



TUGAS AKHIR - KI141502

RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL *PROCUREMENT PROCESS (PURCHASING)* BERORIENTASIKAN *MULTI-TENANCY* DENGAN SISTEM BASIS DATA TERDISTRIBUSI

**MUHAMMAD NICKO RAHMADANO
NRP 5112 100 142**

**Dosen Pembimbing I
Sarwosri, S.Kom, MT.**

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

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

**DESIGN AND IMPLEMENTATION OF
ENTERPRISE RESOURCE PLANNING IN
PROCUREMENT PROCESS (PURCHASING)
WITH MULTI-TENANCY ORIENTED USING
DISTRIBUTED DATABASE**

**MUHAMMAD NICKO RAHMADANO
NRP 5112 100 142**

**Supervisor I
Sarwosri, S.Kom, MT.**

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

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL *PROCUREMENT PROCESS (PURCHASING)* BERORIENTASIKAN *MULTI-TENANCY* DENGAN SISTEM BASIS DATA TERDISTRIBUSI

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

MUHAMMAD NICKO RAHMADANO

NRP. 5112 100 112

Disetujui oleh Dosen Pembimbing 1 dan Pembimbing 2

Sarwosri, S.Kom, MT.

NIP: 19760809 200112 2 001

Prof. Drs. Ec. Ir. Rryanarto Sarno, M.Sc., Ph.D.

NIP: 19590803 198601 1 001



(pembimbing 1)

(pembimbing 2)

SURABAYA
JULI, 2016

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN SISTEM *ENTERPRISE RESOURCE PLANNING* PADA MODUL *PROCUREMENT PROCESS (PURCHASING)* BERORIENTASIKAN *MULTI-TENANCY* DENGAN SISTEM BASIS DATA TERDISTRIBUSI

Nama : Muhammad Nicko Rahmadano
NRP : 5112100142
Jurusan : Teknik Informatika – FTIf ITS
Dosen Pembimbing I : Sarwosri, S.Kom, MT.
Dosen Pembimbing II : Prof. Drs. Ec. Ir. Rinyanarto Sarno, M.Sc.,Ph.D.

ABSTRAK

Enterprise Resource Planning adalah sebuah sistem aplikasi yang digunakan untuk mengelola sebuah perusahaan besar yang dirancang untuk mengkoordinasikan semua sumber daya, informasi dan aktifitas yang diperlukan untuk proses bisnis lengkap. ERP memiliki beberapa modul, salah satu modul yang penting dan yang telah dikerjakan pada tugas akhir ini yaitu *Procurement Process* atau *Purchasing*. *Purchasing* adalah suatu proses pengadaan barang pada perusahaan, *Purchasing* bertanggung jawab terutama pada pengurusan permintaan barang, pembelian barang, memonitor pembelian barang, dan pemilihan supplier.

Dalam tugas akhir ini, dikembangkan aplikasi ERP yang dibangun dengan menggunakan sistem *distributed database* berorientasi *multitenancy* yaitu satu aplikasi dapat digunakan oleh beberapa perusahaan/tenant secara bersamaan dengan memisahkan database setiap tenant ke dalam computer/server tersendiri yang dibedakan oleh IP address setiap computer/server. Hal ini bertujuan untuk mencegah kegagalan sistem jika sebuah computer/server mengalami down, maka

database masih bisa diakses oleh computer server lain. Pada modul Procurement Process dibuat sistem pembelian barang mulai dari permintaan pembelian barang (purchase requisition) sampai melakukan pembelian secara resmi (purchase order), dan menggunakan metode TOPSIS untuk menentukan supplier terbaik saat ingin membeli barang atau raw material berdasarkan dari transaksi pembelian yang telah terjadi sebelumnya.

Hasil implementasi menunjukkan bahwa proses pembelian barang berhasil berjalan dengan baik dan benar. Untuk pemilihan supplier atau supplier analysis berhasil me-ranking supplier terbaik agar pembelian barang bisa berjalan dengan lancar.

Kata kunci: Procurement Process, Purchasing, Supplier Analysis, purchase requisition, purchase order, TOPSIS.

DESIGN AND IMPLEMENTATION OF ENTERPRISE RESOURCE PLANNING IN PROCUREMENT PROCESS (PURCHASING) WITH MULTI-TENANCY ORIENTED USING DISTRIBUTED DATABASE

Student Name : Muhammad Nicko Rahmadano
NRP : 5112100142
Major : Teknik Informatika – FTIf ITS
Supervisor I : Sarwosri, S.Kom, MT.
Supervisor II : Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc.,Ph.D.

ABSTRACT

Enterprise Resource Planning is an application that used to maintain big company to integrating all of the resources, information and activity for the complete business process. ERP has several modules, one of the important modules and what has been done in this final project was Procurement Process or Purchasing. Purchasing is a procurement process at the company , the Purchasing is responsible primarily for the purchase of goods , monitoring purchase of good, and supplier selection. In this final project, have made a procurement process from purchase requisition to purchase order, and proposed a method to determine the best suppliers when buy goods or raw materials based on the purchases that have happened before.

In this final project, ERP Application with distributed database developed, the application are multitenancy oriented, its means that the application can be use by many tenant at the same time. In this procurement process there is a system to request some goods until to buy some goods. TOPSIS used to determine the best supplier.

The implementation results show that the process of purchasing items managed to work properly. For the selection of

the supplier or the supplier analysis has succeed to rank the best suppliers in order to purchase goods can well proceed .

Keywords: Procurement Process, Purchasing, Supplier Analysis, purchase requisition, purchase order, TOPSIS

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
<i>ABSTRACT</i>	ix
DAFTARawaddsss ISI	xiii
DAFTAR GAMBAR.....	xxi
DAFTAR TABEL	xxv
DAFTAR KODE SUMBER.....	xxvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan	2
1.3 Batasan Permasalahan.....	2
1.4 Tujuan3	
1.5 Manfaat	3
1.6 Metodologi	4
1.7 Sistematika Penulisan	5
BAB II DASAR TEORI	7
2.1 Penelitian Terkait (ERP2013)	7
2.2 ERP (<i>Enterprise Resource Planning</i>)	8
2.3 <i>Procurement Process (Purchasing)</i>	8
2.3.1 Purchase Requisition.....	9
2.3.2 Request for Quotation	9
2.3.3 Quotation Order	9
2.3.4 Supplier Analysis	10
2.3.5 Purchase Order.....	10

2.4	Proses Bisnis Umum ERP.....	10
2.4.1	Proses Bisnis Pengadaan (<i>procurement</i>)	11
2.4.2	Proses Bisnis Produksi (<i>production</i>)	11
2.4.3	Proses bisnis penjualan (<i>fullfillment</i>)	12
2.5	TOPSIS	13
2.5.1	Pengertian TOPSIS	13
2.5.2	Langkah-langkah TOPSIS	13
2.6	BPMN (<i>Business Process Model and Notation</i>).....	15
2.7	<i>Data Model</i>	15
2.8	<i>Multi-Tenancy</i>	15
2.9	Framework Yii 2.0	18
2.10	Distributed Database	18
2.10.1	Replikasi	19
2.10.2	Fragmentasi	19
2.11	Database Cluster	19
2.12	MySQL Cluster.....	20
2.13	Arsitektur MySQL Cluster.....	20
2.14	RBAC (<i>Role Based Access Control</i>).....	21
2.15	PHP	22
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		23
3.1	Analisis	23
3.1.1	Analisis Proses Bisnis.....	23
3.1.2	Deskripsi Umum Sistem	35
3.1.3	Spesifikasi Kebutuhan Perangkat Lunak	40
3.1.4	Kasus Penggunaan	42
3.1.5	Analisa Supplier	64

3.2	Perancangan Sistem	70
3.2.1	Perancangan <i>Multitenancy</i>	70
3.2.2	Perancangan RBAC.....	71
3.2.3	Perancangan Basis Data	72
3.2.4	Perancangan Tampilan	77
BAB IV IMPLEMENTASI SISTEM.....		89
4.1	Lingkungan Pengembangan Sistem	89
4.2	Implementasi Distributed Database	89
4.2.1.	Instalasi Data dan SQL node pada node1 dan node2.....	90
4.2.2.	Pemasangan Node Manajemen pada node03	91
4.2.3.	Konfigurasi Manajemen Node.....	92
4.2.4.	Konfigurasi Data dan SQL Node.....	93
4.2.5.	Memulai MySQL Cluster	94
4.3	Implementasi RBAC (Role Base Acces Control)	96
4.3.1.	Membuat Tabel Pengguna	96
4.3.2.	Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna.....	97
4.3.3.	Membuat Modul Admin	99
4.3.4.	Membuat Model Tabel Autentifikasi, <i>Controller</i> dan <i>View</i> Pengguna.....	100
4.3.5.	Menambahkan Kode pada Kelas <i>usercontroller</i>	100
4.4	Implementasi Multi-tenancy	105
4.4.1.	Membuat Halaman Muka Tenant	106
4.4.2.	Mengubah Konfigurasi Database.....	106
4.4.3.	Login Tenant.....	107

4.5	Implementasi Program pada Modul <i>Purchasing</i>	107
4.5.1.	Halaman Utama Modul <i>Purchasing</i>	108
4.5.2.	Melihat Daftar <i>Purchase Requisition</i>	109
4.5.3.	Melihat Rincian <i>Purchase Requisition</i>	110
4.5.4.	Menambah <i>Purchase Requisition</i>	111
4.5.5.	Menyunting <i>Purchase Requisition</i>	112
4.5.6.	Menghapus <i>Purchase Requisition</i>	113
4.5.7.	<i>Convert to RFQ</i>	113
4.5.8.	<i>Convert to PO</i>	114
4.5.9.	Melihat Daftar <i>Request for Quotation</i>	115
4.5.10.	Melihat Rincian <i>Request for Quotation</i>	116
4.5.11.	Menambah <i>Request for Quotation</i>	117
4.5.12.	Menyunting <i>Request for Quotation</i>	117
4.5.13.	Menghapus <i>Request for Quotation</i>	118
4.5.14.	Melihat Daftar <i>Quotation</i>	118
4.5.15.	Menambah <i>Quotation</i>	119
4.5.16.	Melihat Rincian <i>Quotation</i>	120
4.5.17.	Menyunting <i>Quotation</i>	121
4.5.18.	Menghapus <i>Quotation</i>	121
4.5.19.	Melihat Daftar <i>Purchase Order</i>	122
4.5.20.	Melihat Rincian <i>Purchase Order</i>	122
4.5.21.	Menambah <i>Purchase Order</i>	123
4.5.22.	Menyunting <i>Return Order</i>	125
4.5.23.	Menghapus <i>Purchase Order</i>	126
4.5.24.	Melihat Daftar <i>Return Order</i>	127
4.5.25.	Melihat Rincian <i>Return Order</i>	127

4.5.26. Menambah <i>Return Order</i>	128
4.5.27. Menyunting <i>Return Order</i>	130
4.5.28. Menghapus <i>Return Order</i>	131
4.5.29. Menganalisa Supplier	132
4.5.30. Melihat Sejarah Permintaan Barang	138
4.5.31. Melihat Sejarah Harga Barang	140
4.5.32. Melihat Ramalan Kebutuhan Barang	142
4.5.33. Melihat Laporan Pembelian	144
4.6 Implementasi Antarmuka Pengguna	146
4.6.1. Antarmuka Melihat Daftar <i>Purchase Requisition</i>	146
4.6.2. Antarmuka Menambah <i>Purchase Requisition</i> .	147
4.6.3. Antarmuka Menyunting <i>Purchase Requisition</i>	148
4.6.4. Antarmuka Melihat Rincian <i>Purchase Requisition</i>	148
4.6.5. Antarmuka Menghapus <i>Purchase Requisition</i>	149
4.6.6. Antarmuka Melihat Daftar <i>Request for Quotation</i>	150
4.6.7. Antarmuka Menambah <i>Request for Quotation</i>	150
4.6.8. Antarmuka Menyunting <i>Request for Quotation</i>	151
4.6.9. Antarmuka Melihat Rincian <i>Request for Quotation</i>	152
4.6.10. Antarmuka Menghapus <i>Request for Quotation</i>	152
4.6.11. Antarmuka Melihat Daftar <i>Maintain Quotation</i>	153
4.6.12. Antarmuka Menambah <i>Maintain Quotation</i> ..	154

4.6.13.	Antarmuka Menyunting <i>Maintain Quotation</i> .	154
4.6.14.	Antarmuka Melihat Rincian <i>Maintain Quotation</i>	155
4.6.15.	Antarmuka Menghapus <i>Maintain Quotation</i> ..	156
4.6.16.	Antarmuka Melihat Daftar <i>Purchase Order</i> ...	156
4.6.17.	Antarmuka Menambah <i>Purchase Order</i>	157
4.6.18.	Antarmuka Menyunting <i>Purchase Order</i>	157
4.6.19.	Antarmuka Melihat Rincian <i>Purchase Order</i>	158
4.6.20.	Antarmuka Menghapus <i>Purchase Order</i>	159
4.6.21.	Antarmuka Melihat Daftar <i>Return Order</i>	159
4.6.22.	Antarmuka Menambah <i>Return Order</i>	160
4.6.23.	Antarmuka Menyunting <i>Return Order</i>	160
4.6.24.	Antarmuka Melihat Rincian <i>Return Order</i>	161
4.6.25.	Antarmuka Menghapus <i>Return Order</i>	162
4.6.26.	Antarmuka Menganalisa <i>Supplier</i>	162
4.6.27.	Antarmuka Melihat Sejarah Permintaan Barang	163
4.6.28.	Antarmuka Melihat Sejarah Harga Barang	163
4.6.29.	Antarmuka Melihat Ramalan Kebutuhan Barang	164
4.6.30.	Antarmuka Melihat Laporan Pembelian	165
4.6.31.	Antarmuka <i>Dashboard</i> Halaman Depan	165
BAB V PENGUJIAN DAN EVALUASI		166
5.1	Lingkungan Uji Coba.....	168
5.2	Skenario Pengujian	168
5.2.1.	Prosedur Simulasi Pasar	168

5.2.2. Pengujian Basis Data Terdistribusi.....	172
5.2.3. Pengujian RBAC.....	175
5.2.4. Pengujian Multi-Tenant.....	180
5.2.5. Pengujian Fungsionalitas.....	181
5.3 Evaluasi Pengujian.....	197
5.3.1 Evaluasi Pengujian Fungsionalitas Sistem.....	197
BAB VI KESIMPULAN DAN SARAN.....	199
5.4 Kesimpulan.....	199
5.5 Saran	199
DAFTAR PUSTAKA.....	201
DAFTAR ISTILAH.....	203
INDEKS.....	205
LAMPIRAN A – PROSES BISNIS.....	207
6.1 Proses Bisnis Odoo.....	207
6.2 Proses Bisnis InoERP.....	208
6.3 Proses Bisnis Adempiere.....	209
6.4 Proses Bisnis <i>Make-To-Order</i> ERP 2016.....	210
6.5 Proses Bisnis <i>Make-To-Stock</i> ERP 2016.....	211
BIODATA PENULIS.....	213

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1. Proses Bisnis Pengadaan Barang.....	11
Gambar 2.2. Proses Bisnis Produksi.....	12
Gambar 2.3. Proses Bisnis Penjualan.....	12
Gambar 2.4 Separated Database.....	16
Gambar 2.5 Shared Database, Separated Schema.....	17
Gambar 2.6 Shared Database, Shared Schema.....	17
Gambar 3.1 Proses Bisnis ERP Level 0.....	28
Gambar 3.2 Proses Bisnis level 1.....	29
Gambar 3.3. Proses Bisnis Level 2.....	33
Gambar 3.4 Proses Bisnis Level 3.....	34
Gambar 3.5 Rancangan Basis Data Terdistribusi.....	36
Gambar 3.6 Diagram Kasus Penggunaan Aplikasi.....	43
Gambar 3.7 Diagram Aktifitas Menganalisa Supplier.....	52
Gambar 3.8 Diagram Aktifitas Melihat Laporan Pembelian.....	53
Gambar 3.9 Diagram Aktifitas Melihat Sejarah Harga Barang.....	55
Gambar 3.10 Diagram Aktivitas Melihat Ramalan Kebutuhan Barang Mentah.....	56
Gambar 3.11 Diagram Aktivitas Melihat Sejarah Permintaan Barang.....	58
Gambar 3.12 Diagram Aktivitas mengelola Purchase Requisition	59
Gambar 3.13 Diagram Aktifitas mengelola RfQ.....	60
Gambar 3.14 Diagram Aktifitas Mengelola Quotation.....	61
Gambar 3.15 Diagram Aktivitas Mengelola Purchase Order.....	62
Gambar 3.16 Diagram Aktivitas Mengelola Return Order.....	63
Gambar 3.17 Algoritma Analisa Supplier.....	65
Gambar 3.18 Perancangan Multi-tenancy.....	70
Gambar 3.19 Perancangan antarmuka Login.....	71
Gambar 3.20 Perancangan antarmuka add user.....	72
Gambar 3.21 Physical Data Model.....	73
Gambar 3.22 Rancangan Antarmuka Halaman Dashboard.....	78
Gambar 3.23 Rancangan Antarmuka Halaman Purchase Requisition.....	79

Gambar 3.24 Rancangan Antarmuka Halaman Request for Quotation.....	80
Gambar 3.25 Rancangan Antarmuka Halaman Maintain Quotation.....	81
Gambar 3.26 Rancangan Antarmuka Halaman Purchase Order .	82
Gambar 3.27 Rancangan Antarmuka Halaman Return Order	83
Gambar 3.28 Rancangan Antarmuka Halaman Supplier Analysis	84
Gambar 3.29 Rancangan Antarmuka Halaman Item Price History	85
Gambar 3.30 Rancangan Antarmuka Halaman Item Demand History.....	86
Gambar 3.31 Rancangan Antarmuka Halaman Item Forecast ...	87
Gambar 3.32 Rancangan Antarmuka Halaman Purchase Order Report.....	88
Gambar 4.1 Memulai Proses Manajemen Node.....	94
Gambar 4.2 ndbd dan proses mysql server dapat dimulai.....	95
Gambar 4.3 Mysql pada data node aktif.....	95
Gambar 4.4 Data Node Saling Terkoneksi.....	96
Gambar 4.5 Proses Migrasi Tabel Pengguna Berhasil	97
Gambar 4.6 Proses Migrasi Tabel Pengguna Berhasil	98
Gambar 4.7 Proses Migrasi Tabel Pengguna Berhasil	98
Gambar 4.8 Proses Migrasi Tabel Pengguna Berhasil	99
Gambar 4.9 Proses Migrasi Tabel Pengguna Berhasil	99
Gambar 4.10 Tampilan Antarmuka Melihat Daftar Purchase Requisition	147
Gambar 4.11 Tampilan Antarmuka Menambah Purchase Requisition	148
Gambar 4.12 Tampilan Antarmuka Menyunting Purchase Requisition	148
Gambar 4.13 Tampilan Antarmuka Melihat Rincian Purchase Requisition	149
Gambar 4.14 Tampilan Antarmuka Menghapus Purchase Requisition	150

Gambar 4.15 Tampilan Antarmuka Melihat Daftar Request for Quotation.....	150
Gambar 4.16 Tampilan Antarmuka Menambah Request for Quotation.....	151
Gambar 4.17 Tampilan Antarmuka Menyunting <i>Request for Quotation</i>	151
Gambar 4.18 Tampilan Antarmuka Melihat Rincian Request for Quotation.....	152
Gambar 4.19 Tampilan Antarmuka Menghapus Request for Requisition.....	153
Gambar 4.20 Tampilan Antarmuka Melihat Daftar Maintain Quotation.....	153
Gambar 4.21 Tampilan Antarmuka Menambah Maintain Quotation.....	154
Gambar 4.22 Tampilan Antarmuka Menyunting Maintain Quotation.....	155
Gambar 4.23 Tampilan Antarmuka Melihat Rincian <i>Maintain Quotation</i>	155
Gambar 4.24 Tampilan Antarmuka Menghapus Maintain Quotation.....	156
Gambar 4.25 Tampilan Antarmuka Melihat Daftar <i>Purchase Order</i>	156
Gambar 4.26 Tampilan Antarmuka Menambah Purchase Order.....	157
Gambar 4.27 Tampilan Antarmuka Menyunting Purchase Order.....	158
Gambar 4.28 Tampilan Antarmuka Melihat Rincian <i>Purchase Order</i>	158
Gambar 4.29 Tampilan Antarmuka Menghapus Purchase Order.....	159
Gambar 4.30 Tampilan Antarmuka Melihat Daftar Return Order.....	159
Gambar 4.31 Tampilan Antarmuka Menambah Return Order..	160
Gambar 4.32 Tampilan Antarmuka Menyunting Return Order	161

Gambar 4.33 Tampilan Antarmuka Melihat Rincian <i>Return Order</i>	161
Gambar 4.34 Tampilan Antarmuka Menghapus Return Order.	162
Gambar 4.35 Tampilan Antarmuka Hasil Menganalisa <i>Supplier</i>	162
Gambar 4.36 Tampilan Antarmuka Melihat Sejarah Permintaan Barang	163
Gambar 4.37 Tampilan Antarmuka Melihat Sejarah Harga Barang	164
Gambar 4.38 Tampilan Antarmuka Melihat Ramalan Kebutuhan Barang	164
Gambar 4.39 Tampilan Antarmuka Melihat Laporan Pembelian	165
Gambar 4.40 Tampilan Antarmuka Dashboard.....	166
Gambar 5.1 Pengujian fitur Replikasi pada sistem.....	173
Gambar 5.2 Pengujian Fitur Replikasi pada Database Server 1	174
Gambar 5.3 Pengujian Fitur Replikasi pada Databse Server 2..	174
Gambar 5.4 Pengujian Fitur High-Availability pada Sistem.....	175
Gambar 5.5 Pengujian Fitur High-Availability pada Database Server 1	175
Gambar 5.6 Pengujian Fitur High-Availability pada Database Server 2	175
Gambar 5.7 Proses login admin	178
Gambar 5.8 Tampilan awal setelah login berhasil dilakukan....	178
Gambar 5.9 Proses admin menambahkan user baru.....	179
Gambar 5.10 Proses menyunting data user oleh admin.....	179
Gambar 5.11 Proses penghapusan user oleh admin	179
Gambar 5.12 Hasil dari pengujian multi-tenant	180
Gambar 5.13 Hasil Supplier Analysis pada Aplikasi ERP	193
Gambar 0.1 Proses Bisnis Odoo	207
Gambar 0.2. Proses Bisnis InoERP	208
Gambar 0.3. Proses Bisnis Adempiere	209
Gambar 0.4. Proses Bisnis Make-To-Order ERP 2016	210
Gambar 0.5. Proses Bisnis Make-To-Stock ERP 2016	211

DAFTAR TABEL

Tabel 2.1 Perbandingan dengan ERP Sebelumnya	7
Tabel 3.1 Perbandingan Proses Bisnis ERP 2016, Odoo, InoERP, dan Adempire	24
Tabel 3.2. Deskripsi Proses Bisnis Level 1	30
Tabel 3.3 Deskripsi Proses Bisnis Level 2	30
Tabel 3.4 Daftar kebutuhan fungsional perangkat lunak	40
Tabel 3.5 Pengguna Aplikasi	42
Tabel 3.6 Kasus Penggunaan	43
Tabel 3.7 Kasus Penggunaan Mengelola Requisition	44
Tabel 3.8 Kasus Penggunaan Mengelola RfQ	45
Tabel 3.9 Kasus Penggunaan Mengelola Quotation	47
Tabel 3.10 Kasus Penggunaan Purchase Order	48
Tabel 3.11 Kasus Penggunaan Return Order	49
Tabel 3.12 Kasus Penggunaan Menganalisa Supplier	51
Tabel 3.13 Kasus Penggunaan Melihat Laporan Pembelian	52
Tabel 3.14 Kasus Penggunaan Melihat Sejarah Harga Barang ...	54
Tabel 3.15 Kasus Penggunaan Melihat Ramalan Kebutuhan Barang Mentah	55
Tabel 3.16 Kasus Penggunaan Melihat Sejarah Kebutuhan Barang Mentah	57
Tabel 3.17 <i>Keterangan Bobot</i>	64
Tabel 3.18 Gabungan data Purchase Order dan Good Receipt ...	66
Tabel 3.19 Rata-rata Tiap Supplier	67
Tabel 3.20 Matriks Ternormalisasi	67
Tabel 3.21 Bobot setiap kriteria	68
Tabel 3.22 Matriks Terbobot	68
Tabel 3.23 Solusi Ideal Positif dan Negatif	68
Tabel 3.24 Separation dari Positif	68
Tabel 3.25 Separation dari Negatif	69
Tabel 3.26 Supplier's Score	69
Tabel 3.27 Supplier's Rank	69
Tabel 3.28 Keterangan PDM	75
Tabel 5.1. Daftar Barang Jadi	169

Tabel 5.2. Daftar Aset	169
Tabel 5.3. Daftar Bahan Baku	170
Tabel 5.4. Daftar Material	171
Tabel 5.5. Perencanaan Keuangan Perusahaan	172
Tabel 5.6 Pengujian fitur mengelola Role Based Access Control (RBAC)	176
Tabel 5.7 Pengujian Fitur Multi-Tenant.....	180
Tabel 5.8 Pengujian Fitur Mengelola Purchase Requisition	181
Tabel 5.9 Pengujian Fitur Mengelola Request for Quotation.....	183
Tabel 5.10 Pengujian Fitur Mengelola Quotation	185
Tabel 5.11 Pengujian Fitur Mengelola Purchase Order	187
Tabel 5.12 Pengujian Fitur Mengelola Return Order	190
Tabel 5.13 Pengujian Fitur Menganalisa Supplier	192
Tabel 5.14 Perhitungan Supplier Analysis	193
Tabel 5.15 Pengujian Fitur Melihar laporan pembelian.....	193
Tabel 5.16 Pengujian Fitur Melihat Sejarah Harga Barang.....	194
Tabel 5.17 Pengujian Fitur Melihat Ramalan Kebutuhan Barang Mentah.....	195
Tabel 5.18 Pengujian Fitur Melihat Sejarah Kebutuhan Barang Mentah.....	196
Tabel 5.19. Evaluasi Pengujian Fungsionalitas Sistem	197

DAFTAR KODE SUMBER

Kode Sumber 4.1 Membuat grup MySQL pengguna baru dan menambah user MySQL.....	90
Kode Sumber 4.2 Mengubah lokasi ke dalam direktori yang berisi file yang telah didownload, mengubah arsip dan menciptakan symlink ke dalam direktori mysql.	90
Kode Sumber 4.3 Mengubah lokasi ke direktori mysql.	91
Kode Sumber 4.4 Mengatur izin yang diperlukan oleh server MSQL.....	91
Kode Sumber 4.5 Menyalin script startup MySQL ke direktori yang sesuai, mengubah menjadi executable, dan memulai ketika sistem beroperasi	91
Kode Sumber 4.6 Mengubah lokasi ke dalam direktori / var / tmp direktori, mengekstrak ndb_mgm dan ndb_mgmd dari arsip ke direktori yang sesuai seperti / usr / local / bin.....	92
Kode Sumber 4.7 Mengubah lokasi ke dalam direktori tempat file disalin, kemudian dieksekusi.....	92
Kode Sumber 4.8 Membuat direktori tempat file konfigurasi ditemukan kemudian membuat file itu sendiri	92
Kode Sumber 4.9 Mengatur file “config.ini”	93
Kode Sumber 4.10 Data dan SQL Node.....	93
Kode Sumber 4.11 Memulai proses manajemen node	94
Kode Sumber 4.12 Memulai proses ndbd dan proses mysql server	94
Kode Sumber 4.13 Memulai proses ndbd dan proses mysql server	95
Kode Sumber 4.14 Generate tabel RBAC	97
Kode Sumber 4.15 Pembuatan modul admin dan konfigurasi autentifikasi	100
Kode Sumber 4.16 Kode Fungsi Tabel AuthItem	104
Kode Sumber 4.17 Kode Fungsi Tabel ItemChild	105
Kode Sumber 4.18 Kode Fungsi Tabel Auth Assignment	105
Kode Sumber 4.19 Pembuatan Halaman Muka Tenant.....	106
Kode Sumber 4.20 Penambahan Database Tenant Baru	107

Kode Sumber 4.21 Login Tenant	107
Kode Sumber 4.22. Halaman Utama Modul Purchasing.....	109
Kode Sumber 4.23. Melihat Daftar Purchase Requisition.....	110
Kode Sumber 4.24. Melihat Rincian purchase requisition.....	111
Kode Sumber 4.25 Menambah Purchase Requisition	112
Kode Sumber 4.26 Menyunting Purchase Requisition.....	113
Kode Sumber 4.27 Menghapus Purchase Requisition.....	113
Kode Sumber 4.28 Convert to RFQ	114
Kode Sumber 4.29 Convert to PO.....	115
Kode Sumber 4.30. Melihat Daftar Request for Quotation.....	116
Kode Sumber 4.31 Melihat Rincian request for quotation.....	116
Kode Sumber 4.32 Menambah Request for Quotation.....	117
Kode Sumber 4.33 Menyunting Request for Quotation.....	118
Kode Sumber 4.34 Menghapus Request for Quotation.....	118
Kode Sumber 4.35. Melihat Daftar Quotation	119
Kode Sumber 4.36 Menambah Quotation	120
Kode Sumber 4.37 Melihat Rincian Quotation	121
Kode Sumber 4.38 Menyunting Quotation	121
Kode Sumber 4.39 Menghapus Quotation	122
Kode Sumber 4.40. Melihat Daftar Purchase Order.....	122
Kode Sumber 4.41 Melihat Rincian Purchase Order	123
Kode Sumber 4.42 Menambah Purchase Order	125
Kode Sumber 4.43 Menyunting Purchase Order.....	126
Kode Sumber 4.44 Menghapus Purchase Order.....	127
Kode Sumber 4.45. Melihat Daftar Return Order	127
Kode Sumber 4.46 Melihat Rincian Return Order.....	128
Kode Sumber 4.47 Menambah <i>Return Order</i>	130
Kode Sumber 4.48 Menyunting <i>Return Order</i>	131
Kode Sumber 4.49 Menghapus <i>Return Order</i>	131
Kode Sumber 4.50 Menganalisa supplier.....	137
Kode Sumber 4.51 Melihat Sejarah Permintaan Barang.....	140
Kode Sumber 4.52 Melihat Sejarah Harga Barang	142
Kode Sumber 4.53 Melihat Ramalan Kebutuhan Barang	144
Kode Sumber 4.54 Melihat Laporan Pembelian	146

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

Enterprise Resource Planning adalah sebuah sistem yang digunakan untuk mengelola sebuah perusahaan besar yang dirancang untuk mengkoordinasikan semua sumber daya, informasi dan aktifitas yang diperlukan untuk proses bisnis lengkap.

Pada saat ini *ERP* masih diimplementasikan oleh perusahaan-perusahaan besar saja, perusahaan-perusahaan kecil belum banyak yang menggunakan *ERP* karena penggunaannya dianggap terlalu merepotkan, oleh karena itu dibutuhkan sebuah *service ERP* yang bisa menyelesaikan masalah tersebut. Sistem ERP ini juga akan digunakan oleh banyak perusahaan, oleh karena itu diperlukan pendekatan *Multi-Tenancy* agar sistem ERP ini bisa digunakan oleh banyak perusahaan secara bersamaan.

Pada ERP terdapat banyak sekali modul-modul yang dibuat dan diintegrasikan, antara lain adalah *Finance / Accounting, Sales and Delivery, Inventory, Production and Planning, Human Resource Management, Asset Management, Purchasing* dan lainnya. Pada tugas akhir ini, akan dibangun suatu modul aplikasi *Procurement Process*.

Procurement Process atau proses pengadaan adalah sistem yang mengatur proses pengadaan barang dari *inventory request* sampai *inventory and invoice receipt*. Modul ini berfungsi agar pencatatan saat proses pengadaan barang lengkap dan benar, selain itu pada proses pengadaan barang, banyak sekali terjadi kecurangan oleh karena itu dibutuhkan sistem pengadaan barang yang lengkap sehingga tidak terjadi kecurangan.

Pada pengembangan sebelumnya (aplikasi ERP 2013), dikembangkan sebuah program *procurement* pada domain *supplier*

relationship management (SRM) yang tidak begitu berfokus pada *procurement process*, pada domain SRM tidak ada sub modul yang membahas tentang *request for quotation*, *maintain quotation*, dan *supplier analysis* secara lengkap. Aplikasi tersebut menggunakan arsitektur berorientasi *service* (SOA) dengan sistem *Model-View-Controller* (MVC) dan *Workflow* untuk .NET.

Pada tugas akhir ini, lebih mendetailkan proses-proses pada pengadaan barang (*procurement*). Modul *Procurement Process* dikembangkan untuk lebih mendetailkan proses-proses tersebut dimana meliputi submodul *Purchase Requisition*, *Request for Quotation*, *Maintain Quotation*, *Purchase Order*, dan *Supplier Analysis* dengan orientasi basis data terdistribusi. Aplikasi ini menggunakan arsitektur *multitenancy* dengan sistem *Model-View-Controller* (MVC) dan *Workflow* untuk PHP.

1.2 Rumusan Permasalahan

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

1. Bagaimana membuat aplikasi ERP untuk sub-domain *Procurement Process* suatu perusahaan untuk menggantikan proses pengolahan data manual dan sebagai penyedia informasi terkait *Purchasing* sebagai usaha peningkatan kinerja perusahaan?
2. Bagaimana cara menentukan *supplier* terbaik?
3. Bagaimana cara menambahkan dan mengaplikasikan sistem *Multi-Tenancy* pada modul *Procurement Process*?
4. Bagaimana menangani kegagalan sistem pada sebuah *database server* sehingga kinerja *tenant* lainnya tidak terganggu?

1.3 Batasan Permasalahan

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

1. Bahasa pemrograman yang digunakan adalah PHP.
2. Sistem Basis Data yang digunakan adalah MySQL Cluster.
3. *Platform* yang digunakan adalah Yii2 dengan teknologi :

a. *Eloquent*

- b. *Modularity*
 - c. *RBAC (Role Base Access Control)*
4. Aplikasi yang dikembangkan merupakan aplikasi *web*.
 5. Sistem ERP yang dibangun dalam aspek *Multitenancy* dengan *distributed database*.
 6. Modul yang akan diintegrasikan adalah Procurement Process dengan beberapa sub-modul, diantaranya adalah :
 - a. *Master Data Purchasing*
 - b. *Purchase Requisition*
 - c. *Request for Quotations*
 - d. *Maintain Quotation*
 - e. *Purchase Order*
 - f. *Return Order*
 - g. *Supplier Analysis*
 - h. *Purchasing Report*

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat aplikasi yang dapat mempermudah proses pengadaan barang (*purchasing process*) pada perusahaan untuk menggantikan proses pengolahan data secara manual.
2. Menentukan supplier terbaik pada proses pengadaan barang.
3. Mengimplementasikan *multitenancy* pada modul *Procurement Process*.
4. Mengatasi kelemahan *separated database* pada pengembangan sebelumnya dalam hal *availability* untuk arsitektur yang *multitenancy*.
5. Membuat proses bisnis dari modul *Procurement Process* pada sistem ERP

1.5 Manfaat

Tugas akhir ini diharapkan dapat menyediakan sistem pengadaan (*purchasing*) pada ERP (*Enterprise Resource Planning*) sehingga dapat mengatur sistem pembelian secara baik mulai dari

pembuatan *purchase requisition*, *request for quotation*, *maintain quotation*, *supplier Analysis* (memilih pemasok), pembuatan *Purchase Order*, dan sistem ini diharapkan dapat digunakan oleh banyak perusahaan dalam satu waktu.

1.6 Metodologi

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

1. Studi literatur

Pada tahapan ini akan dilakukan studi literatur yang berfokus pada pembuatan aplikasi PHP dengan *framework* Yii 2.0 sesuai dengan proses bisnis model dari ERP, implementasi *multitenancy* dengan *distributed database* serta modul yang akan digunakan seperti *Procurement Process*, serta modul yang berkaitan dengan *Procurement Process* seperti *Inventory and warehouse Management*, *Production Planning*.

2. Analisis dan Perancangan Sistem

Pada tahap analisis dan desain dilakukan perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang digunakan. Tahap ini mendefinisikan alur dari implementasi, perancangan database, perancangan *prototype system*. Desain perangkat lunak akan menggunakan sistem rancang MVC (*Model-View-Controller*) menggunakan *framework* Yii 2.0.

3. Implementasi Perangkat Lunak

Tahap implementasi dilakukan dengan cara menerapkan algoritma-algoritma pada sistem dibangun menggunakan bahasa pemrograman PHP dengan Framework Yii 2.0. Pengolahan database menggunakan database MySQL Cluster.

4. Pengujian dan evaluasi

Pengujian dilakukan oleh kelas *sistem enterprise* 2016, kelas *sistem enterprise* dibagi menjadi 5 kelompok dan masing-masing kelompok terdiri dari 9 orang atau lebih dan mempunyai tanggung jawab masing-masing pada setiap modul ERP.

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 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 IV Implementasi

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

Bab V 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 VI Kesimpulan dan Saran

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 bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

BAB II DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir. Teori-teori tersebut meliputi ERP, Procurement Process, BPMN, Data Model, Multy-Tenancy, TOPSIS, *Supplier Analysis*

2.1 Penelitian Terkait (ERP2013)

Tugas Akhir ini merupakan pengembangan riset berkelanjutan dari riset atau penelitian tentang ERP. Pada pengembangan sebelumnya, telah dikembangkan aplikasi ERP yang diberi nama ERP 2013, aplikasi tersebut menggunakan *Service Oriented Architecture* (SOA) dengan sistem *Model-View-Controller* (MVC) dan *workflow* untuk .NET, pada ERP tersebut terdapat modul yang sama seperti tugas akhir ini yaitu pada domain *supplier chain management* [1]. Didalam modul tersebut, terdapat sub-modul Purchase Requisition dan Purchase Order tetapi tidak menangani permintaan quotation, tidak menangani quotation yang didapatkan dari supplier, dan tidak ada analisa terhadap supplier, pada ERP 2013 juga tidak terdapat laporan yang lengkap, yang berkaitan dengan pembelian. Pada tugas akhir ini menambahkan sub modul untuk mengelola quotation yaitu request for quotation dan maintain quotation, serta terdapat supplier analysis untuk mengevaluasi supplier berdasarkan transaksi-transaksi yang sudah terjadi sebelumnya, serta return order yang menangani pengembalian barang, dan pada ERP 2016 terdapat report-report lengkap yang berkaitan dengan pembelian barang.

Tabel 2.1 Perbandingan dengan ERP Sebelumnya

ERP 2013	ERP 2016
Tidak ada fitur untuk mengelola Quotation	Terdapat fitur untuk mengelola Quotation yaitu pada Request for Quotation dan Maintain Quotation
Tidak terdapat analisis supplier.	Terdapat analisis supplier

	berupa perankingan supplier
Tidak terdapat report yang berkaitan dengan pembelian.	Terdapat report-report yang berkaitan dengan pembelian.
Menggunakan <i>Service Oriented Architecture</i> (SOA)	Menggunakan <i>Distributed Database</i>

2.2 ERP (*Enterprise Resource Planning*)

Enterprise Resource Planning (ERP) adalah sebuah aplikasi manajemen bisnis yang memudahkan pengelolaan bisnis secara terintegrasi [2]. Akhir-akhir ini perusahaan memperhitungkan ERP sebagai bagian penting dari suatu organisasi. ERP mengintegrasikan berbagai sistem informasi di dalam perusahaan. ERP mampu menyediakan informasi secara *real-time* tentang proses bisnis inti perusahaan seperti produksi, *order processing*, *purchasing*, dan *inventory management*. ERP memantau sumber daya perusahaan seperti uang, bahan mentah, kapasitas produksi, dan pegawai. ERP dipergunakan untuk mengelola status komitmen bisnis yang dibuat misalnya *customer orders*, *purchase orders*, and *employee payroll*. Dengan ERP data dapat dimasukkan ke dalam sistem dari berbagai departemen (*manufacturing*, *purchasing*, *sales*, *accounting*, dsb.) ERP mengelola data secara sentral sehingga hanya perlu sekali saja memasukkan data untuk selanjutnya dimanfaatkan oleh berbagai proses bisnis lainnya. ERP melancarkan arus informasi yang berjalan lintas fungsi di dalam organisasi sampai kepada hubungan dengan para *stake holder* di luar perusahaan.

2.3 *Procurement Process* (*Purchasing*)

Procurement Process merupakan proses yang mengatur sistem pengadaan barang dalam ERP. *Procurement Process* tidak hanya berfokus pada pembelian barang namun *Procurement Process* juga meliputi permintaan (*requisition*), meminta penawaran barang atau jasa yang dibutuhkan kepada pemasok sebagai penyedia barang dan jasa, menerima dan menyeleksi surat penawaran dari pemasok, dan menganalisa pemasok agar mendapatkan pemasok yang terbaik dalam pembelian barang [2].

Procurement Process meliputi pembuatan *purchase requisition*, *request for quotation*, *maintain quotation*, *vendor* atau *supplier analysis*, *Purchase Order*, dan segala *report* mengenai *purchasing*.

2.3.1 Purchase Requisition

Purchase Requisition adalah sarana untuk membuat permintaan pembelian, agar pihak *purchasing/buyer* dapat melakukan proses pengadaan barang yang di minta. *Purchase Requisition* merupakan awal dari sebuah proses pengadaan barang. Barang yang diminta adalah barang yang terdaftar pada *Item Master*. *Purchase Requisition* berisi tentang jenis barang yang diminta, deskripsi atau detail barang yang diminta, dan tanggal barang tersebut dibutuhkan.

2.3.2 Request for Quotation

Request for Quotation (RfQ) adalah dokumen yang berisi permintaan penawaran secara resmi kepada supplier. RfQ atau yang lebih dikenal dengan surat permintaan penawaran berisi tentang detail barang yang diminta, jumlahnya, alamat pengiriman serta tanggal batas pengiriman quotation. RfQ bertujuan untuk mendapatkan *quotation order* atau surat penawaran dari supplier secara resmi.

2.3.3 Quotation Order

Quotation Order adalah surat penawaran barang yang diberikan oleh supplier, berisi tentang detail barang, harga barang, diskon, biaya pengiriman, dan pengeluaran untuk pajak. *Quotation* yang didapatkan dari supplier akan dianalisa oleh bagian *purchasing* dan menentukan apakah akan menerima atau menolak penawaran tersebut. *Quotation Order* adalah patokan untuk membuat *purchase order*.

2.3.4 Supplier Analysis

Supplier Analysis adalah sebuah analisa untuk menentukan rekomendasi supplier untuk pembelian barang berdasarkan dari transaksi pembelian dan penerimaan barang yang telah dilakukan sebelumnya. Analisa ini bertujuan agar pihak purchasing tidak salah dalam memilih supplier dan mendapatkan barang dengan kualitas, dan harga yang terbaik.

2.3.5 Purchase Order

Purchase order atau yang lebih dikenal dengan PO merupakan salah satu jenis surat bisnis yang sering ditemui. Saat purchasing melakukan *order* atau pemesanan kepada perusahaan rekanan atau *supplier*, *purchasing* wajib untuk mengirimkan baik via fax, email, maupun pos. Surat ini akan digunakan *supplier* untuk mengetahui secara detail barang-barang apa saja yang dipesan. Ini berguna untuk mengatasi kesalahan dalam memproduksi pesanan. Selain itu, Surat *Purchase Order* juga akan digunakan supplier (dan juga *purchasing*) sebagai file dan bagi bagian *accounting/keuangan/pembukuan*, Surat *Purchase Order* ini akan dijadikan bukti transaksi laporan keuangan.

Di dalam *Purchase Order* harus memuat dengan jelas jenis barang yang dipesan, jumlah yang dipesan, spesifikasi/permintaan khusus bila ada, tanggal pengiriman, serta yang tidak kalah pentingnya adalah harus mencantumkan nomer PO.

2.4 Proses Bisnis Umum ERP

Terdapat beberapa proses bisnis utama yang terdapat pada ERP yang berfungsi sebagai aliran aplikasi ERP dalam sebuah perusahaan. Beberapa proses bisnis utama, antara lain pengadaan (*procurement*), produksi (*production*), penjualan (*sales/fullfillment*) [2].

2.4.1 Proses Bisnis Pengadaan (*procurement*)

Proses bisnis pengadaan merupakan proses dimana suatu perusahaan membeli barang untuk diproduksi. Proses bisnis ini merupakan awal dalam proses *Make-to-Stock*, sebuah proses dimana perusahaan membuat barang sebelum adanya pembelian. Proses business ini dimulai dengan perusahaan membuat dokumen *purchase requisition*. Kemudian dokumen *purchase requisition* ini setelah disetujui akan menjadi dokumen *purchase order* atau perintah pembelian barang. Setelah barang sudah dibeli, maka muncul dokumen *receive material* yang menandakan barang sudah diterima. Kemudian perusahaan mendapatkan tagihan atau *invoice* untuk dibayar. Terakhir perusahaan membayarkan sejumlah uang untuk melunasi tagihan tersebut. Proses bisnis *procurement* ditunjukkan pada Gambar 2.1.



Gambar 2.1. Proses Bisnis Pengadaan Barang

2.4.2 Proses Bisnis Produksi (*production*)

Proses bisnis produksi merupakan proses perusahaan memproduksi barang. Untuk membuat suatu barang, perusahaan harus mengikuti proses bisnis ini. Proses bisnis ini dimulai dengan perusahaan membuat dokumen *request production*. Setelah produksi disetujui maka akan muncul dokumen *authorized production*. Produksi akan dapat berjalan ketika bahan mentah sudah tersedia dan dapat dikeluarkan dari gudang. Proses produksi dimulai ketika dokumen *create product* sudah muncul. Setelah barang sudah jadi maka, barang akan dipindahkan dari tempat produksi ke gudang dengan adanya dokumen *receive*

finished goods. Proses bisnis produksi ditunjukkan pada Gambar 2.2.



Gambar 2.2. Proses Bisnis Produksi

2.4.3 Proses bisnis penjualan (*fullfillment*)

Proses bisnis penjualan merupakan proses perusahaan menjual barang kepada pelanggan. Proses ini merupakan proses terpenting dalam simulasi bisnis. Proses bisnis ini dimulai dengan perusahaan membuat dokumen *sales order* untuk mengelola siapa yang membeli barang, barang yang dibeli, jumlah barang, harga barang, dan total harga. Kemudian proses dilanjutkan dengan mempersiapkan pengiriman barang. Barang akan dikirim dan muncul dokumen *shipment*. Setelah barang diterima oleh *customer*, maka perusahaan mengirimkan tagihan atau *invoice* kepada pelanggan. Proses bisnis ini berakhir ketika pelanggan sudah membayar tagihan tersebut. Proses bisnis penjualan ditunjukkan pada Gambar 2.3.



Gambar 2.3. Proses Bisnis Penjualan

2.5 TOPSIS

2.5.1 Pengertian TOPSIS

Metode TOPSIS adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang pada tahun 1981. Metode ini merupakan salah satu metode yang banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. TOPSIS memiliki konsep dimana alternatif yang terpilih merupakan alternatif terbaik yang memiliki jarak terpendek dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Semakin banyaknya faktor yang harus dipertimbangkan dalam proses pengambilan keputusan, maka semakin relatif sulit juga untuk mengambil keputusan terhadap suatu permasalahan. Apalagi jika upaya pengambilan keputusan dari suatu permasalahan tertentu, selain mempertimbangkan berbagai faktor/kriteria yang beragam, juga melibatkan beberapa orang pengambil keputusan. Permasalahan yang demikian dikenal dengan permasalahan *multiple criteria decision making* (MCDM). Dengan kata lain, MCDM juga dapat disebut sebagai suatu pengambilan keputusan untuk memilih alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Metode TOPSIS digunakan sebagai suatu upaya untuk menyelesaikan permasalahan *multiple criteria decision making*. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan [3].

2.5.2 Langkah-langkah TOPSIS

Langkah-langkah metode TOPSIS adalah sebagai berikut:

1. Menggambarkan alternatif (m) dan kriteria (n) ke dalam sebuah matriks x_{ij} , dimana x_{ij} adalah pengukuran pilihan dari alternatif ke- i dan kriteria ke- j . Matriks ini

kemudian dinormalisasi (R) dengan menghitung nilai r_{ij} yang didapat dari Persamaan 2.1 dan 2.2.

$$r_{ij} = \frac{\frac{1}{x_{ij}}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}, \quad i = 1,2,3,\dots,m; j = 1,2,3,\dots,n \quad (2.1)$$

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}, \quad i = 1,2,3,\dots,m; j = 1,2,3,\dots,n \quad (2.2)$$

2. Membuat pembobotan pada matriks yang telah dinormalisasi. Setelah dinormalisasi, setiap kolom pada matriks R (nilai r_{ij}) dikalikan dengan bobot (w_j) untuk menghasilkan matriks pada Persamaan 2.3.

$$v_{ij} = r_{ij} * w_j, \quad i = 1,2,3,\dots,m; j = 1,2,3,\dots,n \quad (2.3)$$

3. Menentukan nilai solusi ideal positif dan solusi ideal negatif. Solusi ideal dinotasikan A^* , sedangkan solusi ideal negatif dinotasikan A^- . Perhitungan untuk menentukan solusi ideal dapat dilihat pada Persamaan 2.4.

$$\begin{aligned} A^* &= \{v_1^*, \dots, v_n^*\} \text{ untuk solusi positif} \\ A^- &= \{v_1^-, \dots, v_n^-\} \text{ untuk solusi negatif} \end{aligned} \quad (2.4)$$

4. Menentukan *separation measure* (d_i). *Separation measure* ini merupakan pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negatif. Jarak ke solusi ideal positif didapat dari Persamaan 2.5 dan jarak ke solusi ideal negatif didapat dari Persamaan 2.6.

$$d_{ij}^* = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^*)^2}, \quad i = 1, \dots, m \quad (2.5)$$

$$d_{ij}^- = \sqrt{\sum_{i=1}^n (v_{ij} - v_j)^2}, i = 1, \dots, m \quad (2.6)$$

5. Menghitung nilai preferensi (CC_i) untuk setiap alternatif. Untuk menentukan ranking setiap alternatif yang ada maka perlu dihitung terlebih dahulu nilai preferensi dari tiap alternatif. Perhitungan nilai preferensi dapat dilihat melalui Persamaan 2.7.

$$CC_i = \frac{d_i^-}{d_i^* + d_i^-}, I = 1, \dots, m \quad (2.7)$$

Setelah mendapatkan nilai CC_i , maka alternatif dapat diranking berdasarkan urutan CC_i . Dari hasil perankingan ini dapat dilihat alternatif terbaik yaitu alternatif yang memiliki jarak terpendek dari solusi ideal dan berjarak terjauh dari solusi ideal negatif.

2.6 BPMN (*Business Process Model and Notation*)

Tujuan utama dari BPMN adalah untuk menyediakan suatu notasi yang mudah dipahami oleh semua pengguna bisnis, mulai dari bisnis analis yang membuat draft awal dari proses, para pengembang teknis yang bertanggung jawab untuk menerapkan teknologi yang akan melakukan proses-proses tersebut, hingga kepada orang-orang bisnis yang akan mengelola dan memantau proses mereka [4].

2.7 Data Model

Model data adalah sekumpulan konsep-konsep untuk menerangkan data, hubungan-hubungan antara data dan batasan-batasan data yang terintegrasi di dalam suatu organisasi.

2.8 Multi-Tenancy

Multitenancy mengacu pada arsitektur perangkat lunak di mana salah satu contoh dari perangkat lunak yang berjalan pada server dan menyajikan beberapa *tenant* [5]. *Tenant* adalah sekelompok pengguna yang berbagi akses umum dengan hak

akses khusus untuk contoh perangkat lunak. Dengan arsitektur *multitenant*, aplikasi perangkat lunak ini dirancang untuk memberikan setiap tenant sebuah bagian untuk berdedikasi contoh termasuk data, konfigurasi, manajemen pengguna, fungsi individu tenant dan sifat non-fungsional. *Multitenancy* bertentangan dengan arsitektur *multi-instance* misalnya, di mana contoh perangkat lunak terpisah beroperasi atas nama penyewa yang berbeda.

Arsitektur data *Multitenancy* digunakan karena dinilai memadai dan cukup aman dalam mengatasi masalah kurang kepercayaan *tenant* untuk menyerahkan kontrol data bisnis *tenant* kepada pihak ketiga. Terdapat 3 jenis *Multitenancy*, antara lain:

a. ***Separated Database***

Separated Database adalah data setiap *tenant* disimpan pada *database* yang terpisah dengan *tenant* lain. Keuntungan arsitektur ini adalah mudah untuk mengatur kembali model data aplikasi yang digunakan [5]. Tetapi memerlukan biaya yang cukup tinggi untuk menjaga peralatan *server* dan juga *back up* data dari setiap *tenant*. Model arsitektur *Separated Database* ditunjukkan pada Gambar 2.4.

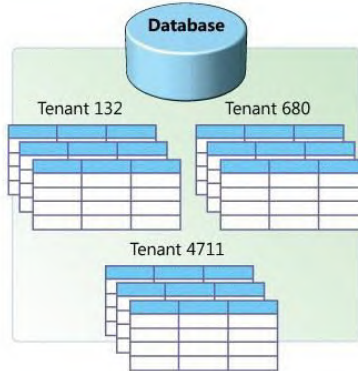


Gambar 2.4 Separated Database

b. ***Shared Database, Separated Schema***

Shared Database, Separated Schema data setiap *tenant* disimpan pada satu *database* tetapi pada *schema* yang terpisah dengan *tenant* lain [5]. Keuntungan dari arsitektur ini adalah mudah untuk digunakan karena tabel yang dibentuk pada awalnya merupakan tabel standar, dan selanjutnya dapat

diubah sesuai keinginan tenant. Akan tetapi apabila terjadi kegagalan maka perlu dilakukan perbaikan untuk semua tabel yang ada dalam *database*. Model arsitektur *Shared Database, Separated Schema* ditunjukkan pada Gambar 2.5



Gambar 2.5 Shared Database, Separated Schema

c. ***Shared Database, Shared Schema***

Shared Database, Shared Schema adalah data setiap *tenant* disimpan pada satu *database* dan satu *schema* [5]. Keuntungan dari arsitektur ini adalah tidak memerlukan biaya yang tinggi, akan tetapi apabila terjadi kegagalan maka perlu dilakukan perbaikan untuk semua tabel yang ada dalam *database*. Model arsitektur *Shared Database, Shared Schema* ditunjukkan pada Gambar 2.6.

TenantID	CustName	Address
4	TenantID	ProductID
1	4	ProductName
6	1	Shipment
4	4711	Date
6	132	324965
4	680	2006-02-21
4	680	115468
	4711	2006-04-08
	4711	654109
	4711	2006-03-27
	4711	324956
	4711	2006-02-23

Gambar 2.6 Shared Database, Shared Schema

2.9 Framework Yii 2.0

Yii adalah *framework* (kerangka kerja) PHP berbasis-komponen, berkinerja tinggi untuk pengembangan aplikasi Web berskala-besar. Yii merupakan Sebuah *framework* untuk bahasa pemrograman PHP dimana *framework* ini telah didukung oleh aspek *modularity* dan *eloquent database* [6]. Nama Yii (dieja sebagai /i:/) singkatan dari "Yes It Is!".

Yii adalah *framework* pemrograman umum *web* yang bisa dipakai untuk mengembangkan semua jenis aplikasi *web*. Dikarenakan sangat ringan dan dilengkapi dengan mekanisme *caching* yang canggih, Yii sangat cocok untuk pengembangan aplikasi dengan lalu lintas-tinggi, seperti portal, forum, sistem manajemen konten (CMS), sistem *e-commerce*, dan lain-lain.

Seperti kebanyakan PHP *framework*, Yii adalah MVC *framework*. Yii melampaui *framework* PHP lain dalam hal efisiensi, kekayaan-fitur, dan kejelasan dokumentasi. Yii didesain dengan hati-hati dari awal agar sesuai untuk pengembangan aplikasi web secara serius. Yii bukan berasal dari produk pada beberapa proyek maupun konglomerasi pekerjaan pihak-ketiga. Yii adalah hasil dari pengalaman kaya para pembuat pada pengembangan aplikasi web dan investigasi *framework* pemrograman web dan aplikasi yang paling populer.

2.10 Distributed Database

Basis data terdistribusi (BDT) adalah suatu basis data yang memiliki kontrol terpusat pada *server* manajemen dengan distribusi penyimpanan data yang tersebar di beberapa *server* data [7]. BDT diterapkan dalam rangka untuk menjaga prinsip *high availability* pada aplikasi ERP. DBT memungkinkan aplikasi tetap berjalan jika terdapat sebuah *server* basis data mengalami *system failure*. Terdapat 2 mekanisme utama dalam BDT, yaitu replikasi dan fragmentasi.

2.10.1 Replikasi

Replikasi adalah mekanisme penyalinan seluruh atau sebagian *table* basis data ke beberapa *server data* [7]. Setiap transaksi (penambahan, penghapusan, atau perubahan data) akan dieksekusi pada semua *server data* yang menyusun sistem BDT. Hal ini menyebabkan mekanisme replikasi membebani performa sistem, namun memiliki tingkat kompleksitas yang paling sederhana.

2.10.2 Fragmentasi

Fragmentasi adalah mekanisme penyalinan sebagian data atau struktur dari setiap tabel basis data [7]. Transaksi basis data yang terjadi harus diolah terlebih dahulu oleh *server manajemen* untuk menentukan letak *server data* dari tabel yang akan terpengaruh. Hal ini menyebabkan mekanisme fragmentasi memiliki tingkat kompleksitas yang tinggi, namun memiliki performa yang lebih baik dibandingkan dengan mekanisme replikasi.

2.11 Database Cluster

Database clustering adalah kumpulan dari beberapa server yang berdiri sendiri yang kemudian bekerjasama sebagai suatu sistem tunggal. Saat ini aplikasi database semakin berkembang, baik dalam hal kegunaan, ukuran, maupun kompleksitas. Hal ini secara langsung berdampak pada server *database* sebagai penyedia layanan terhadap akses *database*, konsekuensi dari semua itu adalah beban *database server* akan semakin bertambah berat dan mengakibatkan kurang optimalnya kinerja dari *server* tersebut.

Oleh karena itu diperlukan perancangan yang tepat dan handal dalam membangun *database server*. *Database* pada masa sekarang ini dituntut agar dapat berjalan dengan cepat, mempunyai kehandalan dan keseterediaan yang tinggi, dengan *clustering database* yang disimpan dapat terbagi ke beberapa mesin dan pada saat aplikasi berjalan, semua mesin yang menyimpan data tersebut dianggap sebagai satu kesatuan. Metode

clustering seperti ini sangat baik untuk *loadbalancing* dan penanganan *system failure* karena kemampuan tiap mesin akan digunakan dan jika ada salah satu mesin yang mengalami *failure* maka sistem tidak akan langsung terganggu karena mesin lain akan tetap berfungsi. Kemampuan *clustering* memungkinkan sebuah *database* tetap hidup dalam waktu yang lama.

2.12 MySQL Cluster

MySQL Cluster merupakan sebuah tipe basis data (database) yang dapat beroperasi dalam ukuran data yang relatif besar (maksimal dalam skala beberapa ratus gigabyte). MySQL Cluster adalah sebuah teknologi baru untuk memungkinkan clustering di dalam memory database dalam sebuah sistem share-nothing. Arsitektur share-nothing memungkinkan sistem dapat bekerja dengan hardware/perangkat keras yang sangat murah, dan tidak membutuhkan perangkat keras dan lunak dengan spesifikasi khusus. Arsitektur tersebut juga handal karena masing-masing komponen mempunyai memory dan disk tersendiri. MySQL Cluster menggabungkan MySQL Server biasa dengan sebuah mesin penyimpanan in-memory tercluster yang dinamakan NDB. NDB berarti bagian dari suatu rangkaian yang dikhususkan sebagai mesin penyimpanan, sedangkan MySQL Cluster diartikan sebagai kombinasi atau gabungan dari MySQL dan mesin penyimpanan yang baru tersebut.

2.13 Arsitektur MySQL Cluster

MySQL Cluster merupakan sebuah database yang menggunakan arsitektur shared-nothing dan antarmuka SQL yang telah umum digunakan. Sistem database ini terdiri dari beberapa node yang dapat didistribusikan ke beberapa perangkat keras dan ke beberapa wilayah/zona yang berbeda sekaligus untuk tetap menjaga ketersediaan data meskipun jaringan ataupun salah satu node sedang mengalami kegagalan (*failure*). Ada tiga node yang menyusun MySQL Cluster, yakni:

1. Data Nodes, digunakan untuk menyimpan semua data yang menjadi milik MySQL Cluster. Semua data direplikasi di node-node ini.
2. Management Server Nodes, digunakan untuk mengendalikan konfigurasi sistem ketika startup. Selain itu, node ini juga dapat digunakan sebagai pengidentifikasi setiap perubahan setting yang terjadi pada cluster.
3. MySQL Server Nodes, berfungsi sebagai pintu akses untuk masuk ke dalam node-node data yang ter-cluster.

2.14 RBAC (*Role Based Access Control*)

RBAC adalah sistem yang diterapkan pada aplikasi yang berhubungan dengan pengontrolan akses sumber daya. RBAC memberikan hak akses untuk peran (*roles*). Perancang kebijakan atau administrator sangat berperan dalam memberikan hak kepada para pelaku, sehingga subjek akan mendapatkan akses ke objek melalui *role* yang telah diberikan oleh administrator (Khayat et al, 2005). Hubungan antara hak akses dengan para pelaku sistem, pertama RBAC memberikan tugas keamanan pada kontrol akses sebagai prioritas tertinggi untuk mengontrol akses ke sumber daya. Hal tersebut mengakibatkan RBAC akan menerapkan keamanan yang sangat ketat dalam melakukan kontrol akses ke sumber daya. Kedua RBAC dalam menerapkan hak akses kepada pengguna membutuhkan waktu yang singkat, dengan cara menghubungkan subjek dengan role, sehingga memerlukan penunggasan hak akses untuk role pada setiap subjek (K hayat et al, 2005).

Kontrol akses dalam mengambil keputusan ditentukan oleh *role*, sehingga pengguna sebagai bagian dari sebuah organisasi akan mendapatkan hak akses sesuai dengan *role* yang didapatkannya. Kebijakan yang dilakukan oleh RBAC akan membuat kontrol akses yang didapatkan oleh pengguna berdasarkan keputusan yang diperoleh dalam sebuah organisasi. Pengguna tidak bisa mengambil hak akses pengguna lain, (Ferraiolo et al, 1992).

Dalam RBAC, role didefinisikan sebagai suatu gagasan yang merupakan dasar dari kebijakan kontrol akses (Sandhu et al, 1999). Dengan kata lain pendefinian role adalah tugas yang didapatkan oleh seseorang sesuai dengan tanggung jawabnya.

2.15 PHP

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 [8]. 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.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis permasalahan dan perancangan tugas akhir. Pada bagian awal dibahas mengenai analisis permasalahan yang ingin diselesaikan. Selanjutnya dibahas mengenai perancangan program untuk memberikan gambaran umum mengenai sistem yang dibuat. Pendekatan yang digunakan dalam perancangan ini adalah pendekatan rancangan berorientasi objek yang direpresentasikan dengan menggunakan diagram UML (*Unified Modelling Language*).

3.1 Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain: cakupan permasalahan, deskripsi umum sistem, spesifikasi kebutuhan perangkat lunak, dan kasus penggunaan sistem. Berikut pembahasan bagian-bagian tahap analisis.

3.1.1 Analisis Proses Bisnis

Aplikasi ERP merupakan aplikasi yang telah lama ada, akan tetapi masing-masing ERP memiliki proses bisnis yang berbeda-beda. Kebanyakan aplikasi ERP yang kompleks dan yang sesuai dengan banyak perusahaan memiliki harga jual yang tinggi. Permasalahannya berada pada bagaimana *business plan* yang telah ada dapat dijalankan secara efisien dengan aplikasi ERP ini. Setiap ERP memiliki proses bisnis yang berbeda-beda. Aplikasi ERP pada Tugas Akhir ini memiliki proses bisnis tersendiri. Pada subbab ini akan dijelaskan tentang analisa proses bisnis yang telah ada. Proses bisnis yang dimiliki oleh Odoo, Adempiere, dan InoERP ditunjukkan pada Lampiran A1, Lampiran A2, dan Lampiran A3. Perbandingan masing-masing ERP di tunjukkan pada Tabel 3.1.

Tabel 3.1 Perbandingan Proses Bisnis ERP 2016, Odoo, InoERP, dan Adempiere

No	Nama Modul	Odoo	InoERP	Adempiere
1	<i>Sales and Distribution</i>	<p>Kontra: 1. Pada proses <i>creating delivery</i>, apabila barang yang dipesan tidak tersedia di gudang/<i>storage</i>, <i>delivery</i> tetap dapat dilakukan</p>	<p>Kontra: 1. Pada proses <i>creating delivery</i>, apabila barang yang dipesan tidak tersedia di gudang/<i>storage</i>, <i>delivery</i> tetap dapat dilakukan</p>	<p>Kontra: 1. Pada proses <i>creating delivery</i>, apabila barang yang dipesan tidak tersedia di gudang/<i>storage</i>, <i>delivery</i> tetap dapat dilakukan</p>
2	<i>Inventory and Warehouse Management</i>	-	<p>kontra: 1. Desain <i>inventory</i> pada InoERP: <i>Warehouse->Locator->Storage</i> Desain <i>inventory</i> pada EZERP: <i>Warehouse -> Inventory -> Storage -> SubStorage -> Bin</i></p>	<p>kontra: 1. Desain <i>inventory</i> pada InoERP: <i>Inventory->SubInventory->Locator</i> Desain <i>inventory</i> pada EZERP: <i>Warehouse -> Inventory -> Storage -> SubStorage -> Bin</i> 2. Pencatatan pengiriman barang diluar gudang tidak ada</p>
3	<i>Purchasing</i>	<p>Kontra: 1. tidak ada sub modul <i>purchase requisition</i>, tidak ada analisa terhadap</p>	<p>Kontra: 1. tidak ada modul untuk mengurus <i>quotation</i> dari <i>supplier</i>, tidak ada</p>	<p>Pro: 1. Fitur <i>purchasing</i> yang terdapat di InoERP sudah lengkap</p>

No	Nama Modul	Odoo	InoERP	Adempiere
		transaksi yang telah dilakukan sebelumnya	analisa terhadap transaksi yang telah dilakukan sebelumnya	Kontra: 1. Tidak ada analisa terhadap transaksi yang sebelumnya telah dilakukan
4	<i>Account Receivable and Account Payable</i>	Kontra: 1. tidak dapat melakukan pembayaran secara kredit Kontra: tidak dapat melakukan pembayaran secara kredit	-	Pro: 1. Fitur sudah lengkap Kontra: 1. tidak ada analisa transaksi yang pernah dilakukan perusahaan 2. tidak ada analisa transaksi yang pernah dilakukan pelanggan
5	<i>Production</i>	Kontra : 1. Tidak ada <i>routing</i> , sehingga penggunaan sumberdaya (mesin dan manusia) tidak tercatat menjadi biaya.	Kontra: 1. <i>Scheduling</i> masih manual (tidak otomatis)	Kontra: 1. <i>Scheduling</i> masih manual (tidak otomatis)
6	<i>Human Resource Management</i>	Kontra : 1. Tidak terdapat KPI dan <i>Payroll</i> masih manual , tidak otomatis.	Kontra : 1. Tidak terdapat KPI dan <i>Payroll</i> masih manual , tidak otomatis.	Kontra : 1. Tidak terdapat KPI dan <i>Payroll</i> masih manual , tidak otomatis.
7	<i>Assets</i>	Kontra:	Kontra :	Pro :

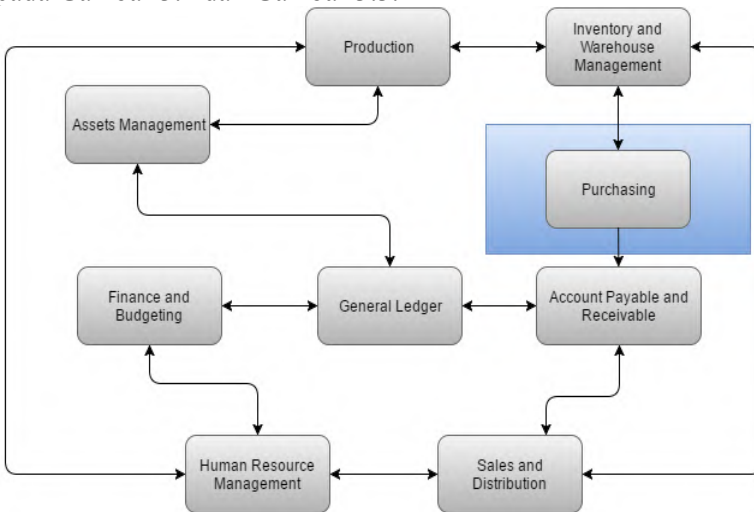
No	Nama Modul	Odoo	InoERP	Adempiere
	<i>Management</i>	1. Produksi tidak melakukan pengecekan status <i>availability</i> mesin 2. <i>Maintenance</i> tidak berdasarkan waktu produksi	1. <i>Maintenance</i> tidak mengikuti waktu produksi	1. Fitur <i>maintenance</i> yang terdapat pada INOERP sudah lengkap terdapat BOM untuk setiap mesin, jadi saat mesin <i>maintenance</i> hanya perlu mengganti partisi yang dibutuhkan Kontra: 1. <i>Maintenance</i> tidak berbasis pada waktu produksi, sistem tidak memberikan <i>warning</i> apabila mesin butuh di <i>maintenance</i>
8	<i>Finance</i>	Kontra : 1. Tidak terdapat COGM maupun COGS sehingga pada modul <i>Sales and Distribution</i> manual	Pro : 1. COGM sudah detail Kontra : 1. <i>Expense</i> atau <i>overhead</i> tidak ada	Pro : 1. Fitur <i>costing</i> yang terdapat pada INOERP sudah lengkap Kontra: 1. Masih manual dalam proses pemberian harga barang jadi, tidak melalui proses COGM, COGS dan <i>overhead</i>
9	<i>General</i>	Pro :	Kontra :	Pro :

No	Nama Modul	Odoo	InoERP	Adempiere
	<i>Ledger</i>	<p>1. Pembuatan akun yang terbilang mudah sehingga mudah digunakan.</p> <p>Kontra : 1. Jurnal masih dapat dibuat secara manual biarpun dalam kondisi belum seimbang</p>	<p>1. Pembuatan Jurnal yang rumit dikarenakan harus menyimpan <i>header</i> dahulu baru <i>line</i>-nya</p>	<p>1. Masing-masing modul sudah menyimpan akun yang akan digunakan sehingga mempermudah proses penjurnalan</p> <p>Kontra : 1. Pembuatan akun yang terbilang rumit, sehingga diaplikasikan algoritma tree dalam pembuatan akun.</p>

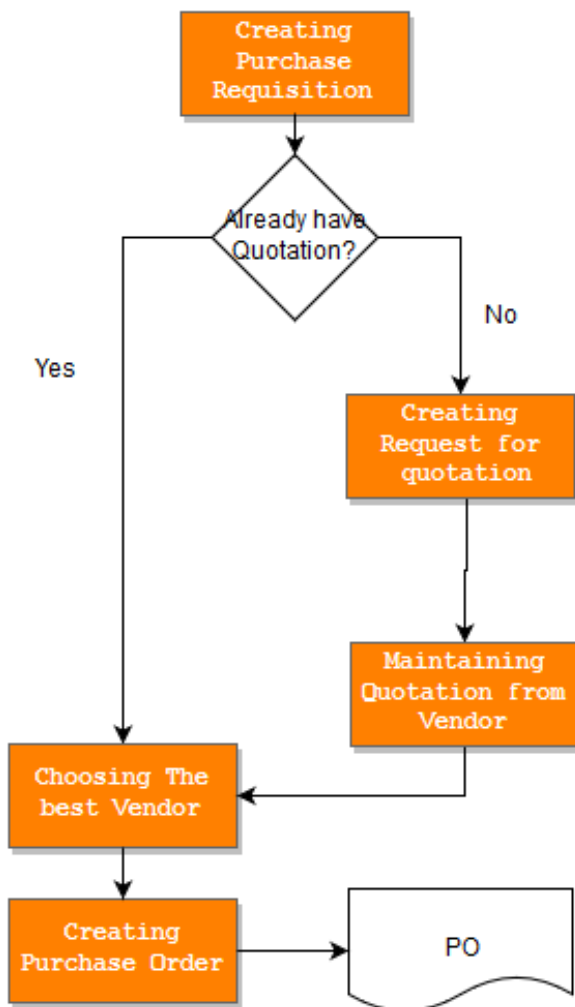
Tabel 3.1 menunjukkan bahwa masih banyak kekurangan atau tidak memenuhi kebutuhan dari *business plan* yang telah dibuat. Berdasarkan kekurangan yang dimiliki oleh setiap ERP maka perlu dikembangkannya sebuah aplikasi ERP yang sesuai dengan kebutuhan pada *business plan*. Terdapat dua jenis proses bisnis yang digunakan sesuai dengan subbab 2.3, yaitu *Make-To-Order* (MTO) dan *Make-To-Stock* (MTS). MTO adalah sebuah proses bisnis dimana produksi didasarkan pada permintaan yang dilakukan oleh *sales order*. MTS adalah sebuah proses bisnis dimana produksi didasarkan oleh peramalan penjualan. Lampiran A4 dan Lampiran A5 menunjukkan proses bisnis pada aplikasi ERP yang telah dirancang sesuai dengan kebutuhan *business plan* yang mengakomodasi kekurangan Odoo, Adempiere, dan InoERP.

Pada Gambar 3.1, modul-modul keuangan (AP/AR, *Finance*, dan *General Ledger*) merupakan sentral dari sistem ERP. Sedangkan, modul *purchasing* berinteraksi dengan modul Pergudangan (*Inventory and Warehouse Management*), Modul *Account Payable*.

Proses bisnis ERP level 0 dapat dipecah kembali menjadi proses-proses bisnis yang lebih rinci berdasarkan subbab 2.3. Proses bisnis secara keseluruhan ditunjukkan pada Lampiran A.4 dan Lampiran A.5. Hasil pemecahan proses bisnis ditunjukkan pada Gambar 3.2 dan Gambar 3.3.



Gambar 3.1 Proses Bisnis ERP Level 0



Gambar 3.2 Proses Bisnis level 1

Pada Gambar 3.2 ditunjukkan proses bisnis *level 1*. Penjelasan proses bisnis *level 1* terdapat pada Tabel 3.2.

Tabel 3.2. Deskripsi Proses Bisnis Level 1

No	Proses Bisnis	Keterangan
1	<i>Creating Purchase Requisition</i>	Merupakan proses bisnis untuk membuat <i>purchase requisition</i> .
2	<i>Creating Request for Quotation</i>	Merupakan proses bisnis untuk membuat <i>request for quotation</i> .
3	<i>Maintaining Quotation from Vendor</i>	Merupakan proses bisnis untuk mengelola <i>Quotation</i> .
4	<i>Choosing the Best Vendor</i>	Merupakan proses bisnis untuk menganalisa supplier.
5	<i>Creating Purchase Order</i>	Merupakan proses bisnis untuk membuat <i>purchase order</i> .

Dari masing-masing proses bisnis *level 1* dapat dipecah kembali menjadi proses bisnis yang lebih rinci lagi. Proses bisnis ditunjukkan pada Gambar 3.3, dan Gambar 3.4.

Pada Gambar 3.3 ditunjukkan proses bisnis *level 2*. Penjelasan proses bisnis *level 2* terdapat pada Tabel 3.3.

Tabel 3.3 Deskripsi Proses Bisnis Level 2

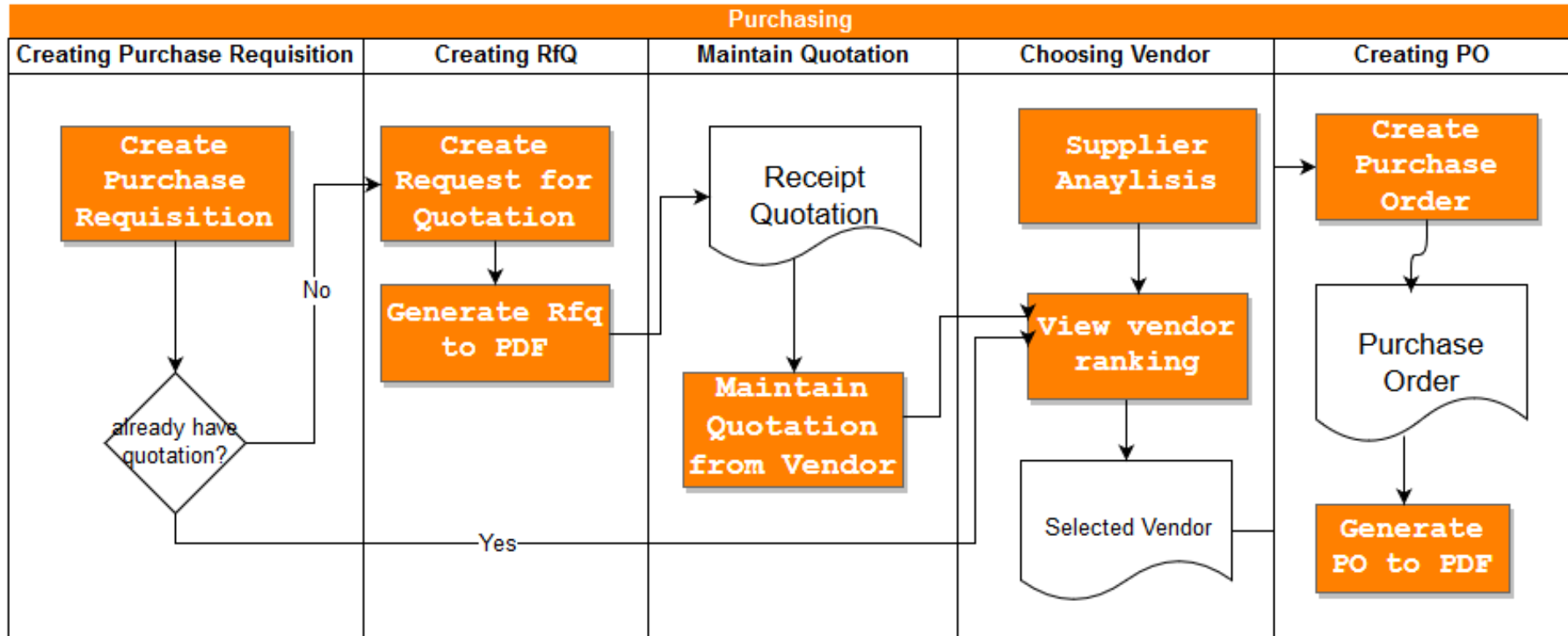
No	Proses Bisnis	Keterangan
1	<i>Create Purchase Requisition</i>	Merupakan proses bisnis untuk membuat <i>purchase requisition</i> .
2	<i>Create Request for Quotation</i>	Merupakan proses bisnis untuk membuat <i>request for quotation</i> .
3	<i>Generate RfQ to PDF</i>	Merupakan proses bisnis untuk membuat dokumen <i>RfQ</i> .
4	<i>Receipt Quotation</i>	Menerima dokumen <i>quotation</i> .

No	Proses Bisnis	Keterangan
5	<i>Maintain quotation from vendor</i>	Merupakan proses bisnis untuk mengelola <i>quotation</i> .
6	<i>Supplier Analysis</i>	Merupakan proses bisnis untuk menganalisa supplier.
7	<i>View Vendor Rangking</i>	Merupakan proses bisnis untuk melihat rangking supplier.
8	<i>Selected Vendor</i>	Merupakan proses bisnis untuk memilih vendor.
9	<i>Create Purchase Order</i>	Merupakan proses bisnis untuk membuat <i>purchase order</i> .
10	<i>Purchase Order</i>	Merupakan proses bisnis untuk melihat <i>purchase order</i> .
11	<i>Generate PO to PDF</i>	Merupakan proses bisnis untuk membuat dokumen <i>PO</i> .

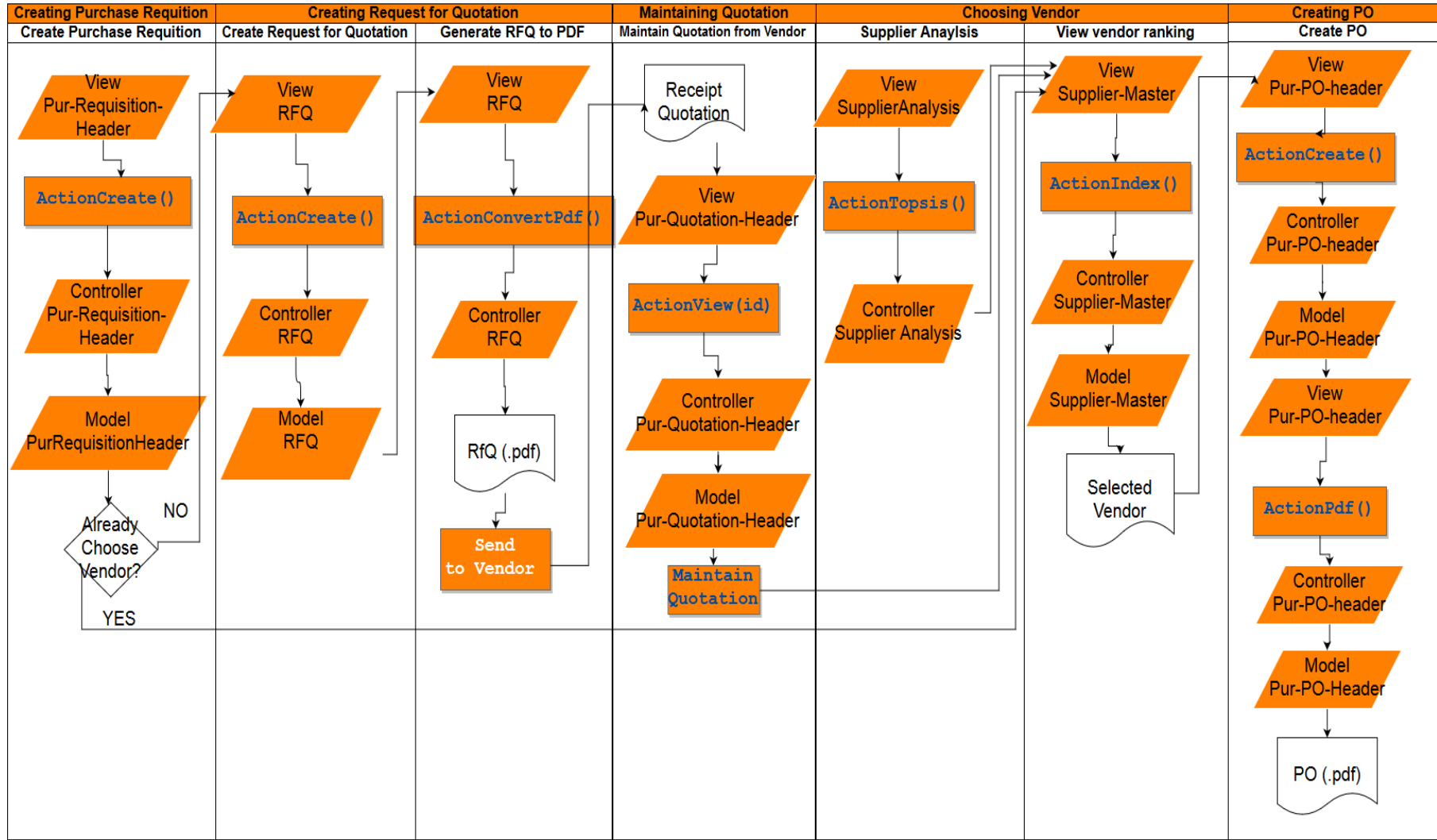
Dari masing-masing proses bisnis *level 2* dapat dipecah kembali menjadi proses bisnis yang lebih rinci lagi. Proses bisnis ditunjukkan pada Gambar 3.4.

Gambar 3.4 merupakan proses bisnis *level 3*. Pada proses bisnis *level 3* menjelaskan detail dari kelas, fungsi, dan halaman yang ditampilkan untuk melakukan proses bisnis yang terdapat pada *level 2*.

[Halaman Ini Sengaja Dikosongkan]



Gambar 3.3. Proses Bisnis Level 2



Gambar 3.4 Proses Bisnis Level 3

3.1.1.1 Analisis Data

Konsep autentikasi dan otorisasi data yang digunakan adalah *multi-tenancy separated schema*. Pada konsep ini, data milik setiap *tenant* akan disimpan di basis data yang terpisah. Sehingga tidak ada 2 atau lebih *tenant* menggunakan *table* dan basis data yang sama. Penambahan *tenant* baru akan menyebabkan penambahan sebuah basis data baru. Penggunaan konsep *separated database* dimaksudkan agar data milik setiap *tenant* dapat terjamin kerahasiaannya. Keuntungan lainnya adalah kerumitan proses *maintenance* basis data dapat dikurangi..

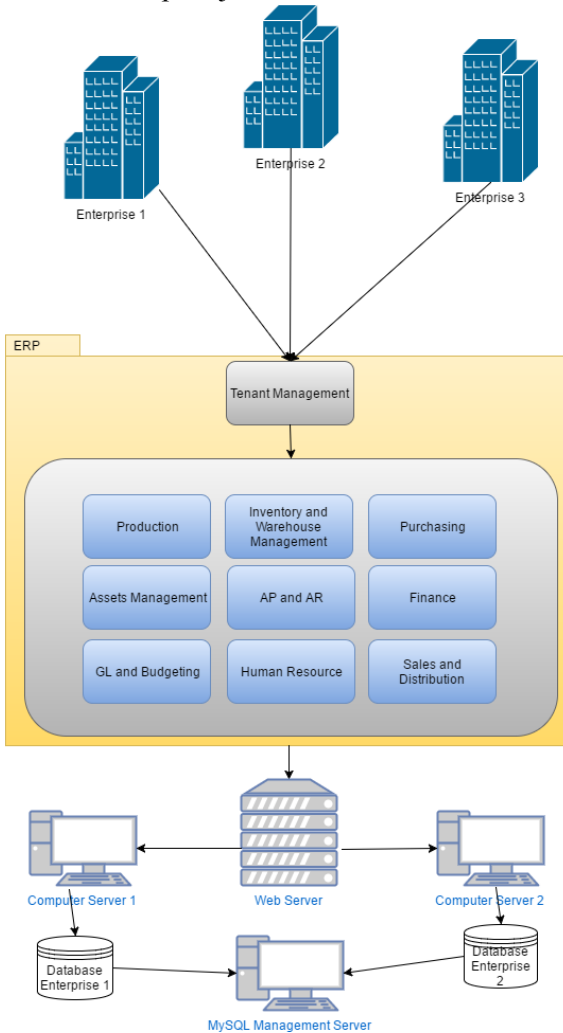
Sedangkan konsep basis data terdistribusi yang diterapkan adalah replikasi. Replikasi memungkinkan penyalinan setiap tabel basis data ke *node-node* penyusun basis data terdistribusi. Kegagalan sebuah *node* tidak akan menyebabkan basis data berhenti bekerja. Sebaliknya, hal tersebut akan memicu mekanisme sinkronisasi jika *node* yang mati kembali hidup.

3.1.2 Deskripsi Umum Sistem

Pada Tugas Akhir ini dibangun aplikasi *Procurement Process* yang terintegrasi ke dalam sebuah sistem *Enterprise Resource Planning* (ERP). Tujuan dari aplikasi *Procurement Process* adalah untuk mengelola dan melakukan monitoring terhadap pergerakan aset tetap. Sistem yang dibangun berorientasi multi-tenancy dengan basis data terdistribusi. Desain basis data terdistribusi dan *multitenancy* yang digunakan dalam sistem ini ditunjukkan pada Gambar 3.5.

Pada Gambar 3.5, dijelaskan bahwa perusahaan 1, perusahaan 2, dan perusahaan 3 dapat menggunakan ERP secara bersamaan. Setiap perusahaan mempunyai 2 basis data yang telah direplikasi yaitu *Computer Server 1* dengan IP *Enterprise 1* dan

Computer Server 2 dengan IP *Enterprise 2*. *Computer 3* berfungsi sebagai *server cluster* untuk melakukan replikasi, sehingga apabila basis data 1 atau *Computer Server 1* dalam keadaan mati, sistem ERP masih tetap berjalan.



Gambar 3.5 Rancangan Basis Data Terdistribusi

Procurement Process meliputi pembuatan *purchase requisition*, *request for quotation*, *maintain quotation*, *vendor* atau *supplier analysis*, *purchase order*, dan segala *report* mengenai *purchasing*.

3.1.2.1 Purchase Requisition

Purchase Requisition adalah sarana untuk membuat permintaan pembelian, agar pihak *purchasing/buyer* dapat melakukan proses pengadaan barang yang di minta. *Purchase Requisition* merupakan awal dari sebuah proses pengadaan barang. Barang yang diminta adalah barang yang terdaftar pada *Item Master*. *Purchase Requisition* berisi tentang jenis barang yang diminta, deskripsi atau detail barang yang diminta, dan tanggal barang tersebut dibutuhkan. Terdapat tiga *trigger* dalam pembuatan *purchase requisition* yang pertama dari ramalan penjualan apabila perusahaan tersebut memakai proses bisnis *make to stock*, yang kedua dari *production request* apabila perusahaan tersebut memakai proses bisnis *make to order*, dan yang ketiga dari department asset.

3.1.2.2 Request for Quotation

Request for Quotation (RfQ) adalah dokumen yang berisi permintaan penawaran secara resmi kepada supplier. RfQ atau yang lebih dikenal dengan surat permintaan penawaran berisi tentang detail barang yang diminta, jumlahnya, alamat pengiriman serta tanggal batas pengiriman *quotation*. RfQ bertujuan untuk mendapatkan *quotation order* atau surat penawaran dari supplier secara resmi. RfQ bisa dibuat dengan cara manual atau bisa juga dibuat dengan cara meng-*convert* dari *Purchase Requisition*, satu *purchase requisition* bisa terdapat pada banyak RfQ

3.1.2.3 Quotation

Quotation Order adalah surat penawaran barang yang diberikan oleh supplier, berisi tentang detail barang, harga barang, diskon, biaya pengiriman, dan pengeluaran untuk pajak. Quotation yang didapatkan dari supplier akan dianalisa oleh bagian *purchasing* dan menentukan apakah akan menerima atau menolak penawaran tersebut. *Quotation* adalah patokan untuk membuat *purchase order*. *Quotation* biasanya didapatkan setelah adanya RfQ ke *supplier*.

3.1.2.4 Supplier Analysis

Supplier Analysis adalah sebuah analisa untuk menentukan rekomendasi *supplier* untuk pembelian barang berdasarkan dari transaksi pembelian dan penerimaan barang yang telah dilakukan sebelumnya, ada tiga kriteria yang terdapat pada analisa *supplier* ini yaitu harga, lamanya keterlambatan, dan kualitas barang. Pengguna dapat memberikan bobot kepada setiap kriteria dengan manual. Analisa ini berupa perankingan *supplier*. Analisa ini bertujuan agar pihak *purchasing* tidak salah dalam memilih *supplier* dan mendapatkan barang dengan kualitas, dan harga yang terbaik. Pembahasan lebih lanjut tentang Supplier Analysis terdapat pada sub-bab 3.1.5.

3.1.2.5 Purchase Order

Purchase order atau yang lebih dikenal dengan PO merupakan salah satu jenis surat bisnis yang sering ditemui. Saat pihak *purchasing* melakukan *order* atau pemesanan kepada perusahaan rekanan atau *supplier*, *purchasing* wajib untuk mengirimkan baik via fax, email, maupun pos. Surat ini akan digunakan *supplier* untuk mengetahui secara detail barang-barang apa saja yang dipesan. Ini berguna untuk mengatasi kesalahan dalam memproduksi pesanan. Selain itu, Surat *Purchase Order* juga akan digunakan *supplier* (dan juga *purchasing*) sebagai file

dan bagi bagian *accounting*/keuangan/pembukuan, Surat *Purchase Order* ini akan dijadikan bukti transaksi laporan keuangan.

Di dalam *Purchase Order* harus memuat dengan jelas jenis barang yang dipesan, jumlah yang dipesan, spesifikasi/permintaan khusus bila ada, tanggal pengiriman, serta yang tidak kalah pentingnya adalah harus mencantumkan nomer PO.

3.1.2.6 Return Order

Return order adalah suatu dokumen pengembalian, bisa berupa pengembalian barang atau pengembalian pembayaran. Pada dasarnya isi dari *return order* sama dengan isi dari *purchase order*. Pada *return order* harus dicantumkan nomor PO dari transaksi sebelumnya, *return order* di trigger dari pihak *warehouse* yang menerima barang langsung dari *supplier*.

3.1.2.7 Item Forecast

Item Forecast adalah suatu laporan peramalan kebutuhan barang mentah atau barang yang akan dipakai untuk produksi. Ramalan dilakukan dengan cara melihat ramalan penjualan pada ulan tertentu dan mengkalikan jumlah ramalannya dengan *bill of material* dari barang tersebut. Pada halaman *Item Forecast* terdapat info mengenai jumlah barang yang sudah dipesan, barang yang sudah terdapat pada gudang, dan ramalan kebutuhan barang pada suatu periode atau bulan tertentu.

3.1.2.8 Item Price History

Item Price History adalah suatu laporan sejarah harga barang yang datanya didapatkan dari transaksi-transaksi *purchase order* sebelumnya. *Item Price History* berupa suatu grafik kenaikan atau penurunan harga barang.

3.1.2.9 Item Demand History

Item Demand History adalah suatu laporan sejarah kebutuhan barang mentah yang datanya didapatkan dari transaksi-transaksi *purchase order* sebelumnya selama periode satu bulan. Pengguna bisa memilih barang yang ingin dilihat sejarah kebutuhannya berupa grafik.

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Berdasarkan uraian mengenai cakupan perangkat lunak pada subbab 3.1.1, maka terdapat beberapa spesifikasi kebutuhan perangkat lunak yang harus dipenuhi agar sistem mampu mengakomodasi permasalahan yang diberikan dan bisa bekerja dengan baik. Spesifikasi kebutuhan perangkat lunak pada Tugas Akhir ini hanya terdiri atas kebutuhan fungsional.

3.1.3.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi kebutuhan utama yang harus dipenuhi oleh sistem agar sistem mampu bekerja dengan baik. Kebutuhan fungsionalitas juga secara tidak langsung mendeskripsikan fungsionalitas perangkat lunak. Daftar kebutuhan fungsionalitas dapat dilihat pada Tabel 3.4.

Tabel 3.4 Daftar kebutuhan fungsional perangkat lunak

Kode Kebutuhan.	Kebutuhan Fungsional	Deskripsi
F-001	Membuat Purchase Requisition	Pengguna dapat membuat dan melihat daftar permintaan barang
F-002	Membuat Request for Quotation	Pengguna dapat membuat request for quotation yang nantinya akan dikirimkan ke supplier
F-003	Mengelola Quotation	Pengguna dapat mencatat quotation yang didapatkan dari supplier
F-004	Membuat Purchase Order	Pengguna dapat membuat dan melihat surat pembelian yang akan

Kode Kebutuhan.	Kebutuhan Fungsional	Deskripsi
		dikirimkan ke supplier
F-005	Membuat Return Order	Pengguna dapat melihat dan membuat surat pengembalian barang
F-006	Menganalisa Supplier	Pengguna dapat melihat supplier terbaik
F-007	Mengeksport Request for Quotatin ke PDF	Pengguna dapat mengubah RfQ yang sudah dibuat ke pdf
F-008	Mengeksport Purchase Order ke PDF	Pengguna dapat mengubah purchase order yang sudah dibuat ke pdf
F-009	Mengeksport Return Order ke PDF	Pengguna dapat mengubah return order yang sudah dibuat ke pdf
F-010	Menerima dan menolak Quotation	Pengguna dapat menerima atau menolak quotation
F-011	Melihat laporan pembelian	Pengguna dapat melihat lapporan pembelian
F-012	Melihat sejarah harga barang	Pengguna dapat melihat sejarah harga barang
F-013	Mengubah Requisition menjadi Request for Quotation	Pengguna dapat mengubah requisition menjadi request for quotation
F-014	Mengubah Requisition menjadi Purchase Order	Pengguna dapat mengubah Requisition menjadi Purchase Order
F-015	Mengubah Request for Quotation menjadi Quotation	Pengguna dapat mengubah Request for Quotation menjadi Quotation
F-016	Mengubah Quotation menjadi Purchase Order	Pengguna dapat mengubah Quotation menjadi Purchase Order
F-017	Melihat ramalan kebutuhan barang mentah	Pengguna dapat melihat ramalan kebutuhan barang mentah

Kode Kebutuhan.	Kebutuhan Fungsional	Deskripsi
F-018	Melihat sejarah kebutuhan barang mentah	Pengguna dapat melihat sejarah kebutuhan barang ment

3.1.3.2 Identifikasi Pengguna

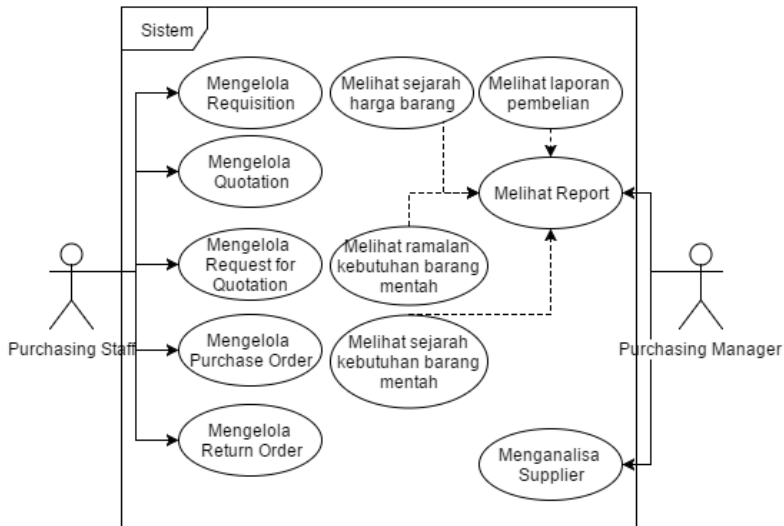
Berdasarkan deskripsi umum yang telah dijelaskan pada Subbab 3.1.2, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini dua, yaitu staf pembelian (*purchasing staff*) dan manajer pembelian (*purchasing manager*). Penjelasan mengenai pengguna yang juga disebut aktor dalam sistem, dijelaskan pada Tabel 3.5.

Tabel 3.5 Pengguna Aplikasi

No.	Aktor	Deskripsi
1.	<i>Purchasing Staff</i>	Bertanggung jawab atas proses-proses pengelolaan data pada modul purchasing membuat (create), memperbaharui (update), dan menghapus (delete).
2.	<i>Purchasing Manager</i>	Mempunyai hak akses untuk melihat (view) data yang telah dikelola oleh staff, dan melihat report-report yang ada.

3.1.4 Kasus Penggunaan

Pada subbab ini akan dijelaskan kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.6 dan kode kasus penggunaan ada pada Tabel 3.6.



Gambar 3.6 Diagram Kasus Penggunaan Aplikasi

Tabel 3.6 Kasus Penggunaan

Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	Mengelola Requisition
UC-002	Mengelola Request for Quotation
UC-003	Mengelola Quotation
UC-004	Mengelola Purchase Order
UC-005	Mengelola Return Order
UC-006	Menganalisa Supplier
UC-007	Melihat Laporan Pembelian
UC-008	Melihat Sejarah Harga Barang

Kode Kasus Penggunaan	Kasus Penggunaan
UC-009	Melihat Ramalan Kebutuhan Barang Mentah
UC-010	Melihat sejarah kebutuhan barang mentah

3.1.4.1 Kasus Penggunaan Mengelola *Requisition* (UC-001)

Pada kasus ini staff *purchasing* mengelola data *purchase requisition*, staff dapat membuat, memperbaharui, menghapus, dan melihat data. *Requisition* harus dibuat sebelum adanya *purchase order*. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.7, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.12.

Tabel 3.7 Kasus Penggunaan Mengelola *Requisition*

Nama	Mengelola Purchase Requisition
Kode	UC-001
Deskripsi	Mengelola(membuat, memperbaharui, menghapus, dan melihat) Purchase Requisition
Tipe	Fungsional
Pemicu	-
Aktor	Staff Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Purchase Requisition.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Staff memilih menu View Purchase Requisition 2. Sistem menampilkan halaman depan purchase requisition 3. Staff memilih aksi <ol style="list-style-type: none"> 3.1 Staff menekan tombol create purchase requisition 4. Sistem menampilkan form create

	<p>purchase requisition.</p> <ol style="list-style-type: none"> 5. Staff mengisi form 6. Staff menekan tombol create 7. Sistem menyimpan data ke database
- Kejadian Alternatif	<ol style="list-style-type: none"> 3.2 Staff menekan tombol Update 4. Sistem menampilkan form dan data purchase requisition 5. Staff memperbaharui 6. Staff menekan tombol Update 7. Data disimpan ke database <ol style="list-style-type: none"> 3.3 Staff menekan tombol delete 4. Sistem menampilkan kotak dialog konfirmasi 5. Sistem menghapus data dari database <ol style="list-style-type: none"> 3.4 Staff menekan tombol view 4. Sistem menampilkan data
Kondisi Akhir	Data pemesanan barang masuk pada basis data.

3.1.4.2 Kasus Penggunaan Mengelola *Request for Quotation* (UC-002)

Pada kasus ini staff *purchasing* mengelola data *request for quotation*, staff dapat membuat, memperbaharui, menghapus, dan melihat data. RfQ harus dibuat untuk mendapatkan *quotation* dari *supplier*. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.8, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.13.

Tabel 3.8 Kasus Penggunaan Mengelola RfQ

Nama	Mengelola Request for Quotation
Kode	UC-002
Deskripsi	Mengelola(membuat, memperbaharui,

	menghapus, dan melihat) Request for Quotation
Type	Fungsional
Pemicu	-
Aktor	Staff Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Request for Quotation.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Staff memilih menu View Request for quotation 2. Sistem menampilkan halaman depan Request for Quotation. 3. Staff memilih aksi: <ol style="list-style-type: none"> 3.1 Staff menekan tombol create Request for Quotation 4. Sistem menampilkan form create purchase requisition. 5. Staff mengisi form 6. Staff menekan tombol create 7. Sistem menyimpan data ke database
- Kejadian Alternatif	<ol style="list-style-type: none"> 3.2 Staff menekan tombol Update 4. Sistem menampilkan from dan data Request for Quotation 5. Staff memperbaharui 6. Staff menekan tombol Update 7. Data disimpan ke database <ol style="list-style-type: none"> 3.3 Staff menekan tombol delete 4. Sistem menampilkan kotak dialog konfirmasi 5. Sistem menghapus data dari database <ol style="list-style-type: none"> 3.4 Staff menekan tombol view 4. Sistem menampilkan data
Kondisi Akhir	Data pemesanan barang masuk pada basis data.

3.1.4.3 Kasus Penggunaan Mengelola *Quotation* (UC-003)

Pada kasus ini staff *purchasing* mengelola data *quotation* yang didapatkan dari supplier, staff dapat membuat, memperbaharui, menghapus, dan melihat data. *quotation* dibuat untuk mencatat dan memilih *quotation* dari supplier. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.9, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.14.

Tabel 3.9 Kasus Penggunaan Mengelola *Quotation*

Nama	Mengelola <i>Quotation</i>
Kode	UC-003
Deskripsi	Mengelola(membuat, memperbaharui, menghapus, dan melihat) <i>Quotation</i>
Tipe	Fungsional
Pemicu	-
Aktor	Staff Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Quotation</i> .
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Staff memilih menu View <i>quotation</i> 2. Sistem menampilkan halaman depan <i>Quotation</i>. 3. Staff memilih aksi: <ol style="list-style-type: none"> 3.1 Staff menekan tombol create <i>Quotation</i> 4. Sistem menampilkan form create <i>Quotation</i>. 5. Staff mengisi form 6. Staff menekan tombol create 7. Sistem menyimpan data ke database
- Kejadian Alternatif	<ol style="list-style-type: none"> 3.2 Staff menekan tombol Update 4. Sistem menampilkan form dan data <i>Quotation</i> 5. Staff memperbaharui 6. Staff menekan tombol Update 7. Data disimpan ke database

	3.3 Staff menekan tombol delete 4. Sistem menampilkan kotak dialog konfirmasi 5. Sistem menghapus data dari database 3.4 Staff menekan tombol view 4. Sistem menampilkan data
Kondisi Akhir	Data pemesanan barang masuk pada basis data.

3.1.4.4 Kasus Penggunaan Mengelola *Purchase Order* (UC-004)

Pada kasus ini staff *purchasing* mengelola data *purchase order*, staff dapat membuat, memperbaharui, menghapus, dan melihat data. *purchase order* dibuat untuk nanti dikirimkan ke *supplier*. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.10, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.15.

Tabel 3.10 Kasus Penggunaan Purchase Order

Nama	Mengelola Purchase Order
Kode	UC-004
Deskripsi	Mengelola(membuat, memperbaharui, menghapus, dan melihat) Purchase Order
Tipe	Fungsional
Pemicu	-
Aktor	Staff Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Purchase Order.
Aliran: - Kejadian Normal	1.Staff memilih menu View Purchase Order 2.Sistem menampilkan halaman depan Purchase Order. 3. Staff memilih aksi: 3.1 Staff menekan tombol create purchase order 4. Sistem menampilkan form create

	Purchase Order. 5. Staff mengisi form 6. Staff menekan tombol create 7. Sistem menyimpan data ke database
- Kejadian Alternatif	3.2 Staff menekan tombol Update 4. Sistem menampilkan form dan data Purchase Order. 5. Staff memperbaharui 6. Staff menekan tombol Update 7. Data disimpan ke database 3.3 Staff menekan tombol delete 4. Sistem menampilkan kotak dialog konfirmasi 5. Sistem menghapus data dari database 3.4 Staff menekan tombol view 4. Sistem menampilkan data
Kondisi Akhir	Data pemesanan barang masuk pada basis data.

3.1.4.5 Kasus Penggunaan Mengelola Return Order

Pada kasus ini staff *purchasing* mengelola data *return order*, staff dapat membuat, memperbaharui, menghapus, dan melihat data. *return order* dibuat untuk nanti dikirimkan ke *supplier*. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.11, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.16.

Tabel 3.11 Kasus Penggunaan Return Order

Nama	Mengelola Return Order
Kode	UC-005
Deskripsi	Mengelola(membuat, memperbaharui, menghapus, dan melihat) Return Order

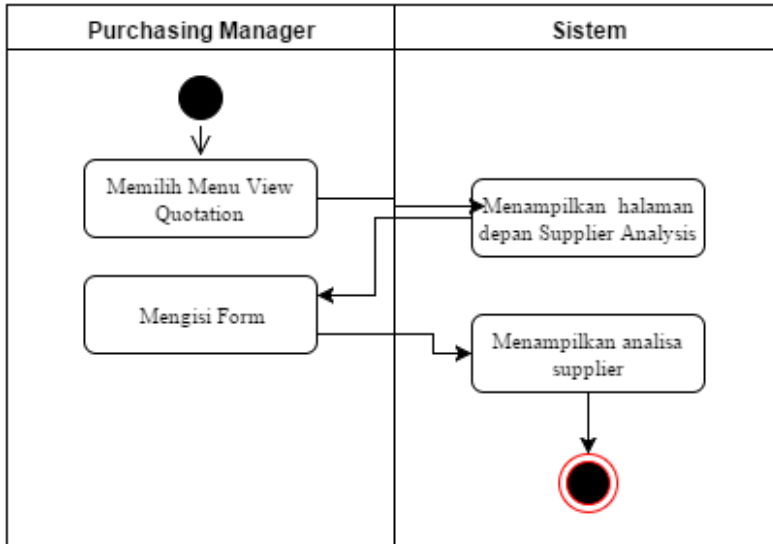
Tipe	Fungsional
Pemicu	-
Aktor	Staff Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Return Order.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Staff memilih menu View Return Order 2. Sistem menampilkan halaman depan Return Order. 3. Staff memilih aksi: <ol style="list-style-type: none"> 3.1 Staff menekan tombol create return order 4. Sistem menampilkan form create Return Order. 5. Staff mengisi form 6. Staff menekan tombol create 7. Sistem menyimpan data ke database
- Kejadian Alternatif	<ol style="list-style-type: none"> 3.2 Staff menekan tombol Update 4. Sistem menampilkan form dan data Return Order. 5. Staff memperbaharui 6. Staff menekan tombol Update 7. Data disimpan ke database <ol style="list-style-type: none"> 3.3 Staff menekan tombol delete 4. Sistem menampilkan kotak dialog konfirmasi 5. Sistem menghapus data dari database <ol style="list-style-type: none"> 3.4 Staff menekan tombol view 4. Sistem menampilkan data
Kondisi Akhir	Data pemesanan barang masuk pada basis data.

3.1.4.6 Kasus Penggunaan Menganalisa Supplier (UC-006)

Pada kasus ini *manager purchasing* menganalisa *supplier* terbaik untuk suatu jenis barang mentah. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.12, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.7.

Tabel 3.12 Kasus Penggunaan Menganalisa Supplier

Nama	Menganalisa Supplier
Kode	UC-006
Deskripsi	Manager Menganalisa supplier untuk mendapatkan supplier terbaik
Tipe	Fungsional
Pemicu	-
Aktor	Manager Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Supplier Analysis.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Staff memilih menu Supplier Analysis 2. Sistem menampilkan halaman depan Supplier Analysis. 3. Staff mengisi form 4. Sistem menampilkan rangking dari supplier terbaik
- Kejadian Alternatif	
Kondisi Akhir	Sistem Menampilkan laporan



Gambar 3.7 Diagram Aktifitas Menganalisa Supplier

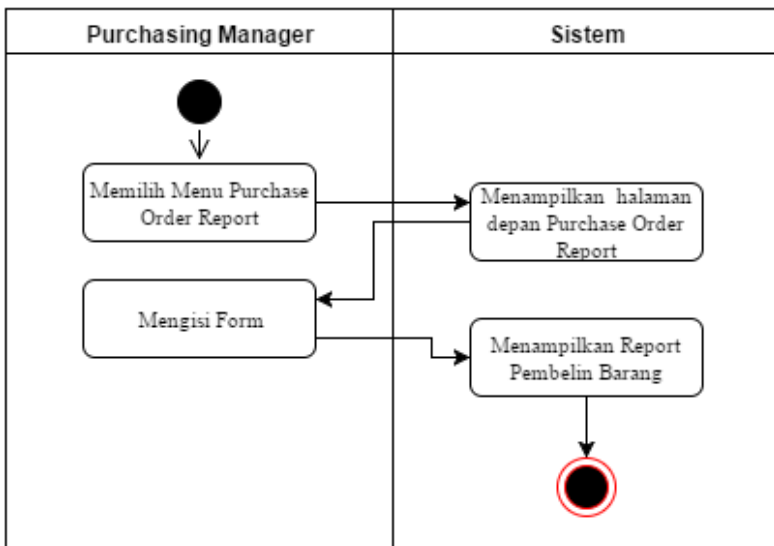
3.1.4.7 Kasus Penggunaan Melihat Laporan Pembelian

Pada kasus ini *manager purchasing* melihat *report* pembelian barang. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.13, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.8.

Tabel 3.13 Kasus Penggunaan Melihat Laporan Pembelian

Nama	Melihat Laporan Pembelian
Kode	UC-007
Deskripsi	Manager Melihat Laporan Pembelian pada periode tertentu
Tipe	Fungsional
Pemicu	-
Aktor	Manager Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Purchase Order Report.
Aliran: - Kejadian	1. Staff memilih menu Purchase Order Report

Normal	<ol style="list-style-type: none"> 2. Sistem menampilkan halaman depan purchase order report. 3. Staff mengisi form 4. Sistem menampilkan report pembelian barang
- Kejadian Alternatif	
Kondisi Akhir	Sistem Menampilkan laporan



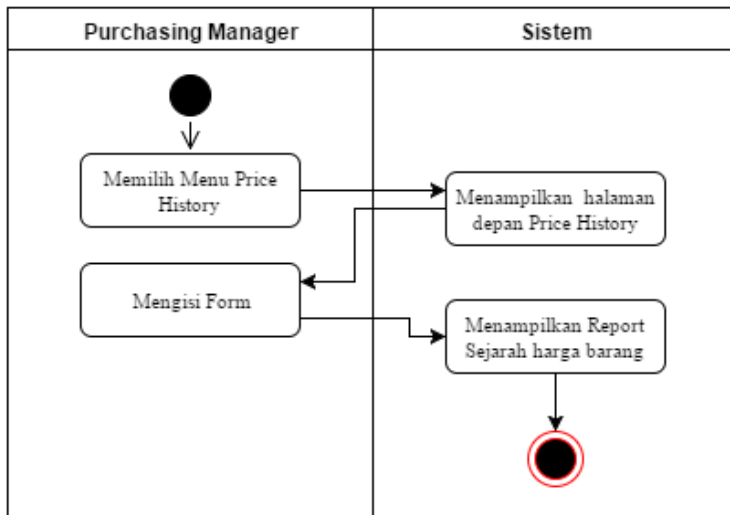
Gambar 3.8 Diagram Aktifitas Melihat Laporan Pembelian

3.1.4.8 Kasus Penggunaan Melihat Sejarah Harga Barang

Pada kasus ini *manager purchasing* melihat *report* pembelian barang. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.14, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.9.

Tabel 3.14 Kasus Penggunaan Melihat Sejarah Harga Barang

Nama	Melihat Sejarah Harga Barang
Kode	UC-008
Deskripsi	Manager melihat laporan pembelian
Tipe	Fungsional
Pemicu	-
Aktor	Manager Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Item Price History.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Staff memilih menu Item Price History 2. Sistem menampilkan halaman depan Item Price History. 3. Staff mengisi form 4. Sistem menampilkan report sejarah harga barang
- Kejadian Alternatif	
Kondisi Akhir	Sistem Menampilkan laporan



Gambar 3.9 Diagram Aktifitas Melihat Sejarah Harga Barang

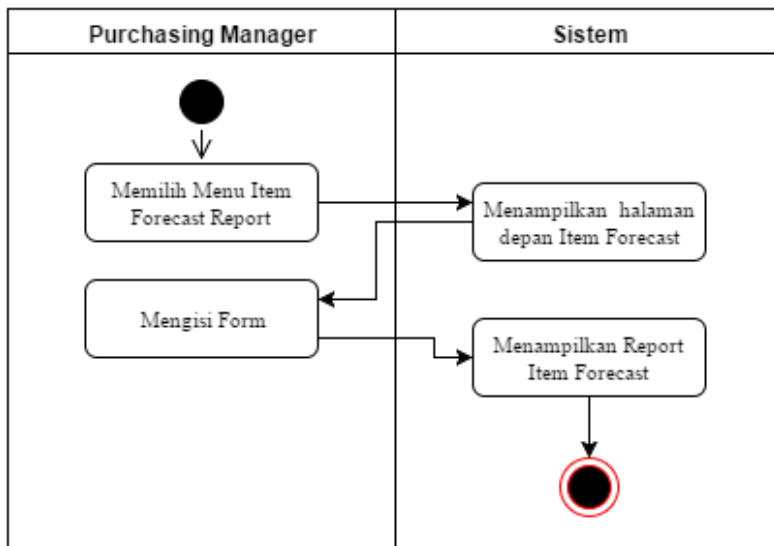
3.1.4.9 Kasus Penggunaan Melihat Ramalan Kebutuhan Barang Mentah

Pada kasus ini *manager purchasing* melihat *report* kebutuhan barang mentah dimasa depan. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.15, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.10.

Tabel 3.15 Kasus Penggunaan Melihat Ramalan Kebutuhan Barang Mentah

Nama	Melihat Ramalan Kebutuhan Barang Mentah
Kode	UC-009
Deskripsi	Manager melihat ramalan kebutuhan barang mentah
Tipe	Fungsional
Pemicu	-
Aktor	Manager Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Item Forecast.
Aliran:	1. Staff memilih menu Item Forecast

- Kejadian Normal	2. Sistem menampilkan halaman depan Item Forecast. 3. Staff mengisi form 4. Sistem menampilkan report ramalan kebutuhan barang
- Kejadian Alternatif	
Kondisi Akhir	Sistem Menampilkan laporan



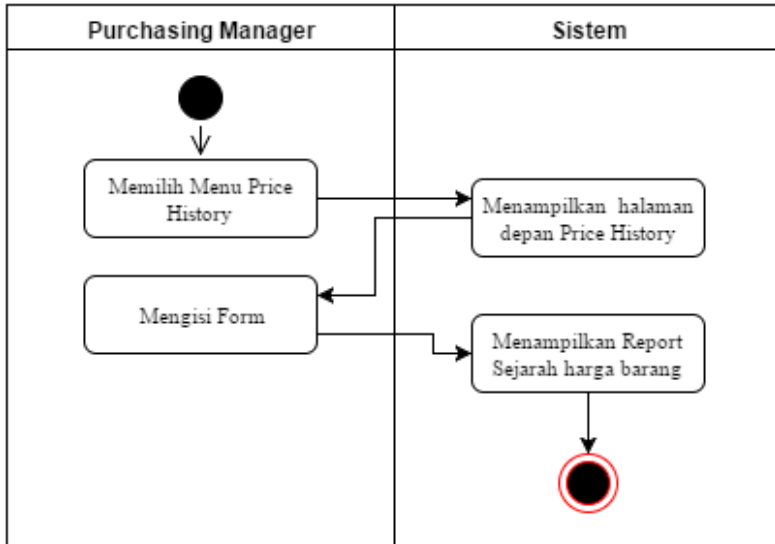
Gambar 3.10 Diagram Aktivitas Melihat Ramalan Kebutuhan Barang Mentah

3.1.4.10 Kasus Penggunaan Melihat Sejarah Kebutuhan Barang Mentah

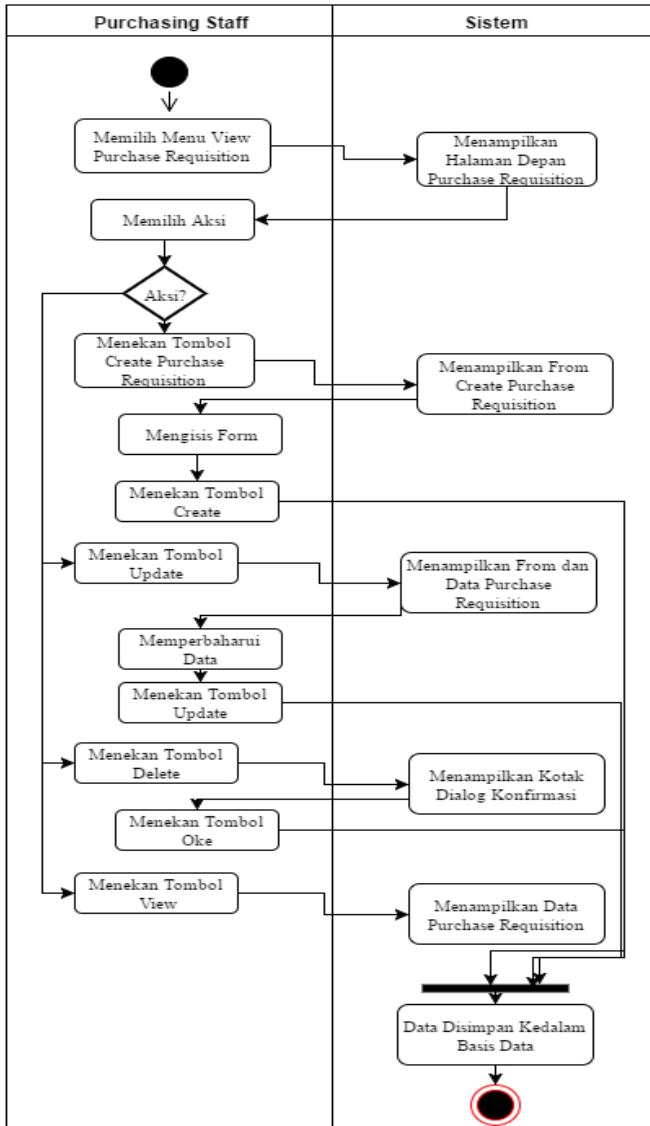
Pada kasus ini *manager purchasing* melihat *report* kebutuhan barang mentah dimasa depan. Rincian kasus penggunaan ini dapat dilihat pada Tabel 3.16, sedangkan diagram aktivitas dapat dilihat pada Gambar 3.11.

Tabel 3.16 Kasus Penggunaan Melihat Sejarah Kebutuhan Barang Mentah

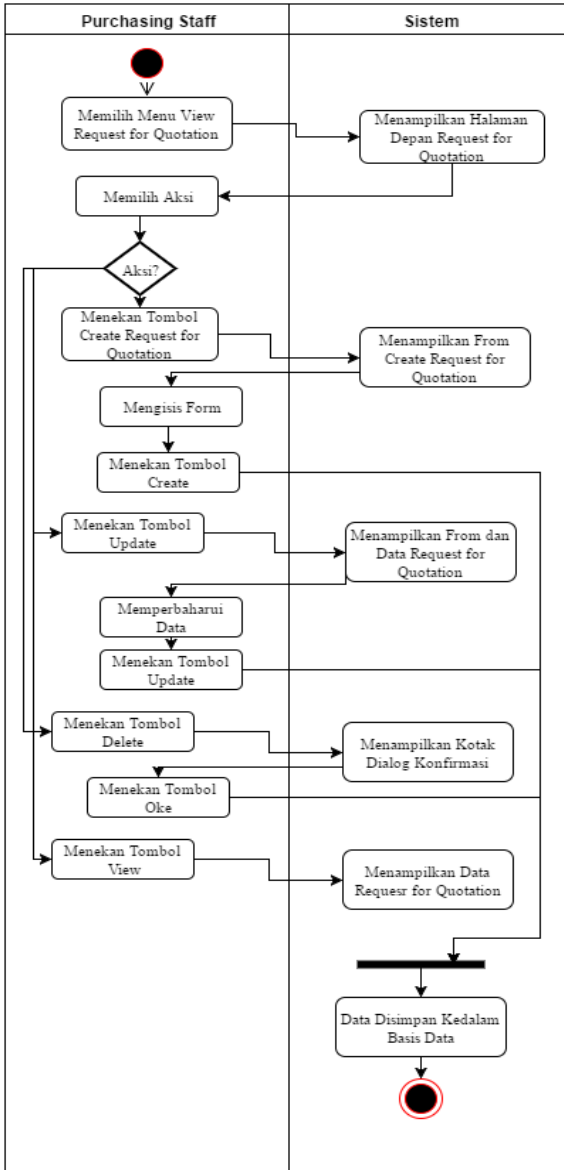
Nama	Melihat Sejarah Kebutuhan Barang Mentah
Kode	UC-010
Deskripsi	Manager melihat sejarah kebutuhan barang mentah
Tipe	Fungsional
Pemicu	-
Aktor	Manager Purchasing
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu Item Demand History.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Staff memilih menu Item Demand History 2. Sistem menampilkan halaman depan Item Demand History. 3. Staff mengisi form 4. Sistem menampilkan laporan sejarah kebutuhan barang
- Kejadian Alternatif	
Kondisi Akhir	Sistem Menampilkan laporan



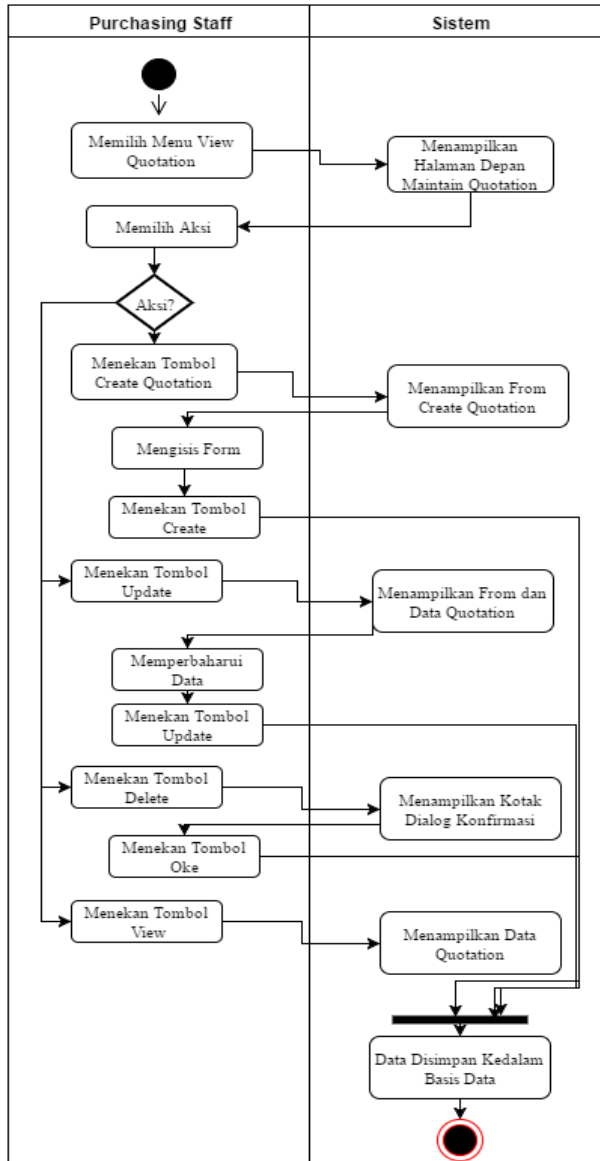
Gambar 3.11 Diagram Aktivitas Melihat Sejarah Permintaan Barang



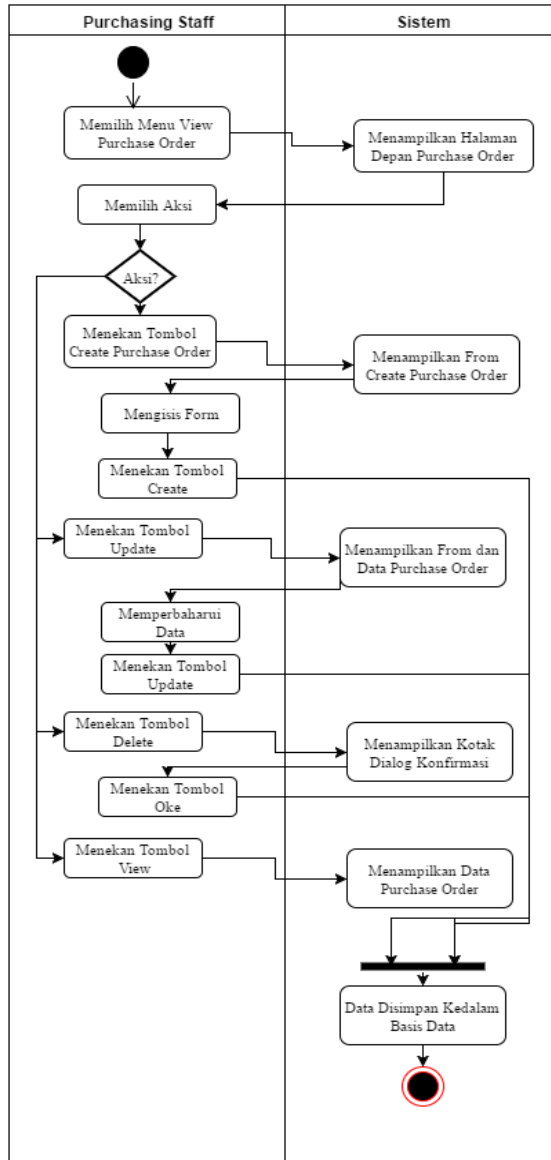
Gambar 3.12 Diagram Aktivitas mengelola Purchase Requisition



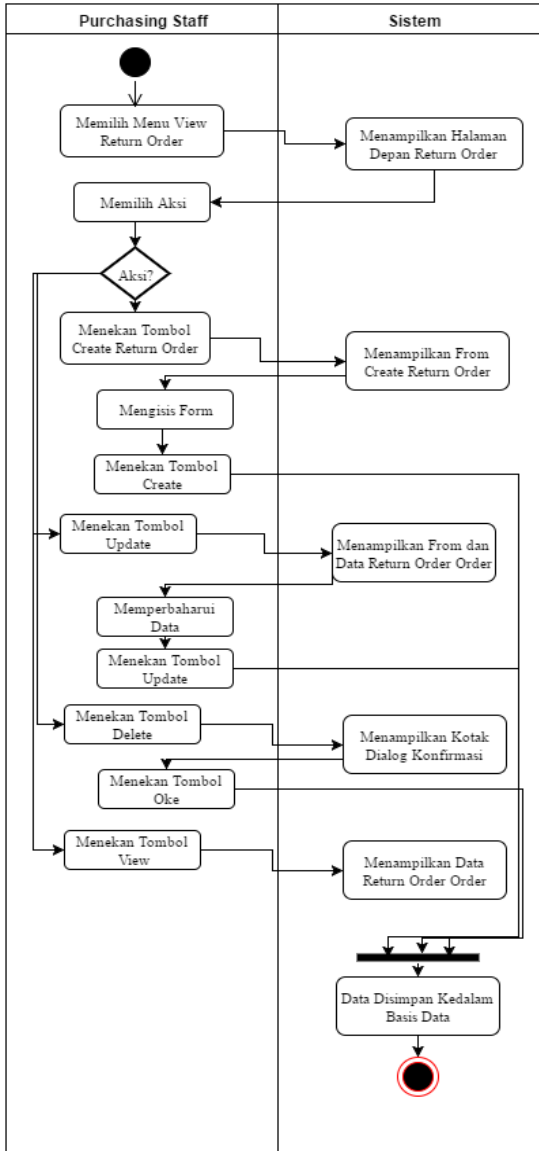
Gambar 3.13 Diagram Aktifitas mengelola RfQ



Gambar 3.14 Diagram Aktifitas Mengelola Quotation



Gambar 3.15 Diagram Aktivitas Mengelola Purchase Order



Gambar 3.16 Diagram Aktivitas Mengelola Return Order

3.1.5 Analisa Supplier

Supplier Analysis adalah sebuah analisa untuk menentukan rekomendasi *supplier* untuk pembelian barang berdasarkan data dari transaksi pembelian dan transaksi penerimaan barang yang telah dilakukan sebelumnya. Analisa ini bertujuan agar pihak *purchasing* tidak salah dalam memilih *supplier* dan mendapatkan barang dengan kualitas, dan harga yang terbaik.

Pada analisa *supplier* terdapat istilah alternatif, dan kriteria, alternatif adalah *supplier-supplier* yang mau dianalisa, dan kriteria adalah data-data yang digunakan untuk mendapatkan *score* dari *supplier*. Ada tiga kriteria dalam menenukan *supplier* terbaik yaitu, harga barang, kualitas barang/jumlah barang cacat, dan lamanya keterlambatan dalam pengiriman, pemilihan tiga kriteria ini berdasarkan dari data-data transaksi yang telah ada sebelumnya yaitu dari data *Purchase Order* dan *Good Receipt*. Data harga barang bisa didapatkan dari data pada *Purchase Order*. Data keterlambatan bisa didapatkan dari data pada *Gr Document Date* di *Good Receipt*, dan data pada *Need by Date* di *Purchase Order*, selisih dari kedua tanggal itu akan dijadikan data keterlambatan. Data kualitas barang bisa didapatkan pada presentase *item defect* di *Good Receipt*. Algoritma atau langkah-langkah analisa supplier terdapat pada Gambar 3.17.

Dari tiga kriteria tersebut, terdapat empat bobot yang bisa dipilih, yaitu *ignore* atau diabaikan, *not important* atau tidak penting, *important* atau penting, dan *very important* atau sangat penting, keterangan pada setiap bobot terdapat pada Tabel 3.17

Tabel 3.17 Keterangan Bobot

Nama Bobot	Besar Bobot
Ignore	0
Not Important	0.3
Important	0.7
Very Important	1



Gambar 3.17 Algoritma Analisa Supplier

Langkah-langkah dari analisa supplier:

3.1.5.1 Gabungkan data Purchase Order dan Good Receipt

Tahap ini adalah tahap sebelum perhitungan TOPSIS, pada tahap ini data-data transaksi yang terdapat pada purchase order dan good receipt digabungkan. Hasilnya dapat dilihat pada Tabel 3.18.

Tabel 3.18 Gabungan data Purchase Order dan Good Receipt

PO	Supplier	Item	Quantity	Price	Delivery late (hour)	Item defect	Item defect (%)
PO001	supplier gear 1	gear	200	20	2	2	1
PO002	supplier gear 1	gear	200	19	0	5	2.5
PO003	supplier gear 1	gear	200	18	4	13	6.5
PO004	supplier gear 2	gear	250	21	2	14	5.6
PO005	supplier gear 2	gear	200	20	6	12	6
PO006	supplier gear 2	gear	150	18	3	17	11.333
PO007	supplier gear 3	gear	175	20	0	2	1.1429
PO008	supplier gear 3	gear	200	20	5	9	4.5
PO009	supplier gear 3	gear	130	19	2	3	2.3077

3.1.5.2 Hitung rata-rata kriteria dari setiap supplier

Pada tahap ini, setiap supplier dihitung rata-ratanya dari setiap transaksi, misalnya supplier gear 1 mengeluarkan harga 20,

19, dan 18, maka rata-ratanya menjadi 18. Hasilnya dapat dilihat pada Tabel 3.19. Hasilnya dinamakan matriks x.

Tabel 3.19 Rata-rata Tiap Supplier

Nama Supplier	Avg Price	Avg Delivery Late	Avg Item Defect (%)
Supplier Gear 1	19	2	3.33333333
Supplier Gear 2	19.66667	3.666667	7.64444444
Supplier Gear 3	19.66667	2.333333	2.65018315

3.1.5.3 Menormalisasi matriks x

Pada tahap ini, dimulai perhitungan TOPSIS, pada tahap ini matriks x yang pada langkah sebelumnya telah didapatkan, dinormalisasi. Hasilnya terdapat pada Tabel 3.20.

Tabel 3.20 Matriks Ternormalisasi

Nama Supplier	Price	Delivery Late	Item Defect
Supplier Gear 1	0.56	0.42	0.38
Supplier Gear 2	0.58	0.77	0.87
Supplier Gear 3	0.58	0.49	0.30

3.1.5.4 Membuat pembobotan pada matriks yang telah dinormalisasi

Setelah dinormalisasi, setiap kolom pada matriks ternormalisasi dikalikan dengan bobot. Bobot dari masing-masing kriteria terdapat pada Tabel 3.21, bobot tersebut didapatkan dari Tabel 3.17, dan hasil dari tahap ini terdapat pada Tabel 3.22.

Tabel 3.21 Bobot setiap kriteria

	Price	Delivery Dtae	Defect
Bobot	0.3	0.7	1

Tabel 3.22 Matriks Terbobot

Nama Supplier	Price	Delivery Late	Item Defect
Supplier Gear 1	0.17	0.29	0.38
Supplier Gear 2	0..18	0.54	0.87
Supplier Gear 3	0.18	0.34	0.30

3.1.5.5 Menentukan nilai solusi ideal positif dan solusi ideal negatif

Solusi ideal dinotasikan A^* , sedangkan solusi ideal negatif dinotasikan A^- . Pada kasus ini, solusi ideal positif adalah yang nilainya paling rendah, dan solusi ideal negative yang nilainya paling tinggi. Hasilnya bisa dilihat pada Tabel 3.23.

Tabel 3.23 Solusi Ideal Positif dan Negatif

	Price	Delivery Late	Item Defect
Ideal Positif	0.17	0.29	0.30
Ideal Negatif	0.18	0.54	0.87

3.1.5.6 Menentukan *Separation Measure* (d_i)

Separation measure ini merupakan pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negatif.

Tabel 3.24 Separatif dari Positif

Nama Supplier	dij(+)
Supplier Gear 1	0.08
Supplier Gear 2	0.62
Supplier Gear 3	0.05

Tabel 3.25 Separation dari Negatif

Nama Supplier	dij(-)
Supplier Gear 1	0.55
Supplier Gear 2	0.00
Supplier Gear 3	0.60

3.1.5.7 Menghitung nilai preferensi (CC_i) untuk setiap alternatif

Untuk menentukan ranking setiap alternatif yang ada maka perlu dihitung terlebih dahulu nilai preferensi dari tiap alternatif. Hasil dari tahap ini terdapat pada Tabel 3.26.

Tabel 3.26 Supplier's Score

Nama Supplier	CC_i (Score)
Supplier Gear 1	0.88
Supplier Gear 2	0.00
Supplier Gear 3	0.92

3.1.5.8 Menyorting Supplier berdasarkan Scorenya

Setelah mendapatkan scorenya, lalu *supplier* diurutkan dari yang memiliki *score* tertinggi ke yang terendah, *supplier* yang memiliki *score* tertinggi adalah supplier yang terbaik atau yang paling direkomendasikan. Hasil pada tahap ini terdapat pada Tabel 3.27.

Tabel 3.27 Supplier's Rank

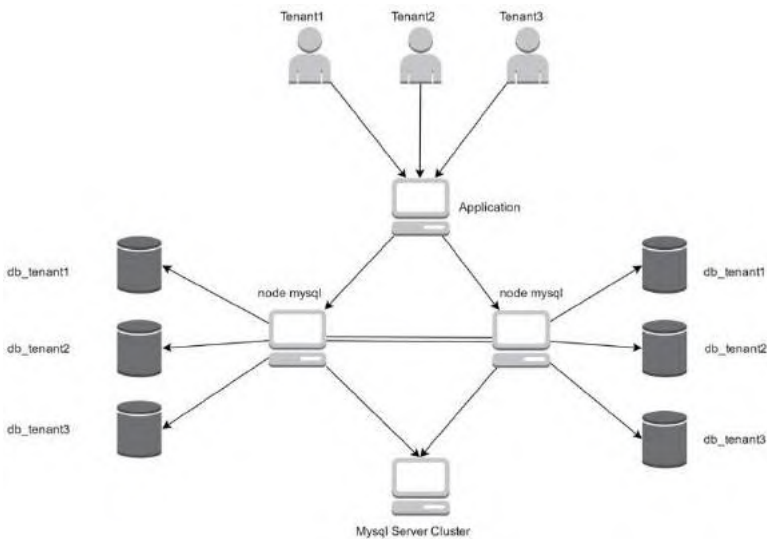
Nama Supplier	CC_i (Score)	Rank
Supplier Gear 3	0.92	1
Supplier Gear 1	0.88	2
Supplier Gear 2	0.00	3

3.2 Perancangan Sistem

Tahap ini meliputi perancangan basis data, tampilan antarmuka, *multitenancy*, dan *RBAC* pada modul *purchasing* yang diharapkan dapat memenuhi tujuan dari pengembangan aplikasi ini.

3.2.1 Perancangan *Multitenancy*

Pada bagian ini akan dijelaskan mengenai perancangan *multi-tenancy* yang terdapat pada sistem ERP 2016. Secara rinci mengenai *multi-tenancy* dijabarkan sebagai berikut:



Gambar 3.18 Perancangan Multi-tenancy

Gambar 3.18 menerangkan bagaimana alur *multi-tenancy* yang terjadi pada sistem ERP. Pembagian *node mysql* sudah dijelaskan pada subbab 3.2.1 yaitu dengan cara *clustering* menjadi 2 database yang berbeda. Basis data terdistribusi merupakan penunjang dari *multi-tenancy*, yang berfungsi untuk

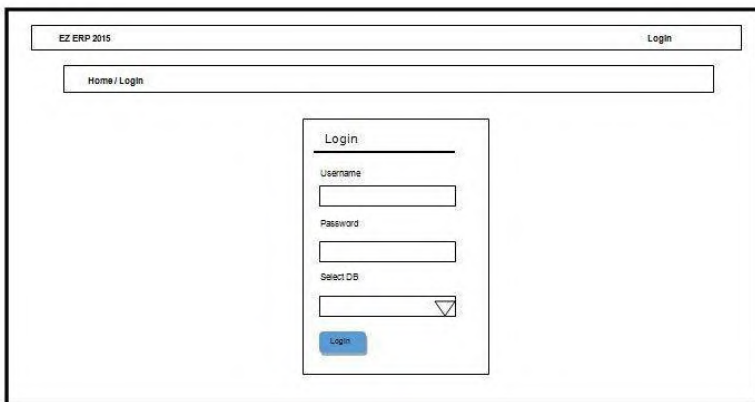
mengatur session setiap perusahaan yang akan mengakses sistem ERP.

3.2.2 Perancangan RBAC

Pada bagian ini akan dijelaskan mengenai perancangan RBAC yang terdapat pada sistem ERP 2016. Secara rinci mengenai RBAC dijabarkan sebagai berikut

1. Perancangan antarmuka login

Rancangan antarmuka *login* dibagi menjadi 3 (tiga) bagian, di bagian paling atas terdapat *header* yang berisi logo aplikasi dan tombol *login*, di bagian atas terdapat navigasi (*breadcrumb*) untuk mengetahui pengguna sedang berada pada halaman tertentu, dan di bagian tengah terdapat form *login* yang memiliki *username*, *password*, dan pemilihan basis data yang digunakan sebagai input. Perancangan antarmuka *login* ditunjukkan pada Gambar 3.19.



The image shows a web browser window displaying the login page for 'EZ ERP 2015'. The page layout includes a top header with the application name on the left and a 'Login' link on the right. Below the header is a breadcrumb trail: 'Home / Login'. The central focus is a 'Login' form containing three input fields: 'Username', 'Password', and 'Select DB' (a dropdown menu). A blue 'Login' button is positioned at the bottom of the form.

Gambar 3.19 Perancangan antarmuka Login

2. Perancangan antarmuka add user

Rancangan antarmuka menambahkan pengguna baru ditampilkan dalam sebuah kotak dialog yang berisi *username*, *email*, dan *password* yang digunakan sebagai input data untuk menambahkan pengguna baru. Perancangan antarmuka menambahkan pengguna baru ditunjukkan pada Gambar 3.20.

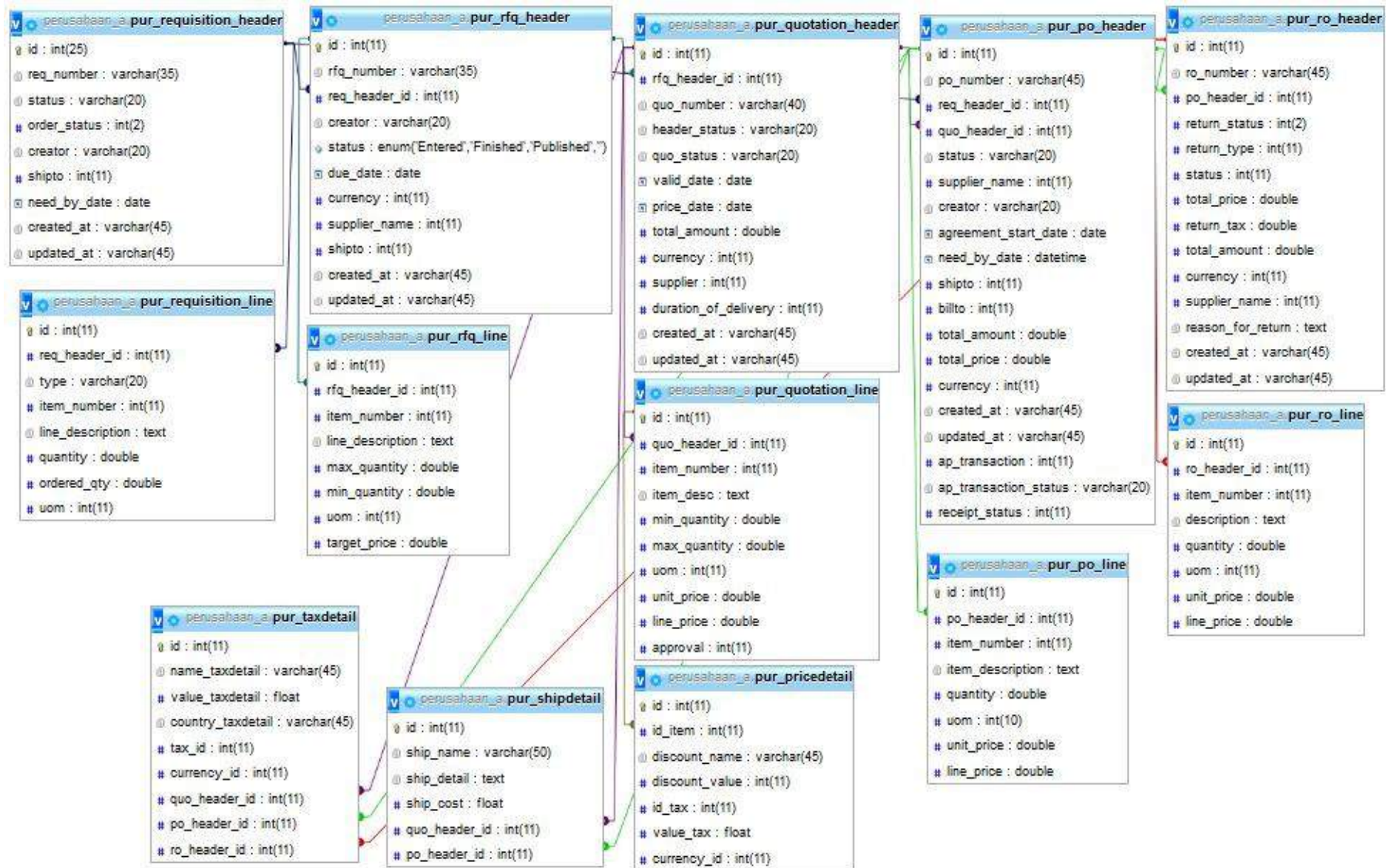


Gambar 3.20 Perancangan antarmuka add user

3.2.3 Perancangan Basis Data

Pada subbab ini akan dijelaskan bagaimana rancangan basis data yang digunakan pada sistem. Basis data pada sistem ini menggunakan sistem manajemen basis data *MySQL Cluster*. *Physical Data Model* (PDM) dari basis data sistem ini ditunjukkan pada Gambar 3.32.

Perancangan PDM pada Gambar 3.21 dijelaskan pada Tabel 3.28.



Gambar 3.21 Physical Data Model

[Halaman Ini Sengaja Dikosongkan]

Tabel 3.28 Keterangan PDM

No	Nama Tabel	Atribut	Keterangan
1	Pur_requisition_header	id: <i>int(25)</i>	Tabel untuk membuat <i>Requisition Header</i>
		req_number: <i>varchar(35)</i>	
		Status: <i>varchar(20)</i>	
		Order_status: <i>int(2)</i>	
		Creator: <i>varchar(20)</i>	
		Shipto: <i>int(11)</i>	
		need_by_date: <i>date</i>	
		created_at: <i>varchar(45)</i>	
		pdated_at: <i>varchar(45)</i>	
2	Pur_requisition_line	Id: <i>int(11)</i>	Tabel untuk memasukkan rincian <i>Requisition</i>
		Req_header_id: <i>int(11)</i>	
		Type: <i>varchar(20)</i>	
		Item_number: <i>int(11)</i>	
		Line_description: <i>text</i>	
		Quantity: <i>double</i>	
		Ordered_qty: <i>double</i>	
Uom: <i>int(11)</i>			
3	Pur_rfq_header	Id: <i>int(11)</i>	Tabel untuk memasukkan <i>Request for Quotation</i>
		Rfq_number: <i>varchar(35)</i>	
		Req_header_id: <i>int(11)</i>	
		Creator: <i>varchar(20)</i>	
		Status: <i>enum</i>	
		Due_date: <i>date</i>	
		Currency: <i>int(11)</i>	
		Supplier_name: <i>int(11)</i>	
		Shipto: <i>int(11)</i>	
		Created_at: <i>varchar(45)</i>	
		Updated_at: <i>varchar(45)</i>	
4	Pur_rfq_line	Id: <i>int(11)</i>	Tabel untuk memasukkan
		Rfq_header_id: <i>int(11)</i>	

No	Nama Tabel	Atribut	Keterangan
		Item number: <i>int(11)</i>	item detail pada <i>Request for Quotation</i>
		Line description: <i>text</i>	
		Max quantity: <i>double</i>	
		Min quantity: <i>double</i>	
		Uom: <i>int(11)</i>	
		Target price: <i>double</i>	
5	Pur_quotation_header	Id: <i>int(11)</i>	Tabel untuk menyimpan header <i>Quotation</i>
		Rfq_header_id: <i>int(11)</i>	
		Quo_number: <i>varchar(40)</i>	
		Header_status: <i>varchar(20)</i>	
		Quo_status: <i>varchar(20)</i>	
		Valid_date: <i>date</i>	
		Price_date: <i>date</i>	
		Total amount: <i>double</i>	
		Currency: <i>int(20)</i>	
		Supplier: <i>int(11)</i>	
		Duration_of_delivery: <i>int(11)</i>	
		Created_at: <i>varchar(45)</i>	
		Updated_at: <i>varchar(45)</i>	
6	Pur_quotation_line	Id: <i>int(11)</i>	Tabel untuk menyimpan rincian item dari <i>Quotation</i>
		Quo_header_id: <i>int(11)</i>	
		Item number: <i>int(11)</i>	
		Item_desc: <i>text</i>	
		Min quantity: <i>double</i>	
		Max quantity: <i>double</i>	
		Uom: <i>int(11)</i>	
		Unit price: <i>double</i>	
		Line price: <i>double</i>	
7	Pur_po_header	Id: <i>int(11)</i>	Tabel untuk memasukkan data <i>Purchase</i>
		Po_number: <i>varchar(45)</i>	
		Req_header_id: <i>int(11)</i>	

No	Nama Tabel	Atribut	Keterangan
		Quo_header_id: <i>int(11)</i>	<i>Order</i>
		Status: <i>varchar(20)</i>	
		Supplier_name: <i>int(11)</i>	
		Creator: <i>varchar(20)</i>	
		Agreement_start_date: <i>date</i>	
		Need_by_date: <i>date time</i>	
		Shipto: <i>int(11)</i>	
		Billto: <i>int(11)</i>	
		Total amount: <i>double</i>	
		Total price: <i>double</i>	
		Currency: <i>int(11)</i>	
		Created_at: <i>varchar(45)</i>	
		Updated_at: <i>varchar(45)</i>	
		Ap_transaction: <i>int(11)</i>	
		Ap_transaction_status: <i>varchar(20)</i>	
		Receipt status: <i>int(11)</i>	
8	Pur_po_line	Id: <i>int(11)</i>	Tabel untuk memasukkan rincian item pada <i>Purchase Order</i>
		Po_header_id: <i>int(11)</i>	
		Item number: <i>int(11)</i>	
		Item description: <i>text</i>	
		Quantity: <i>double</i>	
		Uom: <i>int(10)</i>	
		Unit price: <i>double</i>	
		Line price: <i>double</i>	

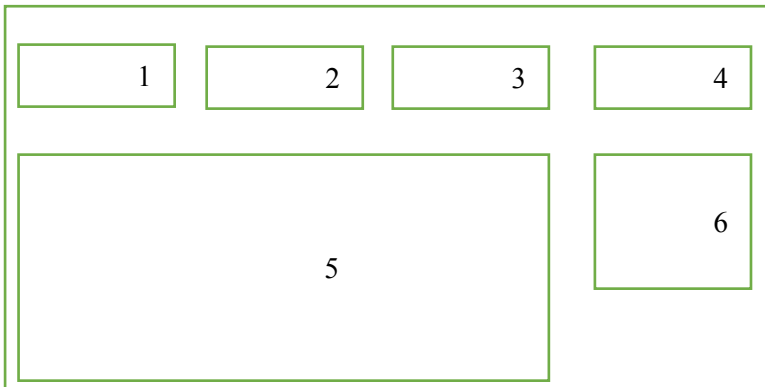
3.2.4 Perancangan Tampilan

Pada subbab ini membahas perancangan antarmuka yang akan digunakan dalam aplikasi ERP khususnya modul

purchasing. Ilustrasi perancangan ditunjukkan sebagai gambar disertai keterangan objek-objek yang ada di dalamnya.

3.2.4.1 Perancangan Antarmuka Halaman *Dashboard*

Rancangan antarmuka halaman *dashboard* ditunjukkan pada gambar di bawah ini. Pada halaman ini ditujukan untuk pengguna agar dapat memonitor keseluruhan modul *purchasing*.



Gambar 3.22 Rancangan Antarmuka Halaman Dashboard

1. Jumlah *Supplier*.
2. *Incomplete Requisition*
3. *Purchase Order today*
4. Total *Expense monthly*
5. Grafik sejarah harga barang
6. Total persentase *purchase order*

3.2.4.2 Perancangan Antarmuka Halaman *Purchase Requisition*

Rancangan antarmuka halaman *purchase requisition* ditunjukkan oleh gambar 3.23. Pada halaman ini, pengguna dapat membuat permintaan barang yang baru.

Gambar 3.23 Rancangan Antarmuka Halaman Purchase Requisition

1. *Input Status*
2. *Input Creator*
3. *Input Dropdown Shipto*
4. *Input Date Need by Date*
5. *Input Dropdown Item Name*
6. *Input Dropdown item type*
7. *Input line description*
8. *Input quantity*
9. *Input Uom id*
10. *Tombol Create*

3.2.4.3 Perancangan Antarmuka Halaman *Request for Quotation*

Rancangan antarmuka halaman *request for quotation* ditunjukkan oleh gambar 3.24. Pada halaman ini, pengguna dapat membuat permintaan untuk *quotation* atau surat penawaran kepada *supplier*.

1

2 3

4 5

6 7

8

9 10 11 12 13

14

Gambar 3.24 Rancangan Antarmuka Halaman Request for Quotation

1. *Input RfQ Number*
2. *Input Dropdown Req Header Id*
3. *Input Creator*
4. *Input Dropdown Status*
5. *Input Date Due Date*
6. *Input Dropdown Currency*
7. *Input Dropdown Supplier Name*
8. *Input Dropdown Shipto*
9. *Input Dropdown Item Name*
10. *Input line description*
11. *Input quantity*
12. *Input Uom id*
13. *Tombol Create*

3.2.4.4 Perancangan Antarmuka Halaman *Maintain Quotation*

Rancangan antarmuka halaman *maintain quotation* ditunjukkan oleh gambar 3.25. Pada halaman ini, pengguna dapat mencatat *quotation* dari *supplier*.

The diagram illustrates the layout of the 'Maintain Quotation' interface. It consists of several key components:

- 1:** A large rectangular area at the top, intended for the Quotation Header.
- 2, 3, 4:** A row of three input fields below the header, used for entering Quotation Line, Tax, and Shipment details.
- 5:** A small rectangular button at the bottom left, representing the 'Create' action.

Gambar 3.25 Rancangan Antarmuka Halaman Maintain Quotation

1. *Form Input untuk Quotatin Header*
2. *Form Input untuk Quotation Line*
3. *Form Input untuk Tax*
4. *Form Input untuk Shipment*
5. *Tombol Create*

3.2.4.5 Perancangan Antarmuka Halaman *Purchase Order*

Rancangan antarmuka halaman *purchase order* ditunjukkan oleh gambar 3.26. Pada halaman ini, pengguna dapat membuat surat permintaan yang akan dikirim ke *supplier*.

The diagram shows a rectangular form layout with five numbered components:

- 1. A large rectangular input field at the top, representing the Purchase Order Header.
- 2, 3, and 4. Three small rectangular input fields arranged horizontally below the header, representing Purchase Order Line, Tax, and Shipment.
- 5. A single rectangular button at the bottom left, representing the Create button.

Gambar 3.26 Rancangan Antarmuka Halaman Purchase Order

1. *Form Input untuk Purchase Order Header*
2. *Form Input untuk Purchase Order Line*
3. *Form Input untuk Tax*
4. *Form Input untuk Shipment*
5. *Tombol Create*

3.2.4.6 Perancangan Antarmuka Halaman *Return Order*

Rancangan antarmuka halaman *return order* ditunjukkan oleh gambar 3.27. Pada halaman ini, pengguna dapat membuat surat pengembalian yang akan dikirim ke *supplier*.

The diagram shows a wireframe for a Return Order form. It consists of a large rectangular area labeled '1' at the top, which is the header input form. Below this, there are three smaller rectangular input fields labeled '2', '3', and '4' arranged horizontally. Below these three fields is a larger rectangular input area. At the bottom left, there is a small rectangular button labeled '5'.

Gambar 3.27 Rancangan Antarmuka Halaman Return Order

1. *Form Input untuk Return Order Header*
2. *Form Input untuk Return Order Line*
3. *Form Input untuk Tax*
4. *Form Input untuk Shipment*
5. *Tombol Create*

3.2.4.7 Perancangan Antarmuka Halaman *Supplier Analysis*

Rancangan antarmuka halaman *supplier analysis* ditunjukkan oleh gambar 3.28. Pada halaman ini, pengguna dapat melihat *ranking supplier*.

The diagram shows a rectangular frame representing the interface layout. Inside the frame, there are five numbered components:

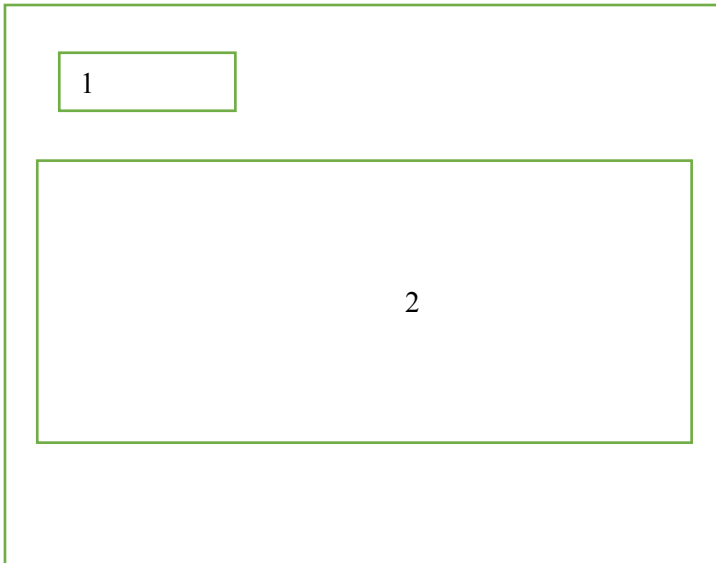
- 1: A small rectangular input field.
- 2: A small rectangular dropdown menu.
- 3: A long horizontal rectangular input field.
- 4: A small rectangular submit button.
- 5: A large rectangular area for displaying the supplier ranking.

Gambar 3.28 Rancangan Antarmuka Halaman Supplier Analysis

1. *Input From Date*
2. *Input Dropdown Item*
3. *Input Criteria's Weight*
4. *Tombol Submit*
5. *Ranking Supplier*

3.2.4.8 Perancangan Antarmuka Halaman *Item Price History*

Rancangan antarmuka halaman *item price history* ditunjukkan oleh gambar 3.29. Pada halaman ini, pengguna dapat melihat *item price history* atau sejarah harga barang.

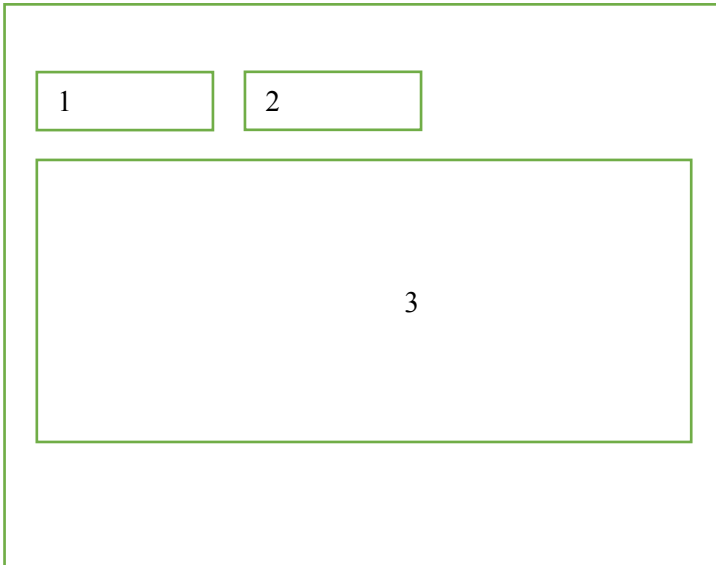


Gambar 3.29 Rancangan Antarmuka Halaman Item Price History

1. *Input Item Name*
2. *Grafik harga Item*

3.2.4.9 Perancangan Antarmuka Halaman *Item Demand History*

Rancangan antarmuka halaman *item demand history* ditunjukkan oleh gambar 3.30. Pada halaman ini, pengguna dapat melihat *item demand history*.



Gambar 3.30 Rancangan Antarmuka Halaman Item Demand History

1. *Input Item Name*
2. *Input Year*
3. *Grafik sejarah permintaan barang*

3.2.4.10 Perancangan Antarmuka Halaman *Item Forecast*

Rancangan antarmuka halaman *item forecast* ditunjukkan oleh gambar 3.31. Pada halaman ini, pengguna dapat melihat ramalan kebutuhan barang.

The wireframe shows a rectangular container with a green border. At the top, there are two input fields labeled '1' and '2'. Below these is a table with four columns, labeled '3', '4', '5', and '6' from left to right.

1	2		
3	4	5	6

Gambar 3.31 Rancangan Antarmuka Halaman Item Forecast

1. *Input Month*
2. *Input Year*
3. *Show Item Name*
4. *Show Item Purchased*
5. *Show Item in Inventory*
6. *Show Item Forecast*

3.2.4.11 Perancangan Antarmuka Halaman *Purchase Order Report*

Rancangan antarmuka halaman *item purchase order report* ditunjukkan oleh gambar 3.32. Pada halaman ini, pengguna dapat melihat laporan pembelian untuk periode tertentu.

The image shows a wireframe of a web interface for a Purchase Order Report. It consists of a large outer rectangle containing five smaller rectangular boxes. Box 1 is a date input field. Box 2 is another date input field. Box 3 is a dropdown menu. Box 4 is a submit button. Box 5 is a large rectangular area, likely for displaying the report content.

Gambar 3.32 Rancangan Antarmuka Halaman Purchase Order Report

1. *Input From Date*
2. *Input To Date*
3. *Input Dropdown Supplier*
4. *Tombol Submit*
5. *Show Purchase Report*
6. *Show Item Forecast*

BAB IV IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem ERP 2016. Di dalamnya mencakup penjelasan lingkungan pengembangan sistem serta proses implementasi *distributed database*, *RBAC*, *multi-tenancy*, dan 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 yang digunakan pada server adalah MySQL Cluster.
2. 2 PC untuk server database menggunakan Ubuntu 12.04.
3. PC untuk server menggunakan Intel® Core™ i3-2120 @3.30GHz , RAM 4GB dengan Sistem Operasi Windows 8.1 Enterprise x64.
4. StarUML untuk pembuatan diagram, Power Designer 12.5, Sublime sebagai teks editor, Pencil untuk pembuatan desain antarmuka, Power Designer untuk pembuatan CDM dan PDM.
5. Mozilla Firefox 46.0.1 dan Chrome 49.0 sebagai antarmuka untuk pengujian aplikasi klien.

4.2 Implementasi Distributed Database

Pada bagian ini akan dijelaskan mengenai implementasi *distributed database* yang terdapat pada sistem ERP 2016. Secara rinci mengenai implementasi *distributed database* dijabarkan sebagai berikut:

4.2.1. Instalasi Data dan SQL node pada node1 dan node2

Pada implementasi instalasi dan sql node pada node1 dan node2, dilakukan langkah-langkah sebagai berikut:

1. Membuat grup MySQL pengguna baru, kemudian menambah user MySQL. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.1.

```
shell> groupadd mysql
shell> useradd-g mysql mysql
```

Kode Sumber 4.1 Membuat grup MySQL pengguna baru dan menambah user MySQL

2. Mengubah lokasi ke dalam direktori yang berisi file yang telah didownload, kemudian mengubah arsip dan menciptakan symlink ke dalam direktori mysql yang bernama "mysql". Hal yang perlu diperhatikan adalah, file yang sebenarnya dan nama direktori bervariasi sesuai dengan jumlah cluster versi MySQL. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.2.

```
hell> cd/usr/local
hell>/usr/local$ tar xzvf mysql-cluster-gpl-7.1.34-
linux-x86_64-glibc23
hell>ln-s/usr/local/mysql-cluster-gpl-7.1.34-linux-
x86-glibc23/usr/local/mysql
hell> export PATH = $ PATH:/usr/local/mysql/bin
hell> echo "export PATH=\$PATH:/usr/local/
mysql/bin">> /etc/bash.bashrc
```

Kode Sumber 4.2 Mengubah lokasi ke dalam direktori yang berisi file yang telah didownload, mengubah arsip dan menciptakan symlink ke dalam direktori mysql.

3. Mengubah lokasi ke dalam direktori mysql dan menjalankan script untuk menciptakan *database system*. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.3.

```
shell> cd mysql
shell>. / scripts / mysql_install_db-user = mysql
```

Kode Sumber 4.3 Mengubah lokasi ke direktori mysql.

4. Mengatur izin yang diperlukan oleh server MSQL. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.4.

```
ll> chown-R root.
shell> chown-R mysql data
shell> chgrp-R mysql.
```

Kode Sumber 4.4 Mengatur izin yang diperlukan oleh server MSQL.

5. Menyalin *script startup MySQL* ke direktori yang sesuai, mengubah menjadi executable, dan memulai ketika sistem beroperasi. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.5.

```
shell> cp support-files/mysql.server / etc /
init.d / mysql
shell> chmod + x / etc / init.d / mysql
shell> update-rc.d mysql defaults
```

Kode Sumber 4.5 Menyalin script startup MySQL ke direktori yang sesuai, mengubah menjadi executable, dan memulai ketika sistem beroperasi

4.2.2. Pemasangan Node Manajemen pada node03

Pemasangan node manajemen memerlukan manajemen server MySQL Cluster (*ndb_mgmd*), diasumsikan bahwa *mysql-cluster-gpl-7.1.5-linux-i686-glibc23.tar.gz* telah ditempatkan di */var / tmp*. Untuk memasang *ndb_mgmd* dan *ndb_mgm* pada host Cluster, sistem sebagai root melakukan langkah-langkah sebagai berikut:

1. Mengubah lokasi ke dalam direktori */ var / tmp* direktori, dan mengekstrak *ndb_mgm* dan *ndb_mgmd* dari arsip ke direktori yang sesuai seperti */ usr / local / bin*. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.6.

```
shell> cd/usr/local
```

```
shell> tar-zxvf mysql-cluster-gpl-7.1.34-linux-
x86-glibc23.tar.gz
shell> cd /usr/local/mysql-cluster-gpl-7.1.34-
linux-x86-glibc235
shell> cp bin / ndb mgm */usr/local/bin
```

Kode Sumber 4.6 Mengubah lokasi ke dalam direktori / var / tmp direktori, mengekstrak ndb_mgm dan ndb_mgmd dari arsip ke direktori yang sesuai seperti / usr / local / bin

2. Mengubah lokasi ke dalam direktori tempat file disalin, kemudian dieksekusi. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.7.

```
shell> cd/usr/local/bin
shell> chmod +x ndb mgm*
```

Kode Sumber 4.7 Mengubah lokasi ke dalam direktori tempat file disalin, kemudian dieksekusi

4.2.3. Konfigurasi Manajemen Node

Konfigurasi pada manajemen node dilakukan dengan langkah-langkah sebagai berikut:

1. Membuat direktori tempat file konfigurasi ditemukan kemudian membuat file itu sendiri. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.8.

```
shell> mkdir / var / lib / mysql-cluster
shell> cd / var / lib / mysql-cluster
shell> vi config.ini
```

Kode Sumber 4.8 Membuat direktori tempat file konfigurasi ditemukan kemudian membuat file itu sendiri

2. Mengatur file “config.ini”. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.9.

```

[ndbd default]
NoOfReplicas=2
DataMemory=10G
IndexMemory=2G
MaxNoOfAttributes=10000
MaxNumberOfTables=2500
MaxOfOrderedIndexes=4086
MaxOfConcurrentOperations=250000
MaxOfConcurrentOperations=250000
[tcp default]
[ndb_mgmd]
hostname=10.151.64.182
datadir=/var/lib/mysql-cluster
[ndbd]
hostname=10.151.64.182
datadir=/usr/local/mysql/data
[ndbd]
hostname=10.151.64.203
datadir=/usr/local/mysql/data
[mysqld]
MaxNoOfAttributes=10000
hostname=10.151.64.182
[mysqld]
MaxNoOfAttributes=10000
hostname=10.151.64.203

```

Kode Sumber 4.9 Mengatur file “config.ini”

4.2.4. Konfigurasi Data dan SQL Node

Konfigurasi data dan SQL Node dilakukan dengan cara mengedit file my.cnf pada direktori /etc/. Untuk setiap data node dan SQL node yang diatur pada my.cnf. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.10.

```

[client]
port=3306
socket=/tmp/mysql.sock
[mysqld]
port=3306
socket=/tmp/mysql.sock
ndbcluster
ndb-connectstring=10.151.64.181
[mysql_cluster]
ndb-connectstring=10.151.64.181

```

Kode Sumber 4.10 Data dan SQL Node

4.2.5. Memulai MySQL Cluster

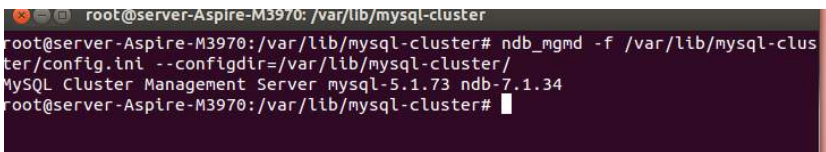
Setiap proses node cluster harus dimulai secara terpisah. Manajemen node harus dimulai terlebih dahulu, kemudian node data. Pada setiap node SQL dilakukan langkah sebagai berikut :

1. Pada node03 (host manajemen), untuk memulai proses manajemen node dari shell sistem dilakukan perintah berikut. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.11.

```
shell>      ndb_mgmd      -f      /var/lib/mysql-  
cluster/config.ini--configdir=/var/lib/mysql-  
cluster/
```

Kode Sumber 4.11 Memulai proses manajemen node

Jika berhasil akan muncul tampilan yang ditunjukkan oleh gambar 4.1



```
root@server-Aspire-M3970: /var/lib/mysql-cluster  
root@server-Aspire-M3970:/var/lib/mysql-cluster# ndb_mgmd -f /var/lib/mysql-clus  
ter/config.ini --configdir=/var/lib/mysql-cluster/  
MySQL Cluster Management Server mysql-5.1.73 ndb-7.1.34  
root@server-Aspire-M3970:/var/lib/mysql-cluster# █
```

Gambar 4.1 Memulai Proses Manajemen Node

2. Jalankan perintah untuk memulai ndbd dan proses mysql server pada masing-masing Data/host SQL. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.12.

```
shell> /usr/local/mysql/bin/ndbd
```

Kode Sumber 4.12 Memulai proses ndbd dan proses mysql server

Jika berhasil akan keluar tampilan seperti Gambar 4.2


```

root@master-Aspire-M3970: /home/master
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ^C^C^C -- exit!
Aborted
root@master-Aspire-M3970: /home/master# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Connected to Management Server at: 10.151.64.181:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0)
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)

[mysqld(API)] 2 node(s)
id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)

ndb_mgm>

```

Gambar 4.2 ndbd dan proses mysql server dapat dimulai

3. Mengaktifkan mysql pada data node. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.13.

```
shell> / etc / init.d / mysql start
```

Kode Sumber 4.13 Memulai proses ndbd dan proses mysql server

Jika berhasil yang maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.3

```

root@master-Aspire-M3970: /home/master
root@master-Aspire-M3970: /home/master# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.73-ndb-7.1.34-cluster-gpl MySQL Cluster Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| db_baru |
| db_baru_1 |
| ezerp |
| mysql |
| ndb_3_fs |
| ndbInfo |
| test |
+-----+
8 rows in set (0.03 sec)

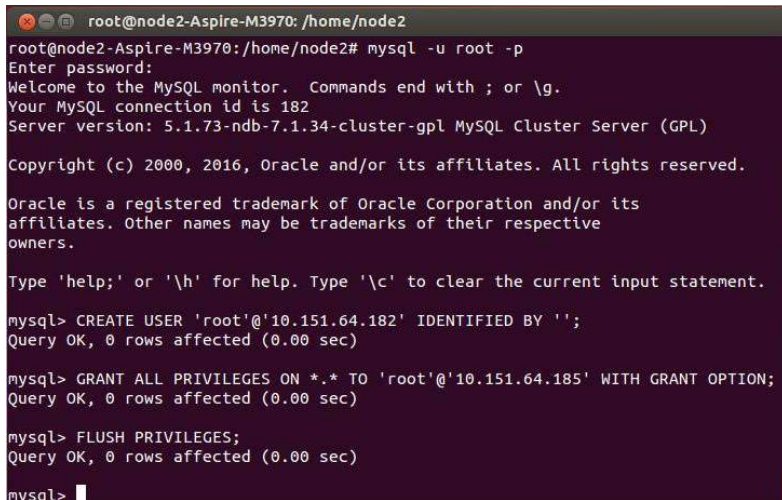
mysql>

```

Gambar 4.3 Mysql pada data node aktif

4. Menguji pada client node dengan perintah `ndb_mgm`. Jika berhasil yang maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.4

5.



```
root@node2-Aspire-M3970: /home/node2
root@node2-Aspire-M3970:/home/node2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 182
Server version: 5.1.73-ndb-7.1.34-cluster-gpl MySQL Cluster Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'root'@'10.151.64.182' IDENTIFIED BY '';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'10.151.64.185' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

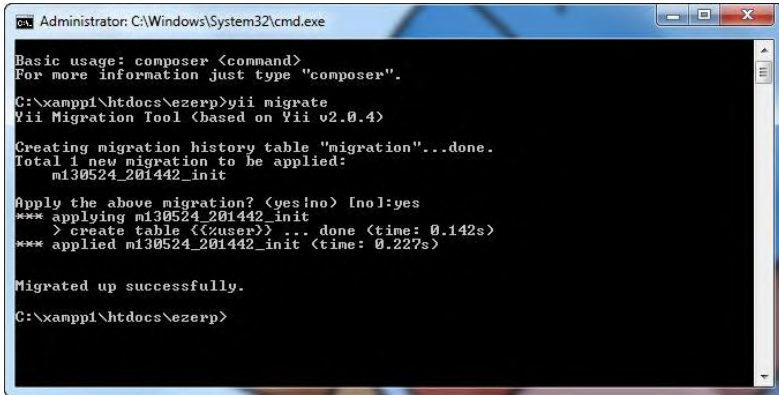
Gambar 4.4 Data Node Saling Terkoneksi

4.3 Implementasi RBAC (Role Base Acces Control)

Pada bagian ini akan dijelaskan mengenai implementasi *RBAC* yang terdapat pada sistem ERP 2016. Secara rinci mengenai implementasi *RBAC* dijabarkan sebagai berikut:

4.3.1. Membuat Tabel Pengguna

Pembuatan tabel pengguna dilakukan dengan menjalankan perintah *yii migrate* pada folder aplikasi. Jika berhasil maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.5.



```

Administrator: C:\Windows\System32\cmd.exe

Basic usage: composer <command>
For more information just type "composer".

C:\xampp\htdocs\ezerp>yii migrate
Yii Migration Tool (based on Yii v2.0.4)

Creating migration history table "migration"...done.
Total 1 new migration to be applied:
  m130524_201442_init

Apply the above migration? (yes/no) [no]:yes
*** applying m130524_201442_init
  > create table <{user}> ... done (time: 0.142s)
*** applied m130524_201442_init (time: 0.227s)

Migrated up successfully.

C:\xampp\htdocs\ezerp>

```

Gambar 4.5 Proses Migrasi Tabel Pengguna Berhasil

4.3.2. Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna

Pada tahap ini dibutuhkan 4 tabel autentifikasi yang terdiri dari:

1. Tabel Item
2. Tabel Child
3. Tabel Assigment
4. Tabel Rule

Cara untuk membuat tabel-tabel tersebut yaitu dengan menjalankan perintah yang ditunjukkan pada Kode Sumber 4.14.

```

Yii migrate --
migrationPath=@yii/rbac/migrations

```

Kode Sumber 4.14 Generate tabel RBAC

Jika berhasil yang maka akan terlihat 4 tabel pada database yaitu table *auth_assignment*, tabel *auth_item*, tabel *auth_item_child* dan tabel *auth_rule*. Tampilan yang ditunjukkan oleh Gambar 4.6, Gambar 4.7, Gambar 4.8, Gambar 4.9

Server: 127.0.0.1 » Database: ezerp » Table: auth_assignment

Browse Structure SQL Search Insert

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	item_name	varchar(64)		No	None	
<input type="checkbox"/>	2	user_id	int(11)		No	None	
<input type="checkbox"/>	3	created_at	int(11)		Yes	NULL	

Gambar 4.6 Proses Migrasi Tabel Pengguna Berhasil

Server: 127.0.0.1 » Database: ezerp » Table: auth_item

Browse Structure SQL Search Insert

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	name	varchar(64)		No	None	
<input type="checkbox"/>	2	type	int(11)		No	None	
<input type="checkbox"/>	3	description	text		Yes	NULL	
<input type="checkbox"/>	4	rule_name	varchar(64)		Yes	NULL	
<input type="checkbox"/>	5	data	text		Yes	NULL	
<input type="checkbox"/>	6	created_at	int(11)		Yes	NULL	
<input type="checkbox"/>	7	updated_at	int(11)		Yes	NULL	

Gambar 4.7 Proses Migrasi Tabel Pengguna Berhasil

Server: 127.0.0.1 » Database: ezerp » Table: auth_item_child

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	parent	varchar(64)			No	None	
2	child	varchar(64)			No	None	

Gambar 4.8 Proses Migrasi Tabel Pengguna Berhasil

Server: 127.0.0.1 » Database: ezerp » Table: auth_rule

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	name	varchar(64)			No	None	
2	data	text			Yes	NULL	
3	created_at	int(11)			Yes	NULL	
4	updated_at	int(11)			Yes	NULL	

Gambar 4.9 Proses Migrasi Tabel Pengguna Berhasil

4.3.3. Membuat Modul Admin

Proses ini bertujuan untuk meletakkan konfigurasi pengguna dan masing-masing model dari 4 tabel autentifikasi. Kemudian dilakukan konfigurasi pada folder config file web.php. Terdapat 3 konfigurasi yaitu:

1. Admin : digunakan sebagai akses ke modul admin
2. Auth Manager : sebagai autentifikasi di yii2 dan mengatur role default sebagai *guest*

3. Session Time Out : mengatur durasi time out session selama 5 menit atau 300 detik.

Masing –masing konfigurasi tersebut ditunjukkan pada kode sumber

```
$config = [
    //...
    'modules' => [
        'admin' => [
            'class'=>'app\modules\admin\AdminModule,
        ],
    ],
    'components' => [
        'authManager' => [
            'class' => 'yii\rbac\DbManager',
            'defaultRoles' => ['guest'],
        ],
        'session' => [
            'timeout' => 300,
        ],
    ],
    //...
];
```

Kode Sumber 4.15 Pembuatan modul admin dan konfigurasi autentifikasi

4.3.4. Membuat Model Tabel Autentifikasi, *Controller* dan *View* Pengguna

Pada tahap ini dilakukan *generate* kelas model dari masing-masing tabel autentifikasi pada modul admin yang diperlukan pada tahap sebelumnya. Kemudian ditambahkan generate kelas *controller* dan *view* pada tabel pengguna. Proses generate ini menggunakan *yii generator* yang telah disediakan oleh framework yii.

4.3.5. Menambahkan Kode pada Kelas *usercontroller*

Pada modul admin, file *usercontroller.php* ditambahkan kode fungsi untuk semua tabel autentifikasi. Masing-masing fungsi ditunjukkan pada Kode Sumber 4.16, Kode Sumber 4.17, Kode Sumber 4.18.

```

public function actionAuthItem()
{
    $auth = Yii::$app->authManager;

    // menambahkan akses sebagai admin ke
    tabel auth_item
        $admin = $auth-
    >createPermission('admin');
        $admin->description = 'Allow user to
    access all page';
        $auth->add($admin);

    // menambahkan akses sebagai asset
    management manajer ke tabel auth_item
        $am_manager = $auth-
    >createPermission('am-manager');
        $am_manager->description = 'Allow
    user as Asset Management Manager';
        $auth->add($am_manager);

    // menambahkan akses sebagai asset
    management staff ke tabel auth_item
        $am_staff = $auth-
    >createPermission('am-staff');
        $am_staff->description = 'Allow user
    as Asset Management Staff';
        $auth->add($am_staff);

    // menambahkan akses sebagai account
    payable manajer ke tabel auth_item
        $ap_manager = $auth-
    >createPermission('ap-manager');
        $ap_manager->description = 'Allow
    user as Account Payable Manager';
        $auth->add($ap_manager);

    // menambahkan akses sebagai account
    payable staff ke tabel auth_item
        $ap_staff = $auth-
    >createPermission('ap-staff');
        $ap_staff->description = 'Allow user
    as Account Payable Staff';
        $auth->add($ap_staff);

    // menambahkan akses sebagai account
    receivable manajer ke tabel auth_item
        $ar_manager = $auth-
    >createPermission('ar-manager');
        $ar_manager->description = 'Allow

```

```

user as Account Receivable Manager';
    $auth->add($ar_manager);

    // menambahkan akses sebagai account
receivable staff ke tabel auth_item
    $ar_staff = $auth-
>createPermission('ar-staff');
    $ar_staff->description = 'Allow user
as Account Receivable Staff';
    $auth->add($ar_staff);

    // menambahkan akses sebagai finance
manajer ke tabel auth_item
    $fi_manager = $auth-
>createPermission('fi-manager');
    $fi_manager->description = 'Allow
user as Finance Manager';
    $auth->add($fi_manager);

    // menambahkan akses sebagai finance
staff ke tabel auth_item
    $fi_staff = $auth-
>createPermission('fi-staff');
    $fi_staff->description = 'Allow user
as Finance Staff';
    $auth->add($fi_staff);

    // menambahkan akses sebagai
accounting/general ledger manajer ke tabel auth_item
    $gl_manager = $auth-
>createPermission('gl-manager');
    $gl_manager->description = 'Allow
user as General Ledger Manager';
    $auth->add($gl_manager);

    // menambahkan akses sebagai
accounting/general ledger staff ke tabel auth_item
    $gl_staff = $auth-
>createPermission('gl-staff');
    $gl_staff->description = 'Allow user
as General Ledger Staff';
    $auth->add($gl_staff);

    // menambahkan akses sebagai human
resource management manajer ke tabel auth_item
    $hrm_manager = $auth-
>createPermission('hrm-manager');
    $hrm_manager->description = 'Allow
user as Human and Resource Manager';
    $auth->add($hrm_manager);

    // menambahkan akses sebagai human

```



```

resource management staff ke tabel auth_item
    $hrm_staff = $auth-
>createPermission('hrm-staff');
    $hrm_staff->description = 'Allow
user as Human and Resource Staff';
    $auth->add($hrm_staff);

    // menambahkan akses sebagai
inventory and warehouse management manajer ke tabel
auth_item
    $iwm_manager = $auth-
>createPermission('iwm-manager');
    $iwm_manager->description = 'Allow
user as Inventory and Warehouse Management Manager';
    $auth->add($iwm_manager);

    // menambahkan akses sebagai
inventory and warehouse management staff ke tabel
auth_item
    $iwm_staff = $auth-
>createPermission('iwm-staff');
    $iwm_staff->description = 'Allow
user as Inventory and Warehouse Management Staff';
    $auth->add($iwm_staff);

    // menambahkan akses sebagai
production planning manajer ke tabel auth_item
    $pp_manager = $auth-
>createPermission('pp-manager');
    $pp_manager->description = 'Allow
user as Producton Planning Manager';
    $auth->add($pp_manager);

    // menambahkan akses sebagai
production planning staff ke tabel auth_item
    $pp_staff = $auth-
>createPermission('pp-staff');
    $pp_staff->description = 'Allow user
as Producton Planning Staff';
    $auth->add($pp_staff);

    // menambahkan akses sebagai
purchasing manajer ke tabel auth_item
    $pur_manager = $auth-
>createPermission('pur-manager');
    $pur_manager->description = 'Allow
user as Purchasing Manager';
    $auth->add($pur_manager);

    // menambahkan akses sebagai
purchasing staff ke tabel auth_item

```

```

        $pur_staff = $auth-
>createPermission('pur-staff');
        $pur_staff->description = 'Allow
user as Purchasing Staff';
        $auth->add($pur_staff);

        // menambahkan akses sebagai sales
and distribution manajer ke tabel auth_item
        $sd_manager = $auth-
>createPermission('sd-manager');
        $sd_manager->description = 'Allow
user as Sales and Distribution Manager';
        $auth->add($sd_manager);

        // menambahkan akses sebagai sales
and distribution staff ke tabel auth_item
        $sd_staff = $auth-
>createPermission('sd-staff');
        $sd_staff->description = 'Allow user
as Sales and Distribution Staff';
        $auth->add($sd_staff);
    }

```

Kode Sumber 4.16 Kode Fungsi Tabel AuthItem

```

public function actionItemChild(){
    $auth = Yii::$app->authManager;

    //admin dapat mengakses semua daftar izin akses
    $am_manager = $auth->createPermission('am-manager');
    $am_staff = $auth->createPermission('am-staff');
    $ap_manager = $auth->createPermission('ap-manager');
    $ap_staff = $auth->createPermission('ap-staff');
    $ar_manager = $auth->createPermission('ar-manager');
    $ar_staff = $auth->createPermission('ar-staff');
    $fi_manager = $auth->createPermission('fi-manager');
    $fi_staff = $auth->createPermission('fi-staff');
    $gl_manager = $auth->createPermission('gl-manager');
    $gl_staff = $auth->createPermission('gl-staff');
    $hrm_manager = $auth->createPermission('hrm-
manager');
    $hrm_staff = $auth->createPermission('hrm-staff');
    $iwm_manager = $auth->createPermission('iwm-
manager');
    $iwm_staff = $auth->createPermission('iwm-staff');
    $pp_manager = $auth->createPermission('pp-manager');
    $pp_staff = $auth->createPermission('pp-staff');
    $pur_manager = $auth->createPermission('pur-
manager');
    $pur_staff = $auth->createPermission('pur-staff');
    $sd_manager = $auth->createPermission('sd-manager');

```

```

$sd_staff = $auth->createPermission('sd-staff');

$admin = $auth->createRole('admin');
$auth->add($admin);
$auth->addChild($admin, $am_manager);
$auth->addChild($admin, $am_staff);
$auth->addChild($admin, $ap_manager);
$auth->addChild($admin, $ap_staff);
$auth->addChild($admin, $ar_manager);
$auth->addChild($admin, $ar_staff);
$auth->addChild($admin, $fi_manager);
$auth->addChild($admin, $fi_staff);
$auth->addChild($admin, $gl_manager);
$auth->addChild($admin, $gl_staff);
$auth->addChild($admin, $hrm_manager);
$auth->addChild($admin, $hrm_staff);
$auth->addChild($admin, $iwm_manager);
$auth->addChild($admin, $iwm_staff);
$auth->addChild($admin, $pp_manager);
$auth->addChild($admin, $pp_staff);
$auth->addChild($admin, $pur_manager);
$auth->addChild($admin, $pur_staff);
$auth->addChild($admin, $sd_manager);
$auth->addChild($admin, $sd_staff);
}

```

Kode Sumber 4.17 Kode Fungsi Tabel ItemChild

```

public function actionAuthAssignment(){
    $auth = Yii::$app->authManager;

    $admin = $auth->createRole('admin');

    $auth->assign($admin, 1);
}

```

Kode Sumber 4.18 Kode Fungsi Tabel Auth Assignment

4.4 Implementasi Multi-tenancy

Pada subbab ini akan dibahas mengenai implementasi alur proses aplikasi yang telah dirancang pada Bab III. Alur proses aplikasi akan dibahas mulai dari pengambilan data partisipan, hingga proses peningkatan level dan penghentian pada setiap *training*. Secara rinci mengenai implementasi *RBAC* dijabarkan sebagai berikut:

4.4.1. Membuat Halaman Muka Tenant

Pada implementasi membuat halaman tenant ini dilakukan pembuatan halaman tenant secara sederhana, kemudian ditambahkan pembuatan database untuk setiap tenant. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.19.

```
// mendapatkan nilai yang dimasukkan dari view
$host = "10.151.64.182";
    $tenant = Yii::$app->request->post('tenant');
    $database = Yii::$app->request->post('database');

// membuka koneksi pada untuk memasukkan data tenant
$koneksidb = mysqli_connect($host, "root", "",
    "multitenant");
if ($koneksidb->connect_error) {
    die("Connection failed: " . $koneksidb->connect_error);
}
    // query untuk memasukkan data tenant
    $insert = "insert into tenant
    (tenant,database_name,created_at)
    ('.$tenant.','.$database.',NOW())";

if($tenant != "" && $database != ""){
    $koneksidb->query($insert);
}
// memutuskan koneksi ke server
$koneksidb->close();
```

Kode Sumber 4.19 Pembuatan Halaman Muka Tenant

4.4.2. Mengubah Konfigurasi Database

Pada tahap ini dilakukan penambahan database untuk tenant baru dengan cara melakukan konfigurasi pada server node mysql. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.20

```
`mysql -u root -p;` lalu tekan enter
`create database <nama database>;` lalu tekan enter
`use <nama database>;` lalu tekan enter
e /tmp/mysql-dump/final-db.sql
```

Kode Sumber 4.20 Penambahan Database Tenant Baru

4.4.3. Login Tenant

Setelah proses pembuatan database dan replikasi pada tahap 4.4.2. selesai, maka tenant melakukan login dengan memilih salah satu database, nama database yang dipilih tersebut disimpan dalam session dan akan digunakan untuk koneksi yang akan dibuat. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.21.

```

$host = '10.151.64.182'; // node mysql
$port = 3306;
$waitTimeoutInSeconds = 1;
$rc =
    @fsockopen($host,$port,$errCode,$errStr,$waitTimeout
    InSeconds);
if(is_resource($rc)){
    $_SESSION['dbserver_ip'] = "10.151.64.182"; // node
    mysql
    // penyimpanan nama database sebagai session untuk
    multitenancy
    $dbname = isset($_SESSION['database_name']) ?
    $_SESSION['database_name'] : 'test';
} else {
    $_SESSION['dbserver_ip'] = "10.151.64.203"; // node
    mysql
}
$connection = [
    'class' => 'yii\db\Connection',
    'dsn' =>
    'mysql:host='.$_SESSION['dbserver_ip'].';dbname='.$d
    bname.'',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8'
];
return $connection;

```

Kode Sumber 4.21 Login Tenant

4.5 Implementasi Program pada Modul *Purchasing*

Pada bagian ini akan dijelaskan mengenai implementasi yang terdapat pada Modul *Purchasing* yang terbagi menjadi tampilan halaman utama atau *dashboard* dan beberapa sub modul,

antara lain: *Purchase Requisition, Request for quotation, Maintain Quotation, Purchase Order, Return Order, dan Report.* Secara rinci mengenai implementasi lapisan antarmuka Modul *Purchasing* dijabarkan sebagai berikut:

4.5.1. Halaman Utama Modul *Purchasing*

Halaman utama modul *Purchasing* menampilkan ringkasan informasi dari beberapa sub modul yang ada pada *Purchasing*. Pengaturan tampilan tersebut diatur di kelas *DefaultController* dalam fungsi *actionIndex()* . Kode tampilan yang dimaksud ditunjukkan oleh kode sumber 4.22

```
class DefaultController extends Controller
{
    public function actionIndex($item = "1")
    {
        $db = Yii::$app->db;
        $supplier = 0;
        $incompletepo = 0;
        $totalpo = 0;
        $incompletereq = 0;
        $thismonth = date('m');
        $today = date('Y-m-d');
        $supplier = Supplier::find()->count();
        $incompletepo = PurPoHeader::find()->where('status =
"Entered" OR status = "Finished")->count();
        $incompletereq = PurRequisitionHeader::find()-
>where('created_at > DATE_SUB(NOW(), INTERVAL 1 DAY)')-
>count();
        $totalpo = PurPoHeader::find()->where('status =
"Published" AND MONTH(updated_at) = '.$thismonth)-
>sum('total_amount');
        $sqlprice = "SELECT avg(pur_po_line.unit_price) as
price, pur_po_header.agreement_start_date as dates from
pur_po_header LEFT JOIN pur_po_line on
pur_po_line.po_header_id = pur_po_header.id WHERE
pur_po_line.item_number = $item AND status = 'published'
group by pur_po_header.agreement_start_date";
        $date = $db->createCommand($sqlprice)->queryAll();
        $popersupplier = "SELECT supplier.supplier_name,
count(pur_po_header.id) as po from pur_po_header LEFT
JOIN supplier on supplier.id =
pur_po_header.supplier_name where pur_po_header.status =
'Published' group by supplier.supplier_name";
        $posupplier = $db->createCommand($popersupplier)-
>queryAll();
```

```

$count = 0;
$priceChart = array();
foreach ($date as $key => $value) {
    $priceChart[$count]['price'] =
round($value['price']);
    $tgl = date('d F Y',
strtotime($value['dates']));
    $priceChart[$count]['date'] = $tgl;
    $count++;
}
$item1 = "SELECT iwm_item_master.id,
iwm_item_master.item_name from iwm_item_master";
$selected = "SELECT item_name from iwm_item_master
WHERE id = $item";
$command3 = $db->createCommand($item1);
$command4 = $db->createCommand($selected);
$items = $command3->queryAll();
$itemname = $command4->queryOne();
return $this->render('index', [
    'supplier' => $supplier,
    'incompletepo' => $incompletepo,
    'incompletereq' => $incompletereq,
    'totalpo' => $totalpo,
    'priceChart' => $priceChart,
    'items' => $items,
    'itemname' => $itemname,
    'posupplier' => $posupplier,
]);
}
}

```

Kode Sumber 4.22. Halaman Utama Modul Purchasing

4.5.2. Melihat Daftar *Purchase Requisition*

Pada implementasi melihat daftar *Purchase Requisition*, sistem menampilkan seluruh daftar permintaan yang berhasil ditambahkan pada *Purchase Requisition*. Pengaturan tampilan ini diatur pada kelas *PurRequisitionHeaderController* fungsi *actionIndex()* Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.23

```

public function actionIndex()
{
    $searchModel = new PurRequisitionHeaderSearch();
    $dataProvider = $searchModel->search(Yii::$app-
>request->queryParams);

```

```

return $this->render('index', [
    'searchModel' => $searchModel,
    'dataProvider' => $dataProvider,
]);
}

```

Kode Sumber 4.23. Melihat Daftar Purchase Requisition

Pada Kode Sumber 4.23, `actionIndex()` berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman utama *purchase requisition*.

4.5.3. Melihat Rincian Purchase Requisition

Pada implementasi melihat rincian *purchase requisition*, sistem menampilkan detail permintaan barang yang disimpan dalam basis data. Pengaturan melihat rincian *purchase requisition* diatur pada kelas *PurRequisitionHeaderController* dan fungsi *actionView()* yang ditunjukkan pada Kode Sumber 4.24.

```

public function actionView($id)
{
    $model = $this->findModel($id);

    $requisitionline= "SELECT pur_requisition_line.id,
pur_requisition_line.req_header_id,
iwm_item_type.item_type_name, iwm_item_master.item_name,
pur_requisition_line.line_description,
pur_requisition_line.quantity,
pur_requisition_line.ordered_qty, uom.uom_name FROM
`pur_requisition_line` left JOIN iwm_item_master on
iwm_item_master.id = pur_requisition_line.item_number left
join iwm_item_type on iwm_item_type.id =
pur_requisition_line.type left join uom on uom.id =
pur_requisition_line.uom WHERE
pur_requisition_line.req_header_id = $model->id";
    $db = Yii::$app->db;
    $command5 = $db->createCommand($requisitionline);
    $results5 = $command5->queryAll();
    $data['dataProvider'] = $results5;
    $providerPurPoHeader = new
\yii\data\ArrayDataProvider([
        'allModels' => $model->purPoHeaders,
    ]);
    $providerPurRequisitionLine = new
\yii\data\ArrayDataProvider([

```



```

        'allModels' => $model->purRequisitionLines,
    ]);
    $providerPurRfqHeader = new
    \yii\data\ArrayDataProvider([
        'allModels' => $model->purRfqHeaders,
    ]);
    $line = $model->purRequisitionLines;
    return $this->render('view', [
        'model' => $this->findModel($id),
        'providerPurPoHeader' => $providerPurPoHeader,
        'providerPurRequisitionLine' =>
        $providerPurRequisitionLine,
        'providerPurRfqHeader' =>
        $providerPurRfqHeader,
        'providerLine' => $results5,
    ], $data);
}

```

Kode Sumber 4.24. Melihat Rincian purchase requisition

Pada Kode Sumber 4.24, *actionView()* berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman rincian *purchase requisition*.

4.5.4. Menambah *Purchase Requisition*

Pada implementasi menambah *Purchase Requisition*, sistem menambahkan permintaan baru ke dalam *Purchase Requisition*. Pengaturan menambah transaksi baru ini diatur pada kelas *PurRequisitionHeaderController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.25

```

public function actionCreate()
{
    $model = new PurRequisitionHeader();

    if ($model->loadAll(Yii::$app->request->post()) &&
        $model->saveAll()) {
        $id = $model['id'];
        $newId =
        str_pad($id, 10, "REQ00000000", STR_PAD_LEFT);
        $model->req_number = $newId;
        $model->order_status = 0;
    }
}

```

```

        $model->update();

        return $this->redirect(['view', 'id' => $model-
>id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4.25 Menambah Purchase Requisition

Fungsi *actionCreate()* pada kode sumber 4.25 berfungsi untuk menambahkan permintaan baru dimana ketika saat ditambahkan, Requisition Number secara otomatis akan terisi dengan format REQ0000000 diakhiri dengan Id-nya. Status akan langsung terisi 0 yang berarti, permintaan tersebut masih belum terpenuhi.

4.5.5. Menyunting *Purchase Requisition*

Pada implementasi menyunting *Purchase Requisition*, sistem memperbarui data pada *Purchase Requisition*. Pengaturan menyunting transaksi ini diatur pada kelas *RequisitionHeaderController* fungsi *actionUpdate()*. Kode untuk menyunting *Purchase Requisition* ditunjukkan pada kode sumber 4.26

```

public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->loadAll(Yii::$app->request->post()) &&
$model->saveAll()) {
        return $this->redirect(['view', 'id' => $model-
>id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4.26 Menyunting Purchase Requisition

4.5.6. Menghapus *Purchase Requisition*

Pada implementasi menghapus *purchase requisition*, sistem menghapus data transaksi pada *purchase requisition*. Pengaturan menyunting transaksi ini diatur pada kelas *RequisitionHeaderController* fungsi *actionDelete()*. Kode untuk menghapus *Purchase Requisition* ditunjukkan pada kode sumber 4.27

```
public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();

    return $this->redirect(['index']);
}
```

Kode Sumber 4.27 Menghapus Purchase Requisition

4.5.7. *Convert to RFQ*

Pada implementasi *convert to RFQ*, sistem merubah *purchase requisition* menjadi *request for quotation*. Pengaturan *convert* transaksi ini diatur pada kelas *RequisitionHeaderController* fungsi *actionConvertrfq()*. Kode untuk *convert to RFQ* ditunjukkan pada kode sumber 4.28

```
public function actionConvertrfq($id)
{
    $model = $this->findModel($id);
    $query1 = "select count(id) as id_req from
pur_rfq_header where req_header_id = $model[id]";
    $db = Yii::$app->db;
    $results1 = $db->createCommand($query1)-
>queryOne();
    if ($results1['id_req'] == NULL)
    {
        $results1['id_req'] = 0;
    }
    else {
        $results1['id_req'] += 1;
    }
    $name = "RfQ-".$model['req_number']."-
```

```

".$results1['id_req'];
    $queryHead=
        "insert
        into
        pur_rfq_header(rfq_number,req_header_id, status, creator,
        due_date, shipto) values ('$name',$model[id], 'Entered',
        '$model[creator]', '$model[need_by_date]',$model[shipto]) ";
    $results1 = $db->createCommand($queryHead)-
>execute();
    $query = "select max(id) as id from
    pur_rfq_header";
    $results = $db->createCommand($query)->queryOne();
    $idbaru = $results['id'];
    foreach ($model['purRequisitionLines'] as $key =>
$value){
        if($value['quantity'] == NULL)
        {
            $value['quantity'] = 0;
        }
        $queryLine =
        "insert
        into
        pur_rfq_line(rfq_header_id, item_number, line_description,
        max_quantity,min_quantity, uom) values
        ($idbaru,$value[item_number], '".addslashes($value['line_desc
        ription'])."', $value[quantity], $value[quantity], $value[uom])
        ";
        $results2 = $db->createCommand($queryLine)-
>execute();
    }
    return $this->redirect('index.php?r=pur/pur-rfq-
header');
}

```

Kode Sumber 4.28 Convert to RFQ

4.5.8. *Convert to PO*

Pada implementasi *convert to PO*, sistem merubah *purchase requisition* menjadi *purchase order*. Pengaturan convert transaksi ini diatur pada kelas *RequisitionHeaderController* fungsi *actionConvertpo()*. Kode untuk *convert to PO* ditunjukkan pada kode sumber 4.29

```

public function actionConvertpo($id)
{
    $model = $this->findModel($id);
    $query = "SELECT id from pur_po_header ORDER BY id
DESC";
    $query1 = "SELECT count(id) as id_req from
pur_po_header where req_header_id = $model[id]";
    $db = Yii::$app->db;
    $results1 = $db->createCommand($query1)-

```

```

>queryOne();
    if ($results1['id_req'] == NULL)
    {
        $results1['id_req'] = 0;
    }
    else {
        $results1['id_req'] += 1;
    }
    $name = "PO-".$model['req_number']."-
".$results1['id_req'];
    $queryHead= "insert into
pur_po_header(po_number,req_header_id, status, creator,
need_by_date, shipto) values ('$name',$model[id], 'Entered',
'$model[creator]', '$model[need_by_date]',$model[shipto]) ";
    $results1 = $db->createCommand($queryHead)-
>execute();
    $query = "SELECT id from pur_po_header ORDER BY id
DESC";
    $results2 = $db->createCommand($query)->queryOne();
    $idbaru = $results2['id'];
    foreach ($model['purRequisitionLines'] as $key =>
$value){
        if($value['quantity'] == NULL)
        {
            $value['quantity'] = 0;
        }
        $queryLine = "INSERT INTO
pur_po_line(po_header_id, item_number, item_description,
quantity, uom) values
($idbaru,$value[item_number], '".addslashes($value['line_desc
ription'])."', $value[quantity], $value[uom])";
        $db->createCommand($queryLine)->execute();
    }
    return $this->redirect('index.php?r=pur/pur-po-
header');
}

```

Kode Sumber 4.29 Convert to PO

4.5.9. Melihat Daftar *Request for Quotation*

Pada implementasi melihat daftar *request for quotation*, sistem menampilkan seluruh daftar *request for quotation*. Pengaturan tampilan ini diatur pada kelas *PurRfqHeaderController* fungsi *actionIndex()* Kode untuk menampilkan seluruh *RFQ* tersebut ditunjukkan pada kode sumber 4.30

```
public function actionIndex()
```

```

{
    $searchModel = new PurRfqHeaderSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

```

Kode Sumber 4.30. Melihat Daftar Request for Quotation

Pada Kode Sumber 4.30, `actionIndex()` berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman utama *request for quotation*.

4.5.10. Melihat Rincian Request for Quotation

Pada implementasi melihat rincian *request for quotation*, sistem menampilkan detail permintaan barang yang disimpan dalam basis data. Pengaturan melihat rincian *request for quotation* diatur pada kelas *PurRfqHeaderController* dan fungsi *actionView()* yang ditunjukkan pada Kode Sumber 4.31.

```

public function actionView($id)
{
    $model = $this->findModel($id);
    $providerPurQuotationHeader = new
    \yii\data\ArrayDataProvider([
        'allModels' => $model->purQuotationHeaders,
    ]);
    $providerPurRfqLine = new
    \yii\data\ArrayDataProvider([
        'allModels' => $model->purRfqLines,
    ]);
    return $this->render('view', [
        'model' => $this->findModel($id),
        'providerPurQuotationHeader' =>
        $providerPurQuotationHeader,
        'providerPurRfqLine' => $providerPurRfqLine,
    ]);
}

```

Kode Sumber 4.31 Melihat Rincian request for quotation

Pada Kode Sumber 4.31, *actionView()* berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman rincian *request for quotation*.

4.5.11. Menambah *Request for Quotation*

Pada implementasi menambah *request for quotation*, sistem menambahkan data baru ke dalam *request for quotation*. Pengaturan menambah data baru ini diatur pada kelas *PurRfqHeaderController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.32.

```
public function actionCreate()
{
    $model = new PurRfqHeader();

    if ($model->loadAll(Yii::$app->request->post()) &&
        $model->saveAll()) {
        return $this->redirect(['view', 'id' => $model-
>id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}
```

Kode Sumber 4.32 Menambah Request for Quotation

Pada kode sumber 4.32, *actionCreate()* berfungsi untuk menambahkan data RFQ baru ke dalam model, apabila tombol *Create* ditekan, maka data yang ada akan disimpan ke dalam basis data dan pengguna akan diarahkan ke halaman rincian *request for quotation*.

4.5.12. Menyunting *Request for Quotation*

Pada implementasi menyunting *request for quotation*, sistem memperbarui data transaksi pada *request for quotation*.

Pengaturan menyunting transaksi ini diatur pada kelas *PurRfqController* fungsi *actionUpdate()*. Kode untuk menyunting *request for quotation* ditunjukkan pada kode sumber 4.33.

```
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->loadAll(Yii::$app->request->post()) &&
        $model->saveAll()) {
        return $this->redirect(['view', 'id' => $model-
>id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}
```

Kode Sumber 4.33 Menyunting Request for Quotation

4.5.13. Menghapus *Request for Quotation*

Pada implementasi menghapus *request for quotation*, sistem menghapus data pada *request for quotation*. Pengaturan menyunting transaksi ini diatur pada kelas *PurRfqHeaderController* fungsi *actionDelete()*. Kode untuk menghapus *request for quotation* ditunjukkan pada kode sumber 4.34.

```
public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();

    return $this->redirect(['index']);
}
```

Kode Sumber 4.34 Menghapus Request for Quotation

4.5.14. Melihat Daftar *Quotation*

Pada implementasi melihat daftar *quotation*, sistem menampilkan seluruh daftar *quotation*. Pengaturan tampilan ini diatur pada kelas *PurQuotationHeaderController* fungsi *actionIndex()* Kode untuk menampilkan seluruh *Quotation* tersebut ditunjukkan pada kode sumber 4.35.


```

public function actionIndex()
{
    $searchModel = new PurQuotationHeaderSearch();
    $dataProvider = $searchModel->search(Yii::$app-
>request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

```

Kode Sumber 4.35. Melihat Daftar Quotation

Pada Kode Sumber 4.35, `actionIndex()` berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman utama *quotation*.

4.5.15. Menambah *Quotation*

Pada implementasi menambah *quotation*, sistem menambahkan data baru ke dalam *quotation*. Pengaturan menambah data baru ini diatur pada kelas *PurQuotationHeaderController* fungsi `actionCreate()`. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.36.

```

public function actionCreate()
{
    $model = new PurQuotationHeader();

    if ($model->loadAll(Yii::$app->request->post()) &&
$model->saveAll()) {

//QuoController::actionPurQuotationHeader($model-
>id,'insert');
        return $this->redirect(['view', 'id' => $model-
>id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4.36 Menambah Quotation

Pada kode sumber 4.36, `actionCreate()` berfungsi untuk menambahkan data *quotation* baru ke dalam model, apabila tombol *Create* ditekan, maka data yang ada akan disimpan ke dalam basis data dan pengguna akan diarahkan ke halaman rician *quotation*.

4.5.16. Melihat Rincian Quotation

Pada implementasi melihat rincian *quotation*, sistem menampilkan detail permintaan barang yang disimpan dalam basis data. Pengaturan melihat rincian *quotation* diatur pada kelas *PurQuotationHeaderController* dan fungsi `actionView()` yang ditunjukkan pada Kode Sumber 4.37.

```
public function actionView($id)
{
    $model = $this->findModel($id);
    $providerPurPoHeader = new
    \yii\data\ArrayDataProvider([
        'allModels' => $model->purPoHeaders,
    ]);
    $providerPurQuotationLine = new
    \yii\data\ArrayDataProvider([
        'allModels' => $model->purQuotationLines,
    ]);
    $providerPurTaxdetail = new
    \yii\data\ArrayDataProvider([
        'allModels' => $model->purTaxdetails,
    ]);
    $providerPurShipdetail = new
    \yii\data\ArrayDataProvider([
        'allModels' => $model->purShipdetails,
    ]);
    return $this->render('view', [
        'model' => $this->findModel($id),
        'providerPurPoHeader' => $providerPurPoHeader,
        'providerPurQuotationLine' =>
        $providerPurQuotationLine,
        'providerPurTaxdetail' =>
        $providerPurTaxdetail,
        'providerPurShipdetail' =>
        $providerPurShipdetail,
    ]);
}
```

Kode Sumber 4.37 Melihat Rincian Quotation

Pada Kode Sumber 4.37, *actionView()* berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman rincian *quotation*.

4.5.17. Menyunting *Quotation*

Pada implementasi menyunting *quotation*, sistem memperbarui data transaksi pada *quotation*. Pengaturan menyunting transaksi ini diatur pada kelas *PurQuotationHeaderController* fungsi *actionUpdate()*. Kode untuk menyunting *quotation* ditunjukkan pada kode sumber 4.38.

```
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->loadAll(Yii::$app->request->post()) &&
        $model->saveAll()) {
        //QuoController::actionPurQuotationHeader($model-
        >id,'update');
        return $this->redirect(['view', 'id' => $model-
        >id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}
```

Kode Sumber 4.38 Menyunting Quotation

4.5.18. Menghapus Quotation

Pada implementasi menghapus *quotation*, sistem menghapus data pada *quotation*. Pengaturan menyunting transaksi ini diatur pada kelas *PurQuotationHeaderController* fungsi *actionDelete()*. Kode untuk menghapus *Quotation* ditunjukkan pada kode sumber 4.39.

```

public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();

    return $this->redirect(['index']);
}

```

Kode Sumber 4.39 Menghapus Quotation

4.5.19. Melihat Daftar *Purchase Order*

Pada implementasi melihat daftar *purchase order*, sistem menampilkan seluruh daftar *purchase order*. Pengaturan tampilan ini diatur pada kelas *PurPoHeaderController* fungsi *actionIndex()* Kode untuk menampilkan seluruh *Purchase Order* tersebut ditunjukkan pada kode sumber 4.40.

```

public function actionIndex()
{
    $searchModel = new PurPoHeaderSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

```

Kode Sumber 4.40. Melihat Daftar Purchase Order

Pada Kode Sumber 4.40, *actionIndex()* berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman utama *purchase order*.

4.5.20. Melihat Rincian *Purchase Order*

Pada implementasi melihat rincian *purchase order*, sistem menampilkan detail permintaan barang yang disimpan dalam basis data. Pengaturan melihat rincian *purchase order* diatur pada kelas *PurPoHeaderController* dan fungsi *actionView()* yang ditunjukkan pada Kode Sumber 4.41.

```

public function actionView($id)
{

```

```

        $model = $this->findModel($id);
        $providerIwmgoodreceipt = new
        \yii\data\ArrayDataProvider([
            'allModels' => $model->iwmgoodreceipts,
        ]);
        $providerPurPoLine = new
        \yii\data\ArrayDataProvider([
            'allModels' => $model->purPoLines,
        ]);

        $providerPurTaxdetail = new
        \yii\data\ArrayDataProvider([
            'allModels' => $model->purTaxdetails,
        ]);
        $providerPurShipdetail = new
        \yii\data\ArrayDataProvider([
            'allModels' => $model->purShipdetails,
        ]);
        return $this->render('view', [
            'model' => $this->findModel($id),
            'providerIwmgoodreceipt' =>
            $providerIwmgoodreceipt,
            'providerPurPoLine' => $providerPurPoLine,
            'providerPurTaxdetail' =>
            $providerPurTaxdetail,
            'providerPurShipdetail' =>
            $providerPurShipdetail,
        ]);
    }

```

Kode Sumber 4.41 Melihat Rincian Purchase Order

Pada Kode Sumber 4.24, *actionView()* berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman rincian *purchase order*.

4.5.21. Menambah *Purchase Order*

Pada implementasi menambah *purchase order*, sistem menambahkan data baru ke dalam *purchase order*. Pengaturan menambah data baru ini diatur pada kelas *PurPoHeaderController* fungsi *actionCreate()*. Saat menambah *purchase order*, jika status diubah menjadi publish, maka akan langsung membuat data *goods receipt* untuk memperingati gudang, lalu langsung memberi tanda juga ke bagian pembayaran,

dan data *purchase order* sudah tidak bisa diubah atau dihapus. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.42.

```
public function actionCreate()
{
    $model = new PurPoHeader();
    if ($model->loadAll(Yii::$app->request->post()) &&
$model->saveAll()) {
        $db = Yii::$app->db;
        $id = $model['id'];
        if ($model['status'] == 'Published')
        {
            PurRequisitionHeaderController::actionUpdateQty($model['req_
header_id']);

            $line = PurPoLine::find()
                ->where('po_header_id = '.$id->all();
            $item = array();
            foreach ($line as $key) {

                $data = array(
                    'item_id' => $key->item_number,
                    'item_qty' => $key->quantity
                );
                array_push($item, $data);
            }

            IwmgoodreceiptController::actionCreateReceipt("Purchase",$i
d,$model['need_by_date'],$item);
        }
        $query1 = "SELECT count(id) as id_req from
pur_po_header where req_header_id = $model[req_header_id]";
        $results1 = $db->createCommand($query1)-
>queryOne();

        $query3 = "SELECT * from pur_requisition_header
where id = $model[req_header_id]";
        $results3 = $db->createCommand($query3)-
>queryOne();
        if ($results1['id_req'] == NULL)
        {
            $results1['id_req'] = 0;
        }
        else {
            $results1['id_req'];
        }
        $newId = "PO-".$results3['req_number']."-
".$results1['id_req'];
        //$newId = str_pad($id,10,"PO-",STR_PAD_LEFT);
    }
}
```

```

        if ($model['po_number'] == NULL)
        {
            $model->po_number = $newId;
        }
        //$model->order_status = 0;
        $model->update();
        return $this->redirect(['view', 'id' => $model-
>id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4.42 Menambah Purchase Order

4.5.22. Menyunting *Return Order*

Pada implementasi menyunting *purchase order*, sistem memperbarui data transaksi pada *quotation*, jika status diubah menjadi *publish*, maka akan langsung membuat data *goods receipt* untuk memperingati gudang, lalu langsung memberi tanda juga ke bagian pembayaran, dan data *purchase order* sudah tidak bisa diubah atau dihapus. Pengaturan menyunting transaksi ini diatur pada kelas *PurPoHeaderController* fungsi *actionUpdate()*. Kode untuk menyunting *purchase order* ditunjukkan pada kode sumber 4.43.

```

public function actionCreate()
{
    $model = new PurPoHeader();
    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {
        $db = Yii::$app->db;
        $id = $model['id'];
        if ($model['status'] == 'Published')
        {
            PurRequisitionHeaderController::actionUpdateQty($model['req_
            header_id']);

            $line = PurPoLine::find()
                ->where('po_header_id = '.$id)->all();
            $item = array();
            foreach ($line as $key) {

                $data = array(
                    'item id' => $key->item number,

```

```

        'item_qty' => $key->quantity
    );
    array_push($item, $data);
}

IwmgoodreceiptController::actionCreateReceipt("Purchase",$id,$model['need_by_date'],$item);
}
$query1 = "SELECT count(id) as id_req from
pur_po_header where req_header_id = $model[req_header_id]";
$results1 = $db->createCommand($query1)-
>queryOne();

$query3 = "SELECT * from pur_requisition_header
where id = $model[req_header_id]";
$results3 = $db->createCommand($query3)-
>queryOne();
if ($results1['id_req'] == NULL)
{
    $results1['id_req'] = 0;
}
else {
    $results1['id_req'];
}
$newId = "PO-".$results3['req_number']."-
".$results1['id_req'];
//$newId = str_pad($id,10,"PO-",STR_PAD_LEFT);
if ($model['po_number'] == NULL)
{
    $model->po_number = $newId;
}
//$model->order_status = 0;
$model->update();
return $this->redirect(['view', 'id' => $model-
>id]);
} else {
    return $this->render('create', [
        'model' => $model,
    ]);
}
}
}

```

Kode Sumber 4.43 Menyunting Purchase Order

4.5.23. Menghapus *Purchase Order*

Pada implementasi menghapus *purchase order*, sistem menghapus data pada *purchase order*. Pengaturan menyunting transaksi ini diatur pada kelas *PurPoHeaderController* fungsi

actionDelete(). Kode untuk menghapus *Purchase Order* ditunjukkan pada kode sumber 4.44.

```
public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();

    return $this->redirect(['index']);
}
```

Kode Sumber 4.44 Menghapus Purchase Order

4.5.24. Melihat Daftar *Return Order*

Pada implementasi melihat daftar *return order*, sistem menampilkan seluruh daftar *return order*. Pengaturan tampilan ini diatur pada kelas *PurRoHeaderController* fungsi *actionIndex()* Kode untuk menampilkan seluruh *Return Order* tersebut ditunjukkan pada kode sumber 4.45

```
public function actionIndex()
{
    $searchModel = new PurRoHeaderSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider, ]);
}
```

Kode Sumber 4.45. Melihat Daftar Return Order

Pada Kode Sumber 4.23, *actionIndex()* berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman utama *return order*.

4.5.25. Melihat Rincian Return Order

Pada implementasi melihat rincian *return order*, sistem menampilkan detail permintaan barang yang disimpan dalam basis data. Pengaturan melihat rincian *return order* diatur pada kelas *PurRoHeaderController* dan fungsi *actionView()* yang ditunjukkan pada Kode Sumber 4.46.

```

public function actionView($id)
    {
        $model = $this->findModel($id);
        $providerIwmgoodreceipt = new
\yii\data\ArrayDataProvider([
    'allModels' => $model->iwmgoodreceipts,
    ]);
        $providerPurPoLine = new
\yii\data\ArrayDataProvider([
    'allModels' => $model->purPoLines,
    ]);
        $providerPurTaxdetail = new
\yii\data\ArrayDataProvider([
    'allModels' => $model->purTaxdetails,
    ]);
        $providerPurShipdetail = new
\yii\data\ArrayDataProvider([
    'allModels' => $model->purShipdetails,
    ]);
Return $this->render('view', ['model'=>$this>
findModel($id),
    'providerIwmgoodreceipt' =>
$providerIwmgoodreceipt,
    'providerPurPoLine' => $providerPurPoLine,
    'providerPurTaxdetail' =>
$providerPurTaxdetail,
    'providerPurShipdetail' =>
$providerPurShipdetail,

```

Kode Sumber 4.46 Melihat Rincian Return Order

Pada Kode Sumber 4.46, *actionView()* berfungsi untuk mengambil data yang berada pada basis data dan kemudian memberikan isi data ke halaman rincian *return order*.

4.5.26. Menambah *Return Order*

Pada implementasi menambah *return order*, sistem menambahkan data baru ke dalam *return order*. Pengaturan manambah data baru ini diatur pada kelas *PurRoHeaderController* fungsi *actionCreate()*. Saat menambah *return order*, jika status diubah menjadi publish dan *return_type* menjadi *return material*, maka akan langsung membuat data

goods receipt untuk memperingati gudang. Jika status diubah menjadi publish dan *return_type* menjadi return payment, maka akan memberitahu bagian pembayaran, dan data *return order* sudah tidak bisa diubah atau dihapus. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.47.

```

public function actionCreate()
{
    $model = new PurRoHeader();

    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {
        $id = $model['id'];
        if ($model['return_status'] == '2' &&
    $model['return_type'] == '0' )
        {
            $line = PurRoLine::find()
                ->where('ro_header_id = '.$id->all();
            $po = PurPoHeader::find()
                ->where('id = '.$model['po_header_id'])-
    >one();

            $item = array();
            foreach ($line as $key) {

                $data = array(
                    'item_id' => $key->item_number,
                    'item_qty' => $key->quantity
                );
                array_push($item, $data);
            }

            IwmgoodreceiptController::actionCreateReceipt("Purchase",$model['po_header_id'],$po['need_by_date'],$item);
        }
        else if ($model['return_status'] == '2' &&
    $model['return_type'] == '1' )
        {
            $db = Yii::$app->db;
            $queryupdatepo = "UPDATE pur_po_header SET
    ap_transaction = '3' WHERE id = $model[id]";
            $results6 = $db->createCommand($queryupdatepo)->execute();

            ApTransactionController::actionReturnStatus($model['id']);
        }
        return $this->redirect(['view', 'id' => $model->id]);
    } else {

```

```

        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4.47 Menambah *Return Order*

4.5.27. Menyunting *Return Order*

Pada implementasi menyunting *return order*, sistem memperbarui data transaksi pada *return order*, Saat menambah *return order*, jika status diubah menjadi publish dan *return_type* menjadi *return material*, maka akan langsung membuat data *goods receipt* untuk memperingati gudang. Jika status diubah menjadi publish dan *return_type* menjadi *return payment*, maka akan memberitahu bagian pembayaran, dan data *return order* sudah tidak bisa diubah atau dihapus. Pengaturan menyunting transaksi ini diatur pada kelas *PurRoHeaderController* fungsi *actionUpdate()*. Kode untuk menyunting *return order* ditunjukkan pada kode sumber 4.48.

```

public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {
        $id = $model['id'];
        if ($model['return_status'] == '2' &&
    $model['return_type'] == '0' )
        {
            $line = PurRoLine::find()
                ->where('ro_header_id = '.$id)->all();
            $po = PurPoHeader::find()
                ->where('id = '.$model['po_header_id'])-
    >one();

            $item = array();
            foreach ($line as $key) {
                $data = array(
                    'item_id' => $key->item_number,
                    'item_qty' => $key->quantity
                );
                array_push($item, $data);
            }

            IwmgoodreceiptController::actionCreateReceipt("Purchase", $model['po_header_id'], $po['need by date'], $item);

```

```

    }

    else if ($model['return_status'] == '2' &&
$model['return_type'] == '1' )
    {
        $db = Yii::$app->db;
        $queryupdatepo = "UPDATE pur_po_header SET
ap_transaction = '3' WHERE id = $model[po_header_id]";
        $results6 = $db-
>createCommand($queryupdatepo)->execute();

ApTransactionController::actionReturnStatus($model['po_heade
r_id']);
    }

    return $this->redirect(['view', 'id' => $model-
>id]);
} else {
    return $this->render('update', [
        'model' => $model,
    ]);
}
}
}

```

Kode Sumber 4.48 Menyunting *Return Order*

4.5.28. Menghapus *Return Order*

Pada implementasi menghapus *return order*, sistem menghapus data pada *return order*. Pengaturan menyunting transaksi ini diatur pada kelas *PurRoHeaderController* fungsi *actionDelete()*. Kode untuk menghapus *Return Order* ditunjukkan pada kode sumber 4.49.

```

public function actionDelete($id)
{
    $this->findModel($id)->deleteWithRelated();

    return $this->redirect(['index']);
}

```

Kode Sumber 4.49 Menghapus *Return Order*

4.5.29. Menganalisa Supplier

Untuk menganalisa supplier, user masuk ke halaman supplier analysis, Pengaturan menganalisa supplier ini terdapat pada kelas *SupplierMasterController* fungsi *actionIndex()* dan fungsi *actionTopsis()*. Kode untuk menganalisa supplier ditunjukkan pada kode sumber 4.50.

```
public function actionIndex($item = NULL, $date = NULL,
    $late = NULL, $defect = NULL, $price = NULL)
    {
        $item = Yii::$app->request->post('item');
        $date = Yii::$app->request->post('date1');
        $late = Yii::$app->request->post('late');
        $defect = Yii::$app->request->post('defect');
        $price = Yii::$app->request->post('price');
        if($item == "") {
            $item = 100000;
        }
        if ($date == "") {
            $date = "2016-01-01 00:00:00";
        }
        if ($late == "") {
            $late = 1;
        }
        if ($defect == "") {
            $defect = 1;
        }
        if ($price == "") {
            $price = 1;
        }
        $data['dataProvider7'] = $date;
        $db = Yii::$app->db;
        if ($item != 100000)
        {
            $query
                = "SELECT
                avg(time_to_sec(timediff(gr_document_date, need_by_date )) /
                3600) as avg_late,
                avg(iwm_good_receipt_f.defect_percent) as
                avg_defect,
                avg(pur_po_line.unit_price) as avg_price
                FROM pur_po_header
                left join iwm_good_receipt_h on pur_po_header.id =
                iwm_good_receipt_h.from_po_id
                left join iwm_good_receipt_f on
                iwm_good_receipt_h.id = iwm_good_receipt_f.gr_id
                left join pur_po_line on pur_po_header.id =
                pur_po_line.po_header id
```

```

        where iwm_good_receipt_f.item_id = '$item' AND
pur_po_line.item_number = '$item'
        AND pur_po_header.agreement_start_date >= '$date'
AND iwm_good_receipt_hreceipt_name NOT LIKE 'Return%';

        $db = Yii::$app->db;
        $command = $db->createCommand($query);
        $results = $command->queryOne();
        $data['dataProvider'] = $results;

        $query5 = "SELECT supplier.supplier_name as
'0',(avg(time_to_sec(timediff(gr_document_date,
need_by_date )) / 3600)) as '1',
        (avg(iwm_good_receipt_f.defect_percent)) as '2',
        (avg(pur_po_line.unit_price)) as '3'
FROM pur_po_header
left join iwm_good_receipt_h on pur_po_header.id =
iwm_good_receipt_h.from_po_id
left join iwm_good_receipt_f on
iwm_good_receipt_h.id = iwm_good_receipt_f.gr_id
left join pur_po_line on pur_po_header.id =
pur_po_line.po_header_id
left join supplier on pur_po_header.supplier_name =
supplier.id
where iwm_good_receipt_f.item_id = '$item' AND
pur_po_line.item_number = '$item'
AND pur_po_header.agreement_start_date >= '$date'
AND iwm_good_receipt_hreceipt_name NOT LIKE
'Return%'
group by pur_po_header.supplier_name ";

        $command5 = $db->createCommand($query5);
        $results5 = $command5->queryAll();

        if ($results['avg_late'] != NULL &&
$results['avg_defect'] != NULL && $results['avg_price'] !=
NULL)
        {
                $query1 = "SELECT supplier.supplier_name as
'0',(avg(time_to_sec(timediff(gr_document_date,
need_by_date )) / 3600))/ $results[avg_late] as '1',
        (avg(iwm_good_receipt_f.defect_percent))/ $results[avg_defect
] as '2',
        (avg(pur_po_line.unit_price))/ $results[avg_price]
as '3'
FROM pur_po_header
left join iwm_good_receipt_h on pur_po_header.id =
iwm_good_receipt_h.from_po_id
left join iwm_good_receipt_f on

```

```

iwm_good_receipt_h.id = iwm_good_receipt_f.gr_id
    left join pur_po_line on pur_po_header.id =
pur_po_line.po_header_id
    left join supplier on pur_po_header.supplier_name
= supplier.id
    where iwm_good_receipt_f.item_id = '$item' AND
pur_po_line.item_number = '$item'
    AND pur_po_header.agreement_start_date >= '$date'
    AND iwm_good_receipt_hreceipt_name NOT LIKE
'Return%'
    group by pur_po_header.supplier_name ";

$command1 = $db->createCommand($query1);
$results1 = $command1->queryAll();
$data['dataProvider1'] = $results1;

$kepentingan = array();
$kepentingan[0] = $late;
$kepentingan[1] = $defect;
$kepentingan[2] = $price;

$data['dataProvider5'] =
SupplierMasterController::actionTopsis($results1,
$kepentingan, $results5);
}

}
$query2 = "SELECT iwm_item_master.item_name from
iwm_item_master where iwm_item_master.id = $item";

$command2 = $db->createCommand($query2);
$results2 = $command2->queryAll();
$data['dataProvider2'] = $results2;

$query3 = "SELECT iwm_item_master.id,
iwm_item_master.item_name from iwm_item_master";

$command3 = $db->createCommand($query3);
$results3 = $command3->queryAll();
$data['dataProvider3'] = $results3;
return $this->render('index', $data);
}

//===== TOPSIS START =====
public function actionTopsis($results1, $kepentingan,
$results5)
{
    $alternatifkriteria = array();
    $pembagi = array();
    for ($i=0;$i<count($results1);$i++)
    {
        $k = 0;
        for ($j=1;$j<4;$j++)

```



```

    {
        $alternatifkriteria[$i][$k] =
$results1[$i][$j];
        $k++;
    }
}
for ($i=0; $i<3; $i++)
{
    $pembagi[$i] = 0;
    for ($j=0; $j<count($alternatifkriteria);
$j++)
    {
        // $j = $j-1;
        $pembagi[$i] = $pembagi[$i] +
($alternatifkriteria[$j][$i] * $alternatifkriteria[$j][$i]);
    }
    $pembagi[$i] = sqrt($pembagi[$i]);
}
$normalisasi = array();
for ($i=0;$i<count($alternatifkriteria);$i++)
{
    for ($j=0;$j<3;$j++)
    {
        if ($pembagi[$j] == 0)
        {
            $pembagi[$j] = 1;
        }

        $normalisasi[$i][$j] =
$alternatifkriteria[$i][$j] / $pembagi[$j];
    }
}
for ($i=0;$i<count($alternatifkriteria);$i++)
{
    for ($j=0;$j<3;$j++)
    {

        $sterbobot[$i][$j] = $normalisasi[$i][$j] *
$kepentingan[$j];
    }
}
$apulus = array();
for ($i=0;$i<3;$i++)
{
    for ($j=0;$j<count($alternatifkriteria);$j++)
    {
        if ($j == 0)
        {
            $apulus[$i] = $sterbobot[$j][$i];
        }
        else
    }
}

```

```

        {
            if ($aplus[$i] > $sterbobot[$j][$i])
            {
                $aplus[$i] = $sterbobot[$j][$i];
            }
        }
    }
}
$amin = array();
for ($i=0;$i<3;$i++)
{
    for ($j=0;$j<count($alternatifkriteria);$j++)
    {
        if ($j == 0)
        {
            $amin[$i] = $sterbobot[$j][$i];
        }
        else
        {
            if ($amin[$i] < $sterbobot[$j][$i])
            {
                $amin[$i] = $sterbobot[$j][$i];
            }
        }
    }
}
$dplus = array();
for ($i=0;$i<count($alternatifkriteria);$i++)
{
    $dplus[$i] = 0;
    for ($j=0;$j<3;$j++)
    {
        $dplus[$i] = $dplus[$i] + (($aplus[$j] -
$sterbobot[$i][$j]) * ($aplus[$j] - $sterbobot[$i][$j]));
    }
    $dplus[$i] = sqrt($dplus[$i]);
}
$dmin = array();
for ($i=0;$i<count($alternatifkriteria);$i++)
{
    $dmin[$i] = 0;
    for ($j=0;$j<3;$j++)
    {
        $dmin[$i] = $dmin[$i] + (($sterbobot[$i][$j]
- $amin[$j]) * ($sterbobot[$i][$j] - $amin[$j]));
    }
    $dmin[$i] = sqrt($dmin[$i]);
}
$hasil = array();
for ($i=0;$i<count($alternatifkriteria);$i++)
{
    if ($dmin[$i] == 0 && $dplus[$i] == 0)

```

```

        {
            $hasil[$i] = 0;
        }
        else
        {
            $hasil[$i] = $dmin[$i] / ($dmin[$i] +
$dplus[$i]);
        }
    }
    //combine result and hasil
    $sortingarray = array();
    $b = count($results5);

    for ($i=0;$i<count($results5);$i++)
    {

        for ($j=0;$j<4;$j++)
        {
            $k = 4;
            $sortingarray[$i][$j] = $results5[$i][$j];

        }
        $sortingarray[$i][$k] = $hasil[$i];
    }
    //2D Bubble Sort
    for ($i=0;$i<count($results5)-1;$i++)
    {
        for ($j=$i;$j<count($results5);$j++)
        {
            if ($sortingarray[$j][4] >
$sortingarray[$i][4])
            {
                //swap the two between each other
                $temp = $sortingarray[$i];
                $sortingarray[$i] = $sortingarray[$j];
                $sortingarray[$j]=$temp;

            }
        }
    }
    $data['dataProvider4'] = $hasil;
    $data['dataProvider5'] = $sortingarray;
    return $sortingarray;
}

```

Kode Sumber 4.50 Menganalisa supplier

4.5.30. Melihat Sejarah Permintaan Barang

Untuk melihat sejarah permintaan barang, user masuk ke halaman Item Demand History, Pengaturan untuk melihat sejarah permintaan barang ini terdapat pada kelas *ItemDemandController* fungsi *actionIndex()*. Kode untuk melihat sejarah permintaan barang ditunjukkan pada kode sumber 4.51.

```
public function actionIndex($item = NULL, $year = NULL )
{
    $db = Yii::$app->db;
    $item = Yii::$app->request->post('ItemParam');
    $years = Yii::$app->request->post('YearParam');
    if($item == "") {
        $item = 1;
    }
    if($years == "Select" || $years == "") {
        $years = date("Y");
    }
    $sqlyear = "SELECT distinct
year(pur_po_header.agreement_start_date) as year from
pur_po_header";
    $sqlmonth = "SELECT distinct
month(pur_po_header.agreement_start_date) as month from
pur_po_header WHERE year(pur_po_header.agreement_start_date)
= $years";
    $sqlprice = "SELECT avg(pur_po_line.unit_price) as
price, pur_po_header.agreement_start_date as dates from
pur_po_header LEFT JOIN pur_po_line
on
pur_po_line.po_header_id = pur_po_header.id WHERE
pur_po_line.item_number = $item group
by
pur_po_header.agreement_start_date";
    $date = $db->createCommand($sqlprice)->queryAll();
    $year = $db->createCommand($sqlyear)->queryAll();
    $month = $db->createCommand($sqlmonth)->queryAll();
    $count = 0;
    $qtyChart = array();
    foreach ($month as $key => $value) {
        $sqlqty = "SELECT sum(pur_po_line.quantity) as
qty from pur_po_header LEFT JOIN pur_po_line
on
pur_po_line.po_header_id = pur_po_header.id WHERE
pur_po_line.item_number = $item AND
month(agreement_start_date) = $value[month] AND
year(agreement_start_date) = $years";
        $qty = $db->createCommand($sqlqty)->queryOne();
        $qtyChart[$count]['qty'] = $qty['qty'];
        if($value['month'] == 1)
        {
            $qtyChart[$count]['date'] = 'January';
        }
    }
}
```

```

}
else if($value['month'] == 2)
{
    $qtyChart[$count]['date'] = 'February';
}
else if($value['month'] == 3)
{
    $qtyChart[$count]['date'] = 'March';
}
else if($value['month'] == 4)
{
    $qtyChart[$count]['date'] = 'April';
}
else if($value['month'] == 5)
{
    $qtyChart[$count]['date'] = 'May';
}
else if($value['month'] == 6)
{
    $qtyChart[$count]['date'] = 'June';
}
else if($value['month'] == 7)
{
    $qtyChart[$count]['date'] = 'July';
}
else if($value['month'] == 8)
{
    $qtyChart[$count]['date'] = 'August';
}
else if($value['month'] == 9)
{
    $qtyChart[$count]['date'] = 'September';
}
else if($value['month'] == 10)
{
    $qtyChart[$count]['date'] = 'October';
}
else if($value['month'] == 11)
{
    $qtyChart[$count]['date'] = 'November';
}
else if($value['month'] == 12)
{
    $qtyChart[$count]['date'] = 'December';
}
$count++;
}
$itemsql = "SELECT iwm_item_master.id,
iwm_item_master.item_name from iwm_item_master";
$selectd = "SELECT iwm_item_master.item_name from
iwm_item_master WHERE id = $item";

```

```

$command3 = $db->createCommand($itemsql);
$command4 = $db->createCommand($selected);
$items = $command3->queryAll();
$itemname = $command4->queryOne();

return $this->render('index', [

    'itemname' => $itemname,
    'items' => $items,
    'year' => $year,
    'qtyChart' => $qtyChart,
    'years' => $years,

]);
}

```

Kode Sumber 4.51 Melihat Sejarah Permintaan Barang

4.5.31. Melihat Sejarah Harga Barang

Untuk melihat sejarah harga barang, user masuk ke halaman Item Price History, Pengaturan untuk melihat sejarah harga barang ini terdapat pada kelas *PriceHistoryController* fungsi *actionIndex()*. Kode untuk melihat sejarah harga barang ditunjukkan pada kode sumber 4.52.

```

public function actionIndex($item = NULL, $year = NULL )
{
    $db = Yii::$app->db;
    $item = Yii::$app->request->post('ItemParam');
    $years = Yii::$app->request->post('YearParam');
    if($item == "") {
        $item = 1;
    }
    if($years == "Select" || $years == "") {
        $years = date("Y");
    }
    $sqlyear = "SELECT distinct
year(pur_po_header.agreement_start_date) as year from
pur_po_header";
    $sqlmonth = "SELECT distinct
month(pur_po_header.agreement_start_date) as month from
pur_po_header WHERE year(pur_po_header.agreement_start_date)
= $years";
    $sqlprice = "SELECT avg(pur_po_line.unit_price) as
price, pur_po_header.agreement_start_date as dates from
pur_po_header LEFT JOIN pur_po_line
on
pur_po_line.po_header_id = pur_po_header.id WHERE
pur_po_line.item_number = $item group by
pur_po_header.agreement_start_date";

```

```

$date = $db->createCommand($sqlprice)->queryAll();
$year = $db->createCommand($sqlyear)->queryAll();
$month = $db->createCommand($sqlmonth)->queryAll();
$count = 0;
$qtyChart = array();
    foreach ($month as $key => $value) {
        $sqlqty = "SELECT sum(pur_po_line.quantity) as
qty    from    pur_po_header    LEFT    JOIN    pur_po_line    on
pur_po_line.po_header_id    =    pur_po_header.id    WHERE
pur_po_line.item_number    =    $item    AND
month(agreement_start_date)    =    $value[month]    AND
year(agreement_start_date) = $years";
        $qty = $db->createCommand($sqlqty)->queryOne();
        $qtyChart[$count]['qty'] = $qty['qty'];
        if($value['month'] == 1)
        {
            $qtyChart[$count]['date'] = 'January';
        }
        else if($value['month'] == 2)
        {
            $qtyChart[$count]['date'] = 'February';
        }
        else if($value['month'] == 3)
        {
            $qtyChart[$count]['date'] = 'March';
        }
        else if($value['month'] == 4)
        {
            $qtyChart[$count]['date'] = 'April';
        }
        else if($value['month'] == 5)
        {
            $qtyChart[$count]['date'] = 'May';
        }
        else if($value['month'] == 6)
        {
            $qtyChart[$count]['date'] = 'June';
        }
        else if($value['month'] == 7)
        {
            $qtyChart[$count]['date'] = 'July';
        }
        else if($value['month'] == 8)
        {
            $qtyChart[$count]['date'] = 'August';
        }
        else if($value['month'] == 9)
        {
            $qtyChart[$count]['date'] = 'September';
        }
        else if($value['month'] ==10)

```

```

        {
            $qtyChart[$count]['date'] = 'October';
        }
        else if($value['month'] == 11)
        {
            $qtyChart[$count]['date'] = 'November';
        }
        else if($value['month'] == 12)
        {
            $qtyChart[$count]['date'] = 'December';
        }
        $count++;
    }
    $itemsql = "SELECT iwm_item_master.id,
iwm_item_master.item_name from iwm_item_master";
    $selected = "SELECT iwm_item_master.item_name from
iwm_item_master WHERE id = $item";
    $command3 = $db->createCommand($itemsql);
    $command4 = $db->createCommand($selected);
    $items = $command3->queryAll();
    $itemname = $command4->queryOne();

    return $this->render('index', [

        'itemname' => $itemname,
        'items' => $items,
        'year' => $year,
        'qtyChart' => $qtyChart,
        'years' => $years,
    ]);
}

```

Kode Sumber 4.52 Melihat Sejarah Harga Barang

4.5.32. Melihat Ramalan Kebutuhan Barang

Untuk melihat ramalan kebutuhan barang, user masuk ke halaman Item Forecast pada halaman ini terdapat juga informasi mengenai total barang yang sudah dipesan, dan total barang yang sudah ada pada *Inventory*. Pengaturan untuk melihat ramalan kebutuhan barang ini terdapat pada kelas *ItemForecastController* fungsi *actionIndex()*. Kode untuk melihat ramalan kebutuhan barang ditunjukkan pada kode sumber 4.53.

```

public function actionIndex($months = NULL, $years = NULL)
{
    $db = Yii::$app->db;
    $months = Yii::$app->request->post('MonthParam');
}

```



```

$years = Yii::$app->request->post('YearParam');
if($years == "Select" || $years == "") {
    $years = date("Y");
}
if($months == "Select" || $months == "") {
    $months = date("n");
}
$date['months'] = $months;
$date['years'] = $years;
$sqlyear = "SELECT distinct
year(sd_forecastline.date_forecastline) as year from
sd_forecastline";
$year = $db->createCommand($sqlyear)->queryAll();
$forcastsql = "SELECT sd_forecast.iwm_item_master_id
as id_item, sum(sd_forecastline.result_forecastline) as qty
from sd_forecast LEFT JOIN sd_forecastline on
sd_forecastline.sd_forecast_id = sd_forecast.id WHERE
year(sd_forecastline.date_forecastline) = $years AND
month(sd_forecastline.date_forecastline) = $months group by
sd_forecast.iwm_item_master_id";
$forcast = $db->createCommand($forcastsql)-
>queryAll();
$i = 0;
$itemqty = array();
foreach ($forcast as $key => $value) {
    $var=
IwmgoodissueController::actionExplodingBom($value['id_item'
]);
    $j = 0;
    foreach ($var as $key => $value1)
    {
        $itemqty[$i][$j]['material_id'] =
$value1['material_id'];
        $itemqty[$i][$j]['quantity'] = $value1['quantity']
* $value['qty'] ;
        $j++;
    }
    $i++;
}
$itemRawsq = "SELECT * FROM iwm_item_master WHERE
item_type = '1'";
$itemRaw = $db->createCommand($itemRawsq)-
>queryAll();
foreach ($itemRaw as $key => $value) {
    $inv = IwmInventory::find()
->select('I.*')
->from('iwm_inventory as I, iwm_inventory_type as
T, iwm_item_type as IT')
->where('T.item_type = IT.id
and T.inventory_id = I.id
and IT.id = '.$value['item_type']->one();

```

```

        $quantity =
IwmquantController::actionGetFreeQtySpesificItem($value['id'
], $inv['id']);
        $itemRaw[$key]['quantity'] = 0;
        $itemRaw[$key]['qtystock'] = $quantity;
    }
    for ($k=0;$k<$i;$k++)
    {
        for ($l=0;$l<$j;$l++)
        {
            foreach ($itemRaw as $key => $value)
            {
                if($value['id'] ==
$itemqty[$k][$l]['material_id'])
                {
                    $itemRaw[$key]['quantity'] +=
$itemqty[$k][$l]['quantity'];
                }
            }
        }
    }
    foreach ($itemRaw as $key => $value) {
        $sqlqty = "SELECT sum(pur_po_line.quantity) as qty
from pur_po_header LEFT JOIN pur_po_line on
pur_po_line.po_header_id = pur_po_header.id WHERE
pur_po_line.item_number = '$value[id]' AND
month(agreement_start_date) = $months AND
year(agreement_start_date) = $years";
        $qty = $db->createCommand($sqlqty)->queryOne();
        if($qty['qty'] == "")
        {
            $qty = 0;
        }
        $itemRaw[$key]['actualqty'] = $qty['qty'];
    }
    return $this->render('index', [
        'itemRaw' => $itemRaw,
        'year' => $year,
        'date' => $date,
    ]);
}

```

Kode Sumber 4.53 Melihat Ramalan Kebutuhan Barang

4.5.33. Melihat Laporan Pembelian

Untuk melihat laporan pembelian, user masuk ke halaman *PO Report*, pada halaman ini terdapat rekap PO selama periode tertentu. Pengaturan untuk melihat laporan pembelian ini terdapat

pada kelas *PoAnalysisController* fungsi *actionIndex()*. Kode untuk melihat laporan pembelian barang ditunjukkan pada kode sumber 4.54.

```

public function actionIndex($supplier = NULL, $date1 = NULL,
    $date2 = NULL)
    {
        $supplier = Yii::$app->request->post('supplier');
        $date1 = Yii::$app->request->post('date1');
        $date2 = Yii::$app->request->post('date2');
        if($date1 == "") {
            $date1 = "2016-01-01";
        }
        if($date2 == "") {
            $date2 = date("Y-m-d");
        }
        $title['date1'] = $date1;
        $title['date2'] = $date2;
        if ($supplier == "")
        {
            $title['supplier'] = "";
            $query = "SELECT      pur_po_header.po_number,
iwm_good_receipt_h.id,      pur_po_header.id      as      po_id,
pur_po_header.total_amount,      supplier.supplier_name,
time_to_sec(timediff(gr_document_date,      need_by_date      )) /
3600 as hour_difference, avg(defect_percent) as defect,
ap_transaction,      codeCurrency,      agreement_start_date FROM
iwm_good_receipt_h      left      join      pur_po_header      on
pur_po_header.id = iwm_good_receipt_h.from_po_id LEFT JOIN
iwm_good_receipt_f      on      iwm_good_receipt_f.gr_id =
iwm_good_receipt_h.id LEFT JOIN supplier on supplier.id =
pur_po_header.supplier_name      LEFT      JOIN      currency      on
currency.id =      pur_po_header.currency      WHERE
pur_po_header.agreement_start_date >= '$date1'      AND
pur_po_header.agreement_start_date <= '$date2' GROUP BY
pur_po_header.id";
        }
        else
        {
            $title['supplier'] = $supplier;
            $query = "SELECT      pur_po_header.po_number,
iwm_good_receipt_h.id,      pur_po_header.id      as      po_id,
pur_po_header.total_amount,      supplier.supplier_name,
time_to_sec(timediff(gr_document_date,      need_by_date      )) /
3600 as hour_difference, avg(defect_percent) as defect,
ap_transaction,      codeCurrency,      agreement_start_date FROM
iwm_good_receipt_h      left      join      pur_po_header      on
pur_po_header.id = iwm_good_receipt_h.from_po_id LEFT JOIN
iwm good receipt f      on      iwm good receipt f.gr id =

```

```

iwm_good_receipt_h.id LEFT JOIN supplier on supplier.id =
pur_po_header.supplier_name LEFT JOIN currency on
currency.id = pur_po_header.currency WHERE
pur_po_header.supplier_name = $supplier AND
pur_po_header.agreement_start_date >= '$date1' AND
pur_po_header.agreement_start_date <= '$date2' GROUP BY
pur_po_header.id ";

$query6 = "SELECT supplier.supplier_name from
supplier where supplier.id = $supplier";
$db = Yii::$app->db;
$command6 = $db->createCommand($query6);
$results6 = $command6->queryAll();
$data['dataProvider6'] = $results6;
}
$query5 = 'SELECT id,supplier_name FROM supplier';
$db = Yii::$app->db;
$command = $db->createCommand($query);
$command5 = $db->createCommand($query5);
$results = $command->queryAll();
$results5 = $command5->queryAll();
$data['dataProvider'] = $results;
$data['dataProvider5'] = $results5;
$data['title'] = $title;
return $this->render('index', $data);
}

```

Kode Sumber 4.54 Melihat Laporan Pembelian

4.6 Implementasi Antarmuka Pengguna

4.6.1. Antarmuka Melihat Daftar *Purchase Requisition*

Pada antarmuka ini pengguna dapat melihat daftar *Purchase Requisition* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan membuat, menyunting, melihat rincian, dan menghapus *Purchase Requisition*. Terdapat *Order Status* yang berisi *incomplete* berarti permintaan tersebut belum sepenuhnya dipesan dan *complete* berarti permintaan tersebut sudah sepenuhnya dibuat. Tampilan antarmuka ini dapat dilihat pada Gambar 4.10.

Purchase Requisition

Create Purchase Requisition Advance Search

Purchase Requisition Showing 1-3 of 3 items

#	Req Number	Status	Order Status	Creator	Shipto	Need By Date	
1	REQ-Forecast-10	Entered	Forecast	Forecast	Raw	2016-07-04	
2	REQ0000014	Finished	Complete	niklo	Raw	2016-05-20	
3	REQ-Forecast-01	Entered	Complete	Forecast	Raw	2016-05-25	

Gambar 4.10 Tampilan Antarmuka Melihat Daftar Purchase Requisition

4.6.2. Antarmuka Menambah *Purchase Requisition*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Purchase Requisition* dan menekan tombol “*Create*” untuk perintah pembuatan, form isian “*Status*” bisa diisi dengan *entered* apabila data tersebut baru dimasukkan, *finished* apabila data tersebut sudah selesai dimasukkan, dan *published* apabila data tersebut sudah dipesan, form isian *creator* akan terisi *forecast* apabila requisition dibuat dari ramalan pembelian dan *production* apabila requisition dibuat oleh pihak produksi. Tampilan antarmuka ini dapat dilihat pada Gambar 4.11.

Create Purchase Requisition

Status: Entered

Creator: Creator

Shipto: Choose Inventory

Need By Date: Choose Need By Date

Pur Requisition Line Showing 1-1 of 1 item

#	Item Name	Type	Line Description	Quantity	UoM Id
1	Choose Item Name	Choose Item Type			Choose UoM

+Add Row

Create

Gambar 4.11 Tampilan Antarmuka Menambah Purchase Requisition

4.6.3. Antarmuka Menyunting *Purchase Requisition*

Pada antarmuka ini pengguna dapat merubah data *Purchase Requisition* dan menekan tombol “*Update*” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.12.

Purchase Requisition: 14

#	Item Name	Type	Line Description	Quantity	UoM id
1	Tire	Rare	Tire's Bika	120	Pcs
2	Seal Kit	Rare	Seal Kit's Bika	120	Pcs

Gambar 4.12 Tampilan Antarmuka Menyunting Purchase Requisition

4.6.4. Antarmuka Melihat Rincian *Purchase Requisition*

Pada antarmuka ini pengguna dapat melihat secara rinci data *Purchase Requisition*. Terdapat tombol *convert to RFQ* untuk merubah *requisition* menjadi RFQ dan tombol *convert to PO* untuk merubah *requisition* menjadi PO, row pada tabel requisition line berwarna hijau apabila item yang diminta telah sepenuhnya dipesan, dan akan berwarna merah apabila belum sepenuhnya dipesan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.13.

Pur Requisition Header 14

PDF Update Delete Convert All to PO Convert All to TRQ

ID	14
Req Number	REQ000014
Status	Finished
Order Status	Cancel
Creator	ricko
Item Inventory	Raw
Need By Date	2016-05-25

Pur Requisition Line 14

Showing 1 of 2 items

#	Type	Item Number	Line Description	Quantity	Ordered Qty	UOM
1	Raw	Tire	Tire's Bkx	120	120	Pcs
2	Raw	Seat K01	Seat K01's Bkx	120	120	Pcs

Convert Selected to TRQ

Convert Selected to PO

Gambar 4.13 Tampilan Antarmuka Melihat Rincian Purchase Requisition

4.6.5. Antarmuka Menghapus *Purchase Requisition*

Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “Yes” atau pembatalan dengan menekan tombol “Cancel”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.14.

10.151.64.124 says:

Are you sure you want to delete this item?

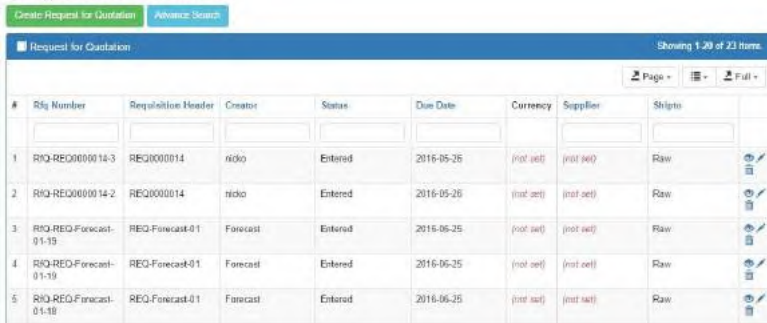
OK Cancel











Gambar 4.14 Tampilan Antarmuka Menghapus Purchase Requisition

4.6.6. Antarmuka Melihat Daftar *Request for Quotation*

Pada antarmuka ini pengguna dapat melihat daftar *Request for Quotation* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan membuat, menyunting, melihat rincian, dan menghapus *Request for Quotation*. Tampilan antarmuka ini dapat dilihat pada Gambar 4.15.

Request for Quotation



#	Rfq Number	Requisition Header	Creator	Status	Due Date	Currency	Supplier	Shipto	
1	RIQ-REQ000014-3	REQ000014	nido	Entered	2016-05-25	(not set)	(not set)	Raw	 
2	RIQ-REQ000014-2	REQ000014	nido	Entered	2016-05-25	(not set)	(not set)	Raw	 
3	RIQ-REQ-Forecast-01-13	REQ-Forecast-01	Forecast	Entered	2016-06-25	(not set)	(not set)	Raw	 
4	RIQ-REQ-Forecast-01-19	REQ-Forecast-01	Forecast	Entered	2016-06-25	(not set)	(not set)	Raw	 
5	RIQ-REQ-Forecast-01-18	REQ-Forecast-01	Forecast	Entered	2016-06-25	(not set)	(not set)	Raw	 

Gambar 4.15 Tampilan Antarmuka Melihat Daftar Request for Quotation

4.6.7. Antarmuka Menambah *Request for Quotation*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Request for Quotation* dan menekan tombol “Create” untuk perintah pembuatan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.16.

Create Request for Quotation

Rfq Number

Req Header ID

Creator

Status

Due Date

Currency

Supplier Name

Ship to

#	Item/Itemmaster	Line Description	Min Quantity	UoM Id	Target Price
1	<input type="text" value="Choose Item/Itemmaster"/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value="Choose UoM"/>	<input type="text" value=""/>

Gambar 4.16 Tampilan Antarmuka Menambah Request for Quotation

4.6.8. Antarmuka Menyunting *Request for Quotation*

Pada antarmuka ini pengguna dapat merubah data *Request for Quotation* dan menekan tombol “*Update*” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.17.

Update Request for Quotation: 26

Rfq Number

Req Header ID

Creator

Status

Due Date

Currency

Supplier Name

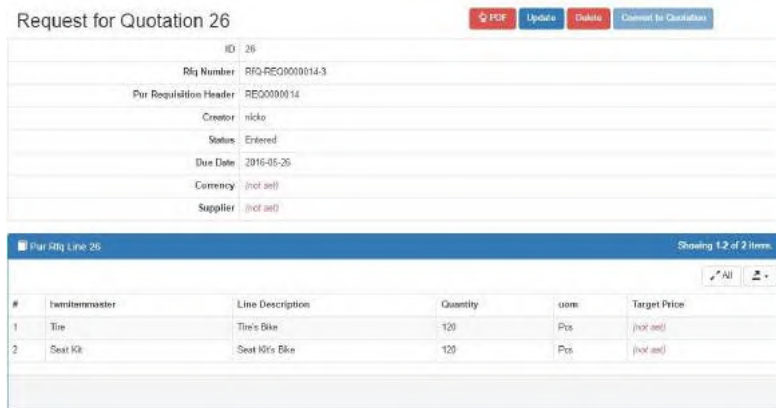
Ship to

#	Item/Itemmaster	Line Description	Min Quantity	UoM Id	Target Price
1	<input type="text" value="Tas"/>	<input type="text" value="Tas's Bille"/>	<input type="text" value="100"/>	<input type="text" value="Pcs"/>	<input type="text" value=""/>
2	<input type="text" value="Seat Pk"/>	<input type="text" value="Seat Pk's Bille"/>	<input type="text" value="100"/>	<input type="text" value="Pcs"/>	<input type="text" value=""/>

Gambar 4.17 Tampilan Antarmuka Menyunting *Request for Quotation*

4.6.9. Antarmuka Melihat Rincian *Request for Quotation*

Pada antarmuka ini pengguna dapat melihat secara rinci data *Request for Quotation*. Terdapat tombol *convert to Quotation* untuk merubah *RFQ* menjadi *Quotation* dan tombol PDF untuk menjadikan *RFQ* ke PDF dan siap untuk dikirimkan ke *supplier*, *RFQ* baru bisa di-*convert* ke *quotation* apabila datanya sudah terisi semua. Tampilan antarmuka ini dapat dilihat pada Gambar 4.18.



Request for Quotation 26

ID: 26

Rfq Number: RFQ-REQ000014-3

Pmr Regulation Header: REQ000014

Creator: niko

Status: Entered

Due Date: 2016-05-26

Currency: (not set)

Supplier: (not set)

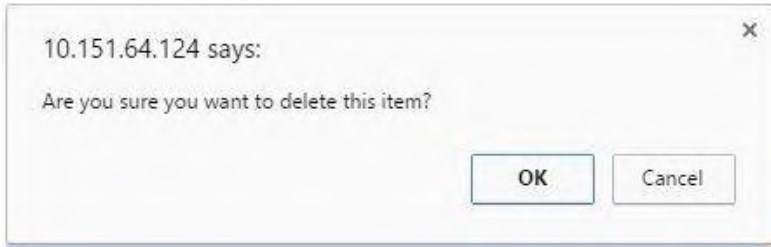
Showing 1-2 of 2 items.

#	Itemmaster	Line Description	Quantity	uom	Target Price
1	Tire	Tire's Bike	120	Pcs	(not set)
2	Seat Kit	Seat Kit's Bike	120	Pcs	(not set)

Gambar 4.18 Tampilan Antarmuka Melihat Rincian Request for Quotation

4.6.10. Antarmuka Menghapus *Request for Quotation*

Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “*Yes*” atau pembatalan dengan menekan tombol “*Cancel*”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.19.



Gambar 4.19 Tampilan Antarmuka Menghapus Request for Requisition

4.6.11. Antarmuka Melihat Daftar *Maintain Quotation*

Pada antarmuka ini pengguna dapat melihat daftar *Maintain Quotation* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan membuat, menyunting, melihat rincian, dan menghapus *Maintain Quotation*. Apabila row tersebut berwarna hijau berarti *quotation* sudah disetujui dan *quotation* itu masih berlaku, dan apabila berwarna merah berarti *quotation* itu belum atau tidak disetujui dan sudah tidak berlaku lagi. Tampilan antarmuka ini dapat dilihat pada Gambar 4.20.

Manage Quotation

Showing 1-3 of 3 items.

#	Quo Number	RFQ Header	Header Status	Quotation Status	Valid Date	Price Date	Total Amount	Currency	Supplier	Duration Of Delivery(days)
1	QUO-RFQ-REQ0000014-3-1	RFQ-REQ0000014-3	Finished	Approved	2016-05-30	2016-05-30	\$1220000	IDR	Baquni Enterprise	10
2	QUO-RFQ-REQ0000014-1-2	RFQ-REQ0000014-1	Entered	(not set)	(not set)	(not set)	(not set)	IDR	supplier1	(not set)
3	QUO-RFQ-REQ-Forecast-01-1-1	RFQ-REQ-Forecast-01-1	Finished	Approved	2016-05-31	2016-05-31	52632912500	IDR	Baquni Enterprise	5

Gambar 4.20 Tampilan Antarmuka Melihat Daftar *Maintain Quotation*

4.6.12. Antarmuka Menambah *Maintain Quotation*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Maintain Quotation* dan menekan tombol “*Create*” untuk perintah pembuatan, terdapat *tab Tax* yang berisi rincian pajak, dan *tab shipment* yang berisi rincian biaya pengiriman, form isian “*Total Amount*” adalah biaya total dari quotation tersebut seperti harga barang, pajak, dan biaya pengiriman. “*Total Amount*”, dan “*Line Price*” akan terisi secara otomatis. Tampilan antarmuka ini dapat dilihat pada Gambar 4.21.

Create Quotation

The screenshot shows a web-based form titled "Create Quotation". At the top, there is a "Quo Number" field. Below it are "Rfq Header ID" and "Header Status" (with a "Status" dropdown). The form is divided into several sections: "Mktl Date" and "Price Date" (each with a "Choose Valid Date" button), "Total Amount" (with a "Take Amount" field and "Choose Currency" dropdown), and "Supplier" (with a "Choose Supplier" dropdown) and "Duration Of Delivery(days)" (with a "Duration Of Delivery" field). Below these fields are two green buttons: "Item Detail" and "Shipment". The main part of the form is a table titled "Item Quotation Line" with a "Showing 1.1 of 1 Item" indicator. The table has columns for "#", "Item Master", "Item Desc", "Min Quantity", "UoM Id", "Unit Price", and "Line Price". The first row is partially filled with dropdown menus for "Item Master" and "UoM Id", and input fields for "Min Quantity" and "Unit Price". Below the table is a green "+Add Row" button and a "Create" button at the bottom left.

Gambar 4.21 Tampilan Antarmuka Menambah *Maintain Quotation*

4.6.13. Antarmuka Menyunting *Maintain Quotation*

Pada antarmuka ini pengguna dapat merubah data *Maintain Quotation* dan menekan tombol “*Update*” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.22.

Update Quotation: 8

Quo Number: QUGR00000145-1

Qty Header ID: RUGR00000145-1

Header Status: Final

Valid Date: 2016-06-30 Price Date: 2016-06-30

Total Amount: 11250000 Currency: IDR

Supplier: Elexis Trijacton Duration Of Delivery (Days): 10

Qty Quotation Line

#	Item Description	Item Desc	Min Quantity	UoM	Unit Price	Line Price
1	Tipe	Tipe 2016	100	Pcs	10000	1000000
2	Item 2016	Item 2016	100	Pcs	10000	1000000

Update

Gambar 4.22 Tampilan Antarmuka Menyunting *Maintain Quotation*

4.6.14. Antarmuka Melihat Rincian *Maintain Quotation*

Pada antarmuka ini pengguna dapat melihat secara rinci data *Maintain Quotation*. Terdapat tombol *convert to PO* untuk merubah *Quotation* menjadi PO dan ada tombol *accept* untuk menyetujui *Quotation*, tombol *reject* untuk menolak *Quotation*. Tampilan antarmuka ini dapat dilihat pada Gambar 4.23.

Quotation 8

Accept This Quotation | Reject This Quotation | Update | Final | Convert to PO

Qty: 8

Quo Number: QUGR00000145-1

Qty Header ID: RUGR00000145-1

Header Status: Final

Quotation Status: Accepted

Valid Date: 2016-06-30

Price Date: 2016-06-30

Total Amount: 11250000

Currency: IDR

Supplier: Elexis Trijacton

Duration Of Delivery (Days): 10

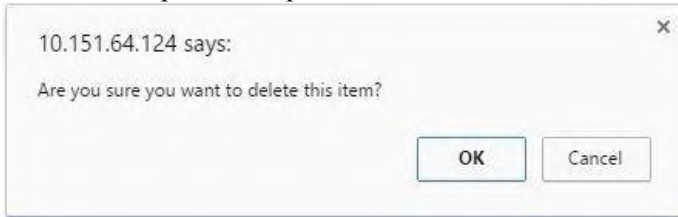
Qty Quotation Line 8

#	Item Description	Item Desc	Quantity	UoM	Unit Price
1	Tipe	Tipe 2016	125	Pcs	90000

Gambar 4.23 Tampilan Antarmuka Melihat Rincian *Maintain Quotation*

4.6.15. Antarmuka Menghapus *Maintain Quotation*

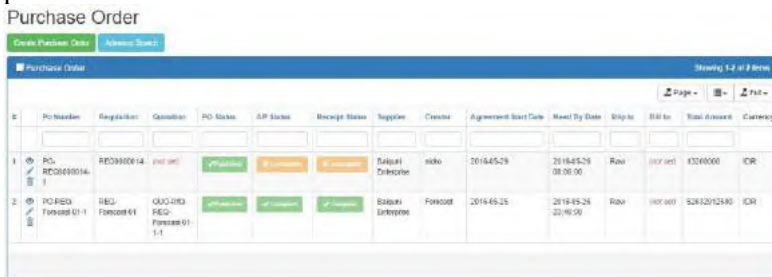
Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “Yes” atau pembatalan dengan menekan tombol “Cancel”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.24.



Gambar 4.24 Tampilan Antarmuka Menghapus *Maintain Quotation*

4.6.16. Antarmuka Melihat Daftar *Purchase Order*

Pada antarmuka ini pengguna dapat melihat daftar *Purchase Order* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan membuat, menyunting, melihat rincian, dan menghapus *Purchase Order*. AP Status merupakan status pembayaran PO tersebut, apabila pembayaran belum lunas isinya “*incomplete*”, dan apabila sudah lunas isinya menjadi “*complete*”. Receipt Status merupakan status penerimaan barang, apabila isinya “*incomplete*” berarti barang di PO tersebut belum terkirim semua, dan apabila isinya “*complete*” berarti barang tersebut sudah terisi semua. Tampilan antarmuka ini dapat dilihat pada Gambar 4.25.



Gambar 4.25 Tampilan Antarmuka Melihat Daftar *Purchase Order*

4.6.17. Antarmuka Menambah *Purchase Order*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Purchase Order* dan menekan tombol “*Create*” untuk perintah pembuatan, pada form isian “*Line Price*” berisi penjumlahan pada harga barang yang dikalikan jumlah pemesanannya, pada form isian “*Total Price*” berisi penjumlahan dari seluruh harga item yang sudah dikalikan dengan jumlahnya, form isian “*total amount*” berisi penjumlahan antara *total price*, pajak, dan biaya pengiriman. Tampilan antarmuka ini dapat dilihat pada Gambar 4.26.

Create Purchase Order

Po Number

Po Number

Req Header ID Qty Header ID

Status Supplier Name

Create Agreement Start Date End By Date

Single Bill

Total Price Total Amount Currency

New Data:

Item Reference	Item Description	Quantity	Unit	Unit Price	Line Price
<input type="text" value="Choose Item Reference"/>	<input type="text" value="Choose Description"/>	<input type="text" value="Quantity"/>	<input type="text" value="Choose Unit"/>	<input type="text" value="Unit Price"/>	<input type="text" value="Line Price"/>

Gambar 4.26 Tampilan Antarmuka Menambah *Purchase Order*

4.6.18. Antarmuka Menyunting *Purchase Order*

Pada antarmuka ini pengguna dapat merubah data *Purchase Order* dan menekan tombol “*Update*” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.27.

Update Purchase Order: 2

Item Number	Item Description	Quantity	Unit	Unit Price	Line Price
1	Tan	120	Pcs	5000	600000
2	Darat Ht	120	Pcs	5000	600000

Gambar 4.27 Tampilan Antarmuka Menyunting Purchase Order

4.6.19. Antarmuka Melihat Rincian *Purchase Order*

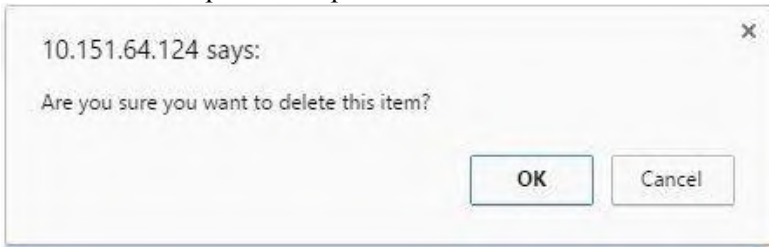
Pada antarmuka ini pengguna dapat melihat secara rinci data *Purchase Order*. Terdapat tombol PDF untuk merubah PO menjadi PO agar siap untuk dikirimkan ke supplier, apabila status berisi *published* maka purchase order tersebut tidak dapat diubah ataupun dihapus. Tampilan antarmuka ini dapat dilihat pada Gambar 4.28.

Item Number	Item Name	Item Description	Quantity	Unit	Unit Price	Line Price
1	Tan	Tan 5000	120	Pcs	5000	600000
2	Darat Ht	Darat Ht 5000	120	Pcs	5000	600000

Gambar 4.28 Tampilan Antarmuka Melihat Rincian *Purchase Order*

4.6.20. Antarmuka Menghapus *Purchase Order*

Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “*Yes*” atau pembatalan dengan menekan tombol “*Cancel*”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.29.



Gambar 4.29 Tampilan Antarmuka Menghapus *Purchase Order*

4.6.21. Antarmuka Melihat Daftar *Return Order*

Pada antarmuka ini pengguna dapat melihat daftar *Return Order* yang terdapat pada sistem. Pengguna juga dapat memilih melakukan kegiatan membuat, menyunting, melihat rincian, dan menghapus *Return Order*. Tampilan antarmuka ini dapat dilihat pada Gambar 4.30.

Return Order

Create Return Order Advance Search

Return Order Showing 1-20 of 29 items

#	Ro Number	PO	RO Status	RO Type	Total Price	Return Tax	Total Amount	Currency	Supplier	Reason F
1	RO-PO-REQ-Forecast120-2	PO-REQ-Forecast120-2	✔ Validated	Return Order	22702600	2270260	24972860	IDR	SANJAYA	barang ru
2	RO-PO-REQ-Forecast120-1	PO-REQ-Forecast120-1	✔ Validated	Return Order	29480200	2948025	32428275	IDR	RESIPAKA	Barang ru
3	RO-PO-REQ0000095-1	PO-REQ0000095-1	✔ Validated	Return Order	1243200	124325	1367575	IDR	SANJAYA	Barang ru

Gambar 4.30 Tampilan Antarmuka Melihat Daftar *Return Order*

4.6.22. Antarmuka Menambah *Return Order*

Pada antarmuka ini pengguna dapat mengisi data yang dibutuhkan terkait *Return Order* dan menekan tombol “*Create*” untuk perintah pembuatan. Form isian “*Return Type*” berisi *Return Payment* yang berarti pengembalian pembayaran, dan *Return Material* yang berarti pengembalian barang. Tampilan antarmuka ini dapat dilihat pada Gambar 4.31.

Create Return Order

The screenshot shows a web-based form for creating a return order. The form is organized into several sections:

- Header Information:** Includes a 'No Member' field and a 'Pi Header ID' dropdown menu.
- Summary Fields:** Contains 'Return Status' and 'Return Type' dropdowns, and 'Total Price', 'Return Tax', 'Total Amount', and 'Currency' fields with associated 'Add' and 'Check' buttons.
- Supplier Information:** A 'Supplier Name' dropdown menu.
- Reason For Return:** A large text area for providing details on why the return is being created.
- Item List Table:** A table with columns for 'Item', 'Description', 'Quantity', 'Unit', 'Unit Price', and 'Line Price'. It currently contains one row with a dropdown for 'Item' and empty fields for the other columns.
- Buttons:** A green 'Create' button is located at the bottom left, and a 'Print Data' button is at the top left.

Gambar 4.31 Tampilan Antarmuka Menambah Return Order

4.6.23. Antarmuka Menyunting *Return Order*

Pada antarmuka ini pengguna dapat merubah data *Return Order* dan menekan tombol “*Update*” untuk perintah penyimpanan perubahan data. Tampilan antarmuka ini dapat dilihat pada Gambar 4.32.

Update Return Order: 30

Ro Number: RO-F0-REG-Fiscal1032

Po Header ID: PO-REG-Fiscal1032

Return Status: Pending

Return Type: Return Material

Total Price: 6000

Return Tax: 0

Total Amount: 6000

Currency: IDR

Supplier Name: SIAURIN

Reason For Return: Barang rusak

Item Detail

#	Item Number	Description	Quantity	Uom	Unit Price	Line Price
1	444-100-000		60	Pcs	100	6000
2	444-100-000		60	Pcs	50	3000

Update

Gambar 4.32 Tampilan Antarmuka Menyunting Return Order

4.6.24. Antarmuka Melihat Rincian *Return Order*

Pada antarmuka ini pengguna dapat melihat secara rinci data *Return Order*. Terdapat tombol PDF untuk merubah RO menjadi PDF agar siap untuk dikirimkan ke supplier. Tampilan antarmuka ini dapat dilihat pada Gambar 4.33.

Return Order 29

ID: 29

Ro Number: RO-F0-REG-Fiscal1031

Po Header: PO-REG-Fiscal1031

Return Status: [PDF](#)

Return Type: [Return Material](#)

Total Price: 2540375

Return Tax: 2540375

Total Amount: 2540375

Currency: IDR

Supplier Name: REGPANG

Reason For Return: Barang rusak

PDF Update Delete

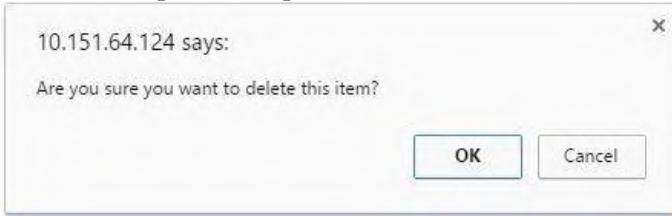
Line

#	Item Number	Description	Quantity	Uom	Unit Price	Line Price
1	Aluminium		100	kg	32500	3250000
2	Red Part 201G		10	Bg Drum	847000	8470000
3	Carbon Fiber		60	kg	152750	9165000

Gambar 4.33 Tampilan Antarmuka Melihat Rincian *Return Order*

4.6.25. Antarmuka Menghapus *Return Order*

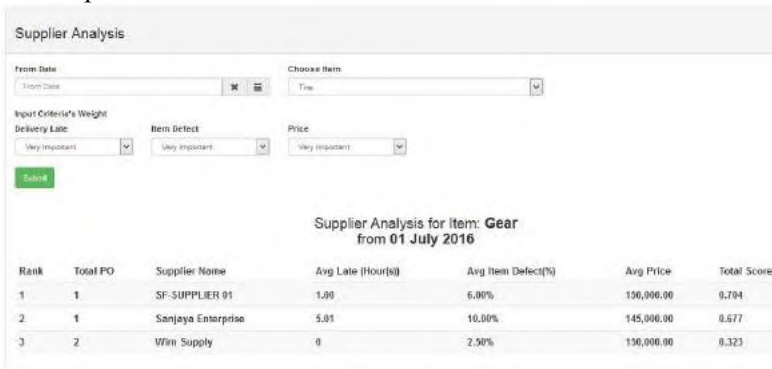
Antarmuka ini berupa halaman *pop-up* untuk melakukan konfirmasi penghapusan data dengan menekan tombol “Yes” atau pembatalan dengan menekan tombol “Cancel”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.34.



Gambar 4.34 Tampilan Antarmuka Menghapus *Return Order*

4.6.26. Antarmuka Menganalisa *Supplier*

Pada antarmuka ini pengguna dapat melihat ranking *supplier* dengan mengisi form yang ada, form tersebut berisi *From Date*, *Choose Item*, Dan bobot dari setiap kriteria yang berbentuk *dropdown list*, terdapat empat bobot yang bisa dipilih, yaitu *ignore* atau diabaikan, *not important* atau tidak penting, *important* atau penting, dan *very important* atau sangat penting. dan menekan tombol “Submit”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.35.



Supplier Analysis

From Date: Choose Item:

Input Criteria's Weight

Delivery Late: Item Defect: Price:

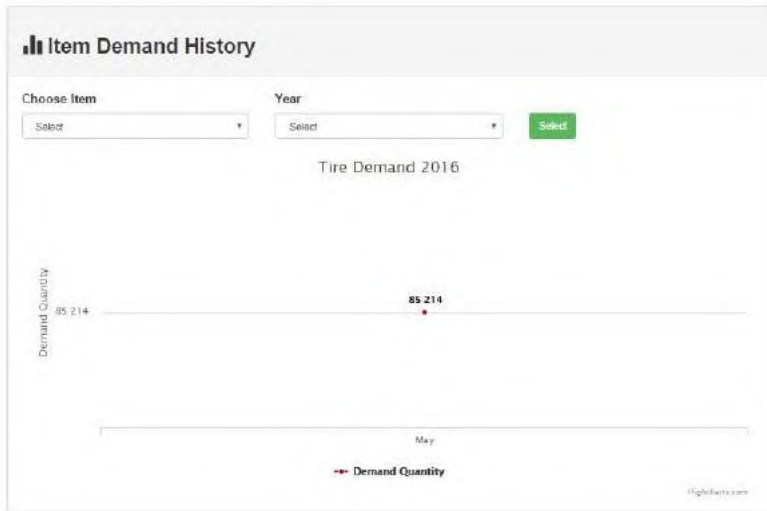
Supplier Analysis for Item: Gear
from 01 July 2016

Rank	Total PO	Supplier Name	Avg Late (Hour(s))	Avg Item Defect(%)	Avg Price	Total Score
1	1	SF-SUPPLIER 01	1.00	5.00%	130,000.00	0.704
2	1	Sanjaya Enterprise	5.01	10.00%	145,000.00	0.577
3	2	Wim Supply	0	2.50%	130,000.00	0.323

Gambar 4.35 Tampilan Antarmuka Hasil Menganalisa *Supplier*

4.6.27. Antarmuka Melihat Sejarah Permintaan Barang

Pada antarmuka ini pengguna dapat melihat sejarah permintaan barang dengan memilih item, memilih tahun dan menekan tombol “Submit”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.36.



Gambar 4.36 Tampilan Antarmuka Melihat Sejarah Permintaan Barang

4.6.28. Antarmuka Melihat Sejarah Harga Barang

Pada antarmuka ini pengguna dapat melihat sejarah harga barang dengan memilih item dan menekan tombol “Submit”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.37.



Gambar 4.37 Tampilan Antarmuka Melihat Sejarah Harga Barang

4.6.29. Antarmuka Melihat Ramalan Kebutuhan Barang

Pada antarmuka ini pengguna dapat melihat ramalan kebutuhan barang, jumlah item yang sudah dipesan, dan item yang sudah ada di gudang dengan memilih bulan, memilih tahun dan menekan tombol “Submit”. Tampilan antarmuka ini dapat dilihat pada Gambar 4.38.

The screenshot shows the 'Item Forecast' interface with a table titled 'Raw Item Purchased and Forecast June, 2016'. The table has five columns: No, Item Name, Purchased Qty, Stock Qty, and Forecast Qty. The data is as follows:

No	Item Name	Purchased Qty	Stock Qty	Forecast Qty
1	Tire	0	4,280	71,618
2	Sweat Kit	0	60	35,889
3	Chain	0	64	35,889
4	Gas	0	14,354	35,889
5	Brake	0	14,252	35,889
6	Handle Bar	0	90	35,889
7	Pedal	0	31,606	71,618
8	Aluminium	0	380	135,103
9	Carbon Fiber	0	3,365	44,040
10	Red Paint 20KG	0	30	1,738

Gambar 4.38 Tampilan Antarmuka Melihat Ramalan Kebutuhan Barang

4.6.30. Antarmuka Melihat Laporan Pembelian

Pada antarmuka ini pengguna dapat melihat laporan pembelian dengan memilih tanggal mulai dan selesai, memilih supplier dan menekan tombol “*Submit*”. Disebelah kanan terdapat tombol “*View Detail*” yang apabila di klik akan menampilkan detail informasi dari *purchase order* tersebut. Tampilan antarmuka ini dapat dilihat pada Gambar 4.39.

Purchase Order Report

From Date: To Date:

Choose Supplier:

Purchase Order Summary Report
from date: 26 May 2016 to date: 26 May 2016

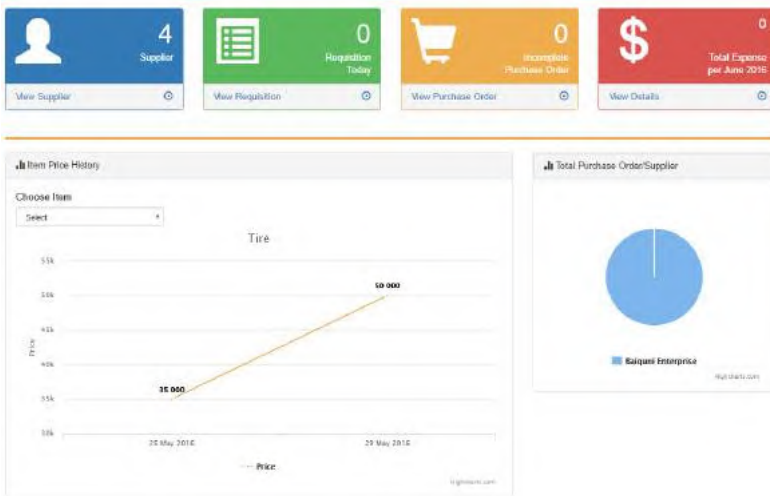
No	PO Number	Supplier Name	Date	Payment Status	Receipt	Item Defect %	Total Amount	
1	PO-REG-Forecast120-1	REGIPAKA	26 May 2016	Incomplete	On Time	0.20%	IDR 13,813,216,450.00	<input type="button" value="View Detail"/>
2	PO-REG-Forecast120-2	SANJANA	26 May 2016	Incomplete	On Time	0.03%	IDR 48,494,087,587.00	<input type="button" value="View Detail"/>
3	PO-REG-Forecast120-3	SURYA	26 May 2016	Incomplete	Late 13.85 hour	0.07%	IDR 59,508,291,425.00	<input type="button" value="View Detail"/>
							Total Expense: IDR 121,815,595,462.00	

Gambar 4.39 Tampilan Antarmuka Melihat Laporan Pembelian

4.6.31. Antarmuka *Dashboard* Halaman Depan

Pada antarmuka ini pengguna dapat melihat laporan singkat dari transaksi-transaksi yang ada, terdapat sejarah harga barang, dan terdapat informasi jumlah PO pada masing-masing supplier. Tampilan antarmuka ini dapat dilihat pada Gambar 4.40.

Purchasing



Gambar 4.40 Tampilan Antarmuka Dashboard

[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): 4 GB
 - c. Tipe sistem: 32-bit sistem operasi
2. Perangkat lunak
 - a. Sistem operasi: Windows 7 Professional
Perangkat pengembang: Python-2.7.11

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 Perencanaan Bisnis

Pengujian perencanaan bisnis dilakukan dengan memasukkan data-data perencanaan bisnis yang telah dibuat ke dalam aplikasi. Perencanaan bisnis ditunjukkan pada Tabel 5.1, Table 5.2, Tabel 5.3, Tabel 5.4, dan tabel 5.5.

5.2.1.1 Daftar Barang Jadi

Perencanaan bisnis berupa daftar barang jadi yang dihasilkan dan dijual oleh perusahaan ditunjukkan pada Tabel 5.1.

Tabel 5.1. Daftar Barang Jadi

No	Nama Barang
1	<i>Red Deluxe Touring Bike</i>
2	<i>Red Profesional Touring Bike</i>

Pada Tabel 5.1, ditunjukkan bahwa perusahaan ini memproduksi dua jenis sepeda. Dua jenis sepeda tersebut memiliki bahan dasar yang berbeda, *Deluxe Touring Bike* berbahan dasar aluminium, sedangkan *Profesional Touring Bike* berbahan dasar karbon.

5.2.1.2 Daftar Aset

Perencanaan bisnis berupa daftar aset yang dimiliki oleh perusahaan ditunjukkan pada Tabel 5.2.

Tabel 5.2. Daftar Aset

No.	Nama	Kuantitas
1	Tanah	180 x 150 m ²
2	Kantor	50 x 50 m ²
3	Parkir	60 x 50 m ²
4	Kantin	40 x 60 m ²
5	Pabrik	108 x 80 m ²
6	Pengolahan Limbah	20 x 40 m ²
7	<i>Raw Materials Inventory</i>	40 x 60 m ²
8	<i>Semi-Finished Good Inventory</i>	40 x 60 m ²
9	<i>Finished Good Inventory</i>	40 x 60 m ²
10	<i>Welding Machine</i>	2 baris
11	<i>Molding Machine</i>	2 baris
12	<i>Laser Cutting Machine</i>	2 baris
13	<i>Spray Painting Machine</i>	2 baris
14	<i>Testing Machine</i>	2 baris
15	Truk	10 buah
16	<i>Forklift</i>	6 buah
17	<i>Heavy Forklift</i>	6 buah

Pada Tabel 5.2, perusahaan memiliki 17 aset pada perencanaannya dengan kuantitas dari masing-masing aset yang ada.

5.2.1.3 Daftar Bahan Baku

Perencanaan bisnis berupa daftar bahan baku yang dimiliki oleh perusahaan ditunjukkan pada Tabel 5.3.

Tabel 5.3. Daftar Bahan Baku

No.	Nama bahan baku	Kuantitas	Satuan unit
1	<i>Tire</i>	1.2	<i>Pcs</i>
2	<i>Seat Kit</i>	0.5	<i>Pcs</i>
3	<i>Chain</i>	1.5	<i>Pcs</i>
4	<i>Gear</i>	1.8	<i>Pcs</i>
5	<i>Brake</i>	0.6	<i>Pcs</i>
6	<i>Handle Bar</i>	1.2	<i>Pcs</i>
7	<i>Pedal</i>	0.4	<i>Pcs</i>
8	<i>Aluminium</i>	2	m
9	<i>Carbon Fiber</i>	2	m
10	<i>Red Paint 20KG</i>	20	<i>Big Drum</i>
11	<i>Velg</i>	2.3	<i>Pcs</i>
12	<i>Tube</i>	0.4	<i>Pcs</i>
13	<i>Hex Nut 5mm</i>	0.04	<i>Pcs</i>
14	<i>Lock Washer 5mm</i>	0.06	<i>Pcs</i>
15	<i>Socket Head Bolt 5mm</i>	0.03	<i>Pcs</i>

Perencanaan yang diadakan oleh perusahaan ini akan menggunakan 15 macam bahan baku seperti yang dijabarkan pada Tabel 5.3. 15 bahan baku ini ditentukan dari daftar material yang akan diterangkan di Tabel 5.4.

5.2.1.4 Daftar Material

Perencanaan bisnis berupa daftar material yang dibutuhkan oleh perusahaan untuk membuat 1 (satu) barang jadi ditunjukkan pada Tabel 5.4.

Tabel 5.4. Daftar Material

No.	Bahan yang dibutuhkan	Kuantitas	Satuan unit	Bahan yang dihasilkan	
1	<i>Aluminium</i>	5	m	Kerangka untuk Red Deluxe Touring Bike	
2	<i>Paint drum with Red Color 20KG</i>	0.05	<i>Big drum</i>		
3	<i>Carbon Fiber</i>	5	m	Kerangka untuk Red Profesional Touring Bike	
4	<i>Paint drum with Red Color 20KG</i>	0.05	<i>Big drum</i>		
5	<i>Hex Nut 5 mm</i>	2	<i>Pcs</i>	Roda sepeda	
6	<i>Lock Washer 5 mm</i>	2	<i>Pcs</i>		
7	<i>Socket Head Bolt 5mm</i>	1	<i>Pcs</i>		
8	<i>Tire's Bike</i>	1	<i>Pcs</i>		
9	<i>Wheel's Tube</i>	1	<i>Pcs</i>		
10	<i>Wheel's Velg</i>	1	<i>Pcs</i>		
11	<i>Frame for Deluxe Touring Bike with Red Color</i>	1	<i>Pcs</i>		<i>Red Deluxe Touring Bike</i>
12	<i>Wheel's Bike</i>	2	<i>Pcs</i>		
13	<i>Chain's Bike</i>	1	<i>Pcs</i>		
14	<i>Handle Bar's Bike</i>	1	<i>Pcs</i>		
15	<i>Seat Kit's Bike</i>	1	<i>Pcs</i>		
16	<i>Pedal's Bike</i>	2	<i>Pcs</i>		
17	<i>Gear's Bike</i>	1	<i>Pcs</i>		
18	<i>Front and Rear Brake's Bike</i>	1	<i>Pcs</i>		
19	<i>Frame for Deluxe Touring Bike with Red Color</i>	1	<i>Pcs</i>	<i>Red Professional Touring Bike</i>	
20	<i>Wheel's Bike</i>	2	<i>Pcs</i>		
21	<i>Chain's Bike</i>	1	<i>Pcs</i>		
22	<i>Handle Bar's Bike</i>	1	<i>Pcs</i>		
23	<i>Seat Kit's Bike</i>	1	<i>Pcs</i>		
24	<i>Pedal's Bike</i>	2	<i>Pcs</i>		
25	<i>Gear's Bike</i>	1	<i>Pcs</i>		
26	<i>Front and Rear Brake's Bike</i>	1	<i>Pcs</i>		

Daftar material yang dijabarkan pada tabel 5.4. menjelaskan material-material yang dibutuhkan untuk memproduksi barang jadi yang dibutuhkan pada perusahaan ini.

5.2.1.5 Sumber Daya

Pada perusahaan ini memiliki beberapa sumber daya yang terdiri atas beberapa pegawai dan beberapa mesin yang terpisah menjadi 2 (dua) baris. Terdapat 150 orang sebagai *direct labour*, 32 orang sebagai *indirect labour*. Untuk mesin, terdapat *welding machine*, *molding machine*, *laser cutting machine*, *spray machine*, dan *testing machine* yang masing-masing mesin untuk 2 (dua) baris.

5.2.1.6 Perencanaan keuangan

Perencanaan keuangan dalam perusahaan ini digunakan untuk merincikan keuangan yang digunakan untuk pengelolaan keuangan dalam perusahaan. Perencanaan keuangan ditunjukkan pada Table 5.5.

Tabel 5.5. Perencanaan Keuangan Perusahaan

No.	Nama perencanaan keuangan	Jumlah (dalam Rp)
1	Total pembelian aset aktif	27.997.311.200,00
2	Total pembelian bahan baku	88.615.744.062,50
3	Biaya lain-lain	3,386,944,737.50
4	Peminjaman dari Bank	120.000.000.000,00

5.2.2. Pengujian Basis Data Terdistribusi

Pengujian Basis Data Terdistribusi (BDT) dilakukan untuk menguji 2 aspek utama dalam konsep BDT, yaitu replikasi dan *high availability*. Sistem BDT menggunakan 4 server fisik. Dengan rincian sebagai berikut:

1. *Server* aplikasi, adalah server yang di dalamnya terdapat *web server* beserta file aplikasi ERP.
2. *Management Node*, adalah *server* yang berfungsi sebagai pusat pengaturan sistem BDT.

3. *Data Node*, adalah 2 buah server penyimpanan data.

- **Prosedur Simulasi Replikasi**

1. Memastikan seluruh *server* dan sistem BDT dalam keadaan *online*.
2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi ERP. yang ditunjukkan pada Gambar 5.3.
3. Melakukan pengecekan terhadap hasil penambahan, perubahan, dan penghapusan data pada seluruh *data node* yang ditampilkan pada Gambar 5.4 dan Gambar 5.5

- ***High Availability***

1. Mematikan *server* aplikasi, *management server* dan sistem BDT pada salah satu *node*. Ditampilkan pada Gambar 5.6.
2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi yang ditunjukkan pada Gambar 5.7.
3. Melakukan pengecekan terhadap hasil pemrosesan data pada *node* yang masih aktif yang ditampilkan pada Gambar 5.8.

Warehouse 3

ID	3
Company	EZTENANT BIKE INDONESIA
Warehouse Name	EZERP Testing BDT

Gambar 5.1 Pengujian fitur Replikasi pada sistem

```
master@master-Aspire-M3970: ~
mysql> use test
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from iwm_warehouse where 1;
+-----+-----+-----+-----+-----+
| id | company_id | ws_name | created_at | updated_at |
+-----+-----+-----+-----+-----+
| 3 | 1 | EZERP Testing BDT | NULL | NULL |
| 1 | 1 | Maju | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```

Gambar 5.2 Pengujian Fitur Replikasi pada Database Server 1

```
root@node2-Aspire-M3970: /home/node2
mysql> use test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from iwm_warehouse where 1;
+-----+-----+-----+-----+-----+
| id | company_id | ws_name | created_at | updated_at |
+-----+-----+-----+-----+-----+
| 3 | 1 | EZERP Testing BDT | NULL | NULL |
| 1 | 1 | Maju | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Gambar 5.3 Pengujian Fitur Replikasi pada Database Server 2

Warehouse 3

ID	3
Company	EZTENANT BIKE INDONESIA
Warehouse Name	EZERP Testing BDT HA

Gambar 5.4 Pengujian Fitur High-Availability pada Sistem

```

root@master:~/Aspire-M3970 ~/home/master
[mysql@API] 2 node(s)
id=1 @10.151.04.182 [mysql-5.1.73 ndb-7.1.34]
id=4 @10.151.04.182 [mysql-5.1.73 ndb-7.1.34]
id=5 @10.151.04.203 [mysql-5.1.73 ndb-7.1.34]

ndb_mgm> Node 21 started (version 7.1.34)
Node 2: Node shutdown initiated
Node 2: Node shutdown completed.
show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 (not connected, accepting connect from 10.151.04.182)
id=3 @10.151.04.203 [mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

[ndb_mgmd(NM)] 1 node(s)
id=1 @10.151.04.181 [mysql-5.1.73 ndb-7.1.34]

[mysql@API] 2 node(s)
id=4 (not connected, accepting connect from 10.151.04.182)
id=5 @10.151.04.203 [mysql-5.1.73 ndb-7.1.34]

ndb_mgm>
  
```

```

master@master:~/Aspire-M3970 ~
mysql> select * from tw_warehouse where id
-----+-----+-----+-----+-----+-----
id | company_id | ws_name | created_at | updated_at
-----+-----+-----+-----+-----+-----
1 | 3 | EZERP Testing BDT HA | NULL | NULL
1 | 1 | Raju | 2016-03-22 21:00:14 | 2016-03-22 21:00:14
2 | 1 | EZERP BIKE INDONESIA 2 | NULL | NULL
-----+-----+-----+-----+-----+-----
3 rows in set (0.00 sec)

mysql>
  
```

Gambar 5.5 Pengujian Fitur High-Availability pada Database Server 1

```

root@node2:~/Aspire-M3970 ~/home/node1
[ndb_mgmd(NM)] 1 node(s)
id=1 @10.151.04.181 [mysql-5.1.73 ndb-7.1.34]

[mysql@API] 2 node(s)
id=4 @10.151.04.182 [mysql-5.1.73 ndb-7.1.34]
id=5 @10.151.04.203 [mysql-5.1.73 ndb-7.1.34]

ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 (not connected, accepting connect from 10.151.04.182)
id=3 @10.151.04.203 [mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

[ndb_mgmd(NM)] 1 node(s)
id=1 @10.151.04.181 [mysql-5.1.73 ndb-7.1.34]

[mysql@API] 2 node(s)
id=4 (not connected, accepting connect from 10.151.04.182)
id=5 @10.151.04.203 [mysql-5.1.73 ndb-7.1.34]

ndb_mgm>
  
```

```

node2@node2:~/Aspire-M3970 ~
node2@node2:~/Aspire-M3970:~$ sudo /etc/init.d/mysql stop
[sudo] password for node2:
Shutting down MySQL
... *
node2@node2:~/Aspire-M3970:~$
  
```

Gambar 5.6 Pengujian Fitur High-Availability pada Database Server 2

5.2.3. Pengujian RBAC

Pengujian ini terdiri dari pengujian mengelola *Role Based Access Control (RBAC)*. Rincian skenario pengujian pada kasus

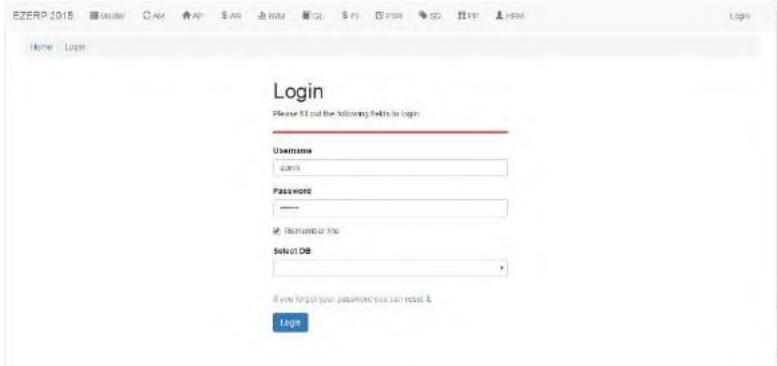
penggunaan pengujian mengelola *Role Based Access Control (RBAC)* dapat dilihat pada Tabel 5.6.

Tabel 5.6 Pengujian fitur mengelola Role Based Access Control (RBAC)

ID	UJ.UC.RBAC
Referensi Kasus Penggunaan	UC. RBAC
Nama	Pengujian mengelola <i>Role Based Access Control (RBAC)</i> .
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Role Based Access Control (RBAC)</i> .
Skenario 1	<i>Pengguna membuat user baru.</i>
Kondisi Awal	Pengguna berada pada halaman dashboard admin.
Data Uji	Masukkan data <i>user</i> baru:
Langkah Pengujian	Pengguna memilih kegiatan <i>Add New User</i> , lalu memasukkan data dan menekan tombol “ <i>Add User</i> ”.
Hasil Yang Diharapkan	<i>User</i> yang baru terdapat pada halaman daftar <i>user</i> .
Hasil Yang Didapat	<i>User</i> yang baru terdapat pada halaman daftar <i>user</i> .
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman halaman daftar <i>user</i> dengan tambahan data <i>user</i> yang baru.
Skenario 2	<i>Pengguna menyunting data user tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>user</i> .
Data Uji	Perubahan data <i>user</i> :
Langkah Pengujian	Pengguna memilih kegiatan menyunting <i>user</i> tertentu, lalu melakukan perubahan data dan menekan tombol “ <i>Update</i> ”.
Hasil Yang Diharapkan	<i>User</i> yang disunting mengalami perubahan data sesuai perubahan.
Hasil Yang Didapat	<i>User</i> yang disunting mengalami perubahan data sesuai perubahan.
Hasil Pengujian	Berhasil.

Kondisi Akhir	Pengguna berada pada halaman daftar user.
Skenario 3	<i>Pengguna menghapus user tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar user.
Data Uji	-
Langkah Pengujian	Pengguna memilih kegiatan menghapus kegiatan tertentu, lalu menekan tombol “Yes” untuk konfirmasi penghapusan.
Hasil Yang Diharapkan	<i>User yang baru dihapus tidak tampil pada daftar user.</i>
Hasil Yang Didapat	<i>User yang baru dihapus tidak tampil pada daftar user.</i>
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>user</i> .

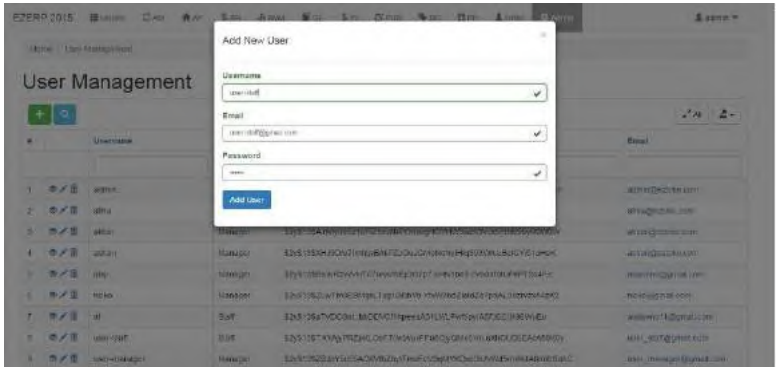
Gambar 5.9 menunjukkan proses login oleh admin. Gambar 5.10 menunjukkan hasil setelah proses login oleh admin. Gambar 5.11 menunjukkan proses pemasukan data *user* untuk skenario 1. Kemudian pada Gambar 5.12 menunjukkan proses penyuntingan data *user* untuk skenario 2. Sedangkan pada Gambar 5.13 dapat dilihat proses penghapusan *user* untuk skenario 3. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan UC.RBAC telah bekerja dengan baik seperti yang diharapkan.



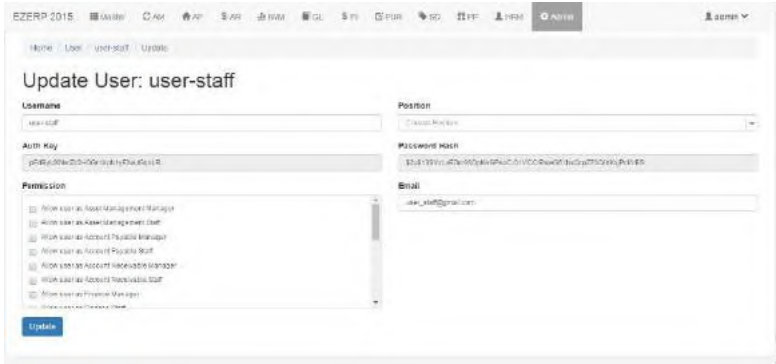
Gambar 5.7 Proses login admin



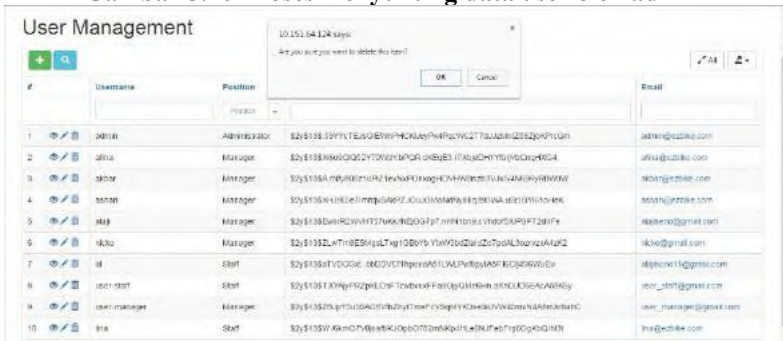
Gambar 5.8 Tampilan awal setelah login berhasil dilakukan



Gambar 5.9 Proses admin menambahkan user baru



Gambar 5.10 Proses menyunting data user oleh admin



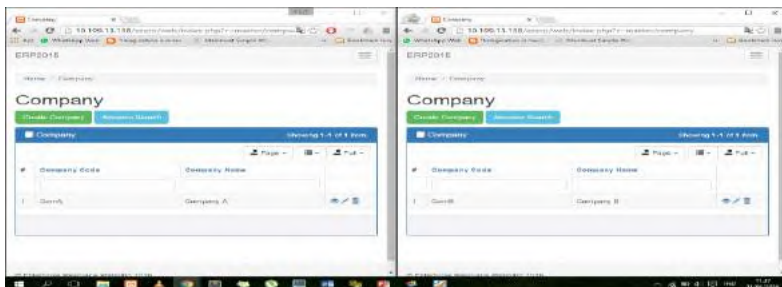
Gambar 5.11 Proses penghapusan user oleh admin

5.2.4. Pengujian Multi-Tenant

Pengujian ini terdiri dari pengujian mengelola *multitenancy*. Pada pengujian *multitenancy*, *tenant* baru yang akan mendaftarkan *tenant*-nya untuk dapat mengakses sistem memasukkan data berupa nama basis data yang akan dibuat. Apabila basis data berhasil dibuat, maka *tenant* dapat mulai mengakses dengan pengguna awal sebagai administrator. Hasil pengujian *Multitenancy* digambarkan pada Gambar 5.14.

Tabel 5.7 Pengujian Fitur Multi-Tenant

ID	UJ-MTC
Referensi	Multitenancy
Nama	Mengelola Multitenancy
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola Multitenancy
Skenario 1	Pengguna mengakses Tenant
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Data Uji	Data <i>Tenant</i>
Langkah Pengujian	Pengguna memasukkan data uji ke form yang tersedia dan memilih tombol login
Hasil Yang Diharapkan	Pengguna berada di halaman utama dengan data <i>tenant</i> -nya
Hasil Yang Didapat	Pengguna berada di halaman utama dengan data <i>tenant</i> -nya
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama



Gambar 5.12 Hasil dari pengujian multi-tenant

5.2.5. Pengujian Fungsionalitas

Pengujian fungsionalitas sistem dilakukan secara mandiri dengan menyiapkan sejumlah skenario sebagai tolak ukur keberhasilan pengujian. Pengujian fungsionalitas dilakukan dengan mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.1.5. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut.

5.2.5.1. Pengujian mengelola *Purchase Requisition*

Pengujian ini terdiri dari pengujian mengelola data *Purchase Requisition*. Pengujian mengelola data *Purchase Requisition* yaitu membuat, menyunting, serta menghapus data *Purchase Requisition* yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.4

Tabel 5.8 Pengujian Fitur Mengelola *Purchase Requisition*

ID	UJ-UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Pengujian Fitur Mengelola <i>Purchase Requisition</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Purchase Requisition</i>
Skenario 1	<i>Pengguna menambah data Purchase Requisition</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Purchase Requisition</i>
Data Uji	Masukkan data <i>Purchase Requisition</i> :
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Purchase Requisition</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	Aktor berada pada halaman melihat data yang baru

	saja dimasukkan.
Skenario 2	Aktor menyunting data Purchase Requisition
Kondisi Awal	Aktor berada pada halaman utama <i>Purchase Requisition</i>
Data Uji	Perubahan data pada Item Type :
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Purchase Requisition</i> yang ada pada sistem, lalu menyunting data tersebut pada halaman menyunting dan menyimpannya kembali ke basis data.
Hasil Yang Diharapkan	Data yang disunting akan berubah dan disimpan kembali ke basis data.
Hasil Yang Didapat	Data yang disunting tersimpan di basis data
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja disunting.
Skenario 3	Aktor menghapus data Purchase Requisition
Kondisi Awal	Aktor berada pada halaman utama <i>Purchase Requisition</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Purchase Requisition</i> yang ada pada sistem, lalu melakukan konfirmasi penghapusan data tersebut dan data akan dihapus oleh sistem.
Hasil Yang Diharapkan	Data yang dihapus akan terhapus di basis data
Hasil Yang Didapat	Data berhasil di hapus.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>Purchase Requisition</i> .

➤ Skenario 1

Skenario ini dimulai ketika aktor menekan tombol Create Purchase Requisition dan sistem menampilkan form pembuatan data baru dan aktor akan mengisi form yang disediakan dan menekan kembali tombol Create. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja dimasukkan ke dalam basis data.

➤ Skenario 2

Skenario ini dimulai ketika aktor menekan tombol Update pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan form penyuntingan data dan aktor akan menyunting data sesuai dengan kebutuhan dan menekan kembali tombol Update. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja disunting ke dalam basis data.

➤ Skenario 3

Skenario Ini dimulai ketika aktor menekan tombol delete pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan konfirmasi untuk penghapusan data. Proses ini berakhir saat aktor melakukan konfirmasi penghapusan data dan system akan menghapus data tersebut di dalam basis data dan menampilkan kembali halaman utama dari Purchase Requisition dengan data yang dihapus sudah terhapus.

5.2.5.2. Pengujian mengelola *Request for Quotation*

Pengujian ini terdiri dari pengujian mengelola data *Request for Quotation*. Pengujian mengelola data *Request for Quotation* yaitu membuat, menyunting, serta menghapus data *Request for Quotation* yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.5

Tabel 5.9 Pengujian Fitur Mengelola Request for Quotation

ID	UJ-UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Pengujian Fitur Mengelola <i>Request for Quotation</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Request for Quotation</i>
Skenario 1	<i>Pengguna menambah data Request for Quotation</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Request for Quotation</i>
Data Uji	Masukkan data <i>Request for Quotation</i> :
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Request for Quotation</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	Aktor berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Aktor menyunting data Request for Quotation</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Request for Quotation</i>
Data Uji	Perubahan data pada <i>Request for Quotation</i> :
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Request for Quotation</i> yang ada pada sistem, lalu menyunting data tersebut pada halaman menyunting dan menyimpannya kembali ke basis data.
Hasil Yang Diharapkan	Data yang disunting akan berubah dan disimpan kembali ke basis data.
Hasil Yang Didapat	Data yang disunting tersimpan di basis data
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja disunting.
Skenario 3	<i>Aktor menghapus data Request for Quotation</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Request for Quotation</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Request for Quotation</i> yang ada pada sistem, lalu melakukan konfirmasi penghapusan data tersebut dan data akan dihapus oleh sistem.
Hasil Yang Diharapkan	Data yang dihapus akan terhapus di basis data
Hasil Yang Didapat	Data berhasil di hapus.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>Request for Quotation</i> .

➤ Skenario 1

Skenario ini dimulai ketika aktor menekan tombol Create Request for Quotation dan sistem menampilkan form pembuatan data baru dan aktor akan mengisi form yang disediakan dan menekan kembali tombol. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja dimasukkan ke dalam basis data.

➤ Skenario 2

Skenario ini dimulai ketika aktor menekan tombol Update pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan form penyuntingan data dan aktor akan menyunting data sesuai dengan kebutuhan dan menekan kembali tombol Update. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja disunting ke dalam basis data.

➤ Skenario 3

Skenario Ini dimulai ketika aktor menekan tombol delete pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan konfirmasi untuk penghapusan data. Proses ini berakhir saat aktor melakukan konfirmasi penghapusan data dan system akan menghapus data tersebut di dalam basis data dan menampilkan kembali halaman utama dari Request for Quotation dengan data yang dihapus sudah terhapus.

5.2.5.3. Pengujian mengelola *Quotation*

Pengujian ini terdiri dari pengujian mengelola data *Quotation*. Pengujian mengelola data *Quotation* yaitu membuat, menyunting, serta menghapus data *Quotation* yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.6

Tabel 5.10 Pengujian Fitur Mengelola *Quotation*

ID	UJ-UC-003
Referensi Kasus Penggunaan	UC-003
Nama	Pengujian Fitur Mengelola <i>Quotation</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Quotation</i>

Skenario 1	<i>Pengguna menambah data Quotation</i>
Kondisi Awal	Aktor berada pada halaman utama Quotation
Data Uji	Masukkan data Quotation :
Langkah Pengujian	Aktor masuk ke halaman membuat data Quotation 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	Aktor berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Aktor menyunting data Quotation</i>
Kondisi Awal	Aktor berada pada halaman utama Quotation
Data Uji	Perubahan data pada Item Quotation :
Langkah Pengujian	Pengguna memilih menyunting salah satu data Quotation yang ada pada sistem, lalu menyunting data tersebut pada halaman menyunting dan menyimpannya kembali ke basis data.
Hasil Yang Diharapkan	Data yang disunting akan berubah dan disimpan kembali ke basis data.
Hasil Yang Didapat	Data yang disunting tersimpan di basis data
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja disunting.
Skenario 3	<i>Aktor menghapus data Quotation</i>
Kondisi Awal	Aktor berada pada halaman utama Quotation
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data Quotation yang ada pada sistem, lalu melakukan konfirmasi penghapusan data tersebut dan data akan dihapus oleh sistem.
Hasil Yang Diharapkan	Data yang dihapus akan terhapus di basis data
Hasil Yang Didapat	Data berhasil di hapus.

Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>Quotation</i> .

➤ Skenario 1

Skenario ini dimulai ketika aktor menekan tombol Create Quotation dan sistem menampilkan form pembuatan data baru dan aktor akan mengisi form yang disediakan dan menekan kembali tombol Create. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja dimasukkan ke dalam basis data.

➤ Skenario 2

Skenario ini dimulai ketika aktor menekan tombol Update pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan form penyuntingan data dan aktor akan menyunting data sesuai dengan kebutuhan dan menekan kembali tombol Update. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja disunting ke dalam basis data.

➤ Skenario 3

Skenario Ini dimulai ketika aktor menekan tombol delete pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan konfirmasi untuk penghapusan data. Proses ini berakhir saat aktor melakukan konfirmasi penghapusan data dan system akan menghapus data tersebut di dalam basis data dan menampilkan kembali halaman utama dari Quotation dengan data yang dihapus sudah terhapus.

5.2.5.4. Pengujian mengelola *Purchase Order*

Pengujian ini terdiri dari pengujian mengelola data *Purchase Order*. Pengujian mengelola data *Purchase Order* yaitu membuat, menyunting, serta menghapus data *Purchase Order* yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.7

Tabel 5.11 Pengujian Fitur Mengelola Purchase Order

ID	UJ-UC-004
-----------	-----------

Referensi Kasus Penggunaan	UC-004
Nama	Pengujian Fitur Mengelola <i>Purchase Order</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Purchase Order</i>
Skenario 1	<i>Pengguna menambah data Purchase Order</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Purchase Order</i>
Data Uji	Masukkan data <i>Purchase Order</i> :
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Purchase Order</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	Aktor berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Aktor menyunting data Purchase Order</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Purchase Order</i>
Data Uji	Perubahan data pada <i>Purchase Order</i> :
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Purchase Order</i> yang ada pada sistem, lalu menyunting data tersebut pada halaman menyunting dan menyimpannya kembali ke basis data.
Hasil Yang Diharapkan	Data yang disunting akan berubah dan disimpan kembali ke basis data.
Hasil Yang Didapat	Data yang disunting tersimpan di basis data
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja disunting.
Skenario 3	<i>Aktor menghapus data Purchase Order</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Purchase Order</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Purchase Order</i> yang ada pada sistem, lalu melakukan

	konfirmasi penghapusan data tersebut dan data akan dihapus oleh sistem.
Hasil Yang Diharapkan	Data yang dihapus akan terhapus di basis data
Hasil Yang Didapat	Data berhasil di hapus.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>Purchase Order</i> .

➤ Skenario 1

Skenario ini dimulai ketika aktor menekan tombol Create Purchase Order dan sistem menampilkan form pembuatan data baru dan aktor akan mengisi form yang disediakan dan menekan kembali tombol Create. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja dimasukkan ke dalam basis data.

➤ Skenario 2

Skenario ini dimulai ketika aktor menekan tombol Update pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan form penyuntingan data dan aktor akan menyunting data sesuai dengan kebutuhan dan menekan kembali tombol Update. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja disunting ke dalam basis data.

➤ Skenario 3

Skenario Ini dimulai ketika aktor menekan tombol delete pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan konfirmasi untuk penghapusan data. Proses ini berakhir saat aktor melakukan konfirmasi penghapusan data dan system akan menghapus data tersebut di dalam basis data dan menampilkan kembali halaman utama dari Purchase Order dengan data yang dihapus sudah terhapus.

5.2.5.5. Pengujian mengelola *Return Order*

Pengujian ini terdiri dari pengujian mengelola data *Return Order*. Pengujian mengelola data *Return Order* yaitu membuat, menyunting, serta menghapus data *Return Order* yang sudah ada

pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.8

Tabel 5.12 Pengujian Fitur Mengelola Return Order

ID	UJ-UC-005
Referensi Kasus Penggunaan	UC-005
Nama	Pengujian Fitur Mengelola <i>Return Order</i>
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Return Order</i>
Skenario 1	<i>Pengguna menambah data Return Order</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Return Order</i>
Data Uji	Masukkan data <i>Return Order</i> :
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Return Order</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	Aktor berada pada halaman melihat data yang baru saja dimasukkan.
Skenario 2	<i>Aktor menyunting data Return Order</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Return Order</i>
Data Uji	Perubahan data pada <i>Return Order</i> :
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Return Order</i> yang ada pada sistem, lalu menyunting data tersebut pada halaman menyunting dan menyimpannya kembali ke basis data.
Hasil Yang Diharapkan	Data yang disunting akan berubah dan disimpan kembali ke basis data.
Hasil Yang Didapat	Data yang disunting tersimpan di basis data
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman melihat data yang baru saja disunting.
Skenario 3	<i>Aktor menghapus data Return Order</i>

Kondisi Awal	Aktor berada pada halaman utama <i>Return Order</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Return Order</i> yang ada pada sistem, lalu melakukan konfirmasi penghapusan data tersebut dan data akan dihapus oleh sistem.
Hasil Yang Diharapkan	Data yang dihapus akan terhapus di basis data
Hasil Yang Didapat	Data berhasil di hapus.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman utama <i>Return Order</i> .

➤ Skenario 1

Skenario ini dimulai ketika aktor menekan tombol Create Return Order dan sistem menampilkan form pembuatan data baru dan aktor akan mengisi form yang disediakan dan menekan kembali tombol Create. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja dimasukkan ke dalam basis data.

➤ Skenario 2

Skenario ini dimulai ketika aktor menekan tombol Update pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan form penyuntingan data dan aktor akan menyunting data sesuai dengan kebutuhan dan menekan kembali tombol Update. Setelah itu sistem akan menampilkan halaman melihat data yang baru saja disunting ke dalam basis data.

➤ Skenario 3

Skenario Ini dimulai ketika aktor menekan tombol delete pada salah satu data yang terdapat pada sistem dan sistem akan menampilkan konfirmasi untuk penghapusan data. Proses ini berakhir saat aktor melakukan konfirmasi penghapusan data dan system akan menghapus data tersebut di dalam basis data dan

menampilkan kembali halaman utama dari Return Order dengan data yang dihapus sudah terhapus.

5.2.5.6. Pengujian Menganalisa Supplier

Pengujian ini terdiri dari pengujian menganalisa supplier. Pengujian menganalisa supplier yaitu melihat ranking supplier berdasarkan data dari purchase order dan goods receipt. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.9

Tabel 5.13 Pengujian Fitur Menganalisa Supplier

ID	UJ-UC-006
Referensi Kasus Penggunaan	UC-006
Nama	Pengujian Fitur Menganalisa Supplier
Tujuan Pengujian	Menguji kemampuan aplikasi untuk menentukan ranking supplier terbaik
Skenario 1	<i>Pengguna menganalisa Supplier</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Supplier Analysis</i>
Data Uji	Data <i>Purchase Order dan Good Receipt</i> :
Langkah Pengujian	Aktor masuk ke halaman supplier analysis dan mengisi form yang ada dan mengklik tombol submit.
Hasil Yang Diharapkan	Berhasil meranking supplier dengan benar.
Hasil Yang Didapat	Berhasil meranking supplier dengan benar.
Hasil Pengujian	Berhasil

➤ Skenario 1

Skenario ini dimulai ketika aktor masuk ke halaman utama supplier analysis, kemudian aktor mengisi form yang ada kemudian menekan tombol submit, hasil dari proses ini bisa dilihat pada Gambar 4.35. Terdapat persamaan antara hasil analisa yang dihitung manual pada Microsoft Exel dengan yang

dihitung pada aplikasi ERP ini. Hasilnya dapat dilihat pada Tabel 5.10.

Tabel 5.14 Perhitungan Supplier Analysis

Perhitungan Manual (Excel)			Perhitungan Aplikasi ERP		
Supplier	Score	Rank	Supplier	Score	Rank
SF-Supplier 01	0.91	1	SF-Supplier 01	0.907	1
Wim Supply	0.79	2	Wim Supply	0.785	2
Sanjaya Enterprise	0.02	3	Sanjaya Enterprise	0.023	3

Supplier Analysis for Item: **Gear**
from 13 July 2016

Rank	Total PO	Supplier Name	Avg Late (Hour(s))	Avg Item Defect(%)	Avg Price	Total Score
1	1	SF-SUPPLIER01	1.80	6.00%	158,090.00	0.907
2	1	Wim Supply	2.81	5.00%	158,090.00	0.785
3	1	Sanjaya Enterprise	5.81	18.00%	145,090.00	0.023

Gambar 5.13 Hasil Supplier Analysis pada Aplikasi ERP

5.2.5.7. Pengujian Melihat Laporan Pembelian

Pengujian ini terdiri dari pengujian melihat laporan pembelian. Pengujian melihat laporan pembelian yaitu melihat laporan singkat dari *purchase order* untuk periode tertentu.. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.15.

Tabel 5.15 Pengujian Fitur Melihar laporan pembelian

ID	UJ-UC-007
Referensi Kasus Penggunaan	UC-007
Nama	Pengujian Fitur Melihat Laporan Pembelian
Tujuan Pengujian	Menguji kemampuan aplikasi untuk melihat laporan pembelian
Skenario 1	<i>Pengguna melihat laporan pembelian</i>
Kondisi Awal	Aktor berada pada halaman utama <i>PO report</i>

Data Uji	Data <i>Purchase Order</i> dan <i>Good Receipt</i> :
Langkah Pengujian	Aktor masuk ke halaman PO Report dan mengisi tanggal mulai dan tanggal selesai, memilih supplier (optional), menekan tombol submit.
Hasil Yang Diharapkan	Berhasil memberikan report purchase order, beserta status pembayaran, jumlah barang rusak, dan total pembelian pada periode tertentu
Hasil Yang Didapat	Berhasil memberikan report purchase order, beserta status pembayaran, jumlah barang rusak, dan total pembelian pada periode tertentu
Hasil Pengujian	Berhasil

➤ Skenario 1

Skenario ini dimulai ketika aktor masuk ke halaman utama PO Report, kemudian aktor mengisi form yang ada kemudian menekan tombol submit, hasil dari proses ini bisa dilihat pada Gambar 4.39

5.2.5.8. Pengujian Melihat Sejarah Harga Barang

Pengujian ini terdiri dari pengujian melihat sejarah harga barang. Pengujian melihat sejarah barang yaitu melihat harga-harga barang dari transaksi sebelumnya untuk barang tertentu.. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.16.

Tabel 5.16 Pengujian Fitur Melihat Sejarah Harga Barang

ID	UJ-UC-008
Referensi Kasus Penggunaan	UC-008
Nama	Pengujian Fitur Melihat Sejarah Harga Barang
Tujuan Pengujian	Menguji kemampuan aplikasi untuk melihat sejarah harga barang
Skenario 1	<i>Pengguna melihat sejarah harga barang</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Item Price History</i>
Data Uji	Data <i>Purchase Order</i> :
Langkah Pengujian	Aktor masuk ke halaman <i>Item Price History</i> , memilih barang yang ingin dilihat sejarah harganya, dan

	mengklik tombol submit.
Hasil Yang Diharapkan	Berhasil memberikan sejarah harga barang
Hasil Yang Didapat	Berhasil memberikan sejarah harga barang
Hasil Pengujian	Berhasil

➤ Skenario 1

Skenario ini dimulai ketika aktor masuk ke halaman utama ItemPrice History, kemudian aktor memilih item yang ingin dilihat sejarah harga barangnya kemudian menekan tombol submit, hasil dari proses ini bisa dilihat pada Gambar 4.37.

5.2.5.9. Pengujian Melihat Ramalan Kebutuhan Barang Mentah

Pengujian ini terdiri dari pengujian melihat ramalan kebutuhan barang mentah. Pengujian melihat ramalan barang mentah yaitu melihat ramalan kebutuhan barang mentah untuk tanggal tertentu, melihat barang yang sudah dipesan pada tanggal tertentu, dan melihat jumlah barang yang berada di gudang saat ini. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.17.

Tabel 5.17 Pengujian Fitur Melihat Ramalan Kebutuhan Barang Mentah

ID	UJ-UC-009
Referensi Kasus Penggunaan	UC-009
Nama	Pengujian Fitur Melihat Ramalan Kebutuhan Barang Mentah
Tujuan Pengujian	Menguji kemampuan aplikasi untuk melihat ramalan kebutuhan barang mentah, melihat jumlah barang yang sudah dipesan pada periode tertentu, dan melihat barang yang sudah ada di gudang.
Skenario 1	<i>Pengguna melihat ramalan kebutuhan barang mentah</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Item Forecast</i>

Data Uji	-
Langkah Pengujian	Aktor masuk ke halaman <i>Item Forecast</i> , memilih tahun dan bulan, dan mengklik tombol submit.
Hasil Yang Diharapkan	Berhasil memperlihatkan ramalan kebutuhan barang mentah, melihat jumlah barang yang sudah dipesan pada periode tertentu, dan melihat barang yang sudah ada di gudang.
Hasil Yang Didapat	Berhasil memperlihatkan ramalan kebutuhan barang mentah, melihat jumlah barang yang sudah dipesan pada periode tertentu, dan melihat barang yang sudah ada di gudang.
Hasil Pengujian	Berhasil

➤ Skenario 1

Skenario ini dimulai ketika aktor masuk ke halaman utama *Item Forecast*, kemudian aktor memasukkan tanggal mulai dan tanggal selesai kemudian menekan tombol submit, hasil dari proses ini bisa dilihat pada Gambar 4.38.

5.2.5.10. Pengujian Melihat Sejarah Kebutuhan Barang Mentah

Pengujian ini terdiri dari pengujian melihat sejarah kebutuhan barang mentah. Pengujian melihat sejarah kebutuhan barang mentah yaitu melihat jumlah barang dari transaksi-transaksi periode sebelumnya untuk barang tertentu.. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.18.

Tabel 5.18 Pengujian Fitur Melihat Sejarah Kebutuhan Barang Mentah

ID	UJ-UC-0010
Referensi Kasus Penggunaan	UC-0010
Nama	Pengujian Fitur Melihat Sejarah Kebutuhan Barang Mentah
Tujuan Pengujian	Menguji kemampuan aplikasi untuk melihat sejarah kebutuhan barang mentah
Skenario 1	<i>Pengguna melihat sejarah kebutuhan barang mentah</i>

Kondisi Awal	Aktor berada pada halaman utama <i>Item Demand History</i>
Data Uji	Data <i>Purchase Order</i> .
Langkah Pengujian	Aktor masuk ke halaman <i>Item Demand History</i> , memilih barang yang ingin dilihat sejarah kebutuhannya, dan mengklik tombol submit.
Hasil Yang Diharapkan	Berhasil memberikan sejarah kebutuhan barang
Hasil Yang Didapat	Berhasil memberikan sejarah kebutuhan barang
Hasil Pengujian	Berhasil

➤ Skenario 1

Skenario ini dimulai ketika aktor masuk ke halaman utama *Item Demand History*, kemudian aktor memilih item yang ingin dilihat sejarah kebutuhan barangnya kemudian menekan tombol submit, hasil dari proses ini bisa dilihat pada Gambar 4.36.

5.3 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional dan evaluasi pengujian kegunaan.

5.3.1 Evaluasi Pengujian Fungsionalitas Sistem

Rangkuman mengenai hasil pengujian fungsionalitas sistem dapat dilihat pada Tabel 5.19. 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 5.19. Evaluasi Pengujian Fungsionalitas Sistem

No.	ID	Nama Kasus Penggunaan	Skenario	Hasil
1	UJ.UC.001	Mengelola <i>Purchase</i>	Skenario 1	Berhasil

No.	ID	Nama Kasus Penggunaan	Skenario	Hasil
		<i>Requisition</i>	Skenario 2	Berhasil
			Skenario 3	Berhasil
2	UJ.UC.002	Mengelola <i>Request for Quotation</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
3	UJ.UC.003	Mengelola <i>Quotation</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
4	UJ.UC.004	Mengelola Purchase Order	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
5	UJ.UC.005	Mengelola Return Order	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
6	UJ.UC.006	Menganalisa Supplier	Skenario 1	Berhasil
7	UJ.UC.007	Melihat Laporan Pembelian	Skenario 1	Berhasil
8	UJ.UC.008	Melihat Sejarah Harga Barang	Skenario 1	Berhasil
9	UJ.UC.009	Melihat Ramalan Kebutuhan Barang Mentah	Skenario 1	Berhasil
10	UJ.UC.010	Melihat Sejarah Kebutuhan Barang Mentah	Skenario 1	Berhasil

[Halaman ini sengaja dikosongkan]

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.

5.4 Kesimpulan

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

1. Aplikasi *procurement process* ini memudahkan pencatatan dalam pemesanan barang, dari permintaan barang sampai pemesanan barang secara resmi kepada *supplier*.
2. Metode TOPSIS berhasil digunakan untuk menentukan *supplier* terbaik dan meranking *supplier*.
3. Transaksi-transaksi pembelian yang sudah dilakukan sebelumnya seperti *purchase order* dan *goods receipt* bisa digunakan sebagai kriteria untuk menganalisa *supplier*.
4. Menggunakan lebih dari satu *database server* dan menggunakan teknik replikasi pada *distributed database* mampu mengatasi kegagalan sistem yang disebabkan oleh *database server*.

5.5 Saran

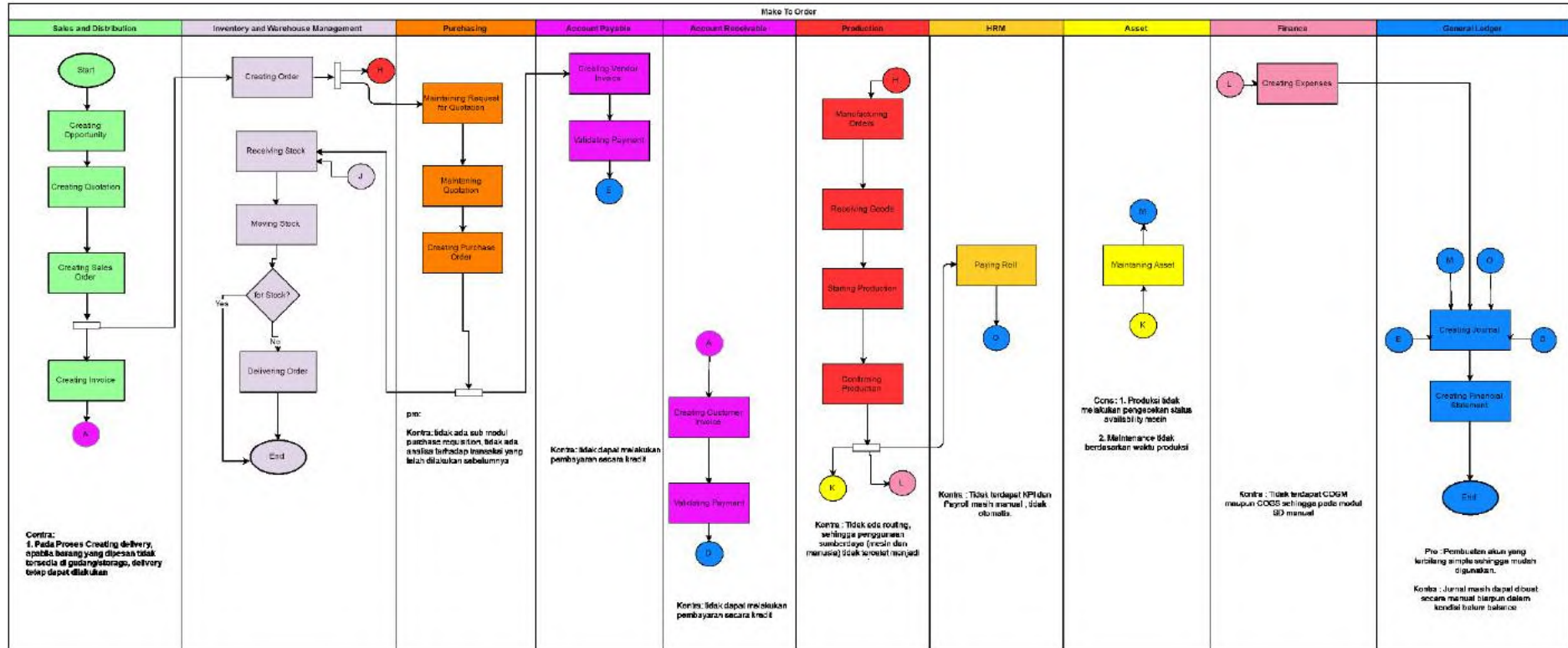
Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Penambahan fitur penjadwalan yang lebih lengkap agar pembelian barang bisa berjalan lebih lancar, dan gudang menjadi lebih efisien.
2. Penambahan fitur *blanket order* pada *purchase order*.
3. Menambah kriteria pada *supplier analysis*

4. Menambahkan fitur yang bisa menganalisa *quotation* secara otomatis

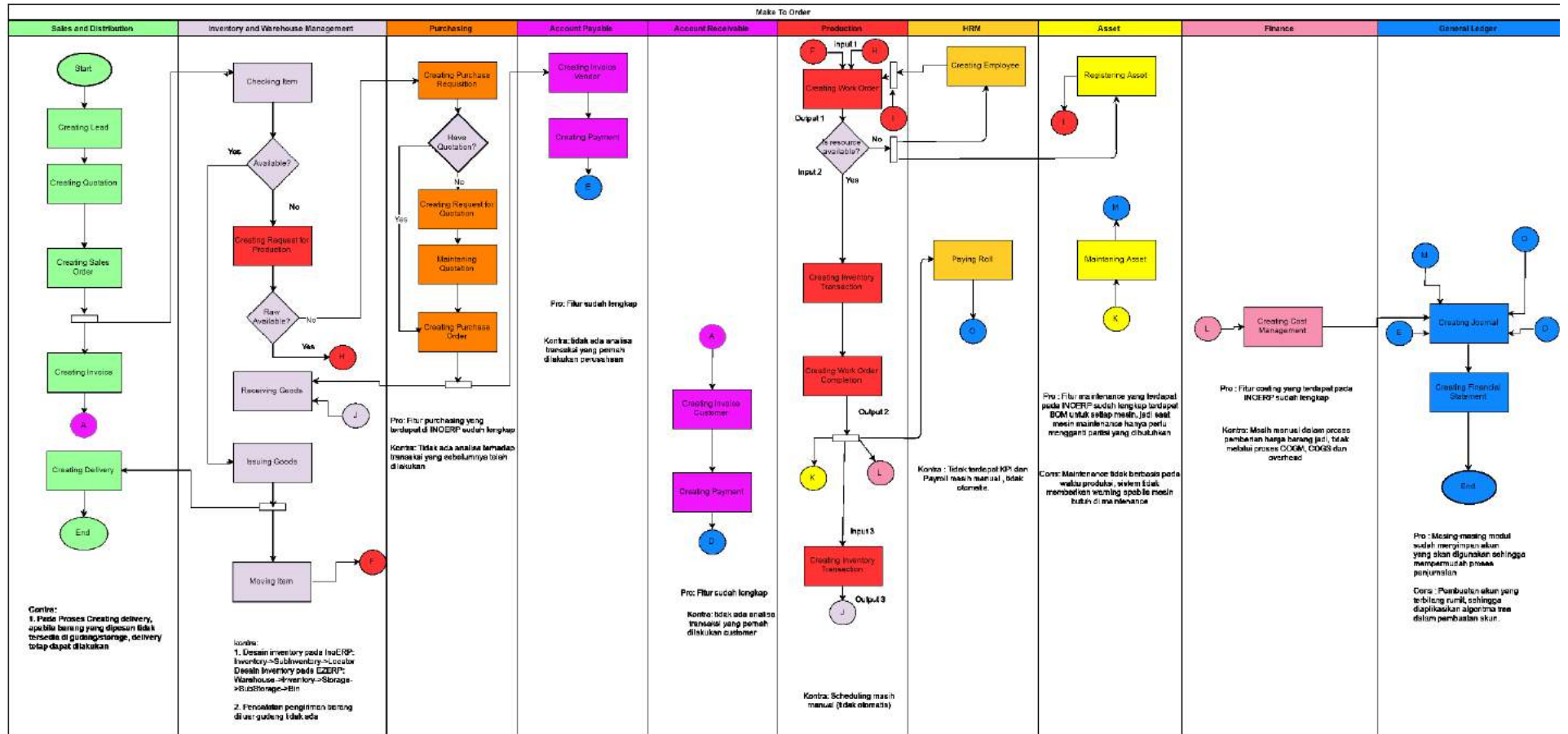
LAMPIRAN A – PROSES BISNIS

A.1 Proses Bisnis Odoo



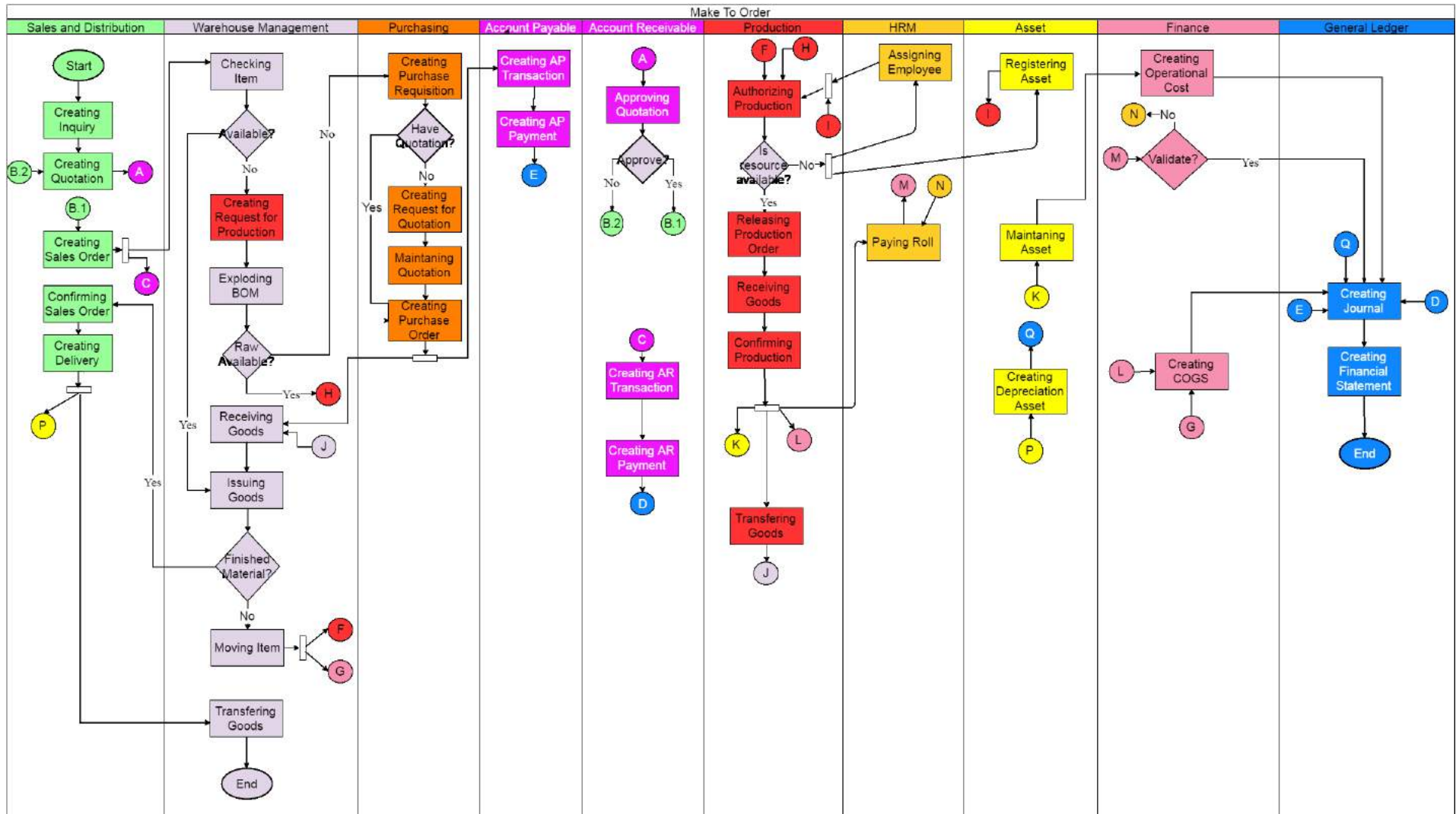
Gambar 0.1 Proses Bisnis Odoo

A.2 Proses Bisnis InoERP



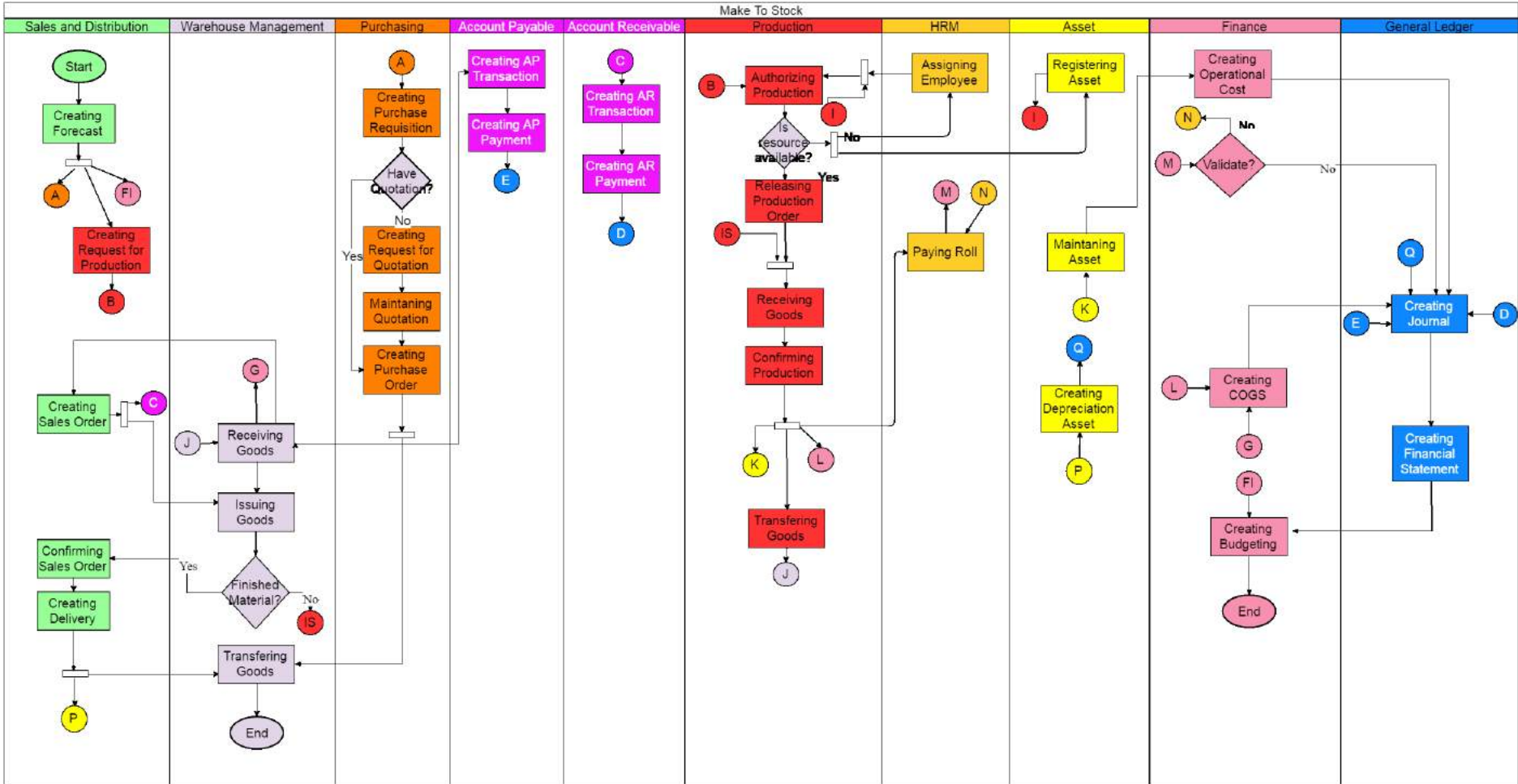
Gambar 0.2. Proses Bisnis InoERP

A.4 Proses Bisnis Make-To-Order ERP 2016



Gambar 0.4. Proses Bisnis Make-To-Order ERP 2016

A.5 Proses Bisnis Make-To-Stock ERP 2016



Gambar 0.5. Proses Bisnis Make-To-Stock ERP 2016

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- 1] A. Rifa'i, R. Sarno and D. Sunaryono, Rancang Bangun Perangkat Lunak Berorientasi Arsitektur Service (SOA) Dengan Pendekatan Workflow Pada Domain Fungsional Supplier Relationship Management (SRM) Untuk Sistem Enterprise Resource Planning, Surabaya, 2013.
- 2] S. R. Magal and J. Word, Integrated Business Processes with ERP Systems, New Jersey: Wiley and Sons, Inc., 2012.
- 3] R. Sarno and U. Yudatama, "Evaluation maturity index and risk management for it governance using Fuzzy AHP and Fuzzy TOPSIS (case Study Bank XYZ)," in *Intelligent Technology and Its Applications (ISITIA), 2015 International Seminar on*, IEEE, 2015, pp. 323--328.
- 4] "BPMN Modeling Reference," [Online]. Available: <https://camunda.org/bpmn/reference/>.
- 5] F. Chong, G. Carraro and R. Wolter, "Multi-Tenant Data Architecture," 2006. [Online]. Available: <https://msdn.microsoft.com/en-us/library/aa479086.aspx>. [Accessed 1 February 2016].
- 6] Sharive, Proyek Membangun Website dengan Yii Framework, Yogyakarta: Lokomedia, 2014.
- 7] Fathansyah, Basis Data, Bandung: Informatika, 2015.
- 8] Kasman and D. A, Trik Mudah Menguasai OOP dengan PHP, Yogyakarta: Lokomedia, 2013.
- 9] U. Yudatama and R. Sarno, "Priority Determination for Higher Education Strategic Planning Using Balanced Scorecard, FAHP and TOPSIS (Case study: XYZ University)," *IOP Conference Series: Materials Science and Engineering*, p. 012040, 2016.
- 10] P. P. Sendy and S. Sony, "Pemilihan Pemasok Terbaik dengan Metode TOPSIS Fuzzy MCDM," *ITS Paper*, 2014.

- "wikipedia," October 2010. [Online]. Available:
11] <https://en.wikipedia.org/wiki/Procurement>.
Andrew, "MySQL Cluster 7.4," 2015. [Online]. Available:
12] <https://www.mysql.com/products/cluster/>. [Accessed 10
February 2016].

BIODATA PENULIS



Muhammad Nicko Rahmadano lahir di Jakarta tanggal 22 November 1994. Anak pertama dari tiga bersaudara ini menempuh pendidikan mulai dari SD Islam Al-Azhar 8 Kembangan (2000-2006), SMP Islam Al-Azhar 10 Kembangan (2006-2009), SMAN 112 Jakarta (2009-2012) dan saat ini sedang menempuh pendidikan Sarjana di Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya angkatan tahun 2012.

Penulis dalam menyelesaikan pendidikan S1 mengambil bidang minat Manajemen Informasi (*Information Management*) dan memiliki ketertarikan di bidang *Enterprise Resource Planning*

Terlibat aktif dalam organisasi kemahasiswaan serta kepanitiaan selama perkuliahan, antara lain staff Ristek dan Teknologi di Himpunan Mahasiswa Teknik Computer-Informatika ITS 2013/2014, Staff Keamanan dan Perizinan acara Schematics 2013, Staff REEVA acara Schematics 2014.

Untuk komunikasi dapat dihubungi melalui surel: nicko.rahmadano12@gmail.com.