



**SISTEM PENDUKUNG KEPUTUSAN PEMBERIAN PINJAMAN PADA  
KOPERASI SIMPAN PINJAM SYARIAH BMT UGT SIDOGIRI  
MENGUNAKAN METODE *WEIGHTED PRODUCT***

**SKRIPSI**

**Oleh**

**Anifatul Baroroh**

**NIM 112410101016**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2015**



**SISTEM PENDUKUNG KEPUTUSAN PEMBERIAN PINJAMAN PADA  
KOPERASI SIMPAN PINJAM SYARIAH BMT UGT SIDOGIRI  
MENGUNAKAN METODE *WEIGHTED PRODUCT***

**SKRIPSI**

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

**Oleh**

**Anifatul Baroroh**

**NIM 112410101016**

**PROGRAM STUDI SISTEM INFORMASI**

**UNIVERSITAS JEMBER**

**2015**

## PERSEMBAHAN

Alhamdulillah, puji dan syukur kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya, sehingga skripsi ini dapat diselesaikan. Dengan penuh kerendahan hati, skripsi ini dipersembahkan untuk:

1. Ibunda Rohanah dan Ayahanda Haidi yang selalu mendoakan dan mencurahkan kasih sayang;
2. Kakak Vika Anggraeni yang selalu memberi semangat dan keceriaan;
3. Guru-guru dari Sekolah Dasar hingga Perguruan Tinggi yang mendidik dan memberikan ilmu yang bermanfaat;
4. Almamater Program Studi Sistem Informasi Universitas Jember;
5. Hengki Irawan yang selalu menciptakan kompetisi untuk memotivasi dalam meraih cita-cita;
6. Sahabat-sahabat yang saling mendukung dari awal perjuangan;
7. Keluarga besar NEFOTION.

## MOTTO

Semua orang akan menghadapi rintangan, termasuk mereka yang sudah sukses saat ini. Bedanya, orang yang sukses mampu mengatasi rintangan itu. (*Rahmad Mr. Power, 2014*)<sup>1</sup>



---

<sup>1</sup> <http://www.motivasi-islami.com/kata-kata-motivasi/>

## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Anifatul Baroroh

NIM : 112410101016

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri Menggunakan Metode *Weighted Product*”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan 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 adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika dikemudian hari pernyataan ini tidak benar.

Jember, 25 Juni 2015

Yang menyatakan,

Anifatul Baroroh

112410101016

## PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri Menggunakan Metode *Weighted Product*”, telah diuji dan disahkan pada:

Hari, tanggal : Kamis, 25 Juni 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

PembimbingI,

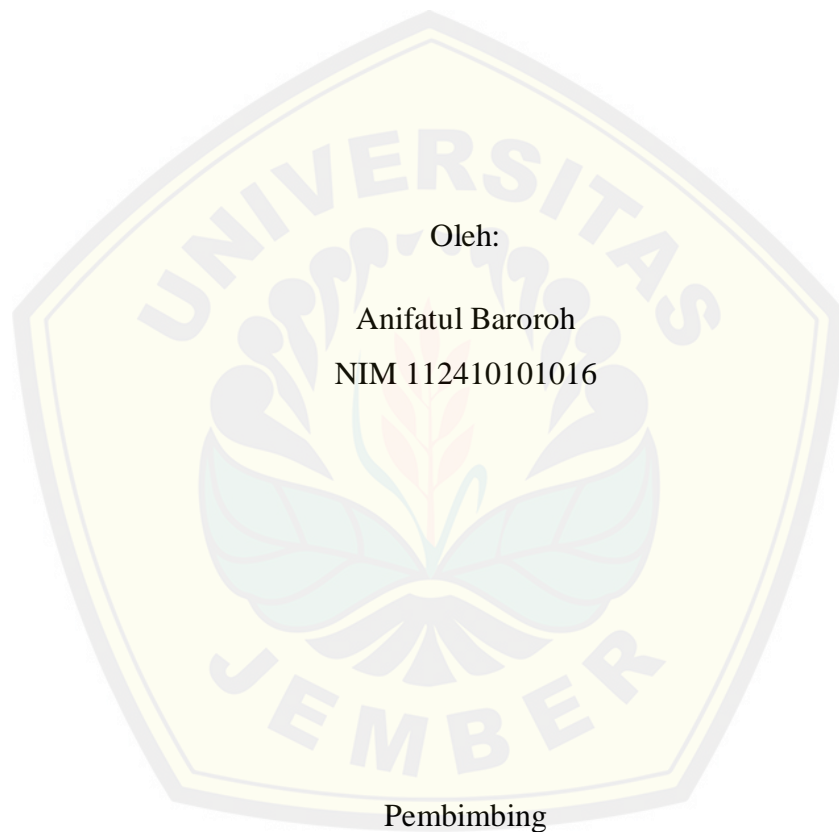
PembimbingII,

Drs. Antonius Cahya P, M.App.,Sc  
NIP. 196909281993021001

Nelly Oktavia Adiwijaya S.Si.,MT  
NIP. 198410242009122008

**SKRIPSI**

**SISTEM PENDUKUNG KEPUTUSAN PEMBERIAN PINJAMAN PADA  
KOPERASI SIMPAN PINJAM SYARIAH BMT UGT SIDOGIRI  
MENGUNAKAN METODE *WEIGHTED PRODUCT***



Dosen Pembimbing Utama : Drs. Antonius Cahya P, M.App.,Sc  
Dosen Pembimbing Anggota : Nelly Oktavia Adiwijaya S.Si.,MT

## PENGESAHAN

Skripsi berjudul “Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri Menggunakan Metode *Weighted Product*” telah diuji dan disahkan pada:

Hari, tanggal : Kamis, 25 Juni 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji:

Penguji I,

Penguji II,

Anang Andrianto, ST.,M.T

NIP. 196906151997021002

Yanuar Nurdiansyah, ST.,M.Cs.

NIP. 198201012010121004

Mengesahkan

Ketua Program Studi Sistem Informasi

Universitas Jember,

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

NIP 196704201992011001



*Loan Decision Support System on Credit Unions Sharia BMT UGT Sidogiri Using  
Weighted Product Method*

**Anifatul Broroh**

Jurusan Sistem Informasi, Program Studi Sistem Informasi, Universitas Jember

**ABSTRACT**

*Capital for developing a business is not major problems anymore. Currently the capital could be obtained from any loan provider. Credit unions Sharia BMT UGT Sidogiri is a cooperative that provides a six-credit services i.e. gadai emas syariah (GES), modal usaha barokah (MUB), multiguna tanpa agunan (MTA), kendaraan bermotor barokah (KBB), pembelian barang elektronik (PBE), and pembiayaan kafalah haji (PKH). Credit unions Sharia BMT UGT Sidogiri found difficult is in determining prospective creditors who are eligible to receive credit due to manual process. In this research, we developed a loan decision support system on Credit Unions Sharia BMT UGT Sidogiri using weighted product method. This system is implemented in a web form so that it can be accessed easily, whenever and wherever, and thus more effective. The system is built using a programming language Page Hypertext Pre-Processor (PHP) and framework Code Igniter (CI). In general the results of measurement of the level of difficulty and complexity of the work, this system belongs to the system simple and easy to use. Thus the user quickly overwhelm the system and no difficulties in operating them.*

**Keywords:** *Loan, Decision Support System, Weighted Product.*

## RINGKASAN

**Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri Menggunakan Metode *Weighted Product***; Anifatul Baroroh; 112410101016; 2015; 140 halaman; Program Studi Sistem Informasi Universitas Jember.

Sistem pendukung keputusan adalah sistem informasi yang mampu memanipulasi data yang ada untuk membantu pengambilan keputusan pada situasi dimana keputusan tidak dapat diambil secara pasti. Pembangunan sistem pendukung keputusan dapat menghasilkan data yang lebih akurat apabila menambahkan metode dalam perhitungannya. Banyak metode yang dapat digunakan dalam sistem pendukung keputusan salah satunya adalah metode *weighted product*. Metode *weighted product* adalah metode pengambilan keputusan yang didasarkan pada beberapa atribut dimana setiap atribut tidak saling bergantung satu dengan yang lainnya. Metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atributnya.

Penelitian ini dilatarbelakangi oleh beberapa hal yang meliputi: (1) Koperasi diminati oleh masyarakat pada umumnya karena pola pembayaran yang tidak rumit, (2) mudahnya proses dan syarat melakukan pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri membuat masyarakat banyak yang mengajukan pinjaman, (3) Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri sulit dalam menentukan calon peminjam yang berhak menerima pinjaman, (4) diperlukan penyelesaian menggunakan sistem pendukung keputusan dengan metode yang mampu menyajikan data kualitatif menjadi data kuantitatif, (5) metode *weighted product* sesuai diterapkan pada sistem pendukung keputusan pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri. Oleh karena itu rumusan masalah dalam penelitian ini adalah: (1) bagaimana pengimplementasian metode *Weighted Product* untuk memutuskan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri?, (2) bagaimana merancang dan membangun sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *Weighted Product*?

Metode *weighted product* pada sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri diterapkan dalam mengambil keputusan layak atau tidaknya seorang nasabah untuk mendapat pinjaman dari koperasi. Metode *weighted product* terletak pada fitur menambah nasabah dan pada fitur edit bobot. Bobot kriteria yang telah diinputkan kedalam sistem akan disimpan pada *database* dan akan dipanggil ketika menambahkan nasabah untuk mengetahui keputusan pemberian pinjaman. Pengambilan keputusan dihitung dengan cara memangkatkan nilai kriteria nasabah. Pangkat didapat dari nilai bobot masing-masing kriteria dibagi nilai bobot keseluruhan. Pangkat bernilai positif jika kriteria merupakan kriteria keuntungan dan pangkat bernilai negatif jika kriteria merupakan kriteria kerugian.

Sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dirancang dan dibangun untuk memberikan bantuan dalam menentukan kelayakan nasabah mendapatkan pinjaman dengan memanfaatkan metode *weighted product*. Penerapan metode *weighted product* dilakukan dengan membangun fitur utama yaitu manajemen data bobot dan manajemen data nasabah. Fitur tambahan pada sistem ini ada fitur melihat pengumuman, fitur melihat info pinjaman, fitur *login*, fitur *logout*, fitur manajemen data pinjaman, fitur manajemen modal koperasi, fitur memberikan persetujuan, dan fitur manajemen data user.

## PRAKATA

Puji syukur kepada Allah SWT yang telah melimpahkan rahmat dan karuniaNya sehingga skripsi yang berjudul “Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri Menggunakan Metode *Weighted Product*” dapat terselesaikan. Shalawat dan salam senantiasa tercurah kepada Nabi Muhammad Saw. Skripsi ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan strata satu (S1) pada Program Studi Sistem Informasi Universitas Jember. Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada:

1. Prof. Drs. Slamin, M.Comp.Sc.,Ph.D. selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Drs. Antonius Cahya P, M.App.,Sc selaku Dosen Pembimbing Utama yang telah memberikan arahan, ilmu dan petunjuk, nasehat, koreksi serta saran dengan penuh kesabaran;
3. Nelly Oktavia Adiwijaya, S.Si.,MT., selaku Dosen Pembimbing Pendamping yang telah memberikan arahan, ilmu dan petunjuk, nasehat, koreksi serta saran dengan teliti dan penuh kesabaran;
4. Anang Andrianto, ST.,MT., selaku dosen penguji I dan Yanuar Nurdiansyah, ST.,M.Cs. selaku dosen penguji II yang telah memberikan masukan dalam penulisan skripsi ini.
5. semua dosen Program Studi Sistem Informasi yang telah memberikan bekal ilmu dan pengetahuan selama penulis menjadi mahasiswa.

Penulis menyadari bahwa penulisan laporan skripsi ini masih belum sempurna. Dengan demikian penulis mengharapkan kritik dan saran yang membangun. Atas perhatian dan dukungannya, penulis sampaikan terima kasih.

Jember, 16 Juni 2015

Penulis

**DAFTAR ISI**

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>PERSEMBAHAN</b> .....	ii
<b>MOTTO</b> .....	iii
<b>PERNYATAAN</b> .....	iv
<b>PENGESAHAN PEMBIMBING</b> .....	v
<b>SKRIPSI</b> .....	vi
<b>PENGESAHAN</b> .....	vii
<b>ABSTRACK</b> .....	viii
<b>RINGKASAN</b> .....	ix
<b>PRAKATA</b> .....	xi
<b>DAFTAR ISI</b> .....	xii
<b>DAFTAR TABEL</b> .....	xv
<b>DAFTAR GAMBAR</b> .....	xvi
<b>DAFTAR LAMPIRAN</b> .....	xvii
<b>BAB 1. PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan .....	3
1.4 Manfaat .....	4
1.5 Batasan Masalah.....	4
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	6
2.1 Penelitian Terdahulu .....	6
2.2 Pengertian Metode <i>Weighted Product</i> .....	7

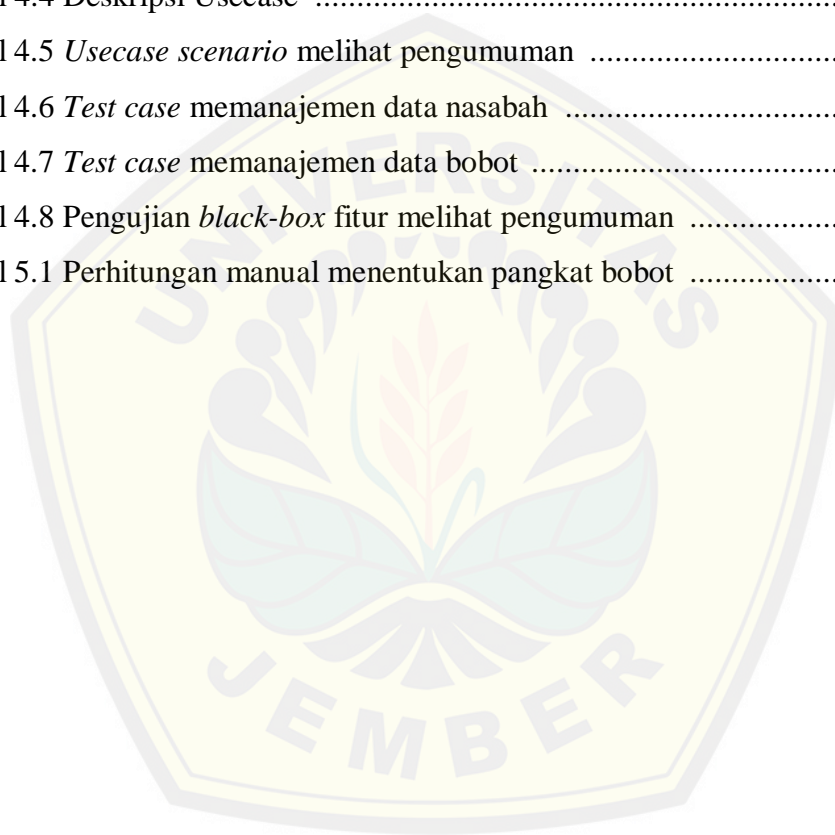
2.3 Konsep Perhitungan dengan Metode <i>Weighted Product</i> .....	8
2.4 Algoritma Wighted Product .....	10
2.5 Konsep Sistem Pendukung Keputusan .....	11
<b>BAB 3. METODE PENELITIAN</b> .....	14
3.1 Jenis Penelitian .....	14
3.2 Alur Penelitian.....	14
3.3 Teknik Pengumpulan Data.....	15
3.4 Model Pengembangan Sistem .....	16
<b>BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM</b> .....	22
4.1 Analisis Sistem.....	22
4.2 Desain Sistem.....	25
4.3 Implementasi.....	48
4.4 Pengujian Sistem .....	54
<b>BAB 5. HASIL DAN PEMBAHASAN</b> .....	61
5.1 Implementasi Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Syariah BMU UGT Sidogiri .....	61
5.2 Hasil Implementasi Metode <i>Weighted Product</i> Pada Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Simpan Pinjam Syariah Bmt Ugt Sidogiri.....	65
5.3 Pembahasan Hasil Implementasi Sistem Pendukung Keputusan Pemberian Pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dan Penerapan Metode <i>Weighted Product</i> .....	69
5.4 Pembahasan Pengujian Sistem.....	72
<b>BAB 6. PENUTUP</b> .....	74
6.1 Kesimpulan.....	74
6.2 Saran .....	75

<b>DAFTAR PUSTAKA</b> .....	76
<b>LAMPIRAN</b> .....	78



**DAFTAR TABEL**

	Halaman
Tabel 3.1 Tabel Uji <i>Black Box</i> .....	20
Tabel 4.1 Kebutuhan Fungsional Sistem .....	24
Tabel 4.1 Kebutuhan Non-fungsional Sistem .....	24
Tabel 4.3 Hak akses sistem pendukung keputusan pemberian pinjaman koperasi .....	27
Tabel 4.4 Deskripsi Usecase .....	27
Tabel 4.5 <i>Usecase scenario</i> melihat pengumuman .....	29
Tabel 4.6 <i>Test case</i> manajemen data nasabah .....	56
Tabel 4.7 <i>Test case</i> manajemen data bobot .....	58
Tabel 4.8 Pengujian <i>black-box</i> fitur melihat pengumuman .....	59
Tabel 5.1 Perhitungan manual menentukan pangkat bobot .....	67





**DAFTAR GAMBAR**

	Halaman
Gambar 2.1 Diagram Alir Algoritma <i>Weighted Product</i> .....	10
Gambar 2.2 Fase Proses Pengambilan Keputusan .....	12
Gambar 3.1 Alur penelitian .....	14
Gambar 3.2 Model Waterfall .....	16
Gambar 3.3 kriteria sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode <i>weighted product</i> .....	7
Gambar 3.4 Alur sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri .....	19
Gambar 3.4 Kriteria Keputusan kelayakan Peminjam .....	23
Gambar 4.2 Business process sistem pendukung keputusan pemberian pinjaman koperasi .....	25
Gambar 4.3 Usecase diagram sistem pendukung keputusan pemberian pinjaman koperasi .....	26
Gambar 4.4 Activity diagram melihat pengumuman .....	31
Gambar 4.5 Sequence diagram melihat pengumuman .....	32
Gambar 4.6 Class diagram .....	43
Gambar 4.7 Entity relationship diagrams (ERD) .....	45
Gambar 4.8 Model prototipe melihat pengumuman .....	46
Gambar 4.9 baris kode fitur melihat pengumuman .....	49
Gambar 4.10 Diagram alir fitur manajemen data nasabah .....	55
Gambar 4.11 Diagram alir fitur manajemen data bobot .....	57
Gambar 5.1 Fitur melihat pengumuman .....	62
Gambar 5.2 Pengumuman pinjaman nasabah .....	62
Gambar 5.3 Alur penerapan metode <i>weighted product</i> .....	68

**DAFTAR LAMPIRAN**

	Halaman
LAMPIRAN A <i>USECASE SCENARIO</i> .....	78
A.1 <i>Usecase Scenario</i> Melihat Info Pinjaman .....	78
A.2 <i>Usecase Scenario</i> Melakukan <i>Login</i> .....	78
A.3 <i>Usecase Scenario</i> Melakukan <i>Logout</i> .....	80
A.4 <i>Usecase Scenario</i> Memanajemen Data Nasabah .....	80
A.5 <i>Usecase Scenario</i> Memanajemen Data Pinjaman .....	82
A.6 <i>Usecase Scenario</i> Memanajemen Modal Koperasi .....	84
A.7 <i>Usecase Scenario</i> Memberikan Persetujuan .....	85
A.8 <i>Usecase Scenario</i> Memanajemen Data Bobot .....	86
A.9 <i>Usecase Scenario</i> Memanajemen Data User .....	87
LAMPIRAN B <i>ACTIVITY DIAGRAM</i> .....	90
B.1 <i>Activity Diagram</i> Melihat Info Pinjaman .....	90
B.2 <i>Activity Diagram</i> Melakukan <i>Login</i> .....	90
B.3 <i>Activity Diagram</i> Melakukan <i>Logout</i> .....	91
B.4 <i>Activity Diagram</i> Memanajemen Data Nasabah .....	91
B.5 <i>Activity Diagram</i> Memanajemen Data Pinjaman .....	92
B.6 <i>Activity Diagram</i> Memanajemen Modal Koperasi .....	93
B.7 <i>Activity Diagram</i> Memberikan Persetujuan .....	93
B.8 <i>Activity Diagram</i> Memanajemen Data Bobot .....	94
B.9 <i>Activity Diagram</i> Memanajemen Data User .....	94
LAMPIRAN C <i>SEQUENCE DIAGRAM</i> .....	96
C.1 <i>Sequence Diagram</i> Melihat Info Pinjaman .....	96
C.2 <i>Sequence Diagram</i> Melakukan <i>Login</i> .....	96
C.3 <i>Sequence Diagram</i> Melakukan <i>Logout</i> .....	96
C.4 <i>Sequence Diagram</i> Memanajemen Data Nasabah .....	97
C.5 <i>Sequence Diagram</i> Memanajemen Data Pinjaman .....	98
C.6 <i>Sequence Diagram</i> Memanajemen Modal Koperasi .....	99
C.7 <i>Sequence Diagram</i> Memberikan Persetujuan .....	99
C.8 <i>Sequence Diagram</i> Memanajemen Data Bobot .....	100

C.9 <i>Sequence Diagram</i> Memanajemen Data User .....	101
LAMPIRAN D PROTOTIPE .....	102
F.1 Prototipe Halaman Melihat Info Pinjaman .....	102
F.2 Prototipe Halaman Melihat Info Pinjaman Selengkapnya .....	102
F.3 Prototipe Halaman <i>Login</i> .....	102
F.4 Prototipe Halaman <i>Login</i> Sukses Admin .....	103
F.5 Prototipe Halaman <i>Login</i> Sukses Teller .....	103
F.6 Prototipe Halaman <i>Login</i> Salah .....	103
F.7 Prototipe Halaman <i>Logout</i> .....	104
F.8 Prototipe Halaman Data Nasabah .....	104
F.9 Prototipe Halaman Tambah Nasabah .....	104
F.10 Prototipe Halaman Edit Nasabah .....	105
F.11 Prototipe Halaman Data Pinjaman .....	105
F.12 Prototipe Halaman Tambah Pinjaman .....	105
F.13 Prototipe Halaman Edit Pinjaman .....	106
F.14 Prototipe Halaman Modal Koperasi .....	106
F.15 Prototipe Halaman Edit Modal Koperasi .....	106
F.16 Prototipe Halaman Memberikan Persetujuan .....	107
F.17 Prototipe Halaman Data Bobot .....	107
F.18 Prototipe Halaman Edit Bobot .....	107
F.19 Prototipe Halaman Data User .....	108
F.20 Prototipe Halaman Tambah User .....	108
F.21 Prototipe Halaman Edit User .....	108
LAMPIRAN E BARIS PROGRAM .....	109
E.1 Baris Program Fitur Melihat Info Pinjaman .....	109
E.2 Baris Program Fitur Melakukan <i>Login</i> .....	109
E.3 Baris Program Fitur Melakukan <i>Logout</i> .....	110
E.4 Baris Program Fitur Memanajemen Data Nasabah .....	110
E.5 Baris Program Fitur Memanajemen Data Pinjaman .....	112
E.6 Baris Program Fitur Memanajemen Modal Koperasi .....	113
E.7 Baris Program Fitur Memberikan Persetujuan .....	113

E.8 Baris Program Fitur Memanajemen Data Bobot .....	114
E.9 Baris Program Fitur Memanajemen Data User .....	115
LAMPIRAN F TAMPILAN FITUR .....	117
F.1 Tampilan Fitur Melihat Info Pinjaman .....	117
F.2 Tampilan Fitur Melihat Info Pinjaman Selengkapnya .....	117
F.3 Tampilan Fitur <i>Login</i> .....	117
F.4 Tampilan Fitur <i>Login</i> Sukses Admin .....	118
F.5 Tampilan Fitur <i>Login</i> Sukses Teller .....	118
F.6 Tampilan Fitur <i>Login</i> Salah .....	118
F.7 Tampilan Fitur <i>Logout</i> .....	119
F.8 Tampilan Fitur Data Nasabah .....	119
F.9 Tampilan Fitur Tambah Nasabah .....	119
F.10 Tampilan Fitur Edit Nasabah .....	120
F.11 Tampilan Fitur Data Pinjaman .....	120
F.12 Tampilan Fitur Tambah Pinjaman .....	120
F.13 Tampilan Fitur Edit Pinjaman .....	121
F.14 Tampilan Fitur Modal Koperasi .....	121
F.15 Tampilan Fitur Edit Modal Koperasi .....	121
F.16 Tampilan Fitur Memberikan Persetujuan .....	122
F.17 Tampilan Fitur Data Bobot .....	122
F.18 Tampilan Fitur Edit Bobot .....	122
F.19 Tampilan Fitur Data User .....	123
F.20 Tampilan Fitur Tambah User .....	123
F.21 Tampilan Fitur Edit User .....	123
LAMPIRAN G <i>LISTING</i> PROGRAM <i>WHITE_BOX</i> .....	124
G.1 <i>Listing</i> Program Memanajemen Data Nasabah .....	124
G.2 <i>Listing</i> Program Memanajemen Data Bobot .....	125
LAMPIRAN H <i>BLACK-BOX TESTING</i> .....	126
H.1 <i>Testing Balck Box</i> Melihat Info Pinjaman .....	126
H.2 <i>Testing Balck Box</i> Melakukan <i>Login</i> .....	126
H.3 <i>Testing Balck Box</i> Melakukan <i>Logout</i> .....	128

H.4 <i>Testing Balck Box</i> Memanajemen Data Nasabah .....	128
H.5 <i>Testing Balck Box</i> Memanajemen Data Pinjaman .....	131
H.6 <i>Testing Balck Box</i> Memanajemen Modal Koperasi .....	134
H.7 <i>Testing Balck Box</i> Memberikan Persetujuan .....	135
H.8 <i>Testing Balck Box</i> Memanajemen Data Bobot .....	136
H.9 <i>Testing Balck Box</i> Memanajemen Data User .....	138



## BAB 1. PENDAHULUAN

Bab ini merupakan pemaparan hal-hal yang terkait dengan pendahuluan yang meliputi latar belakang, rumusan masalah, tujuan, manfaat, dan batasan masalah.

### 1.1 Latar Belakang

Koperasi adalah sebuah wadah kerja sama yang dilakukan oleh orang-orang yang mempunyai kepentingan dan tujuan yang sama. Koperasi merupakan salah satu instansi yang menyediakan layanan simpan pinjam. Selain sebagai instansi yang menyediakan layanan simpan pinjam, koperasi juga memiliki layanan lain sesuai dengan jenis koperasi yang didirikan seperti koperasi produksi, koperasi konsumsi, dan koperasi serba usaha.

Pola pembayaran yang tidak rumit membuat koperasi diminati oleh masyarakat pada umumnya. Koperasi sering menggunakan istilah “kekeluargaan” dan pada sistem syariah menggunakan bagi hasil dalam model transaksinya. Pembayaran dengan cara mengangsur sesuai ketentuan koperasi yang mudah membuat masyarakat merasa ringan melakukan pinjaman pada koperasi tersebut. Pada umumnya di koperasi tidak ada patokan minimum dalam peminjaman. Kalaupun ada beberapa koperasi yang menerapkan batas minimum peminjaman, hal tersebut masih sangat terjangkau oleh masyarakat terutama masyarakat desa yang ingin mengembangkan UKM (Usaha Kecil Menengah). Berdasarkan hal tersebut permasalahan modal dalam membangun sebuah usaha bukanlah sebagai masalah utama lagi.

Koperasi Simpan Pinjam Syariah Baitul Maal wa Tamwil (BMT) UGT Sidogiri merupakan koperasi simpan pinjam yang mengutamakan karakter syariah dalam menjalankannya. Koperasi ini melayani simpan pinjam untuk anggota baik sebagai modal usaha maupun sebagai pemenuh kebutuhan hidup anggota. Koperasi ini memberikan layanan pinjaman dengan syarat yang telah ditentukan oleh koperasi pada umumnya namun berlandaskan syariah.

Mudahnya proses dan syarat melakukan pinjaman serta karakter terpercaya yang diutamakan pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri membuat masyarakat banyak yang mengajukan pinjaman. Namun Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri sulit dalam menentukan calon peminjam yang berhak menerima kredit. Proses lain yang mempersulit penentuan penerima pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri adalah proses manual yang kurang efisien. Selain itu kegiatan administrasi kredit berupa data kualitatif membutuhkan waktu yang lama, juga menghambat proses penentuan keputusan.

Berdasarkan pemaparan yang telah dijelaskan, diperlukan penyelesaian menggunakan sistem pendukung keputusan (*DSS*) menggunakan metode yang mampu menyajikan data kualitatif menjadi data angka atau kuantitatif. Alter (2002:38) mendefinisikan sistem pendukung keputusan atau *Decision Support Systems (DSS)* adalah sistem informasi interkatif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan pada situasi yang semiterstruktur dan situasi yang tidak terstruktur di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Sistem pendukung keputusan pemberian pinjaman ini dibangun untuk memperoleh data penerima pinjaman sesuai dengan kriteria yang telah ditentukan.

Dalam penelitian ini akan membangun sebuah sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *weighted product*. Menurut Ahmadi, dkk (2014) Metode *weighted product* merupakan salah satu dari beberapa metode *MADM (Multi Kriteriae Decision Making)*. Metode *MADM* merupakan metode pengambilan keputusan yang didasarkan pada beberapa kriteria. Metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap kriteria. Metode *weighted product* menggunakan proses normalisasi, dimana rating setiap kriteria harus dipangkatkan dahulu dengan bobot kriteria yang bersangkutan. Pangkat bernilai positif untuk kriteria keuntungan dan pangkat bernilai negatif untuk kriteria biaya. Dengan metode ini data yang disajikan

oleh sistem akan lebih tepat sehingga pengambilan keputusan pemberian pinjaman akan sesuai dengan kriteria yang telah ditentukan.

Berdasarkan pemaparan yang telah dijelaskan maka penelitian dengan judul “Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri Menggunakan Metode *Weighted Product*” sangat diperlukan guna memudahkan dan mengefisienkan kerja staf koperasi dalam menentukan pemberian pinjaman secara otomatis di koperasi pada umumnya.

### **1.2 Perumusan Masalah**

Berdasarkan permasalahan yang telah diuraikan diatas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana pengimplementasian metode *Weighted Product* untuk memutuskan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri?
2. Bagaimana merancang dan membangun sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri?

### **1.3 Tujuan**

Tujuan dari penelitian ini adalah:

1. Mengimplementasikan metode *Weighted Product* yang dapat mendukung pengambilan keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri.
2. Merancang dan membangun sistem pendukung keputusan yang dapat menyimpan data-data calon peminjam dan mengolahnya untuk diurutkan berdasarkan nilai tertinggi hingga terendah.



#### 1.4 Manfaat

Manfaat penelitian ini adalah :

##### 1. Bagi akademis

Penelitian yang dilakukan diharapkan memberikan hasil yang mampu memberikan masukan informasi yang terkait dengan judul penelitian kepada pembaca pada umumnya dan pada Program Studi Sistem Informasi Universitas Jember pada khususnya.

##### 2. Bagi peneliti

- a. Mengetahui data yang digunakan dalam menentukan kelayakan pemberian pinjaman pada koperasi.
- b. Mengetahui proses penerapan sistem pendukung keputusan dalam menentukan pemberian pinjaman pada koperasi.
- c. Mengetahui cara kerja metode *weighted product* dalam menentukan sebuah sistem pendukung sebuah keputusan.

##### 3. Bagi objek penelitian

Mempermudah proses penentuan pemberian pinjaman yang sebelumnya dilakukan secara manual dengan memberikan rangking kelayakan sebagai alat bantu pihak koperasi.

#### 1.5 Batasan Masalah

Beberapa hal yang membatasi penelitian ini adalah:

1. Penelitian dilakukan di Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri cabang Semboro.
2. Kriteria yang digunakan sebagai dasar penilaian diperoleh dari Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri cabang Semboro.
3. Sistem pendukung keputusan ini hanya sebagai alat bantu bagi pihak koperasi dalam menentukan siapa yang layak menerima pinjaman atau tidak berdasarkan kriteria yang ditentukan oleh pihak koperasi dan keputusan akhir tetap berada di pihak koperasi.
4. Metode yang digunakan dalam perancangan sistem ini adalah *Weighted Product*.

5. Hasil dari sistem pendukung keputusan ini adalah urutan prioritas calon peminjam yang layak dan tidak layak menerima pinjaman mulai dari yang tertinggi sampai terendah.
6. Sistem berbasis *website*.



## BAB 2. TINJAUAN PUSTAKA

Dalam bab ini dipaparkan teori-teori dan pustaka yang digunakan dalam penelitian. Teori-teori dan pustaka yang digunakan dan dibahas dalam penelitian ini yaitu penelitian terdahulu, pengertian metode *weighted product*, konsep perhitungan dengan metode *weighted product*, algoritam metode *weighted product*, dan konsep sistem pendukung keputusan.

### 2.1 Penelitian Terdahulu

Bagian ini menguraikan penelitian-penelitian sebelumnya yang mempunyai relevansi dengan tinjauan sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri.

Pada sebuah penelitian sistem pendukung keputusan pencairan kredit pernah dibangun menggunakan metode *scoring system* (Sari, 2012). Metode *scoring system* dapat menyajikan data kualitatif seorang calon kreditur menjadi informasi berbentuk kuantitatif. Nilai kuantitatif didapatkan dengan menentukan prioritas dari setiap kriteria yang mendukung kelayakan calon kreditur dalam pencairan kredit kemudian mengalikannya dengan nilai dari setiap kriteria. Batas maksimal dan minimal nilai ditentukan dengan melihat nilai maksimum dan minimum dari setiap kriteria. Calon kreditur yang memiliki nilai tinggi dinyatakan layak mencairkan kredit.

Metode ini banyak digunakan karena waktu yang dibutuhkan dalam perhitungan lebih singkat. Selain itu metode *scoring system* dengan skala kategorisasi jenjang (ordinal) mampu menghasilkan 5 kategorisasi yaitu tidak layak, kurang layak, cukup layak, layak, dan sangat layak yang dapat digunakan sebagai standar kategorisasi. Tetapi metode ini mengabaikan sifat dari setiap kriteria sehingga semua kriteria dianggap sebagai kriteria keuntungan tanpa memperhatikan kriteria yang memiliki sifat kriteria biaya. Penerapan *scoring system* dalam sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri membuat koperasi tersebut mendapatkan hasil yang kurang akurat karena pengabaian sifat kriteria tersebut.

Berdasarkan pemaparan kekurangan dari metode *scoring system* dibutuhkan sistem pendukung keputusan menggunakan metode yang juga memperhatikan sifat dari setiap kriteria. Metode *scoring system* belum dapat membedakan apakah kriteria masuk ke dalam keuntungan atau biaya. Padahal sifat kriteria ini sangat dibutuhkan oleh sistem karena dapat menampilkan keputusan yang menguntungkan atau merugikan bagi Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri. Oleh karena itu pada penelitian ini sistem pendukung keputusan pinjaman dibangun dengan menggunakan metode *weighted product*.

Metode *weighted product* pernah digunakan pada sistem pendukung keputusan pemberian bonus pegawai (Permatasari, 2013). Metode ini menggunakan proses normalisasi, dimana rating setiap atribut harus dipangkatkan dahulu dengan bobot atribut yang bersangkutan. Pangkat bernilai positif apabila atribut bersifat keuntungan dan pangkat bernilai negatif apabila atribut bersifat biaya. Penerapan sistem pendukung keputusan pemberian bonus menggunakan metode *weighted product* dapat mempercepat kinerja manajer dalam menentukan siapa saja pegawai yang layak menerima bonus pada setiap periodenya dengan akurat.

Metode *weighted product* dapat diterapkan dalam pembangunan sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri karena proses pengambilan keputusan pada koperasi ini hampir sama dengan proses pengambilan keputusan pada penelitian sebelumnya. Metode ini juga dapat melengkapi kekurangan pada metode *scoring system* dengan memperhatikan sifat masing-masing kriteria. Dengan menggunakan metode *weighted product* sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri akan menampilkan hasil yang lebih akurat.

## 2.2 Pengertian Metode *Weighted Product*

Metode *weighted product* merupakan salah satu dari beberapa metode *MADM* (*Multi Attribute Decision Making*). Metode *MADM* merupakan

metode pengambilan keputusan yang didasarkan pada beberapa atribut. Konsep permasalahannya adalah mengevaluasi  $m$  alternatif  $A_i$  ( $i=1,2,\dots,m$ ) terhadap sekumpulan atribut atau kriteria  $C_j$  ( $j=1,2,\dots,n$ ), dimana setiap atribut tidak saling bergantung satu dengan yang lainnya. Metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut (Ahmadi, dkk, 2014).

Metode ini diimplementasikan dalam Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri guna menentukan calon peminjam yang layak mendapatkan pinjaman. Alternatif dalam penelitian ini adalah anggota koperasi yang mengajukan pinjaman sebagai objek yang akan dikualifikasi kelayakannya. Atribut atau kriteria yang dimaksud adalah syarat peminjaman yang ditentukan oleh Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri sendiri. Kualifikasi yang dilakukan akan menghasilkan keputusan yang berbentuk data kuantitatif dengan menentukan nilai atau bobot dari masing-masing kriteria.

### 2.3 Konsep Perhitungan dengan Metode *Weighted Product*

Menurut Ahmadi, dkk (2014) metode *weighted product* menggunakan proses normalisasi, dimana rating setiap atribut harus dipangkatkan dahulu dengan bobot atribut yang bersangkutan. Proses ini diberikan dengan rumus sebagai berikut :

$$S_i = \prod_{j=1}^n X_{ij}^{W_j}$$

dengan  $i = 1, 2, \dots, m$ , dimana:

$S$  menyatakan preferensi alternatif

$x$  menyatakan nilai kriteria,

$w$  menyatakan bobot kriteria,

$n$  menyatakan banyaknya kriteria.

$W_j$  adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya.

### 2.1.1 Alternatif (A<sub>i</sub>)

Alternatif A<sub>i</sub> dengan  $i=1,2,\dots,m$  adalah obyek-obyek yang berbeda dan memiliki kesempatan yang sama untuk dipilih oleh pengambil keputusan.

Preferensi relatif dari setiap alternatif diberikan sebagai :

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{W_j}}{\prod_{j=1}^n X_{ij}^{W_j}}$$

dimana :

V : Preferensi alternatif,

X : Nilai kriteria,

w : Bobot kriteria.

Alternatif yang akan dipilih adalah yang memiliki nilai preferensi tertinggi.

### 2.1.2 Kriteria (C<sub>j</sub>)

Untuk memilih calon peminjam yang layak mendapatkan pinjaman, maka dibutuhkan beberapa kriteria pengambilan keputusan. Kriteria (C<sub>j</sub>) yang ditetapkan sesuai dengan jenis pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri. Jenis pinjaman yang ada pada koperasi tersebut adalah UGT GES (Gadai Emas Syariah), UGT MUB (Modal Usaha Barokah), UGT MTA (Multiguna Tanpa agunan), UGT KBB (Kendaraan Bermotor Barokah), UGT PBE (Pembelian Barang Elektronik), dan UGT PKH (Pembiayaan Kafalah Haji).

### 2.1.3 Bobot (w)

Adapun bobot adalah nilai atau tingkat kepentingan relatif dari setiap kriteria (C<sub>j</sub>) yang diberikan oleh *decision maker*. Nilai bobot diberikan sebagai :

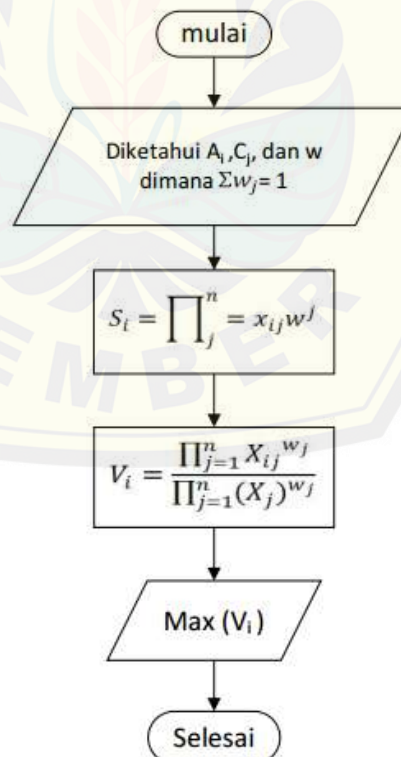
$$W_j = \frac{w}{\sum w}$$

dimana nilai  $\sum W_j = 1$

## 2.4 Algoritma Wighted Product

Algoritma metode *weighted product* dimulai dengan mengitiputkan data yang akan diproses. Dalam sistem penunjang keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri data yang diinputkan adalah data masing-masing kriteria dari calon peminjam yang mengajukan pinjaman. Pada tahap kedua data yang telah diinputkan akan diproses berdasarkan rumus dari metode *weighted product*. Setelah nilai dari setiap calon peminjam telah didapatkan maka akan dilakukan perengkingan dari nilai tertinggi hingga terendah sehingga dapat diketahui calon peminjam mana yang layak mendapatkan pinjaman dari Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri

Langkah-langkah analisis permasalahan dengan metode *weighted product* menurut Ahmadi, dkk (2014) digambarkan dalam diagram alir seperti berikut :



Gambar 2.1 Diagram Alir Algoritma *Weighted Product* (Ahmadi, dkk, 2014)

Tahapan-tahapan yang digambarkan oleh diagram alir pada Gambar 2 dijelaskan sebagai berikut :

1. Input data

Data yang diinputkan adalah nilai alternatif ( $A_i$ ), nilai kriteria ( $C_j$ ) dan bobot ( $w$ ) dimana  $\sum W_j = 1$ .

2. Proses  $S_i$

Ini adalah proses normalisasi, dimana nilai vektor  $S$  yang dicari merupakan nilai preferensi untuk setiap alternatif.

3. Proses  $V_i$

Proses mencari vektor  $V$  sebagai perankingan untuk setiap alternatif.

4.  $V_i$  maksimum

Mencari nilai terbesar dari beberapa alternatif yang ada. Dengan kata lain, alternatif terbesar yang terpilih merupakan alternatif terbaik.

## 2.5 Konsep Sistem Pendukung Keputusan

Konsep sistem pendukung keputusan (*DSS*) dikemukakan pertama kali oleh ScootMorton pada tahun 1971 (Turban, dkk, 2005). Definisi awal *DSS* tersebut yaitu sistem berbasis komputer yang interaktif, yang membantu pengambil keputusan dengan menggunakan data dan model untuk memecahkan persoalan-persoalan tak terstruktur.

Sistem pendukung keputusan merupakan sistem yang mampu mengolah data kualitatif dan menyajikannya dalam bentuk kuantitatif dengan tujuan untuk membantu proses pengambilan keputusan yang sulit untuk dihitung secara langsung. Sistem pendukung keputusan hampir sama dengan sistem informasi manajemen tradisional karena keduanya sama-sama tergantung pada basisdata sebagai sumber data dimana *DSS* menekankan pada fungsi pendukung pembuatan keputusan diseluruh tahap-tahapnya, walaupun keputusan aktual masih tetap wewenang eksekutif sebagai pembuat keputusan (Kendall, dkk, 2006).



Sebuah sistem pendukung keputusan harus melalui empat tahap yang meliputi pemahaman, perancangan, pemilihan, dan implementasi (Simon dalam Kadrasah dan Ramdhani 2012) :

1. Tahap Pemahaman (*Intelligence Phase*)

Tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.

2. Tahap Perancangan (*Design Phase*)

Tahap ini merupakan proses menemukan, mengembangkan, dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi proses untuk mengerti masalah, menurunkan solusi, dan menguji kelayakan solusi.

3. Tahap Pemilihan (*Choice Phase*)

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Tahap ini dimulai dengan mencari solusi dengan menggunakan model, melakukan analisis, menyeleksi alternatif terbaik, melaksanakan rencana untuk mengimplementasikan, dan merancang sistem pengendalian.

4. Tahap Implementasi (*Implementation Phase*)

Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan.



Gambar 2.2 Fase Proses Pengambilan Keputusan (Suryadi, 2002)

Fase ini juga dapat digunakan dalam proses pengambilan keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri. Tahap awal menentukan masalah dalam koperasi tersebut dan mengetahui masukan yang akan diproses. Kemudian merancang tindakan mengenai masalah yang ditemukan dalam koperasi tersebut sehingga dapat dilakukan penyeleksian alternatif terbaik dan selanjutnya mengimplementasikan alternatif yang telah dipilih.



### BAB 3. METODE PENELITIAN

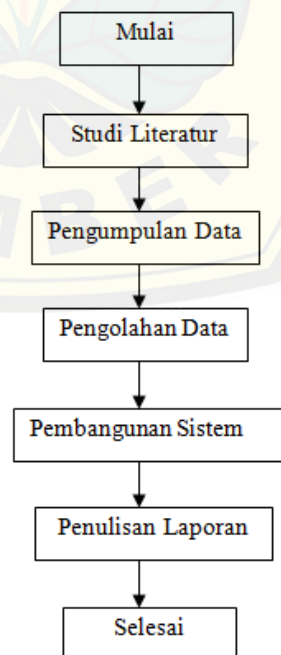
Pada bab metode penelitian dipaparkan beberapa hal yang meliputi jenis penelitian, alur penelitian, teknik pengumpulan data, dan model pengembangan sistem.

#### 3.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian kualitatif dan jenis penelitian kuantitatif. Menggunakan penelitian jenis kualitatif karena pada penelitian ini menganalisa studi literatur yang berhubungan dengan indikator untuk menentukan penerima pinjaman. Menggunakan penelitian jenis kuantitatif karena data yang diolah dalam bentuk angka.

#### 3.2 Alur Penelitian

Alur penelitian merupakan urutan langkah penelitian yang dilakukan mulai dari studi literatur, pengumpulan data, pengolahan data, perancangan sistem, dan pengimplementasian rancangan sistem. diagram alur pada penelitian ini dapat dilihat pada gambar 3.1 berikut.



Gambar 3.1 Alur penelitian

### 3.3 Teknik Pengumpulan Data

Bagian ini menguraikan langkah-langkah yang ditempuh dan teknik yang digunakan untuk mengumpulkan data.

#### 3.1.1 Observasi Partisipatif

Observasi partisipatif adalah peneliti terlibat langsung dalam kegiatan sehari-hari orang yang sedang diamati atau digunakan sebagai sumber data (Sugiyono, 2011). Adapun data yang diambil melalui observasi ini adalah data kualitatif. Data kualitatif adalah data yang diambil dari proses tindakan yang berlangsung berbentuk informasi-informasi deskriptif mengenai Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri.

#### 3.1.2 Wawancara

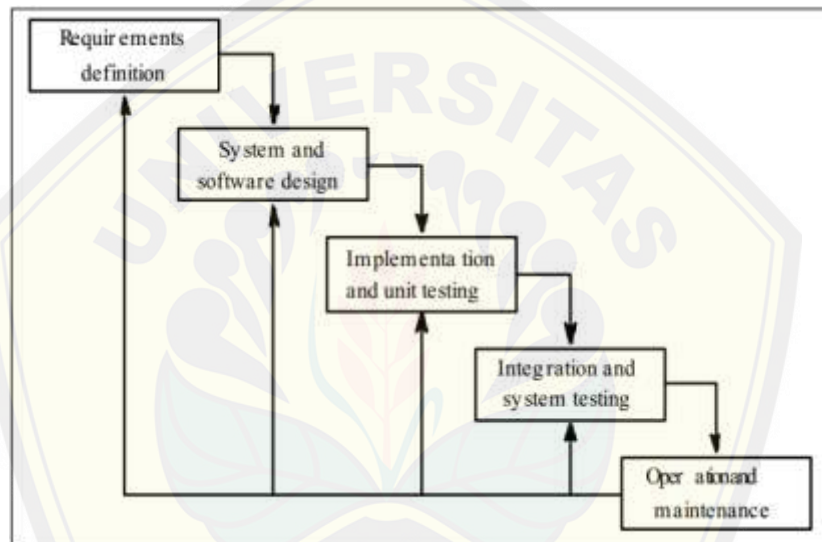
Wawancara adalah percakapan dengan maksud tertentu. Percakapan itu dilakukan oleh dua pihak, yaitu pewawancara (*interviewer*) yang mengajukan pertanyaan dan terwawancara (*interviewee*) yang memberikan jawaban atas pertanyaan itu (Moleong, 2011). Penelitian ini menggunakan teknik wawancara tidak terstruktur. Wawancara tidak berstruktur adalah wawancara yang hanya memuat garis besar pertanyaan dan narasumbernya atau terwawancara merupakan orang-orang tertentu yang memiliki sifat khas. Tujuan dari teknik wawancara dalam penelitian ini yaitu untuk memperoleh informasi yang ada relevansinya dengan pokok persoalan penelitian mengenai latar belakang objek penelitian dan data berupa informasi yang diperlukan dalam membangun sistem pendukung keputusan pemberian pinjaman.

#### 3.1.3 Studi Pustaka

Studi pustaka adalah teknik pencarian data dari arsip-arsip dari perusahaan dan teori yang berhubungan dengan permasalahan yang diangkat dalam penelitian ini. Data hasil studi pustaka dapat menunjang dalam menguraikan data kualitatif berupa informasi mengenai Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dan teori-teori yang digunakan dalam membangun sistem penunjang keputusan pemberian pinjaman.

### 3.4 Model Pengembangan Sistem

Untuk membangun sebuah sistem dibutuhkan sebuah model dalam pengembangannya. Dalam penelitian ini model yang digunakan adalah model waterfall. Model ini dipakai karena umum digunakan dalam pengembangan *software* berskala kecil. Lima tahap dalam model waterfall adalah sebagai berikut, yaitu *Requirement Analysis, System Design, Implementation & Unit Testing, Integration & Testing, Operations & Maintenance*.

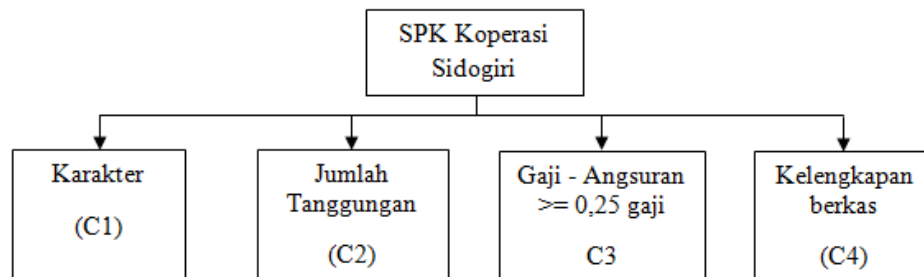


Gambar 3.2 Model *Waterfall* (Sommerville, 2011)

Keterangan model *waterfall* :

#### 1. *Requirement Analysis*

Tahap *requirement analysis* adalah tahap menganalisa apa yang dibutuhkan oleh sistem. Data kebutuhan dibagi menjadi kebutuhan fungsional dan non-fungsional. Pada sistem ini dibutuhkan juga data-data kriteria yang akan digunakan sebagai pendukung keputusan. Data yang sudah terkumpul akan menentukan bagaimana fitur yang akan dibangun pada sistem. Gambar 3.3 berikut merupakan kriteria yang digunakan dalam sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *weighted product*.



Gambar 3. 3 kriteria sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *weighted product*

## 2. System Design

Tahap *system design* adalah tahap yang dilakukan setelah data analisis telah terkumpul. Pada tahap desain ini menggunakan bahasa *Unified Modeling Language (UML)*. Bahasa ini digunakan karena mendukung konsep *Object Oriented Design (OOP)* sesuai dengan kode program yang akan digunakan programmer. Beberapa diagram pada *UML* adalah sebagai berikut :

### a. Business Process

*Business process* adalah diagram yang memperlihatkan inputan yang digunakan sistem, output yang dihasilkan sistem, dan tujuan pembuatan sistem.

### b. Use Case Diagram

*Use case diagram* adalah gambaran fitur dari sistem yang dijalankan oleh aktor. Pada diagram ini dapat dilihat juga hak akses dari aktor.

### c. Sequence Diagram

*Sequence diagram* adalah diagram yang menggambarkan interaksi antara objek satu dengan yang lain di dalam sistem yang dibangun pada urutan waktu. Diagram juga menggambarkan interaksi antara aktor, fitur, serta data yang berjalan.

d. *Activity Diagram*

*Activity diagram* adalah penggambaran alir sistem yang akan dibangun, bagaimana sistem dari awal hingga sistem ditutup, serta bagaimana alir sistem ketika diimplementasikan dengan metode yang digunakan.

e. *Class Diagram*

*Class diagram* adalah diagram yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu kelas dengan yang lain. Dalam kelas ini juga ditampilkan atribut dan operasi yang ada pada sistem.

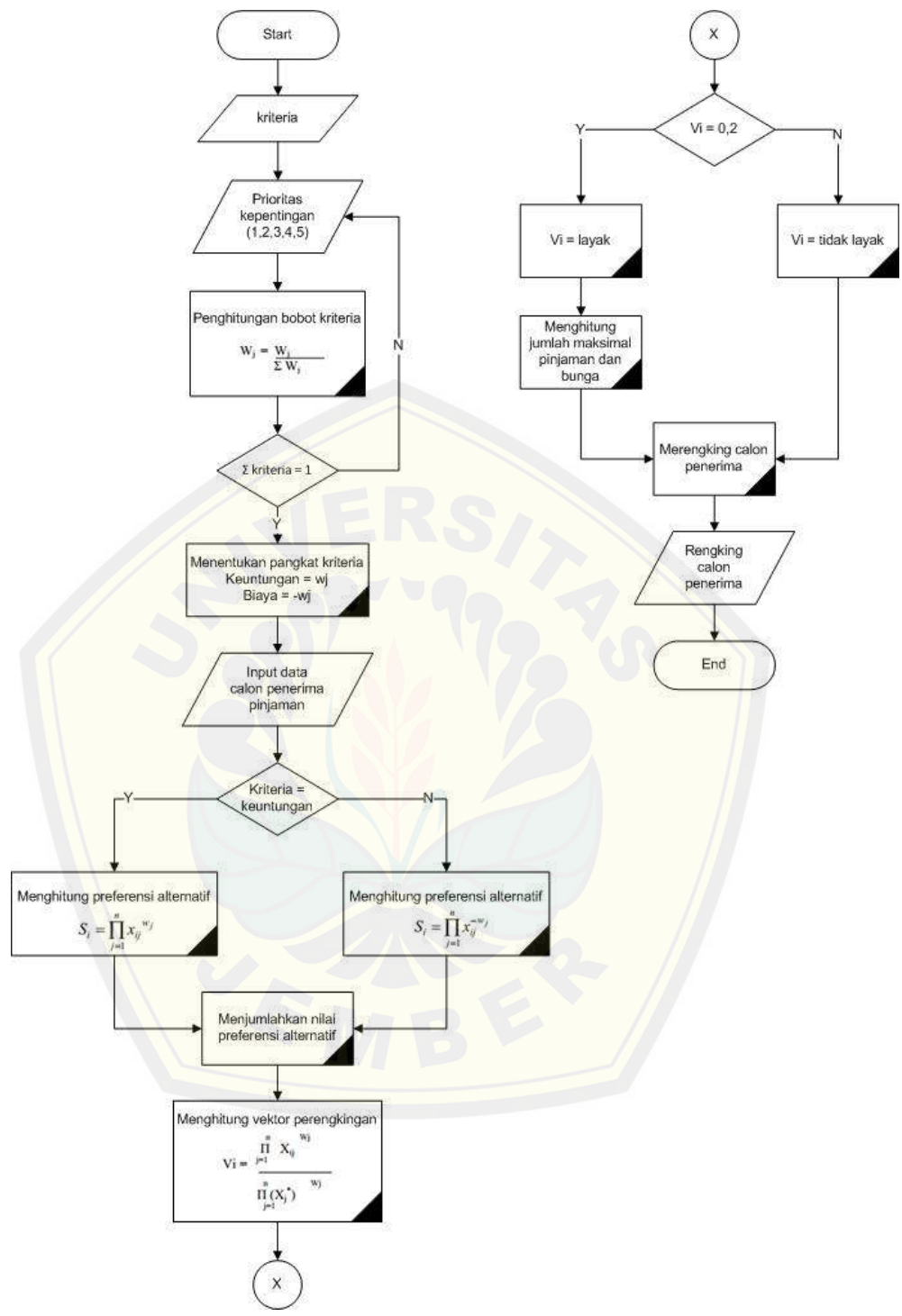
f. *Entity Relation Diagram (ERD)*

*Entity relation diagram (ERD)* adalah diagram yang menggambarkan relasi objek-objek dasar data dalam sebuah basis data.

3. *Implementation & Unit Testing*

Desain yang sudah dibuat pada tahap sebelumnya digunakan sebagai acuan dalam pembangunan sistem. Sistem dibangun menggunakan bahasa pemrograman *Page Hypertext Pre-Processor (PHP)* dengan *tool* yang digunakan Notepad++, dan database yang digunakan adalah *MySQL*. Proses pengkodean menggunakan *framework Code Igniter (CI)* karena mendukung penggunaan konsep *Object Oriented Programming (OOP)*.

Alur sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri setelah diimplementasikan dengan metode *weighted product* dapat dilihat pada gambar 3.4. Proses penghitungan preferensi alternatif melihat dari kriteria yang ada, pangkat bernilai positif untuk kriteria keuntungan dan bernilai negatif untuk kriteria biaya.



Gambar 3.4 Alur sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri



#### 4. *Integration & Testing*

Tahap testing harus dilakukan sebelum sistem diserahkan kepada *user*. Tahap dilakukan agar programmer dapat mengetahui apakah sistem yang dibangun sesuai dengan kebutuhan yang telah dianalisis diawal. Serta agar mengetahui apakah terdapat kesalahan pada sistem yang dibangun. Tahap testing dilakukan guna menyempurnakan sistem sebelum diserahkan kepada *user*. Pada tahap testing ini dilakukan pengujian dengan metode *white-box* dan metode *black-box*.

*White-box testing* adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case* (Pressman 2005). Pada penelitian ini metode yang digunakan dalam pengujian *white box* adalah *independent path* dengan menggunakan *cyclomatic complexity*.

*Cyclomatic complexity* yaitu metrik perangkat lunak yang menyajikan ukuran kuantitatif dari kekompleksan logikal suatu program. *Cyclomatic complexity* digunakan untuk mencari jumlah *path* dalam suatu *flowgraph*. Rumus yang digunakan dalam menghitung *cyclomatic complexity* adalah sebagai berikut :

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

Keterangan :

E : Jumlah *edge* grafik alir

N : Jumlah *node* grafik alir

*Black box testing* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi apakah sesuai dengan spesifikasi yang dibutuhkan (Shalahuddin dan Rosa 2011). Pengujian *black box* seperti pada tabel berikut:

Tabel 3. 1 Tabel Uji *Black Box*

Kelas Uji	Skenario Uji	Hal yang Diharapkan	Kesimpulan

### 5. *Operations & Maintenance*

Tahap *operations & maintenance* adalah tahap akhir dari model waterfall. Tahap ini dilakukan setelah sistem melalui tahap pengujian baik melalui pengujian *white-box* maupun pengujian *black-box*. Tahap *operations & maintenance* dilakukan guna pengembangan sistem atau mengatasi masalah (*bugs*) yang muncul setelah sistem diserahkan kepada *user*.



## BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM

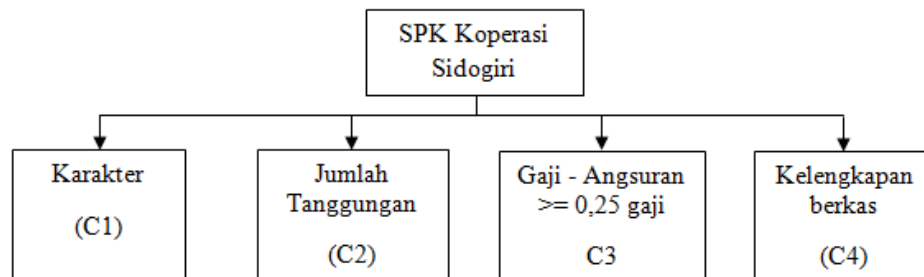
Bab analisis dan pengembangan sistem menjabarkan proses analisis, perancangan, dan pengembangan sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri. Perancangan yang dilakukan meliputi pembuatan *business process*, *usecase diagram*, *usecase scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram (ERD)*.

### 4.1 Analisis Sistem

Sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri adalah sebuah sistem yang digunakan untuk membantu koperasi dalam menentukan nasabah yang layak mendapatkan pinjaman. Sistem menggunakan metode *weighted product* dalam proses perhitungannya sehingga didapatkan nilai kelayakan dari setiap nasabah. Data nasabah yang diinputkan oleh teller akan diolah dengan bobot yang sebelumnya telah ditentukan oleh admin dan disajikan dalam bentuk info nasabah yang dapat dilihat oleh nasabah.

Untuk menentukan kelayakan peminjaman yang dilakukan oleh nasabah Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri memberikan kriteria berdasarkan 6 jenis peminjaman pada koperasi tersebut yaitu UGT GES (Gadai Emas Syariah), UGT MUB (Modal Usaha Barokah), UGT MTA (Multiguna Tanpa agunan), UGT KBB (Kendaraan Bermotor Barokah), UGT PBE (Pembelian Barang Elektronik), dan UGT PKH (Pembiayaan Kafalah Haji).

Keenam jenis pinjaman tersebut memiliki kriteria yang sama dalam menentukan kelayakan peminjam, tetapi berbeda dalam penentuan jumlah pinjaman. Kriteria yang digunakan untuk memutuskan peminjam yang layak dapat dilihat dalam gambar berikut :



Gambar 4.1 Kriteria Keputusan kelayakan Peminjam

Kriteria yang ditentukan oleh Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri adalah karakter dari calon peminjam, jumlah orang yang ditanggung peminjam, rasio pengasilan perbulan dengan angsuran yang tidak boleh lebih dari 0,75 dari pengasilan perbulan tersebut, dan kelengkapan berkas sesuai jenis pinjaman yang diajukan. Nilai pinjaman yang diberikan berbeda pada setiap jenis pinjaman. Pada UGT GES jumlah pinjaman ditentukan dari jumlah emas apabila dirupiahkan. Pada UGT MUB jumlah pinjaman ditentukan dari nilai agunan apabila dirupiahkan. Pada UGT GTA penentuan jumlah pinjaman tergantung pada jumlah pinjaman yang dibutuhkan oleh peminjam. Pada UGT KBB jumlah pinjaman ditentukan dari harga kendaraan yang akan dibeli. Sama dengan UGT KBB pada UGT KBE jumlah pinjaman tergantung pada harga barang yang akan dibeli. Dan pada UGT PKH jumlah pinjaman ditentukan dari jumlah kekurangan pembayaran haji untuk mendapatkan kursi haji.

Dengan menggunakan kriteria tersebut pihak Koperasi Simpan Pinjam Syaria BMT UGT Sidogiri dapat melakukan prosentase penilaian dari setiap kriteria yang digunakan. Proses perhitungan dilanjutkan dengan menggunakan metode *weighted product*.

Agar sistem dapat dijalankan dengan optimal maka diperlukan karakteristik yang dimiliki oleh sistem yang dapat dijelaskan pada kebutuhan fungsional dan kebutuhan non fungsional. Tabel kebutuhan fungsional menjelaskan fitur utama sistem yang digunakan dalam mengolah data yang

akan disajikan. Tabel kebutuhan non fungsional menjelaskan fitur imbuhan yang dapat menunjang kinerja sistem.

Tabel 4. 1 Kebutuhan Fungsional Sistem

ID	Identifikasi
01	Sistem memiliki proses <i>login</i> dan <i>logout</i> sehingga setiap user memiliki hak dalam mengakses sistem.
02	Sistem memberikan hak akses kepada admin untuk menginputkan data bobot, menginputkan data user dan memberikan keputusan pemberian pinjaman.
03	Sistem memberikan hak akses kepada teller untuk mengupdate modal koperasi, menginputkan data nasabah dan menginputkan data pinjaman.
04	Sistem memberikan hak akses kepada nasabah untuk melihat pengumuman keputusan pemberian pinjaman.
05	Sistem mampu mengelola data <i>user</i> untuk melakukan <i>insert</i> dan <i>update</i> .
06	Sistem mampu mengelola data nasabah dan menyajikan siapa yang berhak menerima pinjaman berdasarkan kriteria yang telah diinputkan.

Tabel 4. 2 Kebutuhan Non Fungsional Sistem

ID	Parameter	Identifikasi
01	<i>Reliability</i>	Sistem dapat bekerja sesuai dengan fungsinya dan dapat dijalankan pada komputer dan <i>browser</i> yang berbeda.
02	<i>Availability</i>	Sistem selalu tersedia jika dijalankan serta mampu bekerja selama 24 jam.
03	<i>Comfortability</i>	Sistem memiliki tampilan dan desain yang sederhana serta mudah untuk digunakan.
04	<i>Security</i>	Sistem memiliki keamanan berupa identifikasi <i>username</i> dan <i>password</i> untuk

dilanjutkan

lanjutan

ID	Parameter	Identifikasi
		mengakses sistem.

**4.2 Desain Sistem**

Tahap desain sistem merupakan suatu tahapan dimana sistem yang akan dibuat akan diaplikasikan dalam bentuk dokumen sistem yang meliputi *Business Process*, *Usecase Diagram*, *Usecase Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class Diagram* dan *Entity Relationship Diagrams (ERD)*.

**4.2.1 Business Process**

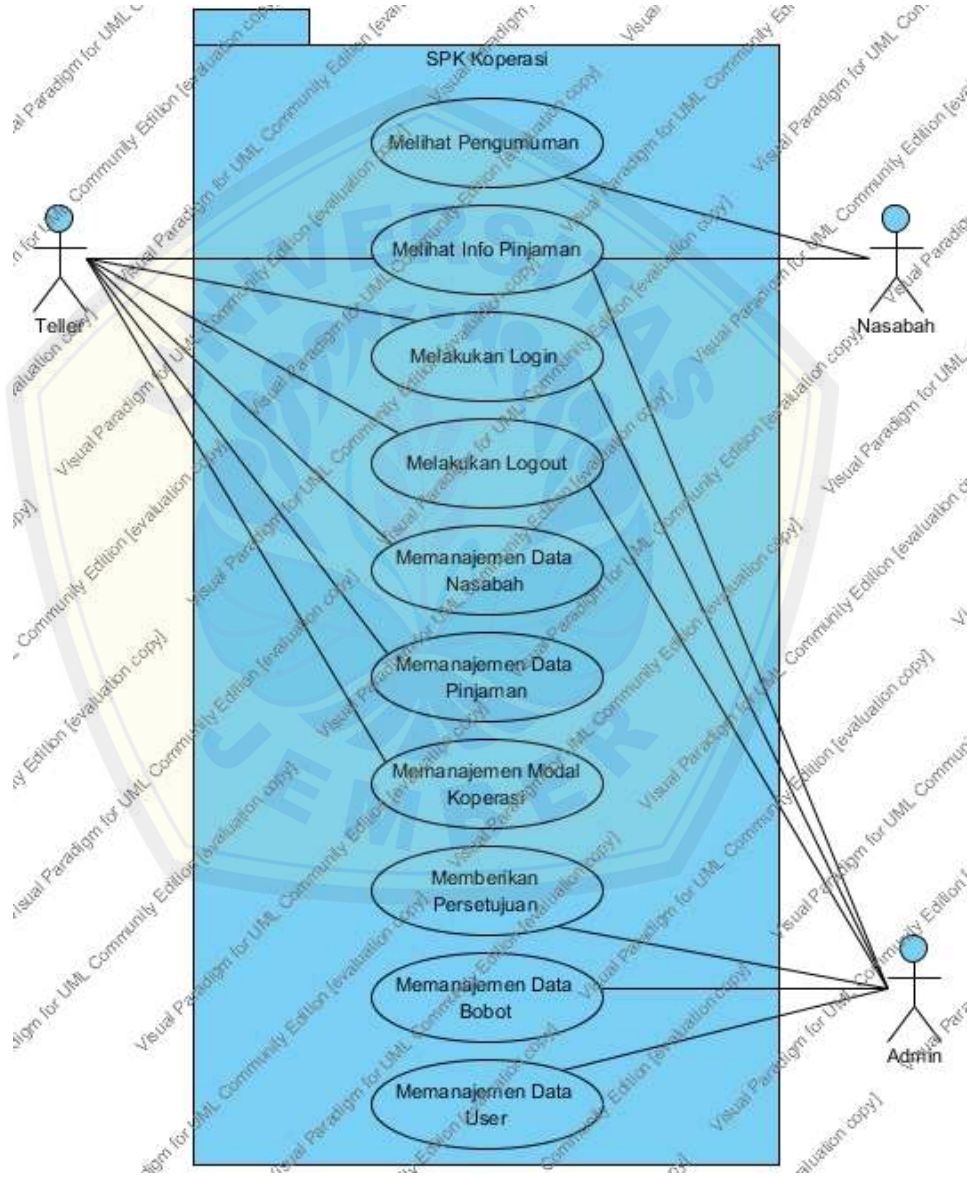
*Business Process* adalah sebuah proses untuk menggambarkan komponen-komponen dalam sistem. Komponen tersebut terdiri dari *input*, *output* dan *goal*. Masukan (*input*) dibutuhkan oleh sistem untuk dapat menghasilkan keluaran (*output*) yang sesuai dengan tujuan (*goal*). Proses *input* dilakukan oleh user yang kemudian dalam pengolahan data *output* akan dilakukan oleh sistem. *Business process* sistem penunjang keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dengan metode *weighted product* dapat dilihat pada Gambar 4.1.



Gambar 4.2 Business process sistem pendukung keputusan pemberian pinjaman koperasi

4.2.2 Usecase Diagram

Usecase Diagram adalah sekumpulan proses dalam sistem yang meliputi aktor dan fitur-fitur yang digunakan di dalam sistem. Gambar 4.2 berikut memperlihatkan detail usecase diagram pada sistem penunjang keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dengan metode *weighted product*.



Gambar 4. 3 Usecase diagram sistem pendukung keputusan pemberian pinjaman koperasi

Pada *usecase diagram* diatas dapat diketahui bahwa sistem memiliki tiga jenis aktor atau hak akses pengguna yang akan menjalankan sistem, yaitu admin, teller, dan nasabah. Penjelasan untuk hak akses pengguna sistem dapat dilihat pada tabel 4.3.

Tabel 4. 3 Hak akses sistem pendukung keputusan pemberian pinjaman koperasi

No	Aktor	Hak Akses
1	Admin	Admin merupakan aktor yang memiliki hak akses terhadap fitur memberikan persetujuan, manajemen data bobot dan manajemen data user.
2	Teller	Teller merupakan aktor yang memiliki hak akses terhadap fitur manajemen data nasabah, manajemen data pinjaman, dan manajemen modal koperasi.
3	Nasabah	Nasabah merupakan aktor yang hanya dapat melihat pengumuman dan melihat info pinjaman.

Sistem yang dirancang memiliki sepuluh fitur *usecase* yang mempunyai fungsi berbeda di setiap fiturnya. Tabel 4.4 berikut adalah penjelasan fitur pada *usecase*.

Tabel 4. 4 Deskripsi Usecase

No	Usecase	Deskripsi
1	Melihat Pengumuman	Fitur melihat pengumuman adalah fitur yang diutamakan untuk nasabah. Fitur ini menampilkan hasil keputusan pemberian pinjaman pada masing-masing nasabah.
2	Melihat Info Pinjaman	Fitur melihat info pinjaman adalah fitur yang dapat dilihat oleh semua aktor, tetapi

dilanjutkan



lanjutan

No	Usecase	Deskripsi
		fitur ini diutamakan untuk nasabah. Fitur ini menampilkan pinjaman apa saja yang disediakan oleh koperasi
3	Melakukan <i>Login</i>	Fitur <i>login</i> digunakan oleh user agar dapat masuk kedalam halaman sesuai level dari setiap user. Pada sistem ini terdapat dua level user yaitu admin dan teller.
4	Melakukan <i>Logout</i>	Fitur ini digunakan oleh user agar dapat keluar dari halaman yang menggunakan proses <i>login</i> .
5	Memanajemen Data Nasabah	Fitur manajemen data nasabah adalah fitur yang hanya dapat dilihat oleh teller. Fitur ini menampilkan data nasabah. Selain itu pada fitur ini dapat dilakukan <i>input</i> , <i>update</i> , dan <i>delete</i> data nasabah.
6	Memanajemen Data Pinjaman	Fitur manajemen data pinjaman adalah fitur yang hanya dapat dilihat oleh teller. Fitur ini menampilkan data pinjaman yang disediakan oleh koperasi. Selain itu pada fitur ini dapat dilakukan <i>input</i> , <i>update</i> , dan <i>delete</i> data pinjaman.
7	Memanajemen Modal Koperasi	Fitur manajemen data pinjaman adalah fitur yang hanya dapat dilihat oleh teller. Fitur ini menampilkan modal yang dimiliki oleh koperasi yang dapat menentukan jumlah maksimal pinjaman anggota koperasi. Pada fitur ini terdapat tombol <i>update</i> yang digunakan untuk merubah jumlah modal koperasi.
8	Memberikan	Fitur memberikan persetujuan adalah fitur

dilanjutkan

lanjutan

No	Usecase	Deskripsi
	Persetujuan	yang hanya dapat dilihat oleh admin. Fitur ini digunakan untuk menentukan apakah nasabah yang telah mengajukan pinjaman dan sudah dinilai oleh sistem berdasarkan metode <i>weighted product</i> disetujui atau tidak.
9	Memajemen Data Bobot	Fitur memajemen data bobot adalah fitur yang hanya dapat dilihat oleh admin. Fitur ini menampilkan data bobot dari setiap kriteria pinjaman. Selain itu pada fitur ini dapat dilakukan <i>update</i> data bobot.
10	Memajemen Data User	Fitur memajemen data user adalah fitur yang hanya dapat dilihat oleh admin. Fitur ini menampilkan data user yang dapat mengakses sistem. Selain itu pada fitur ini dapat dilakukan <i>input</i> , <i>update</i> , dan <i>delete</i> data user.

#### 4.2.2.1 Fitur Melihat Pengumuman

Fitur melihat pengumuman adalah fitur yang digunakan untuk melihat apakah pinjaman nasabah disetujui atau tidak mendapat pinjaman dari koperasi setelah kriteria nasabah yang diinputkan dihitung menggunakan metode *weighted product*. Fitur ini diakses oleh nasabah dimana untuk melihat fitur ini membutuhkan nama dan id nasabah.

##### a. *Usecase Scenario* Melihat Pengumuman

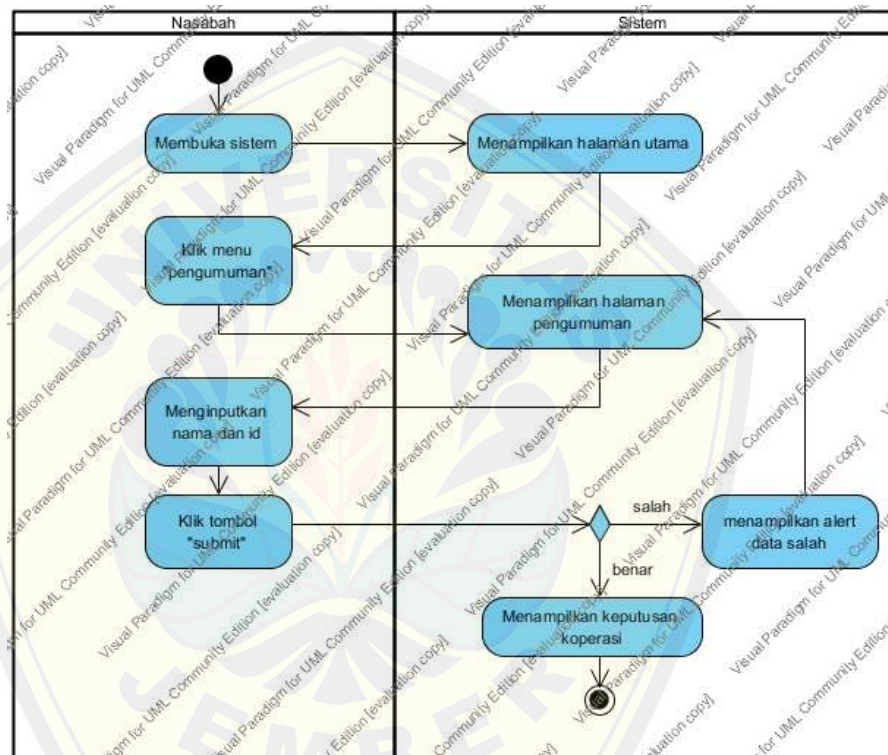
*Usecase scenario* adalah alur sistem yang diimplementasikan ke dalam bentuk tabel. *Usecase scenario* melihat pengumuman menjelaskan langkah untuk dapat melihat nasabah yang mendapat pinjaman dari koperasi dan reaksi sistem ketika mendapat aksi dari user. *Usecase scenario* melihat pengumuman dapat dilihat pada tabel 4.5.

Tabel 4. 5 *Usecase scenario* melihat pengumuman

1. Nama Usecase	: Melihat Pengumuman
2. Aktor	: Nasabah
3. Pra-kondisi	: Nasabah membuka aplikasi
4. Pasca-kondisi	: Nasabah telah melihat pengumuman
<b>Aliran Utama</b>	
<b>Nasabah</b>	<b>Sistem</b>
1. Membuka aplikasi	
	2. Menampilkan halaman utama dengan menu Pengumuman, Info Pinjaman, dan Login
3. Klik menu “pengumuman”	
	4. Menampilkan halaman pengumuman
5. Menginputkan nama dan id nasabah	
6. Klik tombol “submit”	
	7. Memverifikasi nama dan id nasabah
	8. Nama dan id valid
	9. Menampilkan data keputusan pinjaman nasabah sesuai nama dan id nasabah
<b>Aliran Alternatif</b>	
5. Menginputkan nama dan id nasabah	
6. Klik tombol “submit”	
	7. Memverifikasi nama dan id
	8. Nama dan id salah
	9. Menampilkan alert data salah

b. *Activity Diagram* Melihat Pengumuman

*Activity diagram* melihat pengumuman menggambarkan alir aktivitas sistem yang akan dibangun. Diagram diawali dengan nasabah membuka sistem dan mengklik menu pengumuman kemudian diakhiri dengan sistem menampilkan pengumuman nasabah. *Activity diagram* melihat pengumuman dapat dilihat pada gambar 4.4 berikut.

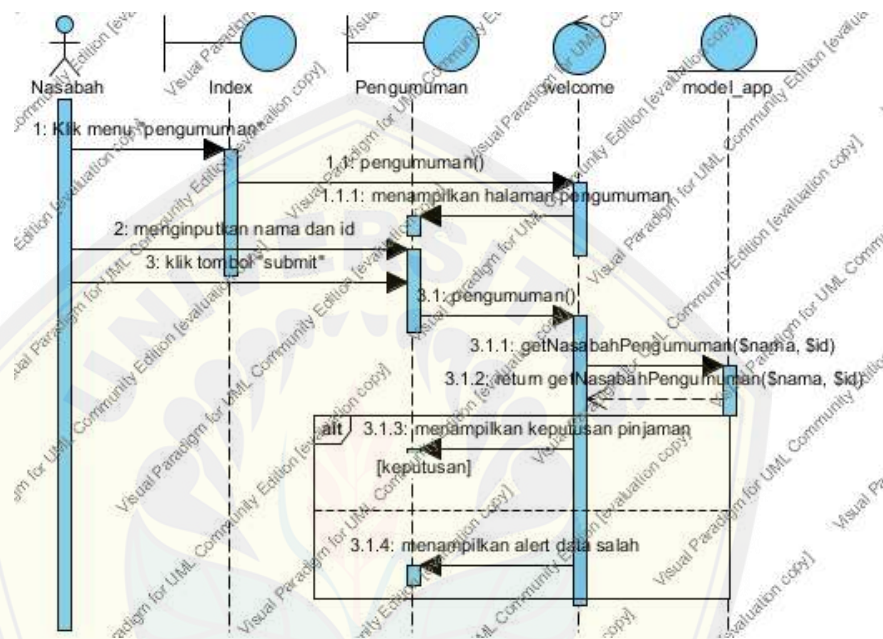


Gambar 4.4 *Activity diagram* melihat pengumuman

c. *Sequence Diagram* Melihat Pengumuman

*Sequence diagram* melihat pengumuman merupakan tahap lanjutan dari *activity diagram* melihat pengumuman sehingga alur prosesnya saling berkaitan. Pada diagram ini digambarkan interaksi antara halaman ‘index’, halaman ‘pengumuman’, *controller* ‘welcome’, dan model ‘model\_app’ dengan waktu yang digunakan. Ketika pesan ‘klik pengumuman’ dari user diterima, maka diartikan pesan menuju *controller* ‘welcome’ dan akan

ditampilkan halaman pengumuman. Ketika user menginputkan data dan klik tombol ‘submit’ maka data diteruskan menuju *controller* ‘welcome’ dan model ‘model\_app’ untuk memanggil data keputusan yang akan dilihat. Pesan yang sampai ke model ‘model\_app’ akan kembalikan pada *controller* ‘welcome’ dan akan ditampilkan pada halaman pengumuman. *Sequence diagram* melihat info nasabah dapat dilihat pada gambar 4.5.



Gambar 4.5 *Sequence diagram* melihat pengumuman

4.2.2.2 Fitur Melihat Info Pinjaman

Fitur melihat info pinjaman adalah fitur yang digunakan untuk melihat pinjaman apa saja yang disediakan oleh koperasi. Fitur ini juga dilengkapi dengan info selengkapnya dari setiap pinjaman yang disediakan koperasi. Fitur ini dapat diakses oleh semua user namun diutamakan untuk nasabah.

a. *Usecase Scenario* Melihat Info Pinjaman

*Usecase scenario* melihat info pinjaman menjelaskan langkah untuk dapat melihat jenis pinjaman yang disediakan oleh koperasi dan info selengkapnya dari jenis pinjaman yang disediakan serta reaksi sistem ketika mendapat aksi dari user. *Usecase scenario* melihat info pinjaman dapat dilihat pada lampiran A.1.

b. *Activity Diagram* Melihat Info Pinjaman

*Activity diagram* melihat info data pinjaman menggambarkan alir aktivitas sistem yang akan dibangun. Diagram diawali dengan nasabah membuka sistem dan mengklik data pinjaman kemudian diakhiri dengan sistem menampilkan halaman info data pinjaman selengkapnya. *Activity diagram* melihat info pinjaman dapat dilihat pada lampiran B.1.

c. *Sequence Diagram* Melihat Info Pinjaman

*Sequence diagram* melihat info pinjaman merupakan tahap lanjutan dari *activity diagram* melihat info pinjaman sehingga alur prosesnya saling berkaitan. Pada diagram ini digambarkan interaksi antara halaman 'index', halaman 'info pinjaman', *controller* 'welcome', dan model 'model\_app' dengan waktu yang digunakan. Ketika pesan 'klik info pinjaman' dari user diterima, maka diartikan pesan menuju *controller* 'welcome' dan diteruskan menuju model 'model\_app' untuk memanggil data pinjaman yang akan dilihat. Pesan yang sampai ke model 'model\_app' akan dikembalikan pada *controller* 'welcome' dan akan ditampilkan pada halaman info pinjaman. Ketika user mengirimkan pesan 'klik selengkapnya' sistem akan mengirimkan pesan pada *controller* 'welcome' dan diteruskan pada model 'model\_app' agar menampilkan data lengkap tentang jenis pinjaman. *Sequence diagram* melihat info nasabah dapat dilihat pada lampiran C.1.

#### 4.2.2.3 Fitur Melakukan Login

Fitur melakukan login adalah fitur yang digunakan untuk masuk dalam halaman akses admin dan halaman akses teller. Pada fitur ini user harus mempunyai *username* dan *password* untuk login. Fitur ini digunakan oleh admin dan teller.

a. *Usecase Scenario* Melakukan Login

*Usecase scenario login* merupakan gambaran sebuah fitur yang digunakan oleh teller dan admin agar dapat masuk halaman manajemen.

*Username* dan *password* yang diinputkan akan diverifikasi oleh sistem dan dilihat apakah user merupakan admin atau teller. Jika user adalah admin maka user akan diantarkan menuju index admin dan jika user adalah teller maka user akan diantarkan menuju halaman teller. Apabila *username* dan *password* yang diinputkan salah sistem akan menampilkan alert bahwa *username* dan *password* tidak sesuai. Lampiran A.2 merupakan *Usecase scenario login*.

b. *Activity Diagram Melakukan Login*

*Activity diagram* melakukan login memperlihatkan alir aktivitas sistem saat login. *Activity diagram login* diawali dengan user membuka aplikasi untuk melakukan login dan diakhiri dengan menampilkan halaman teller atau halaman admin. Diagram juga menampilkan bagaimana sistem mengeksekusi data yang salah. *Activity diagram* melakukan login dapat dilihat pada lampiran B.2.

c. *Sequence Diagram Melakukan Login*

*Sequence diagram login* menggambarkan interaksi antara halaman 'index', halaman 'login' halaman 'index\_admin', halaman 'index\_teller', *controller* 'welcome', *controller* 'c\_login', dan model 'm\_login' dengan waktu yang digunakan. Ketika pesan 'klik login' dari user diterima, maka diartakan pesan menuju *controller* 'welcome' dan akan ditampilkan halaman login. Pesan *username* dan *password* akan diverifikasi pada model 'm\_login' dan dikembalikan pada *controller* 'c\_login'. Apabila *username* dan *password* benar maka user akan diantarkan menuju halaman 'index\_admin' atau 'index\_teller'. Tetapi jika *username* dan *password* salah akan dikirimkan pesan pada index bahwa data tidak sesuai. *Sequence diagram* melakukan *login* dapat dilihat pada lampiran C.2.

#### 4.2.2.4 Fitur Melakukan *Logout*

Fitur melakukan *logout* adalah fitur yang digunakan teller dan admin untuk keluar dari halaman manajemen. Ketika teller atau admin *logout* maka teller atau admin akan diantarkan menuju halaman *index*.

##### a. *Usecase Scenario* Melakukan *Logout*

*Usecase scenario* melakukan *logout* merupakan gambaran fitur yang digunakan oleh teller dan admin untuk keluar dari halaman teller atau admin. *Usecase scenario* melakukan *logout* juga menjelaskan bagaimana aksi sistem ketika mendapat perintah dari user. *Usecase scenario* melakukan *logout* dapat dilihat pada lampiran A.3.

##### b. *Activity Diagram* Melakukan *Logout*

*Activity diagram* melakukan *logout* menggambarkan bagaimana alur sistem ketika user melakukan *logout*. Alur diawali dengan user mengklik menu *logout* dan diakhiri dengan sistem menampilkan halaman *index*. *Activity diagram* melakukan *logout* dapat dilihat pada lampiran B.3.

##### c. *Sequence Diagram* Melakukan *Logout*

*Sequence diagram logout* menggambarkan interaksi antara halaman 'index\_user', halaman 'index', dan *controller* 'c\_login' dengan waktu yang digunakan. Ketika pesan 'klik *logout*' dari user diterima, maka diartikan pesan menuju *controller* 'c\_login' dan akan ditampilkan halaman *index*. Lampiran C.3 merupakan *sequence diagram logout*.

#### 4.2.2.5 Fitur Memanajemen Data Nasabah

*Usecase scenario* manajemen data nasabah merupakan fitur yang dikhususkan bagi teller. Fitur ini digunakan untuk melihat, menginputkan, dan mengedit data nasabah. Pada fitur ini metode *weighted product* diterapkan. Ketika data nasabah yang diinputkan salah maka sistem akan menolak untuk menyimpan data.



a. *Usecase Scenario* Memanajemen Data Nasabah

*Usecase scenario* manajemen data nasabah menggambarkan bagaimana teller melihat, menginputkan, dan mengedit data nasabah. *Usecase scenario* manajemen data nasabah juga menggambarkan bagaimana reaksi sistem ketika teller membatalkan aksi dan reaksi sistem ketika data yang diinputkan oleh teller salah. *Usecase scenario* manajemen data nasabah dapat dilihat pada lampiran A.4.

b. *Activity Diagram* Memanajemen Data Nasabah

*Activity diagram* manajemen data nasabah menggambarkan alur teller dalam manajemen data nasabah. Alur diawali dengan teller mengklik menu data nasabah dan sistem menampilkan halaman data nasabah. Selanjutnya diperlihatkan alur saat teller menambahkan, dan mengedit data nasabah. Kemudian alur diakhiri dengan sistem menampilkan halaman data nasabah dengan data sesuai aksi yang telah dilakukan oleh teller sebelumnya. Lampiran B.4 merupakan *activity diagram* manajemen data nasabah.

c. *Sequence Diagram* Memanajemen Data Nasabah

*Sequence diagram* manajemen data nasabah menggambarkan interaksi antara halaman 'index\_teller', halaman 'data\_nasabah', halaman 'form\_nasabah', halaman 'edit\_nasabah', *controller* 'c\_teller', dan model 'model\_app' dengan waktu yang digunakan. *Sequence diagram* manajemen data nasabah memiliki lima alur alternatif diantaranya alternatif batal menambah data nasabah, menambah data nasabah yang salah, mengedit data nasabah, batal mengedit data nasabah, dan menyimpan data nasabah yang salah ketika selesai mengedit.

Alternatif menambah data nasabah yang salah digunakan apabila data nasabah yang diinputkan tidak sesuai. *Contoller* 'c\_teller' akan mengirimkan pesan pada halaman 'form\_nasabah' bahwa data nasabah yang diinputkan salah. Alternatif mengedit data nasabah digunakan apabila terdapat data nasabah yang ingin dirubah. *Contoller* 'c\_teller' akan mengirimkan pesan

pada halaman 'edit\_nasabah' data dari nasabah yang ingin dirubah. Dan alternatif menyimpan data nasabah yang salah ketika selesai mengedit digunakan ketika teller menyimpan data nasabah yang telah dirubah namun data yang dimasukkan salah. *Contoller* 'c\_teller' akan mengirimkan pesan pada halaman 'edit\_nasabah' bahwa data yang diinputkan salah. *Sequence diagram* manajemen data nasabah dapat dilihat pada lampiran C.4.

#### 4.2.2.6 Fitur Manajemen Data Pinjaman

Fitur manajemen data pinjaman merupakan fitur yang dikhususkan bagi teller. Fitur ini digunakan untuk melihat, menginputkan, dan mengedit data jenis pinjaman yang disediakan oleh koperasi. Hasil dari fitur manajemen data pinjaman ini yang nantinya akan ditampilkan pada fitur melihat info pinjaman dan dilihat oleh nasabah..

##### a. *Usecase Scenario* Manajemen Data Pinjaman

*Usecase scenario* manajemen data pinjaman menggambarkan bagaimana jalannya untuk dapat melihat, menambah, dan mengedit data pinjaman. *Usecase scenario* manajemen data pinjaman juga menggambarkan bagaimana sistem melakukan aksi ketika mendapat perintah dari teller. Pada *usecase scenario* ini terdapat lima alternatif alur yaitu batal menambahkan data pinjaman, menambah data pinjaman yang salah, mengedit data pinjaman, batal mengedit data pinjaman, dan menyimpan data hasil editan yang salah. *Usecase scenario* manajemen data pinjaman dapat dilihat pada lampiran A.5.

##### b. *Activity Diagram* Manajemen Data Pinjaman

*Activity diagram* manajemen data pinjaman menggambarkan alur teller dalam manajemen data pinjaman. Alur diawali dengan teller mengklik menu data pinjaman kemudian sistem menampilkan halaman data pinjaman. Selanjutnya diperlihatkan alur saat teller menambahkan dan mengedit data pinjaman. Alur diakhiri dengan sistem menampilkan halaman data pinjaman dengan data sesuai aksi yang telah dilakukan oleh teller

sebelumnya. Lampiran B.5 merupakan *activity diagram* manajemen data pinjaman.

#### c. *Sequence Diagram* Manajemen Data Pinjaman

*Sequence diagram* manajemen data pinjaman menggambarkan interaksi antara halaman 'index\_teller', halaman 'data\_pinjaman', halaman 'tambah\_pinjaman', halaman 'edit\_pinjaman', *controller* 'c\_teller', dan model 'model\_app' dengan waktu yang digunakan. *Sequence diagram* manajemen data pinjaman memiliki lima alur alternatif diantaranya alternatif batal menambah pinjaman, menambah data pinjaman yang salah, mengedit data pinjaman, batal mengedit data pinjaman, dan menyimpan data pinjaman hasil editan yang salah.

Alternatif menambah data pinjaman yang salah digunakan apabila data pinjaman yang diinputkan tidak sesuai. *Controller* 'c\_teller' akan mengirimkan pesan pada halaman 'tambah\_pinjaman' bahwa data pinjaman yang diinputkan salah. Alternatif mengedit data pinjaman digunakan apabila data pinjaman yang diinputkan untuk mengedit data pinjaman tidak sesuai. *Controller* 'c\_teller' akan mengirimkan pesan pada halaman 'edit\_pinjaman' bahwa data pinjaman yang diinputkan salah. *Sequence diagram* manajemen data pinjaman nasabah dapat dilihat pada lampiran C.5.

#### 4.2.2.7 Fitur Manajemen Modal Koperasi

Fitur manajemen modal koperasi adalah fitur yang digunakan untuk melihat dan mengedit data modal yang dimiliki koperasi. Data modal koperasi digunakan untuk melihat berapa pinjaman maksimal dari masing-masing nasabah. Jumlah pinjaman maksimal masing-masing nasabah diketahui dengan membagi jumlah modal koperasi dengan jumlah anggota koperasi.

##### a. *Usecase Scenario* Manajemen Modal Koperasi

*Usecase scenario* manajemen modal koperasi menggambarkan bagaimana agar teller dapat melihat dan mengedit data modal koperasi.

*Usecase scenario* manajemen modal koperasi juga menggambarkan bagaimana sistem berjalan ketika mendapat perintah dari teller. *Usecase scenario* manajemen modal koperasi dapat dilihat pada lampiran A.6.

b. *Activity Diagram* Manajemen Modal Koperasi

*Activity diagram* manajemen modal koperasi menggambarkan alur teller dalam manajemen modal koperasi. Alur diawali dengan teller mengklik menu modal kemudian sistem menampilkan halaman modal koperasi. Selanjutnya diperlihatkan alur saat teller mengedit modal koperasi. Alur diakhiri dengan sistem menampilkan halaman modal koperasi dengan data sesuai aksi yang telah dilakukan oleh teller sebelumnya. *Activity diagram* manajemen modal koperasi dapat dilihat pada lampiran B.6.

c. *Sequence Diagram* Manajemen Modal Koperasi

*Sequence diagram* manajemen modal koperasi menggambarkan interaksi antara halaman 'index\_teller', halaman 'modal', halaman 'edit\_modal', *controller* 'c\_teller', dan model 'model\_app' dengan waktu yang digunakan. Ketika pesan 'klik modal' dari teller diterima, maka diartikan pesan menuju *controller* 'c\_teller' dan akan ditampilkan halaman modal. Ketika user mengedit data dan klik tombol 'update' maka data diteruskan menuju *controller* 'c\_teller' dan model 'model\_app' untuk menyimpan data yang telah dirubah. Pesan yang sampai ke model 'model\_app' akan kembalikan pada *controller* 'c\_teller' dan akan ditampilkan pada halaman modal. *Sequence diagram* manajemen modal koperasi dapat dilihat pada lampiran C.6.

#### 4.2.2.8 Fitur Memberikan Persetujuan

Fitur memberikan persetujuan adalah fitur yang digunakan oleh admin untuk memberikan keputusannya. Fitur ini yang menentukan apakah data nasabah yang telah dinilai disetujui atau tidak oleh koperasi. Data yang telah diberikan keputusan dapat dilihat pada halaman pengumuman oleh nasabah.

a. *Usecase Scenario* Memberikan Persetujuan

*Usecase scenario* memberikan keputusan menggambarkan bagaimana admin dapat memberikan keputusan persetujuan pada data nasabah. *Usecase scenario* memberikan persetujuan juga menggambarkan bagaimana sistem berjalan ketika mendapat perintah dari admin. *Usecase scenario* memberikan persetujuan dapat dilihat pada lampiran A.7.

b. *Activity Diagram* Memberikan Persetujuan

*Activity diagram* memberikan persetujuan menggambarkan alur admin dalam memberikan persetujuan. Alur diawali dengan admin mengklik menu persetujuan kemudian sistem menampilkan halaman persetujuan. Selanjutnya diperlihatkan alur saat admin klik tombol setuju atau tidak setuju. Alur diakhiri dengan sistem menampilkan halaman persetujuan dengan data sesuai aksi yang telah dilakukan oleh admin sebelumnya. *Activity diagram* manajemen modal koperasi dapat dilihat pada lampiran B.7.

c. *Sequence Diagram* Memberikan Persetujuan

*Sequence diagram* memberikan persetujuan menggambarkan interaksi antara halaman 'index\_admin', halaman 'persetujuan', *controller* 'c\_admin', dan model 'model\_app' dengan waktu yang digunakan. Ketika pesan 'klik persetujuan' dari admin diterima, maka diartikan pesan menuju *controller* 'c\_admin' dan akan ditampilkan halaman persetujuan. Ketika admin mengklik tombol setuju atau tidak setuju maka data diteruskan menuju *controller* 'c\_admin' dan model 'model\_app' untuk menyimpan data yang telah dirubah. Pesan yang sampai ke model 'model\_app' akan kembalikan pada *controller* 'c\_teller' dan akan ditampilkan pada halaman persetujuan. *Sequence diagram* memberikan persetujuan dapat dilihat pada lampiran C.7.

#### 4.2.2.9 Fitur Manajemen Data Bobot

Fitur manajemen data bobot merupakan fitur yang dikhususkan bagi admin. Fitur ini digunakan untuk melihat, dan mengedit nilai bobot

kriteria yang digunakan dalam menentukan nilai nasabah. Data bobot yang disimpan akan dihitung untuk mendapatkan nilai pangkat akhir. Nilai pangkat akhir ini yang akan digunakan untuk melakukan penilaian pada nasabah sehingga mendapatkan keputusan pemberian pinjaman.

a. *Usecase Scenario* Memanajemen Data Bobot

*Usecase scenario* manajemen data bobot menggambarkan bagaimana untuk dapat melihat dan mengedit data bobot. *Usecase scenario* manajemen data bobot juga menjelaskan bagaimana aksi sistem ketika mendapat perintah dari admin dan bagaimana sistem ketika total nilai pangkat akhir bobot tidak sama dengan satu. *Usecase scenario* manajemen data bobot dapat dilihat pada lampiran A.8.

b. *Activity Diagram* Memanajemen Data Bobot

*Activity diagram* manajemen data bobot menggambarkan bagaimana admin manajemen data bobot dan aksi apa yang diberikan sistem. Alur diawali dengan admin mengklik data bobot dan sistem menampilkan halaman data bobot. Selanjutnya admin mengedit data bobot kemudian sistem menghitung data bobot dan memasukkannya dalam database. Alur diakhiri dengan sistem menampilkan halaman data bobot setelah dilakukan pengeditan data bobot. *Activity diagram* manajemen data bobot dapat dilihat pada lampiran B.8.

c. *Sequence Diagram* Memanajemen Data Bobot

*Sequence diagram* manajemen data bobot menggambarkan interaksi antara halaman 'index\_admin', halaman 'data\_bobot', halaman 'edit\_bobot', *controller* 'c\_admin', dan model 'model\_app' dengan waktu yang digunakan. *Sequence diagram* manajemen bobot nasabah menggambarkan bagaimana proses melihat dan mengganti nilai bobot sehingga dapat digunakan dalam menghitung nilai nasabah. *Sequence diagram* manajemen data bobot juga menggambarkan aksi sistem ketika total

pangkat bobot tidak sama dengan 1. *Sequence diagram* manajemen data bobot dapat dilihat pada lampiran C.8.

#### 4.2.2.10 Fitur Manajemen Data User

Fitur manajemen data user merupakan fitur yang dikhususkan bagi admin. Fitur ini digunakan untuk melihat, menginputkan, dan mengedit data user yang dapat mengakses halaman manajemen. Ketika admin memasukkan data yang salah atau *password* dan *repassword* tidak sama maka sistem akan menampilkan alert bahwa data salah.

##### a. *Usecase Scenario* Manajemen Data User

*Usecase scenario* manajemen data user menggambarkan bagaimana alur untuk dapat melihat, menginputkan, dan mengedit data user. *Usecase scenario* manajemen data user juga menggambarkan bagaimana aksi sistem ketika mendapat perintah dari admin dan ketika data yang diinputkan tidak sesuai. *Usecase scenario* manajemen data user dapat dilihat pada lampiran A.9.

##### b. *Activity Diagram* Manajemen Data User

*Activity diagram* manajemen data user menggambarkan alur admin dalam manajemen data user. Alur diawali dengan admin mengklik menu data user dan sistem menampilkan halaman data user. Selanjutnya diperlihatkan alur saat admin menambahkan dan mengedit data user. Alur diakhiri dengan sistem menampilkan halaman data user dengan data sesuai aksi yang telah dilakukan oleh admin sebelumnya. *Activity diagram* manajemen data user dapat dilihat pada lampiran B.9.

##### c. *Sequence Diagram* Manajemen Data User

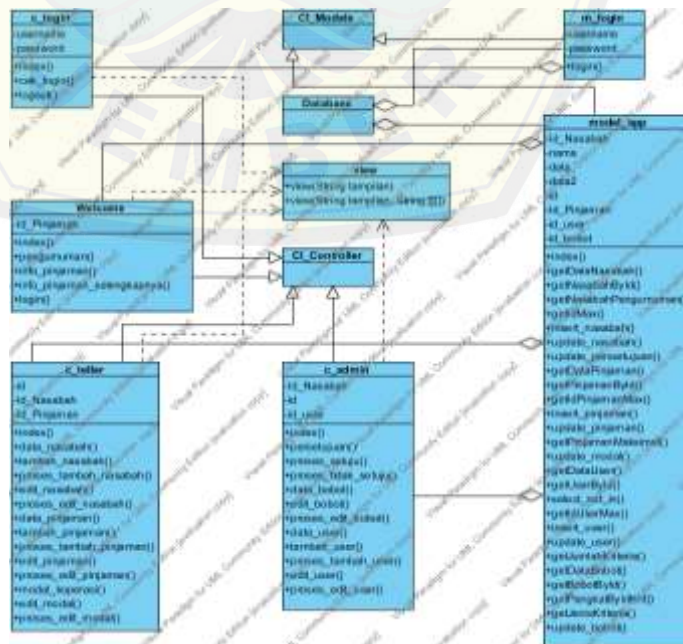
*Sequence diagram* manajemen data user menggambarkan interaksi antara halaman 'index\_admin, halaman 'data\_user, halaman 'tambah\_user, halaman 'edit\_user, *controller* 'c\_admin, dan model 'model\_app' dengan waktu yang digunakan. *Sequence diagram* manajemen data user memiliki

lima alur alternatif diantaranya alternatif batal menambahkan user, menambahkan data user yang salah, mengedit data user, batal mengedit data user, dan menyimpan hasil editan data user yang salah.

Alternatif menambahkan user yang salah digunakan apabila data user yang diinputkan tidak sesuai. *Contoller* 'c\_admin akan mengirimkan pesan pada halaman 'tambah\_user' bahwa data user yang diinputkan salah. Alternatif menyimpan editan user yang salah digunakan apabila data user yang diinputkan untuk mengedit tidak sesuai. *Contoller* 'c\_admin akan mengirimkan pesan pada halaman 'edit\_user' bahwa data user yang diinputkan salah. Lampiran C.9 merupakan *sequence diagram* manajemen data user.

#### 4.2.3 Class Diagram

*Class diagram* adalah diagram yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Gambar 4.6 merupakan *class diagram* sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri yang mempunyai dua model dan empat *controller* yang saling berhubungan.



Gambar 4.6 Class diagram



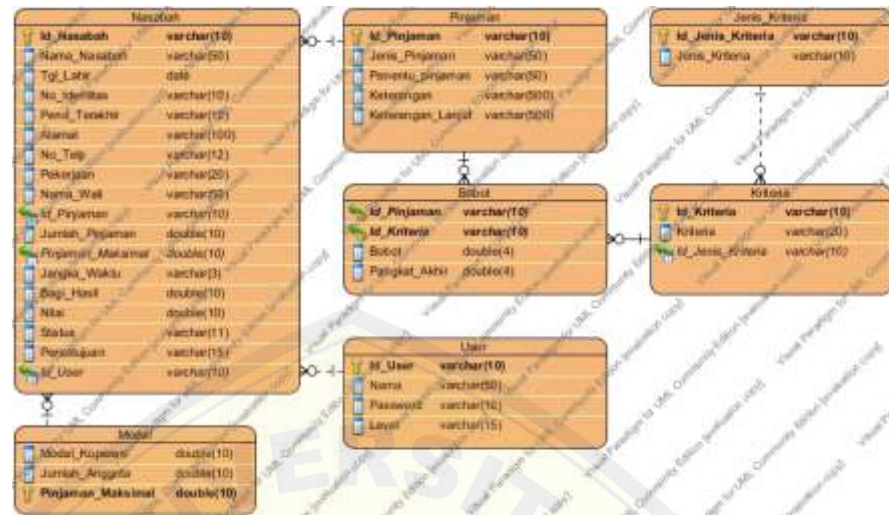
#### 4.2.4 Entity Relationship Diagrams (ERD)

*Entity Relationship Diagrams (ERD)* adalah model yang digunakan untuk menjelaskan hubungan antar data dalam basisdata berdasarkan objek-objek dasar data yang mempunyai relasi. Sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri mempunyai tujuh entitas diantaranya user, nasabah, modal, pinjaman, bobot, kriteria, dan jenis kriteria.

Entitas user memiliki empat atribut yaitu `id_user`, `nama`, `password`, dan `level`. Entitas nasabah memiliki 18 atribut yaitu `id_nasabah`, `nama_nasabah`, `tgl_lahir`, `no_identitas`, `pend_terakhir`, `alamat`, `no_telp`, `pekerjaan`, `nama_wali`, `id_pinjaman`, `jumlah_pinjaman`, `pinjaman_maksimal`, `jangka_waktu`, `bagi_hasil`, `nilai`, `status`, `persetujuan`, dan `id_user`. Entitas modal memiliki tiga atribut yaitu `modal_koperasi`, `jumlah_anggota`, `pinjaman_maksimal`. Entitas pinjaman memiliki lima atribut yaitu `id_pinjaman`, `jenis_pinjaman`, `penentu_pinjaman`, `keterangan`, dan `keterangan_lanjut`. Entitas bobot memiliki empat atribut yaitu `id_pinjaman`, `id_kriteria`, `bobot`, dan `pangkat_akhir`. Entitas kriteria memiliki tiga atribut yaitu `id_kriteria`, `kriteria` dan `id_jenis_kriteria`. Sedangkan entitas `jenis_kriteria` memiliki dua atribut yaitu `id_jenis_kriteria` dan `jenis_kriteria`.

Hubungan antara entitas user dan entitas nasabah adalah *one-to-many*, yaitu satu user bisa menginputkan banyak nasabah. Hubungan entitas modal dengan entitas nasabah adalah *one-to-many*, yaitu jumlah pinjaman maksimal yang sama bisa digunakan oleh banyak nasabah. Hubungan antara entitas pinjaman dan entitas nasabah juga *one-to-many*, yaitu satu jenis pinjaman bisa diajukan oleh banyak nasabah. Hubungan antara entitas pinjaman dan entitas bobot adalah *one-to-many*, yaitu satu jenis pinjaman bisa memiliki bobot yang banyak. Hubungan antara entitas kriteria dan entitas bobot adalah *one-to-many*, yaitu satu jenis kriteria bisa memiliki bobot yang banyak. Hubungan antara entitas `jenis_kriteria` dan entitas kriteria adalah *one-to-many*, yaitu satu jenis kriteria bisa menjadi jenis dari banyak kriteria. Gambar 4.7 adalah *Entity Relationship Diagrams (ERD)* sistem pendukung keputusan

pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri.



Gambar 4.7 Entity relationship diagrams (ERD)

#### 4.2.5 Pembangunan Prototipe

Langkah selanjutnya setelah tahap analisis dan tahap perancangan sistem adalah tahap pemodelan prototipe. Tahap pemodelan prototipe merupakan tahap yang menyajikan tampilan sistem dalam bentuk desain *interface* yang sesuai dengan kebutuhan fungsional yang telah didefinisikan.

##### 4.2.5.1 Melihat Pengumuman

Model prototipe melihat pengumuman adalah halaman yang digunakan nasabah dalam melihat keputusan pinjaman dari koperasi. Pada atas halaman ini terdapat menu pengumuman, info pinjaman, dan login. Pada tengah halaman terdapat kolom nama dan kolom id yang digunakan oleh nasabah menginputkan namanya dan tombol submit yang digunakan untuk menampilkan data persetujuan berdasarkan nama dan id yang diinputkan oleh nasabah. Gambar 4.8 berikut merupakan model prototipe melihat pengumuman.



Gambar 4.8 Model prototipe melihat pengumuman

#### 4.2.5.2 Melihat Info Pinjaman

Model prototipe melihat info pinjaman adalah halaman yang digunakan nasabah dalam melihat jenis pinjaman yang disediakan oleh koperasi. Pada atas halaman ini terdapat menu pengumuman, info pinjaman, dan login. Pada tengah halaman akan berisi jenis-jenis pinjaman yang disediakan oleh koperasi dan tombol selengkapnya. Apabila tombol selengkapnya diklik maka akan menuju ke halaman detail dari jenis pinjaman yang diklik. Gambar model prototipe melihat info pinjaman dapat dilihat di lampiran D.1 dan D.2.

#### 4.2.5.3 Melakukan *Login*

Model prototipe melakukan *login* adalah halaman yang digunakan user agar dapat masuk ke halaman manajemen. Pada atas halaman ini terdapat menu pengumuman, info pinjaman, dan login. Pada tengah halaman terdapat kolom *username* dan *password* serta tombol login. Apabila data benar maka akan masuk ke halaman teller seperti pada lampiran D.5 atau halaman admin seperti pada lampiran D.6. Jika data salah maka sistem akan menampilkan alert data salah seperti pada lampiran D.4. Gambar model prototipe melakukan *login* dapat dilihat di lampiran D.3.

#### 4.2.5.4 Melakukan *Logout*

Model prototipe melakukan *logout* adalah halaman yang digunakan user agar dapat keluar dari halaman manajemen. Ketika user klik tombol *logout* maka sistem akan mengantarkan user menuju halaman *index*. Model prototipe melakukan *logout* dapat dilihat pada lampiran D.7.

#### 4.2.5.5 Memanajemen Data Nasabah

Model prototipe memajemen data nasabah adalah halaman yang digunakan teller untuk melihat, menambah, dan mengedit data nasabah. Model prototipe melihat data nasabah dapat dilihat pada lampiran D.8. Model prototipe menambah data nasabah dapat dilihat pada lampiran D.9. Model prototipe mengedit data nasabah dapat dilihat pada lampiran D.10.

#### 4.2.5.6 Memanajemen Data Pinjaman

Model prototipe memajemen data pinjaman adalah halaman yang digunakan teller untuk melihat, menambah, dan mengedit data pinjaman. Model prototipe melihat data pinjaman dapat dilihat pada lampiran D.11. Model prototipe menambah data pinjaman dapat dilihat pada lampiran D.12. Model prototipe mengedit data pinjaman dapat dilihat pada lampiran D.13.

#### 4.2.5.7 Memanajemen Modal Koperasi

Model prototipe memajemen modal koperasi adalah halaman yang digunakan teller untuk melihat dan mengedit data pinjamodal koperasi. Model prototipe melihat modal koperasi dapat dilihat pada lampiran D.14. Dan model prototipe mengedit modal koperasi dapat dilihat pada lampiran D.15.

#### 4.2.5.8 Memberikan Persetujuan

Model prototipe memberikan persetujuan adalah halaman yang digunakan admin dalam menentukan nasabah mana yang pinjamannya

disetujui oleh koperasi. Model prototipe memberikan persetujuan dapat dilihat pada lampiran D.16.

#### 4.2.5.9 Memenajemen Data Bobot

Model prototipe manajemen data bobot adalah halaman yang digunakan admin untuk melihat dan mengedit data bobot. Model prototipe melihat data bobot dapat dilihat pada lampiran D.17. Dan model prototipe mengedit data bobot dapat dilihat pada lampiran D.18.

#### 4.2.5.10 Memanajemen Data User

Model prototipe manajemen data user adalah halaman yang digunakan admin untuk melihat, menambah, dan mengedit data user. Model prototipe melihat data user dapat dilihat pada lampiran D.19. Model prototipe menambah data user dapat dilihat pada lampiran D.20. Model prototipe mengedit data user dapat dilihat pada lampiran D.21.

### 4.3 Implementasi

Tahap pengkodean sistem adalah tahap pengimplementasian rancangan sistem dalam bentuk *listing* program. Tahap pembangunan sistem ini bertujuan agar sistem yang telah dirancang dapat digunakan sesuai kebutuhan fungsional dan *non-fungsional* sistem. Tahap pengkodean sistem dilakukan dengan menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*) dengan *framework* CI (*Code Igniter*) yang mendukung pengembangan model berbasis MVC (*model, view* dan *Controller*) dan *Bootstrap*.

#### 4.3.1 Fitur Melihat Pengumuman

Fitur melihat pengumuman adalah fitur yang digunakan nasabah untuk melihat persetujuan pinjaman dari koperasi. Untuk mengakses fitur ini dibutuhkan nama dan id nasabah. Baris kode fitur melihat pengumuman dapat dilihat pada gambar 4.9.

```

public function pengumuman()
{
    $nama = $this->input->post('nama');
    $id = $this->input->post('id');

    $query = $this->model_app->getNasabahPengumuman($nama, $id);

    $data["data"] = $query->result();
    $data["total"] = $query->num_rows();

    $this->load->view('pengumuman', $data);
}

function getNasabahPengumuman($nama, $id) {
    return $this->db->query("SELECT * FROM nasabah WHERE p_nama = '$nama' AND p_id = '$id'");
}

```

Gambar 4.9 Baris kode fitur melihat pengumuman

Baris program gambar diatas menunjukkan bahwa yang pertama kali ditampilkan oleh sistem adalah halaman pengumuman. Selanjutnya ketika nasabah penginputkan nama dan id kemudian klik tombol 'submit' maka *controller* akan mengakses 'getNasabahPengumuman' yang artinya sistem memanggil data nasabah berdasarkan nama dan id yang diteloh diinputkan.

#### 4.3.2 Fitur melihat Info Pinjaman

Fitur melihat info pinjaman adalah fitur yang digunakan nasabah untuk melihat pinjaman yang disediakan oleh koperasi. Fitur melihat pinjaman ini memiliki dua halaman yaitu info pinjaman dan info pinjaman selengkapnya. Baris kode fitur melihat info pinjaman dapat dilihat pada lampiran E.1.

Baris program pada lampiran E.1 menunjukkan bahwa yang pertama kali ditampilkan oleh sistem adalah halaman info pinjaman. Selanjutnya ketika nasabah klik tombol 'selengkapnya' maka *controller* akan mengakses 'getPinjamanById' yang artinya sistem memanggil data pinjaman berdasarkan id pinjaman yang diklik.

#### 4.3.3 Fitur Melakukan Login

Fitur melihat *login* adalah fitur yang digunakan *user* untuk masuk kedalam halaman manajemen. Untuk mengakses fitur ini *user* harus mempunyai *username* dan *password*. Baris kode fitur melakukan *login* dapat dilihat pada lampiran E.2.

Baris program pada lampiran E.2 menunjukkan bahwa ketika user menginputkan *username* dan *password* dan klik tombol 'login' maka *controoler* akan mengeksekusi `login($username,$password)` yang artinya memanggil data *username* dan *password*. Selanjutnya *controller* mengecek data yang diinputkan *user*. Jika data benar atau sesuai dengan database maka program akan menampilkan *index user*, tetapi jika salah sistem akan menampilkan alert data salah.

#### 4.3.4 Fitur Melakukan Logout

Fitur melihat *logout* adalah fitur yang digunakan *user* untuk keluar dari halaman manajemen. User cukup klik menu *logout* untuk keluar halaman. Baris kode fitur melakukan *logout* dapat dilihat pada lampiran E.3. Baris kode pada lampiran E.3 menunjukkan bahwa ketika *user* klik menu *logout controller* akan membawa *user* ke halaman *index*.

#### 4.3.5 Fitur Memanajemen Data Nasabah

Fitur manajemen data nasabah adalah fitur yang digunakan *teller* untuk melihat, menambah dan mengedit data nasabah. Fitur manajemen data nasabah ini memiliki tiga halaman yaitu data nasabah, halaman form nasabah, dan halaman edit nasabah. Baris kode fitur manajemen data nasabah dapat dilihat pada lampiran E.4.

Baris program pada lampiran E.4 menunjukkan bahwa yang pertama kali ditampilkan oleh sistem adalah halaman data nasabah. Selanjutnya fungsi `tambah_nasabah($id)` digunakan untuk menampilkan halaman nasabah. *Controller* memanggil *query* `getPinjamanById($id)` untuk menampilkan jenis pinjaman yang dapat dipilih ketika akan menginputkan data nasabah. Kemudian `getDataPinjaman()` digunakan untuk menampilkan penentu pinjaman masing-masing jenis pinjaman. Dan `getPinjamanMaksimal()` digunakan untuk menginputkan jumlah pinjaman maksimal setiap nasabah.

Fungsi `proses_tambah_nasabah()` digunakan untuk memproses data nasabah untuk disimpan ke dalam database dan menghitung nilai nasabah

dengan metode *weighted product*. *Statment* `getPangkatByIdKrit` digunakan untuk memanggil nilai pangkat yang akan digunakan untuk menghitung nilai nasabah. Dan `insert_nasabah($data)` digunakan untuk menyimpan data nasabah yang telah diambil dari form nasabah.

Fungsi selanjutnya pada fitur manajemen data nasabah adalah `edit_nasabah($Id_Nasabah)` dan `proses_ edit_nasabah($Id_Nasabah)`. *Statment* `getNasabahById($Id_Nasabah)` digunakan untuk menampilkan data nasabah yang dipilih ke halaman edit nasabah. Dan `update_nasabah($Id_Nasabah, $data2)` digunakan untuk menyimpan data nasabah hasil editan.

#### 4.3.6 Fitur Manajemen Data Pinjaman

Fitur manajemen data pinjaman adalah fitur yang digunakan teller untuk melihat, menambah dan mengedit data pinjaman. Fitur manajemen data pinjaman ini memiliki tiga halaman yaitu data pinjaman, halaman form pinjaman, dan halaman edit pinjaman. Baris kode fitur manajemen data pinjaman dapat dilihat pada lampiran E.5.

Baris program pada lampiran E.5 menunjukkan bahwa yang pertama kali ditampilkan oleh sistem adalah halaman data pinjaman. Selanjutnya fungsi `tambah_pinjaman($id_pinjaman)` digunakan untuk menampilkan halaman pinjaman. Pada fungsi `tambah_pinjaman()` *controller* memanggil *query* `getIdPinjamanMax()` untuk menampilkan id pinjaman secara otomatis sehingga tidak menyebabkan id pinjaman berulang.

Fungsi `proses_tambah_ pinjaman()` digunakan untuk memproses data pinjaman untuk disimpan ke dalam. *Statment* `insert_pinjaman ($data)` digunakan untuk menyimpan data pinjaman yang telah diambil dari form nasabah. Fungsi selanjutnya pada fitur manajemen data pinjaman adalah `edit_pinjaman($Id_pinjaman)` dan `proses_edit_pinjaman($Id_pinjaman)`. *Statment* `getPinjamanById($Id_pinjaman)` digunakan untuk menampilkan data pinjaman yang dipilih ke halaman edit pinjaman. Dan



`update_pinjaman($Id_pinjaman, $data2)` digunakan untuk menyimpan data pinjaman hasil editan.

#### 4.3.7 Fitur Memanajemen Modal Koperasi

Fitur manajemen modal koperasi adalah fitur yang digunakan teller untuk melihat dan mengedit modal koperasi. Fitur manajemen modal koperasi ini memiliki dua halaman yaitu data modal dan halaman edit modal. Baris kode fitur manajemen modal koperasi dapat dilihat pada lampiran E.6.

Baris program pada lampiran E.6 menunjukkan bahwa yang pertama kali ditampilkan oleh sistem adalah halaman modal koperasi. Fungsi selanjutnya pada fitur manajemen modal koperasi adalah `edit_modal()` dan `proses_edit_modal()`. *Statment* `update_modal($data)` digunakan untuk menyimpan modal koperasi yang telah dirubah.

#### 4.3.8 Fitur Memberikan Persetujuan

Fitur memberikan persetujuan adalah fitur yang digunakan admin untuk memberikat keputusan dari koperasi untuk pinjaman nasabah. Fitur memberikan persetujuan ini memiliki satu halaman yaitu halaman persetujuan. Baris kode fitur memberikan persetujuan dapat dilihat pada lampiran E.8.

Baris program pada lampiran E.8 menunjukkan bahwa yang pertama kali ditampilkan oleh sistem adalah halaman persetujuan. Fungsi selanjutnya pada fitur memberikan persetujuan adalah `proses_setuju($Id_Nasabah)` dan `proses_tidak_setuju($Id_Nasabah)`. *Statment* `update_pesetujuan($Id_Nasabah,$data)` digunakan untuk menyimpan data nasabah yang telah diberikan persetujuan.

#### 4.3.9 Fitur Memanajemen Data Bobot

Fitur manajemen data bobot adalah fitur yang digunakan admin untuk melihat dan mengedit data bobot. Fitur manajemen data bobot ini

memiliki dua halaman yaitu data bobot dan halaman edit bobot. Baris kode fitur manajemen data bobot dapat dilihat pada lampiran E.8.

Baris program pada lampiran E.8 menunjukkan bahwa yang pertama kali ditampilkan oleh sistem adalah halaman data bobot. Fungsi selanjutnya pada fitur manajemen data bobot adalah `edit_bobot()` dan `proses_edit_bobot()`. *Statment* `update_bobot($data)` digunakan untuk menyimpan data bobot yang telah dirubah.

#### 4.3.10 Fitur Manajemen Data User

Fitur manajemen data user adalah fitur yang digunakan admin untuk melihat, menambah dan mengedit data user. Fitur manajemen data user ini memiliki tiga halaman yaitu data user, halaman form user, dan halaman edit user. Baris kode fitur manajemen data user dapat dilihat pada lampiran E.9.

Baris program pada lampiran E.9 menunjukkan bahwa yang pertama kali ditampilkan oleh sistem adalah halaman data user. Selanjutnya fungsi `tambah_user($id_user)` digunakan untuk menampilkan halaman user. Pada fungsi `tambah_user()` *controller* memanggil *query* `getIdUserMax()` untuk menampilkan id user secara otomatis sehingga tidak menyebabkan id user berulang.

Fungsi `proses_tambah_user()` digunakan untuk memproses data user untuk disimpan ke dalam database. *Statment* `insert_user($data)` digunakan untuk menyimpan data user yang telah diambil dari form user. Fungsi selanjutnya pada fitur manajemen data user adalah `edit_user($Id_user)` dan `proses_edit_user($Id_user)`. *Statment* `getUserById($Id_user)` digunakan untuk menampilkan data user yang dipilih ke halaman edit user. Dan `update_user($Id_user, $data2)` digunakan untuk menyimpan data user hasil editan.

#### 4.4 Pengujian Sistem

Tahap *testing* adalah tahap yang harus dilakukan sebelum sistem diserahkan kepada *user*. Tahap dilakukan agar *programer* dapat mengetahui apakah sistem yang dibangun sesuai dengan kebutuhan yang telah dianalisis diawal. Serta agar mengetahui apakah terdapat kesalahan pada sistem yang dibangun. Selain itu tahap *testing* juga digunakan untuk mengetahui kompleksitas sistem yang telah dibangun. Tahap testing mempunyai dua langkah yaitu *white box* dan *black box*.

##### 4.4.1 Pengujian *white-box*

Tahap pengujian *white-box* adalah tahap untuk menghitung kompleksitas sistem yang telah dibangun. Pengujian *white-box* mengecek sistem berdasarkan alur perancangan yang telah dibuat sebelumnya. Pada penelitian ini metode yang digunakan dalam pengujian *white box* adalah *independent path* dengan menggunakan *cyclomatic complexity*.

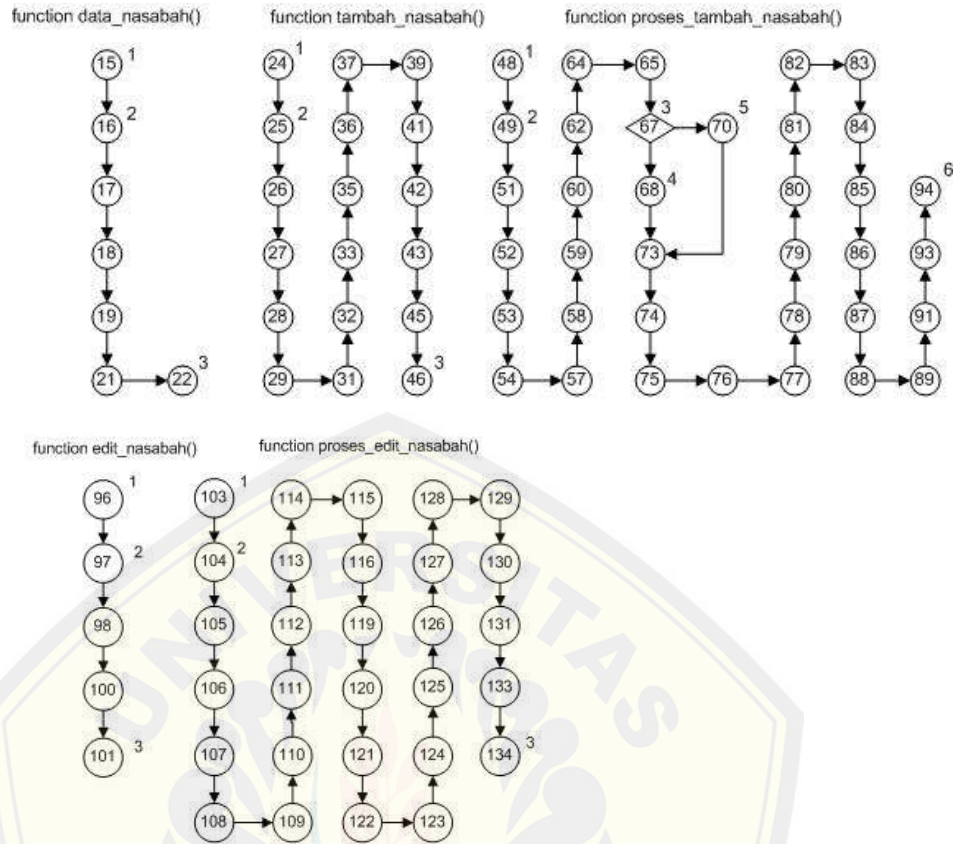
##### 4.4.1.1 Pengujian *White-Box* Memanajemen Data Nasabah

###### a. Baris Program

Tiap *function* pada fitur manajemen data nasabah memiliki berbagai macam statement yang terdapat pada setiap baris program. Baris program yang digunakan untuk *white-box testing* pada fitur manajemen data nasabah dapat dilihat pada lampiran G.1.

###### b. Diagram Alir Program

Diagram alir manajemen data nasabah sesuai program yang telah ditulis dapat dilihat pada gambar 4.10 berikut.



Gambar 4.10 Diagram alir fitur manajemen data nasabah

c. Penghitungan *Cyclomatic Complexity*

Penghitungan diagram alir pada fitur manajemen data nasabah menggunakan kompleksitas siklotatik dapat dilihat sebagai berikut.

- function data\_nasabah() :  $V(G) = E - N + 2 = 6 - 7 + 2 = 1$
- function tambah\_nasabah() :  $V(G) = E - N + 2 = 17 - 18 + 2 = 1$
- function proses\_tambah\_nasabah() :  $V(G) = E - N + 2 = 36 - 36 + 2 = 2$
- function edit\_nasabah() :  $V(G) = E - N + 2 = 4 - 5 + 2 = 1$
- function proses\_edit\_nasabah() :  $V(G) = E - N + 2 = 28 - 29 + 2 = 1$

d. Penghitungan Jalur Program

Berdasarkan diagram alir pada fitur manajemen data nasabah jalur program dapat ditentukan hasil sebagai berikut.

function data_nasabah()	: Jalur 1 – 2 – 3
function tambah_nasabah()	: Jalur 1 – 2 – 3
function proses_tambah_nasabah()	: Jalur 1 – 2 – 3 – 4 – 6
	: Jalur 1 – 2 – 3 – 5 – 6
function edit_nasabah()	: Jalur 1 – 2 – 3
function proses_edit_nasabah()	: Jalur 1 – 2 – 3

e. *Test case*

*Test case* adalah tes yang dilakukan untuk membuktikan kebenaran dari eksekusi program. *Test case* pada fitur manajemen data nasabah dapat ditunjukkan pada tabel 4.6.

Tabel 4. 6 *Test case* manajemen data nasabah

<b><i>Test case function data_nasabah()</i></b>	
<i>Test case</i>	Teller melihat data nasabah
Target yang diharapkan	Sistem dapat menampilkan data nasabah
Hasil pengujian	Benar
Jalur	1 – 2 – 3
<b><i>Test case function tambah_nasabah()</i></b>	
<i>Test case</i>	Teller masuk form tambah nasabah
Target yang diharapkan	Sistem dapat menampilkan form nasabah
Hasil pengujian	Benar
Jalur	1 – 2 – 3
<b><i>Test case function proses_tambah_nasabah()</i></b>	
<b>Jalur 1</b>	
<i>Test case</i>	Teller menginputkan data nasabah nasabah
Target yang diharapkan	Sistem memutuskan data nasabah layak
Hasil pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 6
<b>Jalur 2</b>	
<i>Test case</i>	Teller menginputkan data nasabah nasabah
Target yang diharapkan	Sistem memutuskan data nasabah tidak layak
Hasil pengujian	Benar
Jalur	1 – 2 – 3 – 5 – 6
<b><i>Test case function edit_nasabah()</i></b>	
<i>Test case</i>	Teller masuk form edit nasabah

Target yang diharapkan	Sistem dapat menampilkan halaman edit nasabah
Hasil pengujian	Benar
Jalur	1 – 2 – 3
<b>Test case function proses_edit_nasabah()</b>	
Test case	Teller menyimpan hasil edit data nasabah
Target yang diharapkan	Sistem menyimpan hasil edit data nasabah
Hasil pengujian	Benar
Jalur	1 – 2 – 3

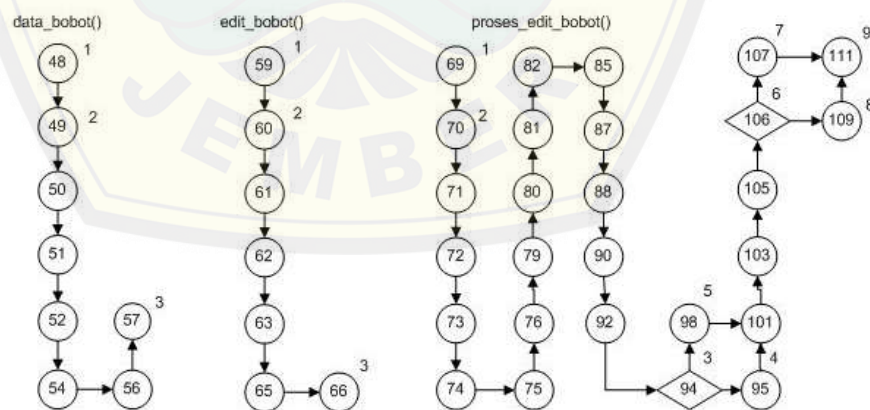
4.4.1.2 Pengujian *White-Box* Memanajemen Data Bobot

a. Baris Program

Tiap *function* pada fitur manajemen data bobot memiliki berbagai macam statement yang terdapat pada setiap baris program. Baris program yang digunakan untuk *white-box testing* pada fitur manajemen data bobot dapat dilihat pada lampiran G.2.

b. Diagram Alir Program

Diagram alir manajemen data bobot sesuai program yang telah ditulis dapat dilihat pada gambar 4.11 berikut.



Gambar 4.11 Diagram alir fitur manajemen data bobot

c. Penghitungan *Cyclomatic Complexity*

Penghitungan diagram alir pada fitur manajemen data bobot menggunakan kompleksitas siklomatik dapat dilihat sebagai berikut.

function data\_bobot() :  $V(G) = E - N + 2 = 7 - 8 + 2 = 1$   
 function edit\_bobot() :  $V(G) = E - N + 2 = 6 - 7 + 2 = 1$   
 function proses\_edit\_bobot() :  $V(G) = E - N + 2 = 28 - 27 + 2 = 3$

d. Penghitungan Jalur Program

Berdasarkan diagram alir pada fitur manajemen data bobot jalur program dapat ditentukan hasil sebagai berikut.

function data\_bobot () : Jalur 1 – 2 – 3  
 function edit\_bobot () : Jalur 1 – 2 – 3  
 function proses\_edit\_bobot () : Jalur 1 – 2 – 3 – 4 – 6 – 7 – 9  
 : Jalur 1 – 2 – 3 – 4 – 6 – 8 – 9  
 : Jalur 1 – 2 – 3 – 5 – 6 – 7 – 9  
 : Jalur 1 – 2 – 3 – 5 – 6 – 8 – 9

e. Test case

Test case adalah tes yang dilakukan untuk membuktikan kebenaran dari eksekusi program. Test case pada fitur manajemen data bobot dapat ditunjukkan pada tabel 4.7.

Tabel 4. 7 Test case manajemen data bobot

<b>Test case function data_bobot()</b>	
Test case	Admin melihat data bobot
Target yang diharapkan	Sistem dapat menampilkan data bobot
Hasil pengujian	Benar
Jalur	1 – 2 – 3
<b>Test case function edit_bobot()</b>	
Test case	Admin akan mengedit bobot
Target yang diharapkan	Sistem dapat menampilkan halaman edit bobot
Hasil pengujian	Benar
Jalur	1 – 2 – 3
<b>Test case function proses_edit_bobot()</b>	
<b>Jalur 1</b>	
Test case	Admin menyimpan data bobot = 1 dan jenis kriteria = “biaya”

dilanjutkan

lanjutan

Target yang diharapkan	Sistem menyimpan data bobot
Hasil pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 6 – 7 – 9
<b>Jalur 2</b>	
<i>Test case</i>	Admin menyimpan data bobot = 1 dan jenis kriteria bukan “biaya”
Target yang diharapkan	Sistem menyimpan data bobot
Hasil pengujian	Benar
Jalur	1 – 2 – 3 – 4 – 6 – 8 – 9
<b>Jalur 3</b>	
<i>Test case</i>	Admin menyimpan data bobot tidak = 1 dan jenis kriteria = “biaya”
Target yang diharapkan	Sistem menampilkan alert bobot tidak = 1
Hasil pengujian	Benar
Jalur	1 – 2 – 3 – 5 – 6 – 7 – 9
<b>Jalur 4</b>	
<i>Test case</i>	Admin menyimpan data bobot tidak = 1 dan jenis kriteria bukan “biaya”
Target yang diharapkan	Sistem menampilkan alert bobot tidak = 1
Hasil pengujian	Benar
Jalur	1 – 2 – 3 – 5 – 6 – 8 – 9

### 5.1.1 Pengujian *Black-Box*

Pengujian *black-box* digunakan untuk menguji fungsionalitas sistem yang telah dibangun. Pengujian dilakukan oleh pihak koperasi pada setiap fitur pada sistem pendukung keputusan pemberian pinjaman. Hasil dari pengujian ini juga dapat digunakan untuk melihat apakah sistem sudah sesuai dengan cara kerja koperasi. Pengujian *black-box* salah satu fitur dapat dilihat pada tabel 4.8 berikut.

Tabel 4. 8 Pengujian *black-box* fitur melihat pengumuman

<b>Pengujian Melihat Pengumuman</b>		
<b>Nasabah</b>	<b>Sistem</b>	<b>Berhasil/Tidak</b>
1. Membuka aplikasi		Berhasil
	2. Menampilkan halaman utama dengan menu Pengumuman, Info	Berhasil

dilanjutkan



lanjutan

	Pinjaman, dan Login	
3. Klik menu “pengumuman”		Berhasil
	4. Menampilkan halaman pengumuman	Berhasil
5. Menginputkan nama dan id nasabah		Berhasil
6. Klik tombol “submit”		Berhasil
	7. Memverifikasi nama dan id nasabah	Berhasil
	8. Nama dan id valid	Berhasil
	9. Menampilkan data keputusan pinjaman nasabah sesuai nama dan id nasabah	Berhasil
<b>Aliran Alternatif</b>		
1. Menginputkan nama dan id nasabah		Berhasil
2. Klik tombol “submit”		Berhasil
	3. Memverifikasi nama dan id	Berhasil
	4. Nama dan id salah	Berhasil
	5. Menampilkan alert	Berhasil

## BAB 5. HASIL DAN PEMBAHASAN

Dalam bab ini dipaparkan hasil dan pembahasan sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *weighted product* yang meliputi implementasi sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dan hasil implementasi metode *weighted product* pada sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri.

### 5.1 Implementasi Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Syariah BMU UGT Sidogiri

Penelitian ini menghasilkan sebuah sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri beserta fitur-fiturnya. Pada sistem terdapat tiga hak akses yaitu nasabah yang dapat mengakses fitur melihat pengumuman dan fitur melihat pinjaman, teller yang dapat mengakses fitur manajemen data nasabah, manajemen data pinjaman, dan manajemen modal koperasi, serta admin yang dapat mengakses fitur memberikan keputusan, manajemen data bobot, dan manajemen data user. Penjelasan mengenai fitur-fitur yang terdapat dalam sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri adalah sebagai berikut :

#### 5.1.1 Fitur Melihat Pengumuman

Pada atas halaman ini terdapat menu pengumuman, info pinjaman, dan login. Pada tengah halaman terdapat kolom nama dan kolom id yang digunakan oleh nasabah menginputkan namanya dan tombol submit yang digunakan untuk menampilkan data persetujuan berdasarkan nama dan id yang diinputkan oleh nasabah. Gambar 5.1 dan gambar 5.2 berikut merupakan fitur melihat pengumuman.



Gambar 5.1 Fitur melihat pengumuman



Gambar 5.2 Pengumuman pinjaman nasabah

#### 5.1.2 Fitur melihat Info Pinjaman

Pada atas halaman ini terdapat menu pengumuman, info pinjaman, dan login. Pada tengah halaman akan berisi jenis-jenis pinjaman yang disediakan oleh koperasi dan tombol selengkapnya. Apabila tombol selengkapnya diklik maka akan menuju ke halaman detail dari jenis pinjaman yang diklik. Gambar fitur melihat info pinjaman dapat dilihat di lampiran F.1 dan F.2.

#### 5.1.3 Fitur Melakukan *Login*

Pada atas halaman ini terdapat menu pengumuman, info pinjaman, dan login. Pada tengah halaman terdapat kolom *username* dan *password* serta

tombol login. Apabila data benar maka akan masuk ke halaman teller seperti pada lampiran F.5 atau halaman admin seperti pada lampiran F.4. Jika data salah maka sistem akan menampilkan alert data salah seperti pada lampiran F.6. Gambar fitur melakukan *login* dapat dilihat di lampiran F.3.

#### 5.1.4 Fitur Melakukan *Logout*

Fitur melakukan *logout* adalah halaman yang digunakan user agar dapat keluar dari halaman manajemen. Ketika user klik tombol logout maka sistem akan mengantarkan user menuju halaman index. Model prototipe melakukan *logout* dapat dilihat pada lampiran F.7.

#### 5.1.5 Fitur Memanajemen Data Nasabah

Fitur memanajemen data nasabah adalah halaman yang digunakan teller untuk melihat, menambah, dan mengedit data nasabah. Fitur melihat data nasabah dapat dilihat pada lampiran F.8. Pada atas halaman ini terdapat menu data nasabah, data pinjaman, dan modal. Pada tengah halaman ini terdapat data nasabah yang telah diinputkan dan tombol tambah serta update. Tombol tambah menuju ke halaman tambah nasabah yang dapat dilihat pada lampiran F.9. Tombol update menuju ke halaman edit nasabah yang dapat dilihat pada lampiran F.10.

#### 5.1.6 Fitur Memanajemen Data Pinjaman

Fitur memanajemen data pinjaman adalah halaman yang digunakan teller untuk melihat, menambah, dan mengedit data pinjaman. Fitur melihat data pinjaman dapat dilihat pada lampiran F.11. Pada atas halaman ini terdapat menu data nasabah, data pinjaman, dan modal. Pada tengah halaman ini terdapat data pinjaman yang telah diinputkan dan tombol tambah serta update. Tombol tambah menuju ke halaman tambah pinjaman yang dapat dilihat pada lampiran F.12. Tombol update menuju ke halaman edit pinjaman yang dapat dilihat pada lampiran F.13.

#### 5.1.7 Fitur Memanajemen Modal Koperasi

Fitur manajemen modal koperasi adalah halaman yang digunakan teller untuk melihat dan mengedit modal koperasi. Fitur melihat data user dapat dilihat pada lampiran F.14. Pada atas halaman ini terdapat menu data nasabah, data pinjaman, dan modal. Pada tengah halaman ini terdapat data modal koperasi yang telah diinputkan dan tombol update. Tombol update menuju ke halaman edit modal koperasi yang dapat dilihat pada lampiran F.15.

#### 5.1.8 Fitur Memberikan Persetujuan

Fitur memberikan persetujuan adalah halaman yang digunakan admin untuk melihat dan memberikan persetujuan. Pada atas halaman ini terdapat menu persetujuan, data bobot, dan data user. Pada tengah halaman ini terdapat data nasabah yang telah diinputkan dan tombol setuju serta tidak setuju. Fitur memberikan persetujuan dapat dilihat pada lampiran F.16.

#### 5.1.9 Fitur Memanajemen Data Bobot

Fitur manajemen data bobot adalah halaman yang digunakan admin untuk melihat dan mengedit data bobot kriteria. Fitur melihat data bobot dapat dilihat pada lampiran F.17. Pada atas halaman ini terdapat menu persetujuan, data bobot, dan data user. Pada tengah halaman ini terdapat data bobot yang telah diinputkan dan tombol update. Tombol update menuju ke halaman edit data bobot yang dapat dilihat pada lampiran F.18.

#### 5.1.10 Fitur Memanajemen Data User

Fitur manajemen data user adalah halaman yang digunakan admin untuk melihat, menambah, dan mengedit data user. Fitur melihat data user dapat dilihat pada lampiran F.19. Pada atas halaman ini terdapat menu persetujuan, data bobot, dan data user. Pada tengah halaman ini terdapat data user yang telah diinputkan dan tombol tambah serta update. Tombol tambah menuju ke halaman tambah user yang dapat dilihat pada lampiran F.20.

Tombol update menuju ke halaman edit user yang dapat dilihat pada lampiran F.21.

## 5.2 Hasil Implementasi Metode *Weighted Product* Pada Sistem Pendukung Keputusan Pemberian Pinjaman Pada Koperasi Simpan Pinjam Syariah Bmt Ugt Sidogiri

Sistem pendukung keputusan sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *weighted product* dalam menghitung nilai dari setiap nasabah. Penggunaan metode *weighted product* terdapat pada fitur manajemen data bobot dan menambahkan data nasabah.

### 5.2.1 Metode *Weighted Product*

#### a. Perhitungan Menggunakan *Listing Program*

Fitur utama pada sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri yang terdapat penerapan metode *weighted product* adalah fitur manajemen data bobot dan fitur menambahkan data nasabah. Pada fitur manajemen data bobot dengan metode *weighted product* diberikan rumus sebagai berikut:

$$W_j = \frac{w}{\sum W}$$

Saat admin memasukkan nilai bobot dari kriteria karakter senilai 5, kriteria jumlah tanggungan senilai 2, kriteria penghasilan perbulan senilai 4, dan kriteria kelengkapan berkas senilai 3 maka sistem akan menghitung pangkat akhir bobot kriteria yang nantinya akan digunakan dalam menghitung nilai nasabah. Pangkat akhir dihitung dengan membagi nilai dari masing-masing bobot kriteria dengan jumlah bobot keseluruhan. Sesuai dengan langkah kerja metode *weighted product* kriteria yang memiliki jenis 'biaya' akan dikalikan minus satu (-1). *Listing program* manajemen data bobot dapat dilihat pada lampiran E.8 dan hasil hitungnyanya pada lampiran F.17 dan F.18.

Fitur selanjutnya yang menggunakan metode *weighted product* adalah fitur manajemen data nasabah. Pada fitur manajemen data nasabah dengan metode *weighted product* diberikan rumus sebagai berikut:

$$S_i = \prod_{j=1}^n X_{ij}^{W_j}$$

Saat teller memasukkan data nasabah beserta nilai kriteria karakter sangat baik (5), kriteria jumlah tanggungan 1 (1), kriteria penghasilan perbulan Rp 2000000-Rp 300000 (4), dan kriteria kelengkapan berkas sangat lengkap (5) dengan nilai bobot kriteria karakter senilai 5, kriteria jumlah tanggungan senilai 2, kriteria penghasilan perbulan senilai 4, dan kriteria kelengkapan berkas senilai 3 maka sistem akan menghitungnya dengan bobot kriteria yang telah diinputkan oleh admin sebelumnya. Nilai kriteria nasabah dipangkatkan dengan bobot masing-masing kriteria kemudian mengalikan hasil pangkat sebelumnya. Dari nilai yang sudah dihitung menggunakan metode *weighted product* dilihat apakah lebih dari 2,0. Jika nilai nasabah kurang dari 2,0 maka nasabah tersebut dinyatakan tidak layak mendapatkan pinjaman, dan jika lebih atau sama dengan 2,0 dinyatakan layak mendapat pinjaman. *Listing program* manajemen data nasabah dapat dilihat pada lampiran E.4 dan hasil hitungannya pada lampiran F.8 dan F.9.

b. Perhitungan secara manual

Perhitungan manual pada manajemen data bobot dimulai dengan menentukan nilai masing-masing kriteria salah satunya pada pinjaman Gadai Emas Syariah (GES) yaitu kriteria karakter senilai 5, kriteria jumlah tanggungan senilai 2, kriteria penghasilan perbulan senilai 4, dan kriteria kelengkapan berkas senilai 3. Dengan menggunakan metode *weighted product* maka nilai pangkat akhir bobot didapat dilihat pada tabel 5.1 berikut.

Tabel 5. 1 Perhitungan manual menentukan pangkat bobot

No.	Kriteria	Nilai	Rumus	Pangkat
1.	Karakter	5	$\frac{5}{14}$	0,35714285714286
2.	Jumlah tanggungan	2	$\frac{2}{14} \times (-1)$	-0,14285714285714
3.	Penghasilan perbulan	4	$\frac{4}{14}$	0,28571428571429
4.	Kelengkapan berkas	3	$\frac{3}{14}$	0,21428571428571

Perhitungan manual pada manajemen data nasabah dimulai dengan menentukan nilai masing-masing kriteria semisal pada nasabah dengan id KSP09 yaitu kriteria karakter sangat baik (5), kriteria jumlah tanggungan 1 (1), kriteria penghasilan perbulan Rp 2000000-Rp 300000 (4), dan kriteria kelengkapan berkas lengkap (5) kemudian dipangkatkan dengan nilai bobot kriteria pada jenis pinjaman semisal pada pinjaman Gadai Emas Syariah (GES) yaitu kriteria karakter senilai 0,35714285714286, kriteria jumlah tanggungan senilai -0,14285714285714, kriteria penghasilan perbulan senilai 0,28571428571429, dan kriteria kelengkapan berkas senilai 0,21428571428571. Dengan menggunakan metode *weighted product* maka nilai nasabah didapat dilihat sebagai berikut.

$$\begin{aligned} \text{Nilai} &= (5^{0,35714285714286}) \times (1^{-0,14285714285714}) \times (4^{0,28571428571429}) \times \\ &\quad (5^{0,21428571428571}) \\ &= 3,727593720315 \end{aligned}$$

Nilai yang didapat oleh nasabah dengan id KSP09 adalah 3,727593720315. Nilai tersebut lebih dari nilai minimum yang ditetapkan yaitu 0,2 sehingga nasabah tersebut dinyatakan layak menerima pinjaman.

c. Perbandingan perhitungan

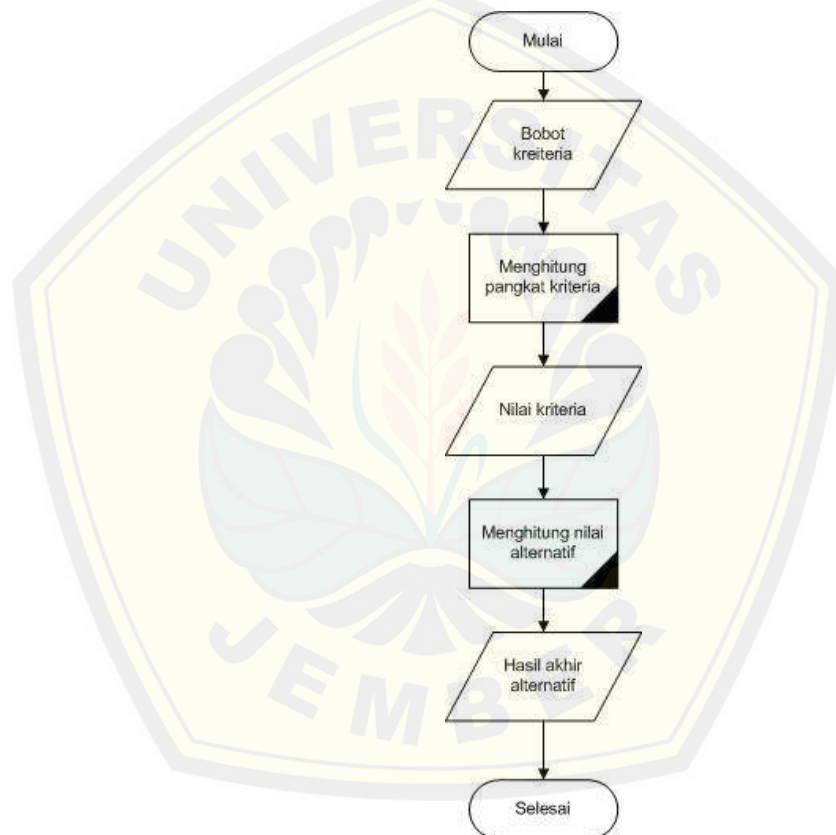
Hasil perhitungan pada *listing program* dan perhitungan manual menunjukkan hasil yang sama. Hasil nilai pangkat bobot pada pinjaman Gadai Emas Syariah (GES) adalah 0,35714285714286, -0,14285714285714,



0,28571428571429, dan 0,21428571428571. Sedangkan hasil nilai nasabah dengan id KSP09 adalah 3,727593720315 yang artinya layak mendapatkan pinjaman.

### 5.2.2 Alur Penerapan Metode *Weighted Product*

Alur penerapan metode *weighted product* pada sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dapat dilihat pada gambar 5.3 berikut.



Gambar 5.3 Alur penerapan metode *weighted product*

Sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri merupakan sistem yang dapat membantu koperasi dalam menentukan kelayakan calon peminjam dengan mengolah data nasabah yang mengajukan pinjaman, data pinjaman, data bobot kriteria pinjaman, dan data modal koperasi kemudian menampilkannya

menjadi data nasabah yang memiliki nilai sehingga dapat dilihat nasabah mana yang memenuhi kriteria. Alur penerapan metode *weighted product* pada sistem ini adalah sebagai berikut:

- 1) Admin menuju ke halaman manajemen data bobot
- 2) Admin klik tombol “update”
- 3) Admin mengedit data bobot kriteria
- 4) Sistem akan menghitung pangkat akhir dari bobot yang diinputkan.
- 5) Teller menuju ke halaman manajemen data nasabah
- 6) Teller klik tombol “tambah”
- 7) Teller menginputkan data nasabah beserta nilai setiap kriteria
- 8) Sistem menghitung nilai nasabah
- 9) Sistem menampilkan nilai akhir nasabah dan hasil kelayakan.

### **5.3 Pembahasan Hasil Implementasi Sistem Pendukung Keputusan Pemberian Pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dan Penerapan Metode *Weighted Product***

Sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri merupakan sistem yang dapat mengolah data nasabah yang mengajukan pinjaman, data pinjaman, data bobot kriteria pinjaman, dan data modal koperasi kemudian menampilkannya menjadi data nasabah yang memiliki nilai sehingga dapat dilihat nasabah mana yang memenuhi kriteria. Sistem pendukung keputusan ini menggunakan metode *weighted product* dalam menghitung nilai dari setiap nasabah. Koperasi yang memiliki sepuluh fitur ini juga menentukan batasan minimum nilai sehingga sistem dapat menampilkan keputusan layak atau tidak layak dari setiap nasabah.

Fitur-fitur pada sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *weighted product* diantaranya fitur melihat pengumuman, fitur melihat info pinjaman, fitur *login*, fitur *logout*, fitur manajemen data nasabah, fitur manajemen data pinjaman, fitur manajemen modal

koperasi, fitur memberikan persetujuan, fitur manajemen data bobot, fitur manajemen data user. Sepuluh fitur tersebut digunakan oleh tiga jenis aktor yaitu nasabah, teller, dan admin. Aktor nasabah hanya dapat mengakses fitur melihat pengumuman dan fitur melihat info pinjaman. Aktor teller dapat mengakses fitur manajemen data nasabah, fitur manajemen data pinjaman, dan fitur manajemen modal koperasi. Dan aktor admin dapat mengakses fitur memberikan persetujuan, fitur manajemen data bobot, dan fitur manajemen data user.

Fitur utama pada sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri yang terdapat penerapan metode *weighted product* adalah fitur manajemen data bobot dan fitur menambahkan data nasabah. Saat admin memasukkan nilai bobot dari masing-masing kriteria sistem akan menghitung pangkat akhir bobot kriteria yang nantinya akan digunakan dalam menghitung nilai nasabah. Pangkat akhir dihitung dengan membagi nilai dari masing-masing bobot kriteria dengan jumlah bobot keseluruhan. Sesuai dengan langkah kerja metode *weighted product* kriteria yang memiliki jenis 'biaya' akan dikalikan minus satu (-1).

Saat teller memasukkan data nasabah beserta nilai kriteria nasabah tersebut sistem akan menghitungnya dengan bobot kriteria yang telah diinputkan oleh admin sebelumnya. Nilai kriteria nasabah dipangkatkan dengan bobot masing-masing kriteria kemudian mengalikan hasil pangkat sebelumnya. Dari nilai yang sudah dihitung menggunakan metode *weighted product* dilihat apakah lebih dari 2,0. Jika nilai nasabah kurang dari 2,0 maka nasabah tersebut dinyatakan tidak layak mendapatkan pinjaman, dan jika lebih atau sama dengan 2,0 dinyatakan layak mendapat pinjaman.

Fitur tambahan pada sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri adalah fitur melihat pengumuman, fitur melihat info pinjaman, fitur *login*, fitur *logout*, fitur manajemen data pinjaman, fitur manajemen modal koperasi, fitur memberikan persetujuan, dan fitur manajemen data user.

Fitur melihat pengumuman digunakan oleh nasabah untuk melihat hasil pengajuan pinjaman oleh nasabah tersebut apakah disetujui oleh koperasi atau tidak. Untuk melihat pengumuman nasabah harus menginputkan nama dan id nasabah kemudian klik tombol “submit”.

Fitur melihat info pinjaman adalah fitur yang digunakan oleh nasabah untuk melihat pinjaman yang disediakan oleh koperasi. Selain jenis-jenis pinjaman yang ditampilkan pada fitur ini juga menampilkan info pinjaman selengkapnya pada masing-masing jenis pinjaman. Untuk melihat info selengkapnya nasabah harus klik tombol “selengkapnya” pada masing-masing jenis pinjaman.

Fitur *login* dan fitur *logout* digunakan oleh aktor user untuk masuk dan keluar dari halaman yang dikhususkan untuk user. Untuk melakukan *login* user harus mempunyai *username* dan *password* yang sesuai dengan data yang dimiliki oleh sistem, sedangkan untuk *logout user* hanya diharuskan klik menu “logout” pada halaman yang sebelumnya melakukan *login*.

Fitur manajemen data pinjaman digunakan oleh teller untuk menginputkan dan mengupdate data pinjaman yang disediakan oleh koperasi. Data pinjaman yang diinputkan oleh teller akan ditampilkan oleh sistem pada fitur halaman data pinjaman yang dilihat atau diakses oleh nasabah untuk mengetahui pinjaman yang disediakan oleh koperasi.

Fitur manajemen modal koperasi adalah fitur yang digunakan oleh teller untuk menginputkan modal yang dimiliki oleh koperasi. Fitur ini digunakan untuk mengetahui pinjaman maksimal yang bisa dilakukan oleh masing-masing nasabah. Dengan menginputkan data modal dan jumlah nasabah maka sistem akan menghitung dengan otomatis nilai pinjaman maksimal tersebut.

Fitur memberikan persetujuan digunakan oleh admin untuk memberikan keputusan akhir dari koperasi setelah dihitung nilai kelayakan masing-masing nasabah. Pada fitur ini terdapat tombol setuju dan tidak setuju yang dapat menentukan keputusan akhir tersebut. Ketika klik tombol setuju

maka pinjaman yang diajukan nasabah disetujui oleh koperasi, dan sebaliknya.

Fitur manajemen data user digunakan oleh admin untuk manajemen user yang dapat mengakses sistem. Ketika admin klik tambah pada formulir tambah user sistem akan melihat apakah *password* dan *repassword* yang diinputkan sama atau tidak. Jika sama maka data akan tersimpan tetapi jika tidak sistem akan menampilkan alert data salah.

#### 5.4 Pembahasan Pengujian Sistem

Sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri telah diuji dengan pengujian *white-box testing* dan *black-box testing*. Pengujian *white-box testing* dilakukan oleh programmer untuk menghitung kompleksitas sistem yang telah dibangun menggunakan metode *cyclomatic complexity*. Berdasarkan metode tersebut programmer mengecek sistem berdasarkan alur perancangan yang telah dibuat sebelumnya. Sedangkan pengujian *black-box testing* dilakukan oleh pihak koperasi guna mengetahui kesesuaian sistem dengan rancangan yang telah dibuat sebelumnya.

Berdasarkan pengujian *white-box testing* dengan metode *cyclomatic complexity* didapat nilai kompleksitas sistem antara 1 sampai 3. Apabila dibandingkan dengan tabel standar nilai *cyclomatic complexity* sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *weighted product* termasuk dalam range 1 sampai 4, yang artinya sistem ini cukup sederhana dengan fitur terbatas, namun memiliki resiko error dan perbaikan rendah. Secara umum hasil pengukuran tingkat kesulitan dan kompleksitas kerja, sistem ini tergolong sistem sederhana dan mudah digunakan. Dengan demikian *user* cepat menguasai sistem ini dan tidak kesulitan dalam mengoperasikannya.

Pada pengujian *black-box testing* yang dilakukan oleh pihak Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri didapat kesimpulan bahwa

sistem sudah sesuai dengan rancangan yang telah diinginkan bersama. Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dalam langkah pengujian ini mengharapkan sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri menggunakan metode *weighted product* diintegrasikan dengan sistem manajemen yang sudah ada dikoperasi sehingga kerja staff lebih efisien.



## BAB 6. PENUTUP

### 6.1 Kesimpulan

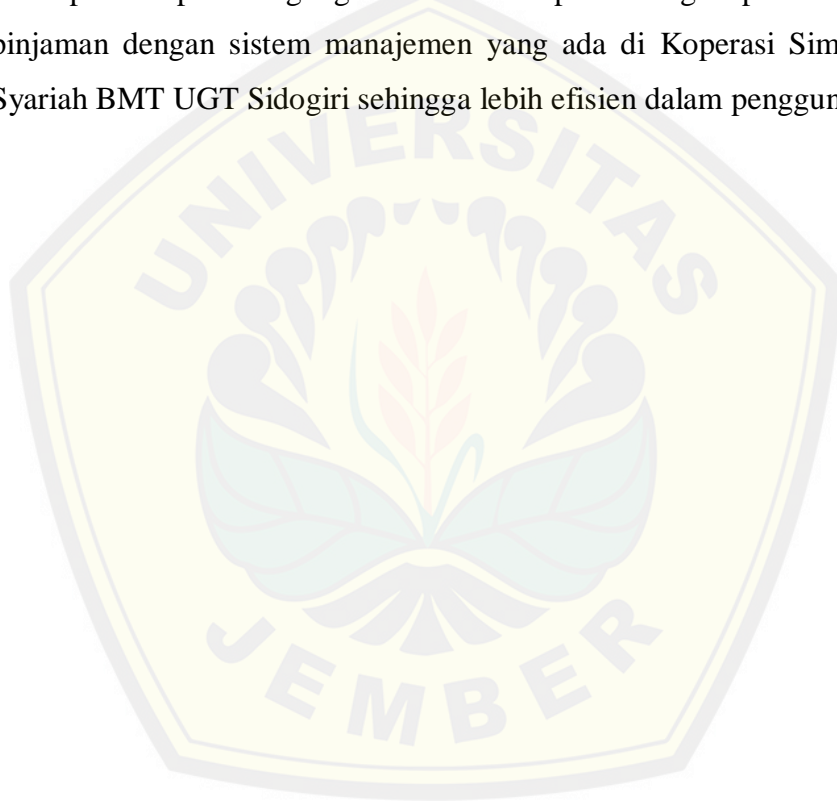
Hasil dari penelitian ini dapat disimpulkan sebagai berikut :

1. Metode *weighted product* pada sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri diterapkan dalam mengambil keputusan layak atau tidaknya seorang nasabah untuk mendapat pinjaman dari koperasi. Pengambilan keputusan dihitung dengan cara memangkatkan nilai kriteria nasabah. Pangkat didapat dari nilai bobot masing-masing kriteria dibagi nilai bobot keseluruhan. Pangkat bernilai positif jika kriteria merupakan kriteria keuntungan dan pangkat bernilai negatif jika kriteria merupakan kriteria biaya.
2. Berdasarkan nilai bobot yang telah diinputkan pada kriteria pinjaman Gadai Emas Syariah (GES) didapat nilai pangkat akhir yaitu kriteria karakter senilai 0,35714285714286, kriteria jumlah tanggungan senilai -0,14285714285714, kriteria penghasilan perbulan senilai 0,28571428571429, dan kriteria kelengkapan berkas senilai 0,21428571428571. Nilai tersebut kemudian digunakan untuk memangkatkan nilai kriteria nasabah dengan id KSP09 sehingga didapat nilai 3,727593720315. Nilai tersebut lebih dari nilai minimum yang ditetapkan yaitu 0,2 sehingga nasabah tersebut dinyatakan layak menerima pinjaman.
3. Sistem pedukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dirancang dan dibangun untuk memberikan bantuan dalam menentukan kelayakan nasabah mendapatkan pinjaman dengan memanfaatkan metode *weighted product*. Penerapan metode *weighted product* dilakukan dengan membangun fitur utama yaitu memanajemen data bobot dan memanajemen data nasabah. Fitur tambahan pada sistem ini ada fitur melihat pengumuman, fitur melihat info pinjaman, fitur *login*, fitur *logout*, fitur memanajemen data

pinjaman, fitur manajemen modal koperasi, fitur memberikan persetujuan, dan fitur manajemen data user.

## 6.2 Saran

*Pertama*, sistem pendukung keputusan pemberian pinjaman pada Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri dapat dikembangkan lebih lanjut dengan menambahkan metode lain secara *hybrid* sehingga perhitungan nilai nasabah dapat lebih akurat. *Kedua*, bagi peneliti selanjutnya diharapkan dapat mengintegrasikan sistem pendukung keputusan pemberian pinjaman dengan sistem manajemen yang ada di Koperasi Simpan Pinjam Syariah BMT UGT Sidogiri sehingga lebih efisien dalam penggunaannya.





## DAFTAR PUSTAKA

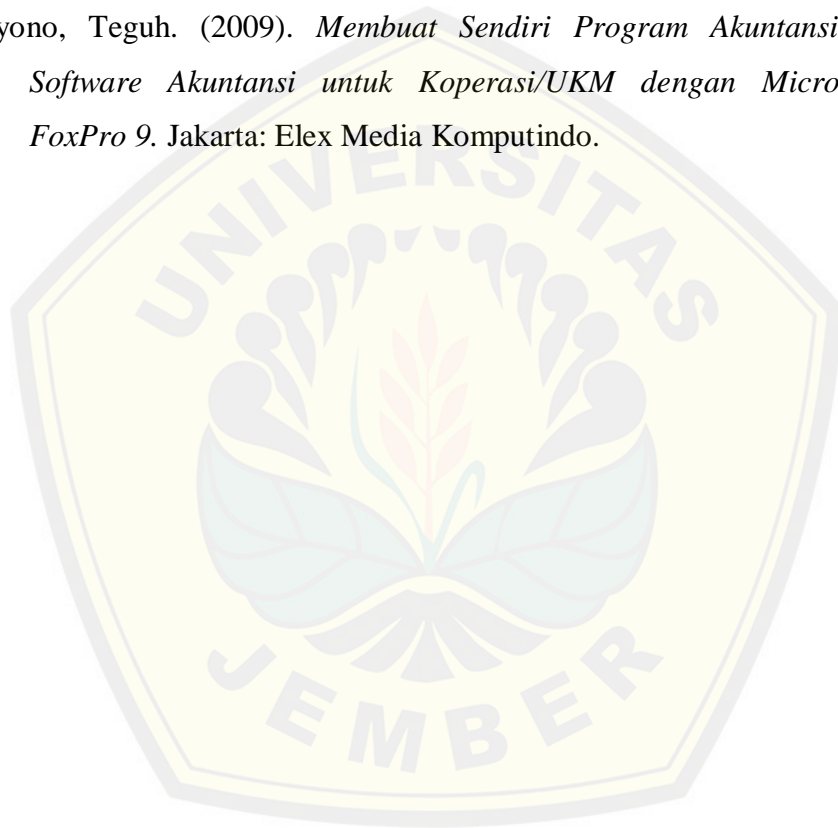
- Ahmadi, Aziz, & Wiyanti, Dian Tri. (2014). Seminar Nasional Aplikasi Teknologi Informasi (SNATI). *Implementasi Weighted Product (WP) Dalam Penentuan Penerima Bantuan Langsung Masyarakat PNPM Mandiri Perdesaan*, 19-21.
- Alter. (2002). *Analisis dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Objek*. Bandung: Informatika.
- Kadrasah, Suryadi, & Ramdhani, Ali. (2002). *Sistem Pendukung Keputusan*. Bandung: Remaja Rosdakarya.
- Kendall, Kenneth E., & E., Kendal Julie. (2006). *Analisis dan Perancangan Sistem Edisi Lima Jilid 1*. Jakarta: Gramedia.
- Moleong, Lexy J. (2011). *Metodologi Penelitian Kualitatif*. Bandung: PT Remaja Rosdakarya.
- Permatasari, Yuke. (2013). *Sistem Pendukung Keputusan Pemberian Bonus Pegawai Pada Hotel Alamanda Klaten Dengan Menggunakan Metode Weighted Product*. Yogyakarta: Sekolah Tinggi Manajemen Informatika dan Komputer Amikom Yogyakarta.
- Pressman, Roger S. (2005). *Rekayasa Perangkat Lunak Buku I*. Yogyakarta: Penerbit Andi.
- Sari, Rizki Yuniar. (2012). *Sistem Pendukung Keputusan Pencairan Kredit Menggunakan Metode Scoring System (Studi Kasus: BMT Bangun Rakyat Sejahtera Yogyakarta)*. Yogyakarta: Tidak Diterbitkan. Skripsi UIN Sunan Kalijaga.
- Shalahuddin, M., & S., Rosa A. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Penerbit Modula.

Sommerville, Ian. (2011). *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.

Sugiyono. (2011). *Metode Penelitian Pendidikan (Pendekatan Kualitatif, Kuantitatif, dan R&D)*. Bandung: Alfabeta.

Turban, & dkk. (2005). *Decision Support System and Intelegent System*. Yogyakarta: Penerbitan Andi.

Wahyono, Teguh. (2009). *Membuat Sendiri Program Akuntansi: Membuat Software Akuntansi untuk Koperasi/UKM dengan Microsoft Visual FoxPro 9*. Jakarta: Elex Media Komputindo.



## LAMPIRAN A USECASE SCENARIO

### A.1 Usecase Scenario Melihat Info Pinjaman

1. Nama Usecase	: Melihat Info Pinjaman
2. Aktor	: Nasabah
3. Pra-kondisi	: Nasabah membuka aplikasi
4. Pasca-kondisi	: Nasabah telah mendapatkan info pinjaman
Aliran Utama	
Nasabah	Sistem
1. Membuka aplikasi	
	2. Menampilkan halaman utama dengan menu Pengumuman, Info Pinjaman, dan Login
3. Klik menu “info pinjaman”	
	4. Menampilkan halaman info pinjaman
	5. Menampilkan daftar pinjaman
6. Klik tombol “selengkapnya”	
	7. Menampilkan detail info pinjaman

### A.2 Usecase Scenario Melakukan Login

1. Nama Usecase	: Melakukan Login
2. Aktor	: User
3. Pra-kondisi	: User membuka aplikasi
4. Pasca-kondisi	: User berhasil masuk ke dalam sistem
Aliran Utama	
User	Sistem
1. Membuka aplikasi	
	2. Menampilkan halaman utama dengan menu Pengumuman, Info

dilanjutkan

lanjutan

	Pinjaman, dan Login
3. Mengklik menu “login”	
	4. Menampilkan halaman login
5. Menginputkan username dan password	
6. Klik tombol “login”	
	7. Memverifikasi username dan password
	8. Username dan password valid
	9. Menampilkan halaman admin
<b>Aliran Alternatif</b>	
5. Menginputkan username dan password	
6. Klik tombol “login”	
	7. Memverifikasi username dan password
	8. Username dan password valid
	9. Menampilkan halaman teller
<b>Aliran Alternatif</b>	
5. Menginputkan username dan password	
6. Klik tombol “login”	
	7. Memverifikasi username dan password
	8. Username dan password salah
	9. Menampilkan alert username/ password salah
	10. Menampilkan halaman beranda

**A.3 Usecase Scenario Melakukan Logout**

1. Nama Usecase	: Melakukan Logout
2. Aktor	: User
3. Pra-kondisi	: User dalam aplikasi
4. Pasca-kondisi	: User berhasil keluar dari sistem
<b>Aliran Utama</b>	
<b>User</b>	<b>Sistem</b>
1. Klik menu “logout”	
	2. Menampilkan halaman beranda awal dengan menu Info Nasabah, Info Pinjaman, dan Login

**A.4 Usecase Scenario Memanajemen Data Nasabah**

1. Nama Usecase	: Memanajemen Data Nasabah
2. Aktor	: Teller
3. Pra-kondisi	: Teller di beranda sistem
4. Pasca-kondisi	: Teller berhasil memanajemen data nasabah
<b>Aliran Utama</b>	
<b>Teller</b>	<b>Sistem</b>
1. Klik menu “data nasabah”	
	2. Menampilkan halaman data serta tombol tambah dan update
3. Klik tombol “tambah”	
	4. Menampilkan halaman form pendaftaran nasabah dengan tombol tambah dan batal
5. Memilih jenis pinjaman	
	6. Menampilkan penentu jumlah pinjaman pada form data nasabah
7. Menginputkan data nasabah	

dilanjutkan

lanjutan

8. Klik tombol “tambah”	
	9. Memverifikasi data nasabah
	10. Data nasabah benar, data tersimpan
	11. Menampilkan halaman data nasabah
<b>Aliran Alternatif</b>	
7. Menginputkan data nasabah	
8. Klik tombol “batal”	
	9. Data tidak tersimpan
	10. Menampilkan halaman data nasabah
<b>Aliran Alternatif</b>	
7. Menginputkan data nasabah	
8. Klik tombol “tambah”	
	9. Memverifikasi data nasabah
	10. Data salah, data tidak tersimpan
	11. Menampilkan alert “data salah” dan menampilkan form pendaftaran nasabah
<b>Aliran Alternatif</b>	
3. Klik tombol “update” pada salah satu baris data nasabah	
	4. Menampilkan halaman edit nasabah dengan tombol update dan batal
5. Mengedit data nasabah	
6. Klik tombol “update”	
	7. Memverifikasi data nasabah
	8. Data nasabah benar, data

dilanjutkan

lanjutan

	tersimpan
	9. Menampilkan halaman data nasabah
<b>Aliran Alternatif</b>	
5. Mengedit data nasabah	
6. Klik tombol “batal”	
	7. Data tidak tersimpan
	8. Menampilkan halaman data nasabah
<b>Aliran Alternatif</b>	
5. Mengedit data nasabah	
6. Klik tombol “update”	
	7. Memverifikasi data nasabah
	8. Data salah, data tidak tersimpan
	9. Menampilkan alert “data salah” dan menampilkan alamat edit nasabah

#### **A.5 Usecase Scenario Memanajemen Data Pinjaman**

1. Nama Usecase	: Memanajemen Data Pinjaman
2. Aktor	: Teller
3. Pra-kondisi	: Teller di beranda sistem
4. Pasca-kondisi	: Teller berhasil memanajemen data pinjaman
<b>Aliran Utama</b>	
<b>Teller</b>	<b>Sistem</b>
1. Klik menu “data pinjaman”	
	2. Menampilkan halaman data pinjaman serta tombol tambah dan update
3. Klik tombol “tambah”	

dilanjutkan

lanjutan

	4. Menampilkan halaman form data pinjaman dengan tombol tambah dan batal
5. Menginputkan data pinjaman	
6. Klik tombol “tambah”	
	7. Memverifikasi data pinjaman
	8. Data pinjaman benar, data tersimpan
	9. Menampilkan halaman data pinjaman
<b>Aliran Alternatif</b>	
5. Menginputkan data pinjaman	
6. Klik tombol “batal”	
	7. Data tidak tersimpan
	8. Menampilkan halaman data pinjaman
<b>Aliran Alternatif</b>	
5. Menginputkan data pinjaman	
6. Klik tombol “tambah”	
	7. Memverifikasi data pinjaman
	8. Data salah, data tidak tersimpan
	9. Menampilkan alert “data salah” dan menampilkan form data pinjaman
<b>Aliran Alternatif</b>	
3. Klik tombol “update” pada salah satu baris data pinjaman	
	4. Menampilkan halaman edit pinjaman dengan tombol update dan batal

dilanjutkan



lanjutan

5. Mengedit data pinjaman	
6. Klik tombol “update”	
	7. Memverifikasi data pinjaman
	8. Data pinjaman benar, data tersimpan
	9. Menampilkan halaman data pinjaman
<b>Aliran Alternatif</b>	
5. Mengedit data pinjaman	
6. Klik tombol “batal”	
	7. Data tidak tersimpan
	8. Menampilkan halaman data pinjaman
<b>Aliran Alternatif</b>	
5. Mengedit data pinjaman	
6. Klik tombol “update”	
	7. Memverifikasi data pinjaman
	8. Data salah, data tidak tersimpan
	9. Menampilkan alert “data salah” dan menampilkan alamat edit pinjaman

#### **A.6 Usecase Scenario Memanajemen Modal Koperasi**

1. Nama Usecase	: Memanajemen Modal Koperasi
2. Aktor	: Teller
3. Pra-kondisi	: Teller di beranda sistem
4. Pasca-kondisi	: Teller berhasil memanajemen modal koperasi
<b>Aliran Utama</b>	
<b>Teller</b>	<b>Sistem</b>
1. Klik menu “modal”	

dilanjutkan

lanjutan

	2. Menampilkan halaman modal koperasi serta tombol update
3. Klik tombol “update”	
	4. Menampilkan halaman update modal
5. Mengganti nilai modal	
6. Klik tombol “update”	
	7. Menampilkan halaman modal koperasi
<b>Aliran Alternatif</b>	
5. Mengganti nilai modal	
6. Klik tombol “batal”	
	7. Data tidak tersimpan
	8. Menampilkan halaman modal koperasi

#### **A.7 Usecase Scenario Memberikan Persetujuan**

1. Nama Usecase	: Memberikan Persetujuan
2. Aktor	: Admin
3. Pra-kondisi	: Admin di beranda sistem
4. Pasca-kondisi	: Admin berhasil memberikan keputusan persetujuan
<b>Aliran Utama</b>	
<b>Admin</b>	<b>Sistem</b>
1. Klik menu “persetujuan”	
	2. Menampilkan halaman persetujuan serta tombol “setujui” dan tombol “tidak disetujui”
3. Mengedit nilai pinjaman	
4. Klik tombol “setujui”	
	5. Menampilkan halaman persetujuan

dilanjutkan

lanjutan

	dengan data nasabah disetujui
<b>Aliran Alternatif</b>	
3. Mengedit nilai pinjaman	
4. Klik tombol “tidak disetujui”	
	5. Menampilkan halaman persetujuan dengan data nasabah tidak disetujui

#### **A.8 Usecase Scenario Memanajemen Data Bobot**

1. Nama Usecase	: Memanajemen Data Bobot
2. Aktor	: Admin
3. Pra-kondisi	: Admin di beranda sistem
4. Pasca-kondisi	: Admin berhasil memanajemen data bobot
<b>Aliran Utama</b>	
<b>Admin</b>	<b>Sistem</b>
1. Klik menu “data bobot”	
	2. Menampilkan halaman data bobot serta select button data pinjaman dan tombol update
3. Memilih data pinjaman yang akan dilihat bobotnya	
	4. Menampilkan data bobot sesuai data pinjaman yang dipilih
5. Klik tombol “update”	
	6. Menampilkan halaman update bobot sesuai data yang yang dipilih dengan tombol update dan batal
7. Mengganti nilai bobot	
8. Klik tombol “update”	
	9. Menghitung nilai bobot

dilanjutkan

lanjutan

	10. Jumlah bobot = 1, nilai tersimpan
	11. Menampilkan halaman data bobot
<b>Aliran Alternatif</b>	
7. Mengganti nilai bobot	
6. Klik tombol “batal”	
	7. Data tidak tersimpan
	8. Menampilkan halaman data bobot
<b>Aliran Alternatif</b>	
7. Mengganti nilai bobot	
8. Klik tombol “update”	
	9. Menghitung nilai bobot
	10. Jumlah bobot tidak sama dengan 1
	11. Menampilkan alert data salah
	12. Menampilkan halaman data bobot

**A.9 Usecase Scenario Memanajemen Data User**

1. Nama Usecase	: Memanajemen Data User
2. Aktor	: Admin
3. Pra-kondisi	: Admin di beranda sistem
4. Pasca-kondisi	: Admin berhasil memanajemen data user
<b>Aliran Utama</b>	
<b>Admin</b>	<b>Sistem</b>
1. Klik menu “data user”	
	2. Menampilkan halaman data user serta tombol tambah dan update
3. Klik tombol “tambah”	
	4. Menampilkan halaman form data user dengan tombol tambah dan batal

dilanjutkan

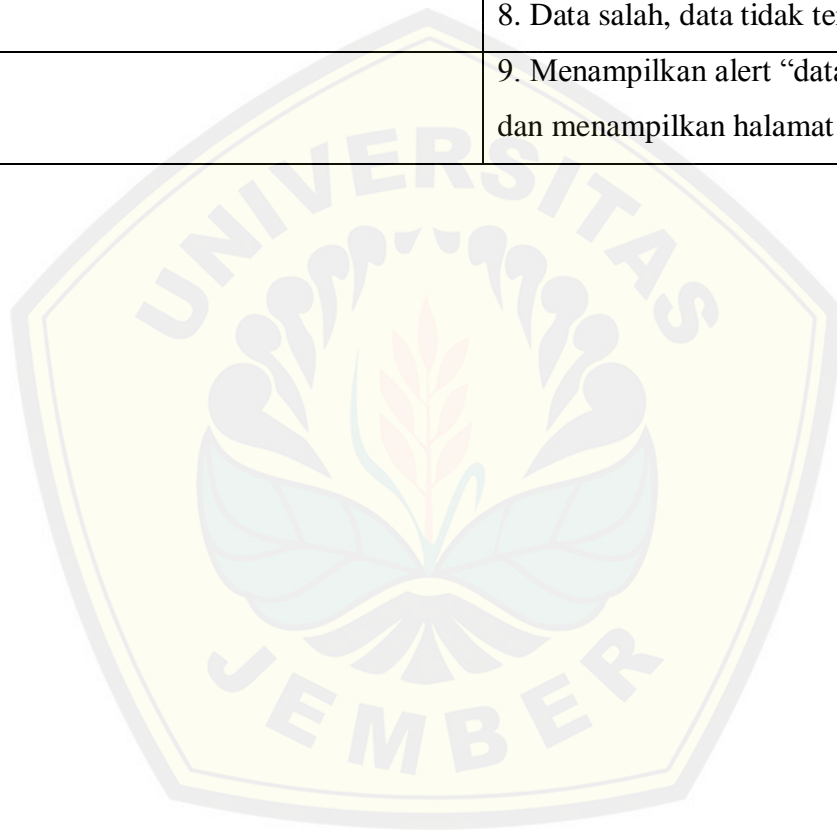
lanjutan

5. Menginputkan data user	
6. Klik tombol “tambah”	
	7. Memverifikasi data user
	8. Data user benar, data tersimpan
	9. Menampilkan halaman data user
<b>Aliran Alternatif</b>	
5. Menginputkan data user	
6. Klik tombol “batal”	
	7. Data tidak tersimpan
	8. Menampilkan halaman data user
<b>Aliran Alternatif</b>	
5. Menginputkan data user	
6. Klik tombol “tambah”	
	7. Memverifikasi data user
	8. Data salah, data tidak tersimpan
	9. Menampilkan alert “data salah” dan menampilkan form data user
<b>Aliran Alternatif</b>	
3. Klik tombol “update” pada salah satu baris data user	
	4. Menampilkan halaman edit user dengan tombol update dan batal
5. Mengedit data user	
6. Klik tombol “update”	
	7. Memverifikasi data user
	8. Data user benar, data tersimpan
	9. Menampilkan halaman data user
<b>Aliran Alternatif</b>	
5. Mengedit data user	

dilanjutkan

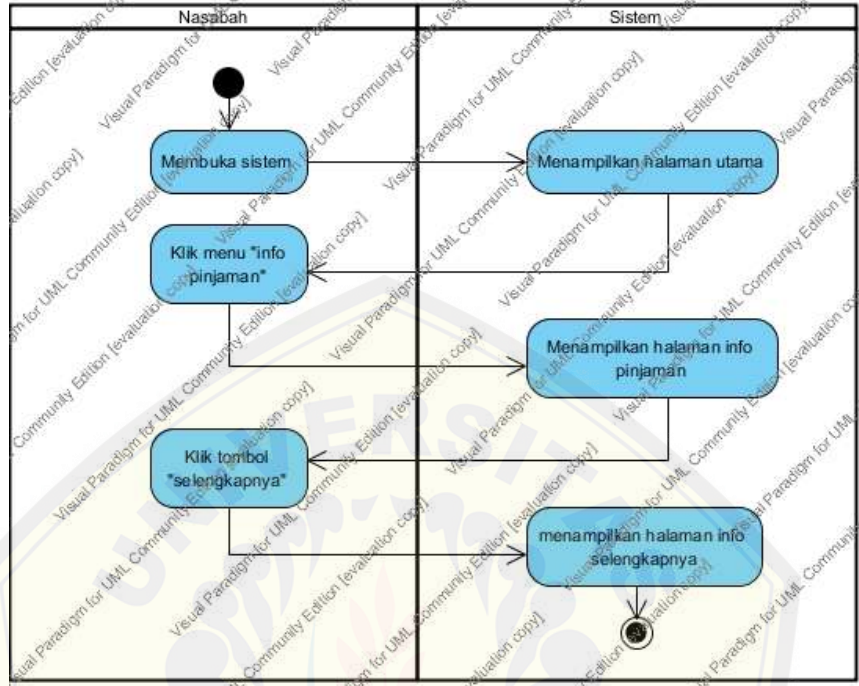
lanjutan

6. Klik tombol “batal”	
	7. Data tidak tersimpan
	8. Menampilkan halaman data user
<b>Aliran Alternatif</b>	
5. Mengedit data user	
6. Klik tombol “update”	
	7. Memverifikasi data user
	8. Data salah, data tidak tersimpan
	9. Menampilkan alert “data salah” dan menampilkan alamat edit user

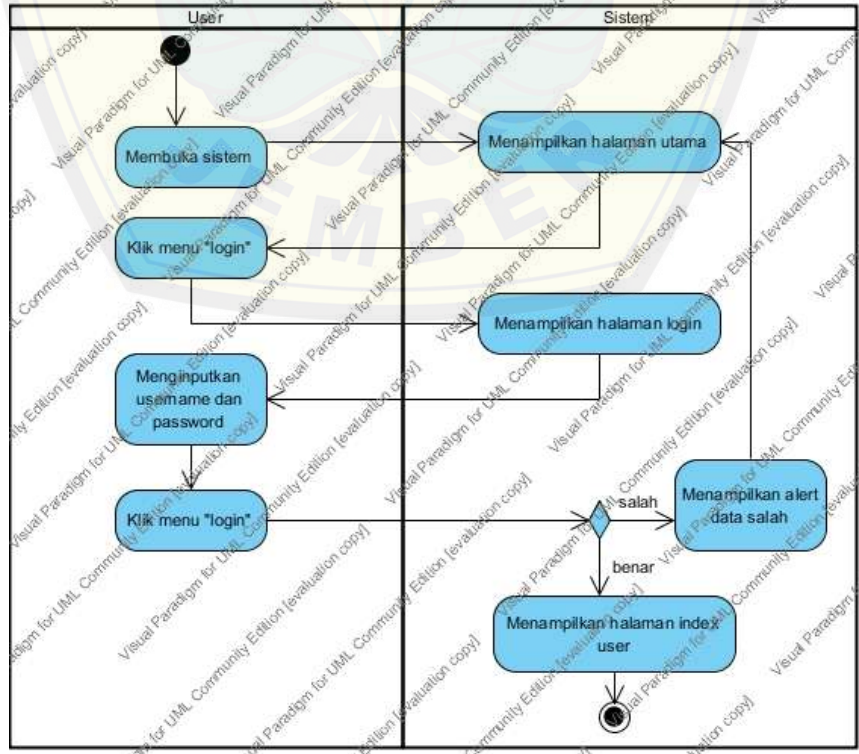


### LAMPIRAN B ACTIVITY DIAGRAM

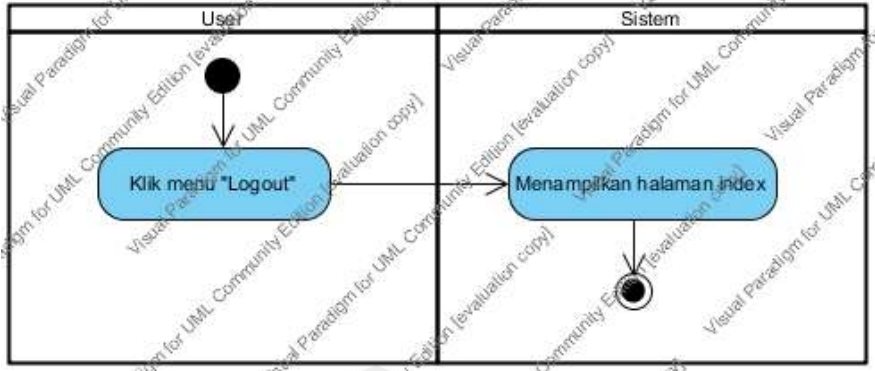
#### B.1 Activity Diagram Melihat Info Pinjaman



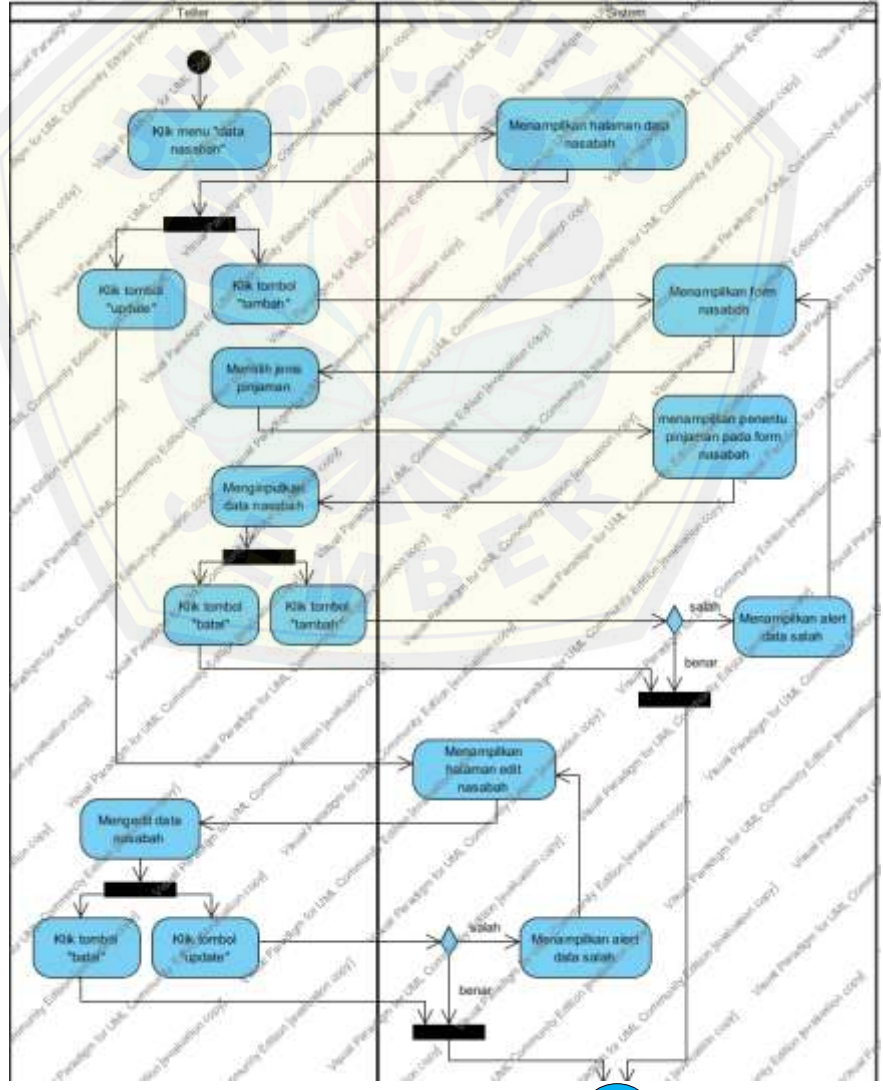
#### B.2 Activity Diagram Melakukan Login



### B.3 Activity Diagram Melakukan Logout

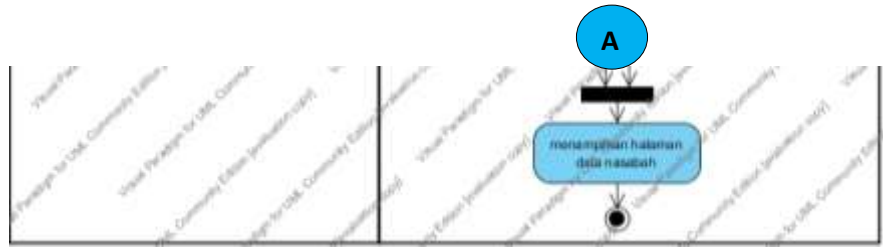


### B.4 Activity Diagram Memanajemen Data Nasabah

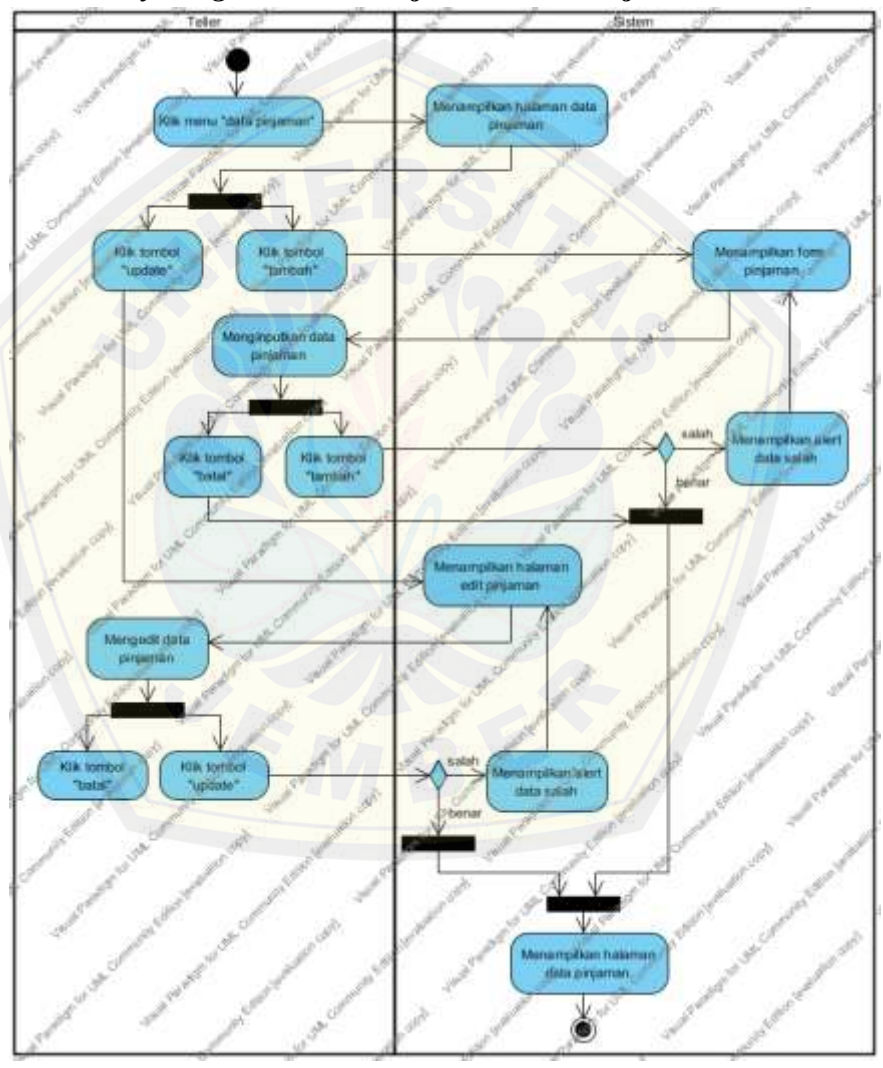


A

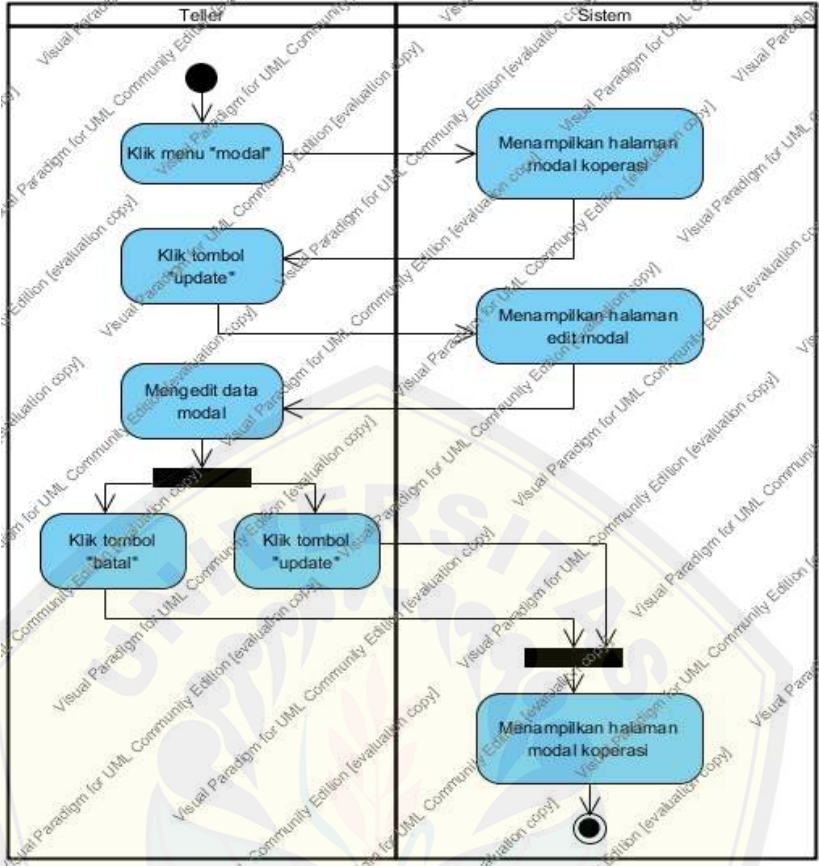




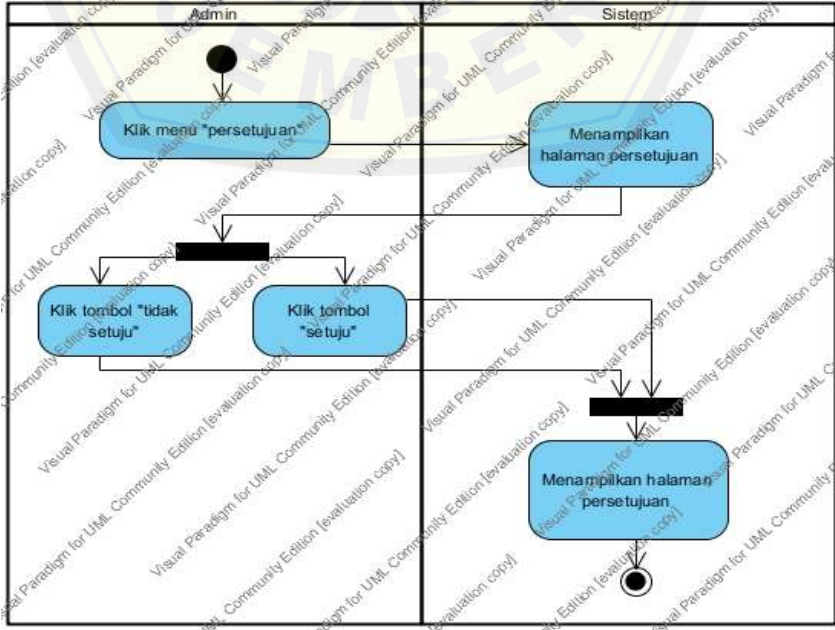
**B.5 Activity Diagram Memanajemen Data Pinjaman**



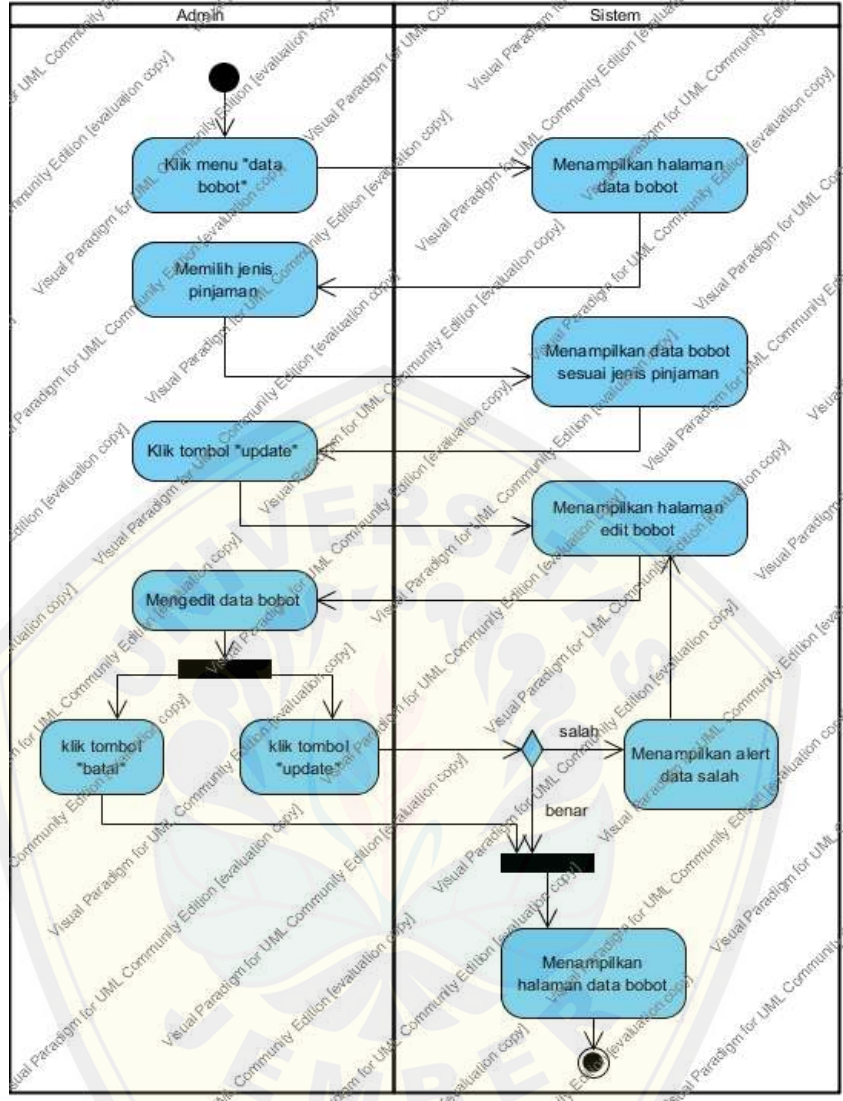
**B.6 Activity Diagram Memanajemen Modal Koperasi**



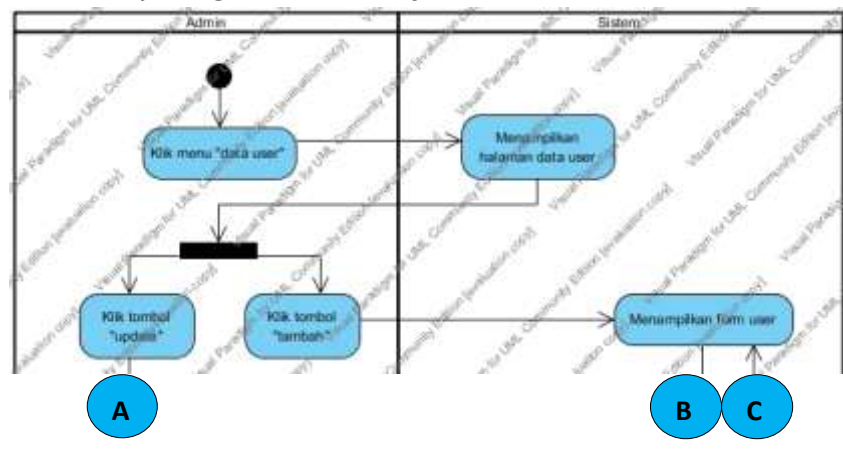
**B.7 Activity Diagram Memberikan Persetujuan**

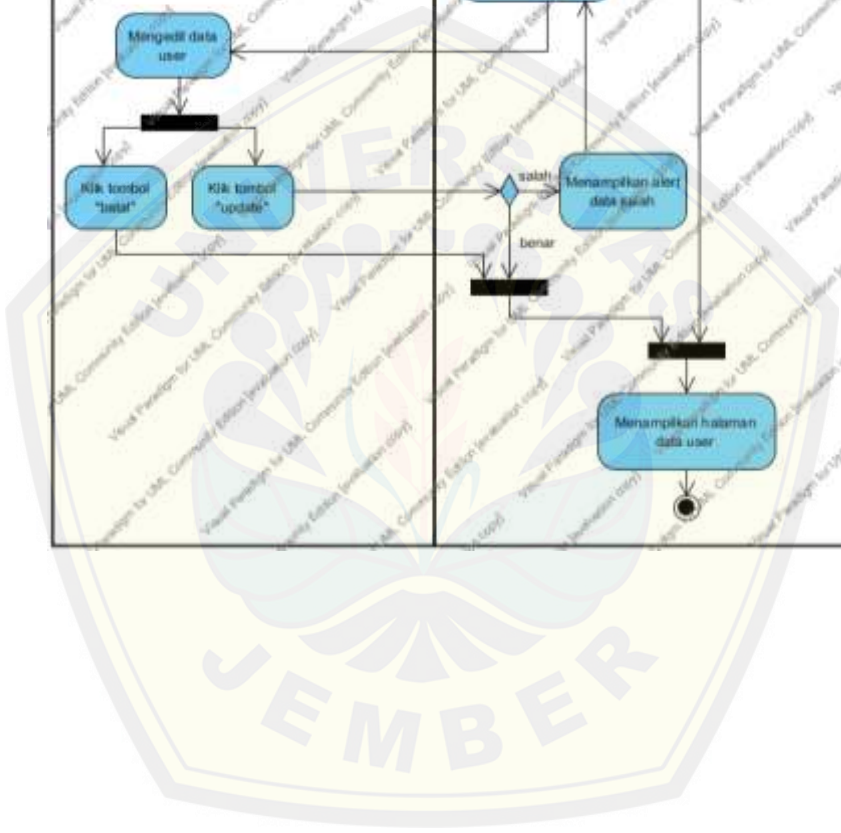
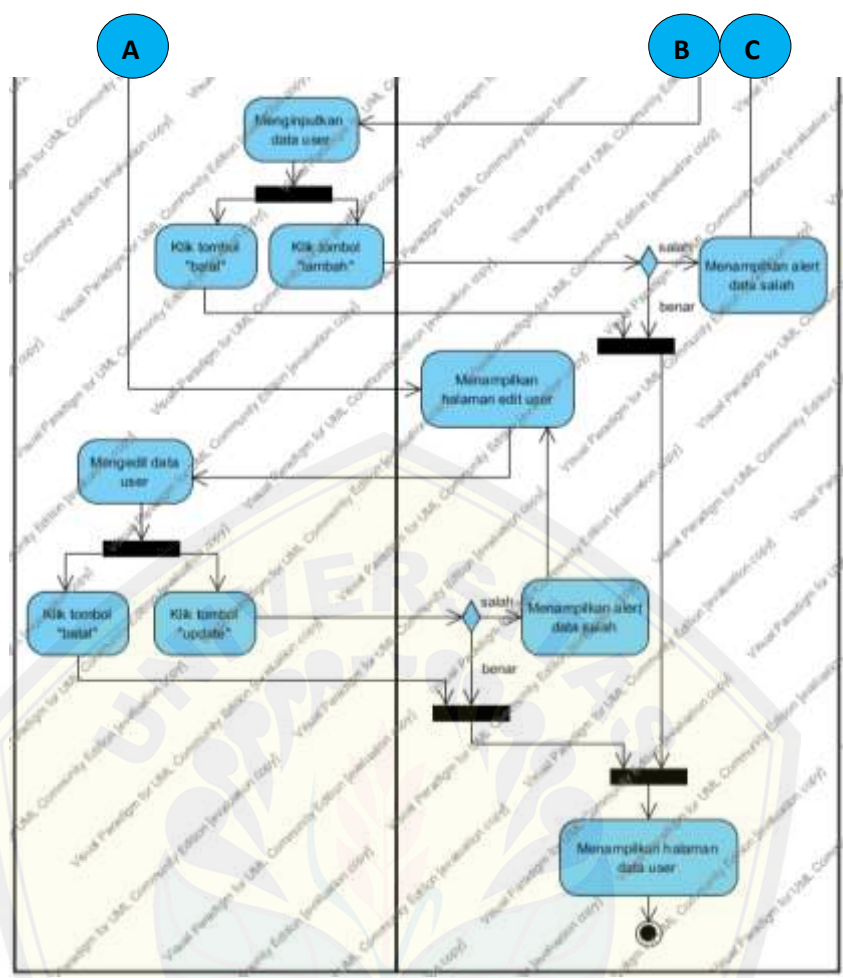


**B.8 Activity Diagram Memanajemen Data Bobot**



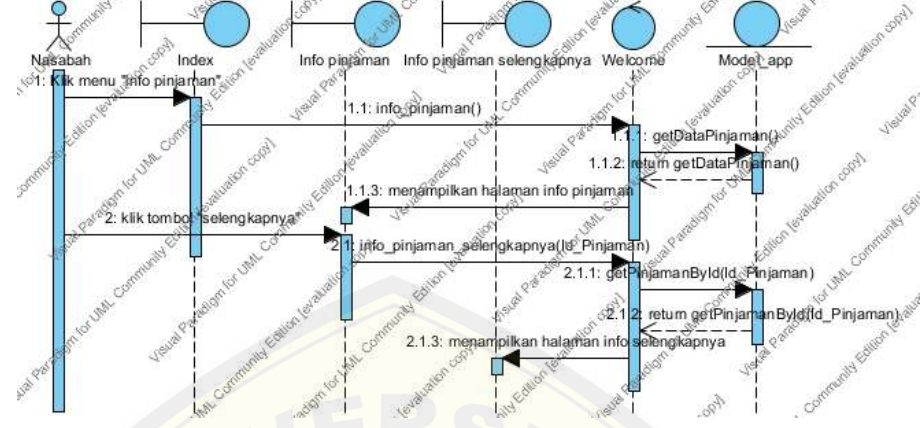
**B.9 Activity Diagram Memanajemen Data User**



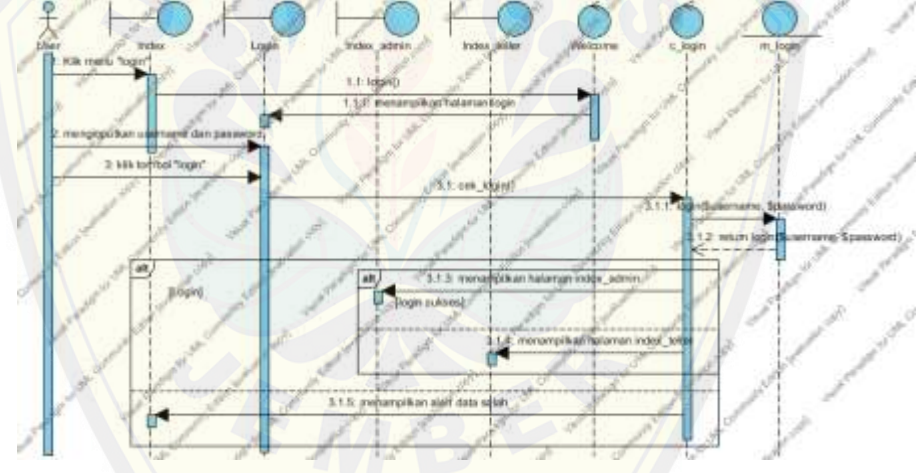


## LAMPIRAN C SEQUENCE DIAGRAM

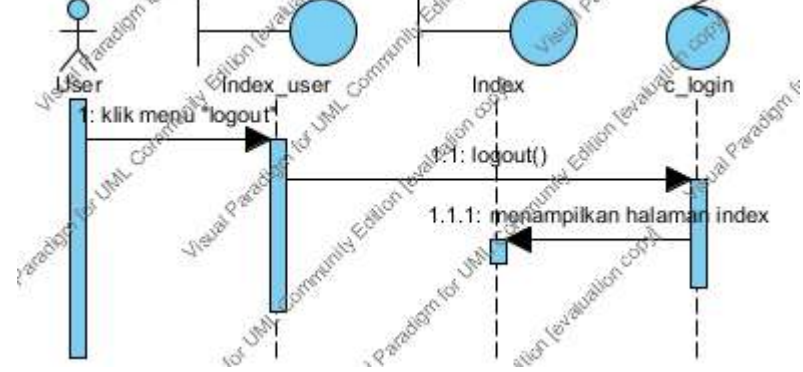
### C.1 Sequence Diagram Melihat Info Pinjaman



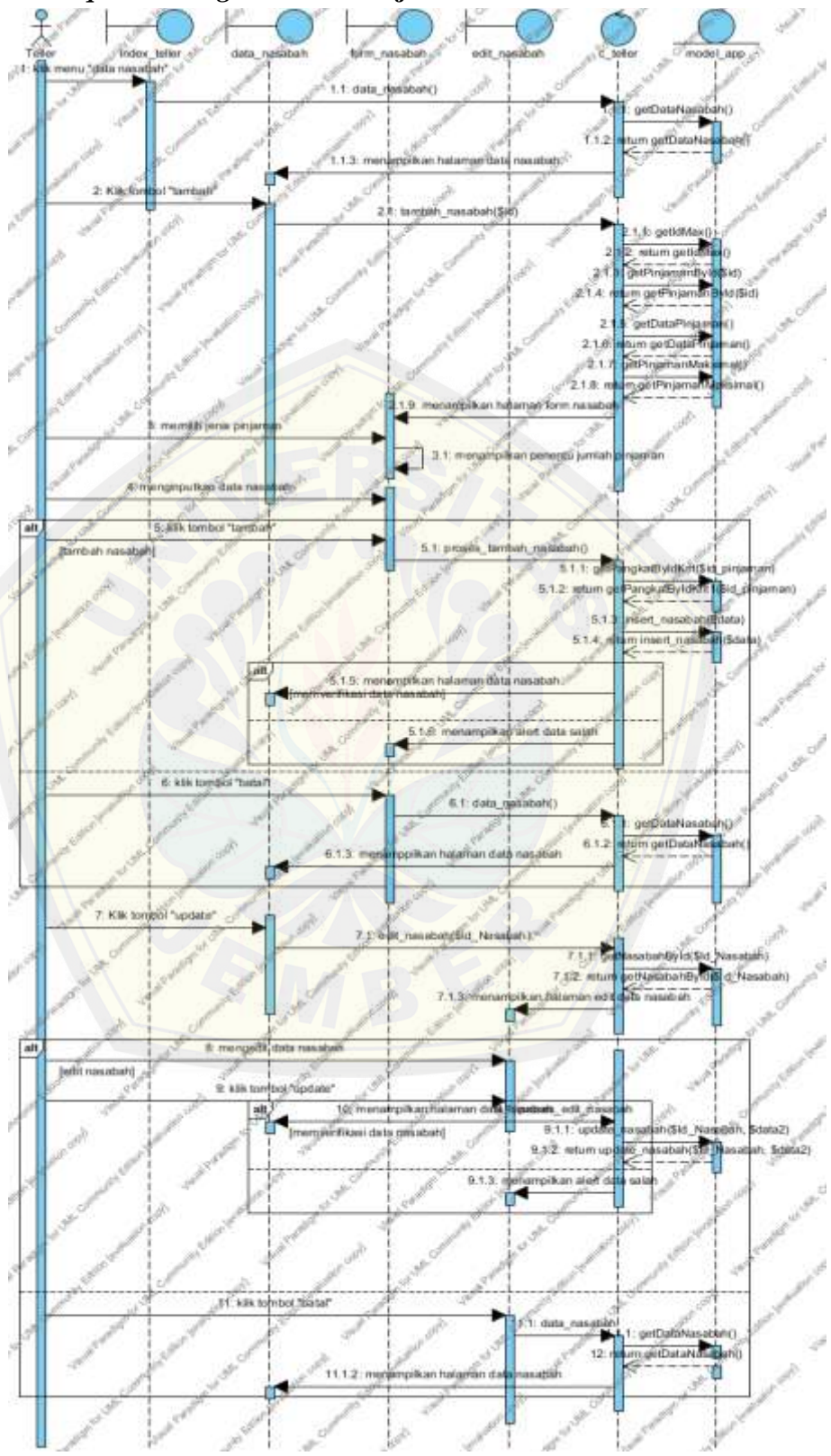
### C.2 Sequence Diagram Melakukan Login



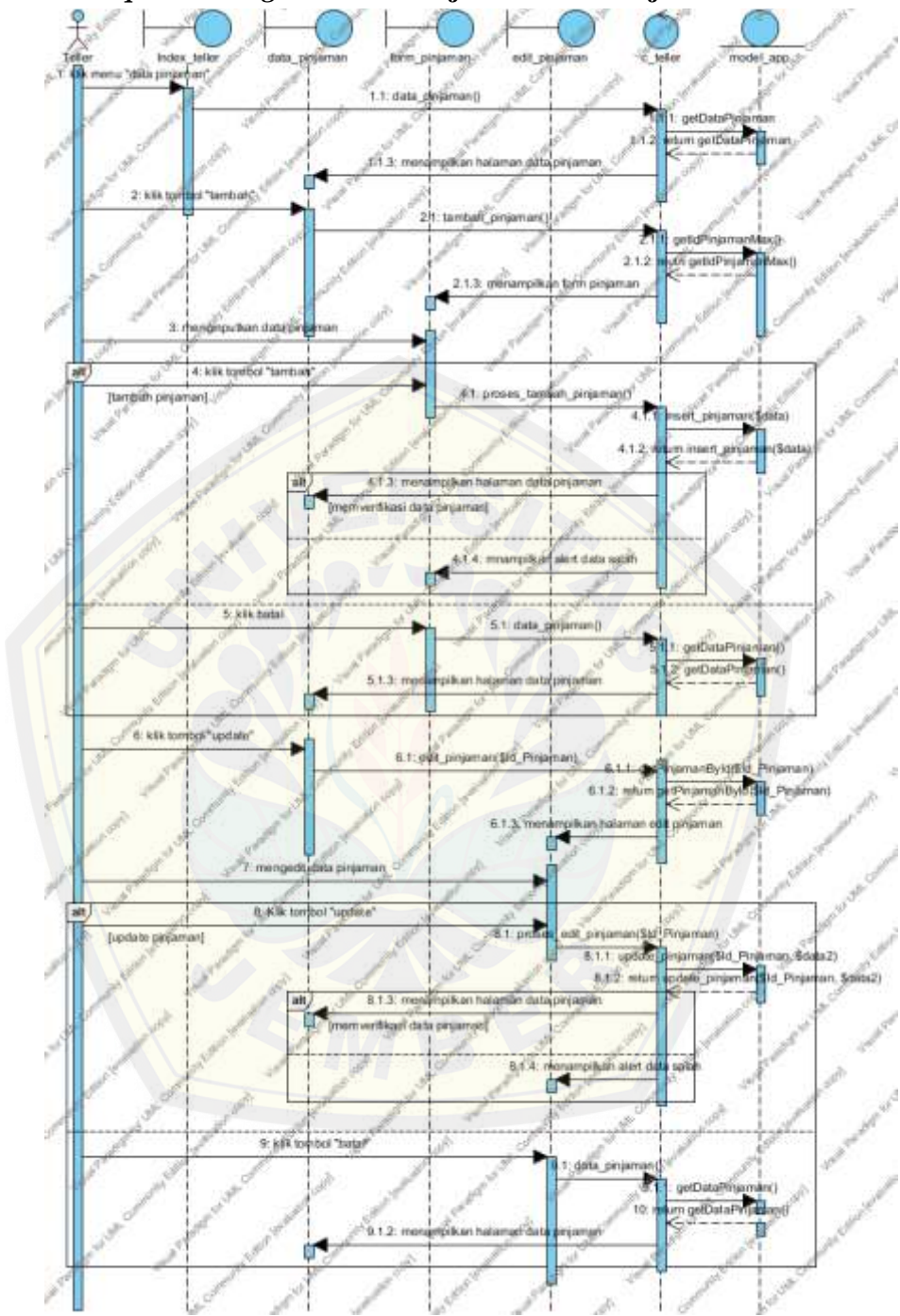
### C.3 Sequence Diagram Melakukan Logout



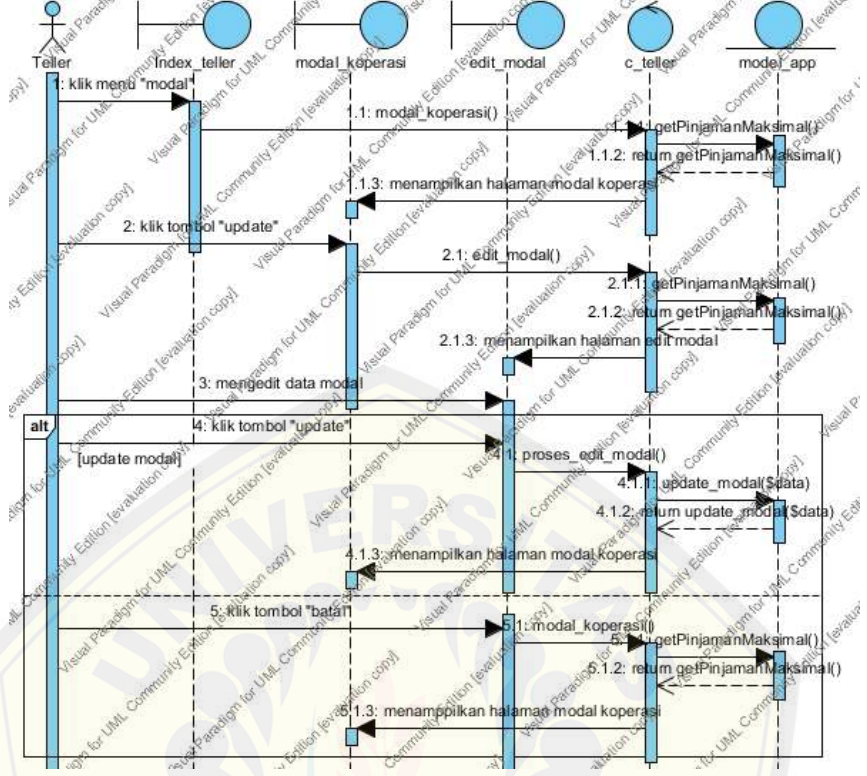
### C.4 Sequence Diagram Memanajemen Data Nasabah



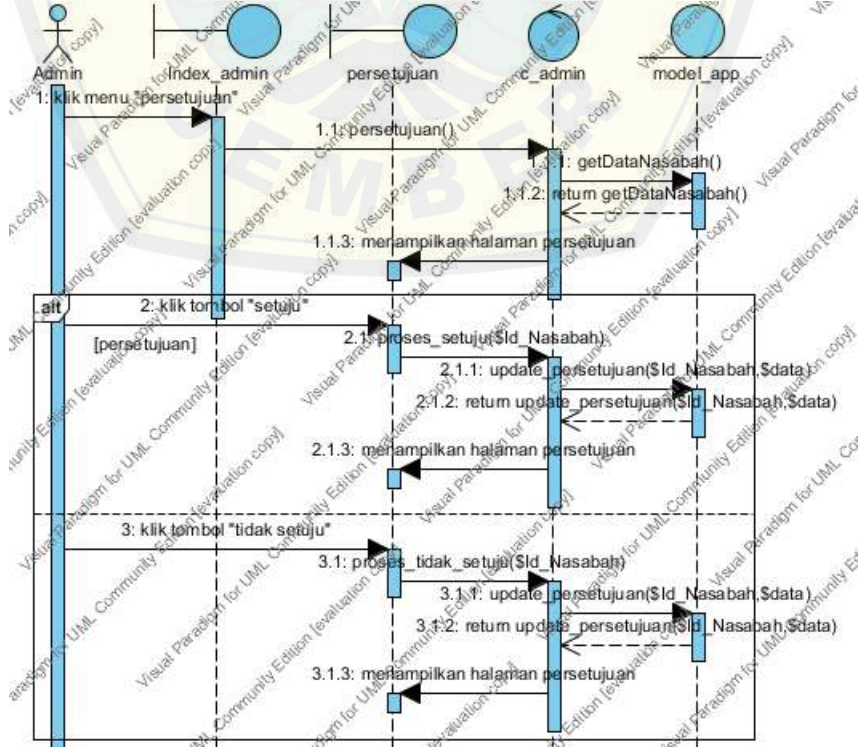
### C.5 Sequence Diagram Memanajemen Data Pinjaman



### C.6 Sequence Diagram Memanajemen Modal Koperasi

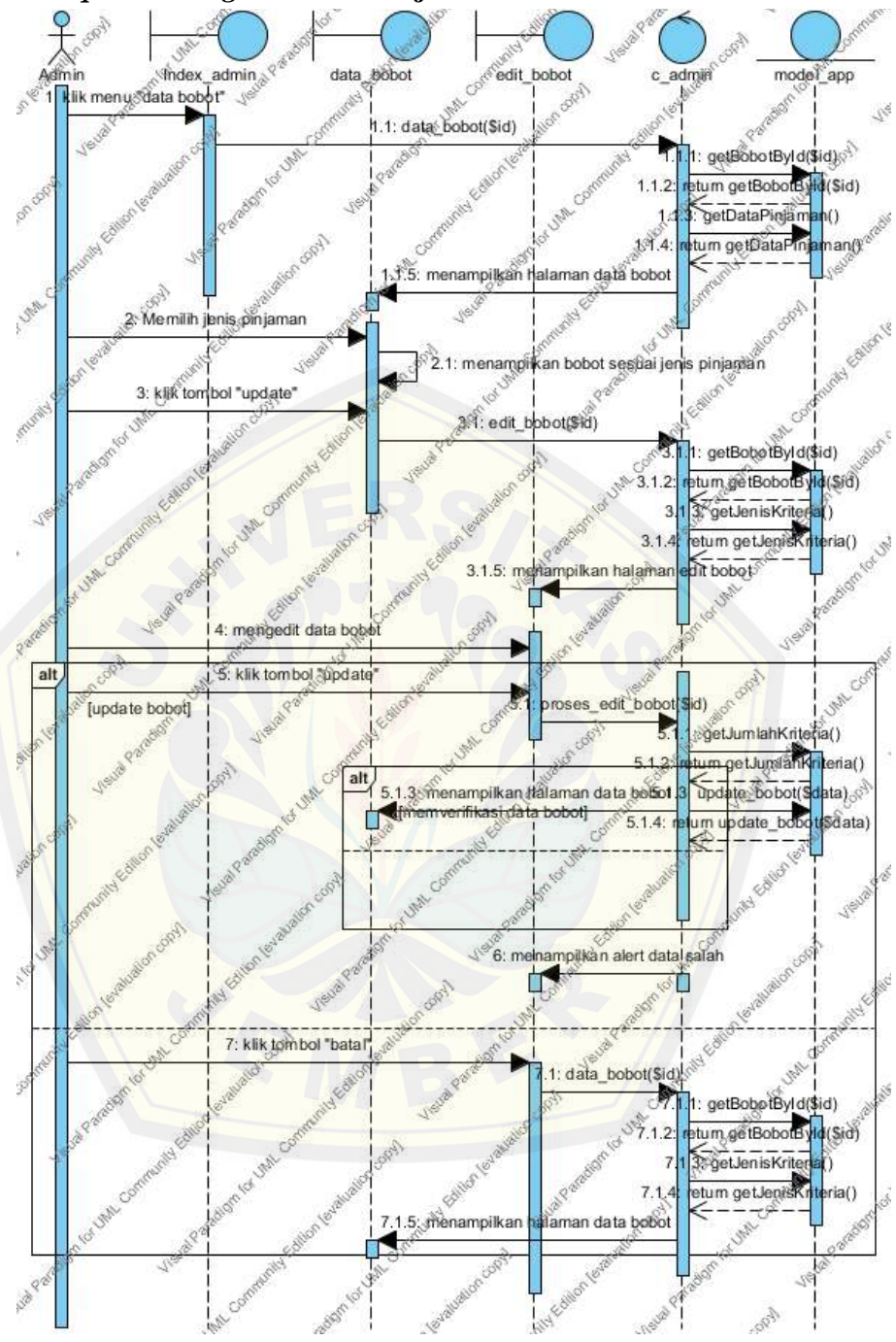


### C.7 Sequence Diagram Memberikan Persetujuan

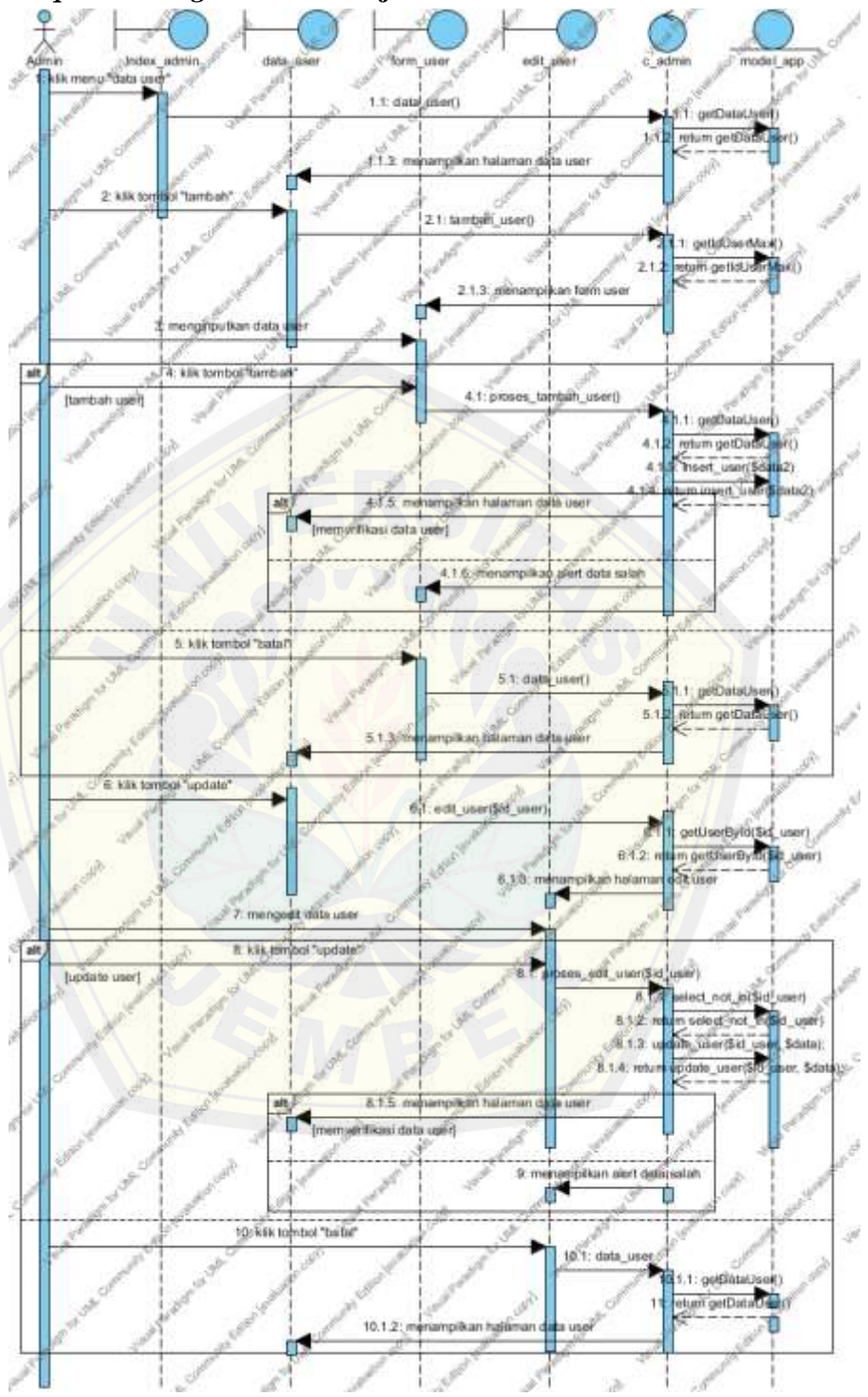




### C.8 Sequence Diagram Memanajemen Data Bobot



### C.9 Sequence Diagram Memanajemen Data User

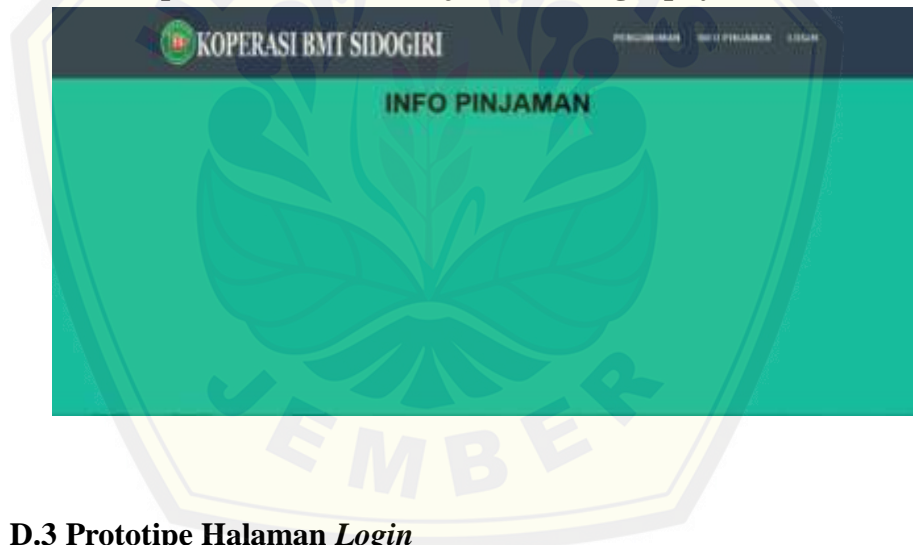


## LAMPIRAN D PROTOTIPE

### D.1 Prototipe Halaman Info Pinjaman



### D.2 Prototipe Halaman Info Pinjaman Selengkapnya



### D.3 Prototipe Halaman *Login*



#### D.4 Prototipe Halaman *Login* Salah



#### D.5 Prototipe Halaman Index Teller



#### D.6 Prototipe Halaman Index Admin



D.7 Prototipe Halaman Logout



D.8 Prototipe Halaman Data Nasabah



D.9 Prototipe Halaman Tambah Nasabah



### D. 10 Prototipe Halaman Edit Nasabah

The screenshot shows the 'UPDATE NASABAH' form. At the top, there is a header with the logo and name 'KOPERASI BMT SIDOGIRI' and navigation links: 'DATA NASABAH', 'DATA PINJAMAN', 'BERITA', and 'LAIN LAIN'. The main title 'UPDATE NASABAH' is centered with a star icon. Below the title is a form with several input fields: 'No. Nasabah' (text), 'Nama' (text), 'Tempat Lahir' (text), 'Tanggal Lahir' (date), 'No. Rekening' (text), 'Pendidikan Terakhir' (dropdown), and 'Alamat' (text). A 'Simpan' button is located at the bottom left of the form area.

### D.11 Prototipe Halaman Pinjaman

The screenshot shows the 'DATA PINJAMAN' dashboard. It features the same header as the previous page. The main title 'DATA PINJAMAN' is centered with a star icon. Below the title, there are three columns: 'No. Pinjaman', 'Nama Pinjaman', and 'Aksi'. A 'Tambah' button is located on the left side. At the bottom, there is a copyright notice: 'Copyright © 2022 Universitas Jember. Semua Hak Dilindungi.'

### D. 12 Prototipe Halaman Tambah Pinjaman

The screenshot shows the 'FORM PINJAMAN' form. It features the same header as the previous pages. The main title 'FORM PINJAMAN' is centered with a star icon. Below the title is a form with several input fields: 'No. Pinjaman' (text), 'Nama Pinjaman' (text), 'Jumlah Pinjaman' (text), and 'Tanggal Pinjaman' (date). A 'Simpan' button is located at the bottom left of the form area.

**D. 13 Prototipe Halaman Edit Pinjaman**



**D. 14 Prototipe Halaman Modal Koperasi**



**D. 15 Prototipe Halaman Edit Modal**



D. 16 Prototipe Halaman Memberikan Persetujuan



D.17 Prototipe Halaman Data Bobot



D.18 Prototipe Halaman Edit Bobot





D. 19 Prototipe Halaman Data User



D.20 Prototipe Halaman Tambah User



D.21 Prototipe Halaman Edit User



## LAMPIRAN E BARIS PROGRAM

### E.1 Baris Program Fitur Melihat Info Pinjaman

#### a. Pada Controller

```

public function info_pinjaman()
{
    $query = $this->model_app->getDataPinjaman();
    $data['title'] = $query->rowCount();
    $data['username'] = $query->user_name;

    $this->load->view('info_pinjaman', $data);
}

public function info_pinjaman_selengkapnya($ID_Pinjaman)
{
    $query = $this->model_app->getPinjamanById($ID_Pinjaman);
    $data['title'] = $query->rowCount();
    $data['username'] = $query->user_name;

    $this->load->view('info_pinjaman_selengkapnya', $data);
}
    
```

#### b. Pada Model

```

// DATA PINJAMAN
function getDataPinjaman() {
    return $this->db->query("SELECT * FROM pinjaman");
}

function getPinjamanById($id) {
    $this->db->select("*");
    $this->db->from('pinjaman');
    $this->db->where('id', $id);
    return $this->db->get();
}
    
```

### E.2 Baris Program Fitur Login

#### a. Pada Controller

```

public function login()
{
    $this->load->view('login');
}

function cek_login()
{
    //field validation succeeded. Validate against database
    $username = $this->input->get('username');
    $password = $this->input->get('password');
    //query the database
    $result = $this->db->login($username, $password);
    if($result)
    {
        $sess_array = array();
        foreach($result as $row)
        {
            //create the session
            $sess_array = array(
                'ID' => $row->ID,
                'NAME' => $row->username,
                'LEVEL' => $row->level,
                'login_status' => $row->login_status,
            );
            $this->session->sess_userdata($sess_array);
            redirect('index.php', 'refresh');
        }
    }
    return TRUE;
}
else {
    //if form validation failed
    redirect(base_url().'index.php', 'refresh');
    return FALSE;
}
}
    
```





```

function getDetail() {
    $q = $this->db->query("SELECT *FROM detail_pinjaman, t) as data_pin FROM pinjaman");
    $data = "";
    if($q->num_rows() > 0) {
        foreach($q->rows() as $row) {
            $tag = ($row->code_mak) <= 1;
            $code = sprintf($tag);
        }
    } else {
        $code = "0";
    }
    return "0000" . $code;
}

function insert_detail() {
    $this->db->insert('detail', $data);
}

```

### E.5 Baris Program Fitur Memanajemen Data Pinjaman

#### a. Pada Controller

```

// DATA PINJAMAN
public function data_pinjaman() {
    $query = $this->model_app->getDetailPinjaman();
    $data['pinjaman'] = $query->result();
    $data['pinjaman'] = $query->num_rows();
    $this->load->view('data_pinjaman', $data);
}

public function tambah_pinjaman() {
    $data=array();
    $data['pinjaman'] = $this->model_app->getIdPinjaman();
    $this->load->view('tambah_pinjaman', $data);
}

public function proses_tambah_pinjaman() {
    $id = $this->input->post('id');
    $jenis = $this->input->post('jenis');
    $keterangan = $this->input->post('keterangan');
    $keterangan2 = $this->input->post('keterangan2');

    $data['id_pinjaman'] = $id;
    $data['jenis_pinjaman'] = $jenis;
    $data['keterangan'] = $keterangan;
    $data['keterangan2'] = $keterangan2;
    $this->model_app->insert_pinjaman($data);
    redirect($site_url.'?index.php/data_pinjaman');
}

public function edit_pinjaman($id_pinjaman) {
    $data['pinjaman'] = $this->model_app->getPinjaman($id_pinjaman)->row();
    $this->load->view('edit_pinjaman', $data);
}

public function proses_edit_pinjaman($id_pinjaman) {
    $id = $this->input->post('id');
    $jenis = $this->input->post('jenis');
    $keterangan = $this->input->post('keterangan');
    $keterangan2 = $this->input->post('keterangan2');

    $data['id_pinjaman'] = $id;
    $data['jenis_pinjaman'] = $jenis;
    $data['keterangan'] = $keterangan;
    $data['keterangan2'] = $keterangan2;
    $this->model_app->update_pinjaman($id_pinjaman, $data);
    redirect($site_url.'?index.php/data_pinjaman');
}

```

#### b. Pada Model

```

function getDetailPinjaman() {
    return $this->db->query("SELECT *FROM pinjaman");
}

function getPinjamanById($id) {
    $this->db->select("");
    $this->db->from('pinjaman');
    $this->db->where('id_pinjaman', $id);
    return $this->db->get();
}

function getIdPinjaman() {
    $q = $this->db->query("SELECT *FROM detail_pinjaman, t) as data_pin FROM pinjaman");
    $code = "";
    if($q->num_rows() > 0) {
        foreach($q->rows() as $row) {
            $tag = ($row->code_mak) <= 1;
            $code = ($tag) ? ($code . $tag) : ($code);
        }
    } else {
        $code = "0";
    }
    return "0000" . $code;
}

```

```
function insert_pinjaman($data)
{
    $this->db->insert('pinjaman', $data);
}

function update_pinjaman($id_pinjaman, $data)
{
    $this->db->where('id_pinjaman', $id_pinjaman);
    $this->db->update('pinjaman', $data);
}
```

**E.6 Baris Program Fitur Memanajemen Modal Koperasi**

a. Pada Controller

```
public function modal_bayarkan()
{
    $query = $this->model_app->getDaftarModalBayar();
    $data['daftarmodal'] = $query->result();
    $data['jumlah'] = $query->num_rows();
    $this->load->view('modal_bayarkan', $data);
}

public function edit_modal()
{
    $data['daftarmodal'] = $this->model_app->getPinjamanBayar($id)->row();
    $this->load->view('edit_modal', $data);
}

public function proses_edit_modal()
{
    $model = $this->model_app->getModal();
    $id_modal = $this->input->post('id_modal');
    $id_smpk = $model->id_smpk;

    $data['modal_bayarkan'] = $model;
    $data['daftar_modal'] = $id_modal;
    $data['daftar_smpk'] = $id_smpk;

    $this->model_app->update_modal($data);
    redirect(site_url('smpk/modal_bayarkan'));
}
```

b. Pada Model

```
function getPinjamanBayar()
{
    return $this->db->query("SELECT * FROM modal");
}

function update_modal($data)
{
    $this->db->update('modal', $data);
}
```

**E.7 Baris Program Fitur Memberikan Persetujuan**

a. Pada Controller

```
public function persetujuan()
{
    $query = $this->model_app->getDaftarKeasah();
    $data['daftarkeasah'] = $query->result();
    $data['jumlah'] = $query->num_rows();
    $this->load->view('persetujuan', $data);
}

public function proses_persetujuan($id_keasah)
{
    $data['daftarkeasah'] = $keasah;

    $this->model_app->update_persetujuan($id_keasah, $data);
    redirect(site_url('smpk/persetujuan'));
}

public function proses_tidak_persetujuan($id_keasah)
{
    $data['daftarkeasah'] = $keasah;

    $this->model_app->update_persetujuan($id_keasah, $data);
    redirect(site_url('smpk/persetujuan'));
}
```

b. Pada Model

```

function getBaratMasuk() {
    return $this->db->query("SELECT * FROM barat_masuk WHERE m_id_Fasilitas = p_id_Fasilitas");
}

function update_pesertaJalan($id_peserta, $data) {
    return $this->db->query("UPDATE barat_masuk SET PesertaJalan='$data[ PesertaJalan ]' WHERE id_peserta='$id_peserta'");
}
    
```

E.8 Baris Program Fitur Memanajemen Data Bobot

a. Pada Controller

```

public function data_bobot($id)
{
    $data['aktif_bobot'] = $this->model_app->getBobotByID($id)->result();
    $data['aktif_peserta'] = $this->model_app->getPesertaJalan()->result();
    $data['id_peserta'] = $id;
    $data['id_bobot'] = $id;
    $this->load->view('data_bobot', $data);
}

public function edit_bobot($id)
{
    $data['id_peserta'] = $id;
    $data['aktif_bobot'] = $this->model_app->getBobotByID($id)->result();
    $data['pes_peserta'] = $this->model_app->getPesertaJalan()->result();
    $this->load->view('edit_bobot', $data);
}

public function proses_edit_bobot()
{
    $jmlk = $this->model_app->getJumlahFasilitas();
    $jmlkriteria = $this->input->post('jumlah_km');
    $data['id_b'] = $this->input->post('id_bobot');
    $total = 0;
    $data['pangkat_bobot'] = 0;
    $total_pangkat = 0;

    for($a=0; $a<$jmlk; $a++){
        $data['Bobot'] = $this->input->post('bobot_'.$a);
        $data['id_s'] = $this->input->post('id_s_'.$a);
        $data['pangkat_bobot'] = $data['Bobot'] * $total;
        $total_pangkat = $total_pangkat + $data['pangkat_bobot'];
    }

    if($data['id_s'] == $jmlkriteria){
        $data['s'] = $data['pangkat_bobot'] * $a - 1;
    } else {
        $data['s'] = $data['pangkat_bobot'] * $a;
    }

    $this->model_app->update_bobot($data);

    echo $data['pangkat_bobot'];

    if($total_pangkat==0){
        $this->data_bobot($data['id_b']);
    } else {
        $this->data_bobot($data['id_b'].'*'.$total_pangkat);
    }
}
    
```

b. Pada Model

```

// DATA MODEL
function getJumlahRisiko() {
    $query = $this->db->query("SELECT * FROM risiko");
    $jmlk = $query->num_rows();
    return $jmlk;
}

function getDetailRisiko() {
    return $this->db->query("SELECT * FROM tabel_r, risiko & WHERE r.id_risiko=r.id_risiko");
}

function getRisikoById($id) {
    return $this->db->query("SELECT * FROM tabel_r, risiko & WHERE r.id_risiko=r.id_risiko AND r.id_risiko=$id");
}

function getFungsiByIdPilih($idpilihan) {
    $q = $this->db->query("SELECT fungsi_risiko FROM tabel_r, risiko & WHERE r.id_risiko=$id_risiko' AND r.id_pilihan=$idpilihan");
    $pangkat = 0;
    foreach($q->result_array() as $row) {
        $pangkat = $row['fungsi_risiko'];
    }
    return $pangkat;
}

function getFungsiByIdPilih($idpilihan) {
    $q = $this->db->query("SELECT fungsi_risiko FROM tabel_r, risiko & WHERE r.id_risiko=$id_risiko' AND r.id_pilihan=$idpilihan");
    $pangkat = 0;
    foreach($q->result_array() as $row) {
        $pangkat = $row['fungsi_risiko'];
    }
    return $pangkat;
}

```

**E.9 Baris Program Fitur Manajemen Data User**  
 a. Pada *Controller*

```

// DATA USER
public function data_user() {
    $query = $this->model_app->getDataUser();
    $data['user'] = $query->result();
    $data['jumlah'] = $query->num_rows();
    $this->load->view('data_user', $data);
}

public function tambah_user() {
    $data=array();
    $data['nama'] = $this->model_app->getIDDefault();
    $this->load->view('tambah_user', $data);
}

public function proses_tambah_user() {
    $pass1 = $this->input->post('password1');
    $pass2 = $this->input->post('password2');
    $nama = $this->input->post('nama');
    $level = $this->input->post('level');

    if($pass1 == $pass2) {
        $status = true;
        $id = $this->input->post('id');
        $user = $this->model_app->getDataUser()->result();

        if($status == true) {
            $data['id_user'] = $id;
            $data['nama'] = $nama;
            $data['password'] = $pass1;
            $data['level'] = $level;
            $this->model_app->insert_user($data);

            redirect(site_url('admin/data_user'));
        } else {
            redirect(site_url('admin/tambah_user'), 'errorpage');
        }

    } else {
        redirect(site_url('admin/tambah_user'), 'errorpage');
    }
}

```



```

public function edit_user($id_user)
{
    $data['user'] = $this->model_app->getById($id_user)->row();
    $this->load->view('edit_user', $data);
}

public function proses_edit_user($id_user)
{
    $password = $this->input->post('password');
    $password2 = $this->input->post('password2');
    $id = $this->input->post('id');
    $nama = $this->input->post('nama');
    $level = $this->input->post('level');

    if($password == $password2){
        $status = true;
        $username = $this->input->post('id');
        $user = $this->model_app->select_user_by($id_user)->result();

        if($status == true){
            $data['id_user'] = $id;
            $data['username'] = $nama;
            $data['password'] = $password;
            $data['level'] = $level;
            $this->model_app->update_user($id_user, $data);

            redirect(site_url('admin/data_user'));
        } else {
            redirect(site_url('admin/edit_user/'.$id_user.'?pesan=error'));
        }
    } else {
        redirect(site_url('admin/edit_user/'.$id_user.'?pesan=error'));
    }
}

```

b. Pada Model

```

function getById($id_user)
{
    return $this->db->query("SELECT * FROM user WHERE id=$id_user");
}

function getById($id_user)
{
    $this->db->select("*");
    $this->db->from('user');
    $this->db->where('id_user', $id_user);
    return $this->db->get();
}

function select_user_by($id_user)
{
    $this->db->select("*");
    $this->db->from('user');
    $this->db->where('id_user', $id_user);
    return $this->db->get();
}

function getById($id_user)
{
    $q = $this->db->query("SELECT * FROM user WHERE id=$id_user");
    $code = "";
    if($q->num_rows() > 0){
        foreach($q->result() as $row){
            $code .= ($row['code_max']+1);
            $code = sprintf("%05s", $code);
        }
    } else {
        $code = "1";
    }
    return "0010".$code;
}

function insert_user($data)
{
    $this->db->insert('user', $data);
}

function update_user($id_user, $data)
{
    $this->db->where('id_user', $id_user);
    $this->db->update('user', $data);
}

```

## LAMPIRAN F TAMPILAN FITUR

### F.1 Tampilan Fitur Melihat Info Pinjaman



### F.2 Tampilan Fitur Melihat Info Pinjaman Selengkapnya



### F.3 Tampilan Fitur Login



**F.4 Tampilan Fitur *Login Sukses Admin***



**F.5 Tampilan Fitur *Login Sukses Teller***



**F.6 Tampilan Fitur *Login Salah***



F.7 Tampilan Fitur Logout



F.8 Tampilan Fitur Data Nasabah

ID	Nama	Jenis	Alamat	Jumlah Pinjaman	Jumlah Uang	Jumlah Bagi Bagi	Pembayaran	Status
60790	Nahana Supriah	6 0772220	14211 Jalan Cendek, Sumantri	200000	0	0000	Belum	Detail
60791	Trianggul Hidayat	6 0827420	1188 Brondongan, Sidogiri, Pajajaran	100000	0	0000	Belum	Detail
60792	Harah	6 1467220	1183 Jalan Cendek, Sumantri	200000	0	0000	Tidak Membayar	Detail
60793	Harah Hidayat	6 146807790	1183 Jalan Cendek, Sumantri	200000	0	0000	Belum	Detail
60794	Ambar Sumantri	6 1209070270	14211 Jalan Cendek, Sumantri	100000	0	0000	Belum	Detail
60795	Harah Hidayat	6 1209070270	14211 Jalan Cendek, Sumantri	100000	0	0000	Tidak Membayar	Detail

F.9 Tampilan Fitur Tambah Nasabah

F.10 Tampilan Fitur Edit Nasabah



F.11 Tampilan Fitur Data Pinjaman



F.12 Tampilan Fitur Tambah Pinjaman



**F.13 Tampilan Fitur Edit Pinjaman**



**F.14 Tampilan Fitur Modal Koperasi**



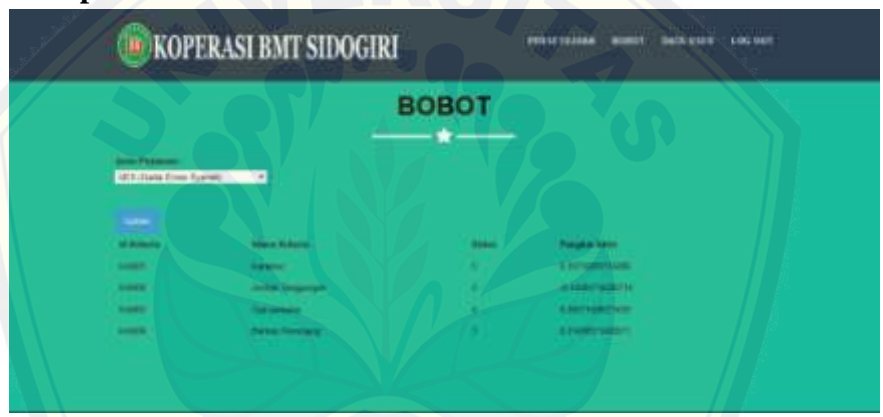
**F.15 Tampilan Fitur Edit Modal Koperasi**



F.16 Tampilan Fitur Memberikan Persetujuan



F.17 Tampilan Fitur Data Bobot



F.18 Tampilan Fitur Edit Bobot



F.19 Tampilan Fitur Data User



F.20 Tampilan Fitur Tambah User



F.21 Tampilan Fitur Edit User





# LAMPIRAN G LISTING PROGRAM WHITE\_BOX

## G.1 Listing Program Memanajemen Data Nasabah

```

// DATA NASABAH
public function data_nasabah()
{
    $query = $this->model_app->getDataNasabah();
    $data['nasabah'] = $query->result();
    $data['nama_yams'] = $query->nama_yams;

    $this->load->view('data_nasabah',$data);
}

public function tambah_nasabah($id)
{
    $data=array(
        'title'=>'tambah_nasabah',
        'link'=>$this->model_app->getLink(),
    );

    $query = $this->model_app->getPinJamsanById($id);
    $data['daftar_golongan_id'] = $query->result();
    $data['nama_yams'] = $query->nama_yams;

    $query = $this->model_app->getDataPinJamsan();
    $data['daftar_golongan'] = $query->result();
    $data['nama_yams'] = $query->nama_yams;

    $data['nama'] = $id;

    $query = $this->model_app->getPinJamsanKawil();
    $data['pda_yams'] = $query->result();
    $data['nama_yams'] = $query->nama_yams;

    $this->load->view('tambah_nasabah',$data);
}

public function proses_tambah_nasabah()
{
    $idpinJamsan = $this->input->post('idpinJamsan');

    $krit1 = $this->model_app->getPangkatByIdKrit1($idpinJamsan);
    $krit2 = $this->model_app->getPangkatByIdKrit2($idpinJamsan);
    $krit3 = $this->model_app->getPangkatByIdKrit3($idpinJamsan);
    $krit4 = $this->model_app->getPangkatByIdKrit4($idpinJamsan);

    $karakter = $this->input->post('karakter');
    $pangpangan = $this->input->post('pangpangan');
    $panghasilan = $this->input->post('panghasilan');
    $status = $this->input->post('status');

    $nilai = pow($karakter,$krit1)*pow($pangpangan,$krit2)*pow($panghasilan,$krit3)*pow($status,$krit4);

    $pinJamsan = $this->input->post('nilai');
    $panghasil = $pinJamsan*$nilai;

    if ($nilai<=1)
    {
        $status = 'Tidak Sesuai';
    }
    else
    {
        $status = 'Lulus';
    }

    $data['id_nasabah'] = $this->input->post('id');
    $data['nama_nasabah'] = $this->input->post('nama');
    $data['tgl_lahir'] = $this->input->post('tgl_lahir');
    $data['No_identifikasi'] = $this->input->post('no_identifikasi');
    $data['jenis_kelamin'] = $this->input->post('jenis_kelamin');
    $data['alamat'] = $this->input->post('alamat');
    $data['telepon'] = $this->input->post('no_telepon');
    $data['nama_yams'] = $this->input->post('nama_yams');

    $data['nama_hala'] = $this->input->post('nama_hala');
    $data['ID_Pinjamsan'] = $this->input->post('idpinJamsan');
    $data['Jumlah_Pinjamsan'] = $this->input->post('jumlah');
    $data['Status_Pinjamsan'] = $this->input->post('status');
    $data['Jumlah_Hala'] = $this->input->post('jumlah_hala');
    $data['Temp_Hala'] = $panghasilan;
    $data['Masa'] = $nilai;
    $data['Status'] = $status;
    $data['id_user'] = 'admin';

    $this->model_app->saveData_nasabah($data);

    redirect($this->url('tambah_nasabah'));
}

public function edit_nasabah($id_nasabah)
{
    $data['nasabah'] = $this->model_app->getNasabahById($id_nasabah)->row();

    $this->load->view('edit_nasabah',$data);
}

```

```

public function proses_edit_nasabah($id_nasabah)
{
    $id = $this->input->post('id');
    $nama = $this->input->post('nama');
    $tgl_lahir = $this->input->post('tgl_lahir');
    $no_identitas = $this->input->post('noidentitas');
    $pendaftaran = $this->input->post('pendaftaran');
    $alamat = $this->input->post('alamat');
    $ortu1g = $this->input->post('no_tlg');
    $pekerjaan = $this->input->post('pekerjaan');
    $nomorwali = $this->input->post('nomorwali');
    $telp = $this->input->post('telp');
    $jangkawaktu = $this->input->post('jangkawaktu');
    $bilang = $this->input->post('bilang');
    $bilangnasabah = $this->input->post('bilangnasabah');

    $data['Nama_Nasabah'] = $nama;
    $data['Tgl_lahir'] = $tgl_lahir;
    $data['No_Identitas'] = $noidentitas;
    $data['Pend_Pendaftaran'] = $pendaftaran;
    $data['Alamat'] = $alamat;
    $data['Ortu1g'] = $ortu1g;
    $data['Pekerjaan'] = $pekerjaan;
    $data['Nomor_Wali'] = $nomorwali;
    $data['No_Telp'] = $telp;
    $data['Jangka_Waktu'] = $jangkawaktu;
    $data['Bngl'] = $bilang;
    $data['BnglNasabah'] = $bilangnasabah;
    $this->model_app->update_nasabah($id_nasabah, $data);

    redirect('site_url/'.$this->url_nasabah);
}

```

G.2 Listing Program Memanajemen Data Bobot

```

public function data_bobot($id)
{
    $data['daftar_bobot'] = $this->model_app->getBobotByIs($id)->result();
    $data['daftar_pinjaman'] = $this->model_app->getDataPinjaman()->result();
    $data['id_pinjaman'] = $id;
    $data['nama'] = $id;
    $this->load->view('data_bobot', $data);
}

public function edit_bobot($id)
{
    $data['id_pinjaman'] = $id;
    $data['daftar_bobot'] = $this->model_app->getBobotByIs($id)->result();
    $data['no_kriteria'] = $this->model_app->getNoKriteria()->result();
    $this->load->view('edit_bobot', $data);
}

public function proses_edit_bobot()
{
    $jmlk = $this->model_app->getJumlahKriteria();
    $jmlkriteria = $this->input->post('jmlkriteria');
    $data['id_p'] = $this->input->post('id_pinjaman');
    $total = 0;
    $data['pangkat_sakit'] = 0;
    $total_pangkat = 0;

    for($k=1; $k<=$jmlk; $k++){
        $data['bobot'] = $this->input->post('bobot_'.$k);
        $total = $total + $data['bobot'];
    }

    for($k=1; $k<=$jmlk; $k++){
        $data['bobot'] = $this->input->post('bobot_'.$k);
        $data['id_k'] = $this->input->post('id_k_'.$k);

        $data['pangkat_sakit'] = $data['bobot'] / $total;

        $total_pangkat = $total_pangkat + $data['pangkat_sakit'];

        if($data['id_k'] == $jmlkriteria){
            $data['a'] = $data['pangkat_sakit']*100;
        }
        else{
            $data['a'] = $data['pangkat_sakit']*10;
        }
    }

    $this->model_app->update_bobot($data);

    echo $data['pangkat_sakit'];
}

if($total_pangkat==100){
    $this->data_bobot($data['id_p']);
}
else{
    $this->data_bobot($data['id_p'].'(pangkat_sakit)');
}

```

## LAMPIRAN H *BLACK-BOX TESTING*

### *H.1 Testing Balck Box Melihat Info Pinjaman*

<b>Pengujian Melihat Info Pinjaman</b>		
<b>Nasabah</b>	<b>Sistem</b>	<b>Berhasil/Tidak</b>
1. Membuka aplikasi		Berhasil
	2. Menampilkan halaman utama dengan menu Pengumuman, Info Pinjaman, dan Login	Berhasil
3. Klik menu “info pinjaman”		Berhasil
	4. Menampilkan halaman info pinjaman	Berhasil
	5. Menampilkan daftar pinjaman	Berhasil
6. Klik tombol “selengkapnya”		Berhasil
	7. Menampilkan detail info pinjaman	Berhasil

### *H.2 Testing Balck Box Login*

<b>Pengujian Melakukan Login</b>		
<b>User</b>	<b>Sistem</b>	<b>Berhasil/Tidak</b>
1. Membuka aplikasi		Berhasil
	2. Menampilkan halaman utama dengan menu Pengumuman,	Berhasil

dilanjutkan

lanjutan

	Info Pinjaman, dan Login	
3. Mengklik menu “login”		Berhasil
	4. Menampilkan halaman login	Berhasil
5. Menginputkan username dan password		Berhasil
6. Klik tombol “login”		Berhasil
	7. Memverifikasi username dan password	Berhasil
	8. Username dan password valid	Berhasil
	9. Menampilkan halaman admin	Berhasil
<b>Aliran Alternatif</b>		
5. Menginputkan username dan password		Berhasil
6. Klik tombol “login”		Berhasil
	7. Memverifikasi username dan password	Berhasil
	8. Username dan password valid	Berhasil
	9. Menampilkan halaman teller	Berhasil
<b>Aliran Alternatif</b>		
5. Menginputkan username dan password		Berhasil
6. Klik tombol “login”		Berhasil

dilanjutkan

lanjukan

	7. Memverifikasi username dan password	Berhasil
	8. Username dan password salah	Berhasil
	9. Menampilkan alert username/ password salah	Berhasil
	10. Menampilkan halaman beranda	Berhasil

### H.3 Testing Balck Box Logout

Pengujian Melakukan Logout		
User	Sistem	Berhasil/Tidak
1. Klik menu “logout”		Berhasil
	2. Menampilkan halaman beranda awal dengan menu Info Nasabah, Info Pinjaman, dan Login	Berhasil

### H.4 Testing Balck Box Memanajemen Data Nasabah

Pengujian Memanajemen Data Nasabah		
Teller	Sistem	Berhasil/Tidak
1. Klik menu “data nasabah”		Berhasil
	2. Menampilkan halaman data serta tombol tambah dan	Berhasil

dilanjutkan

lanjutan

	update	
3. Klik tombol “tambah”		Berhasil
	4. Menampilkan halaman form pendaftaran nasabah dengan tombol tambah dan batal	Berhasil
5. Memilih jenis pinjaman		Berhasil
	6. Menampilkan penentu jumlah pinjaman pada form data nasabah	Berhasil
7. Menginputkan data nasabah		Berhasil
8. Klik tombol “tambah”		Berhasil
	9. Memverifikasi data nasabah	Berhasil
	10. Data nasabah benar, data tersimpan	Berhasil
	11. Menampilkan halaman data nasabah	Berhasil
<b>Aliran Alternatif</b>		
7. Menginputkan data nasabah		Berhasil
8. Klik tombol “batal”		Berhasil
	9. Data tidak tersimpan	Berhasil

dilanjutkan

lanjutan

	10. Menampilkan halaman data nasabah	Berhasil
<b>Aliran Alternatif</b>		
7. Menginputkan data nasabah		Berhasil
8. Klik tombol “tambah”		Berhasil
	9. Memverifikasi data nasabah	Berhasil
	10. Data salah, data tidak tersimpan	Berhasil
	11. Menampilkan alert “data salah” dan menampilkan form pendaftaran nasabah	Berhasil
<b>Aliran Alternatif</b>		
3. Klik tombol “update” pada salah satu baris data nasabah		Berhasil
	4. Menampilkan halaman edit nasabah dengan tombol update dan batal	Berhasil
5. Mengedit data nasabah		Berhasil
6. Klik tombol “update”		Berhasil
	7. Memverifikasi data nasabah	Berhasil
	8. Data nasabah benar, data tersimpan	Berhasil

d lanj utkan

lanjutan

	9. Menampilkan halaman data nasabah	Berhasil
<b>Aliran Alternatif</b>		
5. Mengedit data nasabah		Berhasil
6. Klik tombol “batal”		Berhasil
	7. Data tidak tersimpan	Berhasil
	8. Menampilkan halaman data nasabah	Berhasil
<b>Aliran Alternatif</b>		
5. Mengedit data nasabah		Berhasil
6. Klik tombol “update”		Berhasil
	7. Memverifikasi data nasabah	Berhasil
	8. Data salah, data tidak tersimpan	Berhasil
	9. Menampilkan alert “data salah” dan menampilkan alamat edit nasabah	Berhasil

#### ***H.5 Testing Balck Box Memanajemen Data Pinjaman***

<b>Pengujian Memanajemen Data Pinjaman</b>		
<b>Teller</b>	<b>Sistem</b>	<b>Berhasil/Tidak</b>
1. Klik menu “data pinjaman”		Berhasil

dilanjutkan



lanjutan

	2. Menampilkan halaman data pinjaman serta tombol tambah dan update	Berhasil
3. Klik tombol “tambah”		Berhasil
	4. Menampilkan halaman form data pinjaman dengan tombol tambah dan batal	Berhasil
5. Menginputkan data pinjaman		Berhasil
6. Klik tombol “tambah”		Berhasil
	7. Memverifikasi data pinjaman	Berhasil
	8. Data pinjaman benar, data tersimpan	Berhasil
	9. Menampilkan halaman data pinjaman	Berhasil
<b>Aliran Alternatif</b>		
5. Menginputkan data pinjaman		Berhasil
6. Klik tombol “batal”		Berhasil
	7. Data tidak tersimpan	Berhasil
	8. Menampilkan halaman data pinjaman	Berhasil

dilanjutkan

lanjutan

<b>Aliran Alternatif</b>		
5. Menginputkan data pinjaman		Berhasil
6. Klik tombol “tambah”		Berhasil
	7. Memverifikasi data pinjaman	Berhasil
	8. Data salah, data tidak tersimpan	Berhasil
	9. Menampilkan alert “data salah” dan menampilkan form data pinjaman	Berhasil
<b>Aliran Alternatif</b>		
3. Klik tombol “update” pada salah satu baris data pinjaman		Berhasil
	6. Menampilkan halaman edit pinjaman dengan tombol update dan batal	Berhasil
7. Mengedit data pinjaman		Berhasil
8. Klik tombol “update”		Berhasil
	9. Memverifikasi data pinjaman	Berhasil
	10. Data pinjaman benar, data tersimpan	Berhasil
	11. Menampilkan halaman data pinjaman	Berhasil

dilanjutkan

lanjutan

<b>Aliran Alternatif</b>		
5. Mengedit data pinjaman		Berhasil
6. Klik tombol “batal”		Berhasil
	7. Data tidak tersimpan	Berhasil
	8. Menampilkan halaman data pinjaman	Berhasil
<b>Aliran Alternatif</b>		
5. Mengedit data pinjaman		Berhasil
6. Klik tombol “update”		Berhasil
	7. Memverifikasi data pinjaman	Berhasil
	8. Data salah, data tidak tersimpan	Berhasil
	9. Menampilkan alert “data salah” dan menampilkan halaman edit pinjaman	Berhasil

#### **H.6 Testing Balck Box Memanajemen Modal Koperasi**

<b>Pengujian Memanajemen Modal Koprasi</b>		
<b>Teller</b>	<b>Sistem</b>	<b>Berhasil/Tidak</b>
1. Klik menu “modal”		Berhasil
	2. Menampilkan halaman modal koperasi serta tombol update	Berhasil
3. Klik tombol “update”		Berhasil
	4. Menampilkan	Berhasil

dilanjutkan

lanjutan

	halaman update modal	
5. Mengganti nilai modal		Berhasil
6. Klik tombol “update”		Berhasil
	7. Menampilkan halaman modal operasi	Berhasil
<b>Aliran Alternatif</b>		
5. Mengganti nilai modal		Berhasil
6. Klik tombol “batal”		Berhasil
	7. Data tidak tersimpan	Berhasil
	8. Menampilkan halaman modal operasi	Berhasil

### H.7 Testing Balck Box Memberikan Persetujuan

<b>Pengujian Memberikan Persetujuan</b>		
<b>Admin</b>	<b>Sistem</b>	<b>Berhasil/Tidak</b>
1. Klik menu “persetujuan”		Berhasil
	2. Menampilkan halaman persetujuan serta tombol “setujui” dan tombol “tidak disetujui”	Berhasil
3. Klik tombol “setujui”		Berhasil
	4. Menampilkan halaman persetujuan dengan	Berhasil

dilanjutkan

lanjutan

	data nasabah disetujui	
<b>Aliran Alternatif</b>		
3. Klik tombol “tidak disetujui”		Berhasil
	4. Menampilkan halaman persetujuan dengan data nasabah tidak disetujui	Berhasil

#### ***H.8 Testing Balck Box Memanajemen Data Bobot***

<b>Pengujian Memanajemen Data Bobot</b>		
<b>Admin</b>	<b>Sistem</b>	<b>Berhasil/Tidak</b>
1. Klik menu “data bobot”		Berhasil
	2. Menampilkan halaman data bobot serta select button data pinjaman dan tombol update	Berhasil
3. Memilih data pinjaman yang akan dilihat bobotnya		Berhasil
	4. Menampilkan data bobot sesuai data pinjaman yang dipilih	Berhasil
5. Klik tombol “update”		Berhasil
	6. Menampilkan	Berhasil

dilanjutkan

lanjutan

	halaman update bobot sesuai data yang yang dipilih dengan tombol update dan batal	
7. Mengganti nilai bobot		Berhasil
8. Klik tombol "update"		Berhasil
	9. Menghitung nilai bobot	Berhasil
	10. Jumlah bobot = 1, nilai tersimpan	Berhasil
	11. Menampilkan halaman data bobot	Berhasil
<b>Aliran Alternatif</b>		
7. Mengganti nilai bobot		Berhasil
12. Klik tombol "batal"		Berhasil
	13. Data tidak tersimpan	Berhasil
	14. Menampilkan halaman data bobot	Berhasil
<b>Aliran Alternatif</b>		
7. Mengganti nilai bobot		Berhasil
13. Klik tombol "update"		Berhasil
	14. Menghitung nilai bobot	Berhasil
	15. Jumlah bobot tidak sama dengan 1	Berhasil
	16. Menampilkan alert	Berhasil

dilanjutkan

lanjutan

	data salah	
	17. Menampilkan halaman data bobot	Berhasil

### H.9 Testing Balck Box Memanajemen Data User

Pengujian Memanajemen Data User		
Admin	Sistem	Berhasil/Tidak
1. Klik menu “data user”		Berhasil
	2. Menampilkan halaman data user serta tombol tambah dan update	Berhasil
3. Klik tombol “tambah”		Berhasil
	4. Menampilkan halaman form data user dengan tombol tambah dan batal	Berhasil
5. Menginputkan data user		Berhasil
6. Klik tombol “tambah”		Berhasil
	7. Memverifikasi data user	Berhasil
	8. Data user benar, data tersimpan	Berhasil
	9. Menampilkan halaman data user	Berhasil
Aliran Alternatif		
5. Menginputkan data user		Berhasil

dilanjutkan

lanjutan

6. Klik tombol “batal”		Berhasil
	7. Data tidak tersimpan	Berhasil
	8. Menampilkan halaman data user	Berhasil
<b>Aliran Alternatif</b>		
5. Menginputkan data user		Berhasil
6. Klik tombol “tambah”		Berhasil
	7. Memverifikasi data user	Berhasil
	8. Data salah, data tidak tersimpan	Berhasil
	9. Menampilkan alert “data salah” dan menampilkan form data user	Berhasil
<b>Aliran Alternatif</b>		
3. Klik tombol “update” pada salah satu baris data user		Berhasil
	4. Menampilkan halaman edit user dengan tombol update dan batal	Berhasil
5. Mengedit data user		Berhasil
6. Klik tombol “update”		Berhasil
	7. Memverifikasi data user	Berhasil
	8. Data user benar, data tersimpan	Berhasil

dilanjutkan



lanjutan

	9. Menampilkan halaman data user	Berhasil
<b>Aliran Alternatif</b>		
5. Mengedit data user		Berhasil
6. Klik tombol “batal”		Berhasil
	7. Data tidak tersimpan	Berhasil
	8. Menampilkan halaman data user	Berhasil
<b>Aliran Alternatif</b>		
5. Mengedit data user		Berhasil
6. Klik tombol “update”		Berhasil
	7. Memverifikasi data user	Berhasil
	8. Data salah, data tidak tersimpan	Berhasil
	9. Menampilkan alert “data salah” dan menampilkan halaman edit user	Berhasil