



**IMPLEMENTASI METODE *WEIGHTED PRODUCT* PADA SISTEM  
INFORMASI SELEKSI PENERIMAAN ANGGOTA PSM (PADUAN  
SUARA MAHASISWA) UNIVERSITAS JEMBER**

**SKRIPSI**

Oleh

**Rozha Aulya Nurvianti**

**NIM 142410101037**

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

**2018**



**IMPLEMENTASI METODE *WEIGHTED PRODUCT* PADA SISTEM  
INFORMASI SELEKSI PENERIMAAN ANGGOTA PSM (PADUAN  
SUARA MAHASISWA) UNIVERSITAS JEMBER**

**SKRIPSI**

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

**HALAMAN JUDUL**

Oleh

**Rozha Aulya Nurvianti**

**NIM 142410101037**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER  
2018**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga telah memberikan banyak kemudahan dan kelancaran dalam mengerjakan skripsi;
2. Ibunda Siti Faiqotul Himmah S.Pd dan Ayahanda Drs. A.Sadeli atas ketulusan doa, cinta dan kasih sayang serta pengorbanan selama ini;
3. Saudara saya dr. Reza Kurniawan dan Mohammad Rafli Akbar;
4. Dosen-dosenku selama berada di Fakultas Ilmu Komputer
5. Guru – guruku sejak sekolah dasar hingga perguruan tinggi, yang telah memberikan ilmu dan membimbing dengan penuh kesabaran;
6. Sahabat – sahabat yang selalu mendukung dan menemani dalam masa sulit dan senang;
7. Almamater Program Studi Sistem Informasi Universitas Jember;

**MOTO**

“Jadilah seperti semut dalam ketekunannya”



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Rozha Aulya Nurvianti

NIM : 142410101037

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi Metode *Weighted Product* Pada Sistem Informasi Seleksi Penerimaan Anggota PSM (Paduan Suara Mahasiswa) Universitas Jember” adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 29 Juni 2018

Yang menyatakan,

Rozha Aulya N

NIM 142410101037

**SKRIPSI**

**IMPLEMENTASI METODE *WEIGHTED PRODUCT* PADA SISTEM  
INFORMASI SELEKSI PENERIMAAN ANGGOTA PSM (PADUAN  
SUARA MAHASISWA) UNIVERSITAS JEMBER**

Oleh :

Rozha Aulya Nurvianti

NIM 142410101037

Pembimbing

Dosen Pembimbing Utama : Drs. Antonius Cahya P, M.App., Sc., Ph.D

Dosen Pembimbing Pendamping : Gama Wisnu F, S.Kom., M.Kom

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Implementasi Metode *Weighted Product* Pada Sistem Informasi Seleksi Penerimaan Anggota PSM (Paduan Suara Mahasiswa) Universitas Jember”, telah diuji dan disahkan pada:

hari, tanggal : 29 Juni 2018

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

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

Gama Wisnu F,S.Kom, M.Kom  
NRP. 760015717

**PENGESAHAN PENGUJI**

Skripsi berjudul “Implementasi Metode *Weighted Product* Pada Sistem Informasi Seleksi Penerimaan Anggota PSM (Paduan Suara Mahasiswa) Universitas Jember”, telah diuji dan disahkan pada:

hari, tanggal : 29 Juni 2018

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji:

Penguji I,

Penguji II,

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

Nova El Maidah, S.Si, M.Cs  
NIP. 198411012015042001

Mengesahkan  
Penjabat Dekan Fakultas Ilmu Komputer

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

## RINGKASAN

**Implementasi Metode *Weighted Product* Pada Sistem Informasi Seleksi Penerimaan Anggota PSM (Paduan Suara Mahasiswa) Universitas Jember;** Rozha Aulya N, 142410101037; 2018, 173 halaman; Program Studi Sistem Informasi Universitas Jember.

Universitas Jember merupakan salah satu Universitas Negeri di Kabupaten Jember yang dikenal memiliki berbagai prestasi akademik dan non-akademik mahasiswa dari berbagai jurusan yang ada di Universitas Jember. Salah satu prestasi non-akademik yang pernah diraih Universitas Jember yaitu penampilan terbaik paduan suara Universitas Jember dalam kompetisi Internasional “5th Gran Prix” Pattaya Thailand pada tahun 2012. Untuk menjadi anggota paduan suara mahasiswa Universitas Jember, calon anggota baru harus mengikuti beberapa tes yang telah ditentukan pada prosedur penerimaan.

Dalam memilih anggota baru paduan suara mahasiswa Universitas Jember harus dilakukan dengan tepat, karena hal ini dapat mempengaruhi prestasi non-akademik Universitas Jember khususnya bidang Paduan Suara Mahasiswa (PSM). Dalam melakukan penerimaan anggota paduan suara Universitas Jember, Tim paduan suara Universitas Jember masih menggunakan sistem yang manual. Sistem yang manual memiliki kelemahan yaitu tingginya tingkat kesalahan, kurang efisien waktu dan biaya. Oleh karena itu dibutuhkan sebuah sistem informasi untuk menangani permasalahan di atas.

Sistem informasi yang akan dibuat mampu memberikan pendukung keputusan terhadap pemilihan calon anggota paduan suara Universitas Jember dengan cara pelatih menginputkan nilai hasil tes ke dalam sistem kemudian diolah dan akan menghasilkan *output* nilai anggota paduan suara berupa perangkingan. Hasil perangkingan diambil 30 nilai terbaik. Dalam proses pengambilan keputusan sistem informasi ini mengimplementasikan metode *Weighted Product*. Dengan demikian sistem informasi yang dibuat dapat membantu dalam proses seleksi penerimaan anggota paduan suara Universitas Jember.

## PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya , sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi Metode *Weighted Product* Pada Sistem Informasi Seleksi Penerimaan Anggota Paduan Suara Mahasiswa Universitas Jember (Studi Kasus: Universitas Jember)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

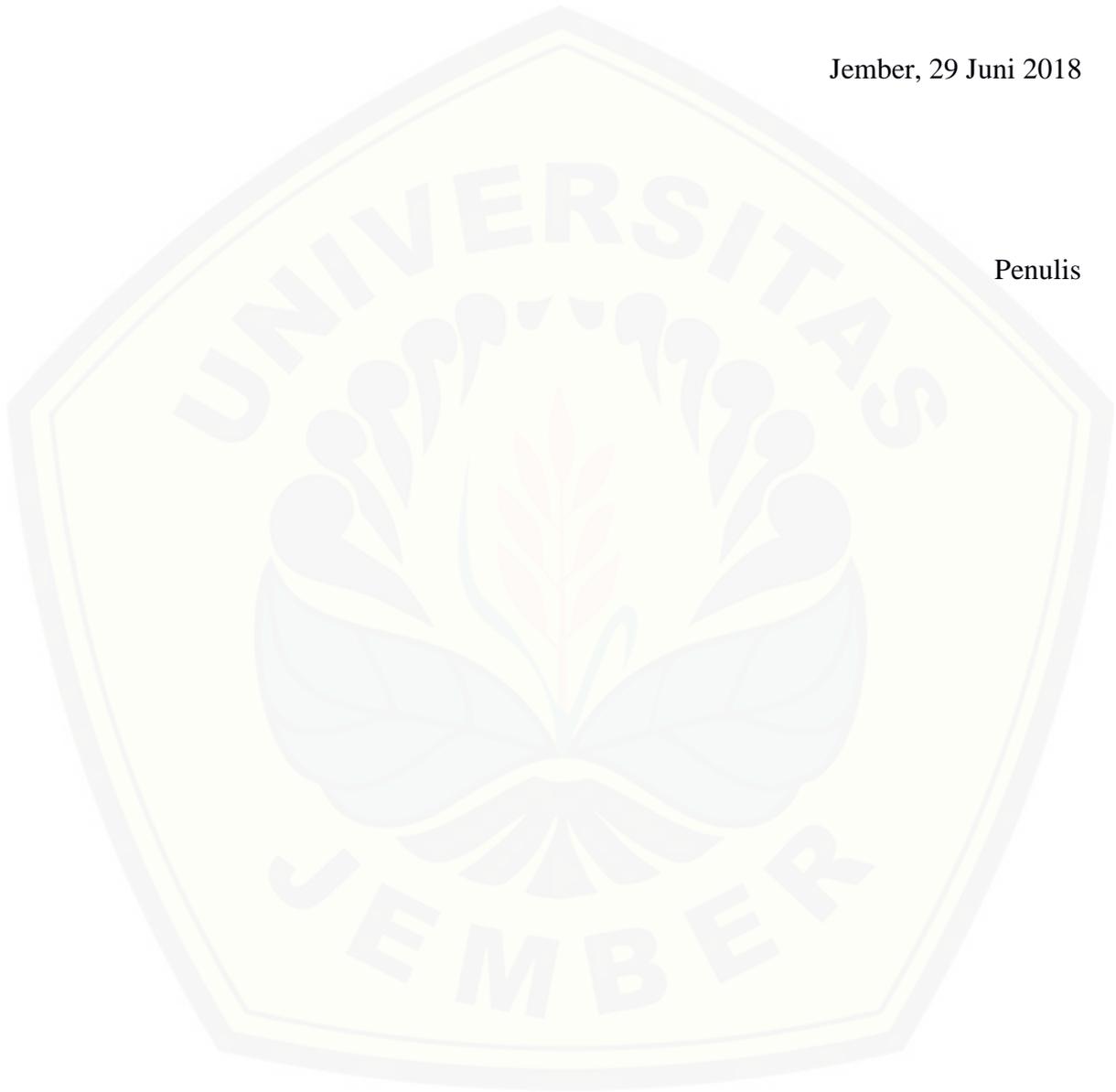
Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Drs. Antonius Cahya P, M.App., Sc., Ph.D. selaku Dosen Pembimbing Utama dan Gama Wisnu Fajarianto, S.Kom., M.Kom selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, tenaga, pikiran, dan perhatian dalam penulisan skripsi;
2. Ibunda Siti Faiqotul Himmah, S. Pd dan Drs. A.Sadeli tercinta, atas dukungan moral, do'a, semangat, nasehat serta kasih sayang yang tiada terhenti dalam setiap perjalanan kehidupan saya;
3. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., dan Rina Hayati, S.Pd sekeluarga, atas doa, bimbingan, dan nasihat selama saya berada di Jember seperti layaknya anak sendiri;
4. Saudaraku, dr. Reza Kurniawan dan Mohammad Rafli Akbar yang telah menjadi inspirasi dan motivasi serta keceriaan dan semangat yang diberikan agar terus berusaha menjadi lebih baik;
5. Sahabat – sahabat yang telah menemani, membantu, mendukung dan memberikan semangat serta doanya yaitu Ino, Ilvi, Laras, Tika, Fajar, Rifqy, Angel, Afif dan Rizal;
6. Teman-teman seperjuangan SENSATION angkatan 2014 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
7. UKM Paduan Suara Universitas Jember;
8. Semua pihak yang tidak dapat disebutkan satu-persatu.

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

Jember, 29 Juni 2018

Penulis



**DAFTAR ISI**

HALAMAN JUDUL.....	i
PERSEMBAHAN .....	ii
MOTO .....	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN .....	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xviii
DAFTAR TABEL.....	xix
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan .....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan .....	4
BAB 2 TINJAUAN PUSTAKA .....	7
2.1 Metode Weighted Product (WP).....	7
2.2 Pengertian Paduan Suara.....	8
2.3 Manajemen Sistem Informasi .....	9
BAB 3 METODOLOGI PENELITIAN.....	10
3.1 Jenis Penelitian.....	10
3.2 Tempat dan Waktu Penelitian .....	10
3.3 Teknik Pengembangan Sistem .....	10
3.3.1 Analisis Kebutuhan.....	11
3.3.2 Desain .....	12

3.3.3 Implementasi.....	13
3.3.4 <i>Testing</i> .....	14
3.3.5 Pemeliharaan.....	14
3.4 Analisis Metode .....	14
3.5 Gambaran Umum Sistem .....	15
<b>BAB 4 PENGEMBANGAN SISTEM</b> .....	<b>17</b>
4.1 Deskripsi Umum Sistem Informasi.....	17
4.2 Analisis kebutuhan sistem.....	17
4.2.1 Hasil Wawancara .....	18
4.2.2 Kebutuhan Fungsional .....	18
4.2.3 Kebutuhan Non-fungsional.....	19
4.3 Desain Sistem Informasi .....	19
4.3.1 Komponen <i>Input Output</i> .....	19
4.3.2 <i>Usecase Diagram</i> .....	20
4.3.3 <i>Use Case Scenario</i> .....	24
4.3.4 <i>Sequence Diagram</i> .....	32
4.3.5 <i>Activity Diagram</i> .....	41
4.3.6 <i>Class diagram</i> .....	49
4.3.7 <i>Entity Relationship Diagram (ERD)</i> .....	49
4.4 Implementasi .....	49
4.5 Pengujian sistem .....	58
<b>BAB 5 HASIL DAN PEMBAHASAN</b> .....	<b>64</b>
5.1 Hasil Perhitungan <i>Weighted Product</i> .....	64
5.2 Hasil Pembahasan Metode <i>Weighted Product</i> pada Sistem Informasi Seleksi Penerimaan Calon Anggota Paduan Mahasiswa Suara Universitas Jember .....	74
5.2.1 Kelebihan Sistem .....	75
5.2.2 Kelemahan Sistem.....	76

5.3 Hasil Implementasi <i>Coding</i> Pada Sistem Informasi Seleksi Penerimaan Calon Anggota Paduan Suara Mahasiswa Universitas Jember .....	76
5.3.1 Tampilan <i>Login</i> .....	76
5.3.2 Tampilan Menambahkan Data Event .....	77
5.3.3 Tampilan Melihat Data Event.....	78
5.3.4 Tampilan Mengubah Data Event .....	78
5.3.5 Tampilan Menambah Galeri Kegiatan.....	78
5.3.6 Tampilan Melihat Galeri Kegiatan .....	79
5.3.7 Tampilan Mengubah Galeri Kegiatan.....	79
5.3.8 Tampilan Pendaftaran .....	79
5.3.9 Tampilan Melihat Data Lolos Tahun Sebelumnya .....	79
5.3.10 Tampilan Lihat Data Pendaftar.....	80
5.3.11 Tampilan Input Nilai Hasil Seleksi.....	80
5.3.12 Tampilan Menambahkan Data Penerimaan.....	81
5.3.13 Tampilan Melihat Data Penerimaan .....	81
5.3.14 Tampilan Mengubah Data Penerimaan.....	81
5.3.15 Tampilan Melihat Vektor S .....	81
5.3.16 Tampilan Melihat Vektor V.....	82
5.3.17 Tampilan Melihat Hasil Perangkingan .....	82
5.3.18 Tampilan Persetujuan Pendaftar Lolos .....	83
5.3.19 Tampilan Cetak Pendaftar Lolos Seleksi.....	83
5.3.20 Tampilan Help .....	83
<b>BAB 6 PENUTUP .....</b>	<b>84</b>
6.1 Kesimpulan .....	84
6.2 Saran.....	85
<b>DAFTAR PUSTAKA .....</b>	<b>86</b>

LAMPIRAN .....	87
LAMPIRAN A. SKENARIO .....	87
1. <i>Use Case Scenario</i> Login .....	88
2. <i>Use Case Scenario</i> Menambah Data Event .....	89
3. <i>Use Case Scenario</i> Melihat Data Event.....	90
4. <i>Use Case Scenario</i> Mengubah Data Event.....	91
5. <i>Use Case Scenario</i> Menambah Galeri Kegiatan .....	92
6. <i>Use Case Scenario</i> Melihat Galeri Kegiatan .....	93
7. <i>Use Case Scenario</i> Mengubah Galeri Kegiatan .....	93
8. <i>Use Case Scenario</i> Pendaftaran .....	94
9. <i>Use Case Scenario</i> Melihat Data Lolos Tahun Sebelumnya.....	95
10. <i>Use Case Scenario</i> Lihat Data Pendaftar.....	96
11. <i>Use Case Scenario</i> Input Nilai Hasil Seleksi.....	97
12. <i>Use Case Scenario</i> Menambah Data Penerimaan.....	98
13. <i>Use Case Scenario</i> Melihat Data Penerimaan .....	100
14. <i>Use Case Scenario</i> Mengubah Data Penerimaan .....	101
15. <i>Use Case Scenario</i> Melihat Hasil Perangkingan .....	102
16. <i>Use Case Scenario</i> Cetak Pendaftar Disetujui.....	102
17. <i>Use Case Scenario</i> Persetujuan Pendaftar Lolos Seleksi .....	103
18. <i>Use Case Scenario</i> Help .....	104
19. <i>Use Case Scenario</i> Logout .....	105
LAMPIRAN B. SEQUENCE DIAGRAM .....	106
1. <i>Sequence Diagram</i> Masuk .....	106
2. <i>Sequence Diagram</i> Menambah Data Event .....	107
3. <i>Sequence Diagram</i> Melihat Data Event.....	107
4. <i>Sequence Diagram</i> Mengubah Data Event.....	108

5. <i>Sequence Diagram</i> Menambah Galeri Kegiatan .....	109
6. <i>Sequence Diagram</i> Melihat Galeri Kegiatan .....	109
7. <i>Sequence Diagram</i> Mengubah Galeri Kegiatan .....	110
8. <i>Sequence Diagram</i> Pendaftaran.....	110
9. <i>Sequence Diagram</i> Melihat Data Lolos Tahun Sebelumnya.....	111
10. <i>Sequence Diagram</i> Lihat Data Pendaftar.....	112
11. <i>Sequence Diagram</i> Input Nilai Hasil Seleksi .....	112
12. <i>Sequence Diagram</i> Menambah Data Penerimaan .....	113
13. <i>Sequence Diagram</i> Melihat Data Penerimaan .....	113
14. <i>Sequence Diagram</i> Mengubah Data Penerimaan .....	114
15. <i>Sequence Diagram</i> Lihat Hasil Perangkingan .....	114
16. <i>Sequence Diagram</i> Persetujuan Pendaftar Lolos Seleksi .....	115
17. <i>Sequence Diagram</i> Cetak Pendaftar Disetujui.....	116
18. <i>Sequence Diagram</i> Help .....	116
19. <i>Sequence Diagram</i> Logout .....	117
LAMPIRAN C. ACTIVITY DIAGRAM .....	118
1. <i>Activity Diagram</i> Masuk.....	118
2. <i>Activity Diagram</i> Menambah Data Event.....	119
3. <i>Activity Diagram</i> Melihat Data Event .....	120
4. <i>Activity Diagram</i> Mengubah Data Event.....	121
5. <i>Activity Diagram</i> Menambah Galeri Kegiatan .....	122
6. <i>Activity Diagram</i> Melihat Galeri Kegiatan.....	123
7. <i>Activity Diagram</i> Mengubah Galeri Kegiatan .....	124
8. <i>Activity Diagram</i> Pendaftaran.....	125
9. <i>Activity Diagram</i> Melihat Data Lolos Tahun Sebelumnya.....	125

10. <i>Activity Diagram</i> Lihat Data Pendaftar .....	126
11. <i>Activity Diagram</i> Input Nilai Hasil Seleksi .....	127
12. <i>Activity Diagram</i> Menambah Data Penerimaan .....	128
13. <i>Activity Diagram</i> Melihat Data Penerimaan .....	128
14. <i>Activity Diagram</i> Mengubah Data Penerimaan .....	129
15. <i>Activity Diagram</i> Melihat Hasil Perangkingan .....	130
16. <i>Activity Diagram</i> Persetujuan Pendaftar Lolos Seleksi .....	130
17. <i>Activity Diagram</i> Cetak Pendaftar Disetujui .....	131
18. <i>Activity Diagram</i> Help .....	131
LAMPIRAN D. TRANSKRIP WAWANCARA .....	132
LAMPIRAN E. TAMPILAN SISTEM .....	134
1. Tampilan Melihat Data Event .....	134
2. Tampilan Mengubah Data Event .....	134
3. Tampilan Menambah Galeri Kegiatan .....	135
4. Tampilan Melihat Galeri Kegiatan .....	135
5. Tampilan Mengubah Galeri Kegiatan .....	136
6. Tampilan Pendaftaran .....	136
7. Tampilan Melihat Data Lolos Tahun Sebelumnya .....	137
8. Tampilan Input Nilai Hasil Seleksi .....	137
9. Tampilan Menambahkan Data Penerimaan .....	138
10. Tampilan Melihat Data Penerimaan .....	138
11. Tampilan Mengubah Data Penerimaan .....	139
12. Tampilan Melihat Vektor V .....	139
13. Tampilan Melihat Hasil Perangkingan .....	140
14. Tampilan Persetujuan Pendaftar Lolos .....	140

15. Tampilan Cetak Pendaftar Lolos Seleksi.....	141
16. Tampilan Help .....	141



**DAFTAR GAMBAR**

Gambar 2. 1 Tahapan Metode Weighted Product.....	8
Gambar 3. 1 Alur SDLC Waterfall .....	11
Gambar 3. 2 Diagram Alir Penerapan Metode Weighted Product (WP).....	13
Gambar 4. 1 Komponen input output.....	20
Gambar 4. 2 Use Case Diagram.....	21
Gambar 4. 3 Sequence Diagram melihat vektor S .....	38
Gambar 4. 4 Sequence Diagram melihat vektor V .....	39
Gambar 4. 5 Activity Diagram melihat vektor S .....	46
Gambar 4. 6 Activity Diagram melihat vektor V.....	47
Gambar 4. 7 Class Diagram .....	50
Gambar 4. 8 Entity Relationship Diagram (ERD) .....	51
Gambar 5. 1 Data Penerimaan Tahun 2017 .....	72
Gambar 5. 2 Hasil Perhitungan Vektor S.....	73
Gambar 5. 3 Hasil Perhitungan Vektor V .....	73
Gambar 5. 4 Hasil Perhitungan Vektor V .....	74
Gambar 5. 5 Hasil perangkaan calon anggota paduan suara .....	75
Gambar 5. 6 Tampilan Login .....	77
Gambar 5. 7 Tampilan Tambah Data Event.....	77
Gambar 5. 8 Tampilan Lihat Data Pendaftar .....	80
Gambar 5. 9 Tampilan Melihat Vektor S.....	82

**DAFTAR TABEL**

Tabel 4. 1 Definisi Aktor .....	21
Tabel 4. 2 Definisi Use Case.....	23
Tabel 4. 3 Skenario Melihat Perhitungan Vektor S .....	29
Tabel 4. 4 Skenario Melihat Perhitungan Vektor V.....	30
Tabel 4.5 Kode Program Kelas c_penerimaan.....	52
Tabel 4. 6 Kode program kelas m_mahasiswa.....	53
Tabel 4. 7 Kode program kelas m_penerimaan .....	53
Tabel 4. 8 Kode program kelas m_vektor_v.....	54
Tabel 4. 9 Kode program kelas m_vektor_v_akhir.....	55
Tabel 4. 10 query dari ambilidakhir.....	56
Tabel 4. 11 query dari view vektor_s.....	56
Tabel 4. 12 query dari view total_vektor_s.....	57
Tabel 4. 13 query dari view abstrak_total_semua_vektor_s.....	57
Tabel 4. 14 query dari view vektor_v .....	58
Tabel 4. 15 query dari view vektor_v_akhir .....	58
Tabel 4. 16 Hasil Pengujian Black Box .....	59
Tabel 5. 1 Data Pendaftar Tahun 2017.....	65
Tabel 5. 2 Pembobotan Berdasarkan Tingat Kepentingan.....	66
Tabel 5. 3 Kriteria .....	66
Tabel 5. 4 Penilaian Calon Anggota Paduan Suara Universitas Jember.....	67
Tabel 5. 5 Nilai Perbaikan Bobot.....	68
Tabel 5. 6 Hasil Nilai Prefensi Vektor S.....	69
Tabel 5. 7 Hasil Nilai Prefensi Vektor V .....	70
Tabel 5. 8 Hasil Perhitungan Prefensi Akhir dan Perangkingan.....	71

## BAB 1 PENDAHULUAN

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

### 1.1 Latar Belakang

Universitas Jember merupakan salah satu universitas negeri di Kabupaten Jember yang dikenal memiliki berbagai prestasi akademik dan non-akademik mahasiswa dari berbagai jurusan yang ada di Universitas Jember. Salah satu prestasi non-akademik yang pernah diraih Universitas Jember yaitu penampilan terbaik paduan suara Universitas Jember dalam kompetisi Internasional “5<sup>th</sup> Gran Prix” Pattaya Thailand pada tahun 2012 serta turut berpartisipasi dalam kompetisi Internasional di negara Korea Selatan pada tahun 2013. Paduan suara mahasiswa Universitas Jember dapat merekrut mahasiswa yang berminat dan memiliki talenta di bidang tarik suara. Para anggota berpartisipasi karena ingin meningkatkan pengetahuan di bidang musik khususnya di bidang tarik suara.

Divisi paduan suara Universitas Jember biasanya menampung anggota baru dengan total 30 mahasiswa Universitas Jember dari berbagai fakultas. Untuk mendapatkan dan mempertahankan prestasi-prestasi yang telah diraih, Tim paduan suara Universitas Jember tidak hanya menekankan pada latihan rutin saja, tetapi juga mengutamakan pemilihan sumber daya manusia yang tepat untuk tetap menjaga dan meningkatkan kualitasnya. Seleksi penerimaan anggota paduan suara Universitas Jember masih menggunakan prosedur yang manual. Hal tersebut dapat menyebabkan terjadinya kesalahan, kurangnya efisiensi waktu dan biaya. Salah satu cara untuk menggantikan prosedur seleksi yang manual adalah dengan menerapkan sistem seleksi penerimaan, sedangkan untuk mengurangi kesalahan pada proses seleksi penerimaan perhitungan manual dapat digantikan dengan perhitungan menggunakan metode perangkangan. Salah satu metode perangkangan yang dapat digunakan adalah metode *weighted product*.

Metode *weighted product* sebelumnya pernah dilakukan oleh (Widianto & Yulianto, 2014) dengan judul “Sistem Informasi Pendaftaran Peserta Didik Baru Pada Sekolah Menengah Kejuruan (SMK) Negeri 2 Pacitan Berbasis Web Dinamis” telah menerapkan sistem informasi untuk pendaftaran peserta didik baru. Dasar penelitian tersebut dikarenakan SMK Negeri 2 Pacitan masih menggunakan sistem konvensional. Hasil dari penelitian tersebut ialah sistem pendaftaran dengan menggunakan sistem yang terkomputerisasi dapat mempermudah panitia pendaftaran serta transparansi update data lebih terjamin sehingga mengurangi tingkat kecurangan yang mungkin terjadi.

Penelitian lain yang dilakukan oleh Niswatin yang berjudul “Sistem Seleksi Penerimaan Mahasiswa Baru Menggunakan Metode *Weighted Product* (WP)”. Penelitian tersebut menggunakan Universitas Nusantara PGRI Kediri sebagai studi kasus. Universitas Nusantara PGRI melakukan penerimaan mahasiswa baru secara manual. Hal tersebut dijadikan landasan peneliti untuk membangun sistem pendukung keputusan penerimaan mahasiswa baru Universitas Nusantara PGRI Kediri menggunakan metode *Weighted Product*. Penelitian ini menghasilkan sebuah sistem pendukung keputusan yang dapat membantu proses seleksi mahasiswa baru di Universitas Nusantara PGRI Kediri menggunakan metode *weighted product*. Kriteria yang digunakan metode *weighted product* untuk proses seleksi yaitu nilai uan matematika, nilai uan bahasa indonesia, nilai uan bahasa inggris, nilai tes matematika, nilai tes bahasa indonesia, nilai tes bahasa inggris, dan nilai psikotes. Sehingga hasil dari sistem tersebut dapat membuat proses penerimaan mahasiswa baru menjadi lebih mudah, cepat, dan tepat. (Niswatin, 2016)

Berdasarkan pemaparan di atas, peneliti membangun sebuah sistem seleksi penerimaan calon anggota paduan suara Universitas Jember dengan menggunakan metode *Weighted Product* dengan tujuan untuk membantu memberikan keputusan pada proses seleksi penerimaan calon anggota paduan suara. Metode ini dipilih karena metode *weighted product* memiliki tingkat akurasi yang tinggi karena memanfaatkan perkalian rating kinerja yang dipangkatkan dengan bobot. Metode *weighted product* merupakan metode penunjang keputusan yang tujuan akhirnya

berupa perankingan. Selain itu pemilihan metode *weighted product* berdasarkan dari kebutuhan atau keperluan dari tujuan penelitian. Kebutuhan dari penelitian ini yaitu untuk menyeleksi anggota paduan suara Universitas Jember berdasarkan 30 nilai terbaik yang diperoleh dari hasil perankingan. Sehingga dapat disimpulkan bahwa metode *weighted product* dapat diterapkan dalam penelitian.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat diambil kesimpulan bahwa permasalahan yang muncul adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode *Weighted Product* (WP) dalam seleksi penerimaan anggota paduan suara Universitas Jember?
2. Bagaimana membangun sistem informasi seleksi penerimaan anggota paduan suara Universitas Jember dengan menggunakan metode *Weighted Product* (WP)?

## 1.3 Tujuan

Tujuan dari penelitian ini adalah:

1. Membangun sebuah sistem informasi seleksi penerimaan anggota paduan suara mahasiswa Universitas Jember dengan menggunakan metode *Weighted Product*.
2. Mengimplementasikan metode *Weighted Product* pemilihan anggota paduan suara mahasiswa Universitas Jember.

## 1.4 Batasan Masalah

Dalam pembuatan sistem informasi seleksi penerimaan anggota paduan suara Universitas Jember, terdapat beberapa batasan antaralain:

1. Studi kasus pada Universitas Jember.
2. Sistem digunakan oleh Universitas Jember untuk menentukan penerimaan anggota paduan suara Universitas Jember.
3. Penelitian dilakukan dengan asumsi bahwa kriteria yang digunakan akan tetap tiap tahun (setiap penerimaan).

4. Sistem menerapkan metode *Weighted Product*
5. Sistem yang dibuat berbasis *web*.

### 1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

#### 1. Pendahuluan

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

#### 2. Tinjauan Pustaka

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

#### 3. Metodologi Penelitian

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

#### 4. Pengembangan Sistem

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

#### 5. Hasil dan Pembahasan

Bab ini menjelaskan mengenai hasil penelitian yang telah dilakukan serta pembahasan sistem yang telah dibuat. Pembahasan dilakukan guna menjelaskan dan memaparkan bagaimana penelitian ini menjawab perumusan masalah serta

tujuan dan manfaat dari penelitian ini seperti apa yang telah ditentukan pada awal penelitian.

#### 6. Penutup

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



## BAB 2 TINJAUAN PUSTAKA

Bab ini menjelaskan tentang tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku, literature, jurnal, dan internet

### 2.1 Metode Weighted Product (WP)

*Multi Attribute Decision Making* (MADM) adalah metode yang digunakan untuk mencari alternatif paling optimal dari sejumlah alternatif optimal dengan kriteria tertentu. Inti MADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses menyusun alternatif sesuai dengan nilai yang telah ada. *Weighted Product* merupakan salah satu yang digunakan untuk menyelesaikan masalah MADM. *Weighted Product* adalah suatu metode yang menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot yang bersangkutan. Proses ini sama halnya dengan proses normalisasi (Sianturi, 2013).

Penelitian dilakukan oleh (Syafitri, Sutardi, & Dewi, 2016) yang berjudul “Penerapan Metode *Weighted Product* Dalam Sistem Pendukung Keputusan Pemilihan Laptop Berbasis Web” menjelaskan ada delapan langkah-langkah untuk menghitung metode *Weighted Product*. Adapun langkah-langkah metode *Weighted Product* sebagai berikut:

1. Menentukan jenis-jenis kriteria yang akan dijadikan acuan dalam penilaian pendukung keputusan.
2. Menentukan bobot setiap alternatif pada setiap kriteria dinilai dengan 1 sampai 5, gradasi pembobotan mengacu pada Skala Likert yaitu:
  - 1= Sangat Tidak Penting
  - 2= Tidak Penting
  - 3= Ragu-ragu

4= Penting

5= Sangat Penting

3. Selanjutnya dilakukan perbaikan bobot yang telah dimasukkan oleh *user* sehingga total bobot harus sama dengan 1. Bobot yang diberikan pada setiap nilai kriteria ditentukan berdasarkan tingkat kepentingan setiap kriteria. Perbaikan bobot dilakukan dengan rumus sebagai berikut

$$W_j = \frac{w_0}{\sum w_0} \quad (2.1)$$

Keterangan:

$W_0$  = Nilai perbaikan bobot *user*

$W_j$  = Nilai bobot dari *user*

$\sum W_0$  = Jumlah nilai bobot dari *user*

4. Menghitung vektor S

$$S_i = \prod_{j=1}^n (X_{ij})^{w_j} \quad (2.2)$$

Keterangan:

$S_i$  = Nilai presensi alternatif  $A_n$  (vektor S)

$W_j$  = Nilai perbaikan bobot *user* (positif jika *benefit* / negatif jika *cost*)

$X_{ij}$  = Nilai rating kerja atribut ke-i terhadap subkriteria ke-j

5. Menentukan nilai vektor V

$$V_i = \frac{\prod_{j=1}^n x_{ij}^{w_j}}{\prod_{j=1}^n (x_j^*)^{w_j}}; \text{ dengan } i = 1, 2, \dots, m \quad (2.3)$$

Keterangan:

$V_i$  = Nilai prefensi relatif dari setiap alternatif (Vektor V)

$W_j$  = Nilai perbaikan bobot *user* (positif jika *benefit* / negative jika *cost*)

$X_{ij}$  = Nilai rating kerja atribut ke-i terhadap subkriteria ke-j

$X_j^*$  = Jumlah nilai prefensi alternatif (Vektor S)

8. Nilai vektor  $V$  terbesar menyatakan bahwa merupakan alternatif terbaik

Hasil dari vektor  $V$  merupakan acuan yang digunakan untuk perankingan dari nilai alternatif yang dihitung. Nilai alternatif terbaik yang dipilih merupakan nilai vektor  $V$  terbesar dari setiap alternatif. Hasil perhitungan metode *Weighted Product* bernilai akurasi 100% berdasarkan perbandingan perhitungan manual dan perhitungan pada sistem pendukung keputusan pemilihan laptop. Tahapan metode *Weighted Product* (WP) dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Tahapan Metode *Weighted Product*

Tahapan metode *Weighted Product* (WP) pada Gambar 2.1, tahap awal yaitu menentukan *benefit* dan *cost* semua kriteria. Kemudian menghitung perbaikan bobot kriteria, diaman apabila dijumlahkan harus sama dengan 1. Tahap selanjutnya, yaitu memangkatkan semua nilai atribut alternatif dengan nilai perbaikan bobot kriteria yang saling berhubungan. Pada tahap tersebut menghasilkan nilai vektor  $S$ . Seluruh nilai vektor  $S$  semua alternatif dijumlahkan. Lalu, membagi nilai vektor  $S$  setiap alternatif dengan hasil penjumlahan vektor  $S$  semua alternatif. Setelah itu akan mendapatkan nilai vektor  $V$  setiap alternatif. Setelah itu akan mendapatkan nilai vektor  $V$  setiap alternatif. Tahap terakhir yaitu meranking vektor  $V$  dari semua alternatif.

## 2.2 Pengertian Paduan Suara

Paduan suara adalah penyajian musik *vocal* yang terdiri dari 15 orang atau lebih yang memadukan berbagai warna suara menjadi satu kesatuan yang utuh dan dapat menampilkan jiwa lagu yang dibawakan. Paduan suara biasanya dipimpin oleh seorang dirigen atau *choirmaster* yang umumnya sekaligus adalah pelatih

paduan suara tersebut. Tiap penyanyi dalam paduan suara harus mampu mengharmoniskan suara mereka antar penyanyi dalam paduan suara yang lain. Tidak boleh ada suara penyanyi yang mendominasi dalam sebuah tim paduan suara (Kob, et al., 2011).

Umumnya jenis suara dalam paduan suara terdiri dari 4 bagian suara (Kob, et al., 2011) antara lain:

- a. Sopran, yaitu suara tertinggi pada wanita
- b. Alto, yaitu suara terendah pada wanita
- c. Tenor, yaitu suara tertinggi pada pria
- d. Bass, yaitu suara terendah pada pria

Untuk menjadi anggota paduan suara dilakukan beberapa tahapan seleksi. Kriteria yang dinilai dalam seleksi antara lain:

- a. Intonasi
- b. Kemampuan membaca not balok
- c. Kemampuan membaca not angka
- d. Artikulasi
- e. Power suara

### **2.3 Manajemen Sistem Informasi**

Kata sistem dapat didefinisikan sebagai sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau yang dihubungkan dengan cara cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan (Edhy, 2005). Fungsi utama diterapkannya sistem informasi manajemen dalam suatu organisasi adalah sebagai berikut:

1. Mempermudah pihak manajemen untuk melakukan perencanaan, pengawasan dan pengarahan pada berbagai bagian yang ada.
2. Meningkatkan efisiensi dan efektifitas data yang tersaji akurat dan tepat waktu.
3. Meningkatkan produktifitas dan penghematan biaya dalam suatu organisasi.
4. Meningkatkan kualitas sumber daya manusia karena unit sistem kerja yang terkoordinir dan sistematis.

## BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tentang jenis penelitian, pengembangan sistem informasi dan pengujian metode *Weighted Product*

### 3.1 Jenis Penelitian

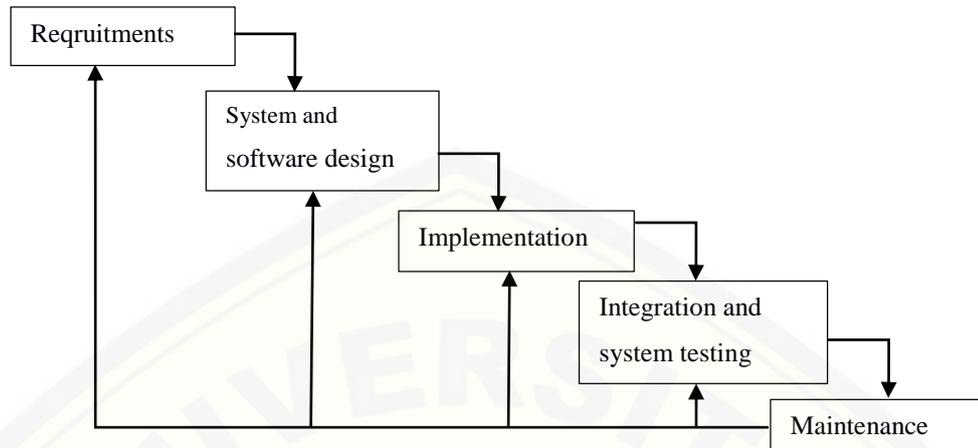
Jenis penelitian yang digunakan merupakan jenis penelitian pengembangan, karena tujuan penelitian adalah untuk membangun sistem informasi seleksi penerimaan anggota paduan suara mahasiswa Universitas Jember menggunakan metode *Weighted Product*. Penelitian pengembangan bertujuan untuk membuat dan mengembangkan suatu produk yang efektif untuk digunakan. Penelitian ini bukan penelitian yang ditujukan untuk menemukan teori atau menguji kebenaran dari suatu teori dalam bentuk eksperimentasi.

### 3.2 Tempat dan Waktu Penelitian

Tempat yang akan dilaksanakan untuk penelitian adalah Universitas Jember. Waktu penelitian dilakukan selama 6 bulan yaitu dimulai dari bulan Oktober 2017 sampai dengan bulan Maret 2018.

### 3.3 Teknik Pengembangan Sistem

Perancangan sistem dilakukan setelah analisis pada sistem telah selesai dilakukan. Perancangan dilakukan untuk memberikan gambaran secara umum mengenai sistem yang akan dibangun. Pembuatan perancangan perangkat lunak ini mengadopsi pola pengembangan metode *waterfall*. Metode *waterfall* merupakan metode yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, test dan pemeliharaan (Pressman, 2002).



Gambar 3. 1 Alur SDLC *Waterfall*

### 3.3.1 Analisis Kebutuhan

Langkah awal dalam perancangan sistem ini adalah analisis kebutuhan sistem yang akan di bangun. Pengumpulan data dalam tahapan ini dapat dilakukan dengan melakukan penelitian, wawancara atau studi literatur. Pada analisis kebutuhan ini sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga data akan di bentuk menjadi kebutuhan fungsional dan non fungsional dari sistem tersebut. Metode yang digunakan adalah sebagai berikut:

#### a. Metode Pengumpulan Data

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

1. Melakukan wawancara, wawancara merupakan cara pengumpulan data yang berhubungan langsung dengan narasumber dengan mengajukan pertanyaan kepada pelatih serta panitia seleksi anggota paduan suara Universitas Jember untuk mendapatkan data dan informasi mengenai proses penyeleksian anggota paduan suara Universitas Jember. Wawancara terhadap narasumber bertujuan untuk memperoleh data yang dibutuhkan dalam penyelesaian penelitian. Adapun data yang diperoleh yaitu kriteria dan bobot untuk menyeleksi anggota paduan suara.

2. Melakukan studi literature, jurnal, media maupun internet mengenai pembangunan sistem informasi seleksi penerimaan anggota paduan suara mahasiswa Universitas Jember seperti data kriteria masing-masing yakni intonasi, penguasaan not balok, penguasaan not angka, artikulasi, serta power suara.

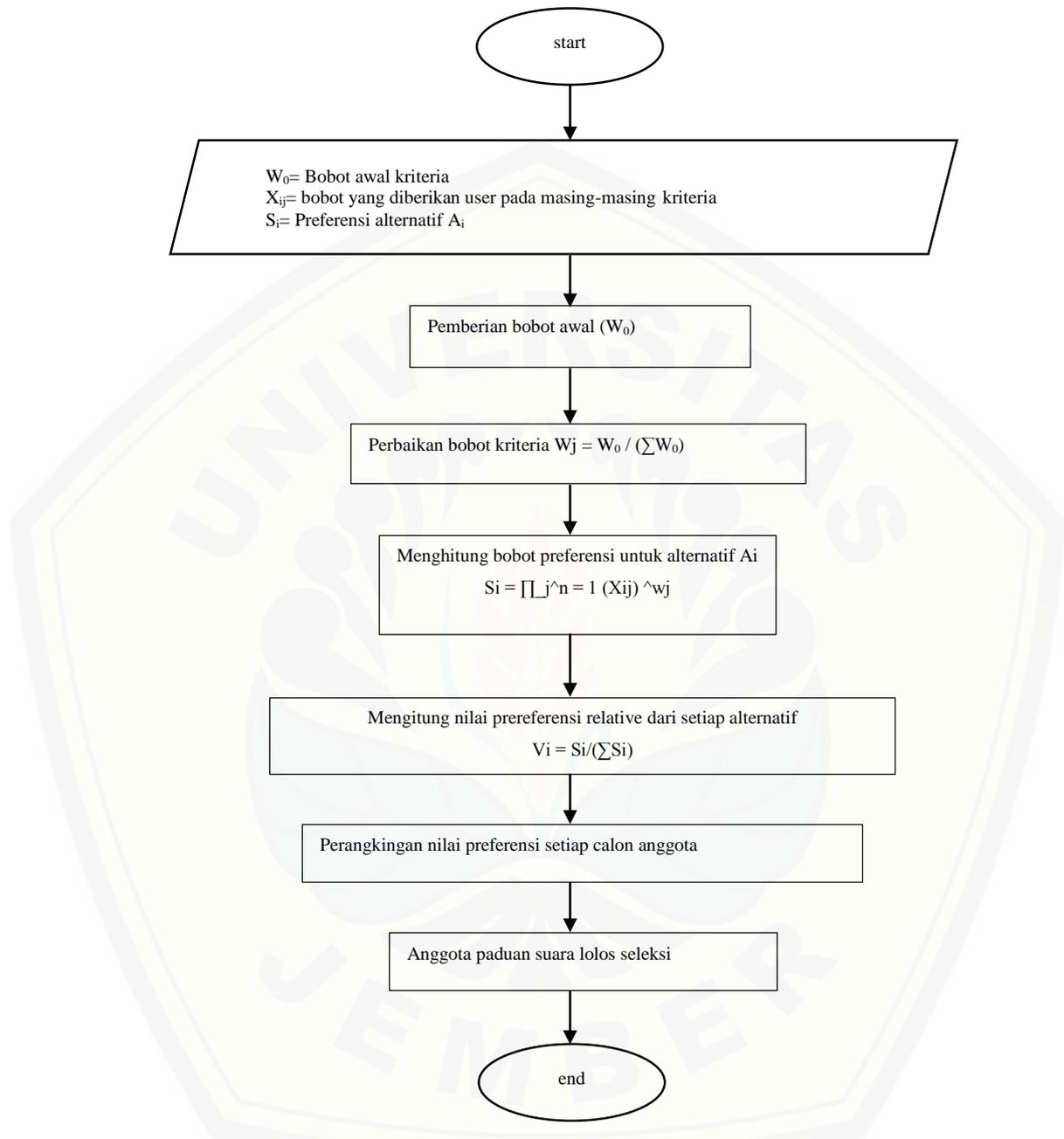
Jenis data yang digunakan dalam penelitian ini adalah data primer dan data sekunder. Data primer merupakan data yang diperoleh melalui metode wawancara. Data sekunder merupakan data yang diperoleh dari studi literatur.

- b. Metode Analisis Data

Tahap analisis data dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Data tersebut diantaranya yakni data kriteria yang ditentukan untuk menyeleksi anggota paduan suara yakni intonasi, penguasaan not balok, penguasaan not angka, artikulasi, seta power suara. Langkah selanjutnya adalah pada setiap kriteria tersebut ditentukan nilai bobot awal untuk di proses menggunakan metode *weighted product*. Data yang dihasilkan berupa nilai mahasiswa dari yang tertinggi ke terendah, dimana nilai 30 mahasiswa tertinggi akan lolos seleksi paduan suara. Adapun menganalisis data dengan menggunakan metode *weighted product* prosesnya dapat dilihat pada Gambar 3.2.

### 3.3.2 Desain

Proses pendesainan sistem dari aplikasi yang akan kita bangun yaitu dengan menggunakan *Unified Modelling Language* (UML). Penggunaan UML karena sudah menggunakan konsep *Object Oriented Design* yang tentunya akan sangat memudahkan *developer* untuk membangun sebuah sistem. Dalam UML ada beberapa diagram yang akan dibuat antara lain: *Business process*, *Use Case Diagram*, *Scenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Entity Relationship Diagram*.



Gambar 3. 2 Diagram Alir Penerapan Metode *Weighted Product* (WP)

### 3.3.3 Implementasi

Setelah proses desain sistem dikerjakan, proses selanjutnya adalah *coding* atau penulisan kode program. Bahasa pemrograman yang dipakai adalah php

(*Hypertext Preprocessor*) dengan *codeigniter framework*, sedangkan *tool* yang digunakan adalah *sublime text*. *Database* yang digunakan yaitu MySQL dengan *tool* yang digunakan yaitu XAMPP.

#### 3.3.4 Testing

Pada tahap ini dilakukan uji coba sistem yang telah dibangun. Kegiatan pengujian yang dilakukan bertujuan untuk menguji spesifikasi sistem yang telah dibangun sehingga teknik pengujian yang dilakukan adalah pengujian *black box*. (Jalote, 2005). Pengujian *black box* merupakan cara pengujian dengan melakukan *running program* dan menguji coba berbagai kemungkinan kesalahan yang ada.

*Black Box Testing* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Pengetahuan khusus dari metode aplikasi, struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Khan, 2011).

#### 3.3.5 Pemeliharaan

Langkah terakhir dalam proses perancangan adalah *maintenance* atau perawatan sistem. Perangkat lunak yang sudah di sampaikan kepada *user* pasti akan mengalami perubahan. *Maintenance* sangat diperlukan untuk menangani masalah pada program yang digunakan oleh *user*. Masalah tersebut bisa terjadi karena mengatasi kesalahan, karena perangkat lunak harus menyesuaikan dengan sistem operasi baru atau karena *user* membutuhkan perkembangan fungsional.

### 3.4 Analisis Metode

Sistem informasi seleksi penerimaan anggota paduan suara merupakan sistem yang dapat membantu proses seleksi penerimaan anggota paduan suara Universitas Jember dengan menerapkan metode *Weighted Product*. Metode *Weighted Product* dapat diterapkan dalam sistem seleksi penerimaan anggota paduan suara Universitas Jember dan menghasilkan nilai yang sama antara perhitungan manual dan perhitungan sistem. Hasil akhir metode *weighted product* berupa perbandingan

nilai tertinggi ke nilai terendah. Nilai tertinggi menempati prioritas utama untuk seleksi penerimaan anggota paduan suara. Hasil perhitungan metode *Weighted Product* lebih teliti dibandingkan dengan metode *Simple Additive Weighted* (SAW) Tingkat keakuratan metode *Weighted Product* 80%. Dikarenakan metode *Weighted Product* memanfaatkan perkalian dari nilai rating kriteria yang dipangkatkan dengan nilai bobot yang telah diperbaiki sehingga urutan ranking yang diberikan lebih tepat dengan hasil yang spesifik.

### 3.5 Gambaran Umum Sistem

Sistem Informasi Seleksi Penerimaan Anggota Paduan Suara Mahasiswa Universitas Jember Siswa merupakan sistem yang dibangun untuk membantu proses manajemen penerimaan calon anggota paduan suara Universitas Jember Pada sistem ini menerapkan metode *Weighted Product* guna untuk menentukan calon anggota paduan suara Universitas Jember. Sistem ini nantinya diakses oleh empat *user*, yaitu admin, admin penerimaan, pelatih dan mahasiswa. Admin penerimaan sebagai pakar yang dapat mengelola bobot, kriteria dari sistem yang akan dijadikan parameter untuk penilaian seleksi penerimaan anggota paduan suara Universitas Jember. *User* kedua adalah admin sebagai *user* yang mengelola informasi-informasi yang berkaitan dengan paduan suara. *User* yang ketiga yaitu pelatih yang berhak memvalidasi mahasiswa yang lolos seleksi serta melihat mahasiswa yang lolos seleksi dari tahun sebelumnya. *User* keempat yaitu mahasiswa yang dapat melakukan registrasi pendaftaran.

Nilai mahasiswa yang mengikuti tes seleksi nantinya akan di kelola atau diinputkan oleh admin penerimaan berdasarkan nilai dari masing-masing kriteria. Kriteria yang digunakan dalam sistem ini berdasarkan yang terdapat pada hasil analisis pribadi melalui wawancara terkait proses seleksi penerimaan anggota paduan suara Universitas Jember. Dari data yang telah diinputkan nantinya akan diberi pembobotan tersendiri yang telah ditentukan sebelumnya oleh pelatih dan di proses menggunakan metode *Weighted Product* (WP) sehingga menghasilkan *output* berupa ranking anggota paduan suara dari yang terbesar hingga terkecil. Hasil dari perankingan akan diambil maksimal 30 nilai terbesar. Dimana ranking

calon anggota terbesar dapat dijadikan penunjang keputusan untuk lolos seleksi penerimaan anggota paduan suara Universitas Jember. Selain hal tersebut status anggota paduan suara yang lolos seleksi akan tersimpan dalam sistem, sehingga memudahkan pengguna dalam melihat catatan anggota paduan. Selain itu sistem informasi ini berbasis web sehingga dapat memudahkan penggunaanya.



## BAB 4 PENGEMBANGAN SISTEM

Bab ini akan membahas tentang implementasi metode *weighted product* pada sistem informasi seleksi penerimaan anggota paduan suara mahasiswa Universitas Jember. Tahap pengembangan dilaksanakan berdasarkan model *waterfall*, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, pengujian, dan pemeliharaan.

### 4.1 Deskripsi Umum Sistem Informasi

Sistem informasi seleksi penerimaan anggota paduan suara merupakan sistem informasi berbasis *web* yang mampu membantu mengambil keputusan pada seleksi penerimaan anggota paduan suara menggunakan metode *Weighted Product*. Data yang dibutuhkan dalam sistem informasi ini yaitu data pendaftar, data kriteria dan data nilai bobot. Sistem informasi ini memiliki tiga hak akses, yaitu admin penerimaan, pelatih, dan admin *website*. Setiap masing-masing hak akses memiliki peran yang berbeda dalam menggunakan sistem informasi. Sistem informasi dapat membantu mengambil keputusan dengan menggunakan metode *Weighted Product* dimana pelatih memberikan nilai hasil seleksi calon anggota paduan suara kemudian admin penerimaan memasukkan nilai pendaftar ke sistem informasi yang selanjutnya sistem informasi akan memberikan keputusan berupa perangkingan dari nilai terbesar ke terkecil. Hasil perangkingan akan diambil 30 nilai terbesar. Selain membantu mengambil keputusan, sistem informasi ini juga dapat menampilkan laporan pendaftar tahun sebelumnya dan mencetak laporan penerimaan. Sistem informasi ini juga menampilkan *event* paduan suara dan menampilkan galeri dokumentasi paduan suara.

### 4.2 Analisis kebutuhan sistem

Tahap analisis kebutuhan sistem merupakan tahapan yang penting dalam pengembangan sebuah sistem informasi. Seluruh kebutuhan didefinisikan pada tahap kebutuhan fungsional dan kebutuhan non-fungsional.

#### 4.2.1 Hasil Wawancara

Wawancara yang telah dilakukan kepada Indah (Ketua Paduan Suara Universitas Jember), Rokhmat Hidayanto S,Si (Pelatih Paduan Suara Universitas Jember) yang menyatakan bahwa Paduan Suara Universitas Jember mengalami kendala dalam seleksi penerimaan anggota paduan suara Universitas Jember yang masih menggunakan sistem manual dalam proses seleksi. Selain itu kurangnya informasi mengenai paduan suara Universitas Jember menyebabkan minat calon anggota paduan suara berkurang. Diharapkan dengan adanya sistem informasi ini dapat menghasilkan solusi dalam menangani permasalahan yang dialami Unit Kegiatan Mahasiswa (UKM) Paduan Suara Universitas Jember. Berdasarkan wawancara yang telah dilakukan, didapatkan hasil berupa data pendaftar, informasi kriteria-kriteria dan nilai bobot dari masing-masing kriteria yang nantinya dapat digunakan dalam melakukan seleksi penerimaan calon anggota paduan suara. Transkrip wawancara dapat dilihat pada Lampiran D.

#### 4.2.2 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berisi fitur-fitur apa saja yang nantinya dapat dilakukan oleh sistem informasi. Kebutuhan fungsional sistem informasi seleksi penerimaan calon anggota paduan suara Universitas Jember antara lain:

1. Sistem mampu mengelola data event (tambah, lihat, ubah)
2. Sistem mampu mengelola galeri kegiatan (tambah, lihat, ubah)
3. Sistem mampu mengelola data penerimaan (lihat, ubah)
4. Sistem mampu mengelola data pendaftar (tambah, lihat, ubah)
5. Sistem mampu melihat data penerimaan mahasiswa tahun sebelumnya
6. Sistem mampu mengubah status data mahasiswa lolos seleksi
7. Sistem mampu melihat data mahasiswa lolos seleksi
8. Sistem mampu mengelola data pendaftar dengan metode *weighted product*
9. Sistem mampu menampilkan data hasil perankingan
10. Sistem mampu mencetak hasil perankingan

#### 4.2.3 Kebutuhan Non-fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang merupakan menjadi pendukung pada sistem informasi. Kebutuhan non-fungsional seleksi penerimaan calon anggota paduan suara Universitas Jember antara lain:

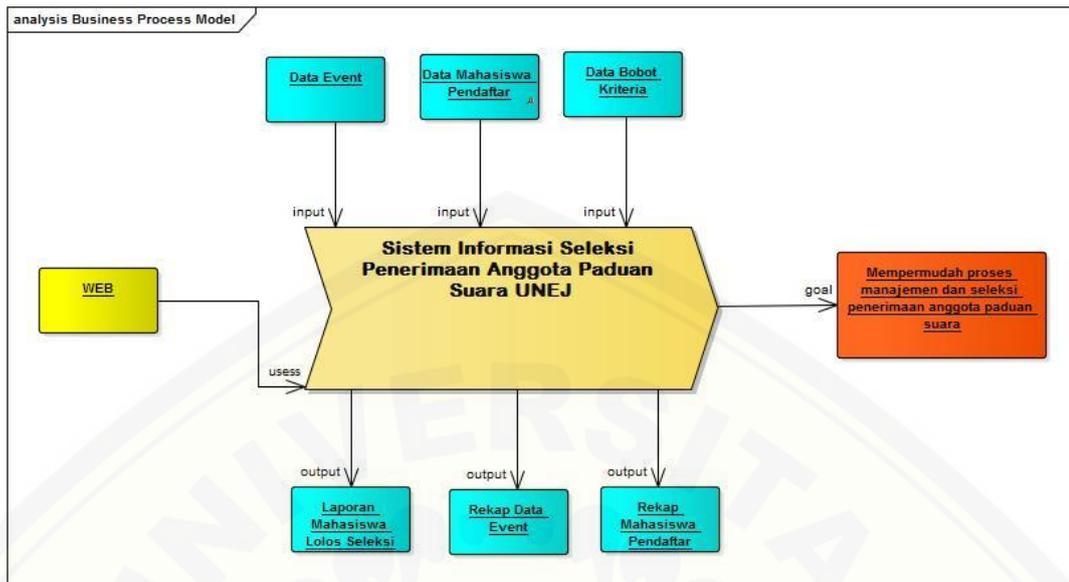
1. Sistem dapat dijalankan oleh beberapa *software web browser* diantaranya *internet explore*, *Google Chrome*, dan *Mozilla Firefox*.
2. Sistem memiliki tampilan (antar muka) yang mudah dipahami.

### 4.3 Desain Sistem Informasi

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

#### 4.3.1 Komponen *Input Output*

Komponen *input output* merupakan model yang menggambarkan proses yang terjadi dalam sistem informasi mulai dari *input*, *output*, *uses*, dan *goal* dari sistem informasi yang dibuat. Sistem informasi seleksi penerimaan calon anggota paduan suara Universitas Jember ini menggunakan platform *website* dan metode *Weighted Product* dalam membantu mengambil keputusan seleksi penerimaan calon anggota paduan suara. Sistem informasi ini menggunakan data pendaftar, data bobot kriteria, dan data event sebagai *input* ke sistem informasi. *Input* tersebut nantinya akan menghasilkan *output* berupa laporan mahasiswa lolos seleksi, rekap mahasiswa pendaftar, dan rekap data event, sedangkan *goal* dari sistem informasi ini adalah mempermudah proses manajemen dan dapat membantu seleksi penerimaan calon anggota paduan suara Universitas Jember. Komponen *input output* dari sistem informasi peramalan persediaan bahan baku ini dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Komponen *input output*

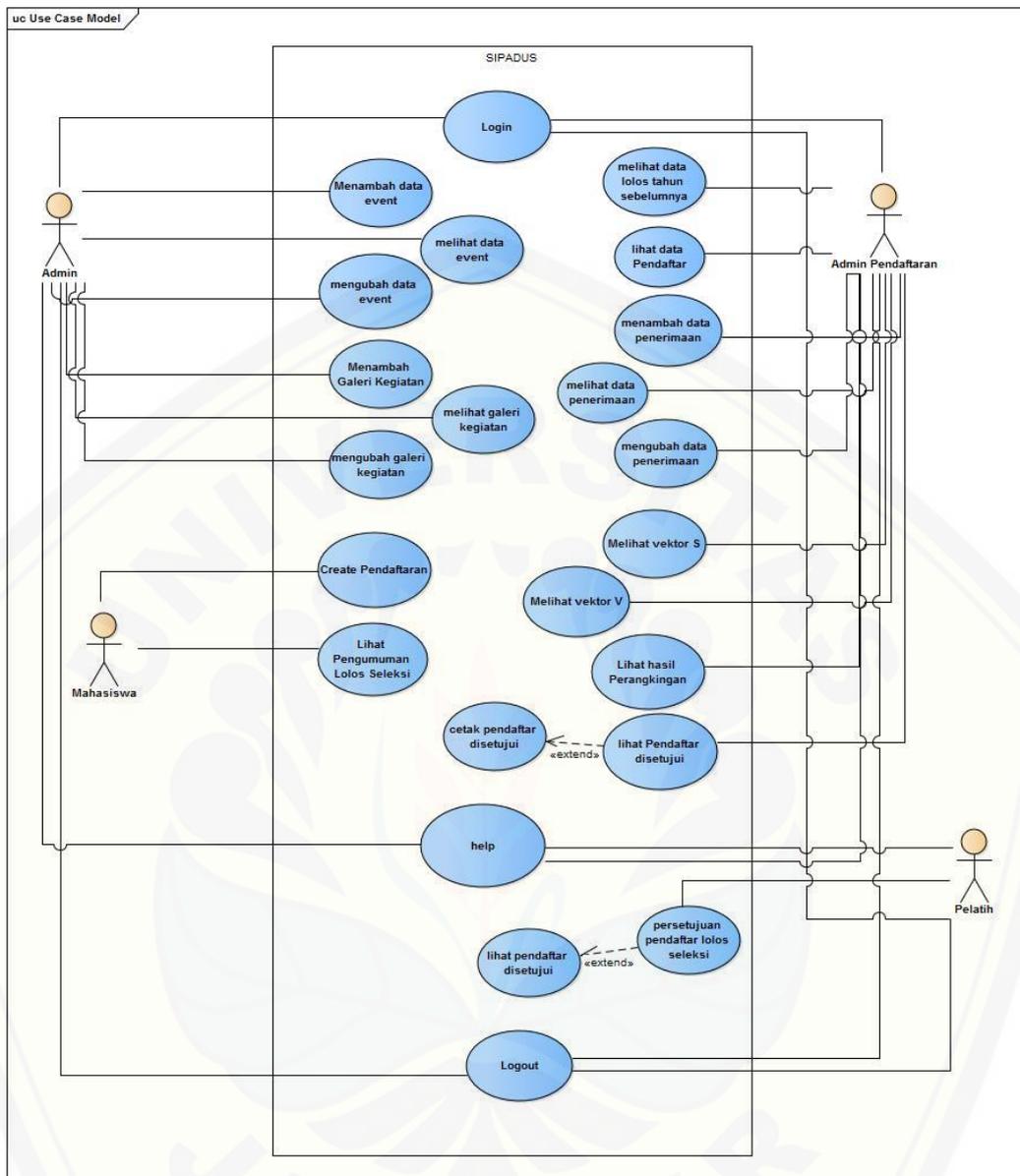
#### 4.3.2 Usecase Diagram

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

Penjelasan tentang definisi aktor dan definisi *use case* dalam *use case* diagram pada Gambar 4.2 akan dijelaskan dibawah ini.

##### 1. Definisi Aktor

Definisi aktor merupakan penjelasan tentang aktor-aktor sebagai pengguna dari sistem informasi seleksi penerimaan anggota paduan suara mahasiswa Universitas Jember dengan mengimplementasikan metode *Weighted Product* yang akan dibangun. Terdapat 4 aktor seperti yang dijelaskan pada Gambar 4.2.



Gambar 4. 2 Use Case Diagram

Tabel 4. 1 Definisi Aktor

No	Aktor	Deskripsi
1.	Admin penerimaan	Aktor Admin penerimaan pada sistem informasi ini memiliki hak akses untuk: 1. Menambah Data Pendaftar Sebelumnya

Dilanjutkan

Lanjutan

No	Aktor	Deskripsi
		<ol style="list-style-type: none"> <li>1. Mengelola Data Penerimaan</li> <li>2. Melihat Data Pendaftar</li> <li>3. Memasukkan Nilai Hasil Seleksi</li> <li>4. Melihat Hasil Perhitungan Vektor S</li> <li>5. Melihat Hasil Perhitungan Vektor V</li> <li>6. Melihat Hasil Perangkingan</li> <li>7. Mencetak Hasil Perangkingan yang telah disetujui oleh pelatih</li> </ol>
2.	. Admin <i>Website</i>	<p>Aktor Admin <i>Website</i> penerimaan pada sistem informasi ini memiliki hak akses untuk:</p> <ol style="list-style-type: none"> <li>1. Menambah Data Event</li> <li>2. Mengelola Galeri Kegiatan</li> </ol>
3.	Pelatih	<p>Aktor Pelatih penerimaan pada sistem informasi ini memiliki hak akses untuk:</p> <ol style="list-style-type: none"> <li>1. Melihat Hasil Perangkingan</li> <li>2. Mengubah Status Penerimaan</li> </ol>
4.	Mahasiswa	<p>Aktor mahasiswa penerimaan pada sistem informasi ini memiliki hak akses untuk:</p> <ol style="list-style-type: none"> <li>1. Melihat Hasil Perangkingan</li> </ol>

## 2. Definisi *Use Case*

Definisi *usecase* yaitu penjelasan tentang fitur-fitur yang tersedia dalam sistem informasi peramalan seleksi penerimaan calon anggota paduan suara Universitas Jember. Terdapat 21 usecase pada Tabel 4.2.

Tabel 4. 2 Definisi Use Case

<b>No.</b>	<b>Use Case</b>	<b>Deskripsi</b>
1.	Masuk sistem	Menggambarkan proses autentifikasi pengguna yang digunakan untuk masuk ke sistem informasi.
2.	Menambah data event	Menggambarkan proses untuk menambah data event.
3.	Melihat data event	Menggambarkan proses untuk melihat data event
4.	Mengubah data event	Menggambarkan proses untuk mengubah data event
5.	Menambah galeri kegiatan	Menggambarkan proses untuk menambah galeri kegiatan
6.	Melihat galeri kegiatan	Menggambarkan proses melihat galeri kegiatan
7.	Mengubah galeri kegiatan	Menggambarkan proses untuk mengubah galeri kegiatan
8.	<i>Create</i> pendaftaran	Menggambarkan proses untuk melakukan registrasi pendaftaran
9.	Melihat data lolos tahun sebelumnya	Menggambarkan proses untuk melihat data pendaftar yang lolos tahun sebelumnya
10.	Melihat data pendaftar	Menggambarkan proses melihat data pendaftar
11.	<i>Input</i> nilai hasil seleksi	Menggambarkan proses menambah nilai hasil seleksi
12.	Menambah data penerimaan	Menggambarkan proses menambah data penerimaan

Dilanjutkan

## Lanjutan

13.	Melihat data penerimaan	Menggambarkan proses melihat data penerimaan
14.	Mengubah data penerimaan	Menggambarkan proses untuk mengubah data penerimaan
15.	Melihat vektor S	Menggambarkan proses untuk melihat hasil perhitungan menggunakan vektor S
16.	Melihat vektor V	Menggambarkan proses untuk melihat hasil perhitungan menggunakan vektor S
17.	Lihat hasil perangkingan	Menggambarkan proses untuk melihat hasil perangkingan
18.	Persetujuan Pendaftar Lolos	Menggambarkan proses persetujuan pendaftar lolos seleksi
19.	Cetak Pendafatar Disetujui	Menggambarkan proses untuk mencetak pendaftar disetujui
20.	Help	Menggambarkan proses untuk menjelaskan hak akses dari masing-masing aktor
21.	Logout	Menggambarkan proses untuk keluar sistem

#### 4.3.3 Use Case Scenario

*Use Case Scenario* digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada use case diagram seperti Gambar 4.2.

##### 1. Skenario *Use Case* login

Skenario masuk merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan masuk ke dalam sistem informasi. Terdapat tiga *user* dalam sistem informasi yang dapat *login* ke dalam sistem ini yaitu Admin Penerimaan, Pelatih dan Admin. Proses *login* dilakukan ketika aktor mengklik button *login* yang ada pada halaman *landing page* sistem. Sistem informasi memiliki reaksi yang berbeda untuk masing-masing aktor sehingga aktor dapat

masuk ke sistem informasi dan menggunakannya. Skenario masuk dapat dilihat pada lampiran A Tabel 1.

## 2. Skenario *Use Case* Menambah Data Event

Skenario menambah data event merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor menambah data event baru. Aktor yang dapat menambah data event yaitu admin. Proses Untuk menambah data event, admin dapat mengklik menu info event pada halaman dashboard kemudian klik submenu tambah data event. Sistem akan menampilkan form tambah event yang dapat diisi oleh admin. Kemudian admin mengklik button submit, maka secara otomatis data event telah ditambahkan. Skenario menambah data event dapat dilihat pada lampiran A Tabel 2.

## 3. Skenario *Use Case* Melihat Data Event

Skenario melihat data event merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat data event yang sudah ditambahkan. Aktor yang dapat melihat data event yaitu admin. Proses untuk melihat data event, aktor admin dapat mengklik menu info event pada halaman *dashboard* kemudian klik submenu lihat data event. Sistem akan menampilkan halaman lihat data event berupa tabel. Skenario melihat data event dapat dilihat pada lampiran A Tabel 3.

## 4. Skenario *Use Case* Mengubah Data Event

Skenario mengubah data event merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor mengubah data event. Aktor yang dapat mengubah data event yaitu admin. Proses untuk mengubah data event, admin dapat mengklik menu info event kemudian klik submeu lihat data event. Sistem akan menampilkan halaman lihat data event berupa tabel dan *action button* edit. Kemudian aktor mengklik *button edit*, maka sistem akan menampilkan form ubah data event. Klik submit apabila proses ubah data event telah selesai, maka data event berhasil diubah. Skenario mengubah data event dapat dilihat pada lampiran A Tabel 4.

#### 5. Skenario *Use Case* Menambah Galeri Kegiatan

Skenario menambah galeri kegiatan merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor menambah galeri kegiatan baru. Aktor yang dapat menambah galeri kegiatan yaitu admin. Proses untuk menambah galeri kegiatan, admin dapat mengklik menu galeri kegiatan pada halaman *dashboard* admin kemudian klik submenu tambah galeri kegiatan. Sistem akan menampilkan form tambah galeri kegiatan yang dapat diisi oleh admin. Kemudian admin mengklik *button submit*, maka secara otomatis galeri kegiatan telah ditambahkan. Skenario menambah galeri kegiatan dapat dilihat pada lampiran A Tabel 5.

#### 6. Skenario *Use Case* Melihat Galeri Kegiatan

Skenario melihat data galeri kegiatan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat data galeri kegiatan yang sudah ditambahkan. Aktor yang dapat melihat galeri kegiatan yaitu admin. Proses untuk melihat galeri kegiatan, admin dapat mengklik menu galeri kegiatan kemudian klik submeu lihat galeri kegiatan. Sistem akan menampilkan halaman lihat galeri kegiatan berupa tabel dan *action button* edit. Skenario melihat galeri kegiatan dapat dilihat pada lampiran A Tabel 6.

#### 7. Skenario *Use Case* Mengubah Galeri Kegiatan

Skenario mengubah galeri kegiatan merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor mengubah galeri kegiatan. Aktor yang dapat mengubah galeri kegiatan yaitu admin. Proses untuk mengubah galeri kegiatan, admin dapat mengklik menu galeri kegiatan kemudian klik submeu lihat galeri kegiatan. Sistem akan menampilkan halaman lihat galeri kegiatan berupa tabel dan *action button* edit. Kemudian aktor mengklik *button edit*, maka sistem akan menampilkan form ubah galeri kegiatan. Klik submit apabila proses ubah galeri kegiatan telah selesai, maka galeri kegiatan berhasil diubah. Skenario mengubah galeri kegiatan dapat dilihat pada lampiran A Tabel 7.

#### 8. Skenario *Use Case* Pendaftaran

Skenario pendaftaran merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melakukan registrasi pendaftaran. Aktor yang dapat melakukan registrasi pendaftaran yaitu mahasiswa (pendaftar). Mahasiswa dapat melakukan proses pendaftaran dengan cara mengklik menu registrasi pada halaman beranda *website*. Kemudian mengisi form registrasi. apabila telah selesai, klik *button submit* untuk menyimpan data. Skenario pendaftaran dapat dilihat pada lampiran A Tabel 8.

#### 9. Skenario *Use Case* Melihat Data Lolos Tahun Sebelumnya

Skenario melihat data pendaftar lolos tahun sebelumnya merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat data pendaftar yang lolos tahun sebelumnya. Aktor yang dapat melihat data lolos tahun sebelumnya yaitu admin penerimaan. Proses untuk melihat data lolos tahun sebelumnya, klik menu lihat semua vektor  $v$  pada *dashboard*. Sistem akan menampilkan halaman data lolos tahun sebelumnya berupa tabel. Skenario melihat data pendaftar lolos tahun sebelumnya dapat dilihat pada lampiran A Tabel 9.

#### 10. Skenario *Use Case* Lihat Data Pendaftar

Skenario lihat data pendaftar merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat data pendaftar. Aktor yang dapat melihat data pendaftar yaitu admin penerimaan. Proses untuk melihat halaman lihat data pendaftar, klik menu lihat pendaftar pada *dashboard* admin penerimaan. Sistem akan menampilkan halaman data pendaftar berupa tabel. Skenario lihat data pendaftar dapat dilihat pada lampiran A Tabel 10.

#### 11. Skenario *Use Case Input* Nilai Hasil Seleksi

Skenario *input* nilai hasil seleksi merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor akan menginputkan nilai hasil seleksi. Aktor yang dapat menginputkan nilai hasil seleksi yaitu admin penerimaan. Proses untuk menginputkan nilai hasil seleksi, klik menu lihat data pendaftar. Sistem akan menampilkan halaman lihat pendaftar berupa tabel dan *button edit*. Klik button edit

kemudian input nilai pendaftar dan di *submit*. Skenario input nilai hasil seleksi dapat dilihat pada lampiran A Tabel 12.

#### 12. Skenario *Use Case* Menambah Data Penerimaan

Skenario menambah data penerimaan merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor menambah data penerimaan baru. Aktor dapat menambah data penerimaan baru. Aktor yang dapat menambah data penerimaan yaitu admin penerimaan. Proses untuk menambah data penerimaan, klik menu data penerimaan pada *dashboard* admin penerimaan kemudian klik submenu klik tambah data penerimaan. Sistem akan menampilkan form tambah data penerimaan kemudian klik *submit* untuk menyimpan data. Skenario menambah data penerimaan dapat dilihat pada lampiran A Tabel 13.

#### 13. Skenario *Use Case* Melihat Data Penerimaan

Skenario melihat data penerimaan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat data penerimaan yang telah ditambahkan. Aktor yang dapat melihat data penerimaan yaitu admin penerimaan. Proses untuk melihat data penerimaan, klik menu data penerimaan pada *dashboard* admin penerimaan. Sistem akan menampilkan halaman lihat data penerimaan berupa tabel. Skenario melihat data penerimaan dapat dilihat pada lampiran A Tabel 14.

#### 14. Skenario *Use Case* Mengubah Data Penerimaan

Skenario mengubah data penerimaan merupakan alur yang menjelaskan aksi dan reaksi sistem informasi pada saat aktor mengubah data penerimaan. Aktor dapat mengubah data penerimaan. Aktor yang dapat mengubah data penerimaan yaitu admin penerimaan. Proses untuk mengubah data penerimaan, klik menu lihat data penerimaan kemudian klik submenu lihat data penerimaan. Sistem akan menampilkan halaman lihat data penerimaan berupa tabel dan *button edit*. Klik *button edit* untuk mengubah data kemudian klik *button submit* untuk menyimpan data. Skenario mengubah data penerimaan dapat dilihat pada lampiran A Tabel 15.

### 15. Skenario *Use Case* Melihat Vektor S

Skenario melihat vektor S merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat hasil perhitungan nilai prefensi dengan vektor S. Aktor dapat melihat hasil perhitungan nilai prefensi dengan menggunakan vektor S. Skenario melihat vektor S dapat dilihat pada Tabel 4.3

Tabel 4. 3 Skenario Melihat Perhitungan Vektor S

No. Usecase	14
Nama usecase	Melihat Perhitungan Vektor S
Aktor	Admin penerimaan
Deskripsi	Fitur untuk melihat hasil dari perhitungan nilai prefensi dengan vektor S
Prakondisi	Memasuki halaman beranda
Pascakondisi	Hasil perhitungan vektor S telah selesai dilihat
Event Flow	
Normal flow : melihat perhitungan vektor S	
Aksi actor	Reaksi sistem
1. Klik menu Vektor S	
2. Klik submenu lihat vektor S	
	3. Menghitung nilai prefensi dengan vektor S menggunakan persamaan: $S_i = \prod_{j=1}^n -1(X_{ij})^w$
	4. Menampilkan halaman lihat vektor S dengan tabel yang berisi: <ul style="list-style-type: none"> <li>a. Nama (varchar 50)</li> <li>b. NIM (varchar 25)</li> <li>c. Vektor S (double)</li> </ul>

### 16. Skenario *Use Case* Melihat Vektor V

Skenario melihat vektor V merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat hasil perhitungan nilai prefensi relatif dari setiap alternatif (vektor V). Aktor dapat melihat hasil

perhitungan nilai prefensi relatif dari setiap alternatif (vektor V). Skenario melihat vektor V dapat dilihat pada Tabel 4.4

Tabel 4. 4 Skenario Melihat Perhitungan Vektor V

No. Usecase	15
Nama usecase	Melihat Perhitungan Vektor V
Aktor	Admin penerimaan
Deskripsi	Fitur untuk melihat hasil dari perhitungan nilai prefensi setiap alternatif (vektor V)
Prakondisi	Memasuki halaman beranda
Prakondisi	Memasuki halaman beranda
Pascakondisi	Hasil perhitungan vektor V telah selesai dilihat
Event Flow	
Normal flow : melihat perhitungan vektor V	
Aksi actor	Reaksi sistem
1. Klik menu Vektor V	
2. Klik submenu lihat vektor V	
	3. Menghitung vektor V dengan menggunakan persamaan: $V_i = \frac{\prod_{j=1}^n x_{ij}^{w_i}}{\prod_{j=1}^n (x_j^*)^{w_j}}; \text{ dengan } i = 1, 2, \dots, m$
	4. Menampilkan halaman lihat vektor V dengan tabel yang berisi: <ul style="list-style-type: none"> <li>a. Nama (varchar 50)</li> <li>b. NIM (varchar 25)</li> <li>c. Vektor V (double)</li> </ul>

#### 17. Skenario Use Case Melihat Hasil Perangkingan

Skenario melihat hasil perangkingan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat hasil perangkingan. Aktor dapat melihat hasil seleksi berupa perangkingan. Aktor yang dapat melihat hasil perangkingan yaitu admin penerimaan dan pelatih. Pelatih dapat mengubah status pendaftar lolos seleksi. Proses untuk melihat hasil perangkingan, klik menu lihat hasil perangkingan pada *dashboard* admin penerimaan. Sistem akan

menamoiikan halaman perankingan berupa tabel. Skenario melihat hasil perankingan dapat dilihat pada lampiran A Tabel 16.

#### 18. Skenario *Use Case* Cetak Pendaftar Disetujui

Skenario cetak pendaftar disetujui merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan mencetak pendaftar disetujui. Aktor dapat mencetak pendaftar disetujui yaitu admin penerimaan. Proses untuk mencetak pendaftar disetujui, klik menu lihat ranking pada halaman *dashboard* admin penerimaan. Sistem akan menampilkan halaman hasil perankingan berupa tabel kemudian klik *button* "download to Pdf" unuk mendownload berkas dan di cetak. Skenario cetak pendaftar disetujui dapat dilihat pada lampiran A Tabel 17.

#### 19. Skenario *Use Case* Persetujuan Pendaftar Lolos Seleksi

Skenario persetujuan pendaftar lolos seleksi merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan menyetujui pendaftar yang lolos seleksi. Aktor dapat mengubah status / menyetujui pendaftar lolos seleksi. Aktor dapat menyetujui / mengubah status yaitu pelatih. Proses untuk menyetujui pendaftar yang lolos seleksi, klik menu lihat ranking pada halaman *dashboard* pelatih. Sistem akan menampilkan halaman hasil perankingan berupa tabel kemudian klik *button* edit unuk mengubah status menjadi disetujui. Klik *button submit* untuk menyimpan perubahan. Skenario persetujuan pendaftar lolos seleksi dapat dilihat pada lampiran A Tabel 18.

#### 20. Skenario *Use Case* Help

Skenario help merupakan alur yang menjelaskan aksi aktor dan reaksi sistem informasi pada saat aktor akan melihat hak ases masing-masing pengguna. Aktor dapat melihat hak akses masing-masing. Fitur help ini terdapat di semua menu aktor. Proses untuk melihat fitur help, klik menu help pada masing-masing menu aktor. Sistem akan menampilkan halaman help berupa penjelasan mengenai hak akses sistem. Skenario help dapat dilihat pada lampiran A Tabel 17.

## 21. Skenario *Use Case* logout

Skenario *use case* logout merupakan alur dari aksi aktor dan reaksi sistem jika akan keluar sistem. Penjelasan urutan aksi aktor dan reaksi sistem pada skenario utama dan skenario alternatif *use case* keluar terdapat pada lampiran A Tabel 18.

### 4.3.4 *Sequence Diagram*

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

#### 1. *Sequence Diagram* Masuk Sistem

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

#### 2. *Sequence Diagram* Menambah Data Event

Pada *sequence diagram* menambah data event merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses menambah data event baru terdapat beberapa interaksi antar kelas yang digunakan. Masing - masing kelas akan ditampilkan secara visual dengan gambar.

Kelas yang berperan dalam *sequence diagram* menambah data event adalah *v\_homeAdmin*, *v\_tambah\_event*, *web*, *c\_admin*, *helpers*, dan *m\_event*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 2.

### 3. *Sequence Diagram* Melihat Data Event

Pada *sequence diagram* melihat data event terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat data event adalah *v\_homeAdmin*, *v\_lihat\_event*, *web*, *c\_admin*, *helpers*, dan *m\_event*. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 3.

### 4. *Sequence Diagram* Mengubah Data Event

Pada *sequence diagram* mengubah data event terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* mengubah data event adalah *v\_homeAdmin*, *v\_lihat\_event*, *v\_edit\_event*, *web*, *c\_admin*, *helpers*, dan *m\_event*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 4.

### 5. *Sequence Diagram* Menambah Galeri Kegiatan

Pada *sequence diagram* menambah galeri kegiatan merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan

proses menambah galeri kegiatan baru terdapat beberapa interaksi antar kelas yang digunakan. Masing - masing kelas akan ditampilkan secara visual dengan gambar. Kelas yang berperan dalam *sequence diagram* menambah galeri kegiatan adalah *v\_homeAdmin*, *v\_tambah\_galeri*, *web*, *c\_admin*, *helpers*, dan *m\_galeri*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 5.

#### 6. *Sequence Diagram* Melihat Galeri Kegiatan

Pada *sequence diagram* mengubah data event terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat galeri kegiatan adalah *v\_homeAdmin*, *v\_lihat\_galeri*, *web*, *c\_admin*, *helpers*, dan *m\_galeri*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 6.

#### 7. *Sequence Diagram* Mengubah Galeri Kegiatan

Pada *sequence diagram* mengubah data event terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* mengubah galeri kegiatan adalah *v\_homeAdmin*, *v\_lihat\_galeri*, *v\_edit\_galeri*, *web*, *c\_admin*, *helpers*, dan *m\_galeri*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas

*model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 7.

#### 8. *Sequence Diagram* Pendaftaran

Pada *sequence diagram* pendaftaran merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses registrasi pendaftaran baru terdapat beberapa interaksi antar kelas yang digunakan. Masing - masing kelas akan ditampilkan secara visual dengan gambar. Kelas yang berperan dalam *sequence diagram* pendaftaran adalah *v\_landing*, *v\_tambah\_registrasi*, *web*, *c\_penerimaan*, *helpers*, dan *m\_siswa*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 8.

#### 9. *Sequence Diagram* Melihat Data Lolos Tahun Sebelumnya

Pada *sequence diagram* melihat data lolos tahun sebelumnya terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat data lolos tahun sebelumnya adalah *v\_homePendaftar*, *v\_lihat\_pendaftar\_sebelumnya*, *web*, *c\_admin*, *helpers*, dan *m\_pendaftar*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 9.

#### 10. *Sequence Diagram* Lihat Data Pendaftar

Pada *sequence diagram* melihat data pendaftar terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat data pendaftar adalah *v\_homePenerimaan*, *v\_lihat\_pendaftar\_baru*, *web*, *c\_penerimaan*, *helpers*, dan *m\_pendaftar*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 10.

#### 11. *Sequence Diagram* input Nilai Hasil Seleksi

Pada *sequence diagram* input nilai hasil seleksi terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* input nilai hasil seleksi adalah *v\_homePenerimaan*, *v\_lihat\_pendaftar\_baru*, *v\_tambahNilai*, *web*, *c\_penerimaan*, *helpers*, dan *m\_pendaftar*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 11.

#### 12. *Sequence Diagram* Menambah Data Penerimaan

Pada *sequence diagram* menambah data penerimaan merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses menambah data penerimaan baru terdapat beberapa interaksi antar kelas yang digunakan. Masing - masing kelas akan ditampilkan secara visual dengan gambar. Kelas yang berperan dalam *sequence diagram* menambah data

penerimaan adalah *v\_homePenerimaan*, *v\_tambah\_penerimaan*, *web*, *c\_penerimaan*, *helpers*, dan *m\_penerimaan*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 12.

### 13. *Sequence Diagram* Melihat Data Penerimaan

Pada *sequence diagram* melihat data penerimaan terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat data penerimaan adalah *v\_homePenerimaan*, *v\_lihatPenerimaan*, *web*, *c\_penerimaan*, *helpers*, dan *m\_penerimaan*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 13.

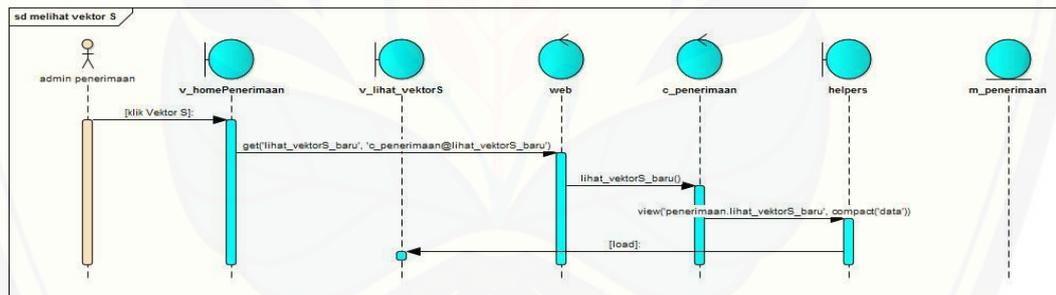
### 14. *Sequence Diagram* Mengubah Data Penerimaan

Pada *sequence diagram* mengubah data penerimaan terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* mengubah data penerimaan adalah *v\_homePenerimaan*, *v\_lihatPenerimaan*, *v\_editPenerimaan*, *web*, *c\_penerimaan*, *helpers*, dan *m\_penerimaan*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil

maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 14.

#### 15. Sequence Diagram Melihat Vektor S

Pada *sequence diagram* melihat Vektor S terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat vektor S adalah *v\_homePenerimaan*, *v\_lihat\_vektorS*, *web*, *c\_penerimaan*, *helpers*, dan *m\_penerimaan*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada Gambar 4.3.

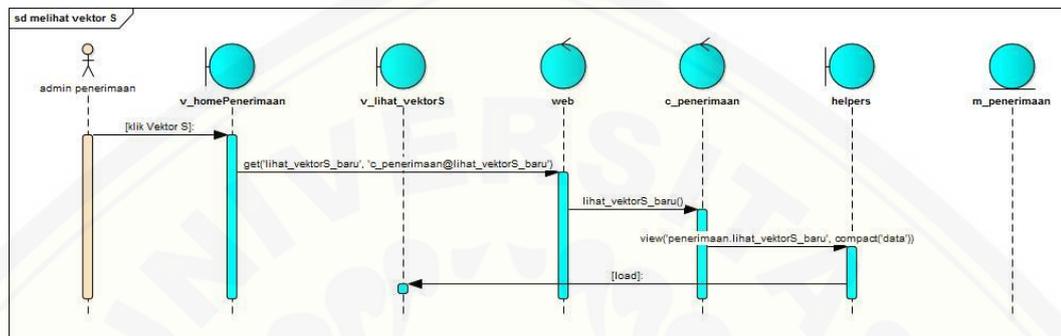


Gambar 4. 3 Sequence Diagram melihat vektor S

#### 16. Sequence Diagram Melihat Vektor V

Pada *sequence diagram* melihat Vektor V terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat vektor S adalah *v\_homePenerimaan*, *v\_lihat\_vektor* Pada *sequence diagram* melihat Vektor S terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* melihat vektor V adalah *v\_homePenerimaan*, *v\_lihat\_vektorS*, *web*, *c\_penerimaan*, *helpers*, dan *m\_penerimaan*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat

pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat Gambar 4.4.



Gambar 4. 4 Sequence Diagram melihat vektor V

#### 17. Sequence Diagram Lihat Hasil Perangkingan

Pada *sequence diagram* lihat hasil perangkingan terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* lihat hasil perangkingan adalah *v\_homePenerimaan*, *v\_hasil\_perangkingan\_baru*, *web*, *c\_penerimaan*, *helpers*, dan *m\_penerimaan*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 15.

#### 18. Sequence Diagram Persetujuan Pendaftar Lolos Seleksi

Pada *sequence diagram* persetujuan pendaftar lolos seleksi terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* persetujuan pendaftar lolos seleksi adalah *v\_homePenerimaan*, *v\_persetujuan\_pendafatr*, *web*, *c\_penerimaan*, *helpers*, dan *m\_penerimaan*. Proses

dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 16.

#### 19. *Sequence Diagram* Cetak Pendaftar Disetujui

Pada *sequence diagram* cetak pendaftar disetujui terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* cetak pendaftar disetujui adalah *v\_homePenerimaan*, *v\_hasil\_perangkingan\_baru*, *v\_cetakPendaftar*, *web*, *c\_penerimaan*, *helpers*, dan *m\_penerimaan*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 17.

#### 20. *Sequence Diagram* Help

Pada *sequence diagram* help terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang berperan dalam *sequence diagram* help adalah *v\_homeHelp*, *v\_help*, *web*, *c\_help*, *helpers*, dan *m\_help*. Proses dari aliran *sequence diagram* diawali ketika aktor melakukan perintah pada suatu *view*, selanjutnya *view* akan mengirim perintah dengan memanggil method yang terdapat pada *controller* dimana *controller* akan melanjutkan memanggil method pada kelas yang berhubungan baik dengan kelas *model* jika terdapat *query* yang diambil maupun kelas *view* jika akan menampilkan sebuah *view* selanjutnya. Interaksi antar kelas

dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 18.

#### 21. *Sequence Diagram* Keluar Sistem

Pada *sequence diagram* keluar merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses keluar dari sistem. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 19.

#### 4.3.5 *Activity Diagram*

*Activity diagram* menggambarkan alur aktivitas pada sistem pada sistem informasi seleksi penerimaan anggota paduan suara mahasiswa jember dengan mengimplementasikan metode *Weighted Product*.

##### 1. *Activity Diagram* Login

*Activity diagram* masuk menggambarkan tentang proses yang terjadi saat aktor melakukan masuk. Aktor memasukkan *username* dan *password* lalu mengklik tombol “*login*”. Kemudian sistem informasi akan melakukan pengecekan apakah inputan kosong atau *username* dan *password* salah, jika benar maka sistem informasi akan menampilkan halaman form Masuk. Namun jika tidak, maka sistem informasi akan menampilkan halaman *dashboard* yang disesuaikan dengan hak akses *user* tersebut. *Activity diagram* masuk dapat dilihat pada lampiran C Gambar 1.

##### 2. *Activity Diagram* Menambah Data Event

*Activity diagram* menambah data event menggambarkan tentang proses yang terjadi saat aktor menambah data event. Aktor yang dapat menambah data event adalah Admin *Website*. Proses untuk menambah data event, admin mengklik menu info event kemudian klik submenu menambah data event. Sistem akan menampilkan halaman tambah data event berupa form. Aktor menambah data event kemudian klik *submit* untuk menyimpan data. Sistem memeriksa data, apabila terdapat data yang kosong maka sistem akan menampilkan pesan “harap isi bidang

ini” dan tetap menampilkan halaman form. Namun apabila aktor mengisi data dengan tepat, maka data berhasil disimpan dan ditambahkan. N *Activity diagram* menambah data event dapat dilihat pada lampiran C Gambar 2.

### 3. *Activity Diagram* Melihat Data Event

*Activity diagram* melihat data event menggambarkan tentang proses yang terjadi saat aktor melihat data event. Aktor yang dapat melihat data event Admin. Proses untuk melihat data event, aktor admin dapat mengklik menu info event kemudian klik submenu lihat data event. Sistem akan menampilkan halaman lihat data event berupa tabel. *Activity diagram* melihat data event dapat dilihat pada lampiran C Gambar 3.

### 4. *Activity Diagram* Mengubah Data Event

*Activity diagram* mengubah data event menggambarkan tentang proses yang terjadi saat aktor mengubah data event. Aktor yang dapat mengubah data event Admin. Admin dapat melihat dan mengubah data event. Proses untuk mengubah data event, admin dapat mengklik menu info event kemudian klik submenu lihat info event. Sistem akan menampilkan halaman lihat data event berupa tabel dengan *action button* edit. Klik *button edit* untuk mengubah data event. Sistem akan menampilkan form ubah data event. Admin mengubah data event kemudian klik *button submit* untuk menyimpan data. *Activity diagram* mengubah data event dapat dilihat pada lampiran C Gambar 4.

### 5. *Activity Diagram* Menambah Galeri Kegiatan

*Activity diagram* menambah galeri kegiatan menggambarkan tentang proses yang terjadi saat aktor menambah galeri kegiatan. Aktor yang dapat menambah data galeri kegiatan adalah Admin. Admin dapat melihat dan menambah galeri kegiatan. Proses untuk menambah galeri kegiatan, admin dapat mengklik menu galeri kegiatan kemudian klik submenu tambah galeri kegiatan. Sistem akan menampilkan halaman tambah galeri kegiatan berupa form. Aktor menambah data galeri kegiatan kemudian klik *submit* untuk menyimpan data. Sistem memeriksa

data, apabila terdapat data yang kosong maka sistem akan menampilkan pesan “harap isi bidang ini” dan tetap menampilkan halaman form. Namun apabila aktor mengisi data dengan tepat dan benar, maka data berhasil disimpan dan ditambahkan. *Activity diagram* menambah galeri kegiatan dapat dilihat pada lampiran C Gambar 5.

#### 6. *Activity Diagram* Melihat Galeri Kegiatan

*Activity diagram* melihat galeri kegiatan menggambarkan tentang proses yang terjadi saat aktor melihat galeri kegiatan. Aktor yang dapat melihat galeri kegiatan adalah Admin. Proses untuk melihat galeri kegiatan, aktor admin dapat mengklik menu galeri kegiatan kemudian klik submenu lihat galeri kegiatan. Sistem akan menampilkan halaman lihat galeri kegiatan berupa tabel. *Activity diagram* melihat galeri kegiatan dapat dilihat pada lampiran C Gambar 6.

#### 7. *Activity Diagram* Mengubah Galeri Kegiatan

*Activity diagram* mengubah galeri kegiatan menggambarkan tentang proses yang terjadi saat aktor mengubah galeri kegiatan. Aktor yang dapat mengubah galeri kegiatan adalah Admin *website*. Admin *website* dapat melihat dan mengubah galeri kegiatan. Proses untuk mengubah data galeri, admin dapat mengklik menu galeri kegiatan kemudian klik submenu lihat galeri kegiatan. Sistem akan menampilkan halaman lihat galeri kegiatan berupa tabel dengan *action button* edit. Klik *button edit* untuk mengubah data galeri kegiatan. Sistem akan menampilkan form ubah galeri kegiatan. Admin mengubah galeri kegiatan kemudian klik *button submit* untuk menyimpan data. *Activity diagram* mengubah galeri kegiatan dapat dilihat pada lampiran C Gambar 7.

#### 8. *Activity Diagram* Pendaftaran

*Activity diagram* pendaftaran menggambarkan tentang proses yang terjadi saat aktor melakukan registrasi pendaftaran. Aktor yang dapat melakukan registrasi pendaftaran adalah mahasiswa (pendaftar). mahasiswa dapat melakukan pendaftaran. Mahasiswa dapat melakukan proses pendaftaran dengan cara

mengklik menu registrasi pada halaman beranda *website*. Kemudian mengisi form registrasi. apabila telah selesai, klik *button submit* untuk menyimpan data. Apabila terdapat form kosong maka sistem akan menampilkan pesan “harap isi bidang ini”. Namun apabila aktor mengisi form dengan tepat dan benar, maka data berhasil di tambahkan dan disimpan. *Activity diagram* pendaftaran dapat dilihat pada lampiran C Gambar 8.

#### 9. *Activity Diagram* Melihat Data Lolos Tahun Sebelumnya

*Activity diagram* melihat data lolos tahun sebelumnya menggambarkan tentang proses yang terjadi saat aktor melihat data pendaftar lolos tahun sebelumnya. Aktor yang dapat melihat data pendaftar lolos tahun sebelumnya adalah Admin penerimaan. Proses untuk melihat data lolos tahun sebelumnya, klik menu lihat semua vektor  $v$  pada *dasboard*. Sistem akan menampilkan halaman data lolos tahun sebelumnya berupa tabel. *Activity diagram* melihat data lolos tahun sebelumnya dapat dilihat pada lampiran C Gambar 9.

#### 10. *Activity Diagram* Lihat Data Pendaftar

*Activity diagram* melihat data pendaftar menggambarkan tentang proses yang terjadi saat aktor melihat data pendaftar. Aktor yang dapat melihat data pendaftar adalah Admin penerimaan. Proses untuk melihat data pendaftar, admin penerimaan dapat mengklik menu lihat data pendaftar pada halaman home admin. Sistem akan menampilkan halaman data pendaftar berupa tabel. *Activity diagram* melihat data pendaftar dapat dilihat pada lampiran C Gambar 10.

#### 11. *Activity Diagram* Input Nilai Hasil Seleksi

*Activity diagram* input nilai hasil seleksi menggambarkan tentang proses yang terjadi saat aktor menginputkan nilai hasil seleksi. Aktor yang dapat menginputkan nilai hasil seleksi adalah Admin penerimaan. Proses untuk menginputkan nilai hasil seleksi, admin penerimaan dapat mengklik menu lihat data pendaftar pada home admin penerimaan. Sistem akan menampilkan halaman lihat data pendaftar berupa tabel dan *action button edit*. Klik *button edit* untuk menginputkan nilai kemudian

klik *submit* untuk menyimpan data. *Activity diagram* input nilai hasil seleksi dapat dilihat pada lampiran C Gambar 11.

#### 12. *Activity Diagram* Menambah Data Penerimaan

*Activity diagram* menambah data penerimaan menggambarkan tentang proses yang terjadi saat aktor menambah data penerimaan. Aktor yang dapat menambah data penerimaan adalah Admin *website*. Admin *website* dapat melihat dan menambah data penerimaan. Proses untuk menambah data penerimaan, admin penerimaan dapat mengklik menu data penerimaan kemudian klik submenu tambah data penerimaan. Sistem akan menampilkan halaman tambah penerimaan berupa form. Admin penerimaan menambahkan data kemudian klik *button submit* untuk menyimpan data yang telah ditambahkan. Apabila terdapat form kosong maka sistem akan menampilkan pesan "harap isi bidang ini". *Activity diagram* menambah data penerimaan dapat dilihat pada lampiran C Gambar 12.

#### 13. *Activity Diagram* Melihat Data Penerimaan

*Activity diagram* melihat data penerimaan menggambarkan tentang proses yang terjadi saat aktor melihat data penerimaan. Aktor yang dapat melihat data penerimaan adalah Admin penerimaan. Proses untuk melihat data penerimaan, admin penerimaan dapat mengklik menu data penerimaan kemudian klik submenu lihat data penerimaan. Sistem akan menampilkan halaman lihat data penerimaan berupa tabel. *Activity diagram* melihat penerimaan dapat dilihat pada lampiran C Gambar 13.

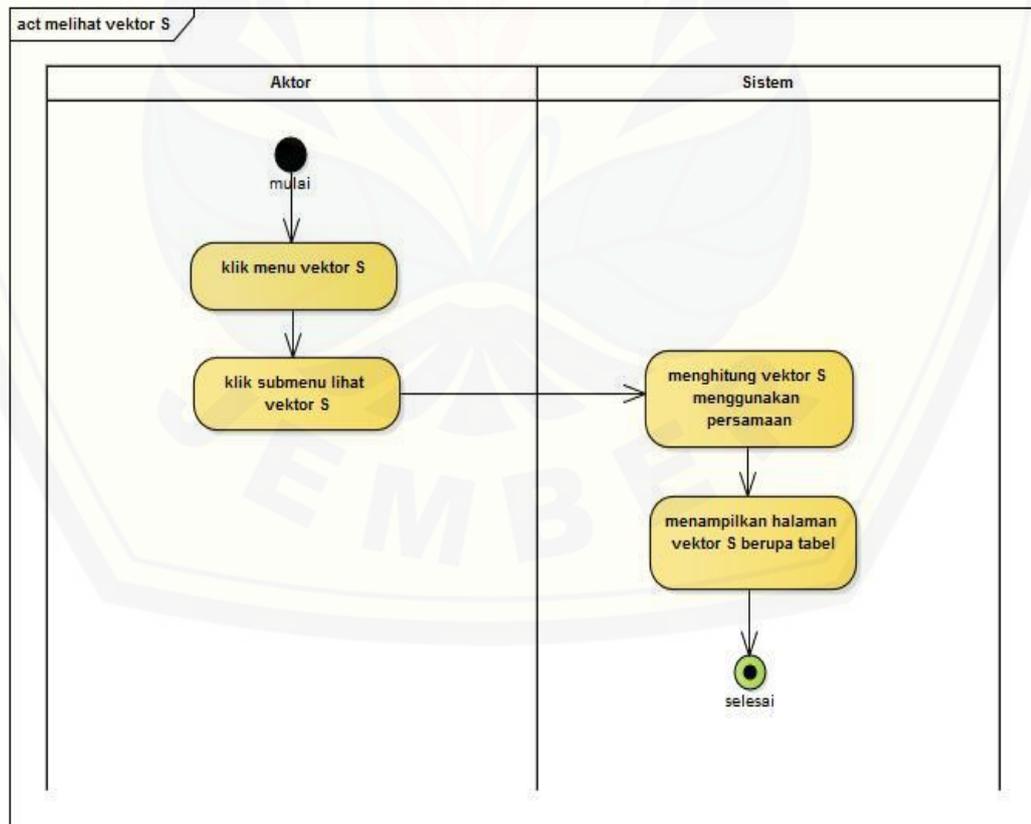
#### 14. *Activity Diagram* Mengubah Data Penerimaan

*Activity diagram* mengubah data penerimaan menggambarkan tentang proses yang terjadi saat aktor akan mengubah data penerimaan. Aktor yang dapat mengubah data penerimaan adalah Admin penerimaan. Admin penerimaan dapat melihat dan mengubah data penerimaan. Proses untuk mengubah data penerimaan, admin penerimaan dapat mengklik menu data penerimaan kemudian klik submenu lihat data penerimaan. Sistem akan menampilkan halaman lihat data penerimaan

berupa tabel dan *action button edit*. klik *button edit* untuk mengubah data penerimaan. Sistem akan menampilkan form ubah data penerimaan. Admin penerimaan mengubah data kemudian klik *button submit* untuk menyimpan data. *Activity diagram* mengubah data penerimaan dapat dilihat pada lampiran C Gambar 14.

#### 15. Activity Diagram Melihat Vektor S

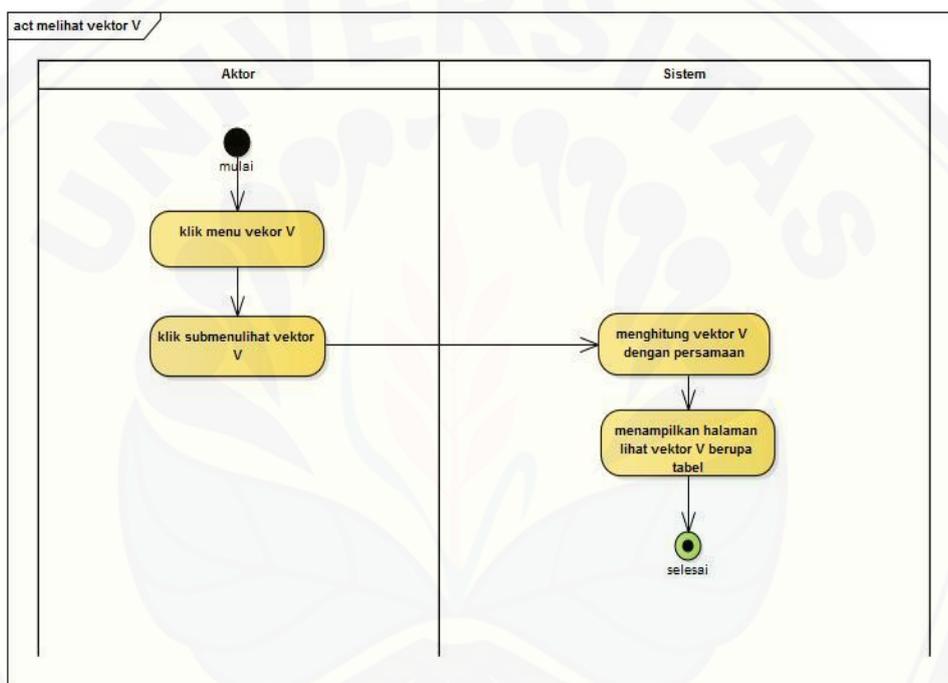
*Activity diagram* melihat vektor S menggambarkan tentang proses yang terjadi saat aktor melihat hasil perhitungan nilai prefensi dengan vektor S. Aktor yang dapat melihat hasil perhitungan nilai prefensi dengan vektor S adalah Admin penerimaan. Proses untuk melihat vektor S, admin penerimaan dapat mengklik menu vektor S kemudian klik submenu lihat vektor S. Sistem akan menampilkan hasil perhitungan nilai prefensi vektor S berupa tabel. *Activity diagram* melihat vektor S dapat dilihat pada Gambar 4.5.



Gambar 4. 5 Activity Diagram melihat vektor S

#### 16. Activity Diagram Melihat Vektor V

*Activity diagram* melihat vektor V menggambarkan tentang proses yang terjadi saat aktor melihat hasil perhitungan nilai prefensi relatif setiap alternatif (vektor V). Aktor yang dapat melihat hasil perhitungan nilai prefensi relatif setiap alternatif (vektor V) adalah Admin penerimaan. Proses untuk melihat vektor V, admin penerimaan dapat mengklik menu vektor V kemudian klik submenu lihat vektor V. Sistem akan menampilkan hasil perhitungan nilai prefensi setiap alternatif vektor V berupa tabel. *Activity diagram* melihat vektor V dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Activity Diagram melihat vektor V

#### 17. Activity Diagram Melihat Hasil Perangkingan

*Activity diagram* melihat hasil perangkingan menggambarkan tentang proses yang terjadi saat aktor melihat hasil perangkingan. Aktor yang dapat melihat hasil perangkingan adalah Admin penerimaan dan pelatih. Proses untuk melihat hasil perangkingan, admin penerimaan dapat mengklik menu lihat hasil rangking. Sistem akan menampilkan hasil perangkingan pendaftar berupa tabel. *Activity diagram* melihat hasil perangkingan dapat dilihat pada lampiran C Gambar 15.

#### 18. *Activity Diagram* Persetujuan Pendaftar Lolos Seleksi

*Activity diagram* persetujuan pendaftar lolos seleksi menggambarkan tentang proses yang terjadi saat aktor menyetujui / mengubah status pendaftar lolos seleksi. Aktor yang dapat menyetujui pendaftar lolos seleksi adalah pelatih. Proses untuk mengubah status, pelatih dapat mengklik menu lihat hasil rangking yang ada pada home pelatih. Sistem akan menampilkan hasil perangkingan berupa tabel dan *action button edit*. Klik *button edit*, sistem akan menampilkan form ubah status menjadi terverifikasi. Klik *button submit* untuk menyimpan data. *Activity diagram* persetujuan pendaftar lolos seleksi dapat dilihat pada lampiran C Gambar 16.

#### 19. *Activity Diagram* Cetak Pendaftar Disetujui

*Activity diagram* cetak pendaftar disetujui menggambarkan tentang proses yang terjadi saat aktor akan mencetak pendaftar yang telah disetujui oleh pelatih. Aktor yang dapat mencetak pendaftar disetujui adalah Admin penerimaan. Proses untuk mencetak pendaftar disetujui, admin penerimaan dapat mengklik menu lihat rangking. Sistem akan menampilkan hasil perangkingan berupa tabel dan *button* "download to Pdf". Klik *button* "download to Pdf" untuk mendownload dan mencetak pendaftar yang telah disetujui oleh pelatih. *Activity diagram* cetak pendaftar disetujui dapat dilihat pada lampiran C Gambar 17.

#### 20. *Activity Diagram* Help

*Activity diagram* help menggambarkan tentang proses yang terjadi saat aktor melihat hak akses dari masing-masing pengguna (help). Aktor yang dapat melihat help adalah Admin penerimaan, Admin *website*, dan pelatih. Proses untuk melihat fitur help, klik menu help pada home masing-masing aktor. Sistem akan menampilkan halaman help berupa gambar yang menjelaskan hak akses dari masing-masing aktor. *Activity diagram* help dapat dilihat pada lampiran C Gambar 18.

## 21. Activity Diagram Logout

*Activity diagram* logout menggambarkan tentang proses yang terjadi saat aktor ingin keluar dari sistem informasi. Proses untuk keluar dari sistem, masing-masing aktor dapat mengklik *button logout* pada pojok kanan atas. Sistem akan menampilkan halaman beranda awal. *Activity diagram* logout dapat dilihat pada lampiran C Gambar 19.

### 4.3.6 Class diagram

*Class diagram* merupakan diagram untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem informasi yang nantinya digunakan. Dalam paradigma OOP (*Object Oriented Programming*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller*. *Class diagram* yang digunakan Sistem Informasi Seleksi Penerimaan Calon Anggota Paduan Suara Universitas Jember dapat dilihat pada Gambar 4.9.

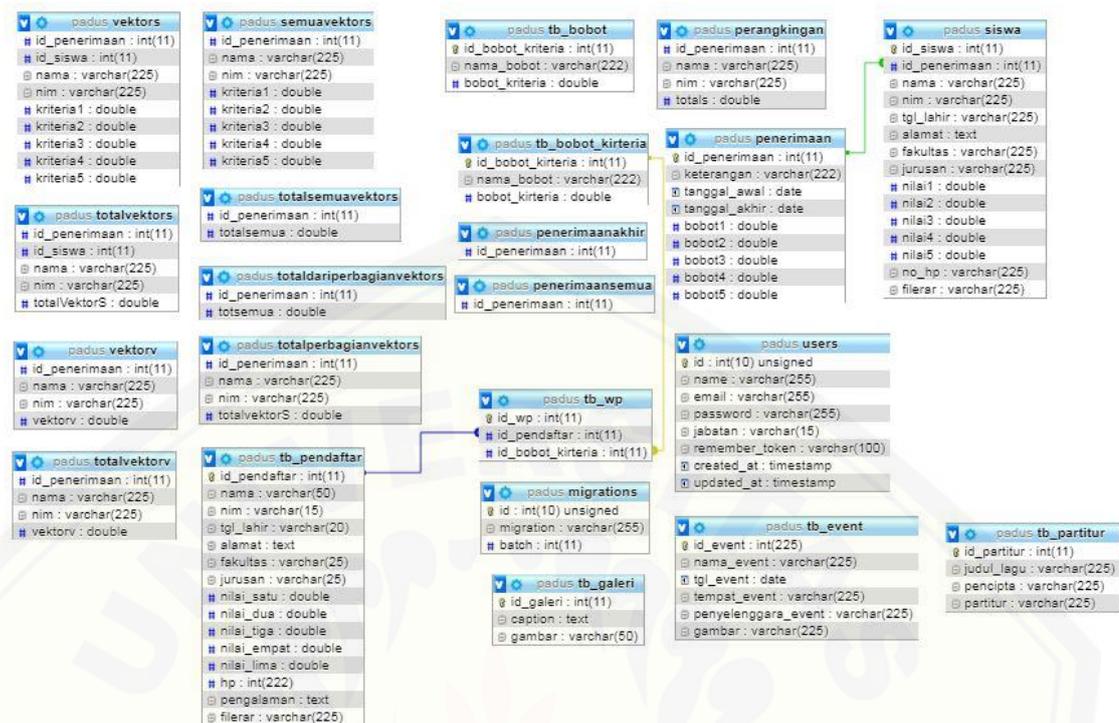
### 4.3.7 Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur *database* yang digunakan dalam pembangunan sistem. ERD pada sistem informasi seleksi penerimaan anggota paduan suara mahasiswa jember dengan mengimplementasikan metode *weighted product* dapat dilihat pada Gambar 4.10.

## 4.4 Implementasi

Tahap implementasi adalah tahap pembuatan program dengan cara pengkodean atau koding sesuai dengan desain sistem yang telah dibuat. Pengkodean dibuat dengan menggunakan bahasa *Page Hyper Text Pre-Processor* (PHP) pada *frameword Codeigniter*. Manajemen database dibuat dengan bantuan *Database Management System MySQL*. Pengkodean dibuat guna mendapatkan sistem yang dapat dioperasikan oleh pengguna.





Gambar 4. 8 Entity Relationship Diagram (ERD)

Pada proses pengkodean mengimplementasikan metode *Weighted Product* pada fitur perangkingan. Fitur tersebut terdapat pada *view* di *database* serta kelas *c\_penerimaan*, *m\_mahasiswa*, *m\_penerimaan*, *m\_vektor\_v*, dan *m\_vektor\_v\_akhir*.

#### 1. Kelas *c\_penerimaan*

Kelas *c\_penerimaan* berfungsi untuk mengatur jalannya data antara model dengan tampilan. Pada kelas terbut terdapat proses untuk menampilkan hasil perangkingan metode *Weighted Product* yang telah dibuat pada *table view* di *database*. Penulisan kode program fitur perangkingan pada kelas *c\_penerimaan* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Kode Program Kelas c\_penerimaan

```

1 public function prosesTambahPenerimaan(Request $request){
2
3     $insert = ([
4         'keterangan' => $request->keterangan,
5         'tanggal_awal' => $request->tanggal_awal,
6         'tanggal_akhir' => $request->tanggal_akhir,
7         'bobot1' => $request->bobot1/20,
8         'bobot2' => $request->bobot2/20,
9         'bobot3' => $request->bobot3/20,
10        'bobot4' => $request->bobot4/20,
11        'bobot5' => $request->bobot5/20,
12
13    ]);
14    m_penerimaan::create($insert);
15    return redirect('/lihatPenerimaan');
16
17 }
18
19 public function lihatVektorS(){
20     $data['vektor_s'] = DB::select('select * from vektor_s');
21     return view('penerimaan.v_lihat_vektors', compact('data'));
22 }
23
24 public function lihatVektorV(){
25     $data = DB::table('vektor_v')->paginate(5);
26
27
28
29
30
31

```

Tabel 4.5 merupakan potongan kode program dari kelas c\_penerimaan. Pada kode program baris 1 – 17 merupakan fungsi prosesTambahPenerimaan() untuk menambah bobot penerimaan. Baris 19 – 23 merupakan fungsi lihatVektorS() untuk menampilkan hasil perhitungan nilai prefensi vektor S. Baris 25 – 31 merupakan fungsi lihatVektorV() untuk menampilkan hasil perhitungan nilai prefensi setiap alternatif dengan vektor V. Baris 33 – 38 merupakan fungsi lihatVektorvPenerimaanAkhir() untuk melihat hasil perhitungan vektor V pada penerimaan gelombang terakhir yang belum terangkai. Sedangkan baris 40 – 48

merupakan fungsi `lihatPerangkingan ()` untuk menampilkan hasil perangkingan nilai akhir.

## 2. Kelas `m_mahasiswa`

Kelas `m_mahasiswa` merupakan kelas yang berfungsi sebagai perantara interaksi antara *database* dan sistem. Penulisan kode program kelas `m_mahasiswa` dapat dilihat pada Tabel 4.6 berikut.

Tabel 4. 6 Kode program kelas `m_mahasiswa`

1	<code>class m_mahasiswa extends Model</code>
2	<code>{</code>
3	<code>protected \$table = 'tb_mahasiswa';</code>
4	<code>protected \$primaryKey = 'id_mahasiswa';</code>
5	<code>protected \$fillable = ['nama', 'nim','tanggal_lahir', 'alamat','fakultas',</code>
6	<code>'kontak', 'status', 'nilai1', 'nilai2', 'nilai3', 'nilai4', 'nilai5', 'filarar' ];</code>
7	
8	<code>public \$timestamps = false;</code>
9	<code>}</code>
10	

Tabel 4.6 merupakan kode program kelas `m_mahasiswa`. Pada baris 1 *class* `m_penerimaan` merupakan anak *class* dari *class* `Model`. Baris 3 – 7 merupakan definisi *koding* untuk penampungan data dari `tb_mahasiswa` yang ada pada *database*. Sedangkan baris 9 berfungsi untuk mengatur *timestamp* data menjadi *false*.

## 3. Kelas `m_penerimaan`

Kelas `m_penerimaan` merupakan kelas yang berfungsi sebagai perantara interaksi antara *database* dan sistem. Penulisan kode program kelas `m_penerimaan` dapat dilihat pada tabel 4.7 berikut.

Tabel 4. 7 Kode program kelas `m_penerimaan`

1	<code>class m_penerimaan extends Model</code>
2	<code>{</code>
3	<code>protected \$table = 'tb_penerimaan';</code>
4	<code>protected \$primaryKey = 'id_penerimaan';</code>
5	

6	protected \$fillable = ['keterangan', 'tanggal_awal','tanggal_akhir',
7	'bobot1', 'bobot2', 'bobot3', 'bobot4', 'bobot5' ];
8	
9	public \$timestamps = false;
	}

Tabel 4.7 merupakan kode program kelas `m_mahasiswa`. Pada baris 1 *class* `m_penerimaan` merupakan anak *class* dari *class* `Model`. Baris 3 – 7 merupakan definisi *koding* untuk penampungan data dari `tb_penerimaan` yang ada pada database. Sedangkan baris 9 berfungsi untuk mengatur *timestamp* data menjadi *false*.

#### 4. Kelas `m_vektor_v`

Kelas `m_vektor_v` merupakan kelas yang berfungsi sebagai perantara interaksi antara *database* dan sistem. Penulisan kode program kelas `m_vektor_v` dapat dilihat pada Tabel 4.8 berikut.

Tabel 4. 8 Kode program kelas `m_vektor_v`

1	class <code>m_vektor_v</code> extends <code>Model</code>
2	{
3	protected <code>\$table</code> = ' <code>vektor_v</code> ';
4	protected <code>\$primaryKey</code> = ' <code>id_penerimaan</code> ';
5	protected <code>\$fillable</code> = [' <code>nama</code> ', ' <code>nim</code> ',' <code>fakultas</code> ', ' <code>keterangan</code> ', ' <code>vektor_v</code> ' ];
6	
7	public <code>\$timestamps</code> = <code>false</code> ;
8	
9	
10	
11	
12	

Tabel 4.8 merupakan kode program kelas `m_vektor_v`. Pada baris 1 *class* `m_vektor_v` merupakan anak *class* dari *class* `Model`. Baris 3 – 7 merupakan definisi *koding* untuk penampungan data dari *class view* `vektor_v` yang ada pada database. Sedangkan baris 9 berfungsi untuk mengatur *timestamp* data menjadi *false*.

#### 5. Kelas `m_vektor_v_akhir`

Kelas `m_vektor_v_akhir` merupakan kelas yang berfungsi sebagai perantara interaksi antara *database* dan sistem. Penulisan kode program kelas `m_vektor_v_akhir` dapat dilihat pada tabel 4.9 berikut.

Tabel 4. 9 Kode program kelas `m_vektor_v_akhir`

1	<code>class m_vektor_v_akhir extends Model</code>
2	<code>{</code>
3	<code>protected \$table = 'vektor_v_akhir';</code>
4	<code>protected \$primaryKey = 'id_mahasiswa';</code>
5	<code>protected \$fillable = ['id_penerimaan','nama', 'nim','fakultas',</code>
6	<code>'keterangan', 'vektor_v' ];</code>
7	<code></code>
8	<code>public \$timestamps = false;</code>
9	<code>public function rengkings()</code>
10	<code>{</code>
11	<code>return \$this-&gt;hasMany('App\m_perengkingan');</code>
12	<code>}</code>
13	<code>}</code>

Tabel 4.9 merupakan kode program kelas `m_vektor_v_akhir`. Pada baris 1 `class m_vektor_v_akhir` merupakan anak *class* dari *class Model*. Baris 3 – 7 merupakan definisi *koding* untuk penampungan data dari *class view* `vektor_v_akhir` yang ada pada *database*. Sedangkan baris 9 berfungsi untuk mengatur *timestamp* data menjadi *false*.

#### 6. *View*

*View* merupakan sebuah tabel yang berasal dari tabel lain atau gabungan dari beberapa tabel. Tujuan penggunaan *view* adalah untuk mempermudah penulisan *query* dan mempercepat proses menampilkan data. *View* perangkaian *Weighted Product* merupakan kumpulan dari beberapa *view* yang digunakan untuk mendapatkan perangkaian dengan menggunakan metode *Weighted Product*. Berikut adalah *query* yang terdapat pada *database* yang digunakan untuk menyelesaikan metode *Weighted Product*.

a. *View ambilidakhir*

*View ambilidakhir* digunakan untuk mendapatkan id\_penerimaan terbaru. *Query* dari *view ambilidakhir* dapat dilihat pada Tabel 4.10 berikut.

Tabel 4. 10 query dari ambilidakhir

<i>view ambilidakhir</i>	
select `padusunej2`.`tb_penerimaan`.`id_penerimaan` AS `id_penerimaan` from `padusunej2`.`tb_penerimaan` `padusunej2`.`tb_penerimaan`.`id_penerimaan` desc limit 1	order by

b. *View vektor\_s*

*View vektor\_s* digunakan untuk mendapatkan nilai hasil pangkat bobot dari masing-masing kriteria. *Query* dari *view vektor\_s* dapat dilihat pada Tabel 4.11 berikut.

Tabel 4. 11 query dari view vektor\_s

<i>view vektor_s</i>	
select `tm`.`id_mahasiswa` AS `id_mahasiswa`,`tp`.`id_penerimaan` AS `id_penerimaan`,`tm`.`nama` AS `nama`,`tm`.`nim` AS `nim`,`tm`.`fakultas` AS `fakultas`,`tp`.`keterangan` AS `keterangan`,`tm`.`status` AS `status`,pow(`tm`.`nilai1`,`tp`.`bobot1`) AS `kriteria1`,pow(`tm`.`nilai2`,`tp`.`bobot2`) AS `kriteria2`,pow(`tm`.`nilai3`,`tp`.`bobot3`) AS `kriteria3`,pow(`tm`.`nilai4`,`tp`.`bobot4`) AS `kriteria4`,pow(`tm`.`nilai5`,`tp`.`bobot5`) AS `kriteria5` from ((`padusunej2`.`tb_mahasiswa` `tm` join `padusunej2`.`view_rangking` `vr` on((`tm`.`id_mahasiswa` = `vr`.`id_mahasiswa`))) join `padusunej2`.`tb_penerimaan` `tp` on((`vr`.`id_penerimaan` = `tp`.`id_penerimaan`)))	

c. *View total\_vektor\_s*

*View total\_vektor\_s* digunakan untuk mendapatkan nilai total dari vektor *s* pada tiap alternatif. *Query* dari *view total\_vektor\_s* dapat dilihat pada Tabel 4.12 berikut.

Tabel 4. 12 query dari view total\_vektor\_s

<i>view total_vektor_s</i>		
select	`padusunej2`.`vektor_s`.`id_mahasiswa`	AS
	`id_mahasiswa`,`padusunej2`.`vektor_s`.`id_penerimaan`	AS
	`id_penerimaan`,`padusunej2`.`vektor_s`.`nama`	AS
	`nama`,`padusunej2`.`vektor_s`.`nim` AS `nim`,`padusunej2`.`vektor_s`.`fakultas` AS	AS
	`fakultas`,`padusunej2`.`vektor_s`.`status`	AS
	`status`,`padusunej2`.`vektor_s`.`keterangan`	AS
	`keterangan`,`(((`padusunej2`.`vektor_s`.`kriteria1`	*
	`padusunej2`.`vektor_s`.`kriteria2`) * `padusunej2`.`vektor_s`.`kriteria3`)	*
	`padusunej2`.`vektor_s`.`kriteria4`) * `padusunej2`.`vektor_s`.`kriteria5`)	AS
	`perkalian_vektors` from `padusunej2`.`vektor_s`	

d. *View abstrak\_total\_semua\_vektor\_s*

*View abstrak\_total\_semua\_vektor\_s* digunakan untuk jumlah nilai total vektor s pada semua alternatif. *Query* dari *view vektor\_s* dapat dilihat pada Tabel 4.13 berikut.

Tabel 4. 13 query dari view abstrak\_total\_semua\_vektor\_s

<i>view abstrak_total_semua_vektor_s</i>		
select	`padusunej2`.`abstrak_total_vektors`.`id_mahasiswa`	AS
	`id_mahasiswa`,`padusunej2`.`abstrak_total_vektors`.`keterangan`	AS
	`keterangan`,`padusunej2`.`abstrak_total_vektors`.`status`	AS
	`status`,`padusunej2`.`abstrak_total_vektors`.`id_penerimaan`	AS
	`id_penerimaan`,`sum(`padusunej2`.`abstrak_total_vektors`.`perkalian_vektors`)	AS
	`hitung_semuavektors` from `padusunej2`.`abstrak_total_vektors` group by	
	`padusunej2`.`abstrak_total_vektors`.`id_penerimaan`	

e. *View vektor\_v*

*View vektor\_v* digunakan untuk mendapatkan nilai vektor v. *Query* dari *view vektor\_v* dapat dilihat pada Tabel 4.14 berikut.

Tabel 4. 14 query dari view vektor\_v

<i>view vektor_v</i>
<pre>select `tvs`.`id_mahasiswa` AS `id_mahasiswa`,`tvs`.`id_penerimaan` AS `id_penerimaan`,`tvs`.`nama` AS `nama`,`tvs`.`nim` AS `nim`,`tvs`.`fakultas` AS `fakultas`,`atsv`.`keterangan` AS `keterangan`,`tvs`.`status` AS `status`,`tvs`.`perkalian_vektors` / `atsv`.`hitung_semuavektors`) AS `vektor_v` from (`padusunej2`.`total_vektor_s` `tvs` join `padusunej2`.`abstrak_total_semua_vektors` `atsv` on((`tvs`.`id_penerimaan` = `atsv`.`id_penerimaan`))) where (`tvs`.`id_penerimaan` = `atsv`.`id_penerimaan`)</pre>

f. *View vektor\_v\_akhir*

*View vektor\_v\_akhir* digunakan untuk mendapatkan nilai pada semua alternative yang sudah diranking dengan id\_penerimaan terakhir. Semakin besar nilai yang didapatkan, semakin tinggi kualitas alternatif tersebut. *Query* dari *view vektor\_v\_akhir* dapat dilihat pada Tabel 4.15 berikut.

Tabel 4. 15 query dari view vektor\_v\_akhir

<i>view vektor_v_akhir</i>
<pre>select `v`.`id_mahasiswa` AS `id_mahasiswa`,`v`.`id_penerimaan` AS `id_penerimaan`,`v`.`nama` AS `nama`,`v`.`nim` AS `nim`,`v`.`fakultas` AS `fakultas`,`v`.`keterangan` AS `keterangan`,`v`.`status` AS `status`,`v`.`vektor_v` AS `vektor_v` from (`padusunej2`.`ambilidakhir` `a` join `padusunej2`.`vektor_v` `v` on((`a`.`id_penerimaan` = `v`.`id_penerimaan`))) where (`v`.`id_penerimaan` = `a`.`id_penerimaan`)</pre>

#### 4.5 Pengujian sistem

Pengujian sistem merupakan salah satu tahapan metode *waterfall*. Tahap tersebut dilakukan setelah melakukan implementasi / kode program. Pengujian merupakan proses menganalisa perangkat lunak untuk mengetahui perbedaan antara kondisi yang ada dengan kondisi yang diinginkan sehingga ditemukan *error* atau *bug*. Sehingga dapat dijadikan acuan evaluasi fitur – fitur perangkat lunak. Proses pengujian yang dilakukan adalah pengujian Black Box. Pengujian dilakukan

bersama dengan pihak yang bersangkutan untuk mendapatkan hasil yang optimal. Hasil pengujian *Black Box* terdapat pada Tabel 4.16.

Tabel 4. 16 Hasil Pengujian Black Box

1.No.	2. Fitur	3. Aksi	4. Hasil	5. Ket
1.	Masuk	Mengisi form username lalu klik tombol Masuk	Masuk berhasil dan user masuk ke dalam sistem informasi sesuai hak akses pengguna	Berhasil
		Klik tombol Masuk namun username atau password kosong	Menampilkan span "Please fill out this field"	Berhasil
		Klik tombol Masuk namun username atau password tidak sesuai	Menampilkan span " <i>These credentials do not match our records</i> "	Berhasil
2.	Menambah Data Event	Klik menu info event, lalu klik submenu tambah data event	Menampilkan halaman berisi form tambah data event dan tombol submit	Berhasil
		Mengisi form tamba data event lalu klik submit	Menyimpan data ke database dan menampilkan halaman lihat data event	Berhasil
		Klik tambahkan jika terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
3.	Melihat Data Event	Klik menu info event kemudian klik submenu lihat data event	Menampilkan halaman berisi tabel data event dan tombol edit	Berhasil

Dilanjutkan

## Lanjutan

1.	2.	3.	4.	5.
4.	Mengubah Data Event	Klik menu info event kemudian klik submenu lihat data event	Menampilkan halaman berisi tabel data event dan tombol edit	Berhasil
		Klik tombol edit salah satu data event pada tabel	Menampilkan halaman form edit data, tombol submit	Berhasil
		Klik tombol submit namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
5.	Menambah galeri kegiatan	Klik menu galeri kegiatan, lalu klik submenu tambah galeri kegiatan	Menampilkan halaman berisi form tambah galeri kegiatan dan tombol submit	Berhasil
		Mengisi form tamba galeri kegiatan lalu klik submit	Menyimpan data ke database dan menampilkan halaman lihat galeri kegiatan	Berhasil
		Klik tombol tambahkan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
6.	Melihat Galeri Kegiatan	Klik menu galeri kegiatan kemudian klik submenu lihat galeri kegiatan	Menampilkan halaman berisi tabel galeri kegiatan dan tombol edit	Berhasil
7.	Mengubah Galeri Kegiatan	Klik menu galeri kegiatan kemudian klik submenu lihat galeri kegiatan	Menampilkan halaman berisi tabel galeri kegiatan dan tombol edit	Berhasil

Dilanjutkan

## Lanjutan

1.	2.	3.	4.	5.
		Klik tombol edit salah satu data galeri kegiatan pada tabel	Menampilkan halaman form edit data, tombol submit	Berhasil
		Klik tombol submit namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
8.	Pendaftaran	Klik menu registrasi	Menampilkan halaman berisi form registrasi pendaftaran dan tombol submit	Berhasil
		Mengisi form pendaftaran lalu klik submit	Menyimpan data ke database dan menampilkan halaman awal website	Berhasil
		Klik tombol tambahkan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
9.	Melihat Data Lolos Tahun Sebelumnya	Klik menu semua vektor V	Menampilkan halaman berisi tabel data pendaftar lolos seleksi tahun sebelumnya	Berhasil
10.	Lihat Pendaftar	Klik menu lihat pendaftar	Menampilkan halaman berisi tabel pendaftar dan tombol edit	Berhasil
11.	Input Nilai Hasil Seleksi	Klik menu lihat pendaftar	Menampilkan halaman berisi tabel pendaftar dan tombol edit	Berhasil

Dilanjutkan

## Lanjutan

1.	2.	3.	4.	5.
		Klik tombol edit salah satu pendaftar untuk diinputkan nilai	Menampilkan halaman form edit data dan tombol submit	Berhasil
		Klik tombol submit	Menyimpan data ke <i>database</i>	Berhasil
12.	Menambah Data Penerimaan	Klik menu penerimaan, klik submenu tambah data penerimaan	Menampilkan halaman berisi form tambah penerimaan dan tombol submit	Berhasil
		Mengisi form tambah penerimaan lalu klik submit	Menyimpan data ke <i>database</i> dan menampilkan halaman lihat data penerimaan	Berhasil
		Klik tambahkan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
13.	Melihat Data Penerimaan	Klik menu data penerimaan kemudian klik submenu lihat penerimaan	Menampilkan halaman berisi tabel data penerimaan dan tombol edit	Berhasil
14.	Mengubah Penerimaan	Klik menu data penerimaan, klik submenu lihat penerimaan	Menampilkan halaman berisi tabel data penerimaan dan tombol edit	Berhasil
		Klik edit pada salah satu data penerimaan pada tabel	Menampilkan halaman form edit data, tombol submit	Berhasil
15.	Melihat vektor S	Klik menu lihat vektor S	Menampilkan halaman berisi tabel hasil perhitungan vektor S	Berhasil

Dilanjutkan

## Lanjutan

1.	2.	3.	4.	5.
16.	Melihat vektor V	Klik menu lihat vektor V	Menampilkan halaman berisi tabel hasil perhitungan vektor V	Berhasil
17.	Melihat hasil perangkingan	Klik menu lihat rangking	Menampilkan halaman berisi tabel hasil perangkingan nilai akhir	Berhasil
18.	Persetujuan Pendaftar Lolos	Klik menu hasil rangking	Menampilkan halaman berisi tabel hasil perangkingan dan tombol edit	Berhasil
		Klik tombol edit salah satu data pendaftar lolos seleksi yang akan di verifikasi pada tabel	Menampilkan halaman form edit data, tombol submit	Berhasil
		Klik tombol submit	Menyimpan data ke <i>database</i>	Berhasil
19.	Cetak Pendaftar Lolos Seleksi	Klik menu hasil rangking	Menampilkan halaman berisi tabel hasil perangkingan dan download	Berhasil
		Klik tombol edit	Menampilkan dokumen hasil perangkingan berupa Pdf	Berhasil
20.	Help	Klik menu help	Menampilkan halaman help berupa gambar penjelasan hak akses aktor	Berhasil

## BAB 6 PENUTUP

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

### 6.1 Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Dari hasil penelitian yang telah dipaparkan diatas, maka dapat ditarik kesimpulan bahwa sistem informasi seleksi penerimaan anggota paduan suara Universitas Jember yang mengimplementasikan metode *Weighted Product* mampu diimplementasikan dengan baik. Dalam mengimplementasikannya terlebih dahulu pelatih menentukan kriteria dan nilai bobot dari masing-masing kriteria. Kemudian menghitung nilai perbaikan bobot (normalisasi) dari masing-masing kriteria. Hasil perbaikan bobot digunakan untuk menghitung nilai prefensi alternatif menggunakan vektor S dengan cara nilai pendaftar per kriteria dipangkatkan dengan hasil normalisasi bobot per kriteria. Kemudian menghitung nilai prefensi relatif dari setiap alternatif menggunakan vektor V. Hasil perhitungan vektor V kemudian dirangking dari nilai terbesar ke nilai terkecil, dimana 30 nilai terbaik akan dinyatakan lolos seleksi penerimaan anggota paduan suara Universitas Jember.
2. Impelementasi Metode *Weighted Product* pada Sistem Informasi Seleksi Penerimaan Calon Anggota Paduan Suara Mahasiswa Universitas Jember mampu menjadi *website* untuk membantu proses penyeleksian calon anggota paduan suara. Alur penelitian yang menerapkan SDLC *waterfall* mampu diterapkan dengan baik pada sistem ini. Tahap pertama yaitu proses analisis yang terdiri dari studi literatur dan wawancara kepada pihak yang bersangkutan mampu mendapatkan hasil berupa data kriteria serta nilai bobot dari masing-masing kriteria. Seluruh kebutuhan fungsional dan non fungsional sistem dapat terpenuhi. Tahap pembuatan desain sistem informasi yang dibangun berdasarkan hasil wawancara dan kebutuhan sistem yang

terdiri dari *business process*, *usecase diagram*, *use case scenario*, *activity diagram*, *class diagram*, dan *Entity Relationship Diagram (ERD)* mampu diintegrasikan dengan data analisis. Implementasi sistem yang dibangun mampu menyesuaikan dengan desain sistem yang dapat membantu pelatih dalam proses penentuan seleksi calon anggota paduan suara seperti penilaian masing-masing kriteria dan mengolah nilai hingga menghasilkan nilai akhir yang menjadi acuan untuk proses seleksi. *Website* ini memudahkan tugas pelatih untuk membantu menentukan anggota paduan suara mahasiswa Universitas Jember.

## 6.2 Saran

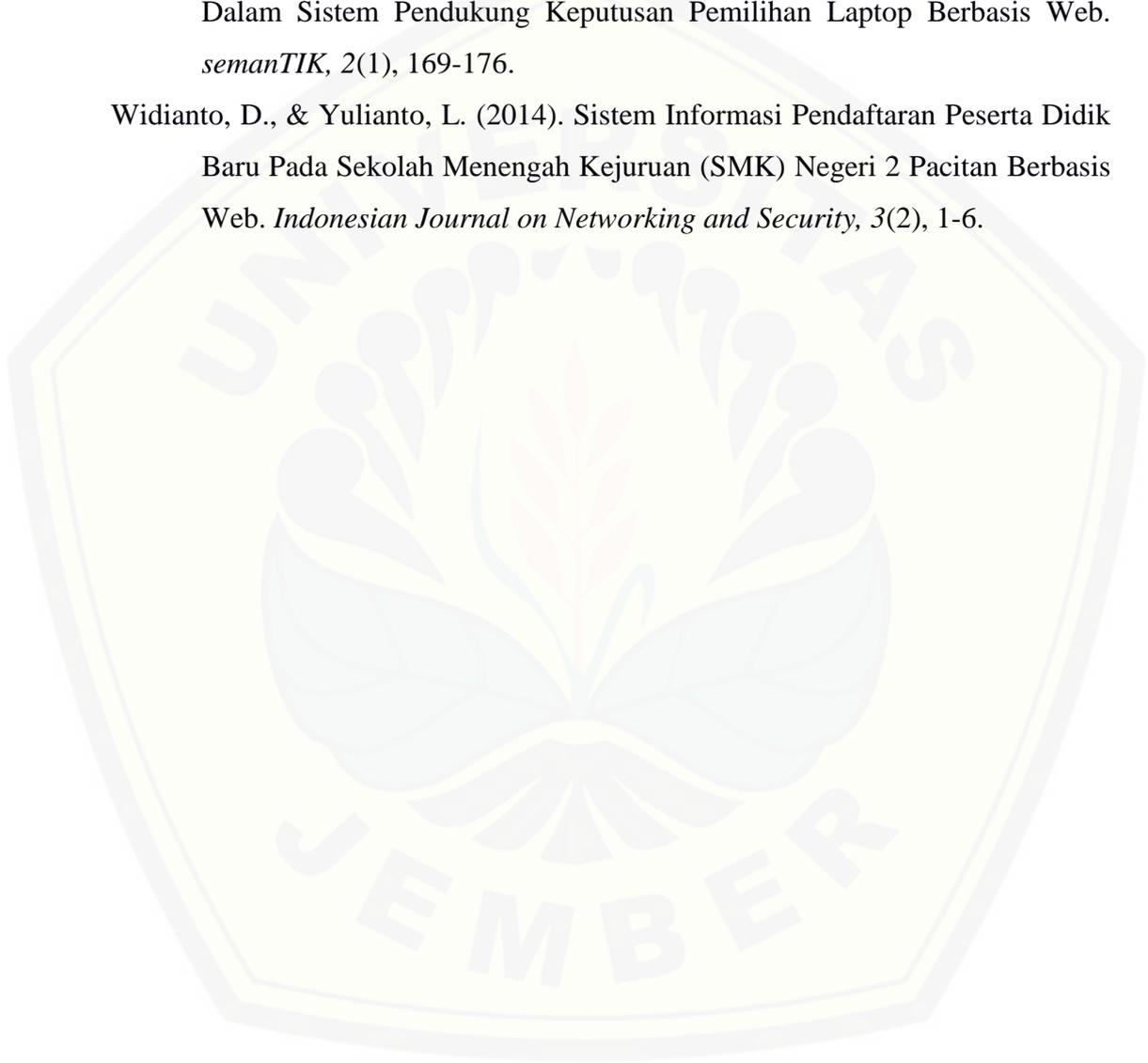
Beberapa saran yang ditujukan untuk memberikan masukan yang lebih baik pada penelitian selanjutnya yaitu:

1. Implementasi metode *Weighted Product* pada sistem informasi seleksi penerimaan anggota paduan suara mahasiswa Universitas Jember ini membutuhkan pengembangan lebih lanjut dengan membangun sistem yang dinamis sehingga pada sistem dapat ditambahkan data kriteria tanpa harus merubah keseluruhan kode program.
2. Implementasi metode *weighted product* pada sistem informasi seleksi penerimaan anggota paduan suara mahasiswa universitas jember ini alangkah baiknya jika ditambah atau digabungkan dengan sistem untuk mengelola data anggota (kepengurusan), manajemen jadwal latihan, informasi mengenai kompetisi paduan suara, serta fitur-fitur manajemen lainnya.

**DAFTAR PUSTAKA**

- Alfita, R. (2010). Perancangan Sistem Pendukung Keputusan Penentuan Prioritas Produk Unggulan Daerah Menggunakan Metode Weight Product (WP).
- Edhy, S. (2005). *Dukungan Sistem Informasi Manajemen (SIM) dalam Kegiatan Manajemen*. manejerial.
- Jalote, P. (2005). *An Intergrated Approach to Software Engineering*. (3. penyunt, Ed.) India: Springer Scence+Business Media.
- Khan, M. E. (2011). Different Approches To Black Box Testing Technique For Finding Error. *International Journal of Software Engineering & Aplication (IJSEA)*, 5.
- Kob, M., Henrich, N., Herzel, H., Howard, D., Tokuda, I., & Wolfe, J. (2011). Analysing and Understading the Singing Voice Recent and Progress and Open Questions. *Current Bioformatics*, 6, 362-374.
- Niswatin, R. K. (2016). Sistem Seleksi Penerimaan Mahasiswa Baru Menggunakan Metode Weghted Product (WP). *Seminar Nasional Teknologi Informasi dan Multimedia 2016*, 1-6.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku Satu)*. Yogyakarta: Andi Offset.

- Sianturi, I. S. (2013). Sistem Pendukung Keputusan Untuk Menentukan Pemilihan Jurusan Siswa Dengan Menggunakan Metode Weighted Product (Studi Kasus: SMA Swasta HKBP Doloksanggul). *Informasi dan Teknologi Ilmiah (INTI)*, 1, 19-22.
- Syafitri, N. A., Sutardi, & Dewi, A. P. (2016). Penerapan Metode Weighted Product Dalam Sistem Pendukung Keputusan Pemilihan Laptop Berbasis Web. *semanTIK*, 2(1), 169-176.
- Widianto, D., & Yulianto, L. (2014). Sistem Informasi Pendaftaran Peserta Didik Baru Pada Sekolah Menengah Kejuruan (SMK) Negeri 2 Pacitan Berbasis Web. *Indonesian Journal on Networking and Security*, 3(2), 1-6.



## LAMPIRAN

## LAMPIRAN A. SKENARIO

## 1. Use Case Scenario Login

Tabel 1 Skenario Use Case Masuk Sistem

No. Usecase	01
Nama usecase	Login
Aktor	Admin, Admin Penerimaan, Pelatih
Deskripsi	Fitur memasuki halaman beranda masing-masing pengguna sistem
Prakondisi	Aktor memasuki halaman utama
Pascakondisi	Pengguna berhasil memasuki halaman beranda masing-masing
Event Flow	
Normal Flow : memasuki sistem	
1. Klik button Login	
	2. Menampilkan halaman masuk sistem denganform : a. Username (varchar 20) b. Pasasword (varchar 50)
3. mengisi form a. Username (varchar 20) b. Password (varchar 50)	
4. klik button login	
	5. pengecekan data
	6. berhasil memasuki halaman beranda
Alternative flow : data kosong	
4. Klik button masuk	
	5. Memeriksa data
	6. Menampilkan pesan “enter username or password”
Alternative flow : data tidak valid	
4. Klik button masuk	
	5. Memeriksa data
	6. Menampilkan pemberitahuan “These credentials do not match our records”

7. Klik button ok	
	8. Menampilkan halaman masuk sistem

## 2. Use Case Scenario Menambah Data Event

Tabel A.2 Skenario Use Case Menambah Data Event

No. usecase	02
Nama usecase	Menambah Data Event
Actor	Admin
Deskripsi	Fitur untuk menambah event paduan suara mahasiswa
Prakondisi	Memasuki halaman beranda
Pascakondisi	Data event telah selesai dikelola
Event Flow	
Normal flow : tambah data kriteria	
Aksi aktor	Reaksi sistem
1. Klik menu info event	
2. Klik submenu tambah event	
	3. Menampilkan halaman tambah event dengan form berisi: <ul style="list-style-type: none"> <li>a. Nama Event (varchar 225)</li> <li>b. Tanggal Event (date)</li> <li>c. Tempat Event (varchar 225)</li> <li>d. Penyelenggara Event (varchar 225)</li> <li>e. Gambar publikasi Event (Varchar 225)</li> </ul> Dan button <ul style="list-style-type: none"> <li>a. Submit</li> </ul>
4. Mengisi Form <ul style="list-style-type: none"> <li>a. Nama Event (varchar 225)</li> <li>b. Tanggal Event (date)</li> <li>c. Tempat Event (varchar 225)</li> <li>d. Penyelenggara Event (varchar 225)</li> <li>e. Gambar Event (Varchar 225)</li> </ul>	
5. Klik button submit	
	7. Memeriksa data
	8. Data event telah disimpan

Alternative flow : form kosong	
5. Klik button submit	
	6. Memeriksa data
	7. Menampilkan pesan “harap ini bidang ini”
Alternative flow : data tidak sesuai format	
5. Klik button submit	
	6. Memeriksa data
	7. Menampilkan pesan “maaf data yang anda masukkan tidak sesuai”

### 3. Use Case Scenario Melihat Data Event

Tabel 3 Skenario Use Case Melihat Data Event

No. usecase	03
Nama usecase	Melihat Data Event
Actor	Admin
Deskripsi	Fitur untuk melihat event paduan suara mahasiswa
Prakondisi	Memasuki halaman beranda
Pascakondisi	Data event telah selesai dikelola
Event Flow	
Normal flow : lihat data event	
Aksi aktor	Reaksi sistem
1. Klik menu info event	
2. Klik submenu lihat event	
	3. Menampilkan halaman event dengan table berisi: <ol style="list-style-type: none"> <li>a. Nama Event (varchar 50)</li> <li>b. Tanggal Event (date)</li> <li>c. Tempat Event (varchar 25)</li> <li>d. Penyelenggara Event (varchar 25)</li> <li>e. Gambar publikasi event (varchar 25)</li> </ol> Dan button <ol style="list-style-type: none"> <li>a. Edit data</li> </ol>

## 4. Use Case Scenario Mengubah Data Event

Tabel 4 Skenario Use Case Mengubah Data Event

No. usecase	04
Nama usecase	Mengubah Data Event
Actor	Admin
Deskripsi	Fitur untuk mengubah event paduan suara mahasiswa
Prakondisi	Memasuki halaman beranda
Pascakondisi	Data event telah selesai diubah
Normal flow : ubah data event	
Aksi aktor	Reaksi sistem
1. Klik menu info event	
2. Klik submenu lihat event	
	3. Menampilkan halaman event dengan table berisi: <ul style="list-style-type: none"> <li>a. Nama Event (varchar 50)</li> <li>b. Tanggal Event (date)</li> <li>c. Tempat Event (varchar 25)</li> <li>d. Penyelenggara Event (varchar 25)</li> <li>e. Gambar publikasi event (25) Dan button</li> </ul>
4. klik button edit data	
	5. Menampilkan form ubah data event <ul style="list-style-type: none"> <li>a. Nama Event (varchar 50)</li> <li>b. Tanggal Event (date)</li> <li>c. Tempat Event (varchar 25)</li> <li>d. Penyelenggara Event (varchar 25)</li> <li>e. Gambar publikasi event (varchar 25)</li> </ul>
6. Mengubah data <ul style="list-style-type: none"> <li>a. Nama Event (varchar 50)</li> <li>b. Tanggal Event (date)</li> <li>c. Tempat Event (varchar 25)</li> <li>d. Penyelenggara Event (varchar 25)</li> <li>e. Gambar publikasi event (varchar 25)</li> </ul>	
7. Klik button submit	
	8. Memeriksa data
	9. Data event telah berhasil diubah

Alternative flow : form kosong	
7. Klik button submit	
	8. Memeriksa data
	9. Menampilkan pesan “harap isi bidang ini”

### 5. Use Case Scenario Menambah Galeri Kegiatan

Tabel 5 Skenario Use Case Menambah Galeri Kegiatan

No. usecase	05
Nama usecase	Menambah Galeri Kegiatan
Actor	Admin
Deskripsi	Fitur untuk menambah galeri kegiatan paduan suara mahasiswa
Prakondisi	Memasuki halaman beranda
Pascakondisi	Galeri kegiatan telah selesai ditambah
Event Flow	
Normal flow : tambah galeri kegiatan	
Aksi aktor	Reaksi sistem
1. Klik menu galeri kegiatan	
2. Klik submenu tambah galeri kegiatan	
	3. Menampilkan halaman tambah galeri kegiatan dengan form berisi: a. caption (text) b. gambar (varchar 50) Dan button a. Reset b. Submit
4. Mengisi Form a. caption (text) b. gambar (varchar 50)	
5. Klik button submit	
	6. Memeriksa data
	7. Data galeri kegiatan telah disimpan
Alternative flow : form kosong	
5. Klik button submit	

	6. Memeriksa data
	7. Menampilkan pesan “harap ini bidang ini”
Alternative flow : data tidak sesuai format	
8. Klik button submit	
	8. Memeriksa data
	9. Menampilkan pesan “maaf data yang anda masukkan tidak sesuai”

#### 6. Use Case Scenario Melihat Galeri Kegiatan

Tabel 6 Skenario Use Case Melihat Galeri Kegiatan

No. usecase	06
Nama usecase	Melihat Galeri Kegiatan
Actor	Admin
Deskripsi	Fitur untuk melihat galeri kegiatan paduan suara mahasiswa
Prakondisi	Memasuki halaman beranda
Pascakondisi	Galeri kegiatan telah selesai dilihat
Event Flow	
Normal flow : lihat galeri kegiatan	
Aksi aktor	Reaksi sistem
1. Klik menu galeri kegiatan	
2. Klik submenu tambah galeri kegiatan	
	3. Menampilkan halaman tambah galeri kegiatan dengan form berisi: c. caption (text) d. gambar (varchar 50) Dan button  c. Reset d. Submit

#### 7. Use Case Scenario Mengubah Galeri Kegiatan

Tabel 7 Skenario Use Case Menambah Data Event

No. usecase	07
Nama usecase	Mengubah Galeri Kegiatan

Actor	Admin
Deskripsi	Fitur untuk mengubah galeri kegiatan paduan suara mahasiswa
Prakondisi	Memasuki halaman beranda
Pascakondisi	Galeri kegiatan telah selesai diubah
Event Flow	
Normal flow : ubah galeri kegiatan	
Aksi aktor	Reaksi sistem
1. Klik menu galeri kegiatan	
2. Klik submenu lihat galeri kegiatan	
	3. Menampilkan halaman galeri kegiatan dengan table berisi: a. Caption (text) b. Gambar (varchar 50) Dan button a. Edit data
4. klik button edit data	
	5. Menampilkan form ubah galeri kegiatan a. Caption (text) b. Gambar (varchar 50)
6. Mengubah data a. Caption (text) b. Gambar (vachar 50)	
7. Klik button submit	
	8. Memeriksa data
	9. Data galeri kegiatan telah berhasil diubah
Alternative flow : form kosong	
7. Klik button submit	
	8. Memeriksa data
	9. Menampilkan pesan “harap isi bidang ini”

#### 8. Use Case Scenario Pendaftaran

Tabel 8 Skenario Pendaftaran

No. Usecase	08
Nama usecase	Pendaftaran
Aktor	Mahasiswa
Deskripsi	Fitur untuk reigistrasi pendaftaran
Prakondisi	Memasuki halaman beranda

Pascakondisi	Data pendaftar baru berhasil ditambahkan
Event Flow	
Normal flow : pendaftaran	
Aksi aktor	Reaksi sistem
1. Klik menu registrasi	
	2. Menampilkan halaman registrasi pendaftaran berupa form dengan isi: <ul style="list-style-type: none"> <li>a. Gelombang Penerimaan</li> <li>b. Nama (varchar 50)</li> <li>c. NIM (varchar 15)</li> <li>d. Tanggal Lahir (varchar 20)</li> <li>e. Alamat (text)</li> <li>f. Fakultas (varchar 25)</li> <li>g. Jurusan (varchar 25)</li> <li>h. No HP (int 222)</li> <li>i. File (varchar 225)</li> </ul> Dan button: <ul style="list-style-type: none"> <li>a. Submit</li> </ul>
3. mengisi form <ul style="list-style-type: none"> <li>a. Nama (varchar 50)</li> <li>b. NIM (varchar 15)</li> <li>c. Tanggal Lahir (varchar 20)</li> <li>d. Alamat (text)</li> <li>e. Fakultas (varchar 25)</li> <li>f. Jurusan (varchar 25)</li> <li>g. No HP (int 222)</li> <li>h. File (varchar 225)</li> </ul>	
4. klik button submit	
	5. memeriksa data
	6. data tersimpan
Alternative flow : data kosong	
4. Klik button submit	
	5. Memeriksa data
	6. Menampilkan pesan “harap isi bidang ini”

### 9. Use Case Scenario Melihat Data Lolos Tahun Sebelumnya

Tabel 9 Skenario Melihat Data Lolos Tahun Sebelumnya

No. usecase	09
Nama usecase	Melihat data lolos tahun sebelumnya
Actor	Admin penerimaan
Deskripsi	Fitur untuk melihat data pendaftar

Prakondisi	Memasuki halaman beranda
Pascakondisi	Data pendaftaran telah selesai dilihat
Event Flow	
Normal flow : melihat data pendaftar	
Aksi aktor	Reaksi sistem
1. Klik menu Semua vektor V	
	2. Menampilkan halaman data hasil akhir semua berupa tabel yang berisi: <ul style="list-style-type: none"> <li>a. Keterangan (varchar 50)</li> <li>b. Nama (varchar 50)</li> <li>c. NIM (varchar 50)</li> <li>d. Fakultas (varchar 25)</li> <li>e. Hasil Akhir (varchar 25)</li> <li>f. Status (varchar 25)</li> </ul>

#### 10. Use Case Scenario Lihat Data Pendaftar

Tabel 10 Skenario Lihat Data Pendaftar

No. usecase	10
Nama usecase	Melihat data pendaftar
Actor	Admin penerimaan
Deskripsi	Fitur untuk melihat data pendaftar
Prakondisi	Memasuki halaman beranda
Pascakondisi	Data pendaftar telah selesai dilihat
Event Flow	
Normal flow : melihat data pendaftar	
Aksi aktor	Reaksi sistem
1. Klik menu data pendaftar baru	
2. Klik submenu lihat data pendaftar baru	
	3. Menampilkan halaman data pendaftar berupa tabel yang berisi: <ul style="list-style-type: none"> <li>a. Nama (varchar 50)</li> <li>b. NIM (varchar 15)</li> <li>c. Fakultas (varchar 25)</li> <li>d. Nilai 1 (double)</li> <li>e. Nilai 2 (double)</li> <li>f. Nilai 3 (double)</li> <li>g. Nilai 4 (double)</li> <li>h. Nilai 5 (double)</li> </ul> Dan button

	a. Edit
--	---------

### 11. Use Case Scenario Input Nilai Hasil Seleksi

Tabel 11 Skenario Lihat Data Pendaftar

No. usecase	12
Nama usecase	Input Nilai Hasil Seleksi
Actor	Admin penerimaan
Deskripsi	Fitur untuk melihat data pendaftar
Prakondisi	Memasuki halaman beranda
Pascakondisi	Data pendaftar telah selesai dilihat
Event Flow	
Normal flow : melihat data pendaftar	
Aksi aktor	Reaksi sistem
1. Klik menu data pendaftar baru	
2. Klik submenu lihat data pendaftar baru	
	3. Menampilkan halaman data pendaftar berupa tabel yang berisi: <ul style="list-style-type: none"> <li>a. Nama (varchar 50)</li> <li>b. NIM (varchar 15)</li> <li>c. Fakultas (varchar 25)</li> <li>d. Nilai 1 (double)</li> <li>e. Nilai 2 (double)</li> <li>f. Nilai 3 (double)</li> <li>g. Nilai 4 (double)</li> <li>h. Nilai 5 (double)</li> </ul> Dan button <ul style="list-style-type: none"> <li>b. Edit</li> </ul>
4. Klik button edit	
	5. Menampilkan form ubah data pendaftar baru <ul style="list-style-type: none"> <li>a. Nama (varchar 50)</li> <li>b. NIM (varchar 15)</li> <li>c. Fakultas (varchar 25)</li> <li>d. Nilai 1 (double)</li> <li>e. Nilai 2 (double)</li> <li>f. Nilai 3 (double)</li> <li>g. Nilai 4 (double)</li> <li>h. Nilai 5 (double)</li> </ul>

6. Klik button submit	
	7. Memeriksa data
	8. Data pendaftar berhasil diubah

## 12. Use Case Scenario Menambah Data Penerimaan

Tabel 12 Skenario Lihat Data Pendaftar

No. usecase	13
Nama usecase	Menambah data penerimaan
Actor	Admin penerimaan
Deskripsi	Fitur untuk menambah data penerimaan paduan suara
Prakondisi	Memasuki halaman beranda
Pascakondisi	Data penerimaan telah selesai ditambah
Event Flow	
Normal flow : tambah penerimaan	
Aksi aktor	Reaksi sistem
1. Klik menu penerimaan	
2. Klik submenu tambah penerimaan	
	3. Menampilkan halaman tambah penerimaan dengan form berisi: <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> <li>f. Bobot artikulasi (double)</li> <li>g. Bobot power suara (double)</li> </ul> Dan button <ul style="list-style-type: none"> <li>e. Submit</li> </ul>
4. Mengisi Form <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> <li>f. Bobot artikulasi (double)</li> <li>g. Bobot power suara (double)</li> </ul>	
5. Klik button submit	

	9. Memeriksa data
	10. Data penerimaan telah disimpan
Alternative flow : form kosong	
5. Klik button submit	
	6. Memeriksa data
	7. Menampilkan pesan “harap ini bidang ini”
Normal flow : ubah data penerimaan	
Aksi aktor	Reaksi sistem
1. Klik menu data penerimaan	
2. Klik submenu lihat penerimaan	
	3. Menampilkan halaman penerimaan dengan table berisi: <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> <li>f. Bobot artikulasi (double)</li> <li>g. Bobot power suara (double)</li> </ul> Dan button <ul style="list-style-type: none"> <li>a. Edit</li> </ul>
4. klik button edit	
	5. Menampilkan form ubah galeri kegiatan <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> <li>f. Bobot artikulasi (double)</li> <li>g. Bobot power suara (double)</li> </ul>
6. Mengubah data <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> <li>f. Bobot artikulasi (double)</li> <li>g. Bobot power suara (double)</li> </ul>	
7. Klik button submit	
	8. Memeriksa data

	9. Data penerimaan telah berhasil diubah
Alternative flow : form kosong	
7. Klik button submit	
	8. Memeriksa data
	9. Menampilkan pesan “harap isi bidang ini”

### 13. Use Case Scenario Melihat Data Penerimaan

Tabel 13 Skenario Melihat Data Penerimaan

No. usecase	14
Nama usecase	Melihat data penerimaan
Actor	Admin penerimaan
Deskripsi	Fitur untuk melihat data penerimaan paduan suara
Prakondisi	Memasuki halaman beranda
Pascakondisi	Data penerimaan telah selesai dilihat
Event Flow	
Normal flow : lihat penerimaan	
Aksi aktor	Reaksi sistem
Aksi aktor	Reaksi sistem
1. Klik menu data penerimaan	
2. Klik submenu lihat penerimaan	
	3. Menampilkan halaman penerimaan dengan table berisi: <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> <li>f. Bobot artikulasi (double)</li> <li>g. Bobot power suara (double)</li> </ul> Dan button <ul style="list-style-type: none"> <li>b. Edit</li> </ul>

## 14. Use Case Scenario Mengubah Data Penerimaan

Tabel 14 Skenario Mengubah Data Penerimaan

No. usecase	15
Nama usecase	Mengubah data penerimaan
Actor	Admin penerimaan
Deskripsi	Fitur untuk mengubah data penerimaan paduan suara
Prakondisi	Memasuki halaman beranda
Pascakondisi	Data penerimaan telah selesai diubah
Event Flow	
Normal flow : ubah data penerimaan	
Aksi aktor	Reaksi sistem
1. Klik menu data penerimaan	
2. Klik submenu lihat penerimaan	
	3. Menampilkan halaman penerimaan dengan table berisi: <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> <li>f. Bobot artikulasi (double)</li> <li>g. Bobot power suara (double)</li> </ul> Dan button <ul style="list-style-type: none"> <li>c. Edit</li> </ul>
4. klik button edit	
	5. Menampilkan form ubah data penerimaan <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> <li>f. Bobot artikulasi (double)</li> <li>g. Bobot power suara (double)</li> </ul>
6. Mengubah data <ul style="list-style-type: none"> <li>a. Keterangan (varchar 225)</li> <li>b. Batas penerimaan (date)</li> <li>c. Bobot intonasi (double)</li> <li>d. Bobot not balok (double)</li> <li>e. Bobot not angka (double)</li> </ul>	

f. Bobot artikulasi (double) g. Bobot power suara (double)	
7. Klik button submit	
	8. Memeriksa data
	9. Data penerimaan telah berhasil diubah
Alternative flow : form kosong	
10. Klik button submit	
	11. Memeriksa data
	12. Menampilkan pesan “harap isi bidang ini”

#### 15. Use Case Scenario Melihat Hasil Perangkingan

Tabel 15 Skenario Melihat Hasil Perangkingan

No. usecase	16
Nama usecase	Melihat Hasil Perangkingan
Aktor	Pelatih
Deskripsi	Fitur untuk melihat hasil dari perangkingan
Prakondisi	Memasuki halaman beranda
Pascakondisi	Hasil perangkingan telah selesai dilihat
Event Flow	
Normal flow : melihat hasil perangkingan	
Aksi aktor	Reaksi sistem
1. Klik menu Lihat rangking	
	2. Menampilkan halaman hasil perangkingan lolos seleksi berupa tabel dengan isi: <ul style="list-style-type: none"> <li>a. Nama (varchar 50)</li> <li>b. NIM (varchar 15)</li> <li>c. Fakultas (varchar 25)</li> <li>d. Total Nilai (double)</li> </ul>

#### 16. Use Case Scenario Cetak Pendaftar Disetujui

Tabel 16 Skenario Cetak Pendaftar Disetujui

No. usecase	17
Nama usecase	Cetak Pendaftar Disetujui
Aktor	Admin penerimaan

Deskripsi	Fitur untuk melihat hasil dari perangkingan
Prakondisi	Memasuki halaman beranda
Pascakondisi	Hasil perangkingan telah selesai dilihat
Event Flow	
Normal flow : melihat hasil perangkingan	
Aksi aktor	Reaksi sistem
3. Klik menu Lihat rangking	
	4. Menampilkan halaman hasil perangkingan lolos seleksi berupa tabel dengan isi: <ul style="list-style-type: none"> <li>e. Nama (varchar 50)</li> <li>f. NIM (varchar 15)</li> <li>g. Fakultas (varchar 25)</li> <li>h. Total Nilai (double)</li> </ul> Button: <ul style="list-style-type: none"> <li>a. Cetak</li> </ul>

#### 17. Use Case Scenario Persetujuan Pendaftar Lolos Seleksi

Tabel 17 Skenario Persetujuan Pendaftar Lolos Seleksi

No. usecase	18
Nama usecase	Persetujuan Pendaftar Lolos Seleksi
Aktor	Admin penerimaan
Deskripsi	Fitur untuk mengubah status pendaftar yang lolos seleksi
Prakondisi	Memasuki halaman beranda
Pascakondisi	Status pendaftar berhasil diubah
Event Flow	
Normal flow : persetujuan pendaftar lolos seleksi	
Aksi aktor	Reaksi sistem
1. Klik menu hasil rangking	
	2. Menampilkan halaman hasil rangking lolos seleksi berupa tabel dengan isi: <ul style="list-style-type: none"> <li>a. Rangking (varchar 50)</li> <li>b. Keterangan (varchar 50)</li> <li>c. Nama (varchar 25)</li> <li>d. NIM (varchar 15)</li> <li>e. Fakultas (varchar 20)</li> <li>f. Hasil Akhir (varchar 20)</li> <li>g. Status (varchar 25)</li> </ul>

	Button a. Edit
3. Klik button edit	
	4. Menampilkan form ubah status dengan isi: a. Rangkaing (varchar 50) b. Keterangan (varchar 50) c. Nama (varchar 25) d. NIM (varchar 15) e. Fakultas (varchar 20) f. Hasil Akhir (varchar 20) g. Status (varchar 25)
5. Mengubah data: a. Status	
6. Klik button Submit	
	7. Memeriksa data
	8. Status berhasil diubah

## 18. Use Case Scenario Help

Tabel 18 Skenario Help

No. usecase	19
Nama usecase	Help
Aktor	Admin penerimaan, Admin, Pelatih
Deskripsi	Fitur untuk melihat hasil dari perangkaing
Prakondisi	Memasuki halaman beranda
Pascakondisi	Fitur help telah selesai dilihat
Event Flow	
Normal flow : help	
Aksi aktor	Reaksi sistem
1. Klik menu Help	
	2. Menampilkan halaman help berupa panduan mengenai hak akses dari masing-masing aktor

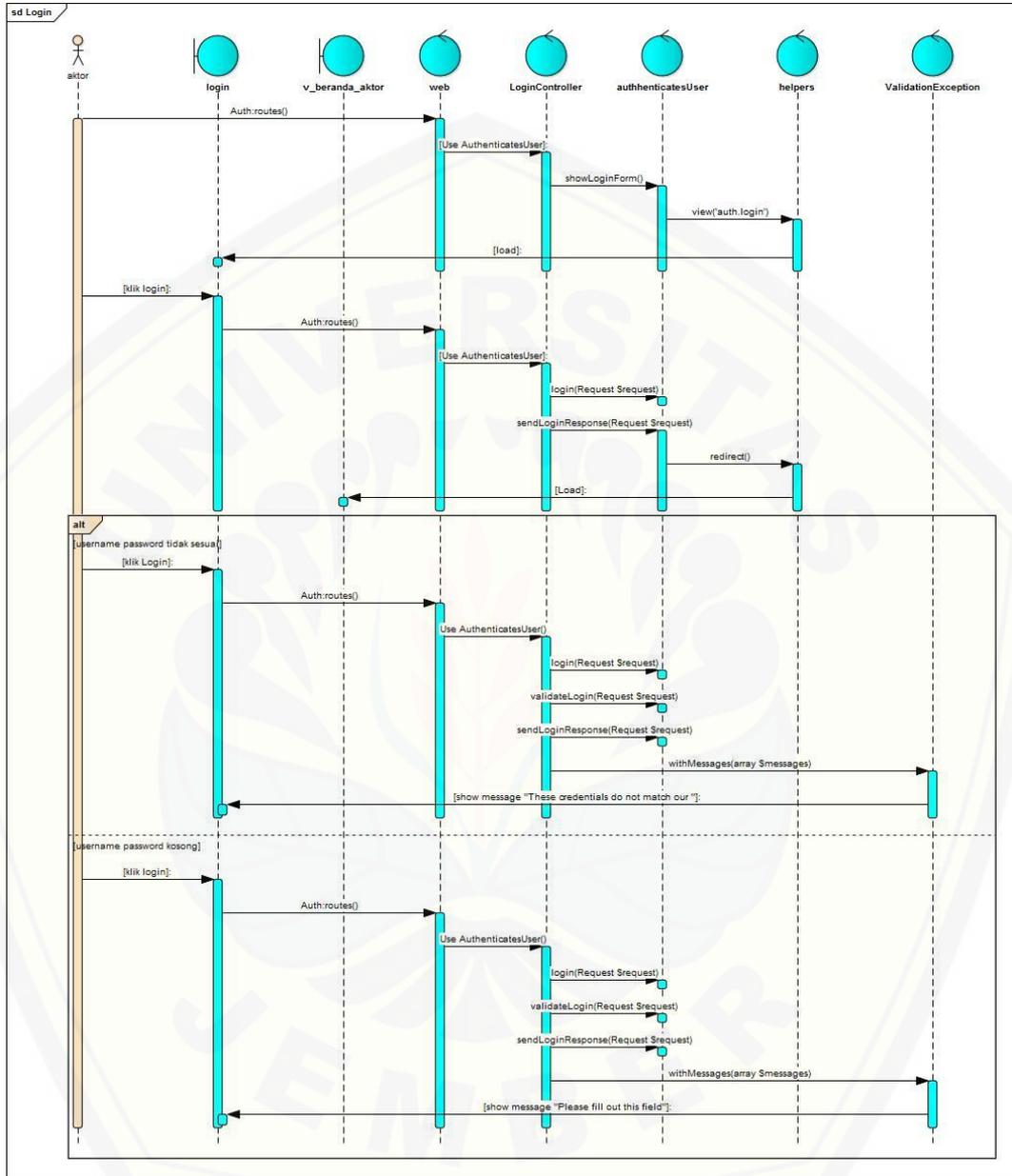
## 19. Use Case Scenario Logout

Tabel 19 Skenario Logout

No. usecase	20
Nama usecase	Logout
Aktor	Admin penerimaan, Admin, Pelatih
Deskripsi	Aktor Keluar dari sistem
Prakondisi	Aktor menggunakan sistem informasi
Pacakondisi	Aktor keluar dari sistem informasi
Event Flow	
Normal flow : Logout	
Aksi aktor	Reaksi sistem
1. Klik button Logout	
	2. Menampilkan halaman beranda website

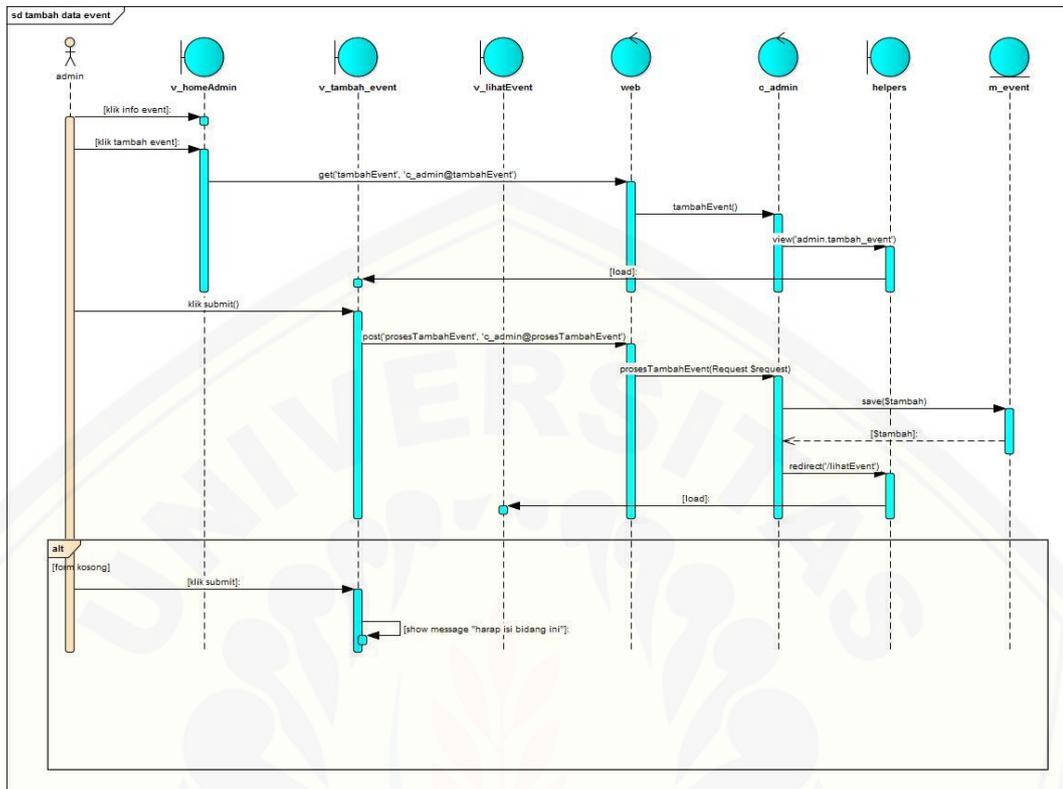
LAMPIRAN B. SEQUENCE DIAGRAM

1. Sequence Diagram Masuk



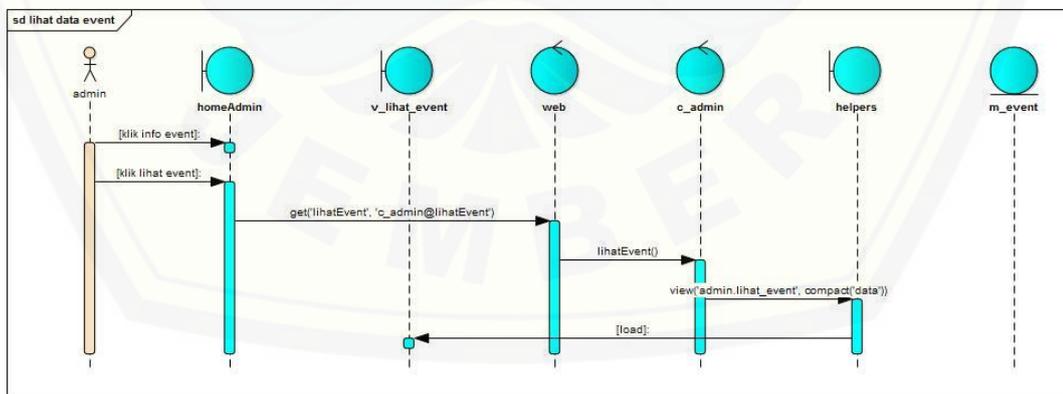
Gambar 1. Sequence Diagram Login

2. Sequence Diagram Menambah Data Event



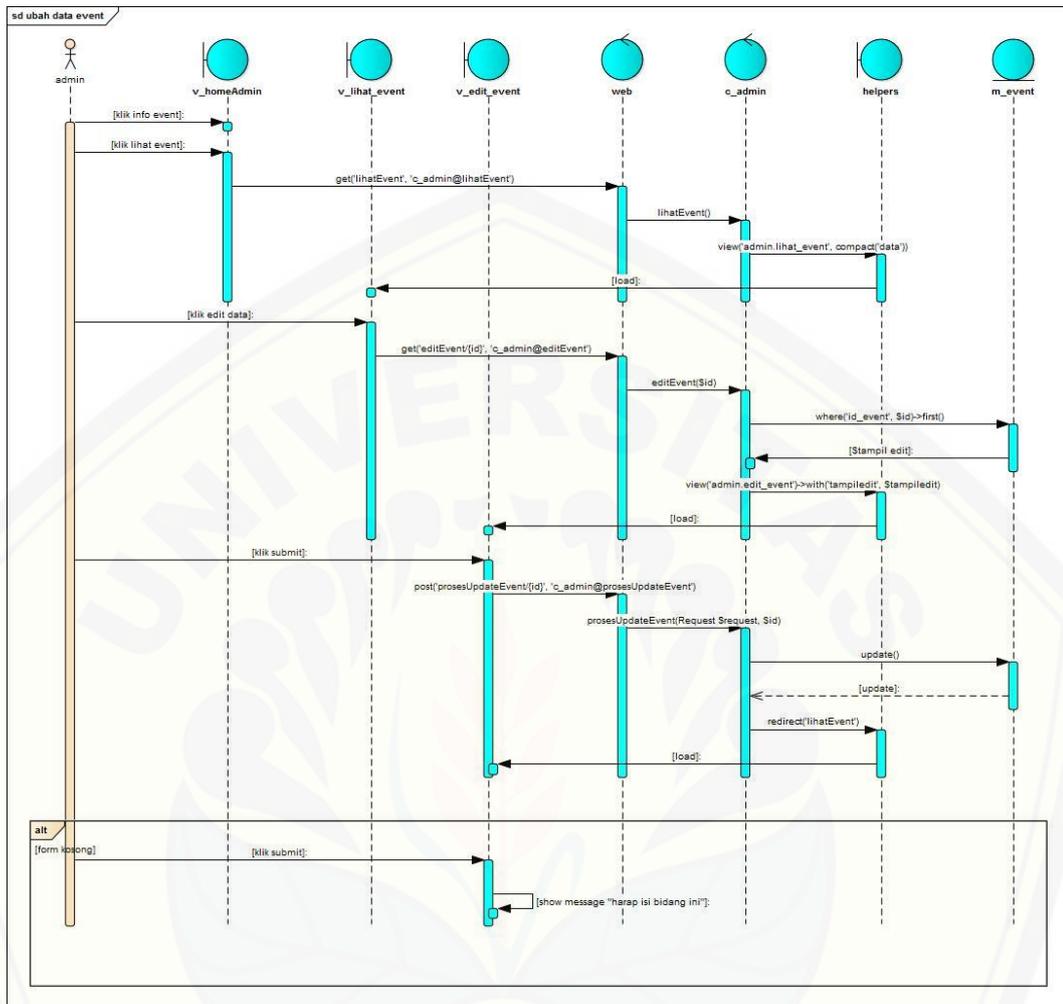
Gambar 2. Sequence Diagram Menambah Data Event

3. Sequence Diagram Melihat Data Event



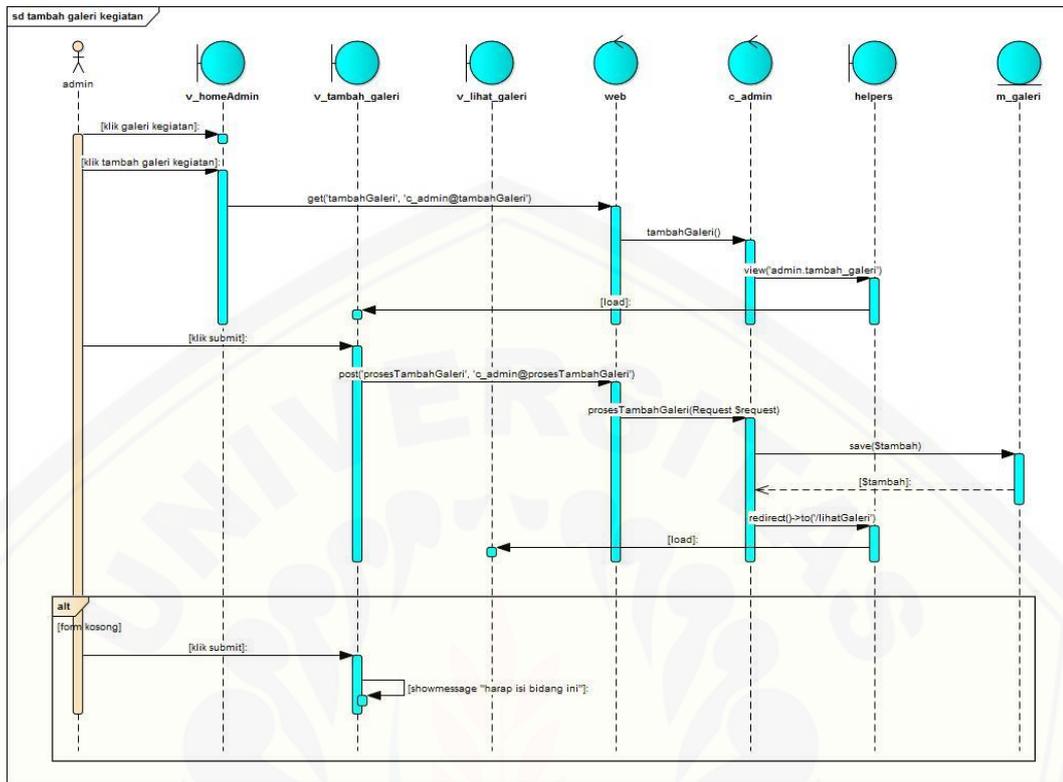
Gambar 3. Sequence Diagram Menambah Data Event

4. Sequence Diagram Mengubah Data Event



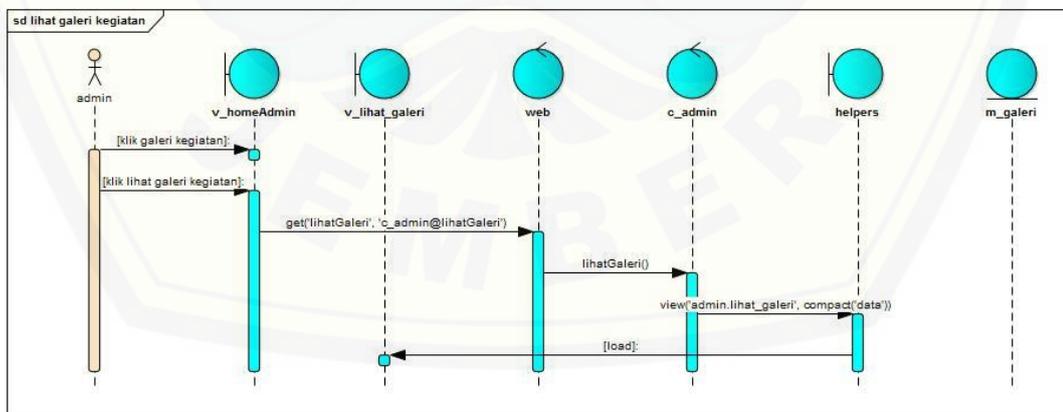
Gambar 4. Sequence Diagram Mengubah Data Event

5. Sequence Diagram Menambah Galeri Kegiatan



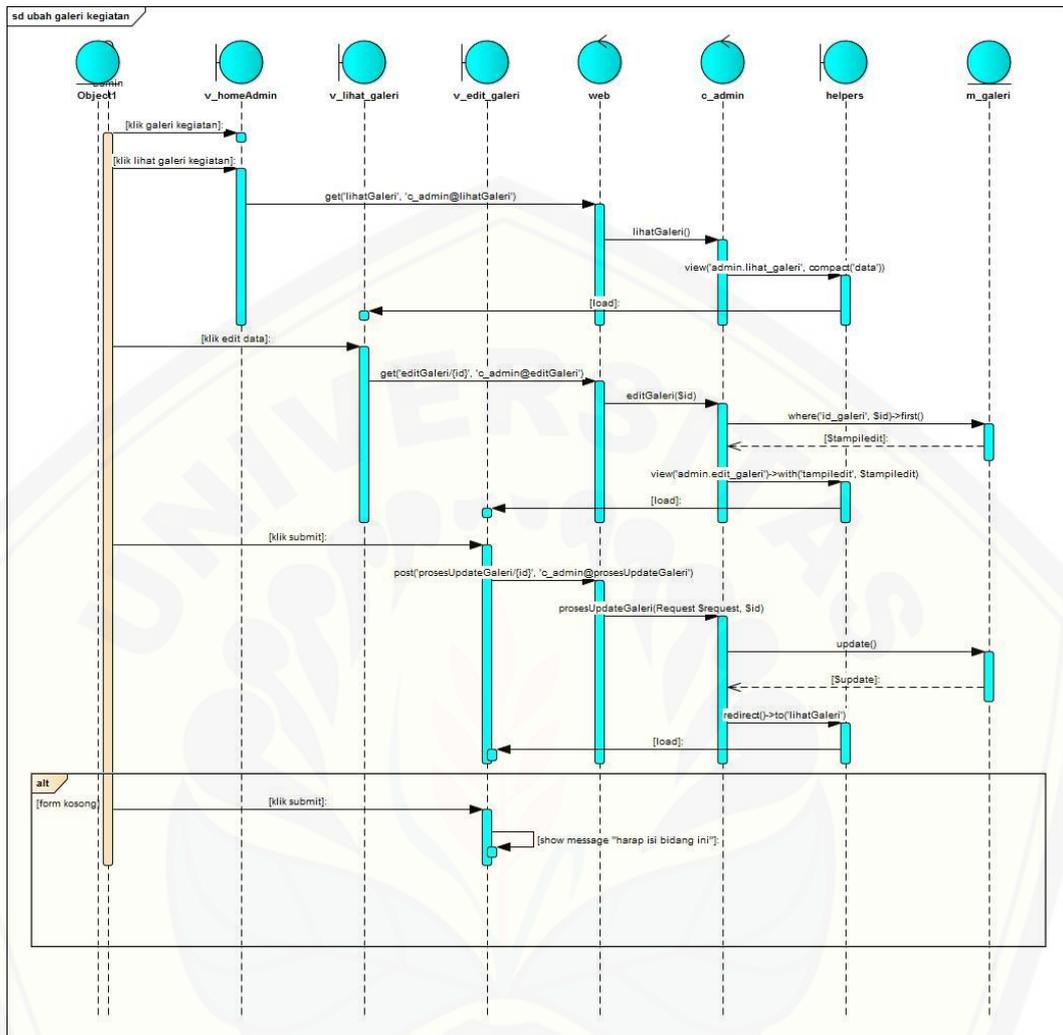
Gambar 5. Sequence Diagram Menambah Galeri Kegiatan

6. Sequence Diagram Melihat Galeri Kegiatan



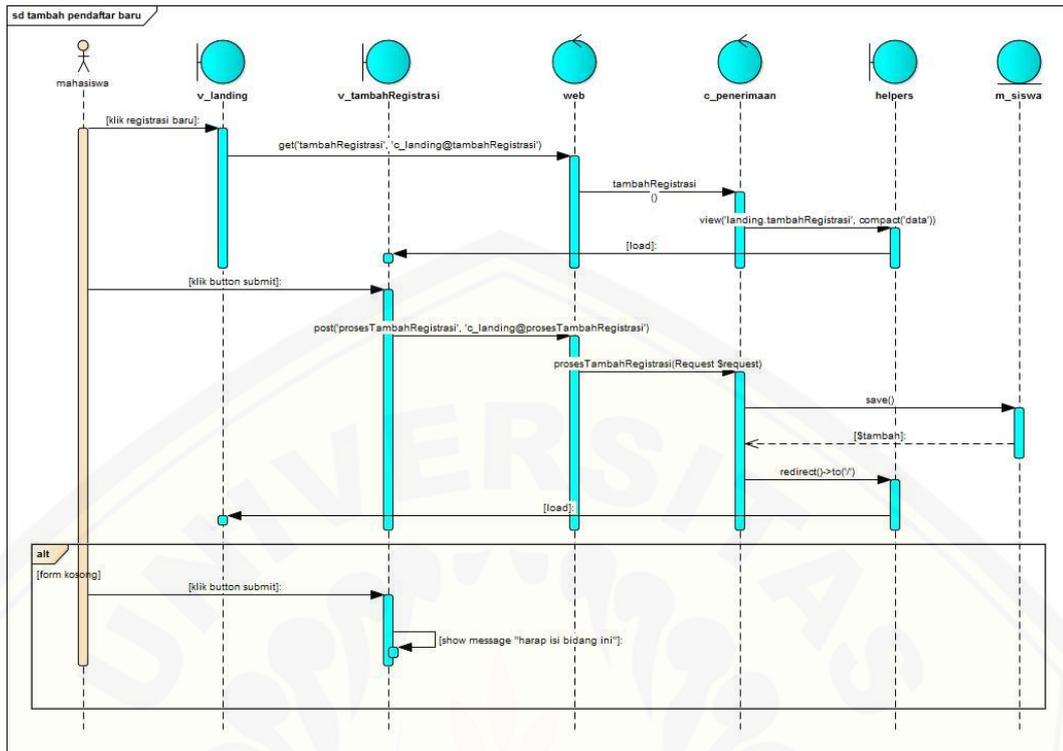
Gambar 6. Sequence Diagram Melihat Galeri Kegiatan

7. Sequence Diagram Mengubah Galeri Kegiatan



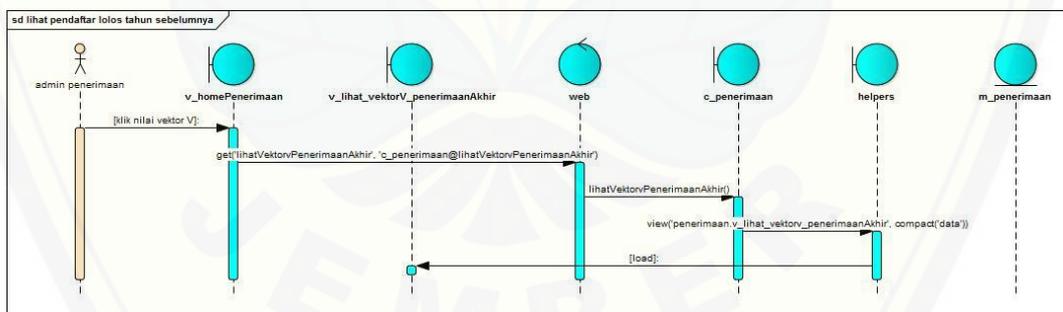
Gambar 7. Sequence Diagram Mengubah Galeri Kegiatan

8. Sequence Diagram Pendaftaran



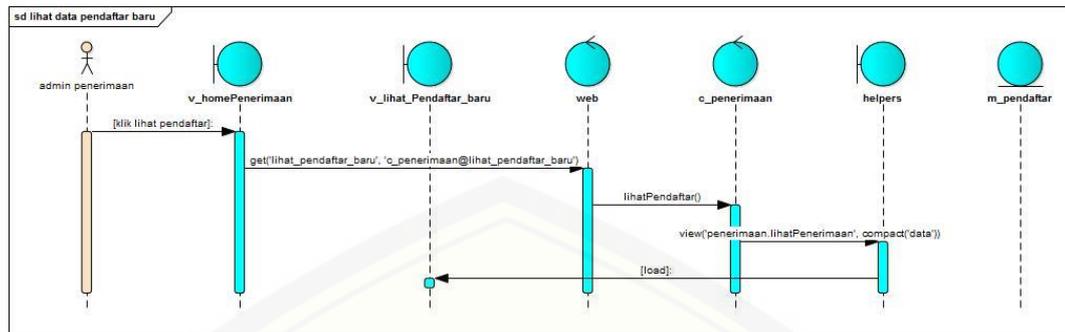
Gambar 8. Sequence Diagram Pendaftaran

9. Sequence Diagram Melihat Data Lolos Tahun Sebelumnya



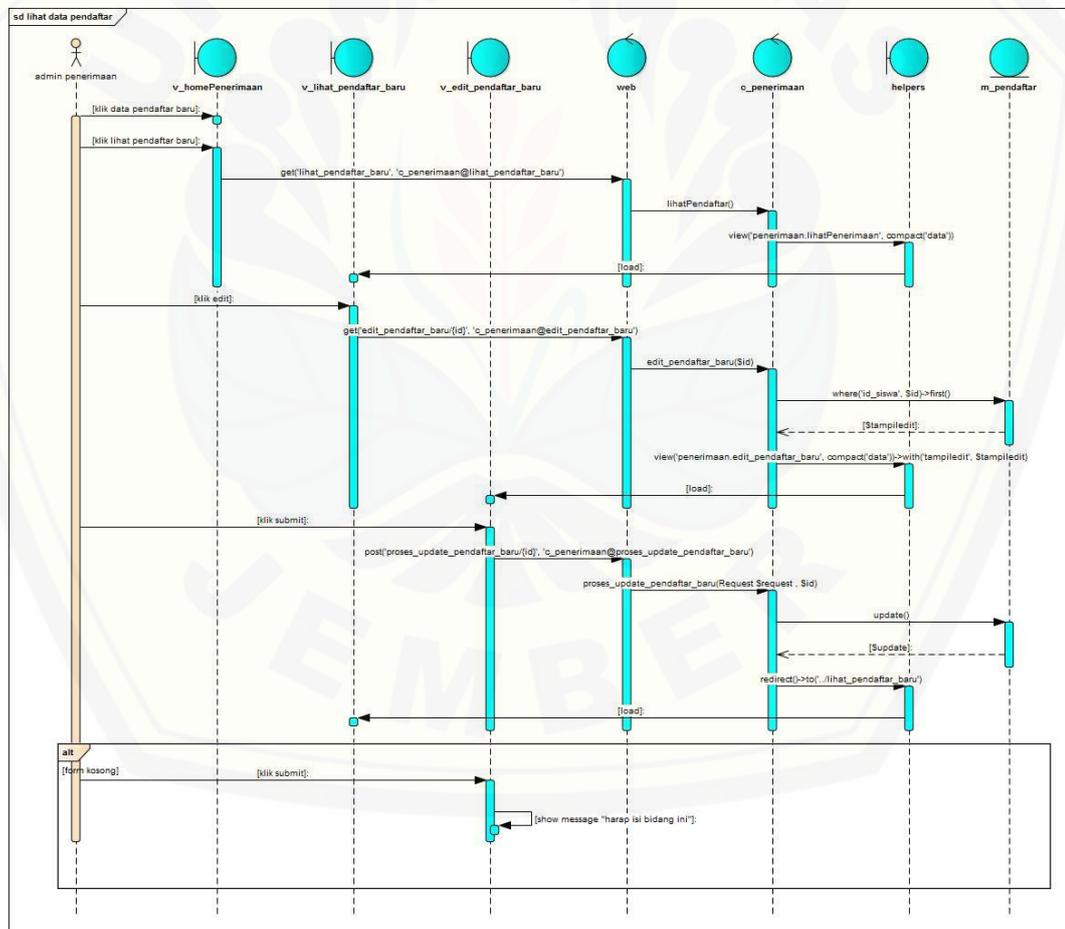
Gambar 9. Sequence Diagram Pendaftaran

10. Sequence Diagram Lihat Data Pendaftar



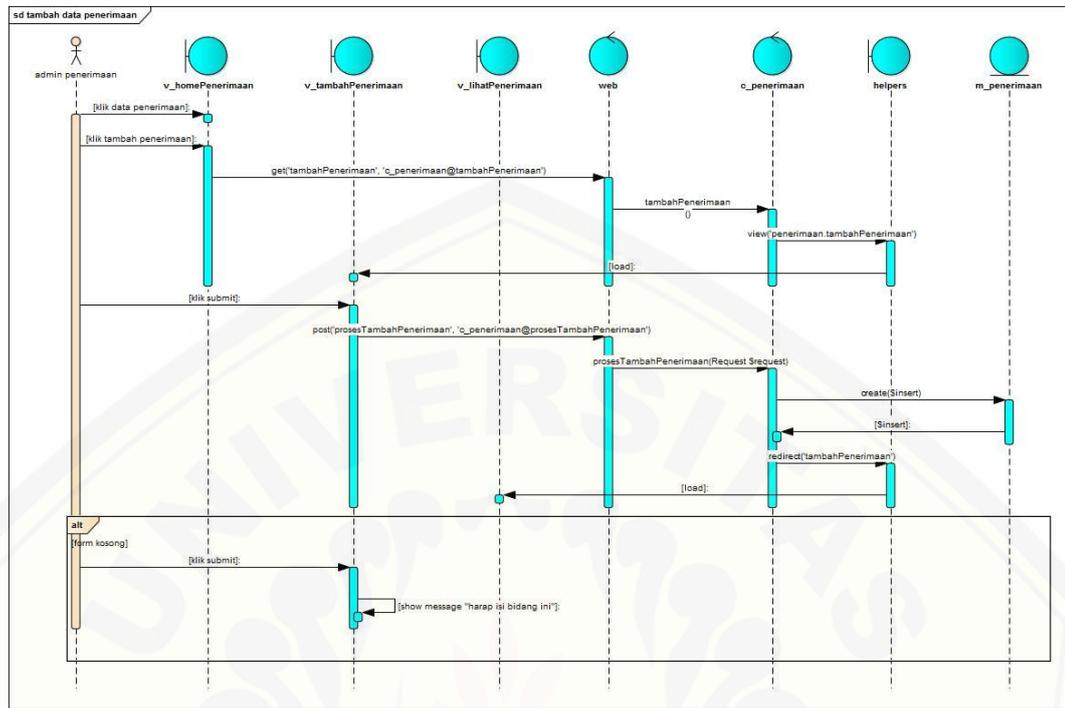
Gambar 10. Sequence Diagram Pendaftaran

11. Sequence Diagram Input Nilai Hasil Seleksi



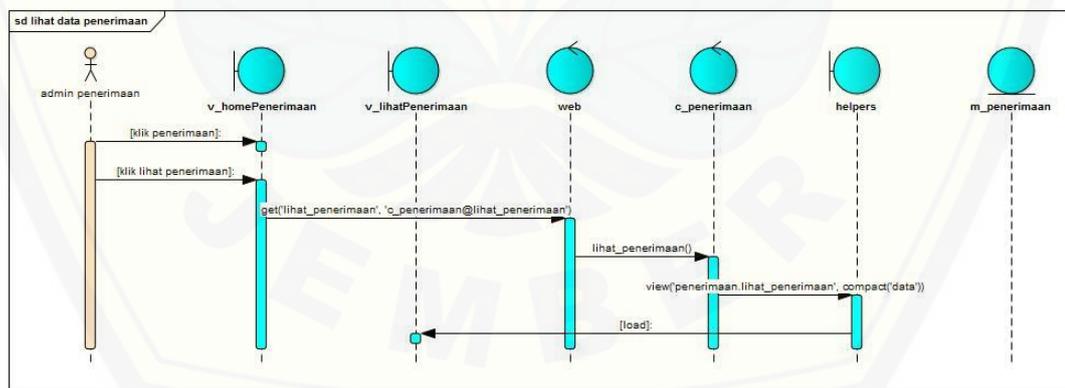
Gambar 11. Sequence Diagram Input Nilai Hasil Seleksi

### 12. Sequence Diagram Menambah Data Penerimaan



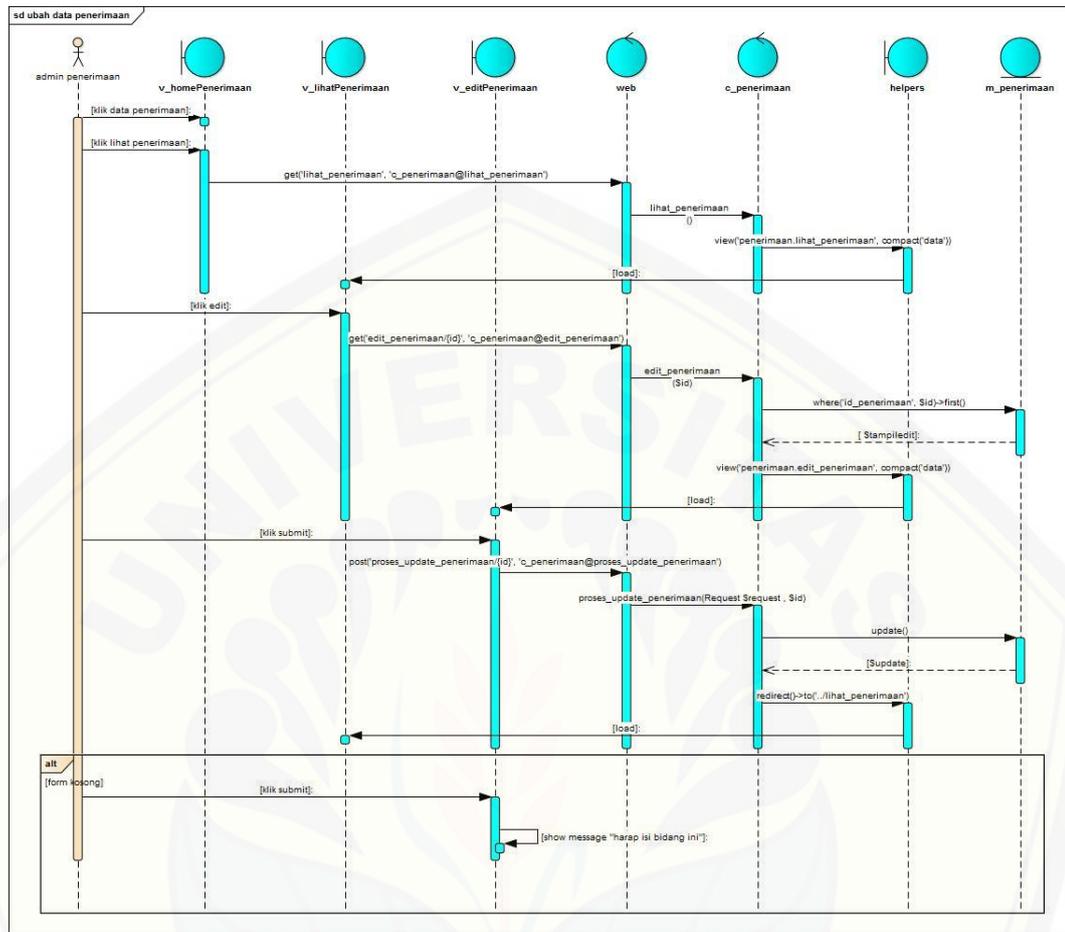
Gambar 12. Sequence Diagram Menambah Data Penerimaan

### 13. Sequence Diagram Melihat Data Penerimaan



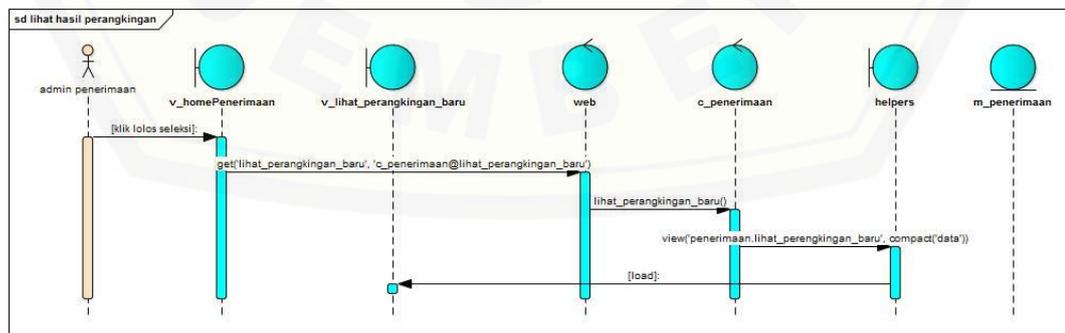
Gambar 13. Sequence Diagram Melihat Data Penerimaan

14. Sequence Diagram Mengubah Data Penerimaan



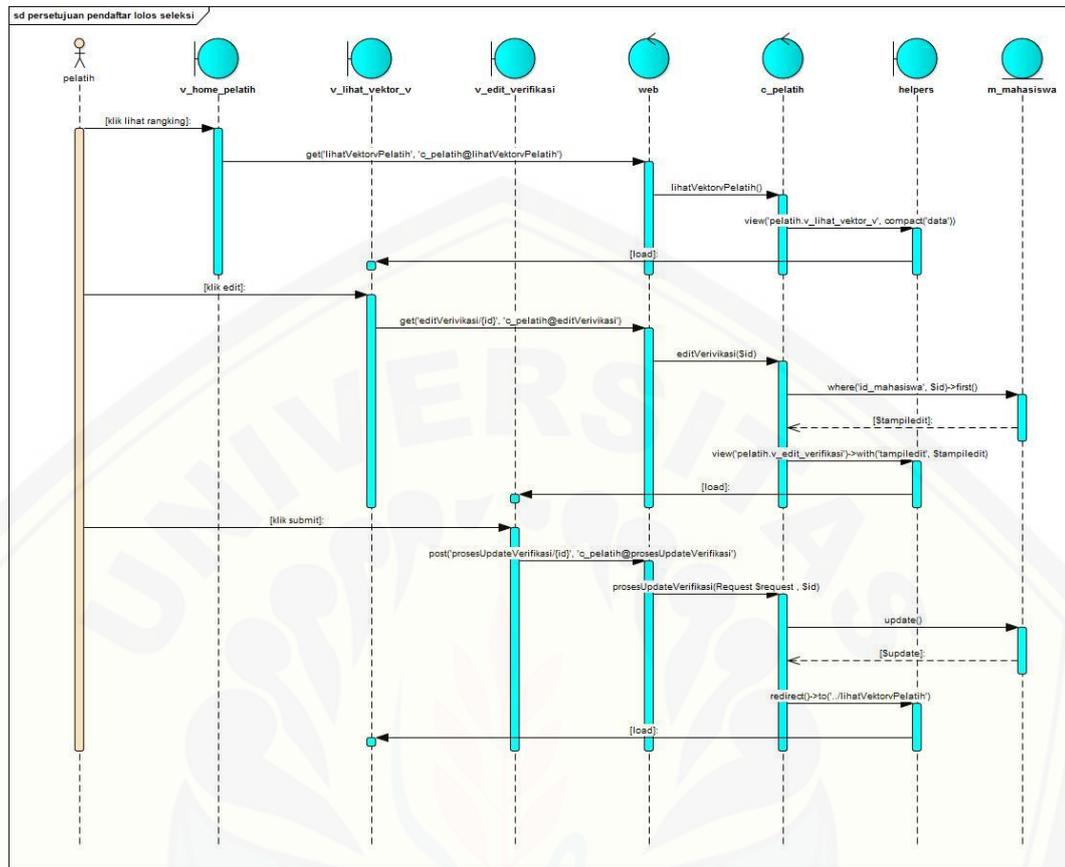
Gambar 14. Sequence Diagram Mengubah Data Penerimaan

15. Sequence Diagram Lihat Hasil Perangkingan



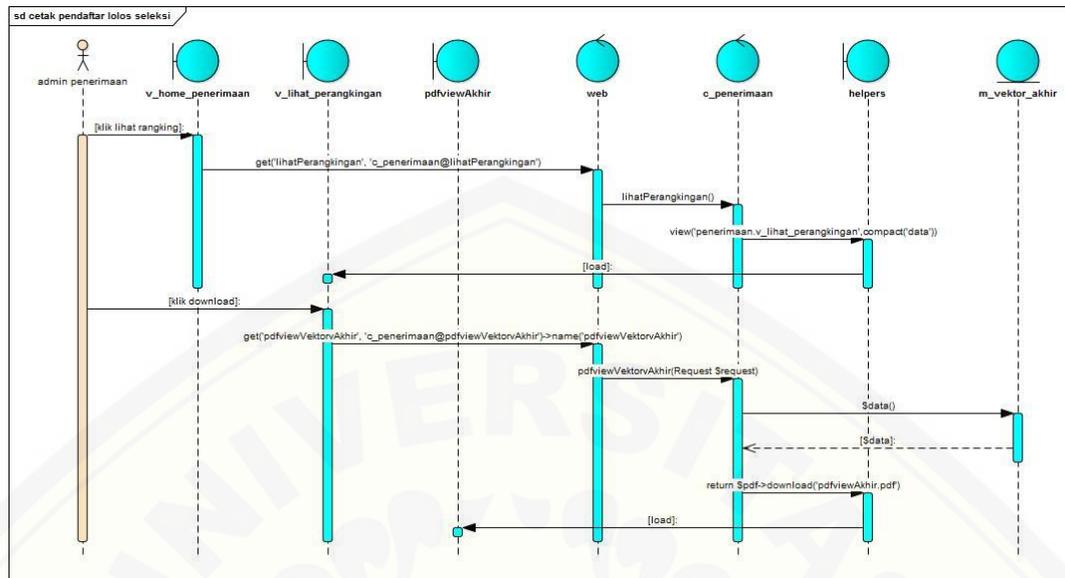
Gambar 15. Sequence Diagram Lihat Hasil Perangkingan

16. Sequence Diagram Persetujuan Pendaftar Lolos Seleksi



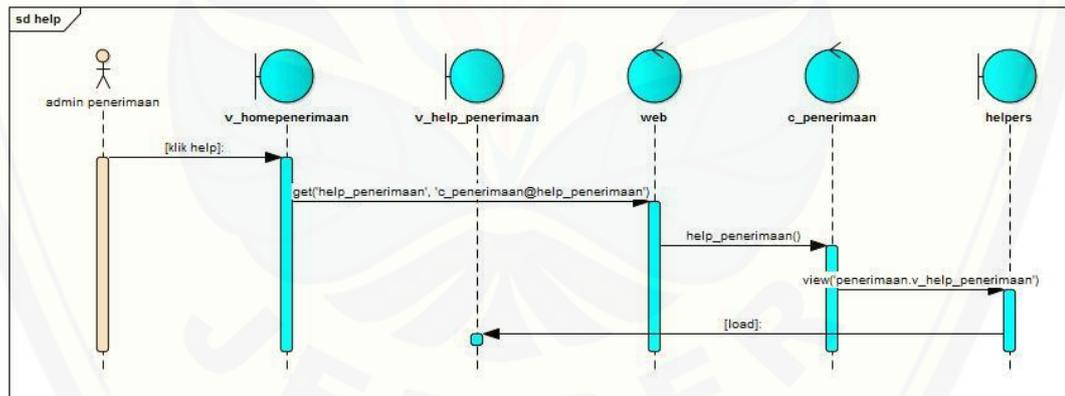
Gambar 16. Sequence Diagram Persetujuan Pendaftar Lolos Seleksi

17. Sequence Diagram Cetak Pendaftar Disetujui



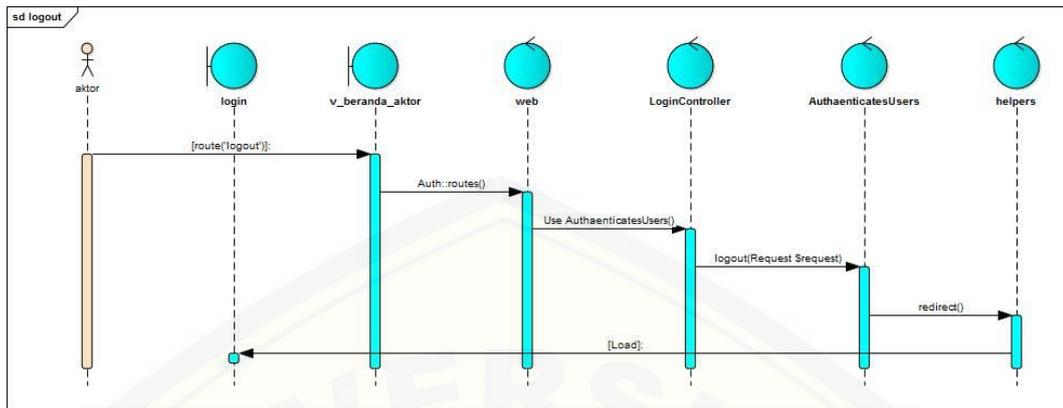
Gambar 17. Sequence Diagram Cetak Pendaftar Lolos Seleksi

18. Sequence Diagram Help



Gambar 18. Sequence Diagram Help

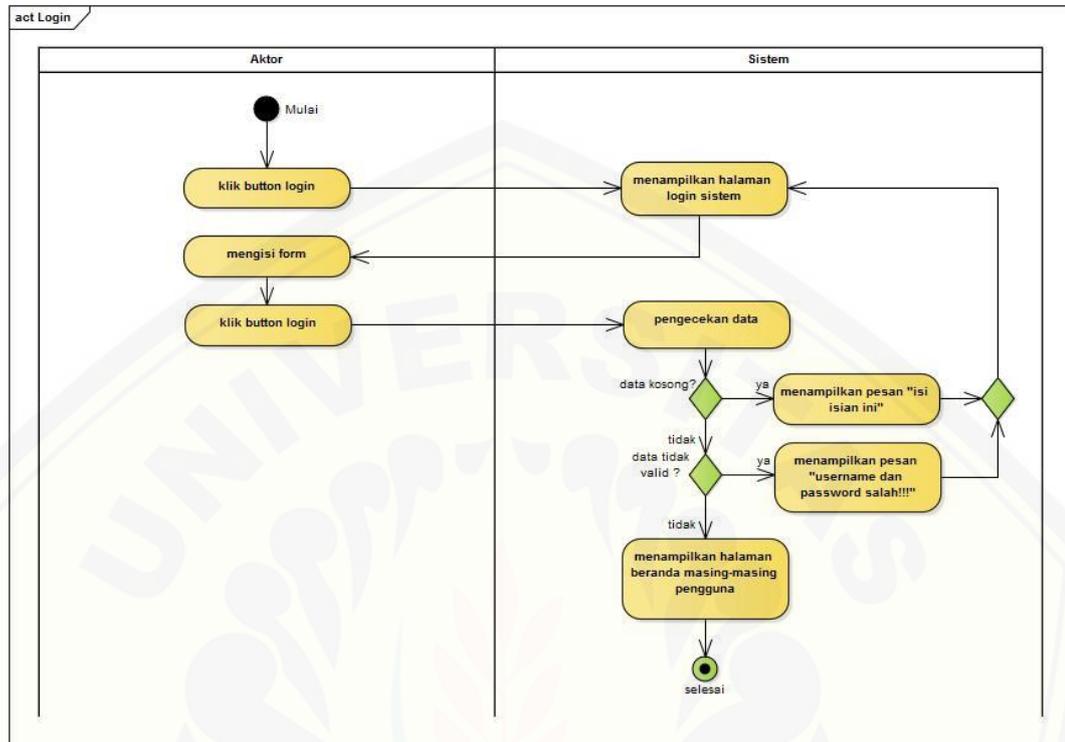
19. Sequence Diagram Logout



Gambar 19. Sequence Diagram logout

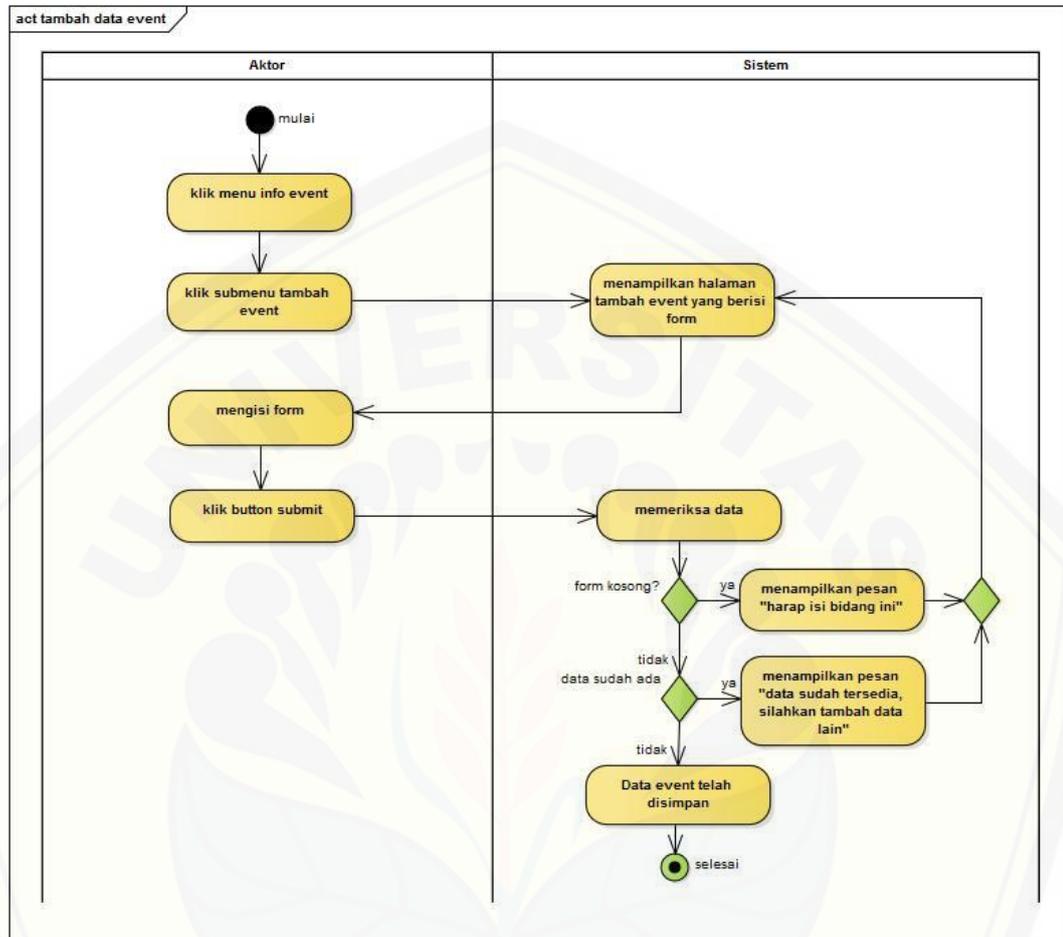
LAMPIRAN C. ACTIVITY DIAGRAM

1. Activity Diagram Masuk



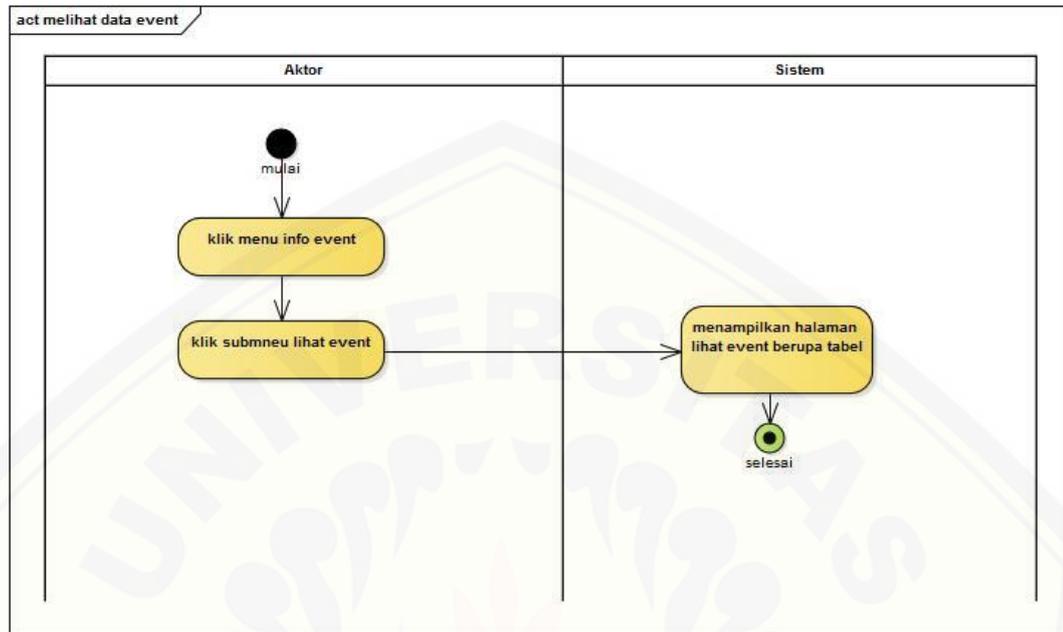
Gambar 1. Activity Diagram Login

## 2. Activity Diagram Menambah Data Event



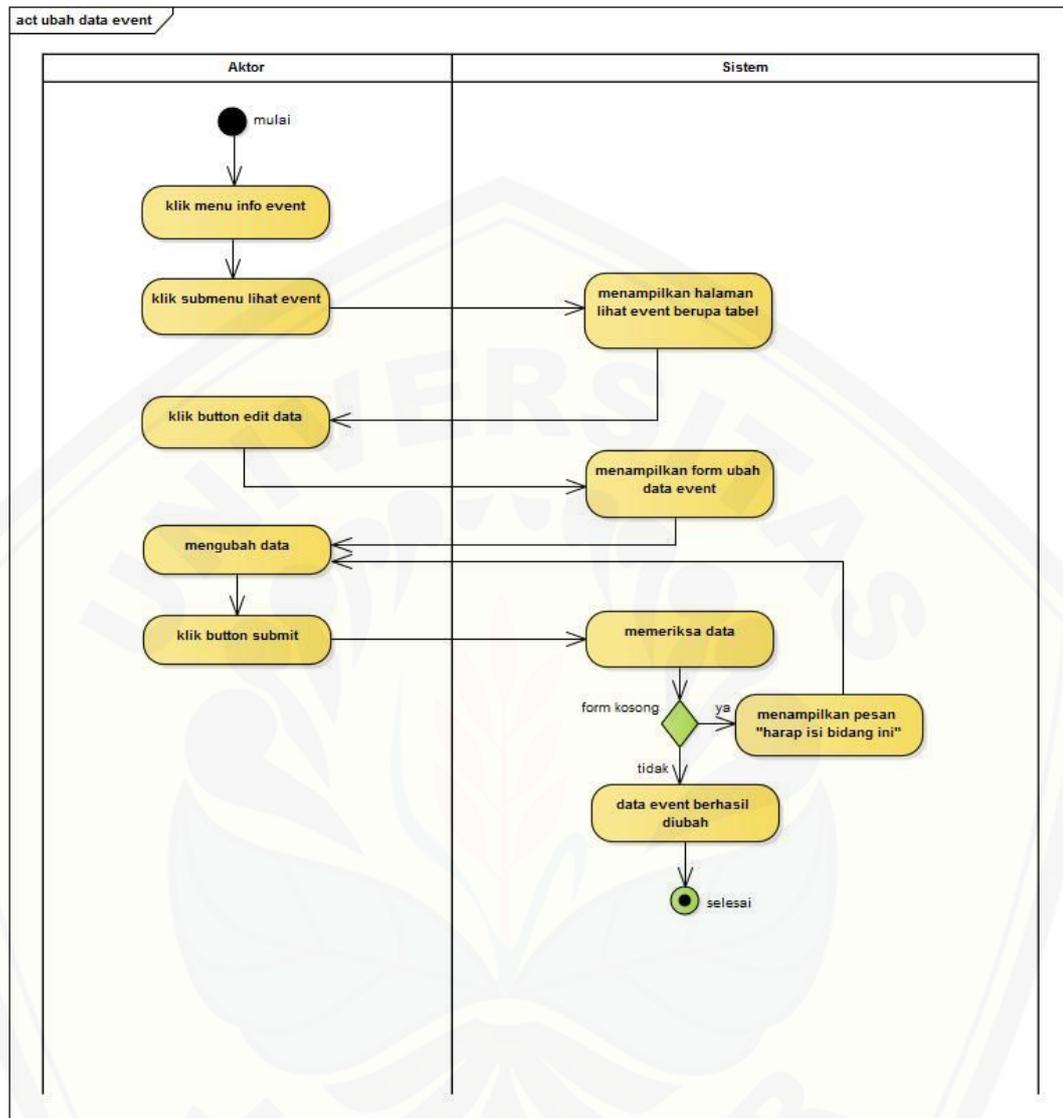
Gambar 2. Activity Diagram Menambah Data Event

### 3. Activity Diagram Melihat Data Event



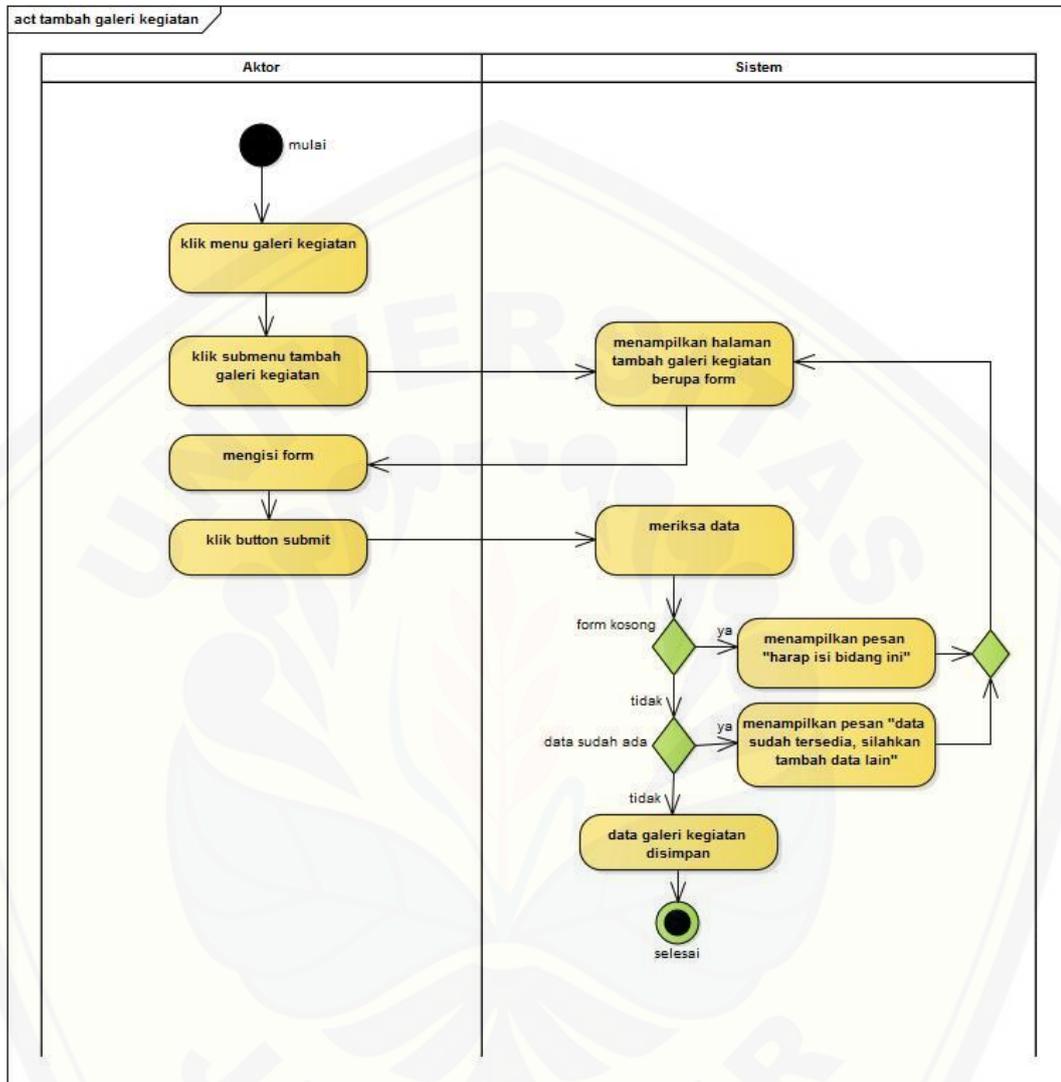
Gambar 3. Activity Diagram Melihat Data Event

## 4. Activity Diagram Mengubah Data Event



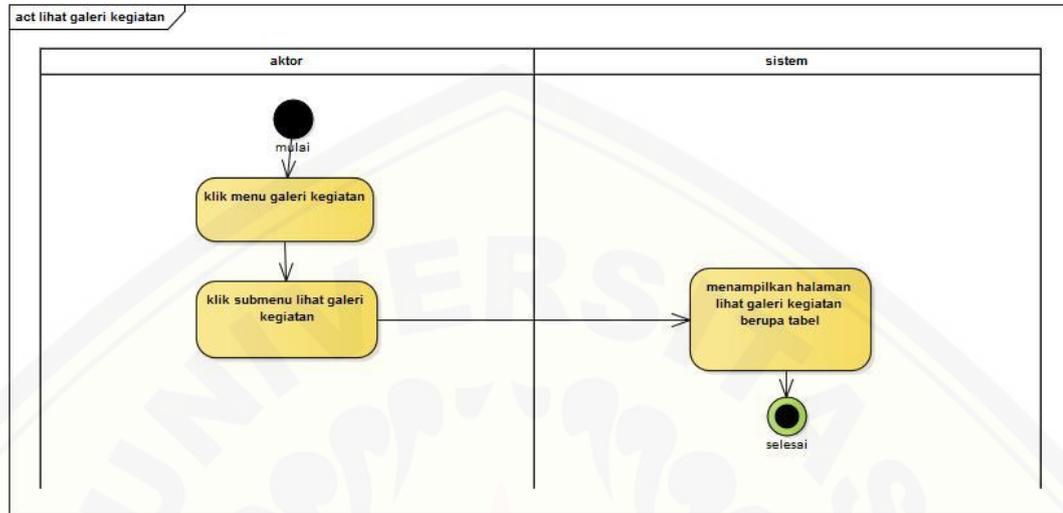
Gambar 4. Activity Diagram Mengubah Data Event

5. Activity Diagram Menambah Galeri Kegiatan



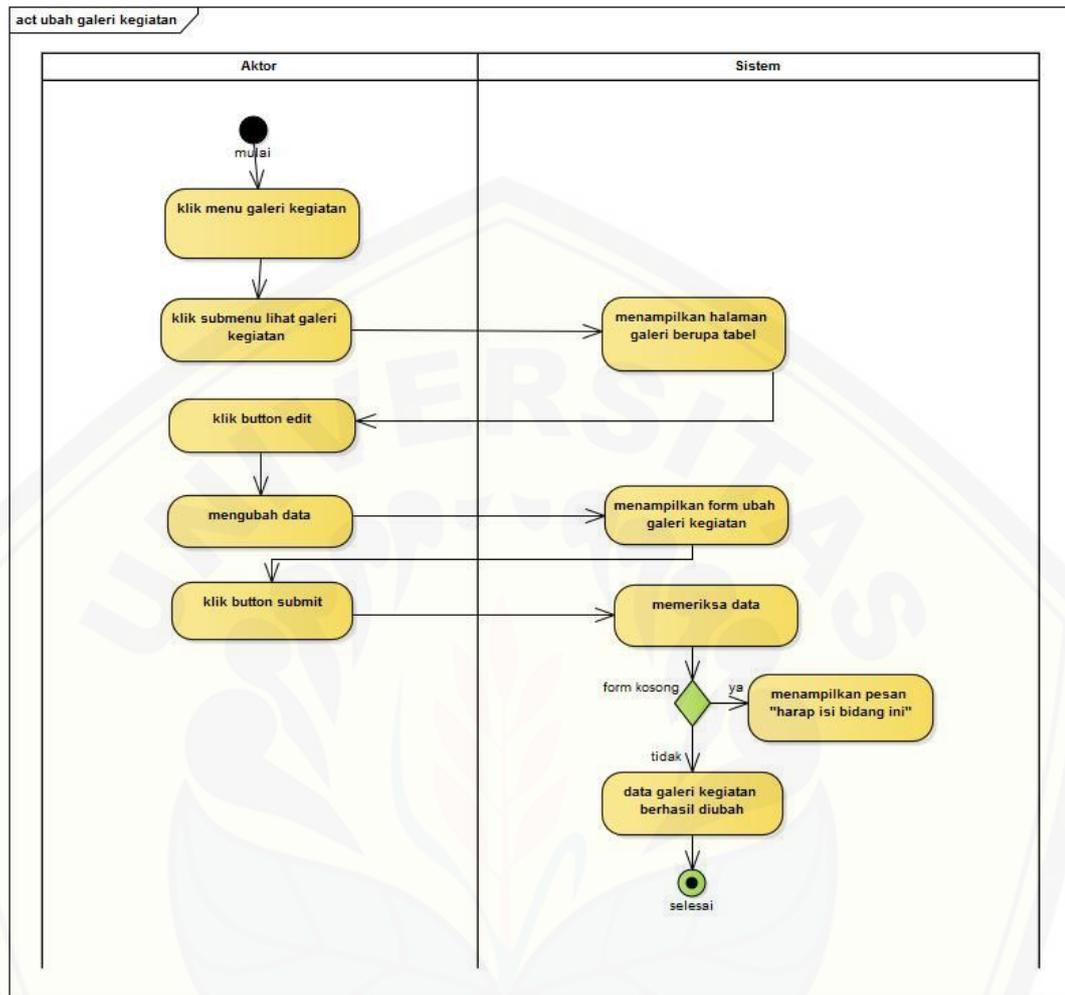
Gambar 5. Activity Diagram Menambah Galeri Kegiatan

### 6. Activity Diagram Melihat Galeri Kegiatan



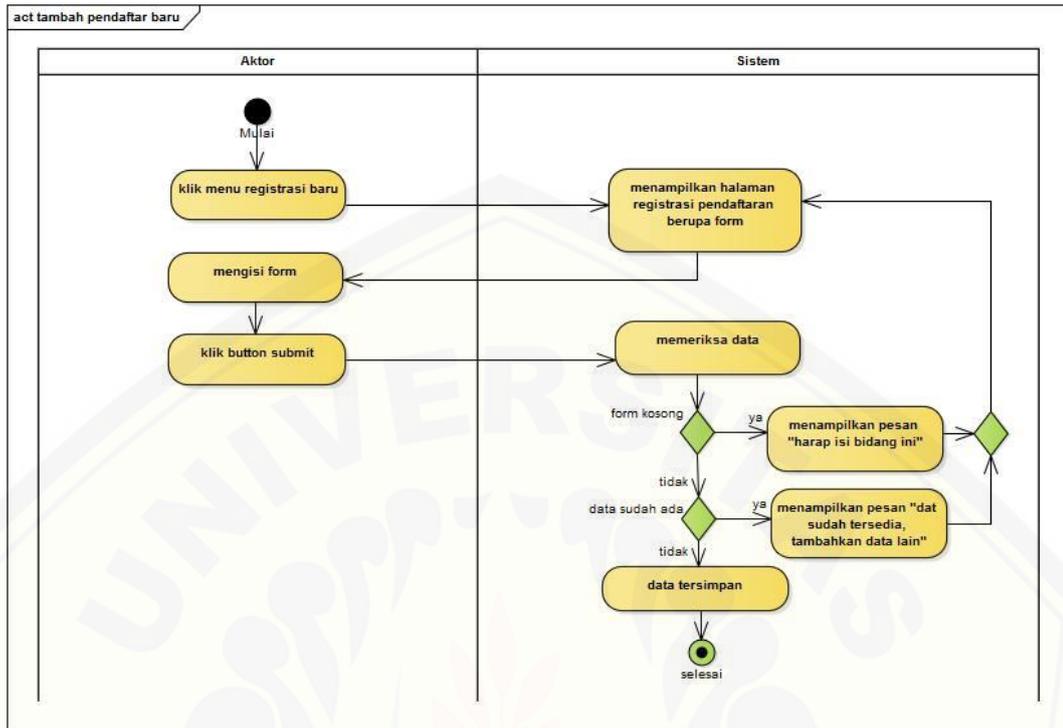
Gambar 6. Activity Diagram Melihat Galeri Kegiatan

7. Activity Diagram Mengubah Galeri Kegiatan



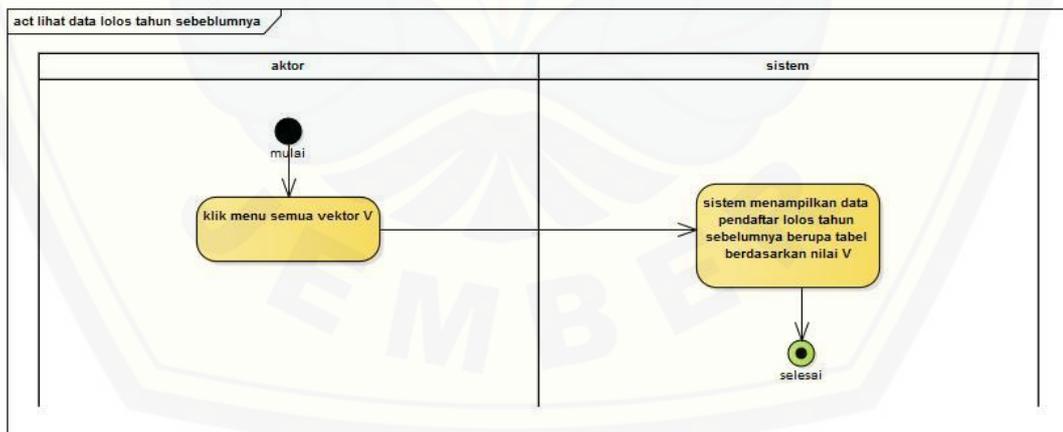
Gambar 7. Activity Diagram Mengubah Galeri Kegiatan

8. Activity Diagram Pendaftaran



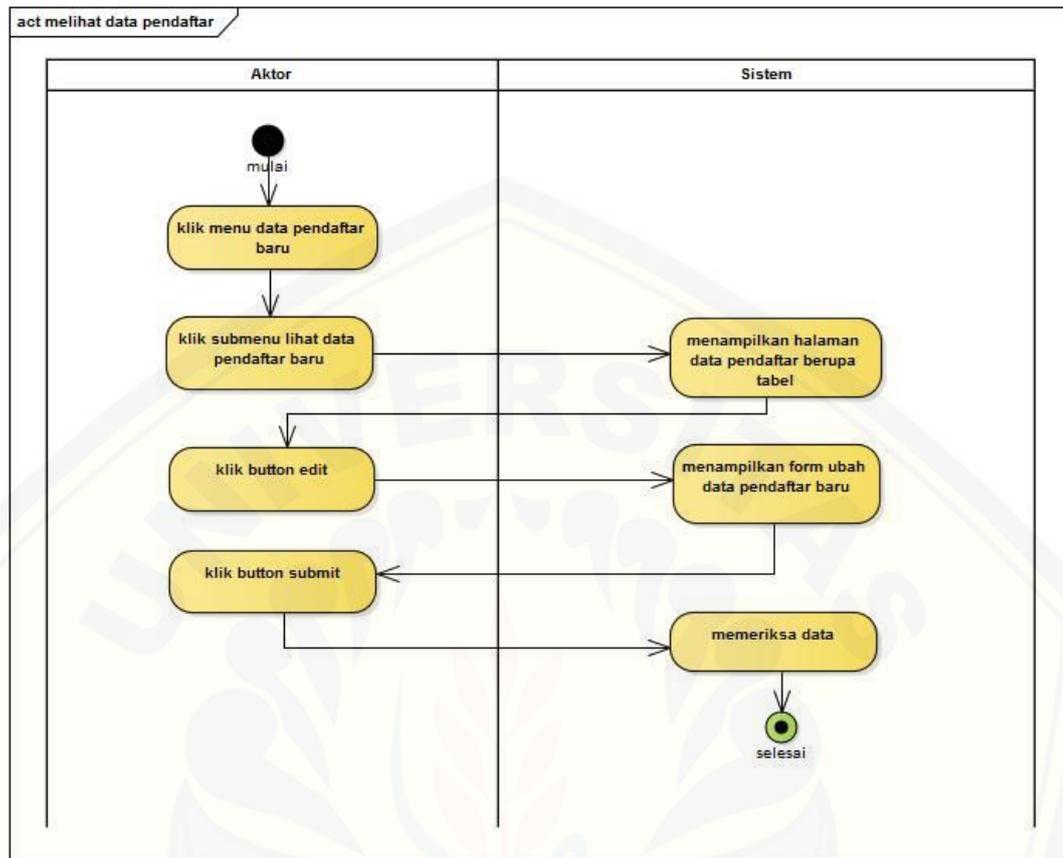
Gambar 2. Activity Diagram Pendaftaran

9. Activity Diagram Melihat Data Lolos Tahun Sebelumnya



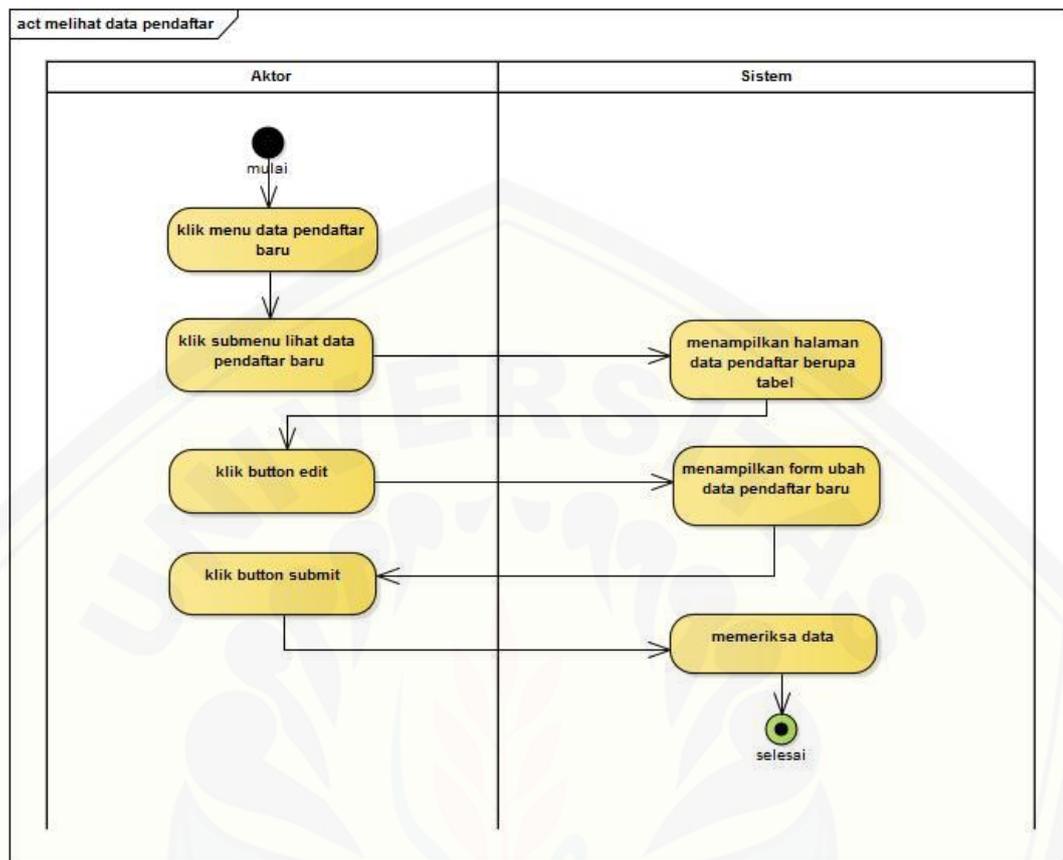
Gambar 9. Activity Diagram Melihat Data Lolos Tahun Sebelumnya

## 10. Activity Diagram Lihat Data Pendaftar



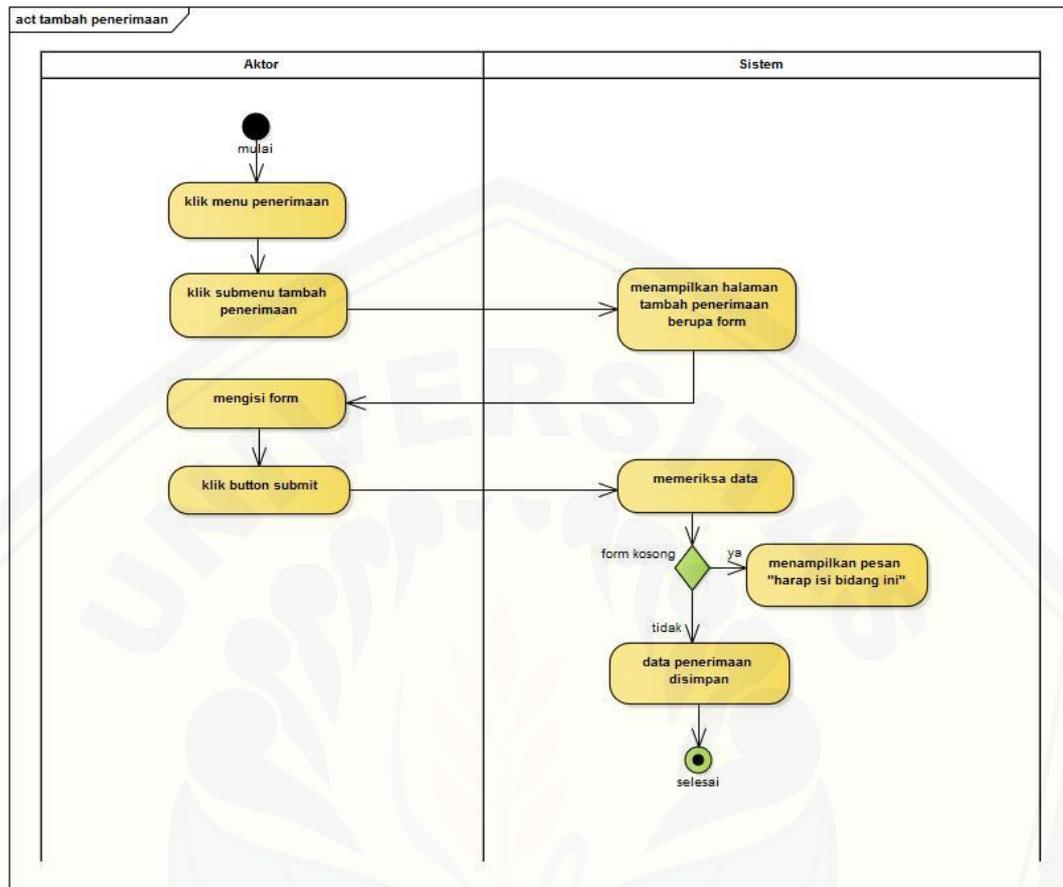
Gambar 10. Activity Diagram Melihat Data Pendaftar

## 11. Activity Diagram Input Nilai Hasil Seleksi



Gambar 11. Activity Diagram Input Nilai Hasil Seleksi

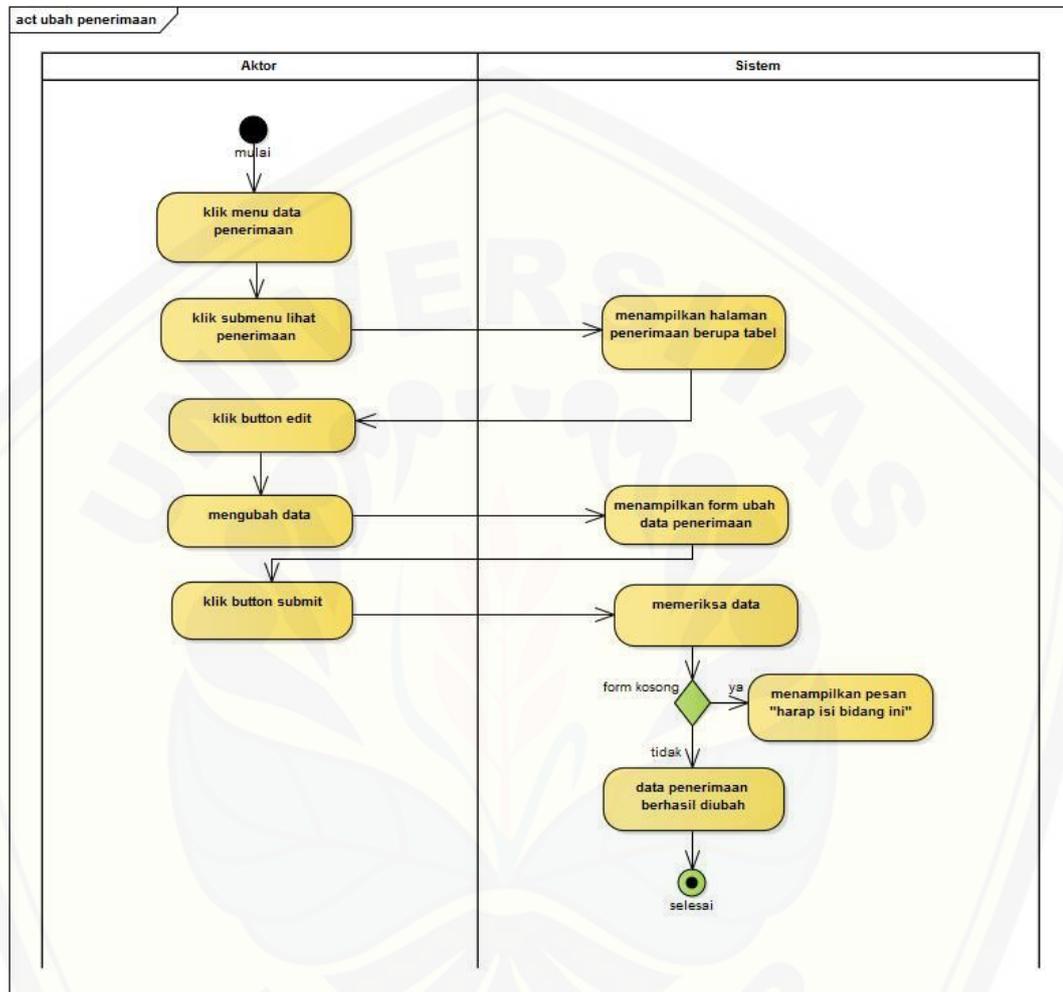
## 12. Activity Diagram Menambah Data Penerimaan



Gambar 12. Activity Diagram Menambah Data Penerimaan

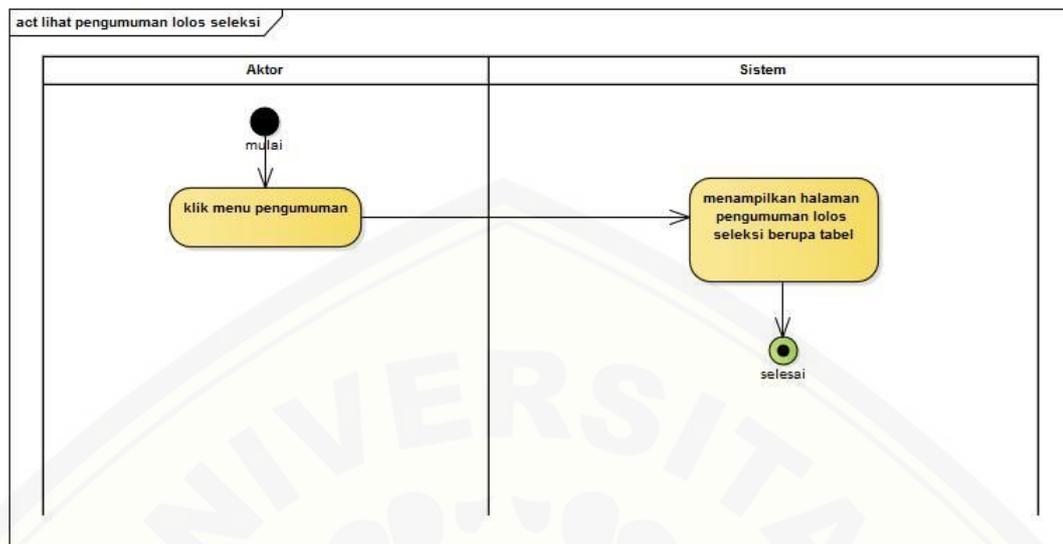
## 13. Activity Diagram Melihat Data Penerimaan

## 14. Activity Diagram Mengubah Data Penerimaan



Gambar 14. Activity Diagram Mengubah Data Penerimaan

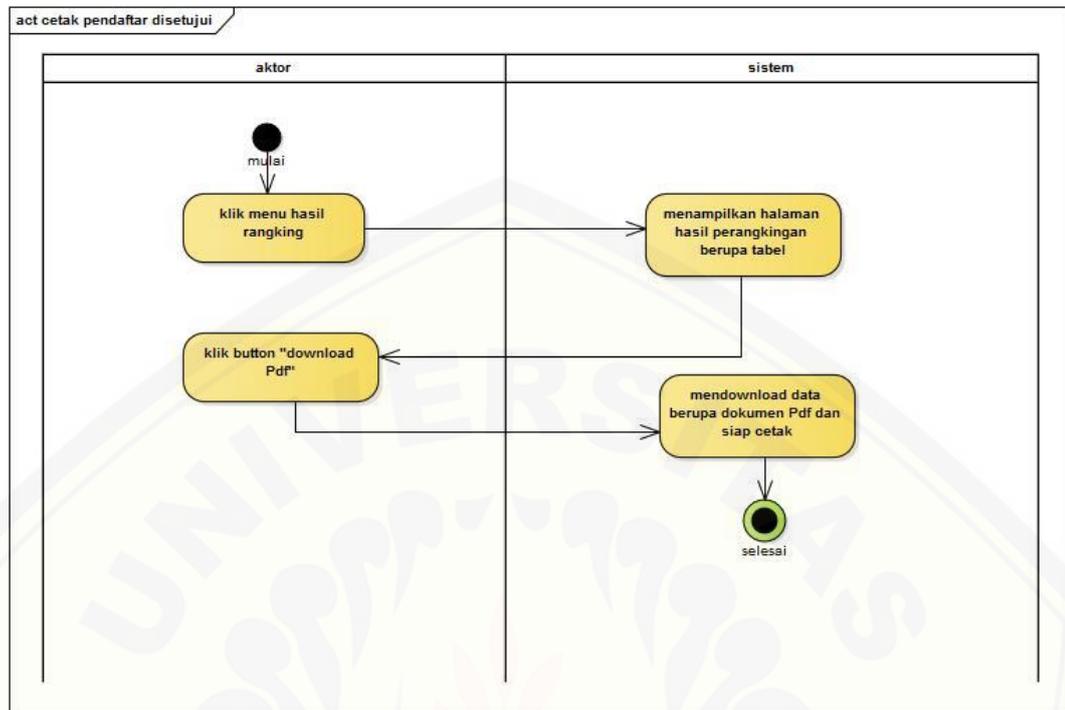
### 15. Activity Diagram Melihat Hasil Perangkingan



Gambar 15. Activity Diagram Melihat Hasil Perangkingan

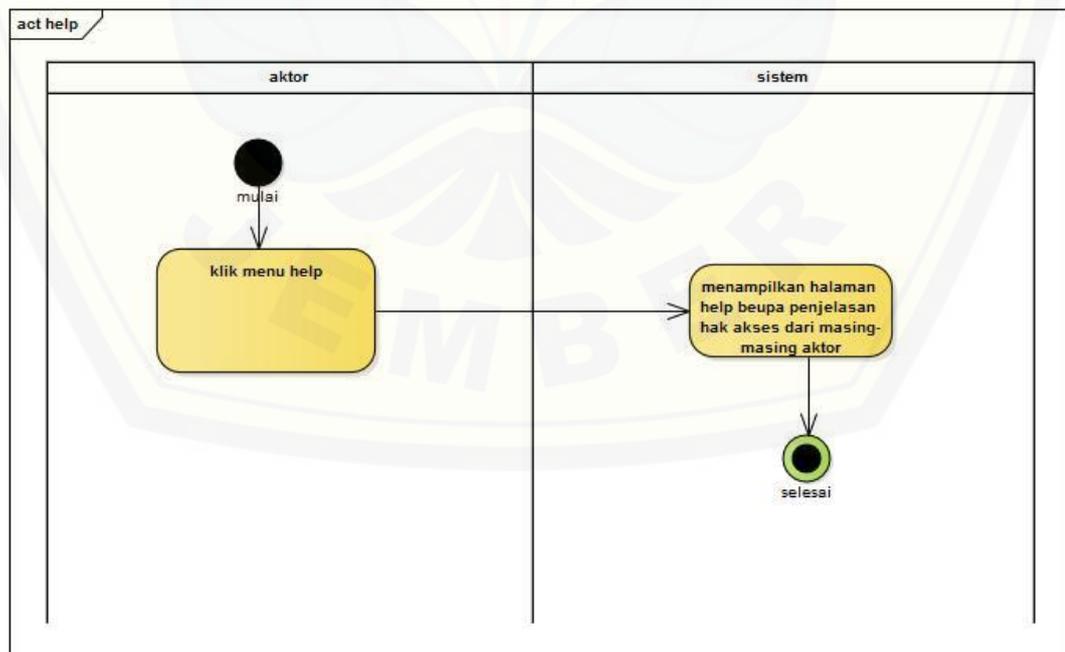
### 16. Activity Diagram Persetujuan Pendaftar Lolos Seleksi

17. Activity Diagram Cetak Pendaftar Disetujui



Gambar 17. Activity Diagram Cetak Pendaftar Disetujui

18. Activity Diagram Help



Gambar 17. Activity Diagram Help

## LAMPIRAN D. TRANSKRIP WAWANCARA

Wawancara Objek Penelitian

Narasumber : Indah Muryati, Rokhmat Hidayanto, S.

Jabatan : Ketua dan Pelatih Paduan Suara Universitas Jember

1. Bagaimana proses seleksi penerimaan anggota paduan suara Universitas Jember?

**Jawaban:**

Proses seleksi penerimaan anggota paduan suara Universitas Jember dari dulu hingga sekarang masih menggunakan sistem yang manual. Proses seleksi dilakukan dengan dua tahap, yang pertama tahap tes kemampuan. Tahap tes kemampuan yang pertama yaitu kemampuan calon anggota dalam menyanyikan nada tone yang sebenarnya (intonasi). Kedua kemampuan dalam membaca not balok. Ketiga kemampuan dalam membaca not angka. Keempat artikulasi. Kelima kemampuan dalam power suara. Setelah calon anggota mengikuti tes kemampuan, pelatih memberikan nilai kepada setiap calon anggota yang mengikuti tes pertama. Nilai kemudian di berikan kepada admin penerimaan untuk diolah. Hasil nilai akhir akan di pilih 30 nilai teratas berdasarkan rangking. Tahap kedua yaitu tes wawancara, tes wawancara dilakukan setelah calon anggota lolos pada tes pertama. Tes wawancara dilakukan oleh panitia seleksi penerimaan.

2. Data apa sajakah yang diperlukan dalam proses seleksi penerimaan?

**Jawaban:**

Data yang diperlukan dalam proses seleksi penerimaan pertama yaitu data pendaftar seperti nama, nim, asal fakultas, prodi, dll. Kedua data kriteria yang digunakan untuk proses seleksi penerimaan serta data nilai bobot dari masing-masing kriteria

3. Kriteria apa sajakah yang digunakan dalam proses seleksi penerimaan anggota paduan suara universitas Jember?

**Jawaban:**

Kriteria yang digunakan ada 5, pertama kriteria intonasi, kedua kemampuan membaca not balok, ketiga kemampuan membaca not angka, keempat artikulasi, dan kelima power suara. Tapi kriteria yang digunakan dalam proses seleksi tidak bisa diubah ataupun ditambah.

4. Apakah kriteria dan nilai bobot yang digunakan tidak berubah setiap tahunnya / setiap proses penerimaan anggota paduan suara?

**Jawaban:**

Untuk kriteria yang digunakan tidak berubah (tetap), namun untuk nilai bobot yang digunakan akan berubah setiap tahunnya sesuai dengan kebutuhan dari tingkat kepentingan atau urgency PSM unej. Tahun 2017, nilai bobot yang paling besar ditekankan pada kriteria intonasi dan power suara dengan nilai 5, kriteria artikulasi memiliki bobot 4, kriteria not balok dan not angka memiliki nilai bobot 3.

## LAMPIRAN E. TAMPILAN SISTEM

## 1. Tampilan Melihat Data Event



ADMIN PADUAN SUARA UJ

ADMIN PADUAN SUARA UJ

LIHAT EVENT  
Paduan Suara Unej

23:59  
Sunday, June 3rd 2018

- DASHBOARD
- HELP
- GALERI
- EVENT

Nama Event	Tanggal Event	Tempat Event	Penyelenggara Event	Gambar Event	Edit
Event A	2018-05-31	GEDUNG SOETARJO	PANITIA PSM MABA		Edit
Event B	2018-05-26	GEDUNG SOERACHMAN	PADUAN SUARA UNEJ		Edit
Event C	2018-06-02	GEDUNG SOETARJO	PADUAN SUARA UNEJ		Edit
Event D	2018-06-08	GEDUNG SOETARJO	PANITIA PSM MABA		Edit
Event E	2018-05-09	GEDUNG SOETARJO	PANITIA PSM MABA		Edit

Gambar 1. Tampilan Melihat Data Event

## 2. Tampilan Mengubah Data Event



ADMIN PADUAN SUARA UJ

ADMIN PADUAN SUARA UJ

TAMBAH EVENT  
Paduan Suara Unej

00:00  
Monday, June 4th 2018

- DASHBOARD
- GALERI
- EVENT

Nama Event:

Tanggal Event:

Tempat Event:

Penyelenggara Event:

GAMBAR EVENT: 

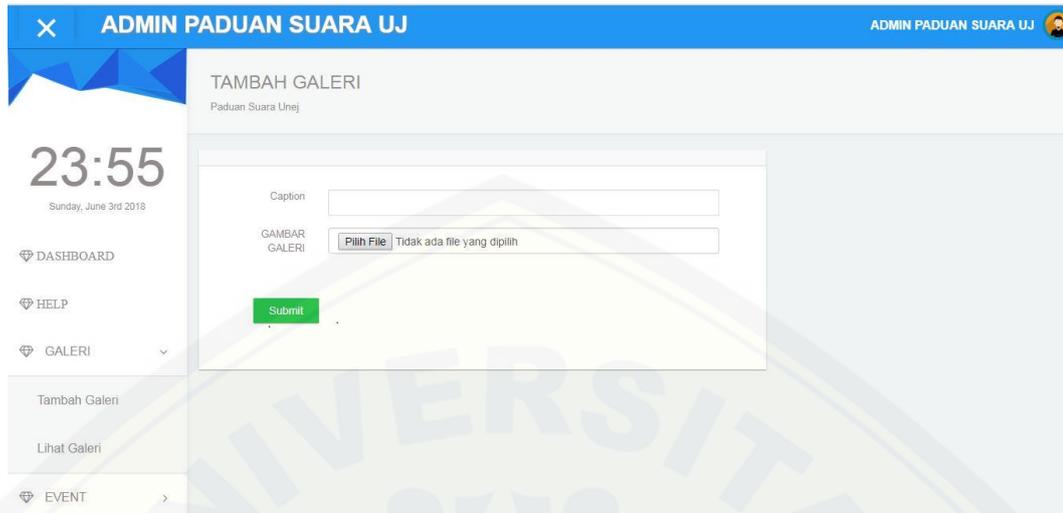
NAMA FILE: GEDUNG SOETARJO

INPUT:  Tidak ada file yang dipilih

Submit

Gambar 2. Tampilan Mengubah Data Event

### 3. Tampilan Menambah Galeri Kegiatan



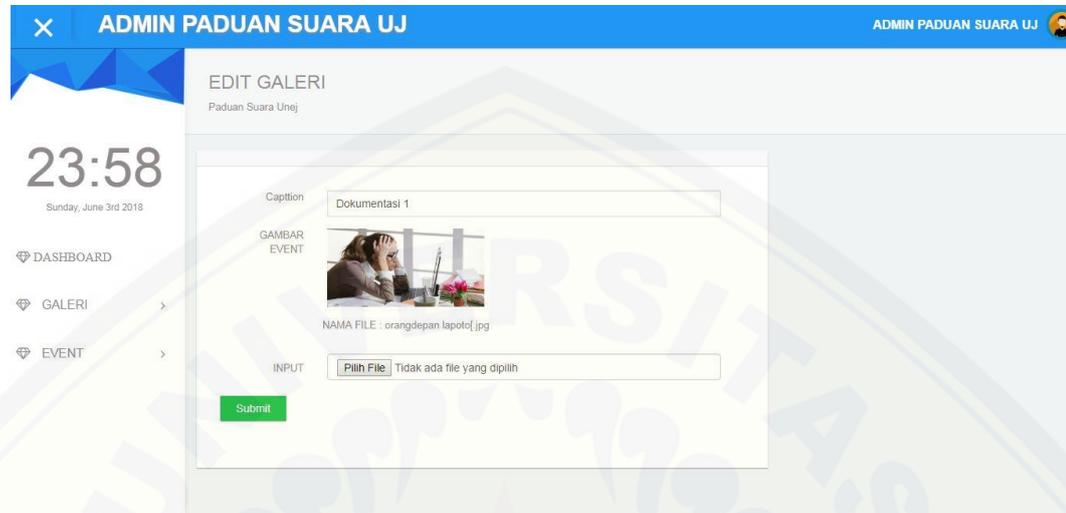
Gambar 3. Tampilan Menambah Galeri Kegiatan

### 4. Tampilan Melihat Galeri Kegiatan



Gambar 4. Tampilan Melihat Galeri Kegiatan

## 5. Tampilan Mengubah Galeri Kegiatan



**ADMIN PADUAN SUARA UJ** ADMIN PADUAN SUARA UJ

### EDIT GALERI

Paduan Suara Unej

23:58  
Sunday, June 3rd 2018

- DASHBOARD
- GALERI >
- EVENT >

Caption: Dokumentasi 1

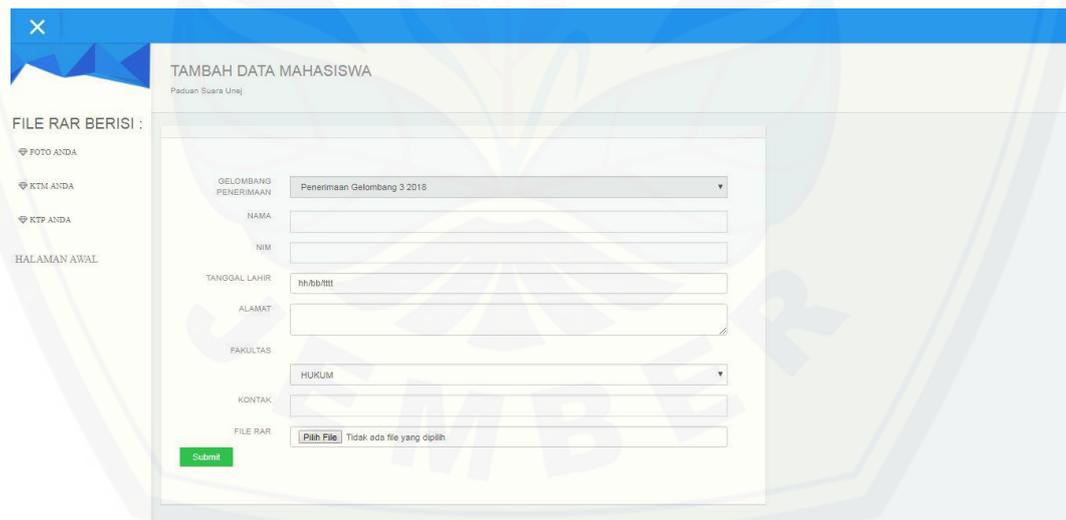
GAMBAR EVENT: 

NAMA FILE : orangdepan lapotoj.jpg

INPUT:  Tidak ada file yang dipilih

Gambar 5. Tampilan Mengubah Galeri Kegiatan

## 6. Tampilan Pendaftaran



**ADMIN PADUAN SUARA UJ**

### TAMBAH DATA MAHASISWA

Paduan Suara Unej

FILE RAR BERISI :

- FOTO ANDA
- KTM ANDA
- KTP ANDA
- HALAMAN AWAL

GELOMBANG PENERIMAAN: Penerimaan Gelombang 3 2018

NAMA:

NIM:

TANGGAL LAHIR: hh:bb:tttt

ALAMAT:

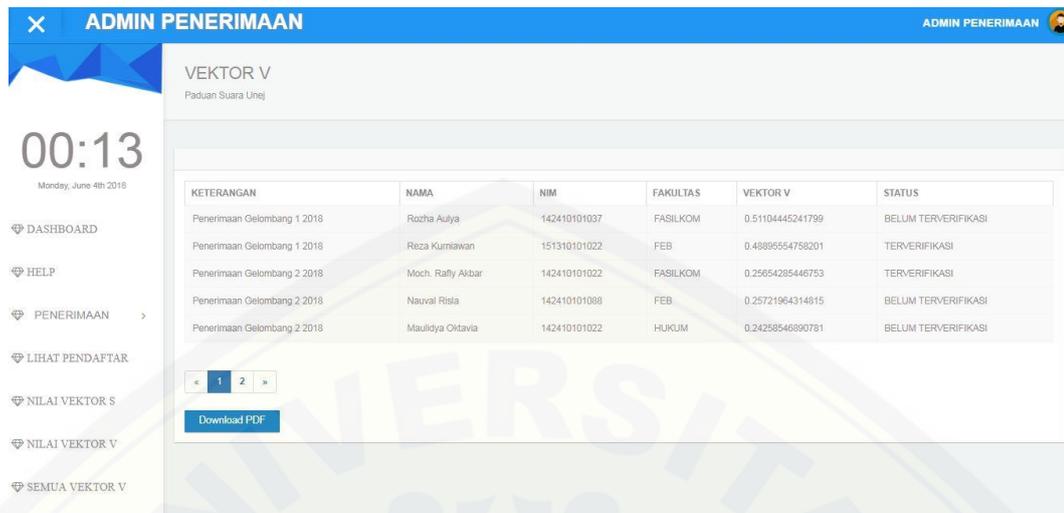
FAKULTAS: HUKUM

KONTAK:

FILE RAR:  Tidak ada file yang dipilih

Gambar 6. Tampilan Pendaftaran

## 7. Tampilan Melihat Data Lolos Tahun Sebelumnya

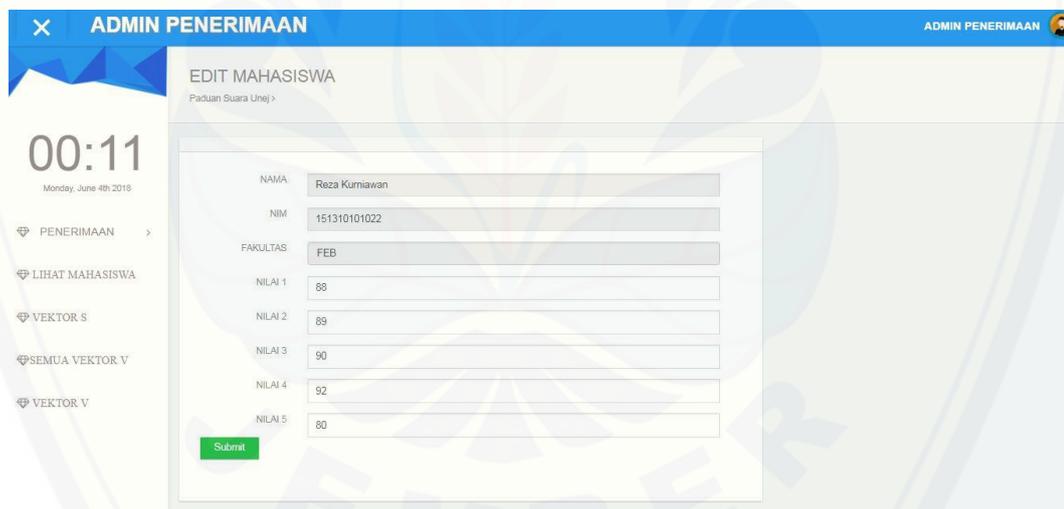


The screenshot shows the 'ADMIN PENERIMAAN' interface. The main content area displays a table titled 'VEKTOR V' with the subtitle 'Paduan Suara Unej'. The table lists admission data for the previous year. The table has six columns: KETERANGAN, NAMA, NIM, FAKULTAS, VEKTOR V, and STATUS. Below the table is a pagination control showing '1' and '2' with arrows, and a 'Download PDF' button.

KETERANGAN	NAMA	NIM	FAKULTAS	VEKTOR V	STATUS
Penerimaan Gelombang 1 2018	Rozha Aulya	142410101037	FASILKOM	0.51104445241799	BELUM TERVERIFIKASI
Penerimaan Gelombang 1 2018	Reza Kurniawan	151310101022	FEB	0.48895554758201	TERVERIFIKASI
Penerimaan Gelombang 2 2018	Moch. Rafly Akbar	142410101022	FASILKOM	0.25654285446753	TERVERIFIKASI
Penerimaan Gelombang 2 2018	Nauval Risa	142410101088	FEB	0.25721964314815	BELUM TERVERIFIKASI
Penerimaan Gelombang 2 2018	Maulidya Oktavia	142410101022	HUKUM	0.24258546890781	BELUM TERVERIFIKASI

Gambar 7. Tampilan Melihat Data Lolos Tahun Sebelumnya

## 8. Tampilan Input Nilai Hasil Seleksi

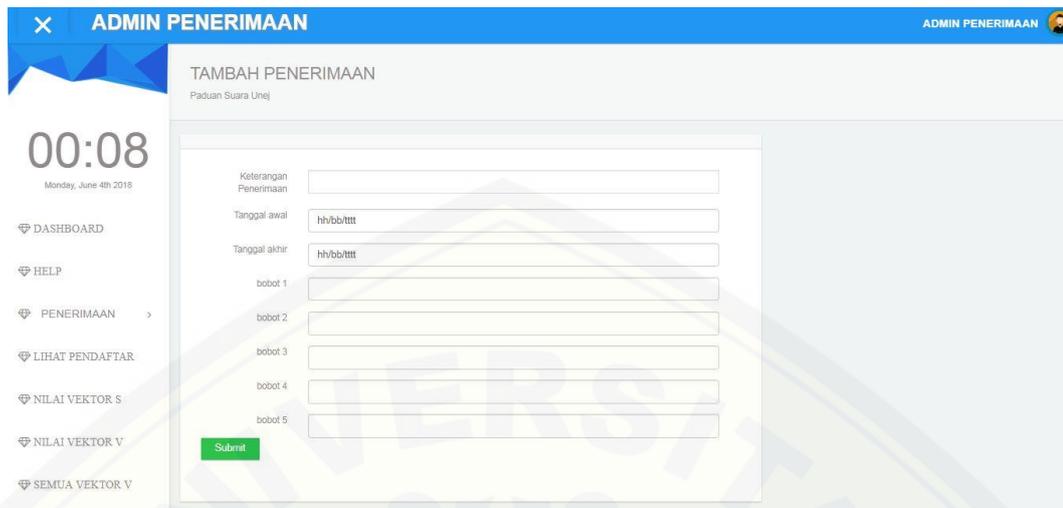


The screenshot shows the 'ADMIN PENERIMAAN' interface with the 'EDIT MAHASISWA' form. The form is for 'Reza Kurniawan' (NIM: 151310101022, FAKULTAS: FEB). It contains five input fields for 'NILAI 1' through 'NILAI 5' with values 88, 89, 90, 92, and 80 respectively. A 'Submit' button is located at the bottom left of the form.

NAMA	Reza Kurniawan
NIM	151310101022
FAKULTAS	FEB
NILAI 1	88
NILAI 2	89
NILAI 3	90
NILAI 4	92
NILAI 5	80

Gambar 8. Tampilan Input Nilai Hasil Seleksi

## 9. Tampilan Menambahkan Data Penerimaan



The screenshot shows the 'ADMIN PENERIMAAN' interface. The main heading is 'TAMBAH PENERIMAAN' with the subtitle 'Paduan Suara Unej'. The form contains the following fields:

- Keterangan Penerimaan:
- Tanggal awal:
- Tanggal akhir:
- Bobot 1:
- Bobot 2:
- Bobot 3:
- Bobot 4:
- Bobot 5:

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

Gambar 3. Tampilan Menambah Data Penerimaan

## 10. Tampilan Melihat Data Penerimaan



The screenshot shows the 'ADMIN PENERIMAAN' interface. The main heading is 'LIHAT PENERIMAAN' with the subtitle 'Paduan Suara Unej'. The table displays the following data:

Keterangan	Tanggal Awal	Tanggal Akhir	Bobot 1	Bobot 2	Bobot 3	Bobot 4	Bobot 5	Action
Penerimaan Gelombang 1 2018	2018-01-01	2018-02-01	0.25	0.15	0.15	0.2	0.25	<a href="#">Edit</a>
Penerimaan Gelombang 2 2018	2018-04-01	2018-06-01	0.25	0.2	0.25	0.15	0.15	<a href="#">Edit</a>
Penerimaan Gelombang 3 2018	2018-06-01	2018-06-08	0.0125	0.0125	0.0075	0.0075	0.01	<a href="#">Edit</a>

Gambar 10. Tampilan Melihat Data Penerimaan

## 11. Tampilan Mengubah Data Penerimaan

ADMIN PENERIMAAN

EDIT PENERIMAAN

Paduan Suara Unej

00:09  
Monday, June 4th 2018

PENERIMAAN >

LIHAT MAHASISWA

VEKTOR S

SEMUA VEKTOR V

VEKTOR V

KETERANGAN: Penerimaan Gelombang 1 2018

TANGGAL AWAL: 01/01/2018

TANGGAL AKHIR: 01/02/2018

BOBOT 1: 0.25

BOBOT 2: 0.15

BOBOT 3: 0.15

BOBOT 4: 0.2

BOBOT 5: 0.25

Submit

Gambar 11. Tampilan Mengubah Data Penerimaan

## 12. Tampilan Melihat Vektor V

ADMIN PENERIMAAN

VEKTOR V PENERIMAAN TERAKHIR

Paduan Suara Unej

00:12  
Monday, June 4th 2018

DASHBOARD

HELP

PENERIMAAN >

LIHAT PENDAFTAR

NILAI VEKTOR S

NILAI VEKTOR V

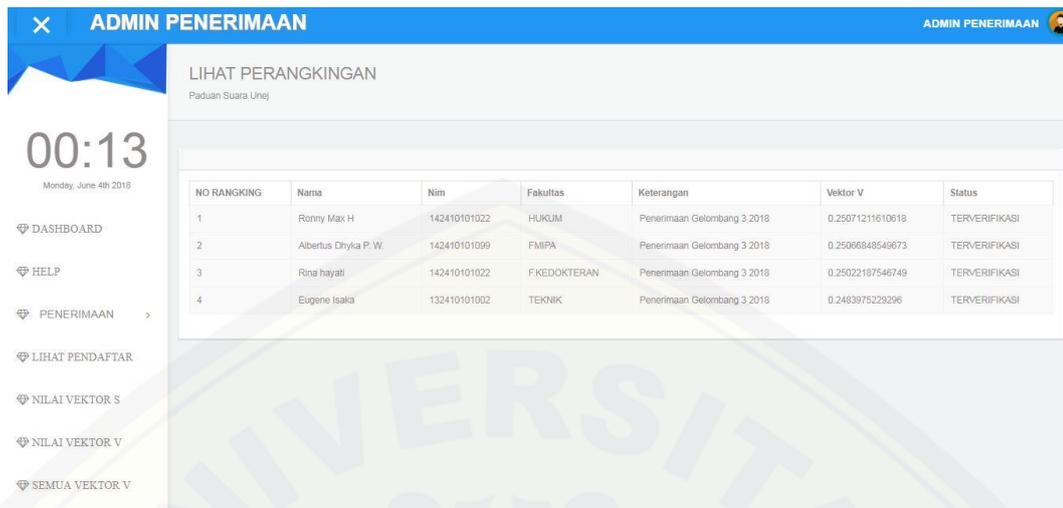
SEMUA VEKTOR V

KETERANGAN	NAMA	NIM	FAKULTAS	VEKTOR V	STATUS
Penerimaan Gelombang 3 2018	Eugene Isaka	132410101002	TEKNIK	0.2483975229296	TERVERIFIKASI
Penerimaan Gelombang 3 2018	Ronny Max H	142410101022	HUKUM	0.25071211610618	TERVERIFIKASI
Penerimaan Gelombang 3 2018	Albertus Dhyka P. W.	142410101099	FMIPA	0.25066848549673	TERVERIFIKASI
Penerimaan Gelombang 3 2018	Rina hayati	142410101022	F.KEDOKTERAN	0.25022187546749	TERVERIFIKASI

Download PDF

Gambar 12. Tampilan Melihat Vektor V

### 13. Tampilan Melihat Hasil Perangkingan

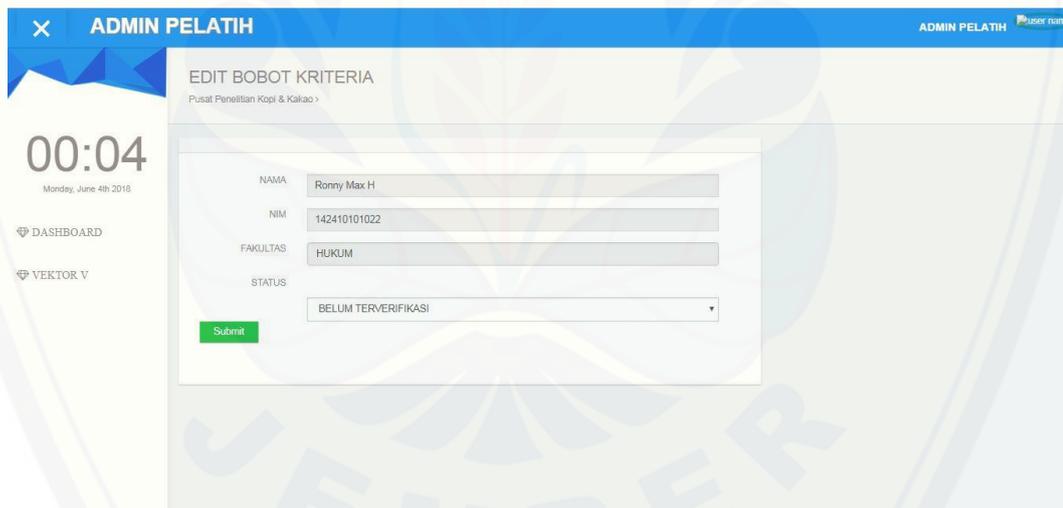


The screenshot shows the 'ADMIN PENERIMAAN' interface. The main content area is titled 'LIHAT PERANGKINGAN' and displays a table of ranking results for 'Paduan Suara Unej'. The table has the following columns: NO RANGKING, Nama, Nim, Fakultas, Keterangan, Vektor V, and Status. The data is as follows:

NO RANGKING	Nama	Nim	Fakultas	Keterangan	Vektor V	Status
1	Ronny Max H	142410101022	HUKUM	Penerimaan Gelombang 3 2018	0.25071211610618	TERVERIFIKASI
2	Albertus Dhyka P. W.	142410101099	FMIPA	Penerimaan Gelombang 3 2018	0.25066848549673	TERVERIFIKASI
3	Rina hayati	142410101022	FKEDOKTERAN	Penerimaan Gelombang 3 2018	0.25022187546749	TERVERIFIKASI
4	Eugene Isaka	132410101002	TEKNIK	Penerimaan Gelombang 3 2018	0.2483975229296	TERVERIFIKASI

Gambar 13. Tampilan Melihat Hasil Perangkingan

### 14. Tampilan Persetujuan Pendaftar Lolos



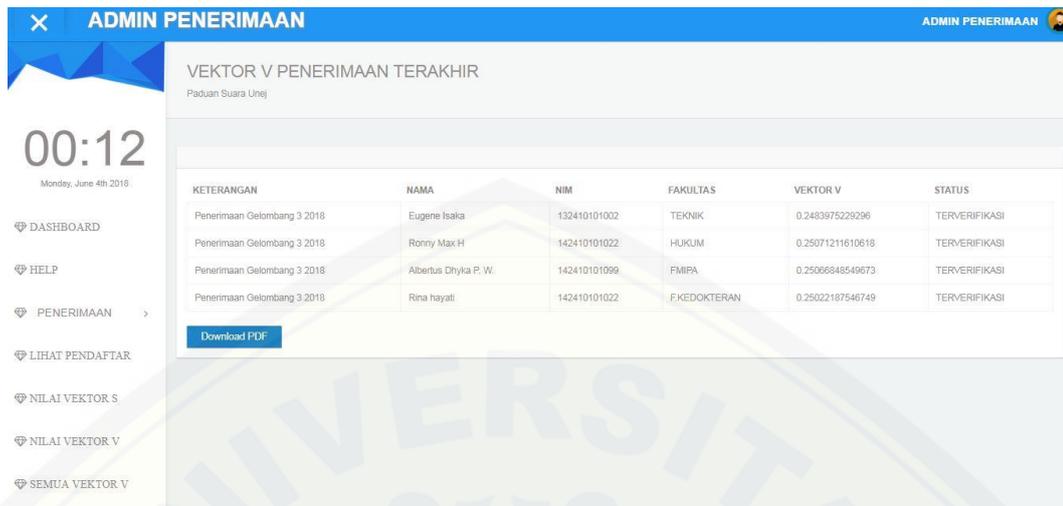
The screenshot shows the 'ADMIN PELATIH' interface. The main content area is titled 'EDIT BOBOT KRITERIA' and displays a form for editing criteria for a successful applicant. The form has the following fields:

- NAMA: Ronny Max H
- NIM: 142410101022
- FAKULTAS: HUKUM
- STATUS: BELUM TERVERIFIKASI

A green 'Submit' button is located below the form.

Gambar 14. Tampilan Persetujuan Pendaftar Lolos

## 15. Tampilan Cetak Pendaftar Lolos Seleksi



**ADMIN PENERIMAAN** ADMIN PENERIMAAN

VEKTOR V PENERIMAAN TERAKHIR  
Paduan Suara Unej

00:12  
Monday, June 4th 2018

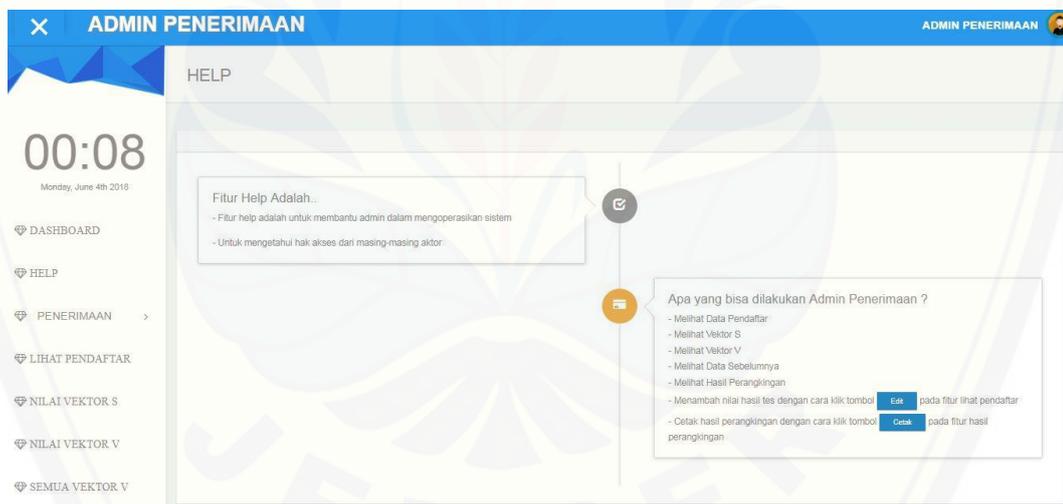
DASHBOARD  
HELP  
PENERIMAAN >  
LIHAT PENDAFTAR  
NILAI VEKTOR S  
NILAI VEKTOR V  
SEMUA VEKTOR V

KETERANGAN	NAMA	NIM	FAKULTAS	VEKTOR V	STATUS
Penerimaan Gelombang 3 2018	Eugene Isaka	132410101002	TEKNIK	0.2483975229296	TERVERIFIKASI
Penerimaan Gelombang 3 2018	Ronny Max H	142410101022	HUKUM	0.25071211610618	TERVERIFIKASI
Penerimaan Gelombang 3 2018	Albertus Dhyka P. W.	142410101099	FMIPA	0.25066848549673	TERVERIFIKASI
Penerimaan Gelombang 3 2018	Rina hayati	142410101022	F.KEDOKTERAN	0.25022187546749	TERVERIFIKASI

Download PDF

Gambar 15. Tampilan Mencetak Pendaftar Lolos Seleksi

## 16. Tampilan Help



**ADMIN PENERIMAAN** ADMIN PENERIMAAN

HELP

00:08  
Monday, June 4th 2018

DASHBOARD  
HELP  
PENERIMAAN >  
LIHAT PENDAFTAR  
NILAI VEKTOR S  
NILAI VEKTOR V  
SEMUA VEKTOR V

Fitur Help Adalah...

- Fitur help adalah untuk membantu admin dalam mengoperasikan sistem
- Untuk mengetahui hak akses dari masing-masing aktor

Apa yang bisa dilakukan Admin Penerimaan ?

- Melihat Data Pendaftar
- Melihat Vektor S
- Melihat Vektor V
- Melihat Data Sebelumnya
- Melihat Hasil Perangkingan
- Menambah nilai hasil tes dengan cara klik tombol **Edit** pada fitur lihat pendaftar
- Cetak hasil perangkingan dengan cara klik tombol **Cetak** pada fitur hasil perangkingan

Gambar 16. Tampilan Help





















