



**PERANCANGAN DAN PEMBUATAN SISTEM INFORMASI SIMPAN
PINJAM PADA KOPERASI PEGAWAI REPUBLIK INDONESIA (KP-RI)
"DIAN PATIRANA" BERBASIS WEB**

SKRIPSI

Oleh

Dwi Andika Febrianto

NIM 092410101004

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015



**PERANCANGAN DAN PEMBUATAN SISTEM INFORMASI SIMPAN
PINJAM PADA KOPERASI PEGAWAI REPUBLIK INDONESIA (KP-RI)
"DIAN PATIRANA" BERBASIS WEB**

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Sistem Informasi (S1) dan mencapai gelar Sarjana Komputer

Oleh

Dwi Andika Febrianto

NIM 092410101004

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2015

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang selalu memberikan rahmat dan hidayah-Nya kepada hamba-Nya.
2. Ayahanda Susanto dan Ibunda Sukaryati yang tercinta;
3. Kakakku Fitri Yunitha Antarini, Adikku Noviana Tri Andayani dan Keponakanku Kanaya Lidya Putri Cahyadi yang tercinta;
4. Pembimbing I Bapak Anang Andrianto, ST., MT serta pembimbing II Bapak Muhammad Arief Hidayat, S.Kom., M.Kom.
5. Teman – teman PSSI angkatan 2009 dan teman – teman junior angkatan 2010 yang telah memberikan dukungan.
6. Guru – guru sejak taman kanak – kanak sampai dengan perguruan tinggi;
7. Sahabat – sahabatku sejak taman kanak – kanak sampai sekarang yang selalu memberikan dukungan dan semangat;
8. Almamater Program Studi Sistem Informasi Universitas Jember

MOTTO

“Sesungguhnya bersama kesukaran itu ada keringanan. Karena itu bila kau sudah selesai (mengerjakan yang lain). Dan berharaplah kepada Tuhanmu.”

(Q.S Al Insyirah : 6-8)

“If you want something in your life you've never had, you'll have to do something, you've never done.”

(JD Houston, 42)

Seorang filsuf Yunani mengatakan,

“Orang pandai belajar dari pengalamannya dan orang bijak belajar dari pengalaman orang lain, tetapi orang bodoh tidak belajar apapun.”

:: BE HONEST ALL THE TIME ::

HALAMAN PERNYATAAN

Saya yang bertandatangan di bawah ini :

Nama : Dwi Andika Febrianto

NIM : 092410101004

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Perancangan dan Pembuatan Sistem Informasi Simpan Pinjam Pada Koperasi Pegawai Republik Indonesia (KP-RI) “DIAN PATIRANA” Berbasis Web” adalah benar – benar hasil karya saya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapatkan sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 01 Juni 2015

Yang menyatakan,

Dwi Andika Febrianto

NIM 092410101004

PENGESAHAN

Skripsi berjudul Perancangan dan Pembuatan Sistem Informasi Simpan Pinjam Pada Koperasi Pegawai Republik Indonesia (KP-RI) DIAN PATIRANA Berbasis Web, telah diuji dan disahkan pada:

Hari tanggal : Senin, 01 Juni 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji

Penguji I,

Penguji II,

Dr. Saiful Bukhori, ST., M.Kom

Nelly Oktavia A, S.Si., MT

NIP. 196811131994121001

NIP. 198410242009122008

Mengesahkan

Ketua Program Studi

Prof. Drs. Slamini, M, Comp.Sc.,Ph.D

NIP. 196704201992011001

PENGESAHAN PEMBIMBING

Karya ilmiah skripsi berjudul Perancangan dan Pembuatan Sistem Informasi Simpan Pinjam Pada Koperasi Pegawai Republik Indonesia (KP-RI) DIAN PATIRANA Berbasis Web telah diuji dan disahkan pada:

Hari, tanggal : Senin, 01 Juni 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Pembimbing:

Pembimbing I,

Pembimbing II,

Anang Andrianto, ST., MT

Muhammad Arief Hidayat, S.Kom., M.Kom

NIP. 196906151997021002

NIP. 198101232010121003

RINGKASAN

Perancangan dan Pembuatan Sistem Informasi Simpan Pinjam Pada Koperasi Pegawai Republik Indonesia (KP-RI) “DIAN PATIRANA” Berbasis Web; Dwi Andika Febrianto, 092410101004; 2015; 211 halaman, Program Studi Sistem Informasi Jember.

Koperasi Pegawai Republik Indonesia (KP-RI) “DIAN PATIRANA” adalah sebuah lembaga keuangan bukan bank yang menyediakan produk berupa barang dan jasa. Koperasi “DIAN PATIRANA” selain sebagai lembaga keuangan, produk utama dengan memberikan pelayanan simpan pinjam (USP) kepada guru-guru pegawai negeri sipil (PNS) dan pegawai dinas unit pelaksana teknis pendidikan TK/SD di kecamatan Grujungan, kabupaten Bondowoso. Untuk meningkatkan pelayanan dan pemrosesan transaksi simpan pinjam menjadi lebih efisien, diperlukan suatu penanganan dan pengolahan data yang lebih baik.

Hasil analisis yang telah didapatkan oleh penulis, maka penelitian ini bertujuan untuk memudahkan pengguna untuk melakukan proses administrasi yang sebelumnya masih dilakukan secara sederhana. Seperti memproses data transaksi simpanan, pinjaman, angsuran, laporan transaksi, penginputan dan penyimpanan data masih dilakukan secara manual. Untuk mengatasi persoalan tersebut, maka diperlukan suatu sistem informasi berbasis komputer.

Sistem informasi simpan pinjam berbasis *web* ini, telah dilengkapi oleh beberapa fitur. Fitur – fitur utama atau fungsional dan fitur-fitur sampingan atau non fungsional yang mungkin dibutuhkan oleh pengguna koperasi nantinya. Metode pengembangan sistem yang telah digunakan pada sistem informasi ini adalah dengan metode *Incremental Model*. Tahapan dalam metode ini yakni analisis, desain sistem, pengkodean sistem dan *test*. Sistem ini menggunakan bahasa pemrograman PHP yang menggunakan konsep OOP dan menggunakan MySQL sebagai manajemen databasenya.

PRAKATA

Puji syukur kehadiran Allah SWT, karena segala limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Perancangan dan Pembuatan Sistem Informasi Simpan Pinjam Pada Koperasi Pegawai Republik Indonesia (KP-RI) “DIAN PATIRANA” Berbasis Web”. Tugas akhir ini disusun guna untuk memenuhi kebutuhan dalam menyelesaikan pendidikan strata satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Dengan selesainya tugas ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Prof. Drs. Slamin, M, Comp.Sc.,Ph.D selaku ketua Program Studi Sistem Informasi Universitas Jember dan selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
2. Anang Andrianto, ST., MT. Selaku Dosen Pembimbing Utama, M. Arief Hidayat,S.Kom., M.Kom. Selaku dosen pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
3. Ayahanda Susanto dan ibunda Sukaryati yang telah memberikan dorongan motivasi serta doa agar terselesaikannya skripsi ini;
4. Semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa masih banyak kekurangan dari tugas akhir ini, baik dari materi maupun teknik penyajiannya, mengingat kembali bahwa kurangnya pengetahuan dan pengalaman penulis. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan.

Terima kasih,

Jember, 01 Juni 2015

Penulis

DAFTAR ISI

Halaman

COVER	i
JUDUL	ii
PERSEMBAHAN	iii
MOTTO	iv
HALAMAN PERNYATAAN	v
PENGESAHAN	vi
PENGESAHAN PEMBIMBING	vii
RINGKASAN	viii
PRAKATA	ix
DAFTAR ISI	x
DAFTAR TABEL	xix
DAFTAR GAMBAR	xx
DAFTAR LAMPIRAN	xxiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan Penelitian.....	2
1.4. Batasan Masalah.....	3
1.5. Manfaat Penelitian.....	3
1.5.1. Bagi Akademis.....	3

1.5.2.	Bagi Koperasi Pegawai “DIAN PATIRANA”	3
1.5.3.	Bagi Peneliti	4
1.6.	Sistematika Penulisan	4
BAB 2. TINJAUAN PUSTAKA.....		5
2.1.	Konsep Dasar Sistem Informasi	5
2.1.1.	Konsep Dasar Sistem	5
2.1.2.	Konsep Dasar Informasi	6
2.1.3.	Konsep Dasar Sistem Informasi.....	6
2.2.	Sistem Informasi	7
2.2.1.	Memiliki Komponen	7
2.2.2.	Batas Sistem (<i>Boundary</i>)	7
2.2.3.	Lingkungan Luar Sistem (<i>Environment</i>).....	7
2.2.4.	Penghubung Sistem (<i>Interface</i>).....	8
2.2.5.	Masukan Sistem (<i>Input</i>)	8
2.2.6.	Keluaran Sistem (<i>Output</i>)	8
2.2.7.	Pengolahan Sistem (<i>Process</i>).....	8
2.2.8.	Sasaran Sistem	9
2.3.	Komponen Aplikasi Sistem Informasi	9
2.3.1.	Administrator	9
2.3.2.	Pimpinan Koperasi	9
2.3.3.	Pengurus Koperasi	9
2.4.	Koperasi	10
2.4.1.	Pengertian Koperasi	10

2.4.2.	Sejarah Koperasi	10
2.5.	Prinsip Koperasi	11
2.6.	Simpan Pinjam Koperasi	12
2.6.1.	Unit Simpan Pinjam (USP).....	12
2.6.2.	Simpanan Anggota.....	12
2.6.3.	Simpanan Pokok	12
2.6.4.	Simpanan Wajib.....	12
2.6.5.	Simpanan Sukarela.....	13
2.6.6.	Pinjaman Anggota.....	13
2.6.7.	Sisa Hasil Usaha (SHU).....	13
2.6.8.	Dana Cadangan	14
2.6.9.	Mekanisme Penghitungan Angsuran	14
2.7.	Pengertian PHP	14
2.8.	Pengertian <i>Database</i>	15
2.9.	<i>Object Oriented Programming</i>	15
2.10.	Pengembangan <i>Incremental Model</i>	17
BAB 3. METODOLOGI PENELITIAN.....		23
3.1.	Tujuan Penelitian.....	23
3.2.	Tempat dan Waktu Penelitian	23
3.3.	Metode Penelitian.....	23
3.3.1.	Analisis Kebutuhan Sistem	26
3.3.2.	<i>System Design</i>	29
3.3.3.	Pengkodean Sistem (<i>Coding</i>).....	30

3.3.4.	<i>System Testing</i>	30
BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM.....		32
4.1.	Pengumpulan Data	32
4.1.1.	Data Transaksi Daftar Tagihan Anggota	32
4.1.2.	Data Transaksi Buku Kas Harian.....	33
4.1.3.	Data Transaksi Rekap Daftar Tagihan Anggota	34
4.1.4.	Data Transaksi Bukti Kas Masuk.....	35
4.1.5.	Data Transaksi Bukti Kas Keluar.....	36
4.2.	Kebutuhan Fungsional dan Non-Fungsional.....	36
4.2.1.	Kebutuhan Fungsional	37
4.2.2.	Kebutuhan Non-fungsional.....	37
4.3.	<i>Business Process</i>	38
4.4.	<i>Use Case Diagram</i>	39
4.5.	<i>Use Case Scenario</i>	43
4.5.1.	Perancangan Inkremen 1 <i>Scenario</i> User Login.....	43
4.5.2.	Perancangan Inkremen 1 <i>Scenario</i> Registrasi Anggota.....	45
4.5.3.	Perancangan Inkremen 1 <i>Scenario</i> Transaksi Simpanan	55
4.5.4.	Perancangan Inkremen 1 <i>Scenario</i> Transaksi Pinjaman.....	61
4.5.5.	Perancangan Inkremen 1 <i>Scenario</i> Transaksi Angsuran.....	65
4.5.6.	Perancangan Inkremen 2 <i>Scenario</i> Manajemen Lembaga.....	67
4.5.7.	Perancangan Inkremen 3 <i>Scenario</i> Cetak Laporan	72
4.5.8.	Perancangan Inkremen 4 <i>Scenario</i> Manajemen Pengurus.....	73
4.5.9.	Perancangan Inkremen 5 <i>Scenario</i> Master Data Transaksi	77

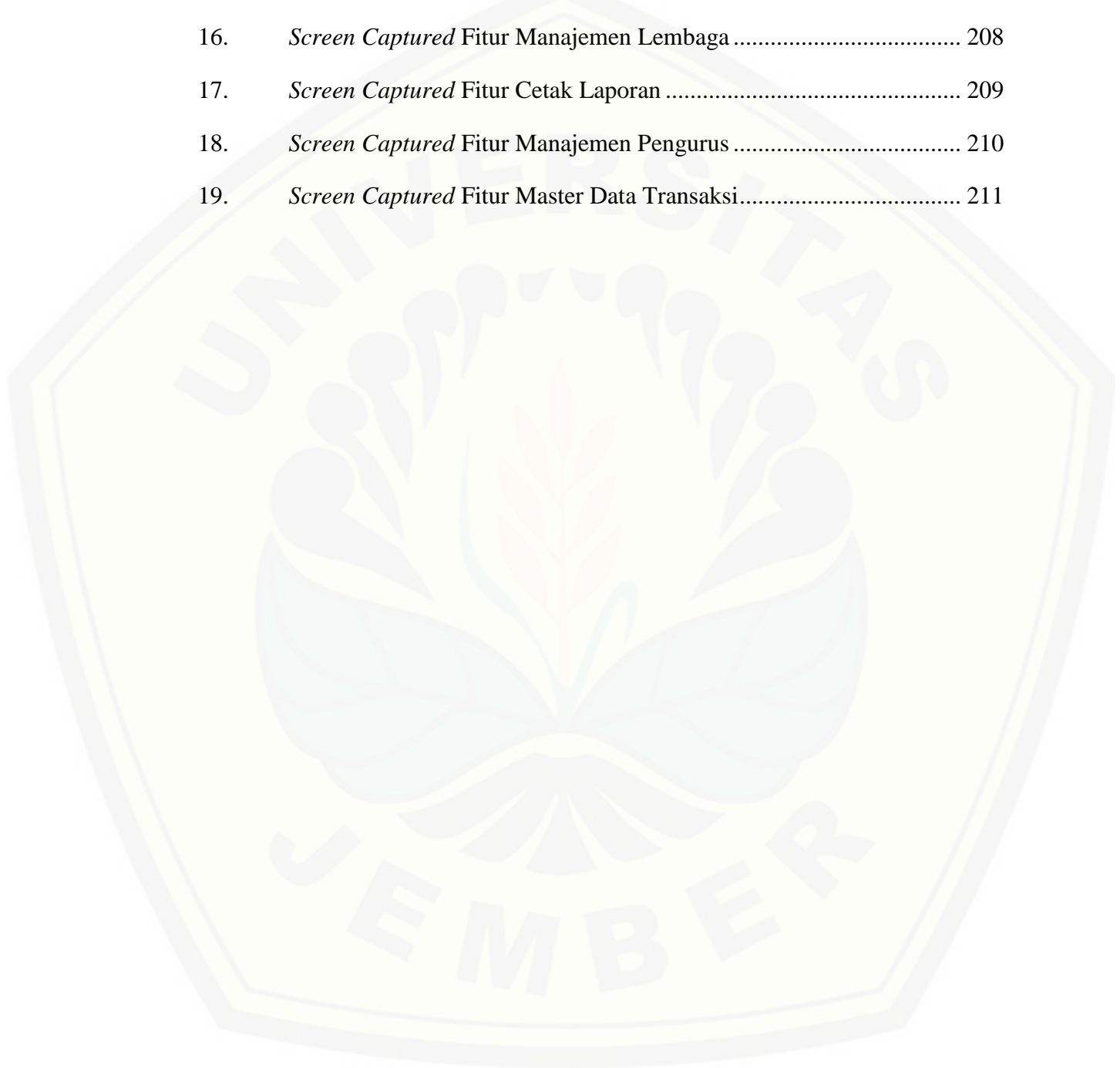
4.6.	<i>Activity Diagram</i>	79
4.6.1.	Perancangan Inkremen 1 <i>Activity</i> User Login.....	79
4.6.2.	Perancangan Inkremen 1 <i>Activity</i> Registrasi Anggota.....	81
4.6.3.	Perancangan Inkremen 1 <i>Activity</i> Transaksi Simpanan	85
4.6.4.	Perancangan Inkremen 1 <i>Activity</i> Transaksi Pinjaman	88
4.6.5.	Perancangan Inkremen 1 <i>Activity</i> Transaksi Angsuran.....	89
4.6.6.	Perancangan Inkremen 2 <i>Activity</i> Manajemen Lembaga.....	90
4.6.7.	Perancangan Inkremen 3 <i>Activity</i> Cetak Laporan	93
4.6.8.	Perancangan Inkremen 4 <i>Activity</i> Manajemen Pengurus	94
4.6.9.	Perancangan Inkremen 5 <i>Activity</i> Master Data Transaksi	96
4.7.	<i>Sequence Diagram</i>	97
4.7.1.	Perancangan Inkremen 1 <i>Sequence</i> User Login.....	98
4.7.2.	Perancangan Inkremen 1 <i>Sequence</i> Registrasi Anggota.....	99
4.7.3.	Perancangan Inkremen 1 <i>Sequence</i> Transaksi Simpanan	104
4.7.4.	Perancangan Inkremen 1 <i>Sequence</i> Transaksi Pinjaman	107
4.7.5.	Perancangan Inkremen 1 <i>Sequence</i> Transaksi Angsuran.....	108
4.7.6.	Perancangan Inkremen 2 <i>Sequence</i> Manajemen Lembaga.....	110
4.7.7.	Perancangan Inkremen 3 <i>Sequence</i> Cetak Laporan	113
4.7.8.	Perancangan Inkremen 4 <i>Sequence</i> Manajemen Pengurus	114
4.7.9.	Perancangan Inkremen 5 <i>Sequence</i> Master Data Transaksi.....	117
4.8.	<i>Class Diagram</i>	119
4.8.1.	Perancangan Inkremen 1 <i>Class Diagram</i> User Login	119
4.8.2.	Perancangan Inkremen 1 <i>Class Diagram</i> Registrasi Anggota.....	120

4.8.3.	Perancangan Inkremen 1 Class Diagram Transaksi Simpanan.....	121
4.8.4.	Perancangan Inkremen 1 <i>Class Diagram</i> Transaksi Pinjaman.....	122
4.8.5.	Perancangan Inkremen 1 <i>Class Diagram</i> Transaksi Angsuran	123
4.8.6.	Perancangan Inkremen 2 <i>Class Diagram</i> Manajemen Lembaga.....	124
4.8.7.	Perancangan Inkremen 3 <i>Class Diagram</i> Cetak Laporan.....	125
4.8.8.	Perancangan Inkremen 4 <i>Class Diagram</i> Manajemen Pengurus.....	126
4.9.	<i>Entity Relationship Diagram</i> (ERD)	128
4.9.1.	Perancangan Inkremen 1 ERD User Login.....	128
4.9.2.	Perancangan Inkremen 1 ERD Registrasi.....	129
4.9.3.	Perancangan Inkremen 1 ERD Transaksi Simpanan	130
4.9.4.	Perancangan Inkremen 1 ERD Transaksi Pinjaman	130
4.9.5.	Perancangan Inkremen 1 ERD Transaksi Angsuran.....	131
4.9.6.	Perancangan Inkremen 2 ERD Manajemen Lembaga	132
4.9.7.	Perancangan Inkremen 3 ERD Cetak Laporan Transaksi.....	132
4.9.8.	Perancangan Inkremen 4 ERD Pengurus	133
4.9.9.	Perancangan Inkremen 5 ERD Master Data Transaksi.....	134
BAB 5 HASIL DAN PEMBAHASAN.....		135
5.1.	Pengujian <i>White Box</i>	135
5.1.1.	Pengujian <i>White Box</i> Inkremen 1 User Login.....	135
5.1.2.	Pengujian <i>White Box</i> Inkremen 1 Registrasi Anggota.....	138
5.1.3.	Pengujian <i>White Box</i> Inkremen 1 Transaksi Simpanan	144
5.1.4.	Pengujian <i>White Box</i> Inkremen 1 Transaksi Pinjaman	148
5.1.5.	Pengujian <i>White Box</i> Inkremen 1 Transaksi Angsuran.....	154

5.1.6.	Pengujian <i>White Box</i> Inkremen 2 Manajemen Lembaga	157
5.1.7.	Pengujian <i>White Box</i> Inkremen 3 Cetak Laporan Transaksi.....	161
5.1.8.	Pengujian <i>White Box</i> Inkremen 4 Manajemen Pengurus	164
5.1.9.	Pengujian <i>White Box</i> Inkremen 5 Master Data Transaksi.....	169
5.2.	Pengujian <i>Black Box</i>	171
5.3.	Implementasi Sistem	178
5.3.1.	Pengkodean User Login	179
5.3.2.	Pengkodean Registrasi Anggota	179
5.3.3.	Pengkodean Transaksi Simpanan Anggota.....	179
5.3.4.	Pengkodean Transaksi Pinjaman Anggota.....	179
5.3.5.	Pengkodean Transaksi Angsuran Anggota	179
5.3.6.	Pengkodean Manajemen Lembaga	179
5.3.7.	Pengkodean Manajemen Pengurus	180
5.3.8.	Pengkodean Master Data Anggota.....	180
5.3.9.	Pengkodean Master Data Simpanan	180
5.3.10.	Pengkodean Master Data Pinjaman	180
5.4.	Pembahasan Sistem	180
5.4.1.	Fitur Login	181
5.4.2.	Fitur Registrasi Anggota	181
5.4.3.	Fitur Transaksi Simpanan	181
5.4.4.	Fitur Transaksi Pinjaman	181
5.4.5.	Fitur Transaksi Angsuran.....	181
5.4.6.	Fitur Manajemen Lembaga	182

5.4.7.	Fitur Cetak Laporan	182
5.4.8.	Fitur Manajemen Pengurus	182
5.4.9.	Fitur Master Data Transaksi.....	182
BAB 6 KESIMPULAN DAN SARAN.....		183
6.1.	Kesimpulan.....	183
6.2.	Saran.....	183
DAFTAR PUSTAKA		184
LAMPIRAN.....		186
Lampiran Implementasi Sistem.....		186
1.	Pengkodean User Login	186
2.	Pengkodean Registrasi Anggota	188
3.	Pengkodean Transaksi Simpanan Anggota.....	192
4.	Pengkodean Transaksi Pinjaman Anggota.....	194
5.	Pengkodean Transaksi Angsuran Anggota	196
6.	Pengkodean Manajemen Lembaga	197
7.	Pengkodean Manajemen Pengurus	199
8.	Pengkodean Master Data Anggota.....	201
9.	Pengkodean Master Data Simpanan	202
10.	Pengkodean Master Data Pinjaman	202
Lampiran Pembahasan Sistem		203
11.	<i>Screen Captured</i> Form Login	203
12.	<i>Screen Captured</i> Fitur Registrasi Anggota	204
13.	<i>Screen Captured</i> Fitur Transaksi Simpanan	205

14.	<i>Screen Captured</i> Fitur Transaksi Pinjaman	206
15.	<i>Screen Captured</i> Fitur Transaksi Angsuran.....	207
16.	<i>Screen Captured</i> Fitur Manajemen Lembaga	208
17.	<i>Screen Captured</i> Fitur Cetak Laporan	209
18.	<i>Screen Captured</i> Fitur Manajemen Pengurus	210
19.	<i>Screen Captured</i> Fitur Master Data Transaksi.....	211



DAFTAR TABEL

	Halaman
Tabel 4.1 Kebutuhan Fungsional	37
Tabel 4.2 Kebutuhan Nonfungsional	38
Tabel 4.3 Deskripsi Aktor	39
Tabel 4.4 Deskripsi <i>Use Case</i>	40
Tabel 4.5 <i>Use Case Scenario</i> User Login (<i>Log-In</i>)	43
Tabel 4.6 <i>Use Case Scenario</i> User Login (<i>Log-Out</i>).....	44
Tabel 4.7 <i>Use Case Scenario</i> Registrasi Anggota (<i>Entry</i>).....	45
Tabel 4.8 <i>Use Case Scenario</i> Registrasi Anggota (<i>Update-Status</i>).....	50
Tabel 4.9 <i>Use Case Scenario</i> Registrasi Anggota (<i>Edit-Update</i>).....	52
Tabel 4.10 <i>Use Case Scenario</i> Transaksi Simpanan Anggota (<i>Entry</i>)	55
Tabel 4.11 <i>Use Case Scenario</i> Transaksi Simpanan Anggota (<i>Edit-Update</i>)	59
Tabel 4.12 <i>Use Case Scenario</i> Transaksi Pinjaman Anggota.....	62
Tabel 4.13 <i>Use Case Scenario</i> Transaksi Simpanan Anggota (<i>Entry</i>)	66
Tabel 4.14 <i>Use Case Scenario</i> Manajemen Lembaga (<i>Entry</i>).....	67
Tabel 4.15 <i>Use Case Scenario</i> Manajemen Lembaga (<i>Edit-Update</i>).....	70
Tabel 4.16 <i>Use Case Scenario</i> Cetak Laporan Transaksi.....	72
Tabel 4.17 <i>Use Case Scenario</i> Manajemen Pengurus (<i>Entry</i>).....	74
Tabel 4.18 <i>Use Case Scenario</i> Manajemen Pengurus (<i>Update-Password</i>).....	76
Tabel 4.19 <i>Use Case Scenario</i> Master Data Transaksi.....	78
Tabel 5.1 Test Case Function userLogin	138
Tabel 5.2 Test Case Function registrasiAnggota	142
Tabel 5.3 Test Case Function entriSimpan	147
Tabel 5.4 Test Case Function entriPinjam.....	152
Tabel 5.5 Test Case Function updateAngsur	156
Tabel 5.6 Test Case Function entriLembaga	160
Tabel 5.7 Test Case Function Content	164
Tabel 5.8 Test Case Function userRegis.....	168
Tabel 5.9 Test Case Function masterAnggota	171
Tabel 5.10 <i>Black Box Testing</i>	172

DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Incremental Model</i>	22
Gambar 3.1 Alir Penelitian	25
Gambar 4.1 Data Transaksi Daftar Tagihan Anggota.....	33
Gambar 4.2 Data Transaksi Buku Kas Harian	34
Gambar 4.3 Data Transaksi Rekap Daftar Tagihan	34
Gambar 4.4 Data Transaksi Bukti Kas Masuk.....	35
Gambar 4.5 Data Transaksi Bukti Kas Keluar.....	36
Gambar 4.6 <i>Business Process</i>	41
Gambar 4.7 <i>Use Case Diagram</i>	42
Gambar 4.8 <i>Activity Diagram</i> User Login (<i>log-In</i>)	80
Gambar 4.9 <i>Activity Diagram</i> User Login (<i>log-Out</i>).....	81
Gambar 4.10 <i>Activity Diagram</i> Registrasi Anggota (<i>Entry</i>).....	82
Gambar 4.11 <i>Activity Diagram</i> Registrasi Anggota (<i>Update-Status</i>).....	83
Gambar 4.12 <i>Activity Diagram</i> Registrasi Anggota (<i>Edit-Update</i>).....	84
Gambar 4.13 <i>Activity Diagram</i> Transaksi Simpanan Anggota (<i>Entry</i>)	86
Gambar 4.14 <i>Activity Diagram</i> Transaksi Simpanan (<i>Edit-Update</i>)	87
Gambar 4.15 <i>Activity Diagram</i> Transaksi Pinjaman Anggota.....	88
Gambar 4.16 <i>Activity Diagram</i> Transaksi Angsuran Anggota	90
Gambar 4.17 <i>Activity Diagram</i> Manajemen Lembaga (<i>Entry</i>).....	91
Gambar 4.18 <i>Activty Diagram</i> Manajemen Lembaga (<i>Edit-Update</i>).....	92
Gambar 4.19 <i>Activity Diagram</i> Cetak Laporan Transaksi.....	93
Gambar 4.20 <i>Activity Diagram</i> Manajemen Pengurus (<i>Entry</i>).....	95
Gambar 4.21 <i>Activity Diagram</i> Manajemen Pengurus (<i>Update-Password</i>).....	96
Gambar 4.22 <i>Activity Diagram</i> Master Data Transaksi.....	97
Gambar 4.23 <i>Sequence Diagram</i> User Login (<i>log-In</i>).....	98
Gambar 4.24 <i>Sequence Diagram</i> Registrasi Anggota (<i>Entry</i>).....	100
Gambar 4.25 <i>Activity Diagram</i> Registrasi Anggota (<i>Update-Status</i>).....	102
Gambar 4.26 <i>Activity Diagram</i> Registrasi Anggota (<i>Edit-Update</i>).....	103

Gambar 4.27 <i>Activity Diagram</i> Transaksi Simpanan Anggota (<i>Entry</i>)	105
Gambar 4.28 <i>Activity Diagram</i> Transaksi Simpanan (<i>Edit-Update</i>)	106
Gambar 4.29 <i>Sequence Diagram</i> Transaksi Pinjaman Anggota	108
Gambar 4.30 <i>Sequence Diagram</i> Transaksi Angsuran Anggota	109
Gambar 4.31 <i>Sequence Diagram</i> Manajemen Lembaga (<i>Entry</i>)	111
Gambar 4.32 <i>Sequence Diagram</i> Manajemen Lembaga (<i>Edit-Update</i>)	112
Gambar 4.33 <i>Sequence Diagram</i> Cetak Laporan Transaksi	114
Gambar 4.34 <i>Sequence Diagram</i> Manajemen Pengurus (<i>Entry</i>)	115
Gambar 4.35 <i>Sequence Diagram</i> Manajemen Pengurus (<i>Update-Password</i>)	116
Gambar 4.36 <i>Sequence Diagram</i> Master Data Transaksi	118
Gambar 4.37 <i>Class Diagram</i> User Login	119
Gambar 4.38 <i>Class Diagram</i> Registrasi Anggot	120
Gambar 4.39 <i>Class Diagram</i> Transaksi Simpanan Anggota	121
Gambar 4.40 <i>Class Diagram</i> Transaksi Pinjaman Anggota	122
Gambar 4.41 <i>Class Diagram</i> Transaksi Angsuran Anggota	123
Gambar 4.42 <i>Class Diagram</i> Manajemen Lembaga	124
Gambar 4.43 <i>Class Diagram</i> Cetak Laporan Transaksi	125
Gambar 4.44 <i>Class Diagram</i> Manajemen Pengurus	127
Gambar 4.45 <i>Class Diagram</i> Master Data Transaksi	128
Gambar 4.46 <i>Entity Relationship Diagram</i> User Login	129
Gambar 4.47 <i>Entity Relationship Diagram</i> Registrasi Anggota	129
Gambar 4.48 <i>Entity Relationship Diagram</i> Transaksi Simpanan Anggota	130
Gambar 4.49 <i>Entity Relationship Diagram</i> Transaksi Pinjaman Anggota	131
Gambar 4.50 <i>Entity Relationship Diagram</i> Transaksi Angsuran Anggota	131
Gambar 4.51 <i>Entity Relationship Diagram</i> Manajemen Lembaga	132
Gambar 4.52 <i>Entity Relationship Diagram</i> Cetak Laporan Transaksi	133
Gambar 4.53 <i>Entity Relationship Diagram</i> Manajemen Pengurus	133
Gambar 4.54 <i>Entity Relationship Diagram</i> Master Data Transaksi	134
Gambar 5.1 Listing Program <i>Function</i> userLogin	136
Gambar 5.2 Diagram Alir <i>Function</i> userLogin	136

Gambar 5.3 Grafik Alir <i>Function</i> userLogin	137
Gambar 5.4 Listing Program <i>Function</i> registrasiAnggota.....	140
Gambar 5.5 Diagram Alir <i>Function</i> registrasiAnggota	141
Gambar 5.6 Grafik Alir <i>Function</i> registrasiAnggota.....	141
Gambar 5.7 Listing Program <i>Function</i> entriSimpan.....	145
Gambar 5.8 Diagram Alir <i>Function</i> entriSimpan	145
Gambar 5.9 Grafik Alir <i>Function</i> entriSimpan.....	146
Gambar 5.10 Listing Program <i>Function</i> entriPinjam.....	150
Gambar 5.11 Diagram Alir <i>Function</i> entriPinjam	150
Gambar 5.12 Grafik Alir <i>Function</i> entriPinjam.....	151
Gambar 5.13 Listing Program <i>Function</i> updateAngsur.....	154
Gambar 5.14 Diagram Alir <i>Function</i> updateAngsur	155
Gambar 5.15 Grafik Alir <i>Function</i> updateAngsur	155
Gambar 5.16 Listing Program <i>Function</i> entriLembaga.....	158
Gambar 5.17 Diagram Alir <i>Function</i> entriLembaga.....	159
Gambar 5.18 Grafik Alir <i>Function</i> entriLembaga	159
Gambar 5.19 Listing Program <i>Function</i> Content.....	162
Gambar 5.20 Diagram Alir <i>Function</i> Content	163
Gambar 5.21 Grafik Alir <i>Function</i> Content.....	163
Gambar 5.22 Listing Program <i>Function</i> userRegis	166
Gambar 5.23 Diagram Alir <i>Function</i> userRegis	166
Gambar 5.24 Grafik Alir <i>Function</i> userRegis.....	167
Gambar 5.25 Listing Program <i>Function</i> masterAnggota.....	170
Gambar 5.26 Diagram Alir <i>Function</i> masterAnggota	170
Gambar 5.27 Grafik Alir <i>Function</i> masterAnggota	170

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Pengkodean User Login	186
Lampiran 2 Pengkodean Registrasi Anggota <i>Entry</i>	188
Lampiran 3 Pengkodean Registrasi Anggota <i>Update Status</i>	190
Lampiran 4 Pengkodean Registrasi Anggota <i>Edit-Update</i>	191
Lampiran 5 Pengkodean Simpanan <i>Entry</i>	192
Lampiran 6 Pengkodean Simpanan <i>Edit</i>	193
Lampiran 7 Pengkodean Pinjaman	194
Lampiran 8 Pengkodean Angsuran	196
Lampiran 9 Pengkodean Lembaga <i>Entry</i>	197
Lampiran 10 Pengkodean Lembaga <i>Update</i>	198
Lampiran 11 Pengkodean Pengurus <i>Entry</i>	199
Lampiran 12 Pengkodean Pengurus <i>Update Password</i>	200
Lampiran 13 Pengkodean Master Data Anggota	201
Lampiran 14 Pengkodean Master Data Simpanan.....	202
Lampiran 15 Pengkodean Master Data Pinjaman.....	202
Lampiran 16 <i>Screen Captured</i> Form Login.....	203
Lampiran 17 <i>Screen Captured</i> Registrasi Anggota	204
Lampiran 18 <i>Screen Captured</i> Registrasi Anggota Form.....	204
Lampiran 19 <i>Screen Captured</i> Transaksi Simpanan	205
Lampiran 20 <i>Screen Captured</i> Transaksi Simpanan Form.....	205
Lampiran 21 <i>Screen Captured</i> Transaksi Pinjaman	206
Lampiran 22 <i>Screen Captured</i> Transaksi Pinjaman Form.....	206
Lampiran 23 <i>Screen Captured</i> Transaksi Angsuran.....	207
Lampiran 24 <i>Screen Captured</i> Transaksi Angsuran Form	207
Lampiran 25 <i>Screen Captured</i> Manajemen Lembaga	208
Lampiran 26 <i>Screen Captured</i> Manajemen Lembaga Form.....	208
Lampiran 27 <i>Screen Captured</i> Cetak Laporan	209
Lampiran 28 <i>Screen Captured</i> Cetak Laporan <i>Preview</i>	209
Lampiran 29 <i>Screen Captured</i> Manajemen Pengurus	210
Lampiran 30 <i>Screen Captured</i> Manajemen Pengurus Form.....	210
Lampiran 31 <i>Screen Captured</i> Master Data Transaksi.....	211
Lampiran 32 <i>Screen Captured</i> Master Data Transaksi <i>Preview</i>	211

BAB I PENDAHULUAN

Bab ini merupakan awal dari penulisan tugas akhir ini. Berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

1.1. Latar Belakang

Koperasi Pegawai Republik Indonesia (KP-RI) “DIAN PATIRANA” merupakan sebuah unit usaha keuangan bukan bank, yang yang melayani bidang simpan pinjam dan juga bertugas memberikan pelayanan masyarakat berupa pinjaman dan tempat penyimpanan uang bagi masyarakat. Produk barang yang disediakan oleh koperasi berupa bahan-bahan pokok, beras, gula pasir, minyak goreng, alat elektronika, perlengkapan rumah tangga, alat tulis, dan kendaraan bermotor.

Bidang yang terdapat pada Koperasi Pegawai Republik Indonesia (KP-RI) “DIAN PATIRANA” ada beberapa macam, yang mana bagian tersebut salah satunya penulis ambil sebagai bahan penelitian untuk penyelesaian tugas akhir. Bagian yang diteliti oleh penulis terfokus pada bagian bendahara, dibidang tersebut bertugas untuk menangani tentang proses simpan pinjam yang ada pada Koperasi Pegawai Republik Indonesia (KP-RI) “DIAN PATIRANA”.

Koperasi “DIAN PATIRANA” memiliki produk berupa jasa seperti unit simpanan anggota meliputi simpanan anggota berupa tabungan, simpanan wajib keanggotaan, simpanan sukarela hingga produk jasa berupa unit simpan-pinjam (USP). Khususnya pada produk unit simpan pinjam dan simpanan anggota, proses mengentrikan data transaksi dan administrasinya yang terjadi di dalam koperasi masih belum dilakukan secara komputasi, namun masih sederhana. Prosesnya dengan cara mengentrikan membuat laporan transaksi kedalam MS Excel, kemudian dicatat ke dalam buku. Sehingga data-data yang ada menjadi tidak terorganisir, dapat membuang waktu yang cukup lama, memproses data transaksi dan penyimpanan

dokumen menjadi tidak efektif dan efisien. Bertitik tolak dengan hal tersebut, beberapa fenomena yang terkait dengan administrasi simpan-pinjam pada koperasi, maka penelitian yang diambil penulis dengan judul “Perancangan dan Pembuatan Sistem Informasi Simpan Pinjam Pada Koperasi Pegawai Republik Indonesia (KP-RI) “DIAN PATIRANA” KECAMATAN GRUJUGAN KABUPATEN BONDOWOSO Berbasis *Web*.”

1.2. Perumusan Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan sebelumnya, maka didapatkan rumusan masalah yang akan dibahas pada penelitian ini adalah:

1. Mendesain dan merancang sebuah sistem untuk memproses data simpan-pinjam pada koperasi ‘DIAN PATIRANA’ sesuai dengan kebutuhan user.
2. Merancang sebuah aplikasi berbasis *web* sehingga dapat memudahkan pencatatan dan pemrosesan transaksi simpan-pinjam.
3. Mengimplementasikan rancangan yang telah dibuat menjadi sebuah perangkat lunak yang direncanakan sebelumnya.

1.3. Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penulisan tugas akhir ini adalah:

1. Menghasilkan sebuah desain sistem yang lebih terstruktur secara komputasi yang dapat memaksimalkan proses kinerja dan transaksi simpan-pinjam pada koperasi menjadi lebih efektif penggunaannya dan lebih efisien dalam pengerjaan pemrosesan data transaksi simpan-pinjam.
2. Menghasilkan sebuah rancangan aplikasi yang dapat memudahkan pengguna dalam melakukan pencatatan dan pemrosesan data transaksi yang terjadi dalam sistem.

3. Menghasilkan sebuah sistem berupa perangkat lunak yang dapat memudahkan pengguna dalam melakukan pencatatan dan pemrosesan transaksi yang terjadi di dalam sistem.

1.4. Batasan Masalah

Dalam membuat sistem ini untuk menentukan bagaimana proses yang dilakukan sistem yang telah dibuat sesuai kebutuhan dan sesuai dengan tema yang dibahas, maka penulis memberikan batasan – batasan pada permasalahannya. Berikut adalah batasan masalah terhadap permasalahan yang diteliti:

1. Bahasa pemrograman yang digunakan untuk mengkodekan sistem adalah *PHP (Hypertext Processor)* dan manajemen database menggunakan *MySQL*.
2. Tidak membahas tentang jaringan komputerisasi dan keamanan data pada sistem yang dibuat.
3. Ruang lingkup sistem hanya pada transaksi simpanan anggota, transaksi unit simpan pinjam yang berupa barang dan jasa, transaksi angsuran yang berkaitan, laporan – laporan transaksi baik berupa rekap maupun kwitansi, manajemen data yang berkaitan dengan sistem.

1.5. Manfaat Penelitian

Manfaat yang ingin didapatkan dari pembuatan sistem ini adalah:

1.5.1. Bagi Akademis

Memberikan informasi dan bagan referensi bagi dunia pendidikan, khususnya bagi yang membutuhkan informasi terkait dengan judul penelitian ini.

1.5.2. Bagi Koperasi Pegawai “DIAN PATIRANA”

Memberikan keuntungan berupa kontribusi dan masukan terhadap pihak koperasi dalam menunjang penerapan sistem yang telah dikerjakan oleh penulis.

1.5.3. Bagi Peneliti

Sebagai salah satu syarat menyelesaikan program Sistem Informasi S1 Program Studi Sistem Informasi Universitas Jember.

1.6. Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut:

BAB 1. PENDAHULUAN

Bab ini terdiri dari latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB 2. TINJAUAN PUSTAKA

Bab ini berisi tentang kajian materi yang digunakan untuk menyelesaikan tugas akhir ini. Dimulai dari object penelitian sampai dengan segala kebutuhan materi yang diperlukan.

BAB 3. METODOLOGI PENELITIAN

Bab ini berisi tentang jenis metode perancangan yang digunakan dan apa yang akan dilakukan oleh penulis. Dimulai dari perencanaan hingga proses pembuatannya.

BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM

Bab ini berisi desain dan perancangan sistem yang digunakan oleh penulis untuk membangun sistem.

BAB 5. HASIL DAN PEMBAHASAN

Bab ini berisi tentang pengujian terhadap sistem yang telah dibangun. Pengujian tersebut meliputi *black box* dan *white box*.

BAB 6. KESIMPULAN DAN SARAN

Bab ini berisi penutup yang merupakan kesimpulan dan saran dari rumusan masalah yang diajukan.

BAB 2. TINJAUAN PUSTAKA

Bab ini menjelaskan pustaka yang berkaitan dengan dasar teori yang mendukung permasalahan, baik pustaka yang berkaitan dengan penelitian terdahulu maupun yang akan diteliti. Dalam merancang dan mengimplementasikan aplikasi koperasi, penulis mengacu pada materi pendukung yang dapat dijadikan landasan teori atau tinjauan pustaka. Tinjauan pustaka dalam arti lebih spesifik merupakan peninjauan kembali pustaka atau laporan penelitian sebelumnya terkait dengan penelitian yang dilakukan pada karya ilmiah ini. Tinjauan pustaka tersebut masing-masing akan dijabarkan pada subbab berikut ini:

2.1. Konsep Dasar Sistem Informasi

2.1.1. Konsep Dasar Sistem

Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Ferdinand Magaline, 2009). Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum, yaitu :

1. Setiap sistem terdiri dari unsur-unsur
2. Unsur-unsur tersebut merupakan bagian terpadu sistem yang bersangkutan
3. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem
4. Suatu sistem merupakan bagian dari sistem lain yang lebih besar

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Dari definisi ini dapat dirinci lebih lanjut pengertian sistem secara umum.

2.1.2. Konsep Dasar Informasi

Informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan (Ferdinand Magaline, 2009). Informasi secara umum dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

2.1.3. Konsep Dasar Sistem Informasi

Informasi merupakan salah satu sumber daya penting dalam suatu organisasi digunakan sebagai bahan pengambilan keputusan (Kadir & Triwahyuni, 2005). Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya (Ferdinand Magaline, 2009). Secara umum definisi dari Sistem Informasi adalah sekelompok elemen-elemen dalam suatu organisasi yang saling berintegrasi dengan menggunakan masukan, proses dan keluaran dengan maksud yang sama untuk mencapai suatu tujuan dan dapat digunakan untuk membantu pengambilan keputusan yang tepat (Whitten, 2004).

2.2. Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu (Jerry Fithgerald, 2009).

Selain itu suatu sistem memiliki memiliki karakteristik- karakteristik yang patut diketahui (Jogiyanto, 2005):

2.2.1. Memiliki Komponen:

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem Dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem, misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem.

2.2.2. Batas Sistem (*Boundary*):

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

2.2.3. Lingkungan Luar Sistem (*Environment*):

Lingkungan luar sistem atau *environment* adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap

dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

2.2.4. Penghubung Sistem (*Interface*):

Penghubung sistem atau *interface* merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran (*output*) dari satu subsistem akan menjadi satu masukan (*input*) bagi subsistem yang lain dan akan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan

2.2.5. Masukan Sistem (*Input*):

Masukan sistem merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

2.2.6. Keluaran Sistem (*Output*):

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Dengan kata lain, merupakan hasil dari energi yang diolah oleh sistem.

2.2.7. Pengolahan Sistem (*Process*):

Pengolahan sistem adalah suatu sistem dapat mempunyai bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan

menjadi keluaran. Atau bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

2.2.8. Sasaran Sistem:

Sasaran sistem adalah ketika sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Suatu sistem harus mempunyai sasaran, karena sasaran sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.3. Komponen Aplikasi Sistem Informasi

2.3.1. Administrator

Seorang admin memiliki peranan yang sangat penting pada sebuah aplikasi koperasi. Tugas dari seorang admin antara lain manajemen data user yang mengakses sistem dan *maintenance* web sistem.

2.3.2. Pimpinan Koperasi

Seorang pimpinan koperasi memiliki akses seperti melihat master data, laporan, serta informasi secara mendetail dan menyeluruh. Guna untuk memonitoring data transaksi yang ada pada sistem.

2.3.3. Pengurus Koperasi

Pengurus koperasi pada sistem yang dibuat memiliki tugas untuk melakukan transaksi-transaksi yang terkait hingga memproses laporan dan kwitansi.

2.4. Koperasi

Koperasi adalah organisasi ekonomi rakyat yang berwatak sosial. Dengan perkataan lain suatu Koperasi adalah satu bentuk organisasi ekonomi rakyat (UU No. 25 tahun 1992).

2.4.1. Pengertian Koperasi

Koperasi adalah organisasi bisnis yang dimiliki dan dioperasikan oleh orang-seorang demi kepentingan bersama. Koperasi melandaskan kegiatan berdasarkan prinsip gerakan ekonomi rakyat yang berdasarkan asas kekeluargaan. Penjelasan UUD 1945 menyatakan bahwa bangunan usaha yang sesuai dengan kepribadian bangsa Indonesia adalah koperasi. Koperasi merupakan gerakan ekonomi rakyat yang dijalankan berdasarkan asas kekeluargaan. Inti dari koperasi adalah kerja sama, yaitu kerja sama di antara anggota dan para pengurus dalam rangka mewujudkan kesejahteraan anggota dan masyarakat serta membangun tatanan perekonomian nasional. Sebagai gerakan ekonomi rakyat, koperasi bukan hanya milik orang kaya melainkan juga milik oleh seluruh rakyat Indonesia tanpa terkecuali. Berikut ini adalah landasan koperasi Indonesia yang melandasi aktifitas koperasi di Indonesia: Landasan Idiil (pancasila), Landasan Mental (Setia kawan dan kesadaran diri sendiri), dan Landasan Struktural dan gerak (UUD 1945 Pasal 33 Ayat 1).

2.4.2. Sejarah Koperasi

Sejarah singkat gerakan koperasi bermula pada abad ke-20 yang pada umumnya merupakan hasil dari usaha yang tidak spontan dan tidak dilakukan oleh orang-orang yang sangat kaya. Koperasi tumbuh dari kalangan rakyat, ketika penderitaan dalam lapangan ekonomi dan sosial yang ditimbulkan oleh sistem kapitalisme semakin memuncak. Beberapa orang yang penghidupannya sederhana dengan kemampuan ekonomi terbatas, terdorong oleh penderitaan dan beban ekonomi yang sama, secara spontan mempersatukan diri untuk menolong dirinya

sendiri dan manusia sesamanya. Koperasi dikenalkan di Indonesia oleh seorang Pamong Praja Patih R.Aria Wiria Atmaja di Purwokerto, Jawa Tengah pada tahun 1896. Pada tanggal 12 juli 1947, pergerakan koperasi di Indonesia mengadakan kongres koperasi yang pertama di Tasikmalaya. Tanggal dilaksanakannya kongres ini kemudian ditetapkan sebagai Hari Koperasi Indonesia.

2.5. Prinsip Koperasi

Prinsip koperasi adalah suatu sistem ide-ide abstrak yang merupakan petunjuk untuk membangun koperasi yang efektif dan tahan lama. Prinsip koperasi terbaru yang dikembangkan *International Cooperative Alliance* (Federasi koperasi non-pemerintah internasional) adalah: Keanggotaan yang bersifat terbuka-sukarela, Pengelolaan yang demokratis, Partisipasi anggota dalam ekonomi, Kebebasan dan otonomi, dan Pengembangan pendidikan, pelatihan, dan informasi. Koperasi melaksanakan prinsip Koperasi sebagai berikut:

1. Keanggotaan bersifat sukarela dan terbuka.
2. Pengelolaan dilakukan secara demokratis.
3. Pembagian sisa hasil usaha dilakukan secara adil sebanding dengan besarnya jasa usaha masing-masing anggota.
4. Pemberian balas jasa yang terbatas terhadap modal.
5. Kemandirian.

Dalam mengembangkan koperasi, maka koperasi melaksanakan pula prinsip koperasi (Kusnadi, 2005) sebagai berikut:

1. Pendidikan perkoperasian.
2. Kerja sama antar koperasi.

2.6. Simpan Pinjam Koperasi

Koperasi Simpan Pinjam adalah koperasi yang kegiatannya hanya usaha simpan pinjam. Kegiatan usaha simpan pinjam adalah kegiatan yang dilakukan untuk menghimpun dana dan menyalurkannya melalui kegiatan usaha simpan pinjam dari dan untuk anggota koperasi yang bersangkutan, calon anggota koperasi yang bersangkutan, koperasi lain dan atau anggotanya. Kegiatan usaha Koperasi Simpan Pinjam dijalankan oleh sekumpulan orang yang disebut unit simpan pinjam.

2.6.1. Unit Simpan Pinjam (USP):

Unit simpan pinjam adalah unit koperasi yang bergerak di bidang usaha simpan pinjam, sebagai bagian dari kegiatan usaha Koperasi yang bersangkutan.

2.6.2. Simpanan Anggota:

Simpanan anggota adalah dana yang dipercayakan oleh anggota, calon anggota, koperasi-koperasi lain dan atau anggotanya kepada koperasi dalam bentuk tabungan.

2.6.3. Simpanan Pokok:

Simpanan pokok adalah sejumlah uang yang wajib disetorkan ke dalam kas koperasi oleh para pendiri atau anggota koperasi pada saat masuk menjadi anggota. Simpanan pokok tidak dapat ditarik kembali oleh anggota koperasi tersebut selama yang bersangkutan masih tercatat menjadi anggota koperasi.

2.6.4. Simpanan Wajib:

Simpanan wajib adalah konsekuensi dari simpanan ini adalah harus dilakukan oleh semua anggota koperasi yang dapat disesuaikan besar kecilnya dengan tujuan usaha koperasi dan kebutuhan dana yang hendak dikumpulkan, arena itu akumulasi

simpanan wajib para anggota harus diarahkan mencapai jumlah tertentu agar dapat menunjang kebutuhan dana yang akan digunakan menjalankan usaha koperasi.

2.6.5. Simpanan Sukarela:

Simpanan sukarela yaitu sama halnya simpanan wajib, akan tetapi pada simpanan sukarela nominal dan kesediaan untuk melakukan sesuai kemampuan oleh anggota yang melakukan simpanan sukarela tanpa adanya paksaan dari pihak lain.

2.6.6. Pinjaman Anggota:

Pinjaman anggota adalah penyediaan uang atau tagihan yang dapat dipersamakan dengan itu, berdasarkan persetujuan atau kesepakatan pinjam-meminjam antara Koperasi dengan pihak lain yang mewajibkan pihak peminjam untuk melunasi hutangnya setelah jangka waktu tertentu disertai dengan pembayaran sejumlah imbalan.

2.6.7. Sisa Hasil Usaha (SHU):

Sisa hasil usaha yaitu pendapatan hasil usaha dan pendapatan lainnya diperoleh dalam satu tahun, dengan dikurangi biaya penyusutan dan biaya yang dikeluarkan dalam satu tahun. SHU digunakan untuk dana cadangan yang diperlukan untuk kebutuhan koperasi dan keperluan lain.

SHU diperuntukkan sebagai berikut:

1. 50% Dana anggota sesuai jasa masing –masing
2. 10% Dana pengurus
3. 5% Dana kesejahteraan karyawan
4. 5% Dana pendidikan
5. 5% Dana social
6. 25% Dana cadangan

2.6.8. Dana Cadangan:

Dana cadangan adalah sejumlah uang yang diperoleh dari sebagian hasil usaha yang tidak dibagikan kepada anggota. Tujuannya adalah untuk memupuk modal sendiri yang dapat digunakan sewaktu-waktu apabila koperasi membutuhkan dana secara mendadak atau menutup kerugian dalam usaha.

2.6.9. Mekanisme Penghitungan Angsuran:

Untuk penghitungan angsuran pinjaman yang dilakukan oleh anggota koperasi dijabarkan sebagai berikut:

Contoh skenario angsuran:

Joni Handoko mengajukan pinjaman kepada koperasi “Dian Patirana” sebesar Rp 10.000.000,- yang diangsur selama 5 bulan sebagai pinjaman nasabah. Sesuai peraturan pihak koperasi yang berlaku, bunga angsuran dikenai 1,5%. Berdasarkan kesepakatan bersama, penghitungan angsuran sebagai berikut:

1. $\text{Rp } 10.000.000 \times 1,5\% = \text{Rp } 150.000$ (bunga pinjaman/bulan)
2. $\text{Rp } 10.000.000 : 5 = \text{Rp } 2.000.000$
3. $\text{Rp } 2.000.000 + \text{Rp } 150.000$

Maka total angsuran yang harus dibayarkan sebesar Rp 2.150.000 /bulan selama 5 kali sesuai kesepakatan yang sudah disetujui oleh kedua belah pihak sebelumnya.

2.7. Pengertian PHP

Proses pembuatan dan perancangan sistem membutuhkan sebuah perangkat, penulis menggunakan satu bahasa pemrograman sebagai perangkat dari komponen-komponen tersebut yaitu: PHP. Menurut Ali (Zaki, 2008), PHP adalah sebuah bahasa

pemrograman *scripting* untuk membuat halaman *web* yang dinamis. *Website* yang menggunakan PHP memerlukan *software* yang bernama *webserver* sebagai tempat pemrosesan kode PHP tersebut. Server *web* yang memiliki *software* PHP *parser* akan memproses *input* berupa kode PHP dan menghasilkan *output* berupa halaman web. Dengan mengacu pada landasan teori tersebut, penulis menilai bahwa PHP merupakan pilihan bahasa pemrograman yang sangat tepat untuk membuat Sistem Informasi berbasis *web*.

2.8. Pengertian Database

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer dan dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Menurut Okta (Oktaviani, 2010), aplikasi yang menggunakan *database* ialah aplikasi yang didesain khusus untuk digunakan oleh user atau pihak lainnya seperti penyedia jasa pemrograman atau konsultan. Basis data merupakan aspek yang sangat penting dalam sistem informasi dimana basis data merupakan gudang penyimpanan data yang akan diolah lebih lanjut.

2.9. Object Oriented Programming

Object oriented merupakan suatu strategi pembangunan perangkat lunak (perencanaan, analisis, perancangan dan implementasi) yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Keuntungan / manfaat utama menggunakan dari Object Oriented yakni (Munawar, 2005):

1. *Uniformity*

Penembang cukup menggunakan satu metodologi dari tahap analisis hingga perancangan. Dengan adanya perkembangan ke arah aplikasi GUI (*graphical user interface*), OMT memungkinkan merancang user interface secara terintegrasi bersama dengan perancangan perangkat lunak sekaligus dengan perancangan basis data

2. *Understandability*

Kode – kode yang dihasilkan dapat diorganisasi ke dalam kelas – kelas yang berhubungan dengan masalah sesungguhnya sehingga lebih mudah dipahami.

3. *Stability*

Kode program yang dihasilkan relatif sebab mendekati permasalahan sesungguhnya di lapangan.

4. *Reusability*

Object Oriented Analysis Design atau yang biasa disingkat OOAD merupakan metode untuk menganalisis dan merancang sistem dengan pendekatan yang berorientasi objek. Objek yang dimaksud merupakan suatu entitas yang memiliki *state*, dan *behavior*. Dapat dicontohkan seperti sebuah mobil yang berperan sebagai entitas. Mobil tersebut memiliki *state* seperti berwarna kuning, berbentuk sedan. Sedangkan *behavior* mobil tersebut dapat bergerak maju dan mundur. (Coad & Yourdon, 1991).

Menurut Bitter, Rick et al (2001, dalam *Object-Oriented Programming in LabVIEW*) Object Oriented memiliki 3 (tiga) ciri, yaitu *encapsulation*, *inheritance*, dan *polymorphism*:

a. *Encapsulation*

Encapsulation merupakan cara untuk menyembunyikan data sehingga data pada suatu objek tidak dapat dimanipulasi dari luar objek. Variabel global merupakan suatu permasalahan yang sangat besar pada aplikasi dengan skala besar. Setiap variabel global jika digunakan pada bermacam – macam fungsi akan menimbulkan kesulitan. Dengan pemrograman berbasis objek dapat mencegah dan memecahkan permasalahan tersebut dengan cara enkapsulasi data. Data yang dienkapsulasi dan tidak dapat diakses dari luar fungsi merupakan “*private data*”.

b. *Inheritance*

Inheritance merupakan kemampuan untuk mewariskan kemampuannya kepada turunannya. Contohnya *software engineer* merupakan turunan dari *engineer*. Ketika

suatu *class* mewariskan kepada *class* yang lainya maka identitas yang dimiliki *class* tersebut akan diwariskan kepada *class* turunanya. *Class* yang mewariskan merupakan “*superclass*” dan yang diwariskan merupakan “*subclass*”.

c. *Polymorphism*

Polymorphism merupakan kemampuan untuk membuat *variable*, fungsi atau objek untuk memiliki lebih dari satu bentuk.

2.10. Pengembangan *Incremental Model*

Pembuatan sistem informasi simpan pinjam koperasi ini menggunakan pengembangan *incremental model*. *Incremental model* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode *incremental* ini memungkinkan *project* yang dikerjakan akan diselesaikan dengan segera, karena sistem yang akan dibuat dibagi – bagi menjadi beberapa modul yang dikerjakan secara terpisah. Jadi ketika terjadi penambahan kebutuhan fungsional akan lebih mudah diverifikasi. Dalam model ini pembangunan sebuah sistem dilakukan secara bertahap dengan menggunakan terlebih dahulu bagian yang telah selesai dibangun tanpa harus menunggu sampai sepenuhnya selesai dibangun. Selain itu *incremental model* akan mengurangi beban pekerjaanya dikarenakan setiap modul akan dilakukan pengujian, verifikasi, serta validasi sehingga perubahan yang akan terjadi pada sistem tidak terlalu besar (Proboyekti, 2008).

Incremental model memiliki beberapa kelebihan dan kekurangan, yaitu:

1. Merupakan model dengan manajemen yang sederhana.
2. Pelanggan tidak perlu menunggu sampai seluruh sistem dikirim untuk mengambil keuntungan dari sistem tersebut. *Increment* yang pertama sdah memenuhi persyaratan mereka yang paling kritis, sehingga perangkat lunak dapat segera digunakan.

3. Pelanggan dapat memakai *increment* yang pertama sebagai bentuk *prototype* dan mendapatkan pengalaman yang dapat menginformasikan persyaratan untuk *increment* sistem berikutnya.
4. Resiko kegagalan proyek secara keseluruhan lebih rendah. Walaupun masalah dapat ditemukan pada beberapa *increment*, bisa saja beberapa *increment* diserahkan dengan sukses kepada pelanggan.
5. Karena layanan dengan prioritas tertinggi diserahkan pertama dan *increment* berikutnya diintegrasikan dengannya, sangatlah penting bahwa layanan sistem yang paling penting mengalami pengujian yang paling ketat. Ini berarti bahwa pelanggan akan memiliki kemungkinan kecil untuk memenuhi kegagalan perangkat lunak pada *increment* sistem yang paling kecil.

Sedangkan kekurangan penggunaan *incremental model* adalah:

1. *Increment* harus relative lebih kecil (tidak lebih dari 20.000 baris kode) dan setiap *increment* harus menyediakan sebagian dari fungsional sistem.
2. Adanya kesulitan untuk memetakan persyaratan pelanggan pada *increment* dengan ukuran yang benar.

Pada gambar 2.1. dapat dijelaskan tahap – tahap *model incremental* sebagai berikut:

1. Analisis Sistem (*System Analysis*)

Pada tahapan ini dilakukan analisis kebutuhan dan perancangan sistem untuk merumuskan solusi yang tepat dalam pembuatan aplikasi serta kemungkinan yang dilakukan untuk mengimplementasikan rancangan tersebut. Analisa kebutuhan berupa fitur-fitur yang akan dibangun pada sistem serta data yang dibutuhkan dalam proses pengembangan sistem.

2. Perancangan Sistem (*Systems Design*)

Pada tahap ini, penulis sebagai *developer* mulai mendesain sebuah sistem. Data yang sebelumnya telah dikumpulkan oleh pihak *developer* akan dikumpulkan

menjadi modul – modul yang dibangun dengan menggunakan konsep *Object-Oriented Programming* (OOP) dengan berlandaskan pada prinsip penataan MVC (*model, view, controller*).

OOP (*Object Oriented Programming*) adalah sebuah metodologi atau cara berpikir dalam melakukan pemrograman dimana pendefinisian tipe data disertai dengan pendefinisian fungsi. Struktur data yang seperti ini disebut dengan istilah *object*. Paradigma pemrograman OOP dapat dilihat sebagai interaksi sebuah *object* dalam melakukan tugasnya. Selanjutnya pada penerapan OOP dalam sistem, penulis melakukan beberapa langkah perancangan antara lain:

a. *Business Process*

Business process didefinisikan sebagai kumpulan dari proses dan berisi kumpulan aktifitas (*tasks*) yang saling berelasi satu sama lain untuk menghasilkan suatu keluaran yang mendukung pada tujuan dan sasaran strategis dari organisasi. Digambarkan dengan *flowchart* yang dimulai dari urutan kerja awal hingga akhir, informasi pendukung, *input-output*, *resources* dan *goal*.

b. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Menggambarkan kebutuhan sistem dari sudut pandang *user*. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

c. *Use Case Scenario*

Use case scenario merupakan penggambaran lebih jelas dari sebuah *use case diagram* sehingga dengan membaca skenario, *user* dapat mengetahui lebih jelas sistem yang akan dibuat.

d. *Activity Diagram*

Activity diagram menggambarkan aktifitas dalam sebuah proses yang dibuat berdasarkan sebuah atau beberapa use case pada use case diagram. Atau berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

e. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

f. *Class Diagram*

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungannya sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain atau kelas-kelas dan paket-paket yang berelasi di dalam *system*.

g. *Entity Relationship Diagram (ERD)*

Entity relationship diagram menggambarkan entitas – entitas mana yang sebaiknya secara konseptual sebaliknya dihubungkan dengan entitas yang lain. Hubungan antar entitas tidak ditentukan oleh *field - field* data yang sama dalam masing – masing entitas.

3. Coding

Pada tahap implementasi atau bisa disebut dengan fase pengkodean sistem. Tahapan yang telah dilakukan pada fase sebelumnya kemudian diimplementasikan kedalam bahasa pemrograman (*Coding*) diubah kedalam bentuk kode bahasa pemrograman, sehingga dapat dimengerti oleh mesin. Dalam tahap implementasi, akan disesuaikan dengan desain sistem yang telah dikerjakan pada tahap perancangan sistem. Kegiatan yang dilakukan selama tahap implementasi antara lain:

- a. Penulisan kode program (*coding*) dengan menggunakan bahasa pemrograman PHP (*Page Hyper Text Pre-Processor*), CSS (*Cascading Style Sheet*) dan *Javascript*.
- b. Manajemen data dengan menggunakan DBMS MySQL.

Sistem ini dibangun dengan menggunakan bahasa pemrograman yang menerapkan konsep *Object Oriented Programming* (OOP) serta dibuat dengan menggunakan aplikasi EditPlus dan XAMPP.

4. Test

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus ditest dahulu apakah sesuai dengan yang diharapkan pada sebelum digunakan sepenuhnya. Pengujian ini dilakukan dengan White Box dan Black. Dalam tahap ini dilakukan dengan menggunakan 2 metode antara lain:

- a. *White Box*

Dilakukan untuk menguji kode pemrograman yang ada pada sistem yang dilakukan pada *statement* dan logika tanpa melibatkan campur tangan *customer*.

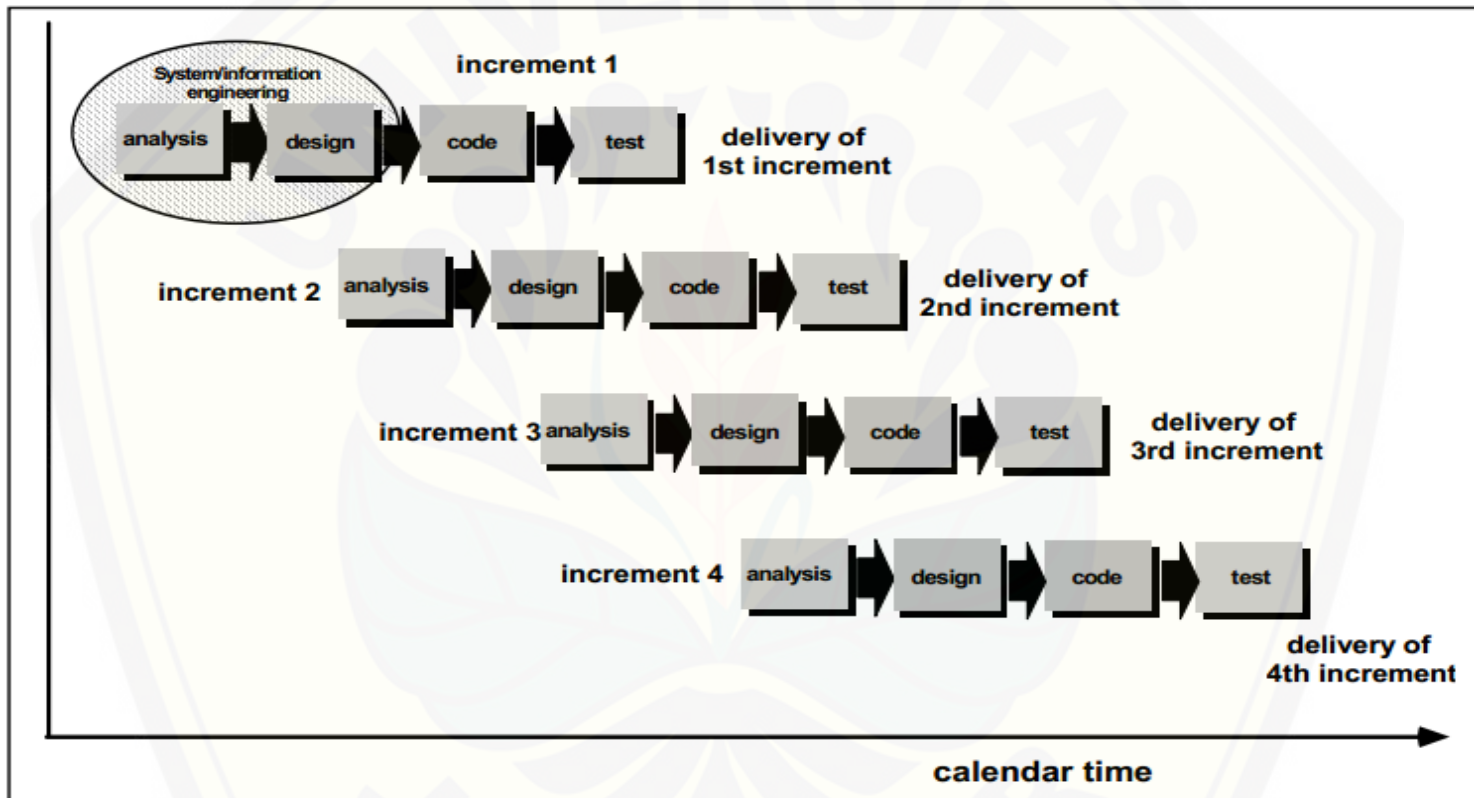
- b. *Black Box*

Dilakukan untuk menyesuaikan apakah sesuai dengan tahap awal yang sudah didefinisikan. Pada pengujian ini memastikan bahwa sistem yang sudah dibuat dapat menjalankan proses dengan benar dan dapat menghasilkan keluaran sesuai dengan perancangan yang sudah dibuat sebelumnya dengan melibatkan campur tangan *customer*.

5. Pemeliharaan Sistem (*Systems Maintenance*)

Setelah sistem berhasil dibangun, maka akan dioperasikan dan berada pada fase perawatan. Jika terdapat *bug* maka akan dilakukan perbaikan. Pemeliharaan sistem dilakukan oleh admin yang ditunjuk untuk menjaga sistem tetap mampu beroperasi secara benar melalui kemampuan sistem dalam mengadaptasikan diri sesuai dengan kebutuhan.

Incremental model pada gambar 2.1. memiliki beberapa tahap dalam *increment* nya, yaitu analisis, desain, *coding* dan test.



Gambar 2.1 *Incremental Model*

(Sumber: Mills)

BAB 3. METODOLOGI PENELITIAN

Metodologi penelitian merupakan bab tentang uraian mengenai tujuan penelitian, tempat dan waktu penelitian hingga metode penelitian yang dilakukan oleh penulis dalam mengerjakan tugas akhir ini. Uraian atau tahap dari metodologi untuk pengembangan sistem ini dapat dibagi menjadi:

3.1. Tujuan Penelitian

Penelitian ini dilakukan dengan bertujuan untuk merancang dan membangun sistem simpan pinjam koperasi “DIAN PATIRANA” berbasis *web*.

3.2. Tempat dan Waktu Penelitian

Penelitian yang telah dilakukan oleh penulis bertempat di Kecamatan Grujungan, Kabupaten Bondowoso, Jawa Timur. Penelitian dilaksanakan pada bulan Oktober 2014 sampai dengan Desember 2014.

3.3. Metode Penelitian

Metode penelitian atau prosedur yang digunakan dalam membangun sistem ini dengan mengimplementasikan teknik pengembangan sistem atau *System Development Life Cycle* (SDLC) dengan menggunakan *Incremental model*. Pengembangan sistem atau *System Development Life Cycle* (SDLC) ini dimulai dari analisis kebutuhan, desain sistem, pengkodean hingga pengujian sistem. Penulis menggunakan *incremental model* karena akan mengurangi beban peneliti dalam pembuatan sistem. *Incremental model* adalah sebuah teknik pengembangan sistem yang digunakan oleh penulis dikarenakan resiko kegagalan yang minimal pada keseluruhan proyek, karena dapat diserahkan beberapa fitur lebih dahulu sebagai *prototype* atau sebagai bahan referensi untuk melengkapi fitur sistem seiring bertambahnya tuntutan kebutuhan user disaat proyek sedang berjalan. Sehingga

ketika ada perubahan atau penambahan fitur, maka tidak akan membebani proyek yang sedang berjalan dan tidak perlu menunggu sampai seluruh sistem selesai agar dapat digunakan *user*.

Tahap awal sebagai metode pengumpulan data untuk mendapatkan informasi dalam membangun modul – modul incremental, diperlukan beberapa kegiatan meliputi studi literatur atau kepustakaan dan penelitian lapangan (*field research*). Berikut ini merupakan penjelasan dari masing-masing metode penelitian yang telah disebutkan sebelumnya. Alir penelitian dalam pembuatan tugas akhir ini dapat dilihat pada gambar 3.1.

1. Studi Literatur atau Kepustakaan

Gambar 3.1 menjelaskan bahwa dimana tahap awal adalah fase studi literatur. Studi literatur dibutuhkan guna untuk mendapatkan informasi dan gambaran terkait mengenai sistem yang akan dibangun. Pada fase ini penulis melakukan beberapa hal seperti pengumpulan data dan informasi dengan cara membaca buku-buku dan referensi dari internet yang dapat dijadikan acuan pembahasan dalam masalah ini.

2. Penelitian Lapangan (*Field Research*)

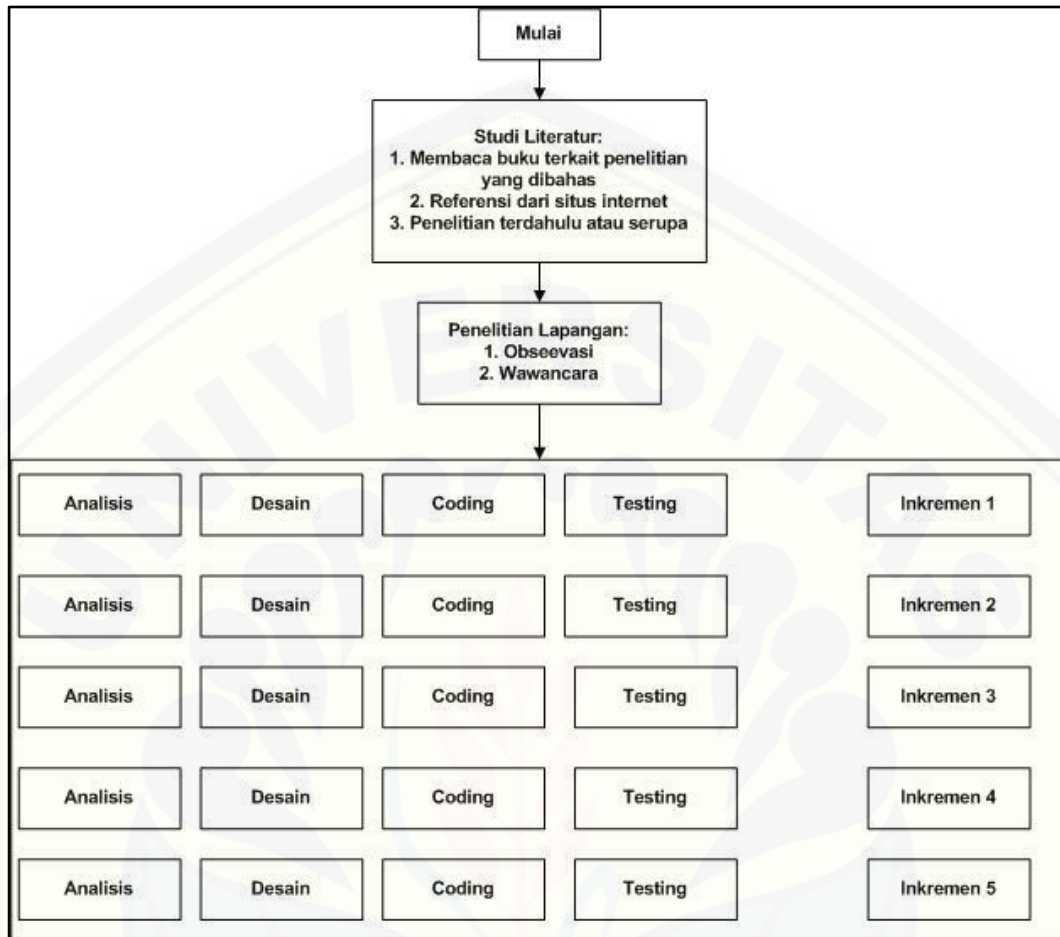
Penelitian yang dilakukan di lapangan untuk memperoleh informasi serta data yang diperlukan. Adapun teknik yang ditempuh.

a. Observasi

Observasi dilakukan oleh penulis dengan pengamatan langsung ke lapangan guna memperoleh data dan informasi terhadap sistem yang sedang berjalan.

b. Wawancara

Wawancara dilakukan setelah penulis menyelesaikan tahap observasi terkait objek yang akan diteliti. Metode wawancara ini merupakan metode tanya jawab untuk mendapatkan data dengan pihak yang berkaitan dengan alur permasalahan.



Gambar 3.1 Alir Penelitian

(Sumber: Hasil Analisis, 2014)

Sehubungan dengan hasil dari penelitian yang telah didapatkan sebelumnya pada metode pengumpulan data, maka didapatkan dokumentasi fitur – fitur yang dibutuhkan oleh pihak koperasi pada aplikasi yang akan dirancang oleh penulis. Penulis membagi beberapa fitur yang dibutuhkan oleh pihak koperasi menjadi 5 inkremen. Pada inkremen pertama, terdiri dari beberapa fitur yang merupakan fitur utama dari sistem yang diprioritaskan untuk diselesaikan terlebih dahulu sebagai *prototype*. Sehingga dapat diserahkan dahulu kepada pengguna tanpa harus

menyelesaikan keseluruhan sistem dan juga dapat menjadi bahan referensi untuk pengembang dalam membangun inkremen - inkremen selanjutnya ketika terjadi penambahan fitur yang mungkin dibutuhkan oleh pengguna.

Pembangunan sistem ini dilakukan secara bertahap yang nantinya akan dilakukan pengujian, verifikasi dan validasi sehingga perubahan yang dilakukan tidak terlalu besar untuk setiap inkremennya. Proses tahapan pada model ini sama dengan *waterfall model*, akan tetapi dilakukan secara berbaris-baris dan setiap baris disebut dengan inkremen. Dimana setiap *increment* menghasilkan fungsi produksi yang berbeda dengan *increment* lainnya. Dalam sistem yang dibangun, peneliti membagi dalam lima modul atau baris *increment* yaitu :

1. Inkremen 1 membahas tentang: User login, registrasi anggota, transaksi simpanan anggota, transaksi pinjaman anggota dan transaksi angsuran anggota.
2. Inkremen 2 membahas tentang: Manajemen lembaga
3. Inkremen 3 membahas tentang: Cetak laporan transaksi
4. Inkremen 4 membahas tentang: Manajemen pengurus koperasi
5. Inkremen 5 membahas tentang: Master data transaksi

Pembangunan sistem ini dilakukan secara bertahap. Adapun tahapan setiap modul inkremen terdiri dari analisis, desain, *coding* dan *testing*.

3.3.1. Analisis Kebutuhan Sistem

Tahap analisis kebutuhan ini merupakan tahap pengumpulan, pencarian dan analisa kebutuhan. Kebutuhan – kebutuhan yang dimaksudkan adalah fitur yang akan dibangun pada sistem serta data – data yang dibutuhkan guna untuk merancang sistem sehingga dapat sesuai keinginan pengguna. Analisis kebutuhan dalam tahap pengembangan sistem pada penelitian ini adalah berupa data yang dibutuhkan dalam membangun sistem ini serta fitur fitur yang nantinya akan dibangun. Analisis

kebutuhan ini akan dilakukan untuk semua inkremen. Analisis kebutuhan dalam membangun sistem ini berdasarkan inkremen adalah :

1. Inkremen 1 yang merupakan modul utama, penulis membangun fitur – fitur: user login, registrasi anggota, simpanan anggota, pinjaman anggota dan angsuran anggota.
2. Inkremen 2, penulis membangun fitur manajemen lembaga.
3. Inkremen 3, penulis membangun fitur cetak laporan transaksi.
4. Inkremen 4, penulis membangun fitur manajemen pengurus koperasi.
5. Inkremen 5, penulis membangun fitur master data transaksi.

Untuk mendapatkan informasi terkait masing – masing inkremen, penulis melakukan wawancara dan meminta data yang dibutuhkan kepada pihak koperasi pada penelitian yang sudah dilakukan sebelumnya. Hasil wawancara dan data yang telah didapatkan, akan digunakan oleh penulis untuk membangun sebuah sistem. Berikut penjelasan dari masing – masing analisis kebutuhan pada tiap inkremen yang telah dilakukan oleh penulis.

1. Analisis kebutuhan inkremen 1

Merupakan tahap pengumpulan, pencarian dan analisa kebutuhan. Kebutuhan – kebutuhan yang dimaksudkan adalah fitur yang akan dibangun pada sistem serta data – data yang dibutuhkan guna untuk merancang sistem sehingga dapat sesuai keinginan pengguna. Pada *increment* tahap 1 ini penulis mengumpulkan kebutuhan *system* terkait apa saja yang dibutuhkan oleh sistem. Fitur yang akan dibangun pada *increment* tahap 1 adalah: user login, registrasi anggota koperasi, transaksi simpanan anggota, transaksi pinjaman anggota dan transaksi angsuran anggota.

2. Analisis kebutuhan inkremen 2

Dari *increment* tahap 1 yaitu: *user login*, registrasi anggota koperasi, transaksi simpanan anggota, transaksi pinjaman anggota dan transaksi angsuran anggota. *User* telah melakukan pengujian dan didapatkan kekurangan sistem. *Increment* tahap 2 tidak akan dilanjutkan jika tahap 1 masih belum benar – benar selesai melakukan perbaikan.

Perbaiki sistem pada hapus data anggota koperasi dengan mengubah basis data *single user login* menjadi basis data *multi user login*, mengubah fitur hapus data anggota menjadi *update* status keanggotaan koperasi, perbaikan transaksi simpanan anggota pada *input form* nominal simpanan sukarela dengan mengizinkan *no value* atau tidak diisikan pada skenario jika tidak melakukan simpanan sukarela, memberikan *exception* tambahan pada input form, pemberian *exception* pada fitur pinjaman anggota dan perbaikan mekanisme angsuran pada fitur angsuran anggota. Sedangkan pada registrasi anggota diberikan penambahan *combo-box* pilihan lembaga yang merupakan fitur manajemen lembaga.

3. Analisis kebutuhan inkremen 3

Dari *increment* tahap 1 dan tahap 2 yaitu: *user login*, registrasi anggota koperasi, transaksi simpanan anggota, transaksi pinjaman anggota, transaksi angsuran anggota dan manajemen lembaga. *User* telah melakukan pengujian dan didapatkan kekurangan sistem. *Increment* tahap 3 tidak akan dilanjutkan jika tahap 2 masih belum benar – benar selesai melakukan perbaikan.

Perbaiki sistem pada penambahan modul ubah data (*edit*) dan hapus data (*delete*) pada fitur manajemen lembaga. Setelah tahap 2 selesai selanjutnya pada *increment* tahap 3 akan diberikan penambahan fitur cetak (*print-out*) laporan transaksi berupa: laporan daftar tagihan, laporan buku kas harian, laporan rekap daftar tagihan, laporan bukti kas masuk dan laporan bukti kas keluar.

4. Analisis kebutuhan inkremen 4

Dari *increment* tahap 1, tahap 2 dan tahap 3 yaitu: *user login*, registrasi anggota koperasi, transaksi simpanan anggota, transaksi pinjaman anggota, transaksi angsuran anggota, manajemen lembaga dan manajemen laporan transaksi. *User* telah melakukan pengujian dan didapatkan kekurangan sistem. *Increment* tahap 4 tidak akan dilanjutkan jika tahap 3 masih belum benar – benar selesai melakukan perbaikan. Perbaikan sistem pada *query* yang digunakan dalam *men-generate* data pada laporan, dimana sebelumnya data tidak ditampilkan pada *print-out* menjadi secara otomatis *ter-generate* ketika memproses laporan mana yang akan dicetak oleh user. Setelah tahap 3 selesai selanjutnya pada *increment* tahap 4 akan diberikan penambahan fitur manajemen pengurus koperasi.

5. Analisis kebutuhan inkremen 5

Dari *increment* tahap 1, tahap 2, tahap 3 dan tahap 4 yaitu: *user login*, registrasi anggota koperasi, transaksi simpanan anggota, transaksi pinjaman anggota, transaksi angsuran anggota, manajemen lembaga, manajemen laporan transaksi dan manajemen pengurus koperasi. *User* telah melakukan pengujian dan didapatkan kekurangan sistem. Pada tahap 5 akan dilakukan perbaikan sistem dan penambahan sistem. Perbaikan sistem pada fitur manajemen pengurus koperasi, dimana terjadi penambahan sub fitur ubah *password user*. Sedangkan pada *increment* tahap 5 akan diberikan penambahan fitur *view master data* untuk data keanggotaan, transaksi simpanan dan transaksi pinjaman.

3.3.2. System Design

Tahap desain ini merupakan tahap untuk membangun sistem berupa dokumen – dokumen rancangan sistem terkait fitur – fitur yang akan dibangun oleh penulis dalam sistem ini. Adapun perbaikan sistem dan penambahan sistem yang sudah dikaji ulang pada tahap analisis, selanjutnya akan dibuat dokumen rancangan sistem pada tahap ini. Dokumen rancangan yang berupa diagram – diagram antara lain:

1. Bisnis Proses
2. *Use Case Diagram*
3. *Use Case Scenario Diagram*
4. *Sequence Diagram*
5. *Class Diagram*
6. *Entity Relationship Diagram (ERD)*

3.3.3. Pengkodean Sistem (*Coding*)

Tahap *coding* ini merupakan tahap untuk merubah desain sistem yang telah dibuat akan dikonversi kedalam bahasa pemrograman. Kegiatan yang dilakukan selama tahap ini adalah :

1. Penulisan kode program dengan menggunakan bahasa PHP (*Page Hyper Text Pre-Processor*), CSS (*Cascading Style Sheet*), *Javascript* serta penggunaan MVC pada pengkodeannya
2. Manajemen basis data dengan menggunakan DBMS MySQL.

3.3.4. *System Testing*

Tahap *testing* ini penulis melakukan kegiatan pengujian sistem dengan *blackbox* dan *whitebox*. Tahap ini memverifikasi apakah sistem yang dibuat telah sesuai dengan apa yang diinginkan atau diharapkan oleh *user* atau tidak. Tahap ini akan diuji kemampuan dan keefektifan sistem sehingga, akan didapatkan kekurangan atau kelemahan sistem yang akan dikaji ulang dan perbaikan pada inkremen selanjutnya.

White box testing merupakan cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan output yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan dicek satu persatu dan diperbaiki, kemudian di-*compile* ulang. Pengembang akan melakukan pengukuran dengan menghitung *Cyclomatic Complexity* (CC).

Black box testing merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Dapat dianalogikan seperti melihat sesuatu dalam kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik atau di dalam kotak hitam tersebut. Sama halnya pengujian *black box*, mengevaluasi perangkat lunak hanya dilakukan dari luarnya saja (*interface*) pada *input* dan *output*. Tanpa mengetahui apa sesungguhnya yang terjadi dalam prosesnya secara detail.

BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM

Bab ini membahas tentang analisis dan perancangan sistem informasi simpan pinjam koperasi berbasis web (studi kasus pada koperasi KP-RI “DIAN PATIRANA”).

4.1. Pengumpulan Data

Pengumpulan data dilakukan setelah penulis sebagai pihak *developer* mendapatkan informasi *user requirements*. Data yang sudah didapatkan sebagian besar berasal dari pihak perusahaan dengan melakukan wawancara dengan pihak perusahaan terkait penelitian yang saat ini sedang dilakukan. Data – data yang telah didapatkan di antaranya:

4.1.1. Data Transaksi Daftar Tagihan Anggota

Daftar tagihan anggota merupakan daftar yang menampilkan tagihan-tagihan kepada pihak lembaga terhadap transaksi yang dilakukan oleh anggota koperasi yang dikelompokkan sesuai dengan lembaga anggota tersebut bekerja dan ditampilkan dalam 1 (satu) bulan. Data yang didapat oleh penulis akan diterangkan pada gambar 4.1.

Gambar 4.1 menunjukkan bahwa dalam 1 (satu) bulan pada sebuah lembaga, sebagian anggota koperasi masih memiliki tanggungan angsuran. Walaupun semua membayarkan simpanan wajib, akan tetapi tidak semua membayar simpanan sukarela.

KPRI "DIAN PATIRANA" Kecamatan Grjugan Badan Hukum No. : 6045/BH/II/86 Tanggal : 05 Oktober 1996			DAFTAR TAGIHAN USP KPRI "DIAN PATIRANA" Untuk SD : Grjugan Kidul 01 No. : 01 Bagian Bulan : JUNI 2013									
No	Nama Anggota	Ke	JENIS TAGIHAN									JUMLAH
			Simpanan			Unit Simpan Pinjam			SIM/almunium/juicer			
			Pokok	S. Rela	Wajib	Ke	Pokok	Jasa	Ke	Pokok	Jasa	
1	Drs. Suhandoko		200.000	100.000					9*	125.000	18.750	443.750
2	Angsri Suryaningsih			100.000	✓14/30	500.000	225.000					882.500
3	Eny Sumartini			100.000	✓2/30	500.000	225.000					825.000
4	Gendut Prasetyo			100.000	✓3/20	1.250.000	375.000	9#	127.600	19.150		1.871.750
5	Purwantini			100.000								142.550
6	Kustiana			100.000	✓4/20	300.000	90.000					490.000
7	Drs. Sugianto			100.000	✓7/30	433.400	195.000					728.400
8	Yuliono			100.000	✓8/50	300.000	225.000					625.000
9	Sri Hardaningsih			100.000	✓19/30	333.400	150.000					583.400
10	Subagio,S.Pd			100.000	✓7/40	500.000	300.000					913.800
11	Drs. Sugianto				✓3/10	300.000	45.000	9	50.000	7.500		402.500
12	Yuliono				✓4/10	150.000	22.500					172.500
									#)	127.600	19.150	
									*)	125.000	18.750	
										50.000	7.500	
JUMLAH			200.000	1.000.000		4.566.800	1.852.500					8.081.150

Gambar 4.1 Data Transaksi Daftar Tagihan Anggota

(Sumber: Hasil Analisis, 2015)

4.1.2. Data Transaksi Buku Kas Harian

Buku kas harian merupakan buku yang mencatat jumlah kas masuk dan keluar pada seluruh lembaga dalam 1 (satu) bulan. Data yang didapatkan oleh penulis akan diterangkan pada gambar 4.2. Pada gambar 4.2 dapat disimpulkan bahwa, setiap lembaga memiliki tagihan yang berbeda-beda tergantung dari transaksi yang dilakukan oleh masing-masing anggota koperasi. Dapat dilihat pada simpanan sukarela dan jenis barang atau jasa setiap lembaga berbeda – beda, akan tetapi pada simpanan wajib tetap akan selalu ada pada setiap laporan karena ketentuan transaksinya diwajibkan oleh seluruh anggota koperasi.

4.1.3. Data Transaksi Rekap Daftar Tagihan Anggota

Rekap daftar tagihan anggota merupakan rekapan data dari daftar tagihan transaksi yang dilakukan oleh anggota koperasi berdasarkan dalam 1(satu) bulan dan dikelompokkan pada setiap lembaga. Data yang telah didapatkan oleh penulis akan diterangkan pada gambar 4.3.

**BUKU KAS HARIAN
BULAN JULI 2013**

TANGGAL	URAIAN	DEBIT	KREDIT	SALDO
01/07/2013	01 Saldo kas bulan Juni 2013	Rp 95.623.254		Rp 95.623.254
01/07/2013	02 Terima dari Bendahara SDN Grujugan Kidul 1			
	1 Simpanan suka rela	Rp ✓ 200.000		Rp 95.823.254
	2 Simpanan wajib	Rp ✓ 1.000.000		Rp 96.823.254
	3 USP	Rp ✓ 4.566.800		Rp 101.390.054
	4 Jasa USP	Rp ✓ 1.852.500		Rp 103.242.554
	5 SIM	Rp ✓ 57.500		Rp 103.300.054
	6 Konfeksi	Rp ✓ 113.850		Rp 103.413.904
	7 Almunium	Rp ✓ 143.750		Rp 103.557.654
	8 Juicer	Rp ✓ 146.750		Rp 103.704.404
01/07/2013	03 Terima dari Bendahara SDN Grujugan Kidul 2			
	1 Simpanan wajib	Rp ✓ 1.000.000		Rp 104.704.404
	2 USP	Rp ✓ 3.790.000		Rp 108.494.404
	3 Jasa USP	Rp ✓ 1.912.500		Rp 110.406.904
	4 Taper Weare	Rp ✓ 189.660		Rp 110.596.564
	5 Juicer	Rp ✓ 220.150		Rp 110.816.714
01/07/2013	04 Terima dari Bendahara SDN Grujugan Kidul 3			
	1 Simpanan wajib	Rp ✓ 1.000.000		Rp 111.816.714
	2 USP	Rp ✓ 6.225.000		Rp 118.041.714
	3 Jasa USP	Rp ✓ 2.430.000		Rp 120.471.714
	4 Konfeksi	Rp ✓ 36.250		Rp 120.507.964
	5 Taper Weare	Rp ✓ 54.650		Rp 120.562.614
	6 Juicer	Rp ✓ 73.400		Rp 120.636.014

Gambar 4.2 Data Transaksi Buku Kas Harian

(Sumber: Hasil Analisis, 2015)

REKAP DAFTAR TAGIHAN USP DAN BARANG KP-RI DIAN PATIRANA KEC. GRUJUGAN BAGIAN BULAN : 2 - 2015

No	Nama Lembaga	Jenis Tagihan											Jumlah				
		S.Pokok	S.Sukarela	S.Wajib	USP	Jasa USP	SIM	Konfeksi	Taperware	Almunium	Sepatu	Tas		Kasur	Juicer	Karpet	Eur
1	SDN Grujugan 1	10000	1500	10000	1000000	15000											1036500
2	SDN Grujugan 2	10000	21000	35000	5000000	750000											50816000
3	SDN Grujugan 3	10000	20000	20000	750000	11250											811250

Gambar 4.3 Data Transaksi Rekap Daftar Tagihan

(Sumber: Hasil Analisis, 2015)

Gambarl 4.3. menunjukkan bahwa setiap lembaga jumlah nominal setiap atribut baik simpanan, pinjaman, maupun barang berbeda – beda tergantung pada berapa dan apa saja transaksi yang dilakukan oleh anggota koperasi tersebut.

4.1.4. Data Transaksi Bukti Kas Masuk

Bukti kas masuk merupakan sebuah kwitansi yang berisi tentang daftar transaksi masukan pada koperasi terhadap transaksi – transaksi yang telah dilakukan oleh anggota koperasi. Data yang telah didapatkan oleh penulis akan diterangkan pada gambar 4.4.

KOPERASI PEGAWAI REPUBLIK INDONESIA "DIAN PATIRANA" "KEC. GRUJUGAN" BH. NO. 6045/BH/II/86 Tgl. 5 September 1996		<u>BUKTI KAS MASUK (BKM)</u>		No. Urut : No. BKM : Tanggal :	KM 01-2-2015
Telah diterima dari :	Bendahara Gaji SDN Grujugan 2				
Banyaknya uang :	SDN Grujugan 2				
Untuk pembayaran :	1. Simpanan Wajib	Rp.	70000		
	2. Simpanan Pokok	Rp.	20000		
	3. Simpanan Sukarela	Rp.	42000		
	4. Unit Simpan Pinjam	Rp.	10000000		
	5. Jasa USP	Rp.	150000		
Terbilang Rp. xxxx					
Diterima oleh :			Disetor oleh :		
MUDJINO HS,S.Pd			xxxxx		

Gambar 4.4 Data Transaksi Bukti Kas Masuk

(Sumber: Hasil Analisis, 2015)

Gambar 4.4. menunjukkan bahwa bukti kas masuk hanya mencatat transaksi debit saja. Sehingga hanya berisi seperti: Simpanan Wajib, Simpanan Pokok, Simpanan Sukarela, Unit Simpan Pinjam dan Jasa Unit Simpan Pinjam.

4.1.5. Data Transaksi Bukti Kas Keluar

Bukti kas keluar merupakan merupakan sebuah kwitansi yang berisi tentang daftar transaksi keluaran pada koperasi terhadap transaksi – transaksi yang telah dilakukan oleh anggota koperasi. Buku kas keluar inilah kwitansi yang selanjutnya akan diberikan kepada pihak lembaga sebagai tagihan – tagihan terhadap anggota koperasi yang nantinya secara langsung dipotong gaji. Data yang telah didapatkan oleh penulis akan diterangkan pada gambar 4.5.

Berdasarkan gambar 4.5. dapat disimpulkan bahwa buku kas keluar hanya mencatat transaksi kredit saja. Sehingga hanya berisi seperti: Unit Simpan Pinjam (kredit), Elektronik, Konfeksi, Taper ware, Almunium dan Karpet.

KOPERASI PEGAWAI REPUBLIK INDONESIA "DIAN PATIRANA" "KEC. GRUJUGAN" BH. NO. 6045/BH/II/86 Tgl. 5 September 1996	<u>BUKTI KAS KELUAR (BKK)</u>	No. Urut : No. BKK : KM Tanggal : 01-2-2015																								
Telah diterima dari : Bendahara Gaji SDN Grujugan 2 Banyaknya uang : SDN Grujugan 2																										
Untuk pembayaran : <table style="width: 100%; margin-left: 20px;"> <tr> <td style="width: 5%;">1.</td> <td style="width: 45%;">Unit Simpan Pinjam</td> <td style="width: 10%;">Rp.</td> <td style="width: 40%; text-align: right;">10000000</td> </tr> <tr> <td>2.</td> <td>Elektronik</td> <td>Rp.</td> <td style="text-align: right;">0</td> </tr> <tr> <td>3.</td> <td>Konfeksi</td> <td>Rp.</td> <td style="text-align: right;">0</td> </tr> <tr> <td>4.</td> <td>Taper Ware</td> <td>Rp.</td> <td style="text-align: right;">0</td> </tr> <tr> <td>5.</td> <td>Almunium</td> <td>Rp.</td> <td style="text-align: right;">0</td> </tr> <tr> <td>5.</td> <td>Karpet</td> <td>Rp.</td> <td style="text-align: right;">0</td> </tr> </table>			1.	Unit Simpan Pinjam	Rp.	10000000	2.	Elektronik	Rp.	0	3.	Konfeksi	Rp.	0	4.	Taper Ware	Rp.	0	5.	Almunium	Rp.	0	5.	Karpet	Rp.	0
1.	Unit Simpan Pinjam	Rp.	10000000																							
2.	Elektronik	Rp.	0																							
3.	Konfeksi	Rp.	0																							
4.	Taper Ware	Rp.	0																							
5.	Almunium	Rp.	0																							
5.	Karpet	Rp.	0																							
Terbilang Rp. xxxx																										
Diterima oleh : MUDJINO HS,S.Pd		Disetor oleh : xxxxx																								

Gambar 4.5 Data Transaksi Bukti Kas Keluar

(Sumber: Hasil Analisis, 2015)

4.2. Kebutuhan Fungsional dan Non-Fungsional

Jenis kebutuhan yang berisi tentang proses apa saja yang dilakukan oleh sistem serta berbagai informasi yang dihasilkan oleh sistem. Berikut merupakan jenis

kebutuhan sistem yang nantinya dapat dilakukan oleh sistem yang telah dibangun oleh pengembang.

4.2.1. Kebutuhan Fungsional

Kebutuhan fungsional adalah kemampuan primer yang nantinya harus dapat dilakukan oleh sistem dan harus terpenuhi oleh sistem. Kebutuhan yang berkaitan dengan fungsi sistem pada kebutuhan fungsional atau *Software Requirement Specification Functional* (SRSF), untuk dapat menghasilkan keluaran yang diinginkan oleh pengembang dan sesuai dengan harapan pengguna akan dijelaskan pada tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional

SRSF_ID	Identifikasi
SRSF 01	Dapat memproses transaksi simpanan anggota, pinjaman anggota, angsuran anggota, serta produk jasa lainnya yang disediakan oleh koperasi.
SRSF 02	Dapat memproses data transaksi yang dilakukan oleh anggota koperasi menjadi laporan transaksi yang diperlukan.
SRSF 03	Dapat mengarsipkan data transaksi agar lebih aman karena disimpan dalam bentuk digital.

(Sumber: Hasl Analisis, 2015)

4.2.2. Kebutuhan Non-fungsional

Kebutuhan non-fungsional adalah kemampuan sekunder yang dimiliki oleh sistem dan secara tidak langsung berkaitan dengan sistem yang dibangun. Kebutuhan yang berkaitan dengan fungsi sistem pada kebutuhan nonfungsional atau *Software Requirement Specification Non Functional* (SRSNF), akan dijelaskan pada tabel 4.2.

Tabel 4.2 Kebutuhan Nonfungsional

SRSNF_ID	Identifikasi
SRSNF 01	<i>Availability</i> . Sistem harus dapat diakses dalam 24 jam dan dimana saja.
SRSNF 02	<i>Response Time</i> . Sistem harus dapat merespon dan memproses paling lama dalam 10 detik.
SRSNF 03	<i>Security</i> . Sistem harus memiliki fitur hak akses pada tingkatan user dalam sistem
SRSNF 04	<i>User Friendly</i> . Sistem harus didesain sedemikian rupa agar dapat memudahkan user untuk mengoperasikannya.
SRSNF 05	<i>Realibility</i> . Sistem harus dapat berjalan sesuai dengan kebutuhan user.

(Sumber: Hasil Analisis, 2015)

4.3. Business Process

Business process didefinisikan sebagai kumpulan dari proses yang berisi aktifitas yang saling berelasi sehingga menghasilkan sebuah keluaran sesuai tujuan atau sasaran. Pada gambar 4.6. *business process* sistem informasi simpan pinjam koperasi memiliki input berupa data transaksi simpanan anggota, data transaksi pinjaman anggota dan data transaksi angsuran anggota. Data – data tersebut dibutuhkan untuk memproses transaksi simpan pinjam koperasi secara komputasi pada sistem yang sedang dibuat oleh penulis.

Sistem informasi simpan – pinjam koperasi dibuat berdasarkan buku anggaran dasar koperasi ini dirancang dengan tampilan *website* dengan bahasa pemrograman PHP dan manajemen basis data MySQL. Sistem ini dibuat dengan tujuan untuk

mempermudah kegiatan simpan-pinjam yang ada dengan cara komputasi sehingga transaksi – transaksi yang ada, dapat dikerjakan lebih cepat dan efisien karena sudah tidak lagi dikerjakan dengan cara manual. Sehingga *output* yang dihasilkan dari sistem ini berupa laporan transaksi simpan pinjam.

Business process sistem simpan pinjam koperasi dapat dilihat pada gambar 4.6.

4.4. Use Case Diagram

Terdapat *use case diagram* yang pada sistem, guna untuk menggambarkan fitur – fitur yang akan dibuat pada sistem. Penggambaran *use case diagram* adalah ditekankan pada apa yang dilakukan oleh sistem, bukan bagaimana yang dilakukan oleh sistem. Sebuah *use case* merpresentasikan interaksi antara *actor* dan sistem, sebuah pekerjaan tertentu yang dilakukan oleh *actor*. *Use case diagram* tersebut akan dijelaskan pada gambar 4.7.

Dari *use case diagram* yang telah dijelaskan pada gambar 4.7. terdapat tiga aktor yang memiliki hak akses masing – masing dalam sistem. Definisi dari aktor – aktor yang terdapat pada sistem, dapat dilihat pada tabel 4.3.

Tabel 4.3 Deskripsi Aktor

No.	Aktor	Definisi Tugas
1.	Pengurus	Melakukan proses transaksi simpan pinjam, memproses arsip transaksi anggota dan mencetak laporan transaksi.
2.	Administrator	Melakukan manajemen user pada sistem, melakukan <i>backup</i> dan <i>recovery</i> basis data dan <i>maintenance</i> aplikasi koperasi.
3.	Pimpinan	Dapat melihat keseluruhan data transaksi simpan pinjam dan data keanggotaan koperasi.

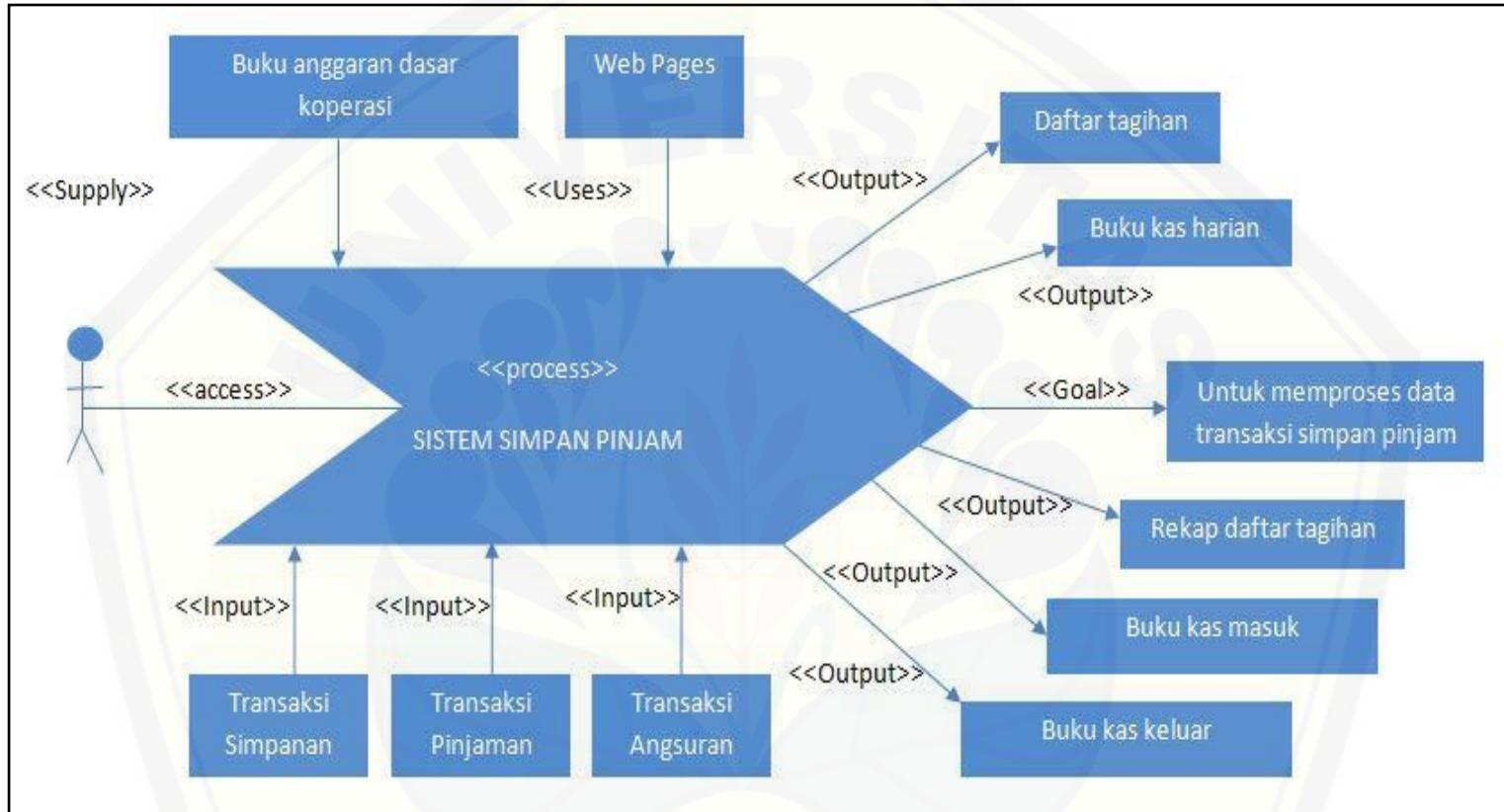
(Sumber: Hasil Analisis, 2015)

Pada *use case* terdapat fitur – fitur yang ada pada sistem yang akan diakses oleh aktor sesuai dengan hak akses masing –masing. Berikut adalah definisi dari masing –masing *use case* dapat dilihat pada tabel 4.4.

Tabel 4.4 Deskripsi *Use Case*

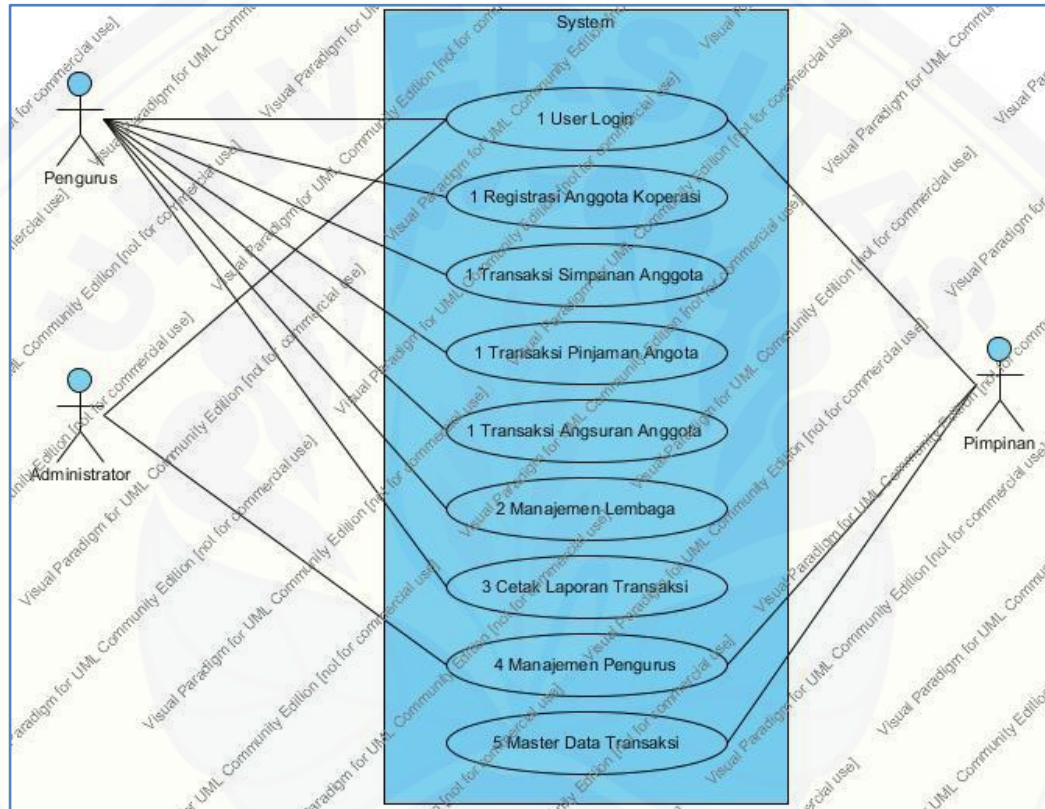
No.	Use Case Diagram	Definisi Use Case
1.	User Login	Use Case untuk user login yang dilakukan oleh semua aktor
2.	Registrasi Anggota Koperasi	Use Case untuk registrasi anggota koperasi yang dilakukan oleh pengurus
3.	Transaksi Simpanan Anggota	Use Case untuk transaksi simpanan anggota koperasi
4.	Transaksi Pinjaman Anggota	Use Case untuk transaksi pinjaman anggota koperasi
5.	Transaksi Angsuran Anggota	Use Case untuk transaksi angsuran pinjaman anggota
6.	Manajemen Lembaga	Use Case untuk fitur manajemen data lembaga yang menjadi lingkup dari koperasi.
7.	Cetak Laporan Transaksi	Use Case untuk sistem keluaran laporan transaksi koperasi
8.	Manajemen Pengurus	Use Case untuk fitur manajemen pengurus yang ada di dalam system
9.	Master Data Transaksi	Use Case untuk fitur detail data transaksi pada transaksi – transaksi anggota dan data anggota.

(Sumber: Hasil Analisis, 2015)



Gambar 4.6 *Business Process*

(Sumber: Hasil Analisis, 2015)



Gambar 4.7 Use Case Diagram

(Sumber: Hasil Analisis, 2015)

4.5. Use Case Scenario

Use case scenario pada subbab ini menjelaskan tentang alur cerita (skenario) pada setiap *use case* yang sudah dijelaskan pada *use case diagram*. *Use case scenario* berisi nama *use case*, aktor, *pre-condition*, *post-condition*, skenario utama dan skenario alternatif.

4.5.1. Perancangan Inkremen 1 *Scenario* User Login

Skenario yang menjelaskan tentang fitur *login* hingga *logout*, sebuah fitur utama yang menjadi kunci pembagian hak akses sesuai tingkatan akun *user* yang sign-in pada sistem.

1. User Login (*Log-In*)

Menjelaskan tentang skenario user login (*log-in*). *Use case* ini menjelaskan alur untuk masuk ke dalam sistem. *Use case scenario* user login (*log-in*) dapat dilihat pada tabel 4.5.

Tabel 4.5 *Use Case Scenario* User Login (*Log-In*)

Nama <i>Use Case</i>	User Login (<i>Log-In</i>)
Aktor	<i>User</i>
<i>Pre-Conditions</i>	<i>User</i> mengakses aplikasi koperasi
<i>Post-Conditions</i>	<i>User</i> berhasil <i>login</i> dan sistem menampilkan <i>home</i>
Skenario Utama	
User	Reaksi Sistem
1. Membuka aplikasi	
	2. Menampilkan halaman <i>form login</i>
3. <i>User</i> memasukkan <i>username</i> dan <i>password</i>	
	4. Menampilkan dialog konfirmasi

	“Login sukses”
	5. <i>Redirect</i> ke halaman <i>home</i>
Skenario Alternatif	
Skenario alternatif jika <i>username</i> dan <i>password</i> salah	
	4a. Menampilkan <i>exception</i> “Gagal Login! <i>password</i> belum diisi, atau kombinasi salah!!”
	5a. Menampilkan halaman <i>form login</i>
9a. <i>User</i> kembali mengisi data pada <i>form</i>	
Skenario alternatif jika <i>username</i> belum diisi	
	4b. Menampilkan <i>exception</i> “Gagal Login! <i>username</i> belum diisi!”
5b. <i>User</i> kembali mengisi data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

2. User Login (*Log-Out*)

Menjelaskan tentang skenario user login (*log-out*). Use case ini menjelaskan alur untuk keluar dari sistem. *Use case scenario* user login (*log-out*) dapat dilihat pada tabel 4.6.

Tabel 4.6 *Use Case Scenario* User Login (*Log-Out*)

Nama <i>Use Case</i>	User Login (<i>Log-out</i>)
Aktor	<i>User</i>
<i>Pre-Conditions</i>	<i>User</i> mengakses aplikasi koperasi
<i>Post-Conditions</i>	<i>User</i> berhasil <i>logout</i> dan sistem menampilkan <i>login form</i>
Skenario Utama	
User	Reaksi Sistem

1. Membuka aplikasi	
	2. Menampilkan halaman website
3. Menekan tombol <i>logout</i> pada menu navigasi	
	4. Menampilkan dialog konfirmasi “Logout”
	5. <i>Redirect</i> ke halaman <i>login form</i>
Skenario Alternatif	
Skenario alternatif jika <i>user</i> tidak mengkonfirmasi dialog konfirmasi logout	
	5. Sistem tidak mengeksekusi perintah apapun halaman akan tetap

(Sumber: Hasil Analisis, 2015)

4.5.2. Perancangan Inkremen 1 *Scenario* Registrasi Anggota

Skenario yang menjelaskan tentang fitur entri data registrasi anggota, *edit* data registrasi anggota dan *delete* data registrasi anggota. Sebuah fitur yang merupakan fitur untuk manajemen data anggota koperasi baik itu data anggota baru maupun data anggota lama.

1. Registrasi Anggota (*Entry*)

Menjelaskan tentang skenario registrasi anggota (*entry*). Use case ini menjelaskan alur untuk entri data registrasi anggota baru kedalam sistem. *Use case scenario* registrasi anggota (*entry*) dapat dilihat pada tabel 4.7.

Tabel 4.7 *Use Case Scenario* Registrasi Anggota (*Entry*)

Nama <i>Use Case</i>	Registrasi Anggota Koperasi (entri data)
Aktor	Pengurus
<i>Pre-Conditions</i>	Pengurus memilih fitur anggota koperasi
<i>Post-Conditions</i>	Pengurus telah selesai meregistrasi anggota baru pada

	<i>database</i>
Skenario Utama	
Pengurus	Reaksi Sistem
1. Pengurus mengakses fitur anggota koperasi pada <i>menu</i> petugas	
	2. Menampilkan halaman daftar anggota koperasi
3. Pengurus menekan tombol 'registrasi' pada halaman daftar anggota dan klik 'OK'	
	4. Menampilkan <i>form</i> registrasi anggota koperasi
5. Pengurus mengisi data sesuai <i>form</i> pengajuan anggota yang sudah diserahkan	
6. Pengurus menekan tombol 'Registrasi Sekarang' dan klik 'OK'	
	7. Sistem meng- <i>entry</i> data pada <i>database</i>
	8. <i>Redirect</i> kembali pada halaman daftar anggota koperasi
Skenario Alternatif	
Skenario alternatif jika nama anggota belum diisikan	
	7a. Menampilkan <i>exception</i> "Nama anggota belum diisikan! mohon entri ulang!"
	8a. Menampilkan halaman <i>form</i>

	registrasi anggota koperasi
9a. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika alamat anggota belum diisikan	
	7b. Menampilkan <i>exception</i> “Alamat anggota belum diisi! mohon entri ulang!”
	8b. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9b. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon belum diisikan	
	7c. Menampilkan <i>exception</i> “Nomor telepon belum belum diisi! mohon entri ulang!”
	8c. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9c. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon bernilai desimal, huruf dan simbol	
	7d. Menampilkan <i>exception</i> “Nomor telepon tidak diijinkan desimal, huruf, atau simbol! mohon entri ulang!”
	8d. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9d. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon bernilai minus	

	7e. Menampilkan <i>exception</i> “Nomor telepon tidak boleh bernilai minus! mohon entri ulang!”
	8e. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9e. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor ktp belum diisikan	
	7f. Menampilkan <i>exception</i> “Nomor ktp belum belum diisi! mohon entri ulang!”
	8f. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9f. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor ktp bernilai desimal, huruf dan simbol	
	7g. Menampilkan <i>exception</i> “Nomor ktp tidak diijinkan desimal, huruf, atau simbol! mohon entri ulang!”
	8g. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9g. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor ktp bernilai minus	
	7h. Menampilkan <i>exception</i> “Nomor ktp tidak boleh bernilai minus! mohon entri ulang!”
	8h. Menampilkan halaman <i>form</i> registrasi anggota koperasi

9h. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika tanggal lahir belum diisikan	
	7i. Menampilkan <i>exception</i> “Tanggal lahir anggota belum diisi! mohon entri ulang!”
	8i. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9i. Pengurus kembali mengentrikan data pada <i>form</i>	
telp jika tempat lahir belum diisikan	
	7j. Menampilkan <i>exception</i> “Tempat lahir anggota belum diisi! mohon entri ulang!”
	8j. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9j. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika gender belum diisikan	
	7i. Menampilkan <i>exception</i> “Gender anggota belum dipilih! mohon entri ulang!”
	8i. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9i. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika alamat <i>e-mail</i> belum diisikan	
	7j. Menampilkan <i>exception</i> “Email

	belum belum diisi! mohon entri ulang!”
	8j. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9j. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika alamat <i>informasi</i> lembaga belum diisi	
	7k. Menampilkan <i>exception</i> “Informasi lembaga belum diisi! mohon entri ulang!”
	8k. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9k. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

2. Registrasi Anggota (*Update Status*)

Menjelaskan tentang skenario registrasi anggota (*update status*). Use case ini menjelaskan alur untuk ubah status keanggotaan pada sistem. *Use case scenario* registrasi anggota (*update-status*) dapat dilihat pada tabel 4.8

Tabel 4.8 *Use Case Scenario* Registrasi Anggota (*Update-Status*)

Nama <i>Use Case</i>	Registrasi Anggota Koperasi (<i>update status</i>)
Aktor	Pengurus
<i>Pre-Conditions</i>	Pengurus memilih fitur anggota koperasi
<i>Post-Conditions</i>	Pengurus telah selesai meng- <i>update</i> status keanggotaan pada <i>database</i>
Skenario Utama	
Pengurus	Reaksi Sistem

1. Pengurus mengakses fitur anggota koperasi pada <i>menu</i> petugas	
	2. Menampilkan halaman daftar anggota
3. Pengurus menekan tombol ‘ubah status’ dan klik ‘OK’	
	4. Menampilkan <i>form update</i> status keanggotaan koperasi
5. Pengurus mengisi <i>form</i> sesuai data yang sesuai	
6. Pengurus menekan tombol ‘Update Sekarang’	
	7. Sistem meng- <i>update</i> data pada <i>database</i>
	8. <i>Redirect</i> kembali pada halaman daftar anggota koperasi
Skenario Alternatif	
Skenario alternatif jika anggota yang akan dinonaktifkan, ternyata masih memiliki tanggungan koperasi	
	7a. Menampilkan <i>exception</i> “Gagal menonaktifkan! Dikarenakan masih memiliki tanggungan pinjaman koperasi! mohon entri ulang!”
	8a. Menampilkan halaman <i>form update</i> status keanggotaan koperasi
9a. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum menselek status menjadi nonaktif	

	7b. Menampilkan <i>exception</i> “Status keanggotaan belum diubah! mohon entri ulang!”
	8b. Menampilkan halaman <i>form update</i> status keanggotaan koperasi
9b. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika tanggal nonaktif belum diinputkan	
	7c. Menampilkan <i>exception</i> “Tanggal nonaktif belum diinputkan! mohon entri ulang!”
	8c. Menampilkan halaman <i>form update</i> status keanggotaan koperasi
9c. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

3. Registrasi Anggota (*Edit-Update*)

Menjelaskan tentang skenario registrasi anggota (*edit-update*). Use case ini menjelaskan alur untuk ubah data anggota pada sistem. *Use case scenario* registrasi anggota (*edit-update*) dapat dilihat pada tabel 4.9

Tabel 4.9 *Use Case Scenario* Registrasi Anggota (*Edit-Update*)

Nama <i>Use Case</i>	Registrasi anggota koperasi (<i>edit-update</i>)
Aktor	Pengurus
Pre-Conditions	Pengurus memilih fitur anggota koperasi
Post-Conditions	Pengurus telah selesai meng- <i>update</i> data keanggotaan pada <i>database</i>

Skenario Utama	
Pengurus	Reaksi Sistem
1. Pengurus mengakses fitur anggota koperasi pada <i>menu</i> petugas	
	2. Menampilkan halaman daftar anggota koperasi
3. Pengurus menekan tombol ' <i>update</i> ' dan klik 'OK'	
	4. Menampilkan <i>form update</i> data keanggotaan koperasi
5. Pengurus mengisi <i>form</i> sesuai data yang sesuai	
6. Pengurus menekan tombol ' <i>Update Sekarang</i> '	
	7. Sistem meng- <i>update</i> data pada <i>database</i>
	8. <i>Redirect</i> kembali pada halaman daftar anggota
Skenario Alternatif	
Skenario alternatif jika belum mengentrikan alamat terbaru	
	7a. Menampilkan <i>exception</i> "Alamat belum diisikan! mohon entri ulang!"
	8a. Menampilkan halaman <i>form update</i> data keanggotaan koperasi
9a. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum menselek status menjadi aktif	
	7b. Menampilkan <i>exception</i> "Status

	keanggotaan belum diubah! mohon entri ulang!”
	8b. Menampilkan halaman <i>form update</i> data keanggotaan koperasi
9b. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon belum diisikan	
	7c. Menampilkan <i>exception</i> “Nomor telepon belum belum diisi! mohon entri ulang!”
	8c. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9c. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon bernilai desimal, huruf dan simbol	
	7d. Menampilkan <i>exception</i> “Nomor telepon tidak diijinkan desimal, huruf, atau simbol! mohon entri ulang!”
	8d. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9d. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon bernilai minus	
	7e. Menampilkan <i>exception</i> “Nomor telepon tidak boleh bernilai minus! mohon entri ulang!”
	8e. Menampilkan halaman <i>form</i> registrasi anggota koperasi

9e. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika alamat <i>e-mail</i> belum diisikan	
	7f. Menampilkan <i>exception</i> “Email belum belum diisi! mohon entri ulang!”
	8f. Menampilkan halaman <i>form</i> registrasi anggota koperasi
9f. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

4.5.3. Perancangan Inkremen 1 *Scenario* Transaksi Simpanan

Skenario yang menjelaskan tentang fitur entri data transaksi simpanan anggota, dan *edit* data transaksi simpanan. Sebuah fitur yang merupakan fitur untuk melakukan transaksi simpanan anggota yang dilakukan oleh pengurus koperasi.

1. Transaksi Simpanan Anggota (*Entry*)

Menjelaskan tentang skenario transaksi simpanan anggota (*entry*). Use case ini menjelaskan alur untuk entri data transaksi simpanan anggota baru kedalam sistem. *Use case scenario* transaksi simpanan anggota (*entry*) dapat dilihat pada tabel 4.10.

Tabel 4.10 *Use Case Scenario* Transaksi Simpanan Anggota (*Entry*)

Nama <i>Use Case</i>	Simpanan anggota koperasi (<i>entry</i>)
Aktor	Pengurus
Pre-Conditions	Pengurus memilih fitur simpanan anggota
Post-Conditions	Pengurus telah selesai mengentri transaksi simpanan anggota pada <i>database</i>
Skenario Utama	
Pengurus	Reaksi Sistem

1. Pengurus mengakses fitur simpanan anggota pada <i>menu</i> transaksi	
	2. Menampilkan halaman transaksi simpanan anggota koperasi
3. Pengurus menekan tombol ' <i>new data</i> ' dan klik 'OK'	
	4. Menampilkan <i>form</i> entri simpanan anggota koperasi
5. Pengurus mengisi <i>form</i> sesuai data yang sesuai	
6. Pengurus menekan tombol 'Entri Sekarang'	
	7. Sistem meng- <i>entry</i> data pada <i>database</i>
	8. <i>Redirect</i> kembali pada halaman transaksi simpanan anggota koperasi
Skenario Alternatif	
Skenario alternatif jika belum memilih anggota yang melakukan transaksi	
	7a. Menampilkan <i>exception</i> "Nama anggota belum dipilih! mohon entri ulang!"
	8a. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9a. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisikan simpanan wajib anggota	
	7b. Menampilkan <i>exception</i> "Nominal

	simpanan wajib belum diisi! mohon entri ulang!”
	8b. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9b. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika simpanan wajib anggota bernilai desimal, huruf dan simbol	
	7c. Menampilkan <i>exception</i> “Nominal simpanan wajib tidak diijinkan desimal, huruf, atau simbol! Mohon entri ulang!”
	8c. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9c. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika simpanan wajib anggota bernilai minus	
	7d. Menampilkan <i>exception</i> “Nominal simpanan wajib tidak boleh bernilai minus! Mohon entri ulang!”
	8d. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9d. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika simpanan sukarela bernilai desimal, huruf dan simbol	
	7e. Menampilkan <i>exception</i> “Nominal simpanan sukarela tidak diijinkan desimal, huruf, atau simbol! Mohon

	entri ulang!”
	8e. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9e. Pengurus kembali mengentrikan data pada <i>form</i>	
Jika simpanan sukarela bernilai minus	
	7f. Menampilkan <i>exception</i> “Nominal simpanan sukarela tidak boleh bernilai minus! Mohon entri ulang!”
	8f. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9f. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisi tanggal transaksi	
	7g. Menampilkan <i>exception</i> “Tanggal transaksi belum diisi! mohon entri ulang!”
	8g. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9g. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

2. Transaksi Simpanan Anggota (*Edit-Update*)

Menjelaskan tentang skenario transaksi simpanan anggota (*edit-update*). Use case ini menjelaskan alur untuk ubah data transaksi simpanan anggota pada sistem. *Use case scenario* transaksi simpanan anggota (*edit-update*) dapat dilihat pada tabel 4.11.

Tabel 4.11 *Use Case Scenario* Transaksi Simpanan Anggota (*Edit-Update*)

Nama <i>Use Case</i>	Simpanan Anggota Koperasi (<i>edit-update</i>)	
Aktor	Pengurus	
Pre-Conditions	Pengurus memilih fitur simpanan koperasi	
Post-Conditions	Pengurus telah selesai meng- <i>update</i> transaksi simpanan anggota pada <i>database</i>	
Skenario Utama		
Pengurus	Reaksi Sistem	
1. Pengurus mengakses fitur simpanan anggota pada <i>menu</i> transaksi		
	2. Menampilkan halaman transaksi simpanan anggota koperasi	
3. Pengurus menekan tombol ubah dan klik 'OK'		
	4. Menampilkan <i>form</i> edit data simpanan anggota koperasi	
5. Pengurus mengisi <i>form</i> sesuai data yang sesuai		
6. Pengurus menekan tombol ' <i>Update data</i> ' dan klik 'OK'		
	7. Sistem meng- <i>update</i> data pada <i>database</i>	
	8. <i>Redirect</i> kembali pada halaman transaksi simpanan anggota koperasi	
Skenario Alternatif		
Skenario alternatif jika simpanan wajib anggota bernilai desimal, huruf dan		

simbol	
	7a. Menampilkan <i>exception</i> “Nominal simpanan wajib tidak diijinkan desimal, huruf, atau simbol! Mohon entri ulang!”
	8a. Menampilkan halaman <i>form</i> edit data simpanan anggota koperasi
9a. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika simpanan wajib anggota bernilai minus	
	7b. Menampilkan <i>exception</i> “Nominal simpanan wajib tidak boleh bernilai minus! Mohon entri ulang!”
	8b. Menampilkan halaman <i>form</i> edit data simpanan anggota koperasi
9b. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika simpanan sukarela bernilai desimal, huruf dan simbol	
	7c. Menampilkan <i>exception</i> “Nominal simpanan sukarela tidak diijinkan desimal, huruf, atau simbol! Mohon entri ulang!”
	8c. Menampilkan halaman <i>form</i> edit data simpanan anggota koperasi
9c. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika simpanan sukarela bernilai minus	
	7d. Menampilkan <i>exception</i> “Nominal

	simpanan sukarela tidak boleh bernilai minus! Mohon entri ulang!”
	8d. Menampilkan halaman <i>form</i> edit data simpanan anggota koperasi
9d. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika simpanan tabungan bernilai minus	
	7e. Menampilkan <i>exception</i> “Nominal simpanan tabungan tidak diijinkan desimal, huruf, atau simbol! Mohon entri ulang!”
	8e. Menampilkan halaman <i>form</i> edit data simpanan anggota koperasi
9e. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika simpanan tabungan bernilai minus	
	7f. Menampilkan <i>exception</i> “Nominal simpanan tabungan tidak boleh bernilai minus! Mohon entri ulang!”
	8f. Menampilkan halaman <i>form</i> edit data simpanan anggota koperasi
9f. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

4.5.4. Perancangan Inkremen 1 *Scenario* Transaksi Pinjaman

Skenario yang menjelaskan tentang fitur untuk transaksi unit simpan pinjam anggota yang dilakukan oleh pengurus koperasi.

Use case scenario transaksi pinjaman anggota dapat dilihat pada tabel 4.12.

Tabel 4.12 *Use Case Scenario* Transaksi Pinjaman Anggota

Nama <i>Use Case</i>	Transaksi Pinjaman Anggota	
Aktor	Pengurus	
Pre-Conditions	Pengurus memilih fitur pinjaman koperasi	
Post-Conditions	Pengurus telah selesai mengentri transaksi pinjaman anggota pada <i>database</i>	
Skenario Utama		
Pengurus	Reaksi Sistem	
1. Pengurus mengakses fitur pinjaman anggota pada <i>menu</i> transaksi		
	2. Menampilkan halaman transaksi pinjaman anggota koperasi	
3. Pengurus menekan tombol <i>new data</i> dan klik 'OK'		
	4. Menampilkan <i>form</i> entri pinjaman koperasi	
5. Pengurus mengisi <i>form</i> sesuai data yang sesuai		
6. Pengurus menekan tombol 'Entri Sekarang' dan klik 'OK'		
	7. Sistem meng- <i>update</i> data pada <i>database</i>	
	8. <i>Redirect</i> kembali pada halaman transaksi pinjaman anggota koperasi	
Skenario Alternatif		

Skenario alternatif jika belum memilih anggota yang melakukan transaksi	
	7a. Menampilkan <i>exception</i> “Nama anggota belum diisikan! mohon entri ulang!”
	8a. Menampilkan halaman <i>form</i> entri pinjaman anggota koperasi
9a. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika anggota yang melakukan transaksi masih memiliki tanggungan koperasi	
	7b. Menampilkan <i>exception</i> “Anggota tersebut sebelumnya masih memiliki tanggungan pinjaman koperasi! mohon entri ulang!”
	8b. Menampilkan halaman <i>form</i> entri pinjaman anggota koperasi
9b. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisi tanggal transaksi	
	7c. Menampilkan <i>exception</i> “Tanggal transaksi belum diisi! mohon entri ulang!”
	8c. Menampilkan halaman <i>form</i> entri pinjaman anggota koperasi
9c. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum memilih jenis transaksi pinjaman	
	7d. Menampilkan <i>exception</i> “Jenis

	transaksi belum ditentukan! mohon entri ulang!”
	8d. Menampilkan halaman <i>form</i> entri pinjaman anggota koperasi
9d. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum memilih pinjaman berupa jasa atau barang	
	7e. Menampilkan <i>exception</i> “Barang atau jasa belum dipilih! mohon entri ulang!”
	8e. Menampilkan halaman <i>form</i> entri pinjaman anggota koperasi
9e. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisi nominal pengajuan	
	7f. Menampilkan <i>exception</i> “Nominal pinjaman belum diisikan! mohon entri ulang!”
	8f. Menampilkan halaman <i>form</i> entri pinjaman anggota koperasi
9f. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nominal pengajuan bernilai desimal, huruf dan simbol	
	7g. Menampilkan <i>exception</i> “Nominal pinjaman tidak diijinkan desimal, huruf, atau simbol! mohon entri ulang!”
	8g. Menampilkan halaman <i>form</i> edit

	data simpanan anggota koperasi
9g. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nominal pengajuan bernilai minus	
	7g. Menampilkan <i>exception</i> “Nominal pinjaman tidak boleh bernilai minus! mohon entri ulang!”
	8g. Menampilkan halaman <i>form</i> edit data simpanan anggota koperasi
9g. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum memilih frekuensi transaksi pinjaman	
	7h. Menampilkan <i>exception</i> “Frekuensi angsuran belum dipilih! mohon entri ulang!”
	8h. Menampilkan halaman <i>form</i> entri pinjaman anggota koperasi
9h. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

4.5.5. Perancangan Inkremen 1 *Scenario* Transaksi Angsuran

Skenario yang menjelaskan tentang fitur untuk transaksi angsuran pinjaman anggota yang dilakukan oleh pengurus koperasi.

Use case scenario transaksi simpanan anggota (*entry*) dapat dilihat pada tabel 4.13.

Tabel 4.13 *Use Case Scenario* Transaksi Angsuran Anggota (*Entry*)

Nama <i>Use Case</i>	Transaksi Angsuran Anggota	
Aktor	Pengurus	
Pre-Conditions	Pengurus memilih fitur angsuran koperasi	
Post-Conditions	Pengurus telah selesai mengentri transaksi angsuran anggota pada <i>database</i>	
Skenario Utama		
Pengurus	Reaksi Sistem	
1. Pengurus mengakses fitur angsuran anggota pada <i>menu</i> transaksi		
	2. Menampilkan halaman transaksi angsuran anggota koperasi	
3. Pengurus menekan tombol angsur dan klik 'OK'		
	4. Menampilkan <i>form</i> entri data angsuran anggota koperasi	
5. Pengurus mengisi <i>form</i> sesuai data yang sesuai		
6. Pengurus menekan tombol 'Tambah Sekarang'		
	7. Sistem meng- <i>update</i> data pada <i>database</i>	
	8. <i>Redirect</i> kembali pada halaman transaksi angsuran anggota koperasi	
Skenario Alternatif		
Skenario alternatif jika tanggal transaksi belum dientrikan		

	7. Menampilkan <i>exception</i> “Tanggal transaksi belum diisi! mohon entri ulang!”
	8. Menampilkan halaman <i>form</i> entri data angsuran anggota koperasi
9. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

4.5.6. Perancangan Inkremen 2 *Scenario* Manajemen Lembaga

Skenario yang menjelaskan tentang fitur entri data lembaga *edit* data lembaga dan *delete* data lembaga. Sebuah fitur yang merupakan fitur untuk manajemen data lembaga yang merupakan lingkup dari koperasi.

1. Manajemen Lembaga (*Entry*)

Menjelaskan tentang skenario manajemen lembaga (*entry*). Use case ini menjelaskan alur untuk entri data lembaga baru kedalam sistem.

Use case scenario manajemen lembaga (*entry*) dapat dilihat pada tabel 4.14.

Tabel 4.14 *Use Case Scenario* Manajemen Lembaga (*Entry*)

Nama <i>Use Case</i>	Manajemen Lembaga (<i>Entry</i>)	
Aktor	Pengurus	
Pre-Conditions	Pengurus memilih fitur lembaga	
Post-Conditions	Pengurus telah selesai mengentri data lembaga baru pada <i>database</i>	
Skenario Utama		
Pengurus	Reaksi Sistem	
1. Pengurus mengakses fitur lembaga pada <i>menu</i> petugas		

	2. Menampilkan halaman daftar lembaga
3. Pengurus menekan tombol ' <i>new data</i> ' dan klik 'OK'	
	4. Menampilkan <i>form</i> entri lembaga
5. Pengurus mengisi <i>form</i> sesuai data yang sesuai	
6. Pengurus menekan tombol 'Register'	
	7. Sistem meng- <i>entry</i> data pada <i>database</i>
	8. <i>Redirect</i> kembali pada halaman daftar lembaga
Skenario Alternatif	
Skenario alternatif jika belum mengisikan nama lembaga	
	7a. Menampilkan <i>exception</i> "Nama lembaga belum belum diisi! mohon entri ulang!"
	8a. Menampilkan halaman <i>form</i> entri lembaga
9a. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisikan alamat lembaga	
	7b. Menampilkan <i>exception</i> "Alamat lembaga belum belum diisi! mohon entri ulang!"
	8b. Menampilkan halaman <i>form</i> entri lembaga

9b. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisi nomor telepon	
	7c. Menampilkan <i>exception</i> “Nomor telepon belum diisi! mohon entri ulang!”
	8c. Menampilkan halaman <i>form</i> entri lembaga
9c. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon bernilai desimal, huruf dan simbol	
	7d. Menampilkan <i>exception</i> “Nomor telepon tidak diijinkan desimal, huruf, atau simbol! Mohon entri ulang!”
	8d. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9d. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon bernilai minus	
	7e. Menampilkan <i>exception</i> “Nomor telepon tidak boleh bernilai minus! Mohon entri ulang!”
	8e. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9e. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisi alamat email lembaga	
	7f. Menampilkan <i>exception</i> “Email

	lembaga belum belum diisi! mohon entri ulang!”
	8f. Menampilkan halaman <i>form</i> entri lembaga
9f. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

2. Manajemen Lembaga (*Edit-Update*)

Menjelaskan tentang skenario manajemen lembaga (*edit-update*). Use case ini menjelaskan alur untuk ubah data lembaga pada sistem.

Use case scenario manajemen lembaga (*edit-update*) dapat dilihat pada tabel 4.15.

Tabel 4.15 *Use Case Scenario* Manajemen Lembaga (*Edit-Update*)

Nama <i>Use Case</i>	Manajemen Lembaga (<i>Edit-Update</i>)
Aktor	Pengurus
Pre-Conditions	Pengurus memilih fitur lembaga
Post-Conditions	Pengurus telah selesai mengubah data lembaga pada <i>database</i>
Skenario Utama	
Pengurus	Reaksi Sistem
1. Pengurus mengakses fitur lembaga pada <i>menu</i> petugas	
	2. Menampilkan halaman daftar lembaga
3. Pengurus menekan tombol ' <i>ubah</i> ' dan klik 'OK'	
	4. Menampilkan <i>form</i> edit data

	lembaga
5. Pengurus mengisi <i>form</i> sesuai data yang sesuai	
6. Pengurus menekan tombol 'Update Data'	
	7. Sistem meng- <i>entry</i> data pada <i>database</i>
	8. <i>Redirect</i> kembali pada halaman daftar lembaga
Skenario Alternatif	
Skenario alternatif jika belum mengisikan nomor telepon	
	7a. Menampilkan <i>exception</i> "Nomor telepon belum belum diisi! mohon entri ulang!"
	8a. Menampilkan halaman <i>form</i> entri lembaga
9a. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon bernilai desimal, huruf dan simbol	
	7b. Menampilkan <i>exception</i> "Nomor telepon tidak diijinkan desimal, huruf, atau simbol! Mohon entri ulang!"
	8b. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9b. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika nomor telepon bernilai minus	
	7c. Menampilkan <i>exception</i> "Nomor

	telepon tidak boleh bernilai minus! Mohon entri ulang!”
	8c. Menampilkan halaman <i>form</i> entri simpanan anggota koperasi
9c. Pengurus kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisikan alamat email lembaga	
	7d. Menampilkan <i>exception</i> “Email lembaga belum belum diisi! mohon entri ulang!”
	8d. Menampilkan halaman <i>form</i> entri lembaga
9d. Pengurus kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

4.5.7. Perancangan Inkremen 3 *Scenario* Cetak Laporan

Skenario yang menjelaskan tentang fitur untuk memproses laporan – laporan transaksi menjadi berupa *file* pdf ataupun *preview* laporan yang akan dicetak menjadi *hard copy*.

Menjelaskan tentang skenario cetak laporan transaksi. Use case ini menjelaskan alur untuk memproses data transaksi menjadi laporan – laporan transaksi.

Use case scenario cetak laporan transaksi dapat dilihat pada tabel 4.16.

Tabel 4.16 *Use Case Scenario* Cetak Laporan Transaksi

Nama <i>Use Case</i>	Cetak Laporan Transaksi
Aktor	Pengurus

Pre-Conditions	Pengurus memilih fitur laporan
Post-Conditions	Pengurus telah selesai memproses laporan transaksi
Skenario Utama	
Pengurus	Reaksi Sistem
1. Pengurus mengakses fitur laporan pada <i>menu</i> petugas	
	2. Menampilkan halaman fitur manajemen sistem laporan koperasi
3. Pengurus men- <i>select selectbox</i> yang diinginkan pada fitur laporan	
	4. Sistem memproses data menjadi laporan transaksi
	5. Sistem menampilkan pada halaman web, dan laporan siap untuk dicetak.
Skenario Alternatif	

(Sumber: Hasil Analisis, 2015)

4.5.8. Perancangan Inkremen 4 *Scenario* Manajemen Pengurus

Skenario yang menjelaskan tentang fitur entri data pengurus koperasi dan *edit password* pengurus koperasi. Sebuah fitur yang merupakan fitur untuk manajemen data pengurus koperasi yang bertanggung jawab akan pengoperasian sistem.

1. Manajemen Pengurus (*Entry*)

Menjelaskan tentang skenario manajemen pengurus (*entry*). Use case ini menjelaskan alur untuk entri data pengurus kedalam sistem.

Use case scenario manajemen pengurus (*entry*) dapat dilihat pada tabel 4.17.

Tabel 4.17 *Use Case Scenario* Manajemen Pengurus (*Entry*)

Nama <i>Use Case</i>	Manajemen Pengurus (<i>Entry</i>)
Aktor	Administrator / Pimpinan
Pre-Conditions	Admin / Pimpinan mengakses menu administrator
Post-Conditions	Admin / Pimpinan telah selesai mengentri data user baru pada <i>database</i>
Skenario Utama	
Pengurus	Reaksi Sistem
1. User mengakses <i>menu</i> administrator	
	2. Menampilkan halaman daftar lembaga
3. User menekan tombol ‘registrasi <i>user</i> baru’ dan klik ‘OK’	
	4. Menampilkan <i>form</i> registrasi user
5. User mengisi <i>form</i> sesuai data yang sesuai	
6. User menekan tombol ‘Register’	
	7. Sistem meng- <i>entry</i> data pada <i>database</i>
	8. <i>Redirect</i> kembali pada halaman administrator
Skenario Alternatif	
Skenario alternatif jika belum mengisikan nama user	
	7a. Menampilkan <i>exception</i> “Nama user belum diisikan! mohon entri ulang!”
	8a. Menampilkan halaman <i>form</i>

	registrasi user
9a. User kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisikan <i>username</i>	
	7b. Menampilkan <i>exception</i> “Username belum diisi! mohon entri ulang!”
	8b. Menampilkan halaman <i>form</i> registrasi user
9b. User kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisikan <i>password</i>	
	7b. Menampilkan <i>exception</i> “Password belum diisi! mohon entri ulang!”
	8b. Menampilkan halaman <i>form</i> registrasi user
9b. User kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum mengisikan konfirmasi <i>password</i>	
	7c. Menampilkan <i>exception</i> “Konfirmasi password belum diisi! mohon entri ulang!”
	8c. Menampilkan halaman <i>form</i> registrasi user
9c. User kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika kombinasi <i>password</i> salah	
	7d. Menampilkan <i>exception</i> “Konfirmasi password salah! mohon

	entri ulang!”
	8d. Menampilkan halaman <i>form</i> registrasi user
9d. User kembali mengentrikan data pada <i>form</i>	
Skenario alternatif jika belum men- <i>select type</i> user	
	7e. Menampilkan <i>exception</i> “Type user belum dipilih! mohon entri ulang!”
	8e. Menampilkan halaman <i>form</i> registrasi user
9e. User kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

2. Manajemen Pengurus (*Update-Password*)

Menjelaskan tentang skenario manajemen lembaga (*entry*). Use case ini menjelaskan alur untuk entri data lembaga baru kedalam sistem.

Use case scenario manajemen pengurus (*update-password*) dapat dilihat pada tabel 4.18.

Tabel 4.18 *Use Case Scenario* Manajemen Pengurus (*Update-Password*)

Nama <i>Use Case</i>	Manajemen Pengurus (<i>Update-Password</i>)
Aktor	Administrator / Pimpinan
Pre-Conditions	Admin / Pimpinan mengakses menu administrator
Post-Conditions	Admin / Pimpinan telah selesai mengubah <i>password user</i> pada <i>database</i>
Skenario Utama	
Pengurus	Reaksi Sistem
1. User mengakses <i>menu</i> administrator	

	2. Menampilkan halaman daftar lembaga
3. User menekan tombol ‘ubah <i>password</i> ’ dan klik ‘OK’	
	4. Menampilkan <i>form update password</i> user
5. User mengisi <i>form</i> sesuai data yang sesuai	
6. User menekan tombol ‘Update’	
	7. Sistem mengubah data pada <i>database</i>
	8. <i>Redirect</i> kembali pada halaman administrator
Skenario Alternatif	
Skenario alternatif jika belum mengisikan <i>password</i> terbaru	
	7. Menampilkan <i>exception</i> “Password terbaru belum diisikan! mohon entri ulang!”
	8. Menampilkan halaman <i>form</i> registrasi user
9. User kembali mengentrikan data pada <i>form</i>	

(Sumber: Hasil Analisis, 2015)

4.5.9. Perancangan Inkremen 5 *Scenario* Master Data Transaksi

Skenario yang menjelaskan tentang fitur untuk memonitoring data transaksi secara menyeluruh dan lengkap tanpa adanya pembatasan.

Menjelaskan tentang skenario master data transaksi. Use case ini menjelaskan alur untuk mengakses master data transaksi yang dilakukan oleh pimpinan koperasi.

Use case scenario master data transaksi dapat dilihat pada tabel 4.19.

Tabel 4.19 *Use Case Scenario* Master Data Transaksi

Nama <i>Use Case</i>	Master Data Transaksi	
Aktor	Pimpinan	
Pre-Conditions	Pimpinan memilih menu pimpinan	
Post-Conditions	Pimpinan telah selesai mengakses fitur master data	
Skenario Utama		
Pengurus	Reaksi Sistem	
1. Pengurus mengakses <i>menu</i> pimpinan		
	2. Menampilkan halaman pimpinan	
3. Pimpinan menekan tombol master data sesuai yang diinginkan		
	4. Sistem memproses data menjadi daftar sesuai format fitur master data transaksi	
	5. Sistem menampilkan pada halaman web.	
Skenario Alternatif		

(Sumber: Hasil Analisis, 2015)

4.6. Activity Diagram

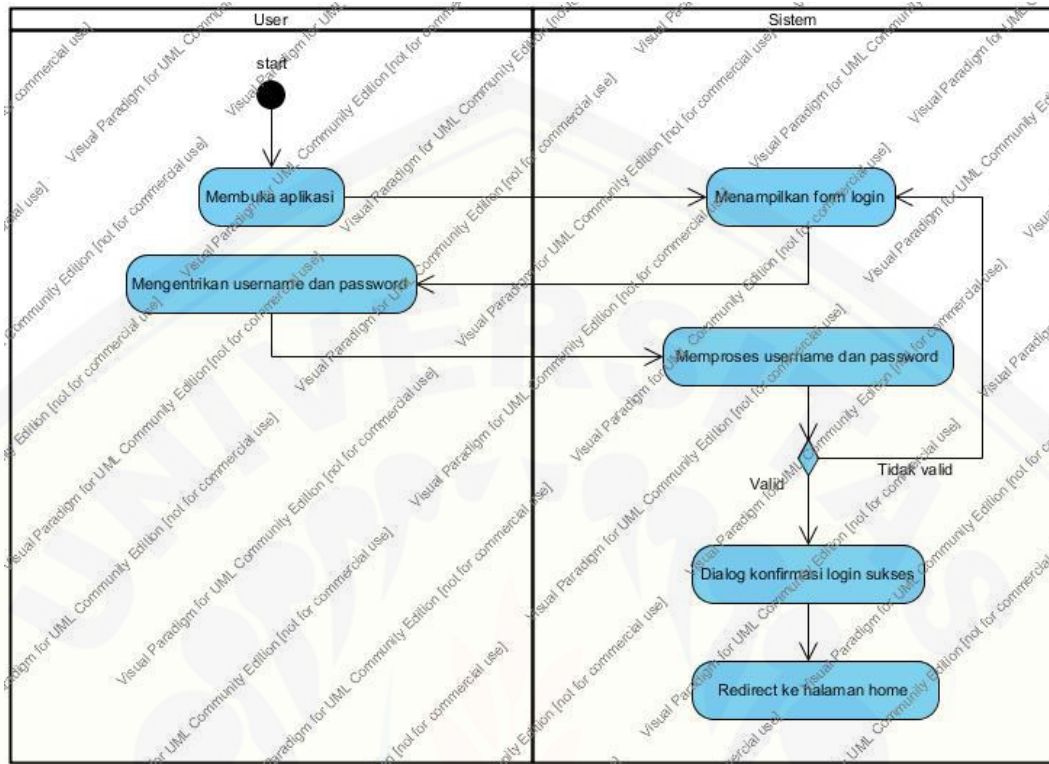
Merupakan sebuah diagram yang menjelaskan tentang alur kerja jalannya aktor ketika senang mengakses sistem yang sedang dirancang. Berikut ini merupakan gambaran dari *activity diagram* pada pembuatan sistem informasi simpan pinjam koperasi.

4.6.1. Perancangan Inkremen 1 Activity User Login

Activity diagram user login menggambarkan alir aktivitas pada proses *sign-in* dan *sign-out*. Fitur ini digunakan untuk akses masuk *user* ke dalam sistem dan sebagai pembagian hak akses pada sistem yang digunakan.

1. User Login (Log-In)

Halaman login yang ditampilkan oleh sistem berisi *field username* dan *field password* serta tombol login. Sistem akan mengecek pada database, apakah *username* dan *password* yang diisikan sesuai dengan database atau tidak. Jika sesuai maka, javascript akan mengecek level dari *user* yang sudah dikonfirmasi dan mengarahkan halaman ke home. Jika *username* dan *password* tidak sesuai pada database, maka sistem akan menampilkan halaman *login* dengan *alert* “Gagal Login!username belum diisi! “ atau “Gagal Login! password belum diisi, atau kombinasi salah!! “. *Activity diagram* user login (log-in) dapat dilihat pada gambar 4.8.

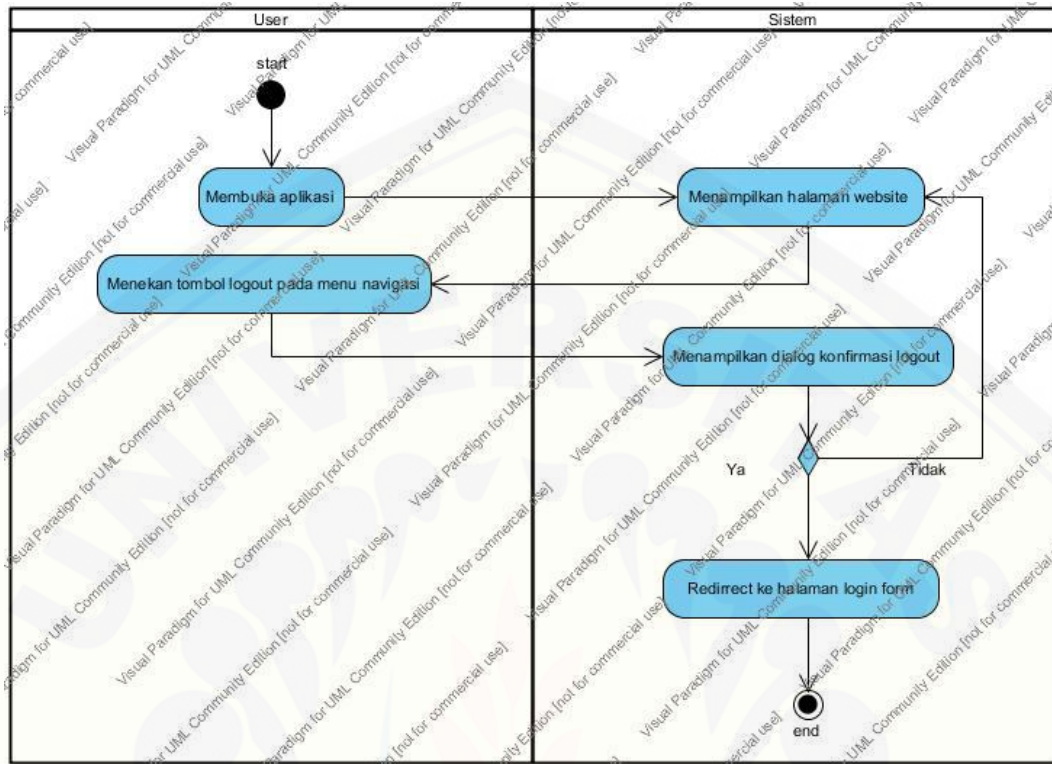


Gambar 4.8 Activity Diagram User Login (log-In)

(Sumber: Hasil Analisis, 2015)

2. User Login (log-Out)

Halaman login akan ditampilkan kembali ketika *session user* telah dihapus dari sistem karena pada *header* disetiap *view* berisipi perintah untuk mengecek apakah ada *session* yang sedang aktif atau tidak. Jika tidak ada *session* yang sedang aktif, maka *view* secara paksa akan diarahkan pada halaman *login*. Ketika *user* menekan tombol *logout*, maka akan muncul dialog konfirmasi. Jika *user* mengkonfirmasi *logout* sistem, maka secara otomatis javascript akan menghapus *session* yang ada. Activity diagram user login (log-out) dapat dilihat pada gambar 4.9.



Gambar 4.9 Activity Diagram User Login (log-Out)

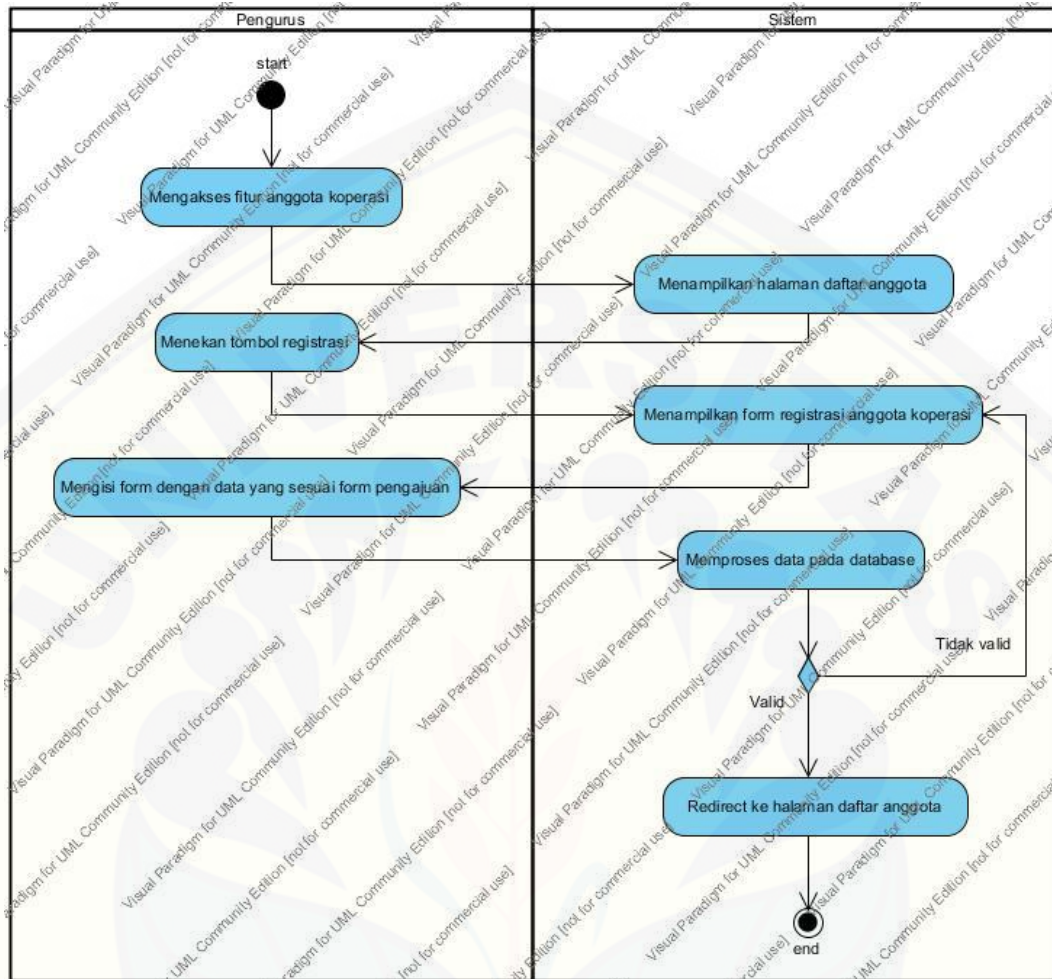
(Sumber: Hasil Analisis, 2015)

4.6.2. Perancangan Inkremen 1 Activity Registrasi Anggota

Activity diagram registrasi anggota menggambarkan alir aktivitas pada proses registrasi anggota (*entry*), registrasi anggota (*update-status*) dan registrasi anggota (*edit-update*). Fitur ini berfungsi untuk melakukan registrasi data keanggotaan baru pada koperasi.

1. Registrasi Anggota (*Entry*)

Activity diagram registrasi anggota (*entry*) dapat dilihat pada gambar 4.10.



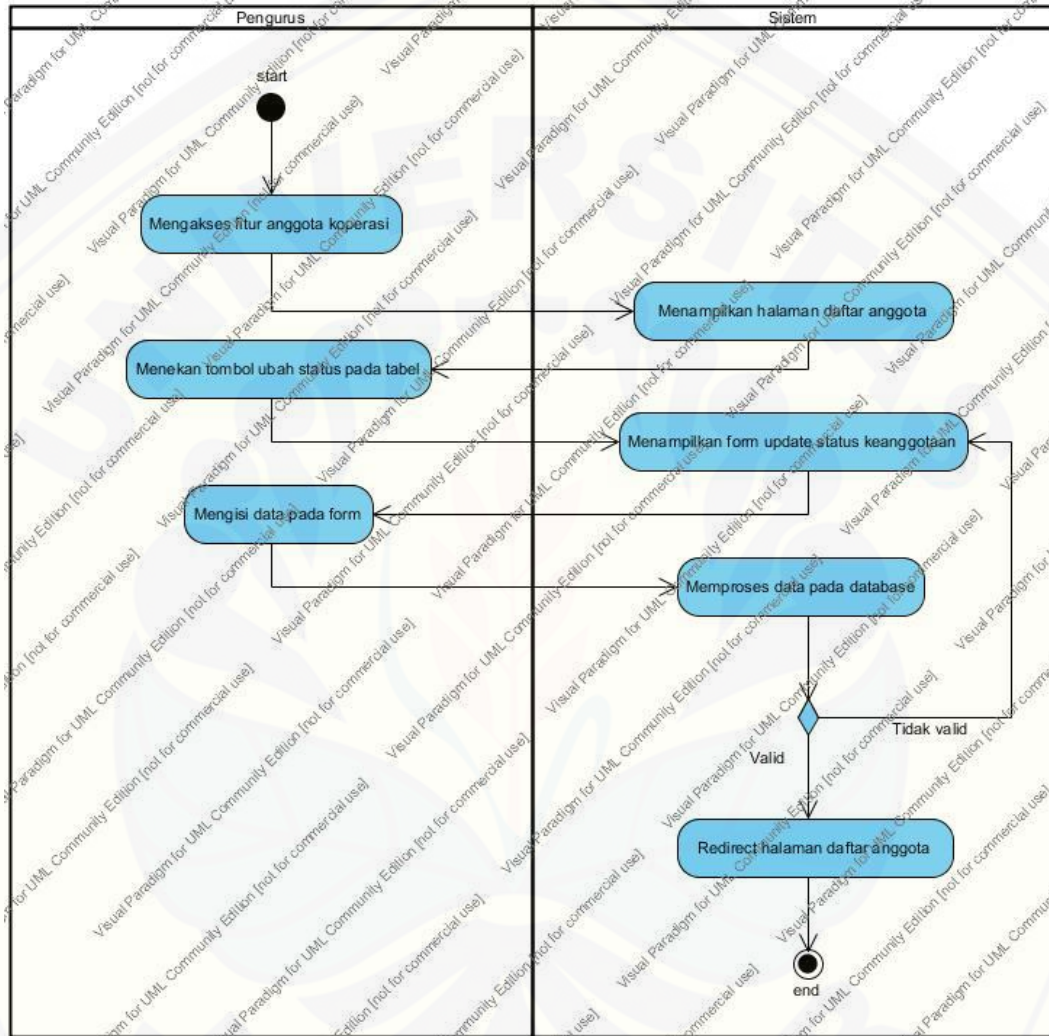
Gambar 4.10 Activity Diagram Registrasi Anggota (Entry)

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.10 setelah data diisikan dan diproses kedalam sistem, sistem akan melakukan pengecekan apakah data yang diisikan sesuai format yang diharuskan atau tidak, jika tidak sesuai maka sistem akan menjalankan *exception* pada halaman form dengan menampilkan kesalahan yang terdeteksi oleh sistem. Jika sesuai format, maka sistem akan lanjut menjalankan *query* untuk memasukkan data pada database.

2. Registrasi Anggota (*Update-Status*)

Activity diagram registrasi anggota (*update-status*) dapat dilihat pada gambar 4.11.



Gambar 4.11 *Activity Diagram* Registrasi Anggota (*Update-Status*)

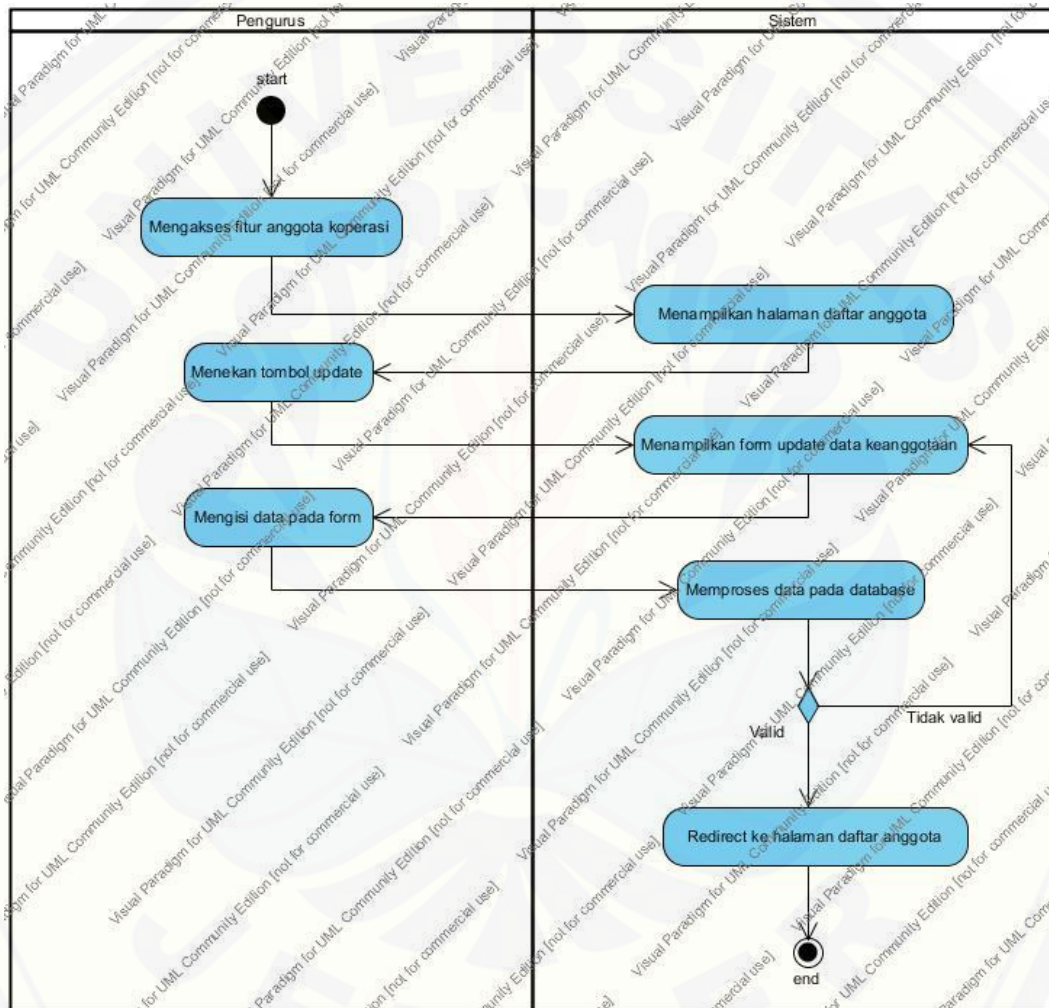
(Sumber: Hasil Analisis, 2015)

Pada gambar 4.11 setelah *user* menekan tombol ubah, sistem akan *select* id yang berkaitan pada database dan menampilkan *value* pada halaman *form* sehingga user dapat meng-*update* data dengan sesuai. Ketika kondisi *exception* terpenuhi,

maka sistem akan menampilkan *exception* pada halaman sehingga *user* dapat kembali meng-*update* data dengan benar.

3. Registrasi Anggota (*Edit-Update*)

Activity diagram registrasi anggota (*edit-update*) dapat dilihat pada gambar 4.12.



Gambar 4.12 *Activity Diagram* Registrasi Anggota (*Edit-Update*)

(Sumber: Hasil Analisis, 2015)

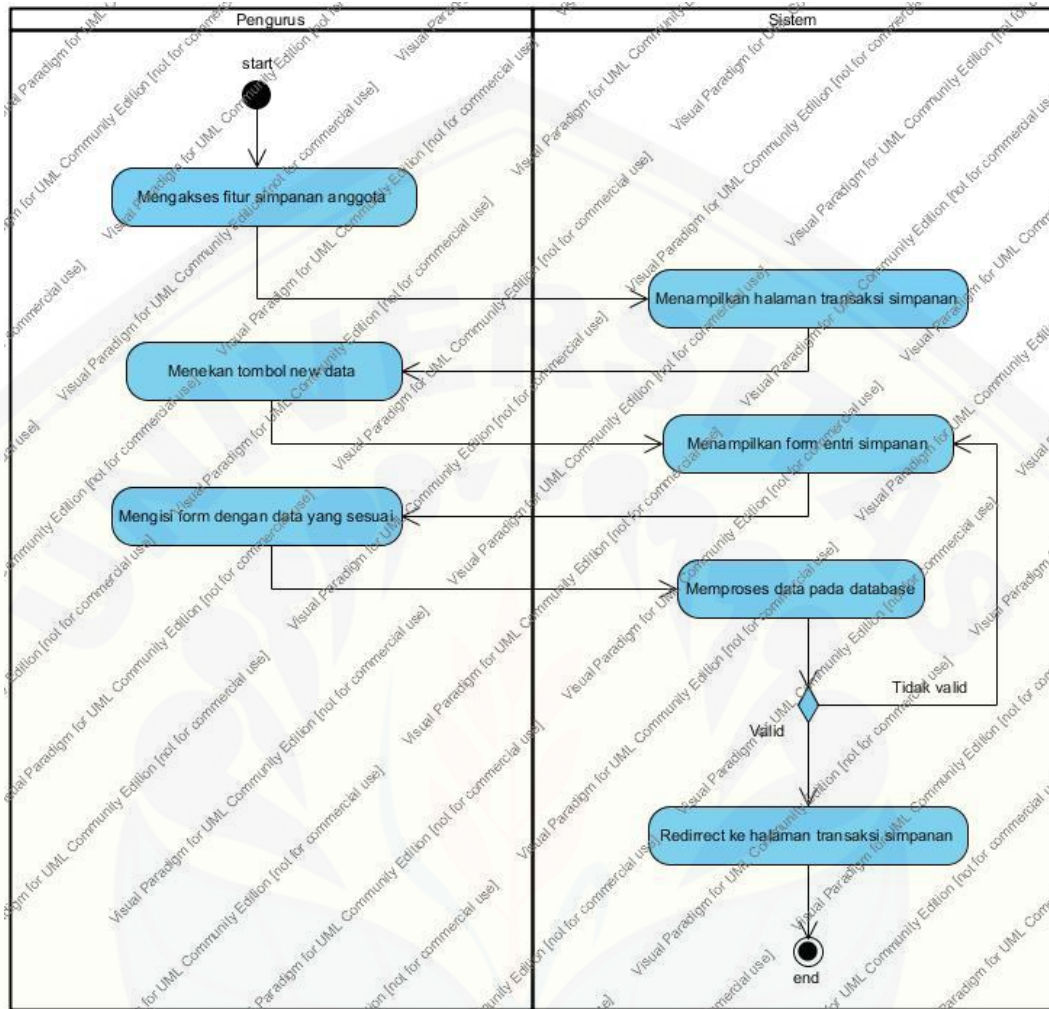
Pada gambar 4.12 setelah *user* menekan tombol ubah, sistem akan men-*select* id yang berkaitan pada database dan menampilkan *value* pada halaman *form* sehingga user dapat meng-*update* data dengan sesuai. Ketika kondisi *exception* terpenuhi, maka sistem akan menampilkan *exception* pada halaman sehingga *user* dapat kembali meng-*update* data dengan benar.

4.6.3. Perancangan Inkremen 1 *Activity* Transaksi Simpanan

Activity diagram transaksi simpanan anggota menggambarkan alir aktivitas pada proses transaksi simpanan anggota (*entry*) dan transaksi simpanan anggota (*edit-update*). Fitur ini berfungsi untuk melakukan proses transaksi simpanan anggota yang berupa simpanan wajib dan simpanan sukarela.

1. Transaksi Simpanan Anggota (*Entry*)

Activity diagram transaksi simpanan anggota (*entry*) dapat dilihat pada gambar 4.13. Pada gambar 4.13 setelah data diisikan dan diproses kedalam sistem, sistem akan melakukan pengecekan apakah data yang diisikan sesuai format yang diharuskan atau tidak, jika tidak sesuai maka sistem akan menjalankan *exception* pada halaman form dengan menampilkan kesalahan yang terdeteksi oleh sistem. Jika sesuai format, maka sistem akan lanjut menjalankan *query* untuk memasukkan data pada database.

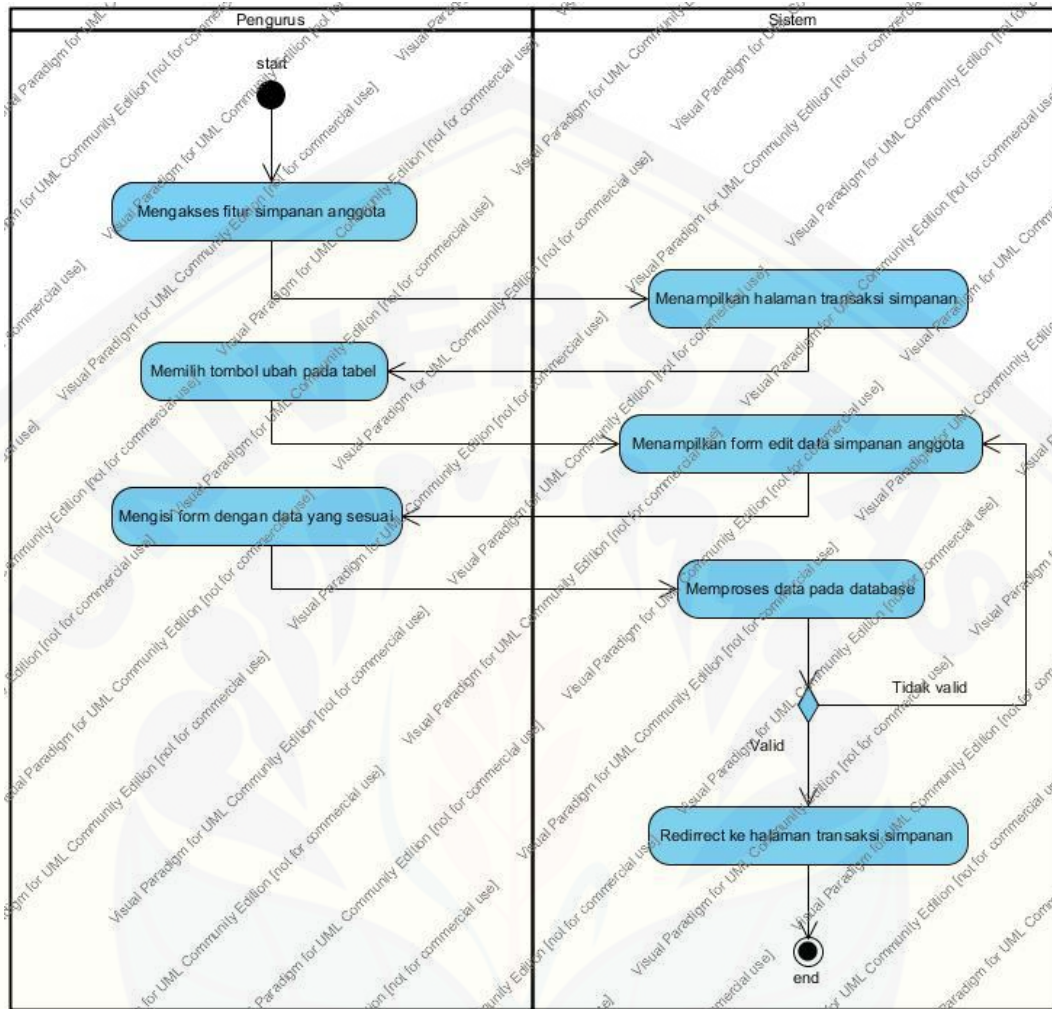


Gambar 4.13 *Activity Diagram* Transaksi Simpanan Anggota (*Entry*)

(Sumber: Hasil Analisis, 2015)

2. Transaksi Simpanan Anggota (*Edit-Update*)

Activity diagram transaksi simpanan anggota (*edit-update*) dapat dilihat pada gambar 4.14.



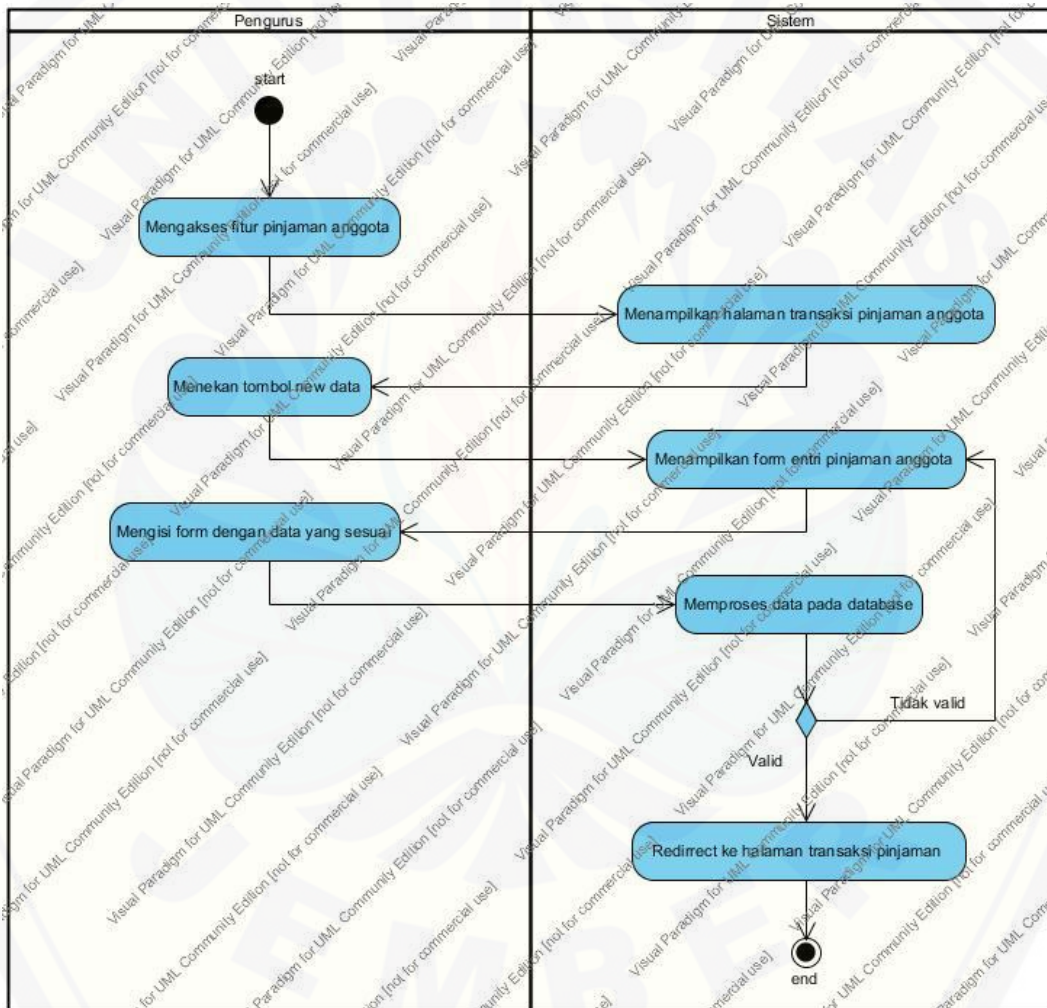
Gambar 4.14 *Activity Diagram* Transaksi Simpanan (*Edit-Update*)

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.14 setelah *user* menekan tombol ubah, sistem akan men-select id yang berkaitan pada database dan menampilkan value pada halaman form sehingga user dapat meng-update data dengan sesuai. Ketika kondisi *exception* terpenuhi, maka sistem akan menampilkan *exception* pada halaman sehingga user dapat kembali mengupdate data dengan benar.

4.6.4. Perancangan Inkremen 1 *Activity* Transaksi Pinjaman

Activity diagram transaksi pinjaman anggota menggambarkan alir aktivitas pada proses transaksi pinjaman berupa uang atau unit simpan pinjam (usp). Fitur ini berfungsi untuk melakukan proses transaksi pinjaman anggota yang berupa pinjaman usp ataupun pinjaman insidental. *Activity diagram* transaksi pinjaman anggota dapat dilihat pada gambar 4.15.



Gambar 4.15 *Activity Diagram* Transaksi Pinjaman Anggota

(Sumber: Hasil Analisis, 2015)

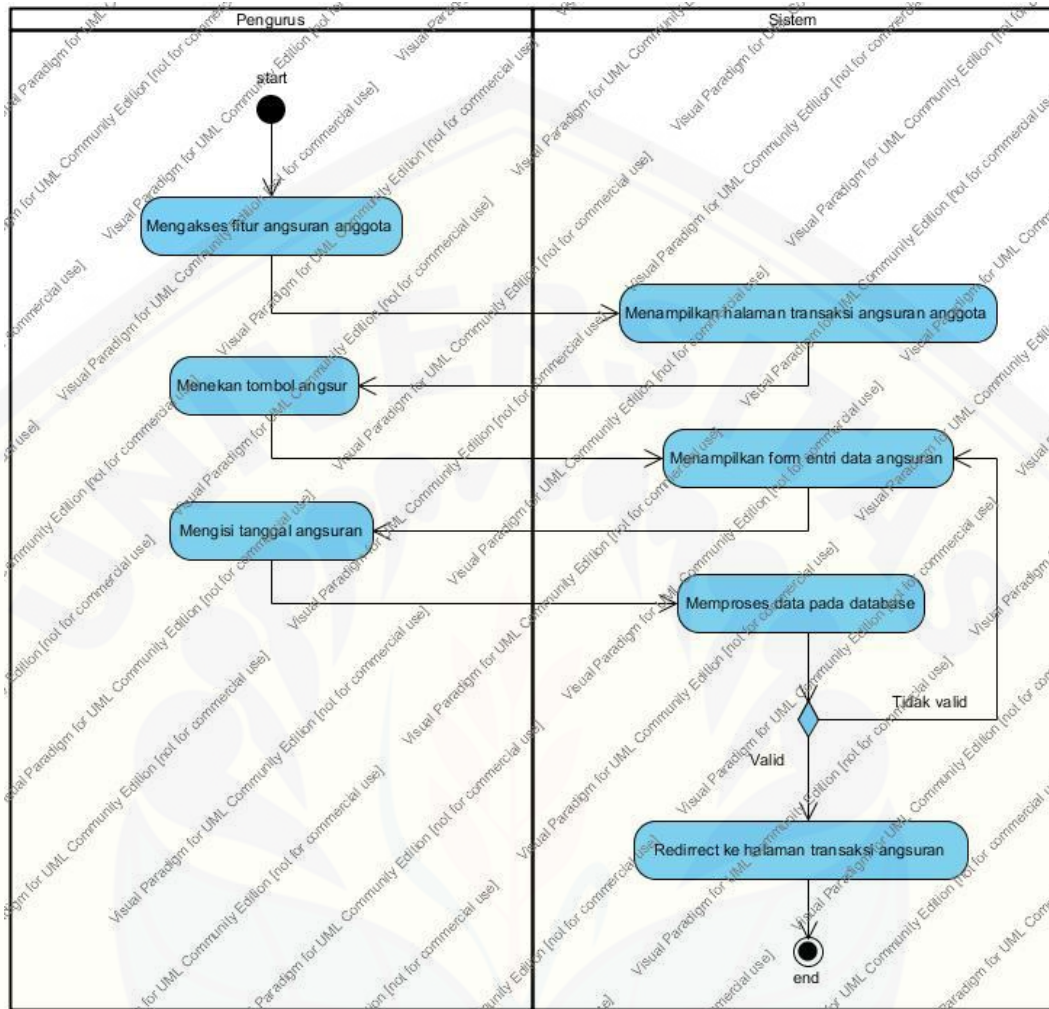
Pada gambar 4.15 setelah data diisikan dan diproses kedalam sistem, sistem akan melakukan pengecekan apakah data yang diisikan sesuai format yang diharuskan atau tidak, jika tidak sesuai maka sistem akan menjalankan *exception* pada halaman form dengan menampilkan kesalahan yang terdeteksi oleh sistem. Jika sesuai format, maka sistem akan lanjut menjalankan *query* untuk memasukkan data pada database.

4.6.5. Perancangan Inkremen 1 *Activity* Transaksi Angsuran

Activity diagram transaksi angsuran anggota menggambarkan alir aktivitas pada proses transaksi angsuran terhadap pinjaman berupa barang atau jasa yang dilakukan oleh anggota koperasi.

Pada gambar 4.16 setelah data diisikan dan diproses kedalam sistem, sistem akan melakukan pengecekan apakah data yang diisikan sesuai format yang diharuskan atau tidak, jika tidak sesuai maka sistem akan menjalankan *exception* pada halaman *form* dengan menampilkan kesalahan yang terdeteksi oleh sistem. Jika sesuai format, maka sistem akan lanjut menjalankan *query* untuk memasukkan data pada database

Activity diagram transaksi angsuran anggota dapat dilihat pada gambar 4.16.



Gambar 4.16 Activity Diagram Transaksi Angsuran Anggota

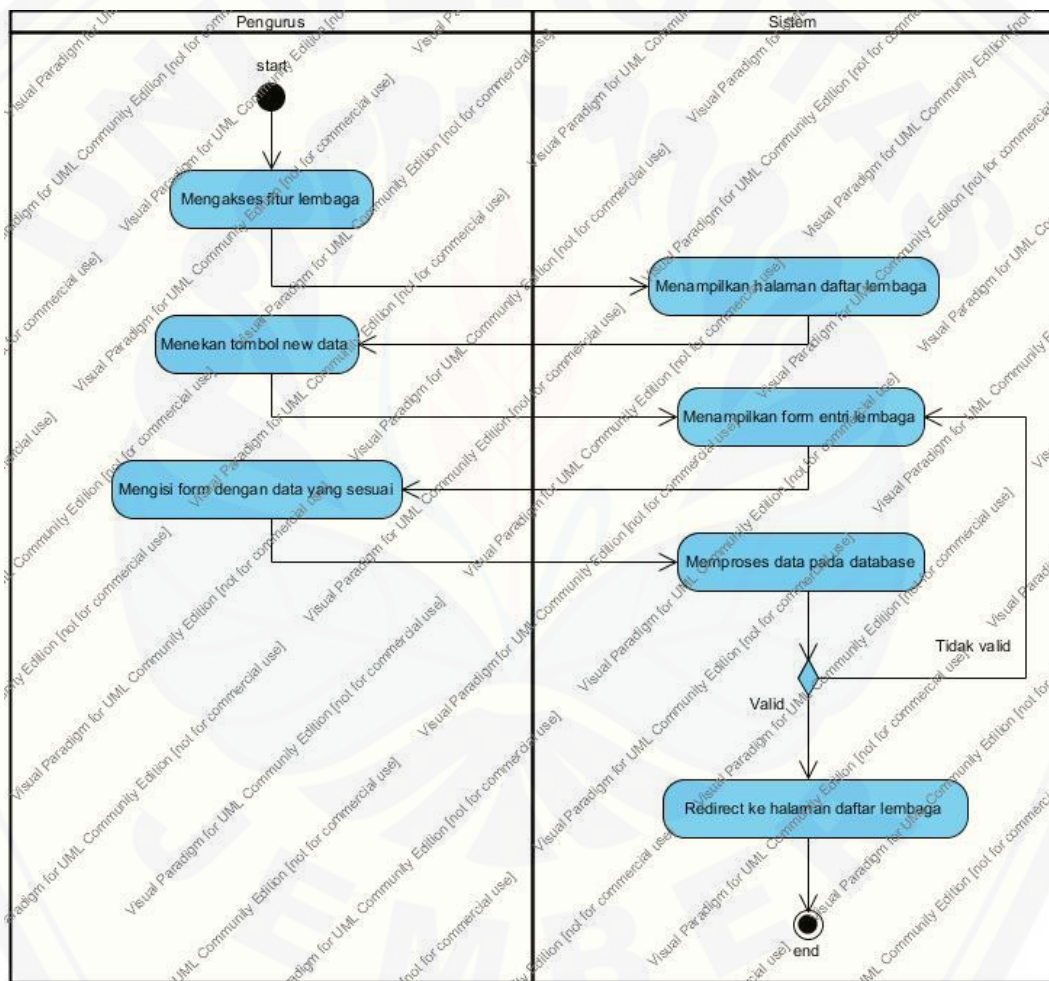
(Sumber: Hasil Analisis, 2015)

4.6.6. Perancangan Inkremen 2 Activity Manajemen Lembaga

Activity diagram manajemen lembaga menggambarkan alir aktivitas pada proses manajemen lembaga (*entry*), manajemen lembaga (*edit-update*) dan manajemen lembaga (*delete*). Fitur ini berfungsi mendaftarkan dan memanajemen informasi data terkait lembaga – lembaga yang berada dan terdaftar pada lingkup koperasi.

1. Manajemen Lembaga (*Entry*)

Activity diagram manajemen lembaga (*entry*) dapat dilihat pada gambar 4.17. Pada gambar 4.17 setelah data diisikan dan diproses kedalam sistem, sistem akan melakukan pengecekan apakah data yang diisikan sesuai format yang diharuskan atau tidak, jika tidak sesuai maka sistem akan menjalankan *exception* pada halaman *form* dengan menampilkan kesalahan yang terdeteksi oleh sistem. Jika sesuai format, maka sistem akan lanjut menjalankan *query* untuk memasukkan data pada database.

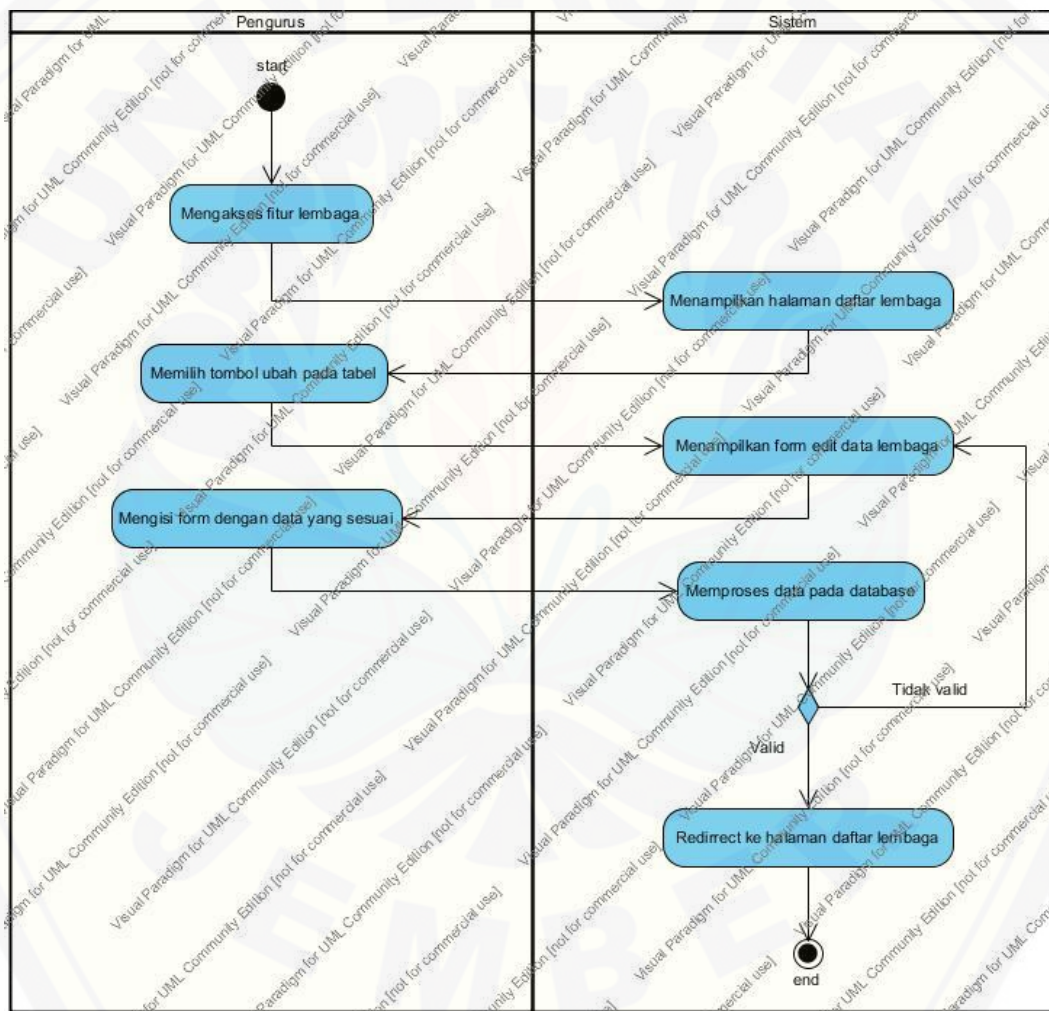


Gambar 4.17 *Activity Diagram* Manajemen Lembaga (*Entry*)

(Sumber: Hasil Analisis, 2015)

2. Manajemen Lembaga (*Edit-Update*)

Activity diagram manajemen lembaga (*edit-update*) dapat dilihat pada gambar 4.18. Pada gambar 4.18 setelah *user* menekan tombol ubah, sistem akan men-select id yang berkaitan pada database dan menampilkan value pada halaman form sehingga user dapat meng-update data dengan sesuai. Ketika kondisi *exception* terpenuhi, maka sistem akan menampilkan *exception* pada halaman sehingga user dapat kembali mengupdate data dengan benar.



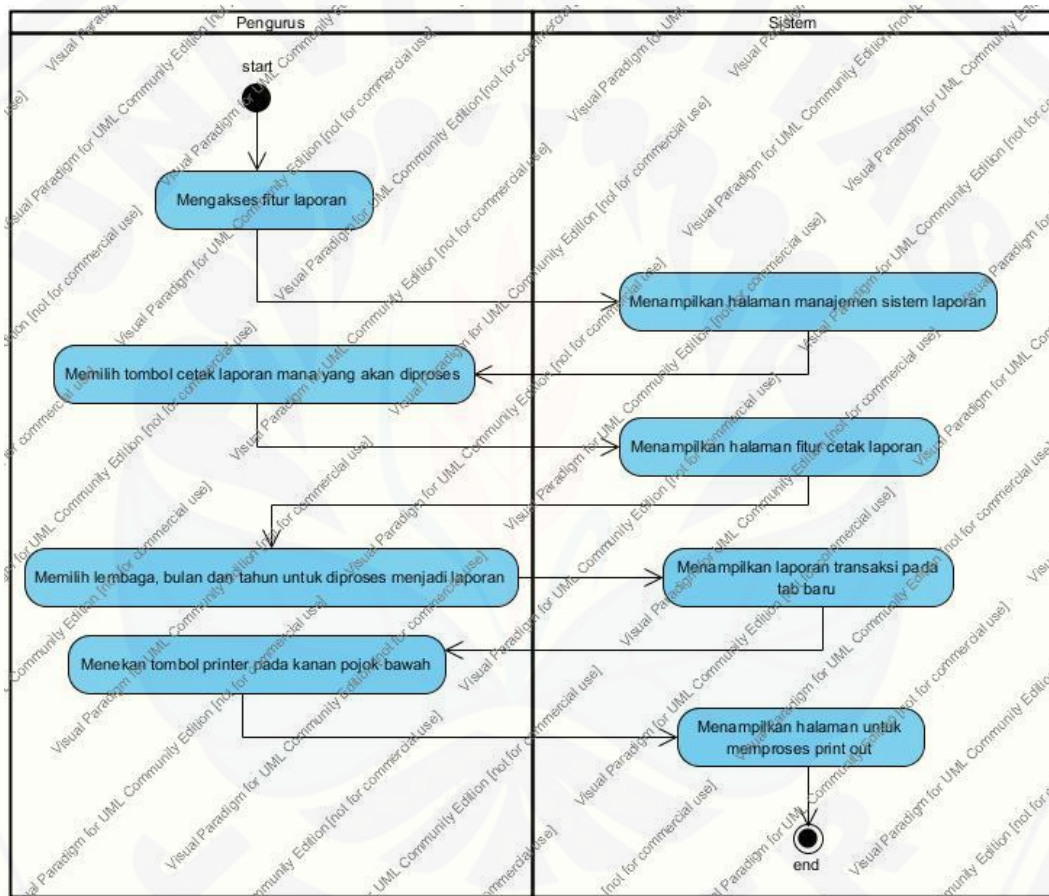
Gambar 4.18 *Activity Diagram* Manajemen Lembaga (*Edit-Update*)

(Sumber: Hasil Analisis, 2015)

4.6.7. Perancangan Inkremen 3 *Activity* Cetak Laporan

Activity diagram cetak laporan transaksi menggambarkan alir aktivitas pada pemrosesan data laporan. Fitur ini berfungsi untuk mencetak dan memproses laporan transaksi yang berupa daftar tagihan, buku kas harian, rekap daftar tagihan, bukti kas masuk dan bukti kas keluar.

Activity diagram cetak laporan transaksi dapat dilihat pada gambar 4.19.



Gambar 4.19 *Activity Diagram* Cetak Laporan Transaksi

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.19 user men-select nama lembaga yang akan diproses laporan transaksinya, bulan transaksi dan tahun transaksi. Maka secara otomatis, sistem akan memproses laporan transaksinya menjadi file pdf ataupun kedalam bentuk *hard copy (print)*.

4.6.8. Perancangan Inkremen 4 *Activity* Manajemen Pengurus

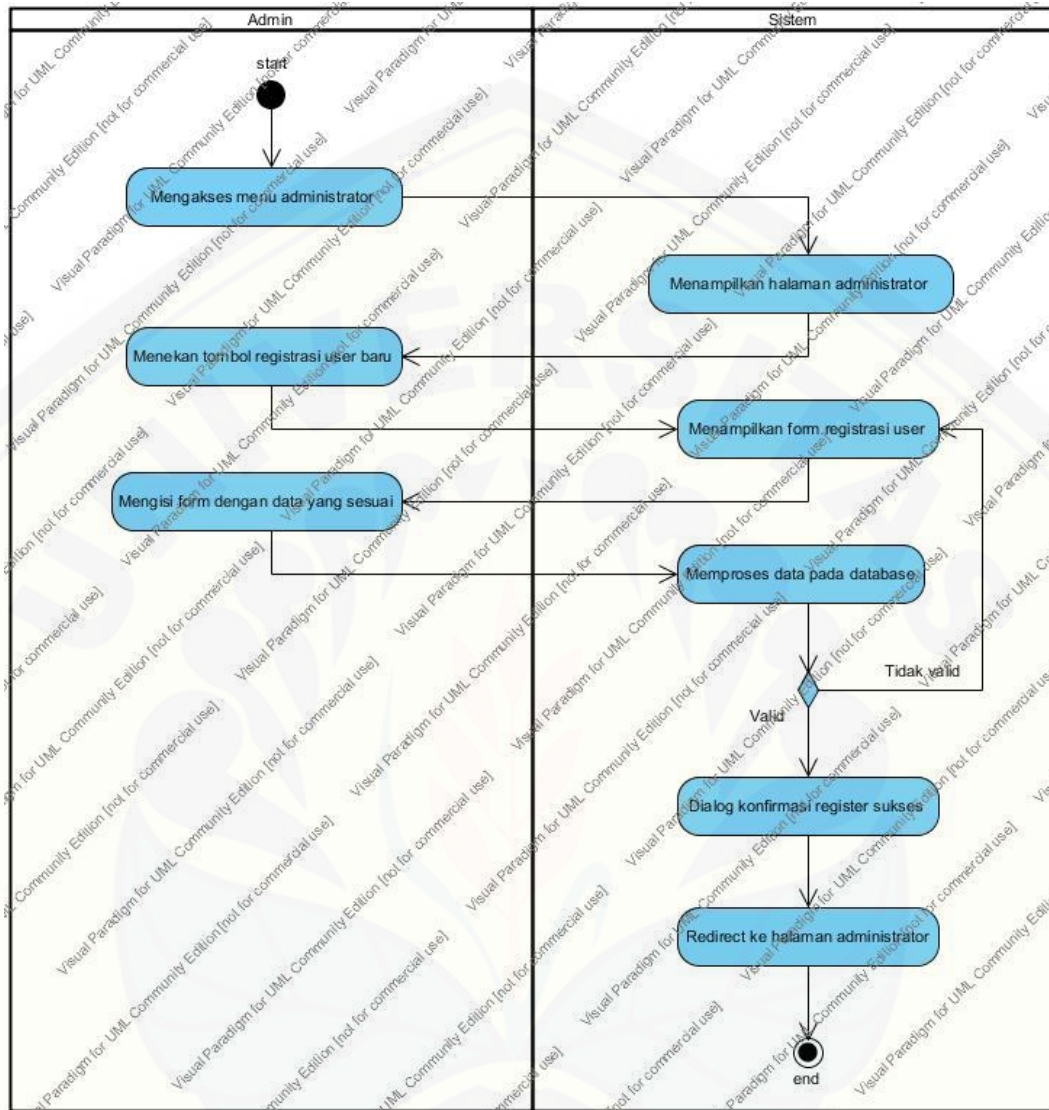
Activity diagram manajemen pengurus menggambarkan alir aktivitas pada proses manajemen pengurus (*entry*) dan manajemen pengurus (*update-password*). Fitur administrator ini berfungsi untuk mendaftarkan dan mengubah *password user* yang bertanggung jawab atas akses terhadap sistem.

1. Manajemen Pengurus (*Entry*)

Activity diagram manajemen pengurus (*entry*) dapat dilihat pada gambar 4.20. Pada gambar 4.20 setelah data diisikan dan diproses kedalam sistem, sistem akan melakukan pengecekan apakah data yang diisikan sesuai format yang diharuskan atau tidak, jika tidak sesuai maka sistem akan menjalankan *exception* pada halaman form dengan menampilkan kesalahan yang terdeteksi oleh sistem. Jika sesuai format, maka sistem akan lanjut menjalankan *query* untuk memasukkan data pada database.

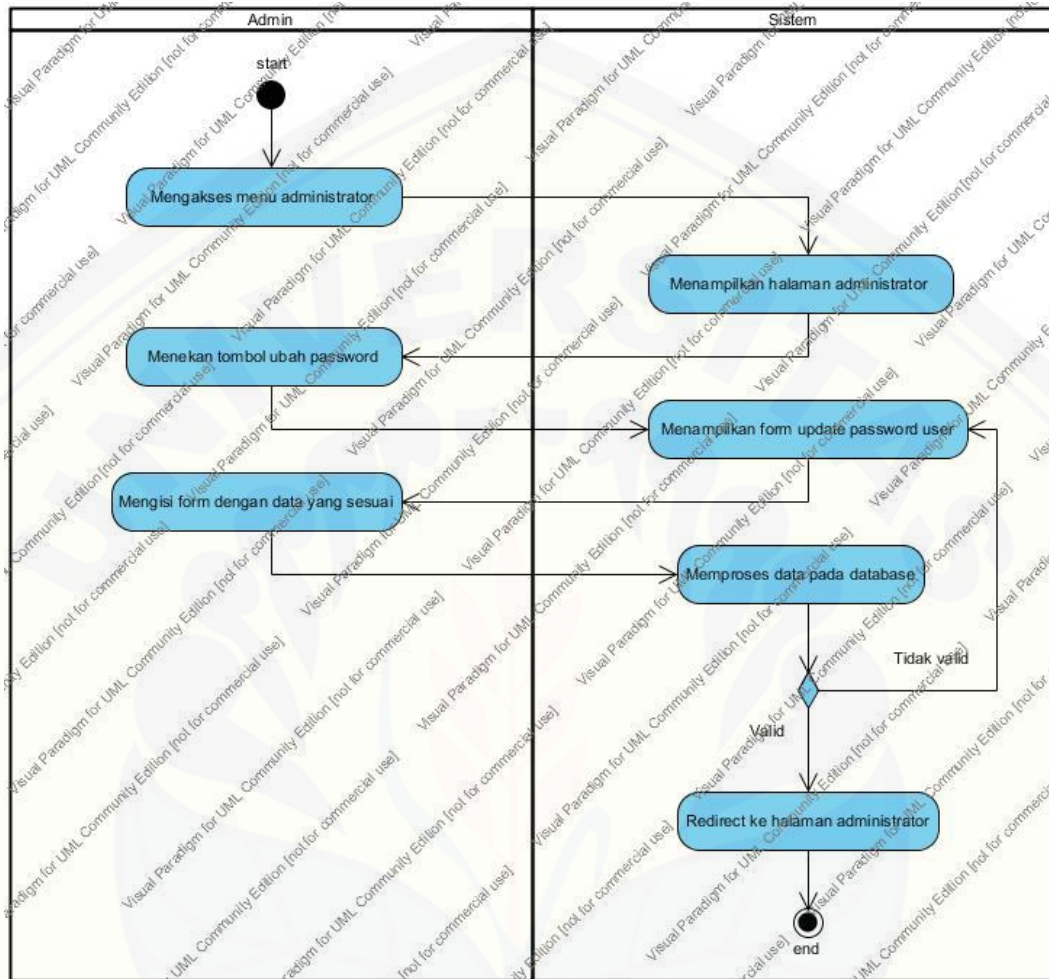
2. Manajemen Pengurus (*Update-Password*)

Activity diagram manajemen pengurus (*update-password*) dapat dilihat pada gambar 4.21. Pada gambar 4.21 setelah user menekan tombol ubah, sistem akan men-select id yang berkaitan pada database dan menampilkan value pada halaman form sehingga user dapat meng-update data dengan sesuai. Ketika kondisi *exception* terpenuhi, maka sistem akan menampilkan *exception* pada halaman sehingga user dapat kembali mengupdate data dengan benar.



Gambar 4.20 Activity Diagram Manajemen Pengurus (Entry)

(Sumber: Hasil Analisis, 2015)



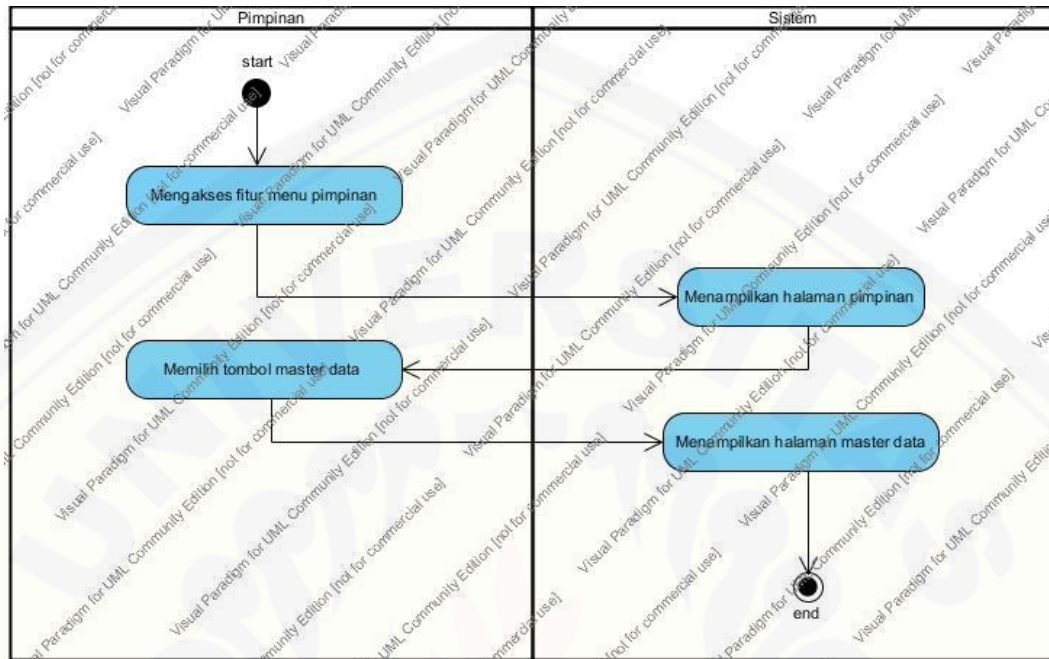
Gambar 4.21 Activity Diagram Manajemen Pengurus (*Update-Password*)

(Sumber: Hasil Analisis, 2015)

4.6.9. Perancangan Inkremen 5 Activity Master Data Transaksi

Activity diagram master data transaksi menggambarkan alir aktivitas pada monitoring data transaksi secara menyeluruh. Fitur yang dimiliki oleh pimpinan koperasi ini hanyalah untuk melihat data atau memonitoring data transaksi secara penuh tanpa dibatasi oleh waktu serta dapat disortir sesuai atribut pada tabel.

Activity diagram master data transaksi dapat dilihat pada tabel 4.22.



Gambar 4.22 *Activity Diagram* Master Data Transaksi

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.22 pimpinan koperasi hanya memilih tombol yang ingin dimonitor data transaksinya sesuai nama pada tombol yang dipilih.

4.7. *Sequence Diagram*

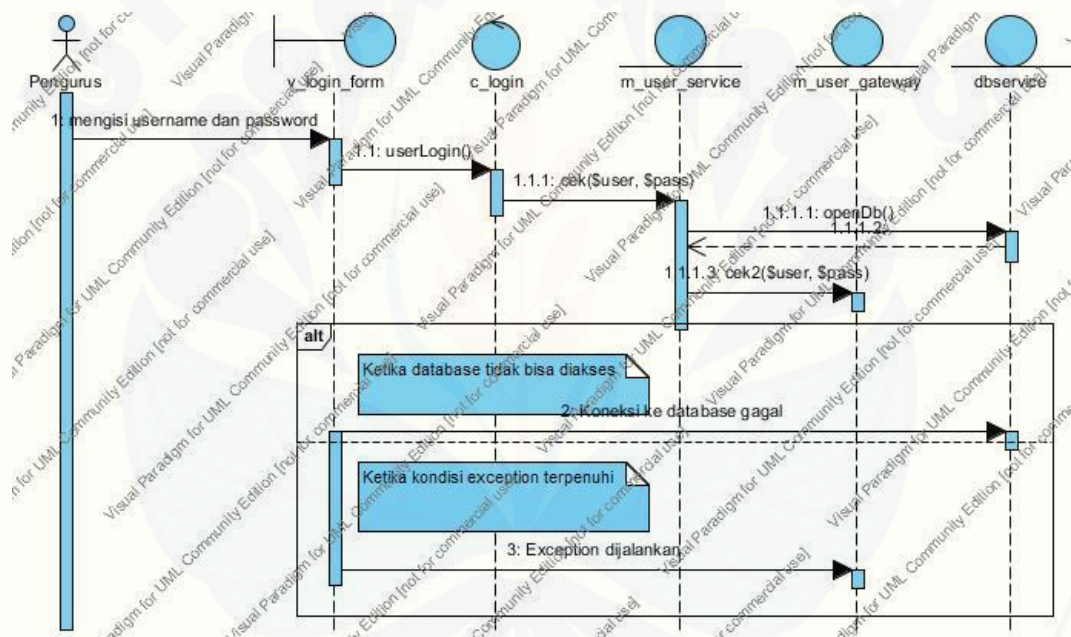
Merupakan sebuah diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi di antara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram. Berikut ini merupakan gambaran dari *activity diagram* pada pembuatan sistem informasi simpan pinjam koperasi.

4.7.1. Perancangan Inkremen 1 *Sequence* User Login

Sequence diagram user login menggambarkan interaksi antar objek pada proses *sign-in* dan *sign-out*. Fitur ini digunakan untuk akses masuk *user* ke dalam sistem dan sebagai pembagian hak akses pada sistem yang digunakan.

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, *v_login_form* sebagai *view*, *c_login* sebagai *controller*, *m_user_service* dan *m_user_gateway* sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* user login pada gambar 4.18

Sequence diagram user login (log-in) dapat dilihat pada gambar 4.23.



Gambar 4.23 *Sequence Diagram* User Login (log-In)

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.23 merupakan *sequence diagram* user login. Ketika user mengisi *username* dan *password* pada *view* *v_login_form*, maka sistem akan mengaksesnya dengan *function* *userLogin()* pada *controller* *c_login*. Kemudian isian yang sudah diambil oleh *controller* dilempar ke *model* *m_user_service* yang akan

memanggil *function* `openDb` pada `dbservice`. Setelah koneksi pada database sudah dibuat, maka *username* dan *password* akan dicek oleh *function* `cek2` pada model `m_user_gateway`. Jika kondisi *exception* terpenuhi, maka sistem akan menampilkan *message* pada view `v_login_form`. Jika *exception* tidak terpenuhi, maka sistem akan membuat *session* login yang diperlukan pada setiap *view* untuk memverifikasi user yang telah login pada sistem.

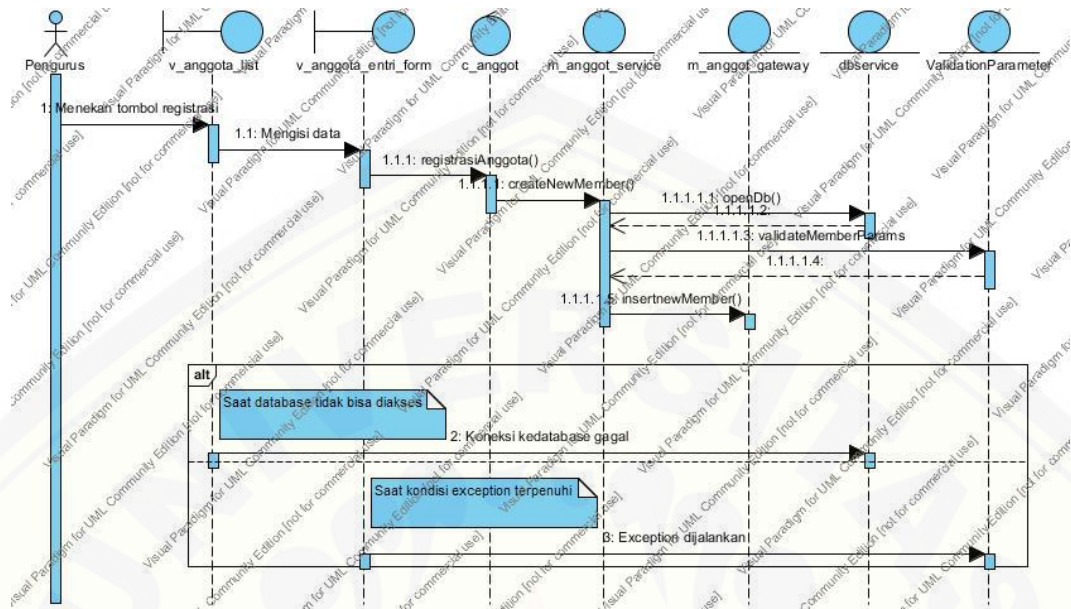
4.7.2. Perancangan Inkremen 1 *Sequence* Registrasi Anggota

Sequence diagram user login menggambarkan interaksi antar objek pada proses registrasi anggota (*entry*), registrasi anggota (*update-status*) dan registrasi anggota (*edit-update*). Fitur ini berfungsi untuk melakukan registrasi data keanggotaan baru pada koperasi.

1. Registrasi Anggota (*Entry*)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_anggota_list` dan `v_anggota_entri_form` sebagai *view*, `c_anggot` sebagai *controller*, `m_anggot_service` dan `m_anggot_gateway` sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* registrasi anggota (*entry*) pada gambar 4.24.

Sequence diagram registrasi anggota (*entry*) dapat dilihat pada gambar 4.24.



Gambar 4.24 *Sequence Diagram* Registrasi Anggota (*Entry*)

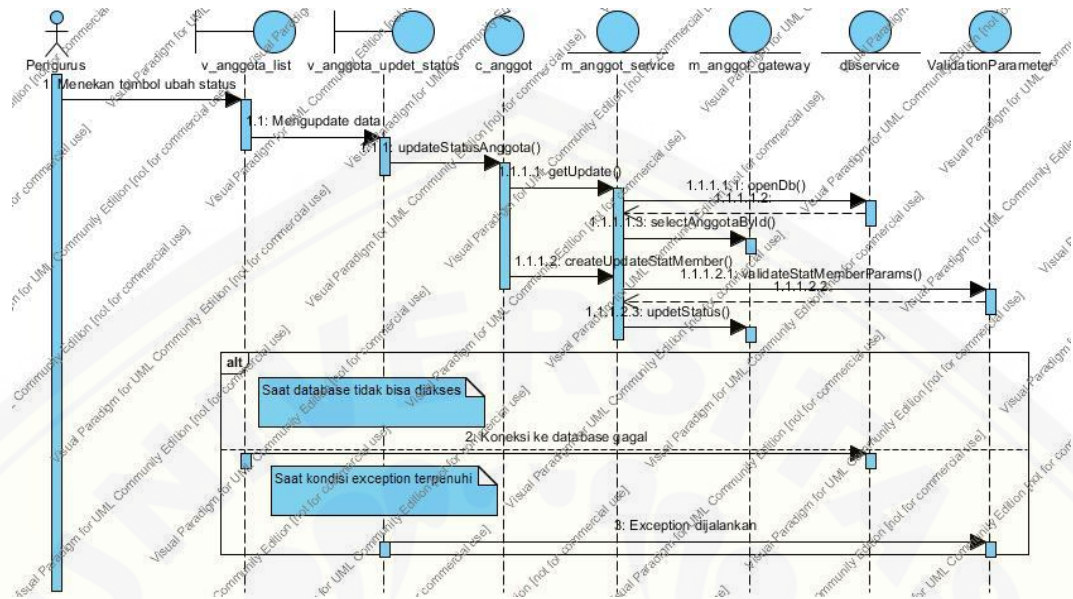
(Sumber: Hasil Analisis, 2015)

Pada gambar 4.24 merupakan *sequence diagram* registrasi anggota (*entry*). Ketika pengurus menekan tombol registrasi pada *view* *v_anggota_list*, maka sistem akan mengalihkan pada *view* *v_anggota_entr_form* sehingga pengurus dapat mengisi data pada form. Setelah mengisi data, maka sistem akan memanggil *function* *registrasiAnggota()* pada *controller* *c_anggot*. Isian yang sudah ditangkap lalu dialihkan pada *function* *createNewMember* pada *model* *m_anggot_service*. Disinilah *model* pertama bertugas untuk membuka koneksi database dengan *function* *openDb* pada *dbservice*, ketika koneksi terbuka maka isian dicek oleh *function* *validationMemberParams* pada *ValidationParameter*. Jika kondisi *exception* terpenuhi, maka sistem akan menampilkan *message* pada *view* *v_anggota_entr_form*. Jika tidak ada *exception* yang terjadi, maka sistem akan melanjutkan ke *function* *insertNewMember* pada *model* *m_anggot_gateway* untuk mengeksekusi *query* kedalam database.

2. Registrasi Anggota (*Update-Status*)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, v_anggota_list dan v_anggota_updet_status sebagai *view*, c_anggot sebagai *controller*, m_anggot_service dan m_anggot_gateway sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* registrasi anggota (*update-status*) pada gambar 4.25. *Activity diagram* registrasi anggota (*update-status*) dapat dilihat pada gambar 4.25.

Pada gambar 4.25 merupakan *sequence diagram* registrasi anggota (*update-status*). Ketika pengurus menekan tombol ubah status pada view v_anggota_list, maka secara otomatis sistem akan mengeksekusi id yang dipilih oleh pengurus dan mengalihkan halaman form pada view v_anggota_updet_status. Sebelumnya, sistem memanggil function `updateStatusAnggota()` pada controller c_anggot, kemudian dialihkan pada function `getUpdate()` pada model m_anggot_service. Pada function `getUpdate()` bertugas untuk membuka koneksi pada database dengan function `openDb()` dan memuat function `selectAnggotaById()` pada model m_anggot_service. Disinilah bagaimana sistem menampilkan sebuah form view beserta id yang telah diselek. Sehingga pengurus koperasi dapat mengubah data melalui form yang telah ditampilkan oleh sistem. Setelah pengurus mengisikan informasi dan menekan tombol update, sistem mengeksekusi function `createUpdateStatMember()` pada model m_anggot_service untuk menangkap value yang telah diisikan oleh user. Kemudian, sistem memuat function `validateStatMemberParams()` pada `ValidationParameter` untuk mengecek bahwa isian valid atau tidak. Jika kondisi exception terpenuhi, maka sistem akan menampilkan message pada halaman form edit. Dan jika kondisi exception tidak terpenuhi, maka sistem akan meneruskan eksekusi dengan memanggil function `updetStatus()` pada model m_anggot_gateway untuk mengeksekusi query pada database.



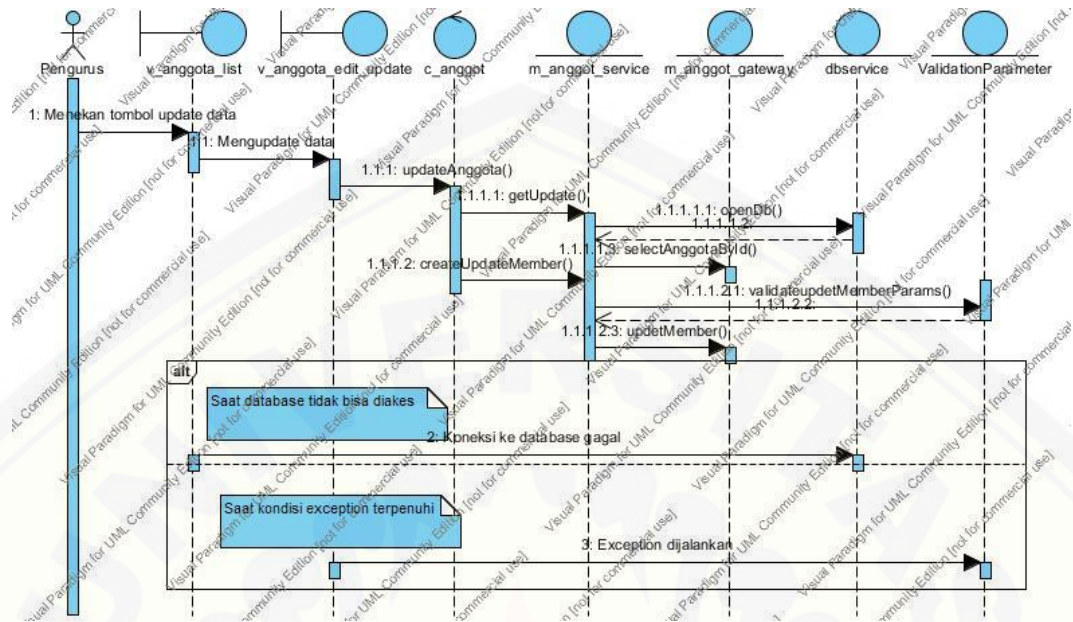
Gambar 4.25 Activity Diagram Registrasi Anggota (*Update-Status*)

(Sumber: Hasil Analisis, 2015)

3. Registrasi Anggota (*Edit-Update*)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, *v_anggota_list* dan *v_anggota_edit_update* sebagai *view*, *c_anggot* sebagai *controller*, *m_anggot_service* dan *m_anggot_gateway* sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* registrasi anggota (*edit-update*) pada gambar.

Activity diagram registrasi anggota (*edit-update*) dapat dilihat pada gambar 4.26.



Gambar 4.26 Activity Diagram Registrasi Anggota (Edit-Update)

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.26 merupakan *sequence diagram* registrasi anggota (*edit-update*). Ketika pengurus menekan tombol ubah data pada view `v_anggota_list`, maka secara otomatis sistem akan mengeksekusi id yang dipilih oleh pengurus dan mengalihkan halaman form pada view `v_anggota_edit_update`. Sebelumnya, sistem memanggil function `updateAnggota()` pada controller `c_anggot`, kemudian dialihkan pada function `getUpdate()` pada model `m_anggot_service`. Pada function `getUpdate()` bertugas untuk membuka koneksi pada database dengan function `openDb()` dan memuat function `selectAnggotaById()` pada model `m_anggot_service`. Disinilah bagaimana sistem menampilkan sebuah form view beserta id yang telah diselek. Sehingga pengurus koperasi dapat mengubah data melalui form yang telah ditampilkan oleh sistem. Setelah pengurus mengisikan informasi dan menekan tombol update, sistem mengeksekusi function `createUpdateMember()` pada model `m_anggot_service` untuk menangkap value yang telah diisikan oleh user. Kemudian, sistem memuat function `validateUpdateMemberParams()` pada `ValidationParameter`

untuk mengecek bahwa isian valid atau tidak. Jika kondisi exception terpenuhi, maka sistem akan menampilkan message pada halaman form edit. Dan jika kondisi exception tidak terpenuhi, maka sistem akan meneruskan eksekusi dengan memanggil function `updetMember()` pada model `m_anggot_gateway` untuk mengeksekusi query pada database.

4.7.3. Perancangan Inkremen 1 *Sequence* Transaksi Simpanan

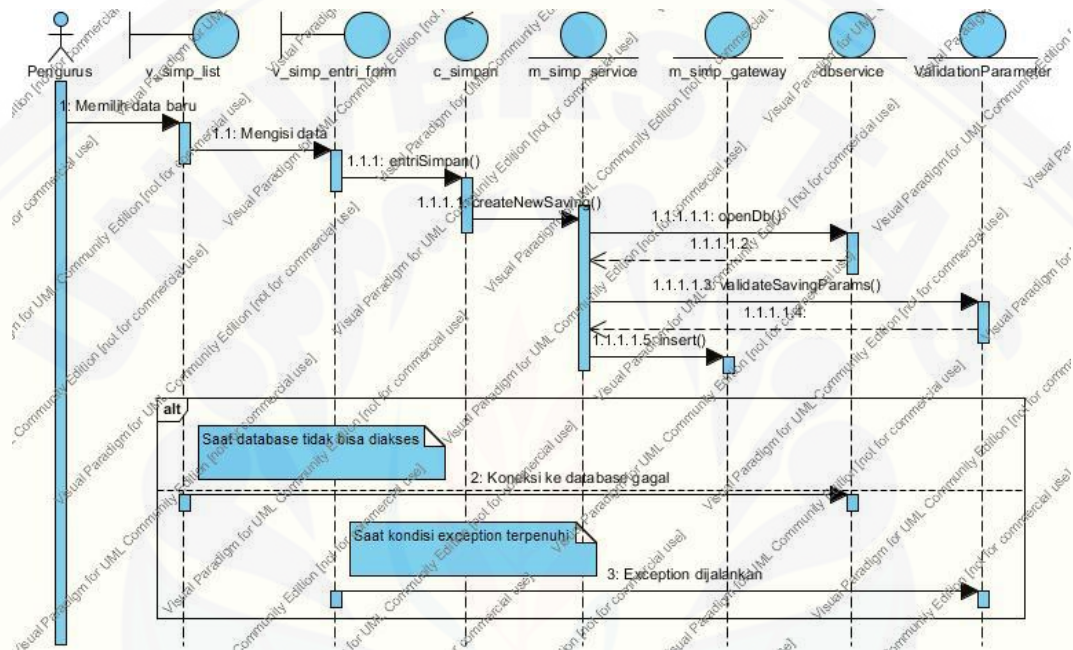
Sequence diagram transaksi simpanan anggota menggambarkan interaksi antar objek pada proses transaksi simpanan anggota (*entry*) dan transaksi simpanan anggota (*edit-update*). Fitur ini berfungsi untuk melakukan proses transaksi simpanan anggota yang berupa simpanan wajib dan simpanan sukarela.

1. Transaksi Simpanan Anggota (*Entry*)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_simp_list` dan `v_simp_entri_form` sebagai *view*, `c_simpan` sebagai *controller*, `m_simp_service` dan `m_simp_gateway` sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* simpanan anggota (*entry*) pada gambar 4.27. *Activity diagram* transaksi simpanan anggota (*entry*) dapat dilihat pada gambar 4.27.

Pada gambar 4.27 merupakan *sequence diagram* transaksi simpanan anggota (*entry*). Ketika pengurus menekan tombol data baru pada view `v_simp_list`, sistem akan mengalihkan halaman pada view `v_simp_entri_form` sehingga pengurus dapat mengisi data transaksi. Setelah pengurus selesai mengisi data dan menekan tombol entri, sistem akan memanggil function `entriSimpan()` pada controller `c_simpan`. Function ini berfungsi untuk menangkap data isian pada form, kemudian dilanjutkan dengan function `createNewSaving()` pada model `m_simp_service`. Pada model pertama ini, sistem membuka koneksi pada database dengan function `openDb()` pada `dbservice`. Ketika koneksi sudah dibuat, sistem memproses data isian form dengan function `validateSavingParams()` pada `ValidationParameter`. Disinilah

merupakan bagian untuk mengecek apakah isian valid atau tidak. Jika kondisi exception terpenuhi, maka sistem akan menampilkan message pada halaman view `v_simp_entri_form`. Jika tidak ada exception atau isian bernilai valid, maka sistem akan melanjutkan eksekusi dengan function `insert()` pada model `m_simp_gateway` untuk mengentrikan data pada database dengan menggunakan query.



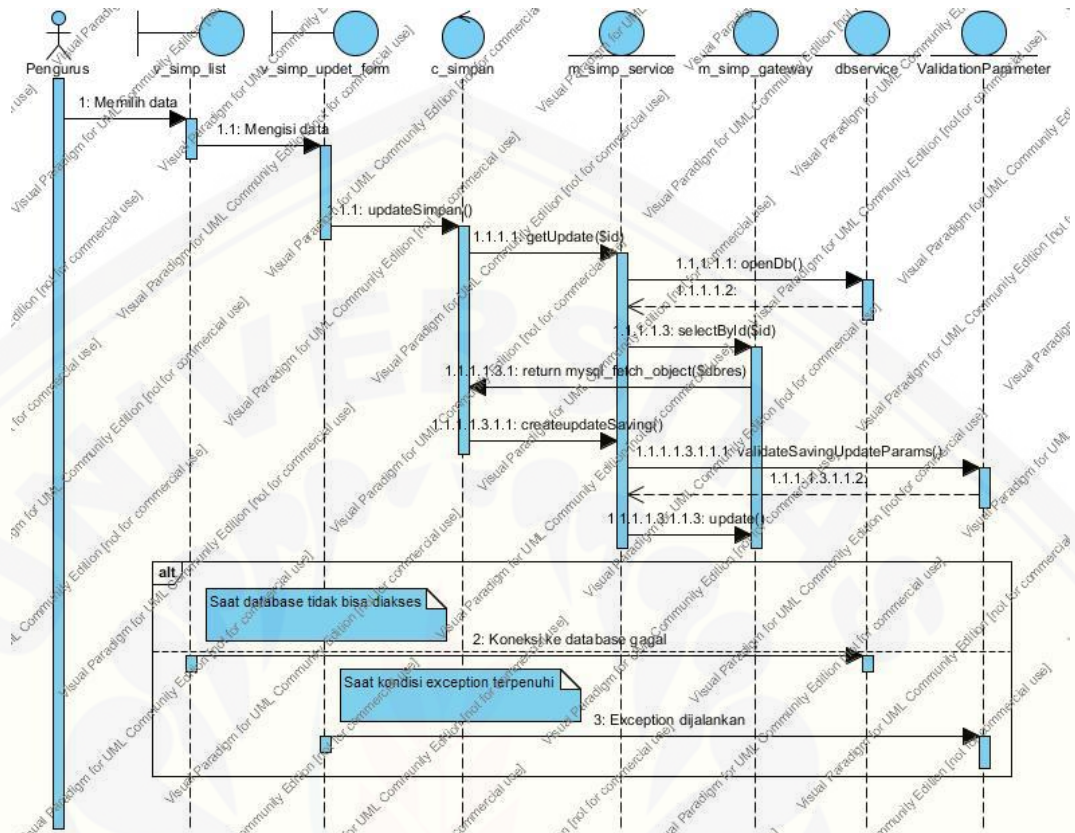
Gambar 4.27 Activity Diagram Transaksi Simpanan Anggota (Entry)

(Sumber: Hasil Analisis, 2015)

2. Transaksi Simpanan Anggota (Edit-Update)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_simp_list` dan `v_simp_updet_form` sebagai view, `c_simpan` sebagai controller, `m_simp_service` dan `m_simp_gateway` sebagai model, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* simpanan anggota (*edit-update*) pada gambar 4.28.

Activity diagram transaksi simpanan anggota (*edit-update*) dapat dilihat pada gambar 4.28.



Gambar 4.28 Activity Diagram Transaksi Simpanan (Edit-Update)

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.28 merupakan *sequence diagram* transaksi simpanan anggota (*edit-update*). Ketika pengurus memilih data pada view `v_simp_list`, maka secara otomatis sistem akan mengeksekusi id yang dipilih oleh pengurus dan mengalihkan halaman form pada view `v_simp_updet_form`. Sebelumnya, sistem memanggil function `updateSimpan()` pada controller `c_simpan`, kemudian dialihkan pada function `getUpdate()` pada model `m_simp_service`. Pada function `getUpdate()` bertugas untuk membuka koneksi pada database dengan function `openDb()` dan memuat function `selectById()` pada model `m_simp_service`. Disinilah bagaimana sistem menampilkan sebuah form view beserta id yang telah diselek. Sehingga pengurus koperasi dapat mengubah data melalui form yang telah ditampilkan oleh

sistem. Setelah pengurus mengisikan informasi dan menekan tombol update, sistem mengeksekusi function `createUpdateSaving()` pada model `m_simp_service` untuk menangkap value yang telah diisikan oleh user. Kemudian, sistem memuat function `validateSavingUpdateParams()` pada `ValidationParameter` untuk mengecek bahwa isian valid atau tidak. Jika kondisi exception terpenuhi, maka sistem akan menampilkan message pada halaman form edit. Dan jika kondisi exception tidak terpenuhi, maka sistem akan meneruskan eksekusi dengan memanggil function `update()` pada model `m_simp_gateway` untuk mengeksekusi query pada database.

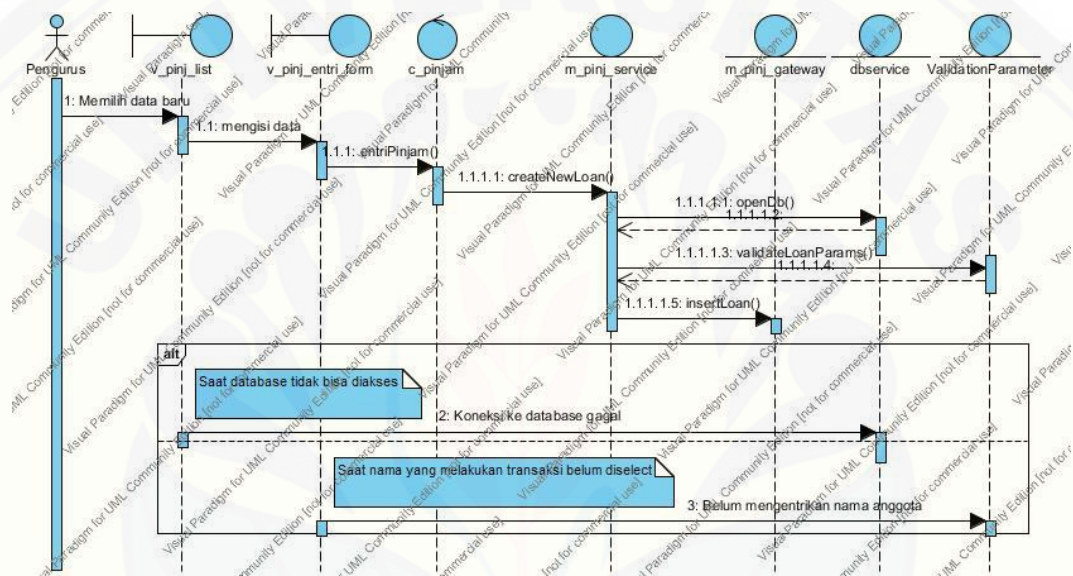
4.7.4. Perancangan Inkremen 1 *Sequence* Transaksi Pinjaman

Sequence diagram transaksi pinjaman anggota menggambarkan interaksi antar objek pada proses transaksi pinjaman berupa uang atau unit simpan pinjam (usp). Fitur ini berfungsi untuk melakukan proses transaksi pinjaman anggota yang berupa pinjaman usp ataupun pinjaman insidental.

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_pinj_list` dan `v_pinj_entri_form` sebagai *view*, `c_pinjam` sebagai *controller*, `m_pinj_service` dan `m_pinj_gateway` sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* transaksi pinjaman anggota pada gambar 4.29. *Sequence diagram* transaksi pinjaman anggota dapat dilihat pada gambar 4.29.

Pada gambar 4.29 merupakan *sequence diagram* transaksi pinjaman anggota. Ketika pengurus menekan tombol data baru pada view `v_pinj_list`, sistem akan mengalihkan halaman pada view `v_pinj_entri_form` sehingga pengurus dapat mengisikan data transaksi. Setelah pengurus selesai mengisikan data dan menekan tombol entri, sistem akan memanggil function `entriPinjam()` pada controller `c_pinjam`. Function ini berfungsi untuk menangkap data isian pada form, kemudian dilanjutkan dengan function `createNewLoan()` pada model `m_pinj_service`. Pada model pertama ini, sistem membuka koneksi pada database dengan function `openDb()` pada `dbservice`. Ketika koneksi sudah dibuat, sistem memproses data isian

form dengan function `validateLoanParams()` pada `ValidationParameter`. Disinilah merupakan bagian untuk mengecek apakah isian valid atau tidak. Jika kondisi exception terpenuhi, maka sistem akan menampilkan message pada halaman view `v_pinj_entr_form`. Jika tidak ada exception atau isian bernilai valid, maka sistem akan melanjutkan eksekusi dengan function `insertLoan()` pada model `m_pinj_gateway` untuk mengentrikan data pada database dengan menggunakan query.



Gambar 4.29 *Sequence Diagram* Transaksi Pinjaman Anggota

(Sumber: Hasil Analisis, 2015)

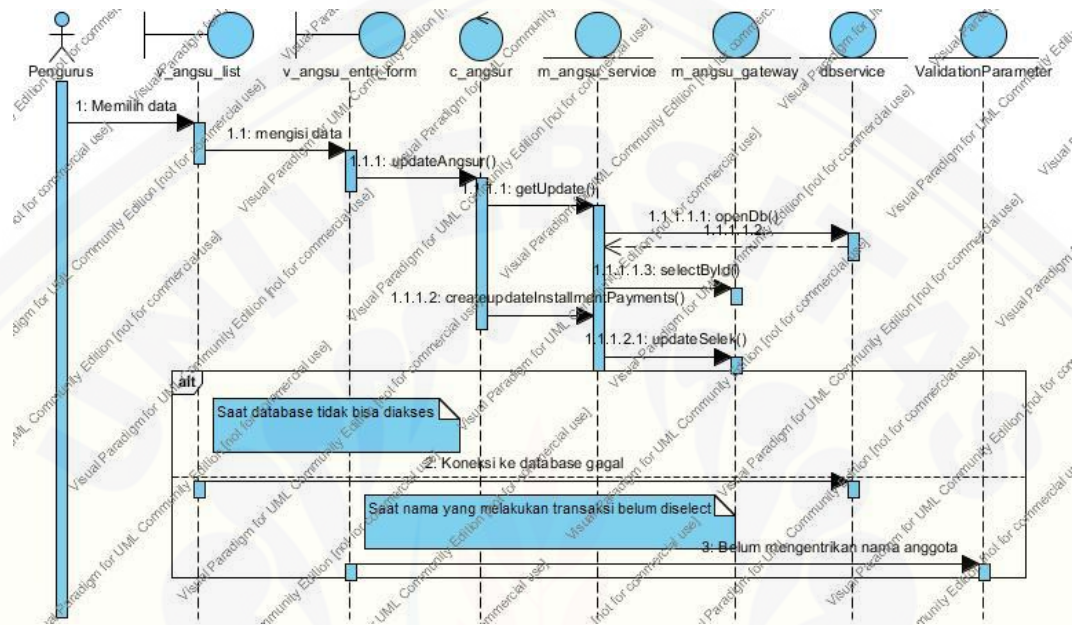
4.7.5. Perancangan Inkremen 1 *Sequence* Transaksi Angsuran

Sequence diagram transaksi angsuran anggota menggambarkan alir aktivitas pada proses transaksi angsuran terhadap pinjaman berupa barang atau jasa yang dilakukan oleh anggota koperasi.

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_angsu_list` dan `v_angsu_entr_form` sebagai *view*, `c_angsur` sebagai *controller*, `m_angsu_service` dan `m_angsu_gateway` sebagai *model*, dan fungsi –

fungsi yang digunakan dapat digambarkan dalam *sequence diagram* transaksi angsuran anggota pada gambar 4.30.

Sequence diagram transaksi angsuran anggota dapat dilihat pada gambar 4.30.



Gambar 4.30 *Sequence Diagram* Transaksi Angsuran Anggota

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.30 merupakan *sequence diagram* transaksi angsuran anggota. Ketika pengurus memilih data yang akan dilakukan transaksi angsuran pada view `v_angsu_list`, maka secara otomatis sistem akan mengeksekusi id yang dipilih oleh pengurus dan mengalihkan halaman form pada view `v_angsu_entri_form`. Sebelumnya, sistem memanggil function `updateAngsur()` pada controller `c_angsur`, kemudian dialihkan pada function `getUpdate()` pada model `m_angsu_service`. Pada function `getUpdate()` bertugas untuk membuka koneksi pada database dengan function `openDb()` dan memuat function `selectById()` pada model `m_simp_service`. Disinilah bagaimana sistem menampilkan sebuah form view beserta id yang telah diselek. Sehingga pengurus koperasi dapat mengubah data melalui form yang telah ditampilkan oleh sistem. Setelah pengurus mengisi informasi dan menekan

tombol update, sistem mengeksekusi function `createUpdateInstalmentPayments()` pada model `m_angsu_service` untuk menangkap value yang telah diisikan oleh user. Kemudian, sistem akan meneruskan eksekusi dengan memanggil function `updateSelek()` pada model `m_angsu_gateway` untuk mengeksekusi query pada database.

4.7.6. Perancangan Inkremen 2 *Sequence* Manajemen Lembaga

Sequence diagram manajemen lembaga menggambarkan interaksi antar objek pada proses manajemen lembaga (*entry*), manajemen lembaga (*edit-update*) dan manajemen lembaga (*delete*). Fitur ini berfungsi mendaftarkan dan memajemen informasi data terkait lembaga – lembaga yang berada dan terdaftar pada lingkup koperasi.

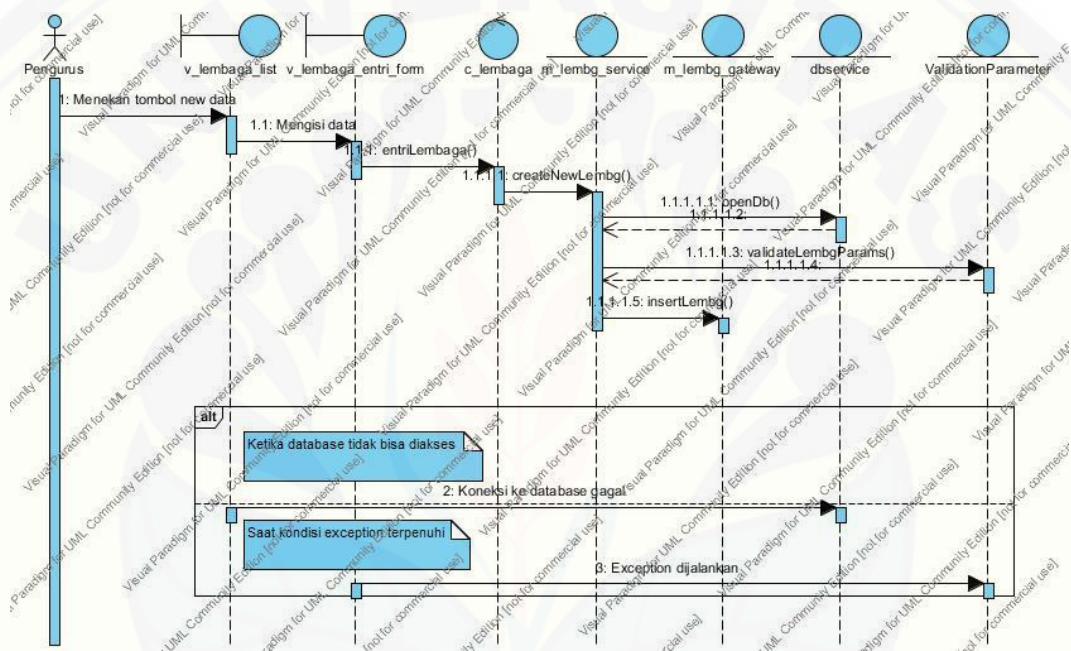
1. Manajemen Lembaga (*Entry*)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_lembaga_list` dan `v_lembaga_entri_form` sebagai *view*, `c_lembaga` sebagai *controller*, `m_lembg_service` dan `m_lembg_gateway` sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* manajemen lembaga (*entry*) pada gambar 4.31.

Pada gambar 4.31 merupakan *sequence diagram* manajemen lembaga (*entry*). Ketika pengurus menekan tombol data baru pada view `v_lembaga_list`, sistem akan mengalihkan halaman pada view `v_lembaga_entri_form` sehingga pengurus dapat mengisikan data transaksi. Setelah pengurus selesai mengisikan data dan menekan tombol entri, sistem akan memanggil function `entriLembaga()` pada controller `c_lembaga`. Function ini berfungsi untuk menangkap data isian pada form, kemudian dilanjutkan dengan function `createNewLembg()` pada model `m_lembg_service`. Pada model pertama ini, sistem membuka koneksi pada database dengan function `openDb()` pada `dbservice`. Ketika koneksi sudah dibuat, sistem memproses data isian form dengan function `validateLembgParams()` pada `ValidationParameter`. Disinilah

merupakan bagian untuk mengecek apakah isian valid atau tidak. Jika kondisi exception terpenuhi, maka sistem akan menampilkan message pada halaman view `v_lembaga_entri_form`. Jika tidak ada exception atau isian bernilai valid, maka sistem akan melanjutkan eksekusi dengan function `insertLembg()` pada model `m_lembg_gateway` untuk mengentrikan data pada database dengan menggunakan query

Sequence diagram manajemen lembaga (*entry*) dapat dilihat pada gambar 4.31.

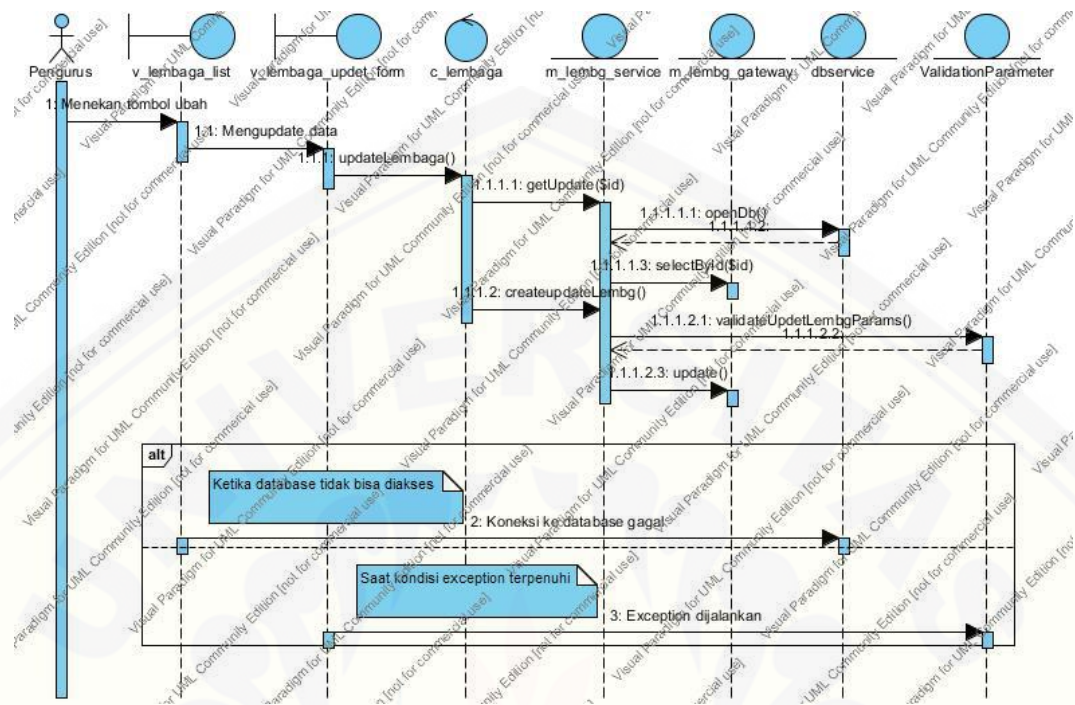


Gambar 4.31 *Sequence Diagram* Manajemen Lembaga (*Entry*)

(Sumber: Hasil Analisis, 2015)

2. Manajemen Lembaga (*Edit-Update*)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_lembaga_list` dan `v_lembaga_updet_form` sebagai *view*, `c_lembaga` sebagai *controller*, `m_lembg_service` dan `m_lembg_gateway` sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* manajemen lembaga (*edit-update*) pada gambar 4.32. *Sequence diagram* manajemen lembaga (*edit-update*) dapat dilihat pada gambar 4.32.



Gambar 4.32 *Sequence Diagram* Manajemen Lembaga (*Edit-Update*)

(Sumber: Hasil Analisis, 2015)

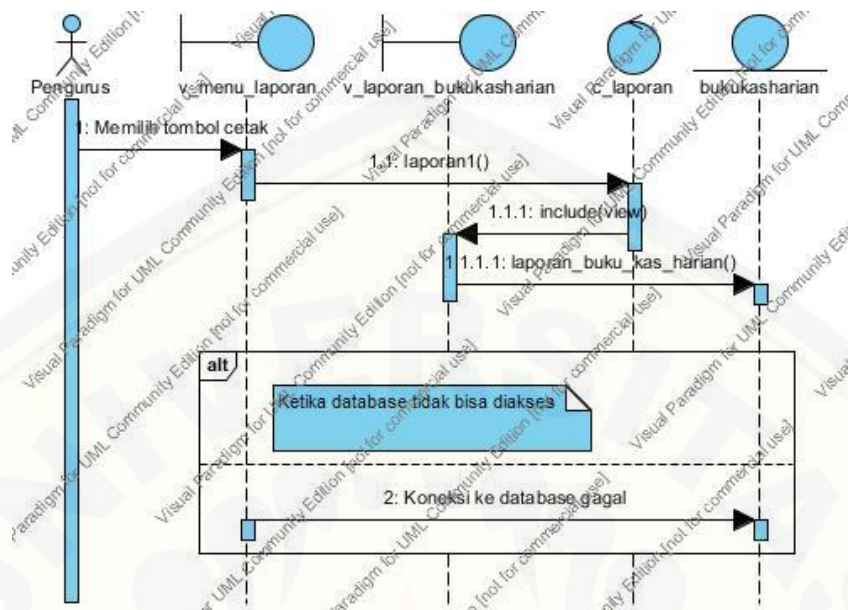
Pada gambar 4.32 merupakan *sequence diagram* manajemen lembaga (*edit-update*). Ketika pengurus memilih data pada view `v_lembaga_list`, maka secara otomatis sistem akan mengeksekusi id yang dipilih oleh pengurus dan mengalihkan halaman form pada view `v_lembaga_updet_form`. Sebelumnya, sistem memanggil function `updatelembaga()` pada controller `c_lembaga`, kemudian dialihkan pada function `getUpdate()` pada model `m_lembg_service`. Pada function `getUpdate()` bertugas untuk membuka koneksi pada database dengan function `openDb()` dan memuat function `selectById()` pada model `m_lembg_service`. Disinilah bagaimana sistem menampilkan sebuah form view beserta id yang telah diselek. Sehingga pengurus koperasi dapat mengubah data melalui form yang telah ditampilkan oleh sistem. Setelah pengurus mengisikan informasi dan menekan tombol update, sistem mengeksekusi function `createUpdateLembg()` pada model `m_lembg_service` untuk menangkap value yang telah diisikan oleh user. Kemudian, sistem memuat function

`validateUpdetLembgParams()` pada `ValidationParameter` untuk mengecek bahwa isian valid atau tidak. Jika kondisi `exception` terpenuhi, maka sistem akan menampilkan message pada halaman form edit. Dan jika kondisi `exception` tidak terpenuhi, maka sistem akan meneruskan eksekusi dengan memanggil function `update()` pada model `m_lembg_gateway` untuk mengeksekusi query pada database.

4.7.7. Perancangan Inkremen 3 *Sequence* Cetak Laporan

Sequence diagram cetak laporan transaksi menggambarkan alir aktivitas pada pemrosesan data laporan. Fitur ini berfungsi untuk mencetak dan memproses laporan transaksi yang berupa daftar tagihan, buku kas harian, rekap daftar tagihan, bukti kas masuk dan bukti kas keluar. *Sequence diagram* cetak laporan transaksi dapat dilihat pada gambar 4.29.

Pada gambar 4.33 merupakan *sequence diagram* cetak laporan transaksi. Ketika pengurus memilih tombol cetak laporan yang diinginkan pada view `v_menu_laporan`, maka secara otomatis sistem akan mengeksekusi function `laporan1()` atau `laporan2()` dan seterusnya pada controller `c_laporan`. Pada controller, akan dipanggil view untuk memberikan akses seleksi berdasarkan lembaga, bulan transaksi dan tahun transaksi untuk laporan yang akan diproses dengan men-include kan pada function. Setelah pengurus menseleksi berdasarkan lembaga, bulan transaksi dan tahun transaksi. Sistem mengeksekusi function `laporan_dafttagih()`, `laporan_buku_kas_harian()`, `laporan_rekap_tagihan()`, `laporan_bkm()` atau `laporan_bkk()` untuk memproses laporan pada model `dafttagih`, `bukukasharian`, `rekapdafttagih`, `bkm` atau `bkk` pada halaman yang telah di-include sebelumnya.



Gambar 4.33 *Sequence Diagram* Cetak Laporan Transaksi

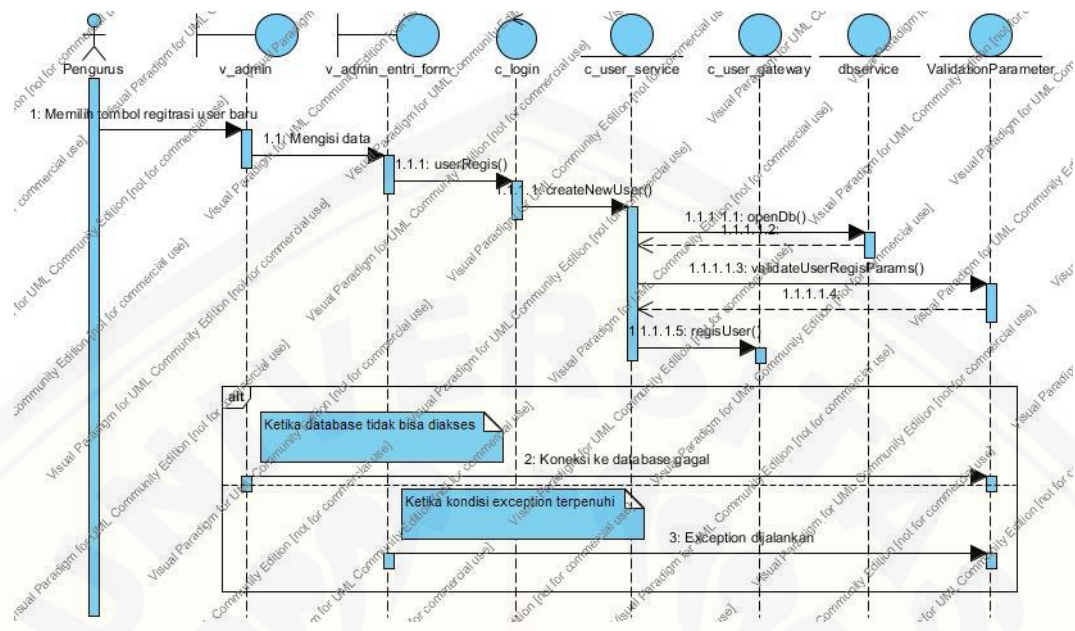
(Sumber: Hasil Analisis, 2015)

4.7.8. Perancangan Inkremen 4 *Sequence* Manajemen Pengurus

Sequence diagram manajemen pengurus menggambarkan interaksi antar objek pada proses manajemen pengurus (*entry*) dan manajemen pengurus (*update-password*). Fitur administrator ini berfungsi untuk mendaftarkan dan mengubah *password user* yang bertanggung jawab atas akses terhadap sistem.

1. Manajemen Pengurus (*Entry*)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, *v_admin_list* dan *v_admin_entri_form* sebagai *view*, *c_login* sebagai *controller*, *m_user_service* dan *m_user_gateway* sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* manajemen pengurus (*entry*) pada gambar 4.34. *Sequence diagram* manajemen pengurus (*entry*) dapat dilihat pada gambar 4.34.



Gambar 4.34 *Sequence Diagram* Manajemen Pengurus (*Entry*)

(Sumber: Hasil Analisis, 2015)

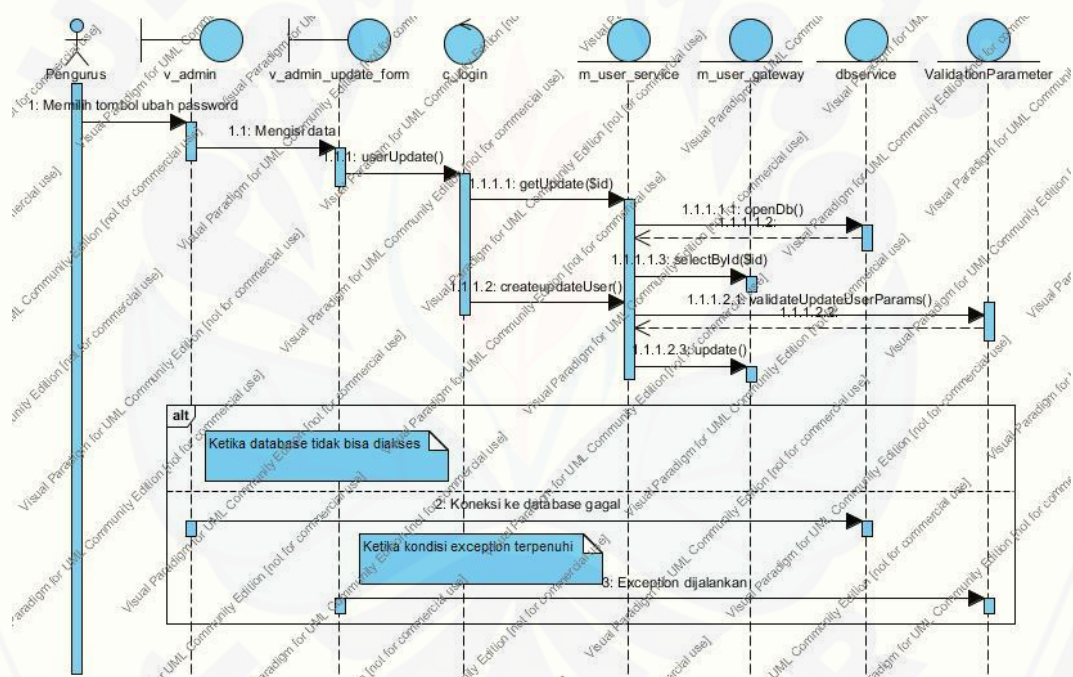
Pada gambar 4.34 merupakan *sequence diagram* manajemen pengurus (*entry*). Ketika pengurus menekan tombol data baru pada view `v_admin`, sistem akan mengalihkan halaman pada view `v_admin_entriform` sehingga pengurus dapat mengisi data transaksi. Setelah pengurus selesai mengisi data dan menekan tombol entri, sistem akan memanggil function `userRegis()` pada controller `c_login`. Function ini berfungsi untuk menangkap data isian pada form, kemudian dilanjutkan dengan function `createNewUser()` pada model `m_user_service`. Pada model pertama ini, sistem membuka koneksi pada database dengan function `openDb()` pada `dbservice`. Ketika koneksi sudah dibuat, sistem memproses data isian form dengan function `validateUserRegisParams()` pada `ValidationParameter`. Disinilah merupakan bagian untuk mengecek apakah isian valid atau tidak. Jika kondisi exception terpenuhi, maka sistem akan menampilkan message pada halaman view `v_admin_entriform`. Jika tidak ada exception atau isian bernilai valid, maka sistem

akan melanjutkan eksekusi dengan function `regisUser ()` pada model `m_user_gateway` untuk mengentrikan data pada database dengan menggunakan query.

2. Manajemen Pengurus (*Update-Password*)

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_admin` dan `v_admin_update_form` sebagai *view*, `c_login` sebagai *controller*, `m_user_service` dan `m_user_gateway` sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* manajemen pengurus (*update-password*) pada gambar 4.35.

Sequence diagram manajemen pengurus (*update-password*) dapat dilihat pada gambar 4.35.



Gambar 4.35 *Sequence Diagram* Manajemen Pengurus (*Update-Password*)

(Sumber: Hasil Analisis, 2015)

Pada gambar 4.35 merupakan *sequence diagram* manajemen pengurus (*update-password*). Ketika pengurus memilih data pada view `v_admin`, maka secara

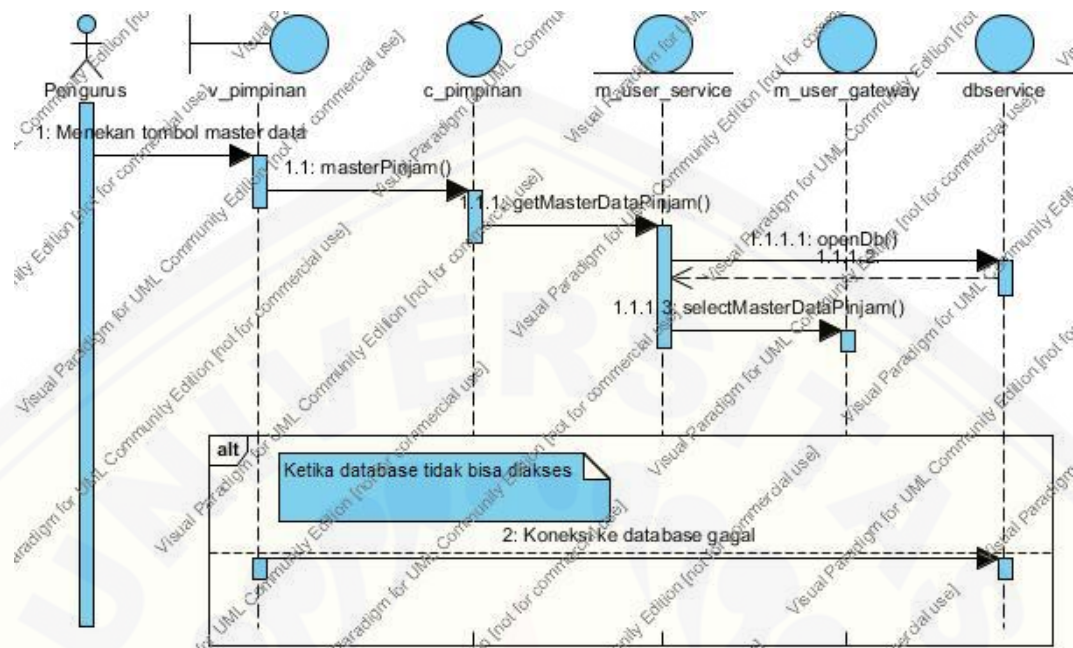
otomatis sistem akan mengeksekusi id yang dipilih oleh pengurus dan mengalihkan halaman form pada view `v_admin_update_form`. Sebelumnya, sistem memanggil function `userUpdate()` pada controller `c_login`, kemudian dialihkan pada function `getUpdate()` pada model `m_user_service`. Pada function `getUpdate()` bertugas untuk membuka koneksi pada database dengan function `openDb()` dan memuat function `selectById()` pada model `m_user_service`. Disinilah bagaimana sistem menampilkan sebuah form view beserta id yang telah diselek. Sehingga pengurus koperasi dapat mengubah data melalui form yang telah ditampilkan oleh sistem. Setelah pengurus mengisi informasi dan menekan tombol update, sistem mengeksekusi function `createupdateUser()` pada model `m_user_service` untuk menangkap value yang telah diisikan oleh user. Kemudian, sistem memuat function `validateUpdateUserParams()` pada `ValidationParameter` untuk mengecek bahwa isian valid atau tidak. Jika kondisi exception terpenuhi, maka sistem akan menampilkan message pada halaman form edit. Dan jika kondisi exception tidak terpenuhi, maka sistem akan meneruskan eksekusi dengan memanggil function `update()` pada model `m_user_gateway` untuk mengeksekusi query pada database.

4.7.9. Perancangan Inkremen 5 *Sequence* Master Data Transaksi

Sequence diagram master data transaksi menggambarkan interaksi antar objek pada proses monitoring data transaksi secara menyeluruh. Fitur yang dimiliki oleh pimpinan koperasi ini hanyalah untuk melihat data atau memonitoring data transaksi secara penuh tanpa dibatasi oleh waktu serta dapat disortir sesuai atribut pada tabel.

Penggambaran interaksi antar objek yang meliputi pengurus koperasi sebagai aktor, `v_pimpinan` sebagai *view*, `c_pimpinan` sebagai *controller*, `m_user_service` dan `m_user_gateway` sebagai *model*, dan fungsi – fungsi yang digunakan dapat digambarkan dalam *sequence diagram* master data transaksi pada gambar 4.36.

Sequence diagram master data transaksi dapat dilihat pada gambar 4.36.



Gambar 4.36 *Sequence Diagram* Master Data Transaksi

(Sumber: Hasil Analisis, 2015)

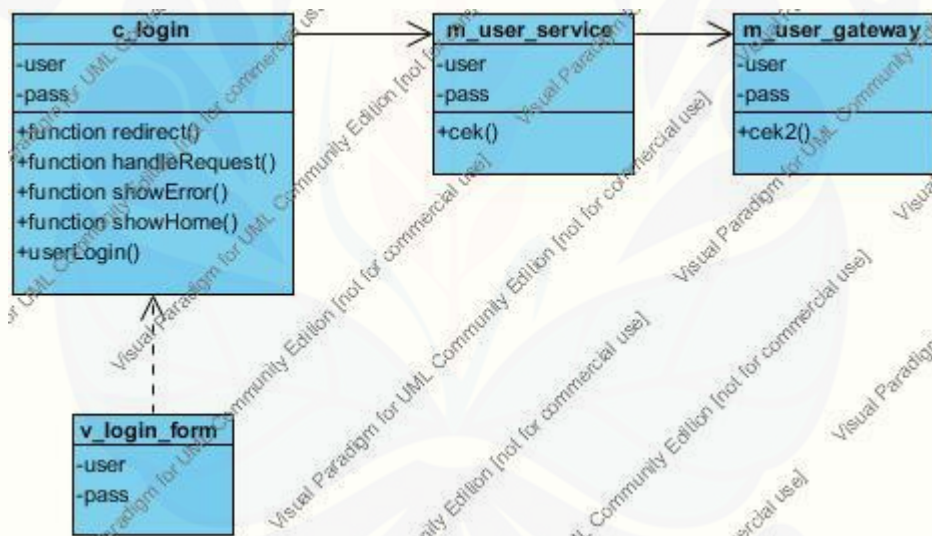
Pada gambar 4.36 merupakan *sequence diagram* registrasi anggota (*update-status*). Ketika pengurus menekan tombol master data simpanan anggota atau master data pinjaman anggota ataupun master data anggota pada view *v_pimpinan*, maka sistem akan memanggil function *masterPinjam()* pada controller *c_pimpinan*. Dimana sistem akan melanjutkan untuk memanggil function *getMasterDataSimpanan()* atau *getMasterDataPinjaman()* ataupun *getMasterDataAnggota* pada model *m_user_service*. Pada model *m_user_service* sistem membuat koneksi pada database dengan function *openDb()* pada *dbservice*. Jika autentikasi database benar maka database diperbolehkan untuk diakses tanpa adanya exception dan sistem akan melanjutkan untuk memanggil function *selectMasterDataPinjam()* atau *selectMasterDataSimpan()* ataupun *selectMasterDataAnggota* pada model *m_user_service*, sehingga data yang diinginkan dapat ditampilkan pada halaman web.

4.8. Class Diagram

Merupakan hubungan antar kelas yang saling terkait dan digunakan untuk membangun sebuah sistem. Fitur – fitur pada sistem informasi simpan pinjam ini terdapat beberapa *class diagram*. *Class - class* tersebut kemudian digabungkan berdasarkan karakteristik sehingga menjadi sebuah *class diagram*.

4.8.1. Perancangan Inkremen 1 *Class Diagram* User Login

Class diagram user login menggambarkan kelas - kelas pada proses *sign-in* dan *sign-out*. Fitur ini digunakan untuk akses masuk *user* ke dalam sistem dan sebagai pembagian hak akses pada sistem yang digunakan. *Class diagram* user login dapat dilihat pada gambar 4.37.



Gambar 4.37 *Class Diagram* User Login

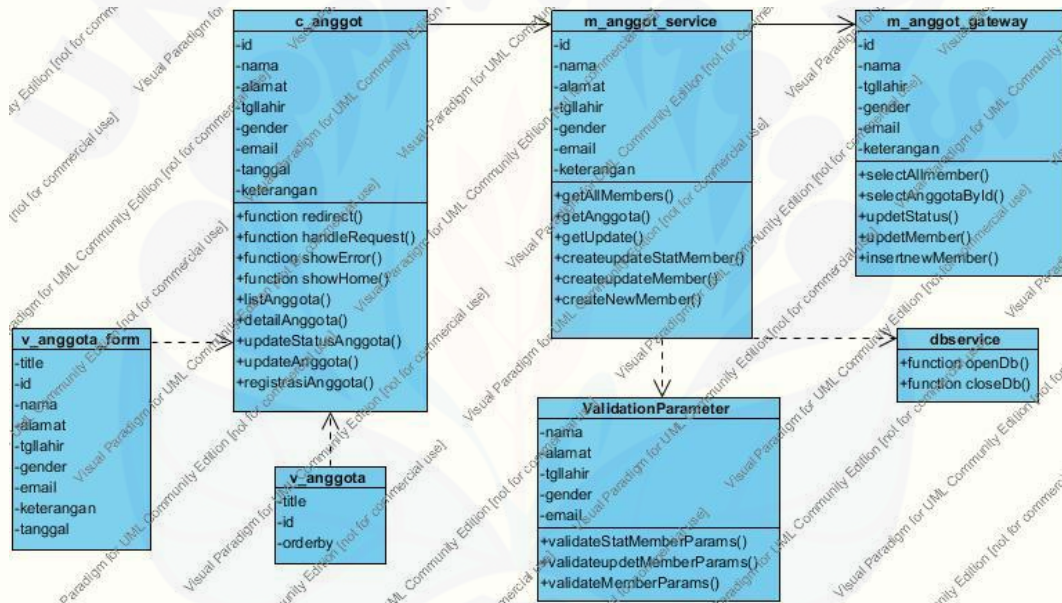
(Sumber: Hasil Analisis, 2015)

Class diagram yang sudah dijelaskan pada gambar 4.37 terdapat `c_login` sebagai controller, `v_login_form` sebagai view, `m_user_service` dan `m_user_gateway` sebagai model. Ada 2 macam hubungan pada class diagram ini, adalah *dependency* pada relasi controller dan view yaitu kelas yang memiliki ketergantungan pada kelas

lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.

4.8.2. Perancangan Inkremen 1 *Class Diagram* Registrasi Anggota

Class diagram registrasi anggota menggambarkan kelas - kelas pada proses registrasi anggota (*entry*), registrasi anggota (*update-status*) dan registrasi anggota (*edit-update*). Fitur ini berfungsi untuk melakukan registrasi data keanggotaan baru pada koperasi. *Class diagram* registrasi anggota dapat dilihat pada gambar 4.38.



Gambar 4.38 *Class Diagram* Registrasi Anggota

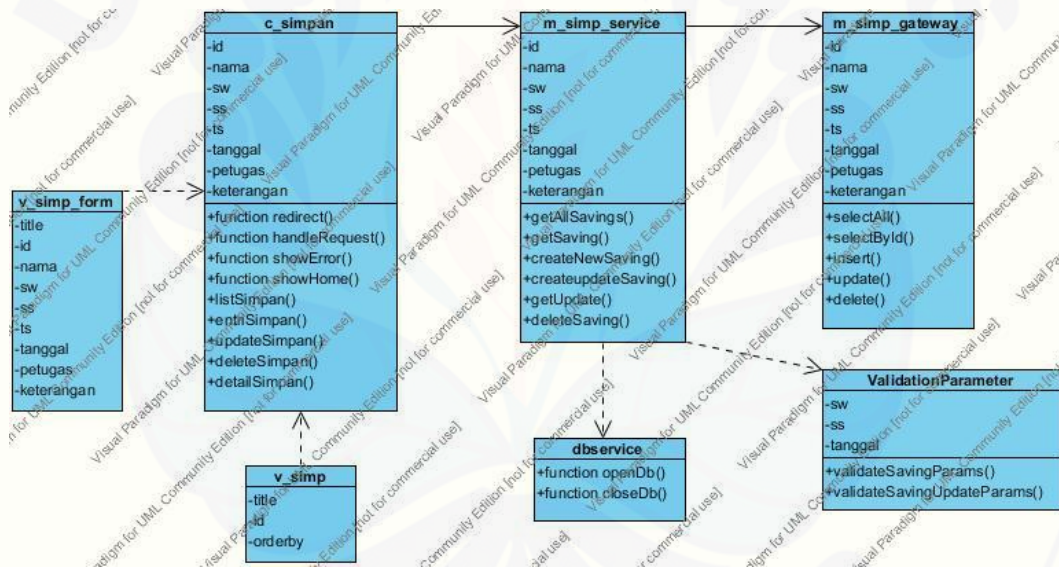
(Sumber: Hasil Analisis, 2015)

Class diagram yang sudah dijelaskan pada gambar 4.38 terdapat *c_anggot* sebagai controller, *v_anggota_form* dan *v_anggota* sebagai view, *m_anggot_service* dan *m_anggot_gateway* sebagai model, *validationparameter* sebagai pengecekan input dan *dbservice* sebagai koneksi database. Ada 2 macam hubungan pada class diagram ini, adalah *dependency* pada relasi controller dan view atau antara model

pada validationparameter dan dbservice. Hubungan *dependency* yaitu kelas yang memiliki ketergantungan pada kelas lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.

4.8.3. Perancangan Inkremen 1 Class Diagram Transaksi Simpanan

Class diagram transaksi simpanan anggota menggambarkan alir aktivitas pada proses transaksi simpanan anggota (*entry*) dan transaksi simpanan anggota (*edit-update*). Fitur ini berfungsi untuk melakukan proses transaksi simpanan anggota yang berupa simpanan wajib dan simpanan sukarela. *Class diagram* transaksi simpanan anggota dapat dilihat pada gambar 4.39.



Gambar 4.39 *Class Diagram* Transaksi Simpanan Anggota

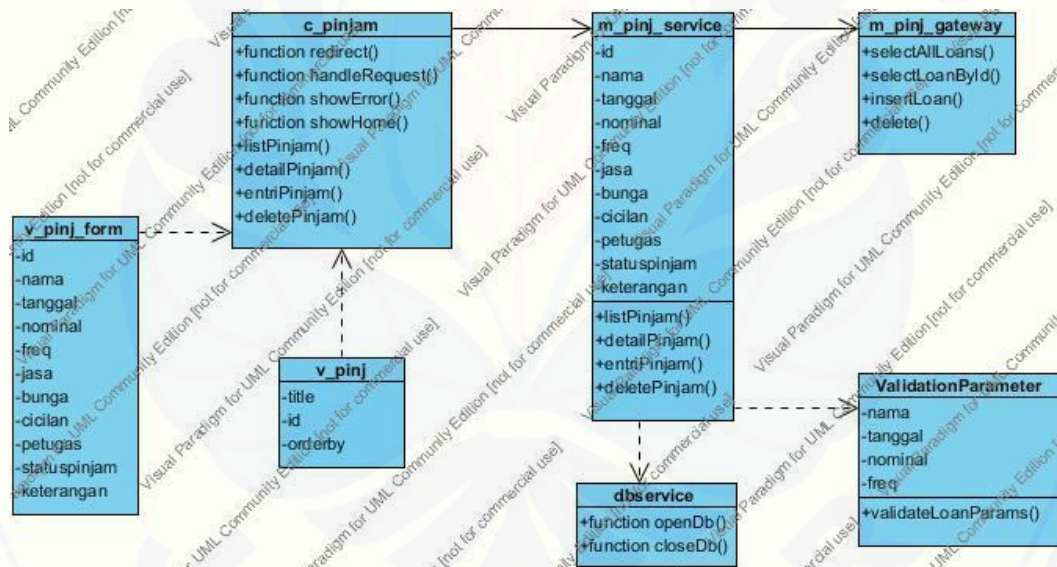
(Sumber: Hasil Analisis, 2015)

Class diagram yang sudah dijelaskan pada gambar 4.39 terdapat *c_simpan* sebagai controller, *v_simp_form* dan *v_simp* sebagai view, *m_simp_service* dan *m_simp_gateway* sebagai model, *validationparameter* sebagai pengecekan input dan *dbservice* sebagai koneksi database. Ada 2 macam hubungan pada class diagram ini,

adalah *dependency* pada relasi controller dan view atau antara model pada validationparameter dan dbservice. Hubungan *dependency* yaitu kelas yang memiliki ketergantungan pada kelas lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.

4.8.4. Perancangan Inkremen 1 *Class Diagram* Transaksi Pinjaman

Class diagram transaksi pinjaman anggota menggambarkan alir aktivitas pada proses transaksi pinjaman berupa uang atau unit simpan pinjam (usp). Fitur ini berfungsi untuk melakukan proses transaksi pinjaman anggota yang berupa pinjaman usp ataupun pinjaman insidental. *Class diagram* transaksi pinjaman anggota dapat dilihat pada gambar 4.40.



Gambar 4.40 *Class Diagram* Transaksi Pinjaman Anggota

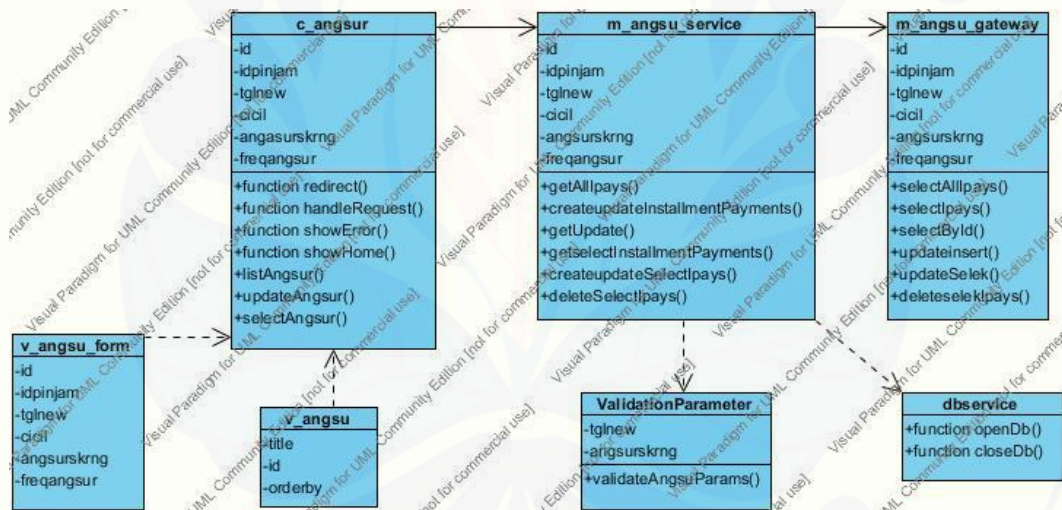
(Sumber: Hasil Analisis, 2015)

Class diagram yang sudah dijelaskan pada gambar 4.40 terdapat c_pinjam sebagai controller, v_pinj_form dan v_pinj sebagai view, m_pinj_service dan m_pinj_gateway sebagai model, validationparameter sebagai pengecekan input dan

dbservice sebagai koneksi database. Ada 2 macam hubungan pada class diagram ini, adalah *dependency* pada relasi controller dan view atau antara model pada validationparameter dan dbservice. Hubungan *dependency* yaitu kelas yang memiliki ketergantungan pada kelas lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.

4.8.5. Perancangan Inkremen 1 *Class Diagram* Transaksi Angsuran

Class diagram transaksi angsuran anggota menggambarkan alir aktivitas pada proses transaksi angsuran terhadap pinjaman berupa barang atau jasa yang dilakukan oleh anggota koperasi. *Class diagram* transaksi angsuran anggota dapat dilihat pada gambar 4.41.



Gambar 4.41 *Class Diagram* Transaksi Angsuran Anggota

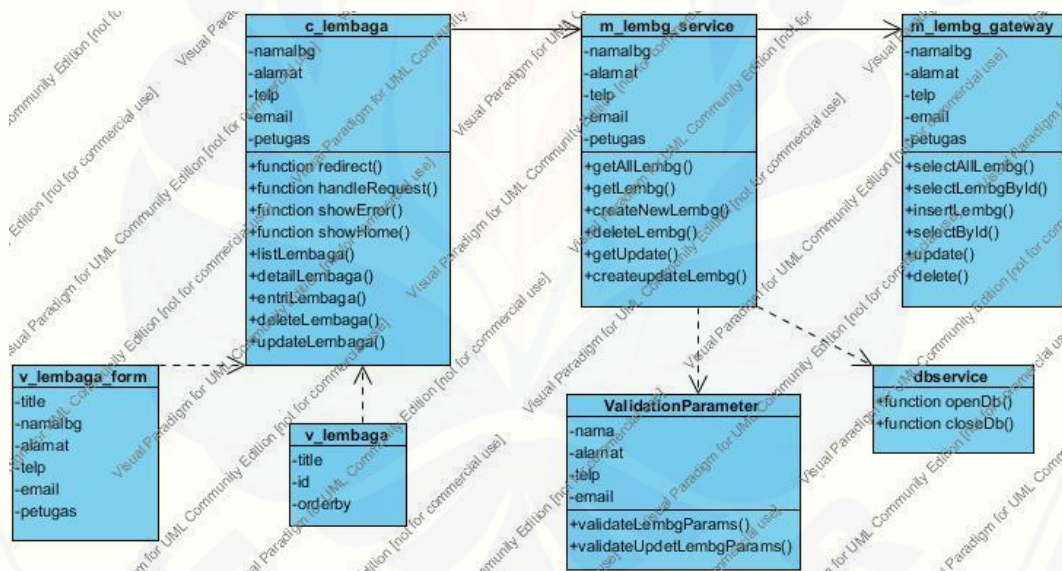
(Sumber: Hasil Analisis, 2015)

Class diagram yang sudah dijelaskan pada gambar 4.41 terdapat c_angsur sebagai controller, v_angsu_form dan v_angsu sebagai view, m_angsu_service dan m_angsu_gateway sebagai model, validationparameter sebagai pengecekan input dan dbservice sebagai koneksi database. Ada 2 macam hubungan pada class diagram ini,

adalah *dependency* pada relasi controller dan view atau antara model pada validationparameter dan dbservice. Hubungan *dependency* yaitu kelas yang memiliki ketergantungan pada kelas lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.

4.8.6. Perancangan Inkremen 2 *Class Diagram* Manajemen Lembaga

Class diagram manajemen lembaga menggambarkan alir aktivitas pada proses manajemen lembaga (*entry*), manajemen lembaga (*edit-update*) dan manajemen lembaga (*delete*). Fitur ini berfungsi mendaftarkan dan memanajemen informasi data terkait lembaga – lembaga yang berada dan terdaftar pada lingkup koperasi. *Class diagram* manajemen lembaga dapat dilihat pada gambar 4.42.



Gambar 4.42 *Class Diagram* Manajemen Lembaga

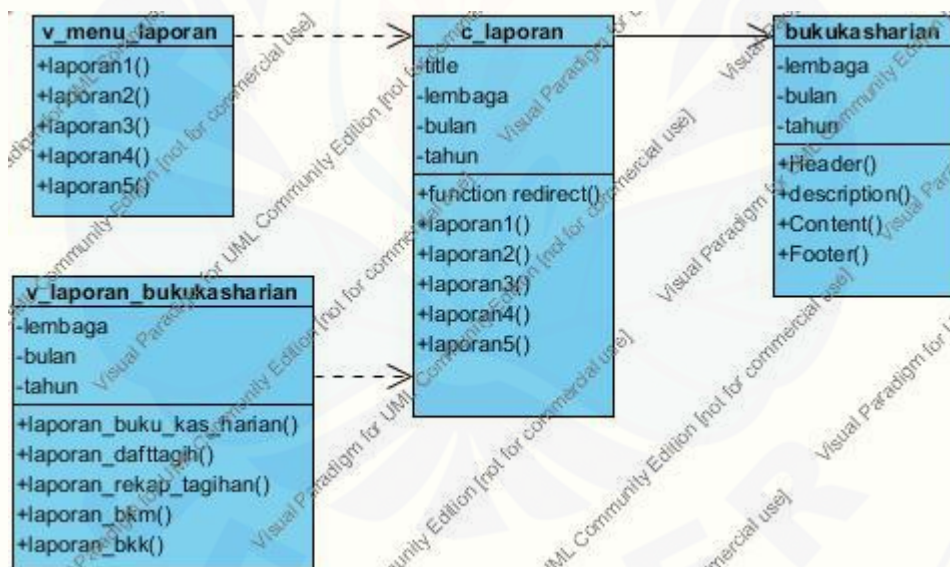
(Sumber: Hasil Analisis, 2015)

Class diagram yang sudah dijelaskan pada gambar 4.42 terdapat *c_lembaga* sebagai controller, *v_lembaga_form* dan *v_lembaga* sebagai view, *m_lembg_service* dan *m_lembg_gateway* sebagai model, *validationparameter* sebagai pengecekan input

dan dbservice sebagai koneksi database. Ada 2 macam hubungan pada class diagram ini, adalah *dependency* pada relasi controller dan view atau antara model pada validationparameter dan dbservice. Hubungan *dependency* yaitu kelas yang memiliki ketergantungan pada kelas lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.

4.8.7. Perancangan Inkremen 3 *Class Diagram* Cetak Laporan

Class diagram cetak laporan transaksi menggambarkan alir aktivitas pada pemrosesan data laporan. Fitur ini berfungsi untuk mencetak dan memproses laporan transaksi yang berupa daftar tagihan, buku kas harian, rekap daftar tagihan, bukti kas masuk dan bukti kas keluar. *Class diagram* cetak laporan transaksi dapat dilihat pada gambar 4.43.



Gambar 4.43 *Class Diagram* Cetak Laporan Transaksi

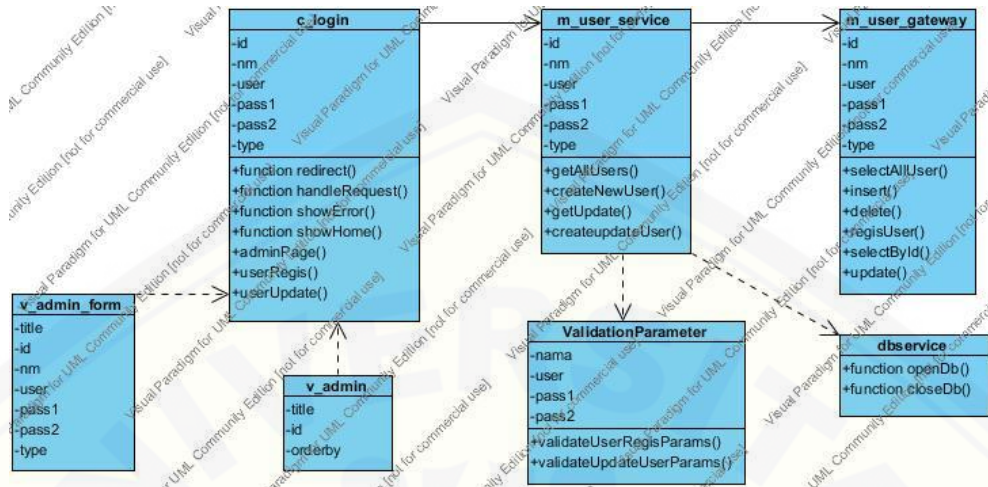
(Sumber: Hasil Analisis, 2015)

Class diagram yang sudah dijelaskan pada gambar 4.43 terdapat *c_laporan* sebagai controller, *v_menu_laporan* dan *v_bukukasharian (sample)* sebagai view, *bukukasharian (sample)* sebagai model. Ada 2 macam hubungan pada class diagram ini, adalah *dependency* pada relasi controller dan view. Hubungan *dependency* yaitu kelas yang memiliki ketergantungan pada kelas lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.

4.8.8. Perancangan Inkremen 4 *Class Diagram* Manajemen Pengurus

Class diagram manajemen pengurus menggambarkan alir aktivitas pada proses manajemen pengurus (*entry*) dan manajemen pengurus (*update-password*). Fitur administrator ini berfungsi untuk mendaftarkan dan mengubah *password user* yang bertanggung jawab atas akses terhadap sistem. *Class diagram* manajemen pengurus dapat dilihat pada gambar 4.44.

Class diagram yang sudah dijelaskan pada gambar 4.44 terdapat *c_login* sebagai controller, *v_admin_form* dan *v_admin* sebagai view, *m_user_service* dan *m_user_gateway* sebagai model, *validationparameter* sebagai pengecekan input dan *dbservice* sebagai koneksi database. Ada 2 macam hubungan pada class diagram ini, adalah *dependency* pada relasi controller dan view atau antara model pada *validationparameter* dan *dbservice*. Hubungan *dependency* yaitu kelas yang memiliki ketergantungan pada kelas lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.



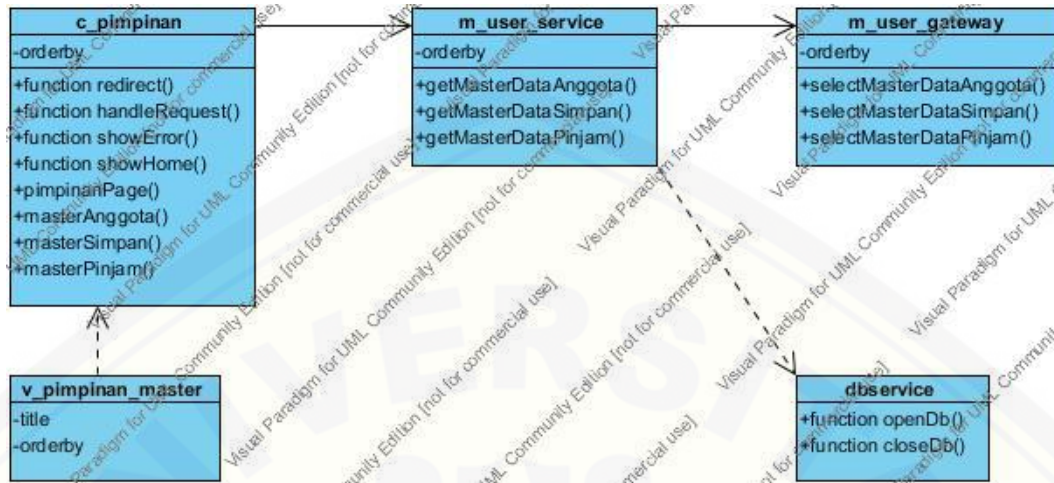
Gambar 4.44 *Class Diagram* Manajemen Pengurus

(Sumber: Hasil Analisis, 2015)

4.8.9. Perancangan Inkremen 5 *Class Diagram* Master Data Transaksi

Class diagram master data transaksi menggambarkan alir aktivitas pada monitoring data transaksi secara menyeluruh. Fitur yang dimiliki oleh pimpinan koperasi ini hanyalah untuk melihat data atau memonitoring data transaksi secara penuh tanpa dibatasi oleh waktu serta dapat disortir sesuai atribut pada tabel. *Class diagram* master data transaksi dapat dilihat pada gambar 4.45.

Class diagram yang sudah dijelaskan pada gambar 4.45 terdapat `c_pimpinan` sebagai controller, `v_pimpinanmaster` sebagai view, `m_user_service` dan `m_user_gateway` sebagai model dan `dbservice` sebagai koneksi database. Ada 2 macam hubungan pada class diagram ini, adalah *dependency* pada relasi controller dan view atau antara model pada `dbservice`. Hubungan *dependency* yaitu kelas yang memiliki ketergantungan pada kelas lain tetapi tidak sebaliknya. Kemudian hubungan *associations* pada relasi controller dan model yaitu sebuah hubungan statis antar kelas yang biasanya menggambarkan atribut yang berupa kelas lainnya.



Gambar 4.45 *Class Diagram* Master Data Transaksi

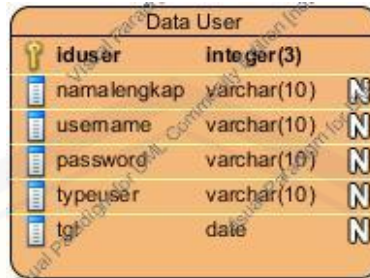
(Sumber: Hasil Analisis, 2015)

4.9. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) menggambarkan tentang hubungan antara entitas yang dibuat dalam skema basisdata yang saling terkait. ERD pada sistem informasi simpan pinjam koperasi dapat dilihat secara keseluruhan pada lampiran 33. ERD desain rancangan sistem informasi simpan pinjam koperasi yang dikelompokkan berdasarkan tahap inkremennya akan dijelaskan pada *point* selanjutnya.

4.9.1. Perancangan Inkremen 1 ERD User Login

ERD user login dapat dilihat pada gambar 4.46. Gambar 4.46 *Entity Relationship Diagram* User Login.

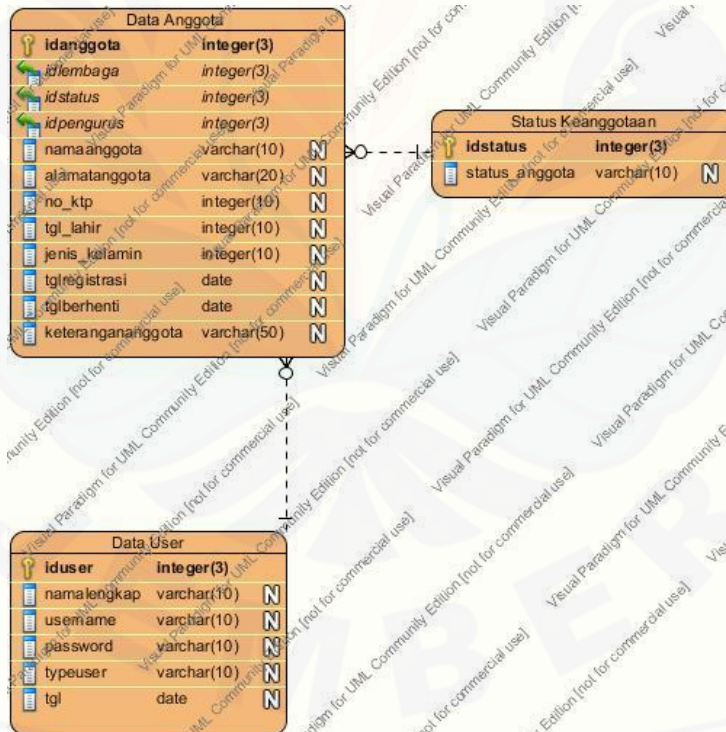


Gambar 4.46 Entity Relationship Diagram User Login

(Sumber: Hasil Analisis, 2015)

4.9.2. Perancangan Inkremen 1 ERD Registrasi

ERD user login dapat dilihat pada gambar 4.47. Gambar 4.47 Entity Relationship Diagram Registrasi Anggota

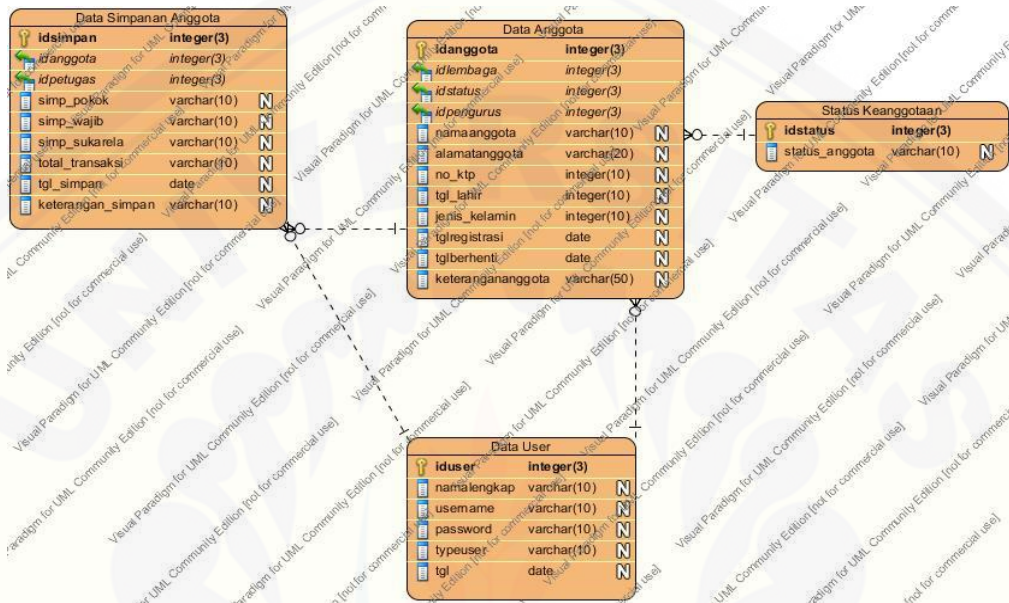


Gambar 4.47 Entity Relationship Diagram Registrasi Anggota

(Sumber: Hasil Analisis, 2015)

4.9.3. Perancangan Inkremen 1 ERD Transaksi Simpanan

ERD user login dapat dilihat pada gambar 4.48. Gambar 4.48 *Entity Relationship Diagram* Transaksi Simpanan Anggota

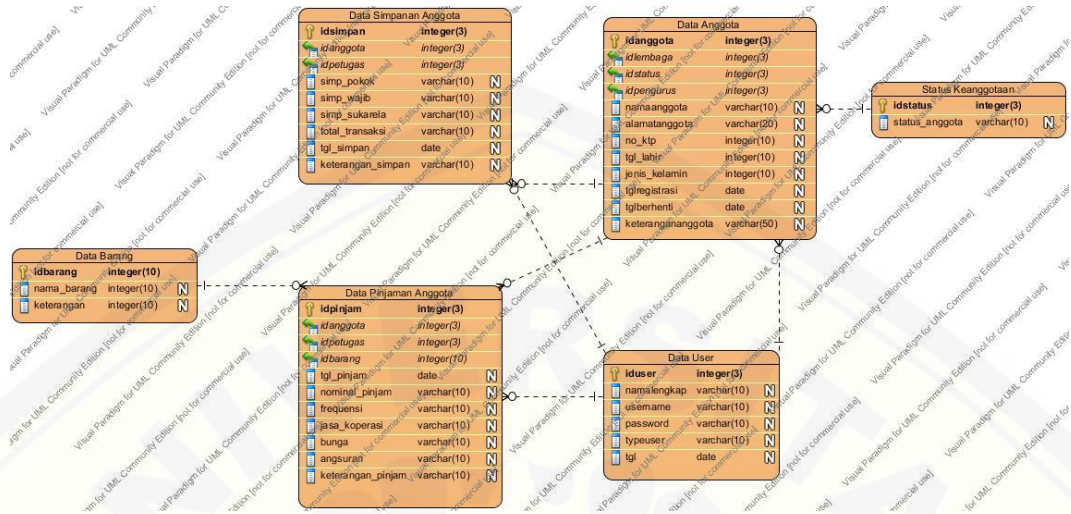


Gambar 4.48 *Entity Relationship Diagram* Transaksi Simpanan Anggota

(Sumber: Hasil Analisis, 2015)

4.9.4. Perancangan Inkremen 1 ERD Transaksi Pinjaman

ERD user login dapat dilihat pada gambar 4.49. Gambar 4.49 *Entity Relationship Diagram* Transaksi Pinjaman Anggota



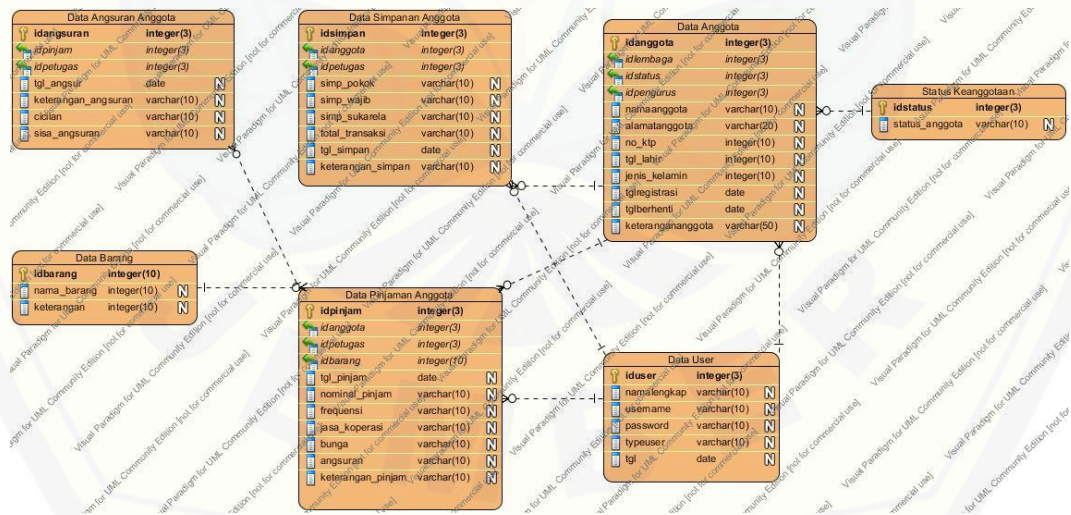
Gambar 4.49 Entity Relationship Diagram Transaksi Pinjaman Anggota

(Sumber: Hasil Analisis, 2015)

4.9.5. Perancangan Inkremen 1 ERD Transaksi Angsuran

ERD user login dapat dilihat pada gambar 4.50.

Gambar 4.50 Entity Relationship Diagram Transaksi Angsuran Anggota



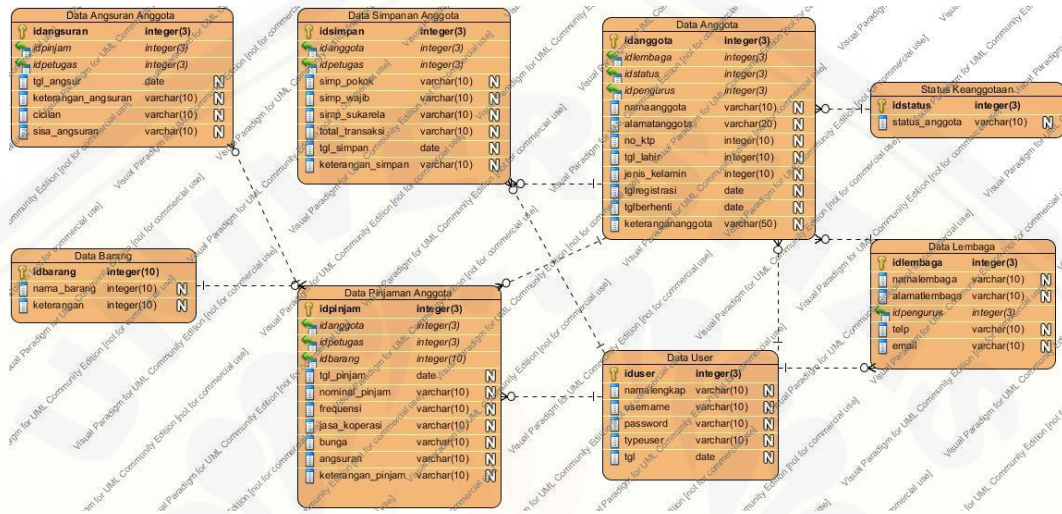
Gambar 4.50 Entity Relationship Diagram Transaksi Angsuran Anggota

(Sumber: Hasil Analisis, 2015)

4.9.6. Perancangan Inkremen 2 ERD Manajemen Lembaga

ERD user login dapat dilihat pada gambar 4.51.

Gambar 4.51 Entity Relationship Diagram Manajemen Lembaga



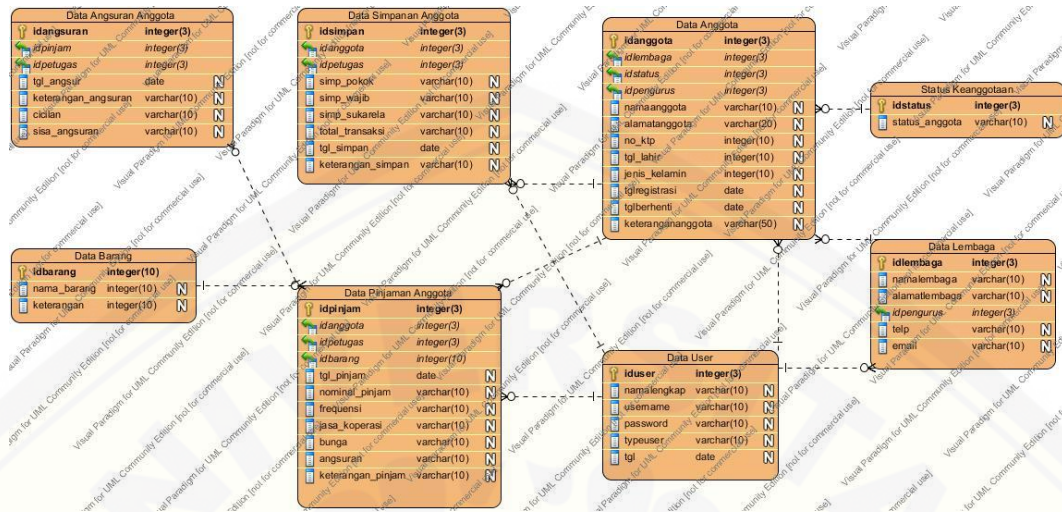
Gambar 4.51 Entity Relationship Diagram Manajemen Lembaga

(Sumber: Hasil Analisis, 2015)

4.9.7. Perancangan Inkremen 3 ERD Cetak Laporan Transaksi

ERD user login dapat dilihat pada gambar 4.52.

Gambar 4.52 Entity Relationship Diagram Cetak Laporan Transaksi

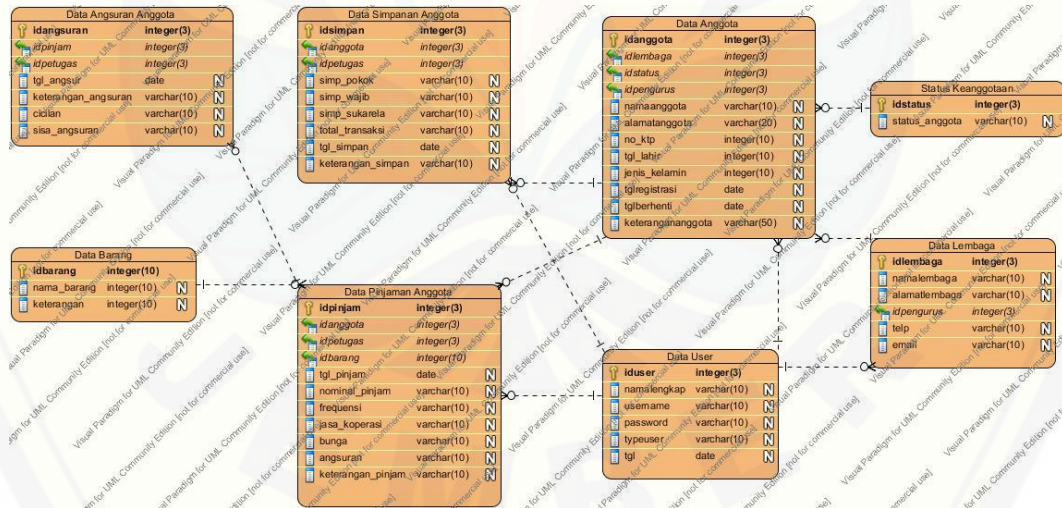


Gambar 4.52 Entity Relationship Diagram Cetak Laporan Transaksi

(Sumber: Hasil Analisis, 2015)

4.9.8. Perancangan Inkremen 4 ERD Pengurus

ERD user login dapat dilihat pada gambar 4.53.

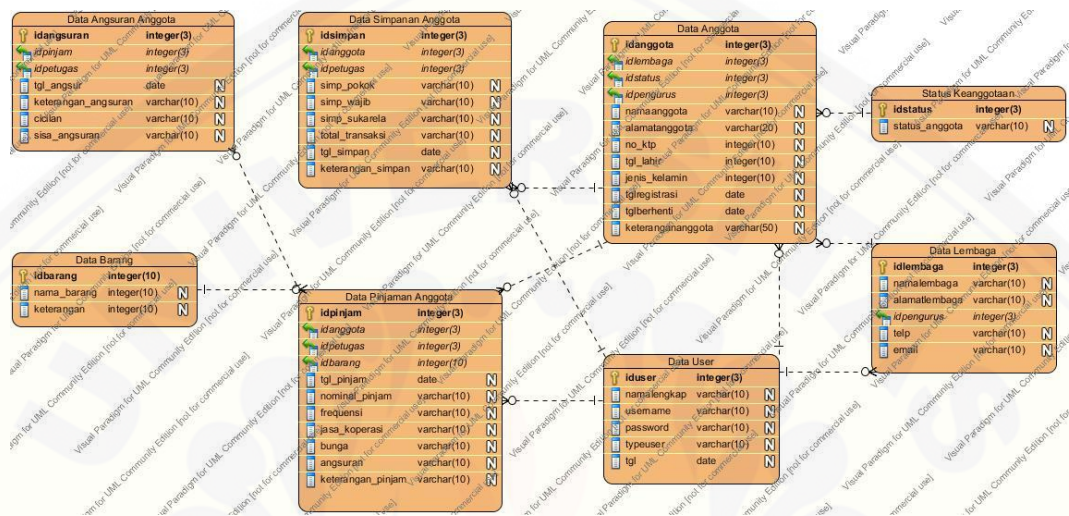


Gambar 4.53 Entity Relationship Diagram Manajemen Pengurus

(Sumber: Hasil Analisis, 2015)

4.9.9. Perancangan Inkremen 5 ERD Master Data Transaksi

ERD user login dapat dilihat pada gambar 4.54.



Gambar 4.54 Entity Relationship Diagram Master Data Transaksi

(Sumber: Hasil Analisis, 2015)

BAB 5 HASIL DAN PEMBAHASAN

Bab ini membahas tentang pengujian sistem dilakukan untuk mengevaluasi program yang telah dibuat. Pada pengujian sistem, dapat dilakukan dengan 2 cara. Yaitu pengujian *white box* dan *black box*.

5.1. Pengujian *White Box*

Pengujian *white box* adalah pengujian apakah pengkodean sistem yang telah dibuat berjalan dengan baik sesuai dengan sebagaimana mestinya. Pengujian ini menggunakan metode *Cyclomatic Complexity* (CC). Teknik pengujian ini menggunakan pengujian jalur dasar (*basis path testing*) yang di dalamnya terdapat beberapa tahapan pengujian antara lain pembuatan diagram alir atau grafik alir, penentuan jalur independen, penghitungan kompleksitas siklomatik jalur independen dan test case. Pengujian dilakukan melalui beberapa tahapan, yaitu: Penomoran listing program, pembuatan diagram alir pembuatan grafik alir, penghitungan kompleksitas siklomatik, penentuan jalur program independen dan *test case*.

5.1.1. Pengujian *White Box* Inkremen 1 User Login

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

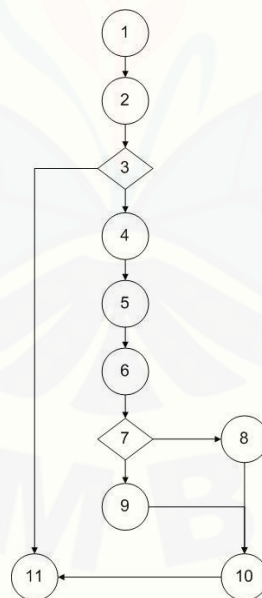
Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function userLogin dapat dilihat pada gambar 5.1.

```
1 { public function userLogin() { //1
2 {   $title = 'Login User';
   $user = '';
   $pass = '';
   $errors = array();
3 { if ( isset($_POST['form_submitted']) ) { //2
4 {   $user = $_POST['user'];
   $pass = md5($_POST['pass']);
5 { try { //3
6 {   $this->kons_dbsevice->openDb();
   $res=mysql_query("select * from tblmultiuser where
   username = '$user' and password = '$pass'");
   $data=mysql_fetch_array($res);
   $name=$data['username'];
   $word=$data['password'];
   $type=$data['type_user'];
7 { if($user==$name && $pass==$word){
8 {   echo '<script>window.location.assign("index.php?operasi=home")</script>';
   }
9 {   else {echo "Gagal Login! password belum diisi, atau kombinasi salah!!";}
   }
10 { $this->kons_dbsevice->closeDb();
   return;
   } //3
10 { catch (ValidationException $e)
   { $errors = $e->getErrors();
11 { } //2
```

Gambar 5.1 Listing Program *Function* userLogin

(Sumber: Hasil Analisis, 2015)

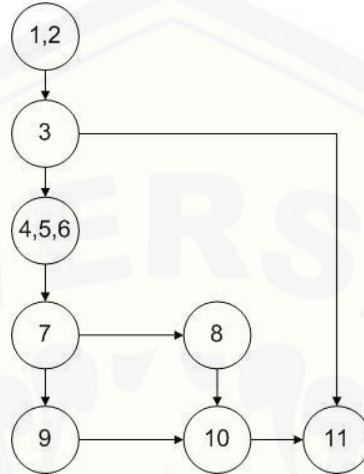
2. Diagram Alir



Gambar 5.2 Diagram Alir *Function* userLogin

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.3 Grafik Alir *Function* userLogin

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexcity (CC)

Penghitungan *Cyclomatic Complexcity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 V(G) &= 10 - 9 + 2 \\
 &= 3
 \end{aligned}$$

$$\text{Cyclomatic Complexcity (CC)} = 3$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-11

Jalur 2 : 1-2-3-4-5-6-7-8-10-11

Jalur 3 : 1-2-3-4-5-6-7-9-10-11

6. *Test Cases*

Tabel 5.1 Test Case Function userLogin

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-11	Belum menekan tombol submit	Form <i>user login</i>	[√] Berhasil [] Gagal
2	1-2-3-4-5-6-7-8-10-11	Semua <i>field</i> sudah diisikan dan valid	Halaman <i>home</i>	[√] Berhasil [] Gagal
3	1-2-3-4-5-6-7-9-10-11	<i>Username</i> belum diinputkan atau <i>password</i> salah	Form <i>user login</i>	[√] Berhasil [] Gagal

5.1.2. Pengujian *White Box* Inkremen 1 Registrasi Anggota

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function userLogin dapat dilihat pada gambar 5.4.

```

108 public function registrasiAnggota() {
109     $title = 'Register';
110
111     $namax = '';
112     $alamatx = '';
113     $telpx = '';
114     $noktpx = '';
115     $tgllahirx = '';
116     $tmptlahirx = '';
117     $genderx = '';
118     $emailx = '';
119     $lembagax = '';
120     $tglx = '';
121     $spokokx = '';
122     $petugasx = '';
123     $ketx = '';
124     $idstatusx = '';
125     $tanggal2x = '';
126
127     $errors = array();
128
129
130
131     if ( isset($_POST['form-submitted']) ) {
132
133         $namax = isset($_POST['nama']) ? $_POST['nama'] :NULL;
134         $alamatx = isset($_POST['alamat']) ? $_POST['alamat'] :NULL;
135         $telpx = isset($_POST['telp']) ? $_POST['telp'] :NULL;
136         $noktpx = isset($_POST['noktp']) ? $_POST['noktp'] :NULL;
137         $tgllahirx = isset($_POST['tgllahir']) ? $_POST['tgllahir'] :NULL;
138         $tmptlahirx = isset($_POST['tempatlahir']) ? $_POST['tempatlahir'] :NULL;
139         $genderx = isset($_POST['gender']) ? $_POST['gender'] :NULL;
140         $emailx = isset($_POST['email']) ? $_POST['email'] :NULL;
141         $lembagax = isset($_POST['lembaga']) ? $_POST['lembaga'] :NULL;
142         $tglx = isset($_POST['tanggal']) ? $_POST['tanggal'] :NULL;
143         $spokokx = isset($_POST['spokok'])? $_POST['spokok'] :NULL;
144         $petugasx = isset($_POST['petugas'])? $_POST['petugas'] :NULL;
145         $ketx = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;
146         $idstatusx = isset($_POST['idstatus'])? $_POST['idstatus'] :NULL;
147         $tanggal2x = isset($_POST['tanggal2'])? $_POST['tanggal2'] :NULL;
148     }
149 }

```

```

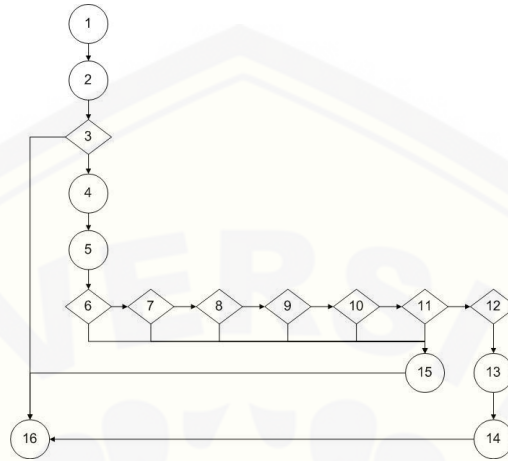
146
147 5  try {
148     if ( !isset($statusubah) || empty($statusubah) ) {
149         $errors[] = 'Status keanggotaan belum diubah!'; 6
150     }
151     elseif ( !isset($alamatbaru) || empty($alamatbaru) ) {
152         $errors[] = 'Alamat belum diinputkan!'; 7
153     }
154     elseif ( !isset($telpbaru) || empty($telpbaru) ) {
155         $errors[] = 'Nomor telepon belum belum diisi!'; 8
156     }
157     elseif ( !is_numeric($telpbaru) ) {
158         $errors[] = 'Nomor telepon tidak diijinkan desimal, huruf, atau simbol!'; 9
159     }
160     elseif ( $telpbaru < 0 ) {
161         $errors[] = 'Nomor telepon tidak boleh bernilai minus!'; 10
162     }
163     elseif ( !isset($emailbaru) || empty($emailbaru) ) {
164         $errors[] = 'Email belum belum diisi!'; 11
165     }
166     elseif ( empty($errors) ) { 12
167         return;
168     }
169
170     $sql="INSERT INTO tblanggota (idlembaga, idstatus, namaanggota, alamatanggota,
171     telpanggota, noktp, tgllahir, tempatlahir, jeniskelamin, emailanggota, tglregistrasi,
172     tglberhenti, statusambildana, tglpengambilan, idpengurus, keterangananggota) VALUES
173     ($db9, $db13, $db1, $db2, $db3, $db4, $db5, $db6, $db7, $db8, $db10, $db14, 'Belum',
174     '-', $db11, $db12)";
175     mysql_query($sql);
176     $lastid=mysql_insert_id();
177     $sql1="INSERT INTO tblsimpanan (idanggota, swajib, spokok, ssukarela, tabungan,
178     totalsimpan, tglsimpan, idpengurus, keterangan simpan) VALUES ($lastid, '0', $db15,
179     '0', '0', $db15, $db10, $db11, $db12)";
180     mysql_query($sql1);
181     $lastid1=mysql_insert_id();
182     return mysql_insert_id();
183
184
185     14  $this->redirect('index.php?operasi=anggota');
186     return;
187 } catch (ValidationException $e) {
188     $errors = $e->getErrors();
189 }
190 }
191 include 'view/v_anggota_entri_form.php'; 16
192 }

```

Gambar 5.4 Listing Program *Function* registrasiAnggota

(Sumber: Hasil Analisis, 2015)

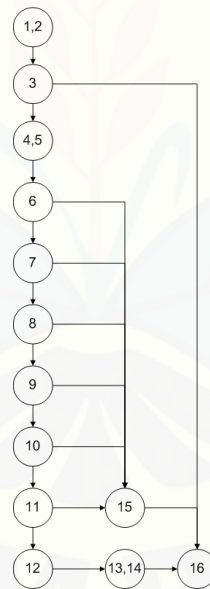
2. Diagram Alir



Gambar 5.5 Diagram Alir *Function* registrasiAnggota

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.6 Grafik Alir *Function* registrasiAnggota

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexity (CC)

Penghitungan *Cyclomatic Complexity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$\begin{aligned} V(G) &= E - N + 2 \\ V(G) &= 19 - 13 + 2 \\ &= 8 \end{aligned}$$

$$\text{Cyclomatic Complexity (CC)} = 8$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-11

Jalur 2 : 1-2-3-4-5-6-15-16

Jalur 3 : 1-2-3-4-5-6-7-15-16

Jalur 4 : 1-2-3-4-5-6-7-8-15-16

Jalur 5 : 1-2-3-4-5-6-7-8-9-15-16

Jalur 6 : 1-2-3-4-5-6-7-8-9-10-15-16

Jalur 7 : 1-2-3-4-5-6-7-8-9-10-11-15-16

Jalur 8 : 1-2-3-4-5-6-7-8-9-10-11-12-13-14-16

6. *Test Cases*

Tabel 5.2 Test Case Function registrasiAnggota

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-11	Belum menekan tombol submit	Form registrasi anggota baru	[√] Berhasil [] Gagal
2	1-2-3-4-5-6-15-16	Belum mengubah status keanggotaan	Form registrasi	[√] Berhasil [] Gagal

			anggota baru	
3	1-2-3-4-5-6-7-15-16	Belum menginputkan alamat anggota	Form registrasi anggota baru	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
4	1-2-3-4-5-6-7-8-15-16	Belum menginputkan nomor telepon	Form registrasi anggota baru	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
5	1-2-3-4-5-6-7-8-9-15-16	Mengisi nomor telepon dengan <i>value</i> desimal, huruf atau simbol	Form registrasi anggota baru	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6	1-2-3-4-5-6-7-8-9-10-15-16	Mengisi nomor telepon dengan angka minus	Form registrasi anggota baru	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7	1-2-3-4-5-6-7-8-9-10-11-15-16	Belum mengisi alamat email anggota	Form registrasi anggota baru	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8	1-2-3-4-5-6-7-8-9-10-11-12-13-14-16	Sudah mengisi semua data yang dibutuhkan dan tidak ada kesalahan pada isian	Halaman fitur registrasi anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

5.1.3. Pengujian *White Box* Inkremen 1 Transaksi Simpanan

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function `entriSimpan` dapat dilihat pada gambar 5.7.

```

41 public function entriSimpan() {
42     1
43     $title = 'Entry';
44     $namax = '';
45     $swajibx = '';
46     $ssukax = '';
47     $stabungx = '';
48     $totalx = '';           2
49     $tglx = '';
50     $petugasx = '';
51     $ketx = '';
52
53     $errors = array();
54
55     3 if ( isset($_POST['form-submitted']) ) {
56
57         $namax = isset($_POST['nama']) ? $_POST['nama'] :NULL;
58         $swajibx = isset($_POST['sw'])? $_POST['sw'] :NULL;
59         $ssukax = isset($_POST['ss'])? $_POST['ss'] :NULL;
60         $stabungx = isset($_POST['st'])? $_POST['st'] :NULL;
61         $totalx = isset($_POST['ts'])? $_POST['ts'] :NULL;           4
62         $tglx = isset($_POST['tanggal'])? $_POST['tanggal'] :NULL;
63         $petugasx = isset($_POST['petugas'])? $_POST['petugas'] :NULL;
64         $ketx = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;
    
```

```

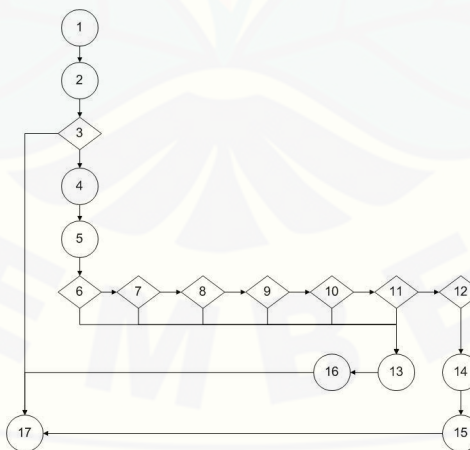
65
66 5   try {
67     if ( !isset($nama) || empty($nama) ) {
68       $errors[] = 'Nama anggota belum dipilih!'; 6
69     }
70     7   elseif ( !is_numeric($suka) ) {
71       $errors[] = 'Nominal simpanan sukarela tidak diijinkan desimal, huruf, atau simbol!';
72     }
73     elseif ( $suka < 0 ) {
74       $errors[] = 'Nominal simpanan sukarela tidak boleh bernilai minus!'; 8
75     }
76     9   elseif ( !is_numeric($tabung) ) {
77       $errors[] = 'Nominal simpanan tabungan tidak diijinkan desimal, huruf, atau simbol!';
78     }
79     elseif ( $tabung < 0 ) {
80       $errors[] = 'Nominal simpanan tabungan tidak boleh bernilai minus!';10
81     }
82     11  elseif ( !isset($tanggal) || empty($tanggal) ) {
83       $errors[] = 'Tanggal transaksi belum diisi!';
84     }
85     elseif ( empty($errors) ) { 12
86       return;
87     }
88     13  throw new ValidationException($errors);
89
90     14  mysql_query("INSERT INTO tblsimpanan (idanggota, swajib, spokok, ssukarela, tabungan,
91     totalsimpan, tglsimpan, idpengurus, keterangan) VALUES ($db1, $db2, '0', $db3,
92     $db4, $db5, $db6, $db7, $db8)");
93     return mysql_insert_id();
94
95     $this->redirect('index.php?operasi=listsimpan'); 15
96     return;
97   } catch (ValidationException $e) {
98     16  $errors = $e->getErrors();
99   }
100 }
101
102 include 'view/v_simp_entri_form.php'; 17
103

```

Gambar 5.7 Listing Program *Function* entriSimpan

(Sumber: Hasil Analisis, 2015)

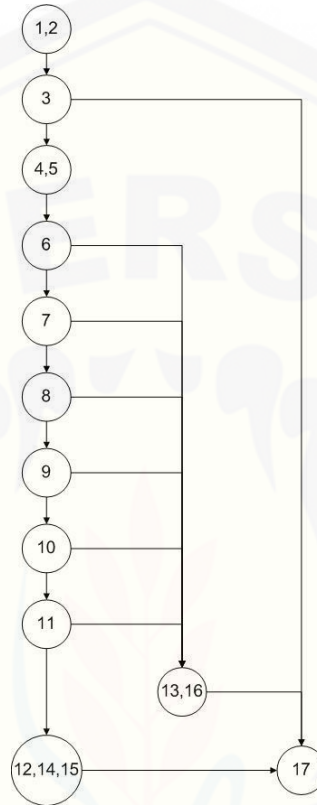
2. Diagram Alir



Gambar 5.8 Diagram Alir *Function* entriSimpan

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.9 Grafik Alir *Function* entriSimpan

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexity (CC)

Penghitungan *Cyclomatic Complexity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 V(G) &= 18 - 12 + 2 \\
 &= 8
 \end{aligned}$$

$$\text{Cyclomatic Complexity (CC)} = 8$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-17

Jalur 2 : 1-2-3-4-5-6-13-16-17

Jalur 3 : 1-2-3-4-5-6-7-13-16-17

Jalur 4 : 1-2-3-4-5-6-7-8-13-16-17

Jalur 5 : 1-2-3-4-5-6-7-8-9-13-16-17

Jalur 6 : 1-2-3-4-5-6-7-8-9-10-13-16-17

Jalur 7 : 1-2-3-4-5-6-7-8-9-10-11-13-16-17

Jalur 8 : 1-2-3-4-5-6-7-8-9-10-11-12-14-15-17

6. Test Cases

Tabel 5.3 Test Case Function entriSimpan

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-17	Belum menekan tombol submit	Form transaksi simpanan	[√] Berhasil [] Gagal
2	1-2-3-4-5-6-13-16-17	Belum menselek nama anggota koperasi	Form transaksi simpanan	[√] Berhasil [] Gagal
3	1-2-3-4-5-6-7-13-16-17	Mengisi simpanan sukarela dengan <i>value</i> desimal, huruf atau simbol	Form transaksi simpanan	[√] Berhasil [] Gagal
4	1-2-3-4-5-6-7-8-13-16-17	Mengisi simpanan sukarela dengan <i>value</i> minus	Form transaksi simpanan	[√] Berhasil [] Gagal

5	1-2-3-4-5-6-7- 8-9-13-16-17	Mengisi simpanan tabungan dengan <i>value</i> desimal, huruf atau simbol	Form transaksi simpanan	[√] Berhasil [] Gagal
6	1-2-3-4-5-6-7- 8-9-10-13-16- 17	Mengisi simpanan tabungan dengan <i>value</i> minus	Form transaksi simpanan	[√] Berhasil [] Gagal
7	1-2-3-4-5-6-7- 8-9-10-11-13- 16-17	Belum mengisi tanggal transaksi simpanan anggota	Form transaksi simpanan	[√] Berhasil [] Gagal
8	1-2-3-4-5-6-7- 8-9-10-11-12- 14-15-17	Sudah mengisi semua data yang dibutuhkan dan tidak ada kesalahan pada isian	Halaman fitur transaksi simpanan	[√] Berhasil [] Gagal

5.1.4. Pengujian *White Box* Inkremen 1 Transaksi Pinjaman

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function entriPinjam dapat dilihat pada gambar 5.9.

```

42 public function entriPinjam() {
43     $title = 'Entry';
44
45     $ax = '';
46     $bx = '';
47     $cx = '';
48     $dx = '';
49     $ex = '';
50     $fx = '';
51     $gx = '';
52     $ix = '';
53     $jx = '';
54     $kx = '';
55     $lx = '';
56     $mx = '';
57     $errors = array();
58
59     3 if ( isset($_POST['form-submitted']) ) {
60
61         $ax = isset($_POST['nama']) ? $_POST['nama'] :NULL;
62         $bx = isset($_POST['tanggal'])? $_POST['tanggal'] :NULL;
63         $cx = isset($_POST['jenis'])? $_POST['jenis'] :NULL;
64         $dx = isset($_POST['barang'])? $_POST['barang'] :NULL;
65         $ex = isset($_POST['nominal'])? $_POST['nominal'] :NULL;
66         $fx = isset($_POST['freq'])? $_POST['freq'] :NULL;
67         $gx = isset($_POST['jasa'])? $_POST['jasa'] :NULL;
68         $hx = isset($_POST['bunga'])? $_POST['bunga'] :NULL;
69         $ix = isset($_POST['cicil1'])? $_POST['cicil1'] :NULL;
70         $jx = isset($_POST['cicil2'])? $_POST['cicil2'] :NULL;
71         $kx = isset($_POST['statuspinjam'])? $_POST['statuspinjam'] :NULL;
72         $lx = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;
73         $mx = isset($_POST['petugas'])? $_POST['petugas'] :NULL;
74
75     5 try {
76         $this->kons_dbsevice->openDb();
77         $query=mysql_query("select status_pinjaman from tblpinjaman where
78         idanggota='".$nama.'" and status_pinjaman='Belum Lunas' ") or die(mysql_error());
79         $blmlunas=mysql_num_rows($query);
80         6 if ( $blmlunas>0 ) {
81             $errors[] = 'Anggota tersebut sebelumnya masih memiliki tanggungan pinjaman kope
82         }
83         7 elseif ( !isset($nama) || empty($nama) ) {
84             $errors[] = 'Nama anggota belum diinputkan!';
85         }
86         elseif ( !isset($tanggal) || empty($tanggal) ) {
87             $errors[] = 'Tanggal transaksi belum diisi!'; 8
88         }
89         9 elseif ( !isset($jenis) || empty($jenis) ) {
90             $errors[] = 'Jenis transaksi belum ditentukan!';
91         }
92         elseif ( !isset($barang) || empty($barang) ) {
93             $errors[] = 'Barang atau jasa belum dipilih!'; 10
94         }
95         11 elseif ( !isset($nomi) || empty($nomi) ) {
96             $errors[] = 'Nominal pinjaman belum diinputkan!';
97         }
98         elseif ( !is_numeric($nomi) ) {
99             $errors[] = 'Nominal pinjaman tidak diijinkan desimal, huruf, atau simbol!'; 12
100         }
101         13 elseif ( $nomi <= 0 ) { // tidak boleh minus
102             $errors[] = 'Nominal pinjaman tidak boleh bernilai minus!';
103         }
104         elseif ( !isset($freq) || empty($freq) ) {
105             $errors[] = 'Frekuensi angsuran belum dipilih!'; 14
106         }
107         15 elseif ( empty($errors) ) {
108             return;
109         }
110         16 throw new ValidationException($errors);

```

```

111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131

```

```

17
    $sql="INSERT INTO tblpinjaman (idanggota, idpengurus, tglpinjam, jenis, idbarang,
    totalpinjam, freqangsur, jasa, bunga, cicilan1, cicilan2, status_pinjaman,
    keteranganpinjam) VALUES ($db1, $db13, $db2, $db3, $db4, $db5, $db6, $db7,
    $db8, $db9, $db10, $db11, $db12)";
    mysql_query($sql);
    $lastid=mysql_insert_id();

    $sql1="INSERT INTO tblangsuran (idpinjam, tglangsur, cicilan, sisaangsur)
    VALUES ($lastid, $db2, $db9, '0')";
    mysql_query($sql1);
    return mysql_insert_id();

```

```

    $this->redirect('index.php?operasi=listpinjam'); 18
    return;

```

```

19
    } catch (ValidationException $e) {
        $errors = $e->getErrors();
    }

```

```

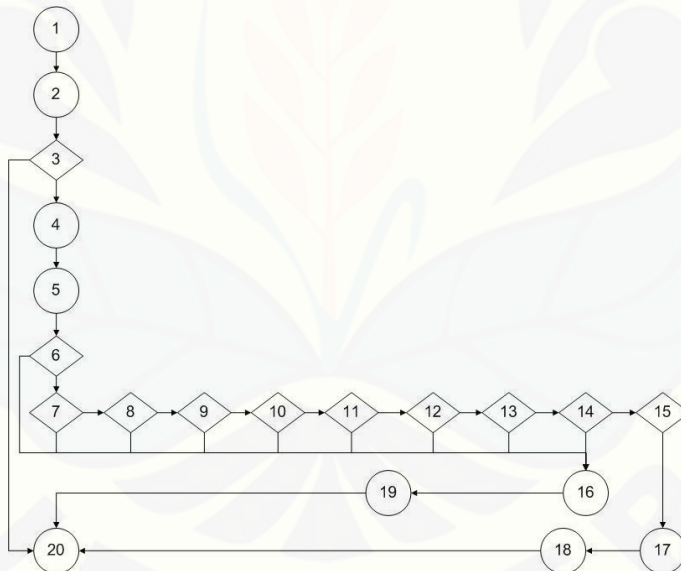
    include 'view/v_pinj_entri_form.php';20
}

```

Gambar 5.10 Listing Program *Function* entriPinjam

(Sumber: Hasil Analisis, 2015)

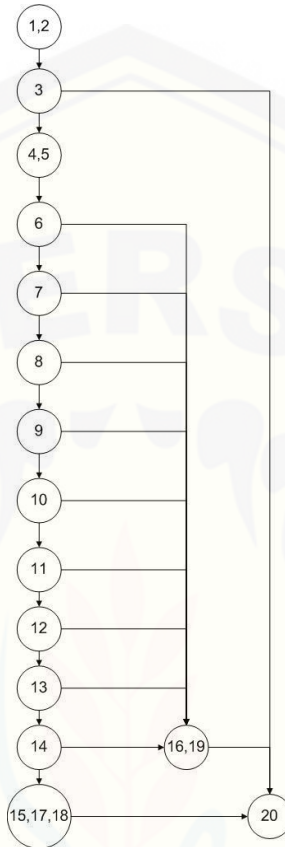
2. Diagram Alir



Gambar 5.11 Diagram Alir *Function* entriPinjam

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.12 Grafik Alir *Function* entriPinjam

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexity (CC)

Penghitungan *Cyclomatic Complexity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 V(G) &= 24 - 15 + 2 \\
 &= 11
 \end{aligned}$$

$$\text{Cyclomatic Complexity (CC)} = 11$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-20

Jalur 2 : 1-2-3-4-5-6-16-19-20

Jalur 3 : 1-2-3-4-5-6-7-16-19-20

Jalur 4 : 1-2-3-4-5-6-7-8-16-19-20

Jalur 5 : 1-2-3-4-5-6-7-8-9-16-19-20

Jalur 6 : 1-2-3-4-5-6-7-8-9-10-16-19-20

Jalur 7 : 1-2-3-4-5-6-7-8-9-10-11-16-19-20

Jalur 8 : 1-2-3-4-5-6-7-8-9-10-11-12-16-19-20

Jalur 9 : 1-2-3-4-5-6-7-8-9-10-11-12-13-16-19-20

Jalur 10 : 1-2-3-4-5-6-7-8-9-10-11-12-13-14-16-19-20

Jalur 11 : 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-17-18-20

6. Test Cases

Tabel 5.4 Test Case Function entriPinjam

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-20	Belum menekan tombol submit	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2	1-2-3-4-5-6-16-19-20	Menselek anggota yang masih memiliki tanggungan pinjaman sebelumnya	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
3	1-2-3-4-5-6-7-16-19-20	Belum menselek nama anggota koperasi	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

4	1-2-3-4-5-6-7- 8-16-19-20	Belum menginputkan tanggal transaksi pinjaman	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
5	1-2-3-4-5-6-7- 8-9-16-19-20	Belum menselek jenis transaksi pinjaman	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6	1-2-3-4-5-6-7- 8-9-10-16-19- 20	Belum menselek transaksi berupa barang atau jasa koperasi	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
7	1-2-3-4-5-6-7- 8-9-10-11-16- 19-20	Belum mengisi nominal transaksi pinjaman anggota	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8	1-2-3-4-5-6-7- 8-9-10-11-12- 16-19-20	Mengisi nominal pinjaman <i>value</i> desimal, huruf atau simbol	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
9	1-2-3-4-5-6-7- 8-9-10-11-12- 13-16-19-20	Mengisi nominal pinjaman <i>value</i> minus	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
10	1-2-3-4-5-6-7- 8-9-10-11-12- 13-14-16-19- 20	Belum menselek frekuensi pinjaman yang dilakukan	Form entri pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
11	1-2-3-4-5-6-7- 8-9-10-11-12- 13-14-15-17- 18-20	Sudah mengisi semua data yang dibutuhkan dan tidak ada kesalahan pada isian	Halaman fitur transaksi pinjaman anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

5.1.5. Pengujian *White Box* Inkremen 1 Transaksi Angsuran

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function `updateAngsur` dapat dilihat pada gambar 5.13.

```

40 public function updateAngsur() {
41     $title = 'Update';
42
43     $errors = array();
44
45     $id = isset($_GET['idpinjam'])?$_GET['idpinjam']:NULL; 2
46     if ( !$id ) {
47         throw new Exception('Internal error.');
```

```

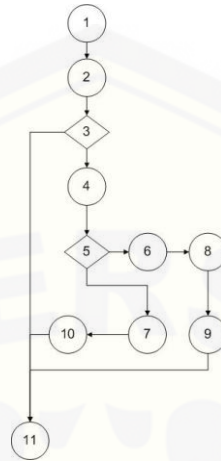
48     }
49     $update = $this->kons_m_angsu_service->getUpdate($id);
50
51     if ( isset($_POST['form-submitted']) ) {
52
53         $idnyax      = isset($_POST['idnya']) ? $_POST['idnya'] :NULL;
54         $idpinjx    = isset($_POST['idpinj']) ? $_POST['idpinj'] :NULL;
55         $tglnewx    = isset($_POST['tglnew']) ? $_POST['tglnew'] :NULL;
56         $cicilx     = isset($_POST['cicil']) ? $_POST['cicil'] :NULL;
57         $angsurkrngx = isset($_POST['angsurkrng']) ? $_POST['angsurkrng'] :NULL;
58         $freqangsurx = isset($_POST['freqangsur']) ? $_POST['freqangsur'] :NULL;
59
60     4 try {
61         if ( !isset($tglnew) || empty($tglnew) ) {
62             $errors[] = 'Tanggal transaksi belum diisi!'; 5
63         }
64         6 elseif ( empty($errors) ) {
65             return;
66         }
67         7 throw new ValidationException($errors);
68
69         8 mysql_query("INSERT INTO tblangsuran (idpinjam, tglangsur, cicilan, sisaangsur)
70             VALUES ($db2, $db3, $db4, $db5)");
71         return mysql_insert_id();
72
73         $this->redirect('index.php?operasi=listangsur') 9
74         return;
75     } catch (ValidationException $e) {
76         10 $errors = $e->getErrors();
77     }
78 }
79 include 'view/v_angsu_entriform.php'; 11
80 }

```

Gambar 5.13 Listing Program *Function* `updateAngsur`

(Sumber: Hasil Analisis, 2015)

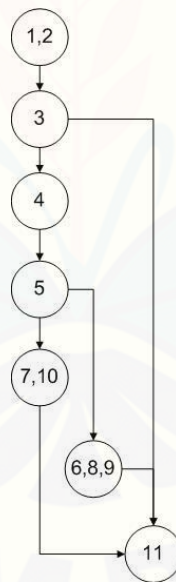
2. Diagram Alir



Gambar 5.14 Diagram Alir *Function* updateAngsur

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.15 Grafik Alir *Function* updateAngsur

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexcity (CC)

Penghitungan *Cyclomatic Complexcity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$V(G) = E - N + 2$$

$$V(G) = 8 - 7 + 2$$

$$= 3$$

$$\text{Cyclomatic Complexcity (CC)} = 3$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-11

Jalur 2 : 1-2-3-4-5-6-8-9-11

Jalur 3 : 1-2-3-4-5-7-10-11

6. *Test Cases*

Tabel 5.5 Test Case Function updateAngsur

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-11	Belum menekan tombol submit	Form entri data angsuran	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
2	1-2-3-4-5-6-8-9-11	Belum mengisi tanggal transaksi angsuran	Form entri data angsuran	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8	1-2-3-4-5-7-10-11	Sudah mengisi semua data yang dibutuhkan dan tidak ada kesalahan pada isian	Halaman fitur angsuran anggota	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

5.1.6. Pengujian *White Box* Inkremen 2 Manajemen Lembaga

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function `entriLembaga` dapat dilihat pada gambar 5.1.

```

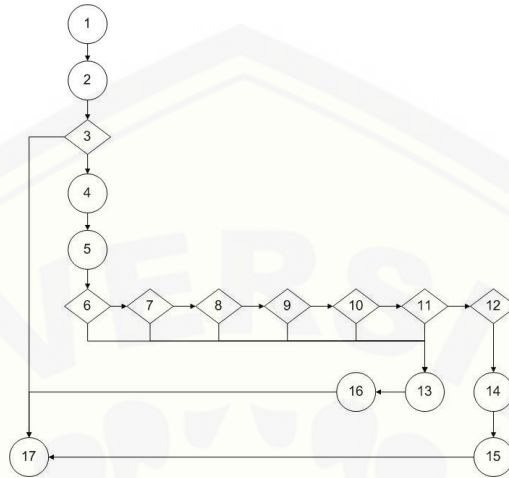
41 public function entriLembaga() {
42     1 $title = 'Entry';
43
44     $namalbgx = '';
45     $alamatx = '';
46     $telpx = '';
47     $emailx = '';
48     $petugasx = '';
49
50     $errors = array();
51
52     3 if ( isset($_POST['form-submitted']) ) {
53
54         $namalbgx = isset($_POST['namalbg']) ? $_POST['namalbg'] :NULL;
55         $alamatx = isset($_POST['alamat']) ? $_POST['alamat'] :NULL;
56         $telpx = isset($_POST['telp']) ? $_POST['telp'] :NULL;
57         $emailx = isset($_POST['email']) ? $_POST['email'] :NULL;
58         $petugasx = isset($_POST['petugas'])? $_POST['petugas'] :NULL;
59
60         5 try {
61             if ( !isset($nama) || empty($nama) ) {
62                 $errors[] = 'Nama lembaga belum diisi!'; 6
63             }
64             7 elseif ( !isset($alamat) || empty($alamat) ) {
65                 $errors[] = 'Alamat lembaga belum diisi!';
66             }
67             elseif ( !isset($telp) || empty($telp) ) {
68                 $errors[] = 'Nomor telepon belum diisi!'; 8
69             }
70             9 elseif ( !is_numeric($telp) ) {
71                 $errors[] = 'Nomor telepon tidak diijinkan desimal, huruf, atau simbol!';
72             }
73             elseif ( $telp <= 0 ) {
74                 $errors[] = 'Nomor telepon tidak boleh bernilai minus!';10
75             }
76             11 elseif ( !isset($email) || empty($email) ) {
77                 $errors[] = 'Email lembaga belum diisi!';
78             }
79             elseif ( empty($errors) ) {
80                 return; 12
81             }
82             13 throw new ValidationException($errors);
83
84             14 mysql_query("INSERT INTO tbllembaga (nama, alamat, telp,
85                 email, idpengurus) VALUES ($db1, $db2, $db3, $db4, $db5)");
86             return mysql_insert_id();
87
88             $this->redirect('index.php?operasi=lembaga'); 15
89             return;
90         } catch (ValidationException $e) {
91             16 $errors = $e->getErrors();
92         }
93     }
94
95     include 'view/v_lembaga_entri_form.php'; 17
96 }

```

Gambar 5.16 Listing Program *Function* entriLembaga

(Sumber: Hasil Analisis, 2015)

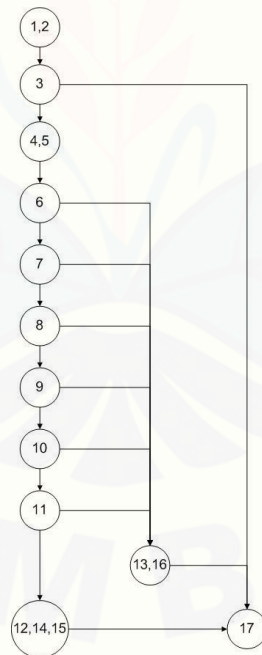
2. Diagram Alir



Gambar 5.17 Diagram Alir *Function* entriLembaga

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.18 Grafik Alir *Function* entriLembaga

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexcity (CC)

Penghitungan *Cyclomatic Complexcity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$\begin{aligned} V(G) &= E - N + 2 \\ V(G) &= 18 - 12 + 2 \\ &= 8 \end{aligned}$$

$$\text{Cyclomatic Complexcity (CC)} = 3$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-17

Jalur 2 : 1-2-3-4-5-6-13-16-17

Jalur 3 : 1-2-3-4-5-6-7-13-16-17

Jalur 4 : 1-2-3-4-5-6-7-8-13-16-17

Jalur 5 : 1-2-3-4-5-6-7-8-9-13-16-17

Jalur 6 : 1-2-3-4-5-6-7-8-9-10-13-16-17

Jalur 7 : 1-2-3-4-5-6-7-8-9-10-11-13-16-17

Jalur 8 : 1-2-3-4-5-6-7-8-9-10-11-12-14-15-17

6. *Test Cases*

Tabel 5.6 Test Case Function entriLembaga

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-17	Belum menekan tombol submit	Form entri lembaga	[√] Berhasil [] Gagal
2	1-2-3-4-5-6-13-16-17	Belum mengisi nama lembaga	Form entri lembaga	[√] Berhasil [] Gagal
3	1-2-3-4-5-6-7-13-16-17	Belum mengisi alamat lembaga	Form entri lembaga	[√] Berhasil [] Gagal

4	1-2-3-4-5-6-7- 8-13-16-17	Belum mengisi nomor telepon lembaga	Form entri lembaga	[√] Berhasil [] Gagal
5	1-2-3-4-5-6-7- 8-9-13-16-17	Mengisi nomor telepon lembaga dengan <i>value</i> desimal, huruf atau simbol	Form entri lembaga	[√] Berhasil [] Gagal
6	1-2-3-4-5-6-7- 8-9-10-13-16- 17	Mengisi nomor telepon lembaga dengan <i>value</i> minus	Form entri lembaga	[√] Berhasil [] Gagal
7	1-2-3-4-5-6-7- 8-9-10-11-13- 16-17	Belum mengisi alamat email lembaga	Form entri lembaga	[√] Berhasil [] Gagal
8	1-2-3-4-5-6-7- 8-9-10-11-12- 14-15-17	Sudah mengisi semua data yang dibutuhkan dan tidak ada kesalahan pada isian	Halaman fitur lembaga	[√] Berhasil [] Gagal

5.1.7. Pengujian *White Box* Inkremen 3 Cetak Laporan Transaksi

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function Content dapat dilihat pada gambar 5.18.


```

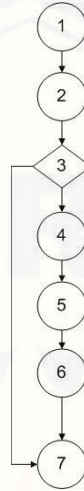
2 function Content(){
3
4 $lembaga=$_GET['lembaga'];
5 $bulan=$_GET['bulan'];
6 $tahun=$_GET['tahun'];
7
8 $query = "
9 SELECT a.idanggota, a.namaanggota, sum(b.swajib) as swajib,sum(b.spokok) as spokok,
10 sum(b.ssukarela) as ssukarela, c.totalpinjam, c.freqangsur,c.idpinjam, c.jasa,
11 sum(IFNULL(b.totalsimpan,0)) + IFNULL(c.totalpinjam,0) as totalsdp, c.idbarang
12 FROM `tblanggota` AS a
13 LEFT JOIN tblsimpanan AS b ON a.idanggota = b.idanggota
14 LEFT JOIN tblpinjaman AS c ON a.idanggota = c.idanggota
15 WHERE month(b.tglsimpan)=".$bulan." and year(b.tglsimpan)=".$tahun."
16 and month(c.tglpinjam)=".$bulan."
17 and year(c.tglpinjam)=".$tahun." and c.status_pinjaman = 'Belum Lunas'
18 and a.idlembaga=".$lembaga."
19 GROUP BY a.idanggota
20 ";
21
22 $db_query = mysql_query($query) or die("Query gagal");
23
24 $i = 0;
25 while($data=mysql_fetch_row($db_query, MYSQL_ASSOC))
26 {
27 $cell[$i][0] = $data[0];
28 $cell[$i][1] = $data[namaanggota];
29 $cell[$i][2] = $swajib;
30 $cell[$i][3] = $spokok;
31 $cell[$i][4] = $ssukarela;
32 $cell[$i][5] = $data[totalsimpan];
33 $i++;
34 }
35
36 $this->SetFont('Times','I',12);
37 for($j=0;$j<$i;$j++)
38 {
39 $this->Cell(10,5,$j+1,1,0,'C',0);
40 $this->Cell(32,5,$cell[$j][1],1,0,'L',0);
41 $this->Cell(13,5,'',1,0,'L',0);
42 $this->Cell(24,5,$cell[$j][2],'LRB',0,'R',0);
43 $this->Cell(24,5,$cell[$j][3],'LRB',0,'R',0);
44 $this->Cell(24,5,$cell[$j][4],'LRB',0,'R',0);
45 $this->Cell(13,5,$cell[$j][6].".$cell[$j][15].".$cell[$j][7],'LRB',0,'C',0);
46 $this->Cell(24,5,$cell[$j][8],'LRB',0,'R',0);
47 $this->Cell(24,5,$cell[$j][9],'LRB',0,'C',0);
48 $this->Cell(13,5,$cell[$j][10].".$cell[$j][16].".$cell[$j][11],'LRB',0,'C',0);
49 $this->Cell(24,5,$cell[$j][12],'LRB',0,'R',0);
50 $this->Cell(24,5,$cell[$j][13],'LRB',0,'C',0);
51 $this->Cell(30,5,$cell[$j][14],'LRB',0,'R',0);
52 $this->Ln();
53 }
54 $this->Ln();
55 $this->Ln();
56 $this->Ln();
57 }
58 }
59
60 $pdf = new PDF();
61 $pdf->AliasNbPages();
62 $pdf->AddPage(L);
63 $pdf->Content();
64 $pdf->Output();
65 ?>

```

Gambar 5.19 Listing Program *Function Content*

(Sumber: Hasil Analisis, 2015)

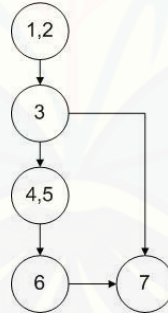
2. Diagram Alir



Gambar 5.20 Diagram Alir *Function Content*

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.21 Grafik Alir *Function Content*

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexcity (CC)

Penghitungan *Cyclomatic Complexcity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$V(G) = E - N + 2$$

$$V(G) = 5 - 5 + 2$$

$$= 2$$

$$\text{Cyclomatic Complexity (CC)} = 2$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-7

Jalur 2 : 1-2-3-4-5-6-7

6. Test Cases

Tabel 5.7 Test Case Function Content

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-7	Menuliskan query dengan tidak valid atau script koneksi ke database salah	Halaman laporan tetapi muncul message error "Query Gagal"	[√] Berhasil [] Gagal
2	1-2-3-4-5-6-7	Tidak ada kesalahan pada query dan koneksi pada database	Halaman laporan siap cetak beserta isi laporan	[√] Berhasil [] Gagal

5.1.8. Pengujian *White Box* Inkremen 4 Manajemen Pengurus

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function userRegis dapat dilihat pada gambar 5.21.

```

53 1  public function userRegis() {
54      $title = 'Register';
55      $nmrx = '';
56      $userx = '';
57      $pass1x = '';
58      $pass2x = '';           2
59      $typex = '';
60      $tglx = '';
61      $errors = array();
62
63 3  if ( isset($_POST['form-submitted']) ) {
64
65      $nmrx = $_POST['nm'];
66      $userx = $_POST['user'];
67      $pass1x = $_POST['pass1'];
68      $pass2x = $_POST['pass2'];           4
69      $typex = $_POST['type'];
70      $tglx = date('Y-m-d H:i:s');
71
72 5  try {
73      if ( !isset($nama) || empty($nama) ) {
74          $errors[] = 'Nama user belum diisikan!';           6
75      }
76      elseif ( !isset($user) || empty($user) ) {
77          $errors[] = 'Username belum diisi!';           7
78      }
79      elseif ( !isset($pass1) || empty($pass1) ) {
80          $errors[] = 'Password belum diisi!';           8
81      }
82      elseif ( !isset($pass2) || empty($pass2) ) {
83          $errors[] = 'Konfirmasi password belum diisi!';           9
84      }
85      elseif ( $pass1 != $pass2 ) {
86          $errors[] = 'Konfirmasi password salah!';           10
87      }
88      elseif ( !isset($type) || empty($type) ) {
89          $errors[] = 'Type user belum dipilih!';           11
90      }
91      elseif ( empty($errors) ) {           12
92          return;
93      }
94      throw new ValidationException($errors);           13

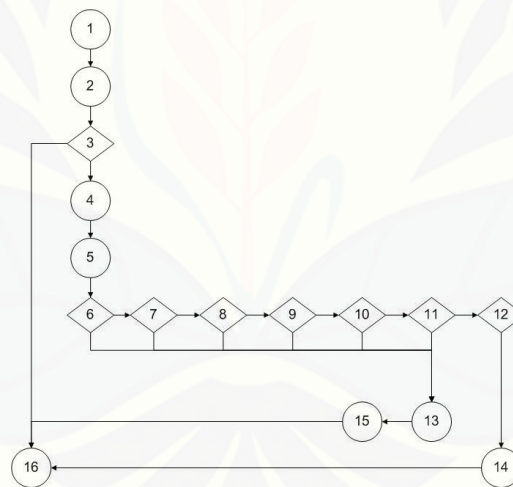
```

```
95  
96  
97  
98 14  mysql_query("INSERT INTO tblsimpanan (idanggota, swajib, spokok,  
99      ssukarela, totalsimpan, tglsimpan, petugassimpan,  
100     keterangan) VALUES ($db1, $db2, '0', $db3, $db4, $db5,  
101     $db6, $db7)");  
102     return mysql_insert_id();  
103  
104     } catch (ValidationException $e) {  
105         $errors = $e->getErrors();  
106     }  
107  
108 16  include 'view/v_admin_entri_form.php';  
109 }
```

Gambar 5.22 Listing Program *Function* userRegis

(Sumber: Hasil Analisis, 2015)

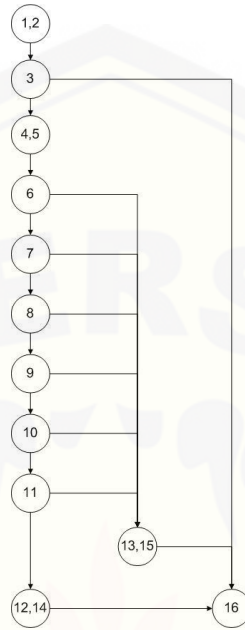
2. Diagram Alir



Gambar 5.23 Diagram Alir *Function* userRegis

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.24 Grafik Alir *Function* userRegis

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexcity (CC)

Penghitungan *Cyclomatic Complexcity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 V(G) &= 18 - 12 + 2 \\
 &= 8
 \end{aligned}$$

$$\text{Cyclomatic Complexcity (CC)} = 8$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-16

Jalur 2 : 1-2-3-4-5-6-13-15-16

Jalur 3 : 1-2-3-4-5-6-7-13-15-16

Jalur 4 : 1-2-3-4-5-6-7-8-13-15-16

Jalur 5 : 1-2-3-4-5-6-7-8-9-13-15-16

Jalur 6 : 1-2-3-4-5-6-7-8-9-10-13-15-16

Jalur 7 : 1-2-3-4-5-6-7-8-9-10-11-13-15-16

Jalur 8 : 1-2-3-4-5-6-7-8-9-10-11-12-14-16

6. Test Cases

Tabel 5.8 Test Case Function userRegis

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-16	Belum menekan tombol submit	Form registrasi user baru	[√] Berhasil [] Gagal
2	1-2-3-4-5-6-13-15-16	Belum mengisi nama lengkap user	Form registrasi user baru	[√] Berhasil [] Gagal
3	1-2-3-4-5-6-7-13-15-16	Belum mengisi <i>username</i>	Form registrasi user baru	[√] Berhasil [] Gagal
4	1-2-3-4-5-6-7-8-13-15-16	Belum mengisi <i>password</i>	Form registrasi user baru	[√] Berhasil [] Gagal
5	1-2-3-4-5-6-7-8-9-13-15-16	Belum mengisis konfirmasi <i>password</i>	Form registrasi user baru	[√] Berhasil [] Gagal
6	1-2-3-4-5-6-7-8-9-10-13-15-16	Kombinasi <i>password</i> salah	Form registrasi user baru	[√] Berhasil [] Gagal
7	1-2-3-4-5-6-7-	Belum memilih <i>type</i>	Form	[√] Berhasil

	8-9-10-11-13-15-16	<i>user</i> atau level <i>user</i>	registrasi user baru	[] Gagal
8	1-2-3-4-5-6-7-8-9-10-11-12-14-16	Sudah mengisi semua data yang dibutuhkan dan tidak ada kesalahan pada isian	Halaman administratif or	[√] Berhasil [] Gagal

5.1.9. Pengujian *White Box* Inkremen 5 Master Data Transaksi

Pengujian user login dilakukan dengan melakukan tes pada sistem menggunakan prosedur pengujian *white box* terkait dengan script atau coding yang dibuat oleh pihak *developer*.

1. Listing Program

Pada tahapan listing program diberikan penomoran yang ditulis berdasarkan statement dan logika pada kode program. Listing program function *masterAnggota* dapat dilihat pada gambar 5.24.

```

26 1  public function masterAnggota() {
27      2  $title = 'Master Anggota';
28          $orderlist = isset($_GET['orderby'])?$_GET['orderby']:NULL;
29
30      3  if ( !isset($order) ) {
31          $order = "idanggota";
32      }
33
34      4  $dbOrder = mysql_real_escape_string($order);
35          $dbres = mysql_query("SELECT * FROM tblanggota, tblstatusanggota,
36          tbllembaga where tblanggota.idstatus = tblstatusanggota.idstatus
37          and tblanggota.idlembaga = tbllembaga.idlembaga ORDER BY
38          $dbOrder ASC");
39
40      5  $masteranggota = array();
41          while ( ($obj = mysql_fetch_object($dbres)) != NULL ) {
42              $masteranggota[] = $obj;
43
44          }
45
46          return $masteranggota;
47
48          include 'view/v_pimpinan_master_anggota.php';
49      }

```


Gambar 5.25 Listing Program *Function* masterAnggota

(Sumber: Hasil Analisis, 2015)

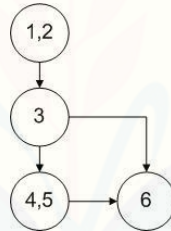
2. Diagram Alir



Gambar 5.26 Diagram Alir *Function* masterAnggota

(Sumber: Hasil Analisis, 2015)

3. Grafik Alir



Gambar 5.27 Grafik Alir *Function* masterAnggota

(Sumber: Hasil Analisis, 2015)

4. Penghitungan Cyclomatic Complexcity (CC)

Penghitungan *Cyclomatic Complexcity* (CC). Penghitungan CC yang disimbolkan oleh $V(G)$ adalah;

$$V(G) = E - N + 2$$

$$V(G) = 4 - 4 + 2$$

$$= 2$$

$$\text{Cyclomatic Complexcity (CC)} = 2$$

5. Jalur Program Independen

Berdasarkan hasil perhitungan dari CC di atas, maka dihasilkan beberapa penentuan jalur independen dari grafik alir yang dijabarkan sebagai berikut;

Jalur 1 : 1-2-3-6

Jalur 2 : 1-2-3-4-5-6

6. *Test Cases*

Tabel 5.9 Test Case Function masterAnggota

NO	JALUR	TEST CASE	HASIL	KESIMPULAN
1	1-2-3-11	Jika user tidak menselek urutan data, secara otomatis sistem mengurutkan berdasarkan id	Halaman master data anggota	[√] Berhasil [] Gagal
2	1-2-3-4-5-6-15-16	User mengurutkan data sesuai dengan pilihan pada tabel	Halaman master data anggota	[√] Berhasil [] Gagal

5.2. Pengujian *Black Box*

Pengujian *black box* adalah pengujian yang bertujuan untuk mengevaluasi operasi perangkat lunak yang sudah dibuat. Yang ditunjukkan dengan fungsi dari perangkat lunak tersebut mengenai pemasukan dan keluaran data yang berjalan sesuai yang diharapkan sebagai mana mestinya.

Adapun hasil dari pengujian black box dapat dilihat pada tabel 5.10.

Tabel 5.10 *Black Box Testing*

NO	USE CASE	AKTOR	SISTEM	STATUS
1	<i>User Login</i>	Jika <i>username, password</i> benar	Tampil dialog konfirmasi login sukses dan masuk halaman <i>home</i>	Berhasil
		Jika <i>field username</i> pada <i>form</i> kosong	Tampil <i>exception</i> “ <i>username belum diisi</i> ”	Berhasil
		Jika <i>field password</i> belum diisi atau salah	Tampil <i>exception</i> “ <i>password belum diisi atau kombinasi salah</i> ”	Berhasil
		Jika <i>field</i> pada <i>form</i> kosong atau salah	Tampil <i>exception</i> yang berkaitan	Berhasil
		Jika menekan tombol ubah password	Akan menampilkan <i>form update password user</i>	Berhasil
		Jika menekan tombol registrasi user baru	Akan menampilkan <i>form registrasi user</i>	Berhasil
		Jika menekan tombol hapus semua	Akan menghapus semua data <i>field</i>	Berhasil
		Jika data yang diisi sudah	Akan <i>redirect</i> pada	Berhasil

		benar	halaman fitur	
2	Registrasi Anggota	Jika <i>field</i> pada <i>form</i> kosong atau salah	Tampil <i>exception</i> yang berkaitan	Berhasil
		Jika nomor telepon diisi berupa desimal, simbol dan minus	Tampil <i>exception</i> tidak diizinkan	Berhasil
		Jika data yang diisi sudah benar	Akan <i>redirect</i> pada halaman fitur	Berhasil
		Jika menekan tombol ubah status	Akan menampilkan <i>form</i> ubah status keanggotaan	Berhasil
		Jika menekan tombol update	Akan menampilkan <i>form update</i> data anggota	Berhasil
		Jika menekan tombol registrasi	Akan menampilkan <i>form</i> registrasi anggota	Berhasil
		Jika menekan tombol hapus semua pada form	Akan menghapus semua data pada <i>field</i>	Berhasil
		Jika memilih ID pada <i>record</i>	Akan menampilkan <i>detail record</i> yang berkaitan	Berhasil
3	Simpanan	Jika menekan tombol ubah	Akan menampilkan <i>form</i> edit data	Berhasil

	Anggota		simpanan	
		Jika menekan tombol hapus	Akan menghapus <i>record</i> yang dipilih	Berhasil
		Jika menekan tombol <i>new data</i>	Akan menampilkan <i>form</i> entri simpanan	Berhasil
		Jika <i>field</i> pada <i>form</i> kosong atau salah	Tampil <i>exception</i> yang berkaitan	Berhasil
		Jika simpanan wajib atau sukarela diisi berupa desimal, simbol dan minus	Tampil <i>exception</i> tidak diizinkan	Berhasil
		Jika data yang diisi sudah benar	Akan <i>redirect</i> pada halaman fitur	Berhasil
		Jika menekan tombol hapus semua pada form	Akan menghapus semua data pada <i>field</i>	Berhasil
		Jika memilih ID pada <i>record</i>	Akan menampilkan <i>detail record</i> yang berkaitan	Berhasil
4	Pinjaman Anggota	Jika menekan tombol hapus	Akan menghapus <i>record</i> yang dipilih	Berhasil
		Jika menekan tombol <i>new data</i>	Akan menampilkan <i>form</i> entri pinjaman	Berhasil
		Jika memilih ID pada <i>record</i>	Akan menampilkan <i>detail record</i> yang	Berhasil

			berkaitan	
		Jika <i>field</i> pada <i>form</i> kosong atau salah	Tampil <i>exception</i> yang berkaitan	Berhasil
		Jika nominal pengajuan diisi berupa desimal, simbol dan minus	Tampil <i>exception</i> tidak diizinkan	Berhasil
		Jika data yang diisi sudah benar	Akan <i>redirect</i> pada halaman fitur	Berhasil
		Jika menekan tombol hapus tanggal pada <i>field</i> tanggal	Akan menghapus data tanggal pada <i>field</i>	Berhasil
		Jika menekan tombol hapus semua pada form	Akan menghapus semua data pada <i>field</i>	Berhasil
5	Angsuran Anggota	Jika menekan detail angsur pada tabel	Menampilkan <i>record</i> angsuran pada id yang dipilih	Berhasil
		Jika menekan nomor pinjaman pada tabel	Menampilkan <i>record</i> pinjaman pada id yang dipilih	Berhasil
		Jika menekan tombol <i>preview all records</i> pada halaman	Menampilkan <i>record</i> transaksi angsuran baik telah lunas dan belum lunas	Berhasil
		Jika menekan tombol	Menampilkan <i>form</i> entri transaksi	Berhasil

		angsur pada tabel	angsuran	
		Jika menekan <i>input field</i> tanggal transaksi angsuran pada <i>form</i>	Menampilkan <i>dialog</i> tanggal untuk menselek data tanggal	Berhasil
		Jika menekan tombol hapus tanggal pada <i>form</i>	Tanggal yang telah diisikan akan terhapus	Berhasil
		Jika menekan tombol <i>entry</i> pada <i>form</i>	Sistem akan mengeksekusi perintah transaksi	Berhasil
		Jika <i>field</i> pada <i>form</i> kosong atau salah	Tampil <i>exception</i> yang berkaitan	Berhasil
6	Manajemen lembaga	Jika menekan tombol ubah	Akan menampilkan <i>form</i> edit data lembaga	Berhasil
		Jika menekan tombol hapus	Akan menghapus <i>record</i> yang dipilih	Berhasil
		Jika menekan tombol <i>new data</i>	Akan menampilkan <i>form</i> entri lembaga	Berhasil
		Jika <i>field</i> pada <i>form</i> kosong atau salah	Tampil <i>exception</i> yang berkaitan	Berhasil
		Jika nomor telepon lembaga diisi berupa desimal, simbol dan minus	Tampil <i>exception</i> tidak diizinkan	Berhasil

		Jika data yang diisi sudah benar	Akan <i>redirect</i> pada halaman fitur	Berhasil
		Jika menekan tombol hapus semua pada form	Akan menghapus semua data pada <i>field</i>	Berhasil
		Jika memilih ID pada <i>record</i>	Akan menampilkan <i>detail record</i> yang berkaitan	Berhasil
7	Cetak laporan transaksi	Jika menekan tombol cetak laporan	Akan menampilkan halaman fitur cetak laporan yang berkaitan	Berhasil
		Jika tidak menselek salah satu di antara lembaga, bulan dan tahun	Laporan yang ditampilkan tidak sesuai	Berhasil
		Jika <i>selectbox</i> lembaga, bulan dan tahun diselek semua	Akan menampilkan laporan sesuai yang diharapkan	Berhasil
8	Manajemen pengurus koperasi	Jika menekan tombol ubah <i>password</i>	Akan menampilkan <i>form update password user</i>	Berhasil
		Jika <i>field password</i> tidak diisikan pada <i>form</i>	Akan menampilkan <i>exception</i>	Berhasil
		Jika menekan tombol registrasi user baru	Akan menampilkan <i>form</i> registrasi user	Berhasil

		Jika <i>field</i> pada <i>form</i> kosong atau salah	Tampil <i>exception</i> yang berkaitan	Berhasil
		Jika data yang diisi sudah benar	Akan <i>redirect</i> pada halaman fitur	Berhasil
		Jika menekan tombol <i>clear</i> pada form	Akan menghapus semua data pada <i>field</i>	Berhasil
9	Master data transaksi	Jika menekan tombol daftar data anggota, daftar data pinjaman atau daftar data simpanan	Akan menampilkan halaman master data sesuai yang diinginkan	Berhasil
		Jika menekan atribut pada menu tabel	Akan mengurutkan data sesuai menu tabel yang dipilih	Berhasil

(Sumber: Hasil Analisis, 2015)

5.3. Implementasi Sistem

Implementasi Sistem adalah tahap pengkodean dari perancangan yang telah dibuat oleh penulis kedalam bahasa pemrograman (*coding*). Pada tahap ini menerapkan perancangan sistem dengan konsep Object Oriented Programming (OOP) yaitu Page Hyper Text Pre-Process (PHP). Sistem informasi simpan pinjam ini di dalamnya penggunaan kode sistem dengan menggunakan pola arsitektur Model View Controller (MVC). Tampilan kode program pada controller dijelaskan pada halaman lampiran.

5.3.1. Pengkodean User Login

Pengkodean untuk user login dapat dilihat pada pengkodean lampiran user login yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.2. Pengkodean Registrasi Anggota

Pengkodean untuk registrasi anggota dapat dilihat pada pengkodean lampiran registrasi anggota yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.3. Pengkodean Transaksi Simpanan Anggota

Pengkodean untuk transaksi simpanan anggotadapat dilihat pada pengkodean lampiran transaksi simpanan anggotadapat yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.4. Pengkodean Transaksi Pinjaman Anggota

Pengkodean untuk transaksi pinjaman anggota dapat dilihat pada pengkodean lampiran transaksi pinjaman anggota yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.5. Pengkodean Transaksi Angsuran Anggota

Pengkodean untuk transaksi angsuran anggota dapat dilihat pada pengkodean lampiran transaksi angsuran anggota yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.6. Pengkodean Manajemen Lembaga

Pengkodean untuk manajemen lembaga dapat dilihat pada pengkodean lampiran manajemen lembaga yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.7. Pengkodean Manajemen Pengurus

Pengkodean untuk manajemen pengurus dapat dilihat pada pengkodean lampiran manajemen pengurus yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.8. Pengkodean Master Data Anggota

Pengkodean untuk master data anggota dapat dilihat pada pengkodean lampiran master data anggota yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.9. Pengkodean Master Data Simpanan

Pengkodean untuk master data simpanan dapat dilihat pada pengkodean lampiran master data simpanan yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.3.10. Pengkodean Master Data Pinjaman

Pengkodean untuk master data pinjaman dapat dilihat pada pengkodean lampiran master data pinjaman yang telah dilampirkan pada halaman lampiran implementasi sistem.

5.4. Pembahasan Sistem

Subbab pembahasan sistem ini menguraikan fitur – fitur yang ada dalam sistem informasi simpan pinjam koperasi. Gambaran umum dari sistem akan dijelaskan pada halaman lampiran.

5.4.1. Fitur Login

Fitur login yang berfungsi sebagai akses untuk masuk kedalam sistem, dapat dilihat pada *screen captured* form login yang telah dilampirkan pada halaman lampiran pembahasan sistem.

5.4.2. Fitur Registrasi Anggota

Fitur registrasi anggota yang berfungsi sebagai manajemen anggota koperasi, dapat dilihat pada *screen captured* registrasi anggota yang telah dilampirkan pada halaman lampiran pembahasan sistem.

5.4.3. Fitur Transaksi Simpanan

Fitur transaksi simpanan yang berfungsi sebagai memproses data transaksi simpanan anggota, dapat dilihat pada *screen captured* transaksi simpanan yang telah dilampirkan pada halaman lampiran pembahasan sistem.

5.4.4. Fitur Transaksi Pinjaman

Fitur transaksi pinjaman yang berfungsi sebagai memproses data transaksi pinjaman anggota, dapat dilihat pada *screen captured* transaksi pinjaman yang telah dilampirkan pada halaman lampiran pembahasan sistem.

5.4.5. Fitur Transaksi Angsuran

Fitur transaksi angsuran yang berfungsi sebagai memproses data transaksi angsuran anggota, dapat dilihat pada *screen captured* transaksi angsuran yang telah dilampirkan pada halaman lampiran pembahasan sistem.

5.4.6. Fitur Manajemen Lembaga

Fitur manajemen lembaga yang berfungsi sebagai manajemen data lembaga, dapat dilihat pada *screen captured* manajemen lembaga yang telah dilampirkan pada halaman lampiran pembahasan sistem.

5.4.7. Fitur Cetak Laporan

Fitur cetak laporan yang berfungsi sebagai memproses data transaksi menjadi laporan berupa *soft-copy* atau *print-out*, dapat dilihat pada *screen captured* cetak laporan yang telah dilampirkan pada halaman lampiran pembahasan sistem.

5.4.8. Fitur Manajemen Pengurus

Fitur manajemen pengurus yang berfungsi sebagai manajemen data pengurus koperasi, dapat dilihat pada *screen captured* manajemen pengurus yang telah dilampirkan pada halaman lampiran pembahasan sistem.

5.4.9. Fitur Master Data Transaksi

Fitur master data transaksi yang berfungsi sebagai memonitoring data transaksi secara keseluruhan, dapat dilihat pada *screen captured* manajemen data transaksi yang telah dilampirkan pada halaman lampiran pembahasan sistem.

BAB 6 KESIMPULAN DAN SARAN

6.1. Kesimpulan

Setelah melakukan rancang bangun sistem, maka diperoleh beberapa kesimpulan. Kesimpulan yang didapat dari penelitian yang telah dilakukan oleh penulis adalah sebagai berikut:

1. Sistem informasi simpan pinjam yang telah dibangun dan dirancang oleh penulis, memberikan fitur – fitur seperti simpanan, pinjaman, angsuran, serta manajemen data lain yang mudah digunakan oleh pihak koperasi.
2. Dengan membandingkan dengan menggunakan sistem sebelumnya yang masih manual, sistem yang dibangun dengan berbasis web ini terbukti lebih efektif dan efisien dalam memproses data transaksi yang berkaitan dengan produk simpan pinjam koperasi.
3. Sistem yang telah dibuat mampu untuk mengatasi pemrosesan laporan transaksi dan bukti transaksi berupa kwitansi yang dibutuhkan untuk pengarsipan data pada koperasi dan pihak lembaga.
4. Terbukti mempermudah dalam pencarian data – data transaksi maupun data lain yang berkaitan dengan koperasi menjadi lebih mudah dan efisien.

6.2. Saran

Berhubung keterbatasan waktu dan kemampuan dalam membangun sistem informasi ini, maka didapatkan saran untuk pengembangan sistem untuk lebih baik. Yaitu diharapkan dapat dikembangkan dengan lebih fleksibel dan memiliki keamanan yang memadai, sehingga dapat diakses secara online dan mobile.

DAFTAR PUSTAKA

- Ade, S. M. (1999). *Aplikasi Web dengan PHP*. Jakarta: Universitas Gunadarma.
- Dharwiyanti, S. d. (2003). *Pengantar Unified Modelling Language (UML)*. Retrieved 04 05, 2012, from Ilmukomputer.com: <http://www.ilmukomputer.com/umum/yanti-uml.php>
- Ferdinand, M. (2009). *Konsep-konsep Dasar Sistem Informasi*. Retrieved from <http://www.scribd.com/doc/43290318/si>
- FithGerald, J. (2002). *Pengertian Sistem*. Jakarta.
- Jogiyanto. (2005). *Analisis dan Desain*. Yogyakarta: Andi, penerbit.
- Jogiyanto. (2005). *Sistem Teknologi Informasi Edisi Ke II*. Yogyakarta: Andi, penerbit.
- Jogiyanto, & Hartono. (2005). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi.
- Kadir, A. d. (2005). *Pengenalan Teknologi Informasi*. Yogyakarta: Andi, penerbit.
- Kadir, A. (1999). *Konsep & Tuntunan Praktis Basis Data*. Yogyakarta: Andi, penerbit.
- Kadir, A. (2003). *Pengenalan Sistem Informasi*. Yogyakarta: Andi, penerbit.
- Kusnadi. (2005). *Peran dan Kebijakan Koperasi*. Jakarta: Penerbit Salemba Empat.
- Kusnadi. (2005). *Peran dan Kebijakan Koperasi*. Jakarta: Salemba Empat.
- Madcoms. (2004). *Aplikasi Program PHP dan MySQL untuk membuat website interaktif*. Yogyakarta: Andi, penerbit.
- McLeod Jr. P, G. S. (2007). *Sistem Informasi Manajemen. Edisi ke-9*. Jakarta: Yuliyanto dan Heri, penerjemah.
- Munawar. (2005). *Pemodelan Visual UML*. Yogyakarta: Graha Ilmu.
- O. H. (2010). *SQL Server 2008 Express*. Yogyakarta: Andi, penerbit.

- P.Coad, E. a. (1991). *Object Oriented design*. Practice hall: Yourdon Press.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak (Pendekatan Praktis)*. Yogyakarta: Andi, penerbit.
- Proboyekti, S. (2008). *Software Process Model I*.
- RS, P. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. Yogyakarta: Andi, penerbit.
- RS, P. (1997). *Rekayasa Perangkat Lunak. Edisi ke-2*. Yogyakarta: LN Harnaningrum, penerjemah.
- S, S. T. (2004). *Pengantar Sistem Informasi*. Yogyakarta: Andi, penerbit.
- Sommerville, I. (2003). *Software Engineering*. Jakarta: Erlangga, penerbit.
- Syafii, M. (2006). *Membangun Aplikasi Berbasis PHP dan MySQL*. Yogyakarta: Andi, penerbit.
- Undang-Undang Perkoperasian, UU no.25 (1992).
- Zaki, A. (2008). *36 menit belajar komputer: PHP dan Mysql*. Jakarta: PT Elex Media Komputindo.

LAMPIRAN

Lampiran Implementasi Sistem

1. Pengkodean User Login

Lampiran 1 Pengkodean User Login

```
public function cek2( $user, $pass ) {
    $res=mysql_query("select * from tblmultiuser where username='$user' and
password='$pass'");
    $data=mysql_fetch_array($res);
    $sidnya=$data['id_user'];
    $nm=$data['nama_lengkap'];
    $name=$data['username'];
    $word=$data['password'];
    $type=$data['type_user'];
    if($user==$name && $pass==$word){
        if($type=="admin"){
            $_SESSION['id_user']=$sidnya;
            $_SESSION['nama_lengkap']=$nm;
            $_SESSION['username']=$name;
            $_SESSION['type_user']=$type;
            echo '<script>
var conn=confirm("Administrator login sukses! Lanjutkan
ke halaman admin?");
window.location.assign("index.php?operasi=home");
if(conn==true){
window.location.assign("index.php?operasi=admin");
}
```

```

        </script>
        ;
        } else{
        $_SESSION['id_user']=$idnya;
        $_SESSION['nama_lengkap']=$nm;
        $_SESSION['username']=$name;
        $_SESSION['type_user']=$type;
        echo '<script>
        var conn=confirm("Berhasil login! klik OK untuk lanjut.");
        window.location.assign("index.php?operasi=logout");
        if(conn==true){
        window.location.assign("index.php?operasi=home");
        }
        </script>
        ;
        }
    }
    elseif ( !isset($user) || empty($user) ) {
        echo "<div style ='font:28px Segoe UI;color:#ff0000' align='center'
        > Gagal Login! </div><div style ='font:18px Segoe Print;color:#000000'
        align='center'>username belum diisi!</b></div>";
    }
    else {
        echo "<div style ='font:28px Segoe UI;color:#ff0000' align='center'
        > Gagal Login! </div><div style ='font:18px Segoe Print;color:#000000'
        align='center'>password belum diisi, atau kombinasi salah!!</b></div>";
    }
}

//----- function cek login
public function regisUser( $nm, $user, $pass1, $pass2, $type, $tgl ) {
    if($pass1==$pass2){

```

```
        $pass=md5($pass1);
        $hasil=mysql_query("insert into tblmultiuser
values('$nm','$user','$pass','$type','$tgl')");
        if($hasil){
            echo '<script>
var conn=confirm("Berhasil Register User! Kembali ke menu admin?");
window.location.assign("index.php?operasi=home");
if(conn==true){
    window.location.assign("index.php?operasi=admin");
}
</script>
';
        }
    }
}
```

(Sumber: Hasil Analisis, 2015)

2. Pengkodean Registrasi Anggota

Lampiran 2 Pengkodean Registrasi Anggota *Entry*

```
public function registrasiAnggota() {
    $title = 'Register';

    $namax = "";
    $salamatx = "";
    $telpx = "";
    $noktpx = "";
    $tgllahirx = "";
    $tmptlahirx = "";
    $genderx = "";
    $emailx = "
```

```
$lembagax = "";
$stglx = "";
$spokokx = "";
$petugasx = "";
$ketx = "";
$sdstatusx = "";
$tanggal2x = "";
$errors = array();

if ( isset($_POST['form-submitted']) ) {

    $namax      = isset($_POST['nama']) ? $_POST['nama'] :NULL;
    $alamatx    = isset($_POST['alamat']) ? $_POST['alamat'] :NULL;
    $telpx      = isset($_POST['telp']) ? $_POST['telp'] :NULL;
    $noktpx     = isset($_POST['noktp']) ? $_POST['noktp'] :NULL;
    $tgllahirx  = isset($_POST['tgllahir']) ? $_POST['tgllahir'] :NULL;
    $tmptlahirx = isset($_POST['tempatlahir']) ? $_POST['tempatlahir'] :NULL;
    $genderx    = isset($_POST['gender']) ? $_POST['gender'] :NULL;
    $emailx     = isset($_POST['email']) ? $_POST['email'] :NULL;
    $lembagax   = isset($_POST['lembaga']) ? $_POST['lembaga'] :NULL;
    $stglx      = isset($_POST['tanggal']) ? $_POST['tanggal'] :NULL;
    $spokokx    = isset($_POST['spokok'])? $_POST['spokok'] :NULL;
    $petugasx   = isset($_POST['petugas'])? $_POST['petugas'] :NULL;
    $ketx       = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;
    $sdstatusx  = isset($_POST['idstatus'])? $_POST['idstatus'] :NULL;
    $tanggal2x  = isset($_POST['tanggal2'])? $_POST['tanggal2'] :NULL;

    try {
        $this->kons_m_anggot_service->createNewMember( $namax, $alamatx, $telpx,
        $noktpx,      $tgllahirx, $tmptlahirx, $genderx, $emailx, $lembagax,$stglx, $spokokx,
        $petugasx, $ketx, $sdstatusx, $tanggal2x );
    }
}
```

```
$this->redirect('index.php?operasi=anggota');  
return;  
} catch (ValidationException $e) {  
    $errors = $e->getErrors();  
}  
}  
include 'view/v_anggota_entr_form.php';  
}
```

(Sumber: Hasil Analisis, 2015)

Lampiran 3 Pengkodean Registrasi Anggota *Update Status*

```
public function updateStatusAnggota() {  
    $title = 'Update';  
    $errors = array();  
    $id = isset($_GET['id'])?$_GET['id']:NULL;  
    if ( !$id ) {  
        throw new Exception('Internal error.');    }  
    $updateMember = $this->kons_m_anggot_service->getUpdate($id);  
    //=====  
    if ( isset($_POST['form-submitted']) ) {  
        $idnyax      = isset($_POST['idnya']) ? $_POST['idnya'] :NULL;  
        $statusubahx = isset($_POST['statusubah'])? $_POST['statusubah'] :NULL;  
        $tanggalberhentix= isset($_POST['tanggalberhenti'])? $_POST['tanggalberhenti']  
:NULL;  
        $petugasx    = isset($_POST['petugas'])? $_POST['petugas'] :NULL;  
        $ketx        = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;  
        try {  
            $this->kons_m_anggot_service->createupdateStatMember( $idnyax,  
$statusubahx,$tanggalberhentix, $petugasx, $ketx );
```

```
$this->redirect('index.php?operasi=anggota');  
return;  
} catch (ValidationException $e) {  
    $errors = $e->getErrors();  
}  
}  
include 'view/v_anggota_updet_status.php';  
}
```

(Sumber: Hasil Analisis, 2015)

Lampiran 4 Pengkodean Registrasi Anggota *Edit-Update*

```
public function updateAnggota() {  
    $title = 'Update';  
    $errors = array();  
    $sid = isset($_GET['id'])?$_GET['id']:NULL;  
    if ( !$sid ) {  
        throw new Exception('Internal error.');    }  
    $updatemember = $this->kons_m_anggot_service->getUpdate($sid);  
    //=====  
    if ( isset($_POST['form-submitted']) ) {  
        $sidnyax      = isset($_POST['idnya']) ? $_POST['idnya'] :NULL;  
        $statusubahx  = isset($_POST['statusubah'])? $_POST['statusubah'] :NULL;  
        $alamatbarux  = isset($_POST['alamatbaru'])? $_POST['alamatbaru'] :NULL;  
        $telpbarux    = isset($_POST['telpbaru'])? $_POST['telpbaru'] :NULL;  
        $emailbarux   = isset($_POST['emailbaru'])? $_POST['emailbaru'] :NULL;  
        $petugasx     = isset($_POST['petugas'])? $_POST['petugas'] :NULL;  
        $ketx         = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;  
        $tanggalberhentix= isset($_POST['tanggalberhenti'])? $_POST['tanggalberhenti']  
:NULL;
```

```
try {
    $this->kons_m_anggot_service->createupdateMember( $idnyax,
    $statusbahx,$alamatbarux, $telpbarux, $emailbarux, $petugasx, $ketx,$tanggalberhentix );
    $this->redirect('index.php?operasi=anggota');
    return;
} catch (ValidationException $e) {
    $errors = $e->getErrors();
}
}
include 'view/v_anggota_edit_update.php';
}
```

(Sumber: Hasil Analisis, 2015)

3. Pengkodean Transaksi Simpanan Anggota

Lampiran 5 Pengkodean Simpanan *Entry*

```
public function entriSimpan() {
    $title = 'Entry';
    $namax = "";
    $swajibx = "";
    $ssukax = "";
    $stabungx = "";
    $totalx = "";
    $stglx = "";
    $petugasx = "";
    $ketx = "";

    $errors = array();
    if ( isset($_POST['form-submitted']) ) {
        $namax      = isset($_POST['nama']) ? $_POST['nama'] :NULL;
        $swajibx    = isset($_POST['sw'])? $_POST['sw'] :NULL;
```

```
$ssukax      = isset($_POST['ss'])? $_POST['ss'] :NULL;
$stabungx   = isset($_POST['st'])? $_POST['st'] :NULL;
$totalx     = isset($_POST['ts'])? $_POST['ts'] :NULL;
$tglx       = isset($_POST['tanggal'])? $_POST['tanggal'] :NULL;
$petugasx   = isset($_POST['petugas'])? $_POST['petugas'] :NULL;
$ketx       = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;

try {
    $this->kons_m_simp_service->createNewSaving( $namax, $swajibx, $ssukax,
                                                $stabungx, $totalx, $tglx, $petugasx, $ketx );
    $this->redirect('index.php?operasi=listsimpan');
    return;
} catch (ValidationException $e) {
    $errors = $e->getErrors();
}
}
include 'view/v_simp_entr_form.php';
}
```

(Sumber: Hasil Analisis, 2015)

Lampiran 6 Pengkodean Simpanan *Edit*

```
public function updateSimpan() {
    $title = 'Update';
    $errors = array();

    $id = isset($_GET['id'])?$_GET['id']:NULL;
    if ( !$id ) {
        throw new Exception('Internal error.');
```



```
//=====
if ( isset($_POST['form-submitted']) ) {

    $idnyax    = isset($_POST['idnya']) ? $_POST['idnya'] :NULL;
    $swajibx   = isset($_POST['sw'])? $_POST['sw'] :NULL;
    $ssukax    = isset($_POST['ss'])? $_POST['ss'] :NULL;
    $stabungx  = isset($_POST['st'])? $_POST['st'] :NULL;
    $totalx    = isset($_POST['ts'])? $_POST['ts'] :NULL;
    $ketx      = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;
    try {
        $this->kons_m_simp_service->createupdateSaving( $idnyax,
        $swajibx,$ssukax,$stabungx, $totalx, $ketx );
        $this->redirect('index.php?operasi=listsimpan');
        return;
    } catch (ValidationException $e) {
        $errors = $e->getErrors();
    }
}
include 'view/v_simp_updet_form.php';
}
```

(Sumber: Hasil Analisis, 2015)

4. Pengkodean Transaksi Pinjaman Anggota

Lampiran 7 Pengkodean Pinjaman

```
public function entriPinjam() {
    $title = 'Entry';
    $ax = "";
    $bx = "";
    $cx = "";
}
```

```
$dx = "";
$ex = "";
$fx = "";
$gx = "";
$ix = "";
$jx = "";
$kx = "";
$lx = "";
$mx = "";
$errors = array();

if ( isset($_POST['form-submitted']) ) {

    $ax      = isset($_POST['nama']) ? $_POST['nama'] :NULL;
    $bx      = isset($_POST['tanggal'])? $_POST['tanggal'] :NULL;
    $cx      = isset($_POST['jenis'])? $_POST['jenis'] :NULL;
    $dx      = isset($_POST['barang'])? $_POST['barang'] :NULL;
    $ex      = isset($_POST['nominal'])? $_POST['nominal'] :NULL;
    $fx      = isset($_POST['freq'])? $_POST['freq'] :NULL;
    $gx      = isset($_POST['jasa'])? $_POST['jasa'] :NULL;
    $hx      = isset($_POST['bunga'])? $_POST['bunga'] :NULL;
    $ix      = isset($_POST['cicil1'])? $_POST['cicil1'] :NULL;
    $jx      = isset($_POST['cicil2'])? $_POST['cicil2'] :NULL;
    $kx      = isset($_POST['statuspinjam'])? $_POST['statuspinjam'] :NULL;
    $lx      = isset($_POST['keterangan'])? $_POST['keterangan'] :NULL;
    $mx      = isset($_POST['petugas'])? $_POST['petugas'] :NULL;

    try {
        $this->kons_m_pinj_service->createNewLoan( $ax, $bx, $cx, $dx, $ex, $fx,$gx,
        $hx, $ix, $jx, $kx, $lx, $mx );
        $this->redirect('index.php?operasi=listpinjam');
```

```
        return;
    } catch (ValidationException $e) {
        $errors = $e->getErrors();
    }
}
include 'view/v_pinj_entr_form.php';
}
```

(Sumber: Hasil Analisis, 2015)

5. Pengkodean Transaksi Angsuran Anggota

Lampiran 8 Pengkodean Angsuran

```
public function updateAngsur() {
    $title = 'Update';
    $errors = array();

    $sid = isset($_GET['idpinjam'])?$_GET['idpinjam']:NULL;
    if ( !$sid ) {
        throw new Exception('Internal error.');
```

=====

```
    if ( isset($_POST['form-submitted']) ) {

        $sidnyax      = isset($_POST['idnya']) ? $_POST['idnya'] :NULL;
        $idpinjx      = isset($_POST['idpinj']) ? $_POST['idpinj'] :NULL;
        $tglnewx      = isset($_POST['tglnew']) ? $_POST['tglnew'] :NULL;
        $cicilx       = isset($_POST['cicil']) ? $_POST['cicil'] :NULL;
        $angsurkrngx  = isset($_POST['angsurkrng']) ? $_POST['angsurkrng'] :NULL;
        $freqangsurx  = isset($_POST['freqangsur']) ? $_POST['freqangsur'] :NULL;
        try {
```

```
$this->kons_m_angsu_service->createupdateInstallmentPayments( $idnyax,  
$idpinjx, $tglnewx, $scicilx, $angsurskrngx, $freqangsurx );  
$this->redirect('index.php?operasi=listangsur');  
return;  
} catch (ValidationException $e) {  
$errors = $e->getErrors();  
}  
}  
include 'view/v_angsu_entri_form.php';  
}
```

(Sumber: Hasil Analisis, 2015)

6. Pengkodean Manajemen Lembaga

Lampiran 9 Pengkodean Lembaga *Entry*

```
public function entriLembaga() {  
$title = 'Entry';  
$namalbgx = "";  
$alamatx = "";  
$telpx = "";  
$emailx = "";  
$petugasx = "";  
  
$errors = array();  
  
if ( isset($_POST['form-submitted']) ) {  
  
$namalbgx = isset($_POST['namalbg']) ? $_POST['namalbg'] :NULL;  
$alamatx = isset($_POST['alamat']) ? $_POST['alamat'] :NULL;  
$telpx = isset($_POST['telp']) ? $_POST['telp'] :NULL;  
$emailx = isset($_POST['email']) ? $_POST['email'] :NULL;
```

```
$petugasx = isset($_POST['petugas'])? $_POST['petugas'] :NULL;

try {
    $this->kons_m lembg_service->createNewLembg( $namalbgx, $alamatx, $telpx,
        $emailx, $petugasx );
    $this->redirect('index.php?operasi=lembaga');
    return;
} catch (ValidationException $e) {
    $errors = $e->getErrors();
}
}
include 'view/v_lembaga_entri_form.php';
}
```

(Sumber: Hasil Analisis, 2015)

Lampiran 10 Pengkodean Lembaga *Update*

```
public function updateLembaga() {
    $title = 'Update';
    $errors = array();

    $sid = isset($_GET['id'])?$_GET['id']:NULL;
    if ( !$sid ) {
        throw new Exception('Internal error.');
```

```
    }
    $update = $this->kons_m lembg_service->getUpdate($sid);

    //=====
    if ( isset($_POST['form-submitted']) ) {

        $sidx = isset($_POST['id']) ? $_POST['id'] :NULL;
```

```
$telpx = isset($_POST['telp']) ? $_POST['telp'] :NULL;
$emailx = isset($_POST['email'])? $_POST['email'] :NULL;

try {
    $this->kons_m_lembg_service->createupdateLembg( $idx, $telpx, $emailx );
    $this->redirect('index.php?operasi=lembaga');
    return;
} catch (ValidationException $e) {
    $errors = $e->getErrors();
}
}
include 'view/v_lembaga_updet_form.php';
}
```

(Sumber: Hasil Analisis, 2015)

7. Pengkodean Manajemen Pengurus

Lampiran 11 Pengkodean Pengurus *Entry*

```
public function userRegis() {
    $title = 'Register';
    $nmX = "";
    $userX = "";
    $pass1x = "";
    $pass2x = "";
    $stypex = "";
    $stglx = "";

    $errors = array();

    if ( isset($_POST['form-submitted']) ) {
```

```
        $nmix  =$_POST['nm'];
        $userx  =$_POST['user'];
        $pass1x=$_POST['pass1'];
        $pass2x=$_POST['pass2'];
        $typex  =$_POST['type'];
        $tglx   =date('Y-m-d H:i:s');

    try {
        $this->kons_m_user_service->createNewUser( $nmix, $userx, $pass1x, $pass2x,
    $typex, $tglx );
        //$this->redirect('index.php?operasi=admin');
        return;
    } catch (ValidationException $e) {
        $errors = $e->getErrors();
    }
}
include 'view/v_admin_entri_form.php';
}
```

(Sumber: Hasil Analisis, 2015)

Lampiran 12 Pengkodean Pengurus *Update Password*

```
public function userUpdate() {
    $title = 'Update';
    $errors = array();

    $id = isset($_GET['id'])?$_GET['id']:NULL;
    if ( !$id ) {
        throw new Exception('Internal error.');
```

```
if ( isset($_POST['form-submitted']) ) {  
  
    $iduserx      = isset($_POST['iduser']) ? $_POST['iduser'] :NULL;  
    $passnewx    = isset($_POST['passnew'])? $_POST['passnew'] :NULL;  
  
    try {  
        $this->kons_m_user_service->createupdateUser( $iduserx, $passnewx );  
        $this->redirect('index.php?operasi=admin');  
        return;  
    } catch (ValidationException $e) {  
        $errors = $e->getErrors();  
    }  
}  
include 'view/v_admin_update_form.php';  
}
```

(Sumber: Hasil Analisis, 2015)

8. Pengkodean Master Data Anggota

Lampiran 13 Pengkodean Master Data Anggota

```
public function selectMasterDataAnggota($order) {  
    if ( !isset($order) ) {  
        $order = "idanggota";  
    }  
    $dbOrder = mysql_real_escape_string($order);  
    $dbres = mysql_query("SELECT * FROM tblanggota, tblstatusanggota, tbllembaga  
where tblanggota.idstatus = tblstatusanggota.idstatus and tblanggota.idlembaga =  
tbllembaga.idlembaga ORDER BY $dbOrder ASC");  
    $masteranggota = array();  
    while ( ($obj = mysql_fetch_object($dbres)) != NULL ) {  
        $masteranggota[] = $obj;  
    }  
}
```



```
}  
return $masteranggota;  
}
```

(Sumber: Hasil Analisis, 2015)

9. Pengkodean Master Data Simpanan

Lampiran 14 Pengkodean Master Data Simpanan

```
public function selectMasterDataSimpan($order) {  
    if ( !isset($order) ) {  
        $order = "idsimpan";  
    }  
    $dbOrder = mysql_real_escape_string($order);  
    $dbres = mysql_query("SELECT * FROM tblsimpanan,tblanggota,tblmultiuser where  
tblsimpanan.idanggota = tblanggota.idanggota and tblsimpanan.idpengurus =  
tblmultiuser.id_user ORDER BY $dbOrder ASC");  
    $mastersimpan = array();  
    while ( ($obj = mysql_fetch_object($dbres)) != NULL ) {  
        $mastersimpan[] = $obj;  
    }  
    return $mastersimpan;  
}
```

(Sumber: Hasil Analisis, 2015)

10. Pengkodean Master Data Pinjaman

Lampiran 15 Pengkodean Master Data Pinjaman

```
public function selectMasterDataPinjam($order) {  
    if ( !isset($order) ) {  
        $order = "idpinjam";  
    }  
}
```

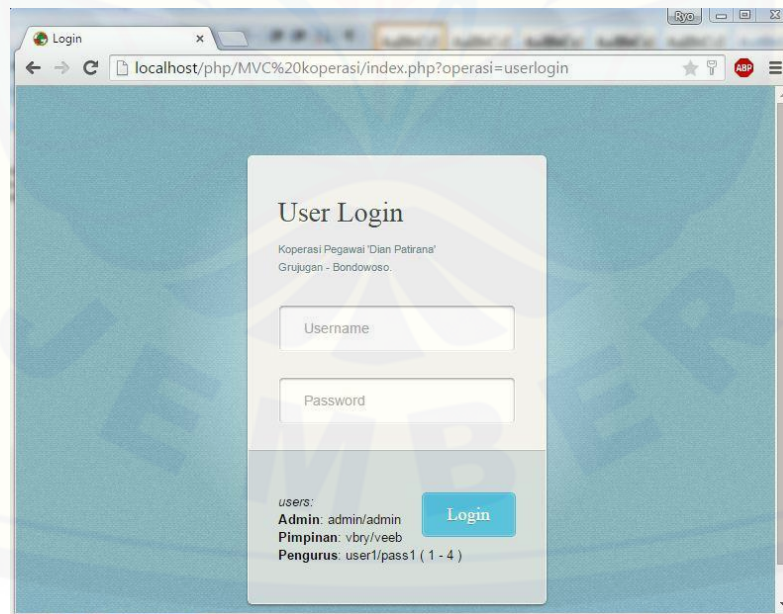
```
$dbOrder = mysql_real_escape_string($order);  
$dbres = mysql_query("select * from tblpinjaman, tblbarang, tblanggota, tblmultiuser  
where tblpinjaman.idanggota = tblanggota.idanggota and tblpinjaman.idbarang =  
tblbarang.idbarang and tblpinjaman.idpengurus = tblmultiuser.id_user and tglpinjam is not  
null ORDER BY $dbOrder ASC");  
$masterpijam = array();  
while ( ($obj = mysql_fetch_object($dbres)) != NULL ) {  
    $masterpijam[] = $obj;  
}  
return $masterpijam;  
}
```

(Sumber: Hasil Analisis, 2015)

Lampiran Pembahasan Sistem

11. *Screen Captured* Form Login

Lampiran 16 *Screen Captured* Form Login



(Sumber: Hasil Analisis, 2015)

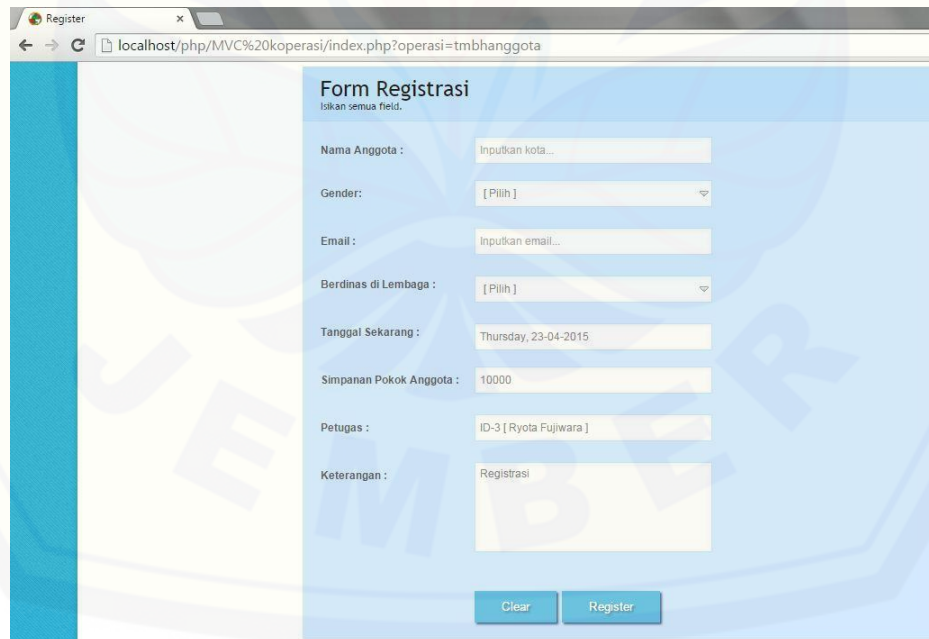
12. *Screen Captured* Fitur Registrasi Anggota

Lampiran 17 *Screen Captured* Registrasi Anggota



(Sumber: Hasil Analisis, 2015)

Lampiran 18 *Screen Captured* Registrasi Anggota Form



(Sumber: Hasil Analisis, 2015)

13. *Screen Captured* Fitur Transaksi Simpanan

Lampiran 19 *Screen Captured* Transaksi Simpanan

No Transaksi	Unit Simpanan Koperasi (USP)						Ubah	Hapus
	Anggota	Tanggal Transaksi	Simpanan Pokok	Simpanan Wajib	Simpanan Sukarela	Simpanan Anggota		
SIMP1	Andi	1 February 2015	Rp. 10.000,00	-	-	-		
SIMP2	Budi	1 February 2015	Rp. 10.000,00	-	-	-		
SIMP3	Cindy	1 February 2015	Rp. 10.000,00	-	-	-		
SIMP4	Andi	8 February 2015	-	Rp. 10.000,00	-	-		
SIMP5	Andi	26 February 2015	-	Rp. 5.000,00	Rp. 1.000,00	-		
SIMP6	Cindy	17 February 2015	-	Rp. 10.000,00	Rp. 1.500,00	-		
SIMP7	Andi	21 February 2015	-	Rp. 10.000,00	Rp. 10.000,00	-		
SIMP8	Budi	11 February 2015	-	Rp. 10.000,00	Rp. 10.000,00	-		
SIMP9	Andi	6 February 2015	-	Rp. 10.000,00	Rp. 10.000,00	-		
SIMP10	Budi	18 February 2015	-	Rp. 10.000,00	Rp. 10.000,00	-		
SIMP11	Doni	17 March 2015	Rp. 10.000,00	-	-	-		
SIMP12	Andi	17 February 2015	-	-	-	Rp. 20.000,00		

Sumber: Hasil Analisis, 2015)

Lampiran 20 *Screen Captured* Transaksi Simpanan Form

Form Entri Transaksi Simpanan
Isikan semua field.

Identitas Anggota : [Pilih] ▾

Simpanan Wajib Anggota : 0 *Lewati jika perlu

Simpanan Sukarela Anggota : 0 *Contoh: 15000

Simpanan Tabungan Anggota : 0 *Contoh: 15000

Total Simpanan Anggota : empty

Tanggal Transaksi : Pilih tanggal...

Petugas : ID-3 [Ryota Fujiwara]

Keterangan : Tidak ada keterangan

(Sumber: Hasil Analisis, 2015)

14. *Screen Captured* Fitur Transaksi Pinjaman

Lampiran 21 *Screen Captured* Transaksi Pinjaman

Unit Pinjaman Koperasi (USP)									
No	Tanggal	Pinjaman	Nominal	Frekuensi	Jasa	Bunga Pinjaman	Kewajiban cicilan	Status	Hapus
PINJ4	17 February 2015	Uang	Rp. 10.000.000,00	10x	1.5%	Rp. 150.000,00	Rp. 1.150.000,00	Belum Lunas	
PINJ3	17 February 2015	Uang	Rp. 500.000,00	5x	1.5%	Rp. 7.500,00	Rp. 107.500,00	Belum Lunas	
PINJ2	17 February 2015	Elektronik	Rp. 250.000,00	5x	1.5%	Rp. 3.750,00	Rp. 53.750,00	Belum Lunas	

(Sumber: Hasil Analisis, 2015)

Lampiran 22 *Screen Captured* Transaksi Pinjaman Form

Form Entri Transaksi Pinjaman
Isikan semua field.

Identitas Anggota :

Tanggal Pengajuan:

Jenis Pinjaman:

Pinjaman :

Nominal Pengajuan: *Contoh: 15000

Frekuensi Pinjaman:

Dikenakan Jasa : %

Keterangan :

(Sumber: Hasil Analisis, 2015)

15. *Screen Captured* Fitur Transaksi Angsuran

Lampiran 23 *Screen Captured* Transaksi Angsuran

Detail Angsur	Unit Angsuran Koperasi (USP)							
	No transaksi	No pinjaman	Tanggal pinjam	Angsuran terakhir	Cicilan wajib	Frekuensi	Angsur ke:	Angsur
Trans ANG2	ANG2	PINJ2	17 February 2015	2 March 2015	Rp. 50.000,00	5 x	4 x	
Trans ANG8	ANG8	PINJ3	17 February 2015	19 February 2015	Rp. 100.000,00	5 x	1 x	
Trans ANG11	ANG11	PINJ4	17 February 2015	17 February 2015	Rp. 1.000.000,00	10 x	2 x	

[preview all records](#)

(Sumber: Hasil Analisis, 2015)

Lampiran 24 *Screen Captured* Transaksi Angsuran Form

Form Entri Transaksi Angsuran
Isikan semua field.

No Pinjaman : 2
Cicilan : 50000
Frekuensi angsuran : 5 x
Angsuran terakhir ke : 4
Angsuran sekarang ke : 5
Tgl angsur sebelumnya : 2 March 2015

Tgl angsur sekarang :

Petugas :

(Sumber: Hasil Analisis, 2015)

16. *Screen Captured* Fitur Manajemen Lembaga

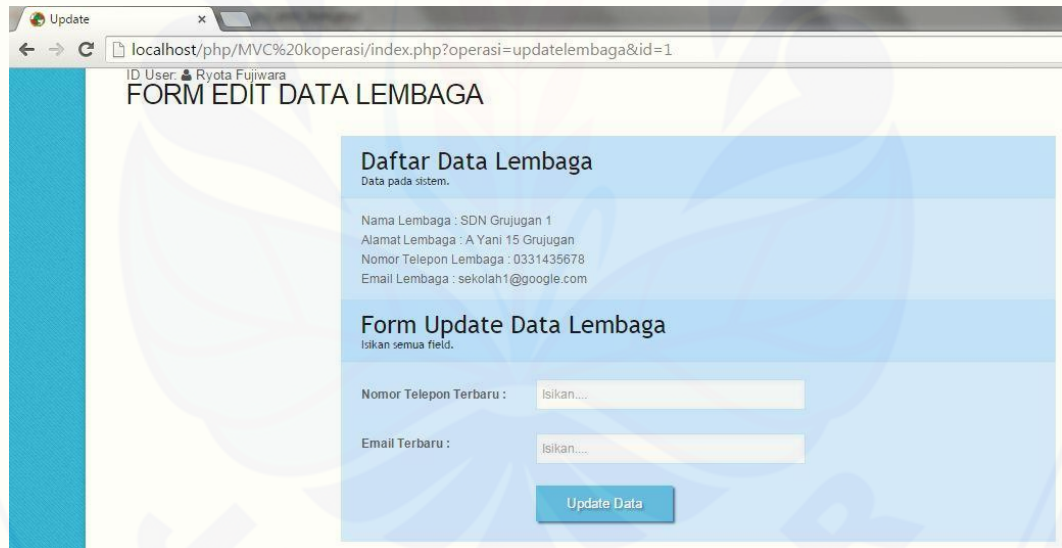
Lampiran 25 *Screen Captured* Manajemen Lembaga



Daftar Lembaga						
ID Lembaga	Nama Lembaga	Alamat Lembaga	No Telepon Lembaga	E-mail Lembaga	Ubah	Hapus
LBG1	SDN Grujugan 1	A Yani 15 Grujugan	0331435678	sekolah1@google.com		
LBG2	SDN Grujugan 2	A Yani 12 Grujugan	0331435678	sekolah2@google.com		
LBG3	SDN Grujugan 3	A Yani 18 Grujugan	0331435678	sekolah1@google.com		

(Sumber: Hasil Analisis, 2015)

Lampiran 26 *Screen Captured* Manajemen Lembaga Form



Daftar Data Lembaga
Data pada sistem.

Nama Lembaga : SDN Grujugan 1
Alamat Lembaga : A Yani 15 Grujugan
Nomor Telepon Lembaga : 0331435678
Email Lembaga : sekolah1@google.com

Form Update Data Lembaga
Isikan semua field.

Nomor Telepon Terbaru :

Email Terbaru :

(Sumber: Hasil Analisis, 2015)

17. *Screen Captured* Fitur Cetak Laporan

Lampiran 27 *Screen Captured* Cetak Laporan



(Sumber: Hasil Analisis, 2015)

Lampiran 28 *Screen Captured* Cetak Laporan Preview



(Sumber: Hasil Analisis, 2015)

18. *Screen Captured* Fitur Manajemen Pengurus

Lampiran 29 *Screen Captured* Manajemen Pengurus

Daftar user pada sistem					
ID	Nama lengkap	Type	Username	Password (MD5)	Ubah password
1	Administrator	admin	admin	21232f297a57a5a743894a0e4a801fc3	
2	Pimpinan koperasi	pimpinan	vbry	907b8c5f11733ccc91e3fca9316fa4bd	
3	Ryota Fujiwara	pengurus	user	1a1dc91c907325c69271ddf0c944bc72	
4	Petugas alpha	pengurus	user1	a722c63db8ec8625af6cf71cb8c2d939	
5	Petugas beta	pengurus	user2	c1572d05424d0ecb2a65ec6a62aeacbf	
6	Petugas charlie	pengurus	user3	3afc79b597f88a72528e864cf81856d2	
7	Petugas delta	pengurus	user4	fc2921d9057ac44e549efaf0048b2512	

(Sumber: Hasil Analisis, 2015)

Lampiran 30 *Screen Captured* Manajemen Pengurus Form

ID User: Administrator

FORM REGISTRASI USER

Isikan semua field.

Nama User :

Username :

Password :

Konfirmasi Password :

Type User :

(Sumber: Hasil Analisis, 2015)

19. *Screen Captured* Fitur Master Data Transaksi

Lampiran 31 *Screen Captured* Master Data Transaksi



(Sumber: Hasil Analisis, 2015)

Lampiran 32 *Screen Captured* Master Data Transaksi *Preview*

Tabel Master Anggota Koperasi											
ID Pinjaman	Anggota	Jenis pinjaman	Bentuk	Nominal	Frekuensi	Status	Petugas	Bunga Pinjaman	Kewajiban cicilan	Status	Tanggal
PINJ1	Andi	USP	Uang	Rp. 10.000.000,00	5x	Lunas	Ryota Fujiwara	Rp. 150.000,00	Rp. 2.150.000,00	Lunas	5 February 2015
PINJ2	Budi	USP	Elektronik	Rp. 250.000,00	5x	Belum Lunas	Ryota Fujiwara	Rp. 3.750,00	Rp. 53.750,00	Belum Lunas	17 February 2015
PINJ3	Cindy	USP	Uang	Rp. 500.000,00	5x	Belum Lunas	Ryota Fujiwara	Rp. 7.500,00	Rp. 107.500,00	Belum Lunas	17 February 2015
PINJ4	Andi	USP	Uang	Rp. 10.000.000,00	10x	Belum Lunas	Ryota Fujiwara	Rp. 150.000,00	Rp. 1.150.000,00	Belum Lunas	17 February 2015

(Sumber: Hasil Analisis, 2015)