



**IMPLEMENTASI LOGIKA *FUZZY* DAN *PROFILE MATCHING*  
PADA SISTEM PENUNJANG KEPUTUSAN SELEKSI  
PENERIMAAN PENGAJAR  
(Studi Kasus: LBB Galileo Jember)**

**SKRIPSI**

Oleh

**Fitriani**

**NIM 122410101047**

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER**

**2017**



**IMPLEMENTASI LOGIKA *FUZZY* DAN *PROFILE MATCHING*  
PADA SISTEM PENUNJANG KEPUTUSAN SELEKSI  
PENERIMAAN PENGAJAR  
(Studi Kasus: LBB Galileo Jember)**

**SKRIPSI**

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

Oleh  
**Fitriani**  
NIM 122410101047

**PROGRAM STUDI SISTEM INFORMASI  
UNIVERSITAS JEMBER  
2017**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang telah memberikan kelancaran dan kemudahan dalam menyelesaikan skripsi ini;
2. Ibu Marfu'ah dan Bapak Rubingan ;
3. Saudara kandungku satu-satunya Vica Izul Zumaroh dan semua sepupuku;
4. Keluarga besar Boyamin dan keluarga besar Abdul Basar ;
5. Seluruh sahabatku seperjuangan yang tidak dapat disebutkan satu persatu;
6. Guru-guruku baik pendidikan formal maupun informal;
7. Almamater Program Studi Sistem Informasi Universitas Jember.

**MOTO**

“Berdoalah (mintalah) kepadaKu (Allah SWT), Pastilah aku kabulkan untukmu”.  
(QS. Al-Mukmin : 60)



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Fitriani

NIM : 122410101047

menyatakan dengan sesungguhnya bahwa karya tulis ilmiah dengan judul “Implementasi Logika *Fuzzy* dan *Profile Matching* pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 19 Mei 2017

Yang menyatakan,

Fitriani

NIM 122410101047

**SKRIPSI**

**IMPLEMENTASI LOGIKA *FUZZY* DAN *PROFILE MATCHING*  
PADA SISTEM PENUNJANG KEPUTUSAN SELEKSI  
PENERIMAAN PENGAJAR  
(Studi Kasus: LBB Galileo Jember)**

Oleh  
**Fitriani**  
**NIM 122410101047**

Pembimbing :

Dosen Pembimbing Utama : Prof. Drs. Slamini, M.Comp.Sc., Ph.D  
Dosen Pembimbing Pendamping : Nelly Oktavia Adiwijaya S.Si., MT.

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Implementasi Logika *Fuzzy* dan *Profile Matching* pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar” telah diuji dan disahkan pada :

hari, tanggal : Jumat, 19 Mei 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

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

NIP.19670420 1992011001

Nelly Oktavia Adiwijaya S.Si., MT.

NIP. 198410242009122008

**PENGESAHAN PENGUJI**

Skripsi berjudul “Implementasi Logika *Fuzzy* dan *Profile Matching* pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar” telah diuji dan disahkan pada :

hari, tanggal : Jumat, 19 Mei 2017

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji,

Penguji I,

Penguji II,

Anang Andrianto, ST., MT  
NIP. 19690615199021002

Fahrobby Adnan, S.Kom,  
NIP. 198706192014041001

Mengesahkan

Ketua Program Studi,

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

NIP. 196704201992011001

## RINGKASAN

**Implementasi Logika *Fuzzy* dan *Profile Matching* pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar (Studi Kasus: LBB Galileo Jember);** Fitriani, 122410101047; 2017; 192 halaman; Program Studi Sistem Informasi Universitas Jember.

UU nomor 20 tahun 2002 tentang Sistem Pendidikan Nasional pada bab 1 disebut orang tua/wali sebagai anggota dari lembaga mandiri dalam pelaksanaan pendidikan, kebanyakan orang tua begitu sibuk dengan rutinitas sehari-hari sehingga peran pengasuhan bergeser menjadi hal sekunder Akhirnya peran ini digantikan oleh lembaga pendidikan dan guru les privat (Widodo, 2015). Galileo menjadi LBB yang sangat di andalkan di kawasan Jember, LBB Galileo berusaha memberikan pelayanan dan fasilitas yang terbaik untuk peserta didik, diantara fasilitas penting tersebut adalah Pengajar, maka hal yang menjadi penting untuk diperhatikan dengan serius adalah proses pemilihan para pengajar.

Masalah umum yang sering terjadi dalam proses pemilihan atau penilaian potensi calon pengajar diantaranya adalah subyektifitas pengambilan keputusan, usaha untuk mengoptimalkan proses penilaian potensi bisa dilakukan dengan perancangan SPK. Mekanisme yang cocok dengan situasi *real* yakni pemilihan pengajar dengan menentukan atribut kriteria dan standar kompetensi pengajar di Lembaga Bimbingan Belajar Galileo Jember adalah *Profile Matching*, yakni sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat *variable* prediktor yang dianggap ideal. Penelitian Muhammad Taufik Irawan (2016) “Penerapan *Profile Matching* untuk Pencarian Siswa SMP Penerima Beasiswa Miskin dan Berprestasi”, dalam penelitian disebutkan bahwa metode *Profile Matching* masih membutuhkan komparasi dengan metode lain. Sebuah prosiding menyebutkan bahwa perhitungan akhir dengan menggunakan *Profile Matching* merupakan nilai yang pasti dan tegas, sehingga mengakibatkan pengukuran dinilai kurang bijaksana pada permasalahan calon yang memiliki potensi tidak jauh beda

(Jumaidi, 2015). Penelitian ini akan dipadukan Logika *Fuzzy* untuk mengatasi persoalan tersebut.



## PRAKATA

Puji syukur kepada Allah SWT, atas segala limpahan rahmat, hidayat dan karuniaNya maka penulis dapat menyelesaikan skripsi yang berjudul “Implementasi Logika *Fuzzy* dan *Profile Matching* pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar”. 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 dan bantuan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Allah SWT, yang telah memberikan kesempatan dan kekuatan yang diberikan sehingga skripsi ini selesai;
2. Ibu Marfu'ah dan Bapak Rubingan atas kesabaran dan dukungan yang diberikan selama ini;
3. Prof. Drs. Slamin, M.Comp.Sc.,Ph.D, selaku Dosen Pembimbing Utama sekaligus ketua Program Studi Sistem Informasi Universitas Jember
4. Nelly Oktavia Adiwijaya S.Si., MT, selaku Dosen Pembimbing Anggota yang telah sabar, memberikan banyak arahan, dan bimbingan dalam penyelesaian skripsi ini;
5. Seluruh dosen dan staf karyawan di Program Studi Sistem Informasi Universitas Jember, yang telah membagi ilmu dan bantuannya selama menempuh pendidikan di perguruan tinggi;
6. Keluarga besar yang selalu memberi dukungan yang luar biasa dan terima kasih untuk segala nasihat, doa, dan perhatian yang selalu diberikan;
7. Lembaga Bimbingan Belajar Galileo yang telah bersedia menerima dan membantu banyak dalam penelitian ini.

8. Sahabat sekaligus teman seperjuangan yang tidak dapat disebutkan satu persatu yang selalu memberi dukungan, doa dan semangat.
9. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis saat pengerjaan skripsi ini;

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

Jember, 19 Mei 2017

Penulis

**DAFTAR ISI**

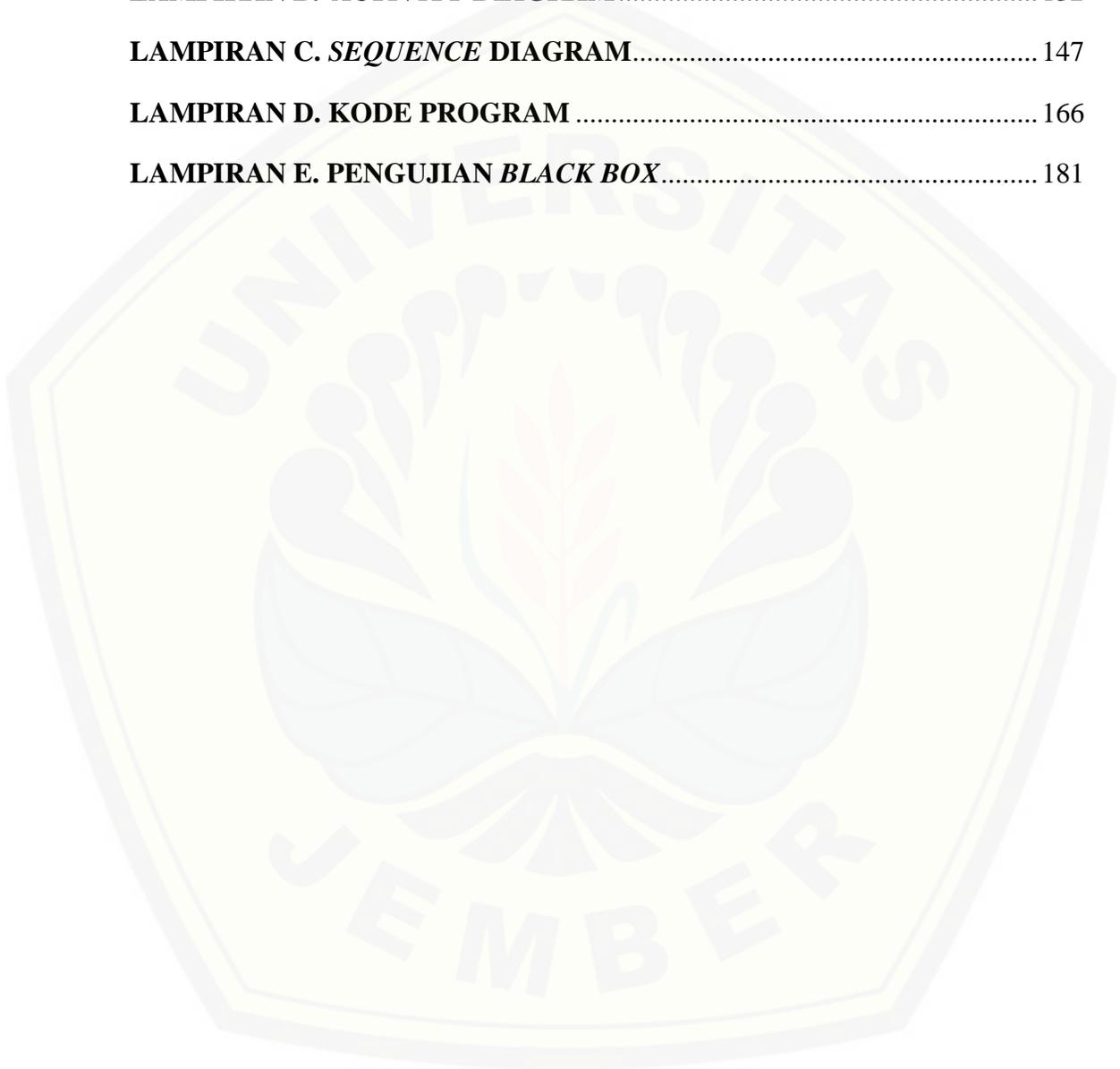
	Halaman
<b>SKRIPSI</b> .....	i
<b>PERSEMBAHAN</b> .....	ii
<b>MOTO</b> .....	iii
<b>PERNYATAAN</b> .....	iv
<b>SKRIPSI</b> .....	v
<b>PENGESAHAN PEMBIMBING</b> .....	vi
<b>PENGESAHAN PENGUJI</b> .....	vii
<b>RINGKASAN</b> .....	viii
<b>PRAKATA</b> .....	x
<b>DAFTAR ISI</b> .....	xii
<b>DAFTAR TABEL</b> .....	xvii
<b>DAFTAR GAMBAR</b> .....	xviii
<b>BAB I. PENDAHULUAN</b> .....	1
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	3
<b>1.3 Tujuan dan Manfaat</b> .....	3
1.3.1 Tujuan.....	3
1.3.2 Manfaat.....	3
<b>1.4 Batasan Masalah</b> .....	4
<b>1.5 Sistematika Penulisan</b> .....	4
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	6

<b>2.1</b>	<b>Penelitian Terdahulu</b> .....	6
<b>2.2</b>	<b>Pengertian Lembaga Bimbingan Belajar</b> .....	6
<b>2.3</b>	<b>Kriteria Penerimaan Pengajar LBB Galileo Jember</b> .....	7
<b>2.4</b>	<b>Decision Support System (DSS)</b> .....	7
2.4.1.	Karakteristik Sistem Pendukung Keputusan.....	7
2.4.2.	Komponen Sistem Pendukung Keputusan.....	8
<b>2.5</b>	<b>Profile Matching</b> .....	8
<b>2.6</b>	<b>Fuzzy Logic</b> .....	11
2.6.1	Himpunan <i>Fuzzy</i> .....	11
2.6.2	Fungsi Keanggotaan.....	12
2.6.3	Sistem Interferensi <i>Fuzzy</i> .....	13
<b>2.7</b>	<b>Pendekatan Fuzzy Logic pada Profile Matching</b> .....	13
<b>BAB 3. METODOLOGI PENELITIAN</b> .....		16
<b>3.1</b>	<b>Jenis Penelitian</b> .....	16
<b>3.2</b>	<b>Tempat dan Waktu Penelitian</b> .....	16
<b>3.3</b>	<b>Tahap Penelitian</b> .....	16
3.4.1	<i>Requirements Definition</i> .....	17
3.4.3	<i>System and Software Design</i> .....	18
3.4.3	<i>Implementation and Unit Testing</i> .....	20
3.4.5	<i>Integration and System Testing</i> .....	20
3.4.5	<i>Maintenance</i> .....	21
<b>BAB 4. PENGEMBANGAN SISTEM</b> .....		22
<b>4.1</b>	<b>Deskripsi Umum Sistem</b> .....	22

4.1.1	SOP ( <i>Statement of purpose</i> ) .....	22
4.1.2	Fungsi sistem.....	22
<b>4.2</b>	<b>Analisis Kebutuhan</b> .....	<b>23</b>
4.3.1.	Kebutuhan Fungsional.....	24
4.3.2.	Kebutuhan Non-Fungsional .....	24
<b>4.3</b>	<b>Desain Sistem</b> .....	<b>25</b>
4.3.1.	Business Process .....	25
4.3.2.	Use Case Diagram.....	26
4.3.3.	<i>Use Case</i> Skenario .....	28
4.3.4	<i>Activity</i> Diagram.....	37
4.3.5	Diagram <i>Sequence</i> .....	42
4.3.6	<i>Class</i> Diagram .....	47
4.3.7	<i>Entity Relationship</i> Diagram (ERD) .....	48
<b>4.4</b>	<b>Implementasi</b> .....	<b>49</b>
<b>4.5</b>	<b>Pengujian</b> .....	<b>52</b>
4.5.1.	Metode <i>White Box</i> .....	52
5)	Metode <i>Black Box</i> .....	61
<b>BAB 5.</b>	<b>HASIL DAN PEMBAHASAN</b> .....	<b>62</b>
<b>5.1</b>	<b>Hasil Penerapan Perhitungan <i>Profile Matching</i> dan Logika <i>Fuzzy</i></b> .....	<b>62</b>
<b>5.2</b>	<b>Hasil Pembuatan Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar Menggunakan Logika <i>Fuzzy</i> dan <i>Profile Matching</i></b> .....	<b>70</b>
5.2.1	Tampilan Awal Sistem .....	70
5.2.2	Fitur <i>Update</i> Mengelola Data Profil User .....	72

5.2.3	Fitur Mengelola data akses.....	74
5.2.4	Fitur Mengelola Nilai Pendaftar.....	75
5.2.5	Fitur Menampilkan Data Profil Pendaftar.....	77
5.2.6	Fitur Menampilkan Data Nilai Pendaftar.....	78
5.2.7	Fitur Mengelola Data Pelajaran.....	79
5.2.8	Fitur Mengelola Kuota Lowongan.....	81
5.2.9	Fitur Mengelola Data Status Pendaftaran Para Pendaftar.....	84
5.2.10	Fitur Menampilkan Data Pendaftar.....	85
5.2.11	Fitur <i>Update</i> Nilai Wawancara dan <i>Microteaching</i> .....	87
5.2.12	Fitur Mengelola Nilai Ideal Kriteria.....	88
5.2.13	Fitur Menyetujui Hasil Penerimaan.....	89
5.2.14	Fitur Melakukan Pendaftaran.....	90
5.2.15	Fitur Logout.....	91
<b>5.3</b>	<b>Implementasi Logika <i>Fuzzy</i> dan <i>Profile Matching</i> pada Seleksi Penerimaan Pengajar.....</b>	<b>92</b>
<b>5.4</b>	<b>Pengujian Sistem Sistem Penunjang Keputusan Seleksi Penerimaan Pengajar Menggunakan Metode <i>Profile Matching</i> dan Logika <i>Fuzzy</i>.....</b>	<b>95</b>
<b>5.5</b>	<b>Pembahasan pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar Menggunakan Logika <i>Fuzzy</i> dan <i>Profile Matching</i>.....</b>	<b>99</b>
5.5.2	Kelebihan Sistem.....	99
5.5.3	Kelemahan Sistem.....	100
<b>BAB 6. PENUTUP.....</b>		<b>101</b>
<b>6.1</b>	<b>Kesimpulan.....</b>	<b>101</b>
<b>6.2</b>	<b>Saran.....</b>	<b>102</b>

<b>DAFTAR PUSTAKA .....</b>	<b>103</b>
<b>LAMPIRAN A. <i>USE CASE</i> SKENARIO.....</b>	<b>105</b>
<b>LAMPIRAN B. <i>ACTIVITY</i> DIAGRAM.....</b>	<b>132</b>
<b>LAMPIRAN C. <i>SEQUENCE</i> DIAGRAM.....</b>	<b>147</b>
<b>LAMPIRAN D. KODE PROGRAM .....</b>	<b>166</b>
<b>LAMPIRAN E. PENGUJIAN <i>BLACK BOX</i>.....</b>	<b>181</b>



## DAFTAR TABEL

	Halaman
Tabel 4. 1 Deskripsi pembagian aktor sistem .....	27
Tabel 4. 2 Deskripsi use case sistem.....	27
Tabel 4. 3 Skenario Mengelola Data Pelajaran.....	31
Tabel 4. 17 Test case function hitungnilai pendaftar() .....	60
Tabel 5. 1 Data Kriteria dan Nilai Standar.....	62
Tabel 5. 2 Tabel input nilai kriteria data test .....	63
Tabel 5. 3 hasil normalisasi nilai kriteria data tes dan nilai minimal.....	64
Tabel 5. 4 hasil pemetaan gap data tes .....	64
Tabel 5. 5 Tabel gap.....	65
Tabel 5. 6 Hasil Konversi nilai gap ke bobot.....	65
Tabel 5. 7 hasil perhitungan pada proses fuzzifikasi .....	66
Tabel 5. 8 hasil pencarian $\alpha$ -predikat untuk data test 1.....	66
Tabel 5. 9 hasil penarikan nilai z data tes 1 .....	68
Tabel 5. 10 hasil penjumlahan $\alpha$ -predikat dan $Z^*$ $\alpha$ -predikat tiap data tes .....	69
Tabel 5. 11 hasil akhir tiap data tes.....	70
Tabel 5. 12 perbandingan nilai akhir yang didapat antara manual dan sistem .....	98
Tabel A. 1 Use case skenario login.....	105
Tabel A. 2 Use case skenario mengelola data profil user .....	105
Tabel A. 3 Use case skenario mengelola data akses .....	108
Tabel A. 4 Use case skenario mengelola nilai pendaftar .....	111
Tabel A. 5 Use case skenario menampilkan data profile pendaftar .....	114
Tabel A. 6 Use Case Skenario menampilkan data nilai pendaftar .....	115
Tabel A. 8 Use case skenario mengelola kuota lowongan.....	117
Tabel A. 9 Use case skenario view mengelola data status pendaftaran .....	122
Tabel A. 10 Use case skenario menampilkan data pendaftar.....	123
Tabel A. 11 Use case skenario update nilai wawancara dan microteaching.....	125

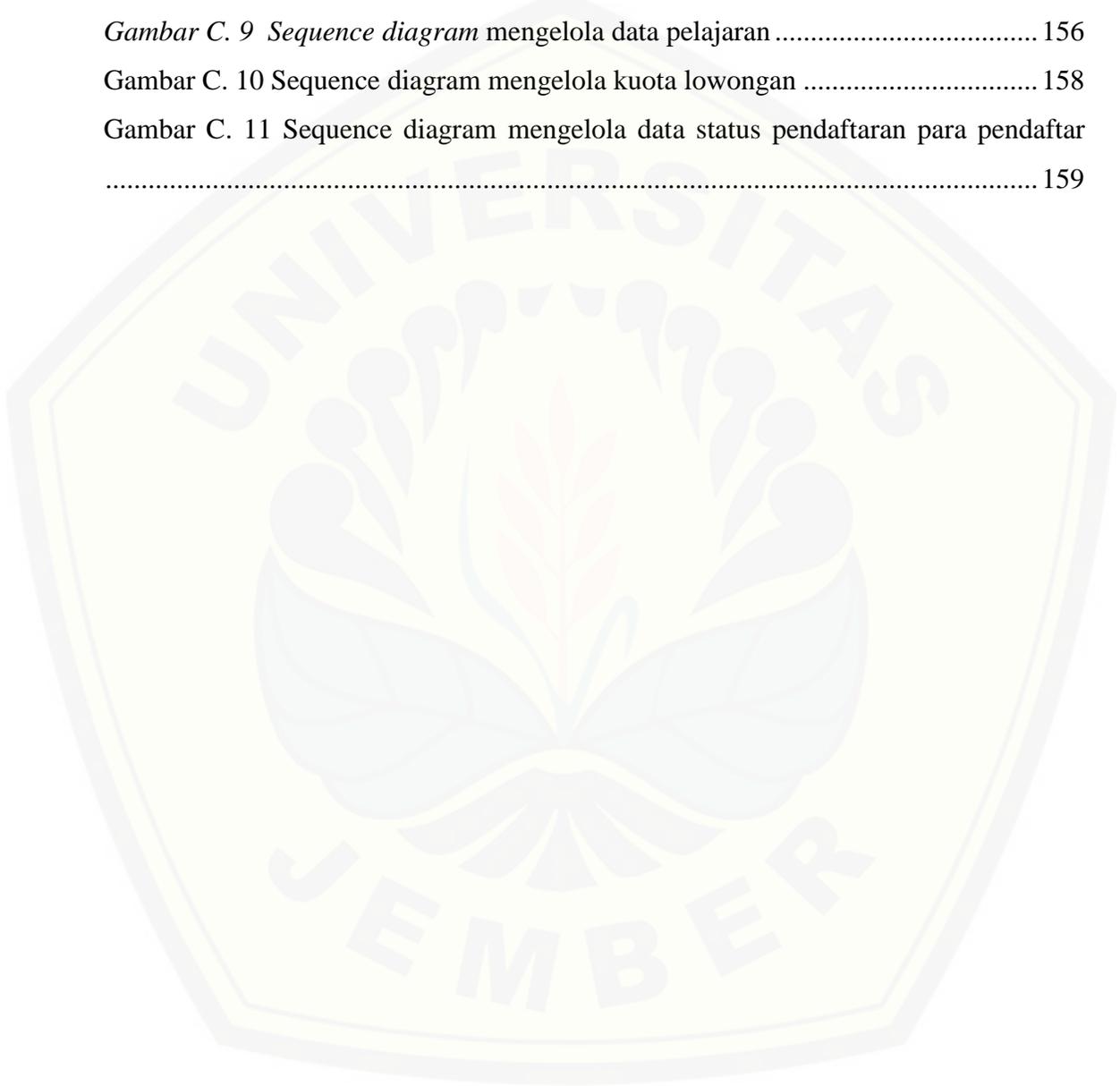
## DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Diagram alir profile matching .....	11
Gambar 2. 2 Representasi kurva segitiga .....	12
Gambar 2. 3 Representasi kurva segitiga .....	12
Gambar 2. 4 Representasi Metode Tsukamoto .....	13
Gambar 2. 5 Diagram alir profile matching .....	15
Gambar 3. 1 Model Waterfall (Sommerville, 2001) .....	17
Gambar 4. 1 Business Process Sistem .....	25
Gambar 4. 2 Use Case Diagram .....	26
Gambar 4. 3 Activity Diagram menampilkan data nilai pendaftar .....	40
Gambar 4. 4 Sequence diagram menampilkan data profile pendaftar .....	44
Gambar 4. 5 Class diagram .....	48
Gambar 4. 6 Entity Relationship Diagram .....	49
Gambar 4. 7 Program function hitungnilai pendaftar(\$var).....	54
Gambar 4. 8 Listing program function hitunggap(\$var).....	55
Gambar 4. 9 Listing program function fuzzyfikasi(\$var) .....	56
Gambar 4. 10 Listing program function hitungratarata().....	58
Gambar 4. 11 Diagram alir function hitungnilai pendaftar(\$var) .....	58
Gambar 4. 12 Diagram alir function hitunggap(\$var) .....	59
Gambar 4. 13 Diagram alir function fuzzyfikasi(\$var).....	59
Gambar 4. 14 Diagram alir function hitungratarata(\$var) .....	59
Gambar 5. 1 Tampilan awal sistem.....	71
Gambar 5. 2 Tampilan halaman login.....	71
Gambar 5. 3 Halaman input data pendaftar .....	72
Gambar 5. 4 Halaman data pembobotan kriteria .....	73
Gambar 5. 5 Halaman data pembobotan kriteria .....	73

Gambar 5. 6 Halaman update pembobotan kriteria .....	74
Gambar 5. 7 Halaman daftar hak akses.....	74
Gambar 5. 8 Halaman tambah hak akses .....	75
Gambar 5. 9 Halaman ubah data akses .....	75
Gambar 5. 10 Halaman daftar nilai pendaftar .....	76
Gambar 5. 11 Halaman mengubah nilai pendaftar.....	76
Gambar 5. 12 Fitur view data profile pendaftar .....	77
Gambar 5. 13 Fitur view detail profile pendaftar.....	77
Gambar 5. 14 Halaman view data pendaftar .....	78
Gambar 5. 15 Halaman view data nilai pendaftar .....	78
Gambar 5. 16 Halaman view detail nilai pendaftar.....	79
Gambar 5. 17 Halaman data mata pelajaran .....	80
Gambar 5. 18 form untuk tambah data pelajaran .....	80
Gambar 5. 19 form untuk mengubah data pelajaran .....	81
Gambar 5. 20 tampilan saat menghapus data pelajaran .....	81
Gambar 5. 21 Halaman untuk mengelola kuota lowongan .....	82
Gambar 5. 22 Halaman tambah lowongan.....	82
Gambar 5. 23 Halaman untuk mengubah data kuota lowongan .....	83
Gambar 5. 24 mengubah status lowongan menjadi ada.....	83
Gambar 5. 25 mengubah status lowongan menjadi tidak ada.....	84
Gambar 5. 26 Halaman pembaruan status pendaftar .....	85
Gambar 5. 27 Halaman untuk menampilkan seluruh data pendaftar .....	85
Gambar 5. 28 Halaman data pendaftar.....	86
Gambar 5. 29 Halaman detail profil pendaftar.....	86
Gambar 5. 30 Halaman detail nilai kriteria pendaftar.....	87
Gambar 5. 31 Halaman list nilai pendaftar .....	87
Gambar 5. 32 Halaman form ubah nilai wawancara dan microteaching .....	88
Gambar 5. 33 Halaman nilai ideal.....	88
Gambar 5. 34 Halaman <i>form</i> ubah nilai ideal .....	89

Gambar 5. 35 Halaman penerimaan.....	90
Gambar 5. 36 Halaman penerimaan.....	90
Gambar 5. 37 Halaman awal sistem.....	91
Gambar 5. 38 Halaman pendaftar peserta.....	91
Gambar 5. 39 view fitur update nilai ideal.....	93
Gambar 5. 40 implementasi perhitungan bobot disistem.....	93
Gambar 5. 41 Implementasi proses fuzzifikasi pada sistem.....	94
Gambar 5. 42 implementasi proses pencarian $\alpha$ -predikat.....	94
Gambar 5. 43 implementasi penarikan nilai Z.....	95
Gambar 5. 44 implementasi proses defuzzifikasi disistem.....	95
Gambar 5. 45 view pemetaan gap.....	96
Gambar 5. 46 view fitur detail perhitungan.....	97
Gambar 5. 47 view daftar pendaftar pada sistem.....	98
Gambar B. 1 Activity Diagram Login.....	132
Gambar B. 2 Activity Diagram Mengelola Data Profil User.....	133
Gambar B. 3 Activity Diagram Mengelola Data Akses.....	134
Gambar B. 4 Activity Diagram Mengelola Nilai Pendaftar.....	135
Gambar B. 5 Activity Diagram Menampilkan data Profile Pendaftar.....	136
Gambar B. 6 Activity Diagram Menampilkan Data Nilai Pendaftar.....	137
Gambar B. 7 Activity Diagram Mengelola Data Pelajaran.....	138
Gambar B. 8 Activity Diagram Skenario Mengelola Kuota Lowongan.....	139
Gambar B. 9 Activity Diagram Mengelola Data Status Pendaftaran.....	140
Gambar B. 10 Activity Diagram Menampilkan Data Pendaftar.....	141
Gambar B. 11 Activity Diagram Update Nilai Wawancara dan <i>Microteaching</i> .....	142
Gambar B. 12 Activity Diagram Mengelola Nilai ideal Kriteria.....	143
Gambar B. 13 Activity Diagram Menyetujui Hasil Penerimaan.....	144
Gambar B. 14 Activity Diagram Logout.....	145
Gambar B. 15 Activity Diagram Logout.....	145
Gambar B. 16 Activity Diagram Melakukan Pendaftaran.....	146

Gambar C. 1 Sequence diagram login (admin).....	147
Gambar C. 2 Sequence diagram login (LITBANG) .....	148
<i>Gambar C. 3 Sequence diagram mengelola data profil user (admin).....</i>	<i>149</i>
<i>Gambar C. 9 Sequence diagram mengelola data pelajaran .....</i>	<i>156</i>
Gambar C. 10 Sequence diagram mengelola kuota lowongan .....	158
Gambar C. 11 Sequence diagram mengelola data status pendaftaran para pendaftar .....	159



## **BAB I. PENDAHULUAN**

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

### **1.1 Latar Belakang**

Pendidikan adalah jalan untuk seseorang mencapai kesuksesan, apabila tanggung jawab pendidikan diletakkan pada sekolah hal itu tidaklah cukup, maka pada UU nomor 20 tahun 2002 tentang Sistem Pendidikan Nasional pada bab 1 disebut orang tua/wali sebagai anggota dari lembaga mandiri dalam pelaksanaan pendidikan. Kebanyakan orangtua ingin anaknya sukses, namun kenyataannya banyak orangtua sibuk dengan rutinitas sehari-hari sehingga peran pengasuhan bergeser menjadi hal sekunder. Akhirnya peran ini digantikan oleh lembaga pendidikan dan guru les privat (Widodo, 2015). Artikel yang sama juga menyebutkan, berdasarkan hasil jajak pendapat Kompas pada 2-4 Desember 2015, sebanyak 144 responden (62 persen) dari 229 responden yang mempunyai anak sekolah menyatakan masih memberikan tambahan les pelajaran, sehingga dapat ditarik kesimpulan kebutuhan akan lembaga bimbingan belajar sudah dianggap penting perannya dikalangan masyarakat umum.

Lembaga bimbingan belajar Galileo Jember adalah salah satu lembaga yang ada di kota Jember, jumlah peserta didik terus menerus mengalami peningkatan, terutama pada peserta persiapan seleksi PTN dalam kurun waktu 3 tahun tak kurang dari 186 peserta tiap tahunnya, sedang peserta regular mencapai angka kurang lebih 80 peserta. Tiap tahunnya peserta didik yang mendaftar pada lembaga bimbingan belajar Galileo terus mengalami peningkatan, hal tersebut menjadi bukti bahwa masyarakat puas dengan layanan bimbingan yang diberikan. Galileo berusaha memberikan pelayanan dan fasilitas yang terbaik untuk peserta didik, diantara fasilitas penting tersebut adalah Pengajar. Atribut penting dalam keberhasilan bimbel adalah

keberhasilan para pengajar dalam mentransfer materi pelajaran kepada para peserta, maka hal yang menjadi penting untuk diperhatikan dengan serius adalah proses pemilihan para pengajar, sehingga diperoleh pengajar yang memiliki kompetensi dalam mengajar.

Masalah umum yang sering terjadi dalam proses pemilihan atau penilaian potensi calon pengajar diantaranya adalah subyektifitas pengambilan keputusan, yang akan semakin terasa terutama jika beberapa calon pengajar memiliki potensi yang tak jauh beda. Usaha untuk mengoptimalkan proses penilaian potensi bisa dilakukan dengan perancangan SPK. Sistem Penunjang Keputusan atau *Decision Support System* adalah “sistem berbasis komputer yang interaktif, membantu pengambilan keputusan memanfaatkan data dan model untuk menyelesaikan masalah masalah tak terstruktur” (Subakti, 2002 ). Sistem tidak akan mutlak menjadi pengambil keputusan, melainkan hanya membantu sebagai pertimbangan bagi pengambil keputusan, namun bagaimana seorang pengambil keputusan dapat mempertanggungjawabkan keputusannya, SPK adalah hal yang dapat diandalkan untuk masalah tersebut.

*Profile Matching* adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat *variable* prediktor yang dianggap ideal. Mekanisme tersebut cocok dengan situasi *real* di Lembaga Bimbingan Belajar Galileo dimana terdapat standar kompetensi seorang pengajar haruslah dipenuhi seorang pendaftar. Nilai ideal pada tiap variabel ditentukan dari musyawarah tim LITBANG LBB Galileo Jember. Menurut penelitian yang terdahulu “Penerapan *Profile Matching* untuk Pencarian Siswa SMP Penerima Beasiswa Miskin dan Berprestasi”, dalam penelitian disebutkan bahwa metode *Profile Matching* masih membutuhkan komparasi dengan metode lain (Muhammad Taufik Irawan, 2016). Sebuah prosiding menyebutkan bahwa perhitungan akhir dengan menggunakan *Profile Matching* merupakan nilai yang pasti dan tegas, sehingga mengakibatkan pengukuran dinilai kurang bijaksana pada permasalahan calon yang memiliki potensi tidak jauh beda (Jumaidi, 2015). Maka pada Penelitian ini akan dipadukan logika

*Fuzzy* untuk mengatasi persoalan tersebut. Perpaduan dua metode yakni *Profile Matching* dan Logika *Fuzzy* diharapkan dapat menghasilkan keputusan yang akurat, tidak kaku, tidak hanya “layak” dan “tidak layak” saja tetapi dapat menghasilkan kategori penerimaan yang lebih detail.

## 1.2 Rumusan Masalah

Berdasarkan dari beberapa permasalahan yang telah diuraikan diatas, maka dapat diambil rumusan masalah sebagai berikut :

- a. Bagaimana mengimplementasikan logika *Fuzzy* dan metode *Profile Matching* di Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar?
- b. Bagaimana merancang dan membangun sebuah sistem pendukung keputusan seleksi penerimaan pengajar?

## 1.3 Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

### 1.3.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

- a. Mengimplementasikan logika *Fuzzy* dan metode *Profile Matching* di Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar di LBB Galileo Jember.
- b. Merancang dan membangun sebuah Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar di LBB Galileo Jember.

### 1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Bagi Instansi

Sebagai sarana yang dapat membantu bagi instansi dalam proses seleksi pengajar, sebagai pertimbangan pengambil keputusan LITBANG dalam

menentukan pengajar yang diterima agar sesuai kompetensi yang diharapkan instansi sehingga dapat meningkatkan kualitas belajar mengajar dan sebagai sarana bantu dokumentasi yang jelas saat proses pemilihan pengajar.

b. Bagi Akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.

c. Bagi Peneliti

Mengetahui bagaimana proses penerapan *Fuzzy Logic* dan *Profile Matching* pada sistem pendukung keputusan seleksi penerimaan pengajar.

d. Bagi pihak lain

Penelitian ini dapat dijadikan bahan referensi bagi peneliti lain yang ingin mengembangkan hasil penelitian ini di kemudian hari.

#### 1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat sebagai parameter pengerjaan penelitian ini diantaranya sebagai berikut :

- a. Sistem yang dibangun berbasis *web*.
- b. Sistem ini hanya digunakan untuk seleksi penerimaan pengajar di LBB Galileo Jember.
- c. Satu kali proses pendaftaran hanya dapat memilih satu mata pelajaran.
- d. Sistem merengking seluruh data pendaftaran menggunakan penerapan logika *Fuzzy* dan metode *Profile Matching*

#### 1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi setelah Bab 1. Pendahuluan adalah sebagai berikut:

a. Bab 2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep sistem pendukung keputusan, dan kajian teori metode yang berkaitan dengan masalah dalam penelitian.

b. Bab 3. Metodologi Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

c. Bab 4. Pengembangan Sistem

Bab ini berisi uraian tentang tentang perancangan desain sistem. Perancangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, kemudian merancang *business process*, *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram* (ERD).

d. Bab 5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

e. Bab 6. Penutup

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

## BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

### 2.1 Penelitian Terdahulu

Penelitian terdahulu mengenai penerapan *profile matching* untuk pencarian siswa SMP penerima beasiswa miskin dan berprestasi, bahwa metode tersebut dapat digunakan untuk melakukan seleksi, namun disarankan untuk melakukan komparasi dengan metode lain (Muhammad Taufik Irawan, 2016).

Penelitian yang lain yakni pendekatan logika *fuzzy* untuk perhitungan *profile matching* dalam menentukan kelayakan proposal di Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung menyebutkan bahwa perhitungan *profile matching* menghasilkan perhitungan tegas dan kurang bijak, dan menyimpulkan bahwa perhitungan menggunakan logika *fuzzy* menghasilkan keputusan yang lebih bijak dan dapat menghindari perhitungan yang tegas (Jumaidi, 2015).

Penelitian kali ini akan diimplementasikan logika *fuzzy* dan *profile matching* pada sistem penunjang keputusan seleksi penerimaan pengajar dengan studi kasus di Lembaga Bimbingan Belajar Galileo Jember, dengan data utama yaitu data kriteria penerimaan.

### 2.2 Pengertian Lembaga Bimbingan Belajar

Lembaga berarti tempat atau wadah, bimbingan belajar adalah suatu proses pemberian bantuan dari guru pembimbing kepada siswa dengan cara mengembangkan suasana belajar yang kondusif dan menumbuhkan kemampuan agar siswa terhindar dari dan atau dapat mengatasi kesulitan belajar yang mungkin dihadapinya sehingga mencapai hasil belajar yang optimal (Dr.Suherman, 2001). Konsep bimbingan belajar adalah mengatasi permasalahan yang dihadapi pelajar dengan bantuan pembimbing. Lembaga Bimbingan Belajar adalah satuan pendidikan

yang diatur di UU Nomor 20 tahun 2003 bab 1 dan poin 10, sebagai salah satu inovasi dari pemerintah dalam usaha pemerataan pendidikan, guna mencapai tujuan penyelenggaraan pendidikan.

### **2.3 Kriteria Penerimaan Pengajar LBB Galileo Jember**

Kriteria penerimaan pengajar merupakan ukuran untuk calon pengajar supaya diterima menjadi Pengajar di LBB Galileo. Kriteria penerimaan dimunculkan sebagai usaha untuk mendapatkan pengajar yang dapat melakukan bimbingan kepada peserta didik dengan baik. Seorang pengajar LBB Galileo harus sarjana pendidikan yang mengajar sesuai dengan bidang keahliannya yang tertera pada ijazah. Kemampuan pengajar dalam menguasai materi dilihat di ijazah, dimana standar minimal yang ditentukan oleh LBB Galileo nilai minimal IPK adalah 3.0 dan lulusan PTN, dilakukan juga *microteaching test*, serta test tulis, dan wawancara oleh tim LITBANG untuk dapat mengetahui kemampuan dan kesungguhan calon pengajar.

### **2.4 Decision Support System (DSS)**

Sistem pendukung keputusan merupakan kumpulan dari aplikasi perangkat lunak dan perangkat keras yang saling terintegrasi yang merupakan tulang punggung dari sebuah proses pengambilan keputusan. (Waghmode & P.P.Jamsandekar, 2004). Sistem Pendukung Keputusan juga didefinisikan sebagai penggabungan sumber-sumber kecerdasan individu dengan kemampuan untuk memperbaiki kualitas keputusan.

#### **2.4.1. Karakteristik Sistem Pendukung Keputusan**

Dari pengertian Sistem Pendukung Keputusan maka dapat ditentukan karakteristik sebagai berikut:

1. Mendukung proses pengambilan keputusan, menitik beratkan pada *management by perception*

2. Dalam adanya interface manusia / mesin dimana manusia (*user*) tetap memegang control proses pengambilan keputusan
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur dan tak terstruktur
4. Memiliki kapasitas untuk memperoleh informasi sesuai dengan kebutuhan
5. Memiliki subsistem – subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan item
6. Membutuhkan struktur data komprehensif yang sapat melayani kebutuhan informasi seluruh tingkat manajemen

#### 2.4.2. Komponen Sistem Pendukung Keputusan

1. Data Management, termasuk *database*, yang mengandung data relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS).
2. Model *Management* melibatkan model *financial*, statistika, *Management science*, atau berbagai model kuantitatif lainnya , sehingga dapat sistem memiliki suatu kemampuan analisis, dan manajemen software yang diperlukan.
3. *Communication* (dialog sistem), *user* dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem atau menyediakan antarmuka.

## 2.5 Profile Matching

*Profile Matching* adalah sebuah metode untuk menentukan kelayakan, “*Profile Matching* secara garis besar merupakan proses membandingkan antara kompetensi individu ke dalam kompetensi jabatan sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap), semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar yang berarti memiliki peluang lebih besar untuk karyawan menempati posisi “ (Andreas Handojo, 2003). Proses pada metode *Profile*

*Matching*, diawali dengan menentukan variable data dan aspek-aspek yang dibutuhkan dalam penilaian, dilanjutkan dengan pemetaan gap yang didahului dengan pendefinisian nilai ideal untuk setiap *variable*. Selisih setiap nilai data *testing* terhadap nilai ideal masing-masing *variable* merupakan gap,

$$\text{Gap} = \text{profile minimal-profile data test} \quad (1).$$

Kemudian nilai selisih yang dihasilkan atau disebut gap konversi ke nilai bobot. Nilai bobot diambil dari tabel bobot sesuai dengan nilai gap yang didapat. Bobot setiap variabel akan kelompok menjadi *variable Core Factor* (CF) dan *Secondary Factor* (SF). *Core Factor* (CF) adalah kriteria yang paling penting atau menonjol atau paling dibutuhkan dalam penilaian, sedangkan *Secondary Factor* (SF) adalah merupakan kriteria pendukung selain yang ada pada *Core Factor*, lalu dari pengelompokan tersebut akan dihitung sesuai dengan langkah berikut

Nilai rata-rata *Core Factor*

$$NCF = \frac{\sum NC}{\sum IC} \quad (2)$$

Keterangan :

NCF : Nilai rata-rata *Core Factor*

NC : Jumlah total nilai *Core Factor*

IC : Jumlah item *Core Factor*

Nilai rata-rata *Secondary Factor*

$$NSF = \frac{\sum NS}{\sum IS} \quad (3)$$

NSF : Nilai rata-rata *Secondary Factor*

NS : Jumlah total nilai *Secondary Factor*

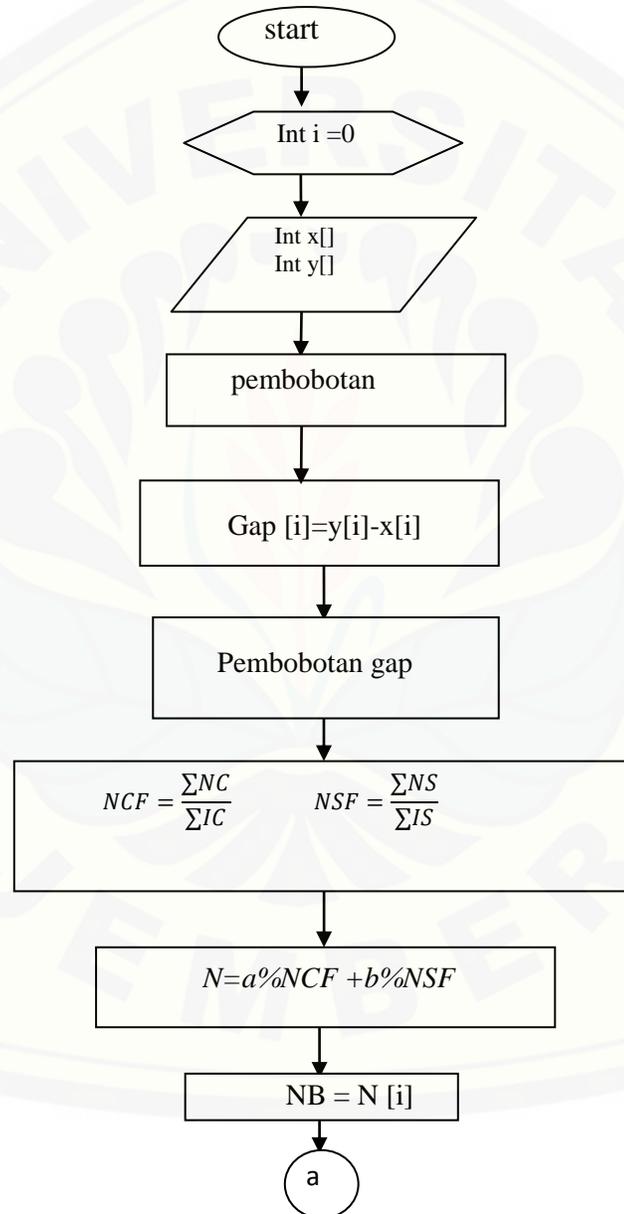
IS : Jumlah item *Secondary Factor*

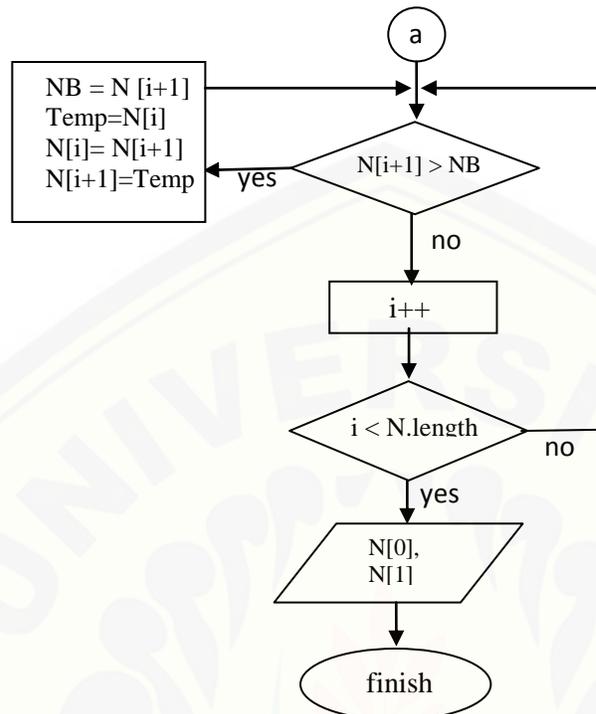
Tahap terakhir dari metode ini, adalah proses akumulasi nilai CF dan SF berdasarkan nilai-nilai variabel data.

$$\text{Nilai akhir} = (x)\%NCF + (y)\%NSF \quad (4).$$

Metode *Profile Matching* dianggap cocok ketika didapati banyak kriteria dan nilai ideal, dan dianggap pas untuk manajemen sumberdaya, dimana kompetensi kebutuhan akan ditentukan terlebih dahulu.

Adapun diagram alir dari perhitungan *Profile Matching* adalah sebagai berikut:



Gambar 2. 1 Diagram alir *profile matching*

Keterangan:

Int x[] : nilai ideal

Int y[] : data alternative

NCF : hasil perhitungan core factor

HSF : hasil perhitungan secondary factor

N : nilai total

NB : nilai terbesar

## 2.6 Fuzzy Logic

Logika *fuzzy* dikembangkan sehubungan dengan ketidakpastian sifat alamiah manusia, guna mensimulasikan proses pertimbangan yang terdapat aturan-aturan yang dianggap dalam keadaan normal (Sri Kusumadewi, 2004).

### 2.6.1 Himpunan *Fuzzy*

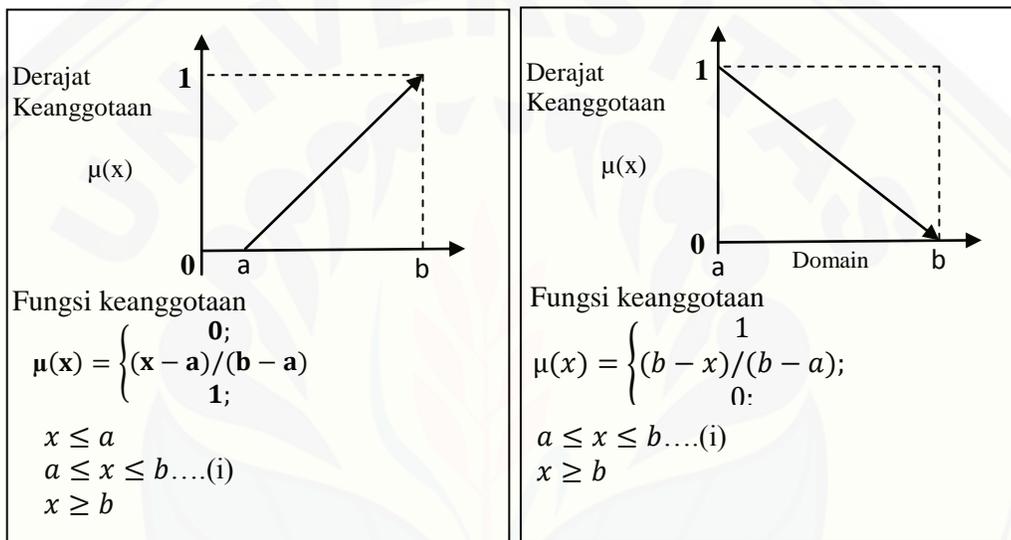
Himpunan *Fuzzy* pada dasarnya adalah gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian sehingga fungsi tersebut akan mencakup bilangan real pada interval  $[0,1]$ . Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar dan masih

ada nilai-nilai yang terletak antara benar dan salah, sehingga dalam logika *Fuzzy* besar dari kebenaran dan kesalahan suatu nilai tergantung pada bobot keanggotaan yang dimiliki.

2.6.2 Fungsi Keanggotaan

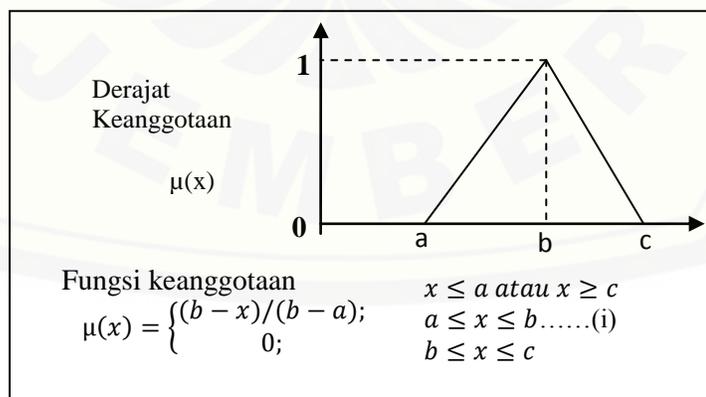
Sebuah kurva menunjukkan pemetaan titik input kedalam nilai keanggotaannya dimana interval yang dimiliki adalah 0 sampai 1

Representasi Linier



Gambar 2. 2 Representasi kurva segitiga

Representasi Kurva Segitiga



Gambar 2. 3 Representasi kurva segitiga

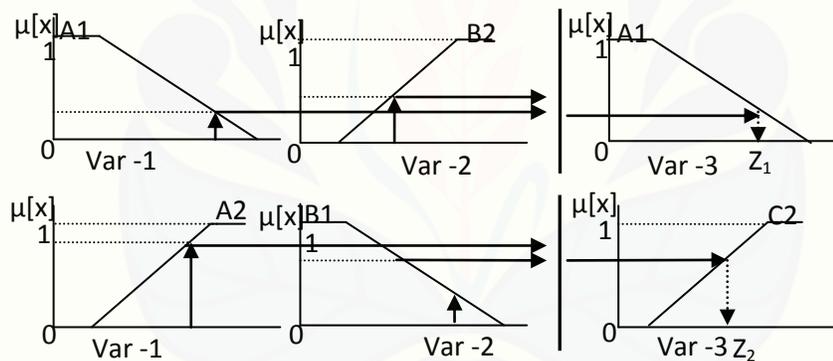
### 2.6.3 Sistem Interferensi *Fuzzy*

#### Metode penalaran Monoton

Metode penalaran Monoton adalah dasar untuk penggunaan teknik implikasi *Fuzzy*, jika dua daerah di relasikan dengan implikasi sederhana maka sistem *Fuzzy* berjalan tanpa proses komposisi dan dekomposisi *Fuzzy*. Nilai *output* dapat diestimasi secara langsung dari derajat keanggotaan yang berhubungan dengan antesedennya.

#### Metode Tsukamoto

Metode Tsukamoto ini dimana setiap konsekuen pada aturan yang berbentuk *If – Then* harus dipresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya *output* dari referensi tiap tiap aturan diberikan secara tegas berdasarkan  $\alpha$ -predikat, direpresentasikan pada kurva berikut :



Gambar 2. 4 Representasi Metode Tsukamoto

Kemudian nilai  $z$  yang dihasilkan akan di defuzzifikasi atau ditarik penegasan, dengan cara di hitung rata-rata dengan menggunakan rata-rata terbobot.

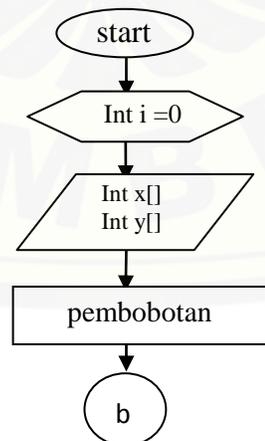
Rata-rata terbobot adalah

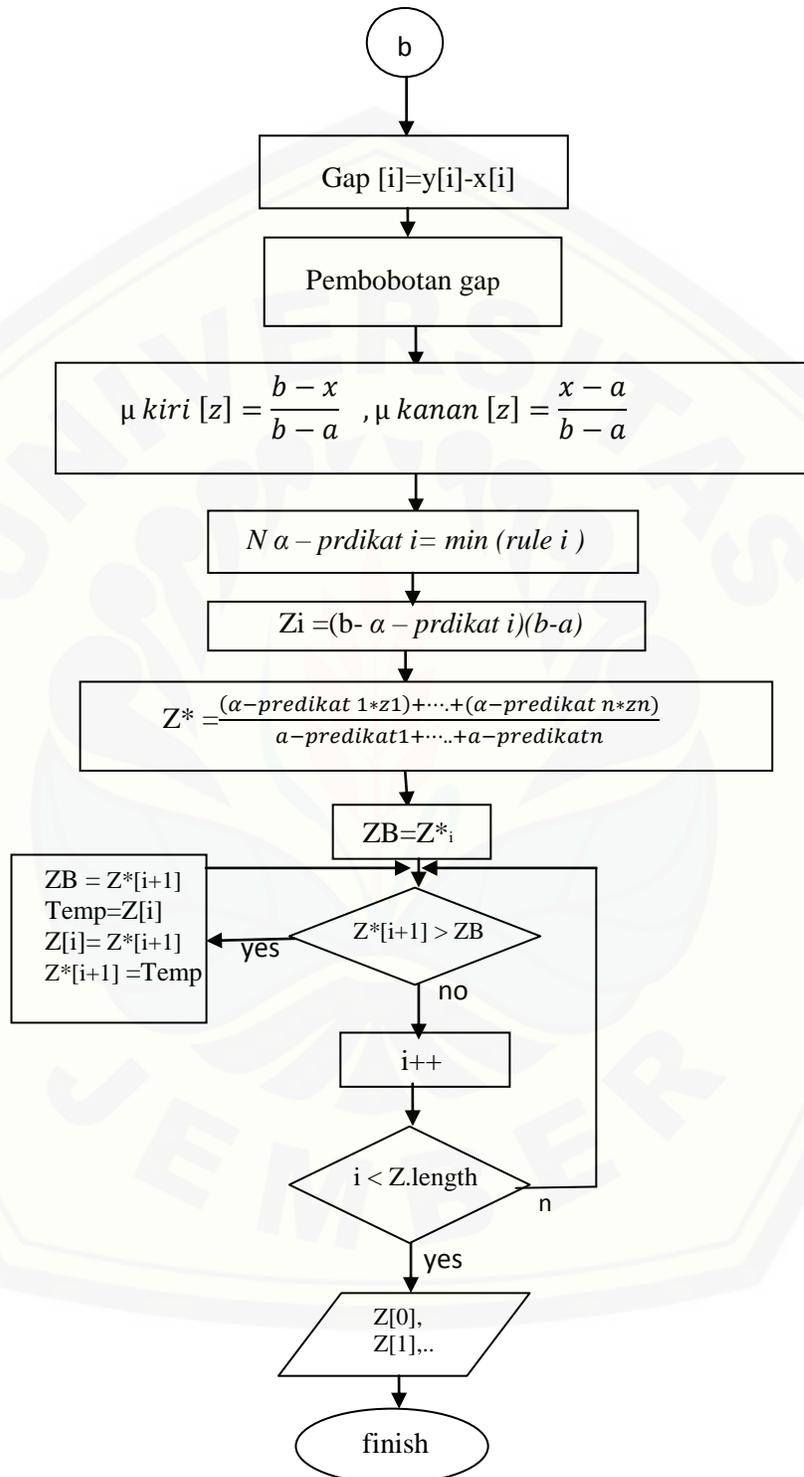
$$Z = \frac{a_1 z_1 + a_2 z_2}{a_1 + a_2}$$

## 2.7 Pendekatan *Fuzzy Logic* pada Profile Matching

Pendekatan *Fuzzy Logic* pada *Profile Matching* pada penelitian ini untuk menghindari nilai akhir yang berupa himpunan tegas, dan menangani masalah ketika

muncul data test dalam jumlah banyak, dimana terdapat nilai-nilai yang hampir sama, pendekatan *Fuzzy* akan menampilkan himpunan nilai akhir yang bijak karena berdasar pada simulasi keputusan yang paling ideal berdasarkan *rules* dan bersifat matematis. Tahapan dimulai dengan menentukan *variable* data dan aspek-aspek yang dibutuhkan dalam penilaian, dilanjutkan dengan pemetaan gap yang didahului dengan pendefinisian nilai ideal untuk setiap kriteria. Selisih setiap nilai data *testing* terhadap nilai ideal masing-masing kriteria merupakan gap. Kemudian nilai gap konversi bobot, nilai bobot diambil dari tabel bobot sesuai dengan gap yang dimiliki. Hasil konversi bobot ini kemudian dihitung pada perhitungan logika *fuzzy* yakni fuzzifikasi dimana nilai per kriteria dimasukkan pada fungsi derajat keanggotaan tinggi dengan persamaan  $= \frac{b-x}{b-a}$  dan fungsi derajat keanggotaan rendah pada persamaan  $\frac{x-a}{b-a}$ , yang berikutnya adalah memasuki fungsi inferensi MIN yakni mengambil nilai minimum dari input berdasar *rules* yang terbentuk. Lalu berdasar pada fungsi keanggotaan dan menggunakan hasil proses inferensi MIN dicari nilai  $z_n = b - \alpha \cdot \text{predikat} (b-a)$ . Tahap terakhir adalah *defuzzifikasi* proses konversi dari *fuzzy output* menjadi *crisp output* dengan menghitung rata-rata terbobot dari semua nilai  $z$  yang dihasilkan dengan persamaan  $\frac{(\alpha \cdot \text{predikat} 1 * z_1) + \dots + (\alpha \cdot \text{predikat} n * z_n)}{a \cdot \text{predikat} 1 + \dots + a \cdot \text{predikat} n}$  (Jumaidi, 2015). Diagram alir dari kombinasi *profile matching* dan logika *fuzzy* adalah sebagai berikut:





Gambar 2. 5 Diagram alir *profile matching*

### **BAB 3. METODOLOGI PENELITIAN**

Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem menggunakan metode pengembangan.

#### **3.1 Jenis Penelitian**

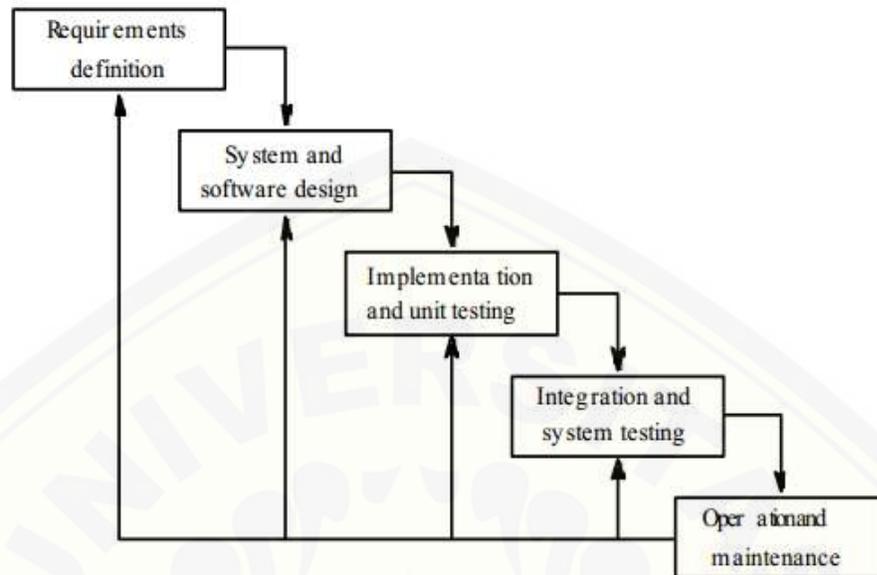
Jenis penelitian yang dilakukan merupakan penelitian kualitatif dan kuantitatif. Pendekatan kualitatif adalah suatu pendekatan yang juga disebut pendekatan investigasi karena biasanya peneliti mengumpulkan data dengan cara bertatap muka langsung dan berinteraksi dengan orang-orang di tempat penelitian (McMillan, 2006). Penggunaan metode kualitatif dalam penulisan ini adalah dengan melalui wawancara langsung dengan pimpinan instansi tentang seperti apa proses seleksi penerimaan pengajar yang dilakukan, menganalisis apa yang menjadi kelebihan serta kekurangannya sistem yang telah ada, dan studi literature yang berkaitan dengan seleksi dan perangkan sumber daya manusia. Pendekatan kuantitatif adalah pendekatan yang bertujuan menggambarkan benar tidaknya fakta-fakta yang ada serta menjelaskan tentang hubungan antar variabel yang diteliti dengan cara mengumpulkan data, mengolah, menganalisis, dan menginterpretasi data dalam pengujian hipotesis statistik, sehingga dapat teruji kebenarannya (Sugiyono, 2010). Penggunaan metode kuantitatif dalam penulisan ini adalah pengumpulan data yang digunakan dalam bentuk angka.

#### **3.2 Tempat dan Waktu Penelitian**

Penelitian dilakukan di Lembaga Bimbingan Belajar Galileo, Jalan Mastrip nomor 6A Jember. Waktu penelitian dilakukan selama 1 (satu) tahun, dimulai pada bulan April 2016 sampai dengan bulan April 2017.

#### **3.3 Tahap Penelitian**

Tahapan penelitian pengembangan sistem mengadopsi dari model *waterfall* dengan tahapan-tahapan seperti pada gambar berikut:



Gambar 3. 1 Model *Waterfall* (Sommerville, 2001)

#### 3.4.1 *Requirements Definition*

*Requirements definition* atau analisis kebutuhan merupakan tahap untuk mengumpulkan data, informasi, serta mencari kebutuhan fungsional dan non fungsional sistem. Pada tahap ini, peneliti mencari permasalahan yang ada untuk dapat dianalisis kebutuhan yang diperlukan, sebagai solusi dari permasalahan yang muncul. Data-data yang telah didapat kemudian dikelompokkan menjadi kebutuhan fungsional dan non-fungsional sistem.

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan melalui beberapa teknik pengumpulan data yaitu:

1. Melakukan wawancara kepada pimpinan Lembaga Bimbingan Belajar Galileo Jember untuk mendapatkan data dan informasi mengenai sistem seleksi penerimaan pengajar.

2. Melakukan studi literature, jurnal, media, maupun internet mengenai pembangunan sistem pendukung keputusan tentang pemecahan masalah kelayakan juga penggunaan kriteria kelayakan dari sebuah lembaga belajar.

#### 3.4.3 System and Software Design

Tahap ini meliputi tahap desain sistem dengan pembuatan diagram menggunakan *tools UML Visual Paradigm*. Diagram-diagram tersebut akan digunakan sebagai acuan pembuatan sistem pada tahap implementasi kebutuhan sistem. Konsep program menggunakan *Object-Oriented Programming* (OOP). Pada bagian desain, pengerjaan yang dilakukan diantaranya adalah:

- a. *Business Process*

*Business Process* merupakan gambaran dari masuknya data serta data yang dihasilkan dari proses yang dijalankan sistem. Data yang dibutuhkan oleh sistem (*input*), keluaran data yang dihasilkan (*output*), media dari sistem (*uses*), dan tujuan dari pembuatan sistem (*goal*). *Input* yang dibutuhkan berupa data pelajaran, data lowongan, data nilai kriteria pendaftar, data profil pendaftar, dan data nilai ideal kriteria . *Output* yang dihasilkan berupa normalisasi data nilai kriteria menggunakan normalisasi MIN-MAX dan hasil *ranking* pendaftar sesuai dengan perhitungan *Profile Matching* dan logika *fuzzy*.

- b. *Use Case Diagram*

*Use Case Diagram* merupakan dokumentasi yang menggambarkan fitur dan aktor yang mampu mengakses fitur-fitur pada sistem yang akan dibangun. Aktor yang dapat mengakses aplikasi ini yaitu pihak admin, LITBANG, dan pendaftar. Fitur yang disediakan pada sistem adalah fitur *login*, mengelola data profil *user*, mengelola data akses, mengelola nilai pendaftar, menampilkan data profil pendaftar, menampilkan data nilai pendaftar, mengelola data pelajaran, mengelola kuota lowongan, mengelola data status pendaftaran para pendaftar, menampilkan data pendaftar, *update* nilai wawancara dan *microteaching*,

mengelola nilai ideal kriteria, menyetujui hasil penerimaan, logout, dan melakukan pendaftaran.

c. *Use Case Scenario*

*Use case scenario* merupakan penjelasan alur sistem sesuai dengan yang tertera pada *use case diagram*, skenario juga menjelaskan reaksi yang akan terjadi pada sistem setelah menerima perlakuan dari aktor. *Use case scenario* menggambarkan keadaan normal saat setiap aktor mengakses aplikasi dan keadaan alternatif yang terjadi pada kondisi tertentu.

d. *Sequence Diagram*

*Sequence diagram* merupakan diagram yang digunakan untuk menggambarkan interaksi yang terjadi di dalam sistem sesuai dengan urutan dijalankannya sistem tersebut. Interaksi tersebut meliputi, aktor yang mengakses fitur tersebut sesuai dengan yang ada pada *use case diagram*, tampilan sistem, *controller*, model, dan pesan yang disampaikan jika terjadi suatu kondisi tertentu.

e. *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas dalam sistem yang sedang dirancang, meliputi awal alur dimulai, *decision* yang terjadi, dan bagaimana alur berakhir yang dapat dilakukan oleh masing-masing aktor. *Activity diagram* juga menggambarkan aktivitas sistem setelah menerima perlakuan dari aktor tersebut.

f. *Class Diagram*

*Class diagram* merupakan sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menggambarkan kelas, *function*, dan atribut yang dibutuhkan oleh seorang programmer dalam membangun sistem.

g. *Entity Relationship Diagram* (ERD)

*Entity relationship diagram* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

### 3.4.3 *Implementation and Unit Testing*

Setelah desain sistem telah selesai dilakukan maka selanjutnya pada tahap ini akan dilakukan pembuatan sistem dan implementasinya. Pembuatan sistem meliputi penulisan kode program, dan pembuatan basis data. Penulisan kode program dilakukan menggunakan *tools sublime text 2* sebagai *editor* dengan bahasa pemrograman *PHP* sedangkan untuk manajemen basis data yang digunakan dalam membangun sistem yaitu *DBMS MySQL*.

### 3.4.5 *Integration and System Testing*

*Integration and System Testing* atau pengujian digunakan untuk mengetahui sejauh mana sistem ini dapat berjalan. *Testing* berfungsi untuk mengetahui apakah sistem ini dapat berfungsi dengan baik sesuai dengan yang diharapkan. Serta untuk mengetahui letak kekurangan yang ada pada sistem ini. Terdapat dua metode yang digunakan untuk pengujian ini yakni :

a. *White box testing*

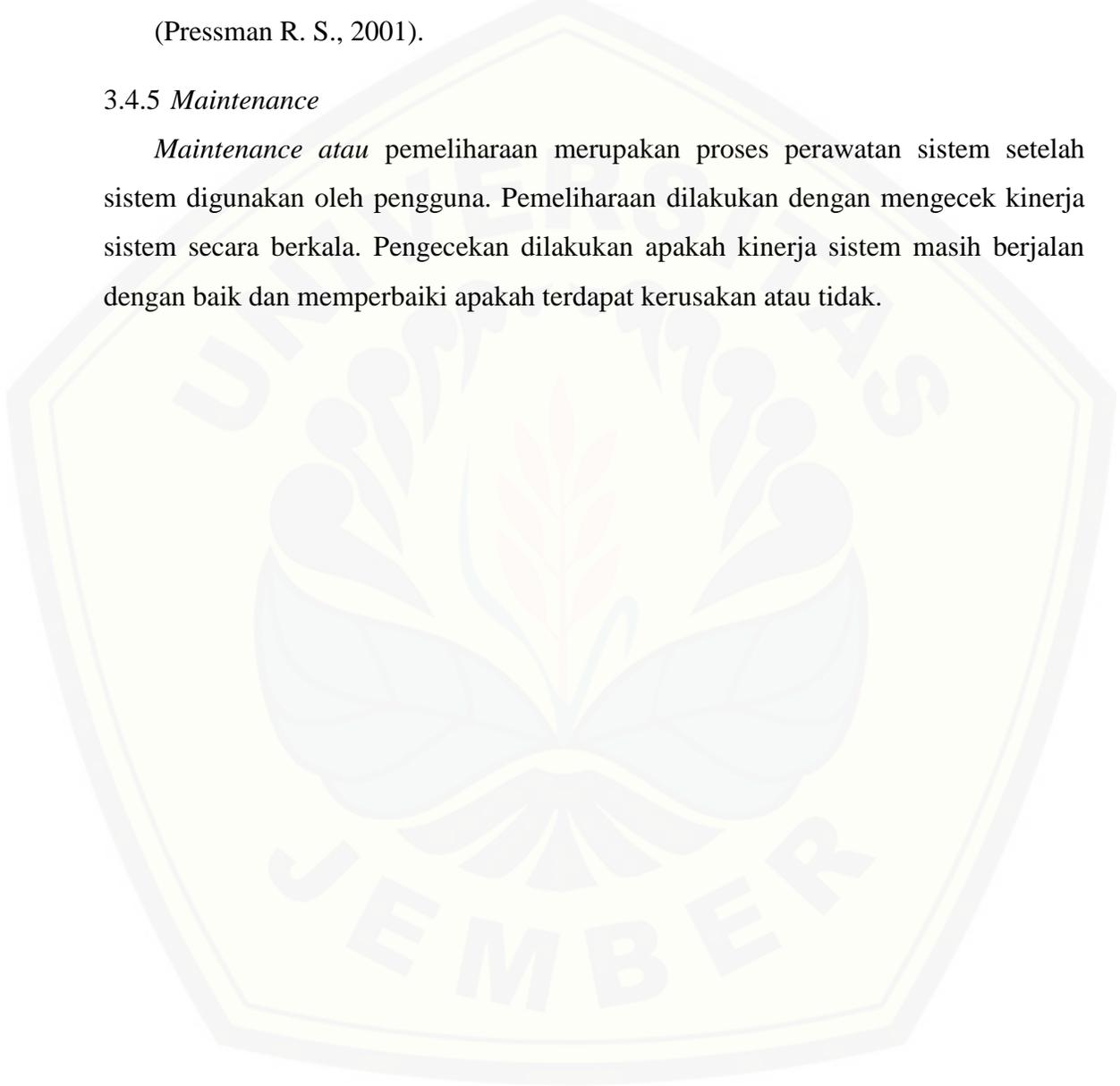
*White box testing* merupakan cara pengujian dengan melihat modul untuk yang telah dibuat dengan program – program yang ada. Pengujian ini dilakukan oleh (*developer*) pembuat program. Modul yang menghasilkan *output* yang tidak sesuai, maka baris-baris program, variabel dan parameter yang terlibat pada unit tersebut satu persatu akan di cek dan diperbaiki, kemudian di *compile* ulang. (Pressman R. S., 2001)

b. *Black box testing*

*Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur *internal* atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Pressman R. S., 2001).

#### 3.4.5 Maintenance

*Maintenance* atau pemeliharaan merupakan proses perawatan sistem setelah sistem digunakan oleh pengguna. Pemeliharaan dilakukan dengan mengecek kinerja sistem secara berkala. Pengecekan dilakukan apakah kinerja sistem masih berjalan dengan baik dan memperbaiki apakah terdapat kerusakan atau tidak.



## BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses perancangan sistem dalam penelitian Implementasi Logika *Fuzzy* dan *Profile Matching* pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar (studi kasus : LBB Galileo). Dimana tahapan analisis hingga pengujian yang dilakukan sesuai dengan metode pengembangan *waterfall*.

### 4.1 Deskripsi Umum Sistem

Deskripsi umum dari Sistem dengan Implementasi Logika *Fuzzy* dan *Profile Matching* pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar (Studi Kasus : LBB Galileo Jember) dijelaskan lebih detail pada SOP (*statement of purpose*) sistem dan fungsi sistem.

#### 4.1.1 SOP (*Statement of purpose*)

Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar menggunakan Logika *Fuzzy* dan *Profile Matching* merupakan sebuah sistem yang membantu pengambil keputusan LBB Galileo Jember dalam proses seleksi penerimaan pengajar. Data yang dibutuhkan adalah data pelajaran, data lowongan, data pelamar dan data kriteria penerimaan yakni IPK, asal lulusan, tes tulis, test wawancara, dan tes *microteaching*. Sistem memiliki 3 hak akses yakni akses bagi pendaftar, LITBANG, dan admin. Sistem dapat mengelola data akses, mengelola data pendaftar, mengelola data pelajaran, mengelola data nilai, mengelola data lowongan, dan memunculkan ranking sebagai hasil perhitungan nilai pendaftar yang akan digunakan sebagai data pendukung untuk para pengambil keputusan.

#### 4.1.2 Fungsi sistem

Fungsi utama sistem yang dibangun dalam penelitian ini terletak pada fitur login yang dapat menentukan hak akses dari setiap pengguna dari sistem ini sendiri. Ketika pengguna melakukan login, maka sistem akan melakukan autentikasi *username* dan

*password* dari pengguna. Selanjutnya sistem akan menyajikan tampilan sistem yang sesuai dengan hak akses dari pengguna yang meliputi :

a. Admin

Admin merupakan pengguna sistem yang telah terdaftar, dan ketika login berhasil maka sistem akan menampilkan halaman home dan halaman-halaman yang sesuai dengan level admin. Admin dapat mengakses beberapa fitur yaitu mengelola data profil user, mengelola data akses, mengelola nilai pendaftar, menampilkan data profile pendaftar, menampilkan data nilai pendaftar, mengelola data pelajaran, mengelola kuota lowongan, mengelola data status pendaftaran para pendaftar. Admin bertanggung jawab atas semua data yang akan diinputkan dan dibutuhkan.

b. LITBANG

LITBANG merupakan pengguna sistem yang telah terdaftar, dan ketika login berhasil maka sistem akan menampilkan halaman home dan halaman-halaman yang sesuai dengan level LITBANG. LITBANG dapat mengakses beberapa fitur yaitu mengelola data profil user, menampilkan data pendaftar, update nilai wawancara dan microteaching, mengelola nilai ideal kriteria, dan menyetujui hasil penerimaan. LITBANG bertanggung jawab atas semua data yang akan diinputkan dan dibutuhkan.

c. Pendaftar

Pendaftar merupakan pengguna sistem tanpa autentikasi yang dapat melakukan menginputkan data pendaftaran. Pendaftar bertanggung jawab atas data yang akan diinputkan dan dibutuhkan.

#### **4.2 Analisis Kebutuhan**

Berdasarkan metode pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahapan analisis kebutuhan. Tahapan ini dilakukan analisis terhadap kebutuhan-kebutuhan dari sistem yang dibangun, baik berupa kebutuhan fungsional maupun kebutuhan non-fungsional. Hasil analisis tersebut sangat mempengaruhi fungsionalitas sistem yang dibangun untuk dapat digunakan sesuai dengan fungsi dan kebutuhan pengguna.

#### 4.3.1. Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari sistem pendukung keputusan seleksi penerimaan pengajar menggunakan logika *fuzzy* dan *profile matching* yaitu:

- a. Sistem mampu mengelola data profil (*input, edit*)
- b. Sistem mampu mengelola data akses (*input, edit*)
- c. Sistem mampu mengelola nilai pendaftar (*input, edit*)
- d. Sistem mampu mengelola nilai data pelajaran (*input, edit, delete*)
- e. Sistem mampu mengelola data lowongan dan kuota lowongan (*input, edit*)
- f. Sistem mampu mengelola nilai ideal kriteria (*input, edit*)
- g. Sistem mampu menghitung dan menampilkan perhitungan metode Logika *Fuzzy* dan *Profile Matching*
- h. Sistem dapat menampilkan ringkasan penilaian.

#### 4.3.2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yaitu berupa fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi kebutuhan dari pengguna. Kebutuhan non-fungsional dari sistem pendukung keputusan seleksi penerimaan pengajar menggunakan logika *fuzzy* dan *profile matching* yaitu :

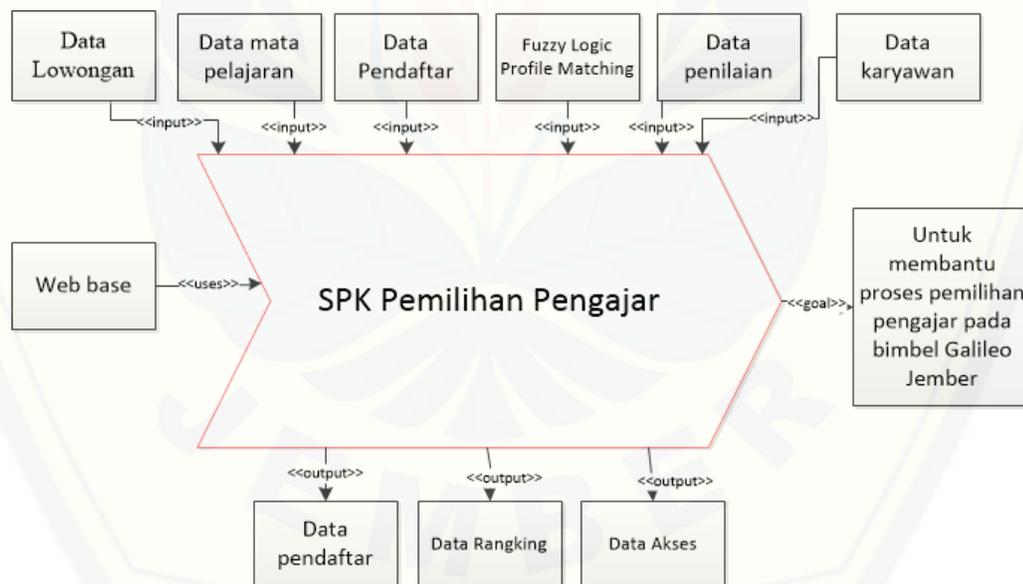
- a. Sistem bekerja sesuai dengan fungsinya dan dapat dijalankan pada semua komputer dan *browser* yang berbeda.
- b. Tampilan dan bahasa komunikasi sistem mudah dimengerti oleh pengguna untuk memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
- c. Sistem menggunakan *username* dan *password* untuk autentikasi akses pengguna terhadap sistem.
- d. Sistem mampu menampilkan hasil dalam waktu maksimal 5 detik

### 4.3 Desain Sistem

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang dapat digambarkan dengan desain sistem. Desain sistem penunjang keputusan seleksi penerimaan pengajar ini meliputi *business process*, *use case diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

#### 4.3.1. Business Process

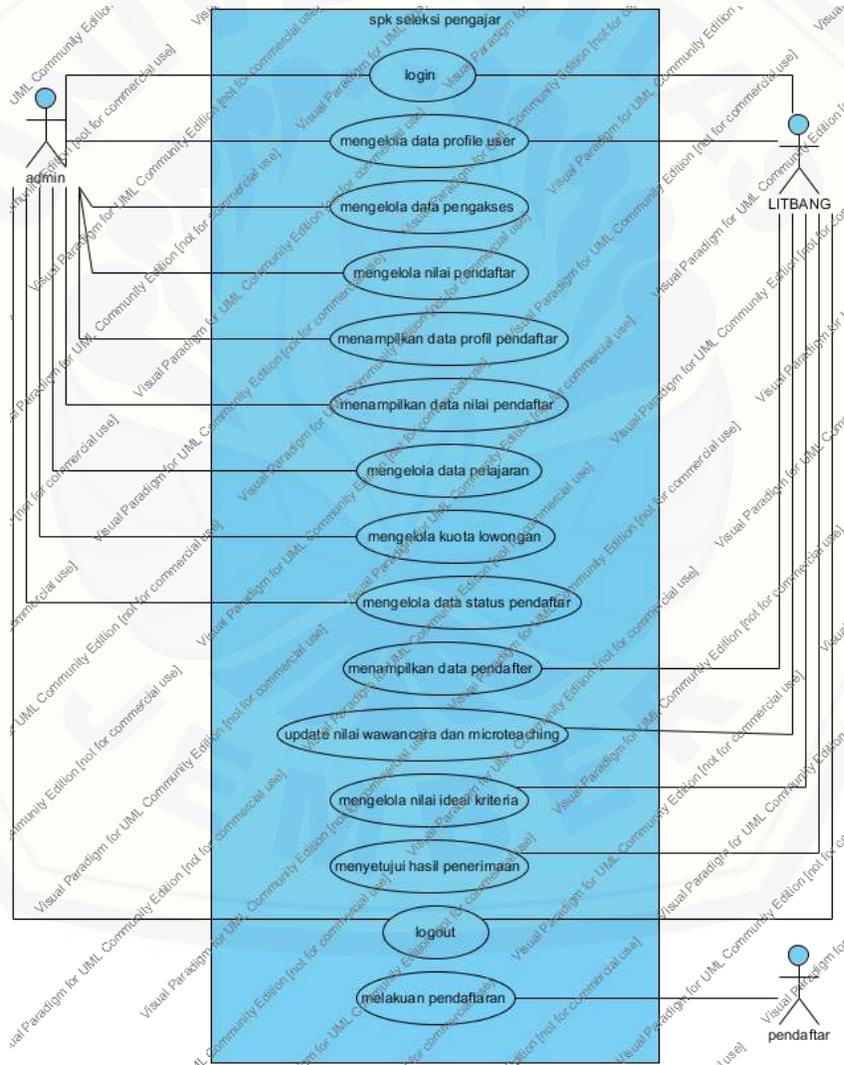
Selain dideskripsikan dalam SOP (*Statement of Purpose*), gambaran umum sistem pendukung keputusan seleksi penerimaan pengajar juga digambarkan dalam *business process*. *Business Proses* adalah gambaran aktivitas data yang terjadi dalam sistem, serta capaian yang telah ditentukan, adapun komponennya adalah input, output, goal dan media yang digunakan oleh sistem. *Business process* yang terjadi pada sistem yang dibangun kali ini seperti pada Gambar 4.1.



Gambar 4. 1 *Business Process* Sistem

4.3.2. Use Case Diagram

Use case diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan hubungan fitur atau suatu fungsi yang terdapat dalam sistem yang dikembangkan dengan actor yang terlibat. Melalui use case diagram dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing masing aktor atau pengguna. Pada Gambar 4.2 digambarkan use case diagram yang terdiri atas tiga aktor dengan lima belas use case.



Gambar 4. 2 Use Case Diagram

Berdasarkan *use case diagram* pada Gambar 4.2 terdapat tiga aktor atau pengguna, yaitu admin, LITBANG, dan pendaftar. Adapun deskripsi dari masing-masing aktor dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Deskripsi pembagian aktor sistem

No.	Aktor	Deskripsi
1	Admin	Admin merupakan aktor yang memiliki hak akses pada fitur dalam sistem yang meliputi <i>login</i> , mengelola data profil admin, mengelola hak akses, mengelola data nilai pendaftar hingga muncul nilai akhir sesuai perhitungan logika <i>fuzzy</i> dan <i>Profile Matching</i> , mengelola data pendaftar, mengelola data pelajaran, data lowongan
2	Litbang	Manager merupakan aktor yang memiliki hak akses pada fitur <i>login</i> , pengelola profil LITBANG, mengelola nilai standar, menginputkan nilai wawancara, menginputkan nilai <i>microteaching</i> , dan penentuan pendaftar yang diterima.
3	Pendaftar	Pendaftar merupakan aktor yang mengakses fitur pendaftaran.

Selain memiliki tiga aktor, dalam *use case diagram* juga terdapat lima belas *use case*. Deskripsi dari *use case* tersebut dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Deskripsi *use case* sistem

No.	Usecase	Deskripsi
1	<i>Login</i>	<i>Login</i> merupakan <i>usecase</i> yang berfungsi untuk memverifikasi hak akses LITBANG dan admin.
2	Mengelola data profil <i>user</i>	Mengelola data profil <i>user</i> merupakan <i>usecase</i> yang diakses admin dan LITBANG, merupakan fitur yang berfungsi untuk merubah data profil sesuai dengan <i>username</i> masing masing.
3	Mengelola data akses	Mengelola data akses merupakan <i>usecase</i> yang diakses admin, merupakan fitur yang berfungsi untuk membuat <i>username</i> untuk dapat mengakses system, mengubah <i>username</i> menjadi memiliki hak akses atau tidak memiliki hak akses.
4	Mengelola nilai pendaftar	Mengelola nilai pendaftar merupakan <i>usecase</i> yang diakses admin, merupakan fitur yang berfungsi untuk mengisikan nilai pendaftar, mengubah nilai pendaftar untuk dihitung menggunakan metode <i>fuzzy</i> dan <i>Profile Matching</i> sehingga dapat nilai akhir pendaftar untuk dirangking.
5	Menampilkan data	Menampilkan data profile pendaftar merupakan <i>usecase</i> yang

	profile pendaftar	diakses admin, merupakan fitur yang berfungsi untuk menampilkan profile pendaftar.
6	Menampilkan data nilai pendaftar	Menampilkan data nilai pendaftar merupakan <i>usecase</i> yang diakses admin, merupakan fitur yang berfungsi untuk menampilkan nilai criteria dan nilai akhir pendaftar.
7	Mengelola data pelajaran	Mengelola data pelajaran merupakan <i>usecase</i> yang diakses admin, merupakan fitur yang berfungsi untuk menambah, mengubah dan menghapus data pelajaran.
8	Mengelola kuota lowongan	Mengelola kuota lowongan <i>usecase</i> yang diakses admin, merupakan fitur yang berfungsi untuk menambah sebuah perekrutan pengajar, mengubah status lowongan.
9	Mengelola data status pendaftaran para pendaftar	Mengelola data status pendaftaran para pendaftar <i>usecase</i> yang diakses admin, merupakan fitur yang berfungsi untuk, mengubah status pendaftaran para pendaftaran sesuai periode perekrutan yang berlaku.
10	Menampilkan data pendaftar	Menampilkan data nilai pendaftar merupakan <i>usecase</i> yang diakses LITBANG, merupakan fitur yang berfungsi untuk menampilkan data profile, nilai criteria dan nilai akhir pendaftar.
11	<i>Update</i> nilai wawancara dan <i>microteaching</i>	<i>Update</i> nilai wawancara dan <i>microteaching</i> merupakan <i>usecase</i> yang diakses LITBANG, merupakan fitur yang berfungsi untuk mengisi dan mengubah nilai hasil wawancara dan tes <i>microteaching</i> pendaftar.
12	Mengelola nilai ideal kriteria	Mengelola nilai ideal kriteria <i>usecase</i> yang diakses LITBANG, merupakan fitur yang berfungsi untuk, mengubah status nilai minimal kriteria atau nilai standar penerimaan sesuai keputusan rapat LITBANG.
13	Menyetujui hasil penerimaan	Menyetujui hasil penerimaan <i>usecase</i> yang diakses LITBANG, merupakan fitur yang berfungsi untuk menyetujui hasil seleksi sebagai keputusan akhir setelah ditunjang hasil rangking sistem sesuai keputusan rapat LITBANG.
14	<i>Logout</i>	<i>Login</i> merupakan <i>usecase</i> yang berfungsi untuk keluar dari sistem.
15	Melakukan pendaftaran	Melakukan pendaftaran <i>usecase</i> yang diakses pendaftar, merupakan fitur yang berfungsi untuk pendaftaran dengan mengisi data pribadi calon pengajar.

#### 4.3.3. Use Case Skenario

*Use case* skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. *Use case* skenario sistem dalam penelitian berjudul “Implementasi Logika *Fuzzy* dan *Profile Matching* pada Sistem Pendukung Keputusan Seleksi Penerimaan Pengajar (studi kasus : LBB Galileo)” adalah sebagai berikut:

a. Skenario *Login*

Skenario *login* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario *login*. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur login antara aksi yang dilakukan oleh aktor yaitu admin dan LITBANG ketika keadaan akan melakukan login, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah melakukan login pada sistem tersebut. Proses lengkap skenario *login* dapat dilihat pada Lampiran A Tabel A.1 *Use case* skenario *login*.

b. Skenario Mengelola Data Profil User

Skenario mengelola data profil *user* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data profil *user*. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur pengelola data profile *user* antara aksi yang dilakukan oleh aktor yaitu admin dan LITBANG ketika keadaan akan mengubah data profil *user*, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengubah data profil *user* pada sistem tersebut. Proses lengkap skenario mengelola data profil user dapat dilihat pada Lampiran A Tabel A.2 *Use case* skenario mengelola data profil *user* (admin/ LITBANG).

c. Skenario Mengelola Data Akses

Skenario mengelola data akses merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data akses. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur menampilkan data akses, menambah data akses, kemudian merubah status akses atau level akses sebuah id *user* . Proses lengkap skenario

mengelola data akses dapat dilihat pada Lampiran A Tabel A.3 *Use case* skenario mengelola data akses.

d. Skenario Mengelola Nilai Pendaftar

Skenario mengelola nilai pendaftar merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola nilai pendaftar. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur pengelola nilai pendaftar antara aksi yang dilakukan oleh aktor yaitu admin ketika keadaan akan mengubah data nilai pendaftar, menampilkan nilai detail, menampilkan daftar nilai pendaftar, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengubah data data nilai pendaftar, menghitung nilai akhir, menampilkan detail nilai, dan menampilkan daftar nilai dari pendaftar pada sistem tersebut. Proses lengkap skenario mengelola nilai pendaftar dapat dilihat pada Lampiran A Tabel A.4 *Use case* skenario mengelola nilai pendaftar.

e. Skenario Menampilkan Data Profile Pendaftar

Skenario menampilkan data profile pendaftar merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menampilkan data profile pendaftar. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur menampilkan data profil pendaftar antara aksi yang dilakukan oleh aktor yaitu admin ketika keadaan akan melihat data profile pendaftar, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah melihat data profil pendaftar pada sistem tersebut. Proses lengkap skenario menampilkan data profile pendaftar dapat dilihat pada Lampiran A Tabel A.5 *Use case* skenario menampilkan data profile pendaftar.

## f. Skenario Menampilkan Data Nilai Pendaftar

Skenario menampilkan data Nilai pendaftar merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menampilkan data nilai pendaftar. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur menampilkan data profil pendaftar antara aksi yang dilakukan oleh aktor yaitu admin ketika keadaan akan melihat data nilai pendaftar, akan melihat detail nilai, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah melihat data nilai pendaftar dan detail nilai pendaftar pada sistem tersebut. Proses lengkap skenario menampilkan data nilai pendaftar dapat dilihat pada Lampiran A Tabel A.6 *Use case* skenario menampilkan data nilai pendaftar.

## g. Skenario Mengelola Data Pelajaran

Skenario mengelola data pelajaran merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data pelajaran. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur mengelola data pengguna yang dilakukan oleh aktor yaitu admin ketika keadaan akan menampilkan, menambah, mengubah, menghapus data pengguna, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data pengguna pada sistem tersebut. Proses lengkap skenario mengelola data pelajaran dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Skenario Mengelola Data Pelajaran

Nama <i>Use Case</i>	Mengelola data pelajaran
Aktor	Admin
Pre Kondisi	Aktor sudah <i>login</i> sistem
Post Kondisi	Aktor dapat menggunakan system untuk mengelola data pelajaran
<b>Skenario Normal</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Pilih menu ‘pembaruan’	

2. Pilih sub menu 'update data pelajaran'	3. Mengambil data mata pelajaran berupa nama pelajaran dari tb_mapel di database
	4. Mengambil data jenjang dari tb_jenjang di database
	5. Menampilkan halaman dengan daftar mata pelajaran, tombol 'tambah baru, tombol 'edit', dan tombol 'hapus'
6. Pilih tombol 'tambah baru' Jika pilih tombol 'edit' ke 5a, Jika pilih tombol 'hapus' ke 5b.	7. Menampilkan form untuk tambah data pelajaran <ul style="list-style-type: none"> <li>• Kolom nama pelajaran</li> <li>• <i>Combobox</i> jenjang</li> <li>• Tombol 'submit'</li> </ul>
8. Mengisi kolom pada form tambah mata pelajaran	
9. pilih 'submit button' jika pilih 'reset button' lanjut ke 7c.	10. Menyimpan data mata pelajaran dan jenjang yang diisikan pada kolom ke tb_mapel di database
	11. Mengambil data mata pelajaran dari tb_mapel di database
	12. Menampilkan halaman dengan daftar mata pelajaran, tombol 'tambah baru, tombol 'edit', dan tombol 'hapus'
	13. Menampilkan halaman dengan daftar mata pelajaran, tombol 'tambah baru, tombol 'edit', dan tombol 'hapus'
Skenario alternative (pilih tombol 'edit')	
Aksi Aktor	Reaksi Sistem
5a. Pilih tombol 'edit'	6a. Mengambil data mata pelajaran berupa nama pelajaran dari tb_mapel di database
	7a. Mengambil data mata pelajaran dari tb_mapel di database
	8a. Menampilkan data mata pelajaran yang terpilih ke halaman form edit <ul style="list-style-type: none"> <li>• Kolom nama pelajaran</li> </ul>

	<ul style="list-style-type: none"> <li>• <i>Combobox</i> jenjang</li> <li>• Tombol '<i>submit</i>'</li> </ul>
9a. Mengisi kolom pada form edit	
10a. Pilih ' <i>submit button</i> ' jika pilih ' <i>reset button</i> ' lanjut ke 8c.	
	11a. Menyimpan data mata pelajaran dan jenjang yang diisikan di <i>form</i> ke <i>tb_mapel</i> di <i>database</i>
	9a. Mengambil daftar mata pelajaran berupa nama pelajaran dari <i>tb_mapel</i> di <i>database</i>
	10 a. Menampilkan halaman dengan daftar mata pelajaran, tombol ' <i>tambah hak akses</i> ', tombol ' <i>edit</i> ', dan tombol ' <i>hapus</i> '
Skenario alternative (pilih tombol ' <i>hapus</i> ')	
Aksi Aktor	Reaksi Sistem
5b. Pilih tombol ' <i>hapus</i> '	6 b. menghapus/ <i>drop</i> mata pelajaran yang telah terpilih dari <i>tb_mapel</i> dari <i>database</i>
	7b. Mengambil data mata pelajaran berupa nama pelajaran dari <i>tb_mapel</i> di <i>database</i>
	8b. Menampilkan halaman dengan daftar mata pelajaran, tombol ' <i>tambah baru</i> ', tombol ' <i>edit</i> ', dan tombol ' <i>hapus</i> '
	8 b. Menampilkan halaman dengan daftar hak akses, tombol ' <i>tambah hak akses</i> ', tombol ' <i>edit</i> ', dan tombol ' <i>hapus</i> '
Skenario alternative (pilih ' <i>reset button</i> ')	
Aksi Aktor	Reaksi Sistem
8c. pilih reset button	9c. menampilkan form pada keadaan awal
Skenario Alternatif (kolom kosong/tidak diisi)	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan pesan "please fill in this field"

h. Skenario Mengelola Kuota Lowongan

Skenario mengelola data lowongan merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola kuota lowongan. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur mengelola data lowongan yang dilakukan oleh aktor yaitu admin ketika keadaan akan menampilkan, menambah, mengubah, menghapus data lowongan, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data pengguna pada sistem tersebut. Proses lengkap skenario mengelola data pelajaran dapat dilihat pada Lampiran A Tabel A.7 *Use case* skenario mengelola data lowongan.

i. Skenario Mengelola Data Status Pendaftaran Para Pendaftar

Skenario mengelola data status pendaftaran para pendaftar merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola data status pendaftaran para pendaftar. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur mengelola data data status pendaftaran para pendaftar yang dilakukan oleh aktor yaitu admin ketika keadaan akan menampilkan, menambah, mengubah, menghapus data status pendaftaran para pendaftar, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data status pendaftaran para pendaftar pada sistem tersebut. Proses lengkap skenario mengelola data pelajaran dapat dilihat pada Lampiran A Tabel A.8 *Use case* skenario mengelola data status pendaftaran para pendaftar

j. Skenario Menampilkan Data Pendaftar

Skenario menampilkan data pendaftar merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menampilkan data pendaftar. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur menampilkan data profil pendaftar antara

aksi yang dilakukan oleh aktor yaitu LITBANG ketika keadaan akan melihat data pendaftar, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah melihat data pendaftar pada sistem tersebut. Proses lengkap skenario menampilkan data pendaftar dapat dilihat pada Lampiran A Tabel A.9 *Use case* skenario menampilkan data pendaftar.

k. Skenario *Update Nilai Wawancara dan Microteaching*

Skenario *update* nilai wawancara dan *microteaching* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario *update* nilai wawancara dan *microteaching*. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur *update* nilai wawancara dan *microteaching* antara aksi yang dilakukan oleh aktor yaitu LITBANG ketika keadaan akan menampilkan, mengubah nilai wawancara dan *microteaching* pendaftar, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah *update* nilai wawancara dan *microteaching* pada sistem tersebut. Proses lengkap skenario *update* nilai wawancara dan *microteaching* dapat dilihat pada Lampiran A Tabel A.10 *Use case* skenario *update* nilai wawancara dan *microteaching*.

l. Skenario Mengelola Nilai Ideal Kriteria

Skenario mengelola nilai ideal kriteria merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario mengelola nilai ideal kriteria. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur mengelola nilai ideal kriteria antara aksi yang dilakukan oleh aktor yaitu LITBANG ketika keadaan akan menampilkan, mengubah data nilai ideal kriteria, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah menampilkan dan mengubah data nilai ideal kriteria pada sistem tersebut. Proses lengkap skenario mengelola nilai

ideal kriteria dapat dilihat pada Lampiran A Tabel A.11 *Use case* skenario mengelola nilai ideal kriteria.

m. Skenario Menyetujui Hasil Penerimaan

Skenario menyetujui hasil penerimaan merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* skenario menyetujui hasil penerimaan. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur data hasil penilaian nilai kriteria pendaftar antara aksi yang dilakukan oleh aktor yaitu LITBANG ketika keadaan akan melihat data hasil penilaian dan menentukan atau menyetujui hasil sebagai penerimaan calon pengajar menjadi pengajar, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah melihat data hasil menyetujui hasil penerimaan pada sistem tersebut. Proses lengkap menyetujui hasil penerimaan dapat dilihat pada Lampiran A Tabel A.12 *Use case* skenario menyetujui hasil penerimaan.

n. Skenario Melakukan Pendaftaran

Skenario melakukan pendaftaran merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *usecase* melakukan pendaftaran. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur pendaftaran antara aksi yang dilakukan oleh aktor yaitu pendaftar ketika keadaan akan melihat form pendaftaran, dan mengisi data pendaftaran, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah melihat form pendaftaran, dan melakukan pendaftaran pada sistem tersebut. Proses lengkap skenario melakukan pendaftaran dapat dilihat pada Lampiran A Tabel A.13 *Use case* skenario melakukan pendaftaran.

o. Skenario *Logout*

Skenario *logout* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario *logout*. Pada skenario tersebut dapat dilihat bagaimana alur perjalanan suatu fitur login antara aksi yang dilakukan oleh aktor yaitu admin dan manager ketika keadaan akan keluar dari sistem, kemudian bagaimana reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah keluar dari sistem tersebut. Proses lengkap skenario *logout* dapat dilihat pada Lampiran A Tabel A.14 *Use case* skenario *logout*.

4.3.4 *Activity Diagram*

*Activity Diagram* menggambarkan aliran aktivitas sistem pendukung keputusan seleksi penerimaan pengajar menggunakan metode logika *Fuzzy* dan *profile matching* di LBB Galileo yang akan dibangun. Sistem ini memiliki lima belas *activity diagram* yaitu sebagai berikut:

a. *Activity Login*

*Activity diagram login* dilakukan oleh admin dan LITBANG. *Activity diagram login* menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan autentikasi hak akses semua aktor dalam menggunakan sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.1 *Activity diagram login*.

b. *Activity Diagram Mengelola Data Profil User*

*Activity diagram* mengelola data profil *user* dapat dilakukan oleh pengguna level satu yakni admin dan LITBANG. *Activity* mengelola data profil *user* menjelaskan tentang bagaimana admin maupun LITBANG dapat mengubah data profil masing masing per id *user* pada sistem sehingga data yang *update*, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.2 *Activity* mengelola data profil *user*.

c. *Activity Diagram Mengelola Data Akses*

*Activity diagram* mengelola data akses dapat dilakukan oleh pengguna level admin. *Activity diagram* melihat dan mengubah data Akses menjelaskan tentang bagaimana admin dapat melihat siapa saja yang dapat mengakses system, membuat sebuah *iduser* dapat atau tidak dapat mengakses sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.3 *Activity* mengelola data akses.

d. *Activity Diagram Mengelola Nilai Pendaftar*

*Activity diagram* mengelola nilai pendaftar dapat dilakukan oleh pengguna yakni admin. *Activity diagram* mengelola nilai pendaftar menjelaskan tentang bagaimana admin dapat mengubah data nilai pendaftar pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.4 *Activity diagram* mengelola nilai pendaftar.

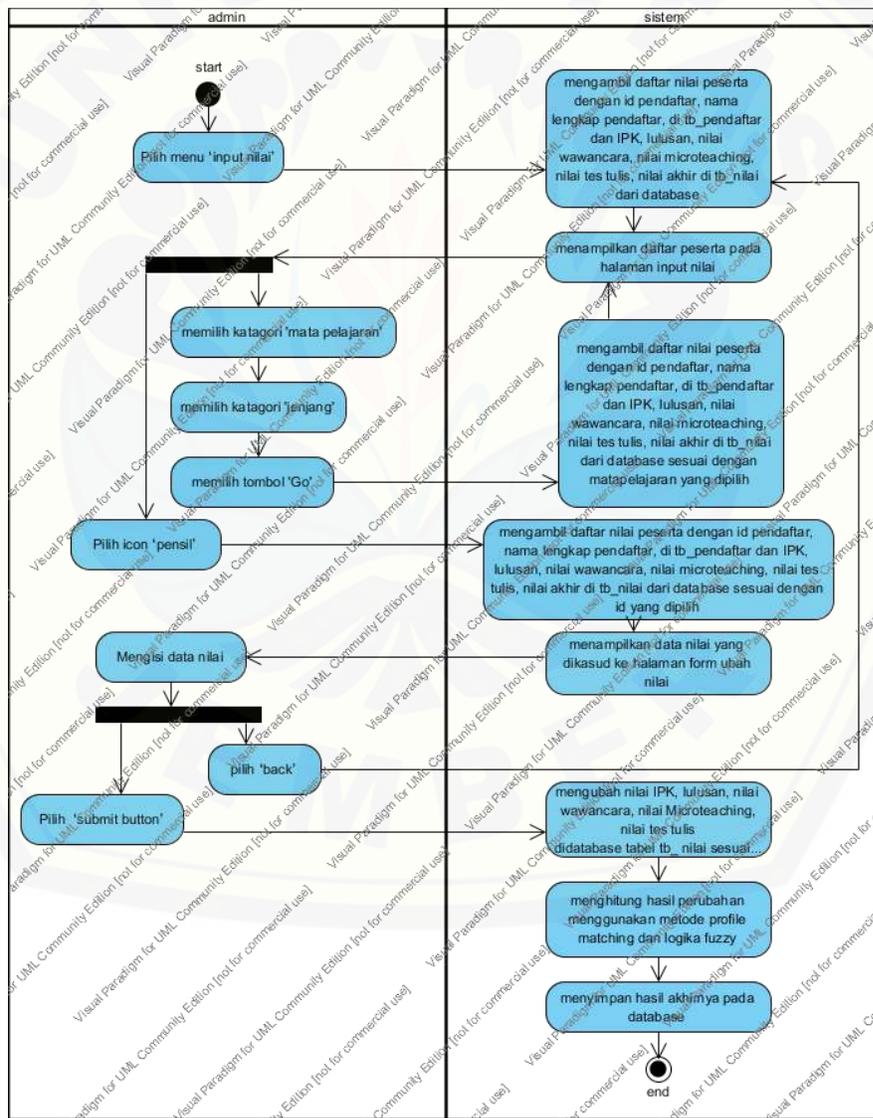
e. *Activity Diagram Menampilkan Data Profile Pendaftar*

*Activity diagram* menampilkan data profile pendaftar dapat dilakukan oleh pengguna admin. *Activity diagram* menampilkan data profil pendaftar menjelaskan tentang bagaimana admin dapat melihat data profil pendaftar pada sistem proses lengkapnya dapat dilihat pada Lampiran B Gambar B.5 *Activity diagram* menampilkan data profile pendaftar.

f. *Activity Diagram Menampilkan Data Nilai Pendaftar*

*Activity diagram* menampilkan data nilai pendaftar dan angka perhitungan tiap tahap dalam metode *profile matching* dan logika *fuzzy* yang dapat dilakukan oleh pengguna level admin. *Activity diagram* menampilkan data nilai pendaftar menjelaskan tentang bagaimana admin dapat melihat nilai para

pendaftar hingga nilai detailnya sesuai hasil perhitungan menggunakan Logika Fuzzy dan *Profile Matching* pada sistem. Fitur ini berada pada menu 'data pendaftar' pada sidebar user admin, tampilan fitur ini adalah daftar pendaftar, lalu admin memilih salah satu nama pendaftar dengan klik detail lalu pilih sub menu 'nilai pendaftar', maka akan ditampilkan halaman detail nilai nilai pendaftar, sedangkan untuk memunculkan detail per tahap perhitungan klik tombol 'show detail' pada halaman detail nilai. *Activity* diagram menampilkan data nilai pendaftar dapat dilihat pada Gambar 4.3.



Gambar 4. 3 *Activity* Diagram menampilkan data nilai pendaftar

g. *Activity* Diagram Mengelola Data Pelajaran

*Activity* diagram mengelola data pelajaran dapat dilakukan oleh pengguna level admin. *Activity* diagram mengelola data pelajaran menjelaskan tentang bagaimana admin dapat menampilkan, mengubah, dan menghapus data pelajaran pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.6 *Activity* diagram mengelola data pelajaran.

h. *Activity* Diagram Mengelola kuota lowongan

*Activity* diagram mengelola kuota lowongan dapat dilakukan oleh admin. *Activity* diagram mengelola kuota lowongan menjelaskan tentang bagaimana admin dapat melihat, mengubah, dan menambah data lowongan, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.7 *Activity* diagram mengelola kuota lowongan.

i. *Activity* Diagram Mengelola Data Status Pendaftaran Para Pendaftar

*Activity* diagram mengelola data status pendaftaran para pendaftar dapat dilakukan oleh admin. *Activity* diagram mengelola data status pendaftaran para pendaftar menjelaskan tentang bagaimana admin dapat melihat dan mengubah status pendaftaran para pendaftar pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.8 *Activity* diagram mengelola data status pendaftaran para pendaftar.

j. *Activity* Diagram Menampilkan Data Pendaftar

*Activity* diagram menampilkan data pendaftar dapat dilakukan oleh pengguna level LITBANG. *Activity* diagram menampilkan data pendaftar menjelaskan tentang bagaimana LITBANG dapat melihat data profil pendaftar dan data nilai

pendaftar pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.9 *Activity* diagram menampilkan data pendaftar.

k. *Activity* Diagram *Update* nilai Wawancara dan *Microteaching*

*Activity* diagram *update* nilai wawancara dan *microteaching* dapat dilakukan oleh pengguna level LITBANG. *Activity* diagram *update* nilai wawancara dan *microteaching* menjelaskan tentang bagaimana LITBANG dapat mengubah data nilai wawancara dan nilai *microteaching* pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.10 *Activity* diagram diagram *update* nilai wawancara dan *microteaching*.

l. *Activity* Diagram Mengelola Nilai Ideal Kriteria

*Activity* diagram mengelola nilai ideal kriteria dapat dilakukan oleh pengguna level LITBANG. *Activity* diagram mengelola nilai ideal kriteria menjelaskan tentang bagaimana LITBANG dapat melihat dan mengubah data nilai kriteria pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.11 *Activity* diagram mengelola nilai ideal kriteria.

m. *Activity* Diagram Menyetujui Hasil Penerimaan

*Activity* diagram menyetujui hasil penerimaan dapat dilakukan oleh pengguna level LITBANG. *Activity* diagram menyetujui hasil penerimaan menjelaskan tentang bagaimana LITBANG dapat melihat data pendaftar dan hasil perangkan nilai akhir para pendaftar, serta litbang dapat menentukan pendaftar yang diterima sebagai pengajar. proses lengkapnya dapat dilihat pada Lampiran B Gambar B.12 *Activity* diagram menyetujui hasil penerimaan.

n. *Activity Diagram Melakukan Pendaftaran*

*Activity* diagram melakukan pendaftaran dapat dilakukan oleh pengguna luar organisasi dalam hal ini pendaftar. *Activity* diagram melakukan pendaftaran menjelaskan tentang bagaimana pendaftar dapat melakukan pendaftaran dan mengisi data data mereka. proses lengkapnya dapat dilihat pada Lampiran B Gambar B.13 *Activity* diagram melakukan pendaftaran.

o. *Activity Diagram Logout*

*Activity* diagram *logout* menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan proses keluar dari sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.14 *Activity* diagram *logout*.

#### 4.3.5 Diagram *Sequence*

Diagram *Sequence* adalah dokumentasi suatu diagram yang menampilkan urutan interaksi - interaksi antar objek didalam sistem. *Sequence* diagram digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam sistem dengan cara *visual*. Diagram *Sequence* dari sistem pendukung keputusan seleksi penerimaan pengajar dengan mengimplementasikan Logika *Fuzzy* dan *Profile Matching* adalah sebagai berikut:

a. *Sequence Diagram Login*

Penggambaran *sequence login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case login*. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram *login* lebih lengkap dijelaskan pada Lampiran C Gambar C.1 *sequence* diagram *login*.

b. *Sequence* Diagram Mengelola Data Profil User

Penggambaran *sequence* diagram mengelola data profil user digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola data profil user. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola data profil user lebih lengkap dijelaskan pada Lampiran C Gambar C.2 *sequence* diagram mengelola data profil user.

c. *Sequence* Diagram Mengelola data akses

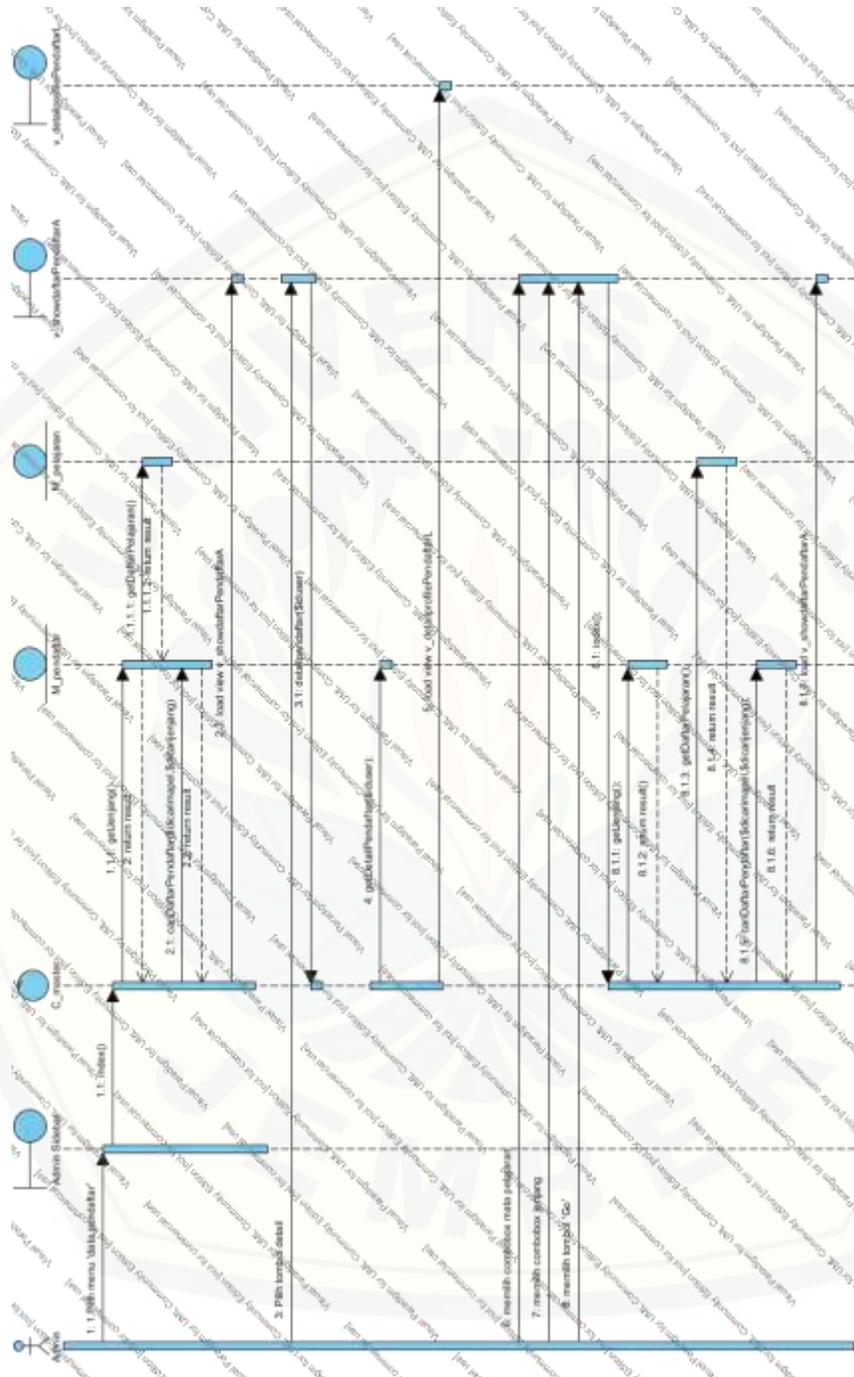
Penggambaran *sequence* diagram Mengelola data akses untuk menjelaskan fungsi atau *method* yang akan dibuat pada mengelola data akses. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola data akses lebih lengkap dijelaskan pada Lampiran C Gambar C.3 *sequence* diagram mengelola data akses.

d. *Sequence* Diagram Mengelola Nilai Pendaftar

Penggambaran *sequence* diagram mengelola nilai pendaftar digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola nilai pendaftar. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola nilai pendaftar lebih lengkap dijelaskan pada Lampiran C Gambar C.4 *sequence* diagram mengelola nilai pendaftar.

e. *Sequence* Diagram Menampilkan Data *Profile* Pendaftar

Penggambaran *sequence* diagram menampilkan data profile pendaftar digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menampilkan data *profile* pendaftar. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menampilkan data *profile* pendaftar lebih lengkap dijelaskan pada Gambar 4.4.



Gambar 4. 4 Sequence diagram menampilkan data profile pendftar

f. Sequence Diagram Menampilkan Data Nilai Pendaftar

Penggambaran *sequence* diagram menampilkan data nilai pendaftar digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use*

*case* menampilkan data nilai pendaftar. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menampilkan data nilai pendaftar lebih lengkap dijelaskan pada Lampiran C Gambar C.5 *sequence* menampilkan data nilai pendaftar.

g. *Sequence* Diagram Mengelola Data Pelajaran

Penggambaran *sequence* diagram mengelola data pelajaran digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case view* mengelola data pelajaran. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram Mengelola data pelajaran lebih lengkap dijelaskan pada Lampiran C Gambar C.6 *sequence* diagram mengelola data pelajaran.

h. *Sequence* Diagram Mengelola Kuota Lowongan

Penggambaran *sequence* diagram mengelola kuota lowongan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola kuota lowongan. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola kuota lowongan lebih lengkap dijelaskan pada Lampiran C Gambar C.7 *sequence* diagram mengelola kuota lowongan.

i. *Sequence* Diagram Mengelola Data Status Pendaftaran Para Pendaftar

Penggambaran *sequence* diagram *edit* data kuesioner pertanyaan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola data status pendaftaran para pendaftar. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola data status pendaftaran para pendaftar lebih lengkap dijelaskan pada Lampiran C Gambar C.8 *sequence* diagram Mengelola data status pendaftaran para pendaftar.

j. *Sequence Diagram Menampilkan Data Pendaftar*

Penggambaran *sequence* diagram menampilkan data pendaftar digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menampilkan data pendaftar. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menampilkan data pendaftar lebih lengkap dijelaskan pada Lampiran C Gambar C.9 *sequence* diagram menampilkan data pendaftar.

k. *Sequence Diagram Update Nilai Wawancara dan Microteaching*

Penggambaran *sequence update* nilai wawancara dan *microteaching* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case update* nilai wawancara dan *microteaching*. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram *update* nilai wawancara dan *microteaching* lebih lengkap dijelaskan pada Lampiran C Gambar C.10 *sequence* diagram *update* nilai wawancara dan *microteaching*.

l. *Sequence Diagram Mengelola Nilai Ideal Kriteria*

Penggambaran *sequence* diagram mengelola nilai ideal kriteria digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola nilai ideal kriteria. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola nilai ideal kriteria dijelaskan pada Lampiran C Gambar C.11 *sequence* diagram mengelola nilai ideal kriteria.

m. *Sequence Diagram Menyetujui Hasil Penerimaan*

Penggambaran *sequence diagram* menyetujui hasil penerimaan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menyetujui hasil penerimaan. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* menyetujui hasil penerimaan dijelaskan pada Lampiran C Gambar C.12 *sequence diagram* menyetujui hasil penerimaan.

n. *Sequence Diagram Melakukan Pendaftaran*

Penggambaran *sequence diagram* melakukan pendaftaran digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* Melakukan Pendaftaran. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* melakukan pendaftaran dijelaskan pada Lampiran C Gambar C.13 *sequence diagram* melakukan pendaftaran.

o. *Sequence Diagram Logout*

Penggambaran *sequence diagram* *logout* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* *logout*. Masing - masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* *logout* lebih lengkap dijelaskan pada Lampiran C Gambar C.14 *sequence diagram* *logout*.

#### 4.3.6 *Class Diagram*

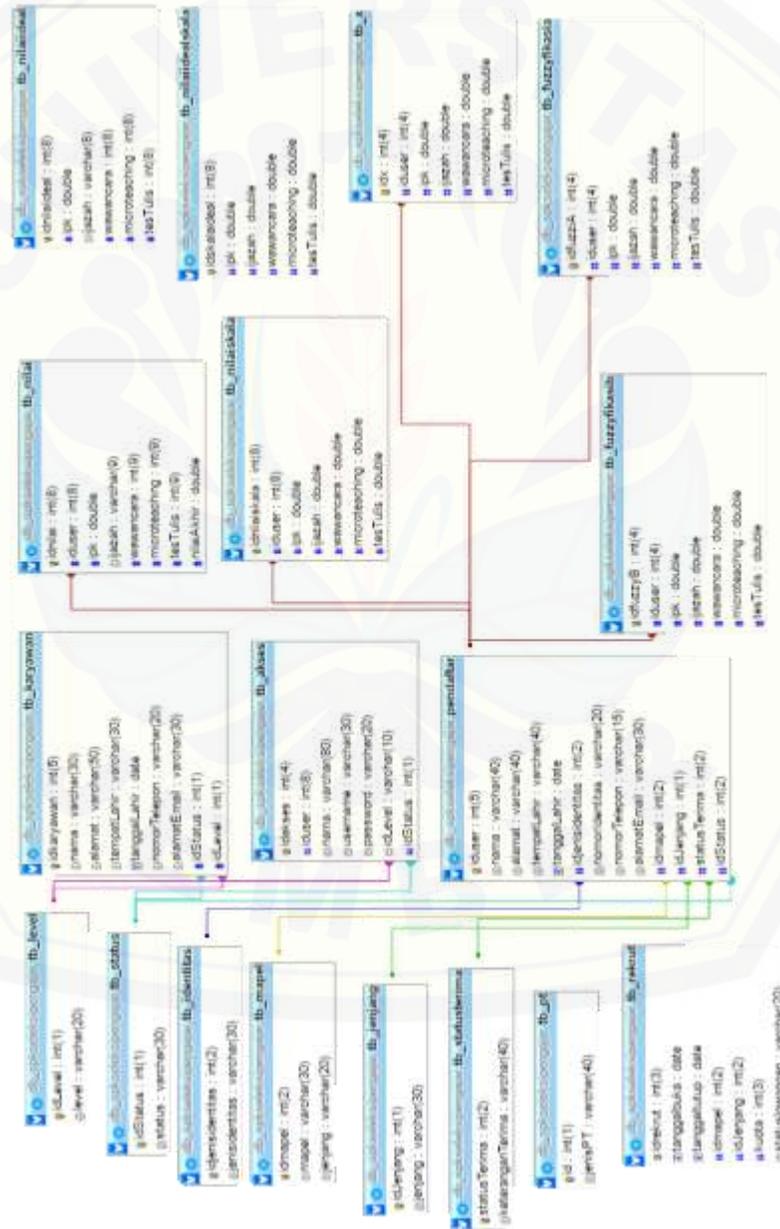
*Class diagram* menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu sistem. Dalam paradigma OOP (*Object Oriented Programming*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller*. Berdasarkan *sequence diagram* yang telah dibangun, *class diagram* sistem pendukung keputusan seleksi penerimaan pengajar menggunakan Logika *fuzzy* dan *Profile Matching* terdiri dari 7



gambaran komponen dan struktur *database* yang digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 18 entitas yang dapat dilihat pada Gambar 4.6.

#### 4.4 Implementasi

Desain akan diimplementasikan ke dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:



Gambar 4. 6 Entity Relationship Diagram

- a. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter (CI)*.
- b. Manajemen basis data menggunakan *DBMS MySQL*.

Kode program perhitungan nilai kriteria metode Logika Fuzzy dan *Profile Matching* terdapat di *class m\_nilai* pada *package models* dan di *class C\_nilai* pada *package controllers*. Kode program perhitungan metode Logika Fuzzy dan *Profile Matching* dapat dilihat pada lampiran D.

Perhitungan metode pada sistem meliputi beberapa tahap yaitu menginputkan nilai, normalisasi nilai, menghitung gap nilai normalisasi, mengkonversi ke nilai bobot, menghitung nilai fuzzifikasi, mencari inferensi dan menghitung nilai defuzzifikasinya, dari tahap-tahap tersebut harus diketahui terlebih dahulu adalah berapa nilai standar penerimaan di LBB Galileo.

Terdapat nilai standar yang ditentukan berdasarkan musyawarah LITBANG diinputkan ke sistem oleh *user level LITBANG* pada menu 'nilai ideal', melalui fungsi `updateDataNilaiIdeal($idUser)`; adapun kode program dapat dilihat pada lampiran D.1. Nilai yang diinputkan pada *form* edit nilai ideal kemudian disimpan pada tabel `tb_nilaiideal` pada *database db\_spkseleksi* pengajar melalui fungsi `editideal($var)` dimana kode program dilampirkan pada lampiran D.2, nilai kemudian dinormalisaikan dan disimpan pada tabel `tb_nilaiidealskala` pada *database db\_spkseleksi* pengajar melalui fungsi `editidealskala($var)` dimana kode program dilampirkan pada lampiran D.3.

Perhitungan metode akan dimulai ketika nilai kriteria pendaftar sudah lengkap dan nilai lebih atau sama dengan nilai standar, sesuai dengan kebutuhan instansi maka nilai dibawah standar akan otomatis tidak dapat diterima, hal tersebut didapati peneliti saat melakukan wawancara kepada tim LITBANG. Proses perhitungan metode diawali dengan input nilai melalui fungsi `updateDataNilai($idUser)` *class controller*, kode program dilampirkan pada Lampiran D.4. Nilai kriteria pendaftar yang diinputkan LITBANG atau ADMIN kemudian disimpan pada `tb_nilai` melalui fungsi

*editdatanilai(\$input)* dimana kode program dilampirkan pada lampiran D.5. Nilai kriteria kemudian juga dinormalisasi dan disimpan pada tabel *tb\_nilaiskala* melalui class model *m\_nilai* melalui fungsi *editnilaix(\$input)*, dimana kode program dilampirkan pada lampiran D.6.

Prose dilanjutkan dengan memeriksa nilai pendaftar apakah nilai diatas atau sama dengan nilai ideal jika nilai pendaftar nilai ideal maka perhitungan tidak akan dilanjutkan ke tahap perhitungan metode yang selanjutnya, kode programnya diImplementasikan difungsi *ceknilaidetail(\$iduser)* pada lampiranD.7, dengan hasil memunculkan nilai boolean. Jika nilai kriteria pendaftar dibawah nilai standar yang ditentukan atau didapatkan nilai *false* pada pengecekan pada fungsi *ceknilaidetail(\$iduser)* maka nilai akhir pendaftar tersebut akan dihapus dari tabel *tb\_nilai* atribut *nilaiAkhir*, berarti sistem akan menjalankan fungsi *dropnilaiakhir(hitungnilai pendaftar())*dimana kode program dilampirkan pada lampiran D.8, hal ini untuk menyesuaikan kebutuhan sistem dimana nilai dibawah standar tidak direngking. Jika didapatkan nilai *true* pada pengecekan pada fungsi *ceknilaidetail(\$iduser)* maka perhitungan metode akan dilanjutkan dengan memanggil fungsi *hitungnilai pendaftar()* pada class model *M\_nilai*, dimana kode program dilampirkan pada lampiran D.9. Fungsi *hitungnilai pendaftar()* akan dilanjutkan proses perhitungan yaitu pemetaan gap melalui fungsi *hitunggap(\$iduser)* yang dipanggil pada class *m\_nilai* dimana kode program dilampirkan pada lampiran D.10, lalu perhitungan fuzzifikasi melalui fungsi *fuzzyfikasi(\$iduser)* dimana kode program dilampirkan pada lampiran D.11 dan defuzzifikasi pada fungsi *hitungratarata(\$iduser)* dimana kode program dilampirkan pada lampiran D.11 dan D.12.

Fungsi *hitunggap(\$iduser)* dipanggil dari *database* nilai kriteria pendaftar yang sudah dinormalisasikan dari *tb\_nilaiskala* dan nilai ideal yang sudah dinormalisasiakan dari *tb\_nilaiidealskala*, kemudian akan dikurangkan sesuai dengan jenis kriteria masing masing, hasilnya adalah gap yang akan disimpan pada tabel *tb\_x*. Fungsi yang dijalankan berikutnya adalah *fuzzyfikasi(\$iduser)*, memproses nilai

gap yang didapatkan ketahap perhitungan fuzzifikasi dengan mengimplementasikan fungsi keanggotaan hingga didapati derajat keanggotaan pada dua domain, lalu derajat keanggotaan tersebut akan disimpan pada tabel *tb\_fuzzyfikasiA* dan tabel *tb\_fuzzyfikasiB* di database *db\_spkseleksi* pengajar. Kode program yang dijalankan selanjutnya adalah *hitungratarata(\$iduser)*, yakni menghitung semua proses setelah fuzzifikasi hingga proses defuzzifikasi, yakni fungsi inferensi, dimana terdapat *rules* yang mewakili keadaan normal keputusan, berdasar *rules* akan dihasilkan nilai  $\alpha$ -predikat yang merupakan nilai minimal dari kombinasi kriteria yang terbentuk. Nilai  $\alpha$ -predikat akan digunakan pada tahap yang selanjutnya yaitu mendapatkan nilai  $z_1$ - $z_n$  berdasar pada persamaan fungsi keanggotaan, lalu nilai yang didapatkan akan di hitung rata-ratanya sebagai *output crisp* nilai ini disimpan pada *tb\_nilai* sebagai *value* dari atribut *nilaiAkhir*, sebagai nilai akhir seorang pendaftar yang akan dibandingkan dengan nilai akhir pendaftar lainnya yang akan menjadi data pendukung untuk para pengambil keputusan dalam menerima pengajar.

## 4.5 Pengujian

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan sebuah metode pengujian sistem untuk mengetahui apakah kebutuhan sistem telah terpenuhi dan sistem layak untuk digunakan oleh *user*. Agar pengujian yang dilakukan lebih valid, maka tahap pengujian sistem pendukung keputusan seleksi penerimaan pengajar menggunakan Logika Fuzzy dan *Profile Matching* ini dilakukan dengan menggunakan dua metode, yaitu *white box* dan *black box*.

### 4.5.1. Metode White Box

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Hal tersebut bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *white box* dilakukan oleh peneliti dengan cara menghitung *independent path* yaitu dengan

menggunakan suatu pengukuran kuantitatif *cyclomatic complexity*, listing program, penentuan jalur independen, dan *test case*. Tahapan pengujian menggunakan metode *white box* meliputi:

a. *Listing Program*

*Listing* program merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program.

b. Diagram alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Diagram alir digambarkan dengan node-node (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program.

c. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis set suatu program. Perhitungan kompleksitas siklomatik menggunakan rumus berikut ini.

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

Keterangan :

V(G) : Kompleksitas siklomatik

E : Jumlah *edge*

N : Jumlah *node*

d. Jalur Independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan

atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi.

e. Pengujian Basis Set (*Test Case*)

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali.

Pengujian sistem pendukung keputusan seleksi penerimaan pengajar dengan metode *Logika Fuzzy* dan *Profile Matching* menggunakan metode *white box* akan diterapkan pada fitur yang dinilai dapat mewakili atau fitur utama sistem pendukung keputusan seleksi penerimaan pengajar menggunakan *Logika Fuzzy* dan *Profile Matching* seperti gambar 4.7 sampai gambar 4.10 .

1). Listing Program Update Nilai Pendaftar

```
134 public function hitungnilaipendaftar($input)
135 {
136     $iduser=$input['iduser'];
137
138     $ipkideals=0.0;
139     $ijazahideals=0.0;
140     $swawancaraideals=0.0;
141     $microteachingideals=0.0;
142     $testulisideals=0.0;
143
144     $nilaiideal = $this->db->query("select * from tb_nilaiidealskala ");
145     foreach ($nilaiideal->result_array() as $nilai)
146     {
147         $ipkideals=$nilai['ipk'];
148         $ijazahideals=$nilai['ijazah'];
149         $swawancaraideals=$nilai['wawancara'];
150         $microteachingideals=$nilai['microteaching'];
151         $testulisideals=$nilai['testulis'];
152     }
153
154     $ipks = 0.0;
155     $ijazabs=0.0;
156     $swawancara=0.0;
157     $microteachings=0.0;
158     $testuliss=0.0;
159
160     $nilaipendaftar = $this->db->query("select * from tb_nilaiskala where iduser=$iduser ");
161     foreach ($nilaipendaftar->result_array() as $nilai)
162     {
163         $ipks=$nilai['ipk'];
164         $ijazabs=$nilai['ijazah'];
165         $swawancara=$nilai['wawancara'];
166         $microteachings=$nilai['microteaching'];
167         $testuliss=$nilai['testulis'];
168     }
169
170     if ($ipks < $ipkideals || $ijazabs < $ijazahideals || $swawancara < $swawancaraideals || $testuliss < $testulisideals ||
171     $microteachings < $microteachingideals)
172     {
173         $this->hitunggap($iduser);
174         $this->fuzzyfikasi($iduser);
175         $this->hitungrata($iduser);
176     }
177 }
178 }
```

Gambar 4. 7 Program *function hitungnilai pendaftar(\$var)*

Dalam proses perhitungan kode fungsi *hitungnilai pendaftar()* memanggil fungsi lainnya yaitu *hitunggap()* untuk proses pemetaan gap, *fuzzyfikasi()* untuk proses fuzzifikasi, dan *hitungratarata()* untuk proses defuzzifikasi, maka akan ditampilkan listing program untuk masing masing tahap berikut yang dimunculkan secara per fungsi.

#### Listing Program Hitung Gap

```
function hitunggap($iduser)// menghitung nilai gap yang didapat dari normalisasi nilai ideal dikurangi nor
{
    $idealipks=0.0;
    $idealijazahs=0.0;
    $idealwawancaras=0.0;
    $idealmicroteachings=0.0;
    $idealtestuliss=0.0;
    $nilaiideal = $this->db->query("select * from tb_nilaiidealskala ");
    foreach ($nilaiideal->result_array() as $nilai) {
        $ipkideals=$nilai['ipk'];
        $ijazahideals=$nilai['ijazah'];
        $wawancaraideals=$nilai['wawancara'];
        $microteachingideals=$nilai['microteaching'];
        $testulisiideals=$nilai['tesTulis'];
    }

    $ipks = 0.0;
    $ijazahs=0.0;
    $wawancaras=0.0;
    $microteachings=0.0;
    $tesTuliss=0.0;
    $nilaipendaftar = $this->db->query("select * from tb_nilaiskala where iduser='$iduser' ");
    foreach ($nilaipendaftar->result_array() as $nilai) {
        $ipks=$nilai['ipk'];
        $ijazahs=$nilai['ijazah'];
        $wawancaras=$nilai['wawancara'];
        $microteachings=$nilai['microteaching'];
        $tesTuliss=$nilai['tesTulis'];
    }

    $x1 =$ipkideals-$ipks;
    $x2 =$ijazahideals-$ijazahs;
    $x3 =$wawancaraideals-$wawancaras;
    $x4 =$microteachingideals-$microteachings;
    $x5 =$testulisiideals-$tesTuliss;

    $this->db->query("UPDATE tb_x SET ipk ='$x1',ijazah='$x2',wawancara='$x3',microteaching='$x4',
    |tesTulis='$x5' where iduser='$iduser'");
}
```

Gambar 4. 8 Listing program function *hitunggap(\$var)*

#### Listing Program Fuzzifikasi

```

623 function fuzzyfikasi($iduser)// menghitung fuzzyfikasi nilai pendftar
624 {
625     $nilaix = 0;
626     $xipk=0;
627     $xijazah=0;
628     $xwawancara=0;
629     $xmicroteaching=0;
630     $xtesTulis=0;
631     $gap=0;
632     $bobot=0;
633     $bobotipk=0;
634     $bobotijazah=0;
635     $bobotwawancara=0;
636     $bobotmicroteaching=0;
637     $bobotTestulis=0;
638
639     $keanggotaanipkA=0;
640     $keanggotaanipkB=0;
641     $keanggotaanijazahA=0;
642     $keanggotaanijazahB=0;
643     $keanggotaanwawancaraA=0;
644     $keanggotaanwawancaraB=0;
645     $keanggotaanmicroteachingA=0;
646     $keanggotaanmicroteachingB=0;
647     $keanggotaantesTulisA=0;
648     $keanggotaantesTulisB=0;
649     $A=0;
650     $B=0;
651     $gapmax=0.0;
652     $nilaimin=0.0;
653     $ipkideals=0.0;
654     $ijazahideals=0.0;
655     $wawancaraideals=0.0;
656     $microteachingideals=0.0;
657     $tesTulisideals=0.0;
658     $nilaiskala = $this->db->query("select * from tb_x where iduser='$iduser'");
659     foreach ($nilaiskala->result_array() as $nilai) {
660         $xipk=$nilai['ipk'];
661         $xijazah=$nilai['ijazah'];
662         $xwawancara=$nilai['wawancara'];
663         $xmicroteaching=$nilai['microteaching'];
664         $xtesTulis=$nilai['tesTulis'];
665     }
666
667     $nilaideal=$this->db->query("select * from tb_nilaidealskala");
668     foreach ($nilaideal->result_array() as $nilai) {
669         $ipkideals=$nilai['ipk'];
670         $ijazahideals=$nilai['ijazah'];
671         $wawancaraideals=$nilai['wawancara'];
672         $microteachingideals=$nilai['microteaching'];
673         $tesTulisideals=$nilai['tesTulis'];
674     }
675
676     $nilaimin=min($ipkideals,$ijazahideals,$wawancaraideals,$microteachingideals,$tesTulisideals);
677     $gapmin=0;
678     $gapmax=1;
679     $A=(((($gapmax-($gapmax-$nilaimin))*$gapmin)/2)); //bobotipk
680     $B=(((($gapmax-($gapmax-$nilaimin))*$gapmax)/2)); //bobotwawancara
681
682
683     $bobotipk=(((($gapmax-($gapmax-$nilaimin))*$xipk)/2);
684     $bobotijazah=(((($gapmax-($gapmax-$nilaimin))*$xijazah)/2);
685     $bobotwawancara=(((($gapmax-($gapmax-$nilaimin))*$xwawancara)/2);
686     $bobotmicroteaching=(((($gapmax-($gapmax-$nilaimin))*$xmicroteaching)/2);
687     $bobotTestulis=(((($gapmax-($gapmax-$nilaimin))*$xtesTulis)/2);
688
689
690     $keanggotaanipka=($B-$bobotipk)/($B-$A);
691     $keanggotaanipkb=($bobotipk-$A)/($B-$A);
692     $keanggotaanijazahA=($B-$bobotijazah)/($B-$A);
693     $keanggotaanijazahB=($bobotijazah-$A)/($B-$A);
694     $keanggotaanwawancaraA=($B-$bobotwawancara)/($B-$A);
695     $keanggotaanwawancaraB=($bobotwawancara-$A)/($B-$A);
696     $keanggotaanmicroteachingA=($B-$bobotmicroteaching)/($B-$A);
697     $keanggotaanmicroteachingB=($bobotmicroteaching-$A)/($B-$A);
698     $keanggotaantesTulisA=($B-$bobotTestulis)/($B-$A);
699     $keanggotaantesTulisB=($bobotTestulis-$A)/($B-$A);
700
701     $query=$this->db->query("select tb_fuzzyfikasi as ipk, $keanggotaanipka, ijazah, $keanggotaanijazahA,
702     wawancara, $keanggotaanwawancaraA, microteaching, $keanggotaanmicroteachingA, tesTulis, $keanggotaantesTulisA where iduser='$iduser'");
703     $query2=$this->db->query("select tb_fuzzyfikasi as ipk, $keanggotaanipkb, ijazah, $keanggotaanijazahB,
704     wawancara, $keanggotaanwawancaraB, microteaching, $keanggotaanmicroteachingB, tesTulis, $keanggotaantesTulisB where iduser='$iduser'");
705     return array ($query,$query2);
706 }

```

Gambar 4. 9 Listing program function fuzzyfikasi(\$var)

## Listing Program Hitung Rata-Rata

```

585 function hitungrata_rata($iduser) // menghitung hasil mesin inferensi dilanjut dengan defusikasi terpusat sel
586 {
587     // (nilai akhir disimpan di tb_nilai)
588     $Alphaipk1=0;
589     $Alphaijazah1=0;
590     $Alphawawancara1=0;
591     $Alphamicroteaching1=0;
592     $Alphatestulis1=0;
593     $Alphaipk2=0;
594     $Alphaijazah2=0;
595     $Alphawawancara2=0;
596     $Alphamicroteaching2=0;
597     $Alphatestulis2=0;
598
599     $jumlahalpha=0;
600     $jumlahi=0;
601     $alphaA=$this->db->query("SELECT * from tb_fuzzyfikasia where iduser='$iduser'");
602     foreach ($alphaA->result_array() as $data) {
603         $Alphaipk1=$data['ipk'];
604         $Alphaijazah1=$data['ijazah'];
605         $Alphawawancara1=$data['wawancara'];
606         $Alphamicroteaching1=$data['microteaching'];
607         $Alphatestulis1=$data['testulis'];
608     }
609     $alphaB=$this->db->query("SELECT * from tb_fuzzyfikasib where iduser='$iduser'");
610     foreach ($alphaB->result_array() as $data) {
611         $Alphaipk2=$data['ipk'];
612         $Alphaijazah2=$data['ijazah'];
613         $Alphawawancara2=$data['wawancara'];
614         $Alphamicroteaching2=$data['microteaching'];
615         $Alphatestulis2=$data['testulis'];
616     }
617     //return
618     $alpha1=min($Alphaipk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis1);
619     $alpha2=min($Alphaipk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis2);
620     $alpha3=min($Alphaipk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis2);
621     $alpha4=min($Alphaipk1,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
622     $alpha5=min($Alphaipk1,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
623     $alpha6=min($Alphaipk2,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis1);
624     $alpha7=min($Alphaipk2,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis2);
625     $alpha8=min($Alphaipk2,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis1);
626     $alpha9=min($Alphaipk2,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
627     $alpha10=min($Alphaipk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
628     $alpha11=min($Alphaipk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis1);
629     $alpha12=min($Alphaipk2,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis1);
630     $alpha13=min($Alphaipk2,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis2);
631     $alpha14=min($Alphaipk1,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis2);
632     $alpha15=min($Alphaipk1,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis2);
633     $alpha16=min($Alphaipk2,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis2);
634     $alpha17=min($Alphaipk1,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis2);
635     $alpha18=min($Alphaipk2,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis2);
636     $alpha19=min($Alphaipk1,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis2);
637     $alpha20=min($Alphaipk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis2);
638     $alpha21=min($Alphaipk1,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis1);
639     $alpha22=min($Alphaipk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis1);
640     $alpha23=min($Alphaipk1,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis1);
641     $alpha24=min($Alphaipk2,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis1);
642     $alpha25=min($Alphaipk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis1);
643     $alpha26=min($Alphaipk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis1);
644     $alpha27=min($Alphaipk1,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis1);
645     $alpha28=min($Alphaipk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
646
647     $nilaiideal=$this->db->query("select * from tb_nilaiidealskala");
648     foreach ($nilaiideal->result_array() as $nilai) {
649         $ipkideals=$nilai['ipk'];
650         $ijazahideals=$nilai['ijazah'];
651         $wawancaraideals=$nilai['wawancara'];
652         $microteachingideals=$nilai['microteaching'];
653         $testulisideals=$nilai['testulis'];
654     }
655
656     $nilaimin=min($ipkideals,$ijazahideals,$wawancaraideals,$microteachingideals,$testulisideals);
657     $gapmin=1;
658     return $alpha1-$gapmin;
659 }

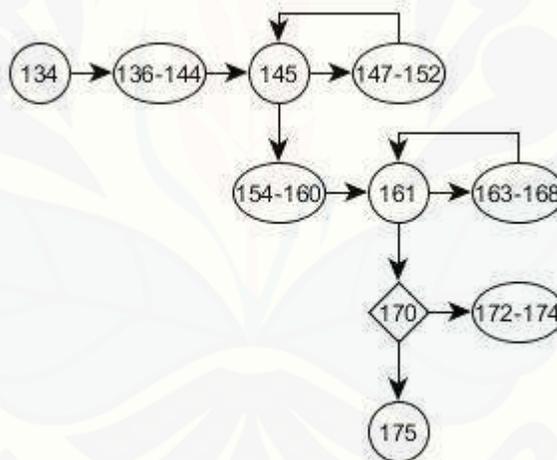
```

```

589 $z8= $Zmax ($alpha8*($Zmax-$Zmin));
590 $z9= $Zmax ($alpha9*($Zmax-$Zmin));
591 $z10= $Zmax ($alpha10*($Zmax-$Zmin));
592 $z11= $Zmax ($alpha11*($Zmax-$Zmin));
593 $z12= $Zmax ($alpha12*($Zmax-$Zmin));
594 $z13= $Zmax ($alpha13*($Zmax-$Zmin));
595 $z14= $Zmax ($alpha14*($Zmax-$Zmin));
596 $z15= $Zmax ($alpha15*($Zmax-$Zmin));
597 $z16= $Zmax ($alpha16*($Zmax-$Zmin));
598 $z17= $Zmax ($alpha17*($Zmax-$Zmin));
599 $z18= $Zmax ($alpha18*($Zmax-$Zmin));
600 $z19= $Zmax ($alpha19*($Zmax-$Zmin));
601 $z20= $Zmax ($alpha20*($Zmax-$Zmin));
602 $z21= $Zmax ($alpha21*($Zmax-$Zmin));
603 $z22= $Zmax ($alpha22*($Zmax-$Zmin));
604 $z23= $Zmax ($alpha23*($Zmax-$Zmin));
605 $z24= $Zmax ($alpha24*($Zmax-$Zmin));
606 $z25= $Zmax ($alpha25*($Zmax-$Zmin));
607 $z26= $Zmax ($alpha26*($Zmax-$Zmin));
608 $z27= $Zmax ($alpha27*($Zmax-$Zmin));
609 $z28= $Zmax ($alpha28*($Zmax-$Zmin));
610 $alphasema = array($alpha1, $alpha2, $alpha3, $alpha4,$alpha5,$alpha6,$alpha7,$alpha8,$alpha9,$alpha10,$alpha11,$alpha12,$alpha13,$alph
611 $zsema = array($z1,$z2,$z3,$z4,$z5,$z6,$z7,$z8,$z9,$z10,$z11,$z12,$z13,$z14,$z15,$z16,$z17,$z18,$z19,$z20,$z21,$z22,$z23,$z24,$z25,$z26
612
613 $jumlahalpha=($alpha1+$alpha2+$alpha3+$alpha4+$alpha5+$alpha6+$alpha7+$alpha8+$alpha9+$alpha10+$alpha11+$alpha12+$alpha13+$alpha14+$alph
614 $jumlahz= (($z1*$alpha1)+($z2*$alpha2)+($z3*$alpha3)+($z4*$alpha4)+($z5*$alpha5)+($z6*$alpha6)+($z7*$alpha7)+($z8*$alpha8)+($z9*$alpha9)+
615
616 $zrata2=($jumlahz)/($jumlahalpha);
617 $query= $this->db->query("UPDATE tb_nilai SET nilaiAkhir='$zrata2' where iduser='$idUser'");
618 echo "rata2"; echo $zrata2;
619 return array($alphasema,$zsema,$query);
620 }
    
```

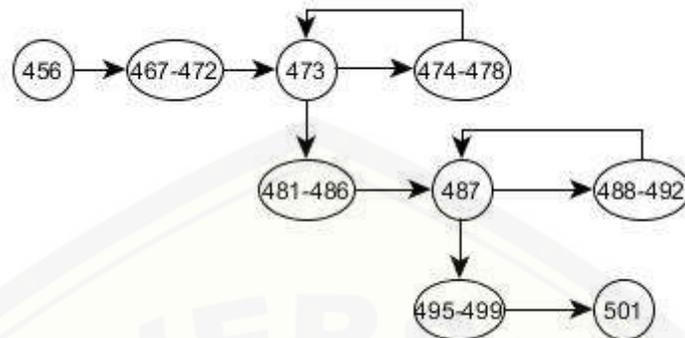
Gambar 4. 10 Listing program function hitungratarata()

2) Diagram Alir *hitungnilaipendaftar()*

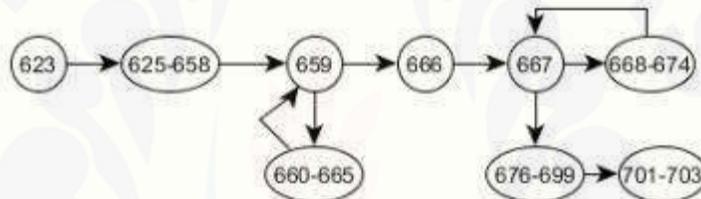


Gambar 4. 11 Diagram alir function *hitungnilai pendaftar(\$var)*

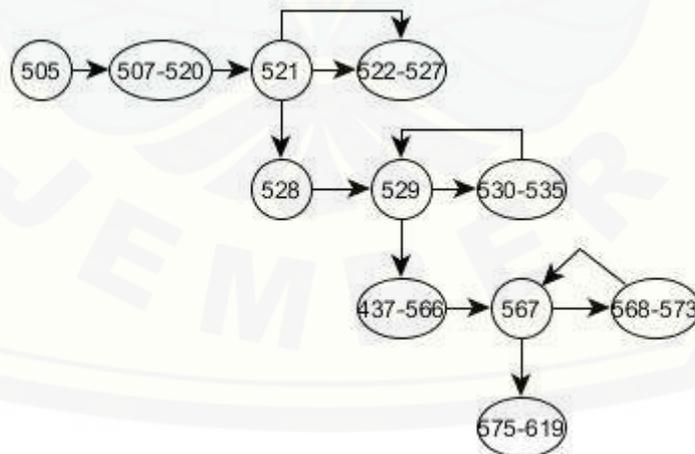
dalam proses *hitungnilai pendaftar()* terdapat fungsi lain yang harus dijalankan maka diagram alir untuk metode profile matching tidak berhenti didiagram atas, melainkan juga diagram alir untuk fungsi *hitunggap*, *fuzzyfikasi()*, dan *hitungratarata()* yang ada dibawah ini yakni pada gambar 4.12 sampai gambar 4.14. kode program untuk perhitungan gap dimulai dari *line* 456 hingga *line* 501



Gambar 4. 12 Diagram alir *function hitunggap(\$var)*  
 kode program untuk *fuzzifikasi* dimulai dari *line 623* hingga *line 703*



Gambar 4. 13 Diagram alir *function fuzzyfikasi(\$var)*  
 kode program untuk *hitungratarata()* dimulai dari *line 506* hingga *line 619*



Gambar 4. 14 Diagram alir *function hitungratarata(\$var)*

3) Kompleksitas Siklomatik (*Cyclomatic Complexity*)

Function hitungnilai pendaftar(), menghitung edge dan node yang dilalui dari hasil yang didapatkan yaitu 6, maka dapat disimpulkan bahwa kode program tergolong sederhana.

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

$$V(G) = 43 - 39 + 2 = 6$$

4) Pengujian Basis Set (*Test Case*)

hitung nilai pendaftar

134-(136-144)-145-(156-160)-161-170-175

134-(136-144)-145-(156-160)-161-170-(172-174)

Terdapat 2 kemungkinan jalur untuk implementasi metode *profile matching* dan logika *fuzzy* dimana dua kemungkinan tersebut akan dilalui juga jalur dibawah ini:

a. hitung gap

146-(146-272)-273-(481-486)-487-(495-499)-501

b. fuzzifikasi

623-(625-658)-659-666-667-(676-699)-(701-703)

c. hitung rata-rata

505-(507-520)-521-528-529-(473-566)-567-(575-619)

Pengujian Basis Set (*Test Case*) atau pelaksanaan jalur di basis set dicontohkan pada tabel 4.17.

Tabel 4. 4 *Test case function hitungnilai pendaftar()*

<i>Test case function hitungnilai pendaftar()</i>	
<b>Jalur 1</b>	
<b>Test Case</b>	Jika nilai pendaftar tidak melampaui nilai minimal
<b>Target yang diharapkan</b>	Menampilkan nilai kriteria pendaftar , tanpa menghitung nilai akhir dan menghapus nilai akhir
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	134-(136-144)-145-(156-160)-161-170-175
<b>Jalur 2</b>	

<b>Test Case</b>	Jika nilai pendaftar memenuhi nilai standar
<b>Target yang diharapkan</b>	Menampilkan nilai kriteria pendaftar, menghitung nilai kriteria dengan perhitungan metode
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	134-(136-144)-145-(156-160)-161-170-(172-174)
<b>Jalur 3</b>	
<b>Test Case</b>	Jika nilai pendaftar memenuhi nilai standar
<b>Target yang diharapkan</b>	Menjalankan perhitungan nilai gap
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	146-(146-272)-273-(481-486)-487-(495-499)-501
<b>Jalur 4</b>	
<b>Test Case</b>	Jika nilai pendaftar memenuhi nilai standar
<b>Target yang diharapkan</b>	Menghitung fuzzifikasi nilai kriteria
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	623-(625-658)-659-666-667-(676-699)-(701-703)
<b>Jalur 5</b>	
<b>Test Case</b>	Jika nilai pendaftar memenuhi nilai standar
<b>Target yang diharapkan</b>	Menghitung defuzzifikasi dari nilai kriteria
<b>Hasil Pengujian</b>	Benar
<b>Path / jalur</b>	505-(507-520)-521-528-529-(473-566)-567-(575-619)

##### 5) Metode *Black Box*

*Black box testing* merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari software. Pengujian *black box* pada sistem sistem pendukung keputusan seleksi penerimaan pengajar menggunakan logika fuzzy dan *Profile Matching* dilakukan untuk mengetahui apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada setiap *usecase*. Hasil pengujian *black box* dapat dilihat pada Lampiran E (Pengujian *Black Box*).

## 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 penelitian yang telah dilakukan adalah:

- a. Implementasi logika *Fuzzy* dan metode *Profile Matching* dalam Sistem Pendukung Keputusan dilakukan untuk menentukan penerimaan pengajar di Lembaga Bimbingan Galileo. Langkah awal untuk melakukan perangkaian yaitu dengan menentukan kriteria penerimaan. Langkah berikutnya yaitu menentukan nilai standar yang dibutuhkan oleh LBB Galileo, dari nilai standart inilah akan di dapati rentang nilai antar nilai kompetensi data tes dengan kompetensi yang diharapkan atau disebut dengan nilai gap, gap kemudian di beri bobot, bobot gap inilah yang kemudian di hitung menggunakan logika fuzzy. Terdapat 5 data tes pada penulisan ini, kemudian didapati bahwa rangking sangat dipengaruhi oleh konversi nilai bobot, dilihat dari beberapa persamaan yang ada, yakni proses fuzzifikasi, mesin inferensi, dan defuzzifikasi. Nilai akhir adalah hasil rata-rata terbobot maka nilai penyebut yang lebih kecil akan menghasilkan nilai akhir yang besar, sedangkan penyebut merupakan jumlah dari  $\alpha$ -predikat. Nilai  $\alpha$ -predikat sangat dipengaruhi oleh data bobot dari gap terbesar, seperti pada pembahasan diketahui rangking terbesar adalah bernilai 0.674753695 dikarenakan penyebut saat dilakukan pembagian adalah penyebut paling kecil yaitu 10.15 yang merupakan penjumlahan dari  $\alpha$ -predikat, sedang  $\alpha$ -predikat didapatkan dari nilai minimal tiap baris rule, dan value tiap atribut dalam baris rule yang digunakan adalah hasil fuzzyfikasi menggunakan bobot gap yang terbilang kecil, untuk nilai gap yang diperoleh dari nilai minimal di kurangi nilai pendaftar. Dari 5 data test, menunjukkan rangking pertama diperoleh tester 1

dengan nilai 0.674754, ranking kedua yaitu tester 4 dengan nilai 0.624576, ranking ketiga yakni tester 3 dengan nilai 0.585686, ranking keempat dan ke lima oleh tester 5 dan tester 2 dengan masing-masing nilai 0.543151 dan 0.5. Maka implementasi profile matching menghasilkan ranking yang dapat menjadi pendukung dalam keputusan penerimaan pengajar di LBB Galileo.

- b. Sistem Pendukung Keputusan dilakukan untuk menentukan penerimaan pengajar di Lembaga Bimbingan Galileo dibangun berbasis website dengan dua hak akses yaitu admin, litbang dan user tanpa hak akses yakni pendaftar dengan fitur utama perangkingan pendaftar. Sistem ini dikembangkan dengan menggunakan model waterfall, yang diawali dengan analisa kebutuhan sistem. Analisis kebutuhan dilakukan dengan cara mendefinisikan sistem secara detail sehingga mendapatkan kebutuhan fungsional dan non fungsional yang lengkap. Tahap selanjutnya adalah pengembangan yang meliputi analisis, desain, implementasi desain kedalam kode program dan pengujian sistem. Penggunaan model waterfall pada penelitian ini sangat sesuai, karena menerapkan sekuensial prosedur atau keterurutan dan sangat mementingkan proses dokumentasi, sehingga membantu pengembang pemula seperti peneliti.

## 6.2 Saran

Adapun saran yang ditujukan untuk membe'rikan masukan yang lebih baik yaitu :

- a. Sistem dengan implementasi Logika *Fuzzy* dan *Profile Matching* ini sudah menghasilkan ranking yang logis, namun kriteria tidak fleksibel hingga kedepan diharapkan akan ditemukan algoritma untuk menghasilkan kombinasi rule.
- b. Sistem seleksi penerimaan pengajar dengan mengimplementasikan perhitungan Logika *Fuzzy* dan *Profile Matching* ini masih perlu ketegasan dalam menyebut nilai ideal dan nilai minimal atau nilai standar.

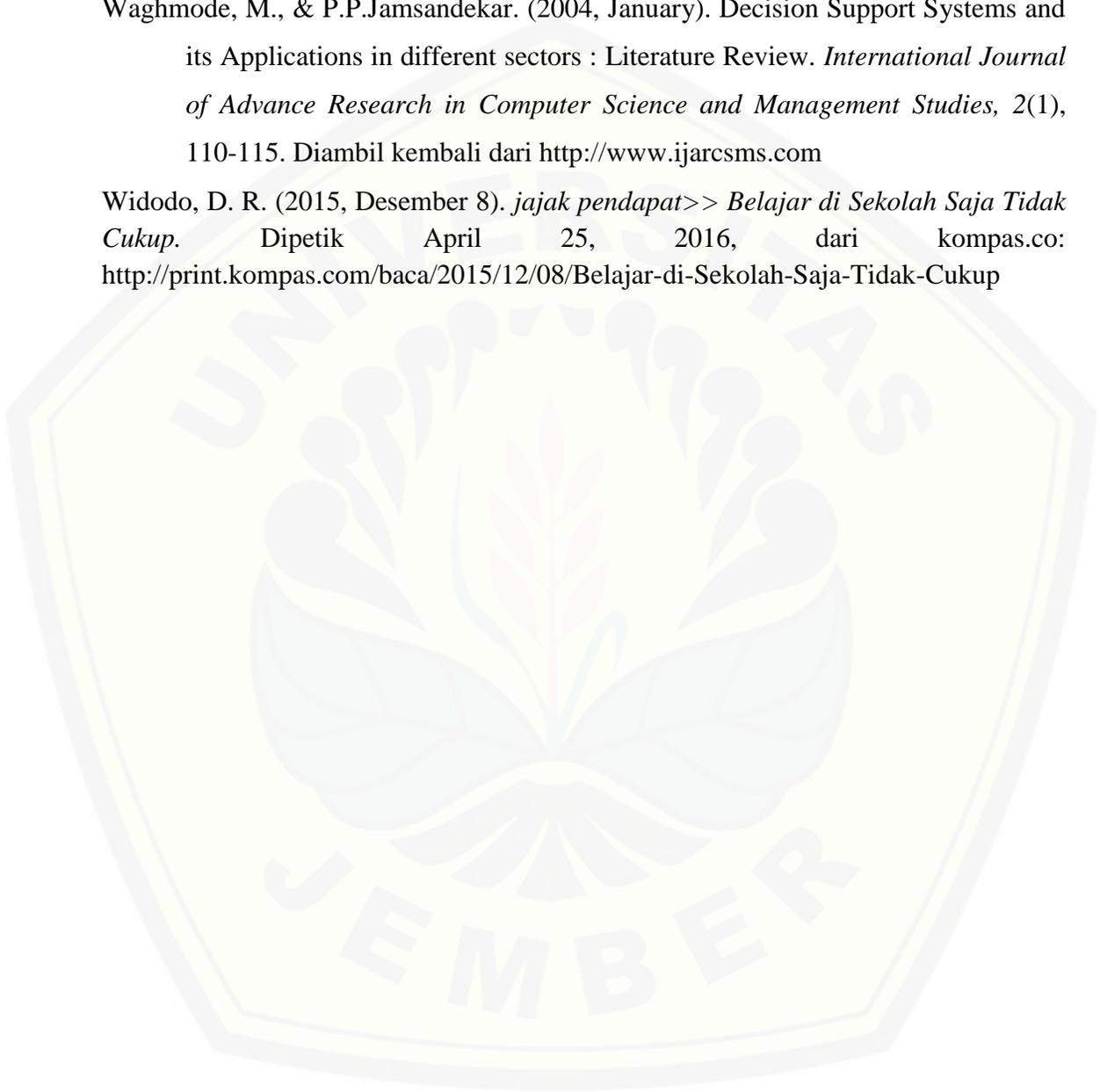
**DAFTAR PUSTAKA**

- Andreas Handoyo, D. H. (2003). Pembuatan Aplikasi Sistem Pendukung Keutusan untuk Proses Kenaikan Jabatan dan Perencanaan Karir pada PT. X. *jurnal informatika*, 101-105.
- Dr.Suherman, M. (2001). *Bimbingan Belajar*. Solo: Universitas Pendidikan Indonesia.
- Jumaidi, C. N. (2015). Pendekatan Logika Fuzzy untuk Perhitungan Gap pada Metode Profile MAtching dalam Menentukan Kelayakan Ptoposal. *Seminar Nasional Sains dan Teknologi* . Bandung: Jurusan Teknik Informatikan Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung.
- Kusumaning Hati Pembayun, R. A. (t.thn.). Sistem Pendukung Keputusan Penerimaan Asisten Praktikum Menggunakan Metode Profile Matching.
- McMillan, J. H. (2006). *Research in Education*. New Jersey: Person.
- Muhammad Taufik Irawan, D. K. (2016). Penerapan Profile Matching untuk Pencarian Siswa SMP Penerima Beasiswa Miskin dan Berprestasi. *Jurnal Informatika dan Komputer(JIKO)-vol. 1, No. 1*.
- Pressman, R. S. (2001). *Software Engineering a practitioner's approach* (5 ed.). New York, America: McGraw-Hill.
- Rouf, A. (t.thn.). *Pengujian Perangkat Lunak dengan Menggunakan Metode White Box dan Bleck Box*. Semarang: Sistem Informasi – STMIK HIMSYA Semarang.
- Sommerville, I. (2001). *Software Engineering*. Addison Wesley.
- Sri Kusumadewi, H. P. (2004). *Aplikasi Logika Fuzzy untuk pendukung keputusan* . Yogyakarta: Graha Ilmu.
- Subakti, I. (2002 ). *IF1526 Sistem Berbasis Pengetahuan*. Surabaya: Jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember Surabaya.
- Sugiyono. (2010). *Metode Penelitian Kuantitatif Kualitatif & RND*. Bandung: Person.

*Undang Undang Republik Indonesia Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional.* (2003).

Waghmode, M., & P.P.Jamsandekar. (2004, January). Decision Support Systems and its Applications in different sectors : Literature Review. *International Journal of Advance Research in Computer Science and Management Studies*, 2(1), 110-115. Diambil kembali dari <http://www.ijarcsms.com>

Widodo, D. R. (2015, Desember 8). *jajak pendapat>> Belajar di Sekolah Saja Tidak Cukup.* Dipetik April 25, 2016, dari [kompas.co: http://print.kompas.com/baca/2015/12/08/Belajar-di-Sekolah-Saja-Tidak-Cukup](http://print.kompas.com/baca/2015/12/08/Belajar-di-Sekolah-Saja-Tidak-Cukup)



## LAMPIRAN

## LAMPIRAN A. USE CASE SKENARIO

## A.1 Use Case Skenario Login

Tabel A. 1 Use case skenario login

Nama use case	<i>Login</i>
Aktor	Admin, LITBANG
Pre Kondisi	Aktor menjalankan sistem
Post Kondisi	Aktor telah melakukan <i>login</i>
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih tombol <i>login</i>	2. Menampilkan halaman <i>login</i> <ul style="list-style-type: none"> <li>• Kolom <i>username</i></li> <li>• Kolom <i>password</i></li> <li>• Tombol <i>Login</i></li> <li>• Tombol <i>Reset</i></li> </ul>
3. Memasukkan <i>username</i>	
4. Memasukkan <i>password</i>	
5. Pilih tombol <i>login</i>	6. Melakukan autentikasi <i>username</i> dan <i>password</i> yang telah dimasukkan, dengan dibandingkan dengan data <i>username</i> dan <i>password</i> yang ada di <i>tb_akses</i> dari database
	7. Menampilkan halaman awal sistem.
Skenario Alternatif (username atau password tidak sesuai)	
Aksi Aktor	Reaksi Sistem
5a. Pilih OK.	4a. Menampilkan pesan 'username dan password salah' dengan tombol OK
	6a. Menampilkan halaman depan sistem tombol <i>login</i>
Skenario Alternatif (username atau password kosong/tidak diisi)	
Aksi Aktor	Reaksi Sistem
	4b. Menampilkan pesan " <i>please fill out this field</i> " dengan menunjuk pada kolom yang belum diisi

## A.2 Use Case Skenario Mengelola Data Profil User

Tabel A. 2 Use case skenario mengelola data profil user

Nama Use Case	Mengelola Data Profil <i>user</i>
Aktor	Admin,LITBANG
Pre Kondisi	Aktor sudah melakukan <i>login</i> pada sistem
Post Kondisi	Aktor dapat mengelola data profil
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih 'profile'	2. Mengambil data profil <i>user</i> <ul style="list-style-type: none"> <li>• Id user</li> <li>• nama lengkap</li> <li>• alamat</li> <li>• tempat lahir</li> <li>• tanggal lahir</li> <li>• email</li> <li>• nomor telepon</li> </ul> dari <i>tb_karyawan</i> di <i>database</i>
	3. Menampilkan halaman profil, dengan, <ul style="list-style-type: none"> <li>• kolom nama lengkap</li> <li>• kolom alamat</li> <li>• kolom tempat lahir</li> <li>• kolom tanggal lahir</li> <li>• kolom nomor telepon</li> <li>• kolom email</li> <li>• tombol ubah profil</li> <li>• tombol ubah <i>password</i></li> </ul>
4. Pilih tombol 'ubah profil' Jika pilih tombol 'ubah password' lanjut ke nomor 4a.	5. Mengambil data profil <i>user</i> <ul style="list-style-type: none"> <li>• Id user</li> <li>• nama lengkap</li> <li>• alamat</li> <li>• tempat lahir</li> <li>• tanggal lahir</li> <li>• email</li> <li>• nomor telepon</li> </ul> dari <i>tb_karyawan</i> di <i>database</i>
	6. Menampilkan halaman form edit profil dengan kolom, <ul style="list-style-type: none"> <li>• alamat</li> <li>• tempat lahir</li> <li>• tanggal lahir</li> <li>• nomor telepon</li> </ul>

	<ul style="list-style-type: none"> <li>• email</li> </ul> <p>dan <i>submit button</i> dan <i>reset button</i></p>
7. Mengubah isi kolom	
8. Pilih simpan Jika <i>reset</i> lanjut ke nomor 7b.	
	<p>9. menyimpan data profil baru</p> <ul style="list-style-type: none"> <li>• nama lengkap</li> <li>• alamat</li> <li>• tempat lahir</li> <li>• tanggal lahir</li> <li>• email</li> <li>• nomor telepon</li> </ul> <p>ke <i>tb_karyawan</i> di <i>database</i></p>
	<p>10. Mengambil data profil <i>user</i></p> <ul style="list-style-type: none"> <li>• nama lengkap</li> <li>• alamat</li> <li>• tempat lahir</li> <li>• tanggal lahir</li> <li>• email</li> <li>• nomor telepon</li> </ul> <p>dari <i>tb_karyawan</i> di <i>database</i></p>
	<p>11. Menampilkan halaman profil, dengan tombol ubah profil dan ubah <i>password</i></p>
<b>Skenario Normal (menekan ubah password)</b>	
4a. Pilih tombol 'ubah password'	
	<p>5a. mengambil data <i>password</i> lama dr <i>tb_akses</i> di <i>database</i></p>
	<p>6a. Menampilkan halaman form edit <i>password</i> dengan kolom</p> <ul style="list-style-type: none"> <li>• <i>Password</i> lama</li> <li>• <i>Password</i> baru</li> </ul> <p>dan tombol <i>submit button</i> dan <i>reset button</i></p>
7a. mengisi password baru	
8a. Pilih simpan Jika <i>reset</i> lanjut ke nomor 7b.	
	<p>9a. Menyimpan data <i>password</i> baru ke <i>tb_akses</i> di <i>database</i>.</p>
	<p>10a. Mengambil data profil <i>user</i></p> <ul style="list-style-type: none"> <li>• nama lengkap</li> <li>• alamat</li> <li>• tempat lahir</li> <li>• tanggal lahir</li> <li>• email</li> </ul>

	<ul style="list-style-type: none"> <li>• nomor telepon dari tb_karyawan di <i>database</i></li> </ul>
	11a. Menampilkan halaman profil, dengan tombol “ubah profil” dan tombol “ubah <i>password</i> ”
Skenario Alternatif (kolom kosong/tidak diisi)	
Aksi Aktor	Reaksi system
	8. Menampilkan pesan “please fill out this field” dengan menunjuk pada kolom yang belum diisi.
Skenario Alternatif	
Aksi Aktor	Reaksi system
7b. Pilih <i>reset</i>	8b. Menampilkan halaman edit profil dengan data awal, tombol <i>submit button</i> dan <i>reset button</i>

### A.3 Use Case Skenario Mengelola Data Akses

Tabel A. 3 Use case skenario mengelola data akses

Nama Use Case	Mengelola Data Akses
Aktor	Admin
Pre Kondisi	Aktor sudah melakukan <i>login</i> pada sistem
Post Kondisi	Aktor dapat menggunakan system untuk mengelola data akses
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih menu ‘data akses’	2. Mengambil data akses yaitu <ul style="list-style-type: none"> <li>• nama lengkap dari tb_karyawan</li> <li>• username</li> <li>• password dari tb_akses di <i>database</i></li> </ul>
	3. Menampilkan halaman dengan daftar hak akses, tombol ‘tambah hak akses’, tombol ‘ <i>edit</i> ’, dan tombol ‘hapus’
4. Pilih tombol ‘tambah hak akses’ Jika pilih tombol ‘ <i>edit</i> ’ lanjut ke 4a, Jika pilih tombol ‘hapus’ ke 4b’	5. Menampilkan form untuk tambah data hak akses dengan,

	<ul style="list-style-type: none"> <li>• Kolom nama lengkap</li> <li>• Kolom <i>username</i></li> <li>• Kolom <i>password</i></li> <li>• <i>Combobox</i> level</li> <li>• <i>Combobox</i> status</li> </ul> <p>Dan <i>submit button</i> dan <i>reset button</i></p>
6. Mengisi kolom pada form tambah hak akses	
7. pilih 'submit button' jika pilih 'reset button' lanjut ke 7c.	
	<p>8. menyimpan data akses yang dimasukkan</p> <ul style="list-style-type: none"> <li>• nama lengkap</li> <li>• <i>username</i></li> <li>• <i>password</i></li> <li>• level</li> <li>• status</li> </ul> <p>ke <i>database</i> pada <i>tb_akses</i></p>
	<p>9. Mengambil data akses yaitu</p> <ul style="list-style-type: none"> <li>• nama lengkap</li> <li>• <i>username</i></li> <li>• <i>password</i></li> </ul> <p>dari <i>tb_akses</i> di <i>database</i></p>
	<p>10. Menampilkan halaman dengan daftar hak akses, tombol 'tambah hak akses', tombol 'edit', dan tombol 'hapus'</p>
Skenario alternative (pilih tombol 'edit')	
Aksi Aktor	Reaksi Sistem
4a. Pilih tombol 'edit'	
	<p>5a. Mengambil data akses</p> <ul style="list-style-type: none"> <li>• <i>user id</i></li> <li>• <i>username</i></li> <li>• level</li> <li>• status</li> </ul> <p>dari <i>tb_akses</i> di <i>database</i></p>
	<p>6a. Menampilkan data akses yang terpilih ke halaman form edit dengan,</p> <ul style="list-style-type: none"> <li>• Kolom <i>user id</i> (<i>read only</i>)</li> <li>• Kolom <i>username</i> (<i>read only</i>)</li> <li>• <i>Combobox</i> level</li> <li>• <i>Combobox</i> status</li> </ul> <p>Dan <i>Submit button</i> dan tombol</p>

	<i>back</i>
6a. Mengisi kolom pada form edit hak akses	
7a. Pilih 'submit button' jika pilih tombol 'back' lanjut ke 7d.	
	8a. mengubah data <ul style="list-style-type: none"> <li>• <i>user id</i></li> <li>• <i>username</i></li> <li>• level</li> <li>• status</li> </ul> yang ada <i>didatabase</i> pada <i>tb_akses</i> dengan data yang diisikan di form
	9a. Mengambil data akses <ul style="list-style-type: none"> <li>• <i>user id</i></li> <li>• <i>username</i></li> <li>• level</li> <li>• status</li> </ul> dari <i>tb_akses</i> di <i>database</i>
	10a. Menampilkan halaman dengan daftar hak akses, tombol 'tambah hak akses', tombol 'edit', dan tombol 'hapus'
<b>Skenario alternative (pilih tombol 'hapus')</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
4b. Pilih tombol 'hapus'	5b. Mengubah status akses data yang telah terpilih pada <i>tb_akses</i> di <i>database</i> menjadi status akses "tidak diizinkan"
	6b. Mengambil data akses yaitu <ul style="list-style-type: none"> <li>• nama lengkap</li> <li>• username</li> <li>• password</li> </ul> dari <i>tb_akses</i> di <i>database</i>
	7b. Menampilkan halaman dengan daftar hak akses, tombol 'tambah hak akses', tombol 'edit', dan tombol 'hapus'
<b>Skenario alternative (pilih 'reset button')</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
7c. pilih reset button	8c. menampilkan form pada keadaan awal
<b>Skenario alternative (pilih tombol 'back')</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>

	7d. Mengambil data akses yaitu <ul style="list-style-type: none"> <li>• nama lengkap</li> <li>• username</li> <li>• password</li> </ul> dari <i>tb_ akses</i> di <i>database</i>
	8d Menampilkan halaman dengan daftar hak akses, tombol ‘tambah hak akses’, tombol ‘ <i>edit</i> ’, dan tombol ‘hapus’
Skenario Alternatif (kolom kosong/tidak diisi)	
Aksi Aktor	Reaksi Sistem
	5 Menampilkan pesan “ <i>please fill out this field</i> ” dengan menunjuk pada kolom yang belum diisi.
Skenario Alternatif (mengisi kolom pencarian)	
Aksi Aktor	Reaksi Sistem
1. Mengisi nama <i>user</i>	
2. Memilih tombol ‘Go’	
	3. Memilih data yang sesuai dengan pencarian pada <i>database</i>
	4. Menampilkan halaman dengan daftar data akses sesuai nama yang dicari

A.4 Use Case Skenario Mengelola Nilai Pendaftar

Tabel A. 4 Use case skenario mengelola nilai pendaftar

Nama Use Case	Mengelola nilai pendaftar
Aktor	Admin
Pre Kondisi	Aktor sudah melakukan <i>login</i> sistem
Post Kondisi	Aktor dapat menggunakan sistem untuk mengelola nilai pendaftar
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih menu ‘input nilai’	
	2. mengambil daftar nilai peserta dengan <ul style="list-style-type: none"> <li>• id pendaftar</li> <li>• nama lengkap pendaftar</li> </ul> di <i>tb_pendaftar</i> dan <ul style="list-style-type: none"> <li>• IPK</li> <li>• lulusan</li> <li>• nilai wawancara</li> <li>• nilai <i>Microteaching</i></li> </ul>

	<ul style="list-style-type: none"> <li>• nilai tes tulis</li> <li>• nilai akhir</li> </ul>
	di <i>tb_nilai</i> dari <i>database</i>
	3. menampilkan daftar peserta pada halaman input nilai dan tombol 'edit'
4. Pilih icon 'pensil'	
	5. mengambil data nilai peserta yang id terpilih <ul style="list-style-type: none"> <li>• id pendaftar</li> <li>• nama lengkap pendaftar</li> </ul>
	di <i>tb_pendaftar</i> dan <ul style="list-style-type: none"> <li>• IPK</li> <li>• lulusan</li> <li>• nilai wawancara</li> <li>• nilai <i>Microteaching</i></li> <li>• nilai tes tulis</li> <li>• nilai akhir</li> </ul>
	di <i>tb_nilai</i> dari <i>database</i>
	6. menampilkan data nilai yang dimaksud ke halaman form ubah nilai, dengan <ul style="list-style-type: none"> <li>• kolom id pendaftar (<i>read only</i>)</li> <li>• kolom nama (<i>read only</i>)</li> <li>• kolom IPK</li> <li>• <i>combobox</i> lulusan</li> <li>• kolom nilai wawancara</li> <li>• kolom nilai <i>Microteaching</i></li> <li>• kolom nilai tes tulis</li> <li>• kolom nilai akhir (<i>read only</i>)</li> </ul>
	dan 'submit button' dan tombol 'back'.
7. Mengisi data nilai pendaftar	
8. Pilih 'submit button'	
	Jika pilih 'back' lanjut ke 8a.
	9. Mengubah data nilai <ul style="list-style-type: none"> <li>• IPK</li> <li>• lulusan</li> <li>• nilai wawancara</li> <li>• nilai <i>Microteaching</i></li> <li>• nilai tes tulis</li> </ul>
	di <i>database</i> tabel <i>tb_nilai</i> sesuai yang di isikan di <i>form</i> input nilai, menghitung hasil perubahan nilai atribut menggunakan metode <i>Profile Matching</i> dan logika <i>fuzzy</i> dan

	menyimpan hasil akhirnya pada atribut nilaiAkhir tabel tb_nilai di <i>database</i>
	10. mengambil daftar nilai peserta dengan <ul style="list-style-type: none"> <li>• id pendaftar</li> <li>• nama lengkap pendaftar</li> </ul> di tb_pendaftar dan <ul style="list-style-type: none"> <li>• IPK</li> <li>• lulusan</li> <li>• nilai wawancara</li> <li>• nilai <i>Microteaching</i></li> <li>• nilai tes tulis</li> <li>• nilai akhir</li> </ul> di tb_nilai dari <i>database</i>
	11. menampilkan daftar peserta pada halaman input nilai dan tombol 'edit'
<b>Skenario Alternatif (memilih tombol 'back')</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
8a. Pilih 'back'	9a. mengambil daftar nilai peserta dengan <ul style="list-style-type: none"> <li>• id pendaftar</li> <li>• nama lengkap pendaftar</li> </ul> di tb_pendaftar dan <ul style="list-style-type: none"> <li>• IPK</li> <li>• lulusan</li> <li>• nilai wawancara</li> <li>• nilai <i>Microteaching</i></li> <li>• nilai tes tulis</li> <li>• nilai akhir</li> </ul> di tb_nilai dari <i>database</i>
	10a. menampilkan daftar peserta pada halaman input nilai dan tombol 'edit'
<b>Skenario Alternatif (melakukan pencarian)</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. memilih <i>combobox</i> mata pelajaran	
2. memilih <i>combobox</i> jenjang	
3. memilih tombol 'Go'	
	4. mengambil daftar nilai pendaftar sesuai dengan kategori pencarian
	5. menampilkan daftar peserta pada halaman input nilai dan tombol

'edit'

A.5 Use Case Skenario Menampilkan data profile pendaftar

Tabel A. 5 Use case skenario menampilkan data profile pendaftar

Nama UseCase	Menampilkan data profile pendaftar
Aktor	Admin
Pre Kondisi	Aktor sudah melakukan <i>login</i> sistem
Post Kondisi	Aktor dapat melihat profile pendaftar
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih menu 'data pendaftar'	2. Mengambil data pendaftar <ul style="list-style-type: none"> <li>• Nama lengkap dari tabel <i>tb_pendaftar</i> dan</li> <li>• Nilai Akhir dari tabel <i>tb_nilai</i> database</li> </ul>
	3. menampilkan halaman dengan daftar pendaftar dan tombol 'detail'
4. Pilih tombol detail	
5. Pilih sub menu 'profil pendaftar'	6. Mengambil data pendaftar <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama lengkap</li> <li>• Alamat</li> <li>• Tempat lahir</li> <li>• Tanggal lahir</li> <li>• Jenis identitas</li> <li>• Nomor identitas</li> <li>• Nomor telepon</li> <li>• Email</li> <li>• Bidang keahlian</li> <li>• Jenjang</li> <li>• Status terima</li> </ul> dari <i>tb_pendaftar</i> database sesuai dengan id yang dipilih dan <ul style="list-style-type: none"> <li>• Ipk</li> <li>• Lulusan</li> </ul> Dari <i>tb_nilai</i> di <i>database</i>
	7. Menampilkan halaman profil pendaftar
Skenario Alternatif (melakukan pencarian)	
Aksi Aktor	Reaksi Sistem
1. memilih <i>combobox</i> mata	

pelajaran	
2. memilih <i>combobox</i> jenjang	
3. memilih tombol 'Go'	
	8. Mengambil data pendaftar <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama lengkap</li> <li>• Alamat</li> <li>• Tempat lahir</li> <li>• Tanggal lahir</li> <li>• Jenis identitas</li> <li>• Nomor identitas</li> <li>• Nomor telepon</li> <li>• Email</li> <li>• Bidang keahlian</li> <li>• Jenjang</li> <li>• Status terima</li> </ul> dari <i>tb_pendaftar</i> database sesuai dengan id yang dipilih dan <ul style="list-style-type: none"> <li>• Ipk</li> <li>• Lulusan</li> </ul> dari <i>tb_nilai</i> di <i>database</i>
	4. menampilkan halaman dengan daftar pendaftar dan tombol 'detail'

A.6 Use Case Skenario Menampilkan Data Nilai Pendaftar

Tabel A. 6 Use Case Skenario menampilkan data nilai pendaftar

Nama Use Case	Menampilkan data nilai pendaftar
Aktor	Admin
Pre Kondisi	Aktor sudah melakukan <i>login</i> sistem
Post Kondisi	Aktor telah dapat melihat nilai pendaftar
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih menu 'data pendaftar'	2. Mengambil data nama pendaftar dari tabel <i>tb_pendaftar</i> dan nilai akhir dari tabel <i>tb_nilai</i> dari <i>database</i>
	3. Menampilkan halaman daftar pendaftar dan tombol 'detail'
4. Pilih tombol 'detail'	9. Mengambil data pendaftar <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama lengkap</li> </ul>

	<ul style="list-style-type: none"> <li>• Alamat</li> <li>• Tempat lahir</li> <li>• Tanggal lahir</li> <li>• Jenis identitas</li> <li>• Nomor identitas</li> <li>• Nomor telepon</li> <li>• Email</li> <li>• Bidang keahlian</li> <li>• Jenjang</li> <li>• Status terima</li> </ul> <p>dari tb_pendaftar database sesuai dengan id yang dipilih dan</p> <ul style="list-style-type: none"> <li>• Ipk</li> <li>• Lulusan</li> </ul> <p>Dari tb_nilai di <i>database</i> sesuai id yang terpilih</p>
	5. Menampilkan data profil pendaftar sesuai id yang terpilih ke halaman detail profil
6. Pilih sub menu 'nilai pendaftar'	
	7. Mengambil data pendaftar <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama lengkap pendaftar</li> </ul> <p>Dari tb_pendaftar <i>didatabase</i></p> <ul style="list-style-type: none"> <li>• Nilai IPK</li> <li>• Lulusan</li> <li>• Nilai wawancara</li> <li>• Nilai test tulis</li> <li>• Nilai tes microteaching</li> <li>• Nilai Akhir</li> </ul> <p>Dari tabel tb_nilai di <i>database</i></p>
	8. Menampilkan data nilai pendaftar ke halaman nilai pendaftar dan tombol 'Show detail'
9. Pilih tombol 'show detail'	
	10. Menampilkan detail perhitungan nilai kriteria dengan <i>Profile Matching</i> dan logika <i>fuzzy</i>
Skenario Alternatif (melakukan pencarian)	
Aksi Aktor	Reaksi Sistem
1. memilih <i>combobox</i> mata pelajaran	
2. memilih <i>combobox</i> jenjang	
3. memilih tombol 'Go'	
	4. Mengambil data nama pendaftar

	dari tabel tb_pendaftar dan nilai akhir dari tabel tb_nilai dari database sesuai dengan bidang keahlian yang dipilih pada filter
5.	menampilkan halaman dengan daftar pendaftar dan tombol 'detail'

### A.7 Use Case Skenario Mengelola Kuota Lowongan

Tabel A. 7 Use case skenario mengelola kuota lowongan

Nama Use Case	Mengelola kuota lowongan
Aktor	Admin
Pre Kondisi	Actor sudah melakukan login sistem
Post Kondisi	Aktor dapat menggunakan sistem untuk mengelola kuota lowongan
Skenario Normal – tambah baru	
Aksi Aktor	Reaksi Sistem
1. Pilih menu 'pembaruan'	
2. Pilih sub menu 'penawaran dan kuota'	
	3. Mengambil data rekrutmen berupa <ul style="list-style-type: none"> <li>• Id rekrut</li> <li>• Tanggal buka</li> <li>• Tanggal tutup</li> <li>• Kuota lowongan</li> <li>• Status lowongan</li> </ul> dari tabel tb_rekrut di database dan <ul style="list-style-type: none"> <li>• Mata pelajaran</li> <li>• Jenjang</li> </ul> Dari tabel tb_mapel didatabase.
	4. Menampilkan halaman dengan daftar penawaran posisi pengajar yaitu <ul style="list-style-type: none"> <li>• Kolom Id rekrut</li> <li>• Kolom Tanggal buka</li> <li>• Kolom Tanggal tutup</li> <li>• Kolom Kuota lowongan</li> <li>• Kolom Status lowongan</li> <li>• Kolom Mata pelajaran</li> <li>• Kolom Jenjang</li> </ul> tombol 'daftar baru', tombol 'ubah', tombol 'buka', dan tombol 'tutup'

---

5. Pilih tombol 'tambah baru' Jika pilih tombol 'buka' ke 4a, Jika pilih tombol 'tutup' ke 4b, Jika pilih tombol 'ubah' ke 4c.	
	6. Menampilkan form penawaran posisi pengajar <ul style="list-style-type: none"><li>• Kolom tanggal buka</li><li>• Kolom tanggal akhir</li><li>• <i>Combobox</i> Jenjang</li><li>• <i>Combobox</i> Mata Pelajaran</li><li>• Kolom kuota</li><li>• <i>Combobox</i> Status lowongan</li><li>• Tombol '<i>submit</i>'</li><li>• tombol '<i>reset</i>'</li></ul>
7. Isi kolom form penawaran pengajar	
8. Pilih ' <i>sumit button</i> '	
	9. Menyimpan data yang diisikan diform berupa <ul style="list-style-type: none"><li>• Id rekrut</li><li>• Tanggal buka</li><li>• Tanggal tutup</li><li>• Kuota lowongan</li><li>• Status lowongan</li><li>• Id Mata pelajaran</li><li>• Id Jenjang</li></ul> ke tabel <i>tb_rekrut</i> di <i>database</i>
	10. Mengambil data rekrutmen berupa <ul style="list-style-type: none"><li>• Id rekrut</li><li>• Tanggal buka</li><li>• Tanggal tutup</li><li>• Kuota lowongan</li><li>• Status lowongan</li></ul> dari tabel <i>tb_rekrut</i> di <i>database</i> dan <ul style="list-style-type: none"><li>• Mata pelajaran</li><li>• Jenjang</li></ul> dari tabel <i>tb_mapel</i> di <i>database</i>
	11. Menampilkan halaman dengan daftar penawaran posisi pengajar yaitu <ul style="list-style-type: none"><li>• Kolom Id rekrut</li><li>• Kolom Tanggal buka</li><li>• Kolom Tanggal tutup</li></ul>

---

- Kolom Kuota lowongan
  - Kolom Status lowongan
  - Kolom Mata pelajaran
  - Kolom Jenjang
- tombol ‘daftar baru’, tombol ‘ubah’,  
tombol ‘buka’ ,dan tombol ‘tutup’

**Skenario Alternatif (pilih tombol ‘buka’)**

Aksi Aktor	Reaksi Sistem
------------	---------------

4a. Pilih tombol ‘buka’

5a. Mengubah status rekrutmen menjadi “ada” pada tabel tb\_rekrut di *database*

6a. Mengambil data rekrutmen berupa

- Id rekrut
  - Tanggal buka
  - Tanggal tutup
  - Kuota lowongan
  - Status lowongan
- dari tabel tb\_rekrut di *database* dan
- Mata pelajaran
  - Jenjang

dari tabel tb\_mapel di *database*

7a. Menampilkan halaman dengan daftar penawaran posisi pengajar yaitu

- Kolom Id rekrut
- Kolom Tanggal buka
- Kolom Tanggal tutup
- Kolom Kuota lowongan
- Kolom Status lowongan
- Kolom Mata pelajaran
- Kolom Jenjang

tombol ‘daftar baru’, tombol ‘ubah’,  
tombol ‘buka’ ,dan tombol ‘tutup’

**Skenario Alternatif (pilih tombol ‘tutup’)**

Aksi Aktor	Reaksi Sistem
------------	---------------

4b. Pilih tombol ‘tutup’

5b. Mengubah status lowongan menjadi “tidak ada” pada tabel tb\_rekrut di *database*

6b. Mengambil data rekrutmen berupa

- Id rekrut
- Tanggal buka
- Tanggal tutup
- Kuota lowongan

- Status lowongan dari tabel *tb\_rekrut* di *database* dan
- Mata pelajaran
- Jenjang

dari tabel *tb\_mapel* di *database*

7b. Menampilkan halaman dengan daftar penawaran posisi pengajar yaitu

- Kolom Id rekrut
- Kolom Tanggal buka
- Kolom Tanggal tutup
- Kolom Kuota lowongan
- Kolom Status lowongan
- Kolom Mata pelajaran
- Kolom Jenjang

tombol 'daftar baru', tombol 'ubah', tombol 'buka', dan tombol 'tutup'

**Skenario Alternatif (pilih tombol 'ubah')**

**Aksi Aktor**

**Reaksi Sistem**

4c. Pilih tombol 'ubah'

5c. mengambil data lowongan sesuai id lowongan yang dipilih berupa

- id rekrut
  - tanggal buka
  - tanggal akhir
  - jumlah kuota
  - status lowongan
- dari tabel *tb\_rekrut*
- nama pelajaran
  - jenjang

dari tabel *tb\_mapel* di *database*

6c. Menampilkan data lowongan pada form ubah lowongan posisi pengajar

- Kolom tanggal buka
- Kolom tanggal akhir
- *Combobox* Jenjang
- *Combobox* Mata Pelajaran
- Kolom kuota
- *Combobox* Status lowongan
- Tombol 'submit'
- tombol 'reset'

7c. Isi kolom form lowongan pengajar

8c. Pilih 'submit button'

9c. mengubah data lowongan yang

	<p>sesuai id lowongan yang dimaksud dengan data yang diisikan diform berupa</p> <ul style="list-style-type: none"> <li>• id rekrut</li> <li>• tanggal buka</li> <li>• tanggal akhir</li> <li>• jumlah kuota</li> <li>• status lowongan</li> <li>• id pelajaran</li> <li>• id jenjang</li> </ul> <p>ke ke tabel <i>tb_rekrutdi database</i></p>
	<p>10c. Mengambil data rekrutmen berupa</p> <ul style="list-style-type: none"> <li>• Id rekrut</li> <li>• Tanggal buka</li> <li>• Tanggal tutup</li> <li>• Kuota lowongan</li> <li>• Status lowongan</li> </ul> <p>dari tabel <i>tb_rekrut</i> di <i>database</i> dan</p> <ul style="list-style-type: none"> <li>• Mata pelajaran</li> <li>• Jenjang</li> </ul> <p>dari tabel <i>tb_mapel</i> di <i>database</i></p>
	<p>11c. Menampilkan halaman dengan daftar penawaran posisi pengajar, tombol ‘daftar baru’, tombol ‘ubah’, tombol ‘buka’, dan tombol ‘tutup’</p>
<b>Skenario Alternatif (kolom kosong/tidak diisi)</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
	<p>5a. Menampilkan pesan “<i>please fill out this field</i>” dengan menunjuk pada kolom yang belum diisi</p>
<b>Skenario Alternatif (pilih reset button)</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
<p>1. Pilih <i>reset</i></p>	<p>2. Menampilkan halaman form dengan kondisi kolom seperti awal, tombol <i>submit button</i> dan <i>reset button</i></p>
<b>Skenario Alternatif (melakukan pencarian)</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
<p>1. memilih <i>combobox</i> mata pelajaran</p>	
<p>2. memilih <i>combobox</i> jenjang</p>	
<p>3. memilih tombol ‘Go’</p>	<p>4. Mengambil data rekrutmen dari <i>database</i></p>
	<p>5. Menampilkan halaman dengan</p>

daftar penawaran posisi pengajar,  
tombol 'daftar baru', tombol 'ubah',  
tombol 'buka', dan tombol 'tutup'

#### A.8 Use Case Skenario Mengelola Data Status Pendaftaran

Tabel A. 8 Use case skenario view mengelola data status pendaftaran

Nama Use Case	Mengelola data status pendaftaran
Aktor	Admin
Pre Kondisi	Aktor sudah login pada sistem
Post Kondisi	Aktor dapat mengelola data status pendaftaran
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih menu 'pembaruan'	
2. Pilih sub menu 'reset daftar pendaftar'	
	3. Mengambil data pendaftar berupa, <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> </ul> Dari tabel tb_pendaftar
	<ul style="list-style-type: none"> <li>• Nilai pendaftar</li> <li>• Bidang keahlian</li> <li>• Status terima</li> </ul> dari tabel tb_rekrut didatabase
	4. Menampilkan halaman dengan daftar pendaftar, tombol 'hapus', dan tombol 'lihat semua'
5. Pilih tombol 'hapus' Jika pilih tombol 'lihat semua' lanjut ke 5a.	
	6. Mengubah status pendaftaran dafri pendaftar yang idnya terpilih
	7. Mengambil data pendaftar dari database
	8. Menampilkan halaman dengan daftar pendaftar, tombol 'hapus', dan tombol 'lihat semua'
Skenario Alternatif (pilih tombol'lihat semua')	
Aksi Aktor	Reaksi Sistem
5a. pilih tombol 'lihat semua'	
	6a. Mengambil semua nama data pendaftar dari tabel tb_pendaftar dari database

7a. Menampilkan halaman dengan daftar pendaftar	
Skenario Alternatif (melakukan pencarian)	
Aksi Aktor	Reaksi Sistem
1. memilih <i>combobox</i> mata pelajaran	
2. memilih <i>combobox</i> jenjang	
3. memilih tombol 'Go'	
	9. Mengambil data pendaftar sesuai dengan mata pelajaran yang dipilih, yang berupa <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> </ul> Dari tabel <i>tb_pendaftar</i> <ul style="list-style-type: none"> <li>• Nilai pendaftar</li> </ul> Dari tabel <i>tb_nilai</i> <ul style="list-style-type: none"> <li>• Bidang keahlian</li> <li>• Status terima</li> </ul> dari tabel <i>tb_rekrut</i> di <i>database</i>
	10. Menampilkan halaman dengan daftar pendaftar, tombol 'hapus', dan tombol 'lihat semua'

A.9 Use Case Skenario Menampilkan Data Pendaftar

Tabel A. 9 Use case skenario menampilkan data pendaftar

Nama Usecase	Menampilkan data pendaftar
Aktor	LITBANG
Pre Kondisi	Aaktor sudah melakukan <i>login</i> sistem
Post Kondisi	Aktor dapat melihat data pendaftar
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih menu 'data pendaftar'	
	2. Mengambil data nama pendaftar dari tabel <i>tb_pendaftar</i> dan nilai akhir dari tabel <i>tb_nilai</i> dari <i>database</i>
	3. Menampilkan halaman daftar pendaftar dan tombol 'detail'
4. Pilih tombol 'detail'	
	5. Mengambil data pendafatar <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama lengkap</li> <li>• Alamat</li> </ul>

	<ul style="list-style-type: none"> <li>• Tempat lahir</li> <li>• Tanggal lahir</li> <li>• Jenis identitas</li> <li>• Nomor identitas</li> <li>• Nomor telepon</li> <li>• Email</li> <li>• Bidang keahlian</li> <li>• Jenjang</li> <li>• Status terima</li> </ul> <p>dari tb_pendaftar database sesuai dengan id yang dipilih dan</p> <ul style="list-style-type: none"> <li>• Ipk</li> <li>• Lulusan</li> </ul> <p>Dari tb_nilai di <i>database</i> sesuai id yang terpilih</p>
	6. Menampilkan data profil pendaftar sesuai id yang terpilih ke halaman detail profil
7. Pilih tombol 'nilai pendaftar'	
	8. Mengambil data pendaftar <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama lengkap pendaftar</li> </ul> <p>Dari tb_pendaftar <i>didatabase</i></p> <ul style="list-style-type: none"> <li>• Nilai IPK</li> <li>• Lulusan</li> <li>• Nilai wawancara</li> <li>• Nilai test tulis</li> <li>• Nilai tes microteaching</li> <li>• Nilai Akhir</li> </ul> <p>Dari tabel tb_nilai di <i>database</i></p>
	9. Menampilkan data pendaftar ke halaman nilai pendaftar
<b>Skenario Alternatif (melakukan pencarian)</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. memilih <i>combobox</i> mata pelajaran	
2. memilih <i>combobox</i> jenjang	
3. memilih tombol 'Go'	
	4. Mengambil data nama pendaftar dari tabel tb_pendaftar dan nilai akhir dari tabel tb_nilai dari <i>database</i>
	5. Menampilkan halaman daftar pendaftar dan tombol 'detail'

A.10 Use Case Skenario Update Nilai Wawancara dan *Microteaching*Tabel A. 10 Use case skenario update nilai wawancara dan *microteaching*

Nama Use Case	Update nilai wawancara dan <i>microteaching</i>
Aktor	LITBANG
Pre Kondisi	Aktor sudah melakukan <i>login</i> sistem
Post Kondisi	Aktor dapat mengelola nilai wawancara dan <i>microteaching</i>
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih menu 'input nilai'	2. Mengambil daftar nilai pendaftar berupa <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> <li>• Nilai wawancara</li> <li>• Nilai tes <i>microteaching</i></li> </ul> Daritabel <i>tb_nilai</i> di <i>database</i>
4. Pilih tombol 'ubah'	3. Menampilkan halaman input nilai dengan daftar nilai pendaftar dan tombol 'ubah'
6. Mengisi kolom nilai	5. Menampilkan halaman dengan <ul style="list-style-type: none"> <li>• kolom nilai wawancara</li> <li>• kolom nilai <i>microteaching</i></li> </ul> dan tombol 'submit' dan tombol 'back'
7. Pilih tombol 'submit' Jika pilih tombol 'back' lanjut 7.a	8. Menyimpan data nilai pendaftar berupa <ul style="list-style-type: none"> <li>• Nilai wawancara</li> <li>• Nilai tes <i>microteaching</i></li> </ul> ke <i>tb_nilai</i> di <i>database</i>
	9. Mengambil daftar nilai pendaftar berupa <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> <li>• Nilai wawancara</li> <li>• Nilai tes <i>microteaching</i></li> </ul> Daritabel <i>tb_nilai</i> di <i>database</i>
	10. Menampilkan halaman input nilai dengan daftar nilai pendaftar dan tombol 'ubah'

Skenario Alternatif (memilih tombol back)	
7a. pilih tombol 'back'	8a. Mengambil daftar nilai pendaftar berupa <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> <li>• Nilai wawancara</li> <li>• Nilai tes <i>microteaching</i></li> </ul> Daritabel tb_nilai di <i>database</i>
	9a. Menampilkan halaman input nilai dengan daftar nilai pendaftar dan tombol 'ubah'

Skenario Alternatif (melakukan pencarian)	
Aksi Aktor	Reaksi Sistem
1. memilih <i>combobox</i> mata pelajaran	
2. memilih <i>combobox</i> jenjang	
3. memilih tombol 'Go'	
	4. Mengambil daftar nilai pendaftar berupa <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> <li>• Nilai wawancara</li> <li>• Nilai tes <i>microteaching</i></li> </ul> Daritabel tb_nilai di <i>database</i>
	5. Menampilkan halaman input nilai dengan daftar nilai pendaftar dan tombol 'ubah'

A.11 Use Case Skenario Mengelola Nilai ideal Kriteria

Tabel A. 12 Use case skenario mengelola nilai ideal kriteria

Nama Use Case	Mengelola nilai ideal kriteria
Aktor	LITBANG
Pre Kondisi	Actor sudah melakukan <i>login</i> sistem
Post Kondisi	Aktor dapat mengelola nilai ideal kriteria
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih menu 'Nilai Ideal'	2. Mengambil data nilai ideal <ul style="list-style-type: none"> <li>• Nilai ideal ipk</li> <li>• Nilai ideal lulusan</li> <li>• Nilai ideal tes wawancara</li> <li>• Nilai ideal tes tulis</li> </ul>

	<ul style="list-style-type: none"> <li>• Nilai ideal tes <i>microteaching</i></li> </ul> <p>Ke tabel <i>tb_nilaiideal</i></p> <ul style="list-style-type: none"> <li>• Skala nilai ideal ipk</li> <li>• Skala nilai ideal lulusan</li> <li>• Skala nilai ideal tes wawancara</li> <li>• Skala nilai ideal tes tulis</li> <li>• Skala nilai ideal tes <i>microteaching</i></li> </ul> <p>Dari tabel <i>tb_nilaiidealskala</i> di database</p>
	3. Menampilkan halaman data nilai ideal, data nilai ideal skala dan tombol 'edit nilai ideal'
4. Pilih tombol 'edit nilai ideal'	
	5. Menampilkan form edit nilai ideal <ul style="list-style-type: none"> <li>• Kolom nilai ideal ipk</li> <li>• Kolom nilai ideal lulusan</li> <li>• Kolom nilai ideal tes wawancara</li> <li>• Kolom nilai ideal tes tulis</li> <li>• Kolom nilai ideal tes <i>microteaching</i></li> </ul>
6. Mengisi kolom kolom nilai ideal	
7. Memilih 'simpan' Jika memilih 'back' lanjut ke 7a.	
	8. mengubah nilai ideal, menghitung hasil normalisasi, dan nilai ideal skala pada <i>database</i>
	9. Mengambil data nilai ideal <ul style="list-style-type: none"> <li>• Nilai ideal ipk</li> <li>• Nilai ideal lulusan</li> <li>• Nilai ideal tes wawancara</li> <li>• Nilai ideal tes tulis</li> <li>• Nilai ideal tes <i>microteaching</i></li> </ul> <p>Ke tabel <i>tb_nilaiideal</i></p> <ul style="list-style-type: none"> <li>• Skala nilai ideal ipk</li> <li>• Skala nilai ideal lulusan</li> <li>• Skala nilai ideal tes wawancara</li> <li>• Skala nilai ideal tes tulis</li> <li>• Skala nilai ideal tes <i>microteaching</i></li> </ul> <p>Dari tabel <i>tb_nilaiidealskala</i> di database</p>

10. Menampilkan halaman data nilai ideal, data nilai ideal skala dan tombol 'edit nilai ideal'

Skenario Alternatif (back)	
Aktor	Sistem

7a. Memilih 'back'

8a. Mengambil data nilai ideal

- Nilai ideal ipk
- Nilai ideal lulusan
- Nilai ideal tes wawancara
- Nilai ideal tes tulis
- Nilai ideal tes *microteaching*

Ke tabel *tb\_nilaiideal*

- Skala nilai ideal ipk
- Skala nilai ideal lulusan
- Skala nilai ideal tes wawancara
- Skala nilai ideal tes tulis
- Skala nilai ideal tes *microteaching*

Dari tabel *tb\_nilaiidealskala* di database

9a. Menampilkan halaman data nilai ideal, data nilai ideal skala dan tombol 'edit nilai ideal'

A.12 Use Case Skenario Menyetujui Hasil Penerimaan

Tabel A. 13 Use case skenario menyetujui hasil penerimaan

Nama Use Case	Menyetujui Hasil Penerimaan
Aktor	LITBANG
Pre Kondisi	Aktor sudah melakukan <i>login</i> sistem
Post Kondisi	Aktor telah dapat menyetujui penerimaan calon pengajar

Skenario Normal	
Aksi Aktor	Reaksi Sistem

1. Pilih 'penerimaan'

2. Mengambil data pendaftar yang memenuhi nilai kriteria berupa

- Id pendaftar
- Nama pendaftar

Dari tabel *tb\_pendaftar* didatabase

- Nilai akhir pendaftar

Dari tabel *tb\_nilai* didatabase

	3. Menampilkan data pendaftar yang memenuhi nilai kriteria kehalaman daftar calon pengajar
4. Memilih tombol 'terima' Jika memilih tombol 'tolak' lanjut ke 4a.	
	5. Mengubah status terima pendaftar menjadi "pendaftar diterima" pada tabel tb_pendaftar di database
	6. Mengambil data pendaftar yang memenuhi nilai kriteria berupa <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> </ul> Dari tabel tb_pendaftar didatabase <ul style="list-style-type: none"> <li>• Nilai akhir pendaftar</li> </ul> Dari tabel tb_nilai didatabase
	7. Menampilkan data pendaftar yang memenuhi nilai kriteria kehalaman daftar calon pengajar

**Skenari Alternatif (memilih tombol 'calon')**

Aktor	Sistem
4a. Memilih tombol 'tolak'	
	5a. Mengubah status terima pendaftar menjadi "pendaftar tereliminasi" pada tabel tb_pendaftar di database
	6a. Mengambil data pendaftar yang memenuhi nilai kriteria berupa <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> </ul> Dari tabel tb_pendaftar didatabase <ul style="list-style-type: none"> <li>• Nilai akhir pendaftar</li> </ul> Dari tabel tb_nilai didatabase
	7a. Menampilkan data pendaftar yang memenuhi nilai kriteria kehalaman daftar calon pengajar

**Skenario Alternatif (melakukan pencarian)**

Aksi Aktor	Reaksi Sistem
1. memilih <i>combobox</i> mata pelajaran	
2. memilih <i>combobox</i> jenjang	
3. memilih tombol 'Go'	
	8. Mengambil data pendaftar yang memenuhi nilai kriteria berupa <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama pendaftar</li> </ul>

	Dari tabel tb_pendaftar didatabase
	<ul style="list-style-type: none"> <li>• Nilai akhir pendaftar</li> </ul>
	Dari tabel tb_nilai didatabase
	9. Menampilkan data pendaftar yang memenuhi nilai kriteria kehalaman daftar calon pengajar

A.13 Use Case Skenario Logout

Tabel A. 14 Use case skenario skenario Logout

Nama Use Case	Logout
Aktor	admin, LITB
Pre Kondisi	Aktor sudah melakukan login sistem
Post Kondisi	Aktor telah keluar dari sistem
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih logout	2. Destroy session
	3. Menampilkan halaman awal sistem

A.14 Use Case Skenario Melakukan Pendaftaran

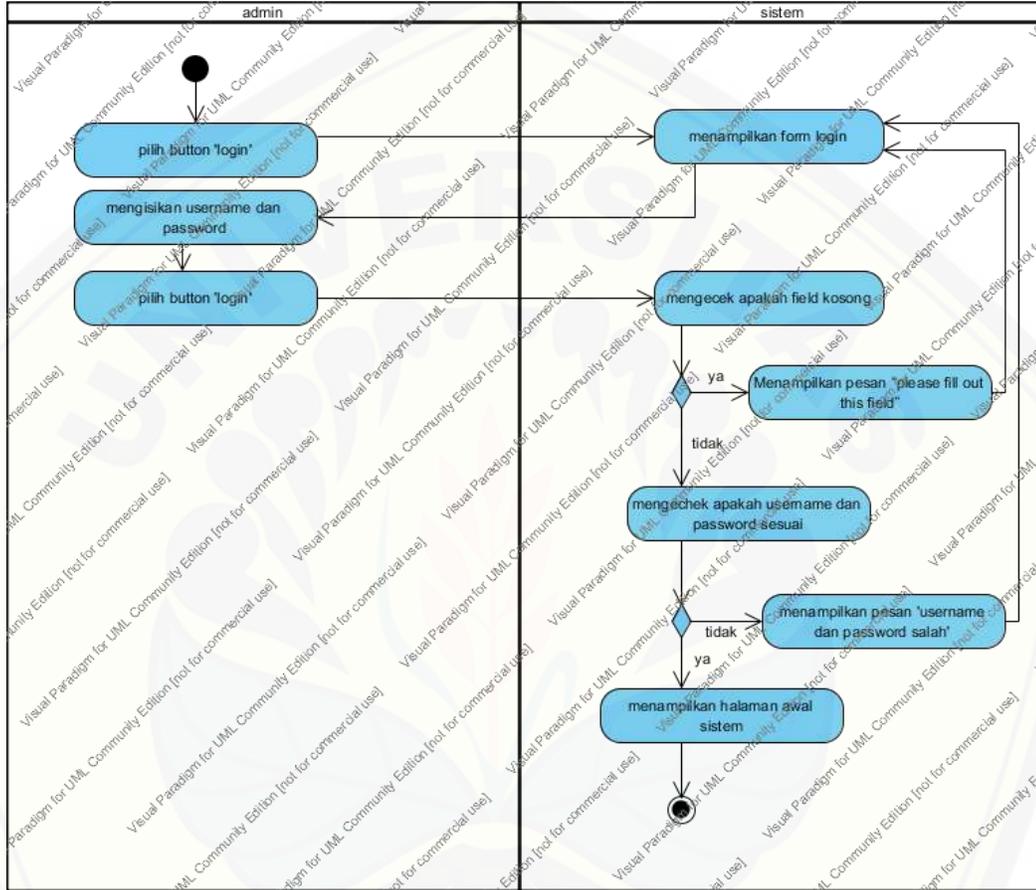
Tabel A. 15 Use case skenario melakukan pendaftaran

Nama Use Case	Melakukan pendaftaran
Aktor	Pendaftar
Pre Kondisi	Aktor sudah membuka sistem
Post Kondisi	Aktor dapat melakukan pendaftaran pada system
Skenario Normal	
Aksi Aktor	Reaksi Sistem
1. Pilih tombol 'daftar' pada halaman pertama	2. Menampilkan formulir pendaftaran dengan
	10. Mengambil data pendafatar
	<ul style="list-style-type: none"> <li>• Kolom Id pendafatar</li> <li>• Kolom Nama lengkap</li> <li>• Kolom Alamat</li> <li>• Kolom Tempat lahir</li> <li>• Kolom Tanggal lahir</li> <li>• Kolom Jenis identitas</li> <li>• Kolom Nomor identitas</li> <li>• Kolom Nomor telepon</li> </ul>

	<ul style="list-style-type: none"> <li>• Kolom Email</li> <li>• <i>Combobox</i> Bidang keahlian</li> <li>• <i>Combobox</i> Jenjang</li> <li>• Status terima</li> <li>• Kolom Ipk</li> <li>• <i>Combobox</i> Lulusan</li> </ul>
3. Mengisi kolom pendaftaran	
4. Memilih tombol 'submit' Jika memilih tombol 'reset' ke 4a.	
	<p>5. Menyimpan data pendaftar berupa</p> <ul style="list-style-type: none"> <li>• Id pendaftar</li> <li>• Nama lengkap</li> <li>• Alamat</li> <li>• Tempat lahir</li> <li>• Tanggal lahir</li> <li>• Jenis identitas</li> <li>• Nomor identitas</li> <li>• Nomor telepon</li> <li>• Email</li> <li>• Bidang keahlian</li> <li>• Jenjang</li> <li>• Status terima</li> </ul> <p>ke <i>tb_pendaftar</i> dan</p> <ul style="list-style-type: none"> <li>• Ipk</li> <li>• Lulusan</li> </ul> <p>dari <i>tb_nilai</i> di <i>database</i> sesuai dengan id yang dipilih</p>
	6. menampilkan ke halaman awal
<b>Skenario Alternatif (memilih tombol 'reset')</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
4a. Memilih tombol 'reset'	5a. Menampilkan formulir pendaftaran dimana kolom kembali kosong, tombol 'submit' dan tombol 'reset'
<b>Skenario Alternatif (terdapat kolom yang tidak diisi/kosong)</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
	5b. Menampilkan pesan " <i>please fill out this field</i> " dengan menunjuk pada kolom yang belum diisi

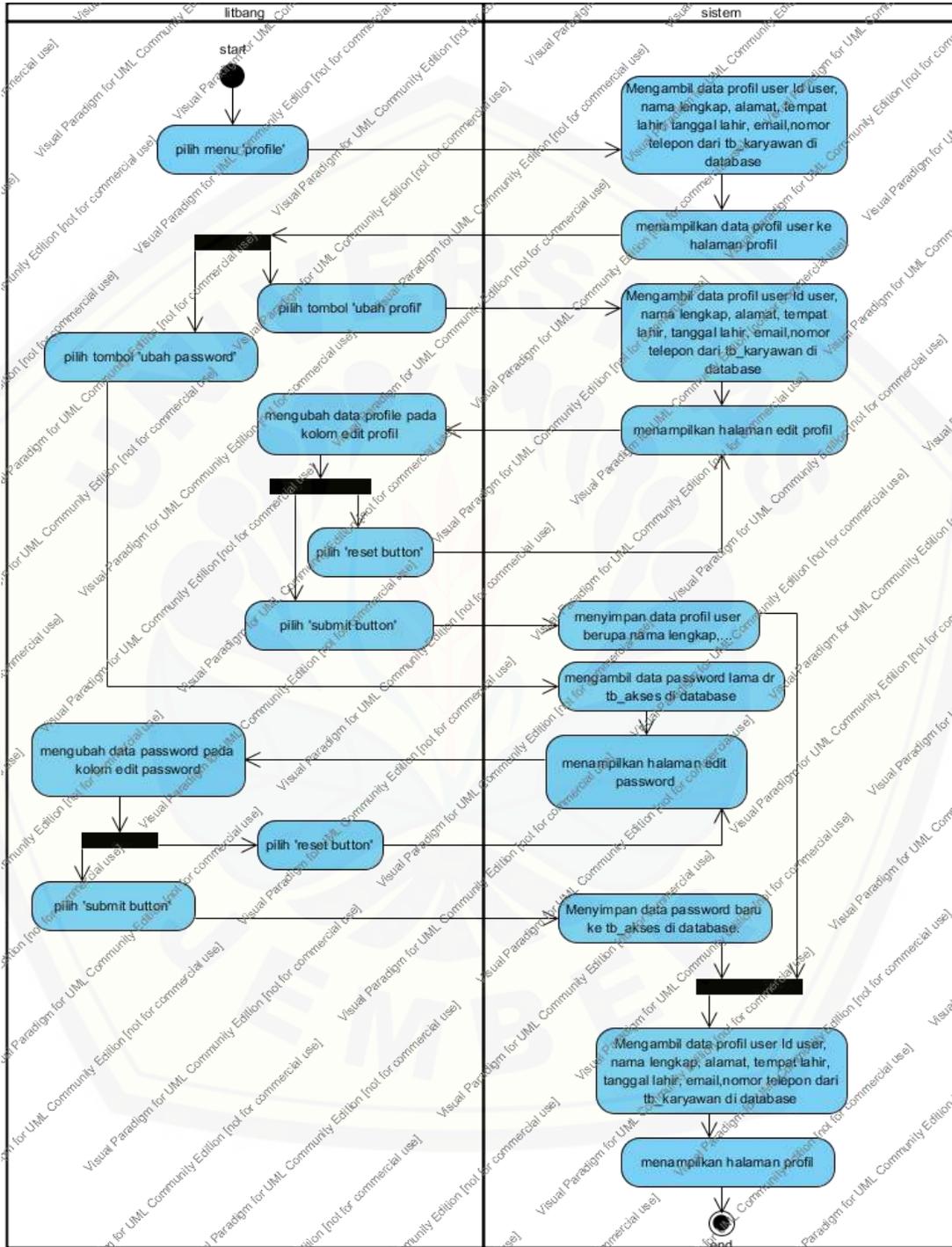
**LAMPIRAN B. ACTIVITY DIAGRAM**

**B.1 Activity Diagram Login**



Gambar B. 1 Activity Diagram Login

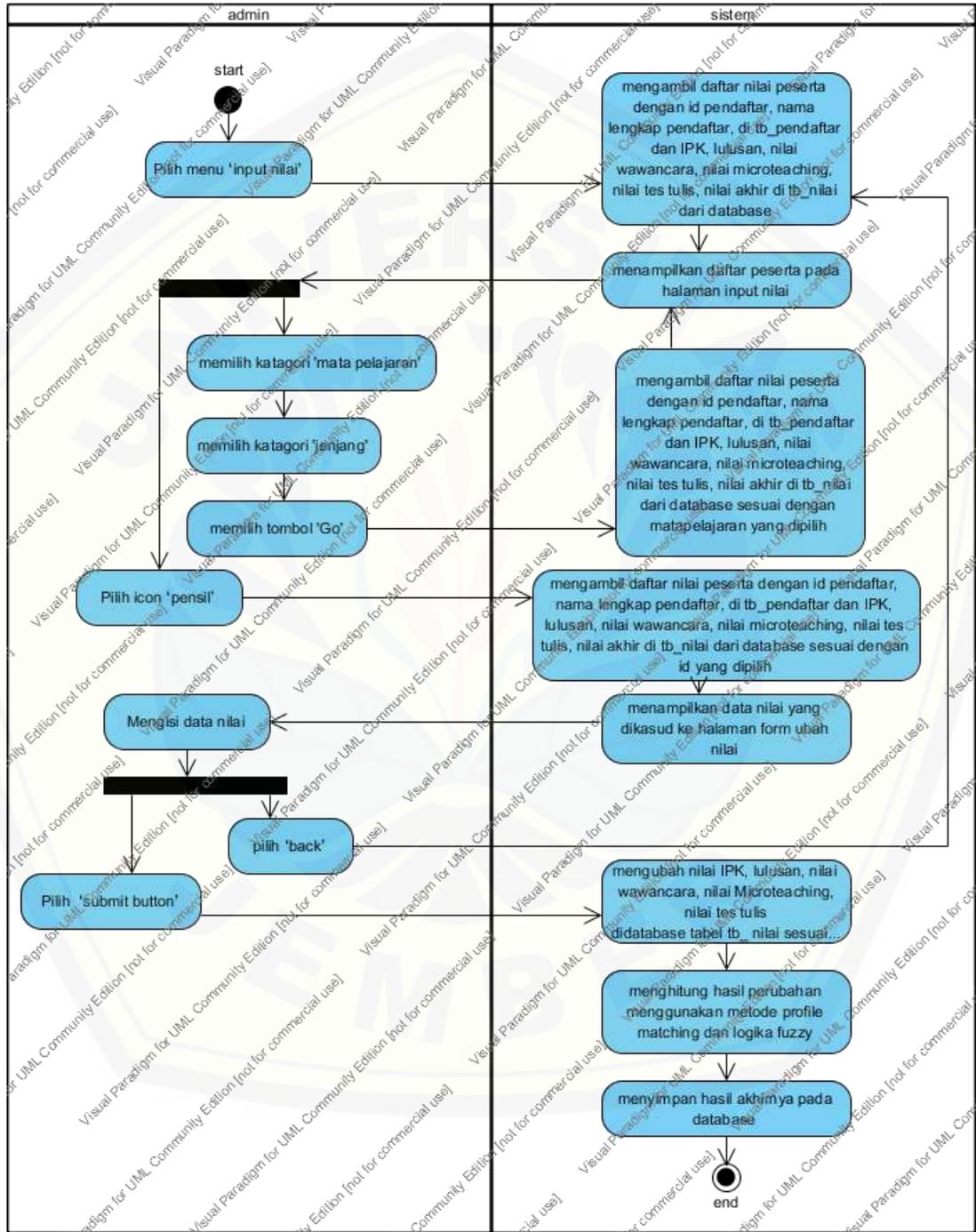
B.2 Activity Diagram Mengelola Data Profil User



Gambar B. 2 Activity Diagram Mengelola Data Profil User

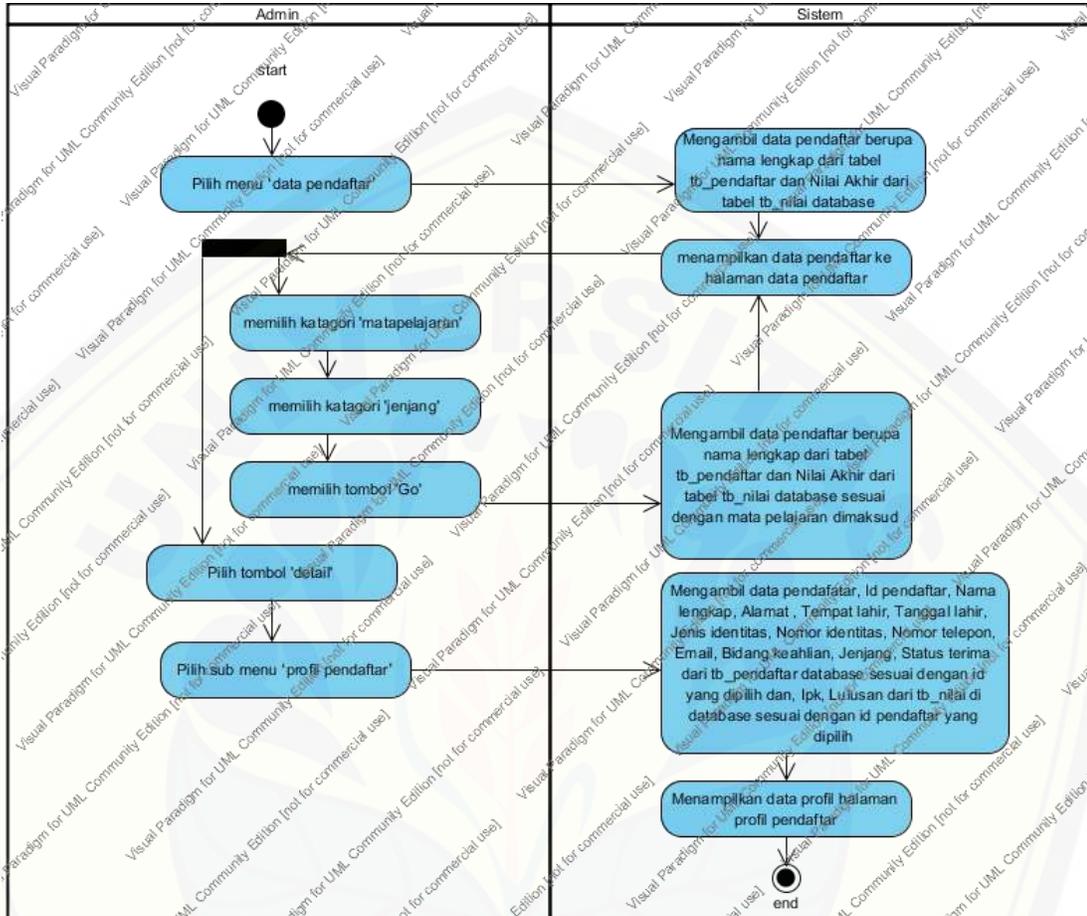


B.4 Activity Diagram Mengelola Nilai Pendaftar



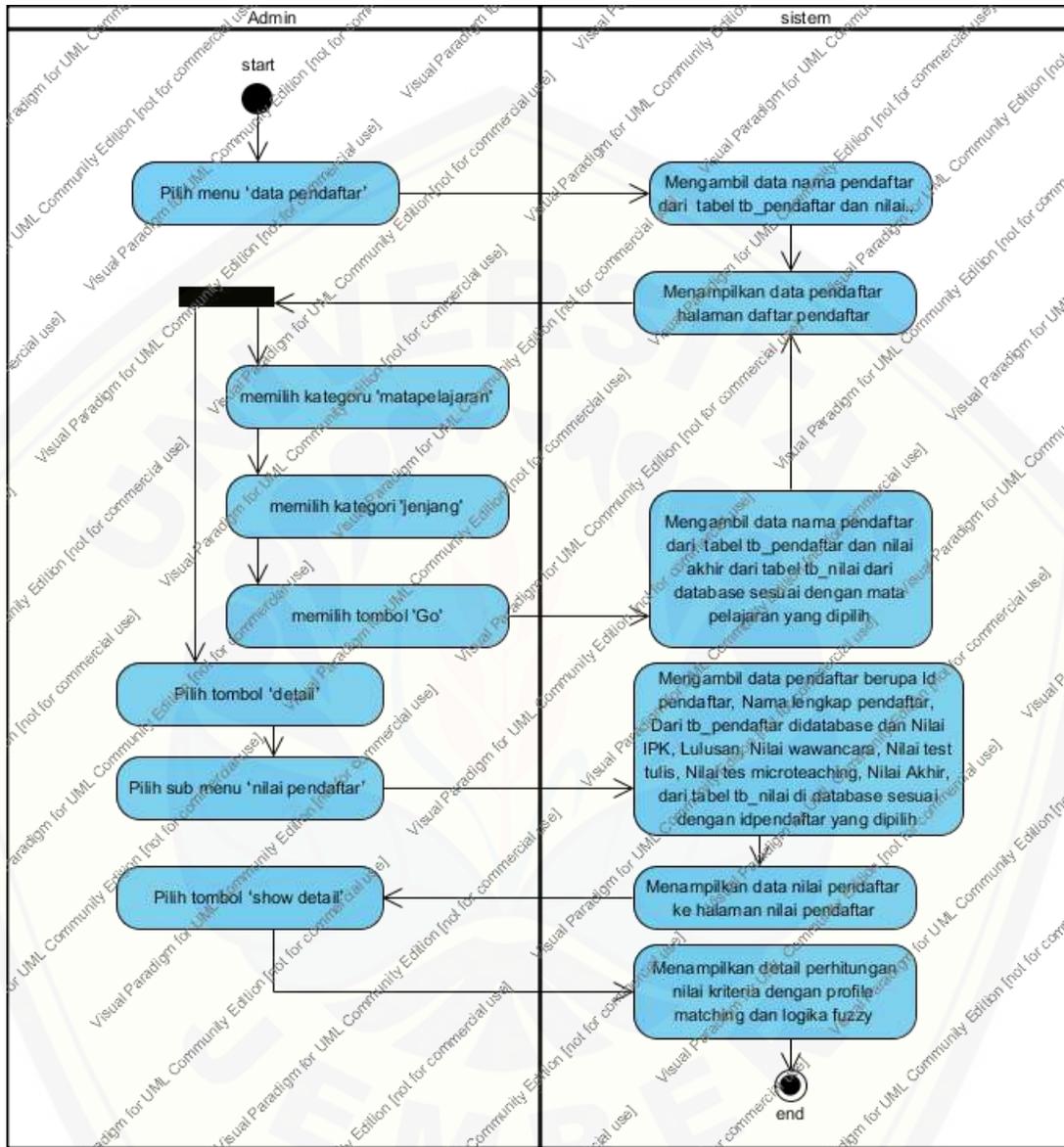
Gambar B. 4 Activity Diagram Mengelola Nilai Pendaftar

B.5 Activity Diagram Menampilkan data Profile Pendaftar



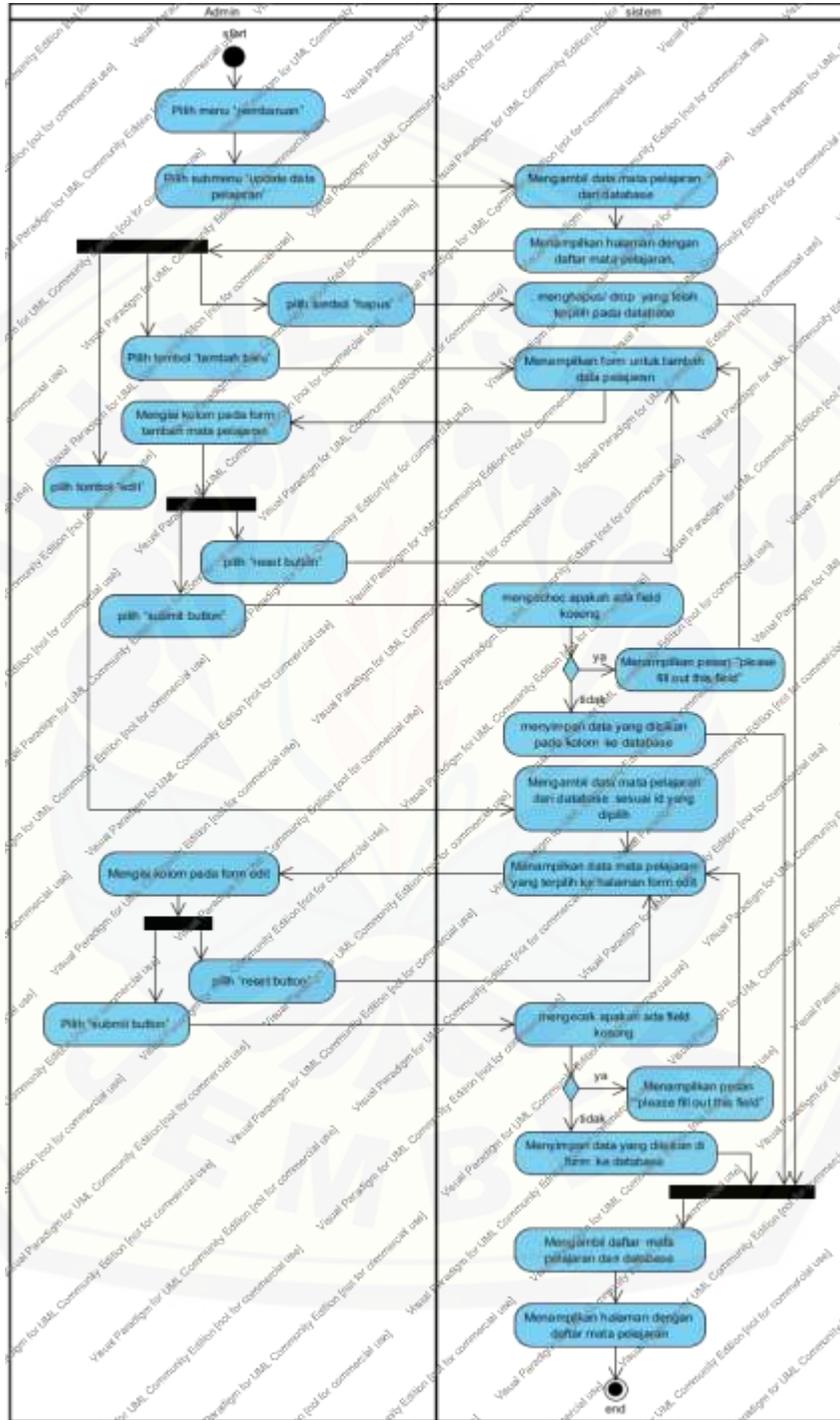
Gambar B. 5 Activity Diagram Menampilkan data Profile Pendaftar

B.6 Activity Diagram Menampilkan Data Nilai Pendaftar



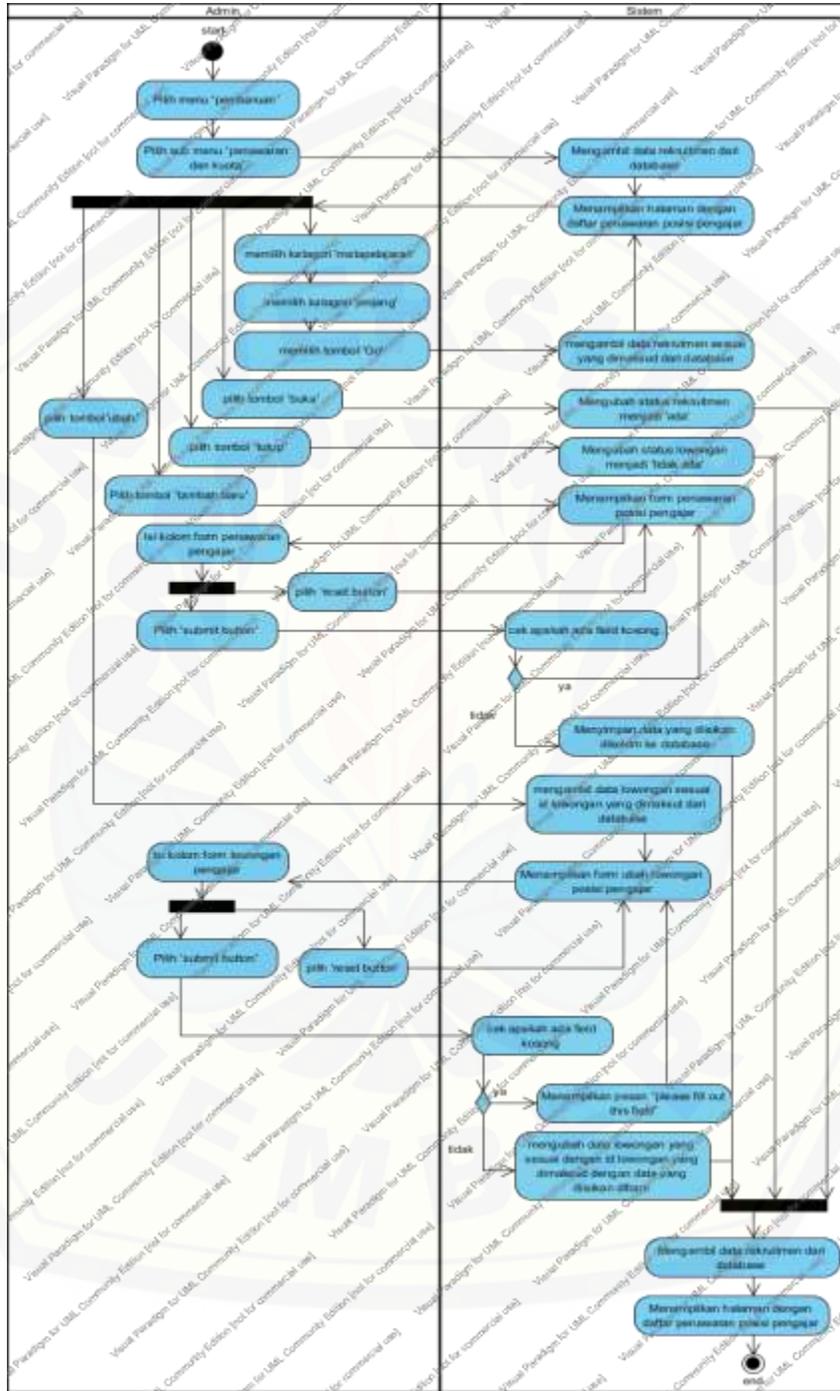
Gambar B. 6 Activity Diagram Menampilkan Data Nilai Pendaftar

B.7 Activity Diagram Mengelola Data Pelajaran



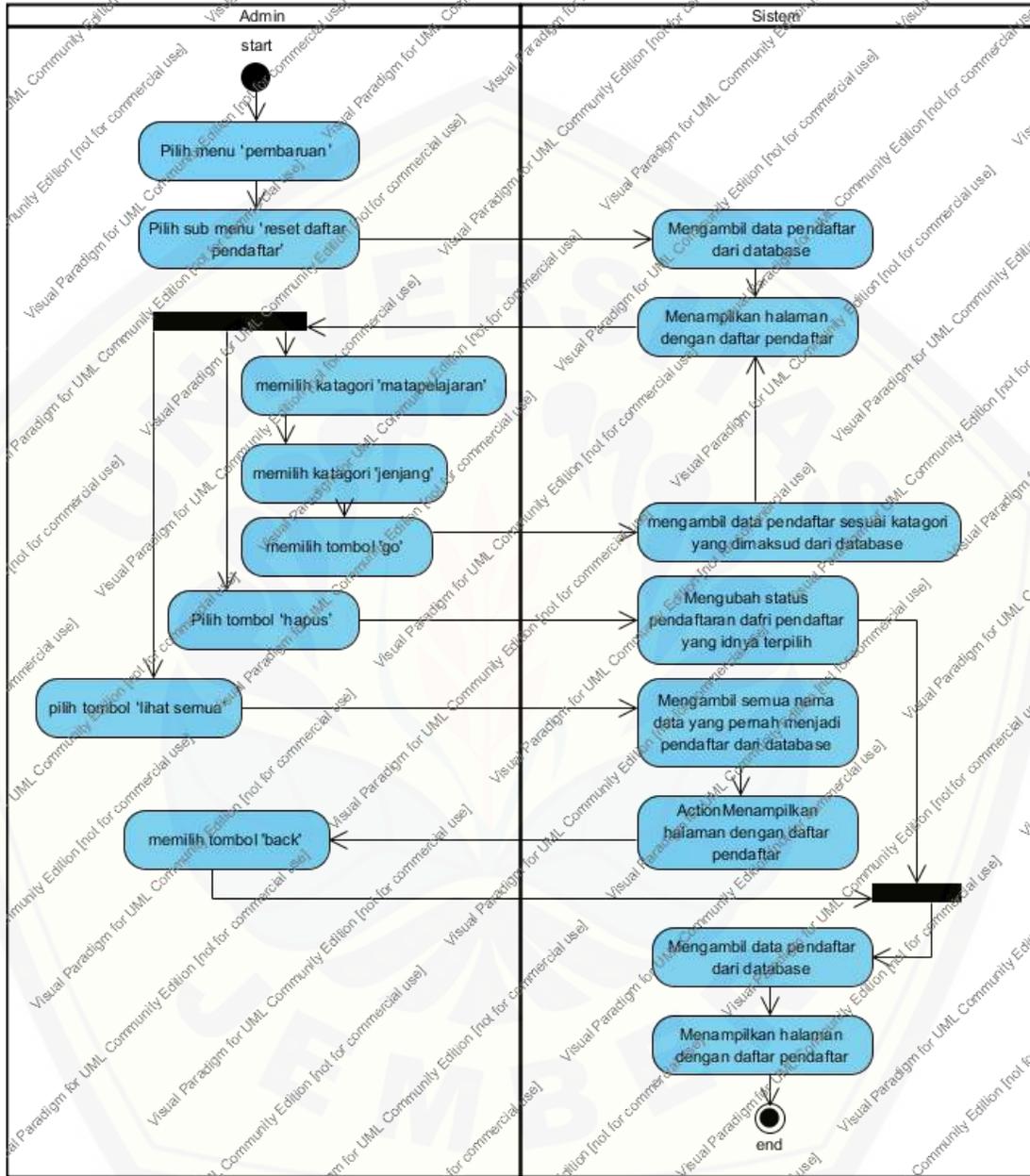
Gambar B. 7 Activity Diagram Mengelola Data Pelajaran

B.8 Activity Diagram Skenario Mengelola Kuota Lowongan



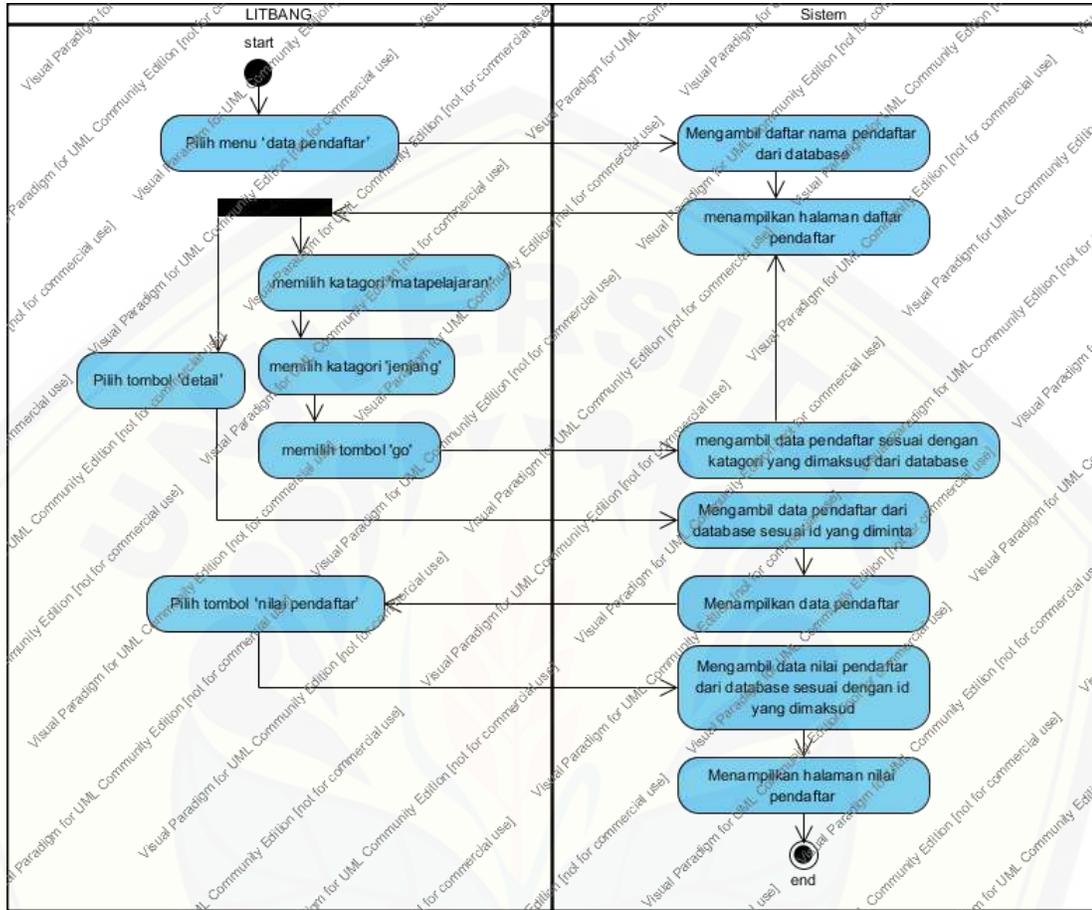
Gambar B. 8 Activity Diagram Skenario Mengelola Kuota Lowongan

B.9 Activity Diagram Mengelola Data Status Pendaftaran



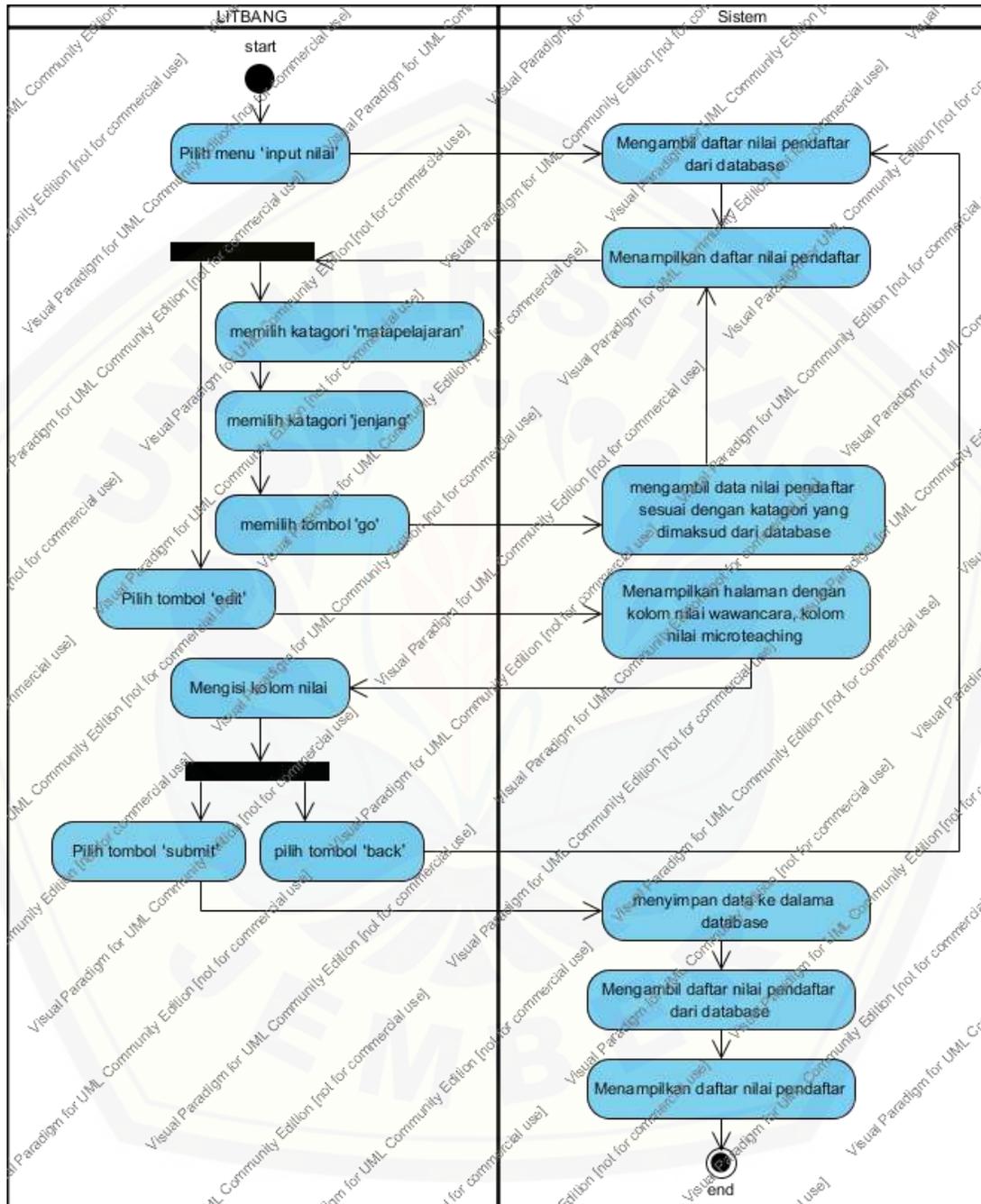
Gambar B. 9 Activity Diagram Mengelola Data Status Pendaftaran

B.10 Activity Diagram Menampilkan Data Pendaftar



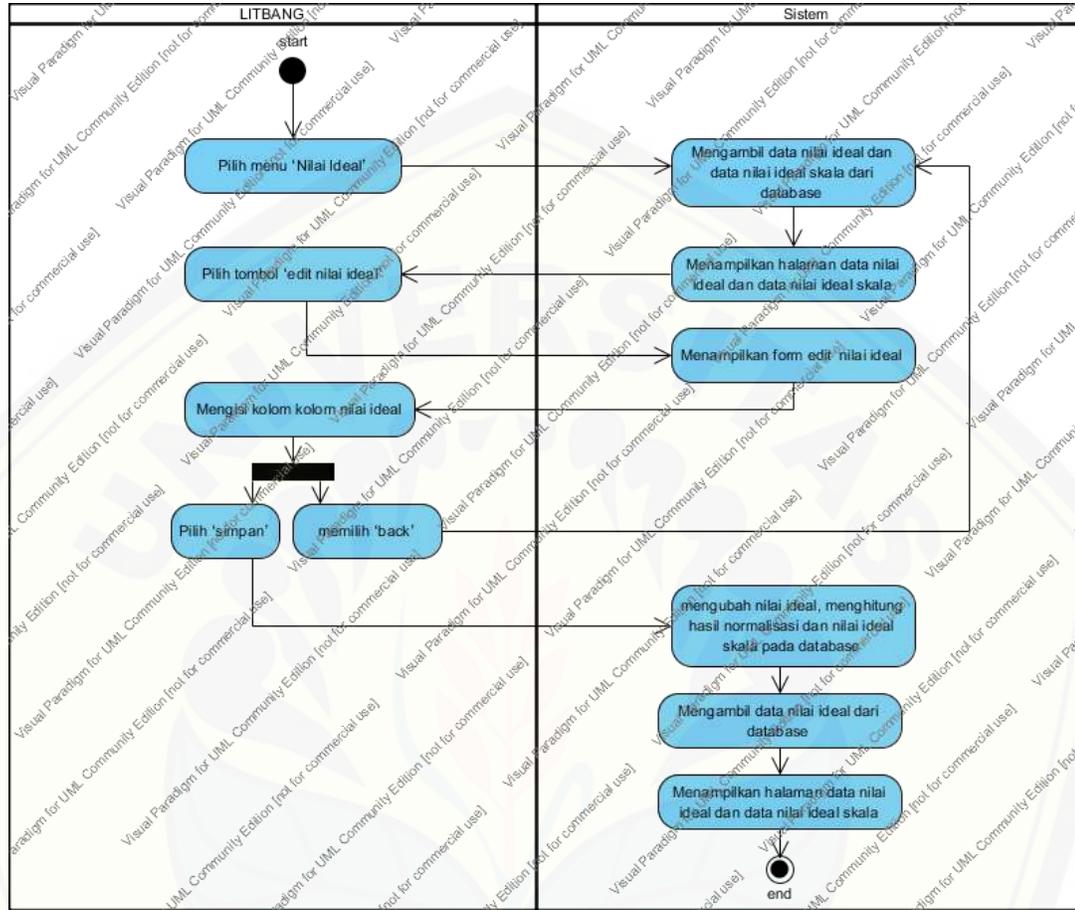
Gambar B. 10 Activity Diagram Menampilkan Data Pendaftar

B.11 Activity Diagram Update Nilai Wawancara dan Microteaching



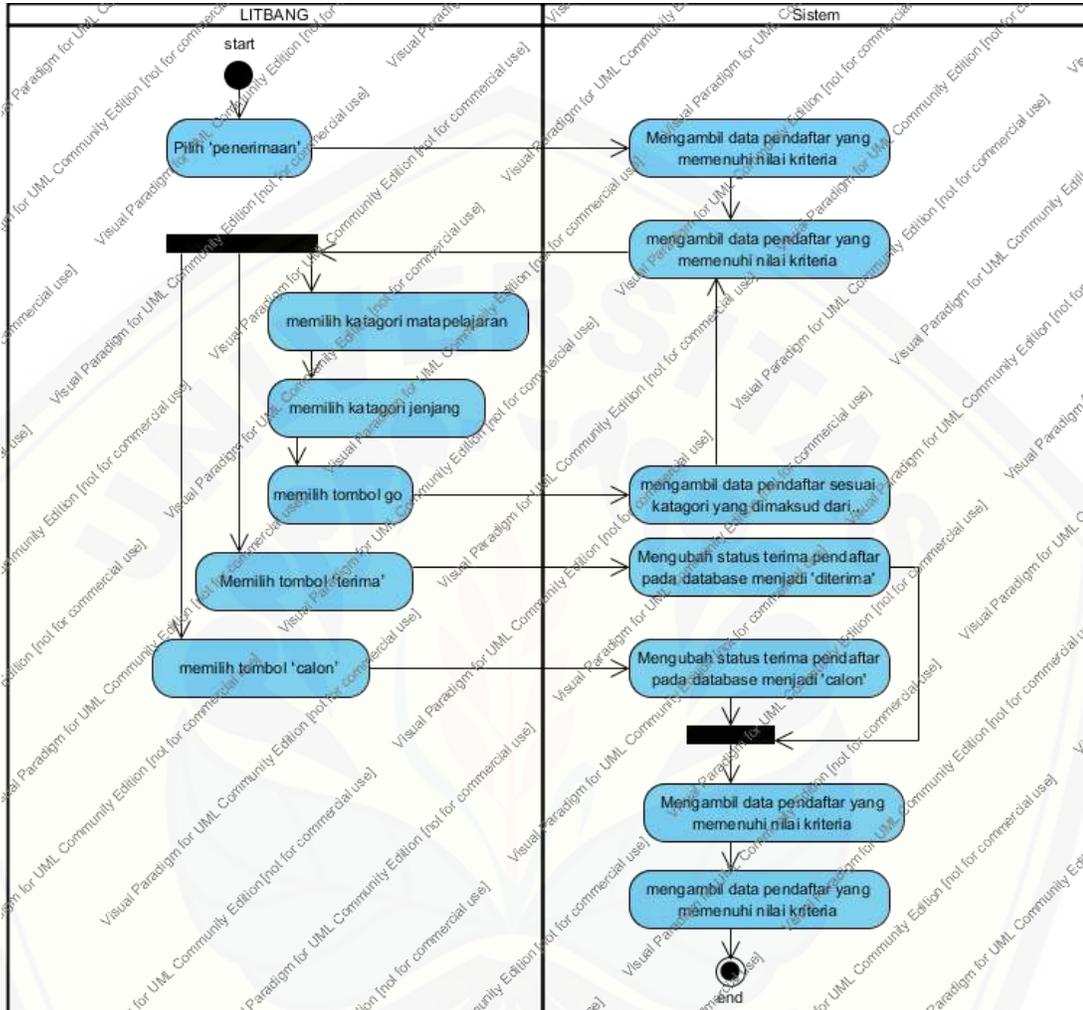
Gambar B. 11 Activity Diagram Update Nilai Wawancara dan Microteaching

B.12 Activity Diagram Mengelola Nilai ideal Kriteria



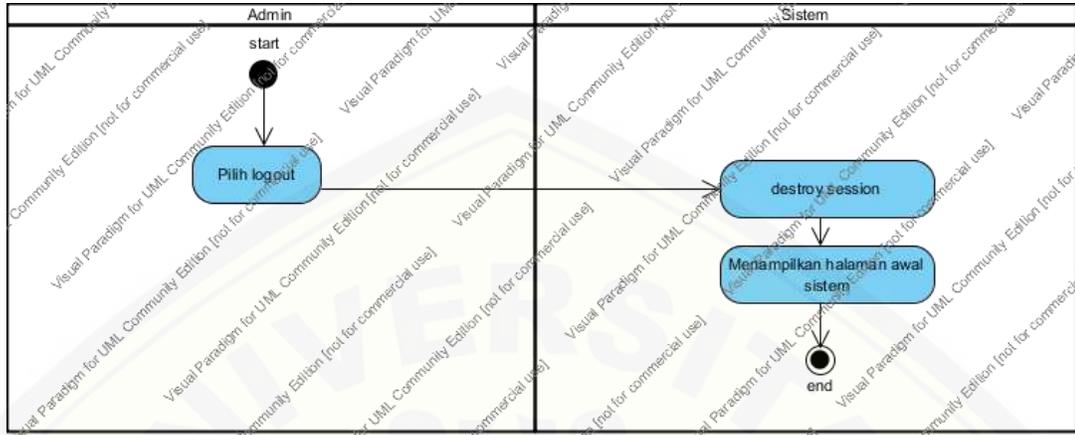
Gambar B. 12 Activity Diagram Mengelola Nilai ideal Kriteria

B.13 Activity Diagram Menyetujui Hasil Penerimaan



Gambar B. 13 Activity Diagram Menyetujui Hasil Penerimaan

B.14 Activity Diagram Logout

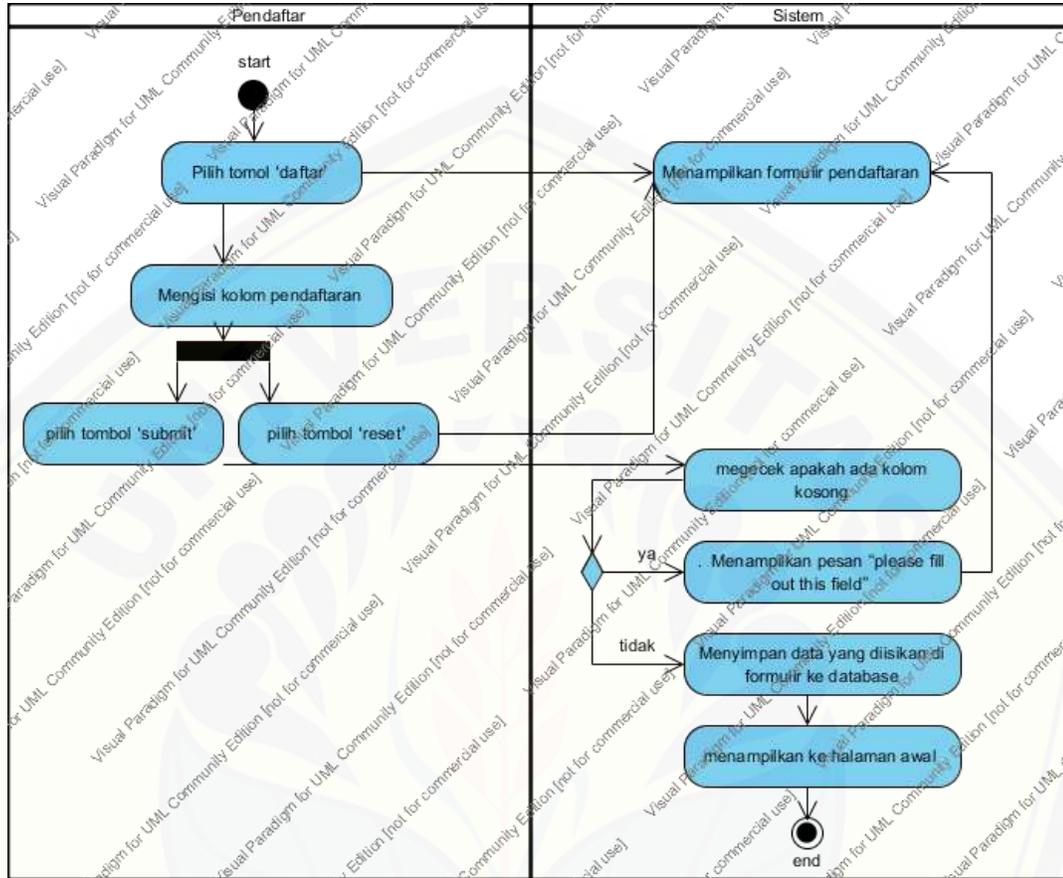


Gambar B. 14 Activity Diagram Logout



Gambar B. 15 Activity Diagram Logout

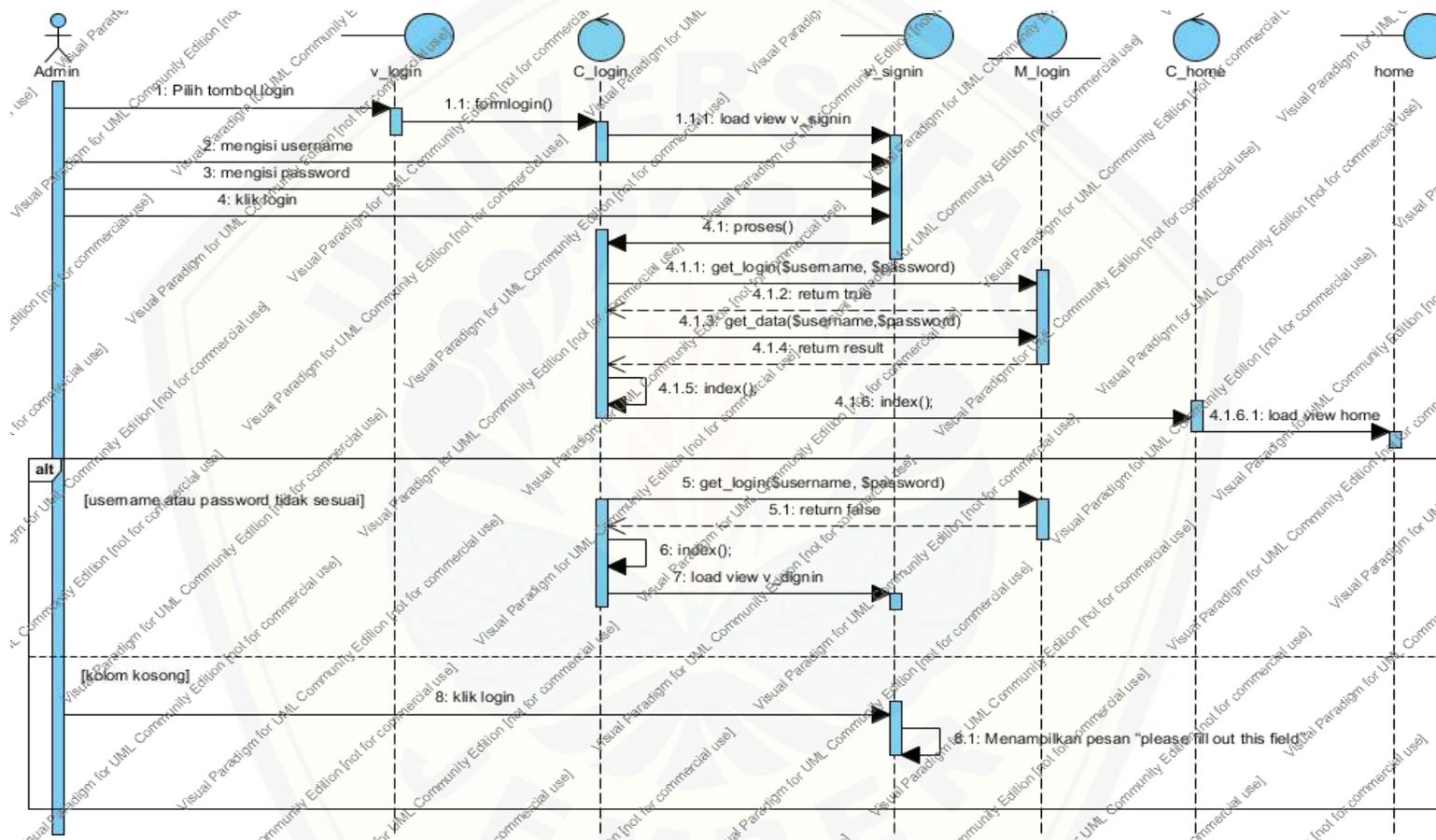
B.15 Activity Diagram Melakukan Pendaftaran



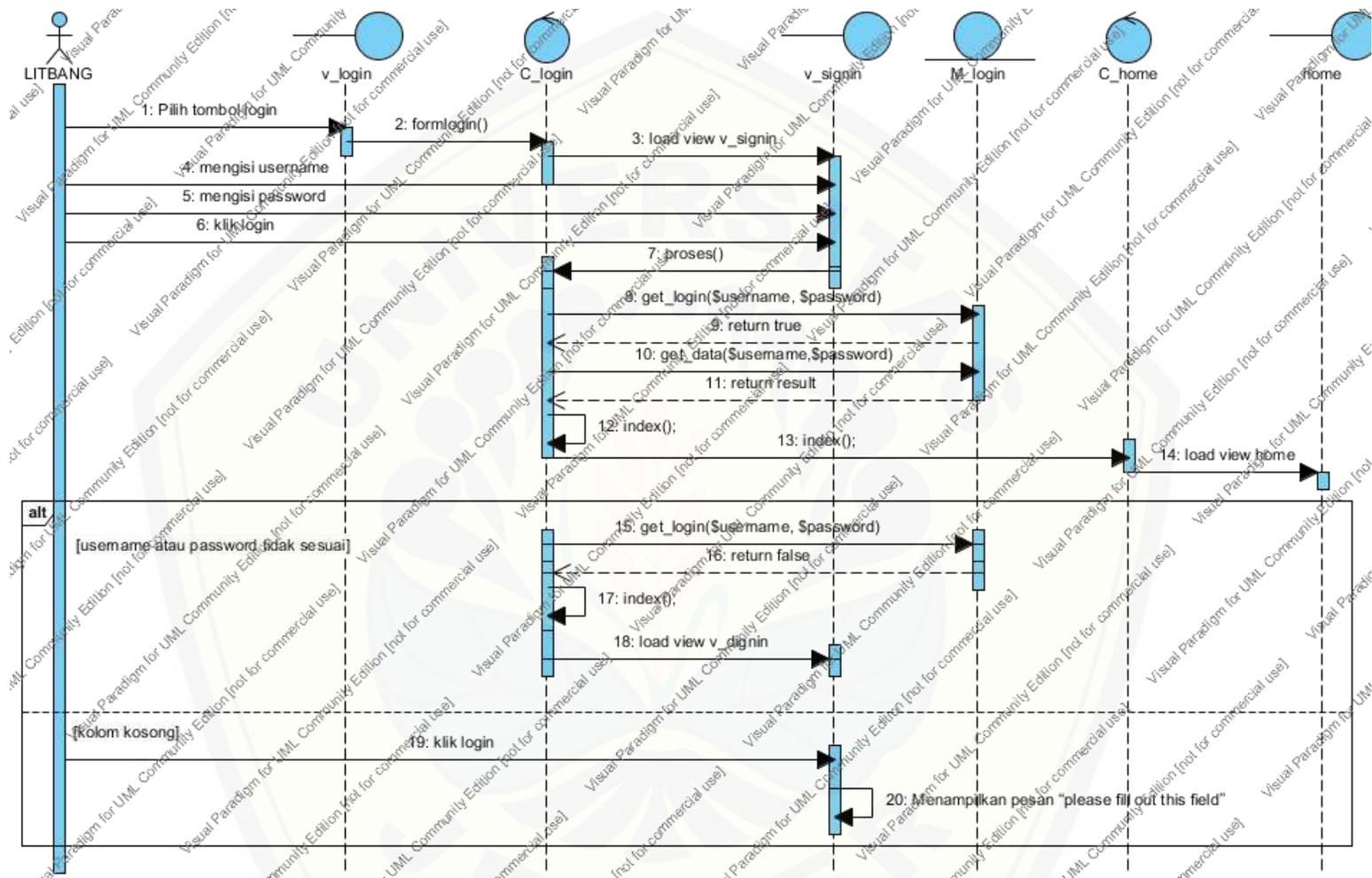
Gambar B. 16 Activity Diagram Melakukan Pendaftaran

LAMPIRAN C. SEQUENCE DIAGRAM

C.1 Sequence Diagram Login

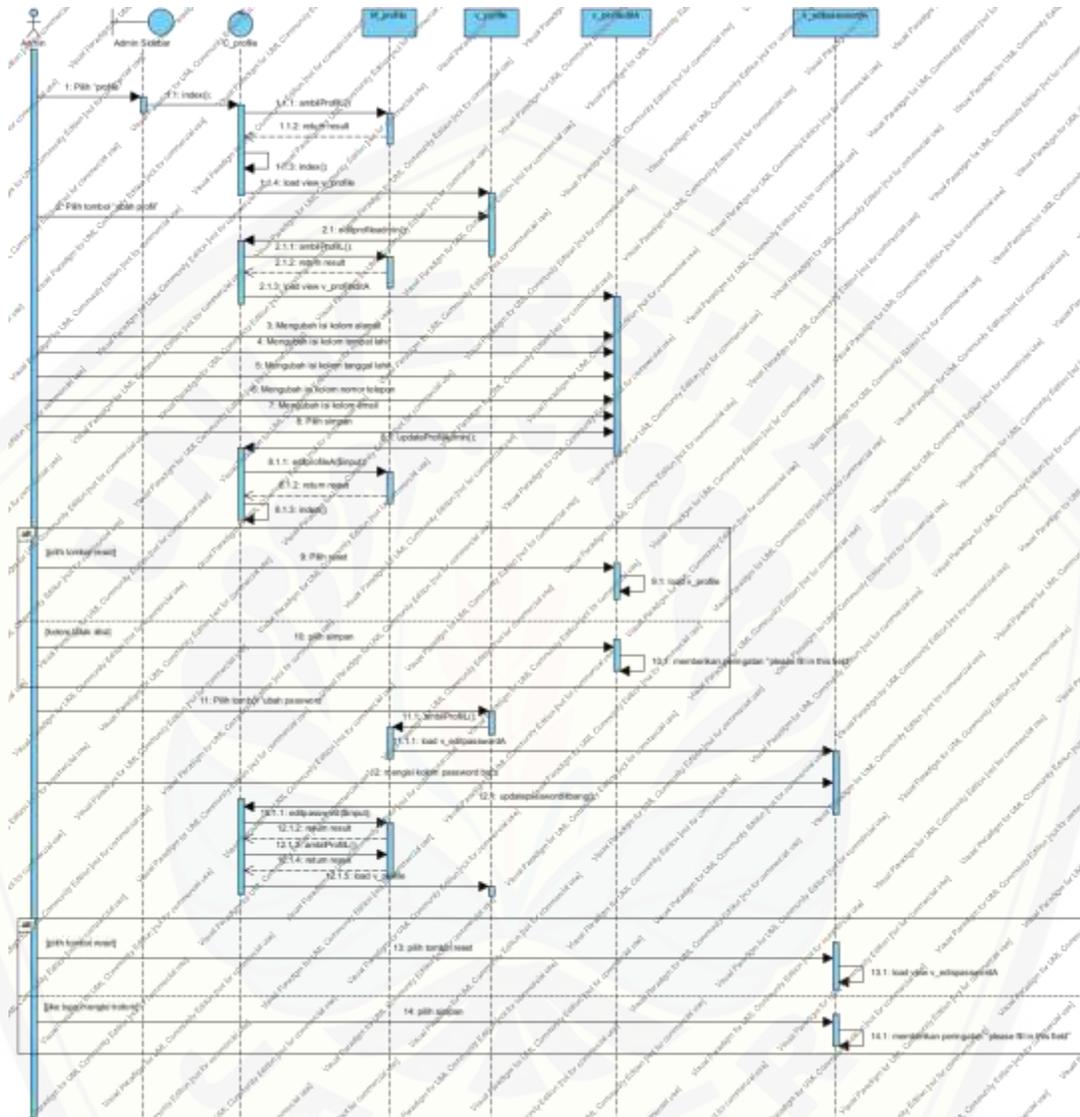


Gambar C. 1 Sequence diagram login (admin)

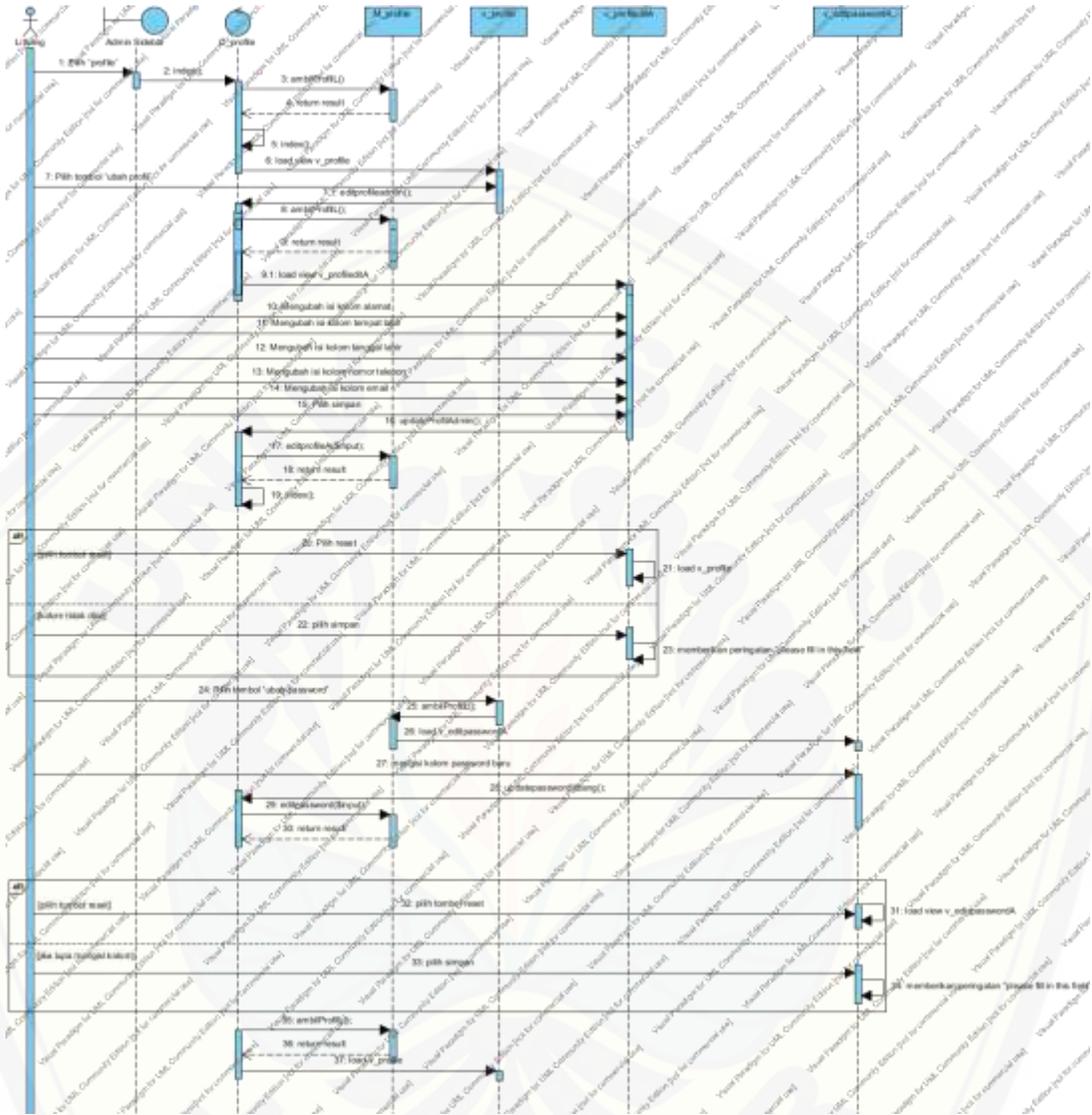


Gambar C. 2 Sequence diagram login (LITBANG)

C.2 Sequence Diagram Mengelola Data Profil User

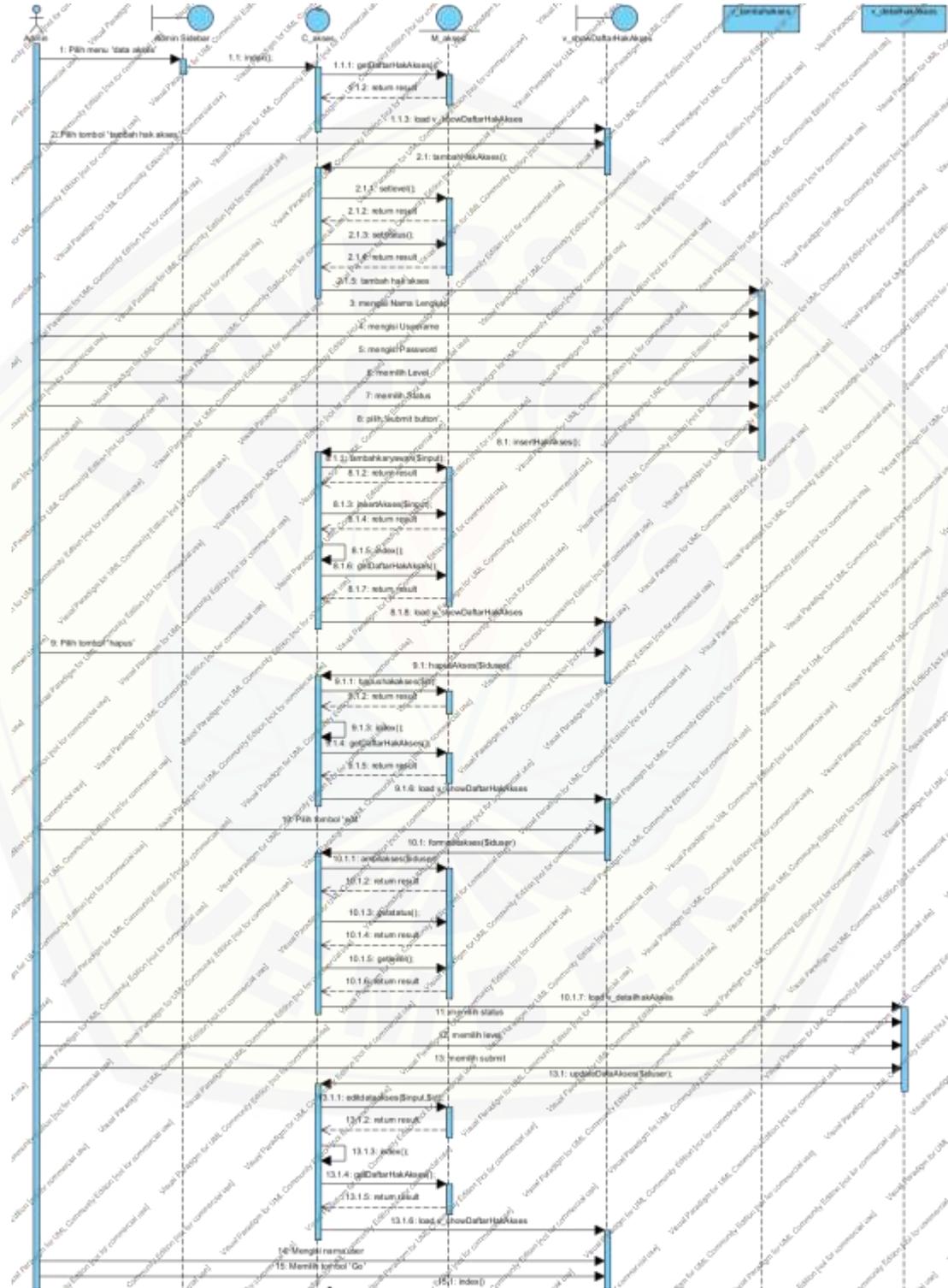


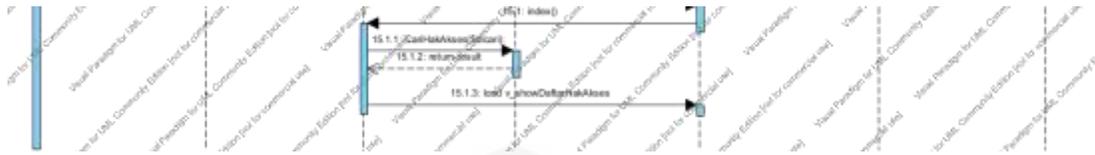
Gambar C. 3 Sequence diagram mengelola data profil user (admin)



Gambar C. 4 Sequence diagram mengelola data profil user (LITBANG)

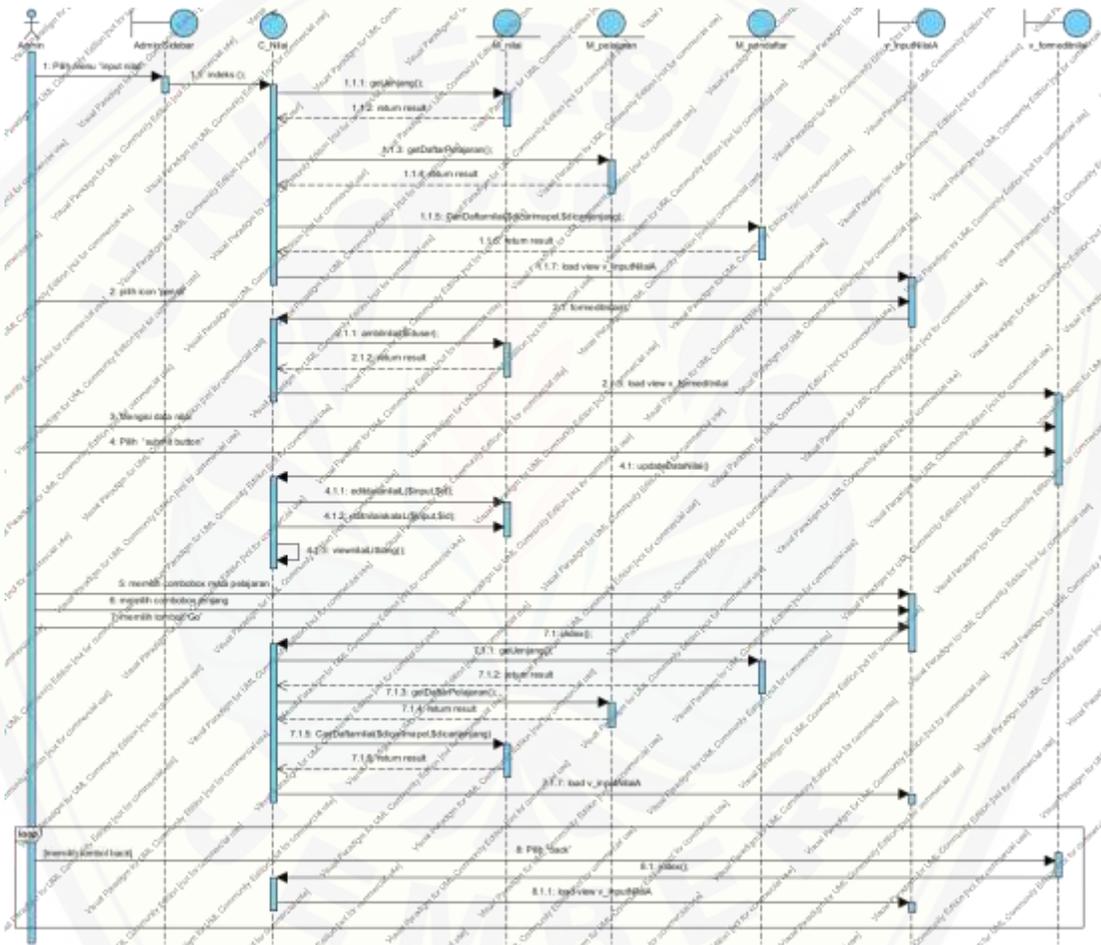
C.3 Sequence Diagram Mengelola Data Akses





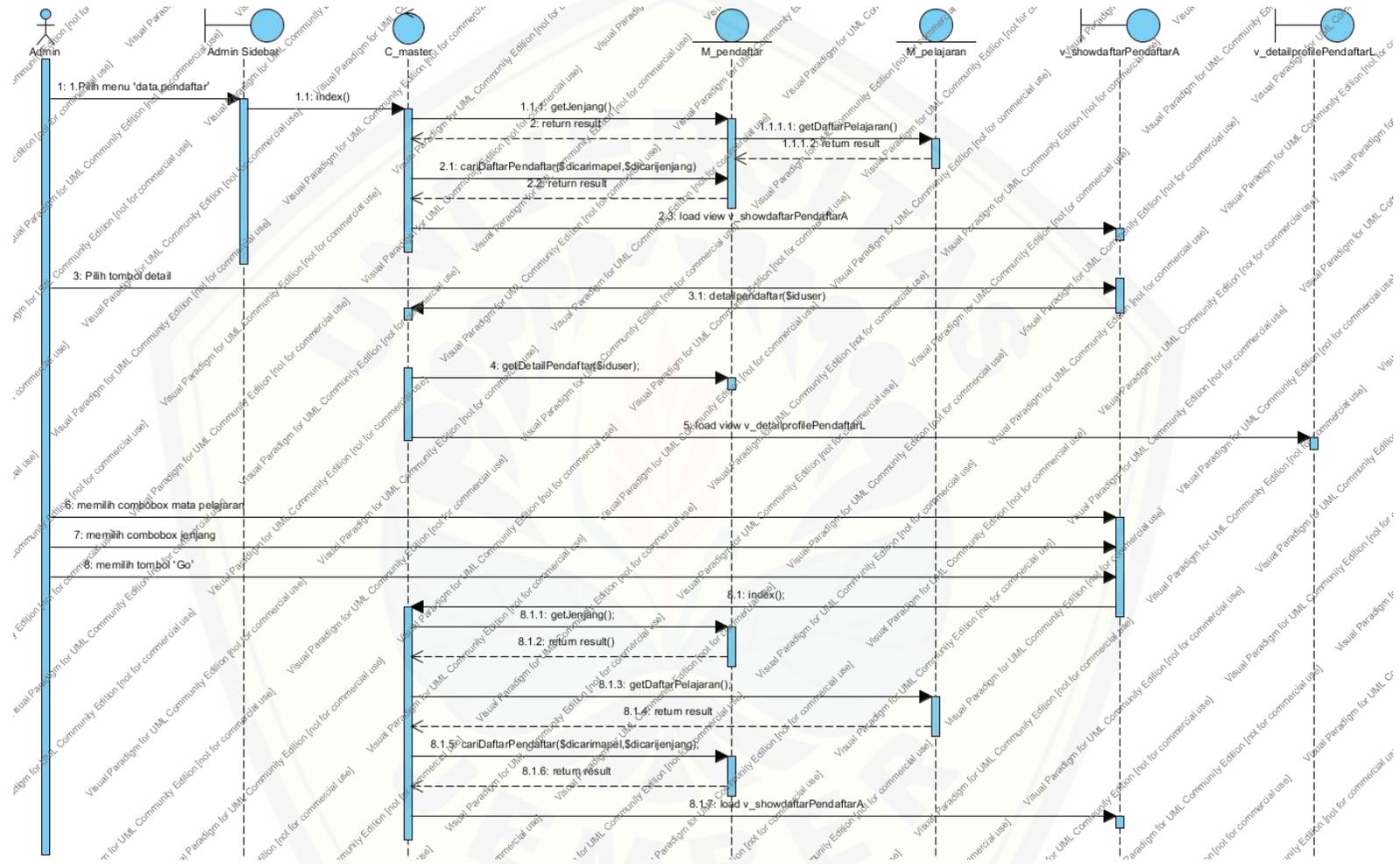
Gambar C. 5 Sequence diagram mengelola data akses

C.4 Sequence Diagram Mengelola Nilai Pendaftar



Gambar C. 6 Sequence diagram mengelola nilai pendaftar

## C.5 Sequence Diagram Menampilkan Data Profile Pendaftar



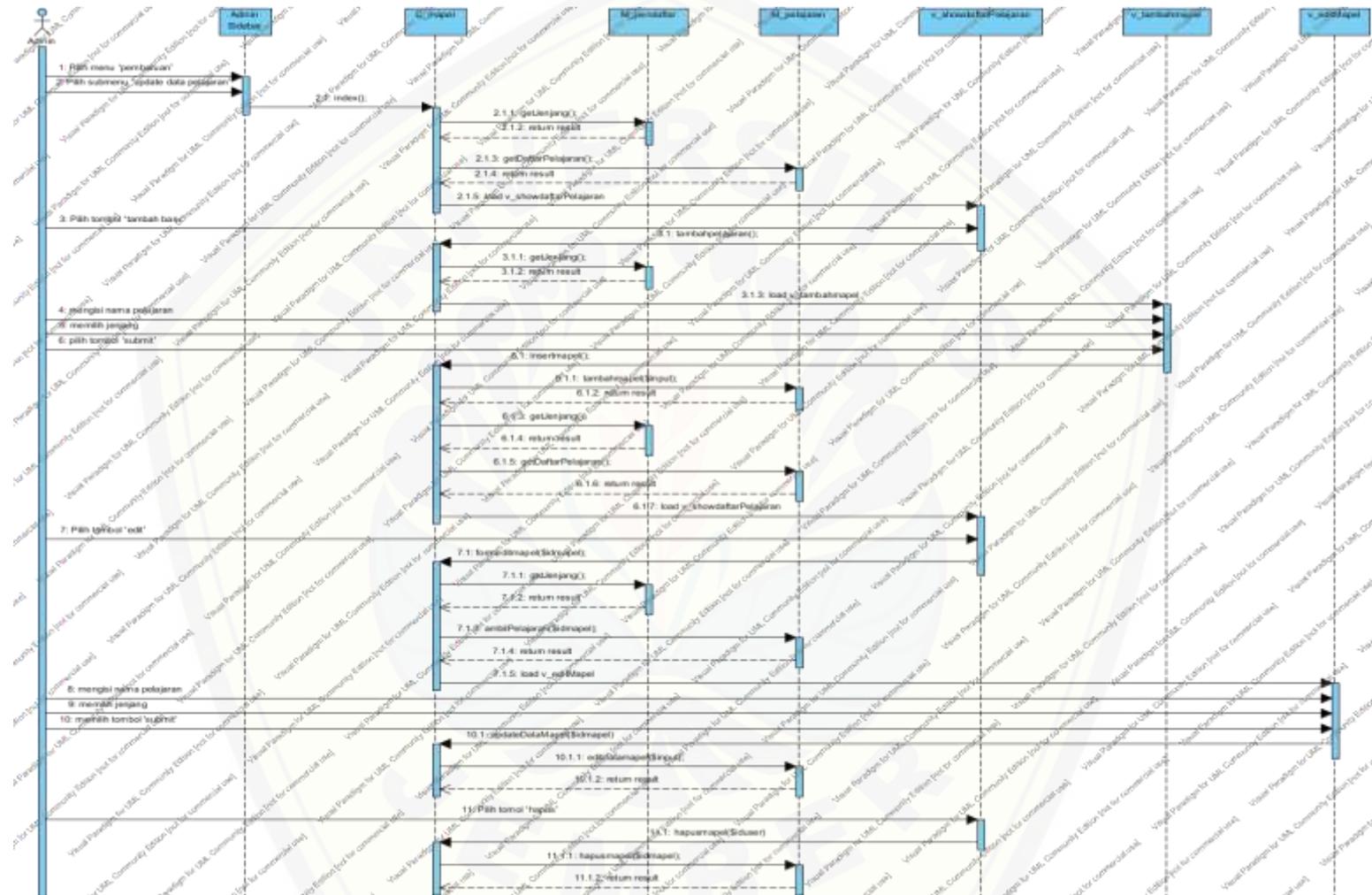
Gambar C. 7 Sequence diagram menampilkan data profile pendaftar

C.6 Sequence Diagram Menampilkan Data Nilai Pendaftar



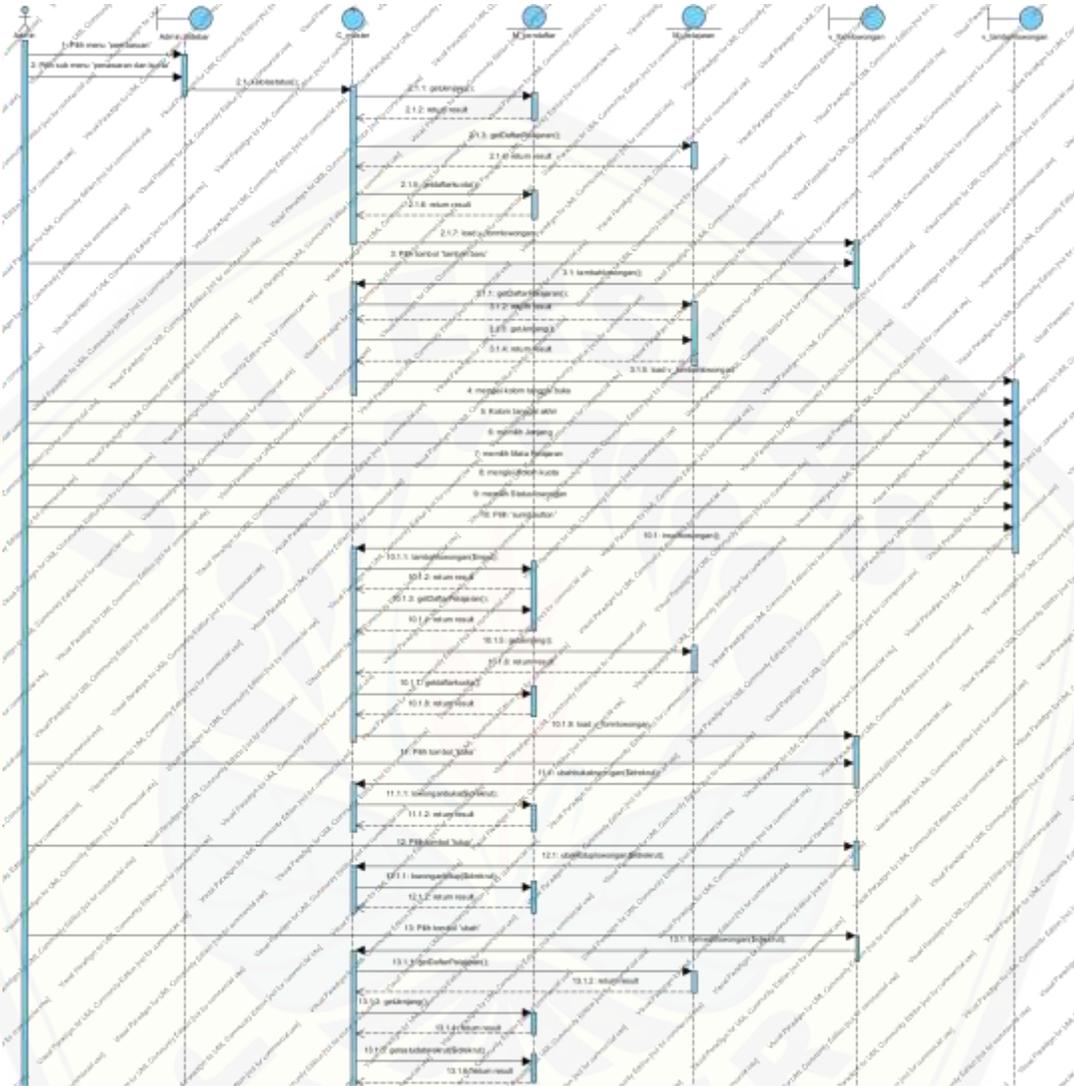


C.7 Sequence Diagram Mengelola Data Pelajaran



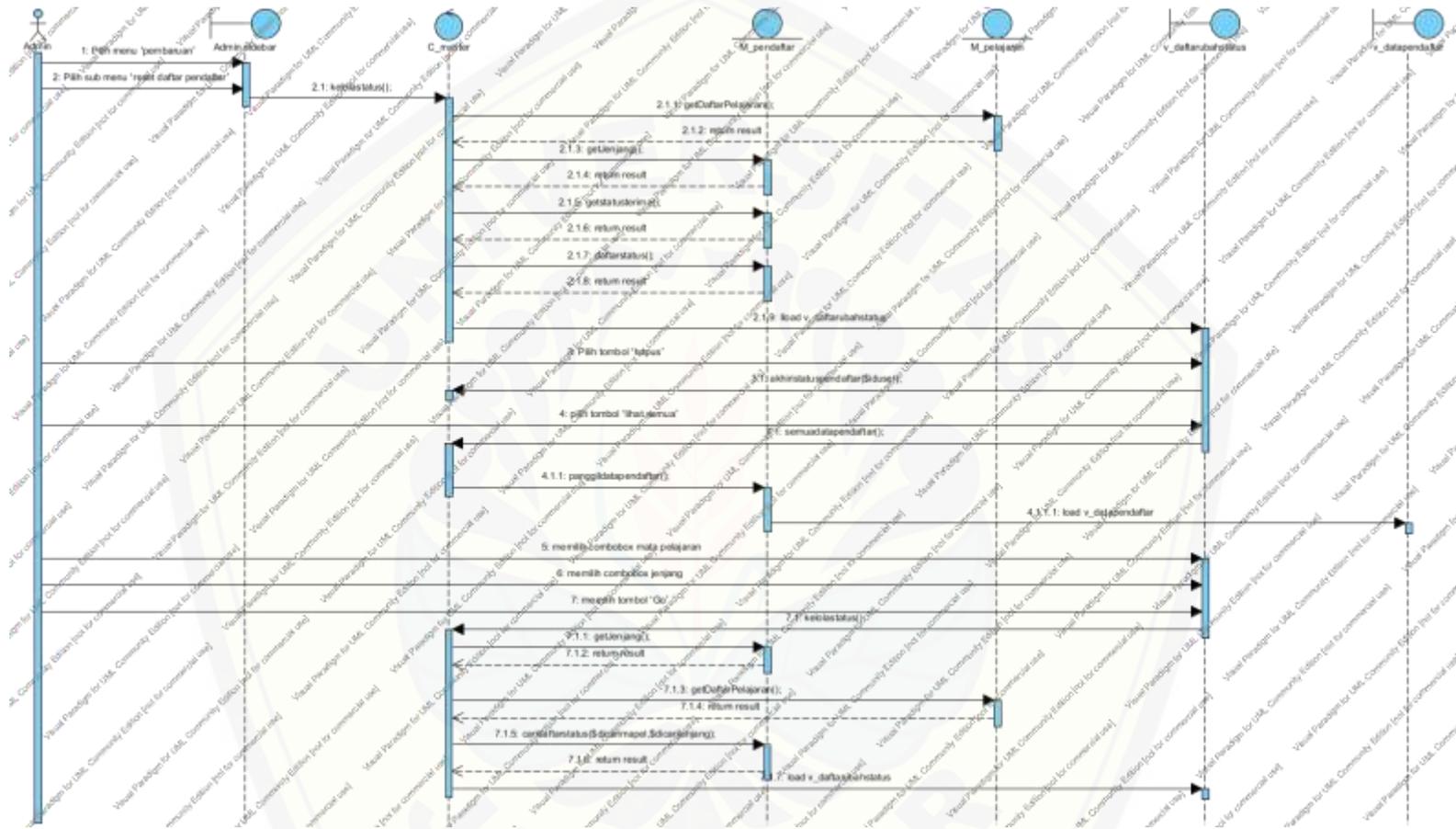
Gambar C. 4 Sequence diagram mengelola data pelajaran

C.8 Sequence Diagram Mengelola Kuota Lowongan



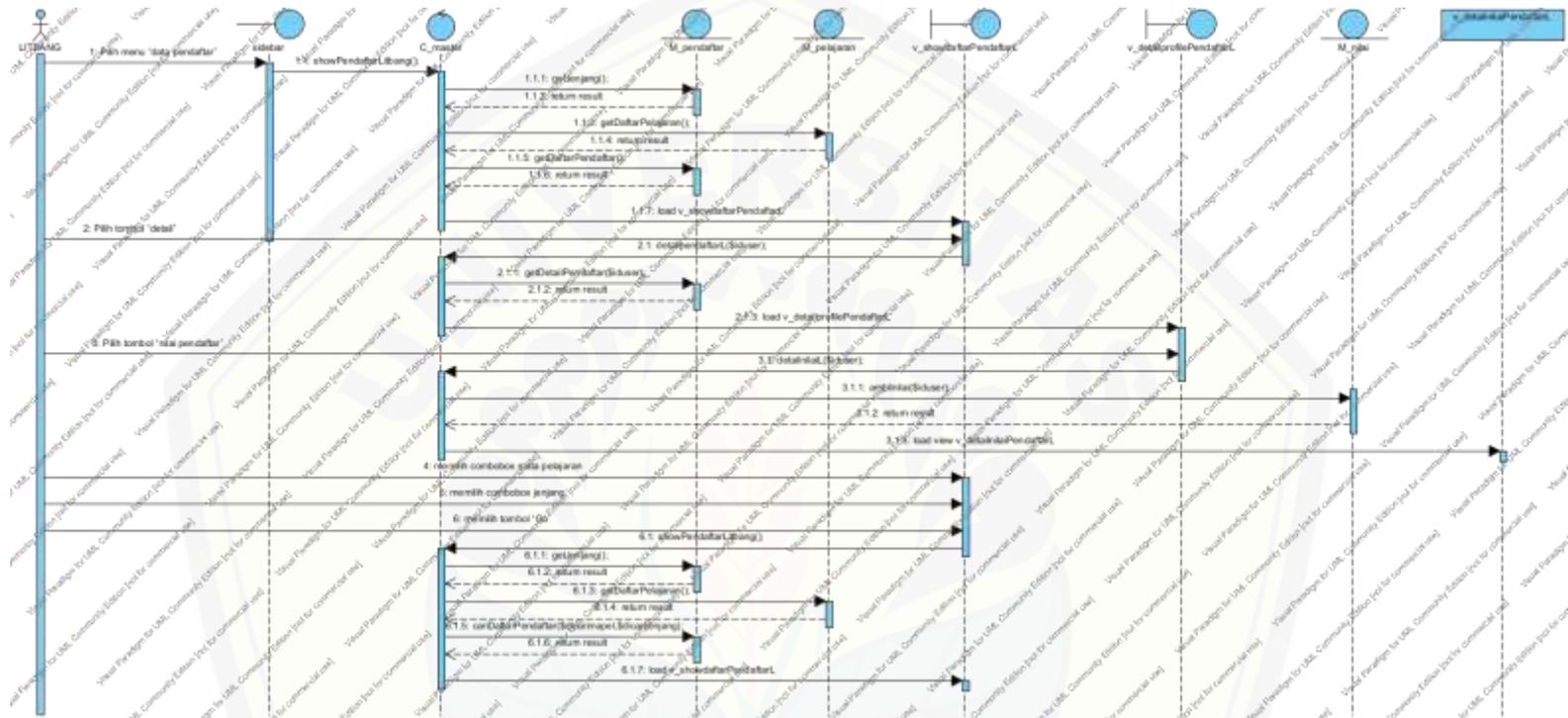


C.9 Sequence Diagram Mengelola Data Status Pendaftaran Para Pendaftar



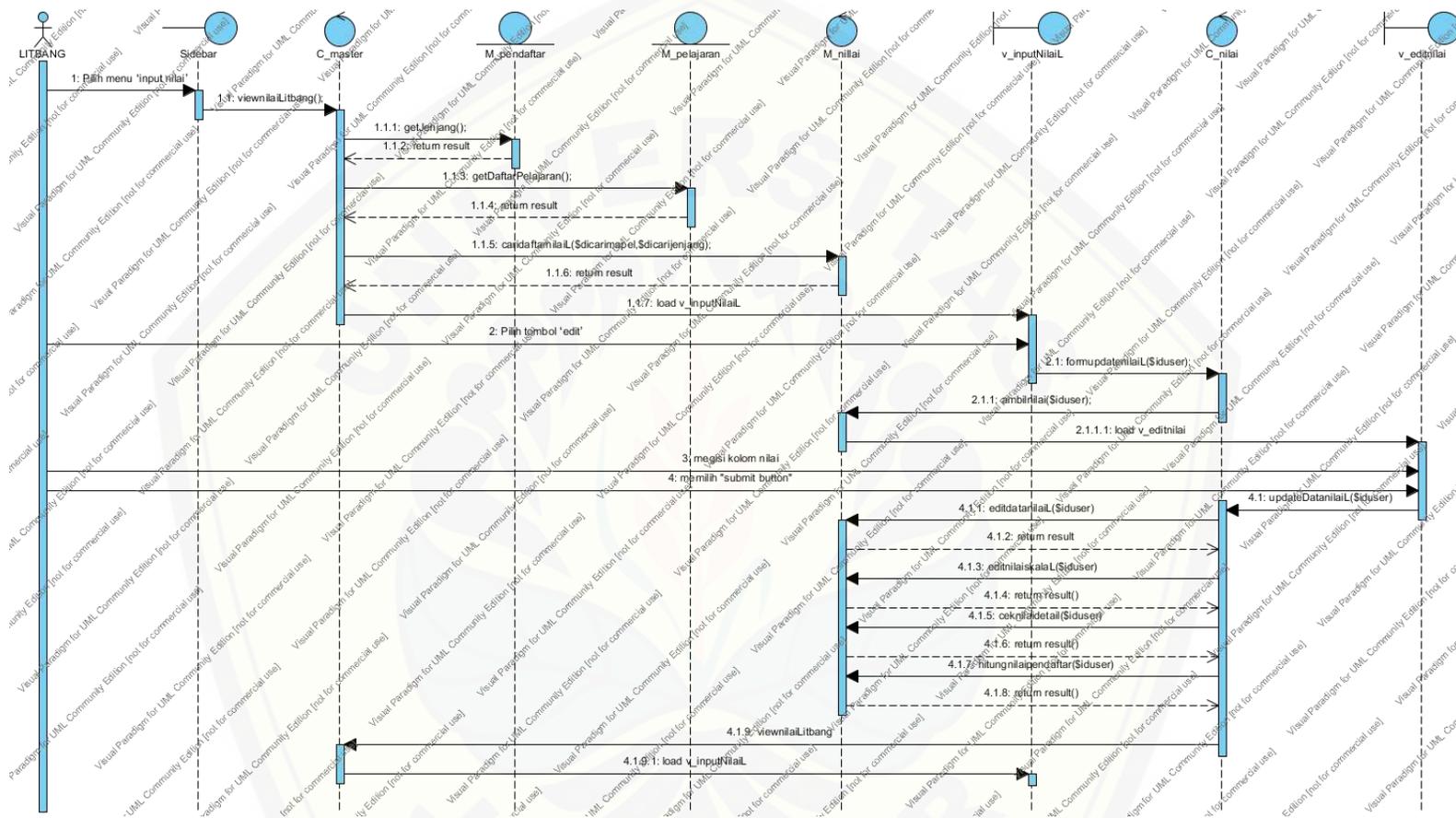
Gambar C. 6 Sequence diagram mengelola data status pendaftaran para pendaftar

C.10 Sequence Diagram Menampilkan Data Pendaftar



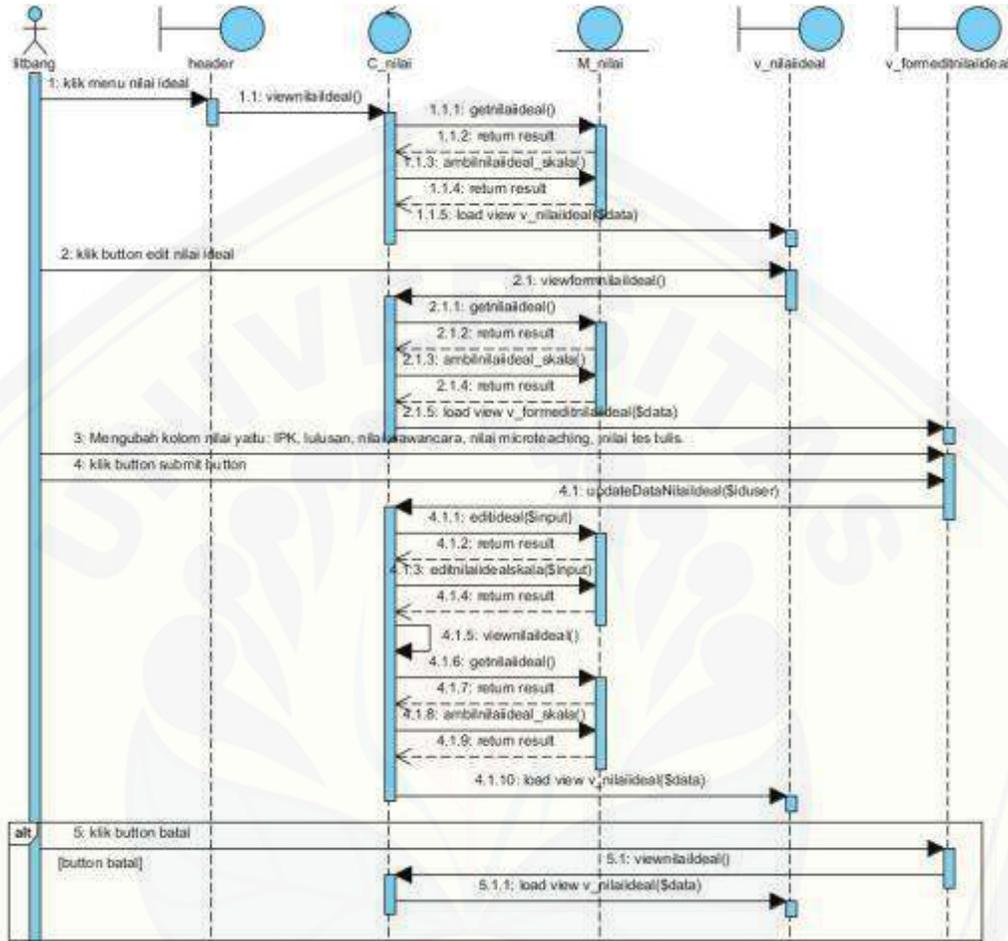
Gambar C. 12 Sequence diagram menampilkan data pendaftar

C.11 Sequence Diagram Update Nilai Wawancara dan Microteaching



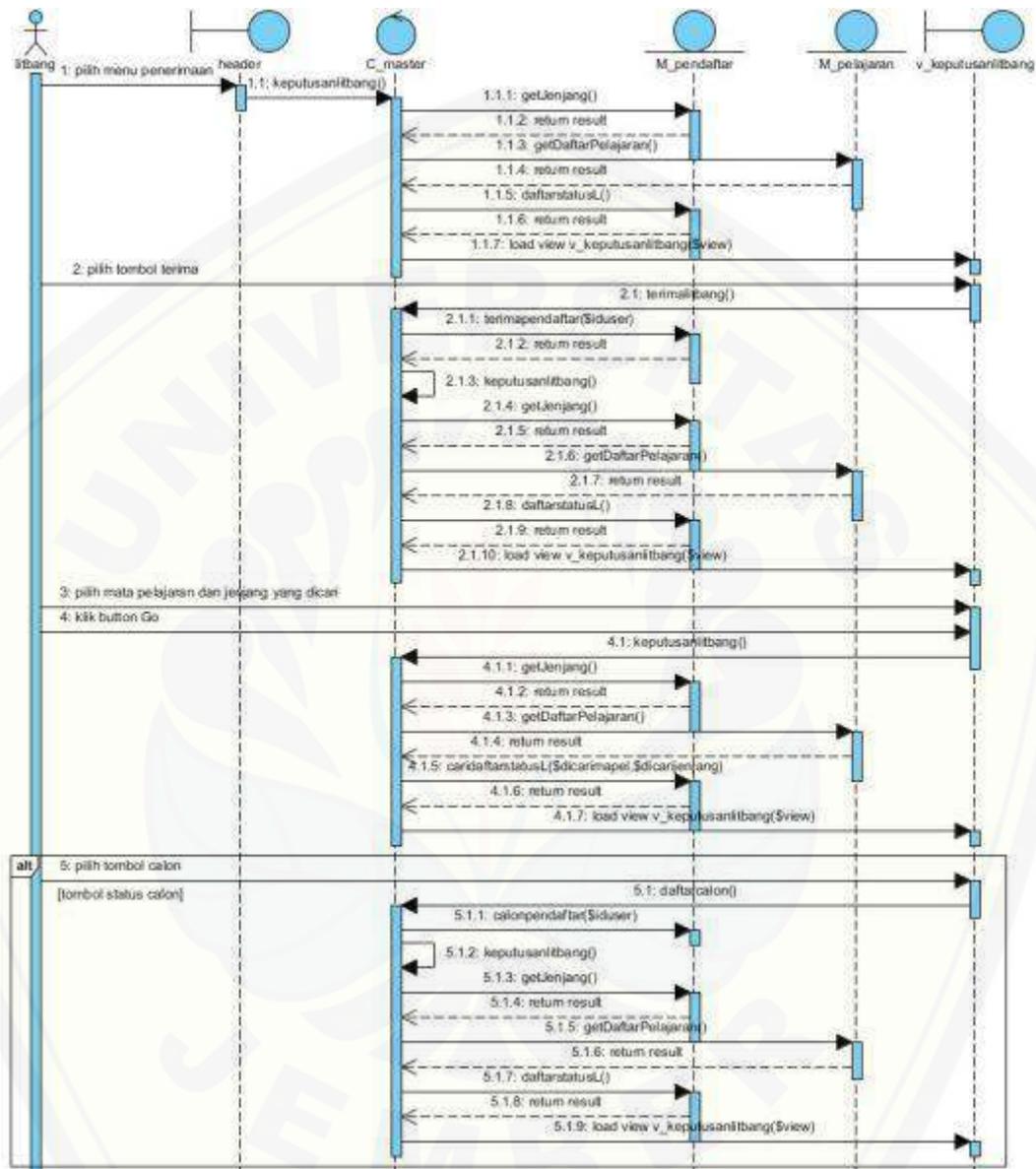
Gambar C. 13 Sequence diagram Update nilai wawancara dan *microteachi*

C.12 Sequence Diagram Mengelola Nilai Ideal Kriteria



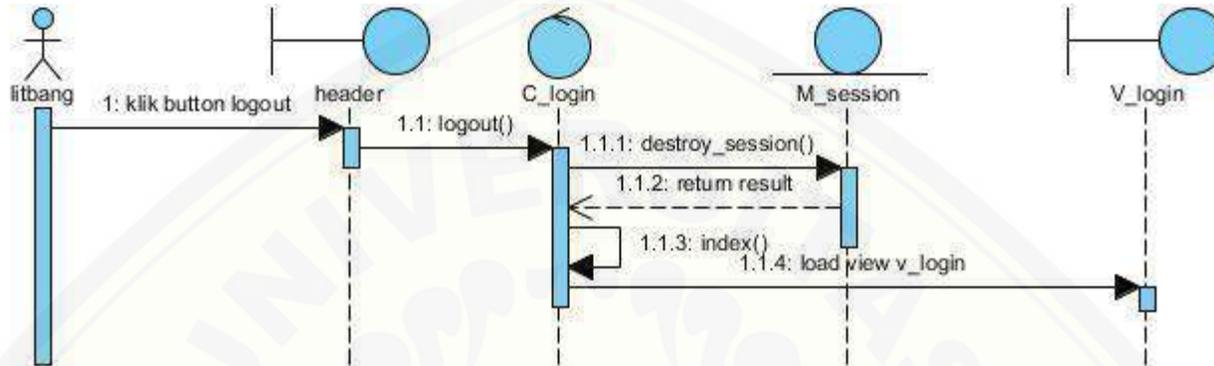
Gambar C. 14 Sequence diagram mengelola nilai ideal kriteria

C.13 Sequence Diagram Menyetujui Hasil Penerimaan

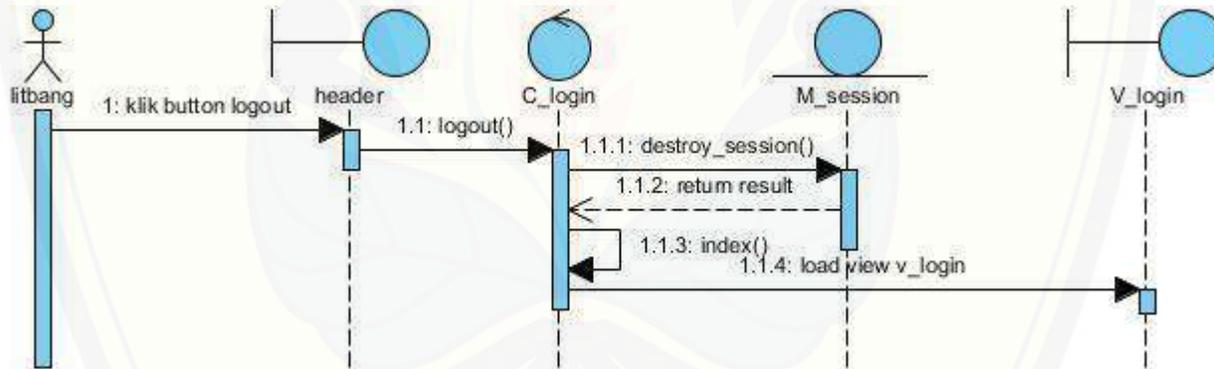


Gambar C. 15 Sequence diagram menyetujui hasil penerimaan

C.14 *Sequence Diagram Logout*

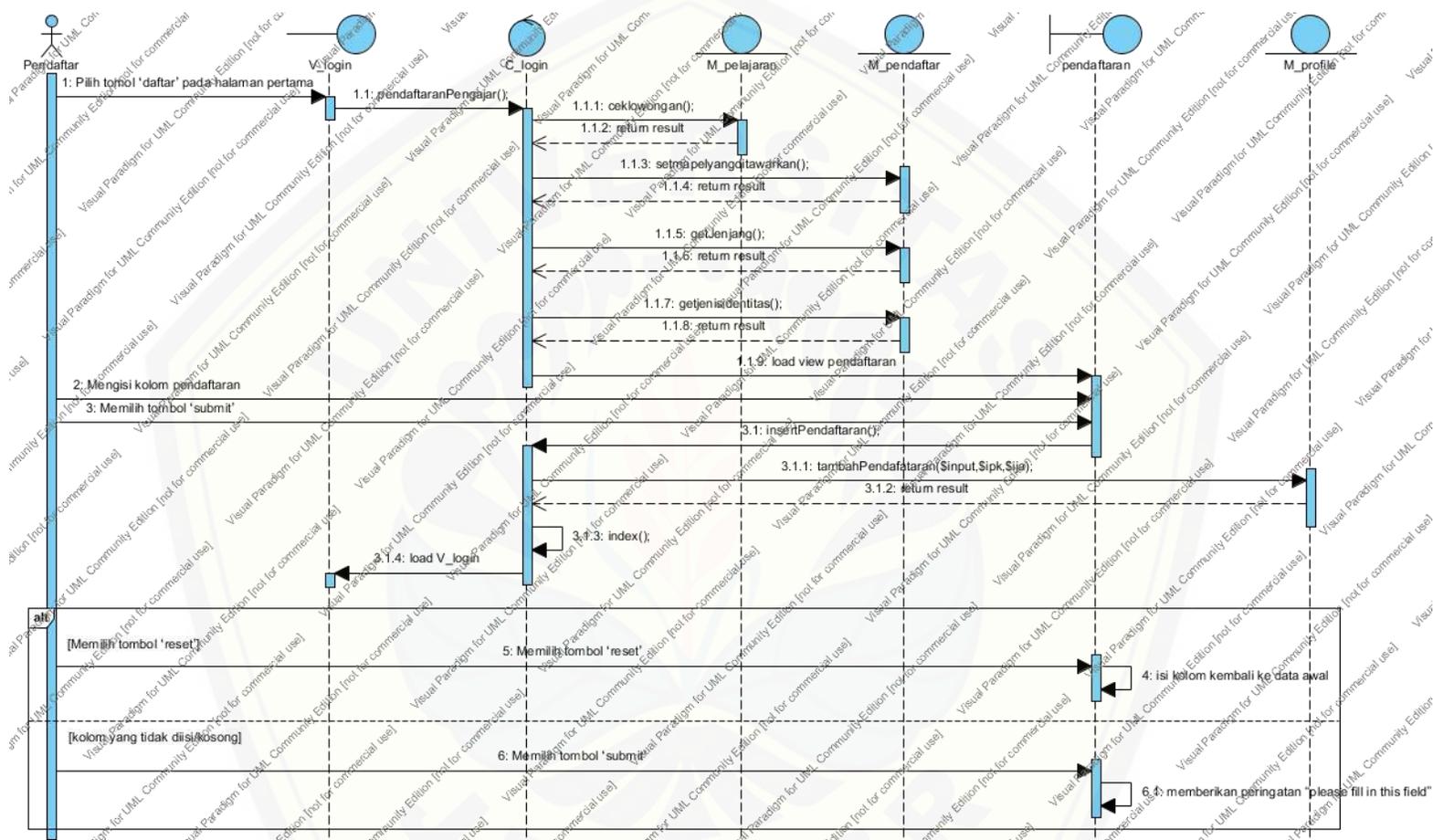


Gambar C. 16 *Sequence diagram logout (admin)*



Gambar C. 17 *Sequence diagram logout (LITBANG)*

## C.15 Sequence Diagram Melakukan Pendaftaran



Gambar C. 18 Sequence diagram melakukan pendaftaran

**LAMPIRAN D. KODE PROGRAM****D.1. Kode program *function* updateDataNilaiIdeal(\$iduser)**

No.	Kode
123	function updateDataNilaiIdeal(\$iduser) {
124	\$input['IPK'] = \$this->input->post('IPK');
125	\$input['ijazah'] = \$this->input->post('ijazah');
126	\$input['wawancara'] = \$this->input->post('wawancara');
127	\$input['microteaching'] = \$this->input->post('microteaching');
128	\$input['tesTulis'] = \$this->input->post('tesTulis');
129	\$this->M_nilai->editideal(\$input);
130	\$this->M_nilai->editnilaiidealskala(\$input);
131	redirect('C_nilai/viewnilaiIdeal');
132	}

**D.2. Kode program *function* editideal (\$iduser)**

No.	Kode
64	public function editideal(\$input)
65	{
66	return \$query= \$this->db->query("UPDATE tb_nilaiideal SET ipk =\$input[IPK],ijazah='\$input[ijazah]',wawancara='\$input[wawancara]',microteaching ='\$input[microteaching]',tesTulis='\$input[tesTulis]' where idnilaiideal=1");
67	}

**D.3. Kode program *function* editnilaiidealskala(\$iduser)**

No.	Kode
340	public function editnilaiidealskala(\$input)
341	{
342	\$ipkideal=0.0;
343	\$ijazahideal;
344	\$wawancaraideal=0.0;
445	\$microteachingideal=0.0;
346	\$testulisideal=0.0;
347	
348	\$idealipks=0.0;

---

```

349     $idealjazahs=0.0;
350     $idealwawancaras=0.0;
351     $idealmicroteachings=0.0;
352     $idealtestuliss=0.0;
353     =====> memanggil nilai yang diinputkan litbang <=====
354     $nilaiideal = $this->db->query("select * from tb_nilaiideal where idnilaiideal=1");
355     foreach ($nilaiideal->result_array() as $nilai) {
356         $ipkideal=$nilai['ipk'];
357         $ijazahideal=$nilai['ijazah'];
358         $wawancaraideal=$nilai['wawancara'];
359         $microteachingideal=$nilai['microteaching'];
360         $stestulisiideal=$nilai['tesTulis'];
361     }
362     =====> menghitung normalisasi nilai standar <=====
363     $idealipks=(((($ipkideal-0)/(4-0))*(1-0))+0);
364     if($ijazahideal== 'PTS'){
365         $ija= 0;
366     }else if ($ijazahideal =='PTN'){
367         $ija= 1;
368     }
369     $idealjazahs=(((($ija-0)/(1-0))*(1-0))+0);
370     $idealwawancaras=(((($wawancaraideal-0)/(100-0))*(1-0))+0);
371     $idealmicroteachings=(((($microteachingideal-0)/(100-0))*(1-0))+0);
372     $idealtestuliss=(((($stestulisiideal-0)/(100-0))*(1-0))+0);
373     =====> menyimpan nilai normalisasi <=====
374     $query= $this->db->query("UPDATE tb_nilaiidealskala SET ipk
= '$idealipks',ijazah='$idealjazahs',wawancara='$idealwawancaras',microteaching='$i
dealmicroteachings',tesTulis='$idealtestuliss' where idskalaideal=1");
    return array ($query);
375 }
376

```

---

#### D.4. Kode program *function* updateDataNilai(\$iduser)

No.	Kode
76	function updateDataNilai(\$iduser){
77	\$input['iduser'] = \$iduser;
78	\$input['IPK'] = \$this->input->post('IPK');
79	\$input['ijazah'] = \$this->input->post('ijazah');

---

---

```

80     $input['wawancara']    = $this->input->post('wawancara');
81     $input['microteaching'] = $this->input->post('microteaching');
82     $input['tesTulis']     = $this->input->post('tesTulis');
83     $this->M_nilai->editdatanilai($input);
84     $this->M_nilai->editnilaix($input);
85     $logic=$this->M_nilai->ceknilaidetail($iduser);
86     if ($logic == false ) {
87         $this->M_nilai->dropnilaiakhir($iduser);
88         redirect('C_nilai');
89     }else{
90         $this->M_nilai->hitungnilai_pendaftar($input);
91         redirect('C_nilai'); }}

```

---

#### D.5. Kode program *function* editdatanilai(\$input)

No.	Kode
60	public function editdatanilai(\$input)
61	{
62	return \$query= \$this->db->query("UPDATE tb_nilai SET IPK =' \$input[IPK]', ijazah=' \$input[ijazah]', wawancara=' \$input[wawancara]', microteaching=' \$input[microteaching]', tesTulis=' \$input[tesTulis]' where iduser=' \$input[iduser]'");
63	}

---

#### D.6. Kode program *function* editnilaix(\$input)

Kode
public function editnilaix(\$input) {
\$minipk=0.0;
\$minijazah=0.0;
\$minwawancara=0.0;
\$minmicroteahing=0.0;
\$minitestulis=0.0;
\$maxipk=0.0;
\$maxijazah=0.0;
\$maxwawancara=0.0;
\$maxmicroteahing=0.0;
\$maxtestulis=0.0;
\$newmin=0.0;

---

---

```

$newmax=1.0;

$ipkideal=0.0;
$ijazahideal=0.0;
$ija=0.0;
$wawancaraideal=0.0;
$microteachingideal=0.0;
$testulisideal=0.0;
=====> memanggil nilai ideal <=====
$nilaiideal = $this->db->query("select * from tb_nilaiideal ");
foreach ($nilaiideal->result_array() as $nilai) {
    $ipkideal=$nilai['ipk'];
    $ijazahideal=$nilai['ijazah'];
    $wawancaraideal=$nilai['wawancara'];
    $microteachingideal=$nilai['microteaching'];
    $testulisideal=$nilai['tesTulis'];
}

$Idealipks=0.0;
$Idealijazahs=0.0;
$Idealwawancaras=0.0;
$Idealmicroteachings=0.0;
$Idealtestuliss=0.0;
$Ipk = 0.0;
$Ijazahs=0.0;
$wawancaras=0.0;
$microteachings=0.0;
$tesTuliss=0.0;

==> menghitung normalisasi nilai kriteria yang diinputkan dan normalisasi nilai ideal
yang dipanggil <=
$Ipk=(((($input['IPK']-0)/(4-0))*(1-0))+0);
$Idealipks=(((($ipkideal-0)/(4-0))*(1-0))+0);
if($input['ijazah']== 'PTS'){
    $ij= 0;
    }else if ($input['ijazah'] =='PTN'){
    $ij= 1;
    }
$Ijazahs=(((($ij-0)/(1-0))*(1-0))+0);

```

---

---

```

if($ijazahideal== 'PTS'){
    $ija= 0;
    }else if ($ijazahideal =='PTN'){
    $ija= 1;
    }
$Idealijazahs=(((($ija-0)/(1-0))*(1-0))+0);
$swawancaras=(((($input['wawancara']-0)/(100-0))*(1-0))+0);
$Idealwawancaras=(((($wawancaraideal-0)/(100-0))*(1-0))+0);
$microteachings=(((($input['microteaching']-0)/(100-0))*(1-0))+0);
$Idealmicroteachings=(((($microteachingideal-0)/(100-0))*(1-0))+0);
$tesTuliss=(((($input['tesTulis']-0)/(100-0))*(1-0))+0);
$Idealtestuliss=(((($testulisideal-0)/(100-0))*(1-0))+0);

=====> menghitung nilai gap <=====
    $x1 =$Idealipks-$ipks;
    $x2 =$Idealijazahs-$ijazahs;
    $x3 =$Idealwawancaras-$wawancaras;
    $x4 =$Idealmicroteachings-$microteachings;
    $x5 =$Idealtestuliss-$tesTuliss;

=====> menyimpan normalisasi nilai kriteria <=====
$query= $this->db->query("UPDATE tb_nilaiskala SET ipk
='$ipks',ijazah='$ijazahs',wawancara='$wawancaras',microteaching='$microteachings',t
esTulis='$tesTuliss' where iduser='$input[iduser]'");

=====> menyimpan nilai gap <=====
$query2= $this->db->query("UPDATE tb_x SET ipk
='$x1',ijazah='$x2',wawancara='$x3',microteaching='$x4',tesTulis='$x5' where
iduser='$input[iduser]'");
    return array ($query,$query2);
}

```

---

#### D.7. Kode program *function* ceknilaidetail(\$iduser)

---

Kode

---

```

public function ceknilaidetail($iduser){
    $ipkideals=0.0;
    $ijazahideals=0.0;
    $wawancaraideals=0.0;
    $microteachingideals=0.0;

```

---

---

```

        $testulsideals=0.0;
=====> mengambil value table normalisasi dari nilai ideal <=====
$nilaiideal = $this->db->query("select * from tb_nilaiidealskala ");
    foreach ($nilaiideal->result_array() as $nilai)
    { $ipkideals=$nilai['ipk'];
      $ijazahideals=$nilai['ijazah'];
      $wawancaraideals=$nilai['wawancara'];
      $microteachingideals=$nilai['microteaching'];
      $testulsideals=$nilai['tesTulis'];
    }
    $ipks = 0.0;
    $ijazahs=0.0;
    $wawancaras=0.0;
    $microteachings=0.0;
    $tesTuliss=0.0;
=====> mengambil value tabel kriteria pendaftar sesuai id pendaftar <=====
$nilai pendaftar = $this->db->query("select * from tb_nilaiskala where
iduser='$iduser' ");
    foreach ($nilai pendaftar->result_array() as $nilai)
    {
      $ipks=$nilai['ipk'];
      $ijazahs=$nilai['ijazah'];
      $wawancaras=$nilai['wawancara'];
      $microteachings=$nilai['microteaching'];
      $tesTuliss=$nilai['tesTulis'];
    }
=====> memunculkan nilai boolean <=====
if ($ipks>=$ipkideals && $ijazahs>=$ijazahideals &&
    $wawancaras>=$wawancaraideals&& $tesTuliss>=$testulsideals&&
    $microteachings>=$microteachingideals)
    {
      return true;
    }else{
return false;
    }}

```

---

#### D.8. Kode program *function* dropnilaiakhir(\$iduser)

---

Kode

---

---

```

public function dropnilaiakhir($iduser)
{
    $this->db->query("UPDATE tb_nilai SET nilaiAkhir ='0' where
iduser='$iduser'");
}

```

---

#### D.9. Kode program *function* hitungnilai pendaftar (\$iduser)

---

Kode

---

```

public function hitungnilai pendaftar($input) {
    $iduser=$input['iduser'];
    $ipkideals=0.0;
    $ijazhideals=0.0;
    $wawancaraideals=0.0;
    $microteachingideals=0.0;
    $testulisideals=0.0;
    =====> mengambil value table normalisasi nilai ideal <=====
    $nilaiideal = $this->db->query("select * from tb_nilaiidealskala ");
    foreach ($nilaiideal->result_array() as $nilai)
    { $ipkideals=$nilai['ipk'];
      $ijazhideals=$nilai['ijazah'];
      $wawancaraideals=$nilai['wawancara'];
      $microteachingideals=$nilai['microteaching'];
      $testulisideals=$nilai['tesTulis'];
    }
    $ipks = 0.0;
    $ijazhs=0.0;
    $wawancaras=0.0;
    $microteachings=0.0;
    $tesTuliss=0.0;
    =====> mengambil value table normalisasi nilai kriteria pendaftar <=====
    $nilai pendaftar = $this->db->query("select * from tb_nilaiskala where
iduser='$iduser' ");
    foreach ($nilai pendaftar->result_array() as $nilai)
    { $ipks=$nilai['ipk'];
      $ijazhs=$nilai['ijazah'];
      $wawancaras=$nilai['wawancara'];
      $microteachings=$nilai['microteaching'];
      $tesTuliss=$nilai['tesTulis'];
    }
}

```

---

---

```

    }
    if ($ipks>=$ipkideals && $ijazahs>=$ijazahideals &&
    $wawancaras>=$wawancaraideals&& $tesTuliss>=$testulsideals&&
    $microteachings>=$microteachingideals)
        { $this->hitunggap($iduser);
          $this->fuzzyfikasi($iduser);
          $this->hitungratarata($iduser);
        }else{ }
    }

```

---

#### D.10. Kode program *function* hitunggap(\$iduser)

---

##### Kode

---

```

function hitunggap($iduser) {
    $idealipks=0.0;
    $idealijazahs=0.0;
    $idealwawancaras=0.0;
    $idealmicroteachings=0.0;
    $idealtestuliss=0.0;
    =====> mengambil value table normalisasi nilai standar <=====
    $nilaiideal = $this->db->query("select * from tb_nilaiidealskala ");
    foreach ($nilaiideal->result_array() as $nilai) {
        $ipkideals=$nilai['ipk'];
        $ijazahideals=$nilai['ijazah'];
        $wawancaraideals=$nilai['wawancara'];
        $microteachingideals=$nilai['microteaching'];
        $testulsideals=$nilai['tesTulis'];
    }
    $ipks = 0.0;
    $ijazahs=0.0;
    $wawancaras=0.0;
    $microteachings=0.0;
    $tesTuliss=0.0;
    =====> memanggil value table normalisasi nilai kriteria <=====
    $nilai pendaftar = $this->db->query("select * from tb_nilaiskala where
    iduser='$iduser' ");
    foreach ($nilai pendaftar->result_array() as $nilai) {
        $ipks=$nilai['ipk'];
        $ijazahs=$nilai['ijazah'];
    }
}

```

---

---

```

    $wawancaras=$nilai['wawancara'];
    $microteachings=$nilai['microteaching'];
    $tesTuliss=$nilai['tesTulis'];
  }
=====> menghitung nilai gap <=====
    $x1=$ipkideals-$ipks;
    $x2=$ijazahideals-$ijazahs;
    $x3=$wawancaraideals-$wawancaras;
    $x4=$microteachingideals-$microteachings;
    $x5=$testulisideals-$tesTuliss;

=====> menyimpan nilai gap <=====
$this->db->query("UPDATE tb_x SET ipk
=$x1',ijazah='$x2',wawancara='$x3',microteaching='$x4',tesTulis='$x5' where
iduser='$iduser'");
  }

```

---

#### D.11. Kode program *function* fuzzyfikasi(\$iduser)

---

##### Kode

---

```

function fuzzyfikasi($iduser) // menghitung fuzzifikasi nilai pendaftar
{
    $nilaix = 0;
    $xipk=0;
    $xijazah=0;
    $xwawancara=0;
    $xmicroteaching=0;
    $xtesTulis=0;
    $gap=0;
    $bobot=0;
    $bobotipk=0;
    $bobotijazah=0;
    $bobotwawancara=0;
    $bobotmicroteaching=0;
    $bobotTestulis=0;

    $keanggotaanipkA=0;
    $keanggotaanipkB=0;
    $keanggotaanijazahA=0;

```

---

---

```

$keanggotaanijazahB=0;
$keanggotaanwawancaraA=0;
$keanggotaanwawancaraB=0;
$keanggotaanmicroteachingA=0;
$keanggotaanmicroteachingB=0;
$keanggotaantesTulisA=0;
$keanggotaantesTulisB=0;
$A=0;
$B=0;
$gapmax=0.0;
$nilaimin=0.0;
$ipkideals=0.0;
$ijazahideals=0.0;
$wawancaraideals=0.0;
$microteachingideals=0.0;
$stesTulisideals=0.0;
=====> mengambil nilai gap <=====
$nilaiskala = $this->db->query("select * from tb_x where iduser='$iduser'");
    foreach ($nilaiskala->result_array() as $nilai) {
        $xipk=$nilai['ipk'];
        $xijazah=$nilai['ijazah'];
        $xwawancara=$nilai['wawancara'];
        $xmicroteaching=$nilai['microteaching'];
        $xtesTulis=$nilai['tesTulis'];
    }

=====> mengambil value table normalisasi nilai ideal <=====
$nilaiideal=$this->db->query("select * from tb_nilaiidealskala");
    foreach ($nilaiideal->result_array() as $nilai) {
        $ipkideals=$nilai['ipk'];
        $ijazahideals=$nilai['ijazah'];
        $wawancaraideals=$nilai['wawancara'];
        $microteachingideals=$nilai['microteaching'];
        $stesTulisideals=$nilai['tesTulis'];
    }

=====> mengambil nilai minimal <=====
$nilaimin=min($ipkideals,$ijazahideals,$wawancaraideals,$microteachingideals,$stesTulisideals);
    $gapmin=-(1);

```

---

---

```

$gapmax=1;
=====> menghitung bobot minimal <=====
$A=((( $gapmax-($gapmax-$nilaimin))+ $gapmin)/2); //bobotminimal
=====> menghitung bobot maksimal <=====
$B=((( $gapmax-($gapmax-$nilaimin))+ $gapmax)/2); //bobotmaximal

=====> menghitung bobot masing-masing nilai kriteria <=====
$bobotipk=((( $gapmax-($gapmax-$nilaimin))+ $xipk)/2);
$bobotijazah=((( $gapmax-($gapmax-$nilaimin))+ $xijazah)/2);
$bobotwawancara=((( $gapmax-($gapmax-$nilaimin))+ $xwawancara)/2);
$bobotmicroteaching=((( $gapmax-($gapmax-$nilaimin))+ $xmicroteaching)/2);
$bobotTestulis=((( $gapmax-($gapmax-$nilaimin))+ $xtesTulis)/2);

=====> menghitung fuzzifikasi kiri dan fuzzykasi kanan <=====
$keanggotaanipkA=($B-$bobotipk)/($B-$A);
$keanggotaanipkB=($bobotipk-$A)/($B-$A);
$keanggotaanijazahA=($B-$bobotijazah)/($B-$A);
$keanggotaanijazahB=($bobotijazah-$A)/($B-$A);
$keanggotaanwawancaraA=($B-$bobotwawancara)/($B-$A);
$keanggotaanwawancaraB=($bobotwawancara-$A)/($B-$A);
$keanggotaanmicroteachingA=($B-$bobotmicroteaching)/($B-$A);
$keanggotaanmicroteachingB=($bobotmicroteaching-$A)/($B-$A);
$keanggotaantesTulisA=($B-$bobotTestulis)/($B-$A);
$keanggotaantesTulisB=($bobotTestulis-$A)/($B-$A);

=====> menyimpan fuzzifikasi <=====
$query= $this->db->query("UPDATE tb_fuzzyfikasia SET ipk
='$keanggotaanipkA',ijazah='$keanggotaanijazahA',wawancara='$keanggotaanwawan
caraA',microteaching='$keanggotaanmicroteachingA',tesTulis='$keanggotaantesTulis
A' where iduser='$iduser'");

$query2= $this->db->query("UPDATE tb_fuzzyfikasib SET ipk
='$keanggotaanipkB',ijazah='$keanggotaanijazahB',wawancara='$keanggotaanwawan
caraB',microteaching='$keanggotaanmicroteachingB',tesTulis='$keanggotaantesTulis
B' where iduser='$iduser'");
return array ($query,$query2);
}

```

---

D.12. Kode program *function* hitungratarata(\$iduser)

## Kode

---

```

function hitungratarata($iduser)
{
    $Alphaipk1=0;
    $Alphaijazah1=0;
    $Alphawawancara1=0;
    $Alphamicroteaching1=0;
    $Alphatestulis1=0;
    $Alphaipk2=0;
    $Alphaijazah2=0;
    $Alphawawancara2=0;
    $Alphamicroteaching2=0;
    $Alphatestulis2=0;
    $jumlahalpha=0;
    $jumlahz=0;
    =====> mengambil nilai fuzzifikasi <=====
        $alphaA= $this->db->query("SELECT * from tb_fuzzyfikasia where
iduser='$iduser'");
        foreach ($alphaA->result_array() as $data) {
            $Alphaipk1=$data['ipk'];
            $Alphaijazah1=$data['ijazah'];
            $Alphawawancara1=$data['wawancara'];
            $Alphamicroteaching1=$data['microteaching'];
            $Alphatestulis1=$data['tesTulis'];
        }
    =====> mengambil nilai fuzzifikasi kiri <=====
    $alphaB= $this->db->query("SELECT * from tb_fuzzyfikasib where
iduser='$iduser'");
        foreach ($alphaB->result_array() as $data) {
            $Alphaipk2=$data['ipk'];
            $Alphaijazah2=$data['ijazah'];
            $Alphawawancara2=$data['wawancara'];
            $Alphamicroteaching2=$data['microteaching'];
            $Alphatestulis2=$data['tesTulis'];
        }
    =====> memprosesnya kedalam mesin inferensi/mencari nilai alpha <=====
    $alpha1=min($Alphaipk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis1);
    $alpha2=min($Alphaipk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis2);
    $alpha3=min($Alphaipk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis2);

```

---

---

```

$alpha4=min($AlphaiPk1,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
$alpha5=min($AlphaiPk1,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
$alpha6=min($AlphaiPk2,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis1);
$alpha7=min($AlphaiPk2,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis2);
$alpha8=min($AlphaiPk2,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis2);
$alpha9=min($AlphaiPk2,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
$alpha10=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);
$alpha11=min($AlphaiPk1,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis1);
$alpha12=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis1);
$alpha13=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis2);
$alpha14=min($AlphaiPk1,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis2);
$alpha15=min($AlphaiPk1,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching1,$Alphatestulis2);
$alpha16=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis2);
$alpha17=min($AlphaiPk1,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis2);
$alpha18=min($AlphaiPk2,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis2);
$alpha19=min($AlphaiPk1,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis2);
$alpha20=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis2);
$alpha21=min($AlphaiPk1,$Alphaijazah1,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis1);
$alpha22=min($AlphaiPk1,$Alphaijazah1,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis1);
$alpha23=min($AlphaiPk1,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis1);
$alpha24=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara1,$Alphamicroteaching2,$Alphatestulis1);
$alpha25=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching1,$Alphatestulis1);
$alpha26=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis1);
$alpha27=min($AlphaiPk1,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis1);
$alpha28=min($AlphaiPk2,$Alphaijazah2,$Alphawawancara2,$Alphamicroteaching2,$Alphatestulis2);

```

---

#### D.13. Kode program *function* hitungratarata(\$idUser) /mencari nilai z

##### Kode

---

```

=====> mengambil value table normalisasi nilai skala <=====
$nilaiideal=$this->db->query("select * from tb_nilaiidealskala");
    foreach ($nilaiideal->result_array() as $nilai) {
        $ipkideals=$nilai['ipk'];
        $ijazahideals=$nilai['ijazah'];
        $wawancaraideals=$nilai['wawancara'];
        $microteachingideals=$nilai['microteaching'];
        $tesTulisideals=$nilai['tesTulis'];
    }

=====> mencari nilai terkecil normalisasi nilai ideal <=====
$nilaimin=min($ipkideals,$ijazahideals,$wawancaraideals,$microteachingideals,$tesTulisideals);
    $gapmin=-1;
    $gapmax=1;

```

---

---

```

====> mencari nilai z minimal <=====
$Zmin = ((($gapmax-($gapmax-$nilaimin))+$gapmin)/2); //bobot minimal
====> mencari nilai z maksimal <=====
$Zmax = ((($gapmax-($gapmax-$nilaimin))+$gapmax)/2); //bobot maksimal

====> mencari nilai zn <=====
$z1= $Zmax-($alpha1*($Zmax-$Zmin));
$z2= $Zmax-($alpha2*($Zmax-$Zmin));
$z3= $Zmax-($alpha3*($Zmax-$Zmin));
$z4= $Zmax-($alpha4*($Zmax-$Zmin));
$z5= $Zmax-($alpha5*($Zmax-$Zmin));
$z6= $Zmax-($alpha6*($Zmax-$Zmin));
$z7= $Zmax-($alpha7*($Zmax-$Zmin));
$z8= $Zmax-($alpha8*($Zmax-$Zmin));
$z9= $Zmax-($alpha9*($Zmax-$Zmin));
$z10= $Zmax-($alpha10*($Zmax-$Zmin));
$z11= $Zmax-($alpha11*($Zmax-$Zmin));
$z12= $Zmax-($alpha12*($Zmax-$Zmin));
$z13= $Zmax-($alpha13*($Zmax-$Zmin));
$z14= $Zmax-($alpha14*($Zmax-$Zmin));
$z15= $Zmax-($alpha15*($Zmax-$Zmin));
$z16= $Zmax-($alpha16*($Zmax-$Zmin));
$z17= $Zmax-($alpha17*($Zmax-$Zmin));
$z18= $Zmax-($alpha18*($Zmax-$Zmin));
$z19= $Zmax-($alpha19*($Zmax-$Zmin));
$z20= $Zmax-($alpha20*($Zmax-$Zmin));
$z21= $Zmax-($alpha21*($Zmax-$Zmin));
$z22= $Zmax-($alpha22*($Zmax-$Zmin));
$z23= $Zmax-($alpha23*($Zmax-$Zmin));
$z24= $Zmax-($alpha24*($Zmax-$Zmin));
$z25= $Zmax-($alpha25*($Zmax-$Zmin));
$z26= $Zmax-($alpha26*($Zmax-$Zmin));
$z27= $Zmax-($alpha27*($Zmax-$Zmin));
$z28= $Zmax-($alpha28*($Zmax-$Zmin));
$alphasemua = array($alpha1, $alpha2, $alpha3,
$alpha4,$alpha5,$alpha6,$alpha7,$alpha8,$alpha9,$alpha10,$alpha11,$alpha12,$alph
a13,$alpha14,$alpha15,$alpha16,$alpha17,$alpha18,$alpha19,$alpha20,$alpha21,$al
pha22,$alpha23,$alpha24,$alpha25,$alpha26,$alpha27,$alpha28);
$zsemua =

```

---

---

```
array($z1,$z2,$z3,$z4,$z5,$z6,$z7,$z8,$z9,$z10,$z11,$z12,$z13,$z14,$z15,$z16,$z17,$z18,$z19,$z20,$z21,$z22,$z23,$z24,$z25,$z26,$z27,$z28);
```

```
====> menghitung jumlah seluruh nilai alpha <=====
```

```
$jumlahalpha=($alpha1+$alpha2+$alpha3+$alpha4+$alpha5+$alpha6+$alpha7+$alpha8+$alpha9+$alpha10+$alpha11+$alpha12+$alpha13+$alpha14+$alpha15+$alpha16+$alpha17+$alpha18+$alpha19+$alpha20+$alpha21+$alpha22+$alpha23+$alpha24+$alpha25+$alpha26+$alpha27+$alpha28);
```

```
====> menghitung jumlah seluruh nilai z <=====
```

```
$jumlahz=($z1*$alpha1)+($z2*$alpha2)+($z3*$alpha3)+($z4*$alpha4)+($z5*$alpha5)+($z6*$alpha6)+($z7*$alpha7)+($z8*$alpha8)+($z9*$alpha9)+($z10*$alpha10)+($z11*$alpha11)+($z12*$alpha12)+($z13*$alpha13)+($z14*$alpha14)+($z15*$alpha15)+($z16*$alpha16)+($z17*$alpha17)+($z18*$alpha18)+($z19*$alpha19)+($z20*$alpha20)+($z21*$alpha21)+($z22*$alpha22)+($z23*$alpha23)+($z24*$alpha24)+($z25*$alpha25)+($z26*$alpha26)+($z27*$alpha27)+($z28*$alpha28));
```

```
====> menghitung defuzzifikasi atau rata rata <=====
```

```
$zrata2=($jumlahz)/($jumlahalpha);
```

```
====> menyimpan hasil defuzzifikasi sebagai nilai akhirnya <=====
```

```
$query= $this->db->query("UPDATE tb_nilai SET nilaiAkhir='$zrata2' where iduser='$iduser'");
```

```
return array($alphasemua,$zsemua,$query);
```

```
}
```

---

LAMPIRAN E. PENGUJIAN *BLACK BOX*

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	<i>Login</i>	Mengisi kolom <i>username</i> dan <i>password</i> lalu klik tombol <i>login</i>	<i>Login</i> berhasil dan menampilkan halaman awal masing-masing aktor	[√] Berhasil [ ] Gagal
		Kolom <i>username</i> atau <i>password</i> tidak sesuai	Menampilkan pesan “ <i>username</i> atau <i>password</i> salah“	[√] Berhasil [ ] Gagal
		Salah satu kolom <i>username</i> atau <i>password</i> kosong	Menampilkan pesan “ <i>please fill out this field</i> ” dengan menunjuk pada kolom yang belum diisi	[√] Berhasil [ ] Gagal
2.	Mengelola data profil <i>user</i>	Klik menu ‘profile’ pada sidebar	Menampilkan halaman data profile	[√] Berhasil [ ] Gagal
		Klik tombol ubah profil dan mengubah data profile lalu klik tombol simpan	Menyimpan data profil dan menampilkan halaman data profile	[√] Berhasil [ ] Gagal
		Klik tombol ubah password dan mengubah password lalu klik tombol simpan	Menyimpan password baru dan menampilkan halaman data profile	[√] Berhasil [ ] Gagal
3.	Mengelola data	Mengisi nama yang ingin	menampilkan data akses yang namanya	[√] Berhasil

	akses	dicari dan klik tombol 'go'	sesuai dengan yang diisikan dikolom pencarian	<input type="checkbox"/> Gagal
		Memilih tombol 'tambah hak akses' dan mengisi data akses lalu klik submit	Menampilkan form tambah akses, menyimpan data akses yang diisikan pada form tambah	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih tombol 'edit', mengisi data akses halaman edit data akses dan memilih tombol submit	Menampilkkan form ubah data akses, menyimpan data akses yang diisikan pada form	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih tombol 'hapus'	Mengubah status dari data hak akses yang dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
4.	Mengelola nilai pendaftar	Memilih menu 'input nilai' pada <i>sidebar</i>	Menampilkan daftar nilai pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi <i>combobox</i> pelajaran dan jenjang lalu pilih tombol 'Go!'	Menampilkan daftar nilai pendaftar sesuai mata pelajaran dan jenjang yang telah dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		memilih <i>icon</i> pensil/ubah, mengisi data nilai, dan	Menampilkan halamn form data nilai, menyimpan nilai yang diisikan.	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

		memilih ' <i>submit button</i> '		
5.	Menampilkan data profile pendaftar	Memilih menu 'Data Pendaftar'	Menampilkan halaman daftar pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi <i>combobox</i> pelajaran dan jenjang lalu pilih tombol 'Go!'	Menampilkan daftar pendaftar sesuai mata pelajaran dan jenjang yang telah dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih <i>icon</i> detail	Menampilkan halaman deail profile pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6.	Menampilkan data nilai pendaftar	Memilih menu 'Data Pendaftar'	Menampilkan halaman daftar pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi <i>combobox</i> pelajaran dan jenjang lalu pilih tombol 'Go!'	Menampilkan daftar pendaftar sesuai mata pelajaran dan jenjang yang telah dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih <i>icon</i> detail	Menampilkan halaman detail profile pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih sub menu 'detail nilai' pada halaman detail profile	Menampilkan halaman detail nilai pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Memilih tombol 'Show detail' pada halaman detail	Menampilkan detail nilai dan detail nilai per tahap perhitungan menggunakan metode	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

	nilai pendaftar	<i>Profile Matching</i> dan <i>fuzzy logic</i>	
7. Mengelola data pelajaran	Memilih menu 'update data pelajaran' pada <i>sidebar</i>	Menampilkan halaman data pelajaran	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Memilih tombol 'tambah baru', mengisi data mata pelajaran, dan memilih 'submit button'	Menampilkan form tambah data pelajaran, menyimpan data pelajaran yang diisikan pada form	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Memilih tombol 'edit', , mengisi data mata pelajaran, dan memilih 'submit button'	Menampilkan form edit data pelajaran, menyimpan data pelajaran yang diisikan pada form	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Memilih tombol 'hapus	Menghapus data mata pelajaran yang dipilih dari daftar mata pelajaran	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8. Mengelola kuota lowongan	Memilih menu 'penawaran dan kuota'	Menampilkan data lowongan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mengisi <i>combobox</i> pelajaran dan jenjang lalu pilih tombol 'Go!'	Menampilkan data lowongan sesuai mata pelajaran dan jenjang yang telah dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Memilih tombol 'tambah baru', mengisi data lowongan, dan memilih	Menampilkan halaman form tambah data lowongan baru, menyimpan data yang diisikan pada form	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

<i>'submit button'</i>			
	Memilih tombol 'buka'	Mengubah status lowongan menjadi ada	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Memilih tombol 'tutup'	Mengubah status lowongan menjadi tidak ada	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Memilih tombol 'ubah, mengisi data lowongan, dan memilih <i>'submit button'</i>	Menampilkan halaman form edit data lowongan baru, menyimpan data yang diisikan pada form	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Kolom kosong tidak diisi	Menampilkan pesan "please fill out this field" dengan menunjuk pada kolom yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
9.	Mengelola data status pendaftaran para pendaftar	Memilih menu 'reset daftar pendaftar'	Menampilkan halaman data pendaftar <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi <i>combobox</i> pelajaran dan jenjang lalu pilih tombol 'Go!'	Menampilkan data pendaftar sesuai mata pelajaran dan jenjang yang telah dipilih <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		memilih tombol 'hapus'	Mengubah data pendaftar menjadi pendaftar tereliminasi <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

	Memilih tombol ‘lihat semua’	Menampilkan halaman data pendaftar secara keseluruhan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
10.	Menampilkan data pendaftar	Memilih menu ‘Data Pendaftar’	Menampilkan halaman daftar pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mengisi <i>combobox</i> pelajaran dan jenjang lalu pilih tombol ‘Go!’	Menampilkan daftar pendaftar sesuai mata pelajaran dan jenjang yang telah dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	Memilih <i>icon</i> detail	Menampilkan halaman detail profile pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	Memilih sub menu ‘detail nilai’ pada halaman detail profile	Menampilkan halaman detail nilai pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
11.	<i>Update</i> nilai wawancara dan microteaching	Memilih menu ‘input nilai’	Menampilkan halaman daftar nilai pendaftar	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	Mengisi <i>combobox</i> pelajaran dan jenjang lalu pilih tombol ‘Go!’	Menampilkan daftar nilai pendaftar sesuai mata pelajaran dan jenjang yang telah dipilih	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	Memilih <i>icon</i> edit, mengisi data nilai wawancara dan	Menampilkan form nilai wawancara dan nilai <i>microteaching</i> , menyimpan data nilai yang di	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	

		<i>microteaching</i> , dan memilih isikan di form sesuai id pendaftaranya 'submit button'	
12.	Mengelola nilai ideal kriteria	Memilih menu 'nilai ideal'	Menampilkan haman data nilai ideal kriteria [√] Berhasil [ ] Gagal
		Memilih tombol 'edit', mengisi nilai ideal, dan mimilih 'submit button'	Menampilkan halaman form edit data nilai ideal kriteria, menyimpan data yang diisikan pada form, menghitung sekaligus menyimpan nilai normalisasinya [√] Berhasil [ ] Gagal
13.	Menyetujui hasil penerimaan	Memilih menu 'penerimaan' pada <i>sidebar</i>	Menampilkan daftar pendaftar [√] Berhasil [ ] Gagal
		Mengisi <i>combobox</i> pelajaran dan jenjang lalu pilih tombol 'Go!'	Menampilkan daftar pendaftar sesuai mata pelajaran dan jenjang yang telah dipilih [√] Berhasil [ ] Gagal
		Memilih tombol 'calon'	Mengubah status terima pendaftar yang dimaksud menjadi 'calon' [√] Berhasil [ ] Gagal
		Memilih tombol 'terima	Mengubah status terima pendaftar yang dimaksud menjadi 'diterima [√] Berhasil [ ] Gagal
14.	Melakukan pendaftaran	Memilih tombol 'daftar' pada halaman awal, mengisi data pendaftar, dan memilih	Menampilkan halaman form pendaftaran, menyimpan data pendaftar [√] Berhasil [ ] Gagal

---

	<i>'submit button'</i>		
	Kolom kosong tidak diisi	Menampilkan pesan “please fill out this field” dengan menunjuk pada kolom yang belum diisi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
15.	<i>Logout</i>	Klik tombol <i>logout</i>	Berhasil keluar sistem dan menampilkan halaman awal <input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

---

