



**SISTEM INFORMASI REKOMENDASI MAHASISWA FAKULTAS  
ILMU KOMPUTER TERHADAP PENERIMAAN BEASISWA  
MENGUNAKAN METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)**

**SKRIPSI**

Oleh

**Lady Anindya Firdauzi**

**NIM 132410101038**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER**

**2018**



**SISTEM INFORMASI REKOMENDASI MAHASISWA FAKULTAS  
ILMU KOMPUTER TERHADAP PENERIMAAN BEASISWA  
MENGUNAKAN METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)**

**SKRIPSI**

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan pendidikan di Program Studi Sistem Informasi Universitas  
Jember dan mendapat gelar Sarjana Sistem Informasi

Oleh

**Lady Anindya Firdauzi**

**NIM 132410101038**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS JEMBER**

**2018**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi.
2. Ibunda tercinta Siti Cholifah, Ayahanda Achmad Junaidi atas segala kesabaran, keikhlasan, limpahan kasih sayang, doa dan motivasi yang luar biasa selama hidup saya.
3. Saudara Perempuan Nadia Julfa Syabania.
4. Saudara-saudaraku beserta seluruh keluarga besar.
5. Almater Program Studi Sistem Informasi Universitas Jember

**MOTTO**

“Maka Sesungguhnya Bersama Kesulitan Ada Kemudahan”

(QS. Al-Insyirah, 6-8)



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Lady Anindya Firdauzi

NIM : 132410101038

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Rekomendasi Mahasiswa Fakultas Ilmu Komputer Terhadap Penerimaan Beasiswa Menggunakan Metode *Simple Additive Weighting* (SAW)” 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 di kemudian hari pernyataan ini tidak benar.

Jember, Juli 2018

Yang menyatakan,

Lady Anindya Firdauzi

NIM 132410101038

**SKRIPSI**

**SISTEM INFORMASI REKOMENDASI MAHASISWA FAKULTAS  
ILMU KOMPUTER TERHADAP PENERIMAAN BEASISWA  
MENGUNAKAN METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)**

Oleh :

Lady Anindya Firdauzi

NIM 132410101038

Pembimbing

Dosen Pembimbing Utama : Prof. Drs. Slamir, M.Comp.Sc.,Ph.D

Dosen Pembimbing Pendamping : Oktalia Juwita, S.Kom., M.MT

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Sistem Informasi Rekomendasi Mahasiswa Fakultas Ilmu Komputer Terhadap Penerimaan Beasiswa Menggunakan Metode *Simple Additive Weighting*”, telah diuji dan disahkan pada:

hari, tanggal : 23 Juli 2018

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Prof. Drs. Slamun, M.Comp.Sc.,Ph.D  
NIP. 19670420 1992011001

Oktalia Juwita, S.Kom., M.MT  
NIP 198110202014042001

**PENGESAHAN PENGUJI**

Skripsi berjudul “Sistem Informasi Rekomendasi Mahasiswa Fakultas Ilmu Komputer Terhadap Penerimaan Beasiswa Menggunakan Metode *Simple Additive Weighting* (Saw)”, telah diuji dan disahkan pada:

hari, tanggal : 23 Juli 2018

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Windi Eka Yulia Retnan, S.Kom., MT    Gama Wisnu Fajarianto, S.Kom., M.Kom

NIP. 198403052010122002

NIP. 760015717

Mengesahkan

Ketua Program Studi

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

NIP. 19670420 1992011001

## RINGKASAN

**Sistem Informasi Rekomendasi Mahasiswa Fakultas Ilmu Komputer Terhadap Penerimaan Beasiswa Menggunakan Metode *Simple Additive Weighting* (SAW);** Lady Anindya Firdauzi, 132410101038; 2018, 107 halaman; Program Studi Sistem Informasi Universitas Jember.

Penelitian rekomendasi mahasiswa terhadap penerimaan beasiswa ini dilakukan untuk mengatasi permasalahan mahasiswa yang memilih beasiswa sesuai kriteria. Dari permasalahan yang telah disebutkan, maka diperlukan sebuah sistem yang dapat mendukung proses rekomendasi beasiswa dengan menggunakan metode *Simple Additive Weighting*. Pengembangan sistem ini menggunakan model SDLC *Waterfall* dan dibangun dengan basis *website*. Implementasi sistem informasi dilakukan dengan melakukan pengkodean yang menggunakan *framework* Laravel 5.4 yang merupakan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP). *Tools* yang digunakan yaitu *Sublime* sebagai editor teks dan *MariaDB* sebagai *database management system*. Tahap pengujian yang dilakukan adalah *black box*, dan pengujian metode *Additive Weighting*.

Hasil dari penelitian ini yaitu sistem dapat menerapkan metode *Simple Additive Weighting* dengan akurat karena sesuai dengan perhitungan manual. Pada tahap pertama sistem akan melakukan perhitungan terhadap nilai kriteria dari data pendaftar. Kemudian sistem akan menampilkan hasil rekomendasi perhitungan metode *Simple Additive Weighting* yang hasil akhirnya di sortir dari nilai tertinggi. Selanjutnya Lembaga melakukan penetapan rekomendasi beasiswa di Fakultas Ilmu Komputer dengan mengacu pada hasil perhitungan menggunakan metode *Simple Additive Weighting*. Penetapan rekomendasi mahasiswa terhadap penerimaan beasiswa merupakan tahapan terakhir dalam sistem ini.

## PRAKATA

*Alhamdulillah* atas segala sesuatu yang diberikan Allah SWT karena kurunia-Nya dan ridha-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Informasi Rekomendasi Mahasiswa Fakultas Ilmu Komputer Terhadap Penerimaan Beasiswa Menggunakan Metode *Simple Additive Weighting*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

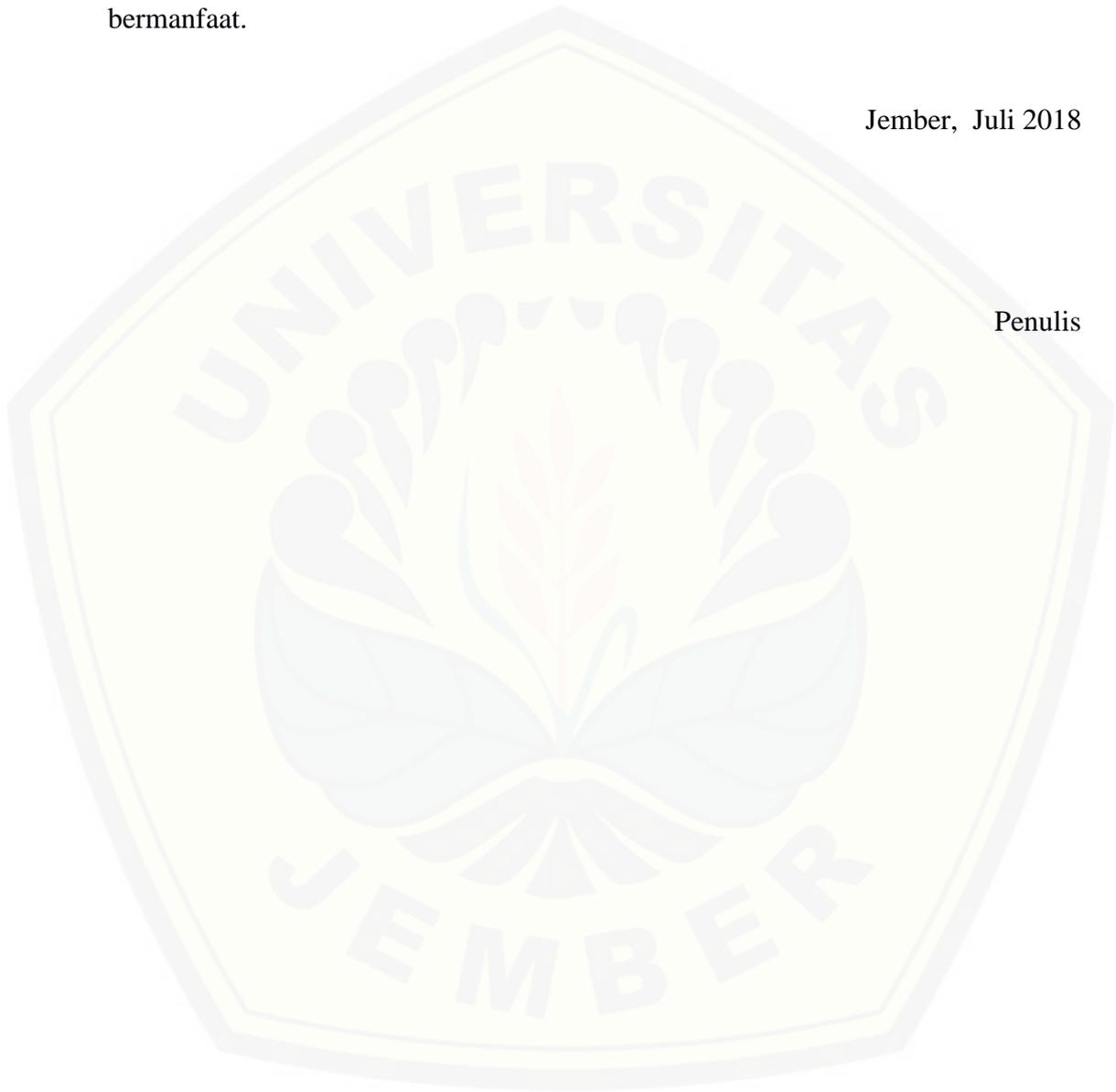
1. Prof. Drs. Slamin, M.Comp.Sc.,Ph.D selaku Dosen Pembimbing Utama dan Oktalia Juwita, S.Kom., M.MT selaku Dosen Pembimbing Anggota yang telah meluangkan waktu membantu dan memberi nasehat dalam menyelesaikan skripsi ini.
2. selaku Dosen Penguji I dan selaku Dosen Penguji II yang telah memberikan saran dan arahan dalam penulisan skripsi ini.
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan Program Studi Sistem Informasi Universitas Jember;
4. Ibunda Siti Cholifah, Ayahanda Achmad Junaidi, saudara perempuan Nadia Julfa Syabania dan keluarga besar yang selalu mendukung dan mendoakan.
5. Achmad Nur Fikri yang telah meluangkan waktu untuk menemani dan membantu melancarkan pengerjaan skripsi ini.
6. Para sahabat M. Bustommy Maulana, Sekar Arum Septiari, Angga Septiawan, Lathfiyya Jamhar, Obby Ilham Nugraha, Nanda Yudha Kawira, Ridlo Pamungkas, Selvia untuk bantuan, dukungan beserta doanya.
7. Teman-teman Intention angkatan 2013.

8. Semua pihak yang tidak dapat disebutkan satu persatu.

Dengan harapan bahwa penelitian ini nantinya akan terus berlanjut dan berkembang, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Juli 2018

Penulis



**DAFTAR ISI**

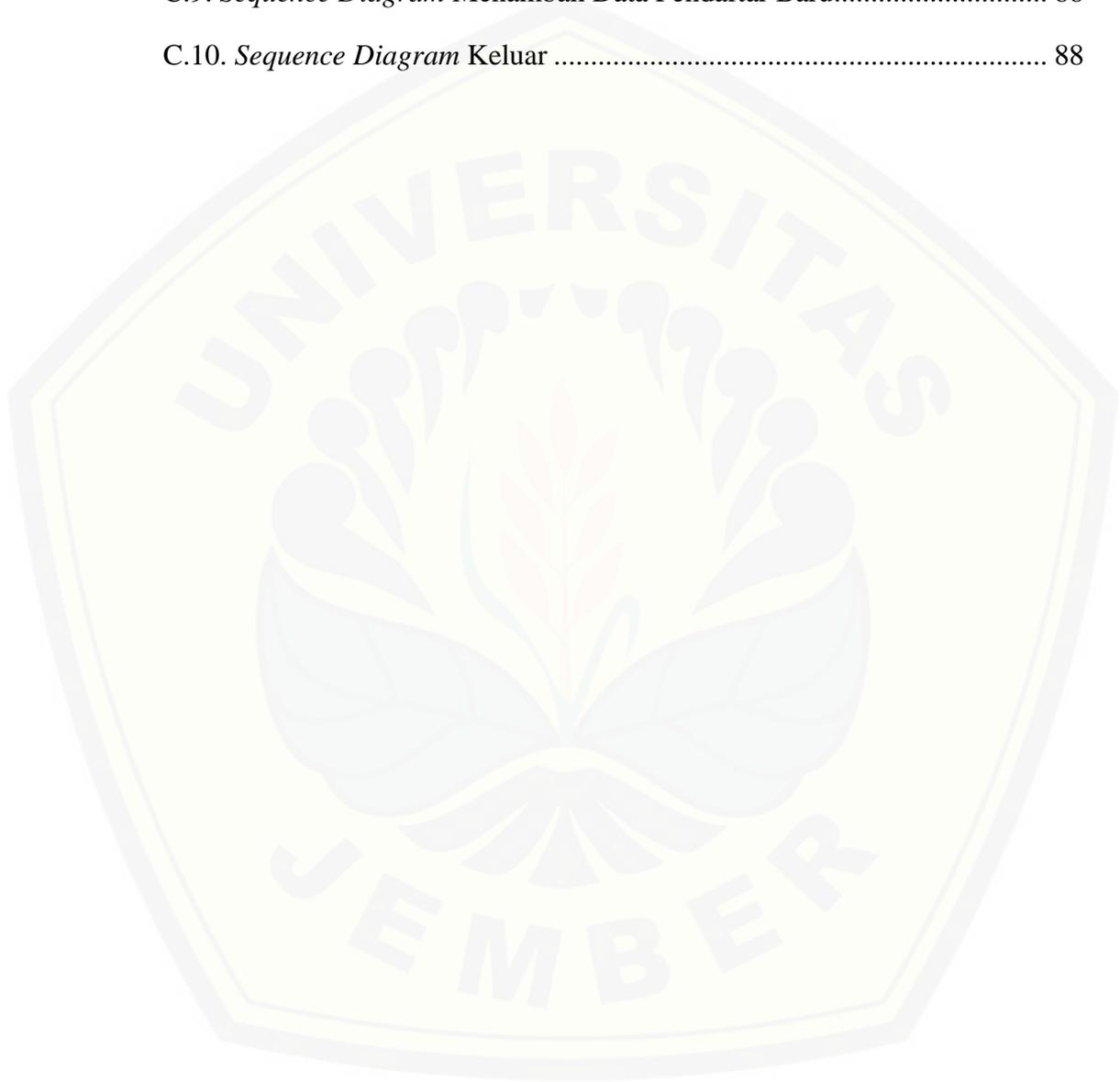
HALAMAN JUDUL.....	<b>Error! Bookmark not defined.</b>
PERSEMBAHAN .....	iii
MOTTO .....	iv
PERNYATAAN.....	v
SKRIPSI.....	vi
PENGESAHAN PEMBIMBING.....	vii
PENGESAHAN PENGUJI.....	viii
RINGKASAN .....	ix
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL.....	xviii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan .....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan .....	3
BAB 2 TINJAUAN PUSTAKA .....	5
2.1 Beasiswa.....	5
2.2 Sistem Pendukung Keputusan.....	5
2.3 Simple Additive Weighting (SAW) .....	6
BAB 3 METODOLOGI PENELITIAN.....	8
3.1 Jenis Penelitian.....	8
3.2 Pengembangan Sistem informasi .....	8
<b>3.2.1</b> Analisis Kebutuhan.....	9
<b>3.2.2</b> Analisis Data.....	10
<b>3.2.3</b> Desain Sistem Informasi .....	10
<b>3.2.4</b> Implementasi.....	11

<b>3.2.5</b> Pengujian dan Evaluasi.....	11
<b>3.2.6</b> Pemeliharaan.....	12
<b>BAB 4 PERANCANGAN SISTEM INFORMASI.....</b>	<b>15</b>
4.1 Deskripsi Umum Sistem Informasi Rekomendasi Mahasiswa Terhadap Penerimaan Beasiswa.....	15
<b>4.1.1</b> Fungsi Sistem.....	16
4.2 Analisis Kebutuhan Sistem Informasi.....	16
<b>4.2.1</b> Kebutuhan Fungsional .....	17
<b>4.2.2</b> Kebutuhan Non Fungsional .....	17
4.3 Desain Sistem Informasi .....	17
<b>4.3.1</b> <i>Business Process</i> .....	18
<b>4.3.2</b> <i>Usecase Diagram</i> .....	18
<b>4.3.3</b> Skenario .....	22
<b>4.3.4</b> <i>Sequence Diagram</i> .....	25
<b>4.3.5</b> <i>Activity Diagram</i> .....	29
<b>4.3.6</b> <i>Class Diagram</i> .....	32
<b>4.3.7</b> <i>Entity Relationship Diagram (ERD)</i> .....	32
4.4 Penulisan Kode Program.....	34
<b>4.4.1</b> Kode Program Masuk .....	34
<b>4.4.2</b> Kode Program Menambah Data Pendaftar Baru .....	34
<b>4.4.3</b> Kode Program Menambah Pendaftaran.....	34
<b>4.4.4</b> Kode Program Melihat Hasil Rekomendasi .....	34
<b>4.4.5</b> Kode Program Mengelola Data Beasiswa .....	34
<b>4.4.6</b> Kode Program Mengelola Data Lembaga .....	35
<b>4.4.7</b> Kode Program Melihat Data Pendaftaran.....	35
<b>4.4.8</b> Kode Program Mengelola Hasil Rekomendasi.....	35

4.4.9 Kode Program Mengelola Kriteria Pendaftaran .....	37
4.4.10 Kode Program Keluar .....	38
4.5 Pengujian Sistem informasi.....	38
4.5.1 Pengujian <i>Black Box</i> .....	38
BAB 5 HASIL DAN PEMBAHASAN.....	51
5.1 Hasil Implementasi <i>Coding</i> Pada Sistem Informasi Rekomendasi Mahasiswa Fakultas Ilmu Komputer Terhadap Penerimaan Beasiswa .....	51
5.1.1 Tampilan Masuk .....	51
5.1.2 Tampilan Menambah Data Pendaftar Baru .....	52
5.1.3 Tampilan Mengelola Data Pendaftaran .....	52
5.1.4 Tampilan Melihat Hasil Rekomendasi .....	53
5.1.5 Tampilan Melihat Data Pendaftar .....	54
5.1.6 Tampilan Mengelola Data Beasiswa .....	54
5.1.7 Tampilan Mengelola Hasil Rekomendasi.....	55
5.1.8 Tampilan Mengelola Data Lembaga.....	56
5.1.9 Tampilan Mengelola Kriteria Pendaftaran .....	56
5.2 Pengujian Sistem Informasi Rekomendasi Mahasiswa Fakultas Ilmu Komputer Terhadap Penerimaan Beasiswa Menggunakan Metode <i>Simple Additive Weighting</i> .....	57
5.2.1 Perhitungan Manual <i>Simple Additive Weighting</i> .....	57
5.3 Hasil Implementasi Metode <i>Simple Additive Weighting</i> Pada Sistem Informasi Rekomendasi Mahasiswa Terhadap Penerimaan Beasiswa.....	60
5.4 Pembahasan Sistem Informasi Peramalan Persediaan Bahan Baku .....	62
BAB 6 PENUTUP .....	63
6.1 Kesimpulan .....	63
6.2 Saran.....	63
DAFTAR PUSTAKA .....	65
LAMPIRAN .....	66

LAMPIRAN A. SKENARIO.....	66
A.1. Skenario Masuk .....	66
A.2. Skenario Mengelola Data Pendaftar Baru.....	68
A.3. Skenario Mengelola Pendaftaran .....	69
A.4. Skenario Mengelola Data Pendaftaran.....	70
A.5. Skenario Mengelola Data Lembaga.....	71
A.6. Skenario Mengelola Kriteria Pendaftaran.....	72
A.7. Skenario Mengelola Hasil Rekomendasi .....	72
A.8. Skenario Keluar .....	74
LAMPIRAN B. ACTIVITY DIAGRAM .....	75
B.1. <b>Activity</b> Diagram Masuk .....	75
B.2. <i>Activity Diagram</i> Menambah Data Pendaftar Baru .....	76
B.3. <i>Activity Diagram</i> Menambah Data Pendaftaran .....	77
B.4. <i>Activity Diagram</i> Melihat Hasil Rekomendasi .....	77
B.5. <i>Activity Diagram</i> Mengelola Data Lembaga.....	78
B.6. <i>Activity Diagram</i> Mengelola Kriteria Pendaftaran .....	79
B.7. <i>Activity Diagram</i> Melihat Data Pendaftar .....	79
B.8. <i>Activity Diagram</i> Mengelola Data Beasiswa .....	80
B.9. <i>Activity Diagram</i> Mengelola Hasil Rekomendasi.....	81
B.10. <i>Activity Diagram</i> Keluar .....	82
LAMPIRAN C. SEQUENCE DIAGRAM .....	83
C.1. <i>Sequence Diagram</i> Masuk.....	83
C.2. <i>Sequence Diagram</i> Menambah Data Pendaftaran.....	83
C.3. <i>Sequence Diagram</i> Melihat Hasil Rekomendasi.....	84
C.4. <i>Sequence Diagram</i> Mengelola Data Lembaga.....	84
C.5. <i>Sequence Diagram</i> Mengelola Kriteria Pendaftaran.....	85

C.6. <i>Sequence Diagram</i> Melihat Data Pendaftaran .....	85
C.7. <i>Sequence Diagram</i> Mengelola Data Beasiswa .....	86
C.8. <i>Sequence Diagram</i> Mengelola Hasil Rekomendasi .....	87
C.9. <i>Sequence Diagram</i> Menambah Data Pendaftar Baru.....	88
C.10. <i>Sequence Diagram</i> Keluar .....	88

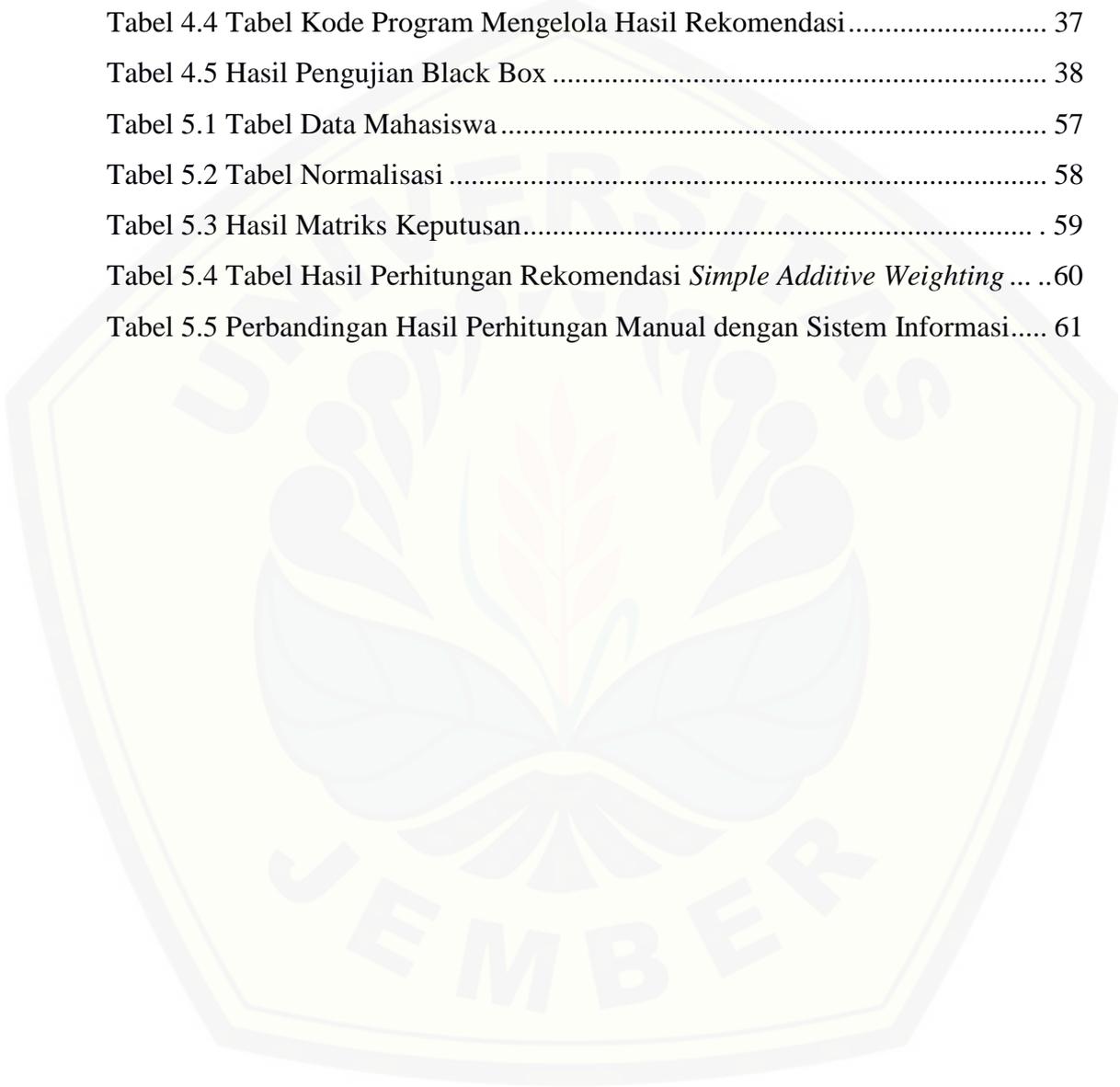


**DAFTAR GAMBAR**

Gambar 3. 1 Bagan Alur Penelitian .....	8
Gambar 3. 2 Model Waterfall (Sommerville, 2011) .....	9
Gambar 4. 1 Komponen Sistem Informasi .....	18
Gambar 4. 2 Usecase Diagram .....	19
Gambar 4. 3 Sequence Diagram Mengelola Hasil Rekomendasi .....	27
Gambar 4. 4 Activity Diagram mengelola hasil rekomendasi .....	31
Gambar 4. 5 Class Diagram .....	<b>Error! Bookmark not defined.</b>
Gambar 4. 6 Entity Relationship Diagram (ERD) .....	<b>Error! Bookmark not defined.</b>
Gambar 5.1 Tampilan Masuk.....	51
Gambar 5.2 Tampilan Menambah Data Pendaftar Baru .....	52
Gambar 5.3 Tampilan Menambah Data Pendaftaran .....	53
Gambar 5.4 Tampilan Melihat Hasil Rekomendasi .....	54
Gambar 5.5 Tampilan Melihat Data Pendaftar .....	54
Gambar 5.6 Tampilan Mengelola Data Beasiswa.....	55
Gambar 5.7 Tampilan Mengelola Hasil Rekomendasi .....	55
Gambar 5.8 Tampilan Mengelola Data Lembaga .....	56

**DAFTAR TABEL**

Tabel 4.1 Definisi Aktor .....	19
Tabel 4.2 Definisi <i>Usecase</i> .....	19
Tabel 4.3 Skenario Mengelola Hasil Rekomendasi.....	21
Tabel 4.4 Tabel Kode Program Mengelola Hasil Rekomendasi.....	37
Tabel 4.5 Hasil Pengujian Black Box .....	38
Tabel 5.1 Tabel Data Mahasiswa .....	57
Tabel 5.2 Tabel Normalisasi .....	58
Tabel 5.3 Hasil Matriks Keputusan.....	59
Tabel 5.4 Tabel Hasil Perhitungan Rekomendasi <i>Simple Additive Weighting</i> ...	60
Tabel 5.5 Perbandingan Hasil Perhitungan Manual dengan Sistem Informasi.....	61



## BAB 1 PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

### 1.1 Latar Belakang

Sehubungan dengan Undang-Undang Dasar 1945 pasal 31 bahwa tiap-tiap warga Negara berhak mendapatkan pengajaran. Berdasarkan pasal tersebut maka pemerintah daerah wajib memberikan layanan dan kemudahan, serta menjamin terselenggaranya pendidikan yang bermutu bagi setiap warga negara tanpa diskriminasi. Untuk menyelenggarakan pendidikan yang bermutu diperlukan biaya pendidikan yang cukup besar. Oleh karena itu bagi setiap peserta didik pada setiap satuan pendidikan berhak mendapatkan biaya pendidikan bagi mereka yang orang tuanya kurang mampu membiayai pendidikannya dan berhak mendapatkan beasiswa bagi mereka yang berprestasi.

Pemerintah melalui Direktorat Jenderal Pendidikan Tinggi Departemen Pendidikan Nasional berupaya mengalokasikan dana untuk memberikan beasiswa kepada mahasiswa yang secara ekonomi tidak mampu untuk membiayai pendidikannya dan memberikan beasiswa kepada mahasiswa yang mempunyai prestasi.

Beasiswa dapat diberikan oleh Lembaga Pemerintah, Perusahaan ataupun Yayasan. Pemberian bantuan belajar berupa beasiswa juga diberikan kepada mahasiswa di Universitas Jember khususnya Pogram Studi Sistem Informasi. Beasiswa yang diberikan antara lain beasiswa Peningkatan Prestasi Akademik (PPA), Beasiswa Djarum Fondation, Beasiswa Bank Indonesia, Beasiswa BCA Finance, Beasiswa Sampoerna Fondation, Beasiswa Bank Mandiri, Beasiswa PT Astra International.

Pentingnya beasiswa bagi penerimanya dan belum adanya alat bantu atau metode yang digunakan untuk menentukan penerima beasiswa, maka dibutuhkan sebuah sistem yang dapat membantu pembuat keputusan dalam memberikan

rekomendasi penerima beasiswa tersebut. Untuk menentukan rekomendasi mahasiswa terhadap penerima beasiswa, maka digunakan metode *Simple Additive Weighting*. Metode *Simple Additive Weighting* dipilih karena dapat menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif. Dengan metode ini didasarkan bobot yang sudah ditentukan sehingga mendapatkan hasil yang akurat terhadap rekomendasi mahasiswa untuk penerima beasiswa.

Penelitian yang dilakukan oleh Apriansyah Putra dari Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Sriwijaya tentang Penentuan Penerima Beasiswa Dengan Menggunakan *Fuzzy Multiple Attribute Decision Making*. Calon penerima beasiswa dipilih berdasarkan kriteria yang telah ditentukan oleh lembaga yang memberi beasiswa. Pemberian beasiswa dilakukan oleh beberapa lembaga untuk membantu seseorang yang kurang mampu maupun sebagai penghargaan bagi mahasiswa yang berprestasi. Penelitian dilakukan dengan mencari nilai bobot untuk setiap atribut. Kemudian dilakukan proses perankingan yang menentukan alternatif optimal, yaitu mahasiswa terbaik (Apriansyah, 2011).

Penelitian lainnya yaitu yang dilakukan oleh Heru Saputra dari STMIK Indonesia Padang. Proses seleksi beasiswa selama ini menggunakan cara manual untuk menentukan Perguruan Tinggi Swasta mana yang berhak menerima beasiswa sehingga pengelolaan data kurang efektif, membutuhkan waktu yang relative lama dan sering terjadi subjektifitas dari para pengelola beasiswa. Dalam penelitian ini sistem pendukung keputusan dengan metode *Simple Additive Weighting* dapat memberikan alternatif terbaik bagi pihak pengelola beasiswa dalam menentukan mahasiswa yang berhak menerima beasiswa sesuai kondisi yang diharapkan (Saputra, 2016).

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas, maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana mengimplementasikan metode *Simple Additive Weighting* dalam merekomendasi mahasiswa terhadap penerimaan beasiswa?
2. Bagaimana merancang dan membangun sistem informasi untuk rekomendasi mahasiswa menggunakan metode *Simple Additive Weighting*?

### 1.3 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Dapat merekomendasikan mahasiswa untuk penerimaan beasiswa dengan menggunakan metode *Simple Additive Weighting*.
2. Dapat merancang dan membangun sebuah sistem informasi penunjang pengambilan keputusan penerimaan beasiswa untuk mahasiswa berprestasi.

### 1.4 Batasan Masalah

Beberapa hal yang menjadi batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini berupa sistem pendukung keputusan berbasis web.
2. Objek pada penelitian ini yaitu mahasiswa dari Program Studi Sistem Informasi Universitas Jember.
3. Penelitian ini berfokus kepada rekomendasi mahasiswa untuk penerimaan beasiswa yang mendaftar.
4. Data yang digunakan pada penelitian ini yaitu data laporan mahasiswa yang mendaftar beasiswa.

### 1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

1. Pendahuluan

Bab Pendahuluan merupakan penjelasan latar belakang dimana alasan skripsi ini disusun, masalah yang terdapat pada latar belakang dijelaskan pada rumusan masalah, tujuan untuk menjawab rumusan masalah, batasan masalah penjelasan batasan-batasan pada skripsi, dan sistematika penulisan skripsi dijabarkan pada sub bab tersendiri pada bab satu ini.

## 2. Tinjauan Pustaka

Bab tinjauan pustaka merupakan bab yang menjelaskan teori-teori yang melandasi penelitian, tinjauan pustaka, dan studi terdahulu yang menjadi acuan dalam penelitian.

## 3. Metodologi Penelitian

Bab metodologi penelitian menyajikan pembahasan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

## 4. Pengembangan Sistem

Bab pengembangan sistem adalah bab yang berisi proses pengembangan sistem yang dibangun dalam penelitian. Adapun tahapan-tahapan pengembangan sistem ini didasarkan pada metodologi penelitian yang telah ditulis pada bab sebelumnya.

## 5. Hasil dan Pembahasan

Bab ini menjelaskan mengenai hasil penelitian yang telah dilakukan serta pembahasan sistem yang telah dibuat. Pembahasan dilakukan guna menjelaskan dan memaparkan bagaimana penelitian ini menjawab perumusan masalah serta tujuan dan manfaat dari penelitian ini seperti apa yang telah ditentukan pada awal penelitian.

## 6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

## BAB 2 TINJAUAN PUSTAKA

Bab ini menjelaskan tentang kajian-kajian teori terkait dengan penelitian yang dapat mendukung penelitian yang dilakukan.

### 2.1 Beasiswa

Beasiswa adalah penghasilan bagi yang menerimanya. Hal ini sesuai dengan ketentuan pasal 4 ayat (1) UU PPh/2000. Disebutkan pengertian penghasilan adalah tambahan kemampuan ekonomis dengan nama dan dalam bentuk apa pun yang diterima atau diperoleh dari sumber Indonesia atau luar Indonesia yang dapat digunakan untuk konsumsi atau menambah kekayaan Wajib Pajak (WP). Karena beasiswa bisa diartikan menambah kemampuan ekonomis bagi penerimanya, berarti beasiswa merupakan penghasilan (Tiyas, 2015)

Beasiswa adalah pemberian berupa bantuan keuangan yang diberikan kepada perorangan, mahasiswa atau pelajar yang digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan. Pemberian beasiswa dapat dikategorikan pada pemberian Cuma-Cuma ataupun pemberian dengan ikatan kerja setelah selesainya pendidikan (Murniasih, 2009).

Beasiswa yang ditawarkan kepada mahasiswa yang dalam seleksinya melibatkan kriteria-kriteria yang ditetapkan oleh lembaga beasiswa. Sebagian besar kriteria yang dibutuhkan merupakan surat atau bukti keterangan beprestasi, surat rekomendasi, dan sebagainya. Secara umum ada dua jenis beasiswa yaitu beasiswa untuk mahasiswa atau calon mahasiswa berprestasi dan beasiswa untuk mahasiswa yang kurang mampu (Eniryati, 2011).

### 2.2 Sistem Pendukung Keputusan

Sistem pendukung keputusan atau Decision Support System (DSS) adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan

dalam situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat. (Turban, Efraim, Aronson, & Jay, 2001).

Sistem pendukung keputusan bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik SPK sebagai sebuah sistem berbasis komputer yang membantu dalam proses pengambilan keputusan. SPK sebagai sistem informasi berbasis computer yang adaptif, interaktif, fleksibel, yang secara khusus dikembangkan untuk mendukung solusi dari permasalahan manajemen yang tidak terstruktur untuk meningkatkan kualitas pengambilan keputusan. Dengan demikian dapat ditarik satu definisi tentang sistem pengambilan keputusan yaitu sebuah sistem berbasis computer yang adaptif, fleksibel, dan interaktif yang digunakan untuk memecahkan masalah-masalah tidak terstruktur sehingga meningkatkan nilai keputusan yang diambil (Yohana, Rani, & Heni, 2010).

### 2.3 Simple Additive Weighting (SAW)

*Simple Additive Weighting* merupakan salah satu cara atau metode perhitungan penjumlahan terbobot. Konsep dasar dari metode *Simple Additive Weighting* ialah mencari penjumlahan terbobot yang didapat dari rating kinerja pada setiap di semua atribut (Fishburn, 1967). Dalam metode SAW ini, rating atau nilai tiap atribut haruslah sudah melewati proses normalisasi sebelumnya. Normalisasi matrik keputusan (X) ke suatu skala yang nantinya bias dilakukan proses perbandingan pada semua nilai yang ada. Formula atau rumus untuk menjalankan proses normalisasi ini adalah

Nilai preferensi terhadap tiap-tiap alternatif ( $V_i$ ) dihitung sebagai berikut:

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\max_i X_{ij}} & \rightarrow \text{Jika } j \text{ adalah attribute keuntungan (benefit)} \\ \frac{\min_i X_{ij}}{X_{ij}} & \rightarrow \text{Jika } j \text{ adalah attribute biaya (cost)} \end{cases}$$

Keterangan:

$V_i$  = ranking pada tiap-tiap alternatif

$W_j$  = nilai bobot dari tiap-tiap kriteria

$R_{ij}$  = rating kinerja yang ternormalisasi

Nilai ( $V_i$ ) yang lebih besar menandakan bahwa yang kemungkinan lebih terpilih adalah alternatif  $A_i$ .

Langkah-langkah penyelesaian *Simple Additive Weighting* sebagai berikut:

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

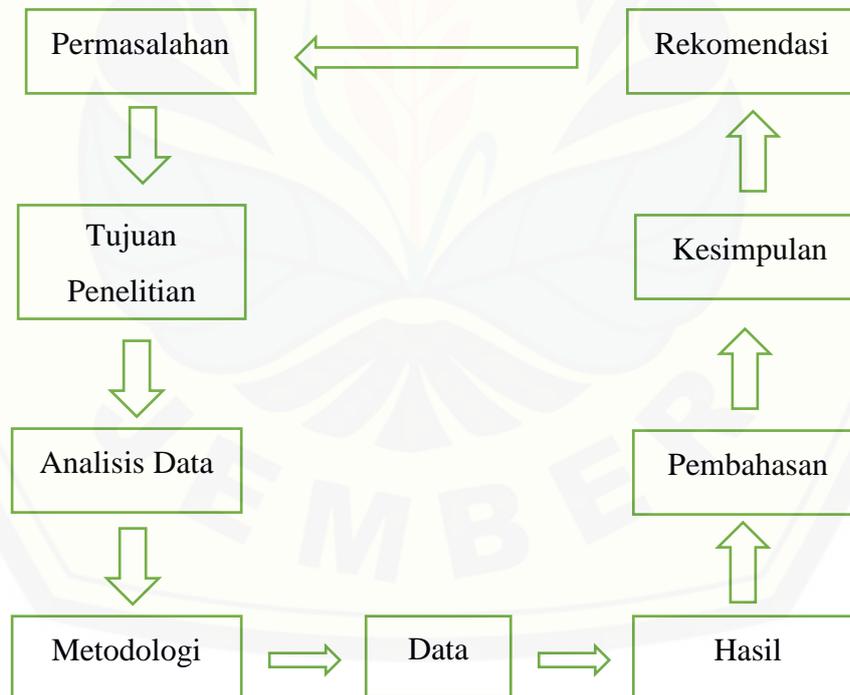
1. Menentukan kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu  $C_i$ .
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria  $C_i$ , kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut sehingga diperoleh matriks ternormalisasi  $R$ .
4. Hasil akhir diperoleh dari proses perangkingan yaitu penjumlahan dari perkalian matriks ternormalisasi  $R$  dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai terbaik ( $A_i$ ) sebagai solusi.

### BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tentang jenis penelitian, pengembangan sistem informasi dan pengujian metode *Simple Additive Weighting*.

#### 3.1 Jenis Penelitian

Jenis penelitian ini adalah penelitian pengembangan, karena ditujukan untuk sistem informasi rekomendasi mahasiswa terhadap penerimaan beasiswa. Penelitian pengembangan dilakukan untuk mempermudah proses rekomendasi mahasiswa yang dilakukan secara manual. Proses manual tersebut dikembangkan dengan proses yang lebih cepat dan akurat dengan menggunakan sistem informasi yang di sesuaikan, sehingga dapat menghasilkan keluaran yang akurat dan sesuai dengan hasil yang dikehendaki.

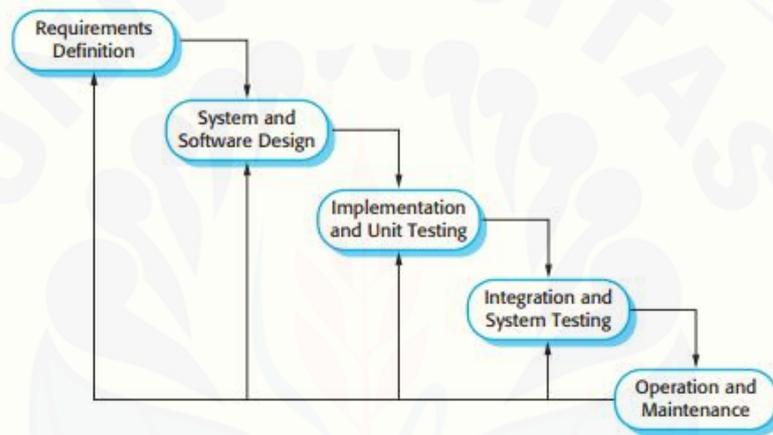


Gambar 3. 1 Bagan Alur Penelitian

#### 3.2 Pengembangan Sistem informasi

Pengembangan perangkat lunak pada penelitian ini dengan menggunakan model *Waterfall*. Model *System Development Life Cycle (SDLC) Waterfall*

digunakan karena keuntungannya, yaitu kualitas sistem informasi yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap, sehingga tidak berfokus pada suatu tahapan. Selain itu, bila kebutuhan sistem informasi dapat didefinisikan dengan baik maka pembangunan sistem informasi akan berjalan lancar. Menurut Pressman (2002) model *Waterfall* melakukan pendekatan pengembangan perangkat lunak yang sistem informasiatis dan sekuensial dimulai pada tingkat kemajuan sistem informasi sampai analisis, desain, kode, pengujian, dan pemeliharaan. Sebagaimana ditunjukkan pada Gambar 3.1, berikut:



Gambar 3. 2 Model Waterfall (Sommerville, 2011)

### 3.2.1 Analisis Kebutuhan

Tahap analisis merupakan langkah menganalisis kebutuhan-kebutuhan sistem informasi yang dibangun. Kebutuhan sistem informasi dibagi menjadi 2 yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan yang didapat berasal dari data yang telah terkumpulkan dan telah diolah yang berhubungan dengan perencanaan aktivitas distribusi pada perusahaan. Data-data tersebut kemudian dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non-fungsional.

Untuk menentukan kebutuhan fungsional dan non-fungsional, maka dilakukan pengumpulan data:

### 1. Wawancara

Wawancara merupakan cara pengumpulan data dengan narasumber dengan mengajukan pertanyaan kepada ketua kemahasiswaan. Wawancara terhadap narasumber bertujuan untuk memperoleh data yang dibutuhkan dalam penyelesaian penelitian.

### 2. Studi Pustaka

Studi Pustaka merupakan cara pengumpulan data dengan mencari informasi yang relevan dengan topik atau masalah yang menjadi obyek penelitian.

### 3.2.2 Analisis Data

Data yang dibutuhkan yaitu data mahasiswa yang pernah mendaftar beasiswa. Data tersebut akan digunakan dalam perhitungan pada rekomendasi untuk penentuan penerimaan beasiswa. Data tersebut akan diolah menggunakan metode *Simple Additive Weighthing*, sehingga informasi yang dihasilkan lebih baik. Selanjutnya sistem informasi akan mengeluarkan hasil dari rekomendasi beserta perhitungan kesalahan (*error*).

### 3.2.3 Desain Sistem Informasi

Desain sistem informasi merupakan tahap pembuatan desain sistem informasi, dimana pada penelitian ini menggunakan *Unified Modeling Language* (UML) yang dirancang dengan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang digunakan sebagai berikut:

#### 1. *Business Process*

*Business process* merupakan diagram yang menjelaskan tentang alur proses sistem informasi yang meliputi input, output dan goal dari tujuan sistem informasi yang dibangun.

#### 2. *Usecase Diagram*

*Usecase diagram* merupakan diagram yang menggambarkan fungsionalitas interaksi antara *user* dengan sistem informasi. *Usecase diagram* juga dapat menggambarkan hak akses dari *user*.

### 3. *Scenario*

*Scenario* digunakan untuk menjelaskan fitur yang ada pada *usecase diagram*.

### 4. *Sequence Diagram*

*Sequance Diagram* menjelaskan interaksi-interaksi objek didalam sistem informasi yang disusun pada sebuah runtutan atau rangkaian waktu.

### 5. *Activity Diagram*

*Activity diagram* merupakan sebuah runtutan aktivitas *user* dalam sistem informasi dan menjelaskan alur yang mungkin terjadi.

### 6. *Class Diagram*

*Class diagram* merupakan gambaran grafis dari struktur objek statis dari sebuah sistem informasi yang menunjukkan kelas objek yang tersusun dari hubungan antar kelas-kelas objek yang lain.

### 7. *Entity Relationship Diagram*

ERD merupakan suatu model yang menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang memiliki hubungan atau relasi.

## 3.2.4 Implementasi

Setelah desain sistem informasi telah selesai dibuat, maka selanjutnya dilakukan tahap implementasi. Perancangan sistem informasi menggunakan *framework* Laravel 5.4 yang merupakan bahasa pemrograman *Page Hypertext Pre-Processor* (PHP). *Tools* yang digunakan yaitu *Sublime* sebagai editor teks dan *MariaDB* sebagai *database management system*.

## 3.2.5 Pengujian dan Evaluasi

Pengujian dan evaluasi digunakan untuk mengetahui sejauh mana sistem informasi ini dapat berjalan. *Testing* berfungsi untuk mengetahui apakah sistem informasi ini dapat berfungsi dengan baik sesuai dengan yang diharapkan. Serta untuk mengetahui letak kekurangan yang ada pada sistem informasi. Pengujian dilakukan oleh tim penguji dari *developer*. Selanjutnya dilakukan evaluasi serta perbaikan terhadap kekurangan-kekurangan yang ada pada sistem informasi ini. Pengujian yang dilakukan pada sistem informasi ini yaitu:

### 1. *Black Box Testing*

*Black Box Testing* merupakan cara pengujian perangkat lunak yang memeriksa fungsionalitas dari sistem informasi yang berkaitan dengan struktur internal kerja. Pengetahuan khusus dari kode sistem informasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari software. (Agissa, 2013)

Pada pengujian *black box* ini, sistem informasi yang dibangun pada penelitian ini akan diuji dengan mengujikan langsung *running* sistem informasi dan melakukan kegiatan pengujian dengan menganalisis proses *input* dan *output* yang dihasilkan sistem informasi. Dalam metode *black box* juga dilakukan pengujian dengan cara memasukkan data normal dan data salah, dari pengimputan ini nantinya akan dilakukan analisis terdapat reaksi yang muncul pada sistem informasi.

### 3.2.6 Pemeliharaan

Pemeliharaan dilakukan setelah sistem informasi melalui tahap pengujian sistem informasi. Tahapan ini dilakukan untuk mengatasi masalah pada sistem informasi di saat penelitian. Selama *user* menemui *bug* pada sistem informasi, maka *user* langsung konfirmasi kepada peneliti untuk segera ditangani.

## 3.3 Analisis Tahap Penerapan Metode

Penelitian ini menghasilkan keakuratan rekomendasi mahasiswa terhadap penerimaan beasiswa yang dihitung dengan menerapkan metode *Simple Additive Weigthing* yang dihitung dengan pengambilan data pada mahasiswa yang mendaftar. Metode ini memberikan pembobotan penjumlahan dari rating kinerja pada setiap alternatif pada semua atribut. Penerapan metode *Simple Additive Weigthing* pada rekomendasi mahasiswa terhadap penerimaan beasiswa dapat dilihat pada Gambar 3.3

Berikut penerapan metode *Simple Additive Weigthing*

1) Menentukan Kriteria beasiswa ( $C_i$ ):

$C_1$  = Semester Aktif Perkuliahan

SMT 1-2 (4)    SMT 5-6 (2)

SMT 3-4 (3)    SMT 7-8 (1)

C2 = IPK

3.00 - 3.25 (1) 3.51 - 3.75 (3)

3.26 - 3.50 (2) 3.76 - 4.00 (4)

C3 = Penghasilan Orangtua

500 - 1.000 (1) 1.500 - 2.000 (3) 2.500 - 3.000 (5)

1.000 - 1.500 (2) 2.000 - 2.500 (4)

C4 = Aktif Prestasi

C5 = Aktif Berorganisasi

JURUSAN (1)

JURUSAN (1)

FAKULTAS (2)

FAKULTAS (2)

2) Memasukkan data dari tiap alternatif

NO	NAMA	C1	C2	C3	C4	C5
1	Raka	4	4	1.000.000	JURUSAN	FAKULTAS
2	Nindy	3	3.5	1.575.000	JURUSAN	JURUSAN
3	Fikri	4	3.35	2.500.000	FAKULTAS	JURUSAN

NO	NAMA	C1	C2	C3	C4	C5
1	Raka	4	4	5	1	2
2	Nindy	3	2	3	1	1
3	Fikri	3	2	4	2	1

3) Membuat matriks keputusan berdasarkan kriteria  $C_i$ , kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut sehingga diperoleh matriks ternormalisasi  $R$

4	4	5	1	2
3	2	3	1	1
3	2	4	2	1

R11	R12	R13	R14	R15
R21	R22	R23	R24	R25
R31	R23	R33	R34	R35

4) Hasil matriks di dapat berdasarkan pembagian dari nilai tengah

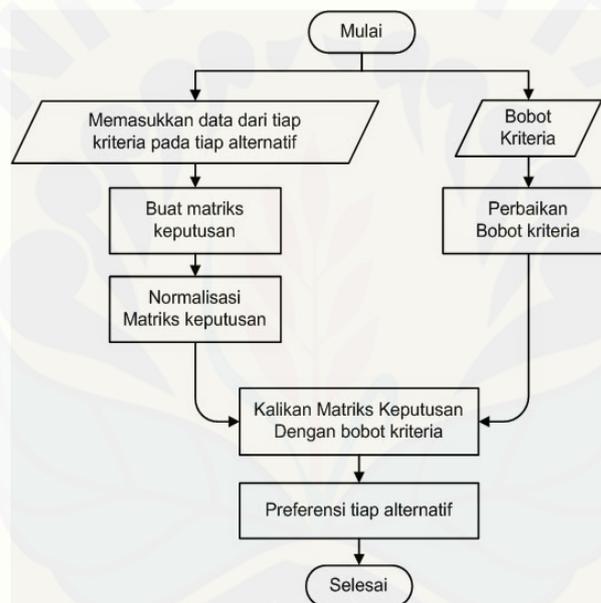
$R_{11}=4/3=1.3$	$R_{12}=4/2=2$	$R_{13}=5/3=1.6$	$R_{14}=1/1=1$	$R_{15}=2/1=2$
$R_{21}=3/3=1$	$R_{22}=2/2=1$	$R_{23}=3/3=1$	$R_{24}=1/1=1$	$R_{25}=1/1=1$
$R_{31}=3/3=1$	$R_{23}=2/2=1$	$R_{33}=4/3=1.3$	$R_{34}=2/1=2$	$R_{35}=1/1=1$
1.3	2	1.6	1	2
1	1	1	1	1
1	1	1.3	2	1

5) Hasil akhir diperoleh dari proses perangkingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar.

$$V1 = 2(1.3) + 4(2) + 6(1.6) + 8(1) + 3(2) = 34.2$$

$$V2 = 2(1) + 4(1) + 6(1) + 8(1) + 3(1) = 23$$

$$V3 = 2(1) + 4(1) + 6(1.3) + 8(2) + 3(1) = 32.8$$



Gambar 3.3 Flowchart Penerapan Metode *Simple Additive Weighing*

## BAB 4 PERANCANGAN SISTEM INFORMASI

Bab ini akan menjelaskan tentang perancangan sistem informasi rekomendasi pemilihan beasiswa pada mahasiswa Program Studi Sistem Informasi menggunakan metode *Simple Additive Weigthing*. Perancangan sistem informasi terdiri dari analisis kebutuhan sistem informasi, desain sistem informasi, penulisan kode program, pengujian, dan pemeliharaan.

### 4.1 Deskripsi Umum Sistem Informasi Rekomndasi Mahasiswa Terhadap Penerimaan Beasiswa

Sistem informasi rekomendasi pengambilan keputusan Mahasiswa Fakultas Ilmu Komputer jurusan Program Studi Sistem Informasi terhadap penerimaan beasiswa merupakan sistem informasi berbasis *web* yang mampu melakukan rekomendasi mahasiswa pada beasiswa menggunakan metode *Simple Additive Weigthing*. Data yang dibutuhkan dalam sistem informasi ini yaitu data kriteria beasiswa, data pendaftar, dan data setiap lembaga beasiswa. Beberapa data kriteria yang digunakan dalam sistem informasi ini adalah jumlah prestasi, jumlah sertifikat organisasi dan hasil nilai Indeks Prestasi Kumulatif (IPK).

Sistem informasi ini memiliki tiga hak akses yaitu Pendaftar, Admin Sistem dan Admin Lembaga (Super Admin). Setiap masing-masing hak akses memiliki peran yang berbeda dalam menggunakan sistem. Sistem dapat melakukan rekomendasi dengan menggunakan metode *Simple Additive Weigthing* dimana pendaftar memasukkan data diri dan Admin Sistem mengelola informasi dari admin lembaga. Admin Lembaga memasukkan data pemberi beasiswa yang selanjutnya sistem akan melakukan rekomendasi pengambilan keputusan.

Sistem akan melakukan perhitungan terhadap nilai kriteria dari data pendaftar. Kemudian sistem akan menampilkan hasil rekomendasi perhitungan metode *Simple Additive Weigthing* yang hasil akhirnya di sortir dari nilai tertinggi. Selanjutnya Lembaga melakukan penetapan rekomendasi beasiswa di Fakultas Ilmu Komputer dengan mengacu pada hasil perhitungan menggunakan metode *Simple Additive*

*Weigthing*. Penetapan rekomendasi mahasiswa terhadap penerimaan beasiswa merupakan tahapan terakhir dalam sistem ini.

#### 4.1.1 Fungsi Sistem

Fungsi utama dari Sistem Informasi Rekomendasi Beasiswa adalah dapat melakukan perhitungan menggunakan metode *Simple Additive Weigthing* (SAW) yang digunakan untuk mendukung dalam rekomendasi mahasiswa terhadap penerimaan beasiswa. Pengguna sistem ini adalah mahasiswa pendaftar beasiswa di Sistem Informasi. Sistem ini memiliki 3 hak akses yaitu Admin Lembaga, Admin Sistem, dan Pendaftar.

a. Admin Lembaga (Super Admin)

Admin Lembaga merupakan pengguna sistem yang memiliki hak akses untuk mengelola data rekomendasi mahasiswa terhadap beasiswa dan pemberi beasiswa. Tugas dari admin lembaga adalah mengelola data Lembaga dan mengelola kriteria pendaftaran. Admin lembaga juga mampu melihat data pendaftaran yang diinputkan oleh pendaftar beasiswa.

b. Admin Sistem

Admin Sistem merupakan pengguna sistem yang memiliki hak akses untuk melihat data pendaftar, mengelola data beasiswa, mengelola hasil rekomendasi dan mengelola data kriteria yang ditentukan oleh admin lembaga.

c. Pendaftar (User)

Pendaftar merupakan pengguna sistem yang memiliki hak akses untuk mengelola data profil miliknya. User merupakan pengguna sistem yang memiliki hak akses untuk menambah data pendaftar baru, menambah pendaftaran dan melihat hasil rekomendasi yang ditentukan oleh admin sistem.

#### 4.2 Analisis Kebutuhan Sistem Informasi

Analisis kebutuhan sistem menjelaskan tentang apa saja kebutuhan yang diperlukan dalam pengembangan sistem. Baik kebutuhan fungsional dan kebutuhan non fungsional.

#### 4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi fitur-fitur yang harus ada di dalam sistem sesuai dengan tuntutan kebutuhan pengguna. Berikut adalah kebutuhan fungsional dan sistem informasi rekomendasi beasiswa menggunakan *Simple Additive Weighthing* antara lain:

1. Sistem memiliki proses login dan logout untuk membedakan hak akses sistem
2. Sistem mampu mengolah data pendaftaran rekomendasi beasiswa yang meliputi *view, insert, edit dan delete*
3. Sistem mampu melakukan perhitungan dengan metode *Simple Additive Weighthing*
4. Sistem mampu menampilkan hasil perhitungan dengan *metode Simple Additive Weighthing*

#### 4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan fitur-fitur pendukung dalam sistem yang dapat mendukung sistem mencapai tujuan. Berikut adalah kebutuhan non fungsional dari sistem informasi rekomendasi beasiswa menggunakan *Simple Additive Weighthing* antara lain :

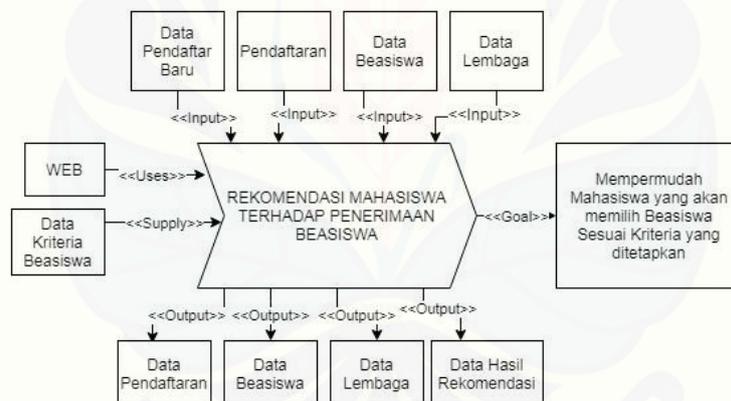
1. Sistem informasi menggunakan autentikasi yang menggunakan *username* dan *password*.
2. User interface yang *user-friendly* sehingga memudahkan pengguna dalam menggunakan sistem informasi.
3. Sistem berbasis *website*
4. Sistem informasi dapat diakses 24 jam.

#### 4.3 Desain Sistem Informasi

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem informasi yaitu tahap perencanaan pembangunan sistem informasi yang dapat digambarkan dengan desain sistem informasi. Desain sistem informasi ini meliputi *business process, usecase diagram, skenario, activity diagram, sequence diagram, class diagram, dan entity relationship diagram (ERD)*.

### 4.3.1 Business Process

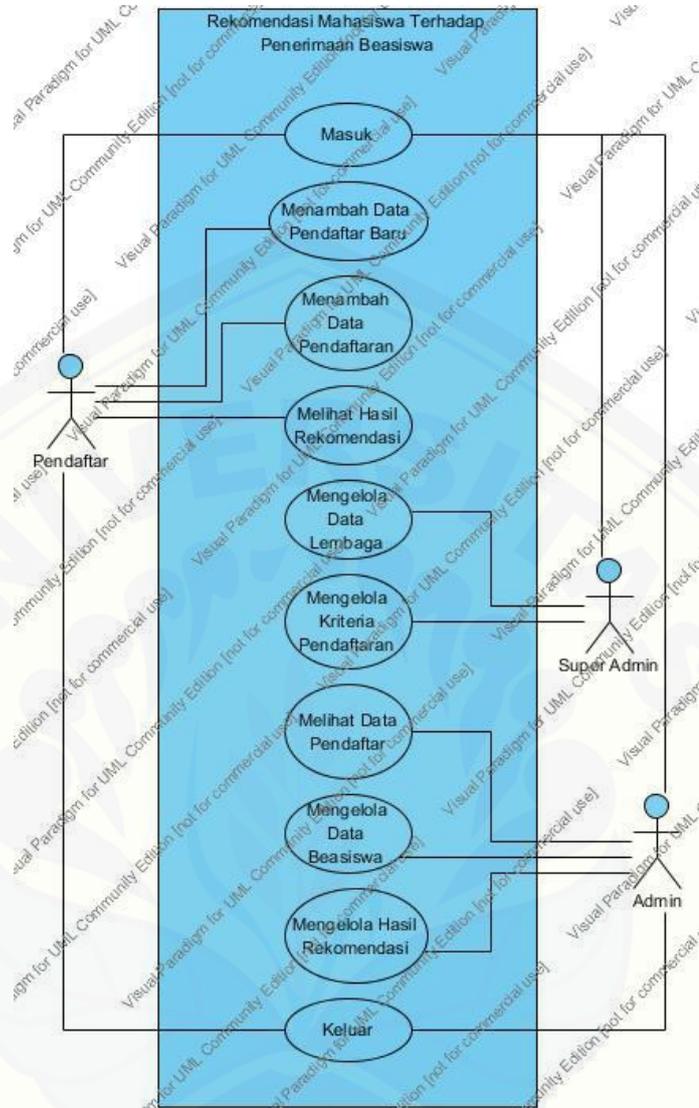
*Business Process* merupakan model yang menggambarkan proses yang terjadi dalam sistem informasi mulai dari *input*, *output* dan *goal* dari sistem informasi yang dibuat. Sistem informasi pengambilan keputusan untuk merekomendasikan beasiswa ini menggunakan platform *website* dan metode *Simple Additive Weighthing* dalam melakukan rekomendasi beasiswa. Sistem ini menggunakan data pendaftar baru, pendaftaran, data beasiswa dan data lembaga sebagai *input* ke sistem informasi. Input tersebut nantinya akan menghasilkan *output* berupa data pendaftaran, data beasiswa, data lembaga dan data hasil rekomendasi, sedangkan *goal* dari sistem informasi ini adalah mempermudah mahasiswa yang akan memilih beasiswa sesuai kriteria yang ditetapkan. *Business Process* dari sistem informasi pengambilan keputusan untuk rekomendas beasiswa ini dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Komponen Sistem Informasi

### 4.3.2 Usecase Diagram

*Usecase Diagram* merupakan diagram yang menggambarkan interaksi antara sistem informasi dan actor sistem informasi. *Usecase Diagram* hanya memberi gambaran singkat hubungan antara *usecase*, aktor dan sistem informasi. Dari *Usecase Diagram* ini dapat diketahui fitur-fitur apa saja yang dimiliki oleh sistem informasi yang dibuat. *Usecase Diagram* pada sistem informasi ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Usecase Diagram*

Penjelasan tentang definisi aktor dan definisi *usecase diagram* pada gambar akan dijelaskan di bawah ini.

#### 1. Definisi aktor

Definisi aktor yaitu penjelasan mengenai aktor-aktor sebagai pengguna dari sistem informasi rekomendasi beasiswa pada beasiswa. Terdapat 3 aktor seperti yang dijelaskan pada tabel 4.1

Tabel 4.1 Definisi Aktor

No.	Aktor	Deskripsi
1.	Pendaftar	Aktor Pendaftar pada sistem informasi ini memiliki hak akses untuk: <ol style="list-style-type: none"> <li>1. Menambah Data Pendaftaran</li> <li>2. Menambah Data Pengguna Baru</li> <li>3. Melihat Hasil Rekomendasi</li> </ol>
2.	Admin Sistem	Aktor Admin Sistem pada sistem informasi ini memiliki peran untuk: <ol style="list-style-type: none"> <li>1. Mengelola Data Beasiswa</li> <li>2. Melihat Data Pendaftaran</li> <li>3. Mengelola Hasil Rekomendasi</li> </ol>
3.	Admin Lembaga	Aktor Admin Lembaga pada sistem informasi ini memiliki peran untuk: <ol style="list-style-type: none"> <li>1. Mengelola Data Lembaga</li> <li>2. Mengelola Kriteria Pendaftaran</li> </ol>

Definisi *usecase* yaitu penjelasan tentang fitur-fitur yang tersedia dalam sistem informasi rekomendasi beasiswa pada beasiswa. Terdapat 10 *usecase* pada table 4.2

Table 4.2 Definisi *Usecase*

No.	<i>Usecase</i>	Deskripsi
1.	Masuk	Menggambarkan proses autentifikasi pengguna yang digunakan untuk masuk ke sistem informasi.
2.	Menambah Data Pendaftar Baru	Menggambarkan proses menambah data pendaftar baru yang meliputi melihat, menambah, mengubah dan menghapus data pendaftar baru.

3.	Menambah Data Pendaftaran	Menggambarkan proses menambah data pendaftaran untuk merekomendasikan mahasiswa terhadap beasiswa.
4.	Melihat Hasil Rekomendasi	Menggambarkan proses melihat hasil data rekomendasi.
5.	Melihat Data Pendaftar	Menggambarkan proses melihat data pendaftar
6.	Mengelola Data Beasiswa	Menggambarkan proses mengelola data beasiswa yang meliputi melihat, menambah, mengubah dan menghapus data beasiswa.
7.	Mengelola Hasil Rekomendasi	Menggambarkan proses mengelola data hasil rekomendasi yang meliputi melihat, menambah, mengubah dan menghapus data hasil rekomendasi.
9.	Mengelola Data Lembaga	Menggambarkan proses mengelola data Lembaga yang meliputi melihat, menambah, mengubah dan menghapus data Lembaga.
10.	Mengelola Kriteria Pendaftaran	Menggambarkan proses mengelola kriteria pendaftaran yang meliputi melihat, menambah, mengubah dan menghapus data kriteria pendaftaran.
11.	Keluar	Menggambarkan proses <i>user</i> untuk keluar dari sistem informasi.

### 4.3.3 Skenario

Skenario merupakan penjabaran alur kerja sistem informasi yang terdapat pada *usecase diagram*. Skenario sistem informasi rekomendasi pada mahasiswa adalah sebagai berikut.

#### 4.3.3.1 Skenario Masuk

Skenario masuk merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan masuk ke dalam sistem informasi. Hanya *user* terdaftar saja yang dapat mengakses sistem informasi. Terdapat 3 *user* dalam sistem informasi ini yaitu Pendaftaran, Admin Sistem dan Admin Lembaga. Sistem informasi memiliki reaksi yang berbeda untuk masing-masing aktor sehingga aktor dapat masuk ke sistem informasi dan menggunakannya. Skenario masuk dapat dilihat pada lampiran A Tabel 1.

#### 4.3.3.2 Skenario Menambah Data Pendaftar Baru

Skenario mengelola data pengguna baru merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor mengelola data pengguna baru. Aktor dapat mengelola data pengguna baru. Skenario mengelola data pengguna baru dapat dilihat pada lampiran A Tabel 2.

#### 4.3.3.3 Skenario Mengelola Data Hasil Rekomendasi

Skenario mengelola data hasil rekomendasi merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor mengelola data hasil rekomendasi. Data hasil rekomendasi yang dilihat merupakan data rekomendasi *beasiswa*. Skenario mengelola data hasil rekomendasi dapat dilihat pada lampiran A Tabel 3.

Tabel 4.3 Skenario Mengelola Hasil Rekomendasi

Nomor Usecase	USC-010
Nama	Mengelola Hasil Rekomendasi
Aktor	Admin
Deskripsi Singkat	Aktor mengelola hasil rekomendasi
Prekondisi	Aktor memilih Menu Beasiswa



	6. Menampilkan spam “ <i>These credentials do not match our records</i> ”
--	---

#### 4.3.3.4 Skenario Mengelola Data Pendaftaran

Skenario mengelola data pendaftaran merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor menambah data pendaftaran atau pengguna baru. Aktor dapat mengelola data pendaftaran. Skenario mengelola data pendaftaran dapat dilihat pada lampiran A Tabel 4.

#### 4.3.3.5 Skenario Mengelola Data Beasiswa

Skenario mengelola data beasiswa merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor menambah data pendaftaran atau pengguna baru. Aktor dapat mengelola data pendaftaran. Skenario mengelola data pendaftaran dapat dilihat pada lampiran A Tabel 5

#### 4.3.3.6 Skenario Kriteria Pendaftaran

Skenario mengelola kriteria pendaftaran merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor mengelola kriteria pendaftaran. Aktor dapat melihat, menambah, mengubah, dan menghapus informasi. Skenario mengelola kriteria pendaftaran dapat dilihat pada lampiran A Tabel 6.

#### 4.3.3.7 Skenario Mengelola Data Lembaga

Skenario mengelola data lembaga merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor mengelola data lembaga. Aktor dapat melihat, menambah, mengubah, dan menghapus informasi. Skenario mengelola data lembaga dapat dilihat pada lampiran A Tabel 7.

#### 4.3.3.8 Skenario Melihat Data Pendaftaran

Skenario melihat data pendaftaran merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor melihat data pendaftaran. Skenario melihat data pendaftaran dapat dilihat pada lampiran A Tabel 8.

#### 4.3.3.9 Skenario Melihat Data Hasil Rekomendasi

Skenario melihat data hasil rekomendasi merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor melihat data hasil rekomendasi. Skenario melihat data hasil rekomendasi dapat dilihat pada lampiran A Tabel 9.

#### 4.3.3.10 Skenario Keluar

Skenario keluar merupakan alur yang menjelaskan aksi actor dan reaksi system informasi pada saat actor melakukan keluar. Scenario keluar dapat dilihat pada lampiran A Tabel 10.

### 4.3.4 *Sequence Diagram*

*Sequence diagram* merupakan diagram yang menggambarkan interaksi antar kelas yang dilakukan sesuai dengan alur masing-masing. *Sequence diagram* merupakan desain perancangan yang mendekati proses pengkodean, maka dari itu di dalam *sequence diagram* bahasa yang dituliskan merupakan bahasa pengkodean, dimana bahasa pengkodean tersebut menjelaskan jalannya alur dalam fitur secara berurutan. *Sequence diagram* dalam sistem informasi ini antara lain sebagai berikut.

#### 4.3.4.1 *Sequence Diagram* Masuk

Pada *sequence diagram* masuk terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling terhubung adalah kelas tampilan antara tiga pengguna yaitu Pendaftar, Admin Sistem dan Admin Lembaga. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Penggambaran *sequence diagram* masuk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang dapat dilihat pada lampiran B Gambar 1.

#### 4.3.4.2 *Sequence Diagram* Menambah Data Pendaftar Baru

Pada *sequence diagram* mengelola data pendaftar baru merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses yang digunakan. Kelas yang saling terhubung adalah kelas tampilan antara tiga pengguna yaitu pendaftar, admin sistem dan admin lembaga.

Masing-masing kelas akan ditampilkan secara visual dengan gambar. Kelas yang berperan dalam *sequence diagram* mengelola data pemberi beasiswa adalah *view*, *home*, *controller home*, *view index*, *blade*, *controller User Controller*, dan *model Users*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 2.

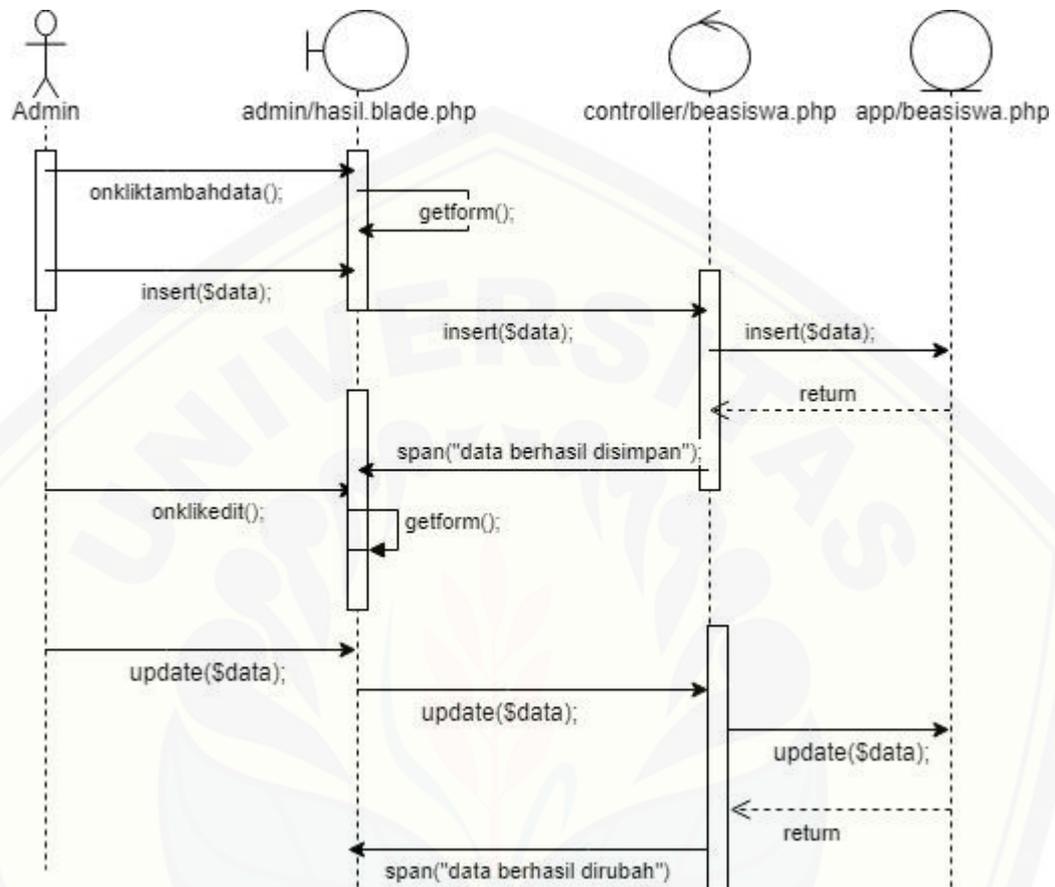
#### 4.3.4.3 *Sequence Diagram* Menambah Pendaftaran

Pada *Sequence Diagram* mengelola pendaftaran terdapat beberapa interaksi antar dua pengguna yaitu admin lembaga dan admin sistem. Kelas yang berperan dalam *sequence diagram* mengelola data penanggung jawab adalah *view*, *home*, *controller home*, *view index*, *blade*, *controller User Controller*, dan *model Users*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 3.

#### 4.3.4.4 *Sequence Diagram* Mengelola Hasil Rekomendasi

Pada *sequence diagram* mengelola hasil rekomendasi merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses mengelola hasil peramalan persediaan bahan baku bulan selanjutnya, terdapat beberapa interaksi antar kelas yang digunakan. Masing – masing kelas akan ditampilkan secara visual dengan gambar. Kelas yang berperan dalam *sequence diagram* mengelola hasil peramalan persediaan bahan baku bulan selanjutnya adalah *view home*, *controller home*, *view index*, *blade*, *controller ForecastController*, dan *model Forecast*. Proses dari aliran *sequence diagram*

diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller*



Gambar 4. 3 Sequence Diagram Mengelola Hasil Rekomendasi

dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada Gambar 4.3.

#### 4.3.4.5 Sequence Diagram Mengelola Data Pendaftaran

Pada *sequence diagram* mengelola data pendaftaran merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses mengelola hasil peramalan persediaan bahan baku bulan selanjutnya, terdapat beberapa interaksi antar kelas yang digunakan. Masing – masing kelas akan ditampilkan secara visual dengan gambar. Kelas yang berperan

dalam *sequence diagram* mengelola hasil peramalan persediaan bahan baku bulan selanjutnya adalah *view home*, *controller home*, *view index. blade*, *controller ForecastController*, dan *model Forecast*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 5.

#### 4.3.4.6 *Sequence Diagram* Mengelola Data Beasiswa

Pada *sequence diagram* mengelola data beasiswa merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses mengelola hasil peramalan persediaan bahan baku bulan selanjutnya, terdapat beberapa interaksi antar kelas yang digunakan. Masing – masing kelas akan ditampilkan secara visual dengan gambar. Kelas yang berperan dalam *sequence diagram* mengelola hasil peramalan persediaan bahan baku bulan selanjutnya adalah *view home*, *controller home*, *view index. blade*, *controller ForecastController*, dan *model Forecast*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 6.

#### 4.3.4.7 *Sequence Diagram* Mengelola Data Lembaga

Pada *sequence diagram* mengelola data lembaga terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat laporan penggunaan bahan baku adalah *view home*, *controller home*, *view index. blade*, *controller SalesController*, dan *model Sales*. Interaksi antar kelas dan

*function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 7.

#### 4.3.4.8 *Sequence Diagram* Mengelola Kriteria Pendaftaran

Pada *sequence diagram* mengelola kriteria pendaftaran terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat laporan penggunaan bahan baku adalah *view home*, *controller home*, *view index.blade*, *controller SalesController*, dan *model Sales*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 8.

#### 4.3.4.9 *Sequence Diagram* Melihat Data Hasil Rekomendasi

Pada *sequence diagram* melihat data hasil rekomendasi terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat laporan penggunaan bahan baku adalah *view home*, *controller home*, *view index.blade*, *controller SalesController*, dan *model Sales*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 9.

#### 4.3.4.10 *Sequence Diagram* Keluar

Pada *sequence diagram* keluar merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses keluar dari sistem. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 10.

### 4.3.5 *Activity Diagram*

*Activity diagram* merupakan diagram yang memodelkan proses-proses yang terjadi pada sistem informasi.

#### 4.3.5.1 *Activity Diagram* Masuk.

*Activity diagram* masuk menggambarkan tentang proses yang terjadi saat aktor melakukan masuk. Aktor memasukkan *username* dan *password* lalu mengklik tombol “masuk”. Kemudian sistem informasi akan melakukan pengecekan apakah inputan kosong atau *username* dan *password* salah, jika benar maka sistem informasi akan menampilkan halaman form Masuk. Namun jika tidak, maka sistem

informasi akan menampilkan halaman *dashboard* yang disesuaikan dengan hak akses *user* tersebut. *Activity diagram* masuk dapat dilihat pada lampiran C Gambar 1.

#### 4.3.5.2 *Activity Diagram* Menambah Data Pendaftar Baru

*Activity diagram* mengelola pendaftaran menggambarkan tentang proses yang terjadi saat aktor mengelola data pendaftaran. Aktor yang dapat mengelola data pendaftaran adalah user (pendaftar). Pendaftar dapat melihat, menambah, mengubah, dan menghapus data pendaftaran. *Activity diagram* mengelola data pendaftaran dapat dilihat pada lampiran C Gambar 3.

#### 4.3.5.3 *Activity Diagram* Menambah Pendaftaran

*Activity diagram* mengelola pendaftaran menggambarkan tentang proses yang terjadi saat aktor mengelola data pendaftaran. Aktor yang dapat mengelola data pendaftaran adalah user (pendaftar). Pendaftar dapat melihat, menambah, mengubah, dan menghapus data pendaftaran. *Activity diagram* mengelola data pendaftaran dapat dilihat pada lampiran C Gambar 3.

#### 4.3.5.4 *Activity Diagram* Melihat Hasil Rekomendasi

*Activity diagram* melihat hasil rekomendasi menggambarkan tentang proses yang terjadi saat aktor melihat hasil rekomendasi. Aktor yang dapat melihat hasil rekomendasi adalah pendaftar. Pendaftar dapat melihat data saja. *Activity diagram* melihat hasil rekomendasi dapat dilihat pada lampiran C Gambar 4.

#### 4.3.5.5 *Activity Diagram* Melihat Data Pendaftar

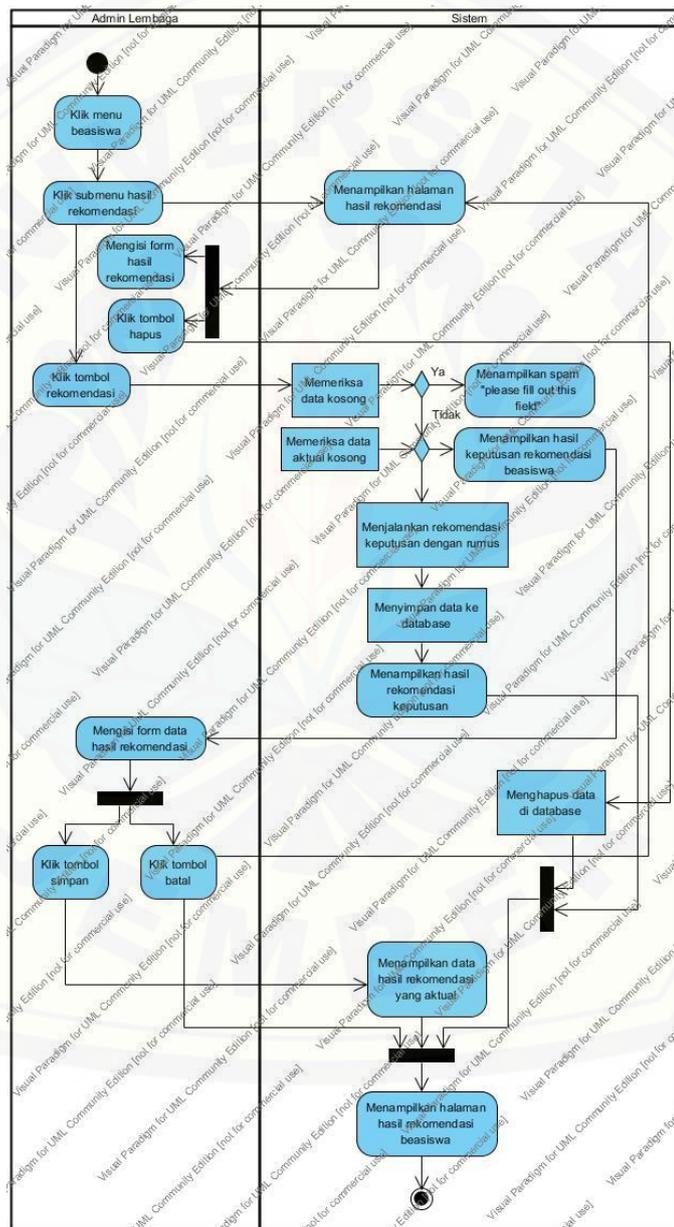
*Activity diagram* melihat data pendaftar menggambarkan tentang proses yang terjadi saat actor melihat data pendaftar. Aktor yang dapat melihat data pendaftar adalah admin. Admin dapat melihat data saja. *Activity diagram* melihat data pendaftar dapat dilihat pada lampiran C Gambar 5.

#### 4.3.5.6 *Activity Diagram* Mengelola Data Beasiswa

*Activity diagram* mengelola data beasiswa menggambarkan tentang proses yang terjadi saat aktor mengelola data beasiswa. Aktor yang dapat mengelola data beasiswa adalah admin. Admin dapat melihat, menambah, mengubah, dan menghapus data beasiswa. *Activity diagram* mengelola data beasiswa dapat dilihat pada lampiran C Gambar 6.

4.3.5.7 Activity Diagram Mengelola Hasil Rekomendasi

Activity Diagram mengelola hasil rekomendasi menggambarkan tentang proses yang terjadi saat aktor mengelola hasil rekomendasi. Aktor yang dapat mengelola hasil rekomendasi adalah Admin. Activity Diagram mengelola hasil rekomendasi dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Activity Diagram mengelola hasil rekomendasi

#### 4.3.5.8 *Activity Diagram* Mengelola Data Lembaga

*Activity diagram* mengelola data lembaga menggambarkan tentang proses yang terjadi saat aktor mengelola data lembaga. Aktor yang dapat mengelola data lembaga adalah Super Admin. Super Admin dapat melihat, menambah, mengubah, dan menghapus data beasiswa. *Activity diagram* dapat dilihat pada lampiran C Gambar 8.

#### 4.3.5.9 *Activity Diagram* Mengelola Kriteria Pendaftaran

*Activity diagram* mengelola data kriteria menggambarkan tentang proses yang terjadi saat aktor mengelola data kriteria. Aktor yang dapat mengelola data kriteria adalah Super Admin. Super Admin dapat melihat, menambah, mengubah, dan menghapus data beasiswa. *Activity diagram* mengelola kriteria pendaftaran dapat dilihat pada lampiran C Gambar 9.

#### 4.3.5.10 *Activity Diagram* Keluar

*Activity diagram* keluar menggambarkan tentang proses yang terjadi saat aktor ingin keluar dari sistem informasi. *Activity diagram* keluar dapat dilihat pada lampiran C Gambar 10.

### 4.3.6 *Class Diagram*

*Class Diagram* merupakan diagram untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem informasi yang nantinya digunakan. Dalam paradigma OOP (*Object Oriented Programming*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller*. *Class diagram* yang digunakan dapat dilihat pada Gambar 4.5

### 4.3.7 *Entity Relationship Diagram* (ERD)

*Entity Relationship Diagram* merupakan gambaran hubungan antar data dalam basis data yang memiliki relasi. ERD yang digunakan Sistem Informasi Rekomendasi Mahasiswa Terhadap Penerimaan Beasiswa dapat dilihat pada Gambar 4.6



#### 4.4 Penulisan Kode Program

Penulisan kode program ini merupakan implementasi dari desain sistem informasi yang telah dirancang sebelumnya.

##### 4.4.1 Kode Program Masuk

Kode program Masuk terletak pada controller *UserController*, model *User*, dan view *page-Masuk*. Terdapat pengecekan username dan password yang diinputkan user pada saat proses Masuk. Jika username dan password tidak sesuai dengan yang terdapat di database, maka user tidak dapat masuk ke dalam sistem informasi.

##### 4.4.2 Kode Program Menambah Data Pendaftar Baru

Kode program menambah data pendaftar baru terletak pada *controller User*, *view User*. Terdapat *function index()* yang digunakan untuk menampilkan halaman yang berisi data pendaftar, *function create()* untuk menampilkan halaman form tambah data pendaftar, *function store()* untuk memasukkan data yang user masukkan pada form tambah data pengguna ke database.

##### 4.4.3 Kode Program Menambah Pendaftaran

Kode program mengelola data pendaftaran terletak pada *controller SalesController*, *modal Sale*, *modal Sales*, dan *view Sales*. Terdapat *function index()* yang digunakan untuk menampilkan halaman yang berisi semua data pendaftaran, *function create()* untuk menampilkan halaman *form* tambah data pendaftaran, *function store()* untuk memasukkan data yang dimasukkan pada form tambah data pendaftaran ke database, *function update()* untuk menampilkan halaman form edit data pendaftaran, *function edit()* untuk memasukkan data yang diubah pada form edit data pendaftaran ke database, dan *function delete()* untuk menghapus data pendaftaran yang dipilih.

##### 4.4.4 Kode Program Melihat Hasil Rekomendasi

Kode program melihat hasil rekomendasi terletak pada *controller*, *view* . Terdapat *function index()* yang digunakan untuk menampilkan halaman Penjualan.

##### 4.4.5 Kode Program Mengelola Data Beasiswa

Kode program mengelola data beasiswa terletak pada *controller MaterialsController*, *view Materials*, *modal Materials*. Terdapat *function index()*

yang digunakan untuk menampilkan halaman yang berisi semua data beasiswa, *function* create() untuk menampilkan form tambah beasiswa, *function* store() untuk memasukkan data yang user masukkan pada form tambah data pengguna ke *database*, *function* update() untuk menampilkan halaman *form* ubah data beasiswa, *function* edit() untuk memasukkan data yang diubah pada form ubah data beasiswa ke *database*, dan *function* delete() untuk menghapus beasiswa yang dipilih.

#### 4.4.6 Kode Program Mengelola Data Lembaga

Kode program mengelola data Lembaga terletak pada *controller* Stock, *view* Stock, *modal* Stock. Terdapat *function* index() yang digunakan untuk menampilkan halaman yang berisi semua data lembaga, *function* create() untuk menampilkan *form* tambah data lembaga, *function* store() untuk memasukkan data yang dimasukkan pada *form* tambah data lembaga ke *database*, *function* update() untuk menampilkan *form* ubah data lembaga bahan baku, *function* edit() untuk memasukkan data yang diubah pada form edit data lembaga ke *database*, dan *function* delete() untuk menghapus data lembaga.

#### 4.4.7 Kode Program Melihat Data Pendaftaran

Kode program melihat data pendaftaran baku terletak pada *controller* Stock, *modal* Stock dan *view* Stock. Terdapat *function* index() yang digunakan untuk menampilkan halaman data pendaftaran.

#### 4.4.8 Kode Program Mengelola Hasil Rekomendasi

Kode program melihat data hasil rekomendasi terletak pada *controller* Forecast, *modal* Forecast dan *view* Forecast. Terdapat deklarasi variabel-variabel yang akan digunakan dalam *function* index() pada Forecast yang kemudian variabel-variabel tersebut akan diberi nilai dengan data yang tersedia di *database*. Pengguna memilih salah satu data rekomendasi, yang dilakukan adalah melakukan perhitungan menggunakan data rekomendasi, akan dilakukan pengecekan kembali terhadap perhitungan hasil rekomendasi.

Tabel 4.4 Tabel Kode Program Mengelola Hasil Rekomendasi

```
1. public function hasil($id){
2.     $beasiswa = Beasiswa::find($id);
3.     $data = DetailBeasiswa::where('beasiswa_id',$id)->get();
```

```
4.     $t = round(count($data)/2);
5.     $arr = array();
6.     $i = 0;
7.     foreach ($data as $key => $value) {
8.         $arr[$i][0] = $value;
9.         $arr[$i][1] = $value->semester_aktif;
10.        $arr[$i][2] = $value->ipk;
11.        $arr[$i][3] = $value->penghasilan_ortu;
12.        $arr[$i][4] = $value->prestasi;
13.        $arr[$i++][5] = $value->organisasi;
14.    }
15.    $data1 = $arr;
16.    $i = 0;
17.    $arr2 = array();
18.    foreach ($data as $key => $value) {
19.        $arr2[$i][0] = $value;
20.        $arr2[$i][1] = $value->semester_aktif/$arr[$t][1];
21.        $arr2[$i][2] = $value->ipk/$arr[$t][2];
22.        $arr2[$i][3] = $value->penghasilan_ortu/$arr[$t][3];
23.        $arr2[$i][4] = $value->prestasi/$arr[$t][4];
24.        $arr2[$i++][5] = $value->organisasi/$arr[$t][5];
25.    }
26.    $data2 = $arr2;
27.    // dd($arr2);
28.    $bobot = Kriteria::all();
29.    // dd($bobot);
30.    $arr3 = array();
31.    $i = 0;
32.    foreach ($data as $key => $value) {
33.        $arr3[$i][0] = $value;
```

```

34.          $arr3[$i][1]          =          ($arr2[$i][1]/$bobot[0]-
>value)+($arr2[$i][2]/$bobot[1]->value)+($arr2[$i][3]/$bobot[2]-
>value)+($arr2[$i][4]/$bobot[3]->value)+($arr2[$i][5]/$bobot[4]->value);
35.          $i++;
36.      }
37.      $data3 = $arr3;
38.      // $arr4 = array();
39.      for ($i=0; $i < count($arr3) ; $i++) {
40.          for ($j=$i+1; $j < count($arr3) ; $j++) {
41.              if($arr3[$i][1] < $arr3[$j][1]){
42.                  $temp = $arr3[$i];
43.                  $arr3[$i] = $arr3[$j];
44.                  $arr3[$j] = $temp;
45.              }
46.          }
47.      }
48.      $data4 = $arr3;
49.      // dd($arr3);
50.      return
view('admin.hasil',compact('arr3','beasiswa','data1','data2','data3','data4'));
}

```

#### 4.4.9 Kode Program Mengelola Kriteria Pendaftaran

Kode program mengelola data kriteria pendaftaran terletak pada *controller* *MaterialsController*, *view* *Materials*, *modal* *Materials*. Terdapat *function* *index()* yang digunakan untuk menampilkan halaman yang berisi semua data kriteria pendaftaran, *function* *create()* untuk menampilkan form tambah beasiswa, *function* *store()* untuk memasukkan data yang user masukkan pada form tambah data pengguna ke *database*, *function* *update()* untuk menampilkan halaman *form* ubah data beasiswa, *function* *edit()* untuk memasukkan data yang diubah pada form ubah

data kriteria pendaftaran ke *database*, dan *function delete()* untuk menghapus kriteria yang dipilih.

#### 4.4.10 Kode Program Keluar

Kode program keluar terletak pada *controller* User. Keluar digunakan *user* untuk keluar dari sistem informasi.

### 4.5 Pengujian Sistem informasi

Pengujian dilakukan untuk mengevaluasi sistem informasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *black box*.

#### 4.5.1 Pengujian *Black Box*

Pengujian *black box* berfungsi untuk menguji sistem informasi dari sisi fungsionalitas sistem informasi dengan tujuan mengetahui apakah fitur-fitur, inputan dan keluaran sistem informasi sesuai dengan kebutuhan pengguna atau tidak. Fitur-fitur yang diujikan mencakup fitur melihat, menambah, mengubah dan menghapus data serta keluaran yang diharapkan dari sistem informasi saat fitur tersebut dijalankan. Hasil pengujian *black box* pada fitur hasil peramalan dapat dilihat pada Tabel 4.5.

Pengujian *black box* berfungsi untuk menguji sistem informasi dari sisi fungsionalitas sistem informasi dengan tujuan mengetahui apakah fitur-fitur, inputan dan keluaran sistem informasi sesuai dengan kebutuhan pengguna atau tidak. Fitur-fitur yang diujikan mencakup fitur melihat, menambah, mengubah dan menghapus data serta keluaran yang diharapkan dari sistem informasi saat fitur tersebut dijalankan. Hasil pengujian *black box* pada fitur hasil peramalan dapat dilihat pada Tabel 4.5.

Tabel 4.1 Hasil Pengujian Black Box

Nomor Usecase	USC-01
Nama	Masuk
Aktor	Pendaftar

Deskripsi Singkat	Aktor membuka halaman Masuk untuk autentifikasi hak akses sistem informasi		
Prekondisi	Aktor membuka halaman Masuk		
Prakondisi	Aktor berhasil masuk ke sistem informasi		
Flow of Events		Kesesuaian	
Skenario Normal : Masuk		Benar	Salah
1. Membuka sistem informasi pada <i>browser</i>		√	
	2. Menampilkan halaman Masuk yang berisi <ul style="list-style-type: none"> <li>• Form yang berisi atribut:                         <ul style="list-style-type: none"> <li>- <i>Username</i> berupa email</li> <li>- <i>Password</i></li> </ul> </li> <li>• Tombol Masuk</li> </ul>	√	
3. Mengisi form Masuk		√	
4. Klik tombol Masuk		√	
	5. Menampilkan halaman Dashboard untuk hak akses Pendaftar	√	
Skenario Alternatif : <i>Username</i> atau <i>Password</i> kosong		√	
4. Klik tombol Masuk		√	
	5. Menampilkan spam " <i>Please fill out this field</i> "	√	
Skenario Alternatif : <i>Username</i> dan <i>password</i> tidak sesuai		√	
4. Klik tombol Masuk		√	

	5. Menampilkan spam “ <i>These credentials do not match our records</i> ”	√	
--	---	---	--

Nomor Usecase	USC-01		
Nama	Masuk		
Aktor	Admin		
Deskripsi Singkat	Aktor membuka halaman Masuk untuk autentifikasi hak akses sistem informasi		
Prekondisi	Aktor membuka halaman Masuk		
Prakondisi	Aktor berhasil masuk ke sistem informasi		
	Flow of Events	Kesesuaian	
	Skenario Normal : Masuk	Benar	Salah
1. Membuka sistem informasi pada <i>browser</i>		√	
	2. Menampilkan halaman Masuk yang berisi <ul style="list-style-type: none"> <li>• Form yang berisi atribut:                         <ul style="list-style-type: none"> <li>- <i>Username</i> berupa email</li> <li>- <i>Password</i></li> </ul> </li> <li>• Tombol Masuk</li> </ul>	√	
3. Mengisi form Masuk		√	
4. Klik tombol Masuk		√	

	5. Menampilkan halaman Dashboard untuk hak akses Admin	√	
Skenario Alternatif : <i>Username</i> atau <i>Password</i> kosong			
4. Klik tombol Masuk		√	
	5. Menampilkan spam “ <i>Please fill out this field</i> ”	√	
Skenario Alternatif : <i>Username</i> dan <i>password</i> tidak sesuai			
4. Klik tombol Masuk		√	
	5. Menampilkan spam “ <i>These credentials do not match our records</i> ”	√	

Nomor Usecase	USC-02		
Nama	Mengelola Data Pengguna Baru		
Aktor	Pendaftar		
Deskripsi Singkat	Aktor mengelola data pengguna baru		
Prekondisi	Aktor memilih Menu Daftar		
Prakondisi	Aktor berhasil melihat, menambah, mengubah dan menghapus data pengguna baru		
	Flow of Events	Kesesuaian	
	Skenario Normal : Menambah Data Pengguna Baru	Benar	Salah
1. Klik Beasiswa		√	
2. Klik Daftar		√	

	<p>3. Menampilkan halaman Pengguna Baru yang meliputi</p> <ul style="list-style-type: none"> <li>• Form tambah data pengguna baru berupa: <ul style="list-style-type: none"> <li>- Nama</li> <li>- Alamat E-mail</li> <li>- Password</li> <li>- Konfirmasi Password</li> </ul> </li> <li>• Tombol aksi “Daftar” dan “Login”</li> </ul>	√	
4. Mengisi form tambah data pendaftar baru kemudian klik tombol Sunting		√	
	5. Menyimpan data ke dalam database	√	
	6. Menampilkan halaman pengguna baru dengan data baru yang telah ditambahkan.	√	

Nomor Usecase	USC-04
Nama	Mengelola Data Pendaftaran
Aktor	Pendaftar
Deskripsi Singkat	Aktor mengelola data pendaftaran
Prekondisi	Aktor memilih menu data pendaftaran
Prakondisi	Aktor berhasil melihat dan menambah data Pendaftaran
	Flow of Events
	Kesesuaian

Skenario Normal : Mengelola Data Pendaftaran		Benar	Salah
1. Klik menu Pendaftaran		√	
	2. Menampilkan halaman Pendaftaran yang meliputi <ul style="list-style-type: none"> <li>• <i>Field</i> pencarian</li> <li>• Tabel data pengguna yang berisi:               <ul style="list-style-type: none"> <li>- Nama Pendaftar</li> <li>- Email</li> <li>- Waktu Dibuat</li> </ul> </li> <li>• Form Tambah data pendaftaran baru berupa:               <ul style="list-style-type: none"> <li>- Nama</li> <li>- Email</li> <li>- <i>Password</i></li> <li>- Konfirmasi <i>Password</i></li> <li>- Tombol Tambahkan</li> </ul> </li> </ul>	√	
3. Mengisi form tambah data pendaftaran baru, kemudian klik tombol Tambahkan		√	
	4. Menyimpan data ke dalam <i>database</i>	√	
	5. Menampilkan halaman Pendaftaran dengan data baru yang telah ditambahkan ke tabel.	√	
Skenario Alternatif : Isian form kosong		√	
3. Klik tombol Tambahkan		√	
	4. Menampilkan span " <i>Please fill out this field</i> "	√	

Nomor Usecase	USC-05		
Nama	Melihat Data Pendaftaran		
Aktor	Admin Sistem		
Deskripsi Singkat	Aktor melihat Data Pendaftaran		
Prekondisi	Aktor memilih menu Pendaftaran		
Prakondisi	Aktor berhasil melihat Pendaftaran		
	Flow of Events		Kesesuaian
	Skenario Normal : Melihat Data Pendaftaran		Benar   Salah
4.3.4. Klik menu Beasiswa		√	
4.3.5. Klik menu Data Pendaftaran		√	
	4.3.6. Menampilkan halaman Data Pendaftaran yang meliputi <ul style="list-style-type: none"> <li>• Tabel data pendaftar</li> <li>• Field pencarian</li> </ul>	√	

Nomor Usecase	USC-07		
Nama	Melihat Data Hasil Rekomendasi		
Aktor	Pendaftar		
Deskripsi Singkat	Aktor melihat Data Hasil Rekomendasi		
Prekondisi	Aktor memilih Menu Hasil Rekomendasi		
Prakondisi	Aktor berhasil Melihat Hasil Rekomendasi		
	Flow of Events		Kesesuaian
	Skenario Normal : Melihat Hasil Rekomendasi		Benar   Salah
3.6.1. Klik menu beasiswa		√	
3.6.2. Klik menu hasil rekomendasi		√	

	4.4. Menampilkan halaman hasil rekomendasi yang meliputi <ul style="list-style-type: none"> <li>• <i>Field</i> pencarian</li> <li>• Tabel data hasil rekomendasi yang berupa persen</li> </ul>	√	
--	--	---	--

Nomor Usecase	USC-07		
Nama	Mengelola Data Beasiswa		
Aktor	Admin		
Deskripsi Singkat	Aktor mengelola data beasiswa		
Prekondisi	Aktor memilih menu data beasiswa		
Prakondisi	Aktor berhasil melihat dan menambah data beasiswa		
	Flow of Events	Kesesuaian	
	Skenario Normal : Mengelola Data Beasiswa	Benar	Salah
4.4.1. Klik menu Beasiswa		√	
	4.4.2. Menampilkan halaman Beasiswa yang meliputi <ul style="list-style-type: none"> <li>• ID</li> <li>• Nama</li> <li>• Lembaga</li> <li>• Tanggal awal</li> <li>• Tanggal akhir</li> <li>• Kuota</li> <li>• Tombol aksi berupa                         <ul style="list-style-type: none"> <li>- Update</li> <li>- Hapus</li> <li>- Hasil</li> </ul> </li> </ul> Form tambah data beasiswa baru	√	
4.4.3. Mengisi form tambah data beasiswa baru, kemudian klik tombol Tambahkan		√	

	4.5. Menyimpan data ke dalam <i>database</i>	√	
	5. Menampilkan halaman beasiswa dengan data baru yang telah ditambahkan ke tabel.	√	
Skenario Alternatif : Isian form kosong		√	
3. Klik tombol Tambahkan		√	
	4. Menampilkan span "Please fill out this field"	√	

Nomor Usecase	USC-08		
Nama	Mengelola Data Lembaga		
Aktor	Admin Lembaga		
Deskripsi Singkat	Aktor mengelola data lembaga		
Prekondisi	Aktor memilih menu data lembaga		
Prakondisi	Aktor berhasil melihat dan menambah data lembaga		
	Flow of Events	Kesesuaian	
	Skenario Normal : Mengelola Data Lembaga	Benar	Salah
5.1.1. Klik menu Lembaga		√	
	5.1.2. Menampilkan halaman Lembaga yang meliputi <ul style="list-style-type: none"> <li>• ID</li> <li>• Nama Lembaga</li> <li>• Tombol aksi berupa                         <ul style="list-style-type: none"> <li>- Update</li> <li>- Hapus</li> </ul> </li> </ul> Form tambah data Lembaga baru	√	
Mengisi form tambah data lembaga baru,		√	

kemudian klik tombol Tambahkan			
	5.2. Menyimpan data ke dalam <i>database</i>	√	
	5.3. Menampilkan halaman beasiswa dengan data baru yang telah ditambahkan ke tabel.	√	
	Skenario Alternatif : Isian form kosong	√	
3. Klik tombol Tambahkan		√	
	4. Menampilkan span “ <i>Please fill out this field</i> ”	√	

Nomor Usecase	USC-09		
Nama	Mengelola Kriteria Pendaftaran		
Aktor	Admin Lembaga		
Deskripsi Singkat	Aktor mengelola kriteria pendaftaran		
Prekondisi	Aktor memilih menu Lembaga kriteria pendaftaran		
Prakondisi	Aktor berhasil melihat dan menambah data kriteria pendaftaran		
	Flow of Events	Kesesuaian	
	Skenario Normal : Mengelola Kriteria Pendaftaran	Benar	Salah
5.3.1. Klik menu Lembaga		√	
	5.3.2. Menampilkan halaman Lembaga kriteria pendaftaran yang meliputi <ul style="list-style-type: none"> <li>• Form tambah data kriteria</li> </ul>	√	
5.3.3. Mengisi form tambah data kriteria baru, kemudian klik tombol Tambahkan		√	

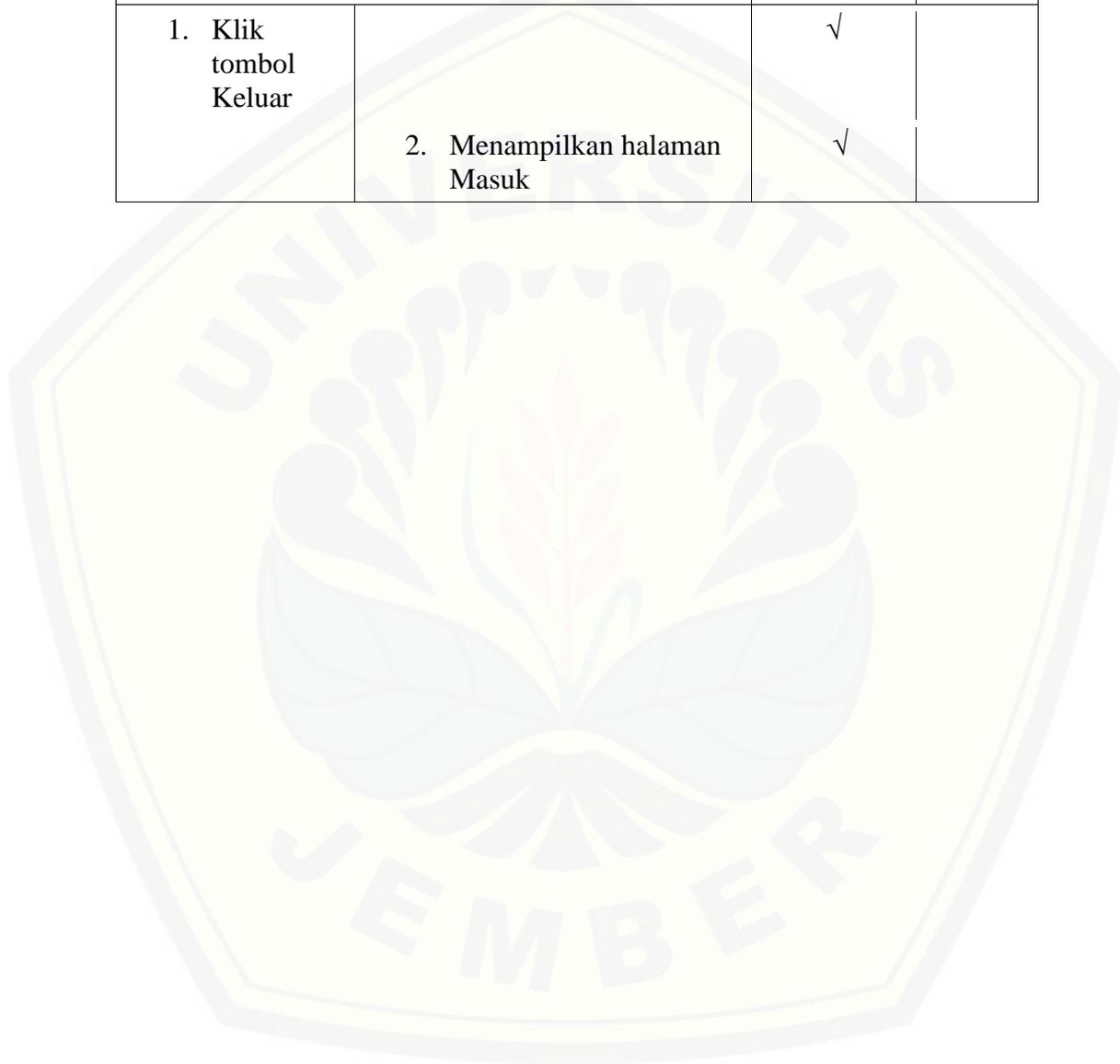
	5.3.4. Menyimpan data ke dalam <i>database</i>	√	
	5.3.5. Menampilkan halaman lembaga dengan data baru yang telah ditambahkan ke tabel.	√	
Skenario Alternatif : Isian form kosong		√	
3. Klik tombol Tambahkan		√	
	4. Menampilkan span " <i>Please fill out this field</i> "	√	

Nomor Usecase	USC-010		
Nama	Mengelola Hasil Rekomendasi		
Aktor	Admin		
Deskripsi Singkat	Aktor mengelola hasil rekomendasi		
Prekondisi	Aktor memilih Menu Beasiswa		
Prakondisi	Aktor berhasil melihat, menambah, mengubah dan menghapus hasil rekomendasi		
	Flow of Events	Kesesuaian	
	Skenario Normal : Mengelola Hasil Rekomendasi	Benar	Salah
5.3.5.1. Klik Beasiswa		√	
	5.3.5.2. Menampilkan halaman Beasiswa yang meliputi <ul style="list-style-type: none"> <li>• ID</li> <li>• Nama</li> <li>• Lembaga</li> <li>• Tanggal awal</li> <li>• Tanggal akhir</li> <li>• Kuota</li> <li>• Tombol aksi berupa                         <ul style="list-style-type: none"> <li>- Update</li> <li>- Hapus</li> <li>- Hasil</li> </ul> </li> </ul>	√	

	Form tambah data beasiswa baru		
5.3.5.3. Mengisi form tambah data hasil rekomendasi baru kemudian klik tombol Sunting		√	
	5.3.5.4. Menyimpan data ke dalam database	√	
	5.3.5.5. Menampilkan halaman beasiswa hasil rekomendasi dengan data baru yang telah ditambahkan.	√	
Skenario Alternatif : Isian form kosong			
3. Klik tombol Sunting		√	
	4. Menampilkan spam “ <i>Please fill out this field</i> ”	√	
Skenario Alternatif : Isian form tidak sesuai			
6. Klik tombol Sunting		√	
	7. Menampilkan spam “ <i>These credentials do not match our records</i> ”	√	

Nomor Usecase	USC-11
Nama	Keluar
Aktor	Admin

Deskripsi Singkat	Aktor keluar dari sistem informasi		
Prekondisi	Aktor menggunakan sistem informasi		
Prakondisi	Aktor keluar dari sistem informasi		
Flow of Events		Kesesuaian	
Skenario Normal : Keluar		Benar	Salah
1. Klik tombol Keluar		√	
	2. Menampilkan halaman Masuk	√	



## BAB 6 PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

### 6.1 Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. *Simple Additive Weighting* dapat menyelesaikan permasalahan penentuan rekomendasi mahasiswa terhadap beasiswa dengan menghitung nilai preferensi untuk masing-masing alternatif dan alternatif yang memiliki nilai preferensi paling tinggi mengindikasikan bahwa alternatif tersebut lebih direkomendasikan untuk memperoleh beasiswa. Hal ini ditunjukkan yang memperoleh beasiswa dengan hasil tertinggi adalah Risa dengan nilai 26.
2. Sistem Rekomendasi akan melakukan perhitungan terhadap nilai kriteria dari data pendaftar. Kemudian sistem akan menampilkan hasil rekomendasi perhitungan metode *Simple Additive Weighting* yang hasil akhirnya di sortir dari nilai tertinggi.

### 6.2 Saran

Beberapa saran yang ditujukan untuk memberikan masukan yang lebih baik pada penelitian selanjutnya yaitu:

1. Sistem informasi yang dikembangkan pada penelitian ini hanya memiliki fitur untuk evaluasi yaitu bertujuan untuk dapat melakukan rekomendasi mahasiswa pada periode selanjutnya. Diharapkan pada penelitian selanjutnya, sistem informasi yang dikembangkan mempunyai penilaian lebih yaitu psikotest dan hasil rekomendasi nilai akhir yang berupa persen.

2. Sistem informasi yang dikembangkan hanya dapat melakukan rekomendasi data tanpa adanya informasi yang ditentukan oleh admin Lembaga. Diharapkan pada penelitian selanjutnya sistem informasi yang dikembangkan dapat melakukan rekomendasi terhadap data informasi yaitu data tahunan beasiswa.



## DAFTAR PUSTAKA

- Apriansyah, P., & Hardiyanti, D. Y. (2011). Penentuan Penerima Beasiswa Dengan Menggunakan Fuzzy Multiple Attribute Decision Making. *Jurnal Sistem Informasi (JSI)*.
- Eniryati, S. (2011). Perancangan Sistem Pendukung Pengambilan Keputusan untuk Penerimaan . *Jurnal Teknologi Informasi Dinamik*.
- Eniyarti, S. (t.thn.). Perancangan Sistem Pendukung Pengambilan Keputusan untuk Penerimaan . Program Studi Sistem Informasi, Universitas Stikubank .
- Herbert , A. S. (2014). Tahap Pengambilan Keputusan.
- Raffy. (2014). Metode SAW Dalam Sistem Pendukung Keputusan dan Penerapannya.
- Saputra, H. (2016). Sistem Pendukung Keputusan Penentu Beasiswa PPA dan BBM pada Perguruan Tinggi Swasta Provinsi Sumbar, Riau, Jambi dan Kepri Di Kopertis Wilayah X Padang Menggunakan Metode AHP (Analytical Hierarchy Process). *Teknosi*.
- Sommerville. (2010). Software Engineering Ninth Edition. *Addision Wesley*.
- Supriyanti, W. (2014). Rancang Bangun Aplikasi Sistem Pendukung Keputusan Penerima Beasiswa dengan Metode SAW. *Citec Journal*.
- Tiyas, S. (2015). Sistem Informasi Pemilihan Mahasiswa Berprestasi Dengan Metode Simple Additive Weighting (Saw) Di Universitas Jember.
- Turban, Efraim, Aronson, & Jay, E. (2001). *Decision Support Systems and Intellegent Systems, 6th Edition*. Upper Saddle River, NJ: Prentice Hall.
- Yohana, D. L., Rani, M. S., & Heni, R. (2010). Sistem Pendukung Keputusan Penentuan Karyawan Terbaik Menggunakan Metode Saw.

## LAMPIRAN

### LAMPIRAN A. SKENARIO

#### A.1. Skenario Masuk

Nomor Usecase	USC-01
Nama	Masuk
Aktor	Pendaftar
Deskripsi Singkat	Aktor membuka halaman Masuk untuk autentifikasi hak akses sistem informasi
Prekondisi	Aktor membuka halaman Masuk
Prakondisi	Aktor berhasil masuk ke sistem informasi
Flow of Events	
Skenario Normal : Masuk	
6. Membuka sistem informasi pada <i>browser</i>	7. Menampilkan halaman Masuk yang berisi <ul style="list-style-type: none"> <li>• Form yang berisi atribut: <ul style="list-style-type: none"> <li>- <i>Username</i> berupa email</li> <li>- <i>Password</i></li> </ul> </li> <li>• Tombol Masuk</li> </ul>
8. Mengisi form Masuk	
9. Klik tombol Masuk	
	10. Menampilkan halaman Dashboard untuk hak akses Pendaftar
Skenario Alternatif : <i>Username</i> atau <i>Password</i> kosong	
6. Klik tombol Masuk	
	7. Menampilkan spam " <i>Please fill out this field</i> "
Skenario Alternatif : <i>Username</i> dan <i>password</i> tidak sesuai	

6. Klik tombol Masuk	7. Menampilkan spam “ <i>These credentials do not match our records</i> ”
----------------------	---

Nomor Usecase	USC-01
Nama	Masuk
Aktor	Admin
Deskripsi Singkat	Aktor membuka halaman Masuk untuk autentifikasi hak akses sistem informasi
Prekondisi	Aktor membuka halaman Masuk
Prakondisi	Aktor berhasil masuk ke sistem informasi
Flow of Events	
Skenario Normal : Masuk	
6. Membuka sistem informasi pada <i>browser</i>	7. Menampilkan halaman Masuk yang berisi <ul style="list-style-type: none"> <li>• Form yang berisi atribut:                         <ul style="list-style-type: none"> <li>- <i>Username</i> berupa email</li> <li>- <i>Password</i></li> </ul> </li> <li>• Tombol Masuk</li> </ul>
8. Mengisi form Masuk	
9. Klik tombol Masuk	
	10. Menampilkan halaman Dashboard untuk hak akses Admin
Skenario Alternatif : <i>Username</i> atau <i>Password</i> kosong	
6. Klik tombol Masuk	
	7. Menampilkan spam “ <i>Please fill out this field</i> ”
Skenario Alternatif : <i>Username</i> dan <i>password</i> tidak sesuai	

6. Klik tombol Masuk	7. Menampilkan spam “ <i>These credentials do not match our records</i> ”
----------------------	---

A.2. Skenario Mengelola Data Pendaftar Baru

Nomor Usecase	USC-02
Nama	Mengelola Data Pengguna Baru
Aktor	Pendaftar
Deskripsi Singkat	Aktor mengelola data pengguna baru
Prekondisi	Aktor memilih Menu Daftar
Prakondisi	Aktor berhasil melihat, menambah, mengubah dan menghapus data pengguna baru
Flow of Events	
Skenario Normal : Menambah Data Pengguna Baru	
1. Klik Beasiswa 2. Klik Daftar	
4. Mengisi form tambah data pendaftar baru kemudian klik tombol Sunting	3. Menampilkan halaman Pengguna Baru yang meliputi <ul style="list-style-type: none"> <li>• Form tambah data pengguna baru berupa:                         <ul style="list-style-type: none"> <li>- Nama</li> <li>- Alamat E-mail</li> <li>- Password</li> <li>- Konfirmasi Password</li> </ul> </li> <li>• Tombol aksi “Daftar” dan “Login”</li> </ul>
	5. Menyimpan data ke dalam database

	6. Menampilkan halaman pengguna baru dengan data baru yang telah ditambahkan.
--	---

A.3. Skenario Mengelola Pendaftaran

Nomor Usecase	USC-04
Nama	Mengelola Data Pendaftaran
Aktor	Pendaftar
Deskripsi Singkat	Aktor mengelola data pendaftaran
Prekondisi	Aktor memilih menu data pendaftaran
Prakondisi	Aktor berhasil melihat dan menambah data Pendaftaran
<p>Flow of Events</p> <p>Skenario Normal : Mengelola Data Pendaftaran</p>	
6. Klik menu Pendaftaran	
	<p>7. Menampilkan halaman Pendaftaran yang meliputi</p> <ul style="list-style-type: none"> <li>• <i>Field</i> pencarian</li> <li>• Tabel data pengguna yang berisi:                             <ul style="list-style-type: none"> <li>- Nama Pendaftar</li> <li>- Email</li> <li>- Waktu Dibuat</li> </ul> </li> <li>• Form Tambah data pendaftaran baru berupa:                             <ul style="list-style-type: none"> <li>- Nama</li> <li>- Email</li> <li>- <i>Password</i></li> <li>- Konfirmasi <i>Password</i></li> <li>- Tombol Tambahkan</li> </ul> </li> </ul>
8. Mengisi form tambah data pendaftaran baru, kemudian klik tombol Tambahkan	
	<p>9. Menyimpan data ke dalam <i>database</i></p> <p>10. Menampilkan halaman Pendaftaran dengan data baru yang telah ditambahkan ke tabel.</p>
<p>Skenario Alternatif : Isian form kosong</p>	
5. Klik tombol Tambahkan	

	6. Menampilkan span “ <i>Please fill out this field</i> ”
--	---

A.4. Skenario Mengelola Data Pendaftaran

Nomor Usecase	USC-07
Nama	Mengelola Data Beasiswa
Aktor	Admin
Deskripsi Singkat	Aktor mengelola data beasiswa
Prekondisi	Aktor memilih menu data beasiswa
Prakondisi	Aktor berhasil melihat dan menambah data beasiswa
Flow of Events	
Skenario Normal : Mengelola Data Beasiswa	
7.1.1. Klik menu Beasiswa	7.1.2. Menampilkan halaman Beasiswa yang meliputi <ul style="list-style-type: none"> <li>• ID</li> <li>• Nama</li> <li>• Lembaga</li> <li>• Tanggal awal</li> <li>• Tanggal akhir</li> <li>• Kuota</li> <li>• Tombol aksi berupa                         <ul style="list-style-type: none"> <li>- Update</li> <li>- Hapus</li> <li>- Hasil</li> </ul> </li> </ul> Form tambah data beasiswa baru
7.1.3. Mengisi form tambah data beasiswa baru, kemudian klik tombol Tambahkan	
	7.2. Menyimpan data ke dalam <i>database</i>
	8. Menampilkan halaman beasiswa dengan data baru yang telah ditambahkan ke tabel.
Skenario Alternatif : Isian form kosong	
5. Klik tombol Tambahkan	

	6. Menampilkan span “ <i>Please fill out this field</i> ”
--	---

## A.5. Skenario Mengelola Data Lembaga

Nomor Usecase	USC-08
Nama	Mengelola Data Lembaga
Aktor	Admin Lembaga
Deskripsi Singkat	Aktor mengelola data lembaga
Prekondisi	Aktor memilih menu data lembaga
Prakondisi	Aktor berhasil melihat dan menambah data lembaga
Flow of Events	
Skenario Normal : Mengelola Data Lembaga	
8.1.1. Klik menu Lembaga	8.1.2. Menampilkan halaman Lembaga yang meliputi <ul style="list-style-type: none"> <li>• ID</li> <li>• Nama Lembaga</li> <li>• Tombol aksi berupa <ul style="list-style-type: none"> <li>- Update</li> <li>- Hapus</li> </ul> </li> </ul> Form tambah data Lembaga baru
8.2. Mengisi form tambah data lembaga baru, kemudian klik tombol Tambahkan	
	8.3. Menyimpan data ke dalam <i>database</i>
	8.4. Menampilkan halaman beasiswa dengan data baru yang telah ditambahkan ke tabel.
Skenario Alternatif : Isian form kosong	
5. Klik tombol Tambahkan	6. Menampilkan span “ <i>Please fill out this field</i> ”

## A.6. Skenario Mengelola Kriteria Pendaftaran

Nomor Usecase	USC-09
Nama	Mengelola Kriteria Pendaftaran
Aktor	Admin Lembaga
Deskripsi Singkat	Aktor mengelola kriteria pendaftaran
Prekondisi	Aktor memilih menu Lembaga kriteria pendaftaran
Prakondisi	Aktor berhasil melihat dan menambah data kriteria pendaftaran
Flow of Events	
Skenario Normal : Mengelola Kriteria Pendaftaran	
8.4.1. Klik menu Lembaga	8.4.2. Menampilkan halaman Lembaga kriteria pendaftaran yang meliputi <ul style="list-style-type: none"> <li>Form tambah data kriteria</li> </ul>
8.4.3. Mengisi form tambah data kriteria baru, kemudian klik tombol Tambahkan	
	8.4.4. Menyimpan data ke dalam <i>database</i>
	8.4.5. Menampilkan halaman lembaga dengan data baru yang telah ditambahkan ke tabel.
Skenario Alternatif : Isian form kosong	
5. Klik tombol Tambahkan	6. Menampilkan span “ <i>Please fill out this field</i> ”

## A.7. Skenario Mengelola Hasil Rekomendasi

Nomor Usecase	USC-010
Nama	Mengelola Hasil Rekomendasi
Aktor	Admin
Deskripsi Singkat	Aktor mengelola hasil rekomendasi
Prekondisi	Aktor memilih Menu Beasiswa

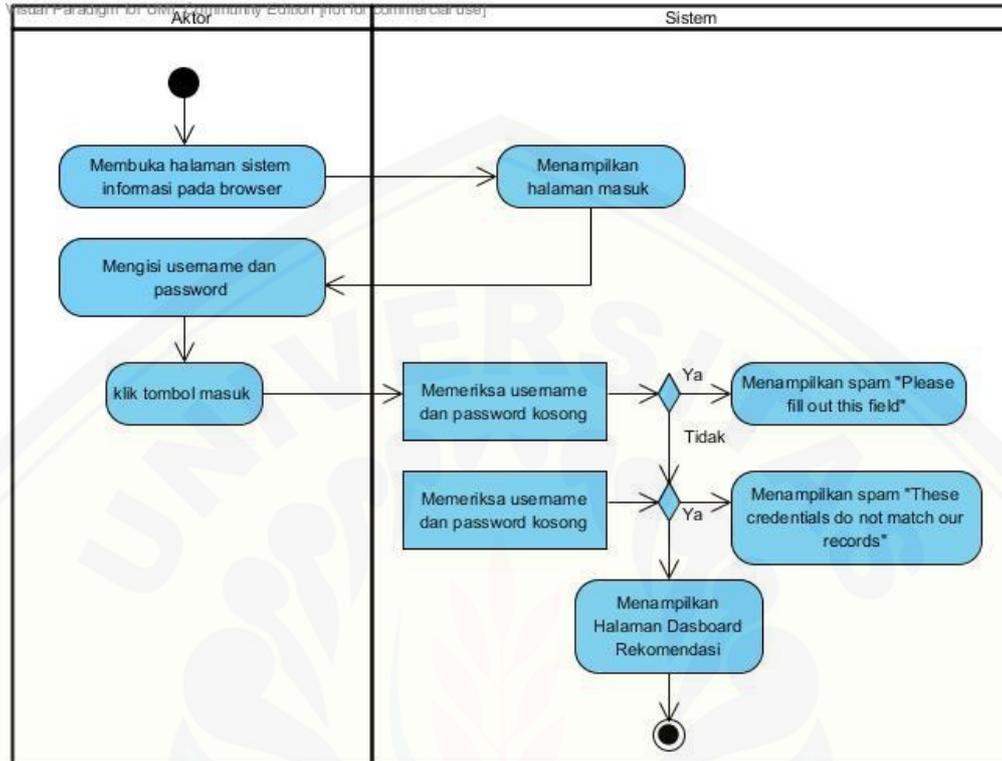
Prakondisi	Aktor berhasil melihat, menambah, mengubah dan menghapus hasil rekomendasi
Flow of Events Skenario Normal : Mengelola Hasil Rekomendasi	
8.4.5.1. Klik Beasiswa	
	<p>8.4.5.2. Menampilkan halaman Beasiswa yang meliputi</p> <ul style="list-style-type: none"> <li>• ID</li> <li>• Nama</li> <li>• Lembaga</li> <li>• Tanggal awal</li> <li>• Tanggal akhir</li> <li>• Kuota</li> <li>• Tombol aksi berupa <ul style="list-style-type: none"> <li>- Update</li> <li>- Hapus</li> <li>- Hasil</li> </ul> </li> </ul> <p>Form tambah data beasiswa baru</p>
8.4.5.3. Mengisi form tambah data hasil rekomendasi baru kemudian klik tombol Sunting	
	8.4.5.4. Menyimpan data ke dalam database
	8.4.5.5. Menampilkan halaman beasiswa hasil rekomendasi dengan data baru yang telah ditambahkan.
Skenario Alternatif : Isian form kosong	
3. Klik tombol Sunting	
	4. Menampilkan spam ““ <i>Please fill out this field</i> ””
Skenario Alternatif : Isian form tidak sesuai	
9. Klik tombol Sunting	
	10. Menampilkan spam ““ <i>These credentials do not match our records</i> ””

## A.8. Skenario Keluar

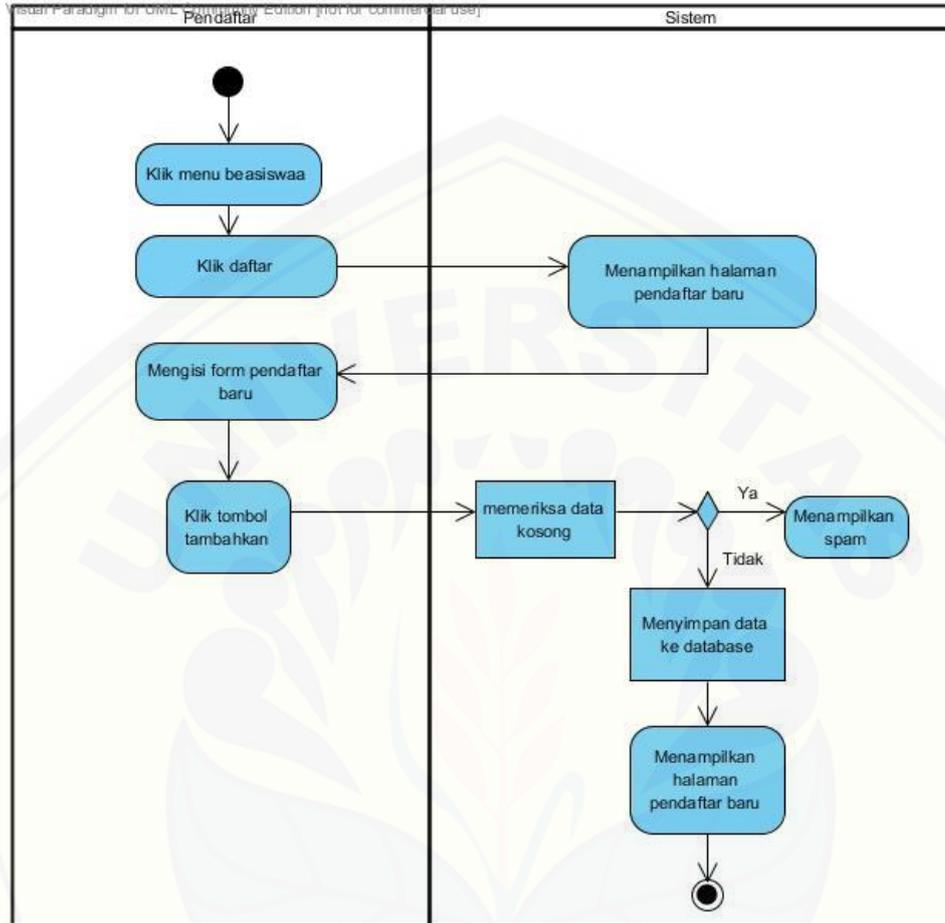
Nomor Usecase	USC-11
Nama	Keluar
Aktor	Admin
Deskripsi Singkat	Aktor keluar dari sistem informasi
Prekondisi	Aktor menggunakan sistem informasi
Prakondisi	Aktor keluar dari sistem informasi
Flow of Events	
Skenario Normal : Keluar	
1. Klik tombol Keluar	2. Menampilkan halaman Masuk

LAMPIRAN B. ACTIVITY DIAGRAM

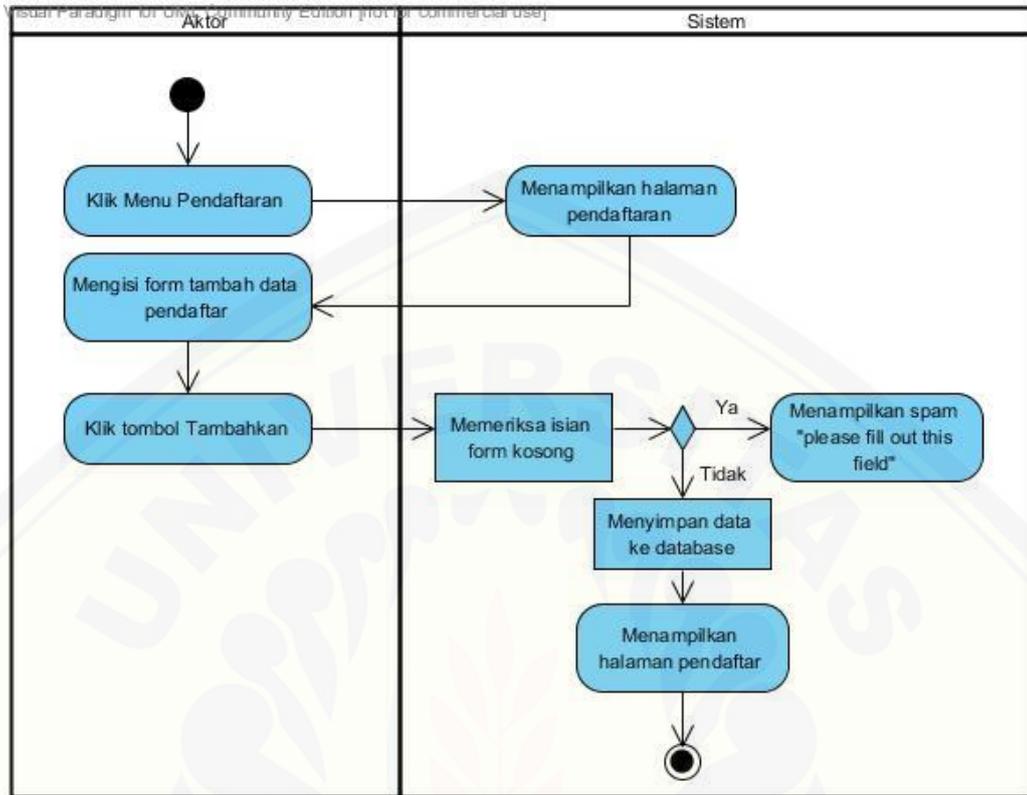
B.1. Activity Diagram Masuk



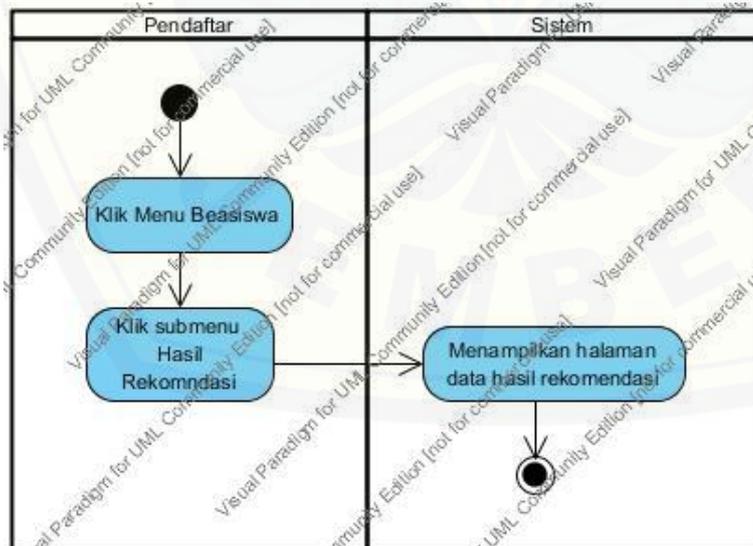
B.2. Activity Diagram Menambah Data Pendaftar Baru



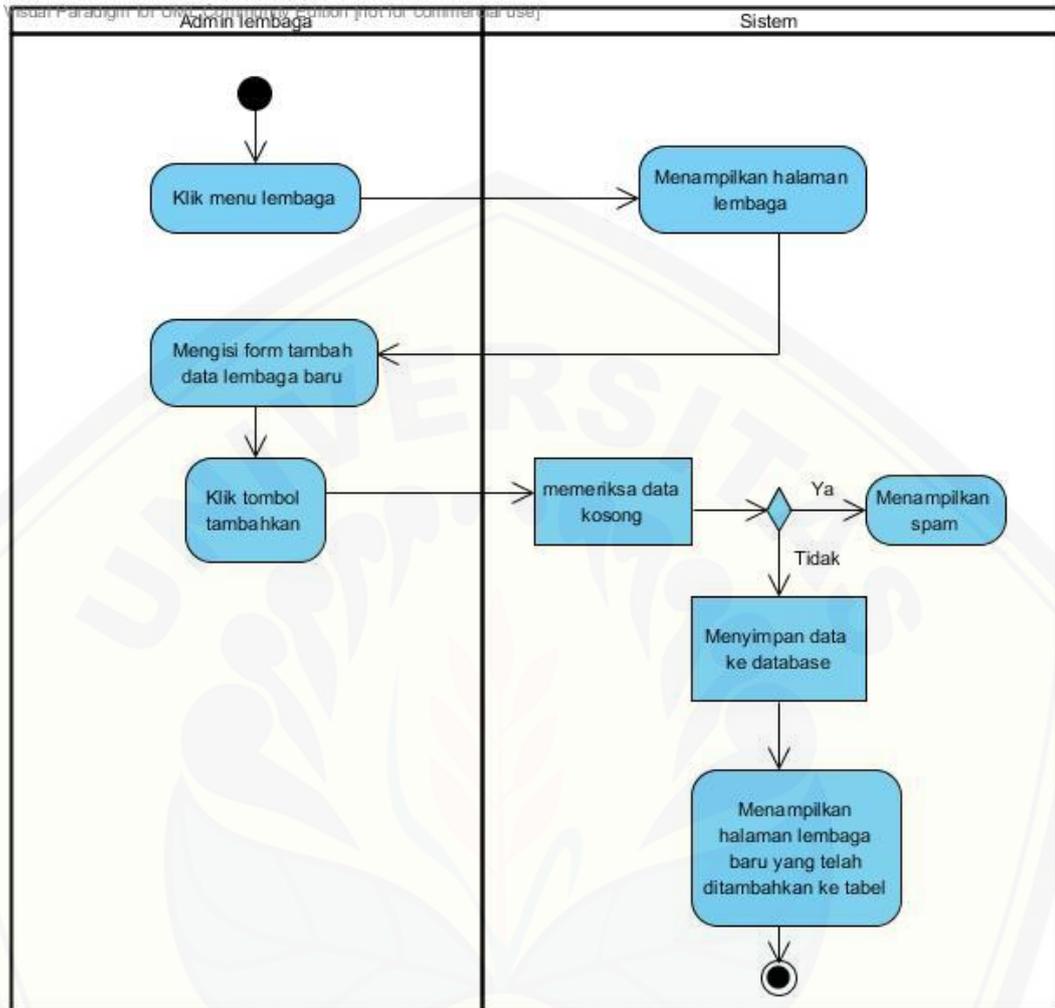
B.3. Activity Diagram Menambah Data Pendaftaran



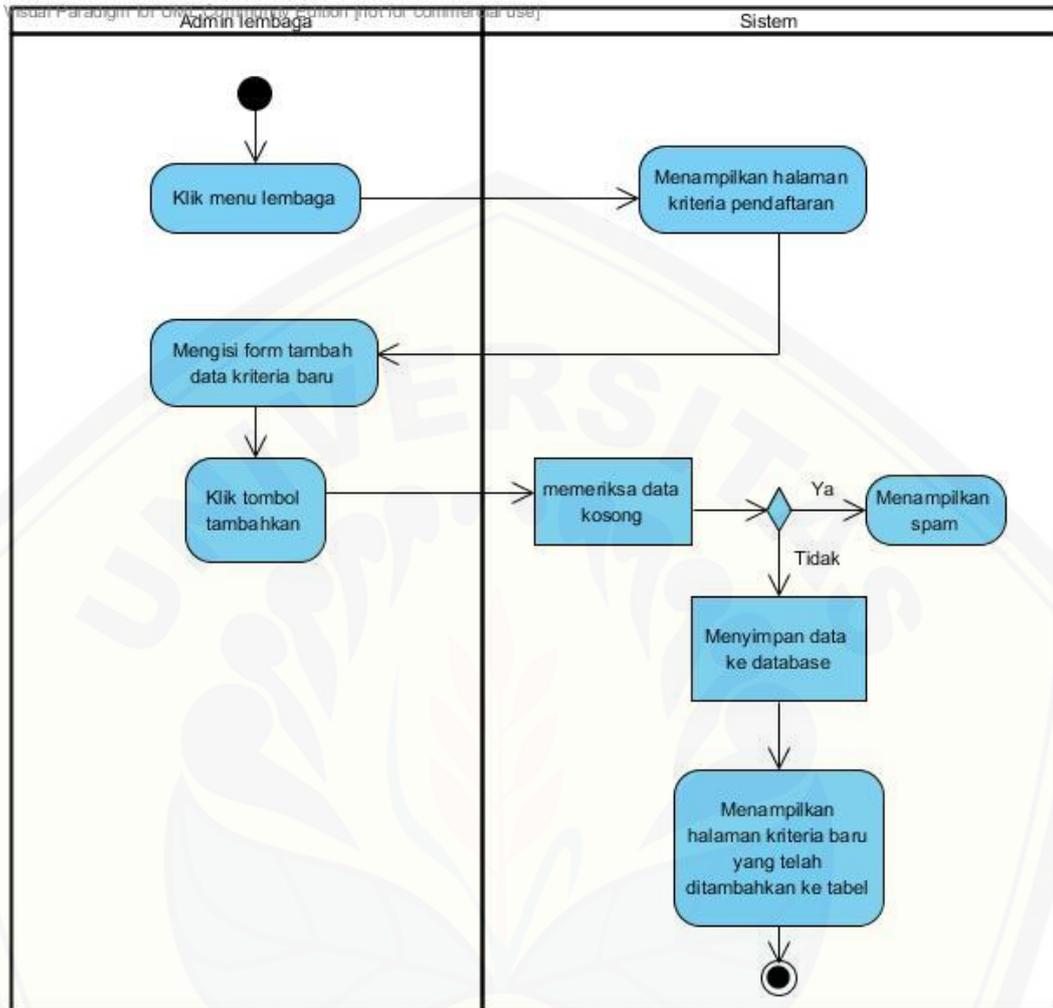
B.4. Activity Diagram Melihat Hasil Rekomendasi



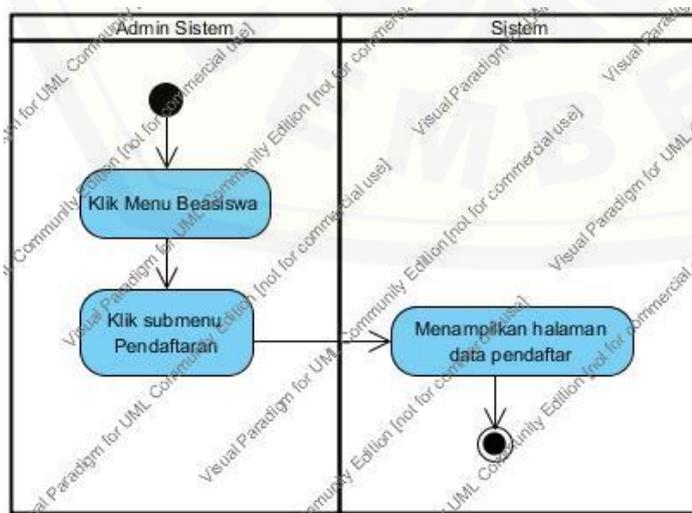
B.5. Activity Diagram Mengelola Data Lembaga



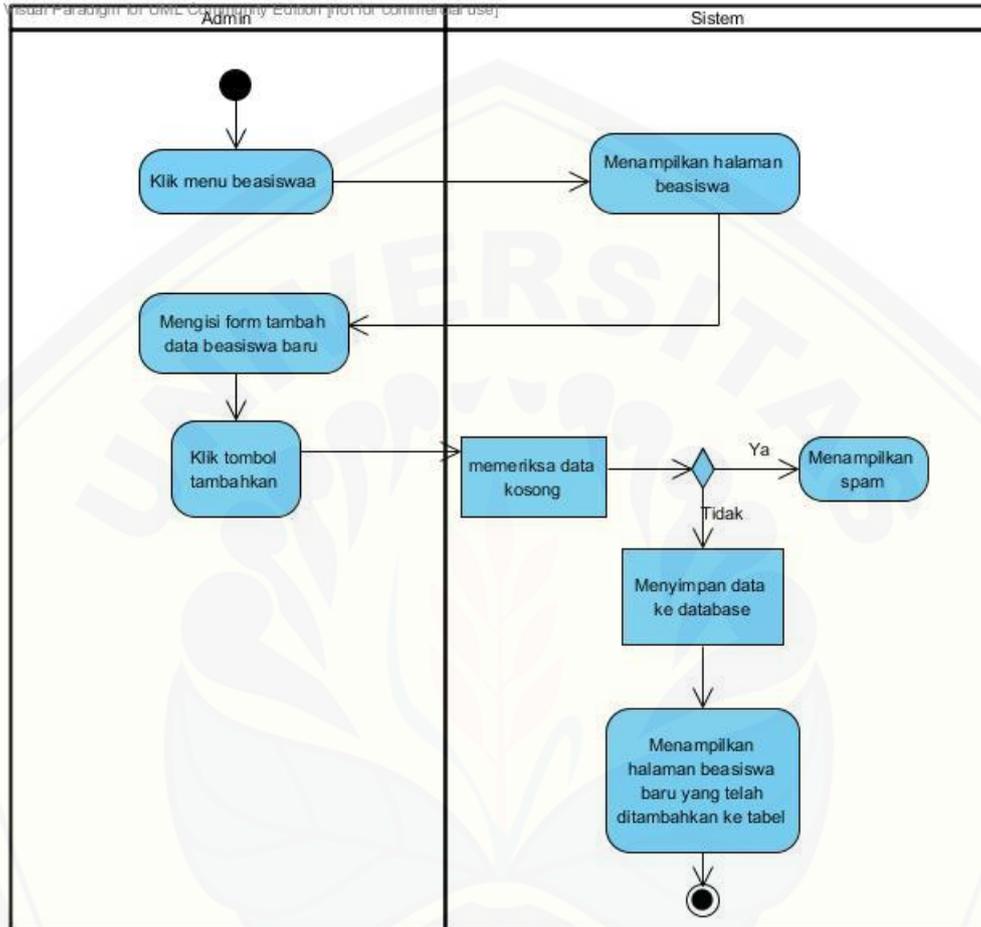
B.6. Activity Diagram Mengelola Kriteria Pendaftaran



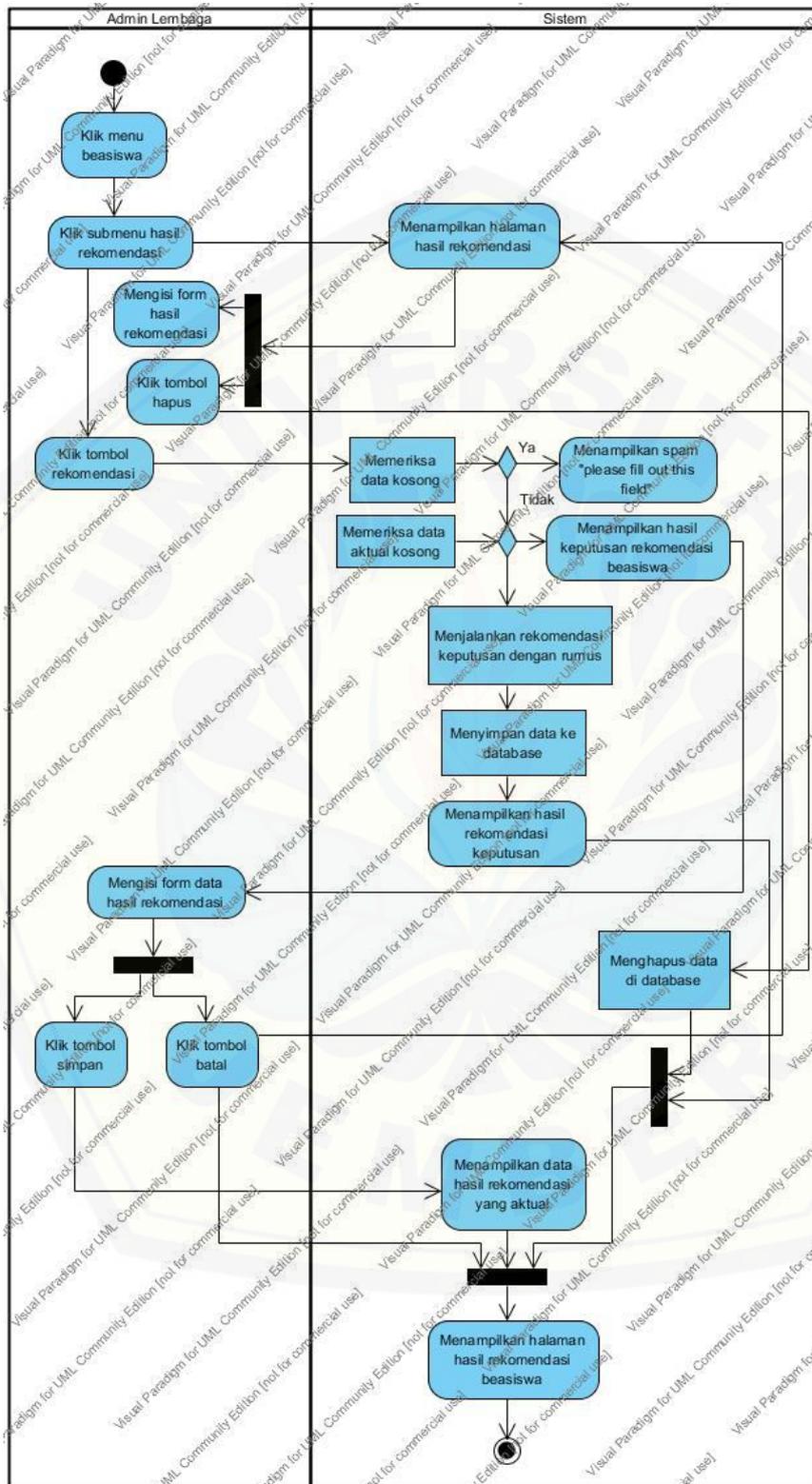
B.7. Activity Diagram Melihat Data Pendaftar



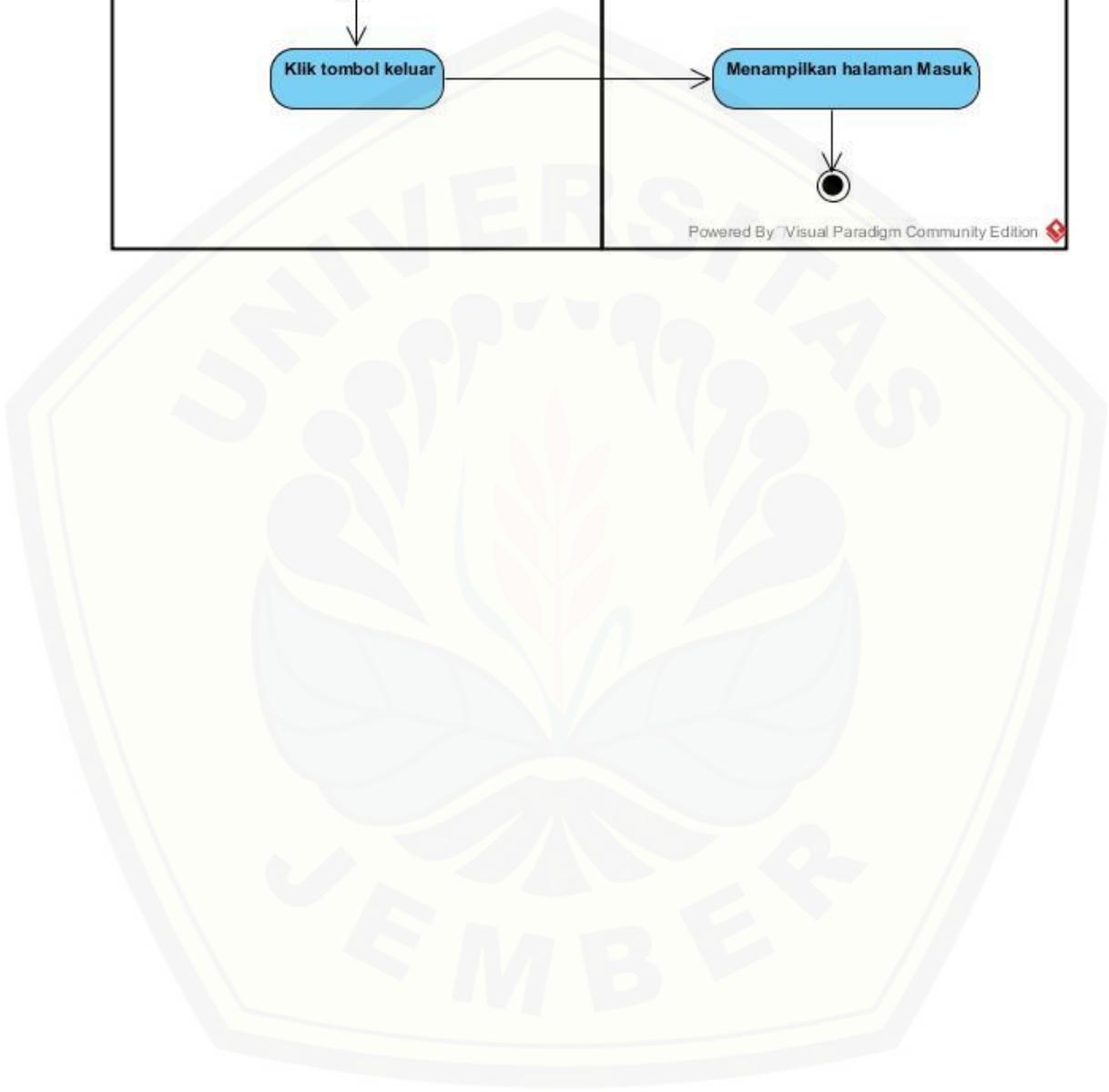
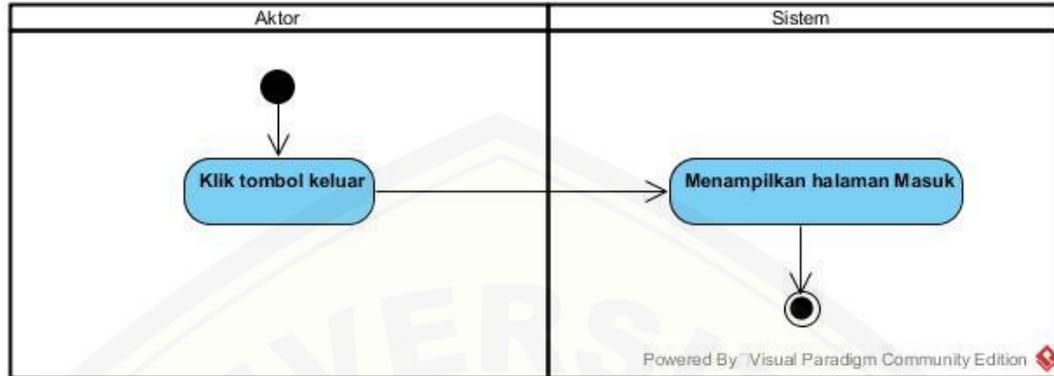
B.8. Activity Diagram Mengelola Data Beasiswa



B.9. Activity Diagram Mengelola Hasil Rekomendasi

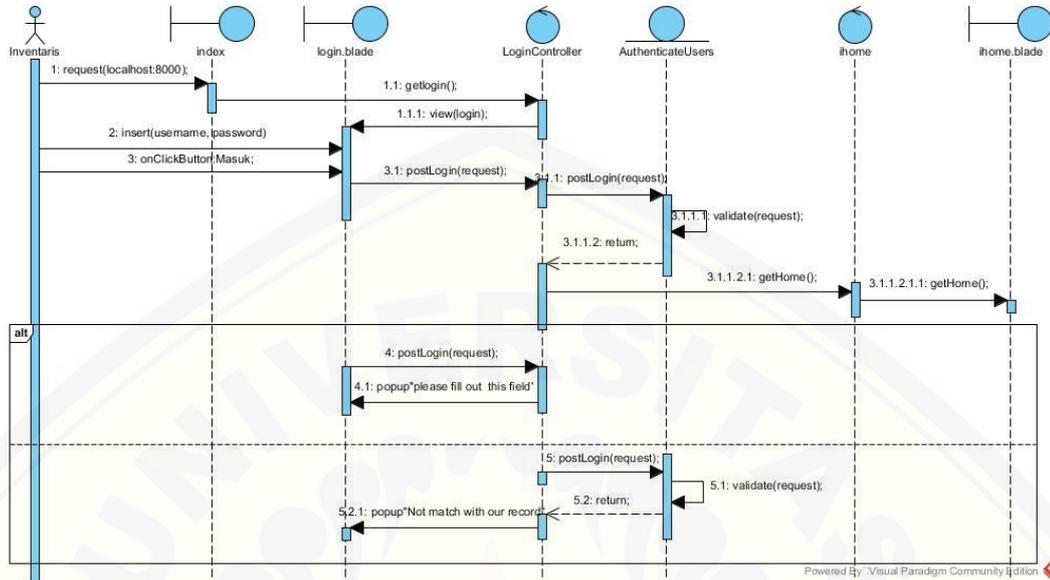


B.10. Activity Diagram Keluar

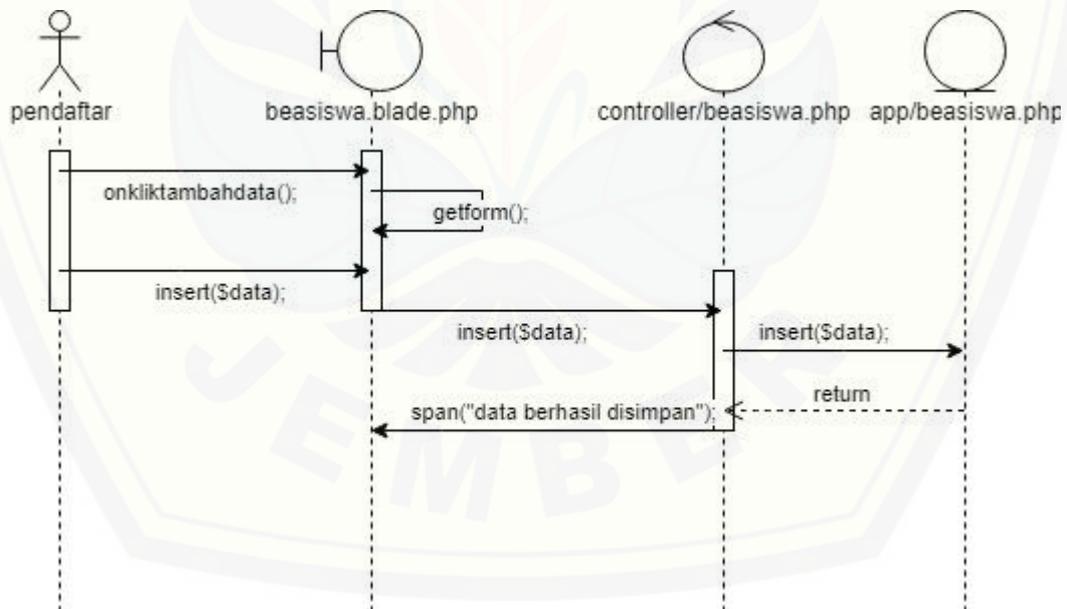


LAMPIRAN C. SEQUENCE DIAGRAM

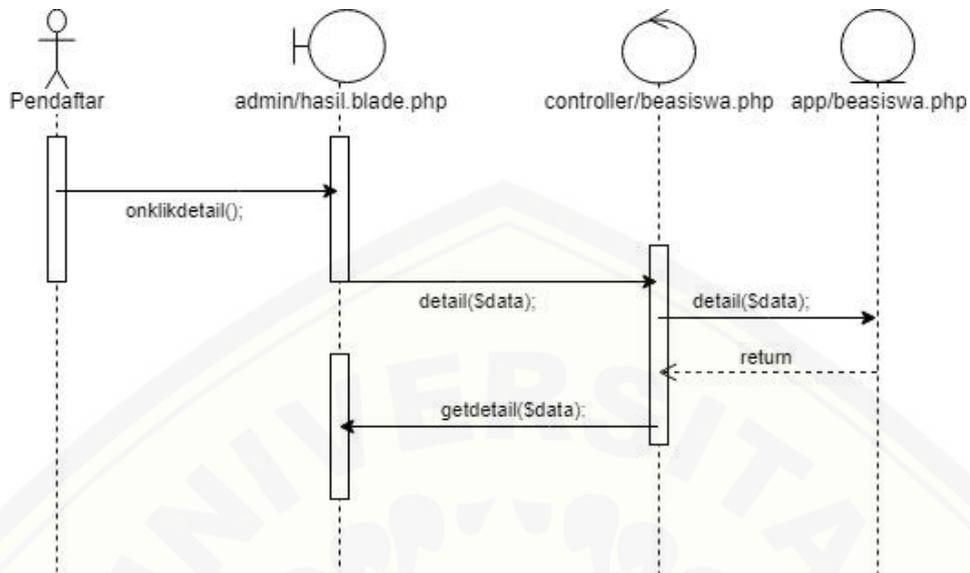
C.1. Sequence Diagram Masuk



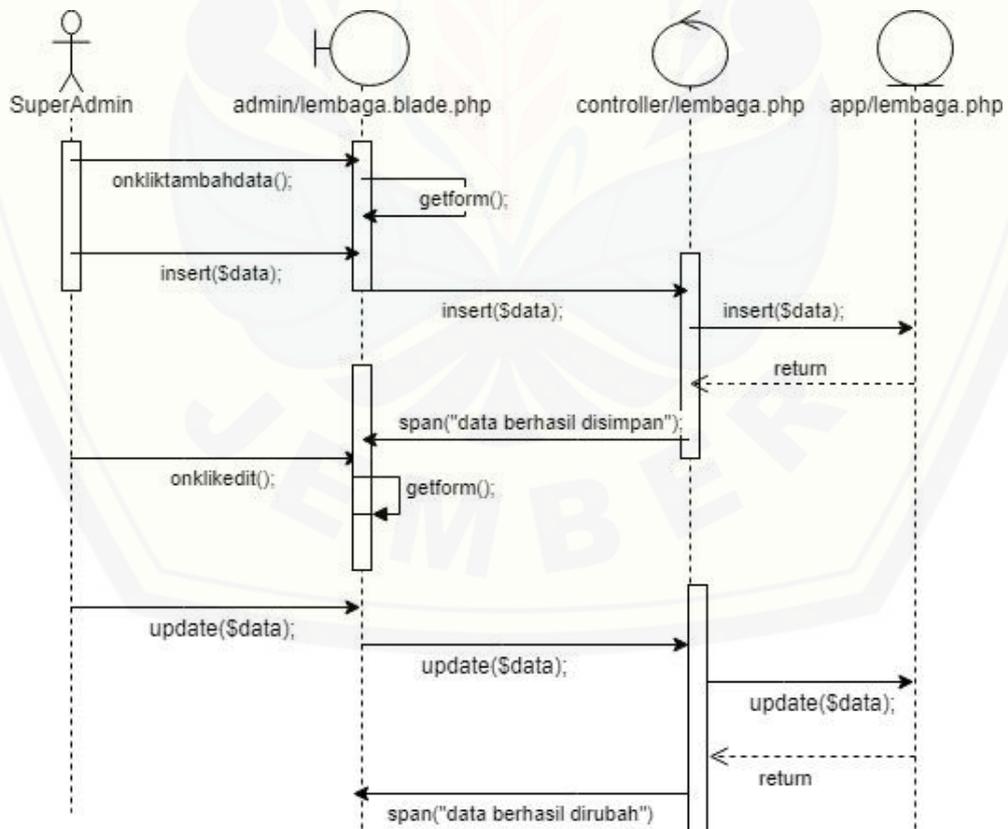
C.2. Sequence Diagram Menambah Data Pendaftaran



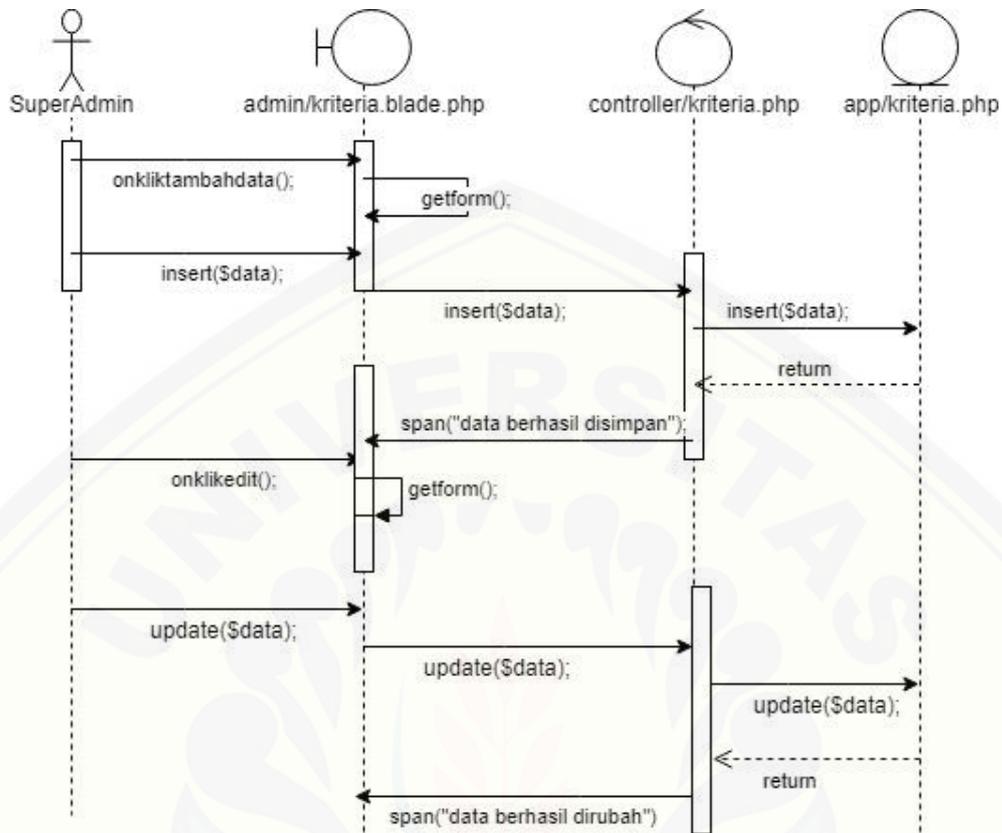
C.3. Sequence Diagram Melihat Hasil Rekomendasi



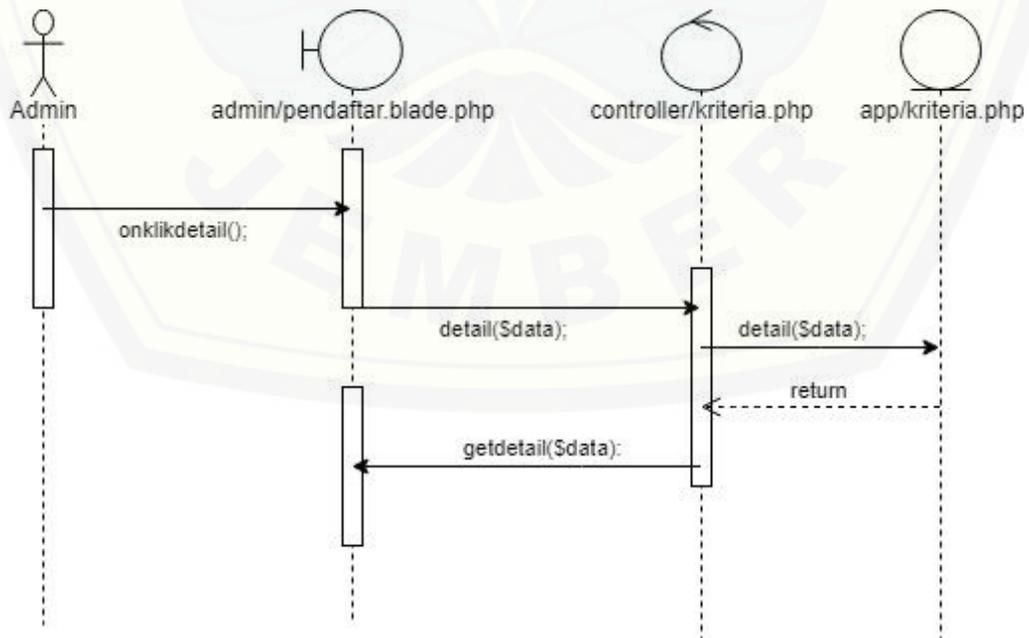
C.4. Sequence Diagram Mengelola Data Lembaga



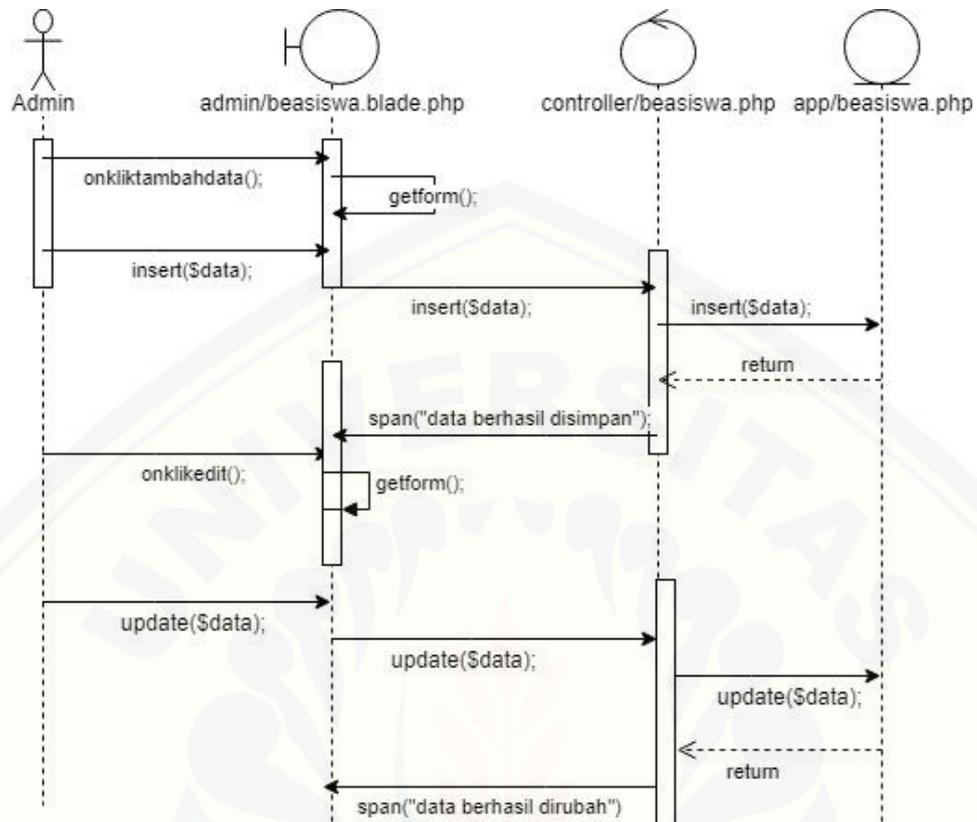
C.5. Sequence Diagram Mengelola Kriteria Pendaftaran



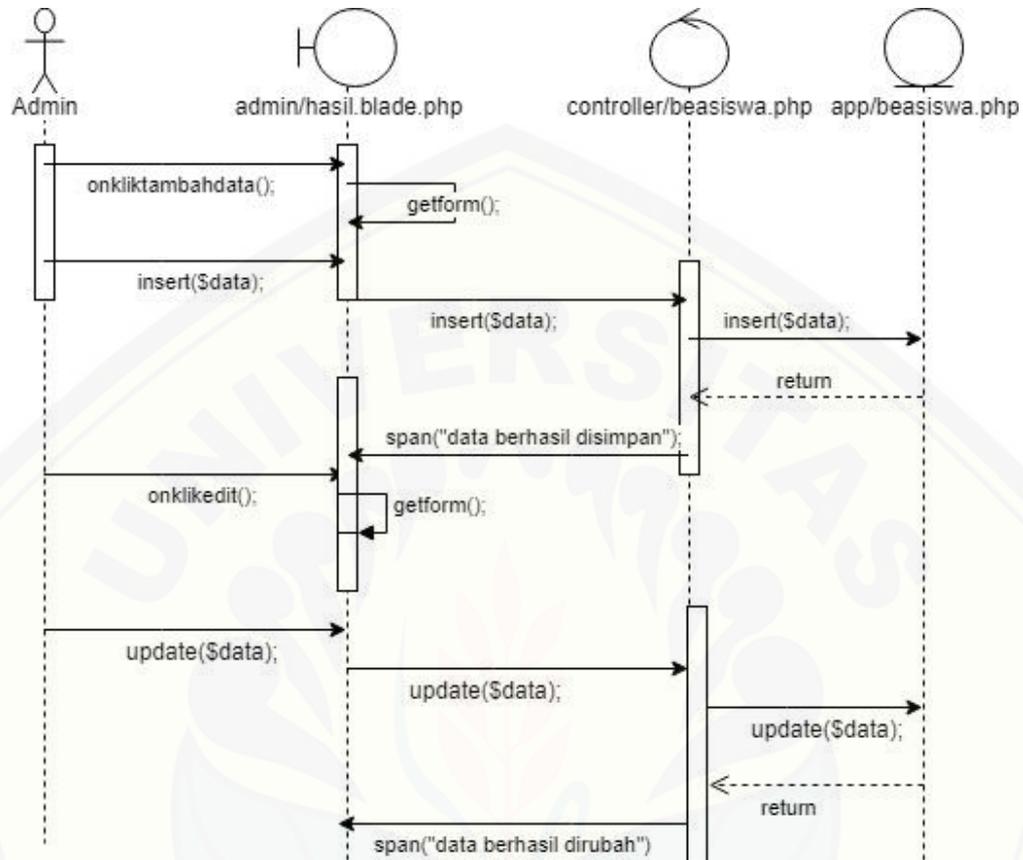
C.6. Sequence Diagram Melihat Data Pendaftaran



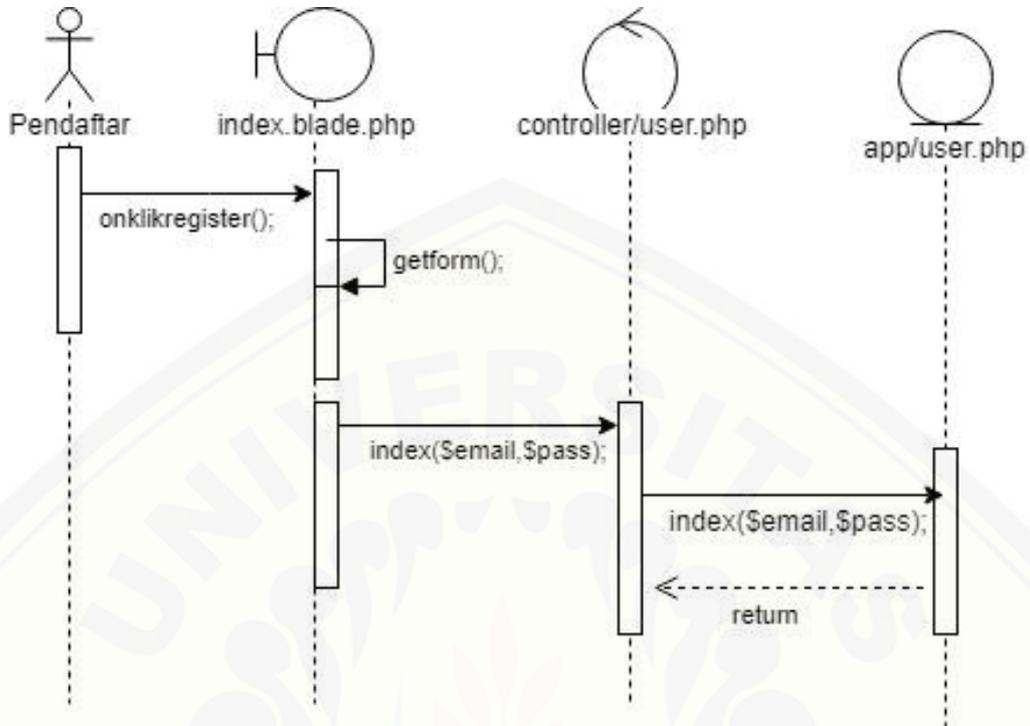
C.7. Sequence Diagram Mengelola Data Beasiswa



C.8. Sequence Diagram Mengelola Hasil Rekomendasi



C.9. Sequence Diagram Menambah Data Pendaftar Baru



C.10. Sequence Diagram Keluar

