



**TUGAS AKHIR - KI141502**

**RANCANG BANGUN APLIKASI ENTERPRISE  
RESOURCE PLANNING PADA MODUL ASSET  
MANAGEMENT BERORIENTASI MULTY-  
TENANCY DENGAN SISTEM DATABASE  
TERDISTRIBUSI**

**MELINDA TRESNAWATI  
NRP 5112 100 025**

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

**Dosen Pembimbing II  
Prof. Drs. Ec. Ir. Rianarto 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 ASSET MANAGEMENT  
WITH MULTITENANT ORIENTED USING  
DISTRIBUTED DATABASE**

**MELINDA TRESNAWATI  
NRP 5112 100 176**

**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 2015**

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

# RANCANG BANGUN APLIKASI ENTERPRISE RESOURCE PLANNING PADA MODUL ASSET MANAGEMENT BERORIENTASI MULTY- TENANCY DENGAN SISTEM DATABASE 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

**MELINDA TRESNAWATI**

NRP. 5112 100 025

Disetujui oleh Dosen Pembimbing Tugas Akhir

Sarwosri, S.Kom., MT.

NIP: 19760809 200112 2 001

(Pembimbing 1)

Prof. Drs. Ec. Ir. RIYANARTO SARNO

M.Sc., Ph.D.

NIP: 19590803 198601 1 001

(Pembimbing 2)

**SURABAYA**

**Juli 2016**

*[Halaman ini sengaja dikosongkan]*

# **Rancang Bangun Sistem *Enterprise Resource Planning* pada Modul *Asset Management* Berorientasi *Multi-Tenancy* dengan Menggunakan Sistem Basis Data Terdistribusi**

Nama Mahasiswa : Melinda Tresnawati  
NRP : 5112 100 025  
Jurusan : Teknik Informatika ITS  
Dosen Pembimbing 1 : Sarwosri, S.Kom., M.T.  
Dosen Pembimbing 2 : Prof. Drs.Ec. Ir. Rianarto Sarno M.Sc., Ph.D.

## **ABSTRAK**

*Enterprise Resource Planning* adalah sebuah sistem yang dirancang untuk mengelola sebuah perusahaan besar dalam mengkoordinasikan semua sumber daya, informasi, dan aktifitas yang diperlukan untuk proses bisnis lengkap. Salah satu modul yang telah dikerjakan pada tugas akhir ini yaitu *Asset Management*. *Asset Management* adalah proses pengelolaan asset (kekayaan) baik berwujud dan tidak berwujud yang memiliki nilai ekonomis, nilai komersial, dan nilai tukar mampu mendorong tercapainya tujuan dari individu dan organisasi. *Asset management* dapat mencatat semua aktivitas yang sedang berjalan pada pengelolaan asset. Kendala yang dihadapi perusahaan dalam mengelola asset adalah tidak adanya pencatatan yang jelas dimana asset berada dan bagaimana keadaan asset, baik yang sedang aktif digunakan maupun sudah tidak aktif (*Retired*)

Dalam tugas akhir ini, dibuat sebuah sistem untuk memudahkan pengguna dalam melakukan pengelolaan aset. Baik dalam perhitungan penyusutan nilai maupun pemeliharaan aset. Hal ini bertujuan untuk mengurangi kecurangan dalam perusahaan.

**Kata kunci:** *Asset Management, Enterprise Resource Planning, Depresiasi Aset, Asset Retirement, Tangible, Intangible*



# **Design and Implementation of Enterprise Resource Planning in Asset Management with Multitenant Oriented using Distributed Database**

Nama Mahasiswa : Melinda Tresnawati  
NRP : 5112 100 025  
Jurusan : Teknik Informatika ITS  
Dosen Pembimbing 1 : Sarwosri, S.Kom., M.T.  
Dosen Pembimbing 2 : Prof. Drs.Ec. Ir. Rivanarto Sarno M.Sc., Ph.D.

## **ABSTRACT**

*Enterprise Resource Planning is a system designed to manage a large company to coordinate all the resources, information, and activities needed to complete business processes. One of the modules that have been done in this thesis, namely Asset Management. Asset Management is management process of wealth, both tangible and intangible that has economic value, commercial value, and the exchange rate to encourage the attainment of the objectives of individuals and organizations. Asset management can record all activity that is running on asset management. Obstacles that companies face in managing assets is the absence of clear recording in which the asset is located and how the state of assets, which is being actively used or are no longer active (Retired)*

*In this final project we created a system to make it easier for users to do asset management. Both in the calculation of depreciation and maintenance of assets. It aims to reduce fraud within the company.*

**Keyword:** *Asset Management, Enterprise Resource Planning, Depresiasi Aset, Asset Retirement, Tangible, Intangible*

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas takhir yang berjudul:

***“RANCANG BANGUN SISTEM ENTERPRISE RESOURCE PLANNING PADA MODUL ASSET MANAGEMENT BERORIENTASIKAN MULTI-TENANCY DENGAN MENGGUNAKAN SISTEM BASIS DATA TERDISTRIBUSI”***

Melalui lembar ini, penulis hanya ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu dan keluarga yang selalu memberikan dukungan penuh untuk menyelesaikan tugas akhir ini.
2. Bapak Riyanarto Sarno beserta keluarga dan Ibu Sarwosri beserta keluarga selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Bapak dan ibu dosen Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan bagi penulis.
4. Seluruh staf dan karyawan FTIf ITS yang banyak memberikan kelancaran administrasi akademik kepada penulis.
5. Dwi Okta, Novita dan Mas Ichank sebagai penghibur penulis dikala sedih
6. Al Aji, Akbar, Vicky, Arya, Nicko, Afina, Yusi, Ashari selaku teman ERP seperjuangan.
7. Teman-teman seperjuangan anak didik Tugas Akhir Prof. Riyanarto Sarno yaitu Dipta, Baiquni, Aldrin, Lukman, Mamin, King, Maya, Kelly, dan Andrean
8. Teman-teman angkatan 2012 jurusan Teknik Informatika ITS yang telah menemani perjuangan selama 4 tahun ini atas saran, masukan, dan dukungan terhadap pengerjaan tugas akhir ini.

9. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu -persatu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juli 2016

Melinda Tresnawati

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK.....	vii
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xviii
DAFTAR TABEL.....	xxii
1. BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Permasalahan.....	3
1.3 Batasan Permasalahan.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Metodologi.....	4
1.7 Sistematika Penulisan.....	5
2. BAB II DASAR TEORI.....	7
2.1 Penelitian Terkait (ERP2013).....	7
2.2 Enterprise Resource Planning (ERP).....	7
2.3 Asset Management.....	8
2.4 Proses Bisnis Umum ERP.....	10
2.5 Metode Garis Lurus (Straight Line Method).....	12
2.6 Metode Saldo Menurun (Reducing Balance Method).....	13
2.7 Metode Penambahan Tahun (Sum of the Year Digits Depreciation).....	14
2.8 Metode Penurunan Aktivitas (Units of Activity Depreciation) 15	
2.9 Multitenancy.....	16
2.10 Basis Data Terdistribusi (Distributed Database).....	18
2.11 Data Model.....	19
2.12 Database Cluster.....	19
2.13 MySQL Cluster.....	20
2.14 Arsitektur MySQL Cluster.....	20
2.15 RBAC (Role Based Access Control).....	21
2.16 Distributed Database.....	22
2.17 Framework Yii.....	23
2.18 PHP.....	23
3. BAB III METODE PEMECAHAN MASALAH.....	25
3.1 Analisis.....	25
3.2 Identifikasi Pengguna.....	33

3.3	<i>Spesifikasi Kebutuhan Perangkat Lunak</i> .....	33
3.4	<i>Kasus Penggunaan</i> .....	34
3.4.1.1	Kasus Penggunaan Mengelola Penambahan Aset .....	36
3.4.1.2	Kasus Penggunaan Mengelola Pencatatan Aset .....	38
3.4.1.3	Kasus Penggunaan Mengelola Depresiasi Aset.....	41
3.4.1.4	Kasus Penggunaan Mengelola Pemeliharaan Aset....	43
3.4.1.5	Kasus Penggunaan Mengelola Penghapusan Aset.....	46
3.4.1.6	Kasus Penggunaan Mengelola Tipe Aset .....	49
3.4.1.7	Kasus Penggunaan Mengelola Kelompok Aset.....	51
3.4.1.8	Kasus Penggunaan Mengelola Metode Depresiasi Aset 53	
3.4.1.9	Kasus Penggunaan Melihat Data Penambahan Aset	55
3.4.1.10	Kasus Penggunaan Melihat Data Aset Aktif .....	56
3.4.1.11	Kasus Penggunaan Melihat Data Penyusutan Aset ...	57
3.4.1.12	Kasus Penggunaan Melihat Data Pemeliharaan Aset	59
3.4.1.13	Kasus Penggunaan Melihat Data Aset Tidak Aktif...	60
3.4.1.14	Kasus Penggunaan Melihat Report Tahunan.....	61
3.5	<i>Perancangan</i> .....	63
3.5.7.1	Perancangan antarmuka login.....	73
3.5.7.2	Perancangan antarmuka add user .....	74
3.5.8.1	Perancangan antarmuka index .....	74
3.5.8.2	Perancangan antarmuka create requisition .....	75
3.5.8.3	Perancangan antarmuka create asset active .....	76
3.5.8.4	Perancangan antarmuka create asset depreciation .....	77
3.5.8.5	Perancangan antarmuka create maintenance .....	78
3.5.8.6	Perancangan antarmuka create asset retirement .....	79
3.5.8.7	Perancangan antarmuka Report Tahunan .....	79
3.5.8.8	Perancangan antarmuka Melihat Data .....	80
4.	<b>BAB IV IMPLEMENTASI SISTEM</b> .....	81
4.1	<i>Lingkungan Pengembangan Sistem</i> .....	81
4.2	<i>Implementasi Basis Data Terdistribusi</i> .....	81
4.3	<i>Implementasi Role Base Access Control</i> .....	86
4.3.1.	Membuat Tabel Pengguna.....	86
4.3.2.	Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna	86
4.3.3.	Membuat 4 Modul Admin dan Konfigurasi Autentifikasi .....	87
4.3.4.	Membuat Model Tabel Autentifikasi, Controller dan View Pengguna .....	88
4.3.5.	Menambahkan Kode pada usercontroller .....	88
4.4	<i>Implementasi Multi-tenancy</i> .....	93

4.4.1.	Membuat Halaman Depan Tenant .....	93
4.4.2.	Menambahkan Basis Data untuk Tenant Baru .....	94
4.4.3.	Login Tenant .....	94
4.5	<i>Implementasi Program pada Modul Asset Management.....</i>	95
4.5.1.	Menambah Asset Type .....	95
4.5.2.	Menyunting Asset Type.....	96
4.5.3.	Menghapus Asset Type.....	96
4.5.4.	Menambah Depreciation Method .....	96
4.5.5.	Menyunting Depreciation Method .....	97
4.5.6.	Menghapus Depreciation Method.....	98
4.5.7.	Menambah Asset Group .....	98
4.5.8.	Menyunting Asset Group.....	98
4.5.9.	Menghapus Asset Group.....	99
4.5.10.	Menambah Asset Requisition .....	99
4.5.11.	Menyunting Asset Requisition.....	101
4.5.12.	Menghapus Asset Requisition .....	101
4.5.13.	Menambah Asset Active .....	101
4.5.14.	Menyunting Asset Active .....	102
4.5.15.	Menghapus Asset Active .....	103
4.5.16.	Menambah Asset Depreciation.....	104
4.5.17.	Menyunting Asset Depreciation .....	109
4.5.18.	Menghapus Asset Depreciation .....	109
4.5.19.	Menambah Asset Maintenance.....	110
4.5.20.	Menyunting Asset Maintenance .....	111
4.5.21.	Menghapus Asset Maintenance .....	111
4.5.22.	Menambah Asset Retirement.....	112
4.5.23.	Menyunting Asset Retirement .....	112
4.5.24.	Menghapus Asset Retirement .....	113
4.5.25.	Melihat Report .....	113
4.6	<i>Implementasi Antarmuka Pengguna.....</i>	114
5.	<b>BAB V PENGUJIAN DAN EVALUASI.....</b>	129
5.1	<i>Lingkungan Uji Coba.....</i>	129
5.2	<i>Skenario Pengujian .....</i>	129
5.2.1.	Pengujian Perencanaan Bisnis .....	129
5.2.1.1	Daftar Barang Jadi .....	130
5.2.1.2	Daftar Aset.....	130
5.2.1.3	Daftar Bahan Baku.....	131
5.2.1.4	Daftar Material.....	132
5.2.1.5	Daftar Material.....	133

5.2.2.	Pengujian Fungsionalitas Sistem.....	133
5.2.3.1	Pengujian Mengelola Tipe Aset .....	133
5.2.3.2	Pengujian Mengelola Metode Depresiasi .....	135
5.2.3.3	Pengujian Mengelola Kelompok Aset .....	136
5.2.3.4	Pengujian Mengelola Penambahan Aset .....	138
5.2.3.5	Pengujian Mengelola Pencatatan Aset .....	139
5.2.3.6	Pengujian Mengelola Penyusutan Nilai Aset .....	141
5.2.3.7	Pengujian Mengelola Pemeliharaan Aset .....	142
5.2.3.8	Pengujian Mengelola Aset Tidak Aktif .....	144
5.2.3.9	Pengujian Melihat Data Penambahan Aset.....	145
5.2.3.10	Pengujian Melihat Data Aset Aktif .....	146
5.2.3.11	Pengujian Melihat Data Penyusutan Nilai Aset.....	147
5.2.3.12	Pengujian Melihat Data Pemeliharaan Aset .....	147
5.2.3.13	Pengujian Melihat Data Aset Tidak Aktif .....	148
5.2.3.14	Pengujian Melihat Report Tahunan .....	149
5.2.3.	Pengujian Fungsionalitas Role Based Access Control (RBAC)	149
5.2.4.	Pengujian Fungsionalitas Basis Data Terdistribusi .....	153
5.3	<i>Implementasi Antar Muka</i> .....	157
5.3.1.	Evaluasi Pengujian Fungsionalitas Sistem .....	158
6.	BAB VI KESIMPULAN DAN SARAN .....	161
6.1	<i>Kesimpulan</i> .....	161
6.2	<i>Saran</i> .....	161
	DAFTAR PUSTAKA .....	163
	DAFTAR ISTILAH.....	165
	LAMPIRAN .....	167
	BIODATA PENULIS .....	173



*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Proses Bisnis Pengadaan Barang .....	11
Gambar 2.2 Proses Bisnis Produksi.....	11
Gambar 2.3 Proses Bisnis Penjualan .....	12
Gambar 2.4 Contoh Kasus menggunakan Sum of the Years Digits .....	15
Gambar 2.5 Separated Database.....	17
Gambar 2.6 Shared Database, Separated Schema .....	17
Gambar 2.7 Shared Database, Shared Schema.....	18
Gambar 2.8 Basis Data Terdistribusi.....	19
Gambar 3.1 Proses Bisnis Level 0.....	28
Gambar 3.2 Proses Bisnis Level 1.....	29
Gambar 3.3 Proses Bisnis Level 2.....	30
Gambar 3.4 Diagram Bisnis Proses Level 3 (bag 1) .....	31
Gambar 3.5 Diagram Bisnis Proses Level 3 (bag 2) .....	32
Gambar 3.6 Diagram Kasus Penggunaan .....	35
Gambar 3.7 Diagram aktifitas mengelola penambahan aset .....	38
Gambar 3.8 Diagram aktifitas kasus penggunaan mengelola pencatatan aset.....	40
Gambar 3.9 Diagram aktifitas depresiasi aset .....	42
Gambar 3.10 diagram aktifitas mengelola pemeliharaan aset .....	45
Gambar 3.11 diagram aktifitas mengelola penghapusan aset.....	48
Gambar 3.12 diagram aktifitas mengelola tipe aset .....	50
Gambar 3.13 Diagram aktifitas mengelola tipe aset.....	52
Gambar 3.14 Diagram aktifitas mengelola metode depresiasi aset .....	54
Gambar 3.15 Diagram aktifitas melihat data penambahan aset ..	56
Gambar 3.16 Diagram aktifitas melihat asset aktif .....	57
Gambar 3.17 Diagram aktifitas melihat penyusutan aset .....	58
Gambar 3.18 Diagram aktifitas melihat data pemeliharaan aset .	60
Gambar 3.19 Diagram aktifitas melihat data aset tidak aktif .....	61
Gambar 3.20 Diagram aktifitas melihat report tahunan .....	62
Gambar 3.21 Alur Proses Perhitungan .....	64
Gambar 3.22 Alur proses penurunan (general) .....	66
Gambar 3.23 Alur proses reducing balance method.....	67

Gambar 3.24 Alur proses Sum of the year .....	69
Gambar 3.25 Diagram Basis Data Terdistribusi.....	71
Gambar 3.26 Perancangan Multi-tenancy .....	72
Gambar 3.27 Perancangan antarmuka Login .....	73
Gambar 3.28 Perancangan antarmuka add user.....	74
Gambar 3.29 Perancangan antarmuka index .....	75
Gambar 3.30 perancangan antarmuka create requisition.....	75
Gambar 3.31 Perancangan antarmuka create asset active .....	76
Gambar 3.32 perancangan antarmuka create asset depreciation ..	77
Gambar 3.33 perancangan antarmuka create asset depreciation ..	78
Gambar 3.34 perancangan antarmuka create asset retirement.....	79
Gambar 3.35 Perancangan antarmuka report tahunan.....	80
Gambar 3.36 perancangan antarmuka melihat data.....	80
Gambar 4.1 Menambah Asset Type .....	115
Gambar 4.2 Menambah Asset Type .....	115
Gambar 4.3 Mengapus Asset type.....	116
Gambar 4.4 menambah depreciation method .....	116
Gambar 4.5 Menyunting depreciation method .....	117
Gambar 4.6 menghapus depreciation method .....	117
Gambar 4.7 menambah asset group.....	118
Gambar 4.8 menyunting asset group .....	118
Gambar 4.9 menghapus asset group .....	119
Gambar 4.10 menambah asset requisition .....	119
Gambar 4.11 Menyunting asset requisition .....	120
Gambar 4.12 menghapus asset requisition .....	120
Gambar 4.13 menambah asset active .....	121
Gambar 4.14 menyunting asset active.....	121
Gambar 4.15 menghapus asset active.....	122
Gambar 4.16 menambah asset depreciation .....	122
Gambar 4.17 menyunting asset depreciation.....	123
Gambar 4.18 menghapus asset depreciation.....	123
Gambar 4.19 menambah asset maintenance.....	124
Gambar 4.20 menyunting asset maintenance .....	124
Gambar 4.21 menghapus asset maintenance .....	125
Gambar 4.22 menambah asset retirement.....	125

Gambar 4.23 menyunting asset retirement.....	126
Gambar 4.24 menghapus asset retirement.....	126
Gambar 4.25 melihat report.....	127
Gambar 5.1 Login RBAC.....	151
Gambar 5.2. Tampilan awal setelah login berhasil dilakukan...	152
Gambar 5.3. Admin menambahkan user baru .....	152
Gambar 5.4. Proses user admin menyunting data user.....	153
Gambar 5.5. Proses user admin menghapus data user.....	153
Gambar 5.6. Pengujian fitur replikasi pada sistem.....	154
Gambar 5.7. Pengujian fitur replikasi pada database server 1...	154
Gambar 5.8. Pengujian fitur replikasi pada database server 2...	155
Gambar 5.9. Pengujian fitur high-availability pada sistem .....	155
Gambar 5.10. Pengecekan availability pada node server .....	156
Gambar 5.11. Pengecekan database pada node 1 .....	156
Gambar 5.12. Pengecekan availability pada node server .....	157
Gambar 5.13. Mematikan database server pada node 2 .....	157

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 2.1 Golongan Penyusutan Aktiva Tetap.....	9
Tabel 3.1 Kekurangan dan kelebihan pada Odoo, Adempiere, dan InoERP .....	25
Tabel 3.2 Daftar Kebutuhan Fungsional Sistem.....	33
Tabel 3.3 Keterangan Kode Kasus Penggunaan.....	35
Tabel 3.4 Kasus Penggunaan Mengelola Penambahan Aset.....	36
Tabel 3.5 Kasus Penggunaan Mengelola Pencatatan Aset.....	39
Tabel 3.6 Kasus Penggunaan Mengelola Depresiasi Aset .....	41
Tabel 3.7 kasus penggunaan mengelola pemeliharaan aset .....	43
Tabel 3.8 kasus penggunaan mengelola penghapusan aset .....	46
Tabel 3.9 kasus penggunaan mengelola tipe aset.....	49
Tabel 3.10 Kasus penggunaan mengelola kelompok aset.....	51
Tabel 3.11 Kasus penggunaan mengelola metode depresiasi aset .....	53
Tabel 3.12 Kasus Penggunaan Melihat Data Penambahan Aset .	55
Tabel 3.13 Kasus penggunaan melihat data aset aktif.....	56
Tabel 3.14 Kasus penggunaan melihat data penyusutan aset....	57
Tabel 3.15 Kasus penggunaan melihat data pemeliharaan aset...	59
Tabel 3.16 Kasus penggunaan melihat data aset tidak aktif.....	60
Tabel 3.17 Kasus penggunaan melihat report tahunan.....	62
Tabel 3.18 Detail Aset.....	64
Tabel 3.19 Detail Aset.....	66
Tabel 3.20 Detail Aset.....	68
Tabel 3.21 Detail Aset.....	69
Tabel 5.1 Daftar Barang Jadi.....	130
Tabel 5.2 Daftar Aset .....	130
Tabel 5.3 Daftar Bahan Baku .....	131
Tabel 5.4 Daftar Material .....	132
Tabel 5.5 Perencanaan Keuangan Perusahaan .....	133
Tabel 5.6 pengujian mengelola tipe aset .....	134
Tabel 5.7 pengujian mengelola metode depresiasi.....	135
Tabel 5.8 pengujian mengelola kelompok aset .....	137
Tabel 5.9 pengujian mengelola penambahan aset .....	138
Tabel 5.10 pengujian mengelola pencatatan aset .....	140

Tabel 5.11 pengujian mengelola penyusutan nilai aset .....	141
Tabel 5.12 pengujian mengelola pemeliharaan aset .....	142
Tabel 5.13 pengujian mengelola aset tidak aktif .....	144
Tabel 5.14 pengujian melihat data penambahan aset .....	145
Tabel 5.15 pengujian melihat data aset aktif .....	146
Tabel 5.16 pengujian melihat data penyusutan nilai aset .....	147
Tabel 5.17 melihat data pemeliharaan aset.....	148
Tabel 5.18 pengujian melihat data aset tidak aktif .....	148
Tabel 5.19 pengujian melihat report tahunan .....	149
Tabel 5.20. Pengujian fitur Role Based Access Control (RBAC) .....	150
Tabel 5.21 evaluasi pengujian .....	158





# **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**

Kondisi dunia bisnis saat ini telah berkembang menjadi semakin kompleks, semakin kompetitif, bergerak dengan cepat serta semakin sulit untuk diprediksi. Agar dapat bersaing dan sukses, perusahaan perlu memadukan bisnis dan sumber daya IT yang dimiliki agar dapat secara fleksibel mengakomodasi adanya perubahan untuk kemudian dilakukan adaptasi terhadap perubahan tersebut secara cepat dan tepat.

Dari pembahasan diatas dapat disimpulkan bahwa kebutuhan akan aplikasi *Enterprise Resource Planning* (ERP) dalam perusahaan merupakan kebutuhan yang perlu agar perusahaan dapat memadukan bisnis dan sumber daya. 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*, dan lainnya. Pada tugas akhir ini, penulis akan membangun paket atau modul aplikasi *Accounting*, khususnya pada sub-domain *Asset Management*.

Sistem ERP yang akan dibangun ini, khususnya untuk modul *Accounting* sebagian besar berkiblat pada aplikasi ERP nomor satu di dunia yaitu SAP, di dalam SAP, modul *Accounting* terdapat dalam satu paket, yaitu *SAP FI-CO (Financial and Controlling)*.

*FI-Financial Accounting*, Software bisnis SAP ditujukan untuk menyediakan pengukuran secara kontinu terhadap profitabilitas perusahaan. Modul *FI* juga mengukur kinerja keuangan perusahaan, berdasarkan pada data transaksi internal maupun eksternal. Modul *FI* menyediakan dokumen keuangan yang mampu melacak (mengaudit) setiap angka yang terdapat dalam suatu laporan keuangan hingga ke data transaksi awalnya.

*CO-Controlling*, Fungsi dari modul *CO* adalah untuk mendukung empat kegiatan operasional yaitu pengendalian capital investment, pengendalian aktivitas keuangan perusahaan, memonitor dan merencanakan pembayaran, pengendalian pendanaan terhadap procurement, pengadaan dan penggunaan dana di setiap area, serta pengendalian biaya dan profit berdasarkan semua aktivitas perusahaan.

Dengan berkaca pada modul *SAP FI-CO* dan dengan sedikit penyesuaian sesuai kebutuhan, maka penulis akan membangun sub-domain *Account Payable*, *Account Receivable* dan *Fixed Asset* pada domain *ERP Accounting*

*Asset management* adalah proses pengelolaan asset (kekayaan) baik berwujud dan tidak berwujud yang memiliki nilai ekonomis, nilai komersial, dan nilai tukar mampu mendorong tercapainya tujuan dari individu dan organisasi. *Asset management* dapat mencatat semua aktivitas yang sedang berjalan pada pengelolaan asset. Kendala yang dihadapi perusahaan dalam mengelola asset adalah tidak adanya pencatatan yang jelas dimana asset berada dan bagaimana keadaan asset, baik yang sedang aktif digunakan maupun sudah tidak aktif (*Retired*). Hal ini menyebabkan banyaknya kecurangan pada perusahaan yang berasal dari pihak internal. Untuk itu dibutuhkan Asset Management.

## 1.2 Rumusan Permasalahan

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

1. Bagaimana membuat aplikasi *ERP* untuk sub-domain *Asset Management* suatu perusahaan untuk menggantikan proses pengolahan data manual dan sebagai penyedia informasi terkait *Accounting* sebagai usaha peningkatan kinerja perusahaan?
2. Bagaimana proses perhitungan penurunan aset pada setiap metode?
3. Bagaimana mengimplementasikan multitenansi pada modul *Asset Management*?
4. Bagaimana mengatasi kelemahan *separated database* pada pengembangan sebelumnya dalam hal *availability* untuk arsitektur yang *multitenancy*?

## 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.
3. *Platform* yang digunakan adalah Yii2 dengan teknologi :
  - a. *Eloquent*
  - b. *Modularity*
  - c. *Web Service*
4. Aplikasi yang dikembangkan merupakan aplikasi *web*.
5. Sistem ERP yang dibangun dalam aspek Multitenancy
6. Modul yang akan diintegrasikan adalah *Asset Management* dengan beberapa sub-modul, diantaranya adalah :
  - a. *Master Data Asset*
  - b. *Asset Requisition*
  - c. *Asset Active*
  - d. *Asset Depreciation*
  - e. *Asset Maintenance*

- f. *Asset Retirement*
- g. *Report*

#### **1.4 Tujuan**

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Membuat aplikasi ERP sub-domain *Asset Management* pada perusahaan untuk menggantikan proses pengolahan data manual dan sebagai penyedia informasi terkait *Accounting* sebagai usaha peningkatan kinerja perusahaan.
2. Menemukan proses model perhitungan penurunan harga asset.
3. Menemukan perhitungan penurunan harga asset yang tepat untuk masing-masing asset pada perusahaan.
4. Mengimplementasikan multitenancy pada modul Asset Management
5. Mengatasi kelemahan *separated database* pada pengembangan sebelumnya dalam hal *availability* untuk arsitektur yang *multitenancy*

#### **1.5 Manfaat**

Manfaat dari hasil tugas akhir ini adalah memberikan kontribusi untuk membangun aplikasi ERP (*Enterprise Resource Planning*) yang berguna untuk kalangan UKM di Indonesia demi meningkatkan pendapatan mereka. Dengan adanya aplikasi ini, diharapkan aplikasi ERP lebih berkembang untuk penggunaan skala UKM.

#### **1.6 Metodologi**

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

1. Studi literatur  
Dalam pembuatan Tugas Akhir ini telah dipelajari tentang hal-hal yang dibutuhkan sebagai ilmu penunjang dalam penyelesaiannya. Pertama adalah tentang metode penurunan harga aset.

2. **Penentuan Metode**  
Pada tahap ini penulis menjabarkan beberapa metode yang dapat digunakan untuk proses penurunan harga barang atau depresiasi
3. **Analisis dan Perancangan Sistem**
4. **Implementasi**  
Pada tahap ini dilakukan pembuatan elemen perangkat lunak yang merupakan implementasi dari rancangan pada proses
5. **Pengujian dan evaluasi**  
Pada tahap ini dilakukan pengujian terhadap elemen perangkat lunak dengan menggunakan data uji yang telah dipersiapkan. Pengujian dan evaluasi perangkat lunak dilakukan untuk mengevaluasi jalannya perangkat lunak, mengevaluasi fitur utama, mengevaluasi fitur-fitur tambahan, mencari kesalahan yang timbul pada saat perangkat lunak aktif, dan mengadakan perbaikan jika ada kekurangan.
6. **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**

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.

## **BAB II**

### **DASAR TEORI**

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

#### **2.1 Penelitian Terkait (ERP2013)**

Tugas akhir ini merupakan pengembangan riset dari tugas akhir sebelumnya. Pada tugas akhir yang dahulu, sudah terdapat aplikasi *Accounting* yang berdiri sendiri-sendiri setiap domainnya, yaitu fungsional domain *Account Payable*, *Account Receivable*, dan *Fixed Asset*.

Pada tugas akhir yang penulis kembangkan saat ini merupakan pengembangan teknologi aplikasi *Fixed Asset*. Dengan penggunaan teknologi workflow maka ERP yang dibangun kali ini akan mempermudah dalam masalah fleksibilitas dari fungsionalitas kebutuhan sistem. Dengan kedinamisan aplikasi yang dibangun maka sangat berpengaruh pada kecepatan performa aplikasi dan juga mempermudah para pelaku bisnis yang berlatar belakang non-OT untuk bisa merubah proses bisnis sesuai kehendaknya.

#### **2.2 Enterprise Resource Planning (ERP)**

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

## **2.3 Asset Management**

Setiap perusahaan pasti memiliki aset (aktiva) yang digunakan untuk mendukung kegiatan usahanya. Aktiva ini dibagi menjadi dua, yaitu aktiva lancar dan aktiva tetap. Aktiva tetap (*Fixed Asset*) merupakan istilah yang digunakan untuk menyatakan sebuah aset atau properti perusahaan yang tidak dapat dengan mudah dikonversikan menjadi uang tunai. Aktiva tetap merupakan aktiva jangka panjang atau dapat dikatakan relatif permanen. Aktiva tersebut dimiliki dan digunakan oleh perusahaan dengan tanpa ada maksud untuk menjualkannya dalam sebuah proses penjualan. Aktiva tetap dapat dibagi menjadi dua, yaitu *tangible* (berwujud) seperti tanah, pabrik, peralatan, properti dan lain sebagainya, serta *intangible* (tidak berwujud) seperti merk dagang, hak paten, *franchise* dan lain sebagainya.

### **2.3.1 Asset Requisition**

Proses bisnis *Asset Active* bertanggung jawab dalam proses pencatatan penambahan aktiva tetap perusahaan. Pencatatan dimulai sejak proses penerimaan barang dari domain *Inventory* hingga nantinya akan disimpan di gedung. Proses ini memiliki satu buah sub bisnis proses yang bernama *Asset Requisition*.

*Asset Requisition* adalah proses permintaan pengadaan aktiva tetap. Didalam proses ini, staff *Asset Management* dapat meminta pengadaan barang, apabila aset perusahaan telah mengalami *retired*, *lost*, atau dirasa kurang maksimal dalam melakukan tugasnya.

### **2.3.2 Asset Depreciation**

Depresiasi adalah proses penyusutan nilai harga dari aktiva tetap yang dilakukan ke harta yang berstatus “Active” (umur aset masih belum habis) sesuai dengan tipenya yang telah ditentukan pada tabel di atas. Proses ini dilakukan per bulan terhadap setiap aset yang dimiliki oleh perusahaan.

Sesuai dengan undang-undang perpajakan di Indonesia yaitu UU No 36 Tahun 2008 Pasal 11, aktiva tetap



dikelompokkan menjadi dua yaitu bangunan dan non-bangunan dengan kriteria seperti terlihat pada Tabel 2.1.

Tabel 2.1 Golongan Penyusutan Aktiva Tetap

Kelompok Harta Berwujud	Masa Manfaat	Tarif Penyusutan Sebagaimana Dimaksud Dalam UU	
<b>Non Bangunan:</b>			
Kelompok 1	4 tahun	25%	50%
Kelompok 2	8 tahun	12,5%	25%
Kelompok 3	16 tahun	6,25%	12,5%
Kelompok 4	20 tahun	5%	10%
<b>Bangunan:</b>			
<b>Permanen</b>	20 tahun	5%	
<b>Non Permanen</b>	10 tahun	10%	

### 2.3.3 Asset Maintenance

*Asset Maintenance* adalah proses pencatatan data pemeliharaan aktiva tetap. Proses ini akan melakukan perubahan status terhadap aktiva tetap sesuai pemeliharaan yang akan dilakukan dan kemudian akan dicatat besar harga pemeliharaan tersebut yang nantinya akan digunakan oleh domain Cash and Bank untuk dilakukannya pembayaran.

Jenis perawatan pada aplikasi yang penulis bangun meliputi dua jenis perawatan, yaitu proses maintenance, yang merupakan proses perawatan yang kurang dari satu hari serta proses reparation, atau proses perbaikan aktiva tetap yang diakibatkan oleh suatu hal sehingga harus diperbaiki selama beberapa hari dan dimungkinkan untuk dibawa keluar perusahaan. Ketika proses reparation ini selesai dilakukan, maka staff fixed

asset akan melakukan proses pencatatan aktiva tetap setelah dilakukan reparasi untuk merubah status dan kondisi aktiva tetap serta mencatat tanggal selesai proses tersebut sesuai dengan tanggal diterimanya.

Setelah proses maintenance ini dilakukan, maka akan dilakukan proses apresiasi, yaitu penambahan nilai aktiva tetap perusahaan.

### **2.3.4 Asset Retirement**

*Asset Retirement* adalah proses penghapusan aktiva tetap perusahaan yang dikarenakan aktiva tetap tersebut hilang, rusak, atau tidak dapat digunakan kembali. Proses ini terkadang didahului oleh proses *stock take*, tetapi perusahaan dapat melakukan perusahaan ini kapan pun juga. Ketika sebuah aktiva tetap perusahaan telah dihapus, maka aktiva tetap tersebut tidak akan dilakukan proses depresiasi dan tentunya secara otomatis aset perusahaan berkurang.

## **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/fulfillment*) [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(*fullfilment*)

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 Metode Garis Lurus (*Straight Line Method*)

Metode garis lurus menghasilkan jumlah beban penyusutan yang sama setiap periodenya sepanjang umur manfaat suatu aktiva tetap. Metode garis lurus sangat sederhana dan digunakan secara luas. Metode ini menciptakan transfer biaya yang wajar ke beban periodik jika pemanfaatan aktiva dan pendapatan yang terkait dengan pemakaian sama dari periode ke periodenya. Untuk menghitung penurunan aset dengan menggunakan metode garis lurus dapat digunakan rumus:

$$\text{Depreciation per annum} = (\text{cost} - \text{residual value}) / \text{useful value}$$

Keterangan:

*Depreciation per annum* : Nilai penyusutan dari aktiva tetap.

- Cost* : Harga aktiva tetap saat pertama kali melakukan pembelian (harga asli).
- Residual value* : Harga yang ditentukan perusahaan untuk menyelamatkan harga asli setelah melalui proses depresiasi. Dalam Tugas Akhir ini, memakai 20% dari harga asli sebagai *residual value*.
- Useful value* : Periode atau umur kebergunaan dari aktiva tetap.

## 2.6 Metode Saldo Menurun (*Reducing Balance Method*)

Metode saldo menurun menghasilkan beban periodik yang terus menurun sepanjang estimasi umur manfaat aktiva. Untuk menerapkan metode ini, tarif penyusutan garis lurus tahunan terlebih dulu digandakan. Untuk menghitung penurunan aset dengan menggunakan metode ini, dapat dilakukan dengan rumus berikut:

$$\text{Depreciation per annum} = (\text{net book value} - \text{residual value}) \times \text{rate}\%$$

Keterangan:

*Depreciation per annum* : Nilai penyusutan dari aktiva tetap

*Cost* : Harga aktiva tetap saat pertama kali melakukan pembelian (harga asli).

*Residual value* : Harga yang ditentukan perusahaan untuk menyelamatkan harga asli setelah melalui proses depresiasi. Dalam Tugas Akhir ini, memakai 20% dari harga asli sebagai *residual value*.

*Net Book Value* : Harga aset awal yang telah dikurangi dengan *Depreciation cost* atau *depreciation per annum*. Jika belum pernah dilakukan depresiasi maka *Net*

*book value* diambil dari harga awal atau *cost*.

*Rate %* : Depresiasi rating yang ditentukan oleh harga pasar.

## 2.7 Metode Penambahan Tahun (*Sum of the Year Digits Depreciation*)

Metode yang melibatkan perhitungan depresiasi berdasarkan penjumlahan dari beberapa tahun masa kebergunaan dari aset. Metode ini seperti metode *reducing balance*, yaitu sebuah teknik percepatan beban depresiasi yang mengalokasikan depresiasi lebih tinggi pada tahun-tahun awal dalam masa kebergunaan dari aset. Perhitungan pada metode ini dapat dilakukan dengan rumus:

$$\text{Depreciation expense} = (\text{un-depreciated useful life (step 3)} / \text{sum of the years digits (step 1)}) \times \text{depreciable amount (step 2)}$$

Keterangan:

*Depreciation expense* : Nilai penyusutan dari aktiva tetap.

*Undepreciated useful life (step 3)*: Umur kebergunaan aktiva tetap yang belum mengalami depresiasi

*Sum of the year digits (step 1)* : Penjumlahan umur kebergunaan aset.

*Depreciable amount (step 2)* : Harga asli aset (*cost*) dikurangi dengan *residual value*.

*Residual value* : Harga yang ditentukan perusahaan untuk menyelamatkan harga asli setelah melalui proses depresiasi. Dalam Tugas Akhir ini, memakai 20% dari

Cost

harga asli sebagai *residual value*.

: Harga aktiva tetap saat pertama kali melakukan pembelian (harga asli).

Step 1: Calculate the sum of the years digits			
Sum of the years' digits = 3 + 2 + 1 = 6			
Step 2: Calculate the depreciable amount			
Depreciable amount = \$100,000 - \$10,000 = \$90,000			
Step 3: Calculate the un-depreciated useful life			
	Year 1	Year 2	Year 3
Un- depreciated useful life (years)	3	2	1
Step 4: Calculate depreciation expense			
Year 1: Depreciation expense =	$\frac{3 \text{ (Step 3)}}{6 \text{ (Step 1)}}$	x \$90,000 (Step 2) =	\$45,000
Year 2: Depreciation expense =	$\frac{2 \text{ (Step 3)}}{6 \text{ (Step 1)}}$	x \$90,000 (Step 2) =	\$30,000
Year 3: Depreciation expense =	$\frac{1 \text{ (Step 3)}}{6 \text{ (Step 1)}}$	x \$90,000 (Step 2) =	\$15,000

Gambar 2.4 Contoh Kasus menggunakan Sum of the Years Digits

## 2.8 Metode Penurunan Aktivitas (*Units of Activity Depreciation*)

Metode ini dikenal dengan metode Units of Activity dan Units of Usage Depresiasi. Metode ini menghitung depresiasi berdasarkan produksi atau penggunaan yang diharapkan. Untuk menggunakan metode ini bisa dilakukan dengan rumus:

1.  $Depreciation \text{ per annum} = (cost - residual \text{ value}) / useful \text{ life.}$
2.  $Stage \text{ of Completion } \% = (value \text{ of work certified as}$

$$\frac{\text{complete/total expected production or usage}}{\text{}} \times 100$$

Keterangan:

- Depreciation per annum* : Nilai penyusutan dari aktiva tetap.
- Cost* : Harga aktiva tetap saat pertama kali melakukan pembelian (harga asli).
- Residual value* : Harga yang ditentukan perusahaan untuk menyelamatkan harga asli setelah melalui proses depresiasi. Dalam Tugas Akhir ini, memakai 20% dari harga asli sebagai *residual value*.
- Useful value* : Periode atau umur kebergunaan dari aktiva tetap.

## 2.9 Multitenancy

*Multitenancy* mengacu pada arsitektur perangkat lunak di mana salah satu contoh dari perangkat lunak yang berjalan pada server dan menyajikan beberapa *tenant*. *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



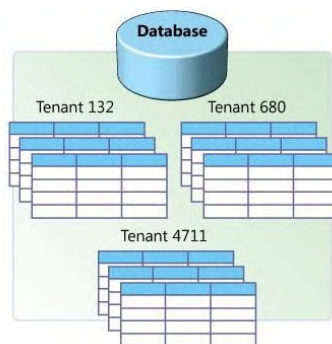
data aplikasi yang digunakan. Tetapi memerlukan biaya yang cukup tinggi untuk menjaga peralatan *server* dan juga *back up* data dari setiap *tenant*. Model arsitektur *Separated Database* ditunjukkan pada Gambar 2.5.



Gambar 2.5 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. 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 2.6.



Gambar 2.6 Shared Database, Separated Schema

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

Shared Database, Shared Schema adalah data setiap *tenant* disimpan pada satu *database* dan satu *schema*. 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 3.

The diagram illustrates a shared database schema where multiple tenants' tables overlap. The visible tables and their data are as follows:

TenantID	CustName	Address
4	TenantID	ProductID
1	4	TenantID
6	1	4711
4	6	132
4	4	680
		4711

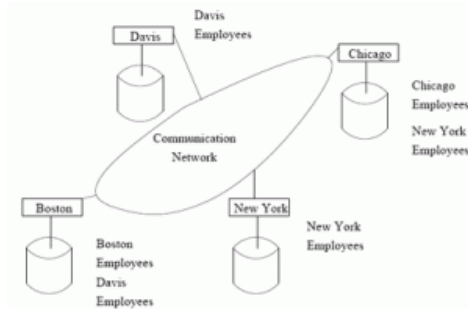
  

ProductID	ProductName
324965	2006-02-21
115468	2006-04-08
654109	2006-03-27
324956	2006-02-23

Gambar 2.7 Shared Database, Shared Schema

**2.10 Basis Data Terdistribusi (*Distributed Database*)**

Basis data terdistribusi (*distributed database*) adalah suatu basis data yang berada di bawah kendali sistem manajemen basis data (DBMS) terpusat dengan piranti penyimpanan (*storage devices*) yang terpisah-pisah satu dari yang lainnya. Tempat penyimpanan ini dapat berada di satu lokasi yang secara fisik berdekatan atau terpisah oleh jarak yang jauh dan terhubung melalui jaringan internet. Penggunaan basis data terdistribusi dapat dilakukan di server internet, intranet atau ektranet kantor, atau di jaringan perusahaan. Gambaran ilustrasi mengenai basis data terdistribusi terdapat pada gambar 2.8



Gambar 2.8 Basis Data Terdistribusi

### 2.11 *Data Model*

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

### 2.12 *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.

### **2.13 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.

### **2.14 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.

### **2.15 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). Fitur yang disediakan oleh RBAC menjadi daya tarik bagi para perusahaan yang menerapkan teknologi informasi, pertama RBAC yang mengorganisir subjek dan role secara alamiah sesuai dengan struktur yang diterapkan pada perusahaan tersebut (Ferraiolo et al, 1997). Hubungan antara hak akses dengan para pelaku yang diterapkan pada perusahaan tersebut, 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, hal

ini adalah dasar perbedaan antara RBAC dan DAC (Ferraiolo et al, 1992).

Dalam menangani kebijakan keamanan, RBAC adalah fondasi utama yang harus didefinisikan dengan baik sebelum Security Constraint Specifiers (SCS) atau disebut juga sebagai petugas yang menentukan kendala dalam menjaga keamanan.

SCS ini akan menentukan keterbatasan yang terjadi seperti kendala dalam penerapan sistem. RBAC memiliki banyak variasi yang akan dilakukan dengan menggambarkan bagaimana menentukan keterbatasan yang terjadi tanpa kehilangan ciri khasnya. Struktur RBAC meliputi roles, permission, user, dan session (Chen et al, 1996).

Dalam RBAC, role didefinisikan sebagai suatu gagasan yang merupakan dasar dari kebijakan kontrol akses (Sandhu et al, 1999). Pendefinisian role masih menjadi perdebatan sehingga diperlukan penjelasan konsep role itu tersendiri, sehingga untuk melakukan kolaborasi dengan kontrol akses masih disesuaikan dengan kebutuhan. Dalam ilmu perilaku, role didefinisikan sebagai pola yang ditentukan sesuai perilaku seseorang dalam situasi tertentu berdasarkan posisi orang tersebut (Hawkins et al, 1983). Dengan kata lain pendefinisian role adalah tugas yang didapatkan oleh seseorang sesuai dengan tanggung jawabnya. Namun demikian, sangat sulit untuk menggambarkan pengertian role karena bahasa yang diterapkan masih bersifat ambigu (Zhu, 2003)

## **2.16 Distributed Database**

Distributed Database adalah suatu kesatuan dari elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk mendistribusikan data, informasi proses, objek dan layanan dari dan kepada user yang terkait didalamnya. Yang melatarbelakangi penggunaan sistem Distributed Database adalah karena data yang tersimpan semakin besar, pemrosesan semakin kompleks, data dapat diakses dengan menggunakan teknologi. Distributed Database ini sendiri bukanlah merupakan sekumpulan file, melainkan sekumpulan Database yang tersebar di beberapa

tempat. Dan Distributed Database tidaklah sama dengan File Sharing.

### **2.17 Framework Yii**

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

### **2.18 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. 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.

*[Halaman ini sengaja dikosongkan]*



## BAB III METODE PEMECAHAN MASALAH

Pada 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: domain permasalahan, deskripsi umum sistem, dan kasus penggunaan sistem. Berikut pembahasan bagian-bagian tahap analisis.

#### 3.1.1 Analisa 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 sub-bab 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 Kekurangan dan kelebihan pada Odoo, Adempiere, dan InoERP

No	Nama Modul	Odoo	InoERP	Adempiere	ERP 2013
1	<i>Assets</i>	<b>Kontra:</b>	<b>Kontra :</b>	<b>Pro :</b>	1.

No	Nama Modul	Odoo	InoERP	Adempiere	ERP 2013
	<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  <b>Kontra:</b> 1. <i>Maintenance</i> tidak berbasis pada waktu produksi, sistem tidak memberikan <i>warning</i> apabila mesin butuh di <i>maintenance</i>	<i>Maintenance</i> tidak mengikuti waktu produksi

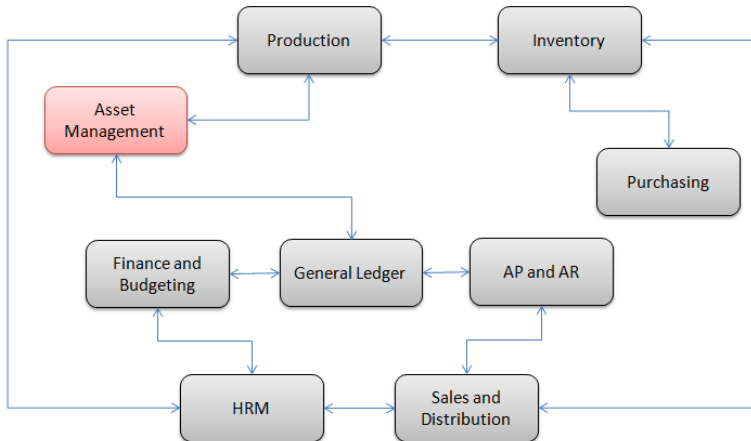
Tabel 3.1 menunjukkan bahwa modul *Asset Management* pada Odoo, InoERP, Adempiere, dan ERP 2013 tidak mendapatkan waktu penggunaan mesin dan jarak penggunaan

kendaraan sehingga jadwal *maintenance* dapat dilakukan sewaktu-waktu atau tidak berdasarkan waktu dan jarak penggunaan. Hal tersebut juga berdampak pada penurunan harga barang, karena proses perhitungan penurunan harga barang tidak sesuai dengan aktivitas yang dilakukan. Modul *Asset Management* pada aplikasi ERP yang dijelaskan di tabel 3.1 menggunakan metode *straight line* untuk menghitung penurunan harga barang pada semua tipe aset sehingga hasil perhitungan tidak akurat. Dibutuhkan metode lain untuk menghitung penurunan harga aset pada tipe dan karakteristik tertentu, khususnya aset bergerak atau yang dipakai terus menerus seperti mesin dan kendaraan.

Berdasarkan sumber [1] dan [2] metode terbaik untuk menghitung penurunan harga (depresiasi) adalah yang dapat menghitung secara rinci pergerakan aktivitas aset, sehingga dipilih metode *Unit of Activity*. Metode tersebut merupakan metode yang cukup unik karena perencanaan umur kebermanfaatannya suatu barang dinyatakan dalam total unit pemakaian yang diharapkan atau aktivitas total aset selama hidupnya. Biaya aset tersebut kemudian dialokasikan ke periode akuntansi berdasarkan penggunaan aset, unit yang diproduksi, aktivitas, dll. Tahun parsial tidak relevan ketika menggunakan metode penyusutan ini. Metode selain *Unit of Activity*, seperti *Straight Line*, *Reducing Balance* dan *Sum of The Year* menggunakan angka parsial sebagai perencanaan umur suatu barang dan mengalokasikan biaya melalui tahun-tahun yang telah digunakan. Metode tersebut dapat digunakan untuk aset yang tidak dipakai terus menerus, atau yang tidak bergerak, seperti tanah, bangunan, dan peralatan Kantor.

Dari kekurangan yang dimiliki oleh setiap ERP maka perlu dikembangkan sebuah aplikasi ERP yang sesuai dengan kebutuhan pada *business plan*. Terdapat dua jenis proses bisnis yang digunakan, yakni Make-To-Order (MTO) dan Make-To-Stock (MTS). MTO adalah sebuah proses bisnis dimana produksi didasarkan pada permintaan. MTS adalah sebuah proses bisnis dimana produksi didasarkan oleh *sales-forecasting*. Berikut

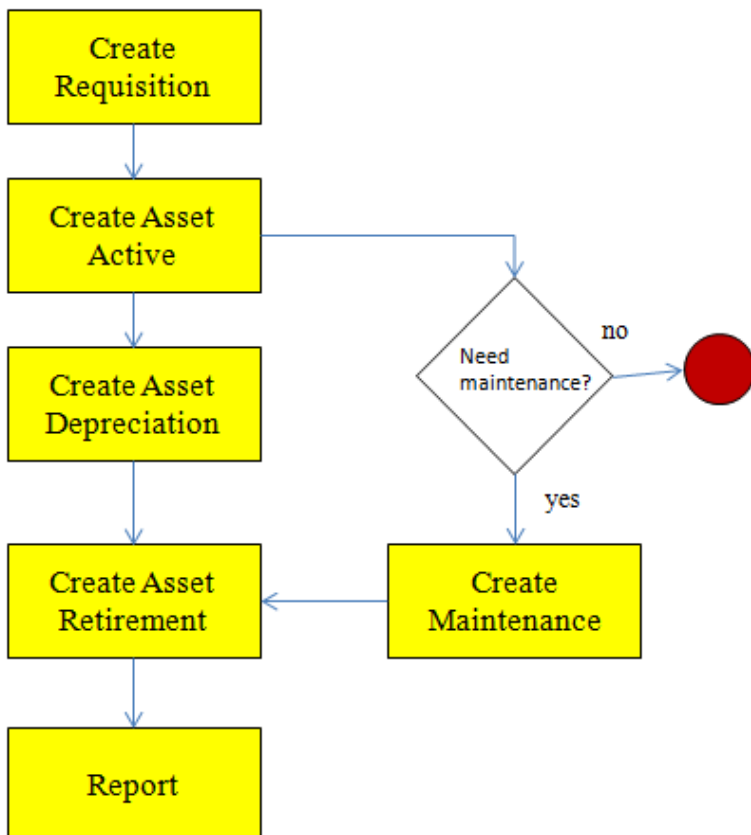
proses bisnis yang dirancang sesuai dengan kebutuhan *business plan* dimana proses bisnis ini telah mengakomodasi kekurangan Odoo, Adempiere, InoERP dan ERP 2013



Gambar 3.1 Proses Bisnis Level 0

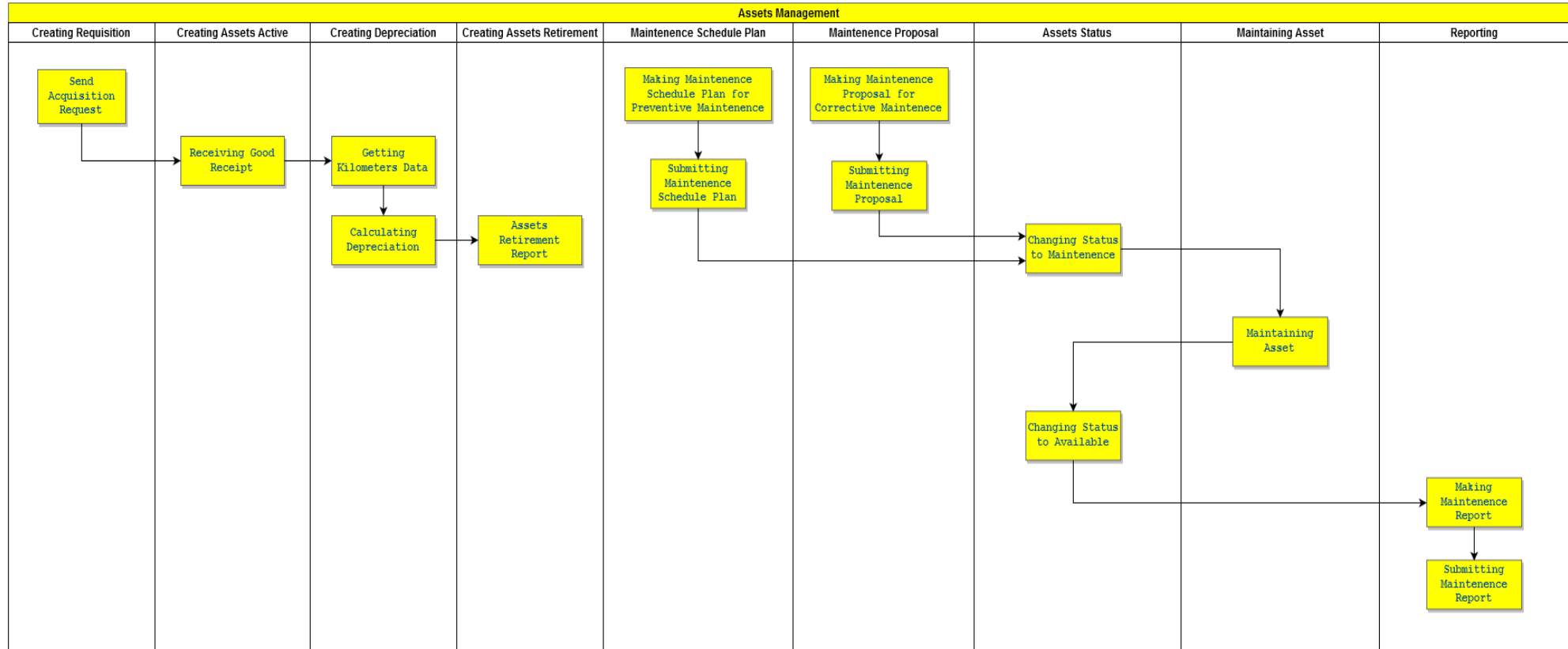
Pada Gambar 3.1, modul-modul keuangan (*AP/AR*, *Finance*, dan *General Ledger*) merupakan sentral dari sistem ERP. Sedangkan, modul *Asset Management* berinteraksi dengan modul *Production* dan *Sales and Distribution*

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

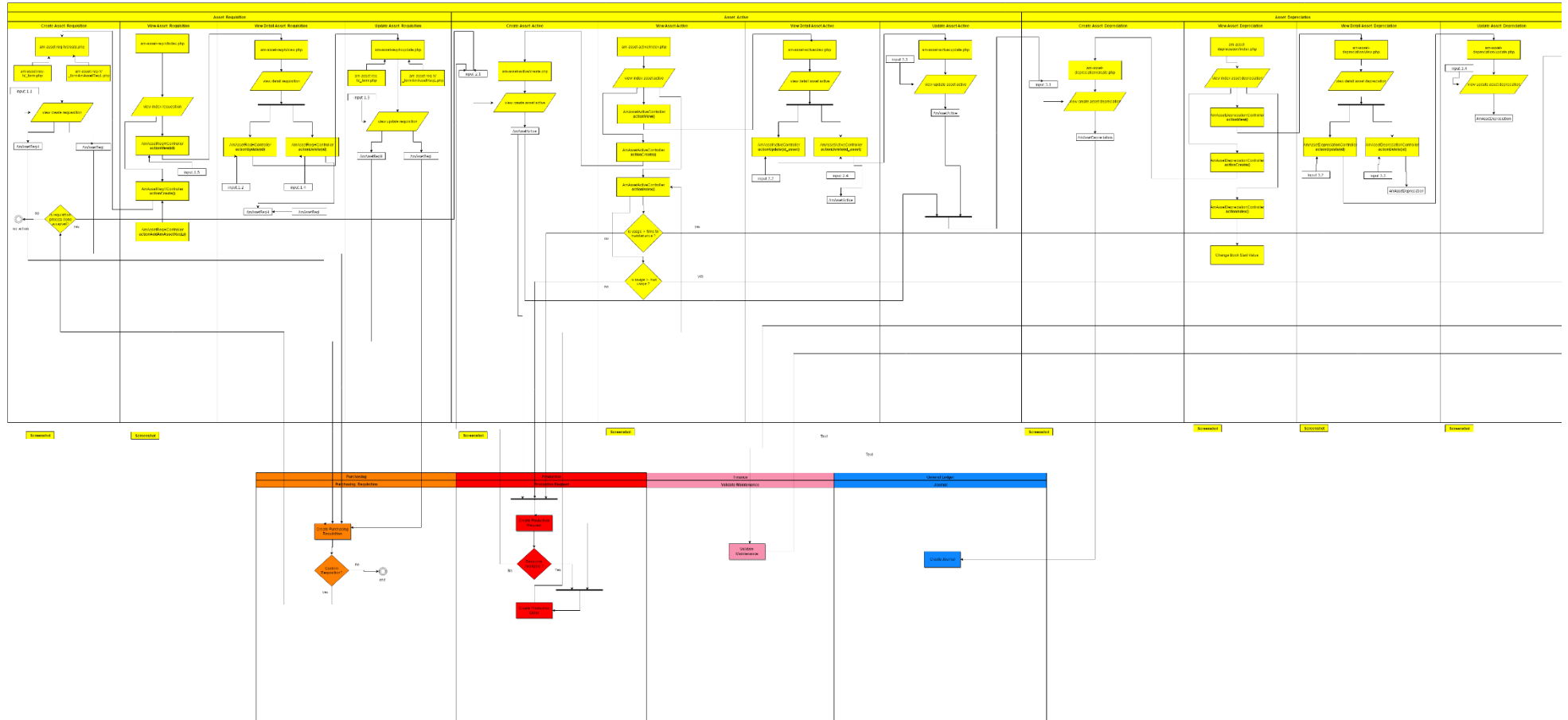


Gambar 3.2 Proses Bisnis Level 1

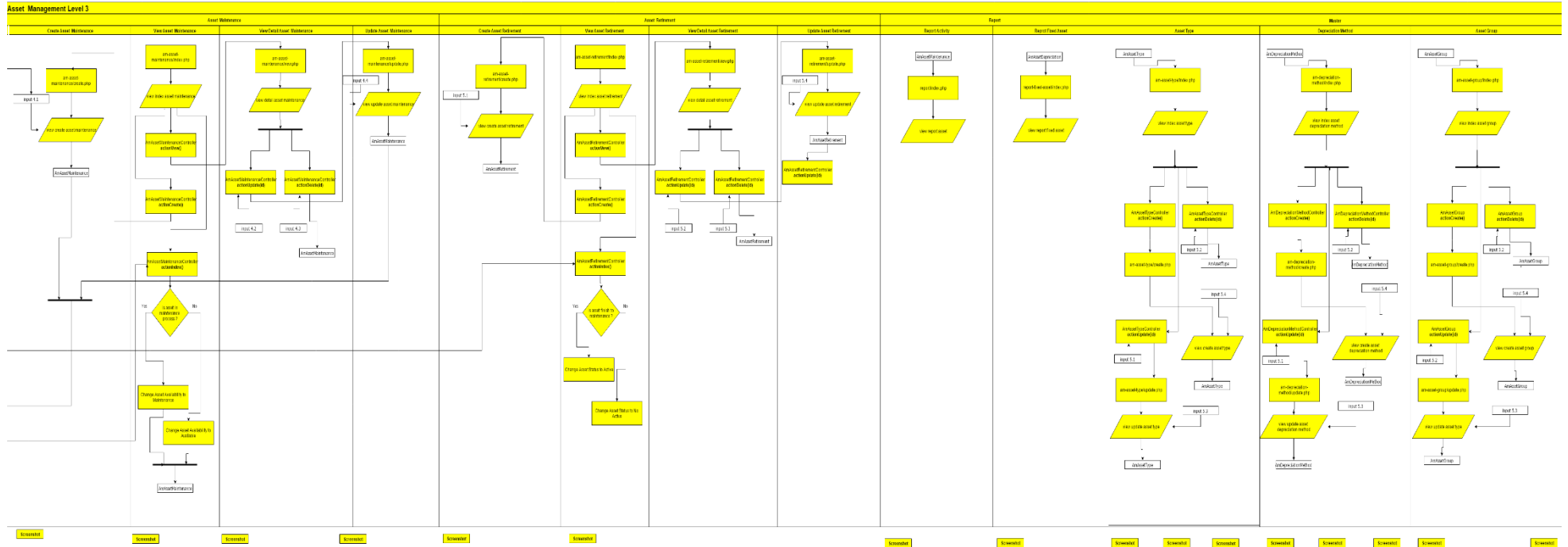
Pada Gambar 3.2 ditunjukkan proses bisnis *level 1*. Level 1 merupakan alur proses bisnis secara umum yang terdapat pada *Asset Management*. Selanjutnya adalah proses bisnis level 2, terdapat pada gambar 3.3. dan pada gambar 3.4 dan 3.5 merupakan proses bisnis level 3



Gambar 3.3 Proses Bisnis Level 2



Gambar 3.4 Diagram Bisnis Proses Level 3 (bag 1)



Gambar 3.5 Diagram Bisnis Proses Level 3 (bag 2)



### 3.2 Identifikasi Pengguna

Pada sistem yang akan dibangun, aktor yang akan menjadi pengguna sistem adalah staff aset dan manager aset. Staff aset mempunyai hak ases dalam pengelolaan aset, yang terdiri atas membuat (*create*), memperbarui (*update*) dan menghapus (*delete*). Sedangkan manager mempunyai hak akses melihat (*view*) data yang telah dikelola oleh staff.

### 3.3 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan dalam sistem ini mencakup kebutuhan fungsional. Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.2

Tabel 3.2 Daftar Kebutuhan Fungsional Sistem

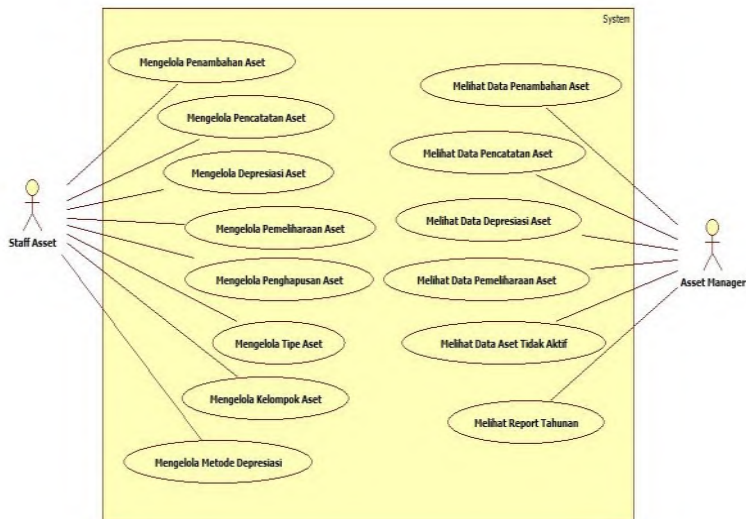
<b>Kode Kebutuhan</b>	<b>Kebutuhan Fungsional</b>	<b>Deskripsi</b>
<b>F-001</b>	Mengelola Penambahan Aset	Pengguna dapat mengelola penambahan aset yang meliputi proses: <i>create</i> , <i>update</i> , <i>delete</i>
<b>F-002</b>	Mengelola Pencatatan Aset	Pengguna dapat mengelola pencatatan aset yang meliputi proses: <i>create</i> , <i>update</i> , <i>delete</i>
<b>F-003</b>	Mengelola Depresiasi Aset	Pengguna dapat mengelola depresiasi aset yang meliputi proses: <i>create</i> , <i>update</i> , <i>delete</i>
<b>F-004</b>	Mengelola Pemeliharaan Aset	Pengguna dapat mengelola pemeliharaan aset yang meliputi proses: <i>create</i> ,

		<i>update, delete</i>
<b>F-005</b>	Mengelola Penghapusan Aset	Pengguna dapat mengelola penghapusan aset yang meliputi proses: <i>create, update</i>
<b>F-006</b>	Mengelola Tipe Aset	Pengguna dapat mengelola tipe aset yang meliputi proses: <i>create, update, delete</i>
<b>F-007</b>	Mengelola Kelompok Aset	Pengguna dapat mengelola kelompok aset yang meliputi proses: <i>create, update, delete</i>
<b>F-008</b>	Mengelola Metode Depresiasi	Pengguna dapat mengelola penambahan aset yang meliputi proses: <i>create, update, delete</i>
<b>F-009</b>	Melihat Data Penambahan Aset	Pengguna dapat melihat data penambahan aset
<b>F-010</b>	Melihat Data Pencatatan Aset	Pengguna dapat melihat data pencatatan aset
<b>F-011</b>	Melihat Data Depresiasi Aset	Pengguna dapat melihat data depresiasi aset
<b>F-012</b>	Melihat Data Pemeliharaan Aset	Pengguna dapat melihat data pemeliharaan aset
<b>F-013</b>	Melihat Data Aset tidak aktif	Pengguna dapat melihat data aset yang tidak aktif
<b>F-014</b>	Melihat Report Tahunan	Pengguna dapat melihat report tahunan

### 3.4 Kasus Penggunaan

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 3.3



Gambar 3.6 Diagram Kasus Penggunaan

Tabel 3.3 Keterangan Kode Kasus Penggunaan

<b>Kode Kasus Penggunaan</b>	<b>Kasus Penggunaan</b>
UC-001	Mengelola Penambahan Aset
UC-002	Mengelola Pencatatan Aset
UC-003	Mengelola Depresiasi Aset
UC-004	Mengelola Pemeliharaan Aset
UC-005	Mengelola Penghapusan Aset
UC-006	Mengelola Tipe Aset

<b>Kode Kasus Penggunaan</b>	<b>Kasus Penggunaan</b>
UC-007	Mengelola Kelompok Aset
UC-008	Mengelola Metode Depresiasi
UC-009	Melihat Data Penambahan Aset
UC-010	Melihat Data Pencatatan Aset
UC-011	Melihat Data Depresiasi Aset
UC-012	Melihat Data Pemeliharaan Aset
UC-013	Melihat Data Aset tidak aktif
UC-014	Melihat Report Tahunan

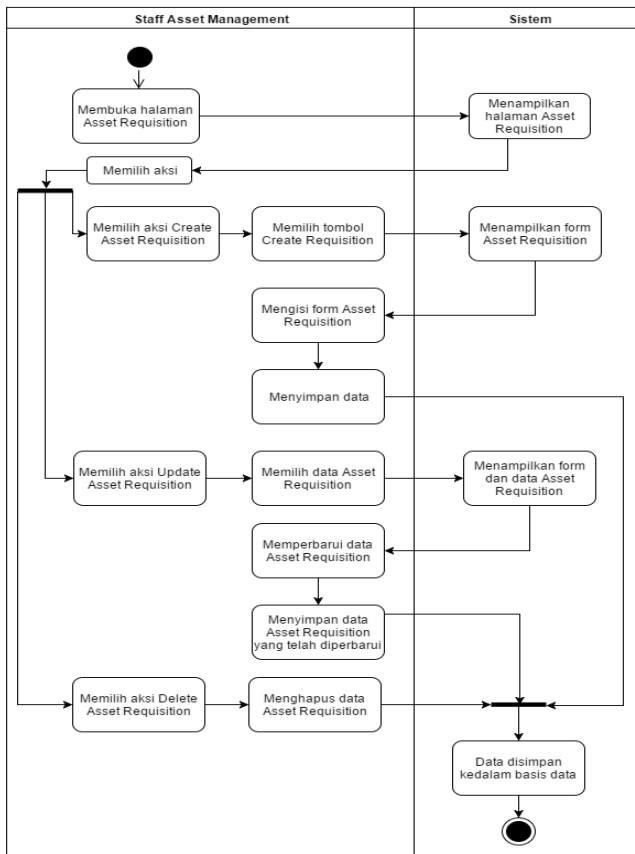
### 3.4.1.1 Kasus Penggunaan Mengelola Penambahan Aset

Pada kasus penggunaan ini, sistem menampilkan halaman untuk melakukan penambahan aset. Pengguna dapat melakukan permintaan penambahan aset dengan mengisi form yang tersedia. Pengguna dapat menambahkan informasi peletakan barang yang akan dikirim, jumlah pemesanan dan barang apa yang akan dipesan. Penjelasan mengenai kasus penggunaan mengelola penambahan aset terdapat pada tabel 3.4 dan gambar 3.7

Tabel 3.4 Kasus Penggunaan Mengelola Penambahan Aset

<b>Nama</b>	Mengelola Penambahan Aset
<b>Kode</b>	UC-001
<b>Deskripsi</b>	Sistem menampilkan halaman permintaan penambahan aset, kemudian sistem menerima masukan data barang dan jumlah barang serta tujuan pengiriman barang. Sistem menyimpan

	masukkan ke dalam basis data
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “Create”
<b>Aktor</b>	Staff Asset Management
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu create Requisition
<b>Aliran:</b> <ul style="list-style-type: none"> <li>• <b>Kejadian Normal</b></li> </ul>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman penambahan aset</li> <li>2. Pengguna memasukkan nama pemesan, tujuan pengiriman, jenis barang.</li> <li>3. Sistem menampilkan pilihan barang yang dapat dipesan.</li> <li>4. Pengguna mengisi nama barang, jumlah dan deskripsi pemesanan.</li> <li>5. Pengguna memilih tombol “Create”</li> <li>6. Sistem mengirimkan data pada basis data.</li> <li>7. Pengguna memilih data yang akan diubah</li> <li>8. Sistem menampilkan data yang telah terpilih</li> <li>9. Pengguna mengubah data yang telah dipilih</li> <li>10. Sistem mengirimkan pada basis data</li> <li>11. Pengguna memilih data yang akan dihapus</li> <li>12. Sistem menampilkan data yang akan dihapus</li> <li>13. Pengguna melakukan penghapusan data</li> <li>14. Sistem menghapus data</li> </ol>
<ul style="list-style-type: none"> <li>• <b>Kejadian Alternatif</b></li> </ul>	<ol style="list-style-type: none"> <li>5a. Pengguna memilih Cancel</li> </ol> <p>Sistem kembali ke kejadian normal 1.</p>
<b>Kondisi Akhir</b>	Data penambahan aset masuk kedalam



Gambar 3.7 Diagram aktifitas mengelola penambahan aset

### 3.4.1.2 Kasus Penggunaan Mengelola Pencatatan Aset

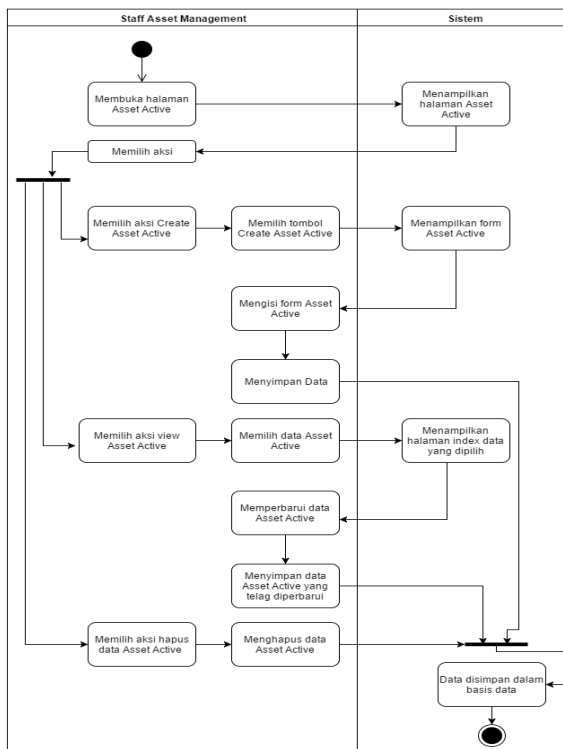
Pada kasus penggunaan ini, sistem menampilkan halaman untuk melakukan pencatatan aset. Aset yang telah dibeli pada proses sebelumnya (UC-001) selanjutnya diperlukan pencatatan agar aset mempunyai nomor registrasi dan keterangan yang dibutuhkan untuk melakukan proses bisnis selanjutnya.

Penjelasan mengenai kasus penggunaan mengelola pencatatan aset terdapat pada tabel 3.5 dan gambar 3.8

Tabel 3.5 Kasus Penggunaan Mengelola Pencatatan Aset

<b>Nama</b>	Mengelola Pencatatan Aset
<b>Kode</b>	UC-002
<b>Deskripsi</b>	Sistem menampilkan halaman pencatatan aset, kemudian sistem menerima masukan nama barang yang telah dibeli melalui proses sebelumnya (UC-001). Sistem menyimpan masukan kedalam basis data
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “Create”
<b>Aktor</b>	Staff Asset Management
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu create Asset Active
<b>Aliran:</b> <ul style="list-style-type: none"> <li>• <b>Kejadian Normal</b></li> </ul>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman pencatatan aset</li> <li>2. Pengguna memasukkan data kelengkapan aset</li> <li>3. Pengguna memilih tombol “Create”</li> <li>4. Sistem mengirimkan data pada basis data.</li> <li>5. Pengguna memilih data yang akan diubah</li> <li>6. Sistem menampilkan data yang telah terpilih</li> <li>7. Pengguna mengubah data yang telah dipilih</li> <li>8. Sistem mengirimkan pada basis data</li> <li>9. Pengguna memilih data yang akan dihapus</li> <li>10. Sistem menampilkan data yang</li> </ol>

	<p>akan dihapus</p> <p>11. Pengguna melakukan penghapusan data</p> <p>12. Sistem menghapus data</p>
<ul style="list-style-type: none"> <li>• <b>Kejadian Alternatif</b></li> </ul>	<p>3a. Pengguna memilih Cancel</p> <p>Sistem kembali ke kejadian normal 1.</p>
<p><b>Kondisi Akhir</b></p>	<p>Data penambahan aset masuk kedalam basis data</p>



Gambar 3.8 Diagram aktifitas kasus penggunaan mengelola pencatatan aset



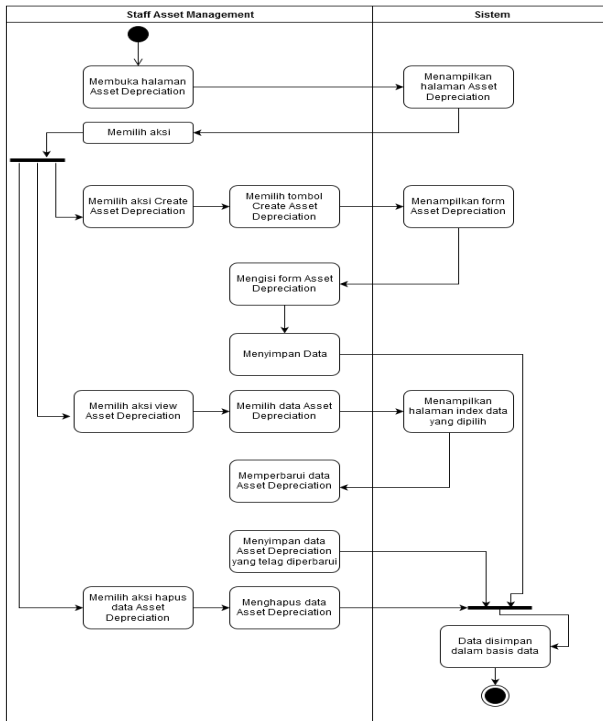
### 3.4.1.3 Kasus Penggunaan Mengelola Depresiasi Aset

Pada kasus penggunaan ini, sistem menampilkan halaman untuk melakukan proses depresiasi aset. Aset yang masih berstatus aktif setiap tahun harus melakukan proses depresiasi untuk mengurangi nilai atau harga aset. Pada proses ini pengguna memilih aset mana yang akan di depresiasi, selanjutnya memilih metode apa yang akan digunakan lalu sistem akan menyimpan masukan yang telah dilakukan oleh pengguna. Penjelasan mengenai kasus penggunaan mengelola depresiasi aset terdapat pada tabel 3.6 dan gambar 3.9

Tabel 3.6 Kasus Penggunaan Mengelola Depresiasi Aset

<b>Nama</b>	Mengelola Depresiasi Aset
<b>Kode</b>	UC-003
<b>Deskripsi</b>	Sistem menampilkan halaman depresiasi aset. Pengguna memasukkan data Sistem menyimpan masukan kedalam basis data
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “Create”
<b>Aktor</b>	Staff Asset Management
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu create Asset Depreciation
<b>Aliran:</b> <ul style="list-style-type: none"> <li>• <b>Kejadian Normal</b></li> </ul>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman depresiasi aset</li> <li>2. Pengguna memasukkan data</li> <li>3. Pengguna memilih tombol “Create”</li> <li>4. Sistem mengirimkan data pada basis data.</li> <li>5. Pengguna memilih data yang akan diubah</li> <li>6. Sistem menampilkan data yang telah terpilih</li> <li>7. Pengguna mengubah data yang telah dipilih</li> </ol>

	<p>8. Sistem mengirimkan pada basis data</p> <p>9. Pengguna memilih data yang akan dihapus</p> <p>10. Sistem menampilkan data yang akan dihapus</p> <p>11. Pengguna melakukan penghapusan data</p> <p>12. Sistem menghapus data</p>
<p>• <b>Kejadian Alternatif</b></p>	<p>3a. Pengguna memilih Cancel</p> <p>Sistem kembali ke kejadian normal 1.</p>
<p><b>Kondisi Akhir</b></p>	<p>Data penambahan aset masuk kedalam basis data</p>



Gambar 3.9 Diagram aktifitas depresiasi aset

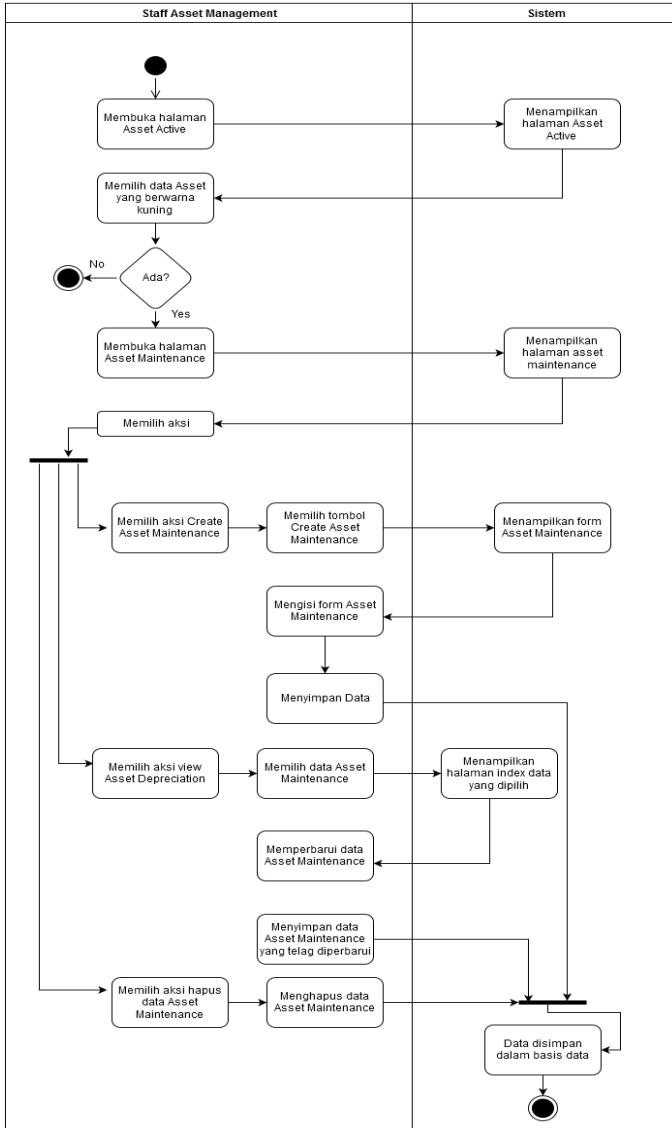
### 3.4.1.4 Kasus Penggunaan Mengelola Pemeliharaan Aset

Pada kasus penggunaan ini, sistem menampilkan halaman untuk melakukan proses pemeliharaan aset. Aset dapat dipelihara (*maintenance*) agar aset tetap pada kondisi yang baik saat digunakan. Proses pemeliharaan diawali dengan, pengguna melihat halaman asset active, lalu memilih tabel yang berwarna kuning, apabila di dalam tabel terdapat warna kuning maka pengguna dapat melakukan pemeliharaan aset dengan cara masuk atau memilih tombol create asset maintenance selanjutnya memilih aset dan mengisi data. Sistem akan menyimpan masukkan pada basis data. Penjelasan mengenai kasus penggunaan mengelola penambahan aset terdapat pada tabel 3.7 dan gambar 3.10

Tabel 3.7 kasus penggunaan mengelola pemeliharaan aset

<b>Nama</b>	Mengelola Pemeliharaan Aset
<b>Kode</b>	UC-004
<b>Deskripsi</b>	Sistem menampilkan halaman pemeliharaan aset. Pengguna memasukkan data Sistem menyimpan masukkan kedalam basis data
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol "Create"
<b>Aktor</b>	Staff Asset Management
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu create Asset Depreciation
<b>Aliran:</b> • <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman asset active</li> <li>2. Pengguna melihat aset dengan tanda kuning</li> <li>3. Pengguna masuk pada halaman asset maintenance</li> <li>4. Sistem menampilkan halaman asset maintenance</li> <li>5. Pengguna memilih data aset yang</li> </ol>

	<p>akan di maintenance</p> <ol style="list-style-type: none"> <li>6. Pengguna melengkapi form asset maintenance</li> <li>7. Pengguna memilih tombol create</li> <li>8. Sistem mengirimkan pada basis data</li> <li>9. Pengguna memilih data yang akan diubah</li> <li>10. Sistem menampilkan data yang telah terpilih</li> <li>11. Pengguna mengubah data yang telah dipilih</li> <li>12. Sistem mengirimkan pada basis data</li> <li>13. Pengguna memilih data yang akan dihapus</li> <li>14. Sistem menampilkan data yang akan dihapus</li> <li>15. Pengguna melakukan penghapusan data</li> <li>16. Sistem menghapus data</li> </ol>
<ul style="list-style-type: none"> <li>• <b>Kejadian Alternatif</b></li> </ul>	<ol style="list-style-type: none"> <li>2a. Pengguna tidak menemukan aset yang perlu untuk di <i>maintenance</i></li> <li>7a. Pengguna memilih Cancel</li> </ol> <p>Sistem kembali ke kejadian normal 1.</p>
<b>Kondisi Akhir</b>	Data penambahan aset masuk kedalam basis data



Gambar 3.10 diagram aktifitas mengelola pemeliharaan aset

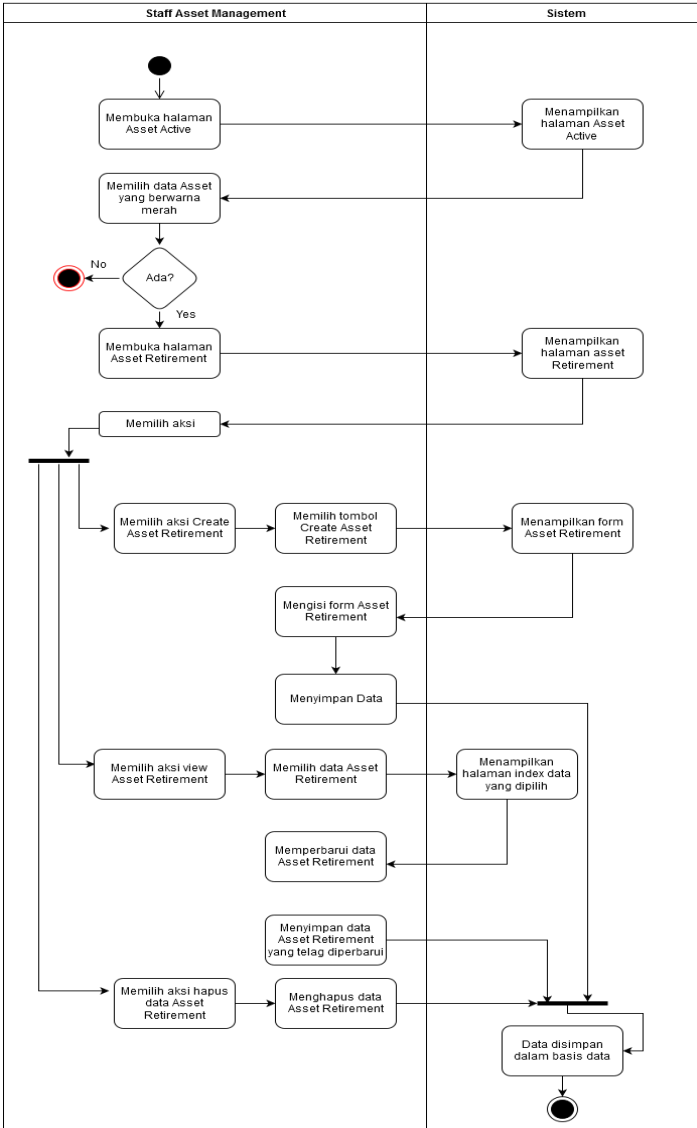
### 3.4.1.5 Kasus Penggunaan Mengelola Penghapusan Aset

Pada kasus penggunaan ini, sistem menampilkan halaman untuk melakukan proses penghapusan aset. Penghapusan aset yang dimaksud adalah mengubah status aset dari aktif menjadi tidak aktif atau retired. Aset yang sudah tidak aktif tidak dapat digunakan kembali. Proses diawali dengan pengguna melihat halaman aset aktif lalu memilih aset yang bertanda merah, selanjutnya pengguna masuk kedalam halaman asset retired dan memilih aset yang telah bertanda merah. Sistem akan menyimpan masukkan pada basis data. Penjelasan mengenai kasus penggunaan mengelola penghapusan aset terdapat pada tabel 3.8 dan gambar 3.11

Tabel 3.8 kasus penggunaan mengelola penghapusan aset

<b>Nama</b>	Mengelola Pemeliharaan Aset
<b>Kode</b>	UC-005
<b>Deskripsi</b>	Sistem menampilkan halaman penghapusan aset. Pengguna memasukkan data Sistem menyimpan masukkan kedalam basis data
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “Create”
<b>Aktor</b>	Staff Asset Management
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu create Asset Depreciation
<b>Aliran:</b> <ul style="list-style-type: none"> <li>• <b>Kejadian Normal</b></li> </ul>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman asset active</li> <li>2. Pengguna melihat aset dengan tanda merah</li> <li>3. Pengguna masuk pada halaman asset retirement</li> <li>4. Sistem menampilkan halaman asset retirement</li> <li>5. Pengguna memilih data aset yang</li> </ol>

	<p>akan di non-aktifkan</p> <ol style="list-style-type: none"> <li>6. Pengguna melengkapi form asset retirement</li> <li>7. Pengguna memilih tombol create</li> <li>8. Sistem mengirimkan pada basis data</li> <li>9. Pengguna memilih data yang akan diubah</li> <li>10. Sistem menampilkan data yang telah terpilih</li> <li>11. Pengguna mengubah data yang telah dipilih</li> <li>12. Sistem mengirimkan pada basis data</li> <li>13. Pengguna memilih data yang akan dihapus</li> <li>14. Sistem menampilkan data yang akan dihapus</li> <li>15. Pengguna melakukan penghapusan data</li> <li>16. Sistem menghapus data</li> </ol>
<ul style="list-style-type: none"> <li>• <b>Kejadian Alternatif</b></li> </ul>	<ol style="list-style-type: none"> <li>2a. Pengguna tidak menemukan aset yang perlu untuk di non-aktif</li> <li>7a. Pengguna memilih Cancel</li> </ol> <p>Sistem kembali ke kejadian normal 1.</p>
<b>Kondisi Akhir</b>	Data penambahan aset masuk kedalam basis data



Gambar 3.11 diagram aktifitas mengelola penghapusan aset



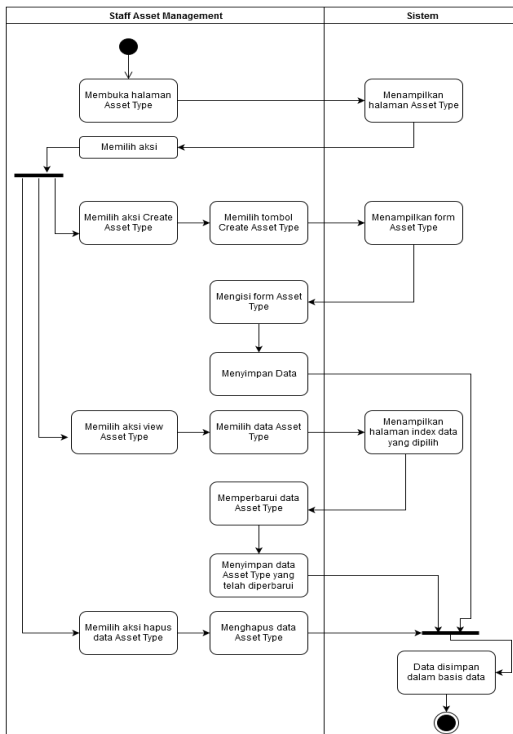
### 3.4.1.6 Kasus Penggunaan Mengelola Tipe Aset

Pada kasus penggunaan ini, sistem menampilkan halaman untuk melakukan pengelolaan tipe aset. Pada proses ini pengguna memasukkan nama, nomor, serta tipe aset. Proses ini diperlukan untuk kasus penggunaan mengelola pencatatan aset aktif. Selanjutnya sistem akan menyimpan masukkan yang telah dilakukan oleh pengguna. Penjelasan mengenai kasus penggunaan mengelola tipe aset terdapat pada tabel 3.9 dan gambar 3.12

Tabel 3.9 kasus penggunaan mengelola tipe aset

<b>Nama</b>	Mengelola Tipe Aset
<b>Kode</b>	UC-006
<b>Deskripsi</b>	Sistem menampilkan halaman tipe aset. Pengguna memasukkan data Sistem menyimpan masukkan kedalam basis data
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol "Create"
<b>Aktor</b>	Staff Asset Management
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu create Asset Depreciation
<b>Aliran:</b> • <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman asset type</li> <li>2. Pengguna memasukkan tipe aset</li> <li>3. Pengguna memilih tombol create</li> <li>4. Sistem mengirimkan pada basis data</li> <li>5. Pengguna memilih data yang akan diubah</li> <li>6. Sistem menampilkan data yang telah terpilih</li> <li>7. Pengguna mengubah data yang telah dipilih</li> <li>8. Sistem mengirimkan pada basis data</li> <li>9. Pengguna memilih data yang akan</li> </ol>

	<p>dihapus</p> <p>10. Sistem menampilkan data yang akan dihapus</p> <p>11. Pengguna melakukan penghapusan data</p> <p>12. Sistem menghapus data</p>
<ul style="list-style-type: none"> <li>• <b>Kejadian Alternatif</b></li> </ul>	<p>3a. Pengguna memilih Cancel</p> <p>Sistem kembali ke kejadian normal 1.</p>
<p><b>Kondisi Akhir</b></p>	<p>Data penambahan aset masuk kedalam basis data</p>



Gambar 3.12 diagram aktifitas mengelola tipe aset

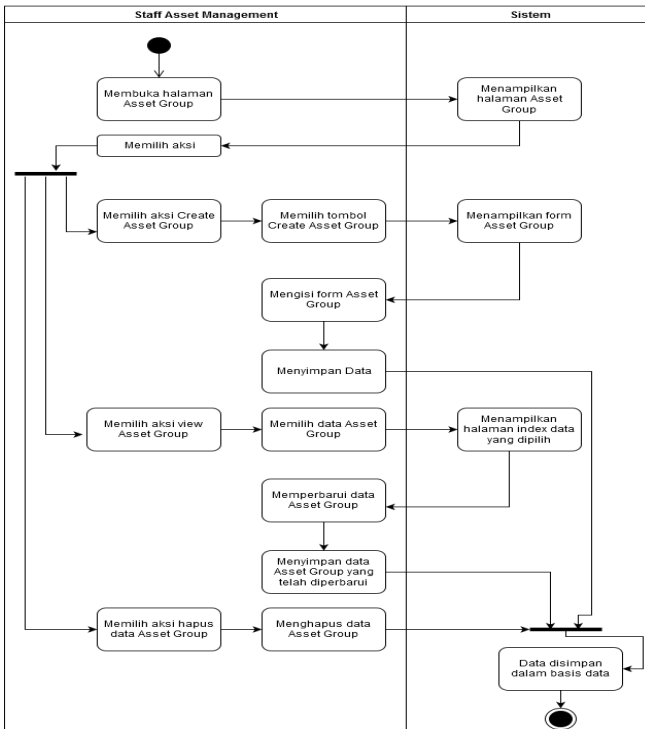
### 3.4.1.7 Kasus Penggunaan Mengelola Kelompok Aset

Pada kasus penggunaan ini, sistem menampilkan halaman untuk melakukan pengelolaan kelompok aset. Pada proses ini pengguna memasukkan nama kelompok, masa penggunaan aset dan rate aset. Proses ini diperlukan untuk kasus penggunaan mengelola pencatatan aset aktif. Selanjutnya sistem akan menyimpan masukan yang telah dilakukan oleh pengguna. Penjelasan mengenai kasus penggunaan mengelola kelompok aset terdapat pada tabel 3.10 dan gambar 3.13

Tabel 3.10 Kasus penggunaan mengelola kelompok aset

<b>Nama</b>	Mengelola Kelompok Aset
<b>Kode</b>	UC-007
<b>Deskripsi</b>	Sistem menampilkan halaman kelompok aset. Pengguna memasukkan data Sistem menyimpan masukan kedalam basis data
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “Create”
<b>Aktor</b>	Staff Asset Management
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu create Asset Depreciation
<b>Aliran:</b> • <b>Kejadian Normal</b>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman asset group</li> <li>2. Pengguna memasukkan tipe kelompok aset</li> <li>3. Pengguna memilih tombol create</li> <li>4. Sistem mengirimkan pada basis data</li> <li>5. Pengguna memilih data yang akan diubah</li> <li>6. Sistem menampilkan data yang telah terpilih</li> <li>7. Pengguna mengubah data yang telah dipilih</li> </ol>

	<p>8. Sistem mengirimkan pada basis data</p> <p>9. Pengguna memilih data yang akan dihapus</p> <p>10. Sistem menampilkan data yang akan dihapus</p> <p>11. Pengguna melakukan penghapusan data</p> <p>12. Sistem menghapus data</p>
<p>• <b>Kejadian Alternatif</b></p>	<p>3a. Pengguna memilih Cancel</p> <p>Sistem kembali ke kejadian normal 1.</p>
<p><b>Kondisi Akhir</b></p>	<p>Data penambahan aset masuk kedalam basis data</p>



Gambar 3.13 Diagram aktifitas mengelola tipe aset

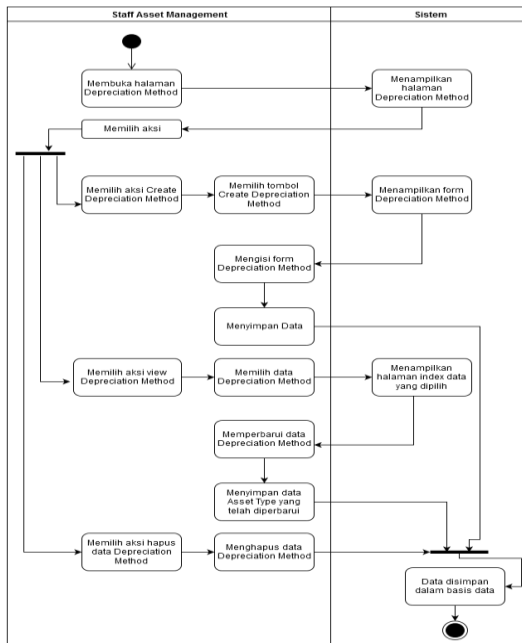
### 3.4.1.8 Kasus Penggunaan Mengelola Metode Depresiasi Aset

Pada kasus penggunaan ini, sistem menampilkan halaman untuk melakukan pengelolaan metode depresiasi aset. Pada proses ini pengguna memasukkan nama metode dan nomor. Proses ini diperlukan untuk kasus penggunaan mengelola depresiasi aset. Selanjutnya sistem akan menyimpan masukkan yang telah dilakukan oleh pengguna. Penjelasan mengenai kasus penggunaan mengelola metode depresiasi aset terdapat pada tabel 3.11 dan gambar 3.14

Tabel 3.11 Kasus penggunaan mengelola metode depresiasi aset

<b>Nama</b>	Mengelola Kelompok Aset
<b>Kode</b>	UC-008
<b>Deskripsi</b>	Sistem menampilkan halaman kelompok aset. Pengguna memasukkan data Sistem menyimpan masukkan kedalam basis data
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “Create”
<b>Aktor</b>	Staff Asset Management
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu create Asset Depreciation
<b>Aliran:</b> • <b>Kejadian Normal</b>	13. Sistem menampilkan halaman asset group 14. Pengguna memasukkan tipe kelompok aset 15. Pengguna memilih tombol cretae 16. Sistem mengirimkan pada basis data 17. Pengguna memilih data yang akan diubah 18. Sistem menampilkan data yang telah terpilih 19. Pengguna mengubah data yang

	<p>telah dipilih</p> <p>20. Sistem mengirimkan pada basis data</p> <p>21. Pengguna memilih data yang akan dihapus</p> <p>22. Sistem menampilkan data yang akan dihapus</p> <p>23. Pengguna melakukan penghapusan data</p> <p>24. Sistem menghapus data</p>
<ul style="list-style-type: none"> <li>• <b>Kejadian Alternatif</b></li> </ul>	<p>3a. Pengguna memilih Cancel</p> <p>Sistem kembali ke kejadian normal 1.</p>
<p><b>Kondisi Akhir</b></p>	<p>Data penambahan aset masuk kedalam basis data</p>



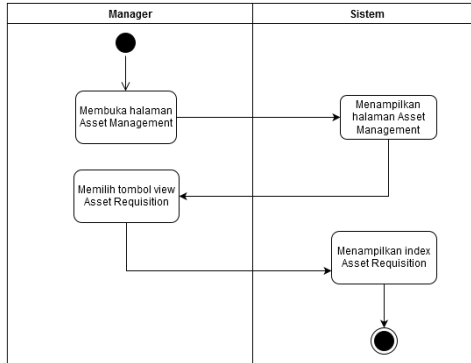
Gambar 3.14 Diagram aktifitas mengelola metode depresiasi aset

### 3.4.1.9 Kasus Penggunaan Melihat Data Penambahan Aset

Pada kasus penggunaan ini, pengguna (manager) hanya dapat melihat data penambahan aset yang telah dikelola oleh staff. Proses ini bertujuan untuk melakukan monitoring terhadap aktifitas yang dilakukan oleh staff. Penjelasan mengenai kasus penggunaan melihat data penambahan aset terdapat pada tabel 3.12 dan gambar 3.15

Tabel 3.12 Kasus Penggunaan Melihat Data Penambahan Aset

<b>Nama</b>	Mengelola Kelompok Aset
<b>Kode</b>	UC-009
<b>Deskripsi</b>	Sistem menampilkan halaman view Asset Requisition
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “View”
<b>Aktor</b>	Manager
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu view asset requisition
<b>Aliran:</b>	
• <b>Kejadian Normal</b>	1. Sistem menampilkan halaman view asset requisition 2. Pengguna melihat halaman view asset requisition
• <b>Kejadian Alternatif</b>	-
<b>Kondisi Akhir</b>	Data ditampilkan pada pengguna



Gambar 3.15 Diagram aktifitas melihat data penambahan aset

### 3.4.1.10 Kasus Penggunaan Melihat Data Aset Aktif

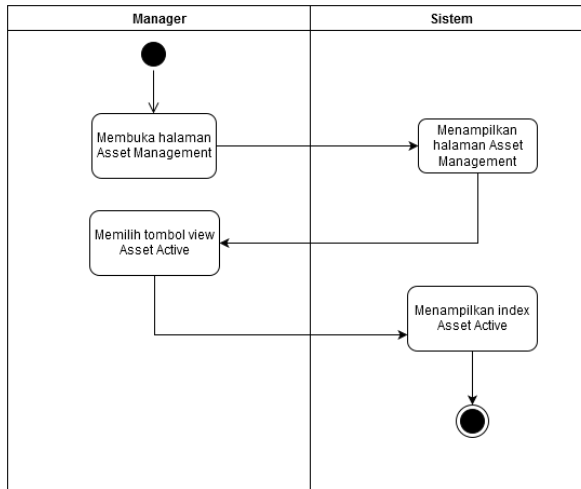
Pada kasus penggunaan ini, pengguna (manager) hanya dapat melihat data aset aktif yang telah dikelola oleh staff. Proses ini bertujuan untuk melakukan monitoring terhadap aktifitas yang dilakukan oleh staff. Serta melihat apakah ada kesalahan yang dilakukan oleh staff dalam memberikan keterangan pada aset. Penjelasan mengenai kasus penggunaan melihat data aset aktif terdapat pada tabel 3.13 dan gambar 3.16

Tabel 3.13 Kasus penggunaan melihat data aset aktif

<b>Nama</b>	Mengelola Kelompok Aset
<b>Kode</b>	UC-010
<b>Deskripsi</b>	Sistem menampilkan halaman view Asset Active
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “View”
<b>Aktor</b>	Manager
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu view asset active
<b>Aliran:</b>	1. Sistem menampilkan halaman view asset active
• <b>Kejadian</b>	



<b>Normal</b>	2. Pengguna melihat halaman view asset active
• <b>Kejadian Alternatif</b>	-
<b>Kondisi Akhir</b>	Data ditampilkan pada pengguna



Gambar 3.16 Diagram aktifitas melihat asset aktif

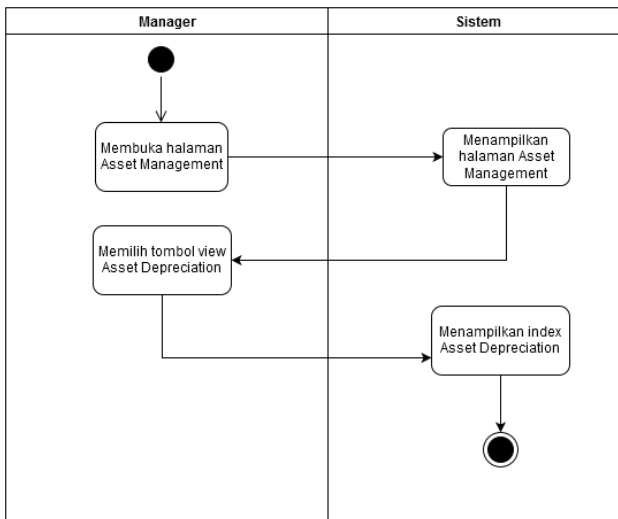
### 3.4.1.11 Kasus Penggunaan Melihat Data Penyusutan Aset

Pada kasus penggunaan ini, pengguna (manager) hanya dapat melihat data penyusutan aset yang telah dikelola oleh staff. Proses ini bertujuan untuk melakukan monitoring terhadap aktifitas yang dilakukan oleh staff. Serta melihat apakah ada kesalahan yang dilakukan oleh staff dalam perhitungan penyusutan aset. Penjelasan mengenai kasus penggunaan melihat data penyusutan aset terdapat pada tabel 3.14 dan gambar 3.17

Tabel 3.14 Kasus penggunaan melihat data penyusutan aset

<b>Nama</b>	Mengelola Kelompok Aset
<b>Kode</b>	UC-011

<b>Deskripsi</b>	Sistem menampilkan halaman view Asset Active
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “View”
<b>Aktor</b>	Manager
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu view asset depreciation
<b>Aliran:</b>	1. Sistem menampilkan halaman view asset depreciation 2. Pengguna melihat halaman view asset depreciation
<ul style="list-style-type: none"> <li><b>Kejadian Alternatif</b></li> </ul>	-
<b>Kondisi Akhir</b>	Data ditampilkan pada pengguna



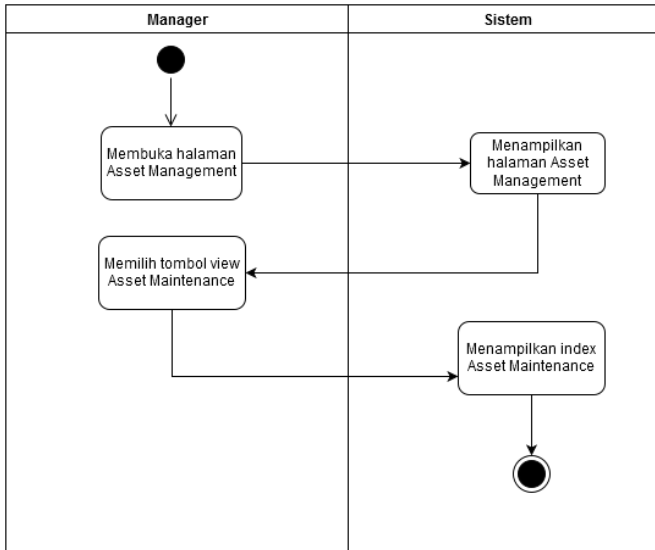
Gambar 3.17 Diagram aktifitas melihat penyusutan aset

### 3.4.1.12 Kasus Penggunaan Melihat Data Pemeliharaan Aset

Pada kasus penggunaan ini, pengguna (manager) hanya dapat melihat data pemeliharaan aset yang telah dikelola oleh staff. Proses ini bertujuan untuk melakukan monitoring terhadap aktifitas yang dilakukan oleh staff. Serta melihat apakah ada kesalahan yang dilakukan oleh staff. Penjelasan mengenai kasus penggunaan melihat data pemeliharaan aset terdapat pada tabel 3.15 dan gambar 3.18

Tabel 3.15 Kasus penggunaan melihat data pemeliharaan aset

<b>Nama</b>	Melihat Data Pemeliharaan Aset
<b>Kode</b>	UC-012
<b>Deskripsi</b>	Sistem menampilkan halaman view Asset Maintenance
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “View”
<b>Aktor</b>	Manager
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu view asset Maintenance
<b>Aliran:</b>	
<ul style="list-style-type: none"> <li>• <b>Kejadian Normal</b></li> </ul>	<ol style="list-style-type: none"> <li>1. Sistem menampilkan halaman view asset depreciation</li> <li>2. Pengguna melihat halaman view asset depreciation</li> </ol>
<ul style="list-style-type: none"> <li>• <b>Kejadian Alternatif</b></li> </ul>	-
<b>Kondisi Akhir</b>	Data ditampilkan pada pengguna



Gambar 3.18 Diagram aktifitas melihat data pemeliharaan aset

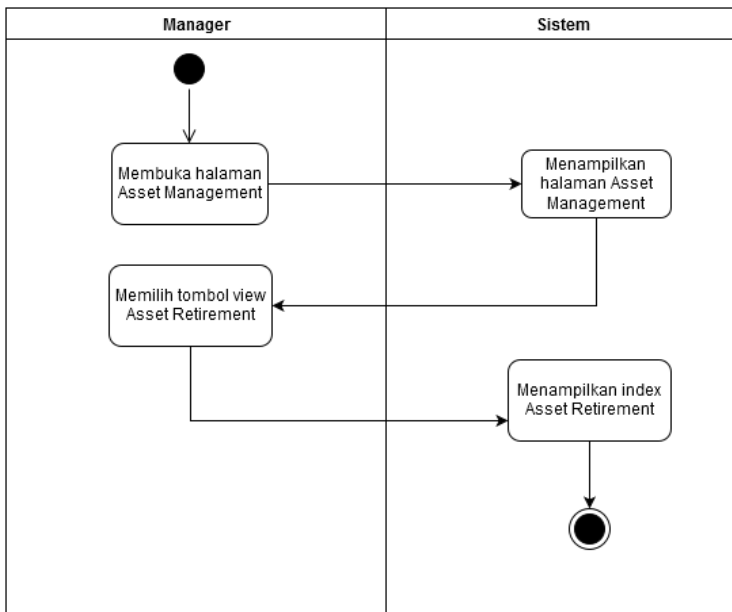
### 3.4.1.13 Kasus Penggunaan Melihat Data Aset Tidak Aktif

Pada kasus penggunaan ini, pengguna (manager) hanya dapat melihat data aset sudah tidak layak digunakan. Proses ini bertujuan untuk melakukan monitoring terhadap aktifitas yang dilakukan oleh staff. Penjelasan mengenai kasus penggunaan melihat data pemeliharaan aset terdapat pada tabel 3.16 dan gambar 3.19

Tabel 3.16 Kasus penggunaan melihat data aset tidak aktif

<b>Nama</b>	Melihat Data Aset Tidak Aktif
<b>Kode</b>	UC-013
<b>Deskripsi</b>	Sistem menampilkan halaman view Asset Maintenance
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “View”
<b>Aktor</b>	Manager

<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu view asset retirement
<b>Aliran:</b> • <b>Kejadian Normal</b>	1. Sistem menampilkan halaman view asset depreciation 2. Pengguna melihat halaman view asset depreciation
• <b>Kejadian Alternatif</b>	-
<b>Kondisi Akhir</b>	Data ditampilkan pada pengguna



Gambar 3.19 Diagram aktifitas melihat data aset tidak aktif

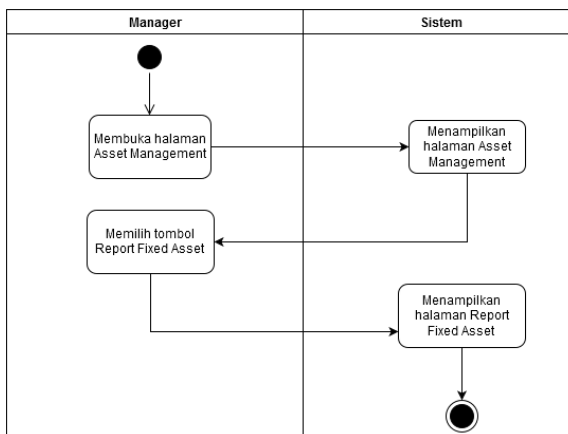
#### 3.4.1.14 Kasus Penggunaan Melihat Report Tahunan

Pada kasus penggunaan ini, pengguna (manager) dapat melihat report tahunan. Report ini berisi aktifitas apa saja yang

dilakukan dalam satu tahun terakhir. Penjelasan mengenai kasus penggunaan melihat data report tahunan terdapat pada tabel 3.17 dan gambar 3.20

Tabel 3.17 Kasus penggunaan melihat report tahunan

<b>Nama</b>	Melihat Report Tahunan
<b>Kode</b>	UC-014
<b>Deskripsi</b>	Sistem menampilkan halaman report
<b>Tipe</b>	Fungsional
<b>Pemicu</b>	Pengguna menekan tombol “View”
<b>Aktor</b>	Manager
<b>Kondisi Awal</b>	Pengguna berada pada menu utama dan memilih menu view asset retirement
<b>Aliran:</b>	1. Sistem menampilkan halaman report tahunan 2. Pengguna melihat halaman report tahunan
• <b>Kejadian Normal</b>	
• <b>Kejadian Alternatif</b>	-
<b>Kondisi Akhir</b>	Data ditampilkan pada pengguna



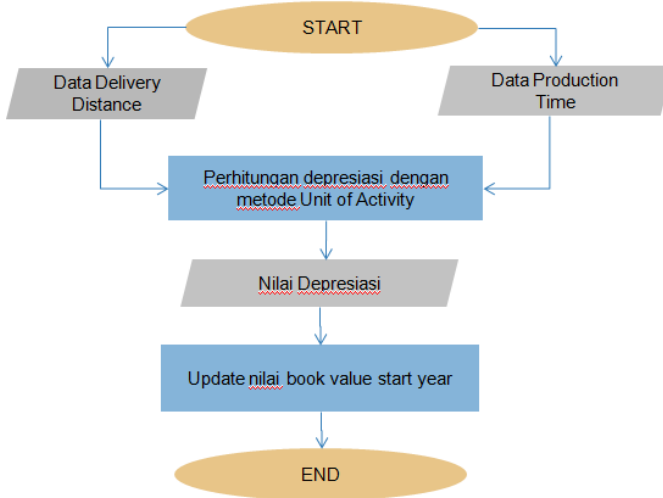
Gambar 3.20 Diagram aktifitas melihat report tahunan

### 3.5 Perancangan

Tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan perhitungan penurunan harga aset (Make-To-Order dan Make-To-Stock), perancangan perhitungan penurunan harga aset (General), perancangan arsitektur basis data terdistribusi, perancangan *multi-tenancy*, perancangan RBAC, dan perancangan antarmuka sistem

#### 3.5.1 Perancangan Perhitungan Penurunan Harga Aset (Make-To-Order dan Make-To-Stock)

Ada beberapa metode untuk proses perhitungan penurunan harga barang seperti yang sudah dijelaskan pada bab 2.5, 2.6 dan 2.7, tetapi untuk barang (aset) yang digunakan pada proses bisnis Make-To-Order dan Make-To-Stock adalah metode *Unit of Activity*. Metode dipilih karena dapat menghasilkan nilai penurunan harga sesuai dengan bobot pekerjaan yang dilakukan, baik waktu maupun jarak. Proses perhitungan dimulai saat modul *Production Planning* melakukan produksi dan modul Sales melakukan pengiriman barang. Setiap modul akan mengirimkan jarak maupun waktu yang telah digunakan, selanjutnya akan dilakukan perhitungan penyusutan harga barang yang dijelaskan pada subbab 2.14 dari perhitungan tersebut akan di hasilkan nilai depresiasi, yang selanjutnya dilakukan update harga barang terbaru yang disebut *book value start year*. Proses-proses tersebut ditunjukkan pada Gambar 3.21 dan detail aset terdapat pada tabel 3.18.



Gambar 3.21 Alur Proses Perhitungan

Berikut contoh proses perhitungan depresiasi dengan metode *Unit of Activity*

Tabel 3.18 Detail Aset

Nama Aset	Harga Aset	Jarak <i>Delivery</i>	Useful Unit	Book Value Start year
TRUC K HINO	Rp. 100.000.000, -	500km	1.000.000k m	99.960.000
		1000km		99.900.040
		2000km		99.740.239
		3000km		99.501.019

### 1. Menghitung *Residual Value*

*Residual Value* adalah nilai sisa saat aset mengalami *retired*, aset masih mempunyai harga jual. Nilai sisa dihitung 20% dari harga aset saat pertama kali dibeli, sehingga  
 $\text{Rp } 100.000.000 \times 20\% = 20.000.000$



## 2. Menghitung Nilai Penurunan Harga Barang

Setelah ditentukan *residual value* dan mendapatkan jarak tempuh dari modul *Sales and Distribution*, maka dapat dilakukan perhitungan penurunan harga barang dengan cara

### - **Delivery 1**

$$\begin{aligned} & (\text{Rp } 100.000.000 - \text{Rp } 20.000.000) / 500\text{km} * 1.000.000\text{km} \\ & = \text{Rp. } 40.000,- \end{aligned}$$

Jadi, pada proses delivery sejauh 500 km kendaraan mengalami penurunan sejumlah Rp. 40.000,-

### - **Delivery 2**

$$\begin{aligned} & (\text{Rp } 100.000.000 - \text{Rp } 20.000.000) / 1000\text{km} * 1.000.000\text{km} \\ & = \text{Rp. } 59.960,- \end{aligned}$$

Jadi, pada proses delivery sejauh 1000 km kendaraan mengalami penurunan sejumlah Rp. Rp. 59.960,-

## 3. Update Nilai Book Value Start Year

Book value start year merupakan harga barang setelah mengalami proses depresiasi. Harga awal dikurangi dengan nilai hasil proses perhitungan selanjutnya merupakan harga aset yang baru

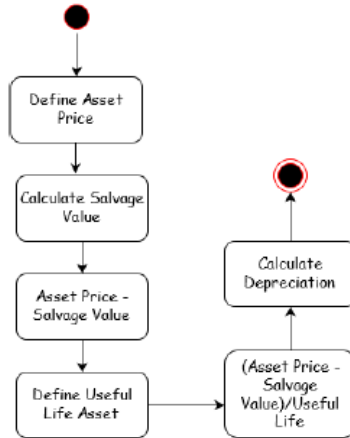
$$\text{Rp. } 100.000.000 - \text{Rp. } 40.0000 = \text{Rp } 99.960.000,-$$

Update dilakukan setiap kali melakukan proses penurunan harga

### 3.5.2 Perancangan Perhitungan Penurunan Harga Aset (*Straight Line Method*)

Bab 3.5.1 telah membahas proses perhitungan penurunan harga untuk aset yang digunakan pada proses *Make-To-Stock* dan *Make-To-Order*. Aset yang tidak bergerak seperti Tanah, bangunan dan peralatan kantor dihitung dengan metode yang berbeda. Perhitungan penurunan dilakukan setiap tahun sesuai dengan tanggal pembelian barang. Pengguna dapat memilih metode penurunan yang telah disediakan

Alur proses perhitungan penurunan harga aset untuk barang-barang yang tidak bergerak dijelaskan pada gambar 3.22 dan Detail asset terdapat pada table 3.19



Gambar 3.22 Alur proses penurunan (general)

Tabel 3.19 Detail Aset

Nama Aset	Harga Aset	Periode	Tahun ke-	Book value start year
GEDUNG KANTOR	Rp. 1.000.000.000,-	20 Tahun	1	960.000.000
			2	920.000.000
			3	880.000.000
			4	840.000.000

### 1. Menghitung Salvage Value

*Salvage Value* adalah nilai sisa saat aset mengalami *retired*, aset masih mempunyai harga jual. Nilai sisa dihitung 20% dari harga aset saat pertama kali dibeli, sehingga  
 $\text{Rp } 1.000.000.000 \times 20\% = 200.000.000$

### 2. Menghitung Penurunan Harga

Proses penurunan harga pada pada barang yang tidak bergerak dilakukan secara statis sehingga mempunyai nilai yang sama setiap tahun

Rp. 1.000.000.000 – Rp. 200.000.000 / 20 Tahun  
 = Rp. 40.000.000,-

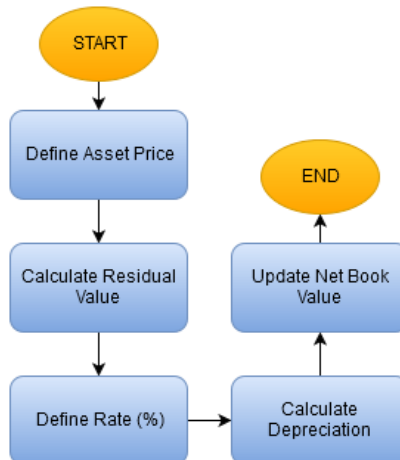
### 3. Update Book value start year

Book value start year merupakan harga barang setelah mengalami proses depresiasi. Harga awal dikurangi dengan nilai hasil proses perhitungan selanjutnya merupakan harga aset yang baru

Rp. 1.000.000.000 – Rp.40.000.000 = Rp 960.000.000,-  
 Update dilakukan setiap tahun

### 3.5.3 Perancangan Perhitungan Penurunan Harga Aset (*Reducing Balance Method*)

Metode lain yang dapat digunakan untuk menghitung penurunan asset adalah metode saldo menurun atau *Reducing Balance Method*. Rumus yang digunakan untuk menghitung penurunan sudah dijelaskan pada subbab 2.6. Alur proses reducing balance method dapat dilihat pada gambar 3.23 dan detail asset terdapat pada tabel 3.20



Gambar 3.23 Alur proses reducing balance method

Tabel 3.20 Detail Aset

Nama Aset	Harga Aset	Periode	Tahun ke-	Net Book Value
PABRIK	Rp. 2.000.000. 000,-	15 Tahun	1	1.880.000.000
			2	1.760.000.000
			3	1.640.000.000
			4	1.520.000.000

### 1. Menghitung Residual Value

*Residual Value* adalah nilai sisa saat aset mengalami *retired*, aset masih mempunyai harga jual. Nilai sisa dihitung 20% dari harga aset saat pertama kali dibeli, sehingga

$$\text{Rp } 2.000.000.000 \times 20\% = 400.000.000$$

### 2. Mendefinisikan Rate

Rate adalah nilai penurunan yang ditentukan oleh pengguna. Pengguna dapat memasukkan rate sesuai dengan keadaan pasar saat ini. Diasumsikan rate pasar adalah 7,5% untuk penurunan harga barang

### 3. Menghitung Penurunan Harga Barang

Perhitungan harga barang dengan metode ini bersifat statis sehingga nilai yang dihasilkan dari tahun ke tahun sama.

$$\begin{aligned} &\text{Rp. } 2.000.000.000 - \text{Rp. } 400.000.000 * 7,5\% \\ &= \text{Rp. } 120.000.000,- \end{aligned}$$

Jadi, pada tahun pertama dan tahun selanjutnya penurunan harga untuk aset adalah Rp. 120.000.000,-

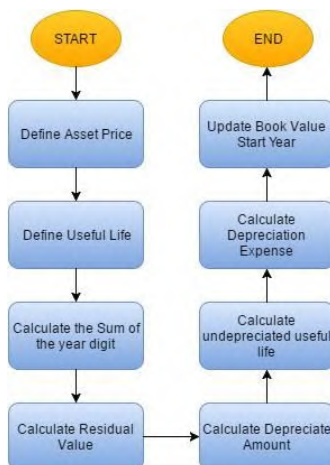
### 4. Update Net Book Value

Net Book Value merupakan harga barang setelah mengalami proses depresiasi. Harga awal dikurangi dengan nilai hasil proses perhitungan selanjutnya merupakan harga aset yang baru

Rp.2.000.000.000 – Rp.120.000.000 = Rp1.880.000.000,-  
Update dilakukan setiap tahun

### 3.5.4 Perancangan Perhitungan Penurunan Harga Aset (*Sum of The Year Method*)

Metode selanjutnya yang dapat digunakan adalah penurunan harga dengan metode penambahan digit angka, rumus yang digunakan dijelaskan pada subbab 2.7. Alur perhitungan dapat dilihat pada gambar 3.24 dan detail asset yang digunakan terdapat pada tabel 3.21



Gambar 3.24 Alur proses Sum of the year

Tabel 3.21 Detail Aset

Nama Aset	Harga Aset	Periode	Tahun ke-	Depreciation Expense	Book Value Start Year
AC	Rp. 5.000.000	5 Tahun	1	1.333.333	3.666.667
			2	1.066.666	2.400.000
			3	800.000	1.600.000
			4	533.333	1.133.333
			5	266.666	866.667

**1. Calculate the sum of the year digit**

Menghitung jumlah tahun dengan menambahkan digit tahun dari awal sampai akhir, karena asset berumur 5 tahun maka:

$$5 + 4 + 3 + 2 + 1 = 15$$

**2. Calculate residual value**

*Residual Value* adalah nilai sisa saat aset mengalami *retired*, aset masih mempunyai harga jual. Nilai sisa dihitung 20% dari harga aset saat pertama kali dibeli, sehingga

$$\text{Rp } 5.000.000 \times 20\% = \text{Rp } 1.000.000.000$$

**3. Calculate the depreciable amount**

depreciable amount adalah harga asset yang akan di depresiasi, pada metode ini tidak menggunakan harga asli pembelian untuk dihitung penurunannya

$$\text{Rp. } 5.000.000 - \text{Rp } 1.000.000 = \text{Rp. } 4.000.0000$$

**4. Calculate undepreciated useful life**

Langkah selanjutnya adalah menyesuaikan umur asset dengan depresiasi nya.

$$\text{Year } 1 = 5$$

$$\text{Year } 2 = 4$$

$$\text{Year } 3 = 3$$

$$\text{Year } 4 = 2$$

$$\text{Year } 5 = 1$$

**5. Calculate depreciation expense**

Perhitungan penurunan dilakukan sesuai dengan *undepreciated useful life*.

$$\text{Year } 1 = \frac{5}{15} \times \text{Rp. } 4.000.000 = \text{Rp. } 1.333.333$$

$$\text{Year } 2 = \frac{4}{15} \times \text{Rp. } 4.000.000 = \text{Rp. } 1.066.000$$

$$\text{Year } 3 = \frac{3}{15} \times \text{Rp. } 4.000.000 = \text{Rp. } 800.000$$

$$\text{Year 4} = \frac{2}{15} \times \text{Rp. 4.000.000} = \text{Rp. 533.333}$$

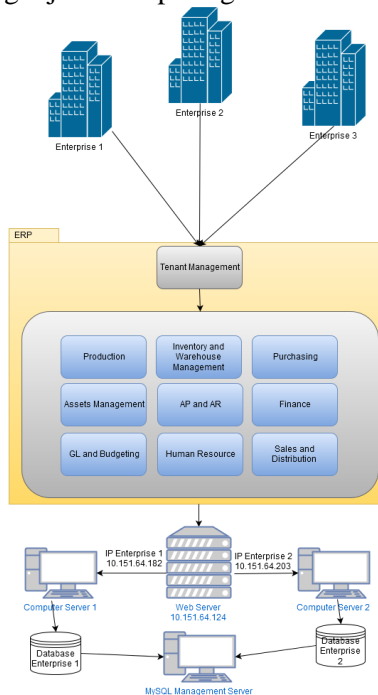
$$\text{Year 5} = \frac{1}{15} \times \text{Rp. 4.000.000} = \text{Rp. 266.666}$$

## 6. Update Book Value Start Year

Book Value Start Year merupakan harga barang setelah mengalami proses depresiasi. Harga awal dikurangi dengan nilai hasil proses perhitungan selanjutnya merupakan harga aset yang baru  $\text{Rp.5.000.000} - \text{Rp. 1.333.333} = \text{Rp 3.666.667,-}$  Update dilakukan setiap tahun

### 3.5.5 Perancangan Basis Data Terdistribusi

Perangkat lunak yang akan dibangun menggunakan basis data terdistribusi yang dijabarkan pada gambar 3.25

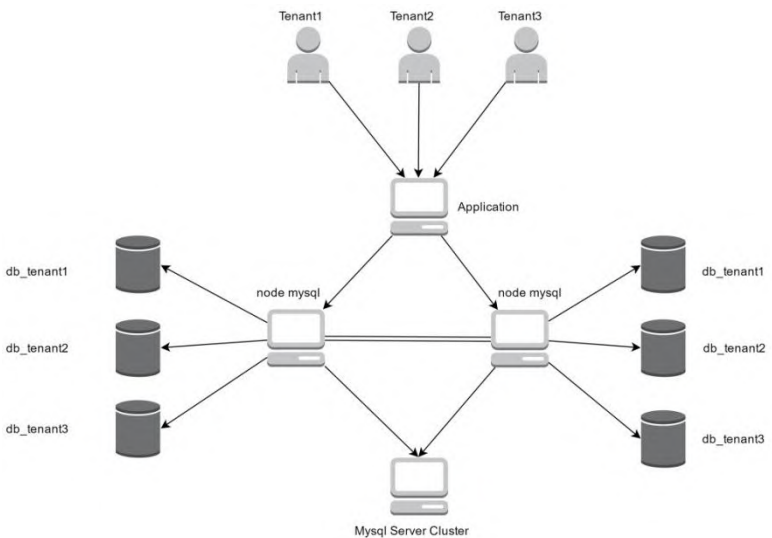


Gambar 3.25 Diagram Basis Data Terdistribusi

Pada gambar diatas menjelaskan bahwa perusahaan 1, perusahaan 2, dan perusahaan 3 dapat menggunakan ERP secara bersamaan. Setiap perusahaan mempunyai 2 database yang telah direplikasi yaitu *Computer Server 1* dengan IP Enterprise 1, 10.151.64.182 dan *Computer Server 2* dengan IP Enterprise 2, 10.151.64.203. komputer 3 berfungsi sebagai server cluster untuk melakukan replikasi, sehingga apabila database 1 atau *Computer Server 1* dalam keadaan mati, sistem ERP masih tetap berjalan.

### 3.5.6 Perancangan *multi-tenancy*

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



Gambar 3.26 Perancangan Multi-tenancy

Gambar 3.26 diatas menerangkan bagaimana alur *multi-tenancy* yang terjadi pada sistem ERP. Pembagian node mysql



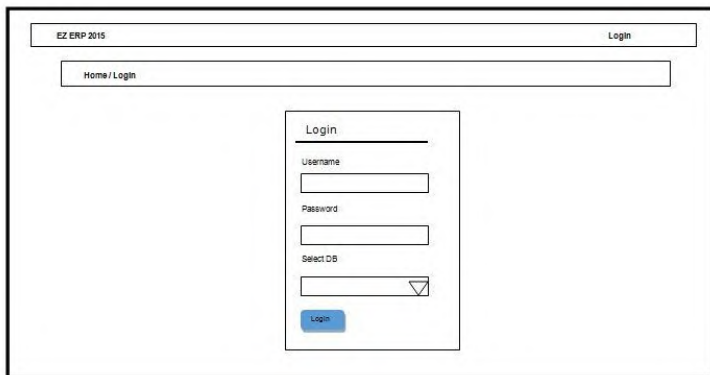
sudah dijelaskan pada sub-bab 3.2.1 yaitu dengan cara clustering menjadi 2 database yang berbeda. Basis data terdistribusi merupakan penunjang dari multi-tenancy, yang berfungsi untuk mengatur session setiap perusahaan yang akan mengakses sistem ERP.

### 3.5.7 Perancangan RBAC (Role Base Acces Control)

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

#### 3.5.7.1 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 user sedang berada pada halaman tertentu, dan di bagian tengah terdapat form login yang memiliki username, password, dan pemilihan basis data yang digunakan sebagai input. Gambar 3.27 merupakan rancangan yang akan digunakan untuk proses Login



The image shows a wireframe of a login page for 'EZ ERP 2016'. At the top, there is a header bar with 'EZ ERP 2016' on the left and 'Login' on the right. Below the header is a breadcrumb trail 'Home / Login'. The main content area contains a 'Login' form with the following elements: a 'Username' input field, a 'Password' input field, a 'Select DB' dropdown menu, and a blue 'Login' button.

Gambar 3.27 Perancangan antarmuka Login

### 3.5.7.2 Perancangan antarmuka *add user*

Agar bisa mendapatkan hak akses maka admin perlu mendaftarkan pengguna. Gambar 3.28 merupakan rancangan antarmuka untuk menambah pengguna.

The image shows a web application interface for 'SZ ERP 2015'. A modal window titled 'Add New User' is centered on the screen. The modal contains three input fields: 'Username', 'Email', and 'Password'. Below these fields is a blue button. The background shows a navigation bar with 'Home/Login' and 'Login' links, and a main content area with a grid of placeholder boxes.

Gambar 3.28 Perancangan antarmuka add user

### 3.5.8 Perancangan antarmuka sistem

Perancangan antarmuka merupakan hal yang penting dalam melakukan perancangan aplikasi. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan-kemudahan dan tampilan yang rapi dan menarik bagi penggunanya.

#### 3.5.8.1 Perancangan antarmuka index

Rancangan antarmuka ini digunakan untuk kasus penggunaan sebelum memilih aksi menambah (*create*), menyunting (*edit*) dan menghapus (*delete*). Perancangan antarmuka index dapat dijelaskan dengan gambar 3.29 dibawah ini:

The diagram shows a web interface layout. On the left is a vertical sidebar with five empty rectangular boxes. The main content area contains: a wide horizontal input field at the top; two smaller rectangular buttons below it, labeled '1' and '2'; and a table with five columns and two rows below that. A callout '3' points to the bottom-left corner of the table.

Gambar 3.29 Perancangan antarmuka index

Penjelasan masing-masing angka pada Gambar diatas adalah sebagai berikut:

1. Tombol Create digunakan untuk menambah transaksi seperti *requisition*, *maintenance*, dsb.
2. Tombok Advance Search untuk mencari transaksi sesuai dengan keinginan user
3. Tombol aksi untuk menyunting dan menghapus

### 3.5.8.2 Perancangan antarmuka create requisition

Rancangan antarmuka ini digunakan untuk kasus penggunaan Create Requisition. Perancangan antarmuka Create Requisition dapat dijelaskan dengan gambar 3.30 dibawah ini

The diagram shows a web interface layout for creating a requisition. On the left is a vertical sidebar with five empty rectangular boxes. The main content area contains: a wide horizontal input field labeled '1'; two smaller rectangular buttons below it, labeled '2' and '3'; two more smaller rectangular buttons below that, labeled '4' and '5'; a large rectangular input field labeled '6' below that; and a green rectangular button labeled '7' at the bottom.

Gambar 3.30 perancangan antarmuka create requisition

Berikut penjelasan masing-masing angka pada gambar diatas:

1. Input untuk Requisition Number
2. Dropdown untuk status
3. Input untuk Name
4. Input untuk Shipto
5. Input untuk Need by date
6. Input Line atau barang yang dibutuhkan
7. Tombol Create

### 3.5.8.3 Perancangan antarmuka create asset active

Rancangan antarmuka ini digunakan untuk kasus penggunaan Create Asset Active. Perancangan antarmuka Create Asset Active dapat dijelaskan dengan gambar 3.31 dibawah ini

The image shows a screenshot of a web application interface for creating an active asset. It features a grid of 18 numbered input fields arranged in four rows. The first row contains two wide input fields (1 and 2). The second row contains three input fields (3, 4, and 5). The third row contains three input fields (6, 7, and 8). The fourth row contains three input fields (9, 10, and 11). The fifth row contains three input fields (12, 13, and 14). The sixth row contains three input fields (15, 16, and 17). Below the grid is a green button labeled '18'. The interface is enclosed in a black border.

Gambar 3.31 Perancangan antarmuka create asset active

Berikut penjelasan masing-masing angka pada gambar diatas:

1. *Dropdown Iwm Item Detail*
2. *Input Asset Name*
3. *Input Price*

4. *Input Bookvalue Start year*
5. *Input Asset Date Register*
6. *Input Periode*
7. *Input Asset Description*
8. *Input Asset type*
9. *Dropdown Asset Condition*
10. *Dropdown Asset Group*
11. *Input Max Usage*
12. *Input Time to maintenance*
13. *Input Usage*
14. *Input time per production*
15. *Input Capacity*
16. *Input Distance to Maintenance*
17. *Input UOM*
18. *Tombol Create*

#### 3.5.8.4 Perancangan antarmuka create asset depreciation

Rancangan antarmuka ini digunakan untuk kasus penggunaan Create Asset Depreciation. Perancangan antarmuka Create Asset Depreciation dapat dijelaskan dengan gambar 3.32 dibawah ini

	1
	2
	3
	4
	5
	6
	7
	8
	9

Gambar 3.32 perancangan antarmuka create asset depreciation

Berikut penjelasan masing-masing angka pada gambar diatas:

1. *Input Asset Number*
2. *Input Useful Life*
3. *Dropdown Depreciation Method*
4. *Textbox Asset Price*
5. *Textbox Book value start year*
6. *Input Asset Rate*
7. *Input Depreciation Date*
8. *Input status*
9. Tombol *create*

### 3.5.8.5 Perancangan antarmuka create maintenance

Rancangan antarmuka ini digunakan untuk kasus penggunaan Create Asset Maintenance. Perancangan antarmuka Create Asset Maintenance dapat dijelaskan dengan gambar 3.33 dibawah ini

	1
	2
	3
	4
	5
	6
	7

Gambar 3.33 perancangan antarmuka create asset depreciation

Berikut penjelasan masing-masing angka pada gambar diatas:

1. *Input Asset Number*
2. *Input Type Maintenance*
3. *Input Maintenance Cost*
4. *Input Maintenance Note*

5. *Input Start date*
6. *Input Finish date*
7. *Tombol create*

### 3.5.8.6 Perancangan antarmuka create asset retirement

Rancangan antarmuka ini digunakan untuk kasus penggunaan Create Asset Retirement. Perancangan antarmuka Create Asset Retirement dapat dijelaskan dengan gambar 3.34 dibawah ini

	1
	2
	3
	4
	5

Gambar 3.34 perancangan antarmuka create asset retirement

Berikut penjelasan masing-masing angka pada gambar diatas:

1. *Input Asset Number*
2. *Input Date Retired*
3. *Input Note*
4. *Input Status*
5. *Tombol Create*

### 3.5.8.7 Perancangan antarmuka Report Tahunan

Rancangan antarmuka ini digunakan untuk kasus penggunaan Report Tahunan. Perancangan antarmuka Report Tahunan dapat dijelaskan dengan gambar 3.27 dibawah ini

The wireframe shows a sidebar on the left with four menu items. The main content area contains a search bar at the top, followed by two buttons labeled '1' and '2'. Below these is a table with five columns and one row.

Gambar 3.35 Perancangan antarmuka report tahunan

Berikut penjelasan masing-masing angka pada gambar

diatas:

1. Dropdown tahun
2. Tombol Submit

### 3.5.8.8 Perancangan antarmuka Melihat Data

Rancangan antarmuka ini digunakan untuk kasus penggunaan Melihat data. Perancangan antarmuka melihat data dapat dijelaskan dengan gambar 3.36 dibawah ini

The wireframe shows a sidebar on the left with four menu items. The main content area contains a search bar at the top, followed by a table with five columns and one row.

Gambar 3.36 perancangan antarmuka melihat data



## **BAB IV IMPLEMENTASI SISTEM**

Bab ini membahas implementasi dari perancangan sistem ERP 2015. 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 SQL Server 2012 R2.
2. PC untuk server menggunakan AMD A6-3400M APU with Radeon HD Graphics @1.40Ghz, RAM 4GB dengan Sistem Operasi Windows 7 Ultimate.

StarUML untuk pembuatan diagram, Power Designer 12.5, Sublime sebagai teks editor, Pencil untuk pembuatan desain antarmuka, Power Designer untuk pembuatan CDM dan PDM.

Mozilla Firefox 46.0.1 sebagai antarmuka untuk pengujian aplikasi klien

### **4.2 Implementasi Basis Data Terdistribusi**

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 node 1 dan node 2**

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

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

Kode Sumber 4.2 Mengubah lokasi direktori, mengubah arsip, dan menciptakan Symlink

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 direktori

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

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

Kode Sumber 4.4. Mengatur perizinan server MySQL

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

#### 4.2.2 Instalasi Data dan SQL node pada node 1 dan node 2

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 direktori

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 direktori

### 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

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 node 3 (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

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 MySQL Cluster

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. Mengaktifkan MySQL

### 4.3 Implementasi Role Base Access 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. Apabila proses migrasi berhasil dilakukan, maka akan muncul keterangan “*migrate successfully*” dan akan muncul tabel *user* di dalam basis data.

#### 4.3.2. Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna

Pada tahap ini dibutuhkan 4 tabel autentifikasi yang terdiri dari (yii2 rbac):

1. Tabel Item, tabel yang menyimpan daftar otorisasi, standar nama tabel adalah *auth\_item*.
2. Tabel Child, tabel yang menyimpan hirarki daftar otoristas, standar nama tabel adalah *auth\_item\_child*.
3. Tabel Assigment, tabel yang menyimpan penetapan user untuk daftar otorisasi, standar nama tabel adalah *auth\_assignment*.
4. Tabel Rule, tabel yang menyimpan aturan-aturan autentifikasi, standar nama tabel adalah *auth\_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 autentifikasi

Apabila proses migrasi berhasil yang maka akan terbuat 4 tabel pada database, yaitu tabel *auth\_assignment*, tabel *auth\_item*, tabel *auth\_item\_child* dan tabel *auth\_rule*.

### 4.3.3. Membuat 4 Modul Admin dan Konfigurasi Autentifikasi

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: menambahkan admin ke daftar modul, dengan tujuan menambah akses ke modul admin, menambahkan Auth Manager pada component sebagai autentifikasi di Yii2 dan mengatur role default sebagai *guest*, dan menambahkan Session Time Out untuk mengatur durasi time out session selama 5 menit atau 300 detik. Masing –masing konfigurasi tersebut ditunjukkan pada Kode Sumber 4.15.

```
$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 kerangka kerja Yii2.

#### 4.3.5. Menambahkan Kode pada 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, dan 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 manager
    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. Menambahkan data ke tabel auth\_item

Kode Sumber 4.16 menunjukkan fungsi untuk menambahkan data ke tabel auth\_item yang telah dibuat sebelumnya pada subbab 4.3.2. Setiap kode yang terdapat di dalam fungsi mendaftarkan setiap modul yang ada pada sistem ERP dan juga hak akses sebagai staf atau manajer.

```

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. Menambahkan data ke tabel auth\_item\_child

Kode Sumber 4.17 menunjukkan fungsi untuk menambahkan data ke tabel auth\_item\_child yang telah dibuat sebelumnya pada subbab 4.3.2. Setiap kode yang terdapat di dalam fungsi bertujuan untuk mengatur user admin untuk dapat mengakses setiap modul yang telah terdaftar di dalam tabel auth\_item.

```

public function actionAuthAssignment()
{

```

```

    $auth = Yii::$app->authManager;

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

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

```

Kode Sumber 4.18. Menambahkan data ke tabel auth\_assignment

Kode Sumber 4.18 menunjukkan fungsi untuk menambahkan data ke tabel auth\_assignment yang telah dibuat sebelumnya pada subbab 4.3.2. Setiap kode yang terdapat di dalam fungsi bertujuan untuk menetapkan user admin yang memiliki id dengan nilai 1 pada tabel user dapat mengakses semua modul yang terdaftar di dalam tabel auth\_item

## 4.4 Implementasi Multi-tenancy

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

### 4.4.1. Membuat Halaman Depan Tenant

Pada implementasi pembuatan halaman tenant, dilakukan pembuatan halaman tenant secara sederhana, kemudian ditambahkan pembuatan basis data 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');
$databse = Yii::$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
$insert = "insert into tenant
(tenant,database_name,created_at) values('".$tenant."','".$
.$databse."','".$NOW())";

```

```

if($tenant != "" && $database != ""){
    $koneksi-db->query($insert);
}

// memutuskan koneksi ke server
$koneksi-db->close();

```

Kode Sumber 4.19. Pembuatan halaman depan tenant

#### 4.4.2. Menambahkan Basis Data untuk Tenant Baru

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

```

shell> mysql -u root -p;
shell> create database <nama database>;
shell> use <nama database>;'
shell> source /tmp/mysql-dump/final-db.sql

```

Kode Sumber 4.20. Penambahan basis data untuk 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;
$src =@fsockopen($host,$port,$errCode,$errStr,$waitTimeout
InSeconds);
if(is_resource($src)){
    $_SESSION['dbserver_ip'] = "10.151.64.182";

    // session multitenancy berdasarkan nama basis data
    $dbname = isset($_SESSION['database_name'])?$_SESSION
['database_name'] : 'test';
} else {
    $_SESSION['dbserver_ip'] = "10.151.64.203";
}

```

```

$connection = [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host='.$_SESSION['dbserver_ip'].';
    dbname='.$_dbname.'',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8'
];
return $connection;

```

Kode Sumber 4.21. Login tenant

## 4.5 Implementasi Program pada Modul Asset Management

Pada bagian ini akan dijelaskan mengenai implementasi yang terdapat pada modul *Asset Management* yang terbagi menjadi tampilan halaman, diantaranya *Asset Type*, *Depreciation Method*, *Asset Group*, *Asset Requisition*, *Asset Active*, *Asset Depreciation*, *Asset Maintenance*, *Asset Retirement*, *Report*

### 4.5.1. Menambah Asset Type

Halaman ini menampilkan informasi mengenai penambahan tipe aset. Pengaturan tampilan tersebut diatur di kelas *AssetAccountController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.22

```

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

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

```

Kode Sumber 4.22 Menambah Asset Type

### 4.5.2. Menyunting Asset Type

Halaman ini menampilkan informasi mengenai penyuntingan tipe aset. Pengaturan tampilan tersebut diatur di kelas *AssetAccountController* fungsi *actionUpdate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.23

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

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

Kode Sumber 4.23 menyunting asset type

### 4.5.3. Menghapus Asset Type

Halaman ini menampilkan informasi mengenai penghapusan tipe aset. Pengaturan tampilan tersebut diatur di kelas *AssetAccountController* fungsi *actionDelete()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.24

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

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

Kode Sumber 4.24 menghapus asset type

### 4.5.4. Menambah Depreciation Method

Halaman ini menampilkan informasi mengenai penambahan tipe depresiasi aset. Pengaturan tampilan tersebut



diatur di kelas *AssetDepreciationMethodController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.25

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

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

Kode Sumber 4.25 menambah depreciation method

#### 4.5.5. Menyunting Depreciation Method

Halaman ini menampilkan informasi mengenai penyuntingan tipe depresiasi aset. Pengaturan tampilan tersebut diatur di kelas *AssetDepreciationMethodController* fungsi *actionUpdate()*. Kode untuk menampilkan seluruh transaksi tersebut 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_method]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}
```

Kode Sumber 4.26 menyunting depreciation method

#### 4.5.6. Menghapus Depreciation Method

Halaman ini menampilkan informasi mengenai penghapusan tipe depresiasi aset. Pengaturan tampilan tersebut diatur di kelas *AssetDepreciationMethodController* fungsi *actionDelete()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.27

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

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

Kode Sumber 4.27 menghapus depreciation method

#### 4.5.7. Menambah Asset Group

Halaman ini menampilkan informasi mengenai penambahan tipe kelompok aset. Pengaturan tampilan tersebut diatur di kelas *AssetGroupController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber

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

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

Kode Sumber 4.28 Menambah Asset group

#### 4.5.8. Menyunting Asset Group

Halaman ini menampilkan informasi mengenai penyuntingan tipe kelompok aset. Pengaturan tampilan tersebut

diatur di kelas *AssetGroupController* fungsi *actionUpdate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.29

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

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

Kode Sumber 4.29 menyunting asset group

#### 4.5.9. Menghapus Asset Group

Halaman ini menampilkan informasi mengenai penghapusan tipe kelompok aset. Pengaturan tampilan tersebut diatur di kelas *AssetGroupController* fungsi *actionDelete()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.30

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

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

Kode Sumber 4.30 menghapus asset group

#### 4.5.10. Menambah Asset Requisition

Halaman ini menampilkan informasi mengenai permintaan penambahan aset. Pengaturan tampilan tersebut diatur di kelas *AssetRequisitionController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.31

```

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

    if ($model->loadAll(Yii::$app->request->post()) &&
$model->saveAll()) {
        // $pur=AmAssetReqH::find()-
>where(['req_number' =>$model['req_number']])->One();
        $item = array();
        foreach ($model['amAssetReqLs'] as $key =>
$value) {
            $data = array(
                'item_id' =>
$value['item_number'],
                'item_qty' => $value['quantity'],
            );
            // print_r($value['item_number']);
            array_push($item, $data);

AccTypeController::actionNewasset($model['asset_type'],
$value['item_number']);

        }
        $date = $model['need_by_date'];

IwmgoodissuehController::actionCreateIssue('Asset',
$model['id'], $model['need_by_date'], $item);

        $newId = str_pad($model['id'],14,"REQ-
ASSET-000",STR_PAD_LEFT);

        $data = AmAssetReqH::findOne(['id' =>
$model['id']]);
        $data->req_number = $newId;
        $data->update();
        return $this->redirect(['view', 'id' =>
$model->id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4.31 menambah asset requisition

#### 4.5.11. Menyunting Asset Requisition

Halaman ini menampilkan informasi mengenai penyuntingan penambahan aset. Pengaturan tampilan tersebut diatur di kelas *AssetRequisitionController* fungsi *actionUpdate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.32

```

*/
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 4.32 menyunting asset requisition

#### 4.5.12. Menghapus Asset Requisition

Halaman ini menampilkan informasi mengenai penghapusan penambahan aset. Pengaturan tampilan tersebut diatur di kelas *AssetRequisitionController* fungsi *actionDelete()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.33

```

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

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

```

Kode Sumber 4.33 menghapus asset requisition

#### 4.5.13. Menambah Asset Active

Halaman ini menampilkan informasi mengenai penambahan aset aktif. Pengaturan tampilan tersebut diatur di

kelas *AssetActiveController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.44

```

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

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

        if($model['asset_type'] == 5)
        {

            SdShiptdetailController::getTransport($model['id_asset'],
            $model['asset_name'], $model['capacity_per_production']);
            $newId
            str_pad($model['id_asset'],7,"VHC0000",STR_PAD_LEFT);

            $data = AmAssetActive::findOne(['id_asset'
            => $model['id_asset']]);
            $data->asset_code = $newId;
            $data->update();
        }
        else {
            $newId
            str_pad($model['id_asset'],8,"ASSET000",STR_PAD_LEFT);

            $data = AmAssetActive::findOne(['id_asset'
            => $model['id_asset']]);
            $data->asset_code = $newId;
            $data->update();
        }
        return $this->redirect(['view', 'id' =>
        $model->id_asset]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

```

Kode Sumber 4.34 menambah asset active

#### 4.5.14. Menyunting Asset Active

Halaman ini menampilkan informasi mengenai penyuntingan aset aktif. Pengaturan tampilan tersebut diatur di

kelas *AssetActiveController* fungsi *actionUpdate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.35

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

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

        if($model['asset_type'] == 5)
        {

            SdShipdetailController::getTransportStatus($model['id_asset'],
                $model['asset_status'], $model['asset_availability']);
        }

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

Kode Sumber 4.35 menyunting asset active

#### 4.5.15. Menghapus Asset Active

Halaman ini menampilkan informasi mengenai penghapusan aset aktif. Pengaturan tampilan tersebut diatur di kelas *AssetActiveController* fungsi *actionDelete()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.36

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

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

Kode Sumber 4.36 menghapus asset active

#### 4.5.16. Menambah Asset Depreciation

Halaman ini menampilkan informasi mengenai penambahan penyusutan aset. Perhitungan penurunan harga juga terdapat pada proses ini, saat pengguna melakukan create maka otomatis akan melakukan perhitungan dengan metode yang telah dipilih, fungsi untuk setiap metode terdapat pada kode sumber, yaitu *function straightLine()* untuk metode garis lurus *functionReducingBalance()* untuk metode penurunan berdasarkan harga pasar *functionsumOfTheYear()* penurunan berdasarkan tahun dan *functionUnitOfActivity()* penurunan berdasarkan aktivitas. Pengaturan tampilan tersebut diatur di kelas *AssetDepreciationController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.37

```

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

    if ($model->loadAll(Yii::$app->request->post()) &&
    $model->saveAll()) {
        $id = $model['id'];
        $aset = $model['asset_number'];

        $newId
        str_pad($model['id'],8,"DEPR0000",STR_PAD_LEFT);

        $data = AmAssetDepreciation::findOne(['id' =>
        $model['id']]);
        $data->asset_depreciation_code = $newId;
        $data->update();

        if($model['method_name'] == 1)
        {
            $this->straightLine($id,
            $model['useful_life'],          $model['asset_cost'],
            $model['salvage_value'], $aset, $model['price']);
        }
        elseif($model['method_name'] == 2)
        {
            $this->ReducingBalance($id,
            $model['useful_life'],          $model['asset_cost'],
            $model['salvage_value'], $model['price']);
        }
    }
}

```



```

        elseif ($model['method_name'] == 3)
        {
            $this->sumOfTheYear($id,
                $model['useful_life'],
                $model['asset_cost'],
                $model['salvage_value'],
                $model['price']);
        }
        elseif ($model['method_name'] == 4)
        {
            $active = AmAssetActive::find()-
>select('am_asset_active.*')->from('am_asset_active,
am_asset_depreciation')->where('am_asset_active.id_asset =
am_asset_depreciation.asset_number AND
am_asset_depreciation.id='.$model['id'])->One();

            $model->salvage_value =
$active['asset_price']*0.2;
            $model->depreciation_expense =
(($active['asset_price']
$active['asset_price']*0.2)/$model['useful_life'])*1;

            $model->book_value_end_year =
$active['bookvalue_start_year'] -
$model->depreciation_expense;
            $model->depreciation_date = date('Y-m-d');
            $model->created_at = date('Y-m-d');
            $model->updated_at = date('Y-m-d');
            $model->update();

            $active->bookvalue_start_year = $model-
>book_value_end_year;
            $active->update();

            //tambahan code

SdShipdetailController::actionDerpreciationCost($aset,
$model->depreciation_expense);
        }

JournalHeaderController::actionDepreciation($id);

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

public function straightLine($id, $useful life,

```

```

$asset_cost, $salvage_value, $aset, $price)
{
    $acc = AmAssetDepreciation::find()->select('*')-
>where('asset_number ='.$aset->count();
    $data = AmAssetDepreciation::findOne($id);

    $sl = (($price - $salvage_value) / $useful_life);
    $book_val = ($asset_cost - $sl);

    $hitung = $acc*$sl;
    $data->accumulated_depreciation = $hitung;

    $data->depreciation_expense = $sl;
    $data->book_value_end_year = $book_val;
    $data->update();

    $register = AmAssetActive::find()-
>select('am_asset_active.*')->from('am_asset_active,
am_asset_depreciation')->where('am_asset_active.id_asset =
am_asset_depreciation.asset_number AND
am_asset_depreciation.id='.$id->One();
    $bookEnd = $data->book_value_end_year;
    $register->bookvalue_start_year = $bookEnd;
    $register->update();

}

public function unitOfActivity($code, $distance, $time,
$date)
{
    $model = AmAssetDepreciation::find()-
>where(['asset_number' => $code])->orderBy('id DESC')-
>One();

    $active = AmAssetActive::find()-
>select('am_asset_active.*')->from('am_asset_active,
am_asset_depreciation')->where('am_asset_active.id_asset =
am_asset_depreciation.asset_number AND
am_asset_depreciation.id='.$model['id']->One();

    $new_model = new AmAssetDepreciation();

    $new_model->price = $active['asset_price'];
    $new_model->asset_cost =
    $active['bookvalue_start_year'];
    $new_model->salvage_value =
    $active['asset_price']*0.2;
    $new_model->depreciation_expense =
    (($active['asset_price'] -
    $active['asset_price']*0.2)/$model['useful life'])*$distanc

```

```

e;
    $new_model->book_value_end_year =
$active['bookvalue_start_year'] - $new_model-
>depreciation_expense;
    $new_model->asset_number = $model['asset_number'];
    $new_model->method_name = $model['method_name'];
    $new_model->asset_rate = $model['asset_rate'];
    $new_model->depreciation_date = $date;
    $new_model->useful_life = $model['useful_life'];
    $new_model->depreciation_status =
$model['depreciation_status'];

    $active->bookvalue_start_year = $new_model-
>book_value_end_year;
    $active->update();

    $new_model->save();

    $newId =
str_pad($model['id'],8,"DEPR0000",STR_PAD_LEFT);

    $data = AmAssetDepreciation::findOne(['id' =>
$new_model['id']]);
    $data->asset_depreciation_code = $newId;
    $data->update();

    // $lastinsertedid = $new_model;

JournalHeaderController::actionDepreciation($new_model-
>id);

}

public function sumOfTheYear($id, $useful_life,
$asset_cost, $salvage_value, $price)
{
    $assets = AmAssetDepreciation::find()->where(['id'
=> $id])->One();
    $sum_year = 0;
    for ($x = $useful_life; $x > 0; $x--)
    {
        $sum_year += $x;
    }
    $dep_amount = ($price - $salvage_value);

    $dep_exp = ($useful_life/$sum_year) * $dep_amount;

    $book_val = ($asset_cost - $dep_exp);

```

```

    $assets->depreciation_expense = round($dep_exp);
    $assets->book_value_end_year = round($book_val);
    $assets->update();

    $register = AmAssetActive::find()-
>select('am_asset_active.*')->from('am_asset_active,
am_asset_depreciation')->where('am_asset_active.id_asset =
am_asset_depreciation.asset_number AND
am_asset_depreciation.id='. $id)->One();
    $bookEnd = $assets->book_value_end_year;
    $register->bookvalue_start_year = $bookEnd;
    $register->update();
}

public function ReducingBalance($id, $useful_life,
$asset_cost, $salvage_value, $asset_rate, $price)
{
    $data = AmAssetDepreciation::findOne($id);
    $accdep = 0;
    $nbv = $asset_cost;
    $a = 1;
    for ($x = $useful_life; $x-1>=0; $x++)
    {
        if($x == 1)
        {
            $dep_val = ($asset_cost - $salvage_value) -
$accdep;
            $accdep+=$dep_val; $salvage_value-=1;
        }
        else
        {
            $dep_val = ($nbv - $salvage_value) *
$asset_rate;
            $accdep+=$dep_val;
            $a++;
            $nbv=$asset_cost-$accdep;
        }
    }

    if ($salvage_value>=0 && $accdep<$asset_cost -
$salvage_value)
    {
        $dep_val = ($asset_cost - $salvage_value) -
$accdep;
        $accdep+=$dep_val; $salvage_value-=1;
    }
    //$book_val = ($asset_cost - $dep_val);

    $data->depreciation_expense = round($dep_val);
    $data->book value end year = round($book_val);
}

```

```

        // $data->accumulated_depreciation = round($accdep);
        $data->update();

        //      $register      =      AmAssetActive::find()-
>select('am_asset_active.*')->from('am_asset_active,
am_asset_depreciation')->where('am_asset_active.id_asset =
am_asset_depreciation.asset_number                                AND
am_asset_depreciation.id='.$id)->One();
        // $bookEnd = $data->book_value_end_year;
        // $register->bookvalue_start_year = $bookEnd;
        // $register->update();
    }

```

Kode Sumber 4.37 menambah asset depreciation

#### 4.5.17. Menyunting Asset Depreciation

Halaman ini menampilkan informasi mengenai penyuntingan penyusutan aset aktif. Pengaturan tampilan tersebut diatur di kelas *AssetDepreciationController* fungsi *actionUpdate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.38

```

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 4.38 menyunting asset depreciation

#### 4.5.18. Menghapus Asset Depreciation

Halaman ini menampilkan informasi mengenai penghapusan penyusutan aset aktif. Pengaturan tampilan tersebut diatur di kelas *AssetDepreciationController* fungsi *actionDelete()*.

Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.39

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

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

Kode Sumber 4.39 menghapus asset depreciation

#### 4.5.19. Menambah Asset Maintenance

Halaman ini menampilkan informasi mengenai penambahan pemeliharaan aset aktif. Pengaturan tampilan tersebut diatur di kelas *AssetMaintenanceController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.40

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

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

        $newId =
str_pad($model['id_asset_maintenance'], 8, "MAIN000", STR_PAD_
LEFT);

        $data =
AmAssetMaintenance::findOne(['id_asset_maintenance' =>
$model['id_asset_maintenance']]);
        $data->asset_maintenance_code = $newId;
        $data->update();

        $code = $model['asset_number'];
        $active = AmAssetActive::find()-
>where(['id_asset' => $code])->One();
        $active->asset_availability='MAINTENANCE';

        $active->update();

MaintenanceController::actionAmAssetMaintenance($model-
>id_asset_maintenance, 'insert');
        return $this->redirect(['view', 'id' => $model-
```

```

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

```

Kode Sumber 4.40 menambah asset maintenance

#### 4.5.20. Menyunting Asset Maintenance

Halaman ini menampilkan informasi mengenai penyuntingan pemeliharaan aset aktif. Pengaturan tampilan tersebut diatur di kelas *AssetMaintenanceController* fungsi *actionUpdate()*. Kode untuk menampilkan 4.41

```

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

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

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

```

Kode Sumber 4.41 menyunting asset maintenance

#### 4.5.21. Menghapus Asset Maintenance

Halaman ini menampilkan informasi mengenai penghapusan pemeliharaan aset aktif. Pengaturan tampilan tersebut diatur di kelas *AssetMaintenanceController* fungsi *actionDelete()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.42

```

public function actionDelete($id)
{

```

```

        $this->findModel($id)->deleteWithRelated();

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

```

Kode Sumber 4.42 menghapus asset maintenance

#### 4.5.22. Menambah Asset Retirement

Halaman ini menampilkan informasi mengenai penambahan data aset tidak aktif. Pengaturan tampilan tersebut diatur di kelas *AssetRetirementController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.43

```

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

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

```

Kode Sumber 4.43 menambah asset retirement

#### 4.5.23. Menyunting Asset Retirement

Halaman ini menampilkan informasi mengenai penyuntingan data aset tidak aktif. Pengaturan tampilan tersebut diatur di kelas *AssetRetirementController* fungsi *actionUpdate()*. Kode untuk menampilkan 4.44

```

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

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

```



```

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

```

Kode Sumber 4.44 menyunting asset retirement

#### 4.5.24. Menghapus Asset Retirement

Halaman ini menampilkan informasi mengenai penghapusan data aset tidak aktif. Pengaturan tampilan tersebut diatur di kelas *AssetRetirementController* fungsi *actionDelete()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.45

```

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

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

```

Kode Sumber 4.45 menghapus asset retirement

#### 4.5.25. Melihat Report

Halaman ini menampilkan informasi mengenai pencatatan pengeluaran mengenai aset per tahun. Pengaturan tampilan tersebut diatur di kelas *ReportController* fungsi *actionIndex()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.46

```

class ReportController extends Controller
{
    function actionIndex($year = NULL)
    {
        $nbv = 0;
        $year = Yii::$app->request->post('year');

        $db = Yii::$app->db;
        if ($year == NULL) {
            $year = date('Y');
        }

        $qReport = "SELECT asset_code as code,

```

```

asset_name as name, asset_price as price,
depreciation_expense as dep_exp, maintenance_cost as cost,
asset_periode as periode, depreciation_date as dates FROM
am_asset_active, am_asset_depreciation,
am_asset_maintenance WHERE
am_asset_depreciation.asset_number
=
am_asset_active.id_asset and am_asset_active.id_asset =
am_asset_maintenance.asset_number and
YEAR(depreciation_date) = $year";
$report = $db->createCommand($qReport)-
>queryAll();

$qYears = "SELECT DISTINCT
YEAR(depreciation_date) as year FROM
am_asset_depreciation";
$year = $db->createCommand($qYears)-
>queryAll();

foreach ($report as $key => $value) {
    $report[$key]['nbv'] =
$value['price'] - $value['dep_exp'] - $value['cost'];
    //$nbv += $report[$key]['nbv'];
}

return $this->render('index', [
    'nbv' => $nbv,
    'report' => $report,
    'year' => $year,
]);
}
}

```

Kode Sumber 4.46 melihat report

## 4.6 Implementasi Antarmuka Pengguna

Pada bagian ini akan dibahas implementasi antarmuka pengguna berdasarkan rancangan antarmuka yang telah dibahas pada bab 3. Lapisan antarmuka merupakan lapisan yang bertugas mengatur tampilan sistem agar dapat berinteraksi dengan pengguna. Lapisan antarmuka berinteraksi langsung dengan lapisan pengendali (*controller*).

### 4.6.1 Antarmuka Menambah Asset Type

Pada antarmuka ini pengguna dapat menambah tipe aset dengan mengisi form yang telah disediakan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.1

The screenshot shows the 'Create Am Asset Type' form in the EZERP2015 application. The form is located on the right side of the dashboard, with a breadcrumb trail: Home > Am Asset Type > Create Am Asset Type. The form fields are:

- Id Account:** A text input field with the value '1'.
- Account Number:** A text input field with the value '220130'.
- Account Name:** A text input field with the value 'Land'.
- Account Note:** A text input field with the value '-'.

A green 'Create' button is positioned below the 'Account Note' field. On the left side, there is a sidebar menu with options: Dashboard, Master, Requestion, Asset Active, Asset Depreciation, Asset Maintenance, Asset Retirement, and Report.

Gambar 4.1 Menambah Asset Type

#### 4.6.2 Antarmuka Menyunting Asset Type

Pada antarmuka ini pengguna dapat mengubah data tipe aset. Tampilan antarmuka ini dapat dilihat pada Gambar 4.2

The screenshot shows the 'Update Am Asset Type' form in the EZERP2015 application. The form is located on the right side of the dashboard, with a breadcrumb trail: Home > Am Asset Type > 1 > Update. The form fields are:

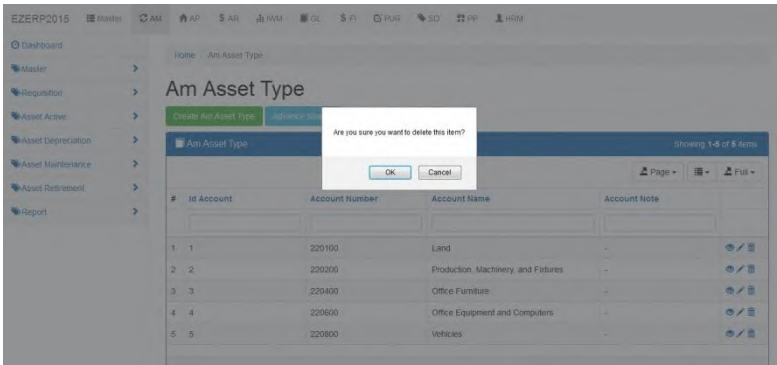
- Id Account:** A text input field with the value '1'.
- Account Number:** A text input field with the value '220130'.
- Account Name:** A text input field with the value 'Land'.
- Account Note:** A text input field with the value '-'.

A blue 'Update' button is positioned below the 'Account Note' field. On the left side, there is a sidebar menu with options: Dashboard, Master, Requestion, Asset Active, Asset Depreciation, Asset Maintenance, Asset Retirement, and Report.

Gambar 4.2 Menambah Asset Type

#### 4.6.3 Antarmuka Menghapus Asset Type

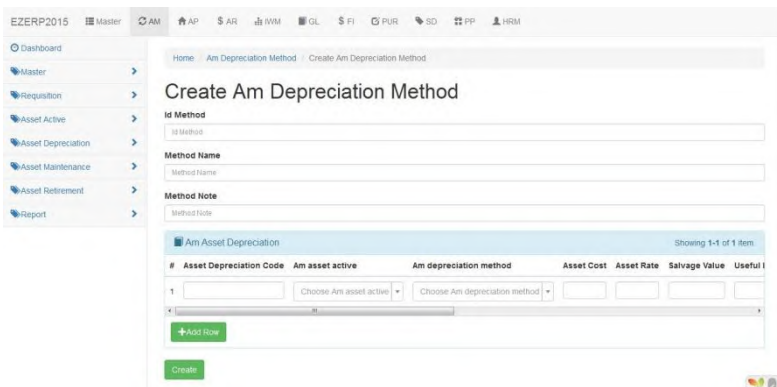
Pada antarmuka ini pengguna dapat menghapus data tipe aset yang sudah tidak dibutuhkan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.3



Gambar 4.3 Mengapus Asset type

#### 4.6.4 Antarmuka Menambah Depreciation Method

Pada antarmuka ini pengguna dapat menambah tipe depresiasi aset dengan mengisi form yang telah disediakan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.4



Gambar 4.4 menambah depreciation method

#### 4.6.5 Antarmuka Menyunting Depreciation Method

Pada antarmuka ini pengguna dapat mengubah data tipe depresiasi aset. Tampilan antarmuka ini dapat dilihat pada Gambar 4.5

Update Am Depreciation Method: 1

Id Method: 1

Method Name: Straight Line Depreciation

Method Note: -

#	Asset Depreciation Code	Am asset active	Am depreciation method	Asset Cost	Asset Rate	Salvage Value	Useful Life	Depreciation E
1		1	1	1717950		3435900	10	1374390
2		1	1	1580514		3435900	10	1236924
3	dep01	3	1	3984950		79299000	10	

Gambar 4.5 Menyunting depreciation method

#### 4.6.6 Antarmuka Menghapus Depreciation Method

Pada antarmuka ini pengguna dapat menghapus data tipe depresiasi aset yang sudah tidak dibutuhkan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.6

Am Depreciation Method

Are you sure you want to delete this item?

OK Cancel

#	Id Method	Method Name	Method Note
1	1	Straight Line Depreciation	-
2	2	Reducing Balance Depreciation	-
3	3	Sum of the year Depreciation	-
4	4	Units of Activity Depreciation	-

Gambar 4.6 menghapus depreciation method

#### 4.6.7 Antarmuka Menambah Asset Group

Pada antarmuka ini pengguna dapat menambah tipe kelompok aset dengan mengisi form yang telah disediakan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.7

The screenshot shows the 'Create Am Asset Group' form in the EZERP2015 system. The form is located in the 'Am Asset Group' section and includes the following fields:

- Id Asset Group:** A text input field.
- Group Name:** A text input field.
- Method Rate:** A text input field.
- Method Estimated Life:** A text input field.
- Method Note:** A text input field.

A green 'Create' button is located at the bottom of the form.

Gambar 4.7 menambah asset group

#### 4.6.8 Antarmuka Menyunting Asset Group

Pada antarmuka ini pengguna dapat mengubah data tipe kelompok aset. Tampilan antarmuka ini dapat dilihat pada Gambar 4.8

The screenshot shows the 'Update Am Asset Group: 1' form in the EZERP2015 system. The form is located in the 'Am Asset Group' section and includes the following fields:

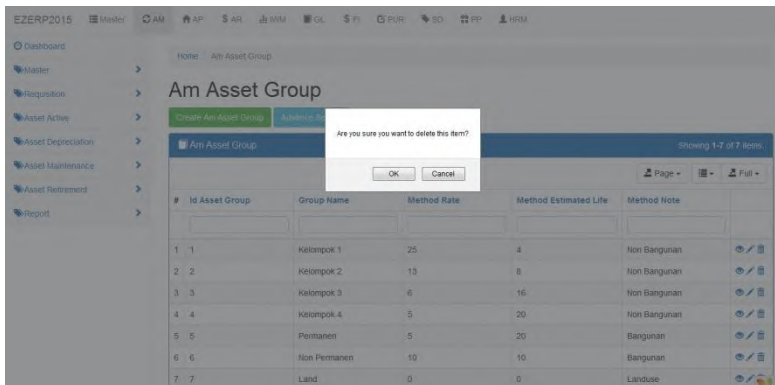
- Id Asset Group:** A text input field with the value '1'.
- Group Name:** A text input field with the value 'Kelompok 1'.
- Method Rate:** A text input field with the value '25'.
- Method Estimated Life:** A text input field with the value '4'.
- Method Note:** A text input field with the value 'Non Bangunan'.

A blue 'Update' button is located at the bottom of the form.

Gambar 4.8 menyunting asset group

#### 4.6.9 Antarmuka Menghapus Asset Group

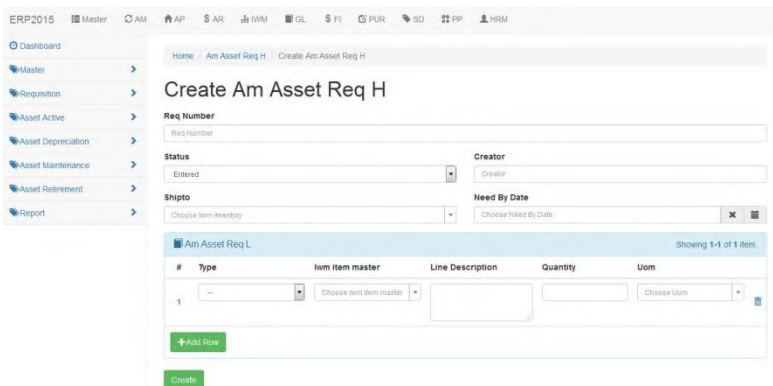
Pada antarmuka ini pengguna dapat menghapus data tipe kelompok aset yang sudah tidak dibutuhkan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.9



Gambar 4.9 menghapus asset group

#### 4.6.10 Antarmuka Menambah Asset Requisition

Pada antarmuka ini pengguna dapat melakukan permintaan penambahan aset dengan mengisi form yang telah disediakan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.10



Gambar 4.10 menambah asset requisition

#### 4.6.11 Antarmuka Menyunting Asset Requisition

Pada antarmuka ini pengguna dapat mengubah permintaan penambahan aset. Tampilan antarmuka ini dapat dilihat pada Gambar 4.11

ERP2015 Master AM AP AR IWM GL FI PUR SD PP HRM

Dashboard Master Requisition Asset Active Asset Depreciation Asset Maintenance Asset Retirement Report

Home Am Asset Req H 1 Update

### Update Am Asset Req H: 1

Req Number: REQ-001

Status: Entered Creator: Melinda

Ship to: Office Need By Date: 2016-05-20

Am Asset Req L Showing 1-1 of 1 item

#	Type	Ivm item master	Line Description	Quantity	Uom
1	Non Product	Pallet Rack Heavy-duty	Its	1	Pcs

+Add Row Update

Gambar 4.11 Menyunting asset requisition

#### 4.6.12 Antarmuka Menghapus Asset Requisition

Pada antarmuka ini pengguna dapat menghapus data tipe kelompok aset yang sudah tidak dibutuhkan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.12

ERP2015 Master AM AP AR IWM GL FI PUR SD PP HRM

Dashboard Master Requisition Asset Active Asset Depreciation Asset Maintenance Asset Retirement Report

Home Am Asset Req H

### Am Asset Req H

Create Am Asset Req H Approve

Are you sure you want to delete this item?

OK Cancel

Am Asset Req H Showing 1-1 of 1 item

#	Req Number	Status	Creator	Ivm Inventory	Need By Date
1	REQ-001	Entered	Melinda	4	2016-05-20

Gambar 4.12 menghapus asset requisition

#### 4.6.13 Antarmuka Menambah Asset Active



Pada antarmuka ini pengguna dapat melakukan registrasi aset dengan mengisi form yang telah disediakan. Registrasi ini bertujuan untuk memberikan keterangan dan nomor pada setiap aset, agar aktifitas aset dapat tercatat dengan baik. Tampilan antarmuka ini dapat dilihat pada Gambar 4.13

Gambar 4.13 menambah asset active

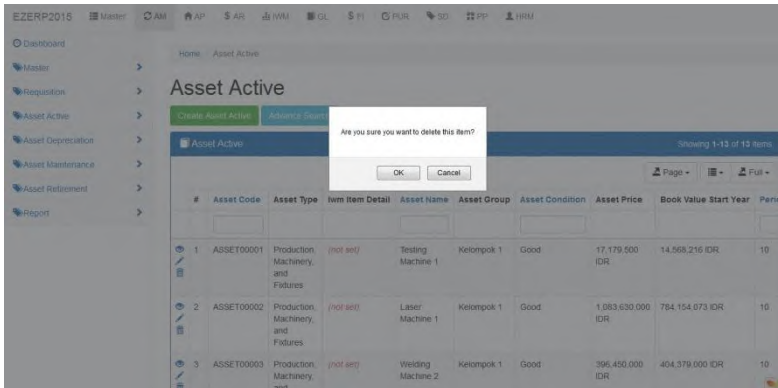
#### 4.6.14 Antarmuka Menyunting Asset Active

Pada antarmuka ini pengguna dapat mengubah data aset yang sudah di registrasi. Tampilan antarmuka ini dapat dilihat pada Gambar 4.14

Gambar 4.14 menyunting asset active

#### 4.6.15 Antarmuka Menghapus Asset Active

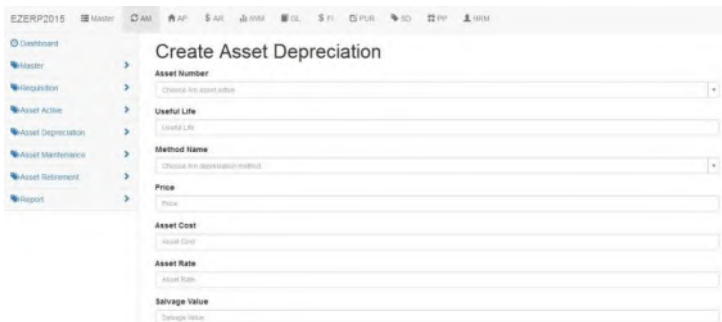
Pada antarmuka ini pengguna dapat menghapus aset yang sudah tidak dibutuhkan atau sudah tidak digunakan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.15



Gambar 4.15 menghapus asset active

#### 4.6.16 Antarmuka Menambah Asset Depreciation

Pada antarmuka ini pengguna dapat melakukan penambahan penyusutan aset dengan mengisi form yang telah disediakan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.16



Gambar 4.16 menambah asset depreciation

#### 4.6.17 Antarmuka Menyunting Asset Depreciation

Pada antarmuka ini pengguna dapat mengubah data penyusutan aset. Tampilan antarmuka ini dapat dilihat pada Gambar 4.17

The screenshot shows the 'Update Asset Depreciation: 13' form in the EZERP2015 system. The form includes the following fields:

- Asset Number:** ASSET0001 - Testing Machine 1
- Useful Life:** 10
- Method Name:** Straight Line Depreciation
- Price:** Price
- Asset Cost:** 17179500
- Asset Rate:** Asset Rate

Gambar 4.17 menyunting asset depreciation

#### 4.6.18 Antarmuka Menghapus Asset Depreciation

Pada antarmuka ini pengguna dapat menghapus data penyusutan aset. Tampilan antarmuka ini dapat dilihat pada Gambar 4.18

The screenshot shows the 'Asset Depreciation' list view in the EZERP2015 system. A confirmation dialog box is overlaid on the table, asking 'Are you sure you want to delete this item?' with 'OK' and 'Cancel' buttons. The table below shows the following data:

#	Asset Depreciation Code	Asset Active Number	Depreciation Method	Asset Price	Asset Cost	Asset Rate	Salvage Value	Useful
1		ASSET00001	Straight Line Depreciation	0 IDR	17.179.500 IDR	(not set)	3.435.900 IDR	10
2		ASSET00001	Straight Line Depreciation	0 IDR	15.805.140 IDR	(not set)	3.435.900 IDR	10
3	dep 01	ASSET00003	(not set)	0 IDR	396.450.000 IDR	(not set)	79.280.000 IDR	(not set)

Gambar 4.18 menghapus asset depreciation

#### 4.6.19 Antarmuka Menambah Asset Maintenance

Pada antarmuka ini pengguna dapat melakukan proses penambahan pemeliharaan aset dengan mengisi form yang telah disediakan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.19

The screenshot shows the 'Create Am Asset Maintenance' form. The form has the following fields:

- Asset Maintenance Code:** A text input field.
- Asset Number:** A dropdown menu with a search icon.
- Maintenance Cost:** A text input field.
- Maintenance Start Date:** A date picker.
- Maintenance Finish Date:** A date picker.

A green 'Create' button is located at the bottom left of the form.

Gambar 4.19 menambah asset maintenance

#### 4.6.20 Antarmuka Menyunting Asset Maintenance

Pada antarmuka ini pengguna dapat mengubah data pemeliharaan aset. Tampilan antarmuka ini dapat dilihat pada Gambar 4.20

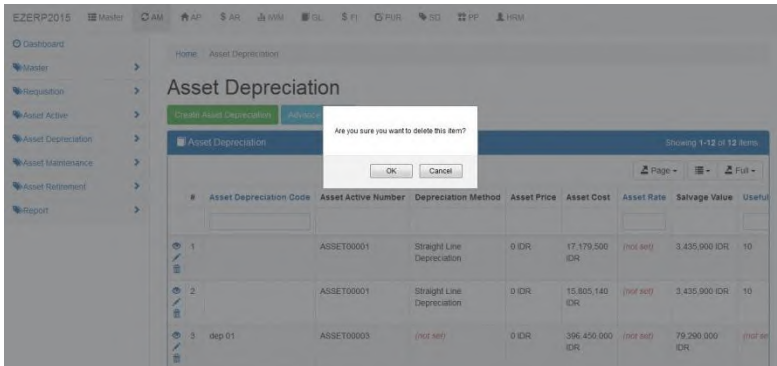
The screenshot shows the 'Update Asset Depreciation: 13' form. The form has the following fields:

- Asset Number:** A dropdown menu with a search icon, showing 'ASSET0001 - Testing Machine 1'.
- Useful Life:** A text input field with the value '10'.
- Method Name:** A dropdown menu with a search icon, showing 'Straight Line Depreciation'.
- Price:** A text input field.
- Asset Cost:** A text input field with the value '17178500'.
- Asset Rate:** A text input field.

Gambar 4.20 menyunting asset maintenance

#### 4.6.21 Antarmuka Menghapus Asset Maintenance

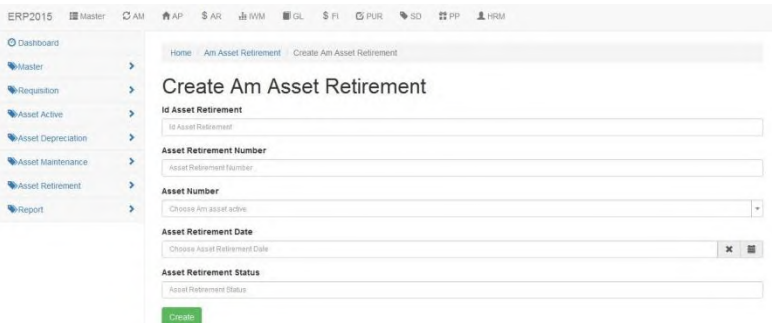
Pada antarmuka ini pengguna dapat menghapus pemeliharaan aset yang sudah tidak dibutuhkan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.21



Gambar 4.21 menghapus asset maintenance

#### 4.6.22 Antarmuka Menambah Asset Retirement

Pada antarmuka ini pengguna dapat melakukan permintaan penambahan aset yang sudah tidak aktif dengan mengisi form yang telah disediakan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.22



Gambar 4.22 menambah asset retirement

#### 4.6.23 Antarmuka Menyunting Asset Retirement

Pada antarmuka ini pengguna dapat mengubah data aset yang sudah tidak aktif. Tampilan antarmuka ini dapat dilihat pada Gambar 4.23

The screenshot shows the 'Update Am Asset Retirement: 1' form in the EZERP2015 system. The form includes the following fields:

- Asset Retirement Number:** RET-001
- Asset Number:** 30
- Asset Retirement Date:** 0000-00-00
- Asset Retirement Status:** OK

An 'Update' button is located at the bottom of the form.

Gambar 4.23 menyunting asset retirement

#### 4.6.24 Antarmuka Menghapus Asset Retirement

Pada antarmuka ini pengguna dapat menghapus data aset tidak aktif. Tampilan antarmuka ini dapat dilihat pada Gambar 4.24

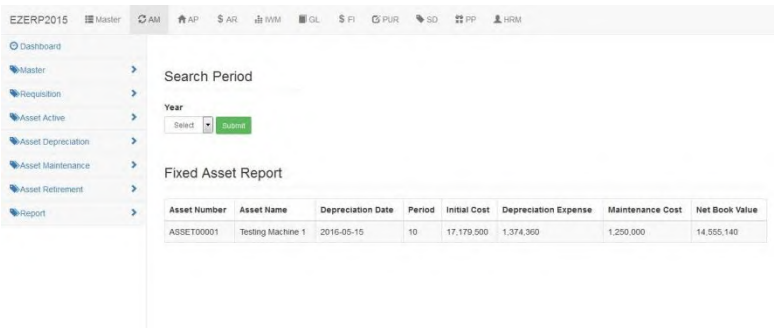
The screenshot shows the 'Asset Retirement' list view in the EZERP2015 system. A confirmation dialog box is displayed over the table, asking: "Are you sure you want to delete this item?". The dialog has 'OK' and 'Cancel' buttons. The background table shows the following data:

#	Asset Retirement Number	Am Asset Active	Asset Retirement Date	Asset Retirement Status
1	RET-001	30	0000-00-00	OK

Gambar 4.24 menghapus asset retirement

#### 4.6.25 Antarmuka Melihat report

Pada antarmuka ini pengguna dapat melihat informasi mengenai pencatatan pengeluaran mengenai aset per tahun. Tampilan antarmuka ini dapat dilihat pada Gambar 4.25



The screenshot displays the EZERP2016 software interface. On the left is a navigation menu with options: Dashboard, Master, Requisition, Asset Active, Asset Depreciation, Asset Maintenance, Asset Retirement, and Report. The main content area is titled 'Fixed Asset Report'. It includes a 'Search Period' section with a 'Year' dropdown menu (set to 'Select') and a green 'Submit' button. Below this is a table with the following data:

Asset Number	Asset Name	Depreciation Date	Period	Initial Cost	Depreciation Expense	Maintenance Cost	Net Book Value
ASSET00001	Testing Machine 1	2016-05-15	10	17.179.500	1.374.360	1.250.000	14.555.140

Gambar 4.25 melihat report

*[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. 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. Mozilla Firefox 46.0.1 dan Chrome 49.0 sebagai antarmuka untuk pengujian aplikasi klien.

#### **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 aluminum, 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 <sup>2</sup>
2	Kantor	50 x 50 m <sup>2</sup>
3	Parkir	60 x 50 m <sup>2</sup>
4	Kantin	40 x 60 m <sup>2</sup>
5	Pabrik	108 x 80 m <sup>2</sup>
6	Pengolahan Limbah	20 x 40 m <sup>2</sup>
7	<i>Raw Materials Inventory</i>	40 x 60 m <sup>2</sup>
8	<i>Semi-Finished Good Inventory</i>	40 x 60 m <sup>2</sup>
9	<i>Finished Good Inventory</i>	40 x 60 m <sup>2</sup>
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

No.	Nama	Kuantitas
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>		Red Deluxe Touring Bike
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>	Red Professional Touring Bike	
20	<i>Wheel's Bike</i>	2	<i>Pcs</i>		
21	<i>Chain's Bike</i>	1	<i>Pcs</i>		

No.	Bahan yang dibutuhkan	Kuantitas	Satuan unit	Bahan yang dihasilkan
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>	

### 5.2.1.5 Daftar Material

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	<b>Peminjaman dari Bank</b>	<b>120.000.000.000,00</b>

### 5.2.2. Pengujian Fungsionalitas Sistem

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 dijabarkan pada subbab berikut.

#### 5.2.3.1 Pengujian Mengelola Tipe Aset

Pengujian ini terdiri dari pengujian mengelola Asset Type. Pengujian mengelola Asset Type yaitu membuat, menyunting serta menghapus data yang sudah ada pada sistem.

Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.6

Tabel 5.6 pengujian mengelola tipe aset

ID	UJ-UC-001
Referensi Kasus Penggunaan	UC-001
Nama	Pengujian Fitur Mengelola Tipe Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data tipe aset
<i>Skenario 1</i>	<b><i>Pengguna membuat data Tipe Aset</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Type</i>
Data Uji	Inputan data tipe aset
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Asset Type</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman melihat <i>Asset Type</i> yang baru saja dibuat
<i>Skenario 2</i>	<b><i>Pengguna menyunting data Asset Type</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Type</i>
Data Uji	Perubahan data pada <i>Asset Type</i>
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Asset Type</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
<i>Skenario 3</i>	<b><i>Aktor menghapus data Asset Type</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Type</i>

Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Asset Type</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	Aktor berada pada halaman utama <i>Asset Type</i>

### 5.2.3.2 Pengujian Mengelola Metode Depresiasi

Pengujian ini terdiri dari pengujian mengelola *Asset Type*. Pengujian mengelola Metode Depresiasi yaitu membuat, menyunting serta menghapus data yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.7

Tabel 5.7 pengujian mengelola metode depresiasi

ID	UJ-UC-002
Referensi Kasus Penggunaan	UC-002
Nama	Pengujian Fitur Mengelola Metode Depresiasi
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data metode depresiasi
Skenario 1	<b><i>Pengguna membuat data metode depresiasi</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Depreciation Method</i>
Data Uji	Inputan data tipe metode depresiasi
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Depreciation Method</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data

Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman melihat <i>Depreciation Method</i> yang baru saja dibuat
Skenario 2	Pengguna menyunting data <i>Depreciation Method</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Depreciation Method</i>
Data Uji	Perubahan data pada <i>Depreciation Method</i>
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Depreciation Method</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
Skenario 3	Aktor menghapus data <i>Depreciation Method</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Depreciation Method</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Depreciation Method</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	Aktor berada pada halaman utama <i>Depreciation Method</i>

### 5.2.3.3 Pengujian Mengelola Kelompok Aset

Pengujian ini terdiri dari pengujian mengelola *Asset Group*. Pengujian mengelola Metode Depresiasi yaitu membuat, menyunting serta menghapus data yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.8



Tabel 5.8 pengujian mengelola kelompok aset

ID	UJ-UC-003
Referensi Kasus Penggunaan	UC-003
Nama	Pengujian Fitur Mengelola Kelompok Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data kelompok aset
<i>Skenario 1</i>	<b><i>Pengguna membuat data Asset Group</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Group</i>
Data Uji	Inputan data kelompok aset
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Asset Group</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman melihat <i>Asset Group</i> yang baru saja dibuat
Skenario 2	Pengguna menyunting data <i>Asset Group</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Group</i>
Data Uji	Perubahan data pada <i>Asset Group</i>
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Asset Group</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
Skenario 3	Aktor menghapus data <i>Asset Group</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Group</i>
Data Uji	-
Langkah	Pengguna memilih menghapus salah satu data <i>Asset</i>

Pengujian	<i>Group</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	Aktor berada pada halaman utama <i>Asset Group</i>

### 5.2.3.4 Pengujian Mengelola Penambahan Aset

Pengujian ini terdiri dari pengujian mengelola permintaan penambahan aset. Pengujian mengelola penambahan aset yaitu membuat, menyunting serta menghapus data yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.9

Tabel 5.9 pengujian mengelola penambahan aset

ID	<b>UJ-UC-004</b>
Referensi Kasus Penggunaan	UC-004
Nama	Pengujian Fitur Mengelola Penambahan Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data penambahan aset
<i>Skenario 1</i>	<b><i>Pengguna membuat data Asset Requisition</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Requisition</i>
Data Uji	Inputan data kelompok aset
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Asset 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	Pengguna berada pada halaman melihat <i>Asset</i>

	<i>Requisition</i> yang baru saja dibuat
Skenario 2	Pengguna menyunting data <i>Asset Requisition</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Requisition</i>
Data Uji	Perubahan data pada <i>Asset Requisition</i>
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Asset 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
Skenario 3	Aktor menghapus data <i>Asset Requisition</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Requisition</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Asset 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	Aktor berada pada halaman utama <i>Asset Requisition</i>

### 5.2.3.5 Pengujian Mengelola Pencatatan Aset

Pengujian ini terdiri dari pengujian mengelola pencatatan aset. Pengujian mengelola pencatatan aset yaitu membuat, menyunting serta menghapus data yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.10

Tabel 5.10 pengujian mengelola pencatatan aset

ID	UJ-UC-005
Referensi Kasus Penggunaan	UC-005
Nama	Pengujian Fitur Mengelola Pencatatan Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data pencatatan aset
Skenario 1	<b><i>Pengguna membuat data Asset Active</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Active</i>
Data Uji	Inputan data aset
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Asset Active</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman melihat <i>Asset Active</i> yang baru saja dibuat
Skenario 2	Pengguna menyunting data <i>Asset Active</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Active</i>
Data Uji	Perubahan data pada <i>Asset Active sset Requisition</i>
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Asset Active</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
Skenario 3	Aktor menghapus data <i>Asset Active</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Active</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Asset Active</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	Aktor berada pada halaman utama <i>Asset Active</i>

### 5.2.3.6 Pengujian Mengelola Penyusutan Nilai Aset

Pengujian ini terdiri dari pengujian mengelola penyusutan nilai aset. Pengujian mengelola penyusutan nilai aset yaitu membuat, menyunting serta menghapus data yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.11

Tabel 5.11 pengujian mengelola penyusutan nilai aset

ID	UJ-UC-006
Referensi Kasus Penggunaan	UC-006
Nama	Pengujian Fitur Mengelola Penyusutan Nilai Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data penyusutan nilai aset
<i>Skenario 1</i>	<b><i>Pengguna membuat data Asset Depreciation</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Depreciation</i>
Data Uji	Inputan data penyusutan nilai aset
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Asset Depreciation</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman melihat <i>Asset Depreciation</i> yang baru saja dibuat

Skenario 2	Pengguna menyunting data <i>Asset Depreciation</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Depreciation</i>
Data Uji	Perubahan data pada <i>Asset Depreciation</i>
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Asset Depreciation</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
Skenario 3	Aktor menghapus data <i>Asset Depreciation</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Depreciation</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Asset Depreciation</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	Aktor berada pada halaman utama <i>Asset Depreciation</i>

### 5.2.3.7 Pengujian Mengelola Pemeliharaan Aset

Pengujian ini terdiri dari pengujian mengelola pemeliharaan aset. Pengujian mengelola pemeliharaan nilai aset yaitu membuat, menyunting serta menghapus data yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.12

Tabel 5.12 pengujian mengelola pemeliharaan aset

ID	UJ-UC-007
----	-----------

ID	UJ-UC-007
Referensi Kasus Penggunaan	UC-007
Nama	Pengujian Fitur Mengelola Pemeliharaan Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data pemeliharaan aset
<i>Skenario 1</i>	<b><i>Pengguna membuat data Asset Maintenance</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Maintenance</i>
Data Uji	Inputan data aset
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Asset Maintenance</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman melihat <i>Asset Maintenance</i> yang baru saja dibuat
Skenario 2	Pengguna menyunting data <i>Asset Maintenance</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Maintenance</i>
Data Uji	Perubahan data pada <i>Asset Maintenance</i>
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Asset Maintenance</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
Skenario 3	Aktor menghapus data <i>Asset Maintenance</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Maintenance</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Asset Maintenance</i> yang ada pada sistem, lalu melakukan konfirmasi penghapusan data tersebut dan data akan

<b>ID</b>	<b>UJ-UC-007</b>
	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	Aktor berada pada halaman utama <i>Asset Maintenance</i>

### 5.2.3.8 Pengujian Mengelola Aset Tidak Aktif

Pengujian ini terdiri dari pengujian mengelola aset tidak aktif. Pengujian mengelola aset tidak aktif yaitu membuat, menyunting serta menghapus data yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.13

Tabel 5.13 pengujian mengelola aset tidak aktif

<b>ID</b>	<b>UJ-UC-008</b>
Referensi Kasus Penggunaan	UC-008
Nama	Pengujian Fitur Mengelola Aset Tidak Aktif
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data aset tidak aktif
<i>Skenario 1</i>	<b><i>Pengguna membuat data Asset Retirement</i></b>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Retirement</i>
Data Uji	Inputan data penyusutan nilai aset
Langkah Pengujian	Aktor masuk ke halaman membuat data <i>Asset Retirement</i> baru dan memasukkan Data Uji ke dalam basis data.
Hasil Yang Diharapkan	Data yang dimasukkan ke dalam basis data berhasil dimasukkan.
Hasil Yang Didapat	Data yang dimasukkan tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman melihat <i>Asset</i>



<b>ID</b>	<b>UJ-UC-008</b>
	<i>Retirement</i> yang baru saja dibuat
Skenario 2	Pengguna menyunting data <i>Asset Retirement</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Retirement</i>
Data Uji	Perubahan data pada <i>Asset Retirement</i>
Langkah Pengujian	Pengguna memilih menyunting salah satu data <i>Asset Retirement</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
Skenario 3	Aktor menghapus data <i>Asset Retirement</i>
Kondisi Awal	Aktor berada pada halaman utama <i>Asset Retirement</i>
Data Uji	-
Langkah Pengujian	Pengguna memilih menghapus salah satu data <i>Asset Retirement</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	Aktor berada pada halaman utama <i>Asset Retirement</i>

### 5.2.3.9 Pengujian Melihat Data Penambahan Aset

Pengujian ini terdiri dari pengujian melihat data penambahan aset. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.14

Tabel 5.14 pengujian melihat data penambahan aset

<b>ID</b>	<b>UJ-UC-009</b>
-----------	------------------

<b>ID</b>	<b>UJ-UC-009</b>
Referensi Kasus Penggunaan	UC-009
Nama	Pengujian Fitur Melihat Data Penambahan Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat data penambahan
<i>Skenario 1</i>	<b><i>Pengguna melihat data Asset Requisition</i></b>
Kondisi Awal	Aktor berada pada halaman index Asset Requisition
Data Uji	Input data pencarian
Langkah Pengujian	Aktor masuk ke halaman index Asset Requisition dan melakukan pencarian berdasarkan data uji
Hasil Yang Diharapkan	Tertampil halaman index dan data yang telah dicari
Hasil Yang Didapat	Data ditampilkan oleh sistem
Hasil Pengujian	Berhasil

### 5.2.3.10 Pengujian Melihat Data Aset Aktif

Pengujian ini terdiri dari pengujian melihat data aset aktif. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.15

Tabel 5.15 pengujian melihat data aset aktif

<b>ID</b>	<b>UJ-UC-010</b>
Referensi Kasus Penggunaan	UC-010
Nama	Pengujian Fitur Melihat Data Aset Aktif
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat data aset aktif
<i>Skenario 1</i>	<b><i>Pengguna melihat data Asset Active</i></b>
Kondisi Awal	Aktor berada pada halaman index Asset Active
Data Uji	Input data pencarian
Langkah Pengujian	Aktor masuk ke halaman index Asset Active dan melakukan pencarian berdasarkan data uji

ID	UJ-UC-010
Hasil Yang Diharapkan	Tertampil halaman index dan data yang yang telah dicari
Hasil Yang Didapat	Data ditampilkan oleh sistem
Hasil Pengujian	Berhasil

### 5.2.3.11 Pengujian Melihat Data Penyusutan Nilai Aset

Pengujian ini terdiri dari pengujian melihat data penyusutan nilai aset. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.16

Tabel 5.16 pengujian melihat data penyusutan nilai aset

ID	UJ-UC-011
Referensi Kasus Penggunaan	UC-011
Nama	Pengujian Fitur Melihat Penyusutan Nilai Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat data aset aktif
<i>Skenario 1</i>	<b><i>Pengguna melihat data Asset Active</i></b>
Kondisi Awal	Aktor berada pada halaman index Asset Active
Data Uji	Input data pencarian
Langkah Pengujian	Aktor masuk ke halaman index Asset Active dan melakukan pencarian berdasarkan data uji
Hasil Yang Diharapkan	Tertampil halaman index dan data yang yang telah dicari
Hasil Yang Didapat	Data ditampilkan oleh sistem
Hasil Pengujian	Berhasil

### 5.2.3.12 Pengujian Melihat Data Pemeliharaan Aset

Pengujian ini terdiri dari pengujian melihat data pemeliharaan aset. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.17

Tabel 5.17 melihat data pemeliharaan aset

ID	UJ-UC-012
Referensi Kasus Penggunaan	UC-012
Nama	Pengujian Fitur Melihat Data Pemeliharaan Aset
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat data pemeliharaan aset
<i>Skenario 1</i>	<b><i>Pengguna melihat data Asset Maintenance</i></b>
Kondisi Awal	Aktor berada pada halaman index <i>Asset Maintenance</i>
Data Uji	Input data pencarian
Langkah Pengujian	Aktor masuk ke halaman index <i>Asset Maintenance</i> dan melakukan pencarian berdasarkan data uji
Hasil Yang Diharapkan	Tertampil halaman index dan data yang telah dicari
Hasil Yang Didapat	Data ditampilkan oleh sistem
Hasil Pengujian	Berhasil

### 5.2.3.13 Pengujian Melihat Data Aset Tidak Aktif

Pengujian ini terdiri dari pengujian melihat data aset tidak aktif. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.18

Tabel 5.18 pengujian melihat data aset tidak aktif

ID	UJ-UC-013
Referensi Kasus Penggunaan	UC-013
Nama	Pengujian Fitur Melihat Data Aset Tidak Aktif
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat data aset tidak aktif
<i>Skenario 1</i>	<b><i>Pengguna melihat data Asset Retirement</i></b>
Kondisi Awal	Aktor berada pada halaman index <i>Asset Retirement</i>
Data Uji	Input data pencarian

Langkah Pengujian	Aktor masuk ke halaman index <i>Asset Retirement</i> dan melakukan pencarian berdasarkan data uji
Hasil Yang Diharapkan	Tertampil halaman index dan data yang telah dicari
Hasil Yang Didapat	Data ditampilkan oleh sistem
Hasil Pengujian	Berhasil

#### 5.2.3.14 Pengujian Melihat *Report Tahunan*

Pengujian ini terdiri dari pengujian melihat data pencatatan atau report tahunan. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada tabel 5.19

Tabel 5.19 pengujian melihat report tahunan

ID	UJ-UC-014
Referensi Kasus Penggunaan	UC-014
Nama	Pengujian Fitur Melihat <i>Report Tahunan</i>
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat data report tahunan
<i>Skenario 1</i>	<b><i>Pengguna melihat Report Tahunan</i></b>
Kondisi Awal	Aktor berada pada halaman Report
Langkah Pengujian	Aktor masuk ke halaman index Report
Hasil Yang Diharapkan	Tertampil halaman index
Hasil Yang Didapat	Data ditampilkan oleh sistem
Hasil Pengujian	Berhasil

#### 5.2.3. Pengujian Fungsionalitas Role Based Access Control (RBAC)

Pengujian ini terdiri dari pengujian mengelola *Role Based Access Control (RBAC)*. Rincian skenario pengujian pada fitur *Role Based Access Control (RBAC)* dapat dilihat pada Tabel 5.20.

Tabel 5.20. Pengujian fitur Role Based Access Control (RBAC)

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

ID	UJ.RBAC
<b>Hasil Yang Didapat</b>	<i>User yang disunting mengalami perubahan data sesuai inputan.</i>
<b>Hasil Pengujian</b>	Berhasil.
<b>Kondisi Akhir</b>	Pengguna berada pada halaman daftar user.
<b>Skenario 3</b>	<b><i>Pengguna menghapus user tertentu.</i></b>
<b>Kondisi Awal</b>	Pengguna berada pada halaman daftar user.
<b>Data Uji</b>	-

Hasil pengujian fitur RBAC dapat dilihat pada Gambar 5.1-Gambar 5.3

The screenshot shows a web browser window with the URL 'EZERP 2015'. The page has a navigation bar with various menu items like 'Master', 'AM', 'AP', 'AR', 'IWM', 'GL', 'FI', 'PUR', 'SD', 'PP', and 'HRM'. The main content area is titled 'Login' and contains a form with the following elements:

- A heading 'Login' followed by the instruction 'Please fill out the following fields to login:'.
- A red horizontal line separating the instruction from the form fields.
- A 'Username' label above an input field containing the text 'admin'.
- A 'Password' label above an input field with masked characters.
- A checked checkbox labeled 'Remember Me'.
- A 'Select DB' label above a dropdown menu.
- A blue 'Login' button at the bottom of the form.
- A link below the button that reads 'If you forgot your password you can reset it.'

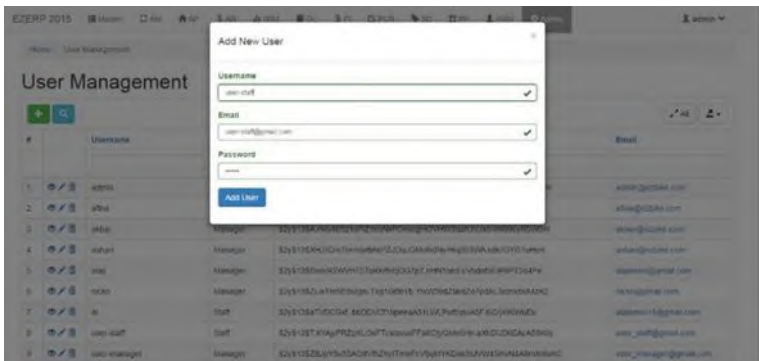
Gambar 5.1 Login RBAC

Gambar 5.2 menunjukkan hasil pengujian untuk proses login oleh user admin.



Gambar 5.2. Tampilan awal setelah login berhasil dilakukan

Gambar 5.3 menunjukkan hasil pengujian berupa tampilan awal setelah proses login oleh user admin.



Gambar 5.3. Admin menambahkan user baru

Gambar 5.3 menunjukkan proses penginputan data *user* untuk skenario 1. Sedangkan untuk Gambar 5.4 menunjukkan hasil pengujianya, dimana dapat dilihat penambahan pada halaman daftar user yang baru.



The screenshot shows a web-based form titled "Update User: user-staff". The form has several sections:
 

- Username:** A text input field containing "user-staff".
- Position:** A dropdown menu with the text "Choose Position".
- Password Hash:** A text input field containing a long alphanumeric string.
- Email:** A text input field containing "user\_staff@gmail.com".
- Permission:** A list of permissions with checkboxes, including "Allow user as Asset Management Manager", "Allow user as Account Payable Manager", "Allow user as Account Receivable Manager", and "Allow user as Inventory Manager".
- Update:** A blue button at the bottom left.

Gambar 5.4. Proses user admin menyunting data user

Gambar 5.5 menunjukkan proses penyuntingan data *user* untuk skenario 2.

The screenshot shows a "User Management" interface with a table of users and a confirmation dialog box. The table has columns for #, Username, Position, Password Hash, and Email. The dialog box asks "Are you sure you want to delete this item?" with "OK" and "Cancel" buttons.

#	Username	Position	Password Hash	Email
1	admin	Administrator	\$2y\$13\$39Fy1TEvG0iH7HCkayF4Pq;WCT7GzJdmZ6ZpPoGm	admin@ezbike.com
2	afna	Manager	\$2y\$13\$06v9CQZ2Y709y78QzLcQ8E3JF7XpDHYBvU3CnpK0G4	afna@ezbike.com
3	jabar	Manager	\$2y\$13\$A rHf80321aRZ 7evh9Pcmwq87vHW8u8t7u3g4W6yR0W0V	jabar@ezbike.com
4	lailani	Manager	\$2y\$13\$XUJ0c7m8qy5ePZ2Du3G8m8M9eHq390VA uf0c9V510rWk	lailani@ezbike.com
5	slap	Manager	\$2y\$13\$DwRQ2WH173uWv7BQZG7P 3tHh0b0c7ho05URP72d4F*	slapneo@gmail.com
6	niko	Manager	\$2y\$13\$ZLw7m8DBpL7q108vY5Y5W08ZdaiZc7pAL3ozvzA4dQ2	niko@gmail.com
7	sl	Staff	\$2y\$13\$8TYDGDx 38C0VCTHpeeaA1UkLPh7yA5F8Cj49W6Ea	slapneo11@gmail.com
8	user-staff	Staff	\$2y\$13\$T XvVq7RZp5Cup7CebvuoF7AbGyG8k6Hs akh0UC6A488kly	user_staff@gmail.com
9	user-manager	Manager	\$2y\$13\$Z8p7u6AGv8vZy7T me7v5g8Y7Dae3LW46vMv4A8hd8a8C	user_manager@gmail.com
10	ina	Staff	\$2y\$13\$W k8mC7Vp8a7kU0p807828m8q4Hs 40N4F e8FgCgk0Qh75	ina@ezbike.com

Gambar 5.5. Proses user admin menghapus data user

Gambar 5.5 dapat dilihat proses penghapusan *user* untuk skenario 3. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan RBAC telah bekerja dengan baik seperti yang diharapkan.

## 5.2.4. Pengujian Fungsionalitas Basis Data Terdistribusi

Pengujian Basis Data Terdistribusi (BDT) dilakukan untuk menguji 2 (dua) aspek utama dalam konsep BDT, yaitu

replikasi dan *high availability*. Sistem BDT menggunakan 4 (empat) 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 (dua) buah server yang berfungsi sebagai penyimpan data.

- **Prosedur Simulasi**

### Replikasi

1. Memastikan seluruh *server* dan sistem BDT dalam keadaan *online*.

#### Warehouse 3

ID	3
Company	EZETANANT BIKE INDONESIA
Warehouse Name	EZERP Testing BDT

Gambar 5.6. Pengujian fitur replikasi pada sistem

2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi ERP yang ditunjukkan pada Gambar 5.7

```

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

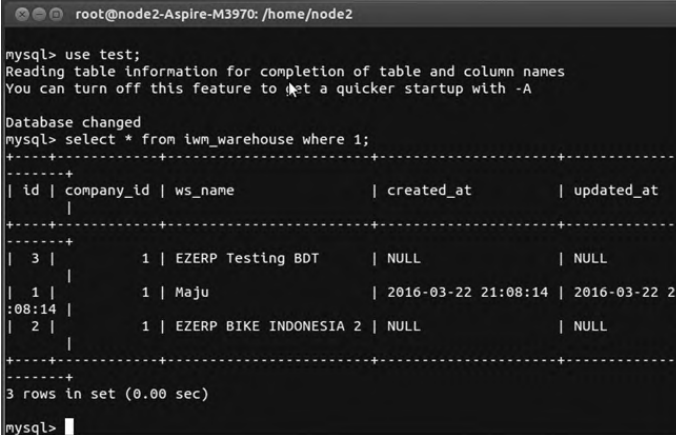
Database changed
mysql> select * from iwm_warehouse where 1;
+-----+-----+-----+-----+-----+
| id | company_id | ws_name | created_at | updated_at |
+-----+-----+-----+-----+-----+
| 3 | 1 | EZERP Testing BDT | NULL | NULL |
| 1 | 1 | Maju | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP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

mysql>

```

Gambar 5.7. Pengujian fitur replikasi pada database server 1

3. Melakukan pengecekan terhadap hasil penambahan, perubahan, dan penghapusan data pada seluruh *data node* yang ditampilkan pada Gambar 5.8 dan Gambar 5.9



```

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 5.8. Pengujian fitur replikasi pada database server 2

### ***High Availability***

1. Mematikan *server* aplikasi, *management server* dan sistem BDT pada salah satu *node*. Ditampilkan pada Gambar 5.9.

#### Warehouse 3

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

Gambar 5.9. Pengujian fitur high-availability pada sistem

2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi yang ditunjukkan pada Gambar 5.10.

```

root@master-Aspire-M3970: /home/master
ld=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)

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

ndb_mgm> Node 2: Started (version 7.1.34)
Node 2: Node shutdown initiated
Node 2: Node shutdown completed.
show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
ld=2 (not connected, accepting connect from 10.151.64.182)
ld=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

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

[mysqld(API)] 2 node(s)
ld=4 (not connected, accepting connect from 10.151.64.182)
ld=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)

ndb_mgm>

```

Gambar 5.10. Pengecekan availability pada node server

```

master@master-Aspire-M3970: ~
| tax |
| uom |
| user |
-----+-----
228 rows in set (0.01 sec)

mysql> select * from lwn_warehouse where 1;
-----+-----+-----+-----+
| id | company_id | ws_name | created_at | updated_at |
-----+-----+-----+-----+
| 3 | 1 | EZERP Testing BDT HA | 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 5.11. Pengecekan database pada node 1

- Melakukan pengecekan terhadap hasil pemrosesan data pada *node* yang masih aktif yang ditampilkan pada Gambar 5.12.

```

root@node2-Aspire-M3970: /home/node2
ld=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

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

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

ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
ld=2 (not connected, accepting connect from 10.151.64.182)
ld=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

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

[mysqld(API)] 2 node(s)
ld=4 (not connected, accepting connect from 10.151.64.182)
ld=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)

ndb_mgm>

```

Gambar 5.12. Pengecekan availability pada node server

```

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 5.13. Mematikan database server pada node 2

### 5.3 Implementasi Antar Muka

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 5.21 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 5.21 evaluasi pengujian

No.	ID	Nama Kasus Penggunaan	Skenario	Hasil
11	UJ.UC.001	Mengelola <i>Asset Type</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
22	UJ.UC.002	Mengelola <i>Depreciation Method</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
33	UJ.UC.003	Mengelola <i>Asset Group</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
34	UJ.UC.004	Mengelola <i>Asset Requisition</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
35	UJ.UC.005	Mengelola <i>Asset Active</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
36	UJ.UC.006	Mengelola <i>Asset Depreciation</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
37	UJ.UC.007	Mengelola <i>Asset Maintenance</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
38	UJ.UC.008	Mengelola <i>Asset Retirement</i>	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil

No.	ID	Nama Kasus Penggunaan	Skenario	Hasil
99	UJ.UC.009	Melihat Data Penambahan Aset	Skenario 1	Berhasil
910	UJ.UC.010	Melihat Data Pencatatan Aset	Skenario 1	Berhasil
911	UJ.UC.011	Melihat Data Penyusutan Nilai Aset	Skenario 1	Berhasil
912	UJ.UC.012	Melihat Data Pemeliharaan Aset	Skenario 1	Berhasil
913	UJ.UC.013	Melihat Data Aset Tidak Aktif	Skenario 1	Berhasil
114	UJ.UC.014	Melihat Report Tahunan	Skenario 1	Berhasil

*[Halaman ini sengaja dikosongka]*



## **BAB VI**

### **KESIMPULAN DAN SARAN**

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

#### **6.1 Kesimpulan**

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

1. Aplikasi Asset Management ini memudahkan dalam pengelolaan aset perusahaan, dari pembelian aset sampai menjadi aset tidak aktif
2. Metode pada perhitungan penurunan aset untuk setiap metode mempunyai nilai atau biaya penurunan yang berbeda, dipilih yang terbaik untuk setiap aset.
3. Menggunakan lebih dari satu *database server* dan menggunakan teknik replikasi pada *distributed database* mampu mengatasi kegagalan sistem yang disebabkan oleh *database server*.
4. Menggunakan 3 server dimana 2 server merupakan *database server* dan satu server bertindak sebagai *management server* yang bertugas dalam memproses replikasi terhadap kedua *database server*.

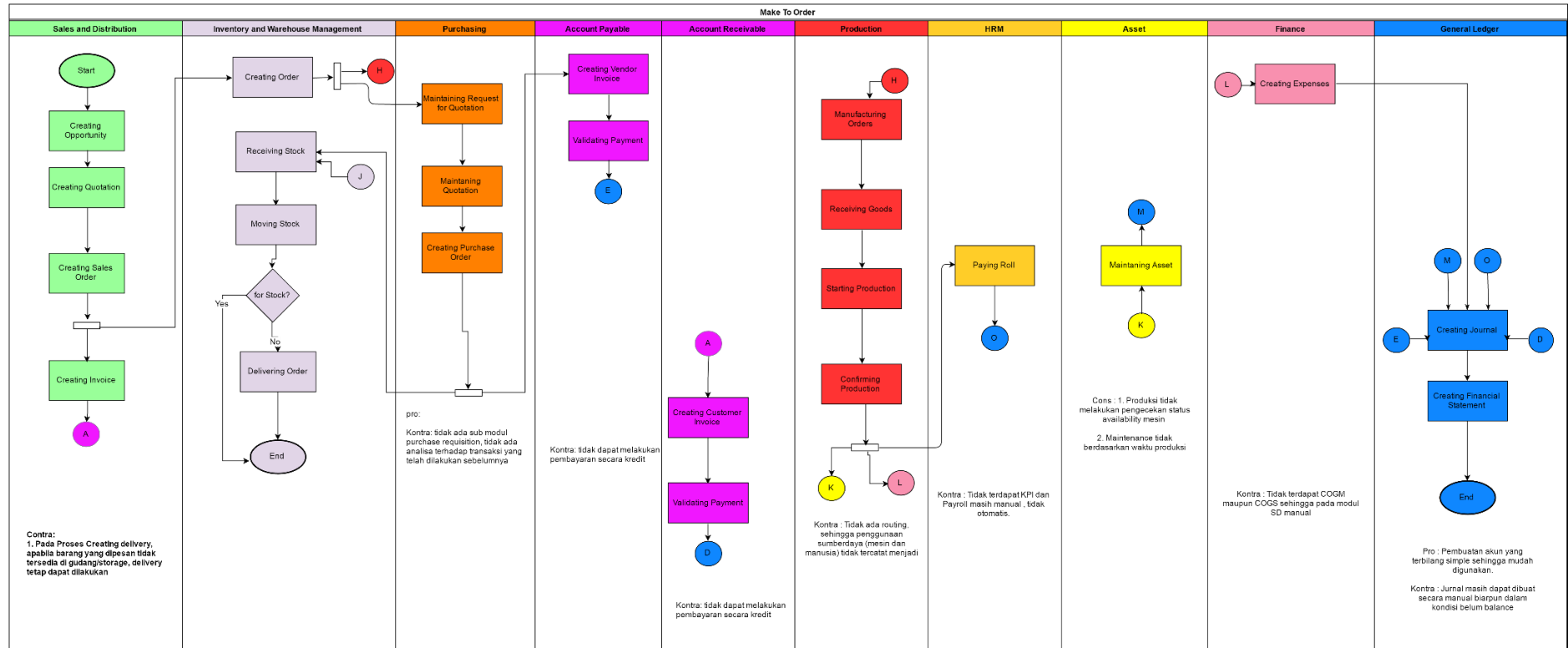
#### **6.2 Saran**

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

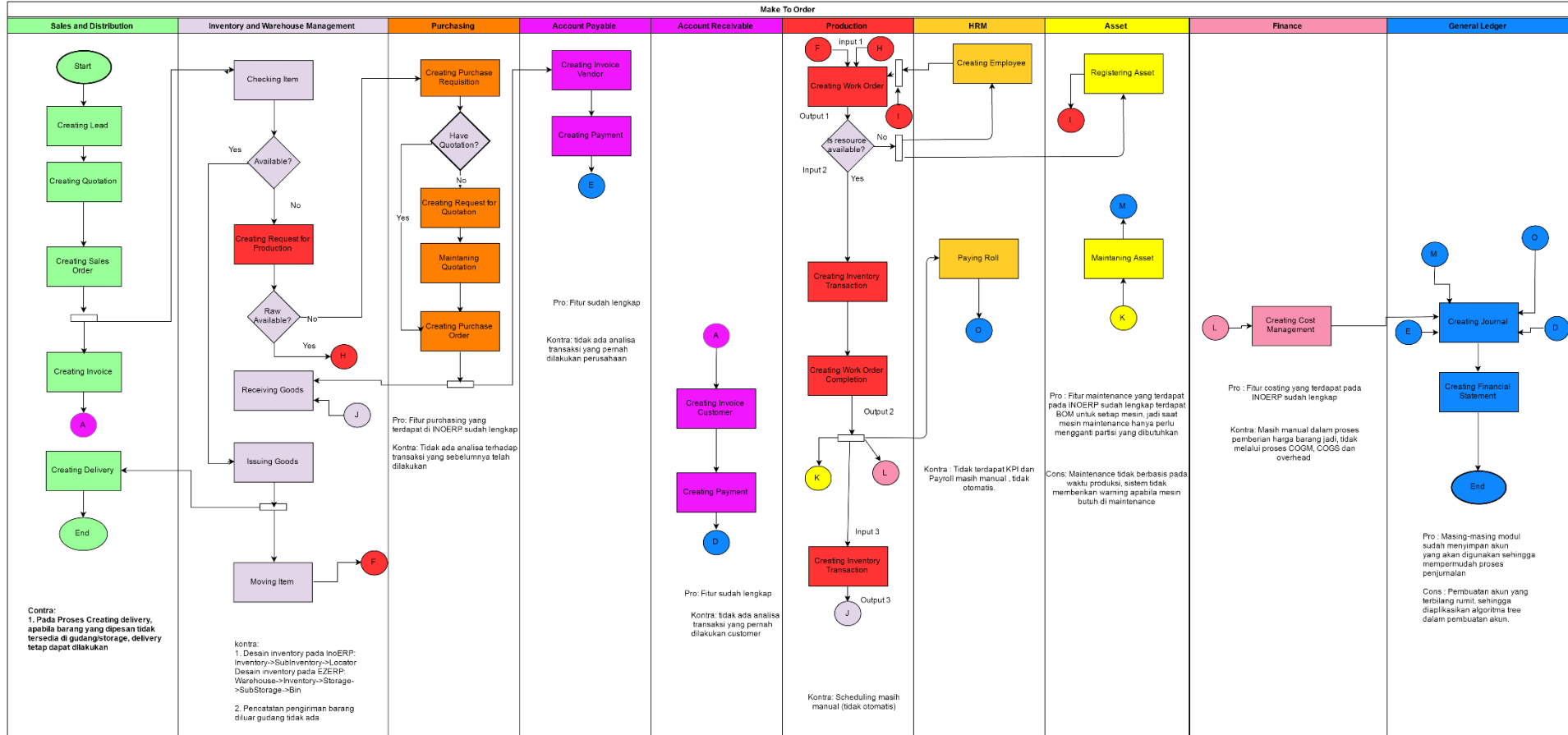
1. Menambahkan routing untuk Asset Maintenance sehingga pada saat ada barang yang perlu di service tidak membutuhkan waktu yang lama

# LAMPIRAN

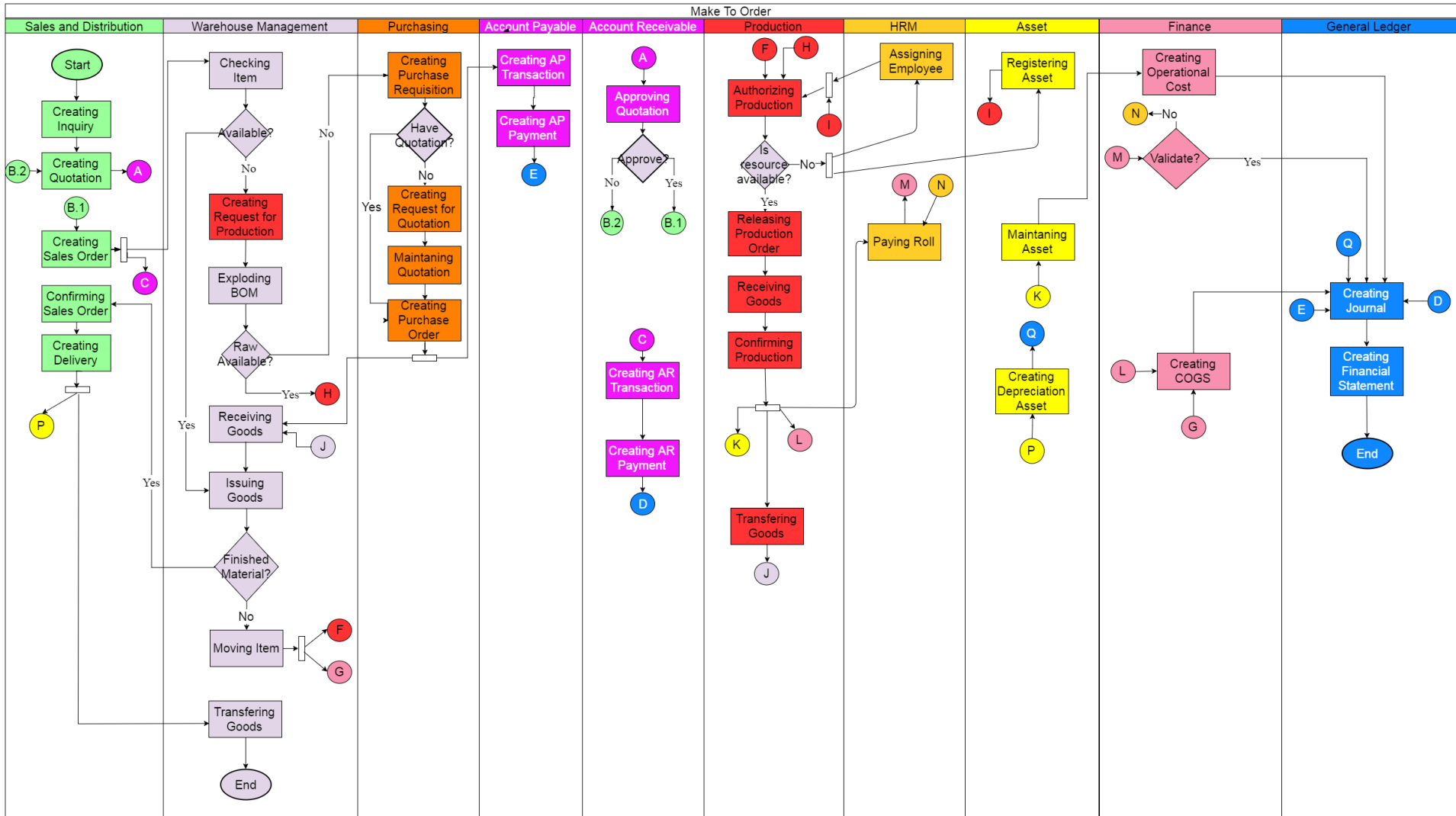
## A.1 Proses Bisnis Odoo



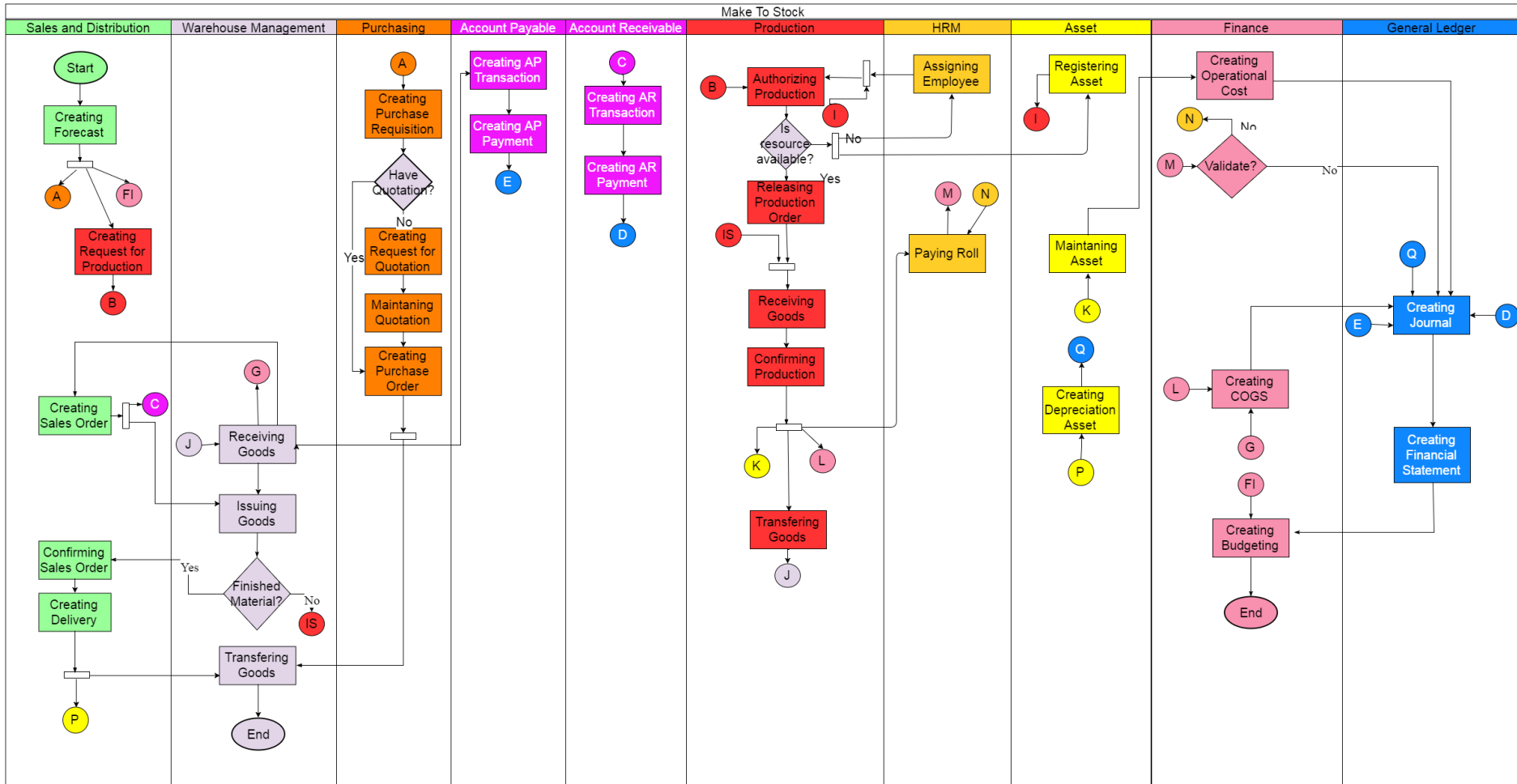
## A.2 Proses Bisnis InoERP



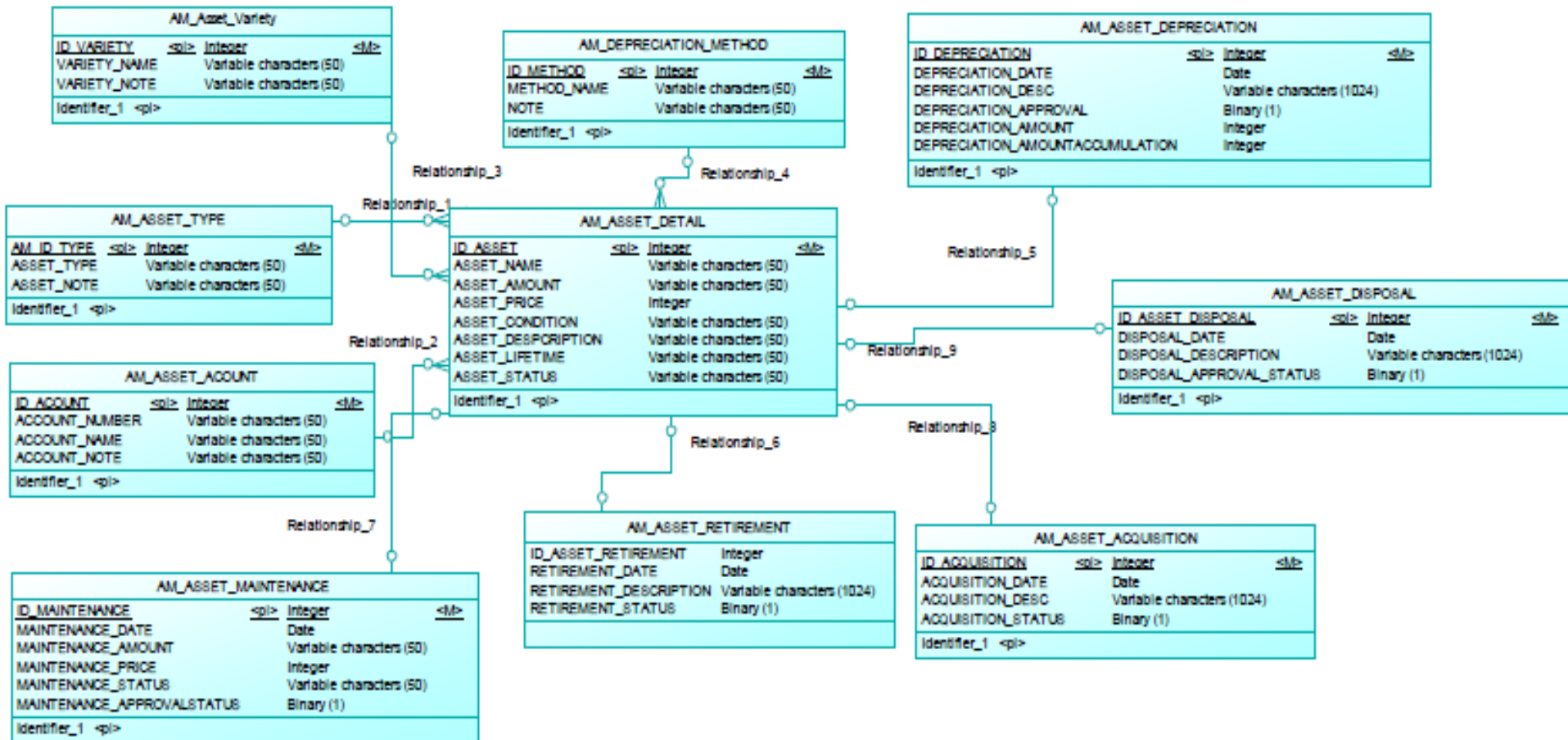
A.3 Proses Bisnis Make-To-Order ERP 2016



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



## A.3 Proses Bisnis Make-To-Order ERP 2016



## **BAB VI**

### **KESIMPULAN DAN SARAN**

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

#### **6.1 Kesimpulan**

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

1. Aplikasi Asset Management ini memudahkan dalam pengelolaan aset perusahaan, dari pembelian aset sampai menjadi aset tidak aktif
2. Metode pada perhitungan penurunan aset untuk setiap metode mempunyai nilai atau biaya penurunan yang berbeda, dipilih yang terbaik untuk setiap aset.
3. Menggunakan lebih dari satu *database server* dan menggunakan teknik replikasi pada *distributed database* mampu mengatasi kegagalan sistem yang disebabkan oleh *database server*.
4. Menggunakan 3 server dimana 2 server merupakan *database server* dan satu server bertindak sebagai *management server* yang bertugas dalam memproses replikasi terhadap kedua *database server*.

#### **6.2 Saran**

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

1. Menambahkan routing untuk Asset Maintenance sehingga pada saat ada barang yang perlu di service tidak membutuhkan waktu yang lama

2. Menambahkan Report perbulan dan perhari untuk mengetahui aktifitas aset pada bulan tersebut dan hari tersebut
3. Melakukan registrasi aset untuk jumlah yang banyak tidak perlu satu per satu.
4. Meningkatkan *running process* untuk proses dengan data yang sangat besar



## DAFTAR PUSTAKA

- [1] A. Rifa'i, R. Sarno dan 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 dan J. Word, Integrated Business Processes with ERP Systems, New Jersey: Wiley and Sons, Inc., 2012.
- [4] "BPMN Modeling Reference," [Online]. Available: <https://camunda.org/bpmn/reference/>.
- [5] F. Chong, G. Carraro dan R. Wolter, "Multi-Tenant Data Architecture," 2006. [Online]. Available: <https://msdn.microsoft.com/en-us/library/aa479086.aspx>. [Diakses 1 February 2016].
- [6] Sharive, Proyek Membangun Website dengan Yii Framework, Yogyakarta: Lokomedia, 2014.
- [7] Fathansyah, Basis Data, Bandung: Informatika, 2015.
- [8] Kasman dan D. A, Trik Mudah Menguasai OOP dengan PHP, Yogyakarta: Lokomedia, 2013.
- [9] "wikipedia," October 2010. [Online]. Available: <https://en.wikipedia.org/wiki/Procurement>.
- [10] Andrew, "MySQL Cluster 7.4," 2015. [Online]. Available: <https://www.mysql.com/products/cluster/>. [Diakses 10 February 2016].

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Melinda Tresnawati atau yang biasa disapa dengan Melinda. Lahir di Surabaya pada tanggal 26 Mei 1994. Telah menempuh pendidikan di SDN Tembok Dukuh IV-86 (2000-2006), SMP Negeri 4 Surabaya (2006-2009), SMA Negeri 2 Surabaya (2009-2012) dan saat ini sedang menempuh pendidikan Sarjana di Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya angkatan tahun 2012.

Memiliki ketertarikan dalam segala bentuk kegiatan sosial. Terlibat aktif dalam organisasi kemahasiswaan serta kepanitiaan selama perkuliahan, antara lain staff Kesejahteraan Mahasiswa di Himpunan Mahasiswa Teknik Computer-Informatika ITS 2013/2014, Staff Divisi REEVA 2013, Staff Hubungan Masyarakat acara Schematics 2014, Untuk komunikasi, dapat dihubungi melalui surel : [melindatresnawati@gmail.com](mailto:melindatresnawati@gmail.com)