



TUGAS AKHIR - KI141502

**RANCANG BANGUN *WEB SERVICE* ENTERPRISE
RESOURCE PLANNING *RETAIL* BERORIENTASIKAN
*SCALABLE AND FLEXIBLE***

Ahmad Zaenal Mustofa
NRP 5113 100 043

Dosen Pembimbing I
Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D

Dosen Pembimbing II
Dwi Sunaryono, S.Kom., M.Kom

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

**RANCANG BANGUN *WEB SERVICE* ENTERPRISE
RESOURCE PLANNING *RETAIL* BERORIENTASIKAN
*SCALABLE AND FLEXIBLE***

**Ahmad Zaenal Mustofa
NRP 5113 100 043**

**Dosen Pembimbing I
Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**Dosen Pembimbing II
Dwi Sunaryono, S.Kom., M.Kom**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

**DESIGN AND IMPLEMENTATION OF *WEB SERVICE*
ENTERPRISE RESOURCE PLANNING *RETAIL* ORIENTED
SCALABLE AND FLEXIBLE**

**Ahmad Zaenal Mustofa
NRP 5113 100 043**

**Supervisor I
Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**Supervisor II
Dwi Sunaryono, S.Kom., M.Kom**

**INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN *WEB SERVICE ENTERPRISE* *RESOURCE PLANNING RETAIL* BERORIENTASIKAN *SCALABLE AND FLEXIBLE*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
AHMAD ZAENAL MUSTOFA
NRP. 5113 100 043

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Prof. Drs. Ec. Ir. RIYANARTO SARI
M.Sc., Ph.D.
NIP: 19590803 198601 1 001

DWI SUNARYONO, S.Kom, M.Kom
NIP: 19720528 199702 1 001



SURABAYA
JULI, 2017

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN *WEB SERVICE* ENTERPRISE RESOURCE PLANNING *RETAIL* BERORIENTASIKAN *SCALABLE AND FLEXIBLE*

Nama : Ahmad Zaenal Mustofa
NRP : 5113100043
Jurusan : Departemen Informatika – FTIf ITS
Dosen Pembimbing I : Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.
Dosen Pembimbing II : Dwi Sunaryono, S.Kom., M.Kom

Abstrak

Enterprise Resource Planning (ERP) adalah sebuah sistem informasi yang digunakan untuk mengintegrasikan dan mengotomasikan proses bisnis yang berhubungan pada aspek operasi, maupun distribusi pada sebuah perusahaan. Namun banyak aplikasi ERP yang tersedia terkadang tidak sesuai dengan proses bisnis perusahaan, mulai dari jumlah modul terlalu banyak, hasil akhir yang tidak sesuai dan lain sebagainya. Untuk itu diperlukan sebuah aplikasi yang dapat dikomposisi sesuai kebutuhan.

Dalam tugas akhir ini, aplikasi ERP retail yang sudah ada kemudian diserviskan agar bisa dikomposisikan dengan menggunakan workflow. Komposisi web service dapat dilakukan dengan cara orkestrasi atau koreografi. Orkestrasi dilakukan untuk komposisi web service dalam lingkup internal perusahaan, sedangkan koreografi digunakan untuk komposisi web service antar perusahaan. Proses komposisi web service dilakukan pada aplikasi ERP retail yang diperuntukan untuk multi-tenant. Sehingga membutuhkan sebuah workflow management untuk multi-tenant. Implementasi workflow management menggunakan platform jBPM yang disesuaikan agar bisa menangani workflow untuk multi-tenant. Integrasi transaksi Business to Business menggunakan metode komposisi web service koreografi.

Hasil implementasi menunjukkan bahwa ERP retail menjadi flexible dan scalable dalam komposisi proses bisnisnya. Komposisi proses bisnis dilakukan dengan cara orkestrasi. Pengguna bisa merubah proses bisnis sesuai dengan kebutuhan tanpa mengganggu proses bisnis perusahaan lain. Ada 31 variasi proses bisnis yang dapat ditangani sistem dengan menerapkan common fragmen proses bisnis. Proses transaksi Business to Business dapat ditangani dengan koreografi web services.

Kata kunci: orkestrasi, koreografi, workflow management, web service, web service composition

DESIGN AND IMPLEMENTATION OF *WEB SERVICE* ENTERPRISE RESOURCE PLANNING *RETAIL* ORIENTED SCALABLE AND FLEXIBLE

Student Name : Ahmad Zaenal Mustofa
NRP : 5113100043
Major : Informatics Department – FTIf ITS
Supervisor I : Prof. Drs.Ec. Ir. Riyanarto Sarno, M.Sc, Ph.D.
Supervisor II : Dwi Sunaryono, S.Kom, M.Kom

Abstract

Enterprise Resource Planning (ERP) is an information system used to integrate and automate business processes that relate to aspects of operations, as well as distribution to a company. Yet many of the available ERP applications sometimes do not fit the company's business processes, ranging from too many modules, unsuitable outcomes and so on. For that required an application that can be composed as needed.

In this final project, an existing retail ERP application is then serviced to be composed by using the workflow. The composition of web service can be done with web service orchestration or choreography. Orchestration is done for the composition of web service within the company's internal, while the choreography is used for the composition of web service s between companies. The process of web service composition is done in retail ERP applications intended for multi-tenant. So it requires a multi-tenant workflow management. Workflow management implementation using a customized jBPM platform to handle multi-tenant workflows. The integration of Busines to Business transactions using the composition method of web service choreography.

The implementation results show that ERP retail is flexible and scalable in the composition of the business process. The

composition of business processes is done by web service orchestration. Users can change business processes according to their needs without disrupting other companies business processes. There are 31 variations of business processes that a system can handle by implementing common fragments of business processes. Business to Business transaction process can be handled with choreography web services.

Keywords: orchestration, Choreography, workflow management, web service, web service composition

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa, karena atas berkat rahmat-Nya, penulis dapat menyelesaikan tugas akhir yang berjudul:

“Rancang Bangun *Web service Enterprise Resource Planning Retail* Berorientasikan *Scalable and Flexible*”

Pengerjaan tugas akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku tugas akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Bapak, Ibu, adik dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Riyanarto Sarno dan bapak Dwi Sunaryono selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan Tugas Akhir ini.
3. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Segenap dosen rumpun mata kuliah Manajemen Informasi.
6. Rekan-rekan pengerjaan satu tim: Andi, Nanda, Ical, Faizal dalam penyelesaian tugas akhir dengan nama “*ERP Retail*”.
7. Teman-teman TC Angkatan 2013 yang selalu mendukung, menyemangati, membantu dan mendengarkan suka duka selama proses pengerjaan tugas akhir.
8. Serta semua pihak yang turut membantu penulis dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan. Dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juli 2017

Ahmad Zaenal Mustofa

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
Abstrak	ix
<i>Abstract</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL	xxv
DAFTAR KODE SUMBER.....	xxvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan.....	3
1.3 Batasan Permasalahan	3
1.4 Tujuan	4
1.5 Manfaat.....	4
1.6 Metodologi	4
1.7 Sistematika Penulisan.....	6
BAB II DASAR TEORI.....	9
2.1 <i>Enterprise Resource Planning (ERP)</i>	9
2.2 <i>Retail</i>	9
2.3 <i>Business Process Model and Notation (BPMN)</i>	9
2.4 <i>Service Oriented Architecture (SOA)</i>	10

2.5	<i>Web service</i>	10
a.	SOAP(<i>Simple Object Access Protocol</i>)	11
b.	REST (<i>Representational State Transfer</i>)	11
2.6	JSON (<i>JavaScript Object Notation</i>)	11
2.7	<i>Enterprise Service Bus</i> (ESB)	11
2.8	<i>Business Process Management System</i> (BPMS)	12
2.9	Orkestrasi <i>Web service</i>	12
2.10	Koreografi <i>Web service</i>	13
2.11	<i>Flexibility</i>	13
2.12	<i>Scalability</i>	13
2.13	Framework Yii	14
2.14	PHP (<i>Hypertext Preprocessor</i>)	14
2.15	Kontribusi Tugas Akhir Sebelumnya	14
	BAB III ANALISIS DAN PERANCANGAN SISTEM	17
3.2	Analisis	17
3.2.1	Cakupan Permasalahan	17
3.2.2	Analisis Fitur Aplikasi ERP <i>Retail</i>	18
3.2.3	Analisis Aplikasi <i>Business Process Management System</i> (BPMS)	21
3.2.4	Analisis <i>Multi-tenancy</i>	22
3.2.5	Deskripsi Umum Sistem	24
3.2.6	Arsitektur	24
3.2.7	Aktor	25
3.2.8	Spesifikasi Kebutuhan Perangkat Lunak <i>Web Client</i>	26
3.2.1	Kasus Penggunaan	27
3.3	Perancangan Sistem	51

3.3.1	Perancangan Implementasi <i>Web service</i> Aplikasi ERP Retail.....	51
3.3.2	Perancangan Komposisi <i>Web service</i>	59
3.3.3	Perancangan Antarmuka	61
	BAB IV IMPLEMENTASI SISTEM.....	67
4.1	Lingkungan Pengembangan Sistem	67
4.2	Implementasi <i>Web service</i> pada Aplikasi ERP Retail.....	67
4.2.1	<i>Service Sales List</i>	67
4.2.2	<i>Service Create Sales</i>	68
4.2.3	<i>Service View Sales</i>	69
4.2.4	<i>Service Purchasing List</i>	70
4.2.5	<i>Service Create Purchasing</i>	71
4.2.6	<i>Service View Purchasing</i>	71
4.2.7	<i>Service Inventory Stock</i>	72
4.2.8	<i>Service Check Availability</i>	73
4.2.9	<i>Service Goods Receipt List</i>	73
4.2.10	<i>Service Create Goods Receipt</i>	74
4.2.11	<i>Service View Goods Receipt</i>	74
4.2.12	<i>Service Check Defect</i>	75
4.2.13	<i>Service Create Return Order</i>	75
4.2.14	<i>Service Goods Issue List</i>	76
4.2.15	<i>Service Create Goods Issue</i>	77
4.2.16	<i>Service View Goods Issue</i>	78
4.2.17	<i>Service Transfer Posting</i>	79
4.2.18	<i>Service Get Discount</i>	80
4.2.19	<i>Service Get Tax</i>	81

4.2.20	<i>Service Purchas Invoice List</i>	82
4.2.21	<i>Service Sales Invoice List</i>	82
4.2.22	<i>Service Detail Invoice</i>	83
4.2.23	<i>Service Pay Invoice</i>	83
4.3	Implementasi Komposisi <i>Web service</i>	85
4.3.1	Orkestrasi.....	85
4.3.2	Koreografi.....	87
4.4	Implementasi Antarmuka Pengguna	88
4.4.1	Antarmuka Melihat Daftar Data <i>Purchasing</i>	88
4.4.2	Antarmuka Menambah Data <i>Purchasing</i>	88
4.4.3	Antarmuka Melihat Data <i>Purchasing</i>	88
4.4.4	Antarmuka Melihat Daftar Data <i>Sales</i>	89
4.4.5	Antarmuka Menambah Data <i>Sales</i>	89
4.4.6	Antarmuka Melihat Data <i>Sales</i>	89
4.4.7	Antarmuka Melihat Daftar Data <i>Stock</i>	89
4.4.8	Antarmuka Melihat Daftar Data <i>Goods Receipt</i>	89
4.4.9	Antarmuka Melihat Data <i>Goods Receipt</i>	89
4.4.10	Antarmuka <i>Receiving Goods</i>	90
4.4.11	Antarmuka Melihat Daftar Data <i>Goods Issue</i>	90
4.4.12	Antarmuka Melihat Data Goods Issue	90
4.4.13	Antarmuka <i>issuing Goods</i>	90
4.4.14	Antarmuka Melihat Daftar Data <i>Invoice</i>	90
4.4.15	Antarmuka Membayar <i>Invoice</i>	90
4.4.16	Antarmuka Melihat Data Laporan <i>Balance Sheet</i>	91
4.4.17	Antarmuka Melihat Data Laporan <i>Cash Flow</i>	91
4.4.18	Antarmuka Melihat Data Laporan <i>Income Statement</i>	91

BAB V PENGUJIAN DAN EVALUASI	93
5.1 Lingkungan Uji Coba	93
5.2 Skenario Pengujian.....	93
5.2.1 Pengujian Orkestrasi.....	93
5.2.2 Pengujian Koreografi.....	97
5.2.3 Pengujian <i>Flexibility</i> Proses Bisnis	99
5.2.4 Pengujian <i>Scalability</i> Proses Bisnis	101
5.2.5 Pengujian <i>Multi-Tenancy</i>	102
5.2.6 Pengujian Fungsionalitas	105
5.3 Evaluasi Pengujian	120
BAB VI KESIMPULAN DAN SARAN.....	123
6.1 Kesimpulan.....	123
6.2 Saran.....	124
DAFTAR PUSTAKA.....	125
LAMPIRAN	127
Gambar	127
Tabel.....	135
BIODATA PENULIS.....	139

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 1. 1 <i>Service Oriented Architecture</i>	10
Gambar 1. 2 <i>Service Orchestration</i>	12
Gambar 1. 3 <i>Service Choreography</i>	13
Gambar 3. 1 Integrasi Modul dalam Pembelian Barang	19
Gambar 3. 2 Integrasi Modul dalam Penjualan Barang..	20
Gambar 3. 3 Integrasi Modul General Ledger dengan yang lainnya	20
Gambar 3. 4 <i>Repository</i> belum ada <i>role</i> akses	23
Gambar 3. 5 Gambaran Umum Aplikasi ERP <i>Retail</i>	24
Gambar 3. 6 Arsitektur ERP <i>Retail</i>	25
Gambar 3. 7 Diagram Kasus Penggunaan	28
Gambar 3. 8 Diagram aktivitas use case UC-001	30
Gambar 3. 9 Diagram aktivitas use case UC-002	32
Gambar 3. 10 Diagram aktivitas use case UC-003	35
Gambar 3. 11 Diagram aktivitas use case UC-004	37
Gambar 3. 12 Diagram aktivitas use case UC-005	39
Gambar 3. 13 Diagram aktivitas use case UC-006	40
Gambar 3. 14 Diagram aktivitas use case UC-007	44
Gambar 3. 15 Diagram aktivitas use case UC-008	46
Gambar 3. 16 Diagram aktivitas use case UC-009	48
Gambar 3. 17 Diagram aktivitas use case UC-010	50
Gambar 3. 18 Orkestrasi <i>Purchase Order</i>	59
Gambar 3. 19 Orkestrasi <i>Sales Order</i>	59
Gambar 3. 20 Orkestrasi <i>Receiving Goods</i>	60
Gambar 3. 21 Orkestrasi <i>Payment Purchase Order</i>	60
Gambar 3. 22 Koreografi <i>retail</i> dengan <i>supplier</i> dalam proses pembelian	61
Gambar 3. 23 Perancangan antarmuka <i>signup</i>	62
Gambar 3. 24 Perancangan antarmuka menambah data submodul	63
Gambar 3. 25 Perancangan antarmuka menyunting data submodul	64

Gambar 3. 26 Perancangan antarmuka detail data submodul.....	65
Gambar 4. 1 Orkestrasi <i>Purchase Order</i>	85
Gambar 4. 2 Orkestrasi <i>Sales Order</i>	85
Gambar 4. 3 Orkestrasi <i>Receiving Goods</i>	86
Gambar 4. 4 Orkestrasi <i>Issuing Goods</i>	87
Gambar 4. 5 Koreografi <i>retail and supplier</i>	87
Gambar 5. 1 Hasil Pengujian Orkestrasi <i>Purchase Order</i>	95
Gambar 5. 2 Hasil Pengujian Orkestrasi <i>Sales Order</i>	96
Gambar 5. 3 Hasil Pengujian Orkestrasi <i>Receiving Goods</i>	96
Gambar 5. 4 Hasil Pengujian Orkestrasi <i>Issuing Goods</i>	97
Gambar 5. 5 Pengujian Pengiriman Pesan <i>Order Request</i>	99
Gambar 5. 6 Pengujian Penerimaan Pesan <i>Order Request</i>	99
Gambar 5. 7 orkestrasi <i>sales older</i>	100
Gambar 5. 8 orkestrasi <i>issuing goods</i>	100
Gambar 5. 9 orkestrasi <i>issuing goods</i>	102
Gambar 5. 10 Editor proses bisnis user1	103
Gambar 5. 11 editor proses bisnis user2.....	104
Gambar A.1 Antarmuka Daftar Data <i>Purchasing</i>	127
Gambar A.2 Antarmuka Menambah Data <i>Purchasing</i> .	127
Gambar A.3 Antarmuka Detail Data <i>Purchasing</i>	128
Gambar A.4 Antarmuka Daftar Data <i>Sales</i>	128
Gambar A.5 Antarmuka Menambah Data <i>Sales</i>	129
Gambar A.6 Antarmuka Detail Data <i>Sales</i>	129
Gambar A.7 Antarmuka Daftar Data <i>Stock</i>	129
Gambar A.8 Antarmuka Daftar Data <i>Goods Receipt</i> ...	130
Gambar A.9 Antarmuka Detail Data <i>Goods Receipt</i>	130
Gambar A.10 Antarmuka Transfer Posting.....	130

Gambar A.11 Antarmuka Daftar Data <i>Goods Receipt</i> .	130
Gambar A.12 Antarmuka Detail Data <i>Goods Receipt</i> ..	131
Gambar A.13 Antarmuka Transfer Posting.....	131
Gambar A.14 Antarmuka Daftar Data <i>Invoice</i>	131
Gambar A.15 Antarmuka Membayar <i>Invoice</i>	132
Gambar A.16 Antarmuka <i>balance Sheet</i>	132
Gambar A.17 Antarmuka <i>cash flow</i>	133
Gambar A.18 Antarmuka <i>Income Statement</i>	134

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Daftar Fitur Aplikasi ERP <i>Retail</i>	18
Tabel 3.2 Perbandingan <i>Open Source</i> BPMS.....	22
Tabel 3.3 Fitur <i>common</i> dan <i>fragment/variasi</i>	23
Tabel 3.4 Daftar Kebutuhan Fungsional Sistem.....	26
Tabel 3.5 Keterangan Kode Kasus Penggunaan.....	27
Tabel 3.6 Spesifikasi Kasus Penggunaan <i>Manage Purchasing</i>	29
Tabel 3.7 Spesifikasi Kasus Penggunaan <i>Manage Sales</i>	31
Tabel 3.8 Spesifikasi Kasus Penggunaan <i>Manage Inventory</i>	33
Tabel 3.9 Spesifikasi Kasus Penggunaan <i>Manage Payment</i>	36
Tabel 3.10 Spesifikasi Kasus Penggunaan <i>Manage Goods Issue</i>	38
Tabel 3.11 Spesifikasi Kasus Penggunaan <i>Manage Purchasing B2B</i>	41
Tabel 3.12 Spesifikasi Kasus Penggunaan <i>Manage Sales B2b</i>	42
Tabel 3.13 Spesifikasi Kasus Penggunaan <i>Manage Service Repository</i>	45
Tabel 3.14 Spesifikasi Kasus Penggunaan <i>Manage Workflow</i>	47
Tabel 3.15 Spesifikasi Kasus Penggunaan <i>Manage General Ledger</i>	49
Tabel 3.16 Daftar Perancangan <i>Web service</i>	51
Tabel 5.1 Tabel Pengujian Orkestrasi	94
Tabel 5.2 Tabel Pengujian Koreografi	97
Tabel 5.3 Tabel Pengujian <i>Flexibility</i>	99
Tabel 5.4 Tabel Pengujian <i>Scalability</i>	101
Tabel 5.5 Tabel Pengujian <i>Workflow Repository</i>	103
Tabel 5.6 Tabel Pengujian Fitur <i>Manage Purchasing</i> ..	105
Tabel 5.7 Tabel Pengujian Fitur <i>Manage Sales</i>	106
Tabel 5.8 Tabel Pengujian Fitur <i>Manage Goods Receipt</i>	108

Tabel 5.9	Tabel Pengujian Fitur <i>Manage Goods Issue</i>	109
Tabel 5.10	Tabel Pengujian Fitur <i>Manage Delivery</i>	111
Tabel 5.11	Tabel Pengujian Fitur <i>Manage Purchasing B2B</i>	112
Tabel 5.12	Tabel Pengujian Fitur <i>Manage Sales B2B</i> ..	114
Tabel 5.13	Tabel Pengujian Fitur <i>Manage Service Repository</i>	117
Tabel 5.14	Tabel Pengujian Fitur <i>Manage Workflow</i> ..	118
Tabel 5.15	Tabel Pengujian Fitur <i>General Ledger</i>	119
Tabel 5.16	Rangkuman Hasil Pengujian	120

DAFTAR KODE SUMBER

Kode Sumber 4. 1 <i>Service Sales List</i>	68
Kode Sumber 4. 2 <i>Service Create Sales</i>	69
Kode Sumber 4. 3 <i>Service View Sales</i>	70
Kode Sumber 4. 4 <i>Service Purchasing List</i>	71
Kode Sumber 4. 5 <i>Service Create Purchasing</i>	71
Kode Sumber 4. 6 <i>Service View Purchasing</i>	72
Kode Sumber 4. 7 <i>Service Inventory Stock</i>	72
Kode Sumber 4. 8 <i>Service Check Availability</i>	73
Kode Sumber 4. 9 <i>Service Goods Receipt List</i>	74
Kode Sumber 4. 10 <i>Service Create Goods Receipt</i>	74
Kode Sumber 4. 11 <i>Service View Goods Receipt</i>	75
Kode Sumber 4. 12 <i>Service View Goods Receipt</i>	75
Kode Sumber 4. 13 <i>Service Create Return Order</i>	76
Kode Sumber 4. 14 <i>Service Goods Issue List</i>	77
Kode Sumber 4. 15 <i>Service Create Goods Receipt</i>	78
Kode Sumber 4. 16 <i>Service View Goods Issue</i>	79
Kode Sumber 4. 17 <i>Service Transfer Posting</i>	80
Kode Sumber 4. 18 <i>Service Get Discount</i>	81
Kode Sumber 4. 19 <i>Service Get Tax</i>	82
Kode Sumber 4. 20 <i>Service Purchase Invoice List</i>	82
Kode Sumber 4. 21 <i>Service Sales Invoice List</i>	83
Kode Sumber 4. 22 <i>Service View Goods Issue</i>	83
Kode Sumber 4. 23 <i>Service Pay Invoice</i>	85

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pengerjaan Tugas Akhir, dan sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

Perusahaan *retail* merupakan salah satu bentuk usaha yang tidak melibatkan proses pengubahan bentuk dari sebuah produk menjadi produk baru. *Retail* berperan untuk mendistribusikan produk atau jasa dari produsen ke konsumen. Dalam perusahaan *retail* terdapat sebuah proses bisnis yang sangat cepat sehingga menuntut perusahaan untuk memiliki sebuah sistem informasi yang terintegrasi untuk melakukan pemrosesan data secara otomatis sehingga perusahaan bisa mendapatkan data yang cepat dan akurat untuk menunjang keputusan bisnis perusahaan.

Enterprise resource planning (ERP) merupakan salah satu solusi sistem informasi terintegrasi dan terpadu yang dapat digunakan oleh sebuah perusahaan dalam menjalankan bisnisnya. Namun banyak aplikasi ERP yang tersedia terkadang tidak sesuai dengan proses bisnis perusahaan, mulai dari jumlah modul terlalu banyak, hasil akhir yang tidak sesuai dan lain sebagainya. Keadaan demikian membuat penggunaan aplikasi menjadi tidak efisien.

Proses bisnis setiap perusahaan yang unik dan kebutuhan perusahaan terhadap aplikasi cenderung berubah-ubah, menuntut aplikasi untuk bisa

beradaptasi terhadap perubahan proses bisnis perusahaan. Variasi proses bisnis setiap perusahaan, perubahan struktur organisasi dan standar operasi merupakan beberapa alasan yang mengubah proses bisnis perusahaan. Dengan kata lain, aplikasi dituntut untuk memiliki sifat yang *flexible* dan *scalable*.

Pada tahun 2015 lalu telah dibuat sebuah aplikasi ERP yang bernama EZERP. Aplikasi ini mempunyai modul yang cukup lengkap dan fokus untuk perusahaan yang bergerak di bidang *manufakturing*. Hal tersebut mengakibatkan penggunaan yang rumit, serta ada modul yang tidak terpakai bagi perusahaan yang bergerak di bagian pengecer dan distributor. Selain itu, aplikasi tersebut belum bisa menyesuaikan proses bisnis sesuai kebutuhan pengguna serta belum bisa dikonfigurasi ulang supaya bisa mengikuti perkembangan proses bisnis perusahaan yang cenderung berubah-ubah dan berkembang. Pada tahun 2011 juga sudah ada perancangan menggunakan metode orkestrasi [1], namun belum mengimplementasikan *choreography* [2], dimana dengan menerapkan metode ini akan memungkinkan aplikasi untuk melakukan transaksi secara B2B (*Business to Business*). Maka dari itu, pada tugas akhir ini akan mengembangkan aplikasi ERP *retail* berbasis *multi-tenancy* yang telah ada menjadi *web service ERP retail* yang *flexible* dan *scalable* guna untuk memudahkan mengikuti kebutuhan pengguna.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mengimplementasikan *web service* pada ERP *retail*?
2. Bagaimana mengimplementasikan orkestrasi pada *web service* ERP *retail*?
3. Bagaimana mengimplementasikan *service* koreografi untuk melakukan transaksi *Business to Business (B2B)* pada ERP *retail*?
4. Bagaimana manajemen proses bisnis untuk *multi-tenancy*?
5. Bagaimana mengkomposisi *web service* ERP *retail* yang *flexible* dan *scalable*?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Aplikasi yang dibangun adalah ERP untuk perusahaan *retail*.
2. *Web service* dibangun dengan library framework Yii 2.0.
3. Aplikasi hanya menggunakan *service* yang terdaftar dalam ESB.
4. Menggunakan jBPM untuk mengkomposisi *web service*.

1.4 Tujuan

Tujuan dalam pembuatan tugas akhir ini adalah sebagai berikut:

1. Mampu mengimplementasikan teknologi *web service* pada aplikasi ERP *retail*.
2. Mampu mengimplementasikan orkestrasi pada *web service* ERP *retail*.
3. Mampu mengimplementasikan *service* koreografi untuk melakukan transaksi *Business to Business (B2B)* pada ERP *retail*.
4. Mampu mememanajemen proses bisnis untuk *multi-tenancy*.
5. Mampu membuat komposisi *web service* ERP *retail* yang *flexible* dan *scalable*.

1.5 Manfaat

Manfaat dari tugas akhir ini adalah menyediakan sistem ERP *retail* yang dapat digunakan oleh banyak pengguna dalam satu waktu dan bisa disesuaikan dengan kebutuhan bisnis pengguna tanpa mempengaruhi proses bisnis perusahaan lain serta dapat terintegrasi dengan sistem lain untuk melakukan transaksi *Business to Business (B2B)*.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1. Studi literatur

Studi literatur yang dilakukan berfokus pada pembuatan *web service* sesuai dengan proses bisnis ERP *retail*. Serta mengintegrasikan *web service* dengan menggunakan metode *orchestration* dan *choreography*.

2. Analisis dan Perancangan Sistem

Proses analisis dalam tugas akhir ini adalah dengan pemecahan masalah yang telah dirumuskan dalam bab rumusan masalah. Kemudian proses disain dihasilkan dari hasil analisa yang berupa konsep-konsep akan diterjemahkan ke dalam desain implementasi yang lebih teknis.

3. Implementasi

Pengembangan aplikasi tugas akhir ini berupa *web service* yang terintegrasi. Sekumpulan *web service* yang bisa dirangkai dengan metode *orchestration* atau *choreography*.

Beberapa hal yang diperlukan dalam implementasi ini adalah:

- a. jBMN BPMS.
- b. Framework PHP Yii 2.0
- c. Java Development Kit 7 (JDK).
- d. Apache *web server*
- e. WSO2 ESB.

4. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan yang menjadi dasar dari pembuatan Tugas Akhir ini.

Bab III Metode Pemecahan Masalah

Bab ini membahas mengenai metode yang digunakan untuk memecahkan masalah yang dipaparkan pada rumusan permasalahan.

Bab IV Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan alur, proses dan perancangan antarmuka pada perangkat lunak.

Bab V Implementasi

Bab ini berisi implementasi dari perancangan perangkat lunak perangkat lunak dan implementasi fitur-fitur penunjang perangkat lunak.

Bab VI Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*) dari perangkat lunak dan pengujian fungsionalitas yang dibuat dengan memperhatikan keluaran yang dihasilkan serta evaluasi terhadap fitur-

fitur perangkat lunak.

Bab VII Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan lembar tambahan untuk mendukung penjelasan yang diuraikan pada bab-bab sebelumnya.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1 *Enterprise Resource Planning (ERP)*

ERP merupakan suatu sistem terintegrasi yang terdiri dari berbagai macam modul proses bisnis perusahaan pada umumnya. Sistem ERP disebut sebagai suatu sistem yang saling terkait dikarenakan penggunaan perangkat lunak sebagai sarana pengintegrasian antarbagian yang diinginkan untuk diintegrasikan. Sistem ERP ini seringkali digunakan perusahaan untuk mengelola data serta meneliti data-data krusial perusahaan yang tersebar di berbagai area bisnisnya, semisal data-data dari bagian keuangan, pemasaran, produksi, sumber daya manusia, manajerial, dan sebagainya. Sistem ERP juga memfasilitasi integrasi aliran data antardepartemen yang terhubung dalam sistem tersebut.

2.2 *Retail*

Eceran atau disebut pula *retail*[3] adalah salah satu cara pemasaran produk meliputi semua aktivitas yang melibatkan penjualan barang secara langsung ke konsumen akhir untuk penggunaan pribadi dan bukan bisnis. Organisasi ataupun seseorang yang menjalankan bisnis ini disebut pula sebagai pengecer. Pada praktiknya pengecer melakukan pembelian barang ataupun produk dalam jumlah besar dari produsen, ataupun pengimport baik secara langsung ataupun melalui grosir, untuk kemudian dijual kembali dalam jumlah kecil.

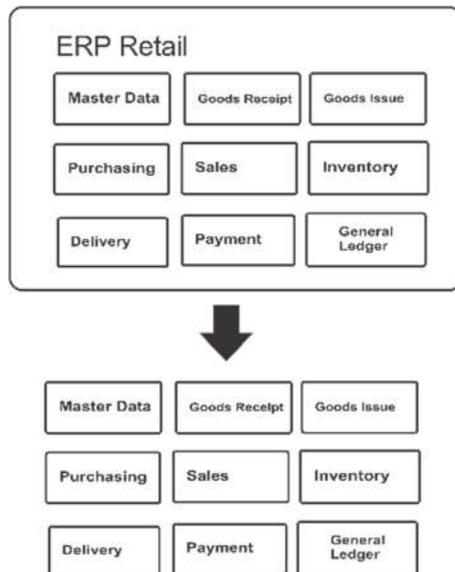
2.3 *Business Process Model and Notation (BPMN)*

BPMN adalah representasi grafis untuk menentukan proses bisnis dalam suatu pemodelan proses Bisnis. Tujuan utama dari BPMN adalah menyediakan suatu notasi standar yang mudah dipahami oleh semua pemangku kepentingan bisnis. Dari analisis bisnis yang menciptakan draft permulaan dari proses-proses sampai dengan pengembang-pengembang teknis yang bertanggung

jawab untuk mengimplementasikan teknologi yang membantu pelaksanaan proses.

2.4 *Service Oriented Architecture (SOA)*

SOA adalah sebuah bentuk teknologi arsitektur yang mengikuti prinsip-prinsip *service-orientation* (berorientasi layanan)[4]. SOA merupakan metode untuk membangun aplikasi yang modular. Dengan dibangun menggunakan SOA, sebuah aplikasi memungkinkan untuk dikomposisi dan berinteraksi dengan aplikasi lain. Gambaran arsitektur SOA dapat dilihat pada Gambar 1.1.



Gambar 1.1 *Service Oriented Architecture*

2.5 *Web service*

Web service adalah sistem pertukaran informasi berbasis XML yang menggunakan internet untuk berinteraksi antar aplikasi[5]. *Web service* dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana

sebuah *database* ditanamkan, dibuat dalam bahasa apa sebuah aplikasi yang mengkonsumsi data, dan di platform apa sebuah data itu dikonsumsi. *Web service* mampu menunjang interoperabilitas. Sehingga *web service* mampu menjadi sebuah jembatan penghubung antara berbagai sistem yang ada. Tipe dari *web service* yaitu SOAP (*Simple Object Access Protocol*) dan RESTful.

a. SOAP (*Simple Object Access Protocol*)

SOAP merupakan sebuah protocol untuk pertukaran informasi pada implementasi *web service*[6]. SOAP menggunakan XML (*Extensible Markup Language*) untuk format pesannya dan menggunakan HTTP (*Hyper Text Transfer Protocol*) untuk transmisinya.

b. REST (*Representational State Transfer*)

REST adalah sebuah arsitektur untuk melakukan pertukaran pesan melalui antarmuka standar seperti antarmuka HTTP[7]. REST tidak berisi lapisan pesan seperti SOAP dan berfokus pada peraturan desain pembuatan *service*. Operasi standar REST yaitu GET, POST, PUT, DELETE, dan HEAD.

2.6 JSON (*JavaScript Object Notation*)

JSON merupakan format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan *degenerate* oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer. Oleh karena itu menjadikan JSON ideal sebagai bahasa pertukaran data.

2.7 *Enterprise Service Bus* (ESB)

Enterprise Service Bus adalah sebuah platform integrasi yang menggabungkan pesan, layanan web, transformasi data dan routing cerdas untuk menghubungkan dan mengkoordinasikan interaksi sejumlah aplikasi yang beragam di suatu perusahaan

dengan integritas transaksional. Dengan kata lain ESB merupakan infrastruktur untuk mengintegrasikan aplikasi dan layanan. ESB memperkuat SOA melalui pengurangan jumlah, ukuran dan kompleksitas interface antara aplikasi dan layanan-layanan.

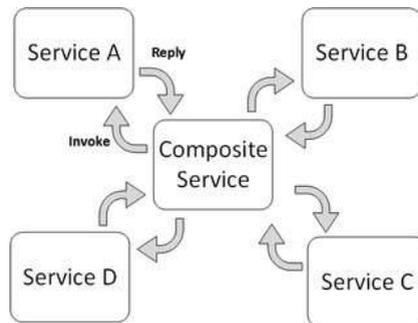
2.8 *Business Process Management System (BPMS)*

BPMS merupakan sebuah *tools* atau perangkat lunak yang digunakan untuk membangaun sebuah aplikasi dengan menggunakan pendekatan *workflow/proses* bisnis. Sehingga pengguna dapat membuat aplikasi hanya dengan menggunakan diagram proses bisnis. Dengan BPMS, tak hanya programer yang bisa mewujudkan aplikasi, bahkan semua orang pun bisa, asalkan memahami apa yang diinginkan dalam aplikasi tersebut dan bisa memodelkan alur prosesnya dalam notasi BPMN.

Dalam tugas akhir ini BPMS digunakan untuk mengkomposisi *web service* . BPMS dipilih karena mendukung notasi BPMN untuk menyusun proses bisnis.

2.9 *Orkestrasi Web service*

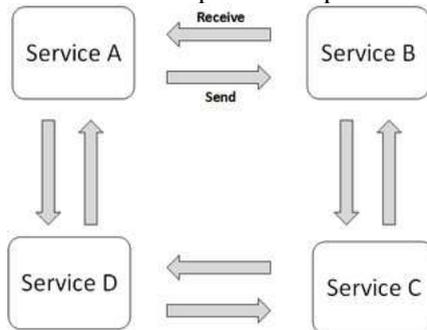
Orchestrtrion[1] yaitu koordinasi atau integrasi beberapa *service* dan mengekspos mereka sebagai *service* tunggal. Orkestrasi digunakan untuk mengkomposisi *web service* dalam lingkup internal perusahaan dan bersifat *private*. Ilustrasi orkestrasi *web service* bisa dilihat pada Gambar 1.1.



Gambar 1. 2 *Service Orchestration*

2.10 Koreografi *Web service*

Koreografi[2] *web service* yaitu mengkomposisi *web service* antar dua atau lebih organisasi atau partisipan. Koreografi *web service* di deskripsikan dengan *Web service Choreography Description Language (WS-CDL)* menyediakan cara untuk membangun komposisi SOA seperti hubungan *peer-to-peer* antara *web service* atau partisipan. Setiap *web service* yang terlibat dalam koreografi mengetahui kapan harus mengeksekusi operasi dan dengan siapa *web service* tersebut akan berinteraksi. Susunan koreografi mempunyai focus pada pertukaran pesan. Semua *web service* pada susunan ini harus tahu proses bisnis apa yang sedang dijalankan, operasi apa yang dieksekusi, pesan apa yang akan dipertukarkan, dan kapan waktunya *web service* harus dijalankan. Ilustrasi koreografi *web service* dapat dilihat pada Gambar 1.3.



Gambar 1. 3 *Service Choreography*

2.11 *Flexibility*

Flexibility adalah kemampuan untuk menerapkan perubahan atau variabilitas dalam proses bisnis, karena perubahan lingkungan. *Flexible* di sini diartikan dapat merubah aliran proses bisnis sesuai kebutuhan serta tidak perlu untuk mematikan sistem ketika melakukan perubahan.

2.12 *Scalability*

Scalability adalah kemampuan suatu sistem, jaringan, atau proses untuk menangani jumlah pekerjaan yang meningkat, atau

potensinya untuk diperbesar untuk mengakomodasi pertumbuhan tersebut. *Scalable* pada sistem ini berarti sistem dapat menyediakan tingkatan level suatu proses bisnis. Misalkan pada awal perusahaan berdiri pada proses bisnis penjualan hanya terdiri dari *task check availability* dan *create order*, ketika perusahaan berkembang kemudian bisa menambahkan *task discount/tax*.

2.13 Framework Yii

Yii adalah kerangka kerja PHP berbasis-komponen dengan performansi tinggi untuk pengembangan aplikasi Web berskala-besar. Ia menyediakan *reusability* maksimum dalam pemrograman Web dan bisa mengakselerasi proses pengembangan secara signifikan.

2.14 PHP (Hypertext Preprocessor)

PHP adalah singkatan dari "PHP: Hypertext Preprocessor", yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari "Personal Home Page Tools". Selanjutnya diganti menjadi FI ("Forms Interpreter"). Sejak versi 3.0, nama bahasa ini diubah menjadi "PHP: Hypertext Preprocessor" dengan singkatannya "PHP". PHP versi terbaru adalah versi ke-5. Berdasarkan survey Netcraft pada bulan Desember 1999, lebih dari sejuta site menggunakan PHP, di antaranya adalah NASA, Mitsubishi, dan RedHat.

2.15 Kontribusi Tugas Akhir Sebelumnya

Tugas Akhir ini merupakan pengembangan riset berkelanjutan dari riset atau penelitian tentang ERP. Pada tahun 2015 lalu telah dibuat sebuah aplikasi ERP yang bernama EZERP. Aplikasi ini mempunyai modul yang cukup lengkap dan fokus untuk perusahaan yang bergerak di bidang *manufakturing*. Hal tersebut mengakibatkan penggunaan

yang rumit, serta ada modul yang tidak terpakai bagi perusahaan yang bergerak di bagian pengecer dan distributor. Selain itu, aplikasi tersebut belum bisa menyesuaikan proses bisnis sesuai kebutuhan pengguna serta belum bisa dikonfigurasi ulang supaya bisa mengikuti perkembangan proses bisnis perusahaan yang cenderung berubah-ubah dan berkembang. Pada tahun 2011 juga sudah ada perancangan menggunakan metode orkestrasi [1], namun belum mengimplementasikan koreografi [2], dimana dengan menerapkan metode ini akan memungkinkan aplikasi untuk melakukan transaksi secara B2B (*Business to Business*). Maka dari itu, pada tugas akhir ini akan mengembangkan aplikasi ERP *retail* berbasis *multi-tenancy* yang telah ada menjadi *web service ERP retail* yang *flexible* dan *scalable* guna untuk memudahkan mengikuti kebutuhan pengguna.

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahapan analisis permasalahan dan perancangan tugas akhir. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir. Analisis kebutuhan mencantumkan kebutuhan-kebutuhan yang diperlukan perangkat lunak. Selanjutnya dibahas mengenai perancangan sistem yang dibuat.

3.2 Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain cakupan permasalahan, analisis fitur aplikasi ERP *retail*, identifikasi dan implementasi *web service*, analisis komposisi *web service*, deskripsi umum sistem, kasus penggunaan sistem, dan kebutuhan perangkat lunak.

3.2.1 Cakupan Permasalahan

Permasalahan utama yang diangkat dalam pembuatan Tugas Akhir ini adalah sebuah aplikasi ERP *retail* yang berbasis web, dimana aplikasi tersebut digunakan untuk multi user atau perusahaan. Seperti yang kita tahu saat ini, setiap perusahaan memiliki proses bisnis yang berbeda dengan perusahaan lain dan ada pula yang terintegrasi dengan aplikasi lain seperti dalam proses *Business to Business* (B2B). Aplikasi ERP *retail* yang ada jika dituntut untuk memenuhi kebutuhan setiap perusahaan maka akan mempengaruhi sistem utama, sedangkan tidak semua perusahaan membutuhkan perubahan tersebut.

Untuk menangani hal tersebut, ERP *retail* perlu dibuat menjadi *web service* agar bisa dikomposisikan sesuai kebutuhan perusahaan. Untuk melakukan komposisi *web service* diperlukan juga sebuah BPMS untuk mengatur *workflow* setiap perusahaan. Dari *workflow* setiap perusahaan dibutuhkan UI/*web client* untuk menjala *workflow* yang telah dibuat.

3.2.2 Analisis Fitur Aplikasi ERP *Retail*

Analisis fitur aplikasi ERP *retail* dilakukan dengan melakukan eksplorasi terhadap aplikasi ERP *retail* yang telah dibuat. Fitur-fitur yang terdapat pada aplikasi ERP *retail* dapat dilihat pada Tabel 3.1.

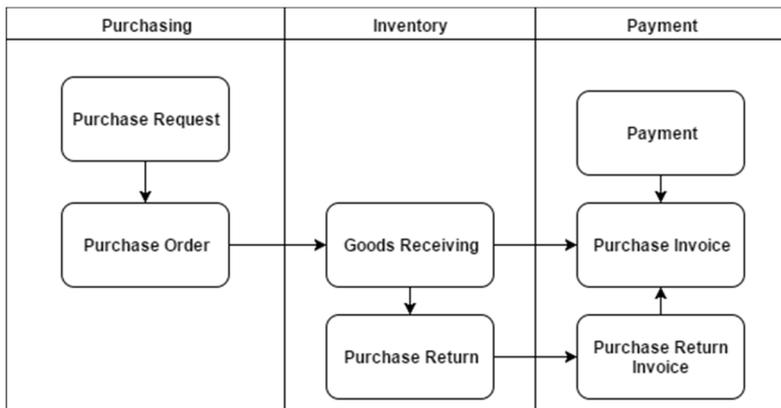
Tabel 3.1 Daftar Fitur Aplikasi ERP *Retail*

No	Jenis Fitur	Keterangan
1.	Pengelolaan <i>purchasing</i>	Meliputi penambahan, dan melihat data <i>purchasing</i> .
2.	Pengelolaan <i>sales</i>	Meliputi penambahan, dan melihat data <i>sales</i> .
3.	Pengelolaan <i>inventory</i>	Meliputi <i>usecase goods receipt, goods issue, dan stock</i> .
4.	Pengelolaan <i>payment</i>	Meliputi <i>usecase purchase payment, dan sales payment</i> .
5.	Pengelolaan <i>product</i>	Meliputi penambahan, pengubahan dan penghapusan data <i>product</i> .
6.	Pengelolaan <i>category</i>	Meliputi penambahan, pengubahan dan penghapusan data <i>category</i> .
7.	Pengelolaan <i>unit of measure (UOM)</i>	Meliputi penambahan, pengubahan dan penghapusan data <i>UOM</i> .
8.	Pengelolaan <i>customer</i>	Meliputi penambahan, pengubahan dan penghapusan data <i>customer</i> .
9.	Pengelolaan <i>supplier</i>	Meliputi penambahan, pengubahan dan penghapusan data <i>supplier</i> .
10.	Pengelolaan modal	Meliputi penambahan, dan pelunasan modal.
11.	Pengelolaan <i>discount</i>	Meliputi penambahan, pengubahan dan penghapusan data <i>discount</i> .
12.	Pengelolaan <i>tax</i>	Meliputi penambahan, pengubahan dan penghapusan data <i>tax</i> .
13.	Pengelolaan <i>delivery</i>	Meliputi penambahan data <i>delivery</i> dan pengiriman.
14.	Pengelolaan <i>general ledger</i>	Meliputi <i>usecase balace sheet, cash flow, dan income statement</i> .

Fitur-fitur pada Tabel 3.1 akan terintegrasi dalam sebuah bisnis proses. Berikut ini adalah beberapa integrasi modul pada beberapa bisnis proses:

a. Integrasi modul dalam pembelian barang

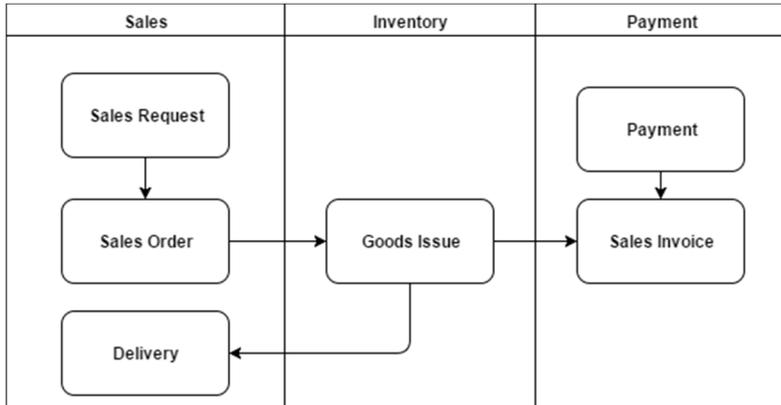
Pada proses pembelian barang, modul yang digunakan adalah modul *purchasing*, *inventory*, dan *payment*. Untuk diagram integrasinya dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Integrasi Modul dalam Pembelian Barang

b. Integrasi modul dalam penjualan barang

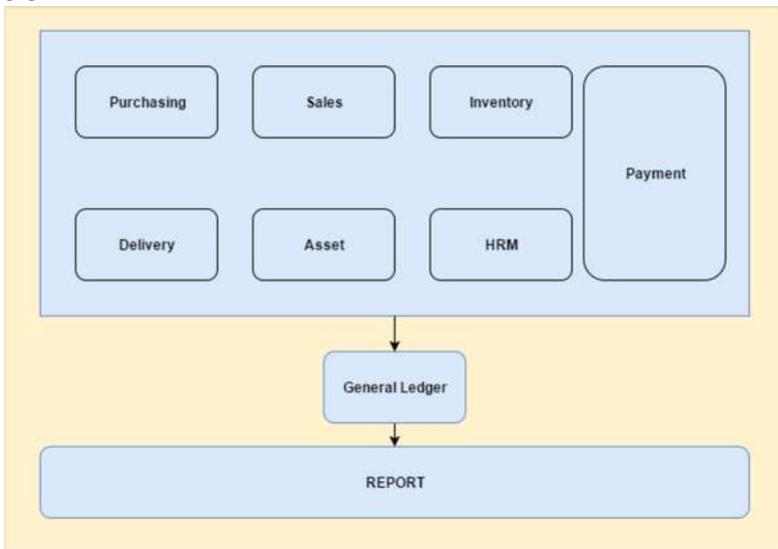
Pada proses penjualan barang, modul yang digunakan yaitu modul *sales*, *inventory*, dan *payment*. Untuk diagram integrasinya dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Integrasi Modul dalam Penjualan Barang

c. Integrasi modul-modul dengan modul *general ledger*

Semua modul-modul yang berhubungan dengan transaksi diintegrasikan dengan modul *general ledger* untuk laporan pembukuan. Untuk diagram integrasinya dapat dilihat pada Gambar 3.3.



Gambar 3. 3 Integrasi Modul General Ledger dengan yang lainnya

3.2.3 Analisis Aplikasi Business Process Management System (BPMS)

BPMS adalah sebuah kakas bantu untuk membangun aplikasi dengan menggunakan diagram dan menitik beratkan pada alur proses seperti yang telah dijelaskan pada subbab 2.6. Pada Tugas Akhir ini BPMS digunakan untuk mengkomposisi *web service* menjadi sebuah aplikasi ERP *retail*. Dengan menggunakan BPMS *web service* ERP *retail* akan mendukung *flexibility* dan *scalability*, karena BPMS memungkinkan untuk implementasi aplikasi menggunakan diagram sehingga bisnis proses bisa diubah sesuai kebutuhan dan setiap pengguna akan memiliki *workflow*/bisnis proses masing-masing tanpa mengganggu proses pengguna lain.

Ada banyak tipe dan vendor yang mengeluarkan perangkat lunak BPMS, beberapa diantaranya adalah Activiti, Bonitasoft, Camunda, jBPM, IBM BPM, Microsoft Biztalk, dsb. Dari berbagai tipe dan vendor BPMS akan dipilih yang memenuhi kriteria kebutuhan dalam tugas akhir ini. Kriteria BPMS yang dibutuhkan dalam tugas akhir ini adalah sebagai berikut:

- a. Bersifat *open source*.
- b. Mendukung komposisi *web service* dengan protocol SOAP.
- c. Hasil komposisi dapat diintegrasikan dengan *web client* (mendukung REST/SOAP).
- d. Mendukung banyak pengguna.
- e. Hasil komposisi dapat dimodifikasi.
- f. Terdapat editor diagram *web-based*.

Untuk menentukan penggunaan BPMS, telah dilakukan survei pada beberapa *open source* BPMS seperti Bonitasoft, Activiti, jBPM, dan Camunda. Hasil survei dapat dilihat pada Tabel 3.2. Dari semua kriteria di atas, jBPM dipilih menjadi BPMS yang digunakan dalam tugas akhir ini. Pada jBPM dukungan multi-tenancy sebenarnya belum secara tegas mendukung *multi-tenancy*. Untuk mendukung kebutuhan multi-tenancy, dalam penelitian ini memanfaatkan fitur

grouping pada jBPM. Untuk mencegah akses *repository* proses bisnis masing-masing perusahaan dilakukan konfigurasi manual pada repository proses bisnis menggunakan tools dukungan “kie-config-cli” yang disediakan terpisah oleh jBPM.

Tabel 3.2 Perbandingan *Open Source* BPMS

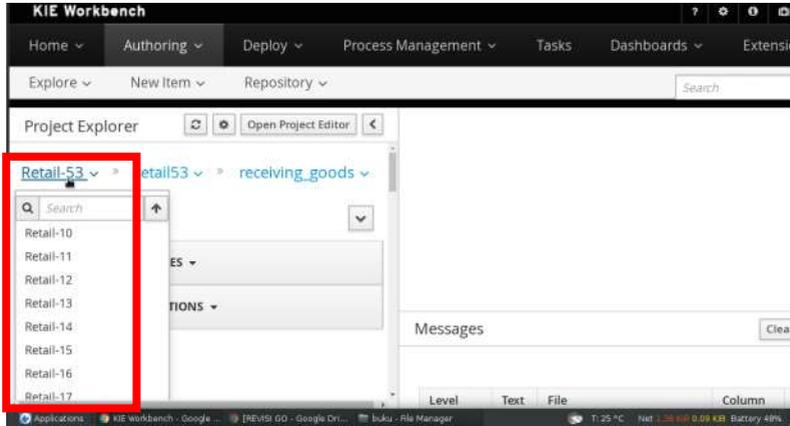
BPMS	BPMN 2.0	Rest	Soap	Multi user	Editor Web-based
Bonitasoft	V	V	?	-	-
Activiti	V	V	?	-	-
jBPM	V	V	V	V	V
Camunda	V	V	V	-	?

3.2.4 Analisis *Multi-tenancy*

Multi-tenancy merupakan kemampuan sistem untuk menangani banyak pengguna. Dalam sistem yang dibangun ini, harus bisa menangani workflow setiap pengguna agar tidak tercampur dengan workflow pengguna lainnya, sehingga dibutuhkan *workflow repository* untuk setiap pengguna. Selain itu sistem juga harus bisa memberikan variasi rangkaian proses bisnis.

3.2.4.1 *Workflow Repository*

Workflow repository berfungsi untuk menampung *workflow* setiap pengguna. Dalam tugas akhir ini *workflow repository* menggunakan fitur dari jBPM. Pada jBPM sudah mendukung banyak pengguna namun untuk *workflow repository* perlu dilakukan konfigurasi tambahan karena belum dapat memberikan *role* akses. Bukti *workflow repository* yang belum bisa memberikan *role* akses dapat dilihat pada Gambar 3.4. Untuk memberikan *role* akses perlu tambahan kakas bantu “kie-config-cli”. Kakas bantu “kie-config-cli” akan memberikan *role* berdasar nama *group* pengguna, sehingga *workflow repository* hanya dapat diakses sesuai *role group* yang telah ditentukan.



Gambar 3. 4 Repository belum ada role akses

3.2.4.2 Variasi Proses Bisnis *Retail*

Proses bisnis adalah sekumpulan aktivitas atau pekerjaan terstruktur yang saling terkait untuk menyelesaikan sebuah proses. Dalam setiap perusahaan memiliki proses bisnis yang bervariasi. Aplikasi ERP yang digunakan pun mengikuti variasi dari proses bisnis perusahaan.

Dalam aplikasi ERP *retail* terdapat beberapa modul yang umum digunakan seperti *purchasing, sales, inventory, payment, delivery, dan general ledger*. Berdasarkan modul yang ada pada aplikasi ERP *retail* yang telah dianalisis pada subbab 3.2.2, dapat dikelompokkan menjadi dua bagian yaitu fitur yang *common* dan fitur variasi/*fragment*. Detail fitur *common* dan variasi/*fragment* dapat dilihat pada Tabel 3.3.

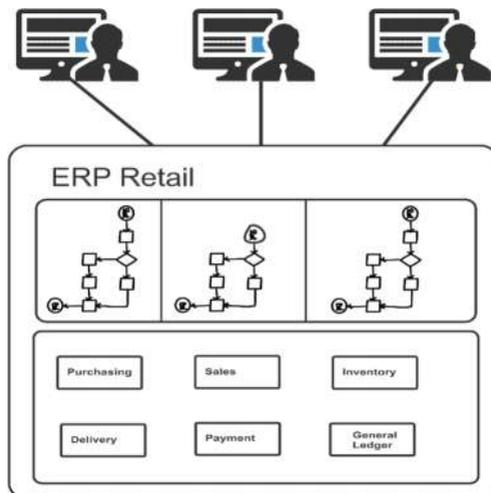
Tabel 3.3 Fitur *common* dan *fragment/variasi*

No	Fitur <i>Common</i>	Fitur <i>Fragment</i>
1.	<i>Purchasing</i>	<i>Delivery</i>
2.	<i>Sales</i>	<i>Discount</i>
3.	<i>Inventory</i>	<i>Tax</i>
4.	<i>Purchase Invoice</i>	<i>Sales Invoice</i>

Dari fitur *common* dan variasi dapat dikombinasikan menjadi banyak variasi proses bisnis. Kombinasi fitur *common* dan variasi menghasilkan 31 variasi proses bisnis. Variasi proses bisnis yang dihasilkan ini yang akan ditangani oleh sistem dalam tugas akhir ini. Variasi proses bisnis yang dihasilkan dari kombinasi fitur dapat dilihat pada Lampiran Tabel A.1.

3.2.5 Deskripsi Umum Sistem

Aplikasi ini dibangun dengan tujuan menciptakan sebuah aplikasi yang melayani banyak *tenant* dan setiap *tenant* mempunyai *workflow* masing-masing serta dapat diubah proses bisnisnya tanpa mempengaruhi proses *tenant* yang lain. Gambaran umum aplikasi dapat dilihat pada Gambar 3.5 .

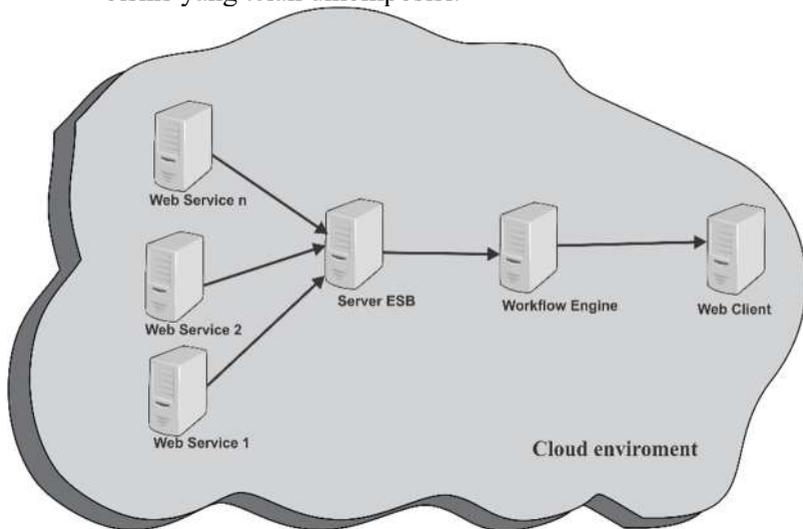


Gambar 3. 5 Gambaran Umum Aplikasi ERP Retail

3.2.6 Arsitektur

Aplikasi yang dibangun berupa aplikasi yang dapat disusun proses bisnisnya sesuai kebutuhan. Arsitektur untuk membangun aplikasi ERP *retail* dapat dilihat pada Gambar 3.6. Terdapat beberapa hal penting dari arsitektur yang diusulkan pada penelitian ini, yaitu:

- a. *Server web service* yang berfungsi untuk menampung *web service* hasil penerapan *web service* pada aplikasi ERP *retail*.
- b. *Server ESB* yang berfungsi untuk mengintegrasikan *web service* agar terpusat pada ESB.
- c. *Server workflow engine* yang berfungsi untuk melakukan komposisi proses bisnis aplikasi.
- d. *Server web client* yang berfungsi untuk menjalankan proses bisnis yang telah dikomposisi.



Gambar 3. 6 *Arsitektur ERP Retail*

3.2.7 Aktor

Pada sistem yang akan dibangun, aktor yang akan menjadi pengguna sistem adalah *administrator* dan pemilik *retail*. *Administratort* adalah yang membuat layanan *web service*. Sedangkan Pemilik *retail* mempunyai hak ases dalam pengelolaan data pada aplikasi ERP *retail*, yang terdiri atas membuat (*create*), melihat (*view*) memperbarui (*update*) dan menghapus (*delete*).

3.2.8 Spesifikasi Kebutuhan Perangkat Lunak *Web Client*

Spesifikasi kebutuhan *web client* mencakup kebutuhan fungsional untuk menjalankan proses bisnis yang telah dirangkai pada BPMS. Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.4

Tabel 3.4 Daftar Kebutuhan Fungsional Sistem

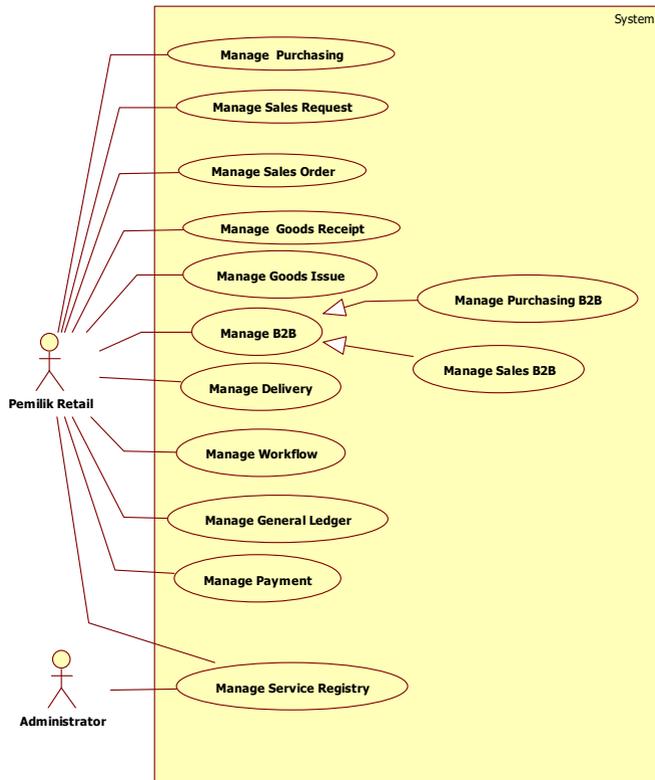
Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	<i>Manage Purchasing</i>	Pengguna dapat mengelola <i>purchasing request</i> yang meliputi proses: <i>create, view</i>
F-002	<i>Manage Sales</i>	Pengguna dapat mengelola <i>sales order</i> yang meliputi proses: <i>create, view</i>
F-003	<i>Manage Goods Receipt</i>	Pengguna dapat mengelola <i>goods receipt</i> yang meliputi proses: <i>view, transfer posting</i>
F-004	<i>Manage Goods Issue</i>	Pengguna dapat mengelola <i>goods issue</i> yang meliputi proses: <i>view, transfer posting</i>
F-005	<i>Manage Delivery</i>	Pengguna dapat mengelola <i>delivery</i> yang meliputi proses: <i>view, shipping</i>
F-006	<i>Manage Purchasing B2B</i>	Pengguna dapat mengelola <i>purchasing B2B</i> yang meliputi proses: <i>create, confirm payment</i>

F-007	<i>Manage Sales B2B</i>	Pengguna dapat mengelola <i>sales B2B</i> yang meliputi proses: <i>confirm order, confirm shipping</i>
F-008	<i>Manage Service Repository</i>	Pengguna dapat mengelola <i>service repository</i> yang meliputi proses: <i>create, update, delete</i>
F-009	<i>Manage Workflow</i>	Pengguna dapat mengelola <i>workflow</i> yang meliputi proses: <i>create, update, delete</i>
F-010	<i>Manage General Ledger</i>	Pengguna dapat mengelola <i>laporan keuangan</i> yang meliputi proses: <i>print, view</i>

Tabel 3.5 Keterangan Kode Kasus Penggunaan

Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	<i>Manage Purchasing</i>
UC-002	<i>Manage Sales Order</i>
UC-003	<i>Manage Inventory</i>
UC-004	<i>Manage Payment</i>
UC-005	<i>Manage Delivery</i>
UC-006	<i>Manage Purchasing B2B</i>

UC-007	<i>Manage Sales B2B</i>
UC-008	<i>Manage Service Repository</i>
UC-009	<i>Manage Workflow</i>
UC-010	<i>Manage General Ledger</i>



Gambar 3. 7 Diagram Kasus Penggunaan

3.2.1 Kasus Penggunaan

Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.7 dan kode kasus penggunaan ada pada tabel 3.5.

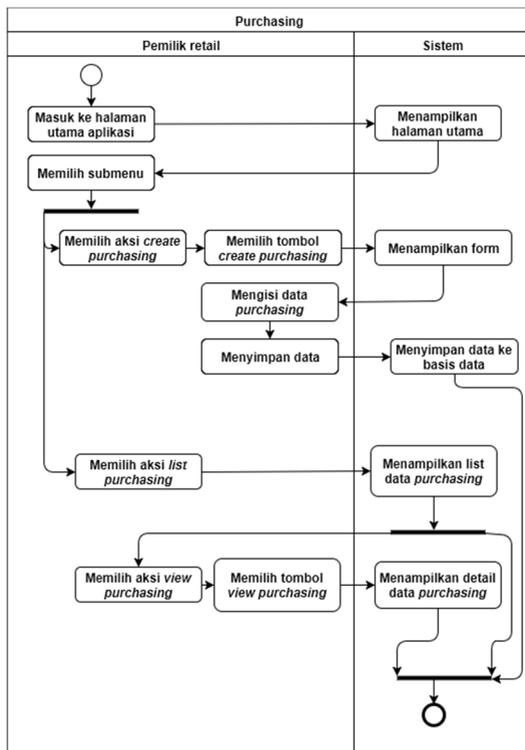
3.2.1.1 UC-001 *Manage Purchasing*

Pada kasus penggunaan ini, pengguna dapat melihat dan melakukan penambahan data *purchasing* dengan mengisi *form* yang tersedia. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.6 dan Gambar 3.8.

Tabel 3.6 Spesifikasi Kasus Penggunaan *Manage Purchasing*

Nama	<i>Manage Purchasing</i>
Nomor	UC-001
Description	Kasus penggunaan ini digunakan untuk melihat, dan menambah data <i>purchasing</i>
Tipe	Fungsional
Aktor	Pemilik <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan halaman dari salah satu submenu <i>purchasing</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih salah satu submenu yang terdapat pada menu <i>purchasing</i> <ol style="list-style-type: none"> A1. Pengguna memilih menu <i>purchasing list</i> A2. Pengguna <i>memilih</i> submenu <i>create purchasing</i> 4. Kasus penggunaan berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pe memilih submenu <i>purchasing list</i> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>purchasing</i> 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1.1. Pengguna memilih kegiatan melihat detail <i>purchasing</i> 3. Berlanjut ke alur normal langkah 4 A2. Pengguna memilih menambah data <i>purchasing</i>

	<ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan data baru 2. Pengguna memasukkan data baru 3. Sistem menyimpan data <i>purchasing</i> 4. Berlanjut ke alur normal langkah 4 <p>A11. Pengguna memilih kegiatan melihat detail <i>purchasing</i>.</p> <p>3.2.1.2 <i>Sistem menampilkan detail data tertentu</i></p> <p>3.2.1.3 <i>Berlanjut ke alur A1 langkah 4</i></p>
--	---



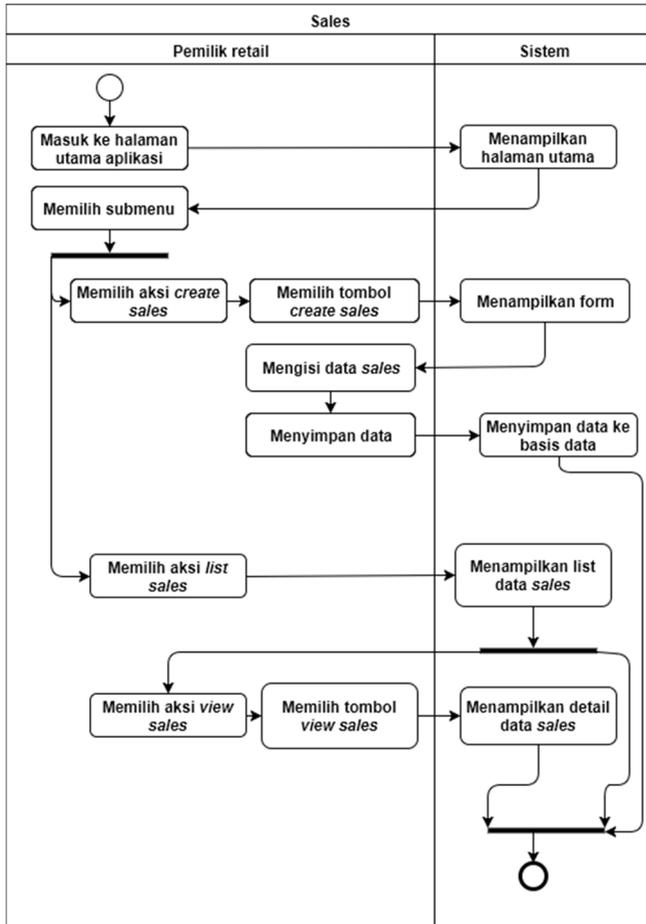
Gambar 3. 8 Diagram aktivitas use case UC-001

3.2.1.4 UC-002 *Manage Sales*

Pada kasus penggunaan ini, pengguna dapat melihat dan melakukan penambahan data *sales* dengan mengisi *form* yang tersedia. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.7 dan Gambar 3.9.

Tabel 3.7 Spesifikasi Kasus Penggunaan *Manage Sales*

Nama	<i>Manage Sales</i>
Nomor	UC-002
Description	Kasus penggunaan ini digunakan untuk melihat, dan menambah data <i>sales</i>
Tipe	Fungsional
Aktor	Pemilik <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan halaman dari salah satu submenu <i>sales</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih salah satu submenu yang terdapat pada menu <i>sales</i> <ol style="list-style-type: none"> A1. Pengguna memilih menu <i>sales list</i> A2. Pengguna <i>memilih</i> submenu <i>create sales</i> 4. Kasus penggunaan berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pe memilih submenu <i>sales list</i> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>sales</i> 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1.1. Pengguna memilih kegiatan melihat detail <i>sales</i> 3. Berlanjut ke alur normal langkah 4 A2. Pengguna memilih menambah data <i>sales</i> <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan data baru 2. Pengguna memasukkan data baru 3. Sistem menyimpan data <i>sales</i> 4. Berlanjut ke alur normal langkah 4 A1.1. Pengguna memilih kegiatan melihat detail <i>sales</i>. <ol style="list-style-type: none"> 1. Sistem menampilkan detail data tertentu 2. Berlanjut ke alur A1 langkah 4



Gambar 3. 9 Diagram aktivitas use case UC-002

3.2.1.5 UC-003 *Manage Inventory*

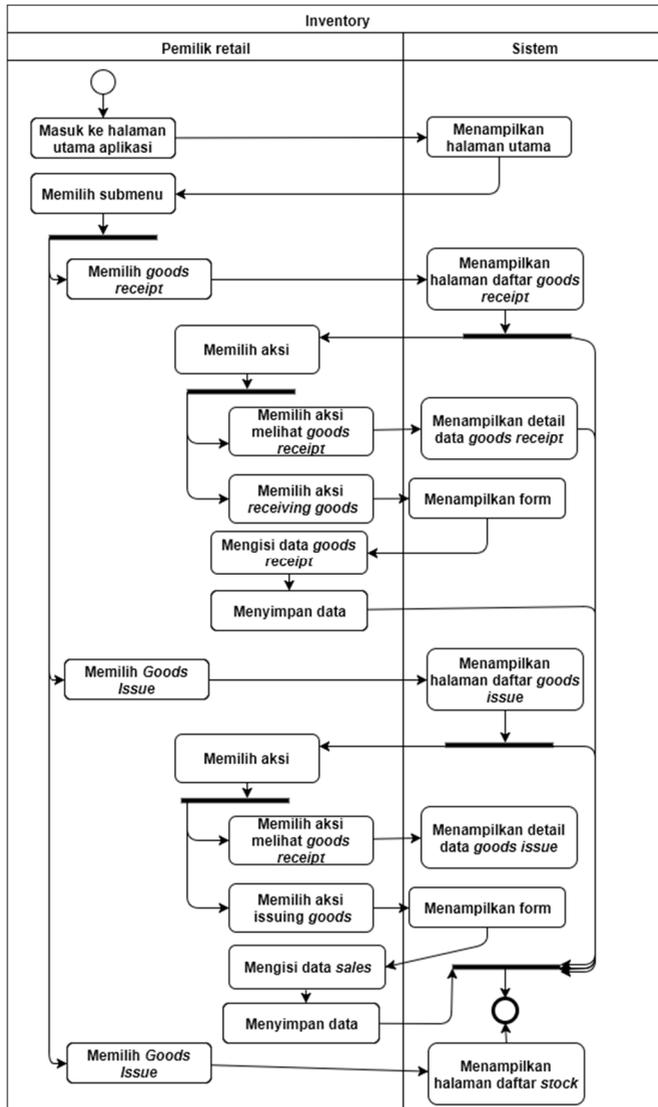
Pada kasus penggunaan ini, pengguna dapat melihat data stok barang, kelola *goods receipt* dan *goods issue*. Spesifikasi

penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.8 dan Gambar 3.10.

Tabel 3.8 Spesifikasi Kasus Penggunaan *Manage Inventory*

Nama	<i>Manage Inventory</i>
Nomor	UC-003
Description	Kasus penggunaan ini digunakan untuk melihat stok barang, mengelola <i>goods receipt</i> , dan <i>goods issue</i>
Tipe	Fungsional
Aktor	Pemilik <i>Retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan halaman dari salah satu submenu <i>Goods Receipt</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih salah satu submenu yang terdapat pada menu <i>inventory</i> <ol style="list-style-type: none"> A1. Pengguna memilih submenu <i>goods receipt</i> A2. Pengguna memilih submenu <i>goods issue</i> A3. Pengguna memilih submenu <i>stock</i> 4. Kasus Penggunaan Berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih submenu <i>goods receipt</i> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>goods receipt</i> 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1.1. Pengguna memilih kegiatan <i>receiving goods</i> A1.2. Pengguna memilih kegiatan melihat detail <i>goods receipt</i> 3. Berlanjut ke alur normal langkah 4 A2. Pengguna memilih submenu <i>goods issue</i> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>goods issue</i> 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A2.1. Pengguna memilih kegiatan <i>issuing goods</i> A2.2. Pengguna memilih kegiatan melihat detail <i>goods issue</i> 3. Berlanjut ke alur normal langkah 4

<p>A3. Pengguna memilih submenu <i>stock</i></p> <ol style="list-style-type: none">1. Sistem menampilkan daftar <i>stock</i>2. Berlanjut ke alur normal langkah 4 <p>A1.1. Pengguna memilih kegiatan <i>receiving goods</i></p> <ol style="list-style-type: none">1. Sistem menampilkan isian data <i>receiving goods</i>2. Pengguna memasukkan data <i>receiving goods</i>3. Sistem menyimpan data4. Berlanjut ke alur A1 langkah 3 <p>A1.2. Pengguna memilih kegiatan melihat detail <i>goods receipt</i></p> <ol style="list-style-type: none">1. Sistem menampilkan isian data <i>receiving goods</i>2. Pengguna memasukkan data <i>receiving goods</i>3. Sistem menyimpan data4. Berlanjut ke alur A1 langkah 3 <p>A2.1. Pengguna memilih kegiatan <i>issuing goods</i></p> <ol style="list-style-type: none">1. Sistem menampilkan isian data <i>issuing goods</i>2. Pengguna memasukkan data <i>issuing goods</i>3. Sistem menyimpan data4. Berlanjut ke alur A2 langkah 3 <p>A2.2. Pengguna memilih kegiatan melihat detail <i>goods issue</i></p> <ol style="list-style-type: none">1. Sistem menampilkan isian data <i>issuing goods</i>2. Pengguna memasukkan data <i>issuing goods</i>3. Sistem menyimpan data4. Berlanjut ke alur A2 langkah 3
--



Gambar 3. 10 Diagram aktivitas use case UC-003

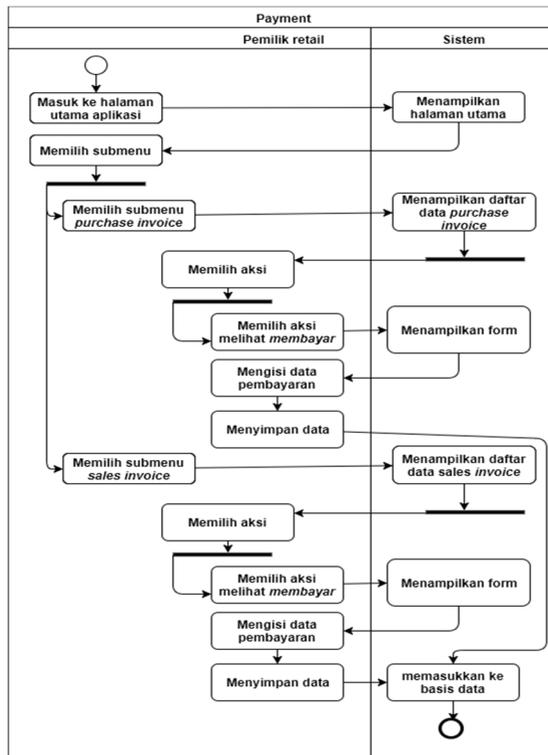
3.2.1.6 UC-004 *Manage Payment*

Pada kasus penggunaan ini, pengguna dapat mengelola *purchase invoice* dan *sales invoice*. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.9 dan Gambar 3.11.

Tabel 3.9 Spesifikasi Kasus Penggunaan *Manage Payment*

Nama	<i>Manage Payment</i>
Nomor	UC-004
Description	Kasus penggunaan ini digunakan untuk mengelola <i>purchase invoice</i> , dan <i>sales invoice</i>
Tipe	Fungsional
Aktor	Pemilik <i>Retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan halaman dari salah satu submenu <i>payment</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih salah satu submenu yang terdapat pada menu <i>inventory</i> <ol style="list-style-type: none"> A1. Pengguna memilih submenu <i>purchase invoice</i> A2. Pengguna memilih submenu <i>sales invoice</i> 4. Kasus Penggunaan Berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih submenu <i>purchase invoice</i> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>purchase invoice</i> 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1.1. Pengguna memilih kegiatan membayar tagihan <i>purchasing</i> 3. Berlanjut ke alur normal langkah 4 A2. Pengguna memilih submenu <i>sales invoice</i> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>goods issue</i> 2. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A2.1. Pengguna memilih kegiatan membayar tagihan <i>sales</i> 3. Berlanjut ke alur normal langkah 4 A1.1. Pengguna memilih kegiatan membayar tagihan <i>purchasing</i>

	<ol style="list-style-type: none"> 1. Sistem menampilkan isian data pembayaran 2. Pengguna memasukkan data pembayaran 3. Sistem menyimpan data 4. Berlanjut ke alur A1 langkah 3 <p>A2.1. Pengguna memilih kegiatan membayar tagihan <i>sales</i></p> <ol style="list-style-type: none"> 1. Sistem menampilkan isian data pembayaran 2. Pengguna memasukkan data pembayaran 3. Sistem menyimpan data 4. Berlanjut ke alur A2 langkah 3
--	--



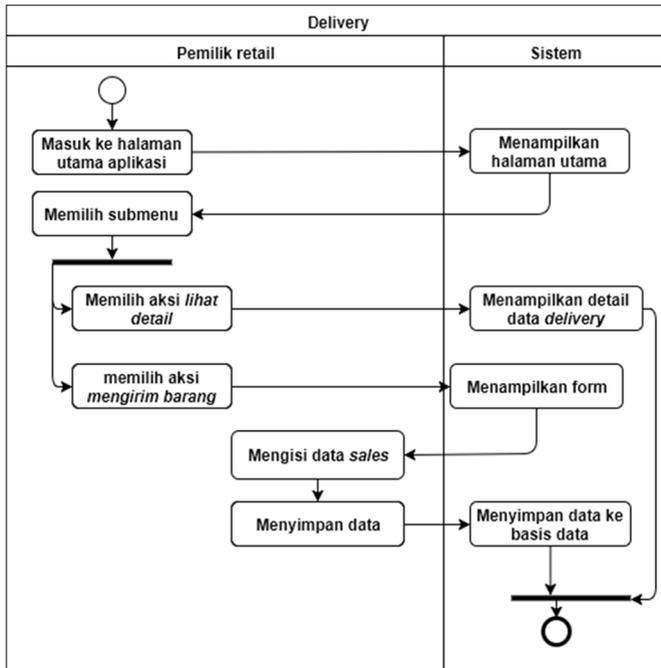
Gambar 3. 11 Diagram aktivitas use case UC-004

3.2.1.7 UC-005 *Manage Delivery*

Pada kasus penggunaan ini, pengguna dapat melihat data *delivery*, dan melakukan pengiriman barang. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.10 dan Gambar 3.12.

Tabel 3.10 Spesifikasi Kasus Penggunaan *Manage Goods Issue*

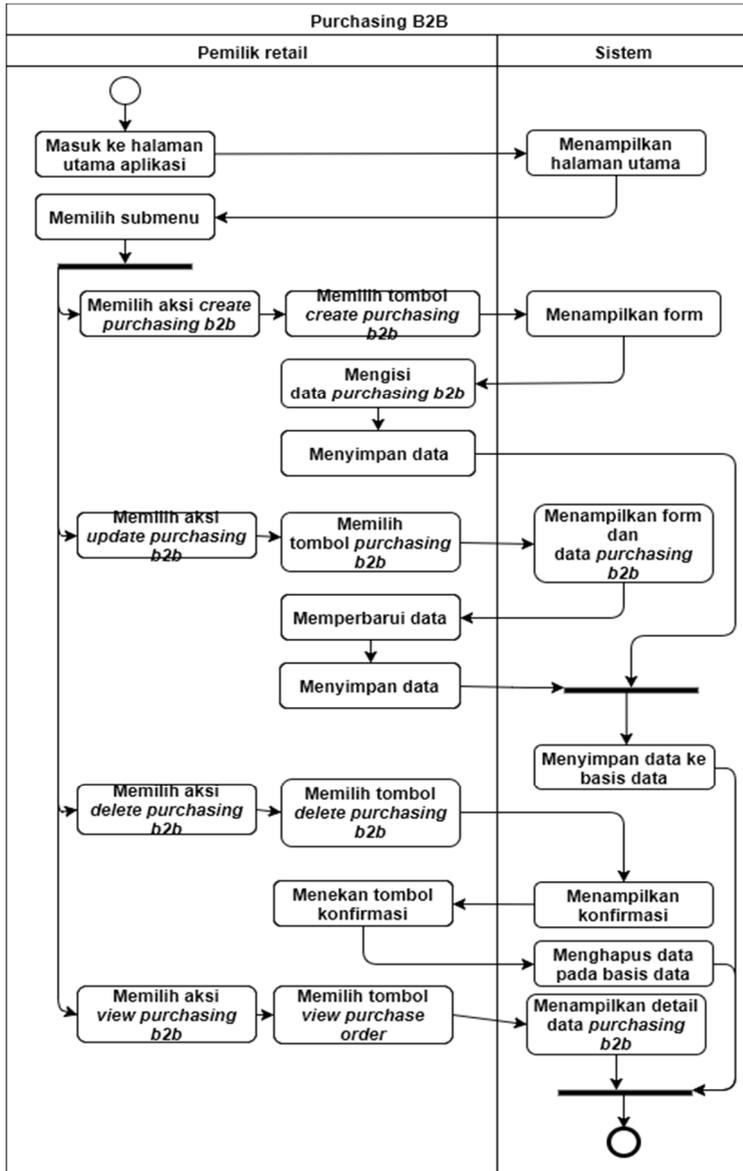
Nama	<i>Manage Delivery</i>
Nomor	UC-005
Description	Kasus penggunaan ini digunakan untuk melihat data <i>delivery</i> dan melakukan pengiriman barang
Tipe	Fungsional
Aktor	Pemilik <i>Retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan halaman utama dari <i>delivery</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pemilik <i>retail</i> membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih menu <i>delivery</i> 4. Sistem menampilkan daftar <i>delivery</i> 5. Pengguna memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. Aktor memilih kegiatan melihat detail data <i>delivery</i> A2. Aktor memilih kegiatan mengirim barang 6. Kasus Penggunaan Berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih melihat data tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan detail data tertentu 2. Berlanjut ke alur normal langkah 6 A1. Pengguna memilih kegiatan mengirim barang <ol style="list-style-type: none"> 1. Sistem menampilkan pesan konfirmasi pengiriman 2. Pengguna mengkonfirmasi pesan 3. Sistem memperbarui status pengiriman 4. Berlanjut ke alur normal langkah 36



Gambar 3. 12 Diagram aktivitas use case UC-005

3.2.1.8 UC-006 Manage Purchasing B2B

Pada kasus penggunaan ini, pengguna dapat melihat data *purchasing B2B*, membuat data baru, menyunting, menghapus, memproses order dan melakukan konfirmasi pembayaran. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.11 dan Gambar 3.13.



Gambar 3. 13 Diagram aktivitas use case UC-006

Tabel 3.11 Spesifikasi Kasus Penggunaan *Manage Purchasing B2B*

Nama	<i>Manage Purchasing B2B</i>
Nomor	UC-006
Description	Kasus penggunaan ini digunakan untuk melihat, menambah, menyunting, memproses order, konfirmasi pembayaran serta menghapus data <i>purchasing B2B</i>
Tipe	Fungsional
Aktor	Pemilik <i>Retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan halaman dari <i>purchasing B2B</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih submenu <i>purchasing</i> yang terdapat pada menu <i>B2B</i> 4. Sistem menampilkan halaman utama pada submenu yang dipilih 5. Pemilik <i>retail</i> akan memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. Pengguna memilih membuat data baru A2. Pengguna memilih menyunting data tertentu A3. Pengguna memilih menghapus data tertentu A4. Pengguna memilih melihat data tertentu A5. Pengguna memilih memproses pemesanan A6. Pengguna memilih konfirmasi pembayaran 6. Kasus Penggunaan Berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih membuat data baru <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan data baru 2. Pengguna memasukkan data baru 3. Sistem menyimpan data baru 4. Berlanjut ke alur normal langkah 6 A2. Pengguna memilih menyunting data tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan isian sunting data tersebut 2. Pengguna menyunting data tersebut 3. Sistem memperbaharui data yang sudah disunting 4. Berlanjut ke alur normal langkah 6 A3. Pengguna memilih menghapus data tertentu

	<ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan data tertentu 2. Pengguna menyetujui penghapusan data 3. Sistem menghapus data tersebut 4. Berlanjut ke alur normal langkah 6 <p>A4. Pengguna memilih melihat data tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan detail data tertentu 2. Berlanjut ke alur normal langkah 6 <p>A5. Pengguna memilih memproses data tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan konfirmasi pemrosesan 2. Pengguna menyetujui pemrosesan 3. Sistem memproses data tersebut 4. Berlanjut ke alur normal langkah 6 <p>A6. Pengguna memilih konfirmasi pembayaran</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan konfirmasi 2. Pengguna menyetujui konfirmasi 3. Sistem memproses konfirmasi 4. Berlanjut ke alur normal langkah 6
--	--

3.2.1.9 UC-007 *Manage Sales B2B*

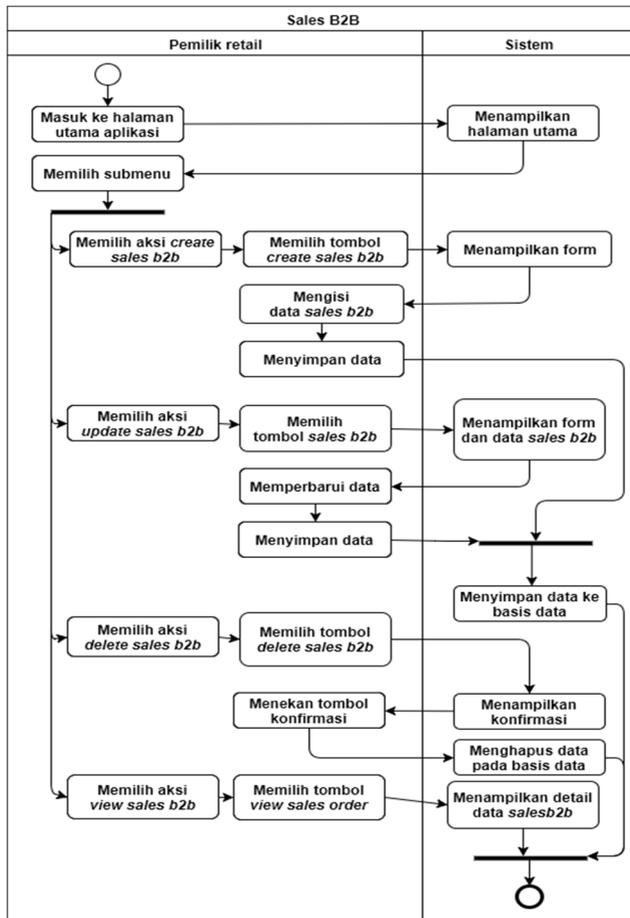
Pada kasus penggunaan ini, pengguna dapat melihat data *sales B2B*, membuat data baru, menyunting, menghapus, konfirmasi pemesanan dan melakukan konfirmasi pengiriman. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.12 dan Gambar 3.14.

Tabel 3.12 Spesifikasi Kasus Penggunaan *Manage Sales B2b*

Nama	<i>Manage Sales B2B</i>
Nomor	UC-007
Description	Kasus penggunaan ini digunakan untuk melihat, menambah, menyunting, konfirmasi order, konfirmasi pengiriman serta menghapus data <i>sales B2B</i>
Tipe	Fungsional
Aktor	Pemilik <i>Retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem

Kondisi Akhir	Sistem menampilkan salah satu submodul dari <i>B2B</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pemilik <i>retail</i> memilih submenu <i>purchasing</i> yang terdapat pada menu <i>B2B</i> 4. Sistem menampilkan halaman utama pada submenu yang dipilih 5. Pemilik <i>retail</i> akan memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. Pengguna memilih membuat data baru A2. Pengguna memilih menyunting data tertentu A3. Pengguna memilih menghapus data tertentu A4. Pengguna memilih melihat data tertentu A5. Pengguna memilih konfirmasi pemesanan A6. Pengguna memilih konfirmasi pengiriman 6. Kasus Penggunaan Berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih membuat data baru <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan data baru 2. Pengguna memasukkan data baru 3. Sistem menyimpan data baru 4. Berlanjut ke alur normal langkah 6 A2. Pengguna memilih menyunting data tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan isian sunting data tersebut 2. Pengguna menyunting data tersebut 3. Sistem memperbaharui data yang sudah disunting 4. Berlanjut ke alur normal langkah 6 A3. Pengguna memilih menghapus data tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan data tertentu 2. Pengguna menyetujui penghapusan data 3. Sistem menghapus data tersebut 4. Berlanjut ke alur normal langkah 6 A4. Pengguna memilih melihat data tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan detail data tertentu 2. Berlanjut ke alur normal langkah 6 A5. Pengguna memilih konfirmasi pemesanan <ol style="list-style-type: none"> 1. Sistem menampilkan pesan konfirmasi 2. Pengguna menyetujui konfirmasi 3. Sistem mengirim pesan konfirmasi 4. Berlanjut ke alur normal langkah 6

	<p>A6. Pengguna memilih konfirmasi pengiriman</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pesan konfirmasi 2. Pengguna menyetujui konfirmasi 3. Sistem memproses konfirmasi pengiriman 4. Berlanjut ke alur normal langkah 6
--	---



Gambar 3. 14 Diagram aktivitas use case UC-007

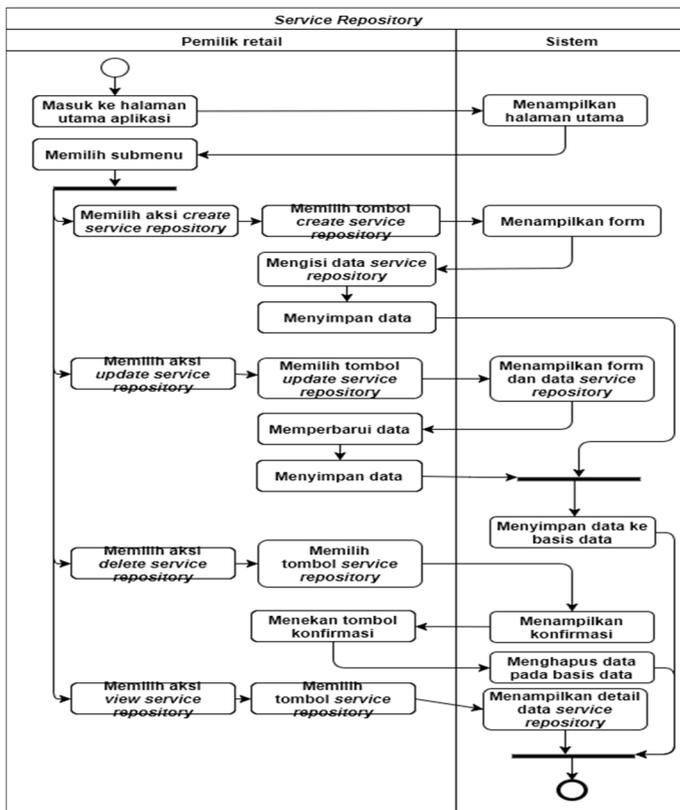
3.2.1.10 UC-008 *Manage Service Repository*

Pada kasus penggunaan ini, pengguna dapat melihat data *sales B2B*, membuat data baru, menyunting, menghapus, konfirmasi pemesanan dan melakukan konfirmasi pengiriman. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.13 dan Gambar 3.15.

Tabel 3.13 Spesifikasi Kasus Penggunaan *Manage Service Repository*

Nama	<i>Manage Service Repository</i>
Nomor	UC-008
Description	Kasus penggunaan ini digunakan untuk melihat, menambah, menyunting, serta menghapus data <i>service repository</i>
Tipe	Fungsional
Aktor	Pemilik <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan halaman dari <i>service repository</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna menu <i>Service Repository</i> 4. Sistem menampilkan halaman utama <i>service repository</i> 5. Pengguna akan memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. pengguna memilih membuat data baru A2. Pengguna memilih menyunting data tertentu A3. Pengguna memilih menghapus data tertentu A4. Pengguna memilih melihat data tertentu 6. Kasus Penggunaan Berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih membuat data baru <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan data baru 2. Pengguna memasukkan data baru 3. Sistem menyimpan data baru 4. Berlanjut ke alur normal langkah 6 A2. Pengguna memilih menyunting data tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan isian sunting data tersebut 2. Pengguna menyunting data tersebut

	<ol style="list-style-type: none"> 3. Sistem memperbaharui data yang sudah disunting 4. Berlanjut ke alur normal langkah 6 <p>A3. Pengguna memilih menghapus data tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan data tertentu 2. Pengguna menyetujui penghapusan data 3. Sistem menghapus data tersebut 4. Berlanjut ke alur normal langkah 6 <p>A4. Pengguna memilih melihat data tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan detail data tertentu 2. Berlanjut ke alur normal langkah 6
--	---



Gambar 3. 15 Diagram aktivitas use case UC-008

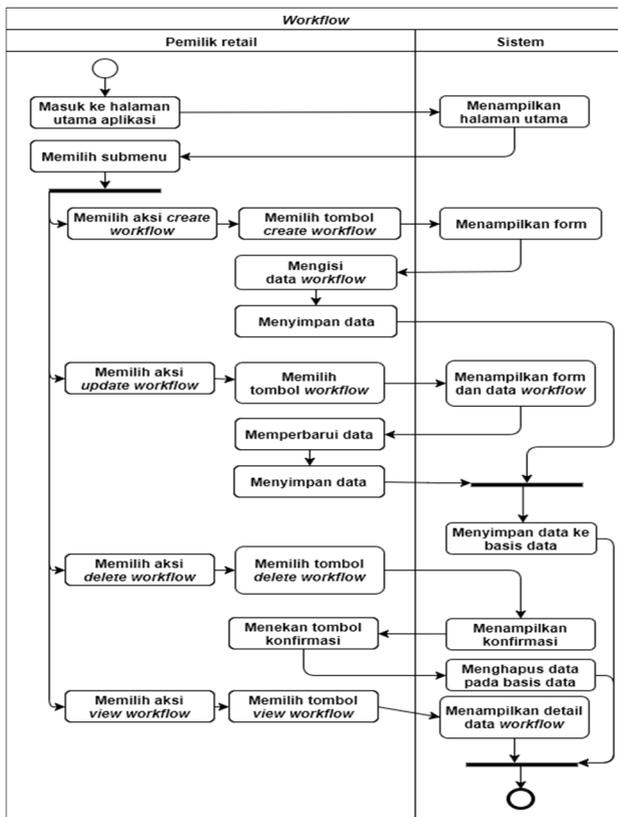
3.2.1.11 UC-009 *Manage Workflow*

Pada kasus penggunaan ini, pengguna dapat melihat data *sales B2B*, membuat data baru, menyunting, menghapus, konfirmasi pemesanan dan melakukan konfirmasi pengiriman. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.14 dan Gambar 3.16.

Tabel 3.14 Spesifikasi Kasus Penggunaan *Manage Workflow*

Nama	<i>Manage Workflow</i>
Nomor	UC-009
Description	Kasus penggunaan ini digunakan untuk melihat, menambah, menyunting, serta menghapus data <i>workflow</i>
Tipe	Fungsional
Aktor	Pemilik <i>retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan halaman dari <i>workflow</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna menu <i>workflow</i> 4. Sistem menampilkan halaman utama <i>workflow</i> 5. Pemilik <i>retail</i> akan memilih kegiatan yang dapat dilakukan <ol style="list-style-type: none"> A1. pengguna memilih membuat data baru A2. Pengguna memilih menyunting data tertentu A3. Pengguna memilih menghapus data tertentu A4. Pengguna memilih melihat data tertentu 6. Kasus Penggunaan Berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih membuat data baru <ol style="list-style-type: none"> 1. Sistem menampilkan isian pembuatan data baru 2. Pengguna memasukkan data baru 3. Sistem menyimpan data baru 4. Berlanjut ke alur normal langkah 6 A2. Pengguna memilih menyunting data tertentu <ol style="list-style-type: none"> 1. Sistem menampilkan isian sunting data tersebut 2. Pengguna menyunting data tersebut

	<ol style="list-style-type: none"> 3. Sistem memperbaharui data yang sudah disunting 4. Berlanjut ke alur normal langkah 6 <p>A3. Pengguna memilih menghapus data tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan pemberitahuan penghapusan data tertentu 2. Pengguna menyetujui penghapusan data 3. Sistem menghapus data tersebut 4. Berlanjut ke alur normal langkah 6 <p>A4. Pengguna memilih melihat data tertentu</p> <ol style="list-style-type: none"> 1. Sistem menampilkan detail data tertentu 2. Berlanjut ke alur normal langkah 6
--	---



Gambar 3. 16 Diagram aktivitas use case UC-009

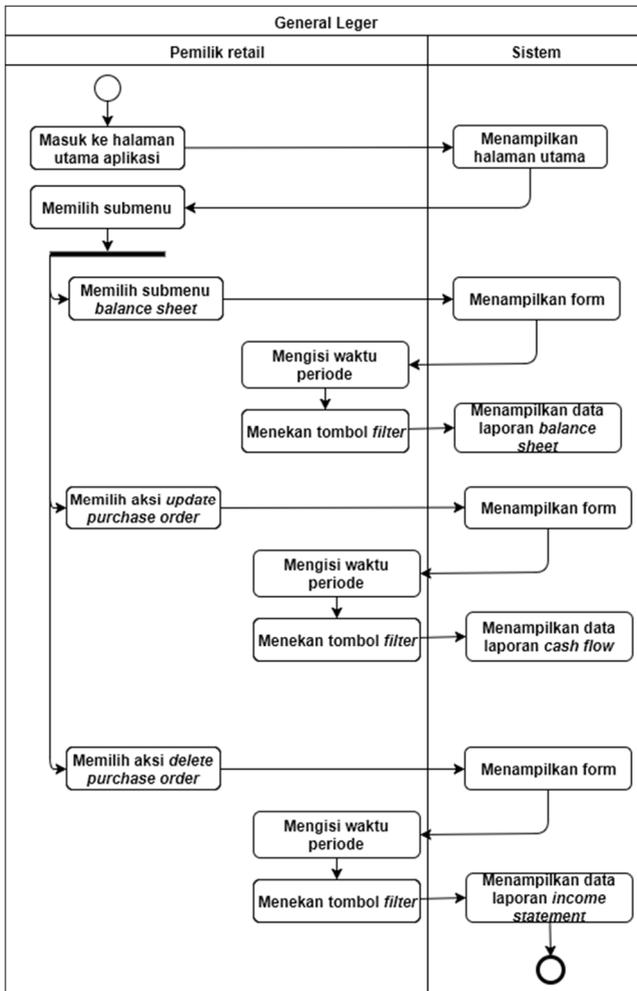
3.2.1.12 UC-010 *Manage General Ledger*

Pada kasus penggunaan ini, pengguna dapat melihat data laporan *balance sheet*, *cash flow*, dan *income statement*. Spesifikasi penggunaan dan diagram aktivitas dapat dilihat pada Tabel 3.15 dan Gambar 3.167

Tabel 3.15 Spesifikasi Kasus Penggunaan *Manage General Ledger*

Nama	<i>Manage General Ledger</i>
Nomor	UC-010
Description	Kasus penggunaan ini digunakan untuk melihat data laporan <i>balance sheet</i> , <i>cash flow</i> dan <i>income statement</i>
Tipe	Fungsional
Aktor	Pemilik <i>Retail</i>
Kondisi Awal	Pengguna sudah masuk ke sistem
Kondisi Akhir	Sistem menampilkan salah satu submenu dari <i>General Ledger</i>
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna membuka halaman utama aplikasi 2. Sistem menampilkan halaman utama aplikasi 3. Pengguna memilih salah satu submenu yang terdapat pada menu <i>general ledger</i> <ol style="list-style-type: none"> A1. Submenu <i>balance sheet</i> A2. Submenu <i>cash flow</i> A3. Submenu <i>income statement</i> 4. Kasus Penggunaan Berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna memilih submenu <i>balance sheet</i> <ol style="list-style-type: none"> 1. Sistem isian periode bulan untuk melihat laporan <i>balance sheet</i> 2. Pengguna menekan tombol <i>filter</i> 3. Sistem menampilkan data laporan 4. Kembali ke alur normal langkah 4 A2. Pengguna memilih submenu <i>cash flow</i> <ol style="list-style-type: none"> 1. Sistem isian periode bulan untuk melihat laporan <i>cash flow</i> 2. Pengguna menekan tombol <i>filter</i> 3. Sistem menampilkan data laporan 4. Kembali ke alur normal langkah 4 A2. Pengguna memilih submenu <i>income statement</i>

	<ol style="list-style-type: none"> 1. Sistem isian periode bulan untuk melihat laporan <i>income statement</i> 2. Pengguna menekan tombol <i>filter</i> 3. Sistem menampilkan data laporan <p>Kembali ke alur normal langkah 4</p>
--	---



Gambar 3. 17 Diagram aktivitas use case UC-010

3.3 Perancangan Sistem

Tahap ini meliputi perancangan implementasi *web service*, komposisi *web service*, dan tampilan antarmuka yang diharapkan dapat memenuhi tujuan dari pengembangan aplikasi ini.

3.3.1 Perancangan Implementasi *Web Service* Aplikasi ERP Retail

Berdasarkan dari hasil analisis fitur aplikasi ERP *retail* yang dijelaskan pada subbab 3.2.2, aplikasi yang dibangun dalam Tugas Akhir ini akan mengimplementasikan beberapa fitur aplikasi ERP *retail* ke dalam *web service* guna untuk bisa melakukan komposisi *web service* dengan menggunakan BPMN. Fitur yang di-*service*-kan hanya yang terlibat dalam integrasi bisnis proses. Hasil identifikasi yang akan diimplementasikan dapat dilihat pada Tabel 3.16.

Tabel 3.16 Daftar Perancangan *Web service*

No	Nama <i>Service</i>	Nama Operasi	Descripsi
1.	<i>Sales</i>	<i>actionIndex</i>	<i>Service</i> ini berfungsi untuk mendapatkan data <i>sales</i> se- <i>suai</i> dengan permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> berupa ID <i>client retail</i> . Sedangkan <i>output</i> -nya berupa string JSON yang berisi data <i>sales</i> sesuai pengguna.
		<i>actionCreate</i>	<i>Service</i> ini berfungsi untuk menambahkan data <i>sales</i> ke basis data dari <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string data <i>sales</i> dan ID <i>retail</i> . Sedangkan <i>output</i> -nya berupa nilai

			boolelan(<i>true/false</i>) yang menandakan proses berhasil atau gagal.
		<i>actionView</i>	<i>Service</i> ini berfungsi untuk melihat detail <i>sales</i> sesuai permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string ID <i>sales</i> . Sedangkan <i>output</i> -nya berupa string JSON data <i>sales</i> .
2.	<i>Purchasing</i>	<i>actionIndex</i>	<i>Service</i> ini berfungsi untuk mendapatkan data <i>purchasing</i> sesuai dengan permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> berupa Id <i>client</i> . Sedangkan <i>output</i> -nya berupa string JSON yang berisi data <i>purchasing</i> sesuai pengguna.
		<i>actionCreate</i>	<i>Service</i> ini berfungsi untuk menambahkan data <i>purchasing</i> ke basis data dari <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string data <i>purchasing</i> dan ID <i>retail</i> . Sedangkan <i>output</i> -nya berupa nilai boolelan(<i>true/false</i>) yang menandakan proses berhasil atau gagal.
		<i>actionView</i>	<i>Service</i> ini berfungsi untuk melihat detail <i>sales</i> sesuai permintaan <i>client</i> . <i>Service</i>

			ini memiliki parameter <i>input</i> string ID <i>sales</i> . Sedangkan <i>output</i> -nya berupa string JSON data <i>sales</i> .
3.	<i>Inventory</i>	<i>actionIndex</i>	<i>Service</i> ini berfungsi untuk mendapatkan daftar data <i>stock</i> sesuai dengan permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> berupa ID <i>client retail</i> . Sedangkan <i>output</i> -nya berupa string JSON yang berisi data <i>stock</i> sesuai pengguna.
		<i>checkAvailability</i>	<i>Service</i> ini berfungsi untuk mendapatkan informasi ketersediaan barang sesuai permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> ID <i>retail</i> dan ID barang. Sedangkan <i>output</i> -nya berupa string yang memberitahukan barang yang diminta tersedia atau tidak.
4.	<i>Goods Receipt</i>	<i>actionIndex</i>	<i>Service</i> ini berfungsi untuk mendapatkan daftar data <i>goods receipt</i> sesuai dengan permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> berupa ID <i>client retail</i> . Sedangkan <i>output</i> -nya berupa string JSON yang berisi data

			<i>goods receipt</i> sesuai pengguna.
		<i>actionCreate</i>	<i>Service</i> ini berfungsi untuk menambahkan data <i>goods receipt</i> ke basis data dari <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string data <i>goods receipt</i> dan ID <i>retail</i> . Sedangkan <i>output</i> -nya berupa nilai boolelan(<i>true/false</i>) yang menandakan proses berhasil atau gagal.
		<i>actionView</i>	<i>Service</i> ini berfungsi untuk melihat detail <i>goods receipt</i> sesuai permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string ID <i>sales</i> . Sedangkan <i>output</i> -nya berupa string JSON data <i>goods receipt</i> .
		<i>actionCheckDefect</i>	<i>Service</i> ini berfungsi untuk mendapatkan informasi adakah barang yang rusak atau tidak. <i>Service</i> ini memiliki parameter <i>input</i> string data penerimaan barang. Sedangkan <i>output</i> -nya berupa string yang menandakan ada atau tidaknya barang yang rusak.
		<i>insertDefect</i>	<i>Service</i> ini berfungsi untuk menambahkan data item <i>defect</i> ke basis data dari

			<i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string data <i>item defect</i> dan ID <i>retail</i> . Sedangkan <i>output</i> -nya berupa nilai boolelan(<i>true/false</i>) yang menandakan proses berhasil atau gagal.
5.	<i>Goods Issue</i>	<i>actionIndex</i>	<i>Service</i> ini berfungsi untuk mendapatkan daftar data <i>goods issue</i> sesuai dengan permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> berupa ID <i>client retail</i> . Sedangkan <i>output</i> -nya berupa string JSON yang berisi data <i>goods issue</i> sesuai pengguna.
		<i>actionCreate</i>	<i>Service</i> ini berfungsi untuk menambahkan data <i>goods issue</i> ke basis data dari <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string data <i>goods issue</i> dan ID <i>retail</i> . Sedangkan <i>output</i> -nya berupa nilai boolelan(<i>true/false</i>) yang menandakan proses berhasil atau gagal.
		<i>actionView</i>	<i>Service</i> ini berfungsi untuk melihat detail <i>goods issue</i> sesuai permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string ID <i>sales</i> . Sedangkan <i>output</i> -

			nya berupa string JSON data <i>goods issue</i> .
6.	<i>Transfer Posting</i>	<i>actionTransfer</i>	<i>Service</i> ini berfungsi untuk melakukan penambahan atau pengurangan data barang pada <i>inventory</i> saat penerimaan barang atau pengeluaran barang. <i>Service</i> ini memiliki parameter input string JSON data <i>goods receipt/goods issue</i> dan ID <i>retail</i> . Sedangkan <i>output</i> -nya berupa boolelan(<i>true/false</i>) yang menandakan proses berhasil atau gagal.
7.	<i>Discount</i>	<i>getDiscount</i>	<i>Service</i> ini berfungsi untuk mengecek adanya <i>discount</i> suatu barang. <i>Service</i> ini memiliki parameter <i>input</i> string JSON data <i>sales</i> . Sedangkan <i>output</i> -nya berupa string JSON.
8.	<i>Tax</i>	<i>getTax</i>	<i>Service</i> ini berfungsi untuk mengecek adanya <i>tax</i> suatu barang. <i>Service</i> ini memiliki parameter <i>input</i> string JSON data <i>sales</i> . Sedangkan <i>output</i> -nya berupa string JSON.
9.	<i>Payment</i>	<i>listPurchaseInvoice</i>	<i>Service</i> ini berfungsi untuk mendapatkan daftar data <i>purchase invoice</i> sesuai dengan permintaan <i>client</i> . <i>Service</i> ini memiliki

			parameter <i>input</i> berupa ID <i>client retail</i> dan type <i>invoice</i> . Sedangkan <i>output</i> -nya berupa string JSON yang berisi data <i>purchase invoice</i> sesuai pengguna.
		<i>listSalesInvoice</i>	<i>Service</i> ini berfungsi untuk mendapatkan daftar data <i>sales invoice</i> sesuai dengan permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> berupa ID <i>client retail</i> dan type <i>invoice</i> . Sedangkan <i>output</i> -nya berupa string JSON yang berisi data <i>sales invoice</i> sesuai pengguna.
		<i>detailInvoice</i>	<i>Service</i> ini berfungsi untuk melihat detail <i>invoice</i> sesuai permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string ID <i>invoice</i> . Sedangkan <i>output</i> -nya berupa string JSON data <i>invoice</i> .
		<i>payInvoice</i>	<i>Service</i> ini berfungsi untuk melakukan pembayaran baik dari <i>purchasing/sales</i> sesuai permintaan <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string JSON data <i>payment</i> . Sedangkan <i>output</i> -nya berupa boolelan(<i>true/false</i>)

			yang menandakan proses berhasil atau gagal.
		<i>actionCreateInvoice</i>	<i>Service</i> ini berfungsi untuk menambahkan data <i>payment</i> ke basis data dari <i>client</i> . <i>Service</i> ini memiliki parameter <i>input</i> string data <i>payment</i> dan ID <i>retail</i> . Sedangkan <i>output</i> -nya berupa nilai boolelan(<i>true/false</i>) yang menandakan proses berhasil atau gagal.

Implementasi *web service* pada modul-modul ERP *retail* dilakukan dengan cara memilih *method* dari suatu kelas pada modul-modul tersebut untuk dijadikan *web service* . Implementasi *web service* menggunakan *library SOAP extention* yang dimiliki oleh framework YII2. *library* ini menyediakan kelas-kelas yang dapat digunakan untuk membuat *web service* , diantaranya *SoapServer* dan *SoapClient*. kelas *SoapServer* dapat digunakan untuk membuat *instance server web service* yang berfungsi untuk menyediakan *web service (service provider)* atau sebagai *listener* yang akan menerima dan merespon permintaan akses terhadap *web service* . sedangkan kelas *SoapClient* digunakan untuk *instance client web service* yang berfungsi untuk menggunakan layanan dari *web service* yang ada atau sebagai *service requestor*.

Aplikasi lain mendapatkan deskripsi layanan yang dimiliki aplikasi ERP *retail* melalui dokumen WSDL. dengan WSDL ini, aplikasi lain dapat mengetahui deskripsi layanan yang dimiliki aplikasi ERP *retail*, yaitu berupa *method* apa saja yang tersedia, parameter masukan, dan parameter keluaran dari *method* tersebut. Dokumen WSDL ini akan di generate secara otomatis oleh SOAP *extention* YII2.

3.3.2 Perancangan Komposisi *Web service*

Perancangan komposisi *web service* dilakukan dengan menggunakan BPMN yang disediakan oleh jBPM. Dalam mengkomposisi *web service* pada penelitian ini menggunakan dua metode yaitu orkestrasi dan koreografi. Orkestrasi digunakan untuk mengkomposisi modul-modul *web service* ERP *retail*, sedangkan koreografi digunakan untuk mengkomposisi *web service* dengan pihak eksternal dalam melakukan transaksi B2B.

3.3.2.1 Orkestrasi

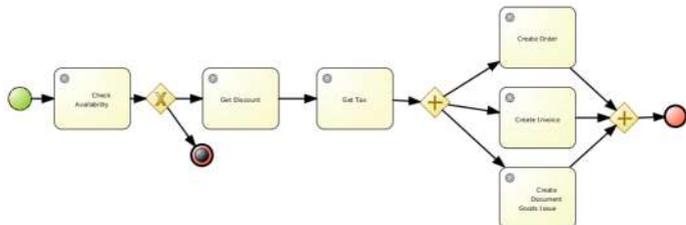
a. Orkestrasi *Purchase Order*



Gambar 3. 18 Orkestrasi *Purchase Order*

Orkestrasi *purchase order* dirancang untuk menggabungkan *service* *Create Purchase Order*, *Create Goods Receipt*, dan *Create Purchase Invoice*. Diagram rangkaian orkestrasi *purchase order* bisa dilihat pada Gambar 3.18.

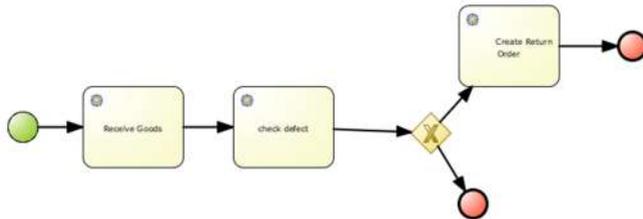
b. Orkestrasi *Sales Order*



Gambar 3. 19 Orkestrasi *Sales Order*

Orkestrasi *sales order* dirancang untuk menggabungkan *service* *Create Sales Order*, *Create Goods Issue*, *Get Discount*, *Get Tax*, *Check Availability* dan *Create Sales Invoice*. Diagram rangkaian orkestrasi *sales order* bisa dilihat pada Gambar 3.19.

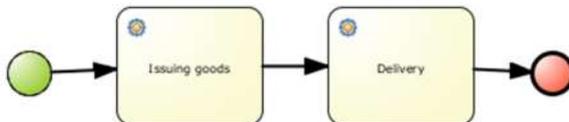
c. Orkestrasi *Receiving Goods*



Gambar 3. 20 Orkestrasi *Receiving Goods*

Orkestrasi *goods receipt* dirancang untuk menggabungkan *service Transfer Posting*, *Check Defect*, dan *Create Defect*. Diagram rangkaian orkestrasi *receiving goods* bisa dilihat pada Gambar 3.20.

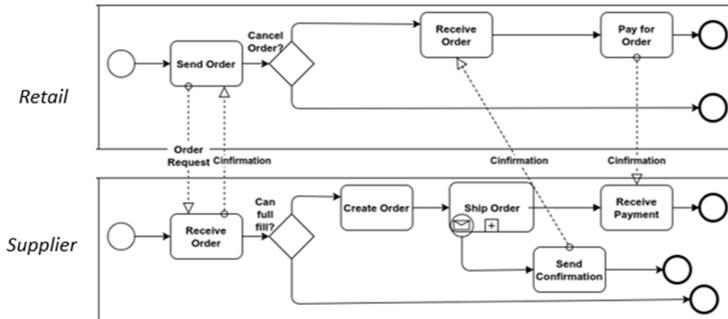
d. Orkestrasi *Issuing Goods*



Gambar 3. 21 Orkestrasi *Payment Purchase Order*

Orkestrasi *goods issue* dirancang untuk menggabungkan *service Transfer Posting*, dan *Create Delivery*. Diagram rangkaian orkestrasi *issuing goods* bisa dilihat pada Gambar 3.21.

3.3.2.2 Koreografi



Gambar 3. 22 Koreografi *retail* dengan *supplier* dalam proses pembelian

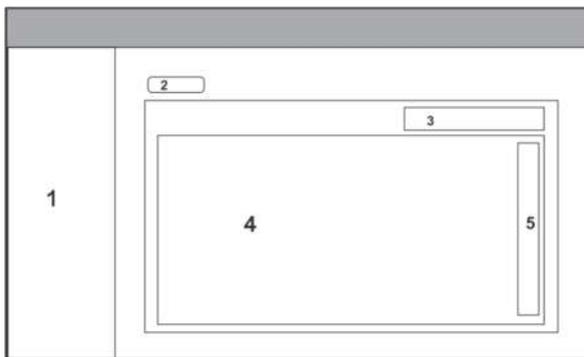
Pada Gambar 3.22 menjelaskan komposisi *web service* antara dua pihak yaitu *retail* dan *supplier* pada proses pembelian barang. Dimulai dari pihak *retail* mengirimkan pesan order barang. Pihak *supplier* mengirimkan pesan konfirmasi. Jika barang sudah dikirim, *supplier* mengirim pesan konfirmasi pengiriman. Selanjutnya pihak *retail* melakukan pembayaran dan melakukan konfirmasi pembayaran kepada *supplier*. Dalam melakukan pengiriman pesan di sini menggunakan *service* dengan protokol SOAP.

3.3.3 Perancangan Antarmuka

Pada sub-bab ini membahas perancangan antarmuka yang akan digunakan dalam aplikasi ERP *retail*. Ilustrasi perancangan ditunjukkan sebagai gambar disertai keterangan objek-objek yang ada di dalamnya.

3.3.3.1 Perancangan Antarmuka Halaman Utama Setiap Submodul

Rancangan antarmuka halaman utama untuk setiap submodul ditunjukkan oleh gambar di bawah ini. Pada halaman ini, pengguna dapat melihat seluruh data yang ada pada submodul tersebut dan melakukan proses pengelolaan baik itu membuat, menyunting, menghapus, melihat dan memproses data, kemudian memasukkan data ke dalam basis data. Rancangan antarmuka halaman utama setiap submodul ditunjukkan pada Gambar 3.23



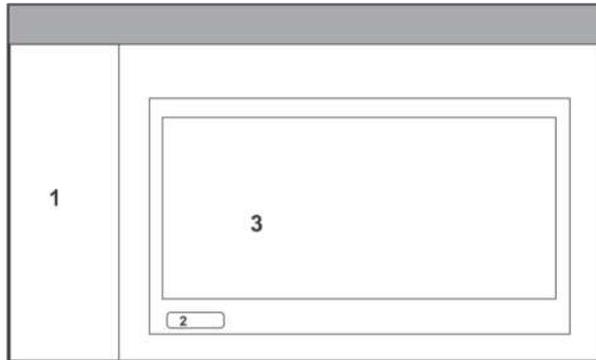
Gambar 3. 23 Perancangan antarmuka *signup*

Berikut penjelasan masing-masing nomor yang ada di dalam masing-masing kotak:

1. Bagian menu modul
2. Tombol untuk *create Data*
3. Form pencarian
4. Halaman yang menampilkan seluruh data pada suatu submodul.
5. Tombol *action (view, edit, delete)*

3.3.3.2 Perancangan Antarmuka Menambah data pada submodul

Antarmuka menambah data submodul dirancang untuk menambah data submodul. Rancangan antarmuka menambah data submodul ditunjukkan pada Gambar 3.24.



Gambar 3. 24 Perancangan antarmuka menambah data submodul

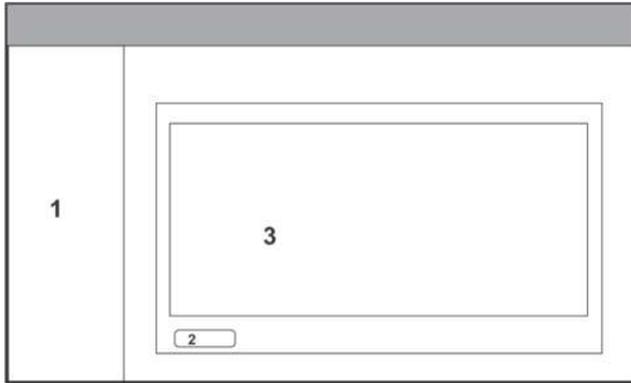
Berikut penjelasan masing-masing nomor yang ada di dalam masing-masing kotak:

1. Bagian menu modul
2. Form isian data
3. Tombol *Create*

Jika tombol ini ditekan maka aplikasi akan menyimpan data yang dimasukkan di dalam *input*. Dan mengarahkan ke halaman utama submodul.

3.3.3.3 Perancangan Antarmuka Menyunting Data Submodul

Antarmuka menyunting data submodul dirancang untuk menyunting data submodul. Rancangan antarmuka menyunting data submodul ditunjukkan pada Gambar 3.25.



Gambar 3. 25 Perancangan antarmuka menyunting data submodul

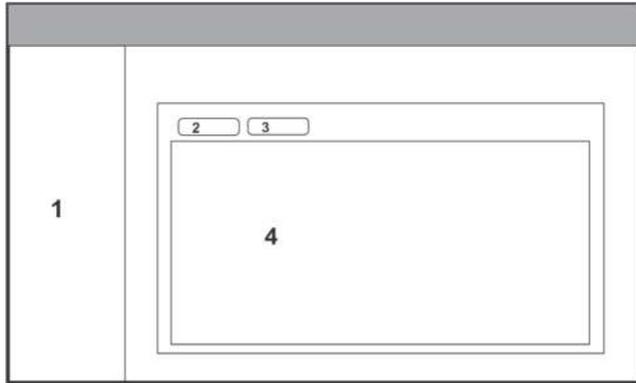
Berikut penjelasan masing-masing nomor yang ada di dalam masing-masing kotak:

1. Bagian menu modul
2. Form isian data
3. Tombol *Create*

Jika tombol ini ditekan maka aplikasi akan menyimpan data yang dimasukkan di dalam *input*. Dan mengarahkan ke halaman utama submodul.

3.3.3.4 Perancangan Antarmuka Detail Data Submodul

Antarmuka detail data submodul dirancang untuk menyunting data submodul. Rancangan antarmuka detail data submodul ditunjukkan pada Gambar 3.26.



Gambar 3. 26 Perancangan antarmuka detail data submodul

Berikut penjelasan masing-masing nomor yang ada di dalam masing-masing kotak:

1. Bagian menu modul
2. Tombol *update*
3. Tombol *delete*
4. Detail data submodul

Jika tombol *update* ditekan maka aplikasi menuju halaman menyunting data submodul. Jika tombol *delete* ditekan maka data submodul akan dihapus dari sistem.

[Halaman ini sengaja dikosongkan]

BAB IV

IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem. Implementasi yang dimaksud di sini meliputi implementasi menservicekan ERP *retail*, mengkompos servis, dan integrasi dengan antarmuka pengguna.

4.1 Lingkungan Pengembangan Sistem

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. Database server MySQL 5.5
2. jBPM untuk BPMS.
3. WSO2 ESB
4. Sublime sebagai teks editor.
5. Mozilla Firefox 46.0.1 sebagai antarmuka untuk pengujian aplikasi klien.
6. 1 PC untuk ESB server menggunakan Ubuntu 14.04.
7. 1 PC untuk BPMS, web servis dan web Client ERP *retail* menggunakan Ubuntu 14.04.

4.2 Implementasi *Web service* pada Aplikasi *ERP Retail*

Pada bagian ini akan dijelaskan mengenai implementasi *web service* yang terdapat pada aplikasi ERP *retail* dari beberapa modul, antara lain: *Master Data (Product, Category, Uom, Customer, Supplier)*, *Purchasing, Sales, Delivery, Inventory, HRM, Asset*, dan *GL*. Secara rinci mengenai implementasi *web service* pada aplikasi ERP *retail* dijabarkan sebagai berikut:

4.2.1 *Service Sales List*

Service ini berfungsi untuk melihat daftar data *sales* yang berhasil ditambahkan. *Service* ini terdapat pada kelas *SalesApiController* fungsi *actionIndex()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.1.

/**

```

* Lists all Sales models.
* @param integer $id
* @return string
* @soap
*/
public function actionIndex($id)
{
    $data = Sales::find()->joinWith('product')-
>joinWith('customer')->where(['sales.id_retail' =>
$id])->asArray()->all();

    return Json::encode($data);
}

```

Kode Sumber 4.1 Service Sales List

4.2.2 Service Create Sales

Service ini berfungsi untuk menambahkan data *purchasing*. *Service* ini terdapat pada kelas *SalesApiController* fungsi *actionCreate()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.2.

```

/**
* @param string $data
* @return boolean
* @soap
*/
public function actionCreate($data2)
{
    $data2=DiscountApiController::getDiscount($data);
    $data=Json::decode($data2);
    if(isset($data['discount'])){
        $discount=$data['discount'];
    }else{
        $discount=0;
    }
    if(isset($data['tax'])){
        $tax=$data['tax'];
    }else{
        $tax=0;
    }
}

```

```

        $total=($data['sales_harga'] *
        $data['sales_jumlah'])*((100+$tax)/100)*(1-
        ($discount/100)) + $data['sales_ongkir'];
        $commando = Yii::$app->db-
>createCommand();
        $result=$commando-
>insert('sales', array(
        'sales_id'=>
        $data['sales_id'],
        'sales_dateStart'=>$data['sales_dateStart'],
        'sales_dateEnd'=>
        $data['sales_dateEnd'],
        'barang_id'=>
        $data['barang_id'],
        'sales_cust'=>
        $data['sales_cust'],
        'sales_jumlah'=>$data['sales_jumlah'],
        'sales_harga'=>$data['sales_harga'],
        'sales_total' =>$total,
        'sales_dist' =>
        $data['sales_dist'],
        'sales_ongkir' =>
        $data['sales_ongkir'],
        'sales_disc' => $discount,
        'sales_tax' =>$tax,
        'id_retail' =>
        $data['id_retail'],
        ))->execute();

        $cek=PaymentController::actionCreateInvoice($data);

        return $result;
    }

```

Kode Sumber 4. 2 Service Create Sales

4.2.3 Service View Sales

Service ini berfungsi untuk melihat data *sales* tertentu. *Service* ini terdapat pada kelas *SalesApiController* fungsi

actionView(). Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.3.

```

/**
 * @param string $data
 * @return boolean
 * @soap
 */
public function actionCreate($data)
{
    $model = new Purchasing();
    $data2=Json::Decode($data);
    $model->attributes=$data2;
    $model->pur_total=($data2['pur_harga'] *
$data2['pur_jumlah']);
    $result=$model->save();
    PaymentController::actionCreateInvoice($data);

    return $result;
}

```

Kode Sumber 4.3 Service View Sales

4.2.4 Service Purchasing List

Service ini berfungsi untuk melihat daftar data *purchasing* yang berhasil ditambahkan. *Service* ini terdapat pada kelas *PurchasingApiController* fungsi *actionIndex()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.4.

```

/**
 * @param string $data
 * @return boolean
 * @soap
 */
public function actionCreate($data)
{
    $model = new Purchasing();
    $data2=Json::Decode($data);
    $model->attributes=$data2;
    $model->pur_total=($data2['pur_harga'] *
$data2['pur_jumlah']);
    $result=$model->save();
    PaymentController::actionCreateInvoice($data);

    return $result;
}

```

```
}

```

Kode Sumber 4. 4 Service Purchasing List

4.2.5 Service Create Purchasing

Service ini berfungsi untuk menambahkan data *purchasing*. *Service* ini terdapat pada kelas *PurchasingApiController* fungsi *actionCreate()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.5.

```
/**
 * @param string $data
 * @return boolean
 * @soap
 */
public function actionCreate($data)
{
    $model = new Purchasing();
    $data2=Json::Decode($data);
    $model->attributes=$data2;
    $model->pur_total=($data2['pur_harga'] *
    $data2['pur_jumlah']);
    $result=$model->save();
    PaymentController::actionCreateInvoice($data);

    return $result;
}

```

Kode Sumber 4. 5 Service Create Purchasing

4.2.6 Service View Purchasing

Service ini berfungsi untuk melihat data *sales* tertentu. *Service* ini terdapat pada kelas *PurchasingApiController* fungsi *actionView()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.3.

```
/**
 * @param string $data
 * @return boolean
 * @soap
 */
public function actionCreate($data)
{
    $model = new Purchasing();

```

```

        $data2=Json::Decode($data);
        $model->attributes=$data2;
        $model->pur_total=($data2['pur_harga'] *
$data2['pur_jumlah']);
        $result=$model->save();
        PaymentController::actionCreateInvoice($data);

        return $result;
    }

```

Kode Sumber 4. 6 Service View Purchasing

4.2.7 Service Inventory Stock

Service ini berfungsi untuk melihat daftar data *stock item* yang berhasil ditambahkan. *Service* ini terdapat pada kelas *InventoryApiController* fungsi *actionIndex()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.7.

```

/**
 * @param integer $id
 * @return string
 * @soap
 */
public function actionIndex($id)
{
    $data = Yii::$app->db
        ->createCommand("SELECT SUM(jumlah) AS jumlah,
id_inv,id_rec,product.`uom_id`,product.`uom_id`,uom
.`uom_name`,category.`category_name`,inventory.`id_
product`,product.`product_name`,inventory.`id_retail`
FROM inventory
INNER JOIN product ON
product.`id_product`=inventory.`id_product`
INNER JOIN uom ON
product.`uom_id`=uom.`id_uom`
INNER JOIN category ON
product.`category_id`=category.`id_category`
WHERE inventory.`id_retail`=".$id."
GROUP BY
inventory.`id_product`,inventory.`id_retail`")
        ->queryAll();
    return Json::Encode($data);
}

```

Kode Sumber 4. 7 Service Inventory Stock

4.2.8 *Service Check Availability*

Service ini berfungsi untuk mengecek ketersediaan *stock item* dalam *inventory*. *Service* ini terdapat pada kelas *InventoryApiController* fungsi *checkAvailability()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.8.

```
/**
 * check stock product
 * If stock available return string array data input
 and status available: 1. if not will return string
 with value 0
 * @param string $data
 * @return string $result
 * @soap
 */
public function checkAvailability($data){
    $data2=Json::decode($data);
    $hasil= Inventory::checkStock($data2);

    if($hasil >= $data2['sales_jumlah']){

        return 1;
    }else{
        return 0;
    }
}
```

Kode Sumber 4. 8 *Service Check Availability*

4.2.9 *Service Goods Receipt List*

Service ini berfungsi untuk melihat daftar data *goods receipt* yang berhasil ditambahkan. *Service* ini terdapat pada kelas *GoodsReceiveApiController* fungsi *actionIndex()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.9.

```
/**
 * Lists all GoodsReceive data.
 * @param integer $id
 * @return string
 * @soap
 */
public function actionIndex($id)
{
```

```

    $data = GoodsReceive::find()-
    >joinWith('receiveDetails')->joinWith('supplier')-
    >where(['goods_receive.id_retail' => $id])-
    >asArray()->all();

    return Json::encode($data);
}

```

Kode Sumber 4. 9 Service Goods Receipt List

4.2.10 Service Create Goods Receipt

Service ini berfungsi untuk menambahkan data *goods receipt*. *Service* ini terdapat pada kelas *GoodsReceiveApiController* fungsi *actionCreate()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.10.

```

/**
 * @param string $data
 * @return boolean
 * @soap
 */
public function actionCreate($data)
{
    $model = new Purchasing();
    $data2=Json::Decode($data);
    $model->attributes=$data2;
    $model->pur_total=($data2['pur_harga'] *
    $data2['pur_jumlah']);
    $result=$model->save();
    PaymentController::actionCreateInvoice($data);

    return $result;
}

```

Kode Sumber 4. 10 Service Create Goods Receipt

4.2.11 Service View Goods Receipt

Service ini berfungsi untuk melihat data *goods receipt* tertentu. *Service* ini terdapat pada kelas *GoodsReceiveApiController* fungsi *actionView()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.11.

```

/**
 * Displays a single GoodsReceive data.
 * @param string $id

```

```

* @return string
* @soap
*/
public function actionView($id)
{
    $data=GoodsReceive::find()-
>joinWith('receiveDetails')->joinWith('supplier')-
>where(['id_rec' => $id])->asArray()->all();
    return Json::Encode($data);
}

```

Kode Sumber 4. 11 Service View Goods Receipt

4.2.12 Service Check Defect

Service ini berfungsi untuk mengecek jika ada barang yang rusak saat penerimaan. *Service* ini terdapat pada kelas *GoodsReceiveApiController* fungsi *actionView()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.12.

```

/**
 * Displays a single GoodsReceive data.
 * @param string $id
 * @return string
 * @soap
 */
public function actionView($id)
{
    $data=GoodsReceive::find()-
>joinWith('receiveDetails')->joinWith('supplier')-
>where(['id_rec' => $id])->asArray()->all();
    return Json::Encode($data);
}

```

Kode Sumber 4. 12 Service View Goods Receipt

4.2.13 Service Create Return Order

Service ini berfungsi untuk menambahkan data *return order*. *Service* ini terdapat pada kelas *GoodsReceiveApiController* fungsi *insertDefect()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.13.

```

/**
 * Deletes an existing GoodsReceive model.
 * If deletion is successful, the browser will
be redirected to the 'index' page.
 * @param string $data

```

```

* @return mixed
* @soap
*/
public function insertDefect($data)
{
    $data2=Json::Decode($data);
    $data = array(
        'po_header_id' =>
$data2['pur_id'],
        'return_type' => 1,
        'product_id' =>
$data2['barang_id'],
        'unit_price' => '',
        'total_price' =>
$data2['pur_total_harga'],
        'supplier_id' =>
$data2['supplier_id'],
        'id_retail' =>
$data2['id_retail'],
        'return_tax' => 0,
        'total_defect' =>
($data2['jumlah_barang']-$data2['jumlah_receive']),
        'total_amount' =>
$data2['pur_total_harga'],
        'reason_for_return' =>
'rusak', );

        $hasil=
ReturnOrderController::actionCreate2($data);

        return $hasil;
}

```

Kode Sumber 4. 13 Service Create Return Order

4.2.14 Service Goods Issue List

Service ini berfungsi untuk melihat daftar data *goods issue* yang berhasil ditambahkan. *Service* ini terdapat pada kelas *GoodsIssueApiController* fungsi *actionIndex()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.14.

```

/**
 * Lists all GoodsIssue data.
 * @param integer $id

```

```

    * @return string
    * @soap
    */
public function actionIndex($id)
{
    $data = GoodsIssue::find()-
>joinWith('issueDetails')->joinWith('customer')-
>where(['goods_issue.id_retail' => $id])->asArray()-
>all();

    return Json::encode($data);
}

```

Kode Sumber 4.14 Service Goods Issue List

4.2.15 Service Create Goods Issue

Service ini berfungsi untuk menambahkan data *goods issue*. *Service* ini terdapat pada kelas *GoodsIssueApiController* fungsi *actionCreate()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.15.

```

/**
 * Creates a new GoodsIssue model.
 * If creation is successful, the browser will
 be redirected to the 'view' page.
 * @param string $data
 * @return boolean
 * @soap
 */
public function actionCreate($data)
{
    $data2=Json::Decode($data);
    if(isset($data2['discount'])){
        $discount=$data2['discount'];
    }else{
        $discount=0;
    }
    if(isset($data2['tax'])){
        $tax=$data2['tax'];
    }else{
        $tax=0;
    }
}

```

```

        $total=($data2['sales_harga']
$discount/100)) + $data2['sales_ongkir'];
        $issueId='ISSUE'.time();
        $result=Yii::$app->db->createCommand()
            ->insert('goods_issue', [
                'id_issue' => $issueId,
                'so_id' => $data2['sales_id'],
                'issue_date' => date('Y-m-d H:m:s'),
                'customer_id' =>
$customer_id'],
                'so_total_harga' => $total,
                'id_retail' => $data2['id_retail'],
            ]->execute();
        $result=Yii::$app->db->createCommand()
            ->insert('issue_detail', [
                'issue_id' => $issueId,
                'barang_id' => $data2['barang_id'],
                'jumlah_barang' =>
$jumlah_barang'],
                'total_harga' => $total,
                'id_retail' => $data2['id_retail'],
            ]->execute();

        return $result;
    }
}

```

Kode Sumber 4.15 Service Create Goods Receipt

4.2.16 Service View Goods Issue

Service ini berfungsi untuk melihat data *goods issue* tertentu. *Service* ini terdapat pada kelas *GoodsIssueApiController* fungsi *actionView()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.16.

```

/**
 * Lists GoodsReceive models.
 * @param string $id
 * @return string
 * @soap
 */
public function actionView($id)

```

```

    {
        $data=GoodsIssue::find()-
>joinWith('issueDetails')->joinWith('customer')-
>where(['id_issue' => $id])->asArray()->all();
        return Json::Encode($data);
    }

```

Kode Sumber 4. 16 Service View Goods Issue

4.2.17 Service Transfer Posting

Service ini berfungsi untuk menambahkan data *goods receipt*. *Service* ini terdapat pada kelas *TransferPostingApiController* fungsi *actionTransfer()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.17.

```

/**
 * @param string $data
 * @return mixed
 * @soap
 */
public function actionTransfer($data)
{
    $data2=Json::Decode($data);
    if(isset($data2['id_rec'])){
        $result=Yii::$app->db->createCommand()
            ->insert('inventory', [
                'id_rec' => $data2['id_rec'],
                'id_product' => $data2['barang_id'],
                'jumlah' =>
                $data2['jumlah_receive'],
                'id_retail' => $data2['id_retail'],
            ]->execute());
        $rec =
        GoodsReceive::findOne($data2['id_rec']);
        $rec-
        >jumlah_receive=$data2['jumlah_receive'];
        $rec->update();
    }elseif(isset($data2['id_issue'])){
        $need=$data2['jumlah_barang'];

        $ambil=GoodsIssueController::actionAmbilStock($data
2);

```

```

        foreach ($sambil as $isi) {
            if ($isi['jumlah']>=$need) {

GoodsIssueController::actionKurangiStock($isi['id_in'], $need);

                $need=$need-$need;
            }else{

GoodsIssueController::actionKurangiStock($isi['id_in'], $isi['jumlah']);

                $need=$need-$isi['jumlah'];
            }
            if ($need<=0) {

                $result = Yii::$app->db
                    ->createCommand("UPDATE
`issue_detail`      set      jumlah_keluar      =
'$data2[jumlah_barang]'      where      issue_id=
'$data2[issue_id]'      AND      id_retail      =
'$data2[id_retail]'")
                    ->execute();
                break;
            }
        }
    }

    return $result;
}

```

Kode Sumber 4. 17 Service Transfer Posting

4.2.18 Service Get Discount

Service ini berfungsi untuk mengambil data *discount* produk pembelian. *Service* ini terdapat pada kelas *DiscountApiController* fungsi *getDiscount()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.18.

```

/**
 * get Discount to get value of discount base
sum of product
 * input data array from purchasing and return
data purchasing + discount
 * @param string $data

```

```

* @return string $discount
* @soap
*/

public function getDiscount($data){
    $data2=Json::decode($data);
    $disc=Discount::getDiscount($data2);
    if($disc){
        $temp=Json::decode($data);
        $temp['discount']=$disc;
    }else{
        $temp=Json::decode($data);
    }

    $discount= Json::Encode($temp);

    return $discount;
}

```

Kode Sumber 4. 18 Service Get Discount

4.2.19 Service Get Tax

Service ini berfungsi untuk mengambil data *discount* produk pembelian. *Service* ini terdapat pada kelas *TaxApiController* fungsi *getTax()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.19.

```

/**
* get Tax to get value of tax
* input data array from purchasing and return
data with tax
* @param string $data
* @return string $tax
* @soap
*/

public function getTax($data){
    $data2=Json::decode($data);
    $tx=Tax::getTax($data2);
    if($tx){
        $temp=Json::decode($data);
        $temp['tax']=$tx;
    }else{

```

```

        $temp=Json::decode($data);
    }
    $tax= Json::Encode($temp);

    return $tax;
}

```

Kode Sumber 4. 19 Service Get Tax

4.2.20 *Service Purchas Invoice List*

Service ini berfungsi untuk melihat daftar data *purchase invoice* yang berhasil ditambahkan. *Service* ini terdapat pada kelas *PaymentApiController* fungsi *listPurchaseInvoice()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.20.

```

/**
 * Lists all Purchase Invoice.
 * @param integer $id
 * @return string
 * @soap
 */
public function listPurchaseInvoice($id)
{
    $data=PaymentMaster::find()->where('pay_status
= 0 and pay_jenis = 1')->andWhere(['id_retail'=>
$id])->all();
    return Json::Encode($data);
}

```

Kode Sumber 4. 20 Service Purchase Invoice List

4.2.21 *Service Sales Invoice List*

Service ini berfungsi untuk melihat daftar data *purchase invoice* yang berhasil ditambahkan. *Service* ini terdapat pada kelas *PaymentApiController* fungsi *listSalesInvoice()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.21.

```

/**
 * Lists all Purchase Invoice.
 * @param integer $id
 * @return string

```

```

    * @soap
    */
public function listPurchaseInvoice($id)
{
    $data=PaymentMaster::find()->where('pay_status
= 0 and pay_jenis = 2')->andWhere(['id_retail'=>
$id])->all();
    return Json::Encode($data);
}

```

Kode Sumber 4. 21 Service Sales Invoice List

4.2.22 Service Detail Invoice

Service ini berfungsi untuk melihat data *invoice* tertentu. *Service* ini terdapat pada kelas *PaymentApiController* fungsi *detailInvoice()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.22.

```

/**
 * Displays a single Payment model.
 * @param integer $id
 * @return string
 * @soap
 */
public function detailInvoice($id)
{
    $data = PaymentMaster::find()-
>where(['pay_id' => $id])->one();
    return Json::encode($data);
}

```

Kode Sumber 4. 22 Service View Goods Issue

4.2.23 Service Pay Invoice

Service ini berfungsi untuk melakukan pembayaran faktur. *Service* ini terdapat pada kelas *PaymentApiController* fungsi *payInvoice()*. Kode untuk implementasi *service* ini ditunjukkan pada Kode Sumber 4.23.

```

/**
 * Creates a new Payment Invoice from Purchasing
or Sales.
 * @param string $data
 * @return boolean
 * @soap

```

```

*/
public function payInvoice($data)
{
    $data2=Json::Decode($data);
    $model = new Payment();

    $total = Yii::$app->db
        ->createCommand("SELECT
pay_total      FROM      `payment_master`      where
pay_id=".$data2['pay_id'])
        ->queryScalar();
    $model->payment_sisa = $total -
$data2['payment_jumlah'];
    $model->payment_date=
$data2['payment_date'];
    $model-
>payment_jumlah=$data2['payment_jumlah'];
    $model-
>payment_transaksi=$data2['payment_transaksi'];
    $model-
>payment_jenis=$data2['payment_jenis'];
    $model->id_retail=$data2['id_retail'];

    if($model->payment_sisa == 0)
    {
        $update = Yii::$app->db
            ->createCommand("UPDATE
`payment_master` set pay_status = 1, pay_total =
".$model->payment_sisa."      where
pay_id=".$data2['pay_id'])
            ->execute();
    }
    else
    {
        $update = Yii::$app->db
            ->createCommand("UPDATE
`payment_master` set pay_total = ".$model-
>payment_sisa." where pay_id=".$data2['pay_id'])
            ->execute();
    }

    $result=$model->save();
}

```

```

return $result;
}

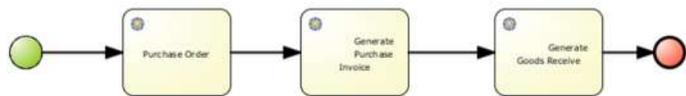
```

Kode Sumber 4. 23 Service Pay Invoice

4.3 Implementasi Komposisi *Web service*

4.3.1 Orkestrasi

a. Orkestrasi *Purchase Order*



Gambar 4. 1 Orkestrasi *Purchase Order*

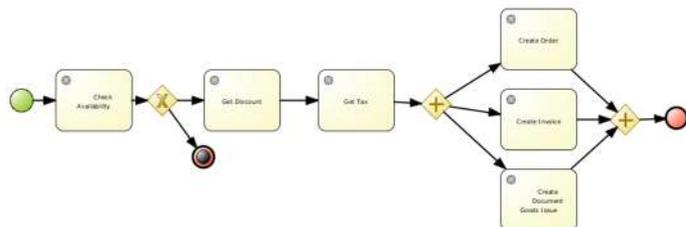
Operasi *service* yang digunakan dalam orkestrasi *purchase order* Gambar 4.1 adalah sebagai berikut:

- a. *Create Purchase Order*
- b. *Create Goods Receipt*
- c. *Create Purchase Invoice*

Berikut lokasi proses orkestrasinya:

http://localhost/jbpm-console/rest/runtime/Retail-A:purchase_order:1.2/process/purchase_order.purchase_order

b. Orkestrasi *Sales Order*



Gambar 4. 2 Orkestrasi *Sales Order*

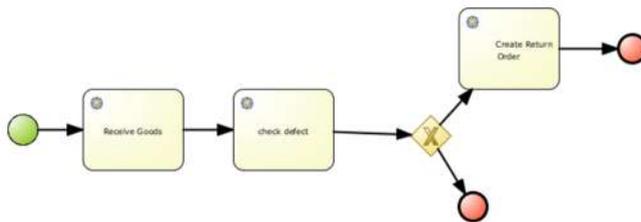
Operasi yang digunakan dalam orkestrasi *purchase order* Gambar 4.2 adalah sebagai berikut:

- a. *Create Sales Order*
- b. *Create Goods Issue*
- c. *Get Discount*
- d. *Get Tax*
- e. *Check Availability*
- f. *Create Sales Invoice*

Berikut lokasi proses orkestrasinya:

http://localhost/jbpm-console/rest/runtime/Retail-A:sales_order:1.2/process/sales_order.sales_order

c. Orkestrasi *Receiving Goods*



Gambar 4. 3 Orkestrasi *Receiving Goods*

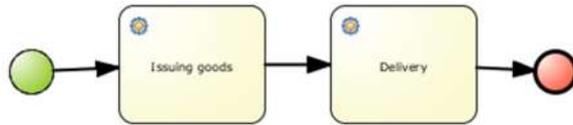
Operasi yang digunakan dalam orkestrasi *receiving goods* Gambar 4.3 adalah sebagai berikut:

- a. *Transfer Posting*
- b. *Check Defect*
- c. *Create Defect*

Berikut lokasi proses orkestrasinya:

http://localhost/jbpm-console/rest/runtime/Retail-A:goods_receive:1.2/process/goods_receive.goods_receive

d. Orkestrasi *Issuing Goods*



Gambar 4. 4 Orkestrasi *Issuing Goods*

Operasi yang digunakan dalam orkestrasi *issuing goods* adalah sebagai berikut:

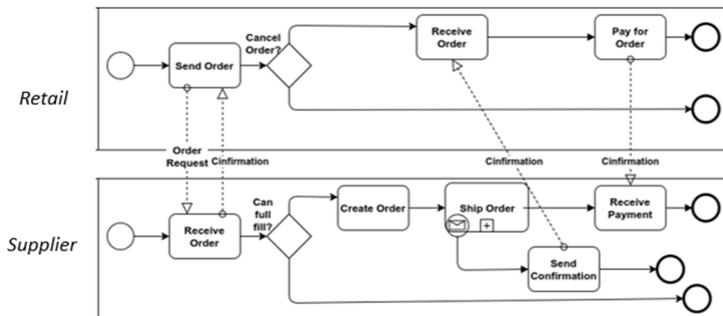
- a. *Transfer Posting*
- b. *Delivery*

Berikut lokasi proses orkestrasinya:

[http://localhost/jbpm-console/rest/runtime/Retail-](http://localhost/jbpm-console/rest/runtime/Retail-A:goods_issue:1.2/process/goods_issue.goods_issue)

[A:goods_issue:1.2/process/goods_issue.goods_issue](http://localhost/jbpm-console/rest/runtime/Retail-A:goods_issue:1.2/process/goods_issue.goods_issue)

4.3.2 Koreografi



Gambar 4. 5 Koreografi *retail and supplier*

Pada implementasi koreografi B2B *seller-buyer* Gambar 4.5 komunikasi *message* menggunakan *service* yang menghubungkan

kedua belah pihak dengan bisnis proses *private* yang dimiliki masing-masing perusahaan.

Berikut *service* yang dibentuk untuk komunikasi pada Gambar 4.5:

Message	Web service
Order Request	http://45.77.36.252:8280/services/B2BPurchaseOrder?wsdl
Confirm Order Request	http://45.77.36.252:8280/services/B2BPurchaseOrderConfirm?wsdl
Shipping Order	http://45.77.36.252:8280/services/ShippingOrder?wsdl
Confirm Payment	http://45.77.36.252:8280/services/B2BConfirmPayment?wsdl

4.4 Implementasi Antarmuka Pengguna

4.4.1 Antarmuka Melihat Daftar Data *Purchasing*

Pada antarmuka ini pengguna dapat melihat daftar *purchasing* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan melihat rincian data *purchasing* tertentu serta membuat *purchasing* baru. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.1.

4.4.2 Antarmuka Menambah Data *Purchasing*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *purchasing* dan menekan tombol “*Create*” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.2.

4.4.3 Antarmuka Melihat Detail Data *Purchasing*

Pada antarmuka ini pengguna dapat melihat detail dari *purchasing*. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.3.

4.4.4 Antarmuka Melihat Daftar Data *Sales*

Pada antarmuka ini pengguna dapat melihat daftar *sales* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan melihat rincian *sales* tertentu serta membuat *sales* baru. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.4.

4.4.5 Antarmuka Menambah Data *Sales*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *sales* dan menekan tombol “*Create*” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.5.

4.4.6 Antarmuka Melihat Data *Sales*

Pada antarmuka ini pengguna dapat melihat detail dari *sales*. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.6.

4.4.7 Antarmuka Melihat Daftar Data *Stock*

Pada antarmuka ini pengguna dapat melihat daftar *stock* pada *inventory* yang terdapat pada sistem. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.7.

4.4.8 Antarmuka Melihat Daftar Data *Goods Receipt*

Pada antarmuka ini pengguna dapat melihat daftar *goods receipt* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan melihat rincian *goods receipt*. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.8.

4.4.9 Antarmuka Melihat Data *Goods Receipt*

Pada antarmuka ini pengguna dapat melihat detail dari *goods receipt*. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.9.

4.4.10 Antarmuka *Receiving Goods*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *receiving goods* dan menekan tombol “*receiving*” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.10.

4.4.11 Antarmuka Melihat Daftar Data *Goods Issue*

Pada antarmuka ini pengguna dapat melihat daftar *goods issue* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan melihat rincian *goods issue*. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.11.

4.4.12 Antarmuka Melihat Data *Goods Issue*

Pada antarmuka ini pengguna dapat melihat detail dari *goods issue*. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.12.

4.4.13 Antarmuka *issuing Goods*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *issuing goods* dan menekan tombol “*issuing*” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.13.

4.4.14 Antarmuka Melihat Daftar Data *Invoice*

Pada antarmuka ini pengguna dapat melihat daftar *Invoice* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan melihat rincian *Invoice*. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.14.

4.4.15 Antarmuka Membayar *Invoice*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *invoice* dan menekan tombol “*Pay*” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.15.

4.4.16 Antarmuka Melihat Data Laporan *Balance Sheet*

Pada antarmuka ini pengguna dapat melihat data laporan *balance sheet* yang terdapat pada sistem dengan cara memasukkan periode waktu pada *form* isian. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.16.

4.4.17 Antarmuka Melihat Data Laporan *Cash Flow*

Pada antarmuka ini pengguna dapat melihat data laporan *cash flow* yang terdapat pada sistem dengan cara memasukkan periode waktu pada *form* isian. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.17.

4.4.18 Antarmuka Melihat Data Laporan *Income Statement*

Pada antarmuka ini pengguna dapat melihat data laporan *income statement* yang terdapat pada sistem dengan cara memasukkan periode waktu pada *form* isian. Tampilan antarmuka ini dapat dilihat pada lampiran Gambar A.18.

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas hasil dan pembahasan pada aplikasi yang dikembangkan. Pada bab ini akan dijelaskan tentang data yang digunakan, hasil yang didapatkan dari penggunaan perangkat lunak dan uji coba yang dilakukan pada perangkat lunak yang telah dikerjakan untuk menguji apakah fungsionalitas aplikasi telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya.

5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi pembuatan sistem pada tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™ i5 CPU @ 2.10GHz
 - b. Memori (RAM): 12 GB
 - c. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak
 - a. Sistem operasi: Ubuntu 14.04
 - b. Web browser: Firefox 48.0

5.2 Skenario Pengujian

Dalam uji coba yang dilakukan dalam tugas akhir ini memiliki beberapa tahapan yang dijelaskan pada subbab ini.

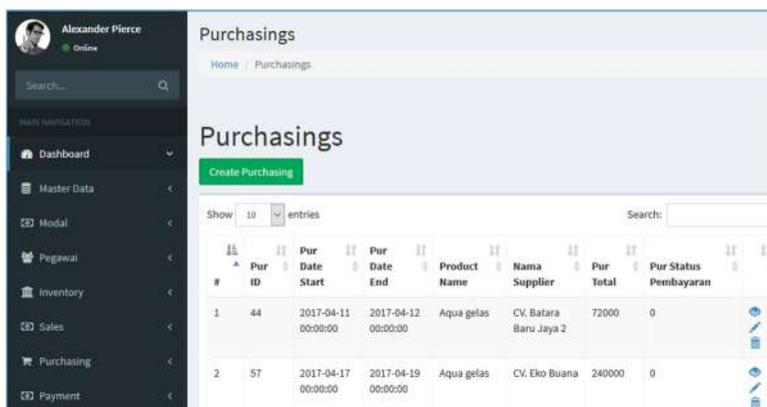
5.2.1 Pengujian Orkestrasi

Pengujian ini dilakukan pada pembuatan rangkaian orkestrasi *purchase order*, *sales order*, *receiving goods*, *issuing goods*. Untuk detail scenario pengujiannya dapat dilihat pada Tabel 5.1 dan hasil uji fungsi bisa dilihat pada Gambar 5.1.

Tabel 5.1 Tabel Pengujian Orkestrasi

ID	UJ-ORC
Referensi Kasus Penggunaan	Orkestrasi
Nama	Pengujian Orkestrasi
Tujuan Pengujian	Menguji kemampuan aplikasi melakukan orkestrasi
Skenario 1	<i>Pengguna melakukan orkestrasi Purchase Order</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman editor jBPM
Data Uji	data pada <i>web service</i> pada <i>service repository</i> :
Langkah Pengujian	Pengguna menyusun orkestrasi pada editor jBPM dan menjalankannya.
Hasil Yang Diharapkan	Orkestrasi proses bisnis <i>Purchase Order</i> Berjalan
Hasil Yang Didapat	Orkestrasi proses bisnis <i>Purchase Order</i> Berjalan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman editor jBPM
Skenario 2	<i>Pengguna melakukan orkestrasi Sales Order</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman editor jBPM
Data Uji	data pada <i>web service</i> pada <i>service repository</i> :
Langkah Pengujian	Pengguna menyusun orkestrasi pada editor jBPM dan menjalankannya.
Hasil Yang Diharapkan	Orkestrasi proses bisnis <i>Sales Order</i> Berjalan
Hasil Yang Didapat	Orkestrasi proses bisnis <i>Sales Order</i> Berjalan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman editor jBPM
Skenario 3	<i>Pengguna melakukan orkestrasi Goods Receipt</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman editor jBPM
Data Uji	data pada <i>web service</i> pada <i>service repository</i> :
Langkah Pengujian	Pengguna menyusun orkestrasi pada editor jBPM dan menjalankannya.

Hasil Yang Diharapkan	Orkestrasi proses bisnis <i>Goods Receipt</i> Berjalan
Hasil Yang Didapat	Orkestrasi proses bisnis <i>Goods Receipt</i> Berjalan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman editor jBPM
Skenario 4	<i>Pengguna melakukan orkestrasi Goods Issue</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman editor jBPM
Data Uji	data pada <i>web service</i> pada <i>service repository</i> :
Langkah Pengujian	Pengguna menyusun orkestrasi pada editor jBPM dan menjalankannya.
Hasil Yang Diharapkan	Orkestrasi proses bisnis <i>Goods Issue</i> Berjalan
Hasil Yang Didapat	Orkestrasi proses bisnis <i>Goods Issue</i> Berjalan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman editor jBPM



The screenshot shows a web application interface for 'Purchasings'. On the left is a dark sidebar with a user profile for 'Alexander Pierce' and a search bar. Below the search bar is a 'MAIN NAVIGATION' menu with items: Dashboard, Master Data, Modal, Pegawai, Inventory, Sales, Purchasing, and Payment. The main content area has a header 'Purchasings' and a sub-header 'Home / Purchasings'. Below this is a 'Create Purchasing' button and a table. The table has a 'Show 10 entries' dropdown and a search box. The table columns are: #, Pur ID, Pur Date Start, Pur Date End, Product Name, Nama Supplier, Pur Total, and Pur Status Pembayaran. There are two rows of data in the table.

#	Pur ID	Pur Date Start	Pur Date End	Product Name	Nama Supplier	Pur Total	Pur Status Pembayaran
1	44	2017-04-11 00:00:00	2017-04-12 00:00:00	Aqua gelas	CV, Batara Baru Jaya 2	72000	0
2	57	2017-04-17 00:00:00	2017-04-19 00:00:00	Aqua gelas	CV, Eko Buana	240000	0

Gambar 5. 1 Hasil Pengujian Orkestrasi *Purchase Order*

Sales

Choose Sales

Show 10 entries

#	Sales ID	Sales Date Start	Sales Date End	Barang ID	Sales Cost
31	32	2017-05-01 00:00:00	2017-05-05 00:00:00	1	1
32	33	2017-05-01 00:00:00	2017-05-30 00:00:00	3	1
33	34	2017-05-01 00:00:00	2017-05-30 00:00:00	1	1
34	35	2017-05-30 00:00:00	2017-05-31 00:00:00	1	1
35	36	2017-05-29 00:00:00	2017-05-31 00:00:00	1	1
36	37	2017-05-29 00:00:00	2017-05-31 00:00:00	1	1
37	38	2017-05-01 00:00:00	2017-05-02 00:00:00	1	1

Showing 31 to 37 of 37 entries

Previous 1 2 3 4 Next text

Copyright © 2014-2018 Alimamul Shalim. All rights reserved. Version 2.3.7

Gambar 5. 2 Hasil Pengujian Orkestrasi Sales Order

Inventories

Show 10 entries

#	Id Rec	Product Name	Jumlah	Uom Name
1	REC14975533365	Aqua gelas	890	Pcs

Showing 1 to 1 of 1 entries

Previous

Gambar 5. 3 Hasil Pengujian Orkestrasi Receiving Goods

The screenshot shows a web interface for an 'Inventories' table. At the top, there is a search bar and a 'Show 10 entries' dropdown. The table has columns for '#', 'Id Rec', 'Product Name', 'Jumlah', and 'User Name'. A single row is visible with the following data: '# 1', 'Id Rec REC1497533365', 'Product Name Aqua gelas', 'Jumlah 890', and 'User Name Pcs'. Below the table, it says 'Showing 1 to 1 of 1 entries' and there is a 'Previous' button.

#	Id Rec	Product Name	Jumlah	User Name
1	REC1497533365	Aqua gelas	890	Pcs

Gambar 5. 4 Hasil Pengujian Orkestrasi *Issuing Goods*

5.2.2 Pengujian Koreografi

Pengujian ini dilakukan pada skenario pembelian barang antara *ritail* dan *supplier*, pengujian ini terdiri dari pengujian proses bisnis koreografi yang meliputi *order request*, *confirm order request*, *confirm delivery*, dan *confirm payment*. Berikut hasil pengujian pada pengiriman pesan transaksi pada proses koreografi. Skenario pengujian dapat dilihat pada Tabel 5.2.

Tabel 5.2 Tabel Pengujian Koreografi

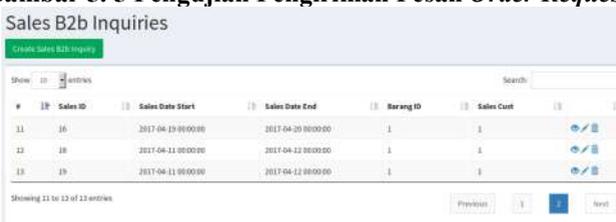
ID	UJ-CHOR
Referensi Kasus Penggunaan	Koreografi
Nama	Pengujian Koreografi
Tujuan Pengujian	Menguji kemampuan aplikasi melakukan koreografi
Skenario 1	<i>Retail melakukan order request ke supplier</i>
Kondisi Awal	<i>Pengguna berada pada halaman utama Purchasing B2B</i>
Data Uji	Inputan data <i>Order Request</i> :
Langkah Pengujian	<i>Pengguna</i> masuk ke halaman membuat data <i>Purchasing B2B</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil

Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Supplier melakukan konfirmasi Order</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama Sales B2B
Data Uji	data pada <i>Order Request</i> :
Langkah Pengujian	<i>Pengguna</i> memilih memproses <i>order request</i>
Hasil Yang Diharapkan	Sistem memproses order dan mengirim konfirmasi ke <i>retail</i>
Hasil Yang Didapat	Order terproses dan konfirmasi terkirim
Hasil Pengujian	Berhasil.
Kondisi Akhir	<i>Pengguna</i> berada pada <i>sales order B2B</i>
Skenario 3	<i>Supplier melakukan konfirmasi Pengiriman</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama Sales B2B
Data Uji	data pada <i>Order Request</i> :
Langkah Pengujian	<i>Pengguna</i> memilih memproses <i>order request</i>
Hasil Yang Diharapkan	Sistem mengirim konfirmasi pengiriman ke <i>retail</i>
Hasil Yang Didapat	Konfirmasi pengiriman terkirim
Hasil Pengujian	Berhasil.
Kondisi Akhir	<i>Pengguna</i> berada pada <i>sales order B2B</i>
Skenario 4	<i>Supplier melakukan konfirmasi Pembayaran</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama Sales B2B
Data Uji	data pada <i>Order Request</i> :
Langkah Pengujian	<i>Pengguna</i> memilih <i>confirm payment order request</i>
Hasil Yang Diharapkan	Sistem mengirim konfirmasi ke <i>retail</i>
Hasil Yang Didapat	konfirmasi pembayaran terkirim
Hasil Pengujian	Berhasil.
Kondisi Akhir	<i>Pengguna</i> berada pada <i>sales order B2B</i>

a. Pesan *Order Request*


The screenshot shows the 'Pur B2b Requisitions' interface. At the top, there is a green button labeled 'Create Pur B2b Requisition'. Below it, there is a search bar and a 'Show' dropdown set to '10 entries'. The main table has the following columns: #, Pur ID, Pur Code, Pur Date Start, Pur Date End, Product Name, Pur Jumlah, Pur Harga, and Pur Status. There is one entry in the table:

#	Pur ID	Pur Code	Pur Date Start	Pur Date End	Product Name	Pur Jumlah	Pur Harga	Pur Status
1	58	PUR1496938504	2017-06-01 00:00:00	2017-06-08 00:00:00	Aqua gelas	200	500	2

Gambar 5. 5 Pengujian Pengiriman Pesan *Order Request*


The screenshot shows the 'Sales B2b Inquiries' interface. At the top, there is a green button labeled 'Create Sales B2b Inquiry'. Below it, there is a search bar and a 'Show' dropdown set to '10 entries'. The main table has the following columns: #, Sales ID, Sales Date Start, Sales Date End, Barang ID, and Sales Cust. There are three entries in the table:

#	Sales ID	Sales Date Start	Sales Date End	Barang ID	Sales Cust
11	16	2017-04-19 00:00:00	2017-04-20 00:00:00	1	1
12	18	2017-04-11 00:00:00	2017-04-12 00:00:00	1	1
13	19	2017-04-11 00:00:00	2017-04-12 00:00:00	1	1

At the bottom, it says 'Showing 11 to 13 of 13 entries' and there are 'Previous', '1', and 'Next' buttons.

Gambar 5. 6 Pengujian Penerimaan Pesan *Order Request*

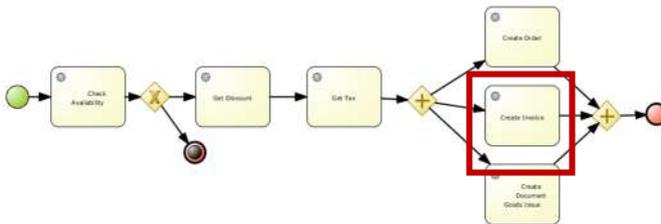
5.2.3 Pengujian *Flexibility* Proses Bisnis

Pengujian ini dilakukan untuk menguji *flexibility* rangkaian proses bisnis. Proses bisnis yang akan diuji adalah proses bisnis yang menangani proses penjualan barang. Detail scenario pengujiannya dapat dilihat pada Tabel 5.3.

Tabel 5.3 Tabel Pengujian *Flexibility*

ID	UJ-FLX
Referensi	<i>Flexibility</i>
Nama	Pengujian <i>Flexibility</i>
Tujuan Pengujian	Menguji kemampuan <i>flexibility</i> proses bisnis
Skenario 1	<i>Pengguna melakukan perubahan proses bisnis</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman editor jBPM
Data Uji	Proses Bisnis <i>Sales Order dan Issuing Goods</i> :

Langkah Pengujian	<i>Pengguna actor merubah alur Sales Order</i>
Hasil Yang Diharapkan	Proses bisnis bisa dirubah dan berjalan
Hasil Yang Didapat	Proses bisnis bisa dirubah dan berjalan
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna berada pada halaman editor jBPM</i>



Gambar 5. 7 orkestrasi *sales older*



Gambar 5. 8 orkestrasi *issuing goods*

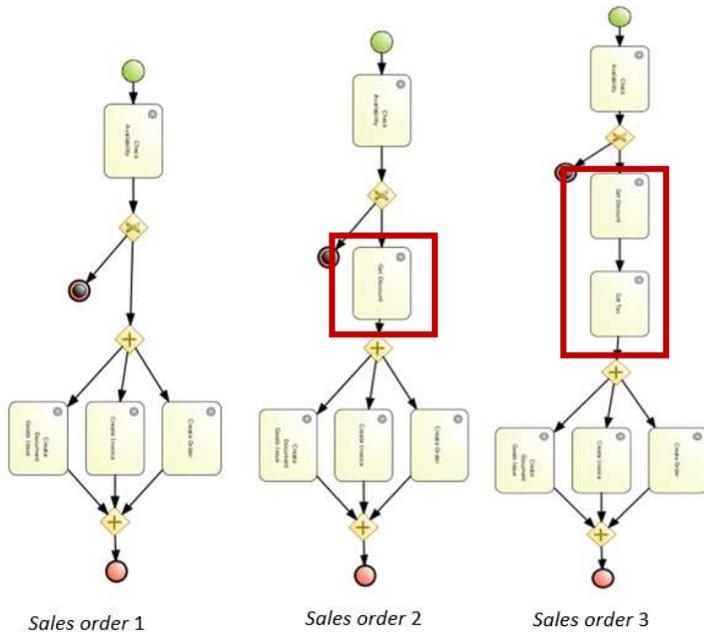
Hasil Pengujian *flexibility* ditunjukkan dengan sistem dapat menangani perubahan alur proses bisnis dimana *task create invoice* bisa diletakkan pada proses bisnis *sales order* ataupun pada proses bisnis *issuing goods*. Hal ini menunjukkan proses pembayaran pada penjualan dapat dilakukan sebelum atau sesudah barang penjualan dikirim ke pembeli. Hasilnya bisa dilihat pada Gambar 5.7 dan Gambar 5.8.

5.2.4 Pengujian *Scalability* Proses Bisnis

Pengujian ini dilakukan untuk menguji *scalability* rangkaian proses bisnis. Pengujian ini dilakukan pada proses bisnis *sales order*. Detail skenario pengujian dapat dilihat pada Tabel 5.4.

Tabel 5.4 Tabel Pengujian *Scalability*

ID	UJ-SCL
Referensi	<i>Scalability</i>
Nama	Pengujian <i>Scalability</i>
Tujuan Pengujian	Menguji kemampuan <i>Scalability</i> proses bisnis
Skenario 1	<i>Pengguna melakukan penambahan task proses bisnis</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman editor jBPM
Data Uji	Proses Bisnis <i>Sales Order</i> :
Langkah Pengujian	<i>Pengguna</i> menambah <i>task</i> pada proses bisnis <i>Sales Order</i>
Hasil Yang Diharapkan	Proses bisnis bisa ditambah <i>task</i> dan berjalan
Hasil Yang Didapat	Proses bisnis bisa ditambah <i>task</i> dan berjalan
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman editor jBPM



Gambar 5. 9 orkestrasi *issuing goods*

Hasil pengujian scalability ditunjukkan dengan sistem dapat menangani penambahan *task get discount*, dan *get tax* pada proses bisnis *sales order*. Hasil dapat dilihat pada Gambar 5.9.

5.2.5 Pengujian *Multi-Tenancy*

Pengujian ini terdiri dari pengujian *workflow repository* untuk menangani *multi-tenancy* dan pengujian variasi proses bisnis yang dapat ditangani oleh sistem.

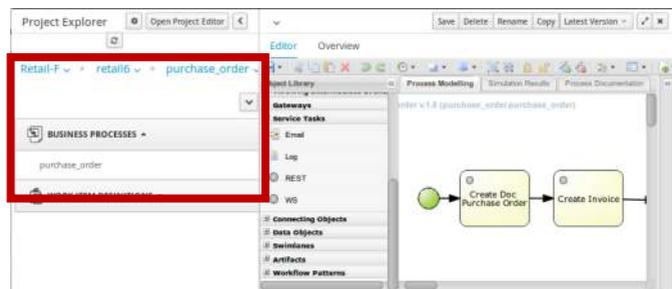
5.2.5.1 *Workflow Repository*

Pengujian ini terdiri dari pengujian untuk mengetahui kemampuan sistem untuk membuat *repository workflow* ketika ada *tenant* baru dan memberikan *role* pada *repository workflow* tersebut

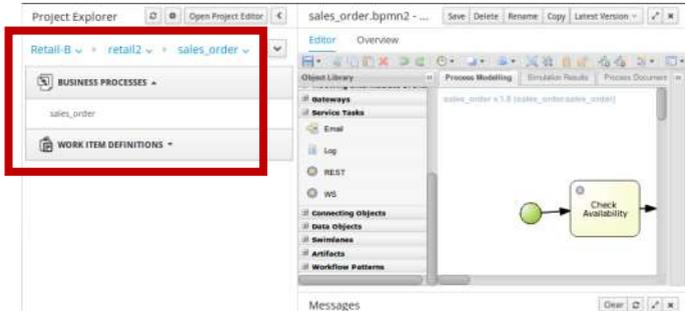
supaya hanya bisa diakses oleh *tenant* yang mempunyai hak akses. Detail pengujian dapat dilihat pada Tabel 5.5.

Tabel 5.5 Tabel Pengujian *Workflow Repository*

ID	UJ-WP
Referensi Analisis Workflow Repository	<i>Workflow Repository</i>
Nama	Pengujian <i>Workflow Repository</i> untuk <i>Multi-tenancy</i>
Tujuan Pengujian	Menguji kemampuan sistem menyediakan <i>workflow repository</i> untuk banyak pengguna dengan memberikan <i>role</i> akses
Skenario 1	<i>Admin membuat akun pengguna</i>
Kondisi Awal	<i>Admin berada pada halaman utama jBPM</i>
Data Uji	Data calon pengguna:
Langkah Pengujian	<i>Admin mendaftarkan data calon pengguna, admin membuatkan workflow repository, admin mengkonfigurasi hak akses Workflow repository</i>
Hasil Yang Diharapkan	<i>Workflow repository</i> berhasil dibuat dan hanya bisa diakses oleh pemiliknya.
Hasil Yang Didapat	<i>Workflow repository</i> hanya bisa diakses oleh pemiliknya.
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Admin berada pada halaman jBPM</i>



Gambar 5. 10 Editor proses bisnis user1



Gambar 5. 11 editor proses bisnis user2

Hasil pengujian ditunjukkan pada Gambar 5.10 dan Gambar 5.11 dimana pada Gambar 5.10 menampilkan *project editor* User2 dengan *repository* “Retail-A” dan proses bisnis “sales_order”. Kemudian pada Gambar 5.11 menampilkan *project editor* User6 dengan *repository* “Retail-F” dan proses bisnis “issuing_goods”. Dari hasil tersebut setiap *tenant* hanya bisa mengakses *repository* proses bisnis sesuai hak aksesnya.

5.2.5.2 Pengujian Variasi Proses Bisnis

Pengujian ini terdiri dari pengujian variasi proses bisnis yang mampu ditangani oleh sistem yang dibangun. Dari proses bisnis yang *common* kemudian dikombinasikan dengan variasi yang telah dijelaskan pada subbab 3.2.4.2 menghasilkan 31 varian proses bisnis. Semua variasi berhasil diterapkan namun ada beberapa yang bermasalah untuk beberapa penerapan variasi. Dari 31 variasi ada 4 penerapan variasi yang ada masalah atau sekitar 87% variasi proses bisnis dapat diterapkan tanpa masalah. Masalah banyak disebabkan pada saat penginputan data informasi *web service* dan saat integrasi dengan *web client* karena masih dilakukan secara manual. Untuk detail hasil pengujiannya dapat dilihat pada Lampiran A.1.

5.2.6 Pengujian Fungsionalitas

5.2.6.1 *Pengujian Fitur Manage Purchasing*

Pengujian ini terdiri dari pengujian membuat data *purchasing*, melihat daftar dan detail data *purchasing*. Data *purchasing* adalah data yang tersimpan dalam sistem saat terjadi transaksi pembelian. Detail skenario pengujian dapat dilihat pada Tabel 5.6.

Tabel 5.6 Tabel Pengujian Fitur *Manage Purchasing*

ID	UJ-UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Pengujian Fitur <i>Manage Purchasing</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Purchasing</i>
Skenario 1	<i>Pengguna melakukan pembuatan data purchasing</i>
Kondisi Awal	Pengguna berada pada halaman utama <i>purchasing</i>
Data Uji	Inputan data <i>Purchasing</i> :
Langkah Pengujian	pengguna masuk ke halaman membuat data <i>Purchasing</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Pengguna melihat daftar data Purchasing</i>
Kondisi Awal	Pengguna berada pada halaman utama <i>purchasing</i>
Data Uji	data <i>Purchasing</i> :
Langkah Pengujian	Pengguna masuk ke halaman <i>Purchasing</i>
Hasil Yang Diharapkan	Sistem menampilkan daftar data <i>purchasing</i>

Hasil Yang Didapat	Sistem menampilkan daftar data <i>purchasing</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman <i>purchasing</i>
Skenario 3	<i>Pengguna melihat detail salah satu data Purchasing</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Purchasing</i>
Data Uji	data pada <i>Purchasing</i> :
Langkah Pengujian	Pengguna memilih melihat salah satu data <i>Sales</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.

5.2.6.2 *Pengujian Fitur Manage Sales*

Pengujian ini terdiri dari pengujian pembuatan data *sales*, melihat daftar dan detail data *sales*. Data *sales* adalah data yang tersimpan dalam sistem saat terjadi transaksi penjualan. Detail skenario pengujian dapat dilihat pada Tabel 5.7.

Tabel 5.7 Tabel Pengujian Fitur *Manage Sales*

ID	UJ-UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Pengujian Fitur <i>Manage Sales</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Sales</i>
Skenario 1	<i>Pengguna melakukan pembuatan data Sales</i>
Kondisi Awal	Pengguna berada pada halaman utama <i>Sales</i>
Data Uji	Inputan data <i>Sales</i> :
Langkah Pengujian	<i>Pengguna</i> masuk ke halaman membuat data <i>Sales</i> baru dan memasukkan Data Uji ke dalam basis data.

Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman <i>sales</i>
Skenario 2	<i>Pengguna melihat daftar data Sales</i>
Kondisi Awal	Pengguna berada pada halaman utama <i>Sales</i>
Data Uji	Inputan data <i>Sales</i> :
Langkah Pengujian	<i>Pengguna</i> masuk ke halaman <i>sales</i>
Hasil Yang Diharapkan	Sistem menampilkan daftar data <i>sales</i>
Hasil Yang Didapat	Sistem menampilkan daftar data <i>sales</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 3	<i>Pengguna melihat detail salah satu data Sales</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Sales</i>
Data Uji	data pada <i>Sales</i> :
Langkah Pengujian	Pengguna memilih melihat salah satu data <i>Sales</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.

5.2.6.3 Pengujian Fitur Manage Goods receipt

Pengujian ini terdiri dari pengujian melihat daftar data *goods receipt*, melihat detail data *goods receipt*, dan melakukan proses penerimaan barang. Detail skenario pengujian dapat dilihat pada Tabel 5.8.

Tabel 5.8 Tabel Pengujian Fitur *Manage Goods Receipt*

ID	UJ-UC-003
Referensi Kasus Penggunaan	UC-003
Nama	Pengujian Fitur <i>Manage Goods Receipt</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Goods Receipt</i>
Skenario 1	<i>Pengguna melihat daftar data Goods Receipt</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Goods Receipt</i>
Data Uji	Inputan data <i>Goods Receipt</i> :
Langkah Pengujian	<i>Pengguna</i> masuk ke halaman membuat data <i>Goods Receipt</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Pengguna melihat detail salah satu data Goods Receipt</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Goods Receipt</i>
Data Uji	data pada <i>Goods Receipt</i> :
Langkah Pengujian	<i>Pengguna</i> memilih melihat salah satu data <i>Goods Receipt</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dipilih.
Skenario 3	<i>Pengguna melakukan penerimaan barang</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Goods Receipt</i>

Data Uji	data pada <i>Goods Receipt</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>Goods Receipt</i> yang akan dilakukan proses penerimaan kemudian memasukkan data terkait penerimaan.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih dan memperbarui data penerimaan
Hasil Yang Didapat	Sistem menampilkan data yang dipilih dan memperbarui data penerimaan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.

5.2.6.4 *Pengujian Fitur Manage Goods Issue*

Pengujian ini terdiri dari pengujian melihat daftar data *goods issue*, melihat detail data *goods issue*, dan melakukan proses penerimaan barang. Detail skenario pengujian dapat dilihat pada Tabel 5.9.

Tabel 5.9 Tabel Pengujian Fitur *Manage Goods Issue*

ID	UJ-UC-004
Referensi Kasus Penggunaan	UC-004
Nama	Pengujian <i>Fitur Manage Goods Issue</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Goods Issue</i>
Skenario 1	<i>Pengguna melihat daftar data Goods Issue</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Goods Issue</i>
Data Uji	Inputan data <i>Goods Issue</i> :
Langkah Pengujian	<i>Pengguna</i> masuk ke halaman membuat data <i>Goods Issue</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dimasukkan.

Skenario 2	<i>Pengguna melihat detail salah satu data Goods Issue</i>
Kondisi Awal	<i>Pengguna berada pada halaman utama Goods Issue</i>
Data Uji	data pada <i>Goods Issue</i> :
Langkah Pengujian	Pengguna memilih melihat salah satu data <i>Goods Issue</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.
Skenario 3	<i>Pengguna melakukan pengeluaran barang</i>
Kondisi Awal	<i>Pengguna berada pada halaman utama Goods Issue</i>
Data Uji	data pada <i>Goods Issue</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>Goods Issue</i> yang akan dilakukan proses penerimaan kemudian memasukkan data terkait pengeluaran.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih dan memperbarui data pengeluaran
Hasil Yang Didapat	Sistem menampilkan data yang dipilih dan memperbarui data pengeluaran
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.

5.2.6.5 *Pengujian Fitur Manage Delivery*

Pengujian ini terdiri dari pengujian melihat daftar data *delivery*, melihat detail data *delivery*, dan melakukan proses pengiriman barang. Detail skenario pengujian dapat dilihat pada Tabel 5.10.

Tabel 5.10 Tabel Pengujian Fitur *Manage Delivery*

ID	UJ-UC-005
Referensi Kasus Penggunaan	UC-005
Nama	Pengujian Fitur <i>Manage Delivery</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Purchasing</i>
Skenario 1	<i>Pengguna melihat daftar data Delivery</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Delivery</i>
Data Uji	Inputan data <i>Delivery</i> :
Langkah Pengujian	<i>Pengguna</i> masuk ke halaman membuat data <i>Delivery</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Pengguna melihat detail salah satu data Delivery</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Delivery</i>
Data Uji	data pada <i>Delivery</i> :
Langkah Pengujian	<i>Pengguna</i> memilih melihat salah satu data <i>Delivery</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dipilih.
Skenario 3	<i>Pengguna memproses pengiriman</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>delivery</i>
Data Uji	data pada <i>Delivery</i> :
Langkah Pengujian	<i>Pengguna</i> memproses pengiriman satu data <i>delivery</i> yang ada pada sistem.

Hasil Yang Diharapkan	Sistem menampilkan memproses perubahan data <i>delivery</i>
Hasil Yang Didapat	Sistem menampilkan memproses perubahan data <i>delivery</i>
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.

5.2.6.6 Pengujian Fitur *Manage Purchasing B2B*

Pengujian ini terdiri dari pengujian membuat data *purchasing B2B*, melihat daftar dan detail data *purchasingB2B*, menyunting dan menghapus data *purchasing*, memproses *order*, dan konfirmasi pembayaran. Data *purchasing B2B* adalah data yang tersimpan dalam sistem saat terjadi transaksi pembelian B2B. Detail skenario pengujian dapat dilihat pada Tabel 5.11.

Tabel 5.11 Tabel Pengujian Fitur *Manage Purchasing B2B*

ID	UJ-UC-006
Referensi Kasus Penggunaan	UC-006
Nama	Pengujian Fitur <i>Manage Purchasing B2B</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Purchasing B2B</i>
Skenario 1	<i>Pengguna membuat data purchasing B2B</i>
Kondisi Awal	Pengguna berada pada halaman utama <i>Purchasing B2B</i>
Data Uji	Inputan data <i>Purchasing B2B</i> :
Langkah Pengujian	Pengguna masuk ke halaman membuat data <i>Purchasing</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman <i>purchasing B2B</i>
Skenario 2	<i>Pengguna melihat detail salah satu data purchasing B2B</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>purchasing B2B</i>

Data Uji	data pada <i>Purchasing B2B</i> :
Langkah Pengujian	Pengguna memilih melihat salah satu data <i>purchasing B2B</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.
Skenario 3	<i>Pengguna penyuntingan data purchasing B2B</i>
Kondisi Awal	pengguna berada pada halaman utama <i>purchasing B2B</i>
Data Uji	data pada <i>Purchasing B2B</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>purchasing B2B</i> yang ada pada sistem dan mengubahnya.
Hasil Yang Diharapkan	Data baru yang dibasukkan menggantikan data lama pada basis data
Hasil Yang Didapat	Data baru tersimpan dalam basis data
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>purchasing B2B</i>
Skenario 4	<i>Pengguna menghapus data purchasing B2B</i>
Kondisi Awal	pengguna berada pada halaman utama <i>purchasing B2B</i>
Data Uji	data pada <i>Purchasing B2B</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>purchasing B2B</i> yang ada pada sistem untuk dihapus.
Hasil Yang Diharapkan	Data terhapus dari basis data
Hasil Yang Didapat	Data terhapus dari basis data
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>purchasing B2B</i>
Skenario 5	<i>Pengguna memproses order data purchasing B2B</i>
Kondisi Awal	pengguna berada pada halaman utama <i>purchasing B2B</i>

Data Uji	data pada <i>Purchasing B2B</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>purchasing B2B</i> yang ada pada sistem dan menekan tombol “ <i>Process Order</i> ”.
Hasil Yang Diharapkan	Status data terproses dan pesan order terkirim ke <i>supplier</i>
Hasil Yang Didapat	Status data terproses dan pesan order terkirim ke <i>supplier</i>
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>purchasing B2B</i>
Skenario 6	<i>Pengguna mengkonfirmasi pembayaran order</i>
Kondisi Awal	pengguna berada pada halaman utama <i>purchasing B2B</i>
Data Uji	data pada <i>Purchasing B2B</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>purchasing B2B</i> yang ada pada sistem dan menekan tombol “ <i>Confirm Paymnet</i> ”.
Hasil Yang Diharapkan	Konfirmasi pembayaran terkirim ke <i>supplier</i>
Hasil Yang Didapat	Konfirmasi pembayaran terkirim ke <i>supplier</i>
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>purchasing B2B</i>

5.2.6.7 *Pengujian Fitur Manage Sales B2B*

Pengujian ini terdiri dari pengujian membuat data *sales B2B*, melihat daftar dan detail data *sales B2B*, menyunting dan menghapus data *purchasing*, konfirmasi *order*, dan konfirmasi pengiriman. Data *sales B2B* adalah data yang tersimpan dalam sistem saat terjadi transaksi pembelian B2B. Detail skenario pengujian dapat dilihat pada Tabel 5.12.

Tabel 5.12 Tabel Pengujian Fitur *Manage Sales B2B*

ID	UJ-UC-007
Referensi Kasus Penggunaan	UC-007
Nama	Pengujian Fitur <i>Manage Sales B2B</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>sales B2B</i>

Skenario 1		Pengguna membuat data sales B2B
Kondisi Awal	Pengguna berada pada halaman utama <i>sales B2B</i>	
Data Uji	Inputan data <i>sales B2B</i> :	
Langkah Pengujian	Pengguna masuk ke halaman membuat data <i>sales B2B</i> baru dan memasukkan Data Uji ke dalam basis data.	
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.	
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data	
Hasil Pengujian	Berhasil	
Kondisi Akhir	Pengguna berada pada halaman <i>sales B2B</i>	
Skenario 2		Pengguna melihat detail salah satu data sales B2B
Kondisi Awal	Pengguna berada pada halaman utama <i>sales B2B</i>	
Data Uji	data pada <i>Purchasing B2B</i> :	
Langkah Pengujian	Pengguna memilih melihat salah satu data <i>sales B2B</i> yang ada pada sistem.	
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih	
Hasil Yang Didapat	Detail data ditampilkan	
Hasil Pengujian	Berhasil.	
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.	
Skenario 3		Pengguna penyuntingan data sales B2B
Kondisi Awal	pengguna berada pada halaman utama <i>sales B2B</i>	
Data Uji	data pada <i>Sales B2B</i> :	
Langkah Pengujian	Pengguna memilih salah satu data <i>sales B2B</i> yang ada pada sistem dan mengubahnya.	
Hasil Yang Diharapkan	Data baru yang dimasukkan menggantikan data lama pada basis data	
Hasil Yang Didapat	Data baru tersimpan dalam basis data	
Hasil Pengujian	Berhasil.	
Kondisi Akhir	Pengguna berada pada halaman utama <i>sales B2B</i>	
Skenario 4		Pengguna menghapus data sales B2B

Kondisi Awal	pengguna berada pada halaman utama <i>sales B2B</i>
Data Uji	data pada <i>sales B2B</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>sales B2B</i> yang ada pada sistem untuk dihapus.
Hasil Yang Diharapkan	Data terhapus dari basis data
Hasil Yang Didapat	Data terhapus dari basis data
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>sales B2B</i>
Skenario 5	<i>Pengguna mengkonfirmasi order request data sales B2B</i>
Kondisi Awal	pengguna berada pada halaman utama <i>sales B2B</i>
Data Uji	data pada <i>sales B2B</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>sales B2B</i> yang ada pada sistem dan menekan tombol “ <i>Confirm Order</i> ”.
Hasil Yang Diharapkan	Status data terproses dan pesan konfirmasi order terkirim ke <i>retail</i>
Hasil Yang Didapat	Status data terproses dan pesan konfirmasi order terkirim ke <i>retail</i>
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>sales B2B</i>
Skenario 6	<i>Pengguna mengkonfirmasi pengiriman</i>
Kondisi Awal	pengguna berada pada halaman utama <i>sales B2B</i>
Data Uji	data pada <i>sales B2B</i> :
Langkah Pengujian	Pengguna memilih salah satu data <i>sales B2B</i> yang ada pada sistem dan menekan tombol “ <i>Confirm Delivery</i> ”.
Hasil Yang Diharapkan	Konfirmasi pengiriman terkirim ke <i>retail</i>
Hasil Yang Didapat	Konfirmasi pengiriman terkirim ke <i>retail</i> dan status data order berubah
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>sales B2B</i>

5.2.6.8 *Pengujian Fitur Manage Service Repository*

Pengujian ini terdiri dari pengujian membuat data *service repository*, melihat daftar dan detail data *workflow*, menyunting dan menghapus data *service repository*. Detail skenario pengujian dapat dilihat pada Tabel 5.13.

Tabel 5.13 Tabel Pengujian Fitur Manage Service Repository

ID	UJ-UC-008
Referensi Kasus Penggunaan	UC-008
Nama	Pengujian Fitur <i>Manage Service Repository</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Service Repository</i>
Skenario 1	<i>Pengguna melihat daftar data Service Repository</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Service Repository</i>
Data Uji	Inputan data <i>Service Repository</i> :
Langkah Pengujian	<i>Pengguna</i> masuk ke halaman membuat data <i>Sales</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Pengguna melihat detail salah satu data Service Repository</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Service Repository</i>
Data Uji	data pada <i>Service Repository</i> :
Langkah Pengujian	<i>Pengguna</i> memilih melihat salah satu data <i>Service Repository</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.

Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.
----------------------	---

5.2.6.9 *Pengujian Fitur Manage Workflow*

Pengujian ini terdiri dari pengujian membuat data *workflow*, melihat daftar dan detail data *workflow*, menyunting dan menghapus data *workflow*. Detail skenario pengujian dapat dilihat pada Tabel 5.13.

Tabel 5.14 Tabel Pengujian Fitur Manage Workflow

ID	UJ-UC-009
Referensi Kasus Penggunaan	UC-009
Nama	Pengujian Fitur <i>Manage Workflow</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>workflow</i>
Skenario 1	<i>Pengguna melihat daftar data Workflow</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Workflow</i>
Data Uji	Inputan data <i>Workflow</i> :
Langkah Pengujian	<i>Pengguna</i> masuk ke halaman membuat data <i>Workflow</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	<i>Pengguna</i> berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Pengguna melihat detail salah satu data Workflow</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>Workflow</i>
Data Uji	data pada <i>Workflow</i> :
Langkah Pengujian	<i>Pengguna</i> memilih melihat salah satu data <i>Workflow</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan

Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih

5.2.6.10 *Pengujian Fitur Manage General Ledger*

Pengujian ini terdiri dari pengujian melihat data laporan dari *balance sheet*, *cash flow*, dan *income statement*. Detail skenario pengujian dapat dilihat pada Tabel 5.15.

Tabel 5.15 Tabel Pengujian Fitur General Ledger

ID	UJ-UC-010
Referensi Kasus Penggunaan	UC-010
Nama	Pengujian Fitur <i>Manage General Ledger</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>General Ledger</i>
Skenario 1	<i>Pengguna melihat detail Balance Sheet</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>General Ledger</i>
Data Uji	data pada <i>General Ledger</i> :
Langkah Pengujian	Pengguna memilih melihat salah satu data <i>General Ledger</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.
Skenario 2	<i>Pengguna melihat detail Cash Flow</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>General Ledger</i>
Data Uji	data pada <i>General Ledger</i> :
Langkah Pengujian	Pengguna memilih melihat salah satu data <i>General Ledger</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih

Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.
Skenario 3	<i>Pengguna melihat detail Income Statement</i>
Kondisi Awal	<i>Pengguna</i> berada pada halaman utama <i>General Ledger</i>
Data Uji	data pada <i>General Ledger</i> :
Langkah Pengujian	Pengguna memilih melihat salah satu data <i>General Ledger</i> yang ada pada sistem.
Hasil Yang Diharapkan	Sistem menampilkan data yang dipilih
Hasil Yang Didapat	Detail data ditampilkan
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja dipilih.

5.3 Evaluasi Pengujian

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.16. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik disimpulkan bahwa fungsionalitas dari program telah bisa bekerja sesuai dengan yang diharapkan.

Tabel 5.16 Rangkuman Hasil Pengujian

ID	Nama	Skenario	Hasil
UJ-ORC	Pengujian orkestrasi	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
UJ-CHOR	Pengujian koreografi	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil

		Skenario 4	Berhasil
UJ-FLX	Pengujian <i>flexibility</i> proses bisnis	Skenario 1	Berhasil
UJ-SCL	Pengujian <i>scalability</i> proses bisnis	Skenario 1	Berhasil
UJ.WP	Pengujian <i>workflow repository</i> untuk <i>multi-tenancy</i>	Skenario 1	Berhasil
UJ-UC-001	Pengujian fitur mengelola <i>Purchasing</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-002	Pengujian fitur mengelola <i>Sales</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-003	Pengujian fitur mengelola <i>Goods Receipt</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ-UC-004	Pengujian fitur mengelola <i>Goods Issue</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
UJ-UC-005	Pengujian fitur mengelola <i>Delivery</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
UJ-UC-006	Pengujian fitur mengelola <i>Purchasing B2B</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
		Skenario 5	Berhasil
		Skenario 6	Berhasil
UJ-UC-007	Pengujian fitur mengelola <i>Sales B2B</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil

		Skenario 5	Berhasil
		Skenario 6	Berhasil
UJ-UC-008	Pengujian fitur mengelola <i>Service Repository</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
UJ-UC-009	Pengujian fitur mengelola <i>Workflow</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
UJ-UC-010	Pengujian fitur mengelola <i>General Ledger</i>	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil

BAB VI KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Penerapan *web service* pada aplikasi ERP *retail* telah berhasil diimplementasikan dengan menggunakan protocol SOAP (*Simple Object Access Protocol*). Implementasi dilakukan terhadap seluruh rancangan implementasi *web service*. Hasil pengujian menunjukkan *web service* berfungsi sesuai fungsionalnya.
2. Implementasi orkestrasi *web service* berhasil dilakukan pada proses bisnis *purchase order*, *sales order*, *receiving goods*, dan *issuing goods*. Orkestrasi dilakukan dengan menggunakan BPMS jBPM. Hasil pengujian menunjukkan orkestrasi berfungsi sesuai rangkaian proses bisnis yang dirangkai.
3. Implementasi koreografi *web service* berhasil dilakukan pada proses pembelian barang yang terjadi antara pihak *retail* dan *supplier*. Koreografi *web service* dilakukan pada layer UI/tampilan dan belum bisa dilakukan dengan diagram seperti pada implementasi orkestrasi. Pesan *service* yang berhasil diimplementasikan adalah *order request*, *confirm order request*, *confirm delivery*, dan *confirm paymnet*. Hasil pengujian menunjukkan pengiriman pesan melalui *service* berfungsi sesuai fungsinya.
4. Managemen proses bisnis untuk *multi-tenant* dilakukan dengan menyediakan *workflow repository* serta *common* dan variasi/*fragment* proses bisnis. Hasil pengujian pemberian *role* pada setiap *repository* menunjukkan *repository* hanya

dapat diakses oleh pemilik *repository*. Hasil pengujian penyediaan *common* dan variasi/*fragment* proses bisnis menunjukkan sistem dapat menangani 31 variasi komposisi proses bisnis dengan tingkat keberhasilan penerapan komposisi 87%.

5. Penerapan *web service* dan BPMS pada aplikasi ERP *retail* membuat proses bisnis aplikasi ERP *retail* menjadi *flexible* dan *scalable*. *Flexibility* ditunjukkan dengan sistem dapat menangani perubahan alur proses bisnis dimana *task create invoice* bisa diletakkan pada proses bisnis *sales order* ataupun pada proses bisnis *issuing goods*. *Scalability* ditunjukkan dengan sistem dapat menangani penambahan *task get discount*, dan *get tax* pada proses bisnis *sales order*.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Penerapan *web service composition* pada lingkup proses bisnis yang lebih besar untuk memaksimalkan fungsi dari teknologi *web service composition*.
2. Penerapan koreografi *web service* dengan menggunakan diagram untuk memudahkan dalam penyusunannya.
3. Menambahkan unsur keamanan pada *web service* untuk menjamin integritas data dalam sistem.
4. Penerapan UI *web client* yang dinamis supaya bisa mengikuti komposisi proses bisnis.
5. Penerapan integrasi otomatis antara proses bisnis yang dirangkai dengan *web client* untuk mengurangi tingkat kesalahan konfigurasi untuk integrasi.

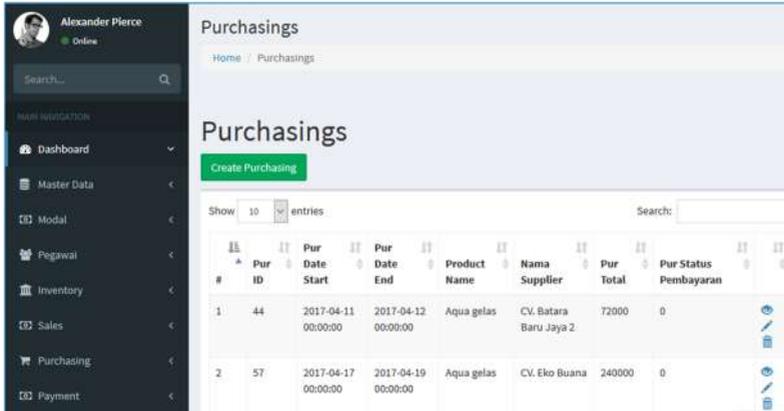
DAFTAR PUSTAKA

- [1] A. D. A. P. D. S. Riyanarto Sarno, “Rancang Bangun Orkestrasi *Web service* serta Implementasi Single Sign On pada Enterprise Resource Planning,” 2011.
- [2] B. Hitpass and J. Leiva, “Modeling a *Retail* B2B Integration Portal as Choreography Intermediary with BPMN 2,0 Choreography Diagrams,” 2013.
- [3] “*Retail*,” [Online]. Available: <https://en.wikipedia.org/wiki/Retail>. [Accessed 13 Desember 2016].
- [4] T.-H. Kuo, C.-H. Chen, H.-Y. Kung and Y.-S. Liao, "Applications of the *Web service* Middleware Framework Based on the BPEL," 2016.
- [5] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne and X. Xu, "*Web services* composition: A decade's overview," 2014.
- [6] "SOAP," [Online]. Available: <https://en.wikipedia.org/wiki/SOAP>. [Accessed 19 June 2017].
- [7] "Representational state transfer," [Online]. Available: https://en.wikipedia.org/wiki/Representational_state_transfer. [Accessed 19 June 2017].
- [8] Y. Vasiliev, in *SOA and WS-BPEL*, Mumbai, Packt Publishing, 2007, p. 25.
- [9] S. Y. Majid Hosseinpour Agdam, “A Flexible and Scalable Framework for QoS-Aware *Web service* s Composition,” *International Symposium on Telecommunications*, 2010.
- [10] S. T. K. D. Emna Fki *, *Automated and flexible composition based on abstract services for a better adaptation to user intentions*, 2016.

- [11] Y. Wang and YuZhang, "The Research and Application of Web-Application Based Workflow Engine," 2011.
- [12] X. Wang, J. Zhang and W. Cui, "A Process Web Client Interaction Plug-in Design and Development Based on jBPM," 2012.

LAMPIRAN

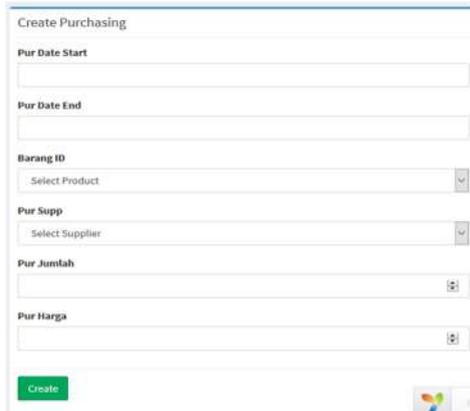
Gambar



The screenshot shows a web application interface for 'Purchasing'. On the left is a dark sidebar with a user profile for 'Alexander Pierce' (Online) and a navigation menu with items: Dashboard, Master Data, Modal, Pegawai, Inventory, Sales, Purchasing (highlighted), and Payment. The main content area has a 'Purchasing' header with a 'Home / Purchasing' breadcrumb and a 'Create Purchasing' button. Below the header is a table with 2 entries. The table has columns for #, Pur ID, Pur Date Start, Pur Date End, Product Name, Nama Supplier, Pur Total, and Pur Status Pembayaran. The first entry has Pur ID 44, dates from 2017-04-11 to 2017-04-12, product 'Aqua gelas', supplier 'CV. Batara Baru Jaya 2', total 72000, and status 0. The second entry has Pur ID 57, dates from 2017-04-17 to 2017-04-19, product 'Aqua gelas', supplier 'CV. Eko Buana', total 240000, and status 0. Each row has edit and delete icons.

#	Pur ID	Pur Date Start	Pur Date End	Product Name	Nama Supplier	Pur Total	Pur Status Pembayaran
1	44	2017-04-11 00:00:00	2017-04-12 00:00:00	Aqua gelas	CV. Batara Baru Jaya 2	72000	0
2	57	2017-04-17 00:00:00	2017-04-19 00:00:00	Aqua gelas	CV. Eko Buana	240000	0

Gambar A.1 Antarmuka Daftar Data *Purchasing*



The screenshot shows the 'Create Purchasing' form. It contains several input fields: 'Pur Date Start' and 'Pur Date End' (text boxes), 'Barang ID' (a dropdown menu with 'Select Product'), 'Pur Supp' (a dropdown menu with 'Select Supplier'), 'Pur Jumlah' (a text box with a spinner), and 'Pur Harga' (a text box with a spinner). At the bottom left is a green 'Create' button, and at the bottom right is a small logo.

Gambar A.2 Antarmuka Menambah Data *Purchasing*

44

Update Delete

Pur ID	44
Pur Date Start	2017-04-11 00:00:00
Pur Date End	2017-04-12 00:00:00
Barang ID	1
Pur Supp	1
Pur Jumlah	30
Pur Harga	2400
Pur Total	72000
Pur Status Pembayaran	0
Pur Status	0
Id Retail	1

Gambar A.3 Antarmuka Detail Data *Purchasing*

Sales

Create Sales

Show 10 entries Search:

#	Sales Id	Sales Date Start	Sales Date End	Product Product Name	Customer Customer Name	Sales Jumlah	Sales Total	
1	SO1497562078	2017-06-01 00:00:00	2017-06-22 00:00:00	Aqua gelas	Andi Putra Kusuma	100	106700	
2	SO1497562410	2017-06-11 00:00:00	2017-06-22 00:00:00	Aqua gelas	Andi Putra Kusuma	10	6300	

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar A.4 Antarmuka Daftar Data *Sales*

Home / Sales / Create Sales

Create Sales

Sales Date Start

Sales Date End

Barang ID
Select Product

Sales Cust
Select Customer

Sales Jumlah

Sales Harga

Gambar A.5 Antarmuka Menambah Data *Sales*

Home / Sales / 1

1

Update Delete

Sales ID	1
Sales Date Start	2017-03-31 00:00:00
Sales Date End	2017-04-21 00:00:00
Barang ID	1
Sales Cust	1
Sales Jumlah	14
Sales Harga	1000
Sales Total	14230

Gambar A.6 Antarmuka Detail Data *Sales*

Inventories

Show 10 entries Search

#	Id Rec	Product Name	Jumlah	Uom Name
1	REC1497553285	Aqua gelas	890	Pcs

Showing 1 to 1 of 1 entries Previous

Gambar A.7 Antarmuka Daftar Data *Stock*

Goods Receives

Create Goods Receive

Show 12 entries

#	ID Rec	Rec Date	Supplier Nama Supplier	Rec Status	
1	REC1497547161	2017-06-15 17:00:21	CV. Eko Buana	0	🔍 ✎
2	REC1497547228	2017-06-15 17:00:20	CV. Karya Surya	0	🔍 ✎
3	REC1497553365	2017-06-15 18:00:45	CV. Batara Baru, Jaya 2	0	🔍 ✎
4	REC1497584049	2017-06-15 10:00:00	CV. Batara Baru, Jaya 2	0	🔍 ✎

Showing 1 to 4 of 4 entries

Previous 1 Next

Gambar A.8 Antarmuka Daftar Data *Goods Receipt*

Gambar A.9 Antarmuka Detail Data *Goods Receipt*

Create Goods Receive

Pur ID

1

Rec Date

Supplier

CV. Eko Buana

Barang

Aqua gelas

Jumlah Pesanan

10

Total Harga

0

Jumlah Barang

Fix Item

Create

Gambar A.10 Antarmuka Transfer Posting

Goods Issues

Create Goods Issue

Show 12 entries

#	ID Issue	Issue Date	Issue Status	
1	ISSUE1497542341	2017-06-15 21:00:23	0	🔍 ✎
2	ISSUE1497582443	2017-06-15 21:00:00	0	🔍 ✎

Showing 1 to 2 of 2 entries

Previous 1 Next

Gambar A.11 Antarmuka Daftar Data *Goods Receipt*

Gambar A.12 Antarmuka Detail Data *Goods Receipt*

Create Goods Receive

Pur ID
1

Rec Date

Supplier
CV. Eko Buana

Barang
Aqua gelas

Jumlah Pesanan
10

Total Harga
0

Jumlah Barang
Fix Item

Create

Gambar A.13 Antarmuka Transfer Posting

Pembayaran Pembelian

Create Payment

Show: 10 entries

#	Pay Trans	Pay Total	Pay Status
1	56	600000	0
2	57	240000	0

Showing 2 of 2 entries

Gambar A.14 Antarmuka Daftar Data *Invoice*

Create Payment

Payment Date

Payment Jumlah

Payment Transaksi

56

Create

Gambar A.15 Antarmuka Membayar *Invoice*

Balance Sheet

Period ID

2017-05

Filter

Periode : 2017-05

Assets

Total Asset Rp 0,-

Liability&Equity

Liability

Total Liability Rp 0,-

Equity

Gambar A.16 Antarmuka *balance Sheet*

Cash Flow	
Period ID	
2017-06	
Filter	
Periode : 2017-06	
<i>Aktivitas Operasional</i>	
Pendapatan	
Total Pendapatan Operasional	Rp 0,-
Biaya Pengurangan	
Total Biaya Operasional	Rp 0,-
<i>Total Aktivitas Operasional</i>	Rp 0,-
<i>Aktivitas Non Operasional</i>	
Lainnya	
<i>Total Aktivitas Non Operasional</i>	Rp 0,-
<i>Perpindahan Kas Bersih</i>	Rp 0,-
<i>Ringkasan</i>	
Saldo Pembukaan	Rp 0,-
Perpindahan Kas Bersih	Rp 0,-
Saldo Penutupan	Rp 0,-

Gambar A.17 Antarmuka *cash flow*

Income Statement	
Period ID	
<input type="text" value="2017-08"/>	
<input type="button" value="Filter"/>	
Periode : 2017-06	
<i>Pendapatan</i>	
Pendapatan dari Penjualan	
Total Pendapatan dari Penjualan	Rp 0,-
Harga Pokok Penjualan	
Total Harga Pokok Penjualan	Rp 0,-
Laba Kotor	Rp 0,-
Biaya Operasional	
Total Biaya	Rp 0,-
Pendapatan Bersih Operasional	Rp 0,-
Pendapatan Lainnya	
Total Pendapatan Lainnya	Rp 0,-
Biaya Lainnya	
Total Biaya Lainnya	Rp 0,-
Pendapatan Bersih	Rp 0,-

Gambar A.18 Antarmuka *Income Statement*

Tabel

Tabel A. 1 Variasi proses bisnis dan hasil pengujiannya

Tipe retail	Purchase Order			Receiving Goods			Sales Order				Issuing Goods			Pengujian
	Create PO	Create Doc Goods Receipt	Purchase invoice	Receiving Goods	Create Return	Create SO	Discount	Tax	Sales Invoice	Create Doc Goods Issue	Issuing Goods	Delivery	Sales Invoice	Keterangan
Retail 1	v	v	v	v		v			v	v	v			berhasil
Retail 2	v	v	v	v		v			v	v	v	v		berhasil
Retail 3	v	v	v	v	v	v			v	v	v			berhasil
Retail 4	v	v	v	v	v	v			v	v	v	v		berhasil
Retail 5	v	v	v	v		v	v		v	v	v			berhasil
Retail 6	v	v	v	v		v	v		v	v	v	v		berhasil
Retail 7	v	v	v	v	v	v	v		v	v	v			berhasil
Retail 8	v	v	v	v	v	v	v		v	v	v	v		berhasil
Retail 9	v	v	v	v		v		v	v	v	v			berhasil
Retail 10	v	v	v	v		v		v	v	v	v	v		berhasil
Retail 11	v	v	v	v	v	v		v	v	v	v			berhasil
Retail 12	v	v	v	v	v	v		v	v	v	v	v		berhasil
Retail 13	v	v	v	v		v	v	v	v	v	v			berhasil*
Retail 14	v	v	v	v		v	v	v	v	v	v	v		berhasil
Retail 15	v	v	v	v	v	v		v	v	v	v			berhasil
Retail 16	v	v	v	v	v	v	v	v	v	v	v	v		berhasil*
Retail 17	v	v	v	v		v				v	v	v	v	berhasil
Retail 18	v	v	v	v	v	v				v	v		v	berhasil
Retail 19	v	v	v	v	v	v				v	v	v	v	berhasil
Retail 20	v	v	v	v		v	v			v	v		v	berhasil
Retail 21	v	v	v	v		v	v			v	v	v	v	berhasil

[Halaman ini sengaja dikosongkan]

Retail 22	v	v	v	v	v	v	v			v	v		v	berhasil
Retail 23	v	v	v	v	v	v	v			v	v	v	v	berhasil
Retail 24	v	v	v	v		v		v		v	v		v	berhasil*
Retail 25	v	v	v	v		v		v		v	v	v	v	berhasil
Retail 26	v	v	v	v	v	v		v		v	v		v	berhasil
Retail 27	v	v	v	v	v	v		v		v	v	v	v	berhasil
Retail 28	v	v	v	v		v	v	v		v	v		v	berhasil
Retail 29	v	v	v	v		v	v	v		v	v	v	v	berhasil*
Retail 30	v	v	v	v	v	v		v		v	v		v	berhasil
Retail 31	v	v	v	v	v	v	v	v		v	v	v	v	berhasil

Ket.

V = Fitur yang digunakan

■ = Fitur *common*

.* = Ada kendala

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Ahmad Zaenal Mustofa, lahir pada tanggal 29 Juni 1994 di Kebumen. Alumni dari MI Muhammadiyah Redisari, SMP N 1 Rowokele dan SMA N 1 Gombang. Sejak 2013 hingga kini, sedang menimba ilmu di jurusan Teknik Informatika, Fakultas Teknologi Informasi (FTIf), Institut Teknologi Sepuluh Nopember (ITS), Surabaya.

Memiliki pengalaman organisasi sebagai Staf Departemen Media dan Informasi HMTC 2014-2015, Ketua Tim Big Event UKM Cinta Rebana ITS 2014-2015, Staf NST Schematic 2014, Bendahara NST Schematic 2015, Staf Media Keluarga Muslim Informatika 2014-2015, Ketua Departemen Media dan Informasi UKM Cinta Reban ITS 2015-2016. Penulis dapat dihubungi melalui surel email: mustofaahmad29@gmail.com.

[Halaman ini sengaja dikosongkan]