



**SISTEM PENDUKUNG KEPUTUSAN SELEKSI PEMAIN BASKET
MENGUNAKAN *SIMPLE ADDITIVE WEIGHTING* (SAW)**

Studi Kasus : Perbasi Jember

SKRIPSI

Oleh

Barana Rizky Setyatama

NIM 122410101108

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2017



**SISTEM PENDUKUNG KEPUTUSAN SELEKSI PEMAIN BASKET
MENGUNAKAN *SIMPLE ADDITIVE WEIGHTING* (SAW)**

Studi Kasus : Perbasi Jember

SKRIPSI

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

Oleh

Barana Rizky Setyatama

NIM 122410101108

PROGRAM STUDI SISTEM INFORMASI

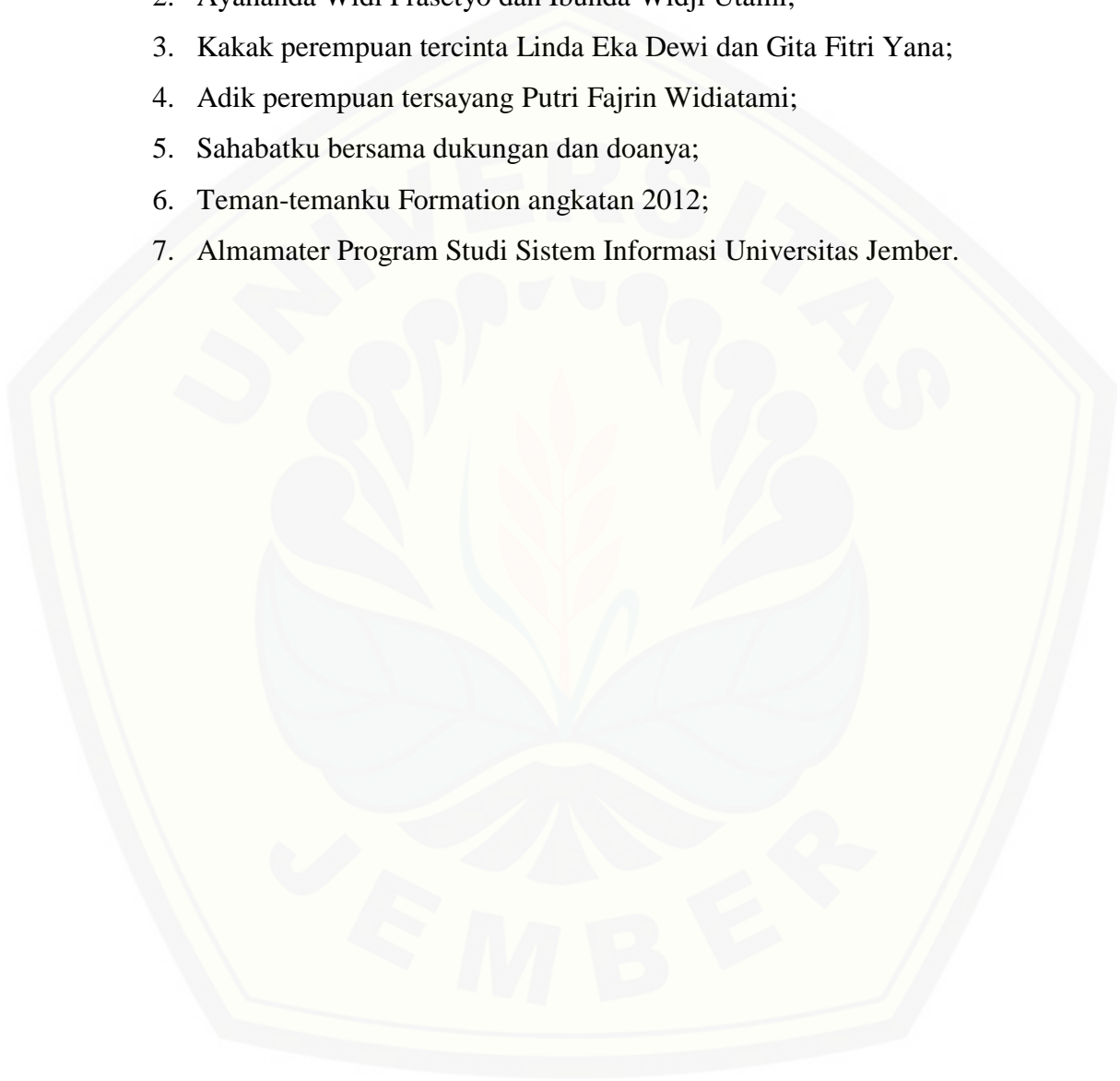
UNIVERSITAS JEMBER

2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Allah SWT;
2. Ayahanda Widi Prasetyo dan Ibunda Widji Utami;
3. Kakak perempuan tercinta Linda Eka Dewi dan Gita Fitri Yana;
4. Adik perempuan tersayang Putri Fajrin Widiatami;
5. Sahabatku bersama dukungan dan doanya;
6. Teman-temanku Formation angkatan 2012;
7. Almamater Program Studi Sistem Informasi Universitas Jember.



MOTO

“Selama masih bisa berkarya, berkarya lah tanpa batas”

(Barana Rizky Setyatama)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Barana Rizky Setyatama

NIM : 122410101108

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan *Simple Additive Weighting* (SAW) Studi Kasus : Perbasi 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 ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 4 Juni 2018

Yang menyatakan,

Barana Rizky Setyatama

NIM 122410101108

SKRIPSI

**SISTEM PENDUKUNG KEPUTUSAN SELEKSI PEMAIN BASKET
MENGUNAKAN *SIMPLE ADDITIVE WEIGHTING* (SAW)**

Studi Kasus : Perbasi Jember

Oleh

Barana Rizky Setyatama

NIM 122410101108

Pembimbing :

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

Dosen Pembimbing Pendamping : Nelly Oktavia Adiwijaya S.Si., MT.

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan Simple Additive Weighting (SAW) Studi Kasus : Perbasi Jember” telah diuji dan disahkan pada:

hari, tanggal : Rabu, 5 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

Nelly Oktavia Adiwijaya S.Si., MT.

NIP. 198410242009122008

PENGESAHAN PENGUJI

“Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan Simple Additive Weighting (SAW) Studi Kasus : Perbasi Jember” telah diuji dan disahkan pada :

hari, tanggal : Jumat, 29 Juni 2018

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji,

Penguji I,

Penguji II,

Fahrobby Adnan, S.Kom., M.MSI
NIP. 198706192014041001

Fajrin Nurman Arifin, ST.,M.Eng
NIP. 198511282015041002

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamini, M.Comp.Sc., Ph.D
NIP. 196704201992011001

RINGKASAN

Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan Simple Additive Weidhting (SAW) Studi Kasus : Perbasi Jember; Barana Rizky Setyatama, 122410101108; 2017; 133 Halaman; Program Studi Sistem Informasi Universitas Jember.

Dalam pembentukan tim bola basket Kabupaten Jember, Perbasi Jember sebagai induk organisasi bola basket yang ada di Jember, melakukan seleksi untuk menentukan pemain yang cocok sesuai dengan kriteria pelatih. Dalam pelaksanaannya, semua data pemain yang akan diseleksi dicatat secara manual di buku evaluasi yang di pegang pelatih. Cara ini akan membutuhkan waktu lama dalam memilih pemain. Hal ini dirasa kurang efisien mengingat jumlah pemain yang diseleksi akan semakin banyak melihat perkembangan basket di Jember yang begitu pesat. Dengan waktu yang semakin lama, maka berbanding lurus dengan anggaran yang dikeluarkan. Maka diperlukan sistem informasi untuk membantu proses seleksi agar lebih efisien menghemat waktu dan anggaran yang dikeluarkan.

Seiring perkembangan teknologi dan ilmu pengetahuan, masyarakat membutuhkan informasi yang cepat dalam melakukan proses pengolahan data dengan memanfaatkan teknologi, contohnya melalui *Personal Computer*. Melihat permasalahan diatas Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan *Simple Additive Weighting* (SAW) diperlukan untuk membantu proses pemilihan pemain yang layak mewakili Kabupaten Jember

Penelitian ini menggunakan jenis penelitian pengembangan. Model pengembangan perangkat lunak untuk perancangan dan pembangunan dari sistem ini menggunakan model *waterfall*. Terdapat 5 tahapan penelitian yaitu : analisis kebutuhan, desain sistem, implementasi sistem, pengujian sistem, dan pemeliharaan sistem. Analisis kebutuhan merupakan tahap untuk mengumpulkan data dan informasi yang dibutuhkan dalam membangun sebuah perangkat lunak. Analisis kebutuhan terdiri dari studi pustaka, wawancara, dan gambaran umum sistem. Desain sistem menggunakan model *Unified Modeling Language* (UML)

yang dirancang menggunakan konsep *Object Oriented Programming* (OOP). Implementasi sistem menggunakan Bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP) dan manajemen basis data MySQL sistem dilakukan dengan menggunakan 2 cara yaitu *White Box* dan *Black Box Testing*. Pemeliharaan sistem ini dilakukan dengan melakukan perbaikan pada *error* yang terdapat dalam perangkat lunak.

Perangkat lunak yang dibangun berbasis aplikasi *web* dengan dua aktor, terdiri dari *user* dan *admin*. Aplikasi dapat memberikan rekomendasi hasil seleksi dengan menerapkan metode *Simple Additive Weighting*. Dari hasil penelitian yang dilakukan, aplikasi dapat menjadi sebuah solusi untuk pelatih dalam memilih dan menyeleksi pemain bola basket sesuai dengan posisinya.

PRAKATA

Puji syukur kepada Allah SWT, atas segala limpahan rahmat, hidayat dan karuniaNya maka penulis dapat menyelesaikan skripsi yang berjudul Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan Simple Additive Weidhting (SAW) Studi Kasus : Perbasi 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 dan bantuan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Drs. Antonius Cahya P, M.App., Sc., Ph.D., selaku Dosen Pembimbing Utama dan Nelly Oktavia A, S.Si, MT., selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran dan perhatian dalam penulisan skripsi;
3. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
4. Ayahanda Widi Prasetyo dan Widji Utami;
5. Kakak perempuan tercinta Linda Eka Dewi dan Gita Fitri Yana;
6. Adik perempuan Putri Fajrin Widiatami;
7. Maya Dinana yang selalu memberikan dukungan, motivasi dan semangat;
8. Sahabat seperjuangan yang selalu menemani dan memberikan semangat serta doa Kresna, Marceli, dan Anton;
9. Teman-teman seperjuangan FORMATION angkatan 2012 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
10. Keluarga besar Unit Kegiatan Mahasiswa Bola Basket UNEJ dan anggota lain yang tidak dapat disebutkan satu persatu;
11. Hisyam, Angga (Narasumber Perbasi Jember)
12. Semua pihak yang tidak dapat disebutkan satu persatu;

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

Jember, 4 Juni 2018

Penulis



DAFTAR ISI

SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTO.....	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN PENGUJI.....	vii
RINGKASAN.....	viii
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB. 1 PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Tujuan dan Manfaat.....	3
1.3.1. Tujuan.....	3
1.3.2. Manfaat.....	4
1.4. Batasan Masalah.....	4
1.5. Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA.....	6
2.1. Penelian Terdahulu.....	6
2.2. Sistem Pendukung Keputusan (SPK).....	7
2.3. Metode Simple Additive Weighting.....	8
2.4. Persatuan Bola Basket Seluruh Indonesia (Perbasi).....	10
BAB 3. METODOLOGI PENELITIAN.....	11
3.1 Jenis Penelitian.....	11
3.2 Tempat dan Waktu.....	11
3.3 Pengembangan Sistem.....	11
3.3.1 Tahapan Analisis Kebutuhan.....	12

3.3.2	Tahapan Desain Sistem	12
3.3.3	Tahapan Implementasi Sistem.....	13
3.3.4	Tahapan Pengujian Sistem	13
3.3.5	Tahapan Pemeliharaan Sistem.....	13
3.3.6	Analisis Penggunaan Metode	14
3.3.7	Ketepatan Implementasi	14
3.3.8	Keakuratan Keputusan.....	14
3.4	Gambaran Umum Sistem	14
BAB 4. PENGEMBANGAN SISTEM.....		18
4.1	Deskripsi Umum Sistem.....	18
4.1.1	SOP (<i>Statement of Purpose</i>).....	18
4.2	Pengumpulan Data.....	18
4.3	Analisis Kebutuhan	19
4.3.1	Kebutuhan Fungsional.....	19
4.3.2	Kebutuhan Non-Fungsional.....	19
4.4	Desain Sistem	20
4.4.1	<i>Business Process</i>	20
4.4.2	<i>Use Case Diagram</i>	21
4.4.3	<i>Use Case Skenario</i>	23
4.4.4	<i>Activity Diagram</i>	27
4.4.5	<i>Sequence Diagram</i>	30
4.4.6	<i>Class Diagram</i>	33
4.4.7	<i>Entity Relationship Diagram (ERD)</i>	34
4.5	Implementasi	35
4.6	Pengujian	37
4.6.1	Metode <i>White Box</i>	38
1.	<i>Listing Program</i>	38
2.	Diagram Alir	39
3.	Pengujian Basis Set (<i>Test Set</i>).....	40
4.6.2	Metode <i>Black Box</i>	41
BAB 5. HASIL DAN PEMBAHASAN.....		42

5.1	Hasil Pembuatan Sistem Pendukung Keputusan Seleksi Pemain Basket menggunakan <i>Simple Additive Weighting</i> (SAW)	42
5.1.1	Tampilan Awal Sistem	42
5.1.2	Fitur Menampilkan <i>Dashboard</i> (Admin)	43
5.1.3	Fitur Menampilkan <i>Dashboard</i> (User).....	43
5.1.4	Fitur Mengelola Biodata Pemain.....	44
5.1.5	Fitur Mengelola Data Kejuaraan	44
5.1.6	Fitur Mengelola Nilai Seleksi.....	45
5.1.7	Fitur Mengelola Data Bobot.....	45
5.1.8	Fitur Menampilkan Perhitungan SAW	46
5.1.9	Fitur Kirim Undangan Seleksi.....	48
5.1.10	Fitur Balas Undangan Seleksi	49
5.1.11	Fitur Menampilkan Ranking.....	49
5.2	Hasil Penerapan Perhitungan Metode <i>Simple Additive Weighting</i> (SAW)	50
5.2.1	Penentuan Data Kriteria dan Posisi	50
5.2.2	Penentuan Nilai Bobot.....	51
5.2.3	Penentuan Skor Seleksi	52
5.2.4	Mencari Nilai Maksimal.....	52
5.2.5	Normalisasi.....	53
5.2.6	Perhitungan Nilai Preferensi.....	54
5.2.7	<i>Ranking</i>	55
5.2.8	Seleksi.....	55
5.3	Pengujian Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan <i>Simple Additive Weighting</i> (SAW) di Perbasi Jember	56
5.4	Pembahasan Pada Sistem Pendukung Keputusan Seleksi Pemain Basket di Perbasi Jember	57
5.6.1	Kelebihan Sistem.....	58
5.6.2	Kelemahan Sistem	58
BAB 6. PENUTUP.....		59
6.1	Kesimpulan.....	59
6.2	Saran.....	59
DAFTAR PUSTAKA		61

LAMPIRAN	63
LAMPIRAN A. <i>USE CASE</i> SKENARIO.....	63
A.1 <i>Use Case</i> Skenario Login.....	63
A.2 <i>Use Case</i> Skenario <i>View Dashboard Admin</i>	64
A.3 <i>Use Case</i> Skenario <i>View Dashboard User</i>	65
A.4 <i>Use Case</i> Skenario Mengelola Biodata Pemain.....	66
A.5 <i>Use Case</i> Skenario Mengelola Data Kejuaraan	67
A.6 <i>Use Case</i> Skenario Mengelola Nilai Seleksi.....	68
A.7 <i>Use Case</i> Skenario Mengelola Nilai Bobot	69
A.8 <i>Use Case</i> Skenario <i>View Perhitungan SAW</i>	70
A.9 <i>Use Case</i> Skenario Kirim Undangan Seleksi.....	71
A.10 <i>Use Case</i> Skenario Balas Undangan Seleksi.....	72
A.11 <i>Use Case</i> Skenario <i>View Ranking</i>	73
A.12 <i>Use Case</i> Skenario <i>Logout</i>	73
LAMPIRAN B. <i>ACTIVITY</i> DIAGRAM	74
B.1 <i>Activity</i> Diagram Login	74
B.2 <i>Activity</i> Diagram <i>View Dashboard Admin</i>	75
B.3 <i>Activity</i> Diagram <i>View Dashboard User</i>	76
B.4 <i>Activity</i> Diagram Mengelola Biodata Pemain.....	77
B.5 <i>Activity</i> Diagram Mengelola Data Kejuaraan	78
B.6 <i>Activity</i> Skenario Mengelola Nilai Seleksi.....	79
B.7 <i>Activity</i> Skenario Mengelola Nilai Bobot	80
B.8 <i>Activity</i> Skenario <i>View Perhitungan SAW</i>	81
B.9 <i>Activity</i> Skenario Kirim Undangan Seleksi.....	82
B.10 <i>Activity</i> Skenario Balas Undangan Seleksi	83
B.11 <i>Activity</i> Skenario <i>View Ranking</i>	84
B.12 <i>Activity</i> Skenario <i>Logout</i>	84
LAMPIRAN C. <i>SEQUENCE</i> DIAGRAM	85
C.1 <i>Sequence</i> Diagram <i>Login</i>	85
C.2 <i>Sequence</i> Diagram <i>View Dashboard Admin</i>	86
C.3 <i>Sequence</i> Diagram <i>View Dashboard User</i>	87
C.4 <i>Sequence</i> Diagram Mengelola Biodata Pemain	88

C.5 <i>Sequence</i> Diagram Mengelola Data Kejuaraan	89
C.6 <i>Sequence</i> Skenario Mengelola Nilai Seleksi.....	90
C.7 <i>Sequence</i> Skenario Mengelola Nilai Bobot	91
C.8 <i>Sequence</i> Skenario View Perhitungan SAW.....	92
C.9 <i>Sequence</i> Skenario Kirim Undangan Seleksi.....	93
C.10 <i>Sequence</i> Skenario Balas Undangan Seleksi	94
C.11 <i>Sequence</i> Skenario View Ranking.....	95
C.12 <i>Sequence</i> Skenario Logout.....	96
LAMPIRAN D. KODE PROGRAM.....	97
D.1 Kode Program <i>class controllers/Admin</i>	97
D.2 Kode Program <i>class controllers/Kirim</i>	106
D.3 Kode Program <i>class controllers/Login</i>	107
D.4 Kode Program <i>class controllers/User</i>	112
D.5 Kode Program <i>class models/m_detailKejuaraan</i>	116
D.6 Kode Program <i>class models/m_kejuaraan</i>	119
D.7 Kode Program <i>class models/m_kriteria</i>	122
D.8 Kode Program <i>class models/m_pesanan</i>	124
D.9 Kode Program <i>class models/m_user</i>	125
LAMPIRAN E. PENGUJIAN BLACK BOX	129
LAMPIRAN D. TRANSKRIP WAWANCARA	134

DAFTAR GAMBAR

Gambar 2. 1 Diagram Alur Metode SAW	10
Gambar 3. 1 Tahapan Metode Waterfall	11
Gambar 3. 2 Diagram Alir Penelitian	15
Gambar 3. 3 Gambaran implementasi SAW	16
Gambar 4. 1 <i>Business Process Sistem</i>	20
Gambar 4. 2 <i>Use Case Diagram</i>	21
Gambar 4. 3 <i>Activity Diagram Login</i>	29
Gambar 4. 4 Sequence Diagram Mengelola Nilai Seleksi.....	32
Gambar 4. 5 <i>Class Diagram</i>	34
Gambar 4. 6 <i>Entity Relationship Diagram</i>	35
Gambar 4. 7 <i>Listing Program Perhitungan()</i>	39
Gambar 4. 8 Diagram Alir <i>Class Controller Admin Function Perhitungan()</i>	39
Gambar 5. 1 Tampilan Halaman <i>Login</i>	42
Gambar 5. 2 Tampilan Halaman <i>Dashboard</i>	43
Gambar 5. 3 Tampilan Halaman <i>Dashboard</i>	43
Gambar 5. 4 Tampilan Halaman Biodata Pemain.....	44
Gambar 5. 5 Tampilan Halaman Mengelola Data Kejuaraan	44
Gambar 5. 6 Tampilan Halaman Mengelola Nilai Seleksi	45
Gambar 5. 7 Tampilan Halaman Mengelola Data Bobot.....	45
Gambar 5. 8 Tampilan Halaman Perhitungan SAW Tahap 1	46
Gambar 5. 9 Tampilan Halaman Perhitungan SAW Tahap 2.....	46
Gambar 5. 10 Tampilan Halaman Perhitungan SAW Tahap 3	47
Gambar 5. 11 Tampilan Halaman Perhitungan SAW Tahap 4.....	47
Gambar 5. 12 Tampilan Halaman Perhitungan SAW Tahap 5	47
Gambar 5. 13 Tampilan Halaman Perhitungan SAW Tahap 6	48
Gambar 5. 14 Tampilan Halaman Fitur Kirim Undangan Seleksi.....	48
Gambar 5. 15 Tampilan halaman Fitur Balas Undangan Seleksi	49
Gambar 5. 16 Tampilan Halaman Ranking (1).....	49
Gambar 5. 17 Tampilan Halaman Ranking (2).....	50

Gambar B. 1 <i>Activity Diagram Login</i>	74
Gambar B. 2 <i>Activity Diagram View Dashboard Admin</i>	75
Gambar B. 3 <i>Activity Diagram View Dashboard User</i>	76
Gambar B. 4 <i>Activity Diagram Mengelola Biodata Pemain</i>	77
Gambar B. 5 <i>Activity Diagram Data Kejuaraan</i>	78
Gambar B. 6 <i>Activity Diagram Mengelola Nilai Seleksi</i>	79
Gambar B. 7 <i>Activity Diagram Mengelola Nilai Bobot</i>	80
Gambar B. 8 <i>Activity Diagram View Perhitungan SAW</i>	81
Gambar B. 9 <i>Activity Diagram Kirim Undangan Seleksi</i>	82
Gambar B. 10 <i>Activity Diagram Balas Undangan Seleksi</i>	83
Gambar B. 11 <i>Activity Diagram View Ranking</i>	84
Gambar B. 12 <i>Activity Diagram Logout</i>	84
Gambar C. 1 <i>Sequence Diagram Login</i>	85
Gambar C. 2 <i>Sequence Diagram View Dashboard Admin</i>	86
Gambar C. 3 <i>Sequence Diagram View Dashboard User</i>	87
Gambar C. 4 <i>Sequence Diagram Mengelola Biodata Pemain</i>	88
Gambar C. 5 <i>Sequence Diagram Mengelola Data Kejuaraan</i>	89
Gambar C. 6 <i>Sequence Diagram Mengelola Nilai Seleksi</i>	90
Gambar C. 7 <i>Sequence Diagram Mengelola Nilai Bobot</i>	91
Gambar C. 8 <i>Sequence Diagram View Perhitungan SAW</i>	92
Gambar C. 9 <i>Sequence Diagram Kirim Undangan Seleksi</i>	93
Gambar C. 10 <i>Sequence Diagram Balas Undangan Seleksi</i>	94
Gambar C. 11 <i>Sequence Diagram View Ranking</i>	95
Gambar C. 12 <i>Sequence Diagram Logout</i>	96

DAFTAR TABEL

Tabel 3. 1 Bobot Setiap Posisi	17
Tabel 4. 1 Deskripsi Pembagian Aktor Sistem	22
Tabel 4. 2 Deskripsi <i>Use Case</i>	22
Tabel 4. 3 Skenario Mengelola Nilai Seleksi.....	25
Tabel 4. 4 Kode Metode <i>SAW function</i> perhitungan()	35
Tabel 4. 5 <i>Test Case Function</i> Perhitungan().....	40
Tabel 5. 1 Data Kriteria.....	51
Tabel 5. 2 Posisi.....	51
Tabel 5. 3 Penentuan Nilai Bobot	52
Tabel 5. 4 Skor Seleksi.....	52
Tabel 5. 5 Nilai Maksimal.....	53
Tabel 5. 6 Normalisasi	53
Tabel 5. 7 Normalisasi Pemain A	54
Tabel 5. 8 Nilai Bobot Posisi <i>Point Guard</i>	54
Tabel 5. 9 Hasil Hitung Nilai Preferensi.....	55
Tabel 5. 10 <i>Ranking</i> Sesuai Nilai Preferensi.....	55
Tabel 5. 11 Hasil Seleksi.....	55
Tabel 5. 12 <i>Decision Maker</i>	57
Tabel 5. 13 Perbandingan Hasil Pengujian Akurasi Sistem.....	57
Tabel A. 1 <i>Use Case Login</i>	63
Tabel A. 2 <i>Use Case</i> Skenario <i>View Dashboard Admin</i>	64
Tabel A. 3 <i>Use Case</i> Skenario <i>View Dashboard User</i>	65
Tabel A. 4 <i>Use Case</i> Skenario Mengelola Biodata Pemain.....	66
Tabel A. 5 <i>Use Case</i> Skenario Mengelola Data Kejuaraan	67
Tabel A. 6 <i>Use Case</i> Skenario Mengelola Nilai Seleksi.....	68
Tabel A. 7 <i>Use Case</i> Skenario Mengelola Nilai Bobot	69
Tabel A. 8 <i>Use Case</i> Skenario Mengelola Nilai Bobot	70
Tabel A. 9 <i>Use Case</i> Skenario Undangan Seleksi	71
Tabel A. 10 <i>Use Case</i> Skenario Balas Undangan Seleksi	72
Tabel A. 11 <i>Use Case</i> Skenario <i>View Ranking</i>	73

Tabel A. 12 *Use Case Skenario Logout* 73



BAB. 1 PENDAHULUAN

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

1.1. Latar Belakang

Permainan bola basket diciptakan oleh Prof A.Naismith salah seorang guru pendidikan jasmani *Young Mens Christian Association (YMCA)* Springfield, Amerika Serikat pada tahun 1891, permainan bola basket dimainkan oleh dua regu berlawanan, tiap-tiap regu yang melakukan permainan di lapangan terdiri dari 5 orang, sedangkan pemain pengganti sebanyak-banyaknya 7 orang, sehingga tiap regu paling banyak terdiri dari 12 orang pemain (Rastafari Horongbala, 2004). Permainan bola basket saat ini mengalami perkembangan yang pesat, mulai dari penyempurnaan peraturan dasar dalam bermain, kompetisi yang semakin banyak, dan munculnya tim serta atlet bola basket pelajar baik di tingkat sekolah maupun perguruan tinggi.

Perkembangan Basket di Jember menunjukkan progres yang pesat. Kabupaten Jember merupakan salah satu kabupaten di Jawa Timur yang mempunyai banyak prestasi di cabang olahraga (Cabor) bola basket. Prestasi yang paling membanggakan adalah salah satu atlet bola basket Jember Sandy Febiansyakh, terpilih menjadi pemain inti di Timnas Indonesia pada kejuaraan *Sea Games 2015* Singapura, tak sampai disitu, Timnas Indonesia juga meraih medali perak dalam kejuaraan tersebut. Prestasi selanjutnya adalah tim bola basket Jember meraih medali perak dalam kejuaraan Pekan Olahraga Provinsi (PORPROV) 2013 Banyuwangi (Wawancara Perbasi Jember 2017). Pada tahun 2016 hampir setiap bulan terdapat satu agenda kompetisi yang berjalan di Jember. Dibandingkan tahun 2012 yang setiap tiga bulan hanya ada satu kompetisi. Khusus untuk kompetisi tingkat Sekolah Dasar (SD) pada Tahun 2016 terdapat empat kompetisi yang berlangsung, di bandingkan tahun 2012 yang tidak ada kompetisi (Wawancara Perbasi Jember 2017).

Persatuan Bola Basket Seluruh Indonesia (Perbasi Jember) merupakan induk organisasi bola basket di Jember. Berlokasi di Jl. Bengawan Solo No. 14 A, Sumpersari, Jember. Perbasi Jember mempunyai program-program kerja untuk meningkatkan kualitas perbasketan di Jember. Salah satu program kerja Perbasi Jember adalah pembinaan atlet bola basket mulai dari usia dini (Wawancara Perbasi Jember 2017). Bentuk kegiatan yang dilakukan adalah mengadakan turnamen antar pelajar dan pemusatan latihan/*training camp* (TC) untuk mendapatkan atlet-atlet terbaik Jember yang akan dikirim untuk mengikuti kejuaraan di dalam kota maupun luar kota mewakili Kabupaten Jember.

Program seleksi Perbasi Jember dipimpin dan dinilai berdasarkan pengamatan dua pelatih. Atlet-atlet terbaik Jember akan dipilih melalui proses seleksi dan dikirim ke kejuaraan seperti Pekan Olahraga Daerah (POPDA) yang diadakan 2 tahun sekali dan Pekan Olahraga Provinsi (PORPROV) yang diadakan 4 tahun sekali. Kejuaraan ini merupakan kejuaraan penting, karena dari sinilah tolak ukur keberhasilan Perbasi Jember dalam membina atlet bola basket. Mekanisme seleksi yang dilakukan yaitu menilai kemampuan dasar-dasar dalam permainan bola basket seperti: *dribble*, *passing*, *shooting*, fisik, dll (Wawancara Perbasi Jember 2017). Nilai dari setiap pemain akan dicatat, sebagai bahan acuan memilih pemain terbaik dari semua peserta seleksi. Proses seleksi akan dibedakan sesuai dengan kemampuan masing-masing pemain dalam menguasai posisi dalam bermain basket. 5 posisi dalam bermain basket yaitu: posisi 1/*Point Guard* (PG), posisi 2/*Shooting Guard* (SG), posisi 3/*Small Forward* (SF), posisi 4/*Power Forward* (PF), posisi 5/*Center* (C) (Rastafari Horongbala, 2004). Setiap posisi mempunyai prioritas penilaian yang akan di nilai sesuai dengan materi yang diberikan. Setiap posisi akan dipilih 3 pemain terbaik. Total luaran yang di hasilkan dalam program seleksi ini adalah 15 orang pemain terbaik mewakili 1 tim. Proses seleksi mengandalkan sistem manual (Wawancara Perbasi Jember 2017). Semua data pemain dicatat di buku evaluasi yang di pegang pelatih. Cara ini akan membutuhkan waktu lama dalam memilih pemain serta ketidaksihlerasan/error antara pemilihan pelatih dan nilai seleksinya karena proses penilaian seleksi tidak menggunakan metode penilaian. Hal ini dirasa kurang efisien mengingat jumlah

pemain yang diseleksi akan semakin banyak melihat perkembangan basket di Jember yang begitu pesat. Dengan waktu yang semakin lama, maka berbanding lurus dengan anggaran yang dikeluarkan. Maka diperlukan sistem informasi untuk membantu proses seleksi agar lebih efisien menghemat waktu dan anggaran yang dikeluarkan.

Seiring perkembangan teknologi dan ilmu pengetahuan, masyarakat membutuhkan informasi yang cepat dalam melakukan proses pengolahan data dengan memanfaatkan teknologi, contohnya melalui *Personal Computer(PC)*. Melihat permasalahan diatas Sistem Pendukung Keputusan Seleksi Pemain Basket Menggunakan *Simple Additive Weighting (SAW)* diperlukan untuk membantu proses pemilihan pemain yang layak mewakili Kabupaten Jember. Metode *SAW* digunakan, karena metode ini bekerja dengan mencari alternatif sesuai dengan penjumlahan terbobot dari setiap rating kriteria pada semua atribut. Melihat posisi dalam permainan bola basket mempunyai bobot prioritas berbeda dengan posisi lainnya (Kusumadewi, 2016).

1.2. Rumusan Masalah

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

- a. Bagaimana penerapan metode *SAW* dalam pengambilan keputusan seleksi pemain basket di Perbasi Jember?
- b. Bagaimana membangun sistem penunjang keputusan pemilihan pemain basket di Perbasi Jember dengan metode *SAW*?

1.3. Tujuan dan Manfaat

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

1.3.1. Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah :

- a. Menerapkan metode *SAW* dalam pengambilan keputusan seleksi pemain basket di Perbasi Jember.

- b. Membangun sistem penunjang keputusan yang dapat membantu Perbasi Jember dalam pengambilan keputusan untuk menentukan pemain basket.

1.3.2. Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

- a. Bagi Instansi

Sebagai penunjang pengambilan keputusan bagi instansi Perbasi untuk memilih pemain basket di Kabupaten Jember.

- b. Bagi Akademis

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

- c. Bagi Peneliti

Mengetahui bagaimana proses penerapan *SAW* pada sistem rekomendasi pemilihan pemain basket di Kabupaten Jember.

- d. Bagi pihak lain

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

1.4. Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat sebagai parameter dalam melakukan penelitian ini, antara lain sebagai berikut :

- a. Lingkup penelitian adalah Perbasi Jember.
- b. Kriteria yang digunakan terbatas sesuai dengan kriteria pelatih yang ada di Jember.

1.5. Sistematika Penulisan

Adapun sistematika penulisan skripsi adalah sebagai berikut:

- a. Bab 1. Pendahuluan

Bab ini terdiri atas pendahuluan atas penelitian yang akan dilakukan.

b. Bab 2. Tinjauan Pustaka

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

c. Bab 3. Metodologi Penelitian

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

d. Bab 4. Pengembangan Sistem

Bab ini berisi uraian tentang langkah-langkah yang ditempuh dalam proses menganalisis dan merancang sistem yang hendak dibangun meliputi desain, kode program, dan pengujian sistem.

e. Bab 5. Hasil dan Pembahasan

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

f. Bab 6. Penutup

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

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.2.1 Penelitian Terdahulu

2.1. Penelian Terdahulu

Metode *Simple Additive Weighting* (SAW) telah diterapkan oleh Sigit Prasetyo Karisma Utomo dan Praditya Kurniawan, 2016. Dalam penelitian yang berjudul “Sistem Penunjang Keputusan Dalam Pemilihan Posisi Tertentu Pada Sepak Bola” yang membahas mengenai permasalahan pemilihan pemain berdasarkan posisi. Karena setiap pemain mempunyai kelebihan di posisi-posisi tertentu. Obyektifitas pemilihan pemain sesuai dengan statistik dan karakteristik pemain. Dalam penelitian ini, penulis menggunakan metode *Simple Additive Weighting* (SAW) untuk pencarian pemain pada sebuah posisi tim utama sepak bola menggunakan beberapa langkah yakni: (a) Menentukan posisi yang dicari (b) Menentukan kriteria-kriteria dalam permainan sepak bola yang mempengaruhi data (c) Menentukan kriteria pada posisi yang akan dicari (d) Mengelola musim pertandingan (e) Mengelola data pemain yang mengikuti seleksi posisi (g) melakukan proses SAW yang di pilih posisi dan data statistik berdasarkan musim yang dicari.

Dalam menentukan posisi pemain sepak bola diperlukan data statistik dari setiap pemain dalam periode/musm tertentu. Data tersebut didapat saat pemain bermain dalam beberapa pertandingan periode waktu tertentu. Contoh beberapa ktiteria yang digunakan dalam menentukan posisi pemain adalah: *goal, total shot, chances created, assist, total passes, total cross, total clearance, yellow card, red card, dan dribbling*. Dengan data tersebut siste harus mampu memberikan rekomendasi pemain untuk posisi tertentu. Dasar Metode SAW adalah mencari jumlah pembobotan dari rating kinerja pada setiap alternatif di semua atribut. Penelitian ini memiliki pengujian menggunakan 10 kriteria, setiap kriteria mempunyai bobot tertentu sesuai keinginan pelatih, alternatif yang digunakan

adalah pemain yang bermain di setiap musim. Dan berhasil dalam memilih pemain sesuai dengan kebutuhan pelatih. Pembobotan dalam setiap posisi yang digunakan di penelitian ini akan sangat cocok bila diterapkan di proses seleksi pemain basket, dikarenakan permainan bola basket memiliki 5 posisi yang masing-masing posisi memiliki tugas bermain yang berbeda-beda. Berdasarkan hasil penelitian Prasetyo Karisma Utomo, dkk dapat disimpulkan bahwa metode SAW dapat menentukan pemain terbaik berdasarkan posisi posisinya sehingga rekomendasi pemain lebih objektif.

Sedangkan sistem penunjang keputusan seleksi basket sudah diterapkan oleh Tondo Kusumo dan Karis Widyatmoko. Dalam penelitian yang berjudul “Sistem Penunjang Keputusan Seleksi Atlet Basket Pada PPLPD Jateng” dengan menggunakan metode profile matching dalam penyelesaiannya. Penelitian ini dilakukan untuk membantu pelatih menentukan pemain. Dengan jumlah pemain sebanyak 14 pemain. Kemampuan yang dinilai meliputi *vertical jump*, *sprint*, dan *agility*. Berdasarkan hasil penelitian tersebut, dengan adanya Sistem Pendukung Keputusan Seleksi Atlet Basket Pada PPLPD Jateng, para pelatih akan lebih mengontrol para pemainnya. Setiap pemain dapat tercatat kemampuannya dari setiap latihan. Dengan begitu pelatih juga mendapatkan statistik perkembangan pemain dari bulan ke bulan. Penggunaan metode profile matching tidak cocok digunakan ke dalam sistem pendukung seleksi pemain basket. Secara garis besar dalam proses metode profile matching adalah membandingkan antara kompetensi data terdahulu dengan data yang sekarang. PERBASI Jember tidak mempunyai data yang dapat dibandingkan, meskipun ada, penggunaannya dirasa kurang efisien, karena kualitas pemain dulu dengan sekarang berbeda.

2.2. Sistem Pendukung Keputusan (SPK)

SPK merupakan penggabungan dari sumber-sumber kecerdasan individu yang memiliki kemampuan untuk dapat memperbaiki kualitas dari suatu keputusan. Dari pengertian tersebut dapat disimpulkan SPK bukan merupakan alat pengambil keputusan melainkan SPK adalah sistem yang membantu dalam mengambil keputusan atau menunjang keputusan yang didasari oleh informasi

dari data yang diolah dengan relevan yang diperlukan untuk membuat keputusan tentang suatu masalah dengan cepat dan akurat, sehingga sistem tidak dimaksud untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan (Munawan & Shiddiq, 2012). Sistem penunjang keputusan adalah bagian dari sistem informasi berbasis komputer (termasuk berbasis pengetahuan) yang dipakai untuk mendukung pengambil keputusan dari masalah semi-terstruktur yang spesifik dalam suatu organisasi atau perusahaan (Dinal, 2011).

Dari definisi tersebut diatas dapat disimpulkan bahwa Sistem Penunjang Keputusan adalah sistem komputer yang menciptakan informasi yang berguna untuk mendukung para pengambil keputusan untuk mengambil keputusan yang tepat dalam situasi semi-terstruktur maupun tidak terstruktur.

Sistem Penunjang Keputusan memiliki beberapa karakteristik, yaitu :

- a. Sistem pendukung keputusan dirancang untuk mengambil keputusan dalam memecahkan masalah yang sifatnya semi terstruktur ataupun tidak terstruktur.
- b. Dalam proses pengolahannya, sistem pendukung keputusan mengkombinasikan penggunaan model-model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari atau interogasi informasi.
- c. Sistem pendukung keputusan dirancang sedemikian rupa sehingga dapat digunakan dengan mudah oleh orang-orang yang tidak memiliki dasar kemampuan pengoperasian komputer yang tinggi. Oleh karena itu pendekatan yang digunakan biasanya model interaktif.
- d. Sistem pendukung keputusan dirancang dengan menekankan pada aspek fleksibilitas dan adaptasi yang tinggi, sehingga mudah disesuaikan dengan perubahan lingkungan yang terjadi.

2.3. Metode Simple Additive Weighting

Metode ini merupakan metode yang paling dikenal dan banyak digunakan untuk menghadapi situasi *Multiple Attribute Decision Making* (MDAM) (Kusumadewi, 2006). Metode ini mengharuskan pembuat keputusan menentukan

bobot pada setiap atribut. Skor total untuk sebuah alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating yang dapat dibandingkan lintas atribut. Rating tiap atribut telah melewati proses normalisasi sebelumnya.

Metode SAW dikenal dengan istilah penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif di semua atribut (Kusumadewi, 2006).

Metode SAW membutuhkan proses normalisasi matriks keputusan ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada. Formula untuk melakukan normalisasi diberikan pada persamaan I berikut:

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max x_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah attribute biaya (cost)} \end{cases} \dots \text{persamaan (I)}$$

dimana :

r_{ij} = rating kinerja ternormalisasi

Max_{ij} = nilai maksimum dari setiap baris dan kolom

Min_{ij} = nilai minimum dari setiap baris dan kolom

X_i = baris dan kolom dari matriks

Dengan r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$. Nilai preferensi setiap alternatif (V_i) diberikan pada persamaan II.

$$V_i = \sum_{j=1}^n w_j r_{ij} \dots \text{persamaan (II)}$$

dimana :

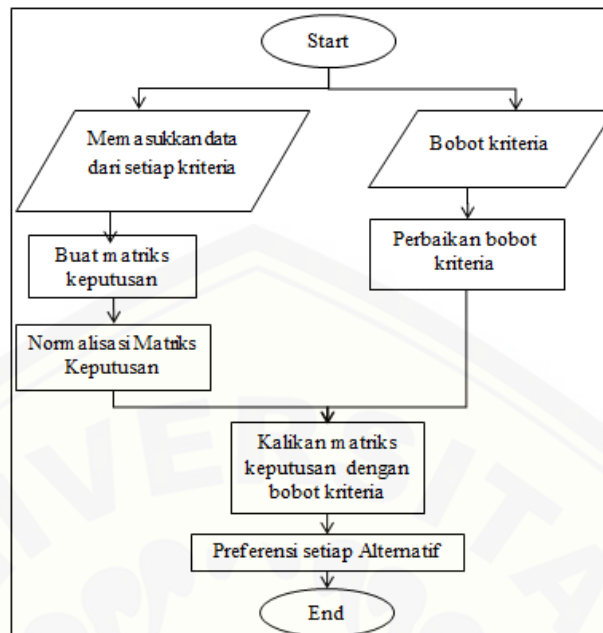
V_i = Nilai akhir dari alternatif

W_j = Bobot yang telah ditentukan

R_{ij} = Normalisasi matriks

Nilai V_i lebih besar megindikasikan bahwa alternatif A_i lebih terpilih (Kusumadewi, 2006).

Algoritma metode SAW dapat dilihat pada gambar :



Gambar 2. 1 Diagram Alur Metode SAW

(Sumber:Kusumadewi, 2006)

2.4. Persatuan Bola Basket Seluruh Indonesia (Perbasi)

Perbasi adalah satu satunya badan yang memiliki wewenang dalam mengkoordinasikan dan membina segala kegiatan bola basket di seluruh wilayah Indonesia. Perbasi didirikan pada tahun 1952 oleh Tony Wen dan Wim latumeten. Perbasi menganut sistem vertical berjenjang, yang dimulai dari tingkat perkumpulan, pengurus cabang (Pencab), pengurus daerah (Pengda), sampai ke pengurus besar (PB) Perbasi (Perbasi, 2017).

Persatuan Bola Basket Seluruh Indonesia (Perbasi Jember) merupakan induk organisasi bola basket di Jember. Berlokasi di Jl. Bengawan Solo No. 14 A, Sumpersari, Jember. Perbasi Jember mempunyai program-program kerja untuk meningkatkan kualitas perbasketan di Jember. Salah satu program kerja Perbasi Jember adalah pembinaan atlet bola basket mulai dari usia dini (Wawancara Perbasi Jember 2017)

BAB 3. METODOLOGI PENELITIAN

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

3.1 Jenis Penelitian

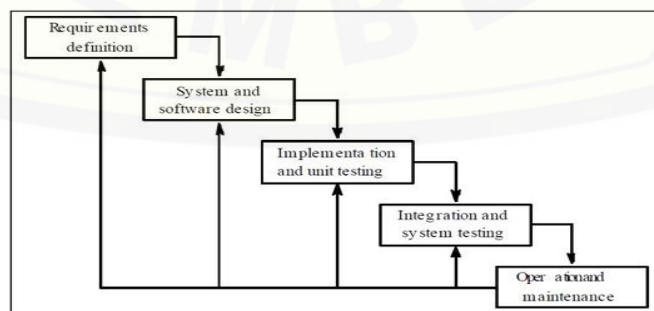
Penelitian ini menggunakan jenis penelitian pengembangan, di mana penelitian ini membuat dan mengembangkan suatu sistem pembantu keputusan sehingga dapat membantu menentukan pemain basket di Jember dengan menggunakan metode SAW dan penelitian ini bukan dimaksudkan untuk menemukan teori baru atau menguji kebenaran dari suatu teori atau metode dalam penelitian.

3.2 Tempat dan Waktu

Penelitian ini dilakukan di Perbasi Jember yang beralamat di Jl. Bengawan Solo No 14A Kabupaten Jember. Waktu penelitian dilakukan selama 2 (dua) bulan, dimulai pada bulan Juli 2016 sampai dengan bulan Agustus 2016.

3.3 Pengembangan Sistem

System Development Life Cycle (SDLC) yang digunakan adalah metode *Waterfall*. Menurut Pressman (1997) dalam Proboyekti (2011) model *waterfall* adalah model klasik yang bersifat berurutan dalam membangun *software*. Alur perancangan model *waterfall* dapat dilihat pada gambar 3.1 :



Gambar 3. 1 Tahapan Metode Waterfall

(Sumber: Proboyekti, 2013)

3.3.1 Tahapan Analisis Kebutuhan

Tahapan ini dilakukan untuk mengumpulkan kebutuhan secara lengkap melalui studi literatur dan survei lapangan kemudian menganalisa kebutuhan yang harus dipenuhi oleh sistem yang akan dibangun. Penelitian dilakukan berdasarkan hasil wawancara dengan Sekretaris Umum dan Bidang Pelatihan Perbasi Jember yang bersektariat di Jl. Bengawan Solo No. 14A, Sumbersari, Jember, Jawa Timur dan jurnal-jurnal yang berkaitan dengan penelitian ini. Waktu penelitian ini selama tiga bulan.

3.3.2 Tahapan Desain Sistem

Pembuatan desain sistem pada penelitian ini menggunakan *Unified Modeling Language (UML)* yang dirancang menggunakan konsep *Object Oriented Programming (OOP)*. Pemrograman berorientasi objek atau dalam bahasa Inggrisnya disebut dengan *Object Oriented Programming* (selanjutnya disingkat *OOP*) adalah teknik pemrograman yang menggunakan objek (Gery Sabar, 2015). Diagram yang akan dibuat, yaitu :

- a. *Business Process* digunakan untuk menggambarkan inputan data yang dibutuhkan sistem dan output dari sistem serta tujuan pembuatan sistem.
- b. *Use Case Diagram* merupakan rangkaian kelompok yang saling terkait dan nantinya dapat membentuk sistem secara teratur yang diawasi oleh aktor.
- c. *Scenario* merupakan penjelasan fitur yang ada pada *use case diagram*.
- d. *Activity Diagram* menggambarkan alur aktivitas didalam sistem yang dirancang
- e. *Sequence Diagram* digunakan untuk menggambarkan rangkaian pesan yang dikirim antar *object* juga interaksi yang terjadi di dalamnya.
- f. *Entity Relationship Diagram (ERD)* merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data.

3.3.3 Tahapan Implementasi Sistem

Implementasi koding dilakukan dengan mengacu kepada perancangan perangkat lunak yang dibuat. Implementasi koding ini menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*) dan *framework Code Igniter (CI)* yang akan digunakan untuk membuat dan mengembangkan sistem web. *Software* yang digunakan dalam membuat aplikasi ini adalah Notepad++ dan XAMPP. Notepad++ adalah sebuah *software* untuk media pemrograman web yang mendukung banyak bahasa pemrograman seperti Java, C++, CSS, PHP, Javascript, HTML dan masih banyak lagi. Sedangkan XAMPP merupakan tools yang digunakan untuk server sementara di PC/laptop.

3.3.4 Tahapan Pengujian Sistem

Tahap pengujian dilakukan apabila sistem yang dibuat telah selesai dan siap untuk digunakan pengguna. Pengujian yang dilakukan berguna untuk mengetahui sejauhmana pengimplementasian metode yang digunakan. Tahap pengujian dilakukan dengan mencari kesalahan-kesalahan yang mungkin terjadi, serta melakukan perbaikan untuk penyempurnaan aplikasi ini. Proses pengujian menggunakan *white box* dan *black box*. Menurut Pressman (2005), *white-box testing* adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test-case*. Pengujian *black box* menurut Pressman, (2005) adalah metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak.

3.3.5 Tahapan Pemeliharaan Sistem

Pada tahap ini dilakukan pemeliharaan sistem untuk mengatasi *bug* atau sistem *error* yang tidak ditemukan sebelumnya, atau jika ada penambahan fitur baru.

3.3.6 Analisis Penggunaan Metode

Tahap analisa metode *Simple Additive Weighting* (SAW) ke dalam sistem penunjang keputusan seleksi pemain basket di Jember. Sesuai dengan data yang di dapat dari analisa kebutuhan sistem.

3.3.7 Ketepatan Implementasi

Pengujian implementasi langsung metode *Simple Additive Weighting* (SAW) ke dalam sistem penunjang keputusan seleksi pemain basket yang ada di Jember, melalui program seleksi yang dilakukan oleh pelatih.

3.3.8 Keakuratan Keputusan

Pengujian dilakukan dengan membandingkan hasil rekomendasi dari SPK dengan hasil keputusan dari *decision maker* (penyeleksi). Nilai akurasi dihitung menggunakan rumus *accuracy*.

3.4 Gambaran Umum Sistem

Sistem pendukung keputusan seleksi pemain basket di Jember merupakan suatu sistem penunjang keputusan yang dapat digunakan untuk membantu Perbasi Jember dalam menentukan pemain basket yang memenuhi kriteria pelatih sesuai dengan posisi bermain dalam bola basket (Wawancara Perbasi 2017). Adapun kriteria dan posisi yang ditentukan pelatih sebagai berikut :

Kriteria :

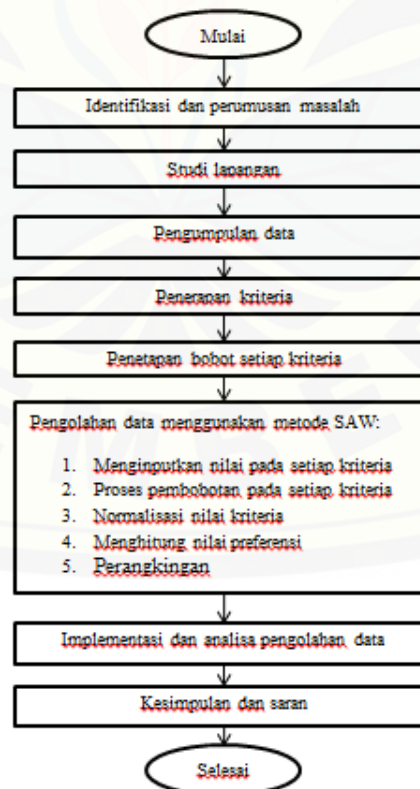
1. *Dribbling* (menggiring bola)
2. *Passing* (mengoper bola)
3. *Shooting* (menembak bola ke ring)
4. Ketahanan fisik pemain
5. *Basketball IQ* (kecerdasan dalam bermain bola basket)
6. Tinggi Badan
7. Berat Badan

Posisi :

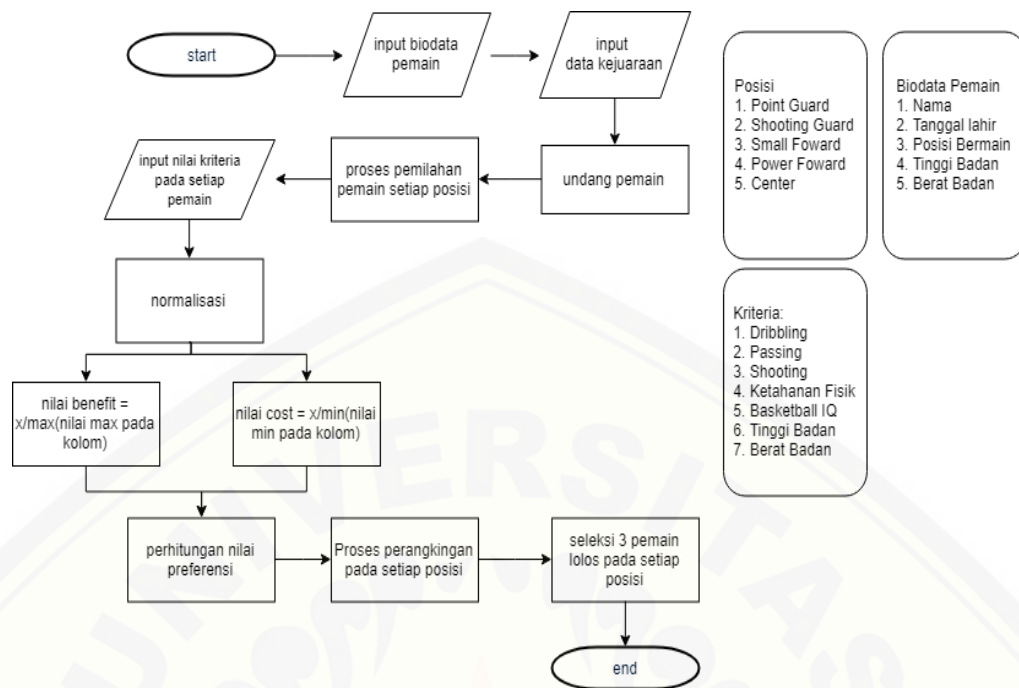
1. *Point Guard* (PG)

2. *Shooting Guard* (SG)
3. *Small Forward* (SF)
4. *Power Forward* (PF)
5. *Center* (C)

Alur penelitian dimulai dengan cara identifikasi dan perumusan masalah, lalu dilanjutkan dengan studi lapangan dan pengumpulan data. Pengumpulan data didapat dengan cara wawancara kepada Kepala Bidang Pelatihan Perbasi Jember. Proses selanjutnya adalah penentuan dan pembobotan kriteria sesuai dengan posisinya. Setiap posisi mempunyai nilai bobot kriteria yang berbeda dikarenakan pemain dengan posisi *point guard* (PG) dan *center* (C) mempunyai tugas yang berbeda dalam bermain bola basket begitu juga dengan posisi lainnya. Kemudian setelah penilaian dan pembobotan dilakukan, dilanjutkan perhitungan serta implementasi metode SAW dan terakhir proses perankingan. Dari uraian di atas, gambaran alir sistem dapat dilihat pada gambar 3.2 dan gambaran implementasi SAW pada gambar 3.3:



Gambar 3. 2 Diagram Alir Penelitian



Gambar 3. 3 Gambaran implementasi SAW

Pada Gambar 3.3, proses pembobotan dilakukan dengan cara membedakan posisi dasar dalam bermain basket. Setiap Posisi mempunyai bobot kriteria yang berbeda, bobot tersebut merupakan keinginan dari pelatih sendiri untuk memberikan prioritas di setiap kriteria yang di. Dalam permainan bola basket terdapat posisi utama pemain, yaitu posisi 1(*Point Guard/PG*), posisi 2(*Shooting Guard/SG*), posisi 3(*Small Forward/SF*), posisi 4(*Power Forward/PF*), posisi 5 (*Center/C*). Setiap Posisi mempunyai tugas yang berbeda beda. PG mempunyai tugas memimpin penyerangan, maka seorang PG harus memiliki kemampuan membawa bola (*Dribbling*) diatas rata-rata dan harus memiliki distribusi bola (*Passing*) yang baik. SG mempunyai tugas menembak (*shooting*) tiga angka, maka seorang SG harus mempunyai tembakan yang baik. SF merupakan seorang *allrounder* yaitu pemain yang mampu mencetak angka dari semua kesempatan menyerang, baik itu *shooting* dan penetrasi kedalam pertahanan lawan (*drive*), maka seorang SF harus mempunyai kemampuan *dribbling* dan *shooting* sama baiknya. PF bertugas menangkap bola pantul (*rebound*) yang gagal masuk ke ring, terutama saat bertahan, maka seorang PF harus bertubuh tinggi, semaking tinggi

pemain, maka semakin mudah untuk melakukan rebound. Biasanya seorang PF juga memiliki tembakan dua angka yang baik. Posisi *center* harus ditempati oleh pemain yang bertubuh paling tinggi dan besar, karena tugas dari seorang *center* menjadi pertahanan terakhir dalam bermain bola basket (Wawancara Perbasi, 2017).

Angka atau nilai bobot yang diberikan mempunyai total 100% (Kusumadewi,2006). Nilai pada setiap bobot mengacu pada tugas utama dalam posisi bermain basket. Berikut data pembobotan kriteria pada setiap posisi pada tabel 3.1 bersumber pada hasil wawancara dengan Kepala Bidang kepelatihan di PERBASI Jember.

Tabel 3. 1 Bobot Setiap Posisi

Posisi	Nilai Bobot						
	Dribbling	Passing	Shooting	Ketahanan Fisik	Basketball IQ	Tinggi Badan	Berat Badan
Posisi 1 (PG)	30%	20%	10%	10%	20%	5%	5%
Posisi 2 (SG)	10%	10%	40%	10%	20%	5%	5%
Posisi 3 (SF)	20%	10%	20%	10%	20%	10%	10%
Posisi 4 (PF)	5%	5%	10%	10%	20%	30%	20%
Posisi 5 (C)	5%	5%	5%	10%	20%	35%	20%

(Sumber: Kepala Bidang Kepelatihan Perbasi Jember)

Setelah proses pembobotan dilakukan berdasarkan kriteria yang dipilih, maka proses selanjutnya akan dimulai dengan proses perankingan pada setiap posisi dengan menggunakan metode SAW. Setiap posisi akan diambil 3 terbaik untuk dijadikan satu tim. Total satu team terdiri dari 15 orang pemain terbaik yang sudah lolos seleksi. Semua data dan informasi yang digunakan adalah hasil dari wawancara dengan Kepala Bidang Kepelatihan di PERBASI Jember.

BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember Tahapan-tahapan pembuatan sistem dilakukan berdasarkan metode *waterfall*.

4.1 Deskripsi Umum Sistem

Deskripsi umum dari sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember yang dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (*statement of purpose*) sistem dan fungsi sistem.

4.1.1 SOP (*Statement of Purpose*)

Sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember merupakan sistem yang mampu menampilkan hasil seleksi berdasarkan kriteria, posisi dan bobot yang telah ditentukan dengan menggabungkan perhitungan dengan menggunakan *Simple Additive Weighting* (SAW). Adapun kriteria yang digunakan adalah *dribbling*, *passing*, *shooting*, ketahanan fisik pemain, *basketball IQ*, tinggi badan, berat badan. Dan posisi yang di gunakan adalah *point guard*, *shooting guard*, *small forward*, *power forward*, *center*. Selanjutnya hasil seleksi akan ditampilkan kepada *user* melalui login pada *website*

4.2 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan melalui wawancara langsung ke kantor sekretariat Perbasi Jember tepatnya dengan Sekertaris Umum dan Bidang Kepelatihan Perbasi Jember. Data yang diperlukan adalah data kriteria, posisi, dan bobot yang nantinya digunakan dalam membangun Sistem.

4.3 Analisis Kebutuhan

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

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) yaitu:

- a. Sistem mampu mengelola data pemain (*view, insert, edit*).
- b. Sistem mampu mengelola data nilai seleksi (*view, insert, edit*).
- c. Sistem mampu mengelola data kejuaraan (*view, insert, edit*).
- d. Sistem mampu mengelola data bobot (*view, edit*).
- e. Sistem mampu menampilkan *ranking* pemain. (*user*).
- f. Sistem mampu mengirimkan undangan melalui dan hasil seleksi kepada *user*.
- g. Sistem mampu balas undangan seleksi kepada (*user*).

Kebutuhan fungsional tersebut didapat dari hasil wawancara kebutuhan sistem dengan Perbasi Jember.

4.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi kebutuhan dari pengguna. Kebutuhan non-fungsional dari sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember yaitu :

- a. Sistem bekerja sesuai dengan fungsinya dan dapat dijalankan pada semua komputer dan semua *browser* yang berbeda.

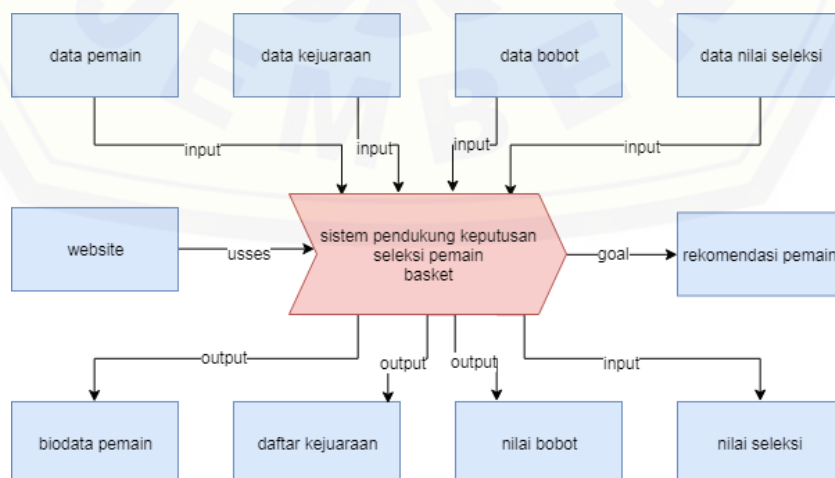
- b. Tampilan dan bahasa komunikasi sistem mudah dimengerti oleh pengguna untuk memberikan kenyamanan pemakaian dan memudahkan pengoperasian.
- c. Sistem menggunakan *username* dan *password* untuk autentifikasi akses admin terhadap sistem, sedangkan *user* hanya dapat *login* untuk melihat undangan seleksi dan hasil seleksi.

4.4 Desain Sistem

Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang dapat digambarkan dengan desain sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember meliputi *business process*, *use case diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

4.4.1 Business Process

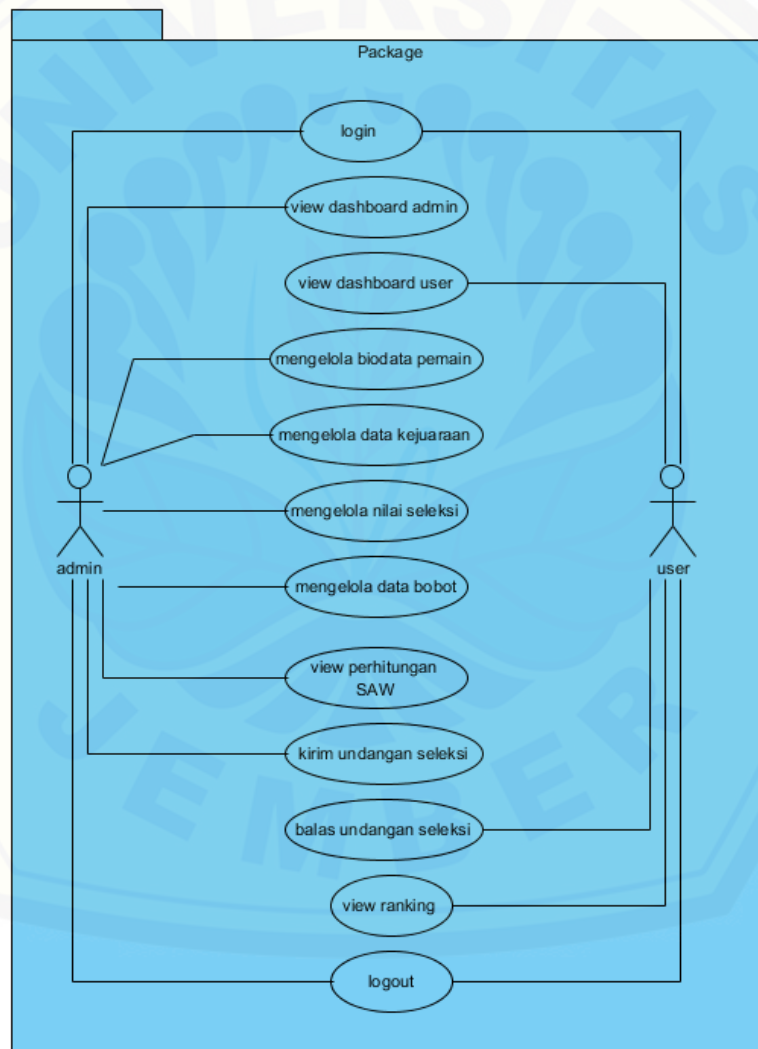
Selain dapat dideskripsikan dalam sebuah SOP (*Statement Of Purpose*), gambaran sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember dapat digambarkan melalui sebuah *business proses*. Seperti yang dapat kita lihat pada Gambar 4.1 menggambarkan data-data yang digunakan sebagai masukan, data keluaran, *uses* sistem yang dibangun hingga *goal* dari dibangunnya sistem sendiri.



Gambar 4. 1 *Business Process* Sistem

4.4.2 Use Case Diagram

Use case diagram merupakan permodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember yang akan dibangun. Melalui *use case* diagram dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing-masing aktor atau pengguna. Pada Gambar 4.2 digambarkan *use case* diagram yang terdiri atas dua aktor dengan sepuluh *use case*.



Gambar 4. 2 Use Case Diagram

Berdasarkan *use case* diagram pada Gambar 4.2 terdapat dua pengguna atau aktor, yaitu *admin* dan *user*. Berikut deskripsi dari masing-masing aktor yang dapat dilihat pada tabel 4.1

Tabel 4. 1 Deskripsi Pembagian Aktor Sistem

No	Aktor	Deskripsi
1.	<i>Admin</i>	Aktor ini memiliki hak untuk mengelola data pemain, data kejuaraan, data nilai seleksi, kirim undangan dan mengelola data bobot.
2.	<i>User</i>	Aktor hanya memiliki hak akses terbatas. Aktor ini hanya dapat login, balas undangan dan melihat hasil ranking seleksi.

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

Tabel 4. 2 Deskripsi *Use Case*

No	<i>Use Case</i>	Deskripsi
1.	<i>Login</i>	<i>Use case</i> yang menggambarkan proses masuk ke dalam Sistem.
2.	<i>View dashboard admin</i>	<i>Use case</i> yang menampilkan halaman dashboard admin yang berisi deskripsi sistem .
3.	<i>View dashboard user</i>	<i>Use case</i> yang menampilkan halaman dashboard user yang berisi deskripsi sistem .
4.	Mengelola biodata pemain	<i>Use case</i> yang menggambarkan proses pengolahan biodata pemain.
5.	Mengelola data kejuaraan	<i>Use case</i> yang menggambarkan proses pengolahan data kejuaraan.
6.	Mengelola nilai seleksi	<i>Use case</i> yang menggambarkan proses pengolahan nilai seleksi.

Dilanjutkan

Lanjutan

7.	Mengelola data bobot	<i>Use case</i> yang menggambarkan proses pengolahan data bobot.
8.	View perhitungan SAW	<i>Use case</i> yang menampilkan halaman perhitungan metode SAW.
9.	Kirim undangan seleksi	<i>Use case</i> yang menggambarkan proses mengirim undangan seleksi.
10.	Balas undangan seleksi	<i>Use case</i> yang menggambarkan proses membalas undangan seleksi.
11.	<i>View ranking</i>	<i>Use case</i> yang menampilkan halaman hasil ranking
12.	<i>Logout</i>	<i>Use case</i> yang menggambarkan proses keluar dari Sistem.

4.4.3 *Use Case* Skenario

Use case skenario adalah dokumentasi terhadap kebutuhan fungsional sistem. sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember adalah sebagai berikut:

a. Skenario *Login*

Skenario *login* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario *login*. Pada skenario tersebut dapat dilihat bagaimana sebuah alur *login* berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon reaksi tersebut sehingga aktor telah melakukan *login* pada sistem tersebut. Proses lengkap skenario *login* dapat dilihat pada Lampiran A Tabel A.1.

b. Skenario Menampilkan *Dashboard admin*

Skenario menampilkan *dashboard admin* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *use case* skenario menampilkan *dashboard admin*. Pada skenario tersebut dapat dilihat bagaimana sebuah alur menampilkan *dashboard* berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon aksi tersebut sehingga aktor dapat mengakses halaman

dashboard admin. Proses lengkap skenario menampilkan *dashboard admin* dapat dilihat pada Lampiran A Tabel A.2.

c. Skenario Menampilkan *dashboard user*

Skenario menampilkan *dashboard user* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *use case* skenario menampilkan *dashboard user*. Pada skenario tersebut dapat dilihat bagaimana sebuah alur menampilkan *dashboard user* berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon aksi tersebut sehingga aktor dapat mengakses halaman *dashboard user*. Proses lengkap skenario menampilkan *dashboard user* dapat dilihat pada Lampiran A Tabel A.3.

d. Skenario Mengelola Biodata Pemain

Skenario mengelola biodata pemain merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario mengelola biodata pemain. Pada skenario tersebut dapat dilihat bagaimana sebuah alur mengelola biodata pemain berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola biodata pemain. Proses lengkap skenario mengelola biodata pemain dapat dilihat pada Lampiran A Tabel A.4.

e. Skenario Mengelola Data Kejuaraan

Skenario mengelola data kejuaraan merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario mengelola data kejuaraan. Pada skenario tersebut dapat dilihat bagaimana sebuah alur mengelola data kejuaraan berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data kejuaraan. Proses lengkap skenario mengelola data data kejuaraan dapat dilihat pada Lampiran A Tabel A.5.

f. Skenario Mengelola Nilai Seleksi

Skenario mengelola nilai seleksi merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario mengelola nilai seleksi. Pada skenario tersebut dapat dilihat bagaimana sebuah alur mengelola nilai seleksi berjalan setelah mendapat aksi dari aktor dan reaksi

dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola nilai seleksi. Proses lengkap skenario mengelola nilai seleksi dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Skenario Mengelola Nilai Seleksi

Nama Use Case	Mengelola nilai seleksi
Participating Actor	Admin
Entry Condition	Daftar kejuaraan
Exit Condition	Nilai telah selesai dikelola
Event Flow	
Normal Flow : memasukkan nilai pemain	
Aktor	Reaksi Sistem
1.klik tombol info pada menu daftar kejuaraan	
	2. menampilkan halaman data pemain seleksi
3. klik tombol tambah	
	4. menampilkan menu tambah pemain seleksi
5. pilih pemain	
6. input nilai seleksi	
7. klik tombol submit	
	8. data nilai selesai di input
Alternative Flow : Form kosong	
7. klik tombol submit	
	8. Menampilkan pesan "please fill out this field"
Normal Flow : edit nilai pemain	
1.klik tombol info pada menu daftar kejuaraan	
	2. menampilkan halaman data pemain seleksi
3. klik tombol edit	
	4. menampilkan halaman form ubah nilai
5. input nilai seleksi	
6. klik tombol submit	
	7. berhasil ubah nilai seleksi
Alternative Flow : Form kosong	
6. klik tombol submit	
	7. Menampilkan pesan "please fill out this field"

g. Skenario Mengelola Data Bobot

Skenario mengelola data bobot merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif *use case* skenario mengelola data bobot. Pada skenario tersebut dapat dilihat bagaimana sebuah alur mengelola data bobot berjalan setelah mendapat aksi dari aktor dan reaksi dari sistem untuk merespon aksi tersebut sehingga aktor telah mengelola data bobot. Proses lengkap skenario mengelola data bobot dapat dilihat pada Lampiran A Tabel A.7.

h. Skenario Menampilkan perhitungan SAW

Skenario menampilkan perhitungan SAW merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *use case* skenario menampilkan perhitungan SAW. Pada skenario tersebut dapat dilihat bagaimana sebuah alur menampilkan perhitungan SAW berjalan setelah mendapat aksi dari aktor dan reaksi sistem untuk merespon aksi tersebut sehingga aktor dapat melihat halaman perhitungan SAW. Proses lengkap skenario menampilkan perhitungan SAW dapat dilihat pada Lampiran A Tabel A.8.

i. Skenario Kirim Undangan Seleksi

Skenario kirim undangan seleksi merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *use case* skenario mengirim undangan seleksi. Pada skenario tersebut dapat dilihat bagaimana sebuah alur mengirim undangan seleksi berjalan setelah mendapat aksi dari aktor dan reaksi sistem untuk merespon aksi tersebut sehingga aktor dapat mengirim undangan seleksi. Proses lengkap skenario kirim undangan seleksi dapat dilihat pada Lampiran A Tabel A.9.

j. Skenario Balas Undangan Seleksi

Skenario balas undangan seleksi merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *use case* skenario membalas undangan seleksi. Pada skenario tersebut dapat dilihat bagaimana sebuah alur membalas undangan seleksi berjalan setelah mendapat aksi dari aktor dan reaksi sistem untuk merespon aksi tersebut sehingga aktor dapat membalas undangan seleksi.

Proses lengkap skenario balas undangan seleksi dapat dilihat pada Lampiran A Tabel A.10.

k. Skenario Menampilkan *Ranking*

Skenario menampilkan *ranking* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *use case* skenario menampilkan *ranking*. Pada skenario tersebut dapat dilihat bagaimana sebuah alur menampilkan *ranking* berjalan setelah mendapat aksi dari aktor dan reaksi sistem untuk merespon aksi tersebut sehingga aktor dapat melihat halaman *ranking*. Proses lengkap skenario menampilkan *ranking* dapat dilihat pada Lampiran A Tabel A.11.

l. Skenario Logout

Skenario *logout* merupakan penjelasan urutan reaksi aktor dan reaksi sistem pada skenario normal *use case* skenario mengelola perhitungan. Pada skenario tersebut dapat dilihat bagaimana sebuah alur *logout* berjalan setelah mendapat aksi dari aktor dan reaksi sistem untuk merespon aksi tersebut sehingga aktor telah keluar dari sistem. Proses lengkap skenario *logout* dapat dilihat pada Lampiran A Tabel A.12.

4.4.4 Activity Diagram

Activity Diagram menggambarkan aliran aktivitas sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember yang akan dibangun. sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) memiliki sebelas *activity* diagram yaitu sebagai berikut:

a. *Activity* Diagram *Login*

Activity diagram *Login* dilakukan oleh admin Perbasi Jember dan *user*/pemain. *Activity* diagram *login* menjelaskan tentang bagaimana sistem dapat menjalankan fungsi melakukan autentifikasi hak akses aktor dalam menggunakan sistem. Proses lengkapnya dapat dilihat pada Lampiran B Gambar B.1.

b. *Activity* Diagram Menampilkan *Dashboard admin*

Activity diagram menampilkan *dashboard admin* dapat dilakukan oleh admin Perbasi Jember. *Activity* diagram menampilkan *dashboard admin* berisi halaman

dashboard dengan deskripsi sistem. Proses lengkapnya dapat dilihat pada Lampiran B Gambar B.2.

c. *Activity Diagram Menampilkan Dashboard user*

Activity diagram menampilkan dashboard user dapat dilakukan *user/pemain*. *Activity diagram menampilkan dashboard user* berisi halaman dashboard dengan deskripsi sistem. Proses lengkapnya dapat dilihat pada Lampiran B Gambar B.3.

d. *Activity Diagram Mengelola Biodata Pemain*

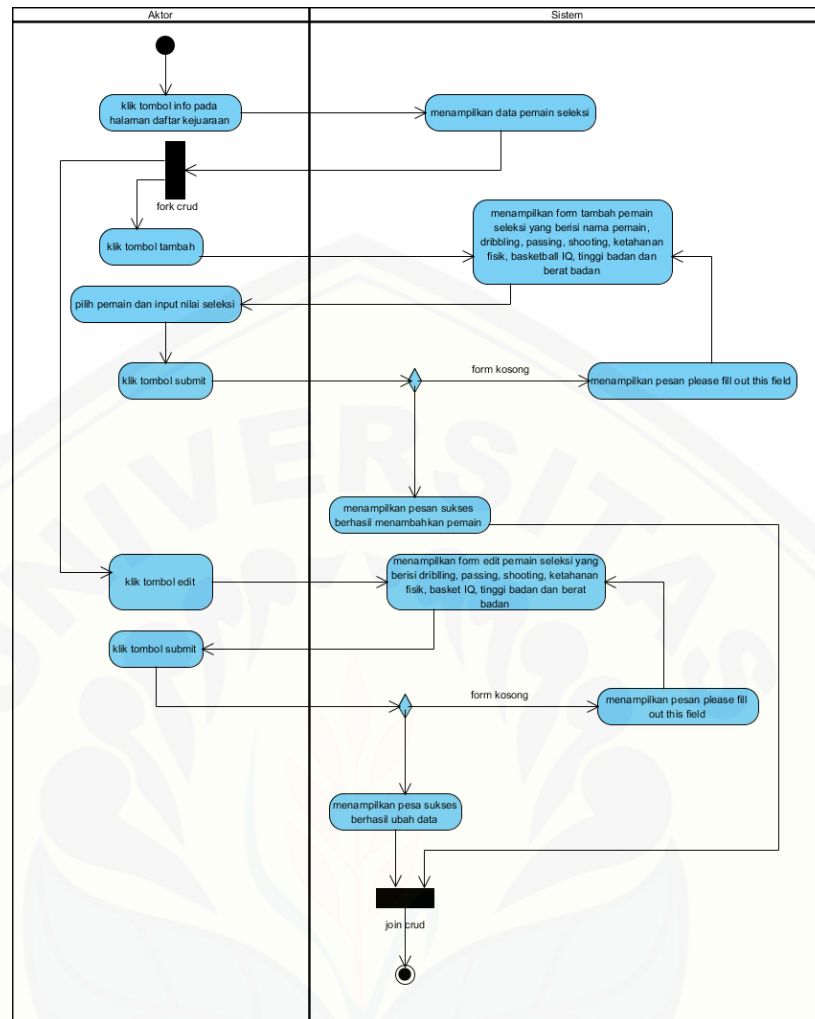
Activity diagram mengelola biodata pemain dapat dilakukan oleh admin Perbasi Jember. *Activity diagram mengelola biodata pemain* menjelaskan bagaimana admin dapat mengelola biodata pemain pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.4.

e. *Activity Diagram Mengelola Data Kejuaraan*

Activity diagram mengelola data kejuaraan dapat dilakukan oleh admin Perbasi Jember. *Activity diagram mengelola data kejuaraan* menjelaskan bagaimana admin dapat mengelola data kejuaraan pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.5.

f. *Activity Diagram Mengelola Nilai Seleksi*

Activity diagram mengelola nilai seleksi dapat dilakukan oleh admin Perbasi Jember. *Activity diagram mengelola nilai seleksi* menjelaskan bagaimana admin dapat mengelola nilai seleksi pada sistem, proses lengkapnya dapat dilihat pada Gambar 4.3.



Gambar 4. 3 Activity Diagram Login

g. Activity Diagram Mengelola Data Bobot

Activity diagram mengelola data bobot dapat dilakukan oleh admin Perbasi Jember. Activity diagram mengelola data bobot menjelaskan bagaimana admin dapat mengelola data bobot pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.7.

h. Activity Diagram Menampilkan Perhitungan SAW

Activity diagram menampilkan perhitungan SAW dapat dilakukan oleh Admin Perbasi Jember. Activity diagram menampilkan perhitungan SAW berisi sebuah diagram yang menjelaskan bagaimana admin menampilkan informasi perhitungan SAW pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.8.

i. Activity Diagram Kirim Undangan Seleksi

Activity diagram kirim undangan seleksi dapat dilakukan oleh *admin* Perbasi Jember. *Activity* diagram kirim undangan seleksi berisi sebuah diagram yang menjelaskan bagaimana *admin* mengirim pesan/*e-mail* yang kemudian diterima oleh user pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.9.

j. Activity Diagram Balas Undangan Seleksi

Activity diagram balas undangan seleksi dapat dilakukan oleh *user/pemain*. *Activity* diagram balas undangan seleksi berisi sebuah diagram yang menjelaskan bagaimana *user* membalas pesan/*e-mail* yang kemudian dikirim ke *admin* pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.10.

k. Activity Diagram Menampilkan *Ranking*

Activity diagram menampilkan perhitungan *ranking* dapat dilakukan oleh *user/pemain*. *Activity* diagram menampilkan *ranking* berisi sebuah diagram yang menjelaskan bagaimana *user* menampilkan informasi *ranking* pada sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.11.

l. Activity Diagram Logout

Activity diagram *logout* dapat dilakukan oleh *admin* Perbasi Jember dan *user/pemain*. *Activity* diagram *logout* berisi sebuah diagram yang menjelaskan bagaimana *admin* dan *user* untuk keluar dari sistem, proses lengkapnya dapat dilihat pada Lampiran B Gambar B.12.

4.4.5 Sequence Diagram

Sequence diagram adalah dokumentasi suatu diagram yang menampilkan urutan interaksi – interaksi objek di dalam sistem. *Sequence* diagram digunakan untuk menggambarkan skenario dan aliran logika dalam sistem dengan cara *visual*. *Sequence* diagram dari sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember adalah sebagai berikut:

a. *Sequence* Diagram Login

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

b. *Sequence* Diagram Menampilkan *Dashboard Admin*

Penggambaran *sequence* menampilkan *dashboard admin* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menampilkan *dashboard admin*. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menampilkan *dashboard admin* lebih lengkap akan dijelaskan pada Lampiran C Gambar C.2.

c. *Sequence* Diagram Menampilkan *Dashboard user*

Penggambaran *sequence* menampilkan *dashboard user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menampilkan *dashboard user*. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menampilkan *dashsboard user* lebih lengkap akan dijelaskan pada Lampiran C Gambar C.3.

d. *Sequence* Diagram Mengelola Biodata Pemain

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

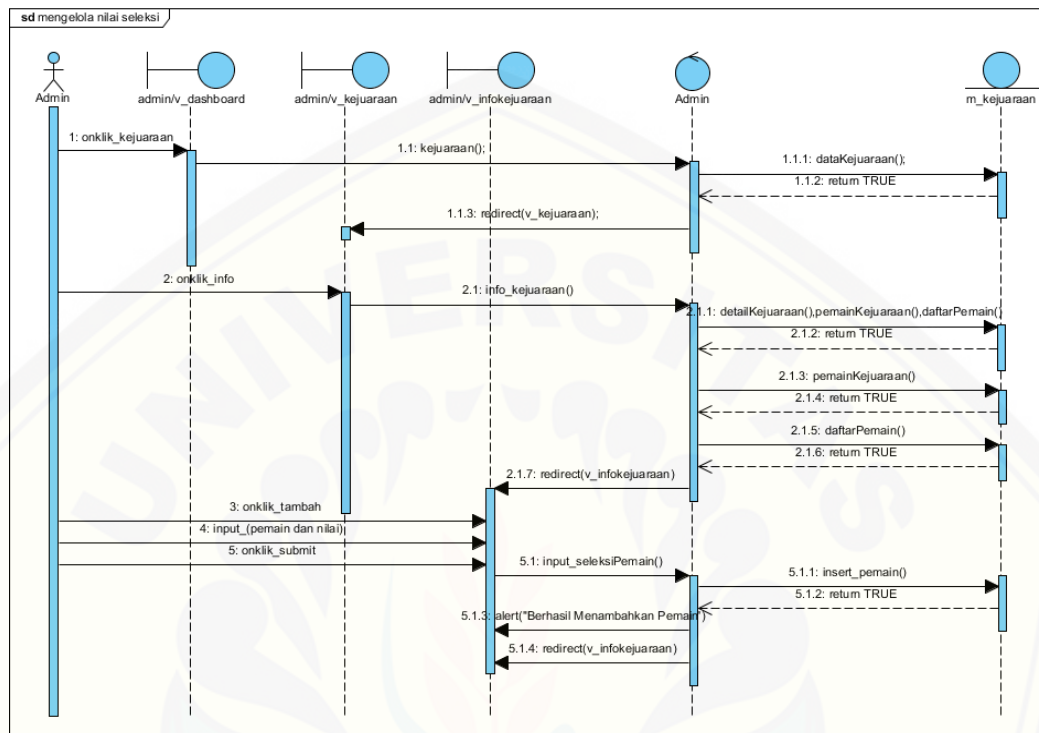
e. *Sequence* Diagram Mengelola Data Kejuaraan

Penggambaran *sequence* diagram mengelola data kejuaraan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola data kejuaraan. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola data kejuaraan lebih lengkap akan dijelaskan pada Lampiran C Gambar C.5.

f. *Sequence* Diagram Mengelola Nilai Seleksi

Penggambaran *sequence* diagram mengelola nilai seleksi digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* mengelola nilai

seleksi. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram mengelola nilai seleksi lebih lengkap akan dijelaskan pada Gambar 4.4.



Gambar 4. 4 Sequence Diagram Mengelola Nilai Seleksi

g. *Sequence* Diagram Mengelola Data Bobot

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

h. *Sequence* Diagram Menampilkan Perhitungan SAW

Penggambaran *sequence* diagram menampilkan perhitungan SAW digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menampilkan perhitungan SAW. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence* diagram menampilkan perhitungan SAW lebih lengkap akan dijelaskan pada Lampiran C Gambar C.8.

i. *Sequence Diagram* Kirim Undangan Seleksi

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

j. *Sequence Diagram* Balas Undangan Seleksi

Penggambaran *sequence diagram* balas undangan seleksi digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* balas undangan seleksi. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* balas undangan seleksi lebih lengkap akan dijelaskan pada Lampiran C Gambar C.10.

k. *Sequence Diagram* Menampilkan *Ranking*

Penggambaran *sequence diagram* menampilkan *ranking* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* menampilkan *ranking*. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* menampilkan *ranking* lebih lengkap akan dijelaskan pada Lampiran C Gambar C.11.

l. *Sequence Diagram* Logout

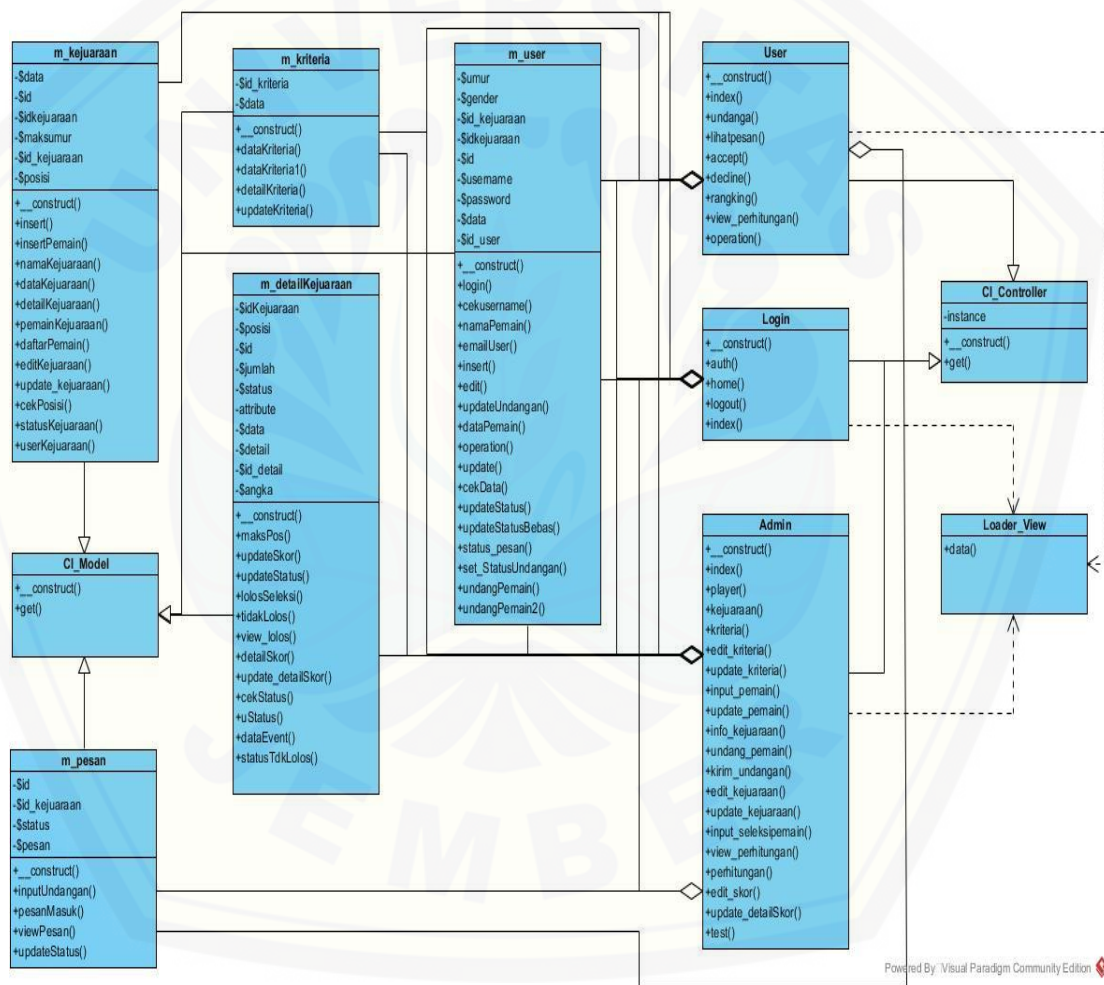
Penggambaran *sequence diagram* *logout* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat pada *use case* *logout*. Masing – masing *class* akan ditampilkan secara *visual* dengan gambar. *Sequence diagram* *logout* lebih lengkap akan dijelaskan pada Lampiran C Gambar C.12.

4.4.6 *Class Diagram*

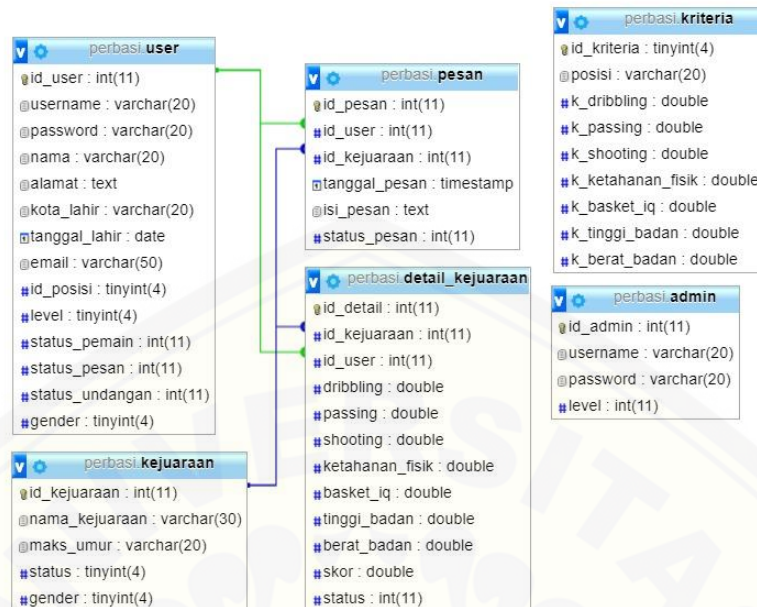
Class diagram menggambarkan hubungan antara kelas yang digunakan untuk membangun suatu sistem. Dalam paradigma OOP (*Object Oriented Programming*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller*. Berdasarkan *sequence diagram* yang telah dibangun, *class diagram* sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember dapat dilihat pada Gambar 4.5.

4.4.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember merupakan gambaran komponen, relasi dan struktur *database* yang digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 5 entitas yang dapat dilihat pada Gambar 4.6.



Gambar 4. 5 Class Diagram



Gambar 4. 6 Entity Relationship Diagram

4.5 Implementasi

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

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

Kode program perhitungan metode *Simple Additive Weighting (SAW)* terdapat di class Admin pada *package controller*. Penulisan kode program perhitungan metode *Simple Additive Weighting* dapat dilihat pada Tabel 4.5.

Tabel 4. 4 Kode Metode SAW *function* perhitungan()

Kode
<pre>public function perhitungan(\$id_kejuaraan){ \$idkejuaraan = \$id_kejuaraan; \$posisi = array(1,2,3,4,5); \$data['pemain'] = \$this->kejuaraan->pemainKejuaraan(\$idkejuaraan); foreach (\$posisi as \$key => \$value) { \$data['pos'][\$value] = \$this->dKejuaraan->maksPos(\$id_kejuaraan,\$value); } }</pre>
Dilanjutkan

Lanjutan

```

foreach ($data['pemain'] as $keyPemain => $pemain) {
    foreach ($data['pos'] as $keyPos => $pos) {
        if($pemain['id_posisi'] == $keyPos){
            $result[$keyPemain]['id_detail'] = $pemain['id_detail'];
            $result[$keyPemain]['id_kejuaraan'] =
            $pemain['id_kejuaraan'];
            $result[$keyPemain]['id_user'] = $pemain['id_user'];
            $result[$keyPemain]['id_posisi'] = $pemain['id_posisi'];
            $result[$keyPemain]['nama'] = $pemain['nama'];

//NORMALISASI
            $result[$keyPemain]['nDribling'] = $pemain['dribbling']
            / $pos['maksDrib'];
            $result[$keyPemain]['nPass'] =
            $pemain['passing'] / $pos['maksPass'];
            $result[$keyPemain]['nShoot'] =
            $pemain['shooting'] / $pos['maksShoot'];
            $result[$keyPemain]['nFisik'] =
            $pemain['ketahanan_fisik'] / $pos['maksFisik'];
            $result[$keyPemain]['nIq'] =
            $pemain['basket_iq'] / $pos['maksIq'];
            $result[$keyPemain]['nTB'] =
            $pemain['tinggi_badan'] / $pos['maksTB'];
            $result[$keyPemain]['nBB'] =
            $pemain['berat_badan'] / $pos['maksBB'];

//PERKALIAN KE NILAI KRITERIA
            $result[$keyPemain]['skor_Drib'] =
            round($result[$keyPemain]['nDribling'] * $pemain['k_dribbling'],4);
            $result[$keyPemain]['skor_Pass'] =
            round($result[$keyPemain]['nPass'] * $pemain['k_passing'],4);
            $result[$keyPemain]['skor_Shoot'] =
            round($result[$keyPemain]['nShoot'] * $pemain['k_shooting'],4);
            $result[$keyPemain]['skor_Fisik'] =
            round($result[$keyPemain]['nFisik'] * $pemain['k_ketahanan_fisik'],4);
            $result[$keyPemain]['skor_Iq'] =
            round($result[$keyPemain]['nIq'] * $pemain['k_basket_iq'],4);
            $result[$keyPemain]['skor_TB'] =
            round($result[$keyPemain]['nTB'] * $pemain['k_tinggi_badan'],4);
            $result[$keyPemain]['skor_BB'] =
            round($result[$keyPemain]['nBB'] * $pemain['k_berat_badan'],4);

//JUMLAH AKHIR
            $result[$keyPemain]['jumlah'] =
            round($result[$keyPemain]['skor_Drib'] +
            $result[$keyPemain]['skor_Pass'] +
            $result[$keyPemain]['skor_Shoot'] +
            $result[$keyPemain]['skor_Fisik'] +
            $result[$keyPemain]['skor_Iq'] +
            $result[$keyPemain]['skor_TB'] +
            $result[$keyPemain]['skor_BB'],4);
            $update = $this->dKejuaraan->updateSkor($result[$keyPemain]['id_detail'],$result[$keyPemain]['jumlah']);
            $update = $this->user->updateStatusBebas($pemain['id_user'],$pemain['id_kejuaraan']);

```

Dilanjutkan

```

Lanjutkan

$update = $this->user->updateUndangan($pemain['id_user'],0);
    }
}

foreach ($posisi as $key => $value) {
    $data['lolosSeleksi'][$value] = $this->dKejuaraan-
>lolosSeleksi($id_kejuaraan,$value);
}

for ($i=1; $i <= count($posisi) ; $i++) {
    foreach ($data['lolosSeleksi'][$i] as $key => $value) {
        $data[$i][$key]['id_detail'] = $value['id_detail'];
        $data[$i][$key]['id_kejuaraan'] = $value['id_kejuaraan'];
        $data[$i][$key]['id_posisi'] = $value['id_posisi'];
        $data[$i][$key]['skor'] = $value['skor'];
        $data[$i][$key]['status'] = 1;

        $update = $this->dKejuaraan-
>updateStatus($data[$i][$key]['id_detail'],$data[$i][$key]['status']);
    }
}

$data['tidakLolos'] = $this->dKejuaraan->tidakLolos($id_kejuaraan);
foreach ($data['tidakLolos'] as $key => $value) {
    $update = $this->dKejuaraan->statusTdkLolos($value['id_detail']);
}

$update = $this->kejuaraan->statusKejuaraan($idkejuaraan);
pesan('sukses', 'Seleksi pemain selesai');
redirect(base_url(). 'Admin/view_perhitungan/'.$idkejuaraan);
}

```

Pada Tabel 4.4 merupakan kode perhitungan metode *Simple Additive Weighting* pada package *controller* Admin. Pada kode program tersebut menjelaskan tahapan atau langkah – langkah proses perhitungan metode *Simple Additive Weighting* yang menghasilkan perhitungan kriteria yang ada. Hasil perhitungan tersebut adalah ranking hasil seleksi.

4.6 Pengujian

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan sebuah metode pengujian sistem untuk mengetahui apakah kebutuhan sistem telah

terpenuhi dan sistem layak untuk digunakan oleh pengguna. Agar pengujian yang dilakukan lebih valid, maka tahap pengujian ini dilakukan dengan menggunakan dua metode, yaitu *white box* dan *black box*.

4.6.1 Metode White Box

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Hal tersebut bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *white box* dilakukan oleh peneliti dengan cara menghitung *independent path* yaitu dengan menggunakan suatu pengukuran kuantitatif *cyclomatic complexity*, listing program, penentuan jalur independen, dan *test case*. Tahapan pengujian menggunakan metode *white box* pada sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember meliputi:

1. Listing Program

Listing program merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program. Berikut *listing program* sistem pada class controller Admin *function* perhitungan() di gambar 4.7.

```

248 public function perhitungan($id_kejuaraan){
249     $idkejuaraan = $id_kejuaraan;
250     $posisi = array(1,2,3,4,5);
251     $data['pemain'] = $this->kejuaraan->pemainkejuaraan($idkejuaraan);
252     foreach ($posisi as $key => $value) {
253         $data['pos'][$value] = $this->kejuaraan->maksPos($id_kejuaraan,$value);
254     }
255
256     foreach ($data['pemain'] as $keyPemain => $pemain) {
257         foreach ($data['pos'] as $keyPos => $pos) {
258             if($pemain['id_posisi'] == $keyPos){
259                 $result[$keyPemain]['id_detail'] = $pemain['id_detail'];
260                 $result[$keyPemain]['id_kejuaraan'] = $pemain['id_kejuaraan'];
261                 $result[$keyPemain]['id_user'] = $pemain['id_user'];
262                 $result[$keyPemain]['id_posisi'] = $pemain['id_posisi'];
263                 $result[$keyPemain]['nama'] = $pemain['nama'];
264
265                 //NORMALISASI
266                 $result[$keyPemain]['nDribling'] = $pemain['dribbling'] / $pos['maksDrib'];
267                 $result[$keyPemain]['nPass'] = $pemain['passing'] / $pos['maksPass'];
268                 $result[$keyPemain]['nShoot'] = $pemain['shooting'] / $pos['maksShoot'];
269                 $result[$keyPemain]['nFisik'] = $pemain['ketahanan_fisik'] / $pos['maksFisik'];
270                 $result[$keyPemain]['nIq'] = $pemain['basket iq'] / $pos['maksIq'];
271                 $result[$keyPemain]['nTB'] = $pemain['tinggi_badan'] / $pos['maksTB'];
272                 $result[$keyPemain]['nBB'] = $pemain['berat_badan'] / $pos['maksBB'];
273
274                 //PERKALIAN KE NILAI KRITERIA
275                 $result[$keyPemain]['skor_Drib'] = round($result[$keyPemain]['nDribling'] * $pemain['k_dribling'],4);
276                 $result[$keyPemain]['skor_Pass'] = round($result[$keyPemain]['nPass'] * $pemain['k_passing'],4);
277                 $result[$keyPemain]['skor_Shoot'] = round($result[$keyPemain]['nShoot'] * $pemain['k_shooting'],4);
278                 $result[$keyPemain]['skor_Fisik'] = round($result[$keyPemain]['nFisik'] * $pemain['k_ketahanan_fisik'],4);
279                 $result[$keyPemain]['skor_Iq'] = round($result[$keyPemain]['nIq'] * $pemain['k_basket iq'],4);
280                 $result[$keyPemain]['skor_TB'] = round($result[$keyPemain]['nTB'] * $pemain['k_tinggi_badan'],4);
281                 $result[$keyPemain]['skor_BB'] = round($result[$keyPemain]['nBB'] * $pemain['k_berat_badan'],4);
282
283                 //JUMLAH AKHIR
284                 $result[$keyPemain]['jumlah'] = round($result[$keyPemain]['skor_Drib'] + $result[$keyPemain]['skor_Pass'] + $result[$keyPemain]['skor_Shoot'] + $result[$keyPemain]['skor_Fisik'] + $result[$keyPemain]['skor_Iq'] + $result[$keyPemain]['skor_TB'] + $result[$keyPemain]['skor_BB'],4);
285
286                 $update = $this->kejuaraan->updateSkor($result[$keyPemain]['id_detail'],$result[$keyPemain]['jumlah']);
287                 $update = $this->user->updateStatusBebas($pemain['id_user'],$pemain['id_kejuaraan']);
288                 $update = $this->user->updateUndangan($pemain['id_user'],0);
289             }

```

```

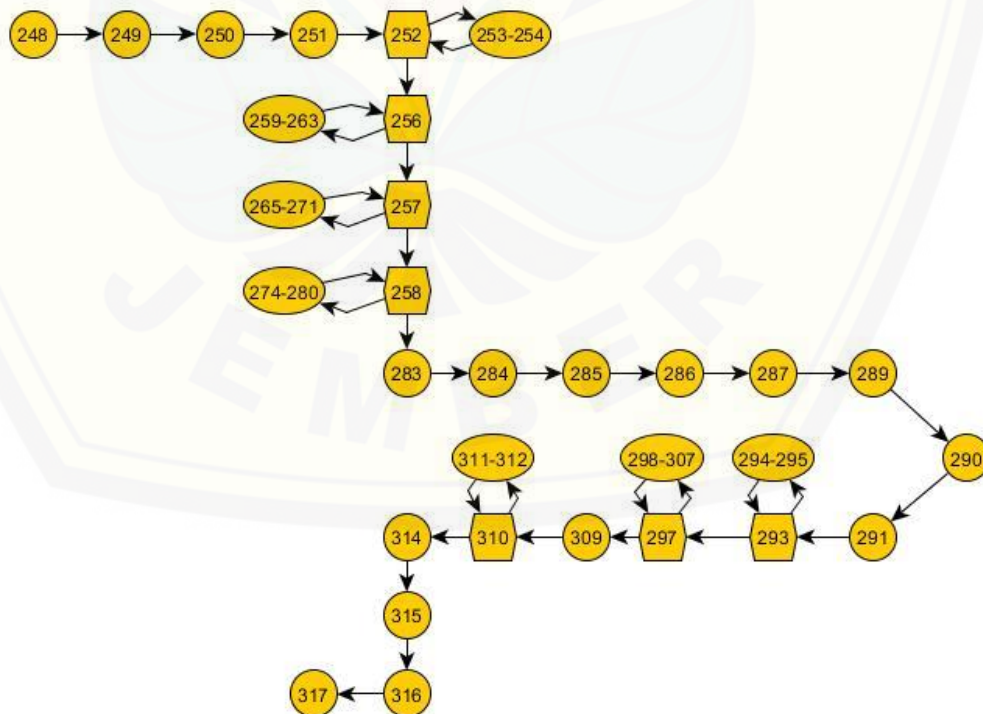
290     }
291   }
292
293   foreach ($posisi as $key => $value) {
294     $data['lolosSeleksi'][$value] = $this->dKejuaraan->lolosSeleksi($id_kejuaraan,$value);
295   }
296
297   for ($i=1; $i <= count($posisi) ; $i++) {
298     foreach ($data['lolosSeleksi'][$i] as $key => $value) {
299       $data[$i][$key]['id_detail'] = $value['id_detail'];
300       $data[$i][$key]['id_kejuaraan'] = $value['id_kejuaraan'];
301       $data[$i][$key]['id_posisi'] = $value['id_posisi'];
302       $data[$i][$key]['skor'] = $value['skor'];
303       $data[$i][$key]['status'] = 1;
304     }
305     $update = $this->dKejuaraan->updateStatus($data[$i][$key]['id_detail'],$data[$i][$key]['status']);
306   }
307 }
308
309 $data['tidakLolos'] = $this->dKejuaraan->tidakLolos($id_kejuaraan);
310 foreach ($data['tidakLolos'] as $key => $value) {
311   $update = $this->dKejuaraan->statusTidakLolos($value['id_detail']);
312 }
313
314 $update = $this->kejuaraan->statusKejuaraan($idkejuaraan);
315 pesan('sukses', 'Seleksi pemain selesai');
316 redirect(base_url(). 'Admin/view_perhitungan/' . $idkejuaraan);
317 }

```

Gambar 4. 7 Listing Program Perhitungan()

2. Diagram Alir

Diagram alir merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari listing program. Diagram alir digambarkan dengan node-node (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program. Berikut diagram alir class controller Admin *function* perhitungan() pada gambar 4.8.



Gambar 4. 8 Diagram Alir Class Controller Admin Function Perhitungan()

a. Kompleksitas Siklomatik (*cyclomatic complexity*)

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

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

Keterangan :

V(G) : Kompleksitas siklomatik

E : Jumlah *edge*

N : Jumlah *node*

Berikut kompleksitas siklomatik sistem pada class controller Admin *function* perhitungan().

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 37 - 31 + 2 \\ &= 8 \end{aligned}$$

3. Pengujian Basis Set (*Test Set*)

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis set yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali. Jalur independen pada class controller Admin *function* perhitungan() dapat di lihat pada tabel

Tabel 4. 5 *Test Case Function* Perhitungan()

<i>Test case function</i> perhitungan()	
Jalur 1	
<i>Test Case</i>	Jika Berhasil Mengelola Nilai Seleksi
Target yang diharapkan	Menampilkan hasil perhitungan Nilai Seleksi setiap tahap

Dilanjutkan

Lanjutan

Hasil Pengujian	Benar
Path/Jalur	248-249-250-251-252-256-257-258-283-284-285-287-289-290-291-293-297-309-310-314-315-316-317
Jalur 2	
Test Case	Jika Berhasil Mengelola Nilai Seleksi
Target yang diharapkan	Menampilkan hasil perhitungan Nilai Seleksi setiap tahap
Hasil Pengujian	Benar
Path/Jalur	248-249-250-251-252-(253-254)-256-(259-263)-257-(265-271)-258-(274-280)-283-284-285-286-287-289-290-291-293-(294-295)-297-(298-307)-309-310-(311-312)-314-315-316-317

4.6.2 Metode *Black Box*

Black box testing merupakan metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari software. Pengujian *black box* pada sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember dilakukan untuk mengetahui apakah masukan dan keluaran dari sistem sesuai dengan kebutuhan fungsional atau tidak. Pengujian dilakukan pada setiap *usecase*. Hasil pengujian *black box* dapat dilihat pada Lampiran E (Pengujian *Black Box*).

BAB 6. PENUTUP

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah:

- a. Penerepan metode *Simple Additive Weighting* dalam sistem pendukung keputusan seleksi pemain basket ini mampu menentukan pemain yang cocok sesuai dengan pelatih berdasarkan skor setiap kriteria yang nantinya akan digunakan rumus Normalisasi untuk menormalisasikan nilai pada tiap skor kriteria. Setelah tahap normalisasi kemudian perhitungan akan dilanjutkan penghitungan nilai preferensi menggunakan rumus Preferensi pada tiap data normalisasi. Perangkingan dilakukan berdasarkan nilai preferensi dan data yang diuji diurutkan berdasarkan nilai preferensi tertinggi yang kemudian di seleksi 3 teratas berdasarkan nilai preferensi pada setiap posisi, setelah setiap posisi mendapatkan 3 teratas, maka sistem memberikan 15 pemain yang lolos seleksi. Pada penelitian ini metode *Simple Additive Weighting* memiliki tingkat akurasi yang tinggi yaitu 100% dengan pengambil keputusan tetap pada *decision maker*-nya atau pelatih.
- b. Pembangunan sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) berdasarkan model *waterfall*. Pada tahap Analisa, persyaratan pembangunan sistem didapatkan dari hasil observasi dan wawancara kepada pihak terkait. Pada tahap – tahap berikutnya pembangunan sistem dilakukan berdsarkan hasil analisa dan implementasi sistem. Pembuatan sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) di Perbasi Jember yang digunakan untuk memberikan rekomendasi pemain kepada pelatih.

6.2 Saran

Adapun saran yang ditujukan untuk memberikan masukan yang lebih baik yaitu :

- a. Sistem pendukung keputusan seleksi pemain basket menggunakan *Simple Additive Weighting* (SAW) ini membutuhkan pengembangan lebih lanjut

dengan membangun sistem yang dinamis sehingga pada sistem dapat ditambahkan kriteria tanpa merubah kode pemrograman dikarenakan dalam penambahan kriteria pada metode ini akan merubah nilai bobot yang ada dalam kode program.



DAFTAR PUSTAKA

- Eniyati, Sri. 2011. Perancangan Sistem Pendukung Pengambilan Keputusan untuk Penerimaan Beasiswa dengan Metode Simple Additive Weighting (SAW), *Jurnal Teknologi Informasi Dinamik*, Vol. 16 No. 2
- Horongbala, Rastafari. 2014, Teknik Dasar dalam Permainan Bola Basket, Desember 2014
- Kusrini. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Penerbit Andi Offset, Yogyakarta.
- Kusumadewi, Hartati, Harjoko, dan Wardoyo. 2006. *Fuzzy Multi-Attribute Decision Making (Fuzzy MADM)*. Yogyakarta: Graha Ilmu.
- Murdani. 2013. Sistem Pendukung Keputusan Pemilihan Pekerjaan Bagi Alumni Teknik Informatika Dengan Metode Simple Additive Weighting (SAW) *.Pelita Informatika Budi Darma*. Vol. IV No. 2, Agustus 2013.
- Munawan dan Siddiq, A.F. 2012. Sistem Pendukung Keputusan Menggunakan Metode Technique for Order by Similarity to Ideal Solution (TOPSIS). *Jurnal Sistem Informasi*, Vol. 4 No. 1, April 2012. hal. 398 – 412
- Perbasi. (2017). Persatuan Bola Basket Seluruh Indonesia. Dipetik September 10, 2017, dari http://perbasi.kontra.web.id/ind/20251918/Perbasi_51326_ibmb_perbasi-kontra.html
- Powers, D.M.W. 2011. “Evaluation: from precision, recall and f- measure to ROC, informedness, markedness & correlation”. *Journal 12 of Machine Learning Technologies*. ISSN: 2229-3981 & ISSN: 2229- 399X, Volume 2, Issue 1, 2011, pp- 37-63
- Pressman, Roger S. 2005. *Rekayasa Perangkat Lunak*. Penerbit Andi Offset, Yogyakarta.

- Pressman, R. S. (2010). *Software Engineering : a Practitioner's Approach Seventh Edition*. New York: McGraw-Hill Companies, Inc.
- Proboyekti, U. (2013, Oktober 23). *Software Process Model*. Dipetik September 9, 2016, dari <http://repository.binus.ac.id/content/A0194/A019411884.pdf>
- Rusadi, W. A. (2002). *Hard Mathematics Applied to Soft Decision*. Surabaya, Jawa Timur, Indonesia: INSAHP II. Universitas Kristen Petra.
- Rusadi, W. A. (2013). *White Box Testing & Black Box Testing*. Dipetik Agustus 2016, 10, dari <http://bangwildan.web.id/berita-176-white-box-testing--black-box-testing.html>
- Sabar, Gery. 2015, *Pemrograman Berorientasi Objek Dengan PHP5*, Januari 2015
- Sigit, Praditya. 2016, *Sistem Penunjang Keputusan Dalam Pemilihan Pemain Untuk Posisi Tertentu Pada Sepak Bola. Seminar Nasional Teknologi Informasi dan Multimedia*, 6-7 Februari 2016.
- Sommerville, Ian. 2011. *Software Engineering (Rekayasa Perangkat Lunak)*. Erlangga, Jakarta.
- Sutariyani, H.Ary Setyadi. 2014. *Sistem Pendukung Keputusan Untuk Menentukan Posisi Pemain Dalam Olahraga Sepakbola Menggunakan Metode AHP*, Jurnal Ilmiah Go Infotech, Vol 20, No.2
- Turban, E., Aronson, J., & Liang, T. (2009). *Decision Support System and Intelligent Systems* (jilid I ed., Vol. 7). Yogyakarta: Andi Offset.

LAMPIRAN
LAMPIRAN A. USE CASE SKENARIO

A.1 Use Case Skenario Login

Tabel A. 1 Use Case Login

Nama Use Case	Login
Participating Actor	Admin dan user
Entry Condition	Halaman login
Exit Condition	Halaman Dashboard
Event Flow	
Normal Flow : menampilkan home	
Aktor	Reaksi Sistem
1. Memasukkan username dan password	
2. Kil Tombol login	
	3. Menampilkan pesan sukses dan menuju Halaman Dasboard
Alternative Flow : Username / password salah	
2. Klik tombol login	
	3. Menampilkan pesan "username/password salah"
Alternative Flow : Form kosong	
2. Klik tombol login	
	3. Menampilkan pesan "please fill out this field"

A.2 Use Case Skenario View Dashboard Admin

Tabel A. 2 Use Case Skenario View Dashboard Admin

Nama Use Case	View dashboard admin
Participating Actor	Admin
Entry Condition	Menampilkan halaman dashboard admin
Exit Condition	Menampilkan halaman login
Event Flow	
Normal Flow : menampilkan dashboard admin	
Aktor	Reaksi Sistem
1. Memasukkan username dan password Admin	
2. Klik Tombol login	
	3. Menampilkan Halaman Dasboard Admin
Alternative Flow : Username / password salah	
2. Klik tombol login	
	3. Menampilkan pesan "username/password salah"
Alternative Flow : Form kosong	
2. Klik tombol login	
	3. Menampilkan pesan "please fill out this field"

A.3 Use Case Skenario View Dashboard User

Tabel A. 3 Use Case Skenario View Dashboard User

Nama Use Case	View dashboard user
Participating Actor	user
Entry Condition	Menampilkan halaman dashboard user
Exit Condition	Menampilkan halaman login
Event Flow	
Normal Flow : menampilkan dashboard user	
Aktor	Reaksi Sistem
1. Memasukkan username dan password user	
2. Klik Tombol login	
	3. Menampilkan Halaman Dashboard user
Alternative Flow : Username / password salah	
2. Klik tombol login	
	3. Menampilkan pesan "username/password salah"
Alternative Flow : Form kosong	
2. Klik tombol login	
	3. Menampilkan pesan "please fill out this field"

A.4 Use Case Skenario Mengelola Biodata Pemain

Tabel A. 4 Use Case Skenario Mengelola Biodata Pemain

Nama Use Case	Mengelola biodata pemain
Participating Actor	Admin
Entry Condition	Daftar pemain
Exit Condition	Data telah selesai dikelola
Event Flow	
Normal Flow : memasukkan biodata pemain	
Aktor	Reaksi Sistem
1. Klik tombol player	
	2. Menampilkan data pemain
3. Klik tombol tambah	
	4. Menampilkan form tambah pemain
4. Mengisi biodata yang telah disediakan	
5. Klik tombol submit	
	6. Menampilkan pesan "sukses berhasil menambahkan data"
Alternative Flow : Username telah digunakan	
5. Klik Tombol Submit	
	6. Menampilkan pesan "Perhatian username telah digunakan"
Alternative Flow : Form Kosong	
5. Klik Tombol Submit	
	6. Menampilkan pesan "Please fill out this field"
Normal Flow : Ubah biodata pemain	
1. Klik tombol ubah	
	2. Menampilkan form ubah biodata pemain
3. Merubah data kejuaraan	
4. Klik tombol update	
	5. Menampilkan pesan "Sukses berhasil merubah data"
Alternative Flow : Form Kosong	
5. Klik Tombol update	
	6. Menampilkan pesan "Please fill out this field"

A.5 Use Case Skenario Mengelola Data Kejuaraan

Tabel A. 5 Use Case Skenario Mengelola Data Kejuaraan

Nama Use Case	Mengelola data kejuaraan
Participating Actor	Admin
Entry Condition	Daftar kejuaraan
Exit Condition	Data telah selesai dikelola
Event Flow	
Normal Flow : memasukkan data kejuaraan	
Aktor	Reaksi Sistem
1. Klik tombol tambah	
	2. Menampilkan pop up modal tambah kejuaraan
3. Mengisi form yang disediakan	
4. Klik Tombol Submit	
	5. Data berhasil ditambahkan
Alternative Flow : Form kosong	
4. Klik Tombol Submit	
	5. Menampilkan pesan "please fill out this field"
Normal Flow : Edit data kejuaraan	
1. Klik tombol edit	
	2. menampilkan form ubah data kejuaraan
3. merubah data kejuaraan	
4. klik tombol update	
	5. Data berhasil diubah
Alternative Flow : Form kosong	
4. klik tombol update	
	5. Menampilkan pesan "please fill out this field"

A.6 Use Case Skenario Mengelola Nilai Seleksi

Tabel A. 6 Use Case Skenario Mengelola Nilai Seleksi

Nama Use Case	Mengelola nilai seleksi
Participating Actor	Admin
Entry Condition	Daftar kejuaraan
Exit Condition	Nilai telah selesai dikelola
Event Flow	
Normal Flow : memasukkan nilai pemain	
Aktor	Reaksi Sistem
1.klik tombol info pada menu daftar kejuaraan	
	2. menampilkan halaman data pemain seleksi
3. klik tombol tambah	
	4. menampilkan menu tambah pemain seleksi
5. pilih pemain	
6. input nilai seleksi	
7. klik tombol submit	
	8. data nilai selesai di input
Alternative Flow : Form kosong	
7. klik tombol submit	
	8. Menampilkan pesan "please fill out this field"
Normal Flow : edit nilai pemain	
1.klik tombol info pada menu daftar kejuaraan	
	2. menampilkan halaman data pemain seleksi
3. klik tombol edit	
	4. menampilkan halaman form ubah nilai
5. input nilai seleksi	
6. klik tombol submit	
	7. berhasil ubah nilai seleksi
Alternative Flow : Form kosong	
6. klik tombol submit	
	7. Menampilkan pesan "please fill out this field"

A.7 Use Case Skenario Mengelola Nilai Bobot

Tabel A. 7 Use Case Skenario Mengelola Nilai Bobot

Nama Use Case	Mengelola nilai bobot
Participating Actor	Admin
Entry Condition	Daftar kriteria
Exit Condition	Data bobot selesai di kelola
Event Flow	
Normal Flow : Kelola nilai bobot	
Aktor	Reaksi Sistem
1.klik tombol menu kriteria	
	2. menampilkan data kriteria
3. klik tombol edit	
	4. menampilkan form ubah kriteria
5. edit nilai bobot	
6. klik tombol update	
	7. berhasil ubah data bobot
Alternative Flow : Form kosong	
6. klik tombol update	
	7. Menampilkan pesan "please fill out this field"
Alternative Flow : Total bobot lebih besar dari 100%	
6. klik tombol update	
	7. menampilkan pesan "perhatian total bobot harus = 100"

A.8 Use Case Skenario View Perhitungan SAW

Tabel A. 8 Use Case Skenario Mengelola Nilai Bobot

Nama Use Case	View Perhitungan SAW
Participating Actor	Admin
Entry Condition	Daftar kejuaraan
Exit Condition	Daftar kejuaraan
Event Flow	
Normal Flow : View perhitungan SAW	
Aktor	Reaksi Sistem
1. klik menu kejuaraan	
	2. menampilkan halaman daftar kejuaraan
3. klik tombol perhitungan	
	4. menampilkan halaman perhitungan seleksi
5. klik tombol Tahap 2	
	6. menampilkan halaman menentukan skor
7. klik tombol Tahap 3	
	8. menampilkan halaman menentukan nilai maksimal
9. klik tombol Tahap 4	
	10. menampilkan halaman menentukan normalisasi 1
11. klik tombol Tahap 5	
	12. menampilkan halaman menentukan normalisasi 2
13. klik tombol Tahap 6	
	14. menampilkan halaman peringkat

A.9 Use Case Skenario Kirim Undangan Seleksi

Tabel A. 9 Use Case Skenario Undangan Seleksi

Nama Use Case	Kirim Undangan seleksi
Participating Actor	Admin
Entry Condition	Login
Exit Condition	Daftar kejuaraan
Event Flow	
Normal Flow : Kirim undangan seleksi (Admin)	
Aktor	Reaksi Sistem
1. login sebagai admin	
	2. menampilkan halaman dashboard admin
3.klik tombol Kejuaraan	
	4. menampilkan halaman daftar kejuaraan
5. klik tombol info	
	6. menampilkan halaman data pemain seleksi
7. klik tombol undang pemain	
	8 menampilkan halaman undang pemain
9. klik tombol action undang	
	10. menampilkan pesan "SUKSES undangan telah dikirm"

A.10 Use Case Skenario Balas Undangan Seleksi

Tabel A. 10 Use Case Skenario Balas Undangan Seleksi

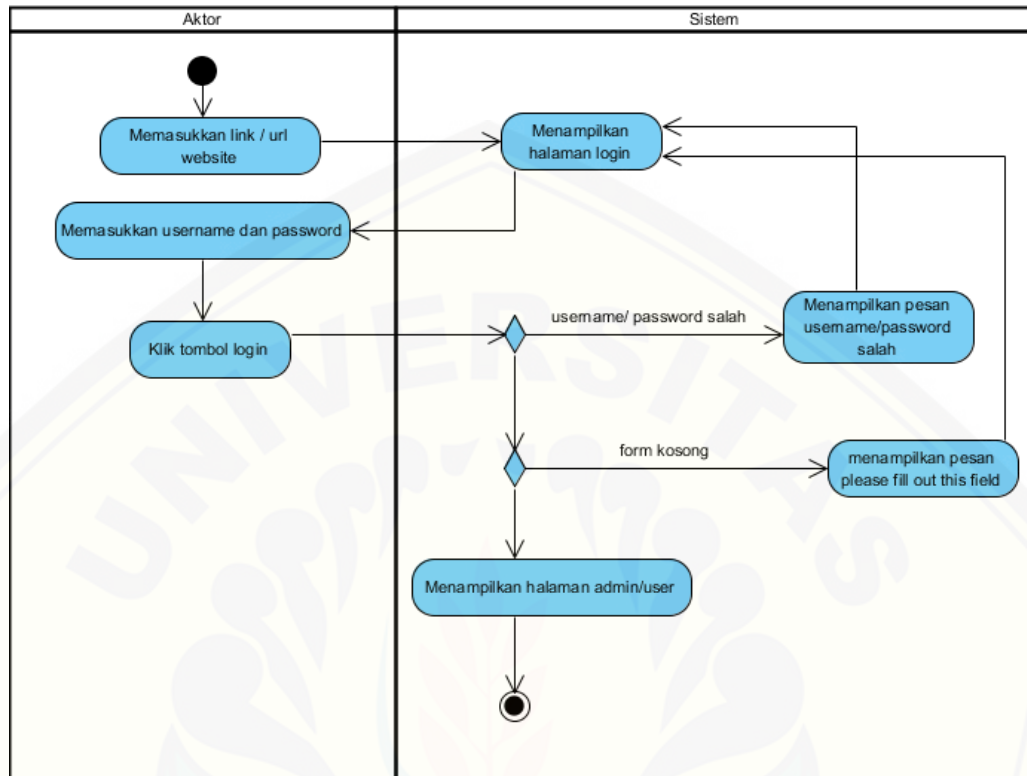
Nama Use Case	Balas Undangan Seleksi
Participating Actor	User
Entry Condition	Login
Exit Condition	Dashboard user
Event Flow	
Normal Flow : Terima undangan seleksi	
1. login sebagai user	
	2. menampilkan halaman dashboard user
3. klik tombol undangan	
	4. menampilkan halaman pesan masuk
5. klik action pesan	
	6. menampilkan isi pesan masuk
7. klik tombol terima	
	8. menampilkan pesan "SUKSES terima kasih atas partisipasinya"
Alternative Flow : Tolak undangan seleksi	
7. klik tombol tolak	
	8 menampilkan pesan "Perhatian Ok ditunggu partisipasinya"

A.11 *Use Case Skenario View Ranking*Tabel A. 11 *Use Case Skenario View Ranking*

Nama Use Case	View Ranking
Participating Actor	User
Entry Condition	Dashboard user
Exit Condition	Dashboard user
Event Flow	
Normal Flow : View Ranking	
Aktor	Reaksi Sistem
1.klik tombol Ranking	
	2. menampilkan halaman daftar seleksi yang diikuti

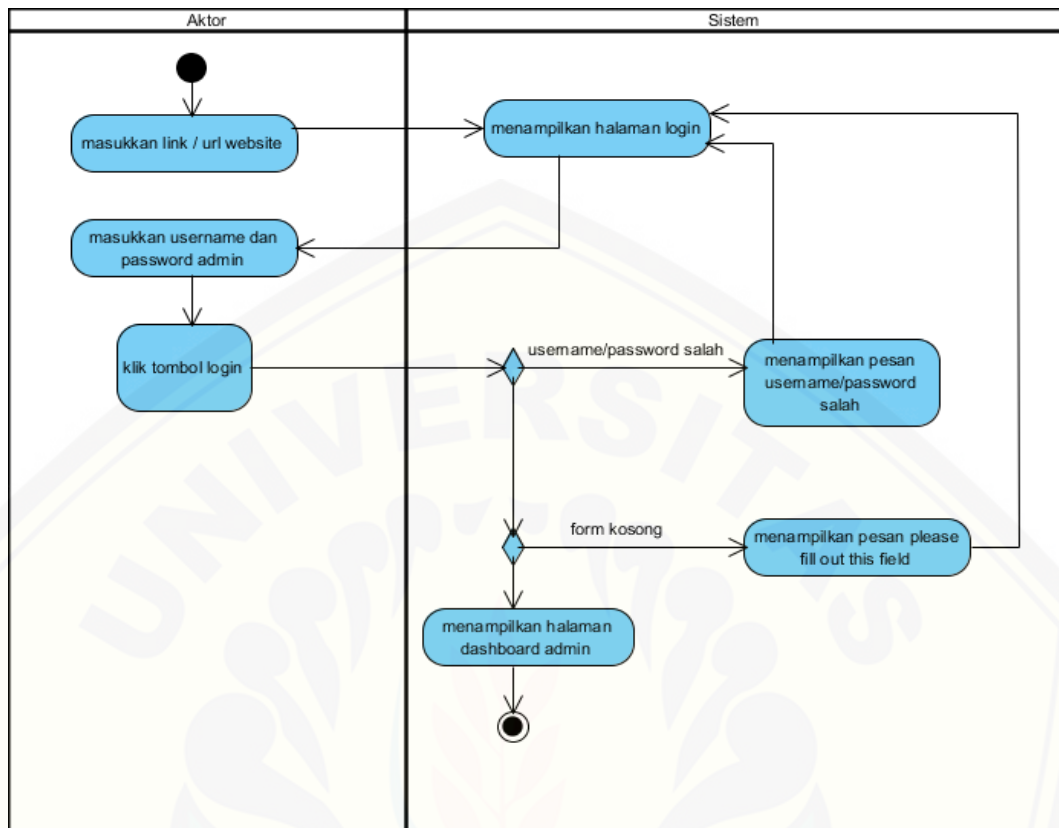
A.12 *Use Case Skenario Logout*Tabel A. 12 *Use Case Skenario Logout*

Nama Use Case	Logout
Participating Actor	Admin dan user
Entry Condition	Menampilkan halaman login
Exit Condition	Menampilkan halaman login
Event Flow	
Normal Flow : Logout	
Aktor	Reaksi Sistem
1.Klik tombol logout	
	2. menampilkan halaman login

LAMPIRAN B. ACTIVITY DIAGRAM**B.1 Activity Diagram Login**

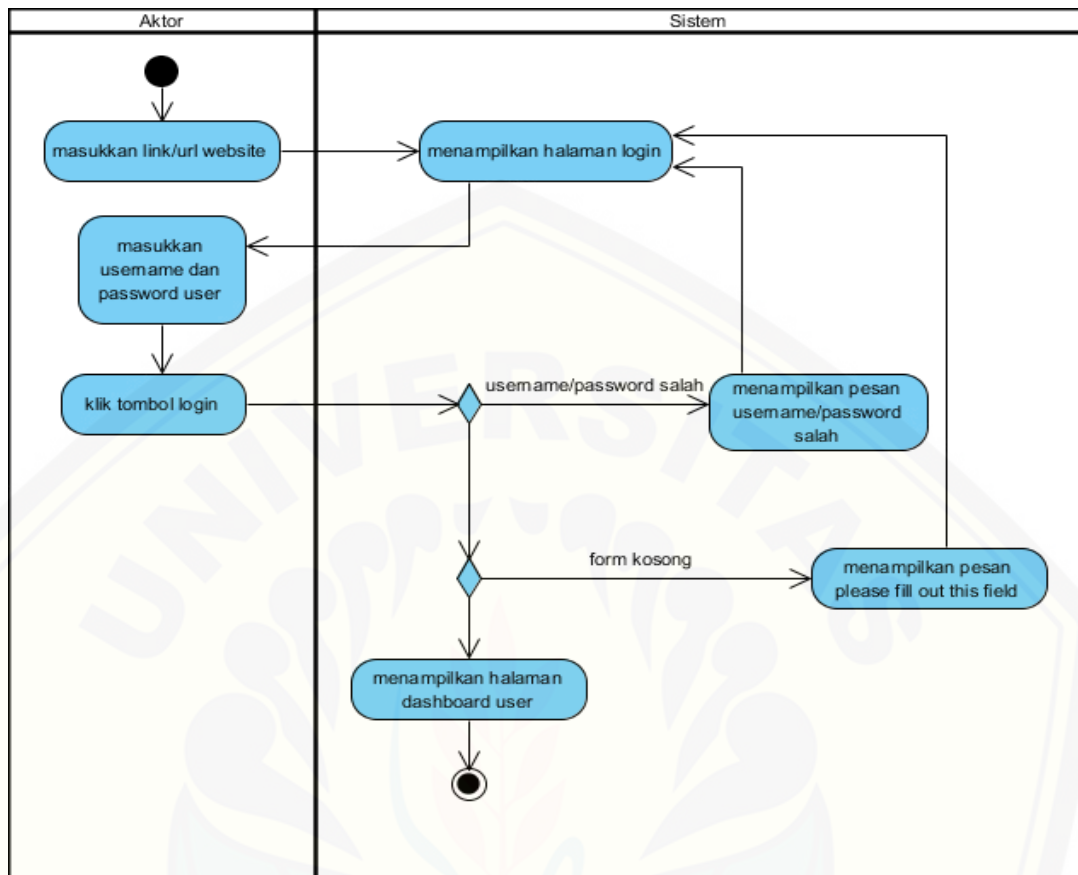
Gambar B. 1 Activity Diagram Login

B.2 Activity Diagram View Dashboard Admin



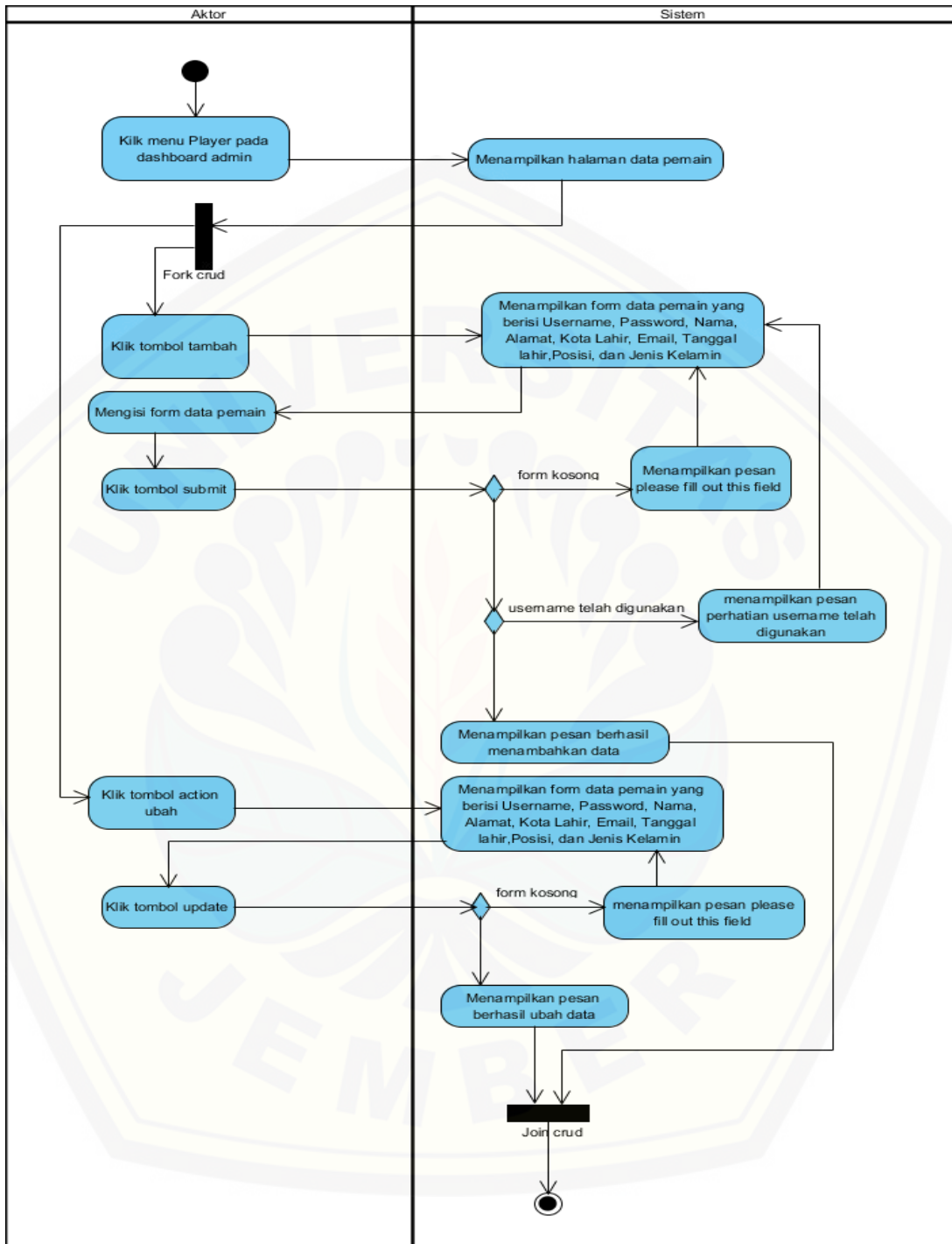
Gambar B. 2 Activity Diagram View Dashboard Admin

B.3 Activity Diagram View Dashboard User



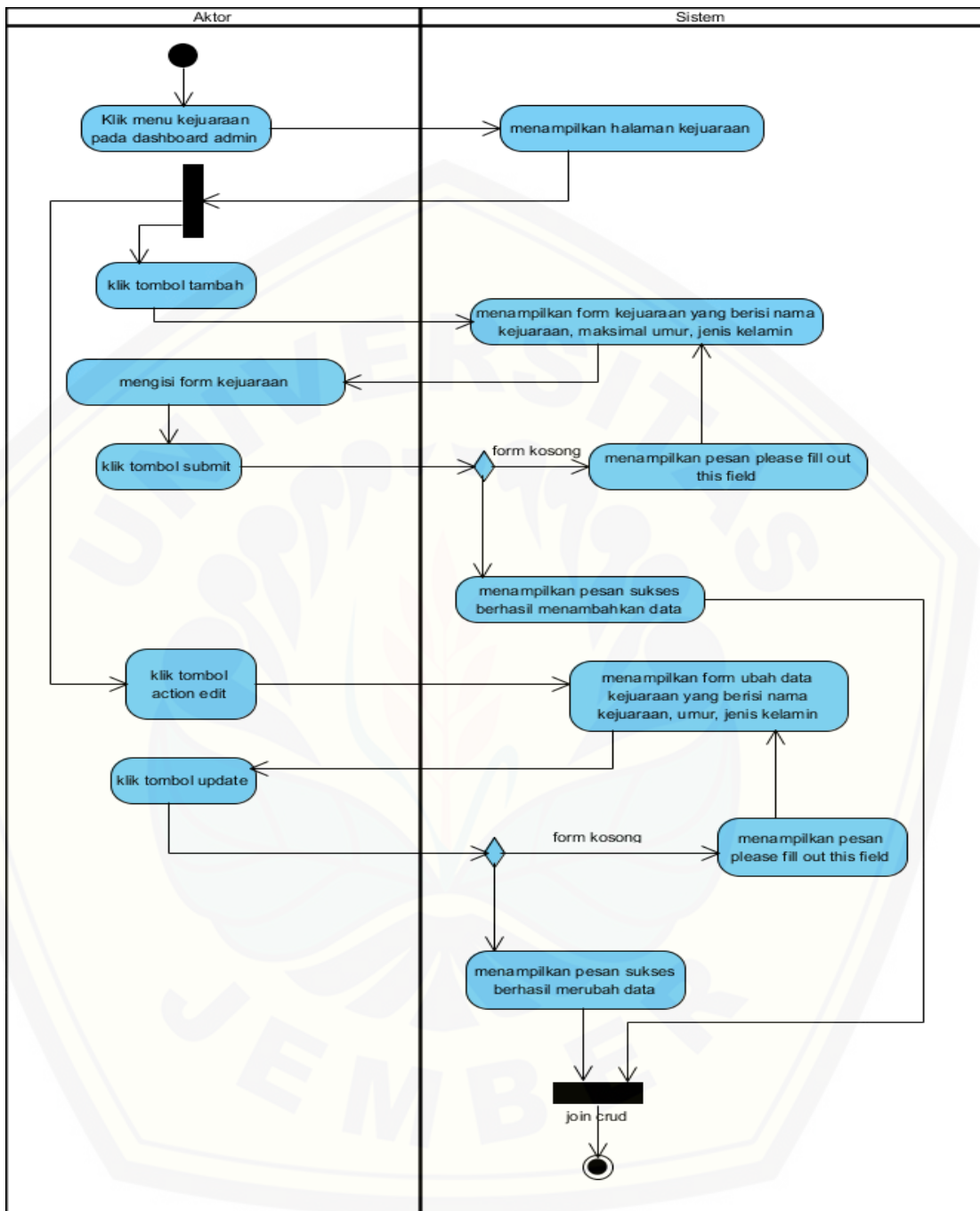
Gambar B. 3 Activity Diagram View Dashboard User

B.4 Activity Diagram Mengelola Biodata Pemain



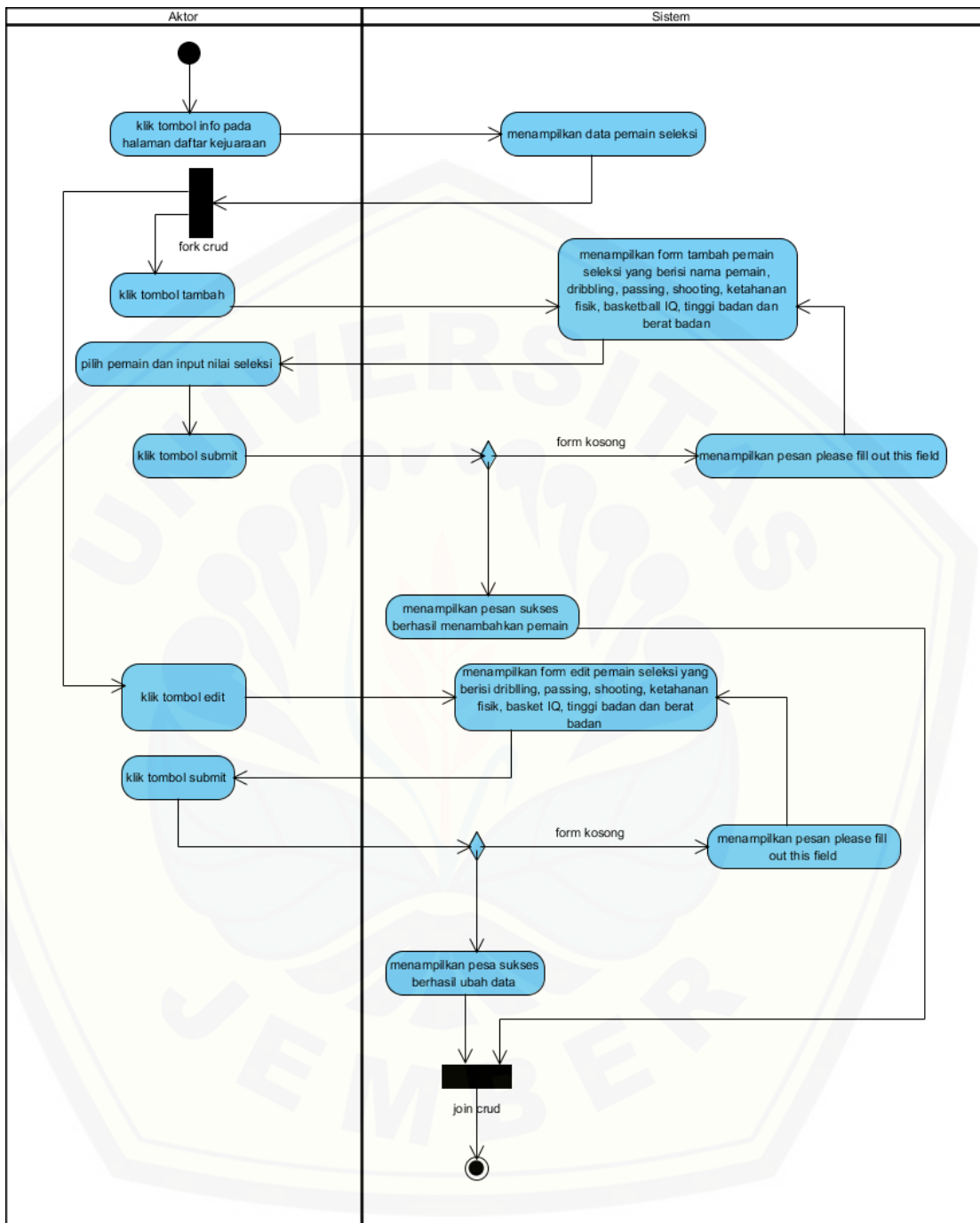
Gambar B. 4 Activity Diagram Mengelola Biodata Pemain

B.5 Activity Diagram Mengelola Data Kejuaraan



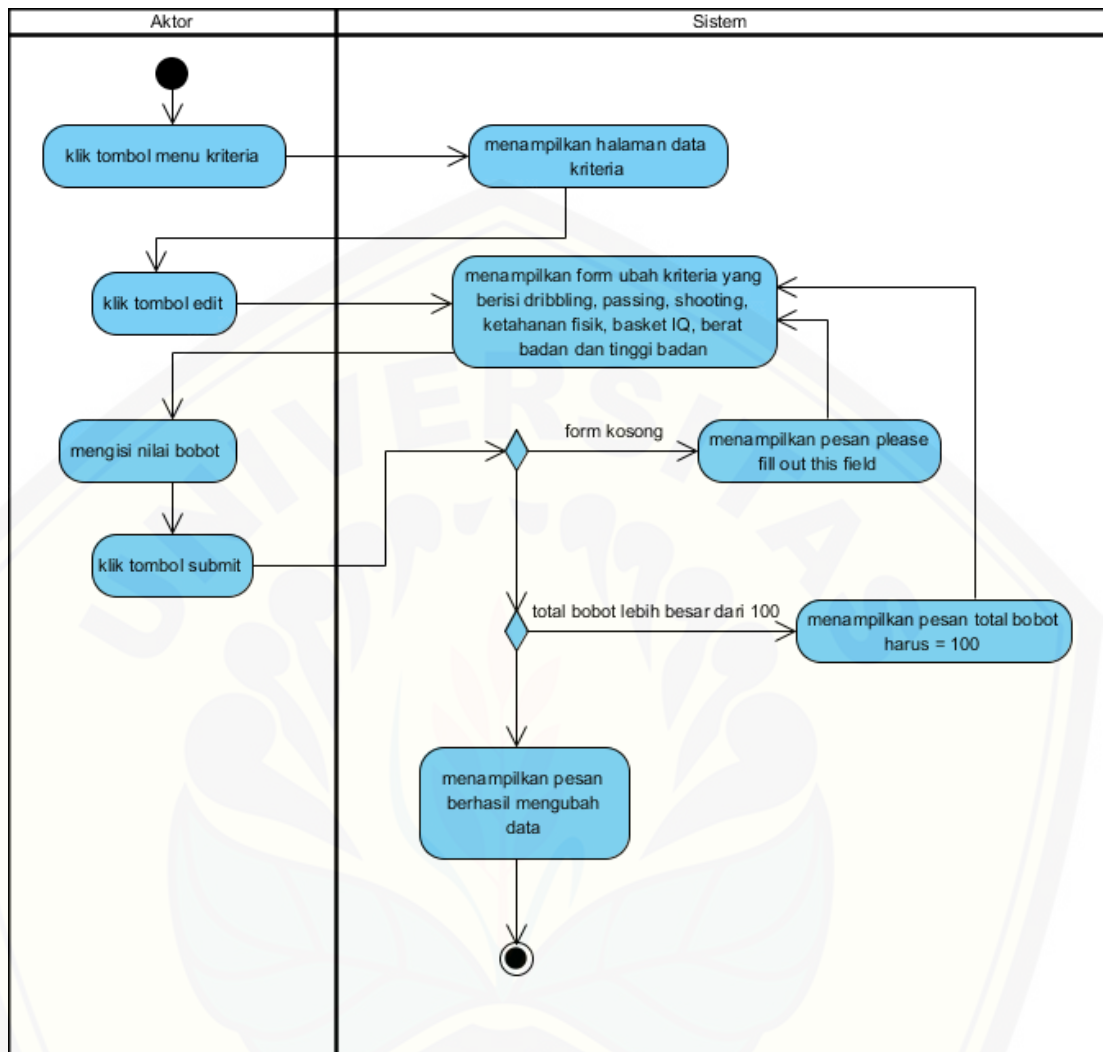
Gambar B. 5 Activity Diagram Data Kejuaraan

B.6 Activity Skenario Mengelola Nilai Seleksi



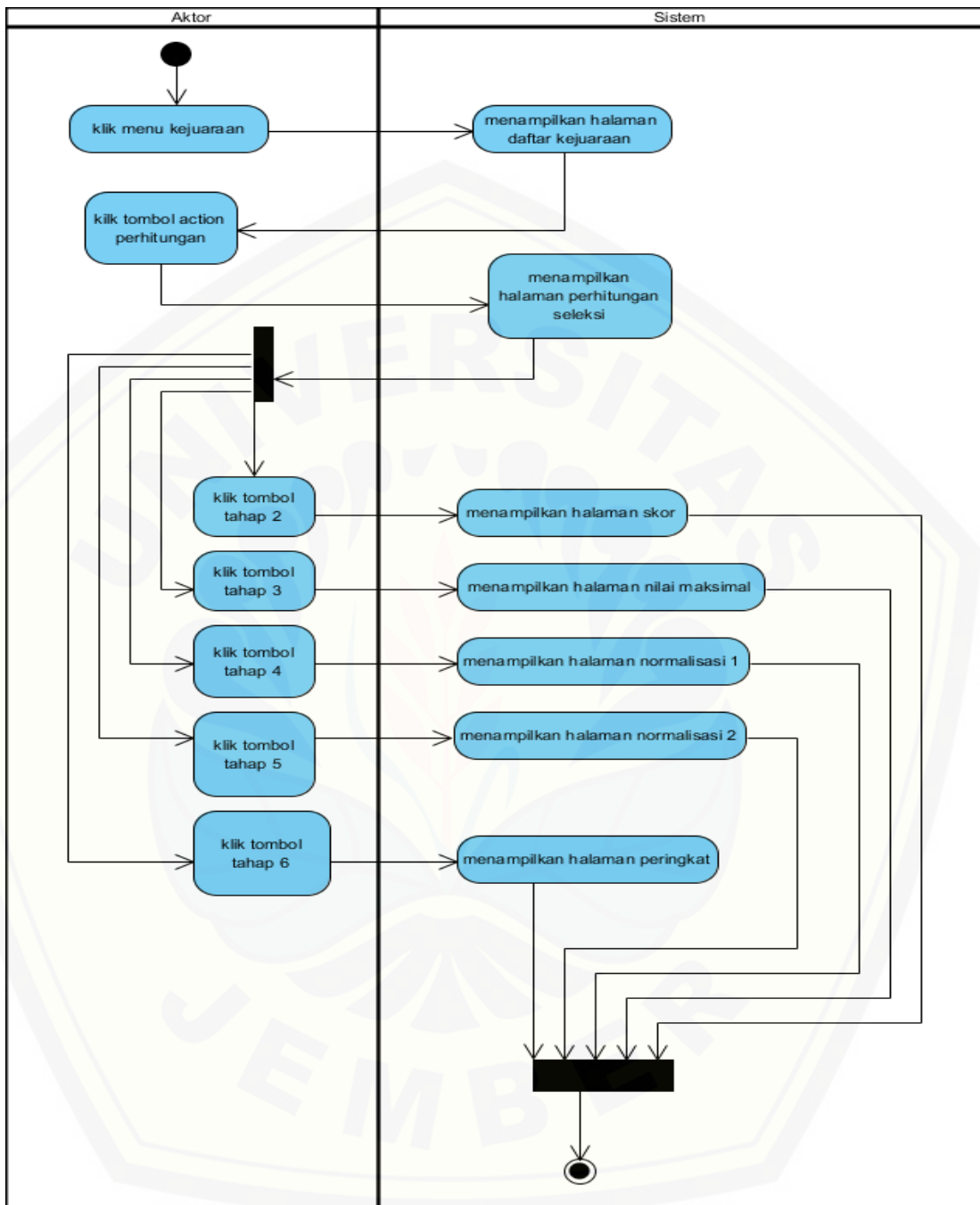
Gambar B. 6 Activity Diagram Mengelola Nilai Seleksi

B.7 Activity Skenario Mengelola Nilai Bobot



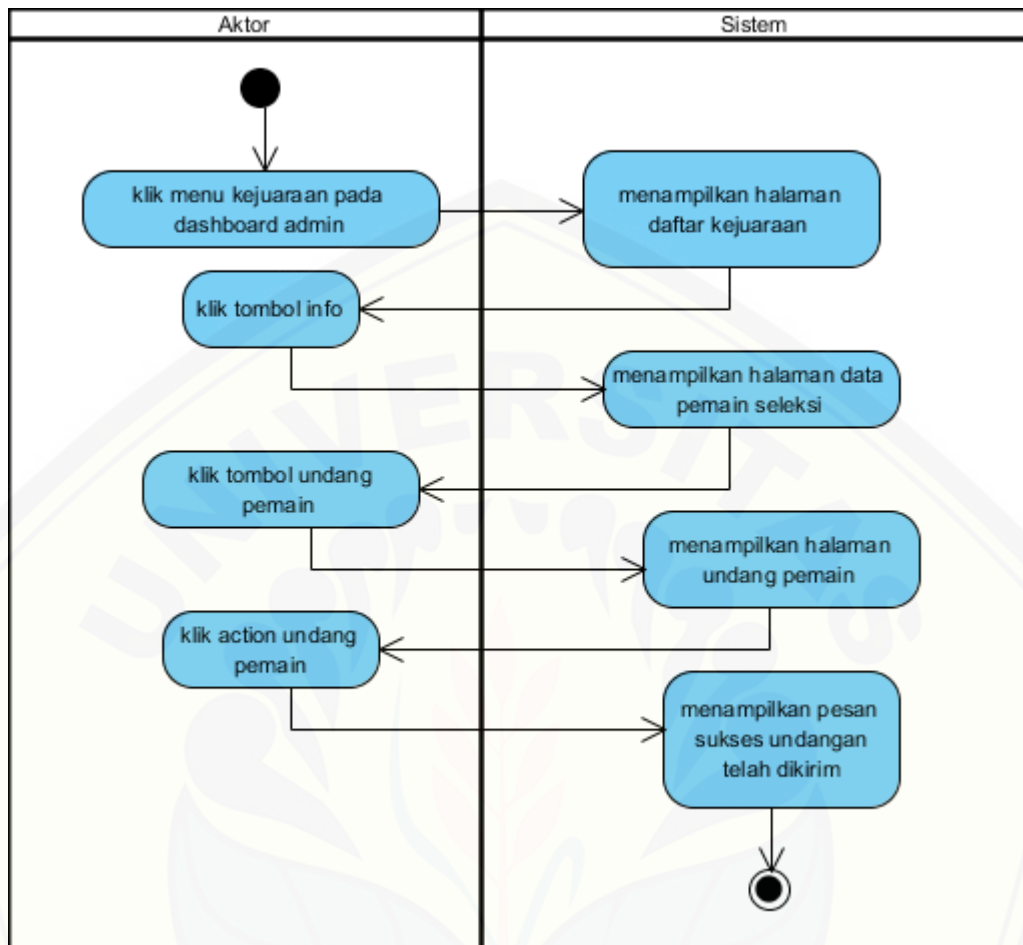
Gambar B. 7 Activity Diagram Mengelola Nilai Bobot

B.8 Activity Skenario View Perhitungan SAW



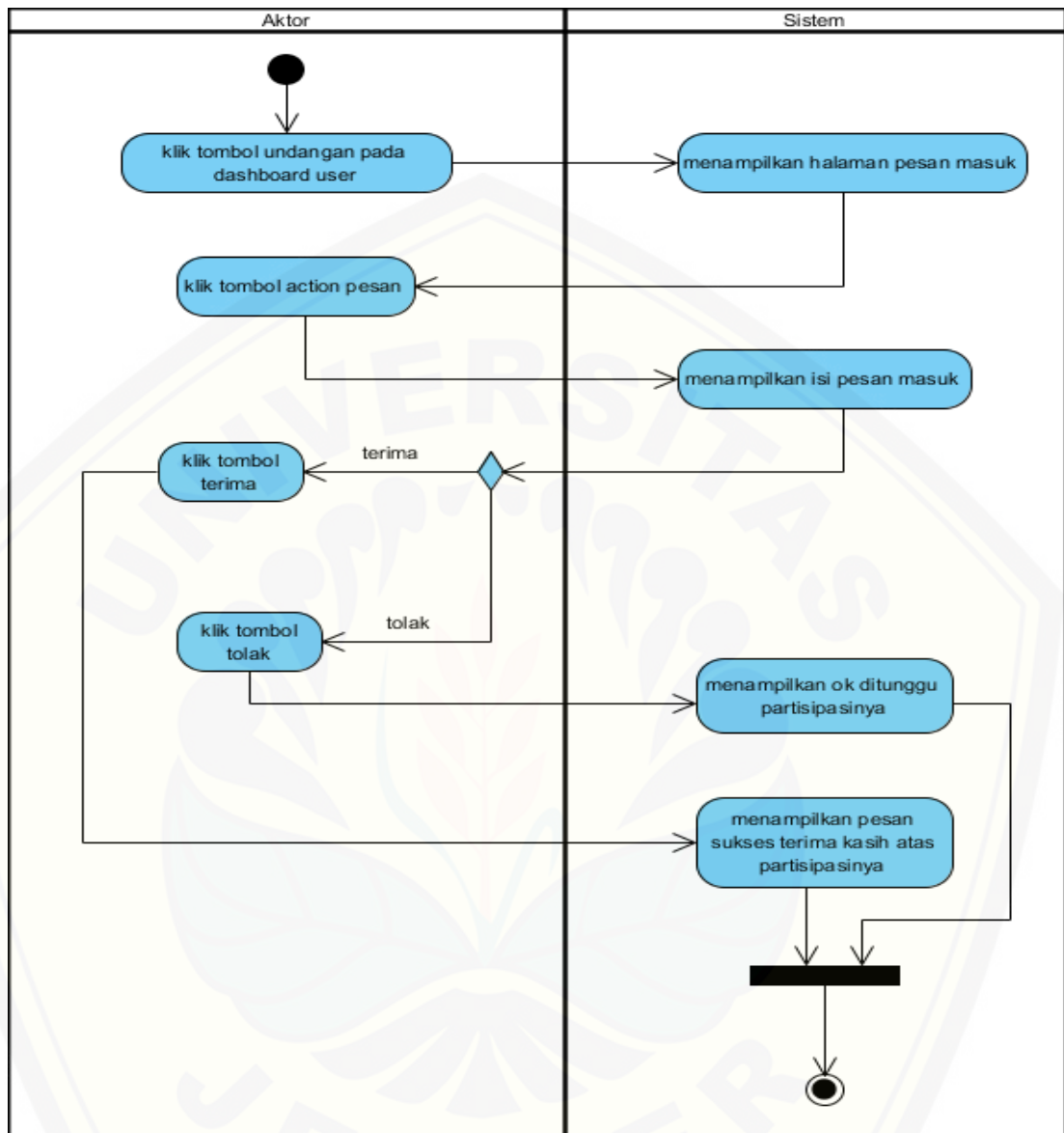
Gambar B. 8 Activity Diagram View Perhitungan SAW

B.9 Activity Skenario Kirim Undangan Seleksi



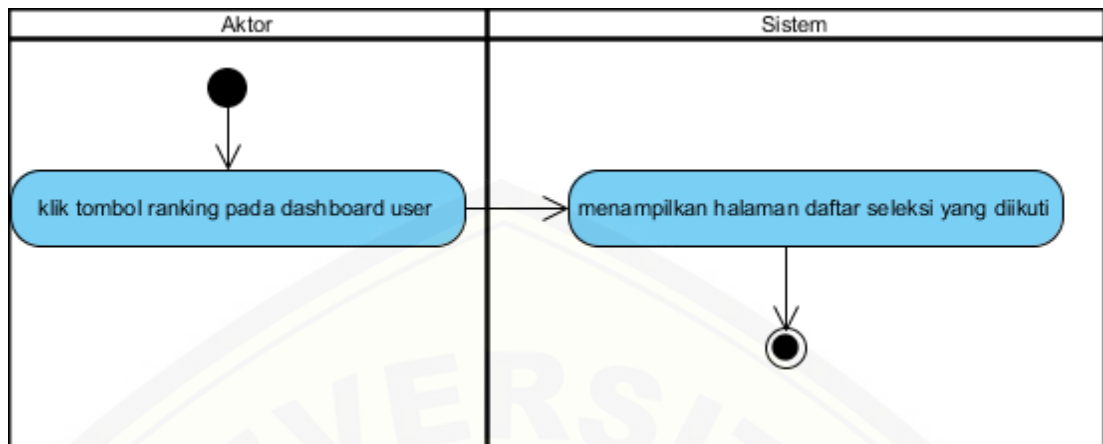
Gambar B. 9 Activity Diagram Kirim Undangan Seleksi

B.10 Activity Skenario Balas Undangan Seleksi



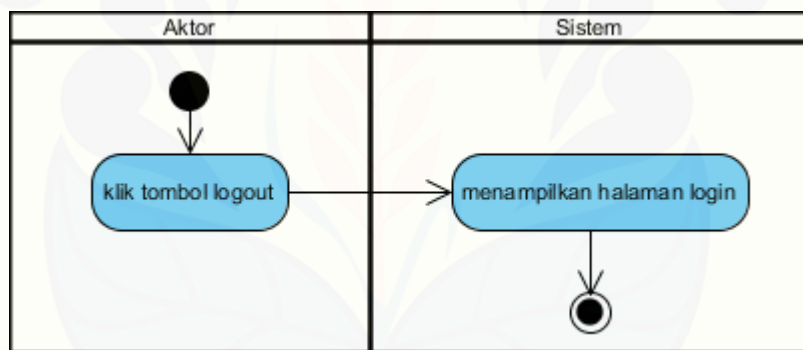
Gambar B. 10 Activity Diagram Balas Undangan Seleksi

B.11 Activity Skenario View Ranking



Gambar B. 11 Activity Diagram View Ranking

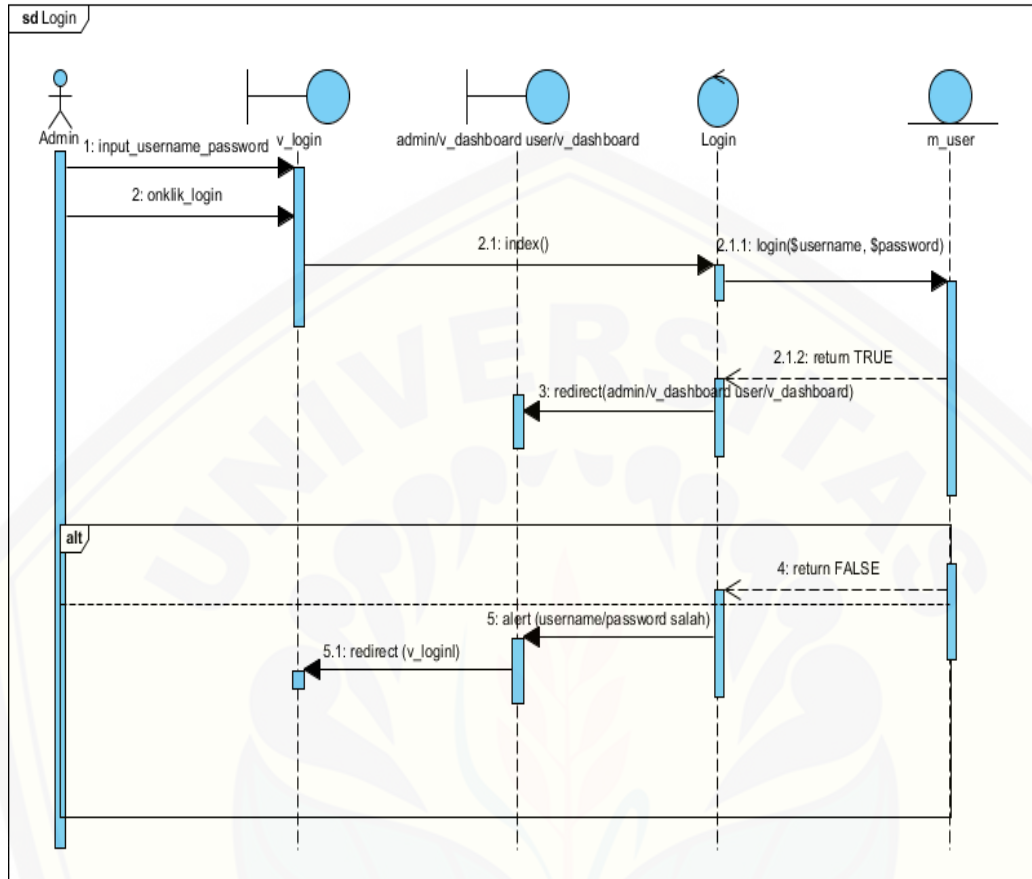
B.12 Activity Skenario Logout



Gambar B. 12 Activity Diagram Logout

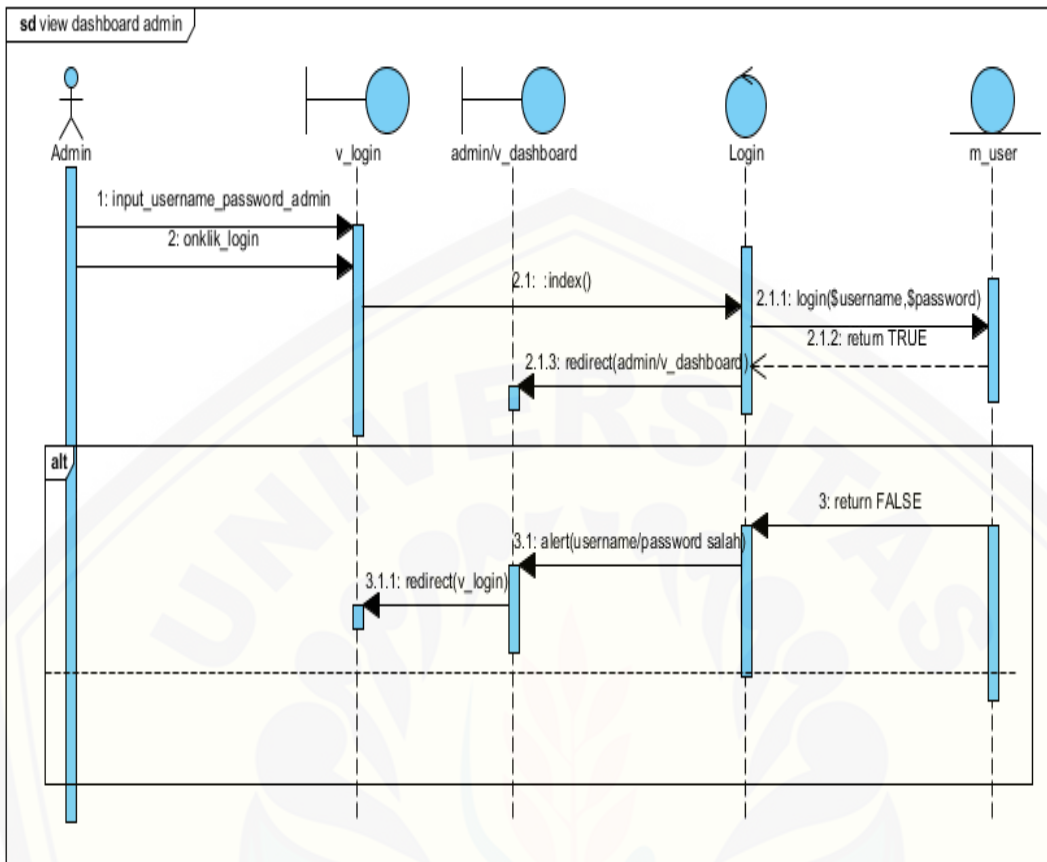
LAMPIRAN C. SEQUENCE DIAGRAM

C.1 Sequence Diagram Login



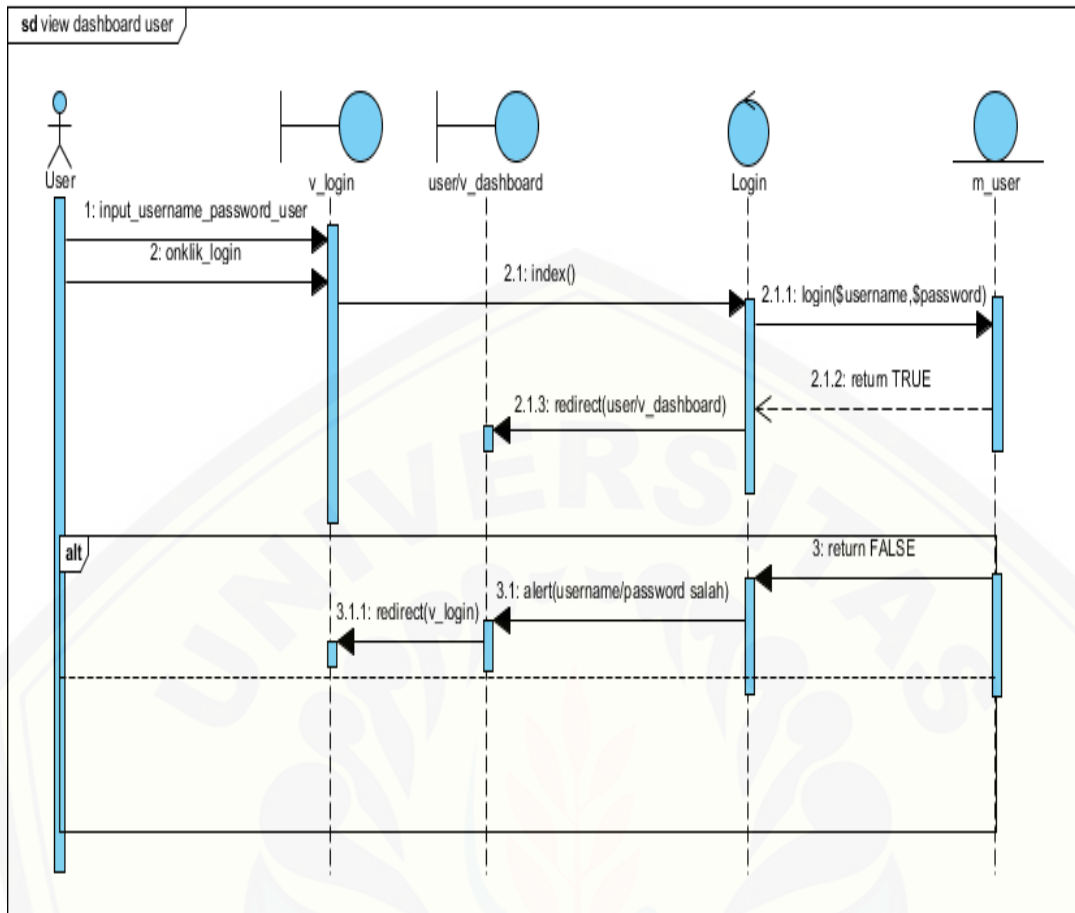
Gambar C. 1 Sequence Diagram Login

C.2 Sequence Diagram View Dashboard Admin



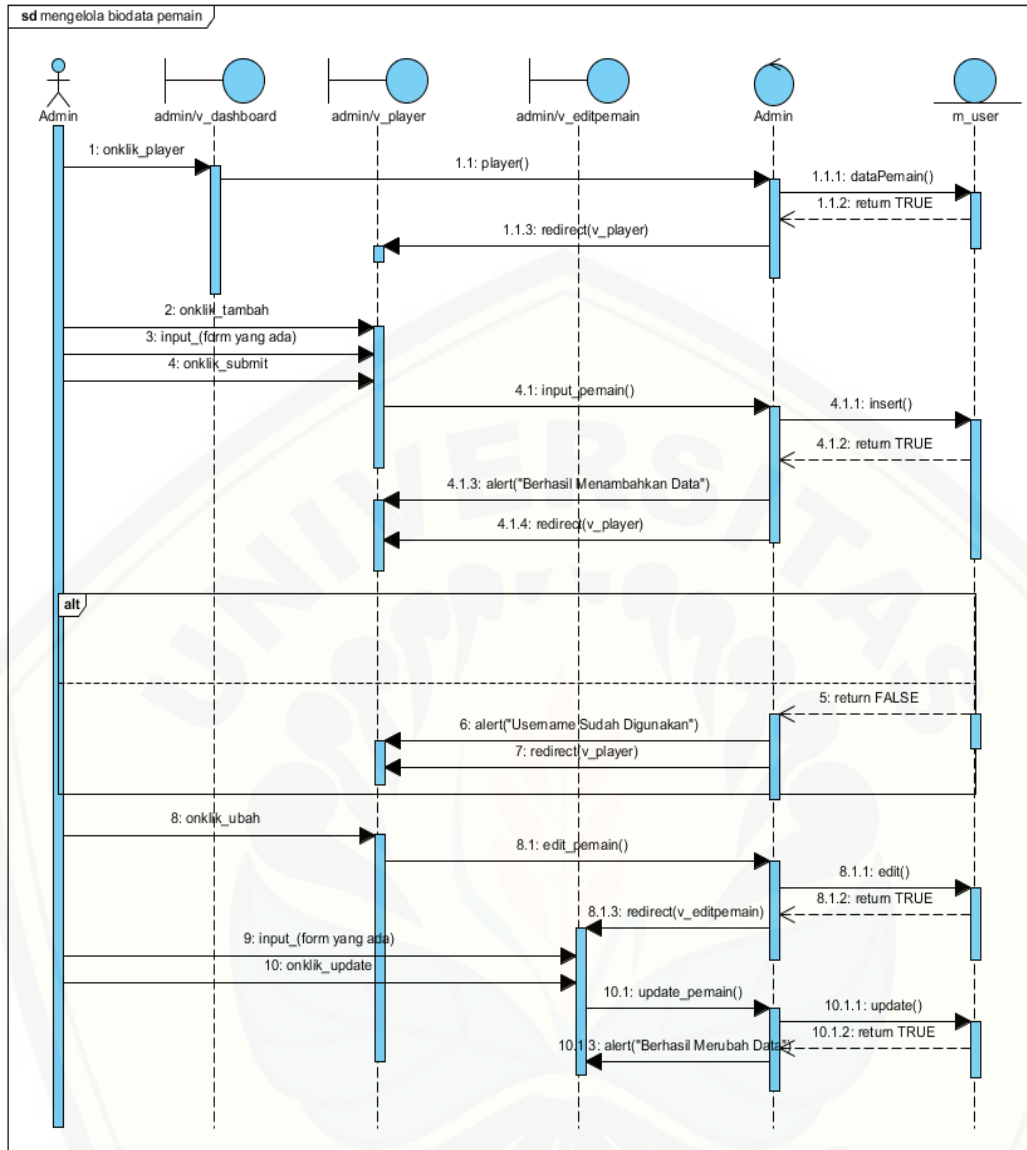
Gambar C. 2 Sequence Diagram View Dashboard Admin

C.3 Sequence Diagram View Dashboard User



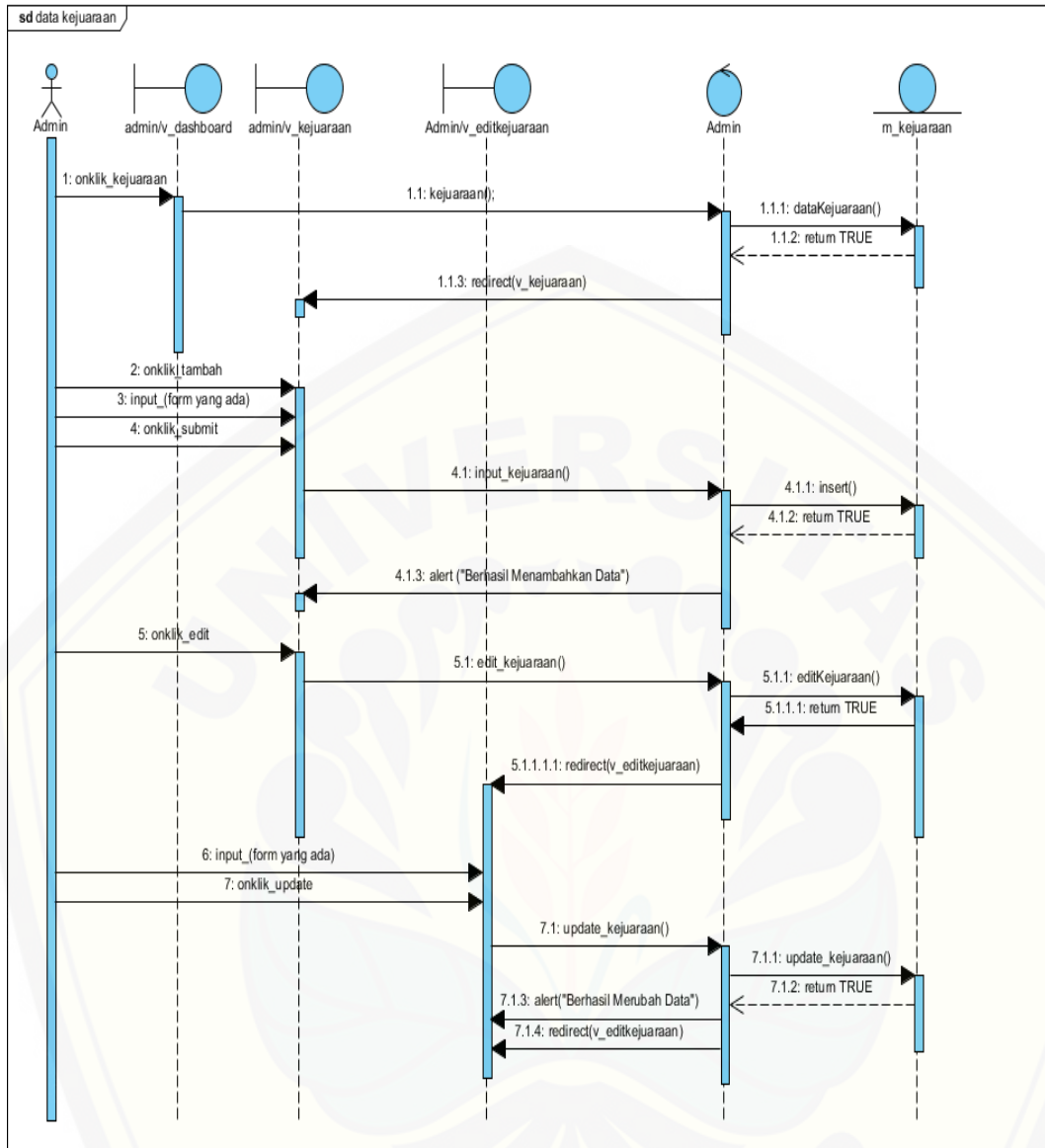
Gambar C. 3 Sequence Diagram View Dashboard User

C.4 Sequence Diagram Mengelola Biodata Pemain



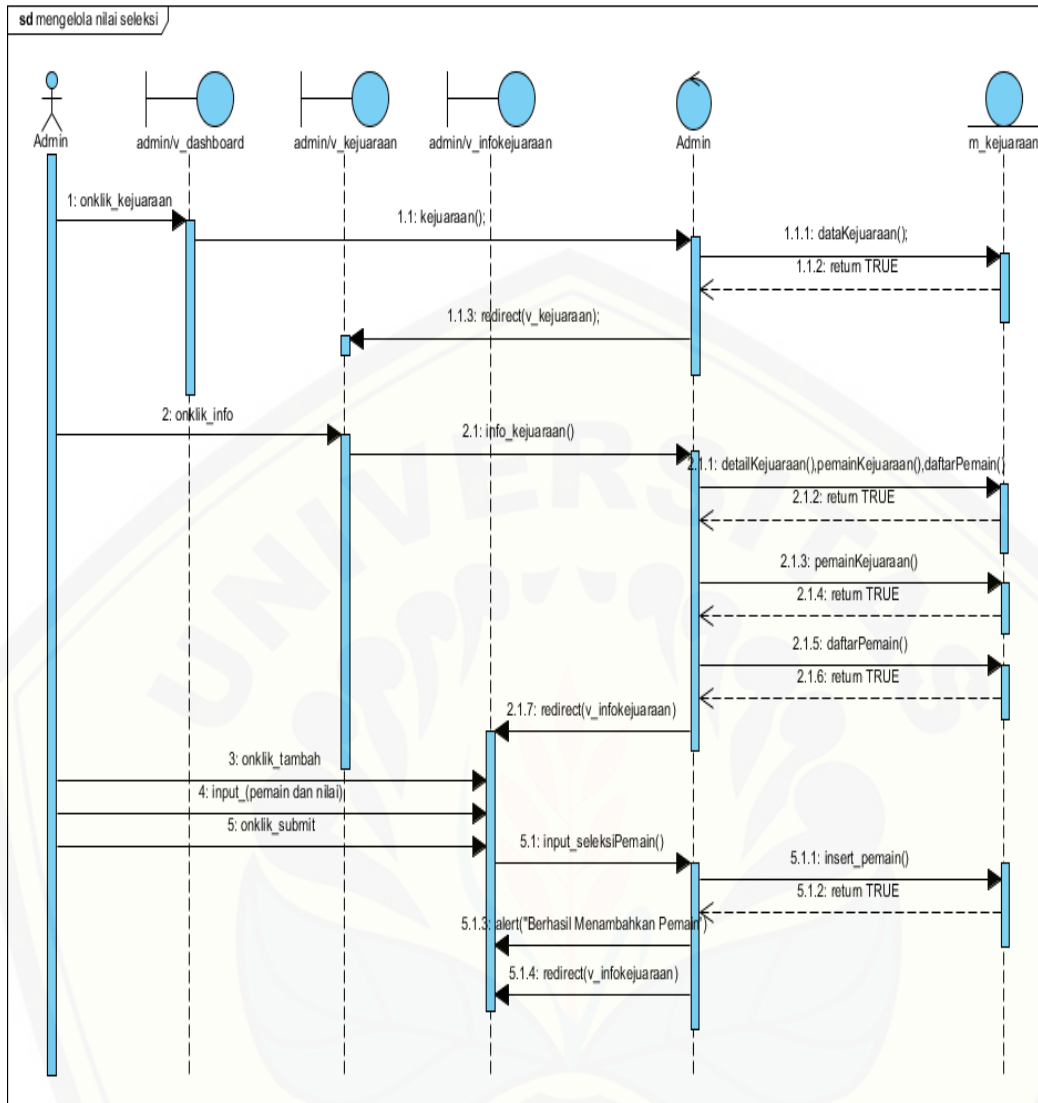
Gambar C. 4 Sequence Diagram Mengelola Biodata Pemain

C.5 Sequence Diagram Mengelola Data Kejuaraan



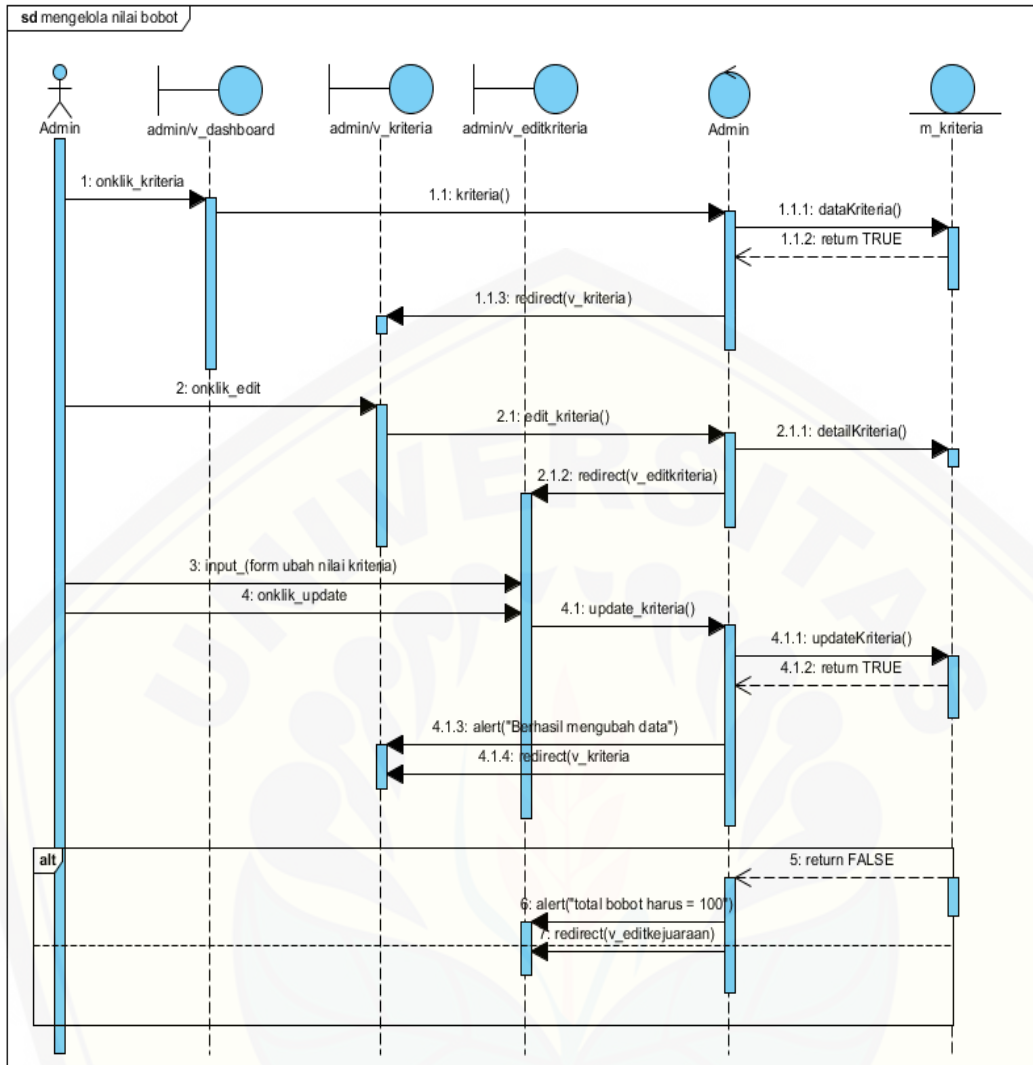
Gambar C. 5 Sequence Diagram Mengelola Data Kejuaraan

C.6 Sequence Skenario Mengelola Nilai Seleksi



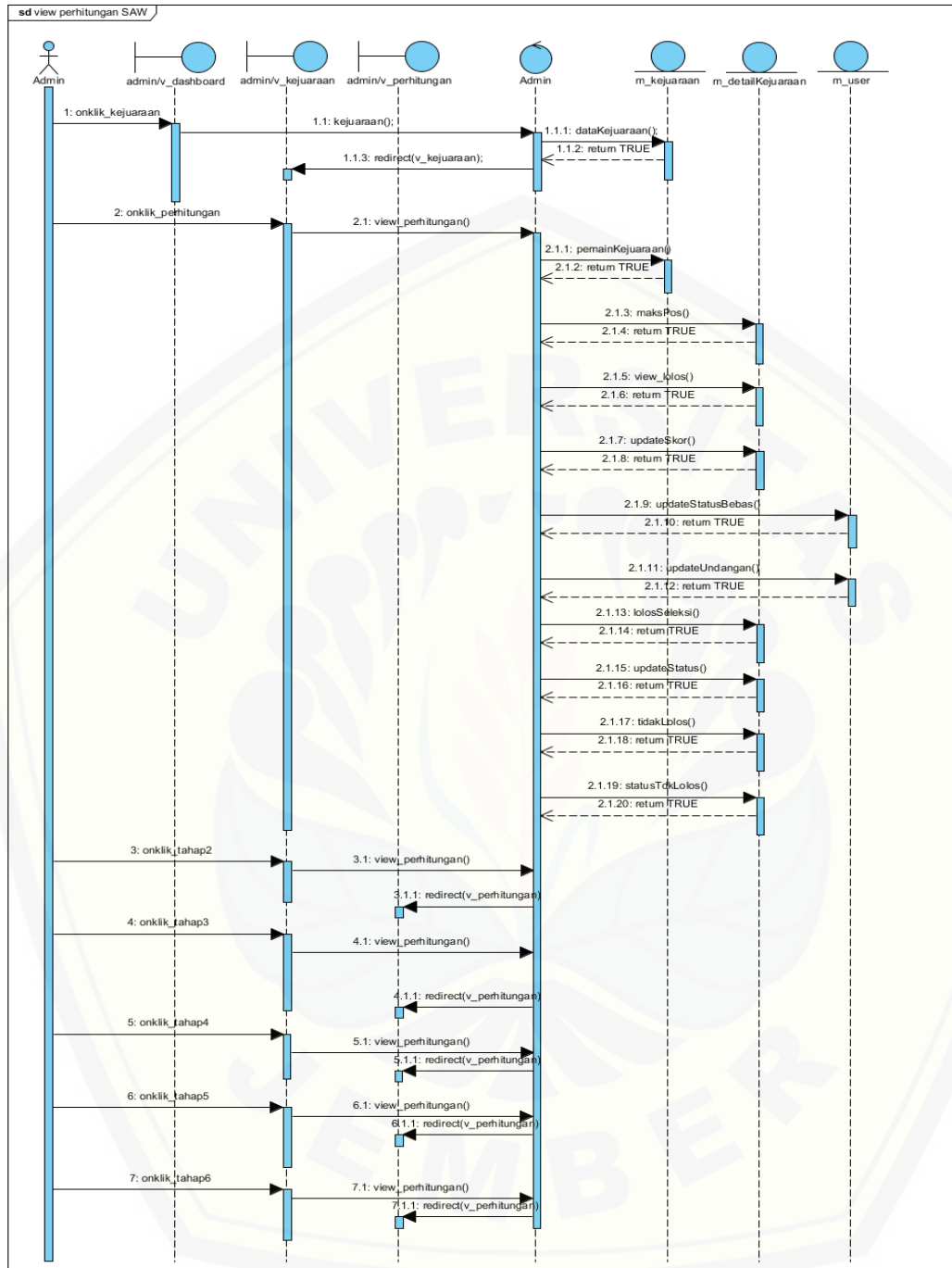
Gambar C. 6 Sequence Diagram Mengelola Nilai Seleksi

C.7 Sequence Skenario Mengelola Nilai Bobot



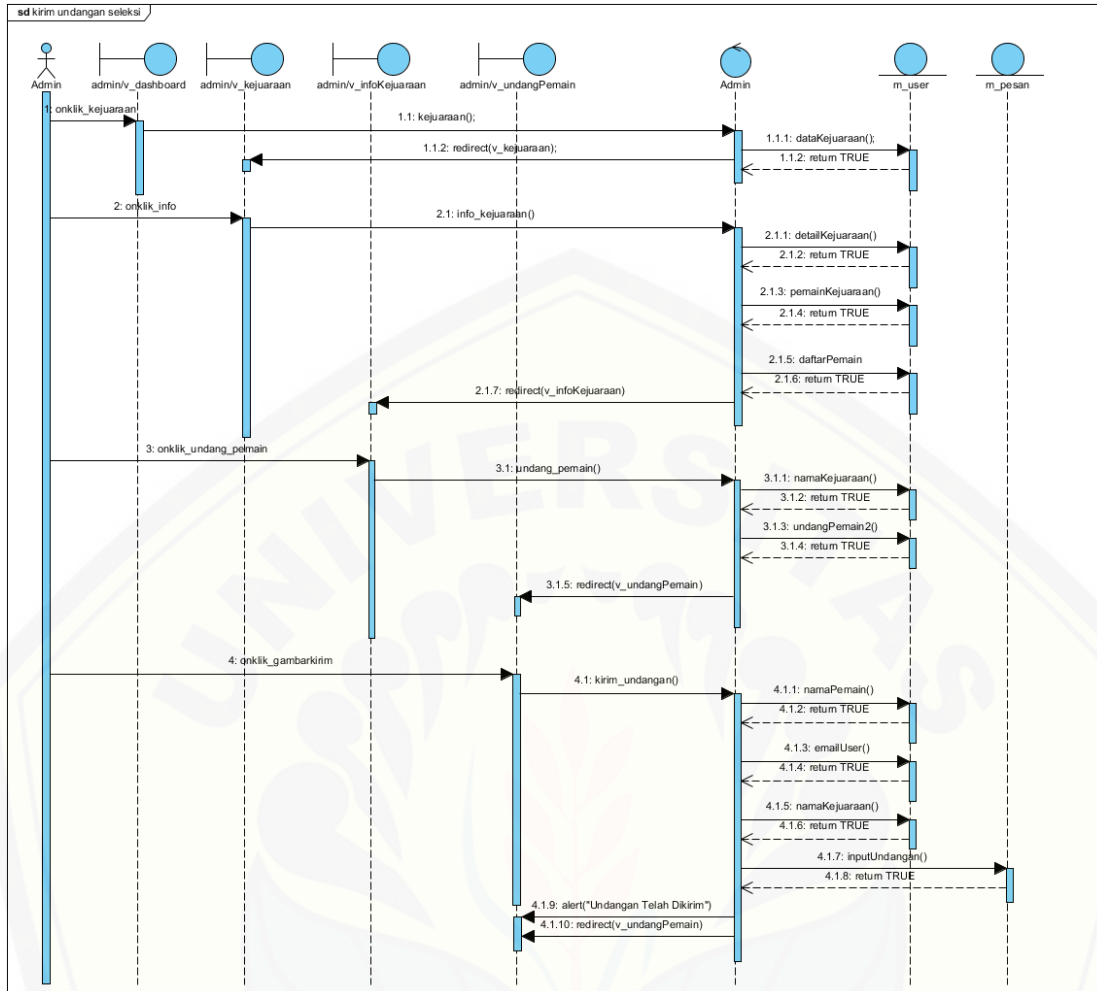
Gambar C. 7 Sequence Diagram Mengelola Nilai Bobot

C.8 Sequence Skenario View Perhitungan SAW



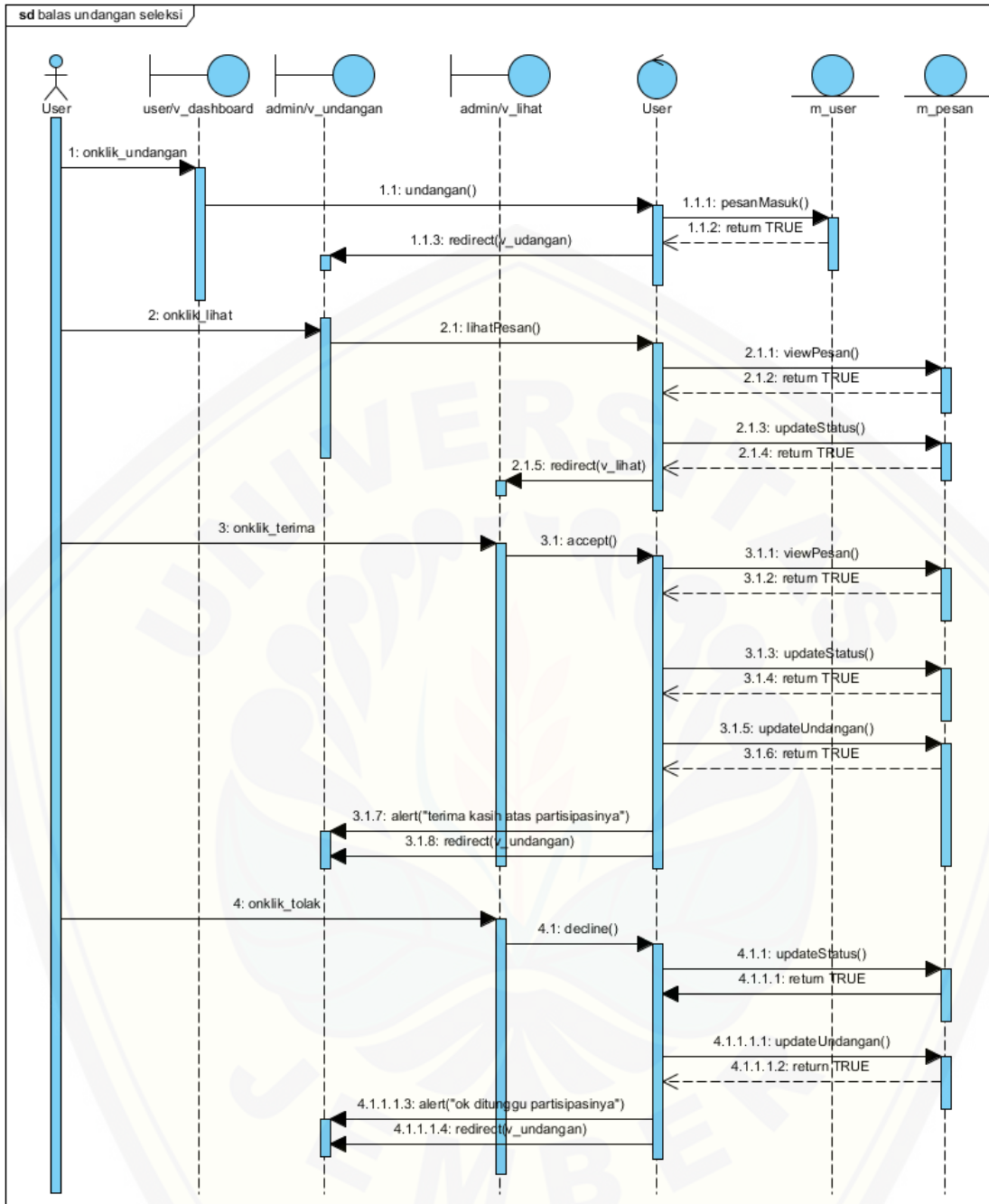
Gambar C. 8 Sequence Diagram View Perhitungan SAW

C.9 Sequence Skenario Kirim Undangan Seleksi



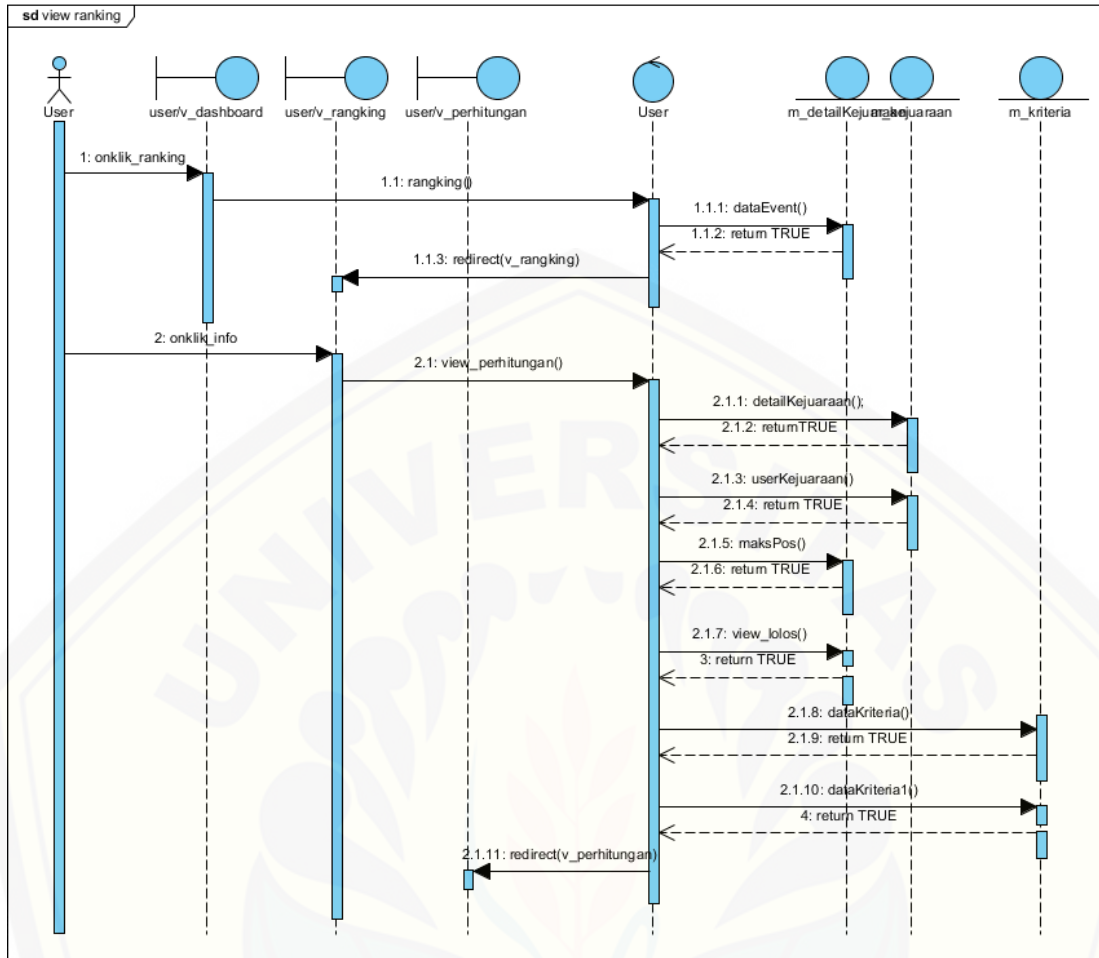
Gambar C. 9 Sequence Diagram Kirim Undangan Seleksi

C.10 Sequence Skenario Balas Undangan Seleksi



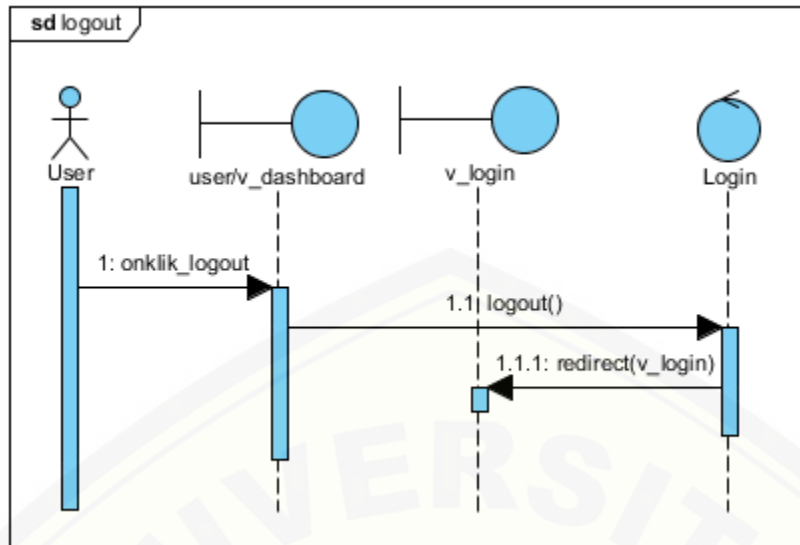
Gambar C. 10 Sequence Diagram Balas Undangan Seleksi

C.11 Sequence Skenario View Ranking



Gambar C. 11 Sequence Diagram View Ranking

C.12 Sequence Skenario Logout



Gambar C. 12 Sequence Diagram Logout

LAMPIRAN D. KODE PROGRAMD.1 Kode Program *class controllers/Admin*

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Admin extends CI_Controller {

/**
 * Index Page for this controller.
 *
 * Maps to the following URL
 *      http://example.com/index.php/welcome
 *  - or -
 *      http://example.com/index.php/welcome/index
 *  - or -
 * Since this controller is set as the default controller in
 * config/routes.php, it's displayed at http://example.com/
 *
 * So any other public methods not prefixed with an underscore will
 * map to /index.php/welcome/<method_name>
 * @see https://codeigniter.com/user_guide/general/urls.html
 */

public function __construct()
{
    parent::__construct();
    $this->load->library('session');
    $this->sess = $this->session->userdata('admin');
    $this->load->model('m_user','user');
    $this->load->model('m_kejuaraan','kejuaraan');
    $this->load->model('m_detailKejuaraan','dKejuaraan');
    $this->load->model('m_kriteria','kriteria');
    $this->load->model('m_pesan','pesan');
}

public function index() //dashboard
{
    $data['pemain'] = $this->user->dataPemain();
    $this->load->view('admin/v_dashboard');
}
}
```

Dilanjutkan

Lanjutan

```
public function player(){
    $data['pemain'] = $this->user->dataPemain();
    $this->load->view('admin/v_player',$data);
}

public function kejuaraan(){
    $data['kejuaraan'] = $this->kejuaraan->dataKejuaraan();

    // dd($data);
    $this->load->view('admin/v_kejuaraan',$data);
}

public function kriteria(){
    $data['kriteria'] = $this->kriteria->dataKriteria();
    $this->load->view('admin/v_kriteria',$data);
}

public function edit_kriteria($id_kriteria){
    $data['detailKriteria'] = $this->kriteria->detailKriteria($id_kriteria)[0];
    $this->load->view('admin/v_editkriteria',$data);
}

public function update_kriteria(){
    $total = $_POST['k_dribbling'] + $_POST['k_passing'] +
    $_POST['k_shooting'] + $_POST['k_ketahanan_fisik'] + $_POST['k_basket_iq'] +
    $_POST['k_tinggi_badan'] + $_POST['k_berat_badan'];
    if ($total == 100) {
        $id_kriteria = $_POST['id_kriteria'];
        $data['k_dribbling'] = $_POST['k_dribbling']/100;
        $data['k_passing'] = $_POST['k_passing']/100;
        $data['k_shooting'] = $_POST['k_shooting']/100;
        $data['k_ketahanan_fisik'] = $_POST['k_ketahanan_fisik']/100;
        $data['k_basket_iq'] = $_POST['k_basket_iq']/100;
        $data['k_tinggi_badan'] = $_POST['k_tinggi_badan']/100;
        $data['k_berat_badan'] = $_POST['k_berat_badan']/100;
        $update = $this->kriteria->updateKriteria($id_kriteria,$data);
        pesan('sukses', 'Berhasil mengubah data');
        redirect(base_url() . 'Admin/kriteria');
    }else{
        $id_kriteria = $_POST['id_kriteria'];
        pesan('gagal', 'Total bobot harus = 100');
        redirect(base_url() . 'Admin/edit_kriteria/'.$id_kriteria);
    }
}
```

Dilanjutkan

```
Lanjutan
}

public function input_pemain(){
    $data = $this->input->post();
    $data['level'] = 1;
    $cekusername = $this->user->cekusername($data['username'])[0]['jumlah'];
    //cekusernaem sudah ada atau belum

    if ($cekusername == 0) {
        $insert = $this->user->insert($data);
        pesan('sukses', 'Berhasil menambahkan data');
        redirect(base_url() . 'Admin/player');
    }else{
        pesan('gagal', 'username sudah digunakan!!!!');
        redirect(base_url() . 'Admin/player');
    }
}

public function edit_pemain($id_user){
    $data['edit'] = $this->user->edit($id_user)[0];
    $this->load->view('admin/v_editpemain',$data);
}

public function update_pemain(){
    $data = $this->input->post();
    $id = $data['id_user'];
    $result = $this->user->update($data, $id);
    pesan('sukses', 'Berhasil merubah data');
    redirect(base_url() . 'Admin/player');
}

public function input_kejuaraan(){
    $data = $this->input->post();
    pesan('sukses', 'Sukses menambahkan data');
    $insert = $this->kejuaraan->insert($data);
    pesan('sukses', 'Berhasil menambahkan data');
    redirect(base_url() . 'Admin/kejuaraan');
}
```

Dilanjutkan

Lanjutan

```
public function info_kejuaraan($id_kejuaraan){
    $idkejuaraan = $id_kejuaraan;
    $data['kejuaraan'] = $this->kejuaraan->detailKejuaraan($idkejuaraan);
    $this->session->
>set_userdata('id_kejuaraan',$data['kejuaraan'][0]['id_kejuaraan']);
    $id = $this->session->userdata('id_kejuaraan');
    // dd($id);
    $maksumur = $data['kejuaraan'][0]['maks_umur'];
    $data['pemain'] = $this->kejuaraan->pemainKejuaraan($idkejuaraan);
    $data['daftar'] = $this->kejuaraan->
>daftarPemain($maksumur,$idkejuaraan);
    $data['idkejuaraan'] = $id_kejuaraan;
    // dd($data);
    $this->load->view('admin/v_infoKejuaraan',$data);
}

public function undang_pemain($id_kejuaraan){
    $session = array(
        'id_kejuaraan' => $id_kejuaraan);
    $data['id'] = $id_kejuaraan;
    $this->session->set_userdata('admin', $session);
    $data['kejuaraan'] = $this->kejuaraan->namaKejuaraan($id_kejuaraan);
    // dd($data);
    // dd($data);
    $data['pemain'] = $this->user->
>undangPemain2($data['kejuaraan'][0]['maks_umur'],$id_kejuaraan,$data['kejuar
aan'][0]['gender']);
    $this->load->view('admin/v_undangPemain',$data);
}

public function kirim_undangan($id){
    $id_kejuaraan = $this->sess['id_kejuaraan'];
    $nama = $this->user->namaPemain($id)[0]['nama'];
    $username = $this->user->namaPemain($id)[0]['username'];
    $password = $this->user->namaPemain($id)[0]['password'];

    $email = $this->user->emailUser($id)[0]['email'];
    $namaKejuaraan = $this->kejuaraan->namaKejuaraan($id_kejuaraan)[0];
    // dd($namaKejuaraan);
}
```

Dilanjutkan

Lanjutan

```

$pesan = 'Hai '.$nama.' anda telah diundang PERBASI JEMBER, untuk mengikuti
seleksi Bola Basket guna mengikuti Event '.$namaKejuaraan['nama_kejuaraan'].'.
Mohon segera dikonfirmasi anda bersedia atau tidak mengikuti seleksi. Terima
Kasih. TTD Ketua Perbasis Jember. NB : Pesan ini tidak perlu dibalas. Silahkan
klik link berikut untuk menerima undangan. Silahkan login menggunakan <br>
Username : <b>'.$username.'</b> <br> Password <b> : '.$password.' </b> <br>
<a href="http://localhost/latihan">Klik disini untuk login</a>;
    $insert = $this->pesan->inputUndangan($id,$id_kejuaraan,$pesan);
    send_email($email,'Admin Perbasi Jember',$pesan);
    // $update = $this->user->updateUndangan($id,$id_kejuaraan);
    pesan('sukses', 'Undangan Telah Dikirim');
    redirect(base_url(). 'Admin/undang_pemain/'.$id_kejuaraan);
}

public function edit_kejuaraan($id_kejuaraan){
    $data['edit'] = $this->kejuaraan->editKejuaraan($id_kejuaraan);
    // dd($data);
    $this->load->view('admin/v_editkejuaraan',$data);
}

public function update_kejuaraan(){

    $data = $this->input->post();

    $id = $data['id_kejuaraan'];

    $result = $this->kejuaraan->update_kejuaraan($data,$id);
    pesan('sukses', 'Berhasil merubah data');

    redirect(base_url(). 'Admin/kejuaraan');
}

public function input_seleksiPemain(){
    $data = $this->input->post();
    $id = $data['id_user'];
    // dd($data);
    $update = $this->user->updateStatus($id);
    $insert = $this->kejuaraan->insertPemain($data);

    pesan('sukses', 'Berhasil menambah pemain');
    redirect(base_url(). 'Admin/info_kejuaraan/'.$data['id_kejuaraan']);
}

```

Dilanjutkan

Lanjutan

```

public function view_perhitungan($id_kejuaraan){
    $idkejuaraan = $id_kejuaraan;
    $data['kejuaraan'] = $this->kejuaraan->detailKejuaraan($idkejuaraan)[0];
    $posisi = array(1,2,3,4,5);
    $data['pemain'] = $this->kejuaraan->pemainKejuaraan($idkejuaraan);
    foreach ($posisi as $key => $value) {
        $data['pos'][$value] = $this->dKejuaraan-
>maksPos($id_kejuaraan,$value);
    }

    foreach ($data['pemain'] as $keyPemain => $pemain) {
        foreach ($data['pos'] as $keyPos => $pos) {
            if($pemain['id_posisi'] == $keyPos){
                $data['hasil'][$keyPemain]['id_detail'] =
                $pemain['id_detail'];
                $data['hasil'][$keyPemain]['posisi'] =
                $pemain['posisi'];
                $data['hasil'][$keyPemain]['nama'] =
                $pemain['nama'];
                $data['hasil'][$keyPemain]['id_user'] =
                $pemain['id_user'];
                $data['hasil'][$keyPemain]['id_posisi'] =
                $pemain['id_posisi'];
                $data['hasil'][$keyPemain]['nama'] =
                $pemain['nama'];
                //NORMALISASI
                $data['hasil'][$keyPemain]['nDribling'] =
                round($pemain['dribbling'] / $pos['maksDrib'],4);
                $data['hasil'][$keyPemain]['nPass'] =
                round($pemain['passing'] / $pos['maksPass'],4);
                $data['hasil'][$keyPemain]['nShoot'] =
                round($pemain['shooting'] / $pos['maksShoot'],4);
                $data['hasil'][$keyPemain]['nFisik'] =
                round($pemain['ketahanan_fisik'] / $pos['maksFisik'],4);
                $data['hasil'][$keyPemain]['nIq']
                = round($pemain['basket_iq'] / $pos['maksIq'],4);
                $data['hasil'][$keyPemain]['nTB']
                = round($pemain['tinggi_badan'] / $pos['maksTB'],4);
                $data['hasil'][$keyPemain]['nBB']
                = round($pemain['berat_badan'] / $pos['maksBB'],4);
            }
        }
    }
}

```

Dilanjutkan

Lanjutan

```
//PERKALIAN KE NILAI KRITERIA
        $data['hasil'][$keyPemain]['skor_Drib']      =
round($data['hasil'][$keyPemain]['nDribbling']      * $pemain['k_dribbling'],4);
        $data['hasil'][$keyPemain]['skor_Pass']      =
round($data['hasil'][$keyPemain]['nPass']          * $pemain['k_passing'],4);
        $data['hasil'][$keyPemain]['skor_Shoot']     =
round($data['hasil'][$keyPemain]['nShoot']         * $pemain['k_shooting'],4);
        $data['hasil'][$keyPemain]['skor_Fisik']     =
round($data['hasil'][$keyPemain]['nFisik']         * $pemain['k_ketahanan_fisik'],4);
        $data['hasil'][$keyPemain]['skor_Iq']
    = round($data['hasil'][$keyPemain]['nIq']        *
    $pemain['k_basket_iq'],4);
        $data['hasil'][$keyPemain]['skor_TB']
    = round($data['hasil'][$keyPemain]['nTB']        *
    $pemain['k_tinggi_badan'],4);
        $data['hasil'][$keyPemain]['skor_BB']
    = round($data['hasil'][$keyPemain]['nBB']        *
    $pemain['k_berat_badan'],4);

//JUMLAH AKHIR
        $data['hasil'][$keyPemain]['jumlah']        =
round($data['hasil'][$keyPemain]['skor_Drib'] +
    $data['hasil'][$keyPemain]['skor_Pass'] + $data['hasil'][$keyPemain]['skor_Shoot']
    + $data['hasil'][$keyPemain]['skor_Fisik'] + $data['hasil'][$keyPemain]['skor_Iq']
    + $data['hasil'][$keyPemain]['skor_TB'] +
    $data['hasil'][$keyPemain]['skor_BB'],4);
    }
}
}

    foreach ($posisi as $key => $value) {
        $data['lolosSeleksi'][$value] = $this->dKejuaraan-
>view_lolos($id_kejuaraan,$value);
    }
    $update = $this->user->set_StatusUndangan($idkejuaraan);
    $data['kriteria'] = $data['kriteria'] = $this->kriteria->dataKriteria();
// dd($data);
    $this->load->view('admin/v_perhitungan',$data);
}
}
```

Dilanjutkan

Lanjutan

```

public function perhitungan($id_kejuaraan){
    $idkejuaraan = $id_kejuaraan;
    $posisi = array(1,2,3,4,5);
    $data['pemain'] = $this->kejuaraan->pemainKejuaraan($idkejuaraan);
    foreach ($posisi as $key => $value) {
        $data['pos'][$value] = $this->dKejuaraan-
>maksPos($id_kejuaraan,$value);
    }

    foreach ($data['pemain'] as $keyPemain => $pemain) {
        foreach ($data['pos'] as $keyPos => $pos) {
            if($pemain['id_posisi'] == $keyPos){
                $result[$keyPemain]['id_detail'] =
                $pemain['id_detail'];
                $result[$keyPemain]['id_kejuaraan'] =
                $pemain['id_kejuaraan'];
                $result[$keyPemain]['id_user'] = $pemain['id_user'];
                $result[$keyPemain]['id_posisi'] =
                $pemain['id_posisi'];
                $result[$keyPemain]['nama'] = $pemain['nama'];
                //NORMALISASI
                $result[$keyPemain]['nDribling'] =
                $pemain['dribbling'] / $pos['maksDrib'];
                $result[$keyPemain]['nPass'] =
                $pemain['passing'] / $pos['maksPass'];
                $result[$keyPemain]['nShoot'] =
                $pemain['shooting'] / $pos['maksShoot'];
                $result[$keyPemain]['nFisik'] =
                $pemain['ketahanan_fisik'] / $pos['maksFisik'];
                $result[$keyPemain]['nIq'] =
                $pemain['basket_iq'] / $pos['maksIq'];
                $result[$keyPemain]['nTB'] =
                $pemain['tinggi_badan'] / $pos['maksTB'];
                $result[$keyPemain]['nBB'] =
                $pemain['berat_badan'] / $pos['maksBB'];
                //PERKALIAN KE NILAI KRITERIA
                $result[$keyPemain]['skor_Drib'] =
                round($result[$keyPemain]['nDribling'] * $pemain['k_dribbling'],4);
                $result[$keyPemain]['skor_Pass'] =
                round($result[$keyPemain]['nPass'] * $pemain['k_passing'],4);
                $result[$keyPemain]['skor_Shoot'] =

```

Dilanjutkan

Lanjutan

```

round($result[$keyPemain]['nShoot'] * $pemain['k_shooting'],4);
    $result[$keyPemain]['skor_Fisik'] =
round($result[$keyPemain]['nFisik'] * $pemain['k_ketahanan_fisik'],4);
    $result[$keyPemain]['skor_Iq'] =
round($result[$keyPemain]['nIq'] * $pemain['k_basket_iq'],4);
    $result[$keyPemain]['skor_TB'] =
round($result[$keyPemain]['nTB'] * $pemain['k_tinggi_badan'],4);
    $result[$keyPemain]['skor_BB'] =
round($result[$keyPemain]['nBB'] * $pemain['k_berat_badan'],4);

//JUMLAH AKHIR
    $result[$keyPemain]['jumlah'] =
round($result[$keyPemain]['skor_Drib'] + $result[$keyPemain]['skor_Pass']
+$result[$keyPemain]['skor_Shoot'] + $result[$keyPemain]['skor_Fisik'] +
$result[$keyPemain]['skor_Iq'] + $result[$keyPemain]['skor_TB'] +
$result[$keyPemain]['skor_BB'],4);

    $update = $this->dKejuaraan-
>updateSkor($result[$keyPemain]['id_detail'],$result[$keyPemain]['jumlah']);
    $update = $this->user-
>updateStatusBebas($pemain['id_user'],$pemain['id_kejuaraan']);
    $update = $this->user-
>updateUndangan($pemain['id_user'],0);

    }
}

foreach ($posisi as $key => $value) {
    $data['lolosSeleksi'][$value] = $this->dKejuaraan-
>lolosSeleksi($id_kejuaraan,$value);
}

for ($i=1; $i <= count($posisi) ; $i++) {
    foreach ($data['lolosSeleksi'][$i] as $key => $value) {
        $data[$i][$key]['id_detail'] = $value['id_detail'];
        $data[$i][$key]['id_kejuaraan'] = $value['id_kejuaraan'];
        $data[$i][$key]['id_posisi'] = $value['id_posisi'];
        $data[$i][$key]['skor'] = $value['skor'];
        $data[$i][$key]['status'] = 1;
    }
}

```

Dilanjutkan

Lanjutkan

```

        $update = $this->dKejuaraan-
>updateStatus($data[$i][$key]['id_detail'],$data[$i][$key]['status']);
        }
    }

    $data['tidakLolos'] = $this->dKejuaraan->tidakLolos($id_kejuaraan);
    foreach ($data['tidakLolos'] as $key => $value) {
        $update = $this->dKejuaraan->statusTdkLolos($value['id_detail']);
    }

    $update = $this->kejuaraan->statusKejuaraan($idkejuaraan);
    pesan('sukses', 'Seleksi pemain selesai');
    redirect(base_url(). 'Admin/view_perhitungan/'.$idkejuaraan);
}

public function edit_skor($id_detail){
    $data['skor_pemain'] = $this->dKejuaraan->detailSkor($id_detail)[0];
    $this->load->view('admin/v_editskor',$data);
}

public function update_detailSkor(){
    $data = $_POST;
    $id_detail = $data['id_detail'];
    unset($data['id_detail']);
    $update = $this->dKejuaraan->update_detailSkor($data,$id_detail);
    pesan('sukses', 'Berhasil mengubah data');
    redirect(base_url(). 'Admin/info_kejuaraan/'.$data['id_kejuaraan']);
}

}Dilanjutkan

```

D.2 Kode Program *class* controllers/Kirim

```

<?php

/**
 *
 */
class kirim extends CI_Controller
{

```

Dilanjutkan

Lanjutan

```
function __construct()
{
    parent::Controller();
}

public function index(){
    dd(5);
    $config = array(
        'protocol' => 'smtp',
        'smtp_host' => 'ssl://smtp.googlemail.com',
        'smtp_port' => 465,
        'smtp_user' => 'krsndwipayana@gmail.com',
        'smtp_pass' => '!Dwipayana081914'
    );

    $this->load->library('email',$config);
    $this->email->set_newline("\r\n");

    $this->email->from('barana@gmail.com','Admin');
    $this->email->to('rizky@gmail.com');
    $this->email->subject('Coba Emaik');
    $this->email->message('ini bekerja sangat baik');

    if ($this->email->send()) {
        echo "Pesan Terkirim";
    }else{
        show_error($this->email->print_debugger());
    }
}
}
?>
```

D.3 Kode Program *class controllers/Login*

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Login extends CI_Controller {

/**
 * Index Page for this controller.
 */
```

Dilanjutkan

Lanjutan

```

* Maps to the following URL
*      http://example.com/index.php/welcome
*      - or -
*      http://example.com/index.php/welcome/index
*      - or -
* Since this controller is set as the default controller in
* config/routes.php, it's displayed at http://example.com/
*
* So any other public methods not prefixed with an underscore will
* map to /index.php/welcome/<method_name>
* @see https://codeigniter.com/user_guide/general/urls.html
*/

```

```

public function __construct()
{
    parent::__construct();
    $this->load->library('session');
    $this->load->model("m_user","user");
    $this->load->model("m_detailKejuaraan","dKejuaraan");
}

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

public function auth(){

    $username = $_POST['username'];
    $password = $_POST['password'];
    $data['login'] = $this->user->login($username,$password)[0];
    if ($data['login']['jumlah'] == 1) {
        if ($data['login']['level'] == 0) {
            pesan('sukses', 'Selamat datang '.$username);
            redirect(base_url() . 'Admin');
        }else{
            $data = $this->user->cekData($username,$password)[0];
            if ($data['status_pesan'] == 0) {
                $data = $this->user-
>cekData($username,$password)[0];
                $session = array(
                    'id_user'    => $data['id_user'],
                    'nama'       => $data['nama'],

```

Dilanjutkan

Lanjutan

```
'id_posisi' => $data['id_posisi'],
    );
    $this->session->set_userdata('user', $session);
    pesan('sukses', 'Selamat datang '.$data['nama']);
    redirect(base_url() . 'User');
} else {
    $cekstatus = $this->dKejuaraan-
>cekStatus($data['id_user'],$data['status_pesan'])[0];
    if ($cekstatus['status'] == 1) {
        $session = array(
            'id_user' => $data['id_user'],
            'nama' => $data['nama'],
            'id_posisi' => $data['id_posisi'],
        );
        $this->session->set_userdata('user',
$session);
        pesan('sukses', 'Selamat anda lolos seleksi
'.$cekstatus['nama_kejuaraan']);
        $update = $this->dKejuaraan-
>uStatus($cekstatus['id_detail'],2);
        $update = $this->user-
>status_pesan($data['id_user']);
        redirect(base_url() . 'User');
    } elseif ($cekstatus['status'] == 3) {
        $session = array(
            'id_user' => $data['id_user'],
            'nama' => $data['nama'],
            'id_posisi' => $data['id_posisi'],
        );
        $this->session->set_userdata('user',
$session);
        pesan('gagal', 'Anda Gagal Lolos Seleksi
'.$cekstatus['nama_kejuaraan'].' Tetap Semangat!!!!');
        $update = $this->dKejuaraan-
>uStatus($cekstatus['id_detail'],4);
        $update = $this->user-
>status_pesan($data['id_user']);
        redirect(base_url() . 'User');
    } else {
        $data = $this->user-
>cekData($username,$password)[0];
```

Dilanjutkan

Lanjutan

```
                                $session = array(
                                    'id_user'      => $data['id_user'],
                                    'nama'         => $data['nama'],
                                    'id_posisi'    => $data['id_posisi'],
                                );
                                $this->session->set_userdata('user',
$session);
                                pesan('sukses', 'Selamat datang
'. $data['nama']);
                                redirect(base_url() . 'User');
                            }
                        }
                    }
                }else{
                    pesan('gagal', 'Username / Password Salah ');
                    redirect(base_url() . 'Login');
                }
            }
        }

    public function auth1(){

        $username = $_POST['username'];
        $password = $_POST['password'];
        $data['login'] = $this->user->login($username,$password)[0];
        if ($data['login']['jumlah'] == 1) {
            if ($data['login']['level'] == 0) {
                pesan('sukses', 'Selamat datang '.$username);
                redirect(base_url() . 'Admin');
            }if($data['login']['level'] == 1){
                $data = $this->user->cekData($username,$password)[0];
                $cekstatus = $this->dKejuaraan-
>cekStatus($data['id_user'])[0];
                if ($cekstatus['status'] == 3) {
                    $session = array(
                        'id_user'      => $data['id_user'],
                        'nama'         => $data['nama'],
                        'id_posisi'    => $data['id_posisi'],
                    );
                    $this->session->set_userdata('user', $session);
                    pesan('gagal', 'Anda Gagal Lolos Seleksi
'. $cekstatus['nama_kejuaraan'].' Tetap Semangat!!!!');
                    $update = $this->dKejuaraan-
>uStatus($cekstatus['id_detail'],4);
                }
            }
        }
    }
}
```

Dilanjutkan

Lanjutan

```

        redirect(base_url() . 'User');
    }elseif($cekstatus['status'] == 1){
        $session = array(
            'id_user'    => $data['id_user'],
            'nama'       => $data['nama'],
            'id_posisi' => $data['id_posisi'],
        );
        $this->session->set_userdata('user', $session);
        pesan('sukses', 'Selamat anda lolos seleksi
        '.$cekstatus['nama_kejuaraan']);
        $update = $this->dKejuaraan-
        >uStatus($cekstatus['id_detail'],2); redirect(base_url() . 'User');
    }else{
        $session = array(
            'id_user'    => $data['id_user'],
            'nama'       => $data['nama'],
            'id_posisi' => $data['id_posisi'],
        );
        $this->session->set_userdata('user', $session);
        pesan('sukses', 'Selamat datang '.$data['nama']);
        redirect(base_url() . 'User');
    }
    }else{
        $data = $this->user->cekData($username,$password)[0];
        $session = array(
            'id_user'    => $data['id_user'],
            'nama'       => $data['nama'],
            'id_posisi' => $data['id_posisi'],
        );
        $this->session->set_userdata('user', $session);
        pesan('sukses', 'Selamat datang '.$data['nama']);
        redirect(base_url() . 'User');
    }
    }else{
        pesan('gagal', 'Username / Password Salah ');
        redirect(base_url() . 'Login');
    }
}

public function home(){ $this->load->view('admin/v_admin'); }

```

Dilanjutkan

Lanjutan

```
public function logout(){
    redirect(base_url(). 'Login');
}

}
```

D.4 Kode Program *class controllers/User*

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class User extends CI_Controller {

/**
 * Index Page for this controller.
 *
 * Maps to the following URL
 *      http://example.com/index.php/welcome
 *  - or -
 *      http://example.com/index.php/welcome/index
 *  - or -
 * Since this controller is set as the default controller in
 * config/routes.php, it's displayed at http://example.com/
 *
 * So any other public methods not prefixed with an underscore will
 * map to /index.php/welcome/<method_name>
 * @see https://codeigniter.com/user_guide/general/urls.html
 */

public function __construct()
{
    parent::__construct();
    $this->load->library('session');
    $this->sess = $this->session->userdata('user');
    $this->load->model('m_user','user');
    $this->load->model('m_kejuaraan','kejuaraan');
    $this->load->model('m_detailKejuaraan','dKejuaraan');
    $this->load->model('m_kriteria','kriteria');
    $this->load->model('m_pesan','pesan');
}
}
```

Dilanjutkan

Lanjutan

```
public function index() //dashboard
{
    $data['nama'] = $this->sess['nama'];
    $this->load->view('user/v_dashboard',$data);
}

public function undangan(){
    $id = $this->sess['id_user'];
    $data['nama'] = $this->sess['nama'];
    $data['pesan'] = $this->pesan->pesanMasuk($id);
    // dd($data);
    $this->load->view('user/v_undangan',$data);
}

public function lihatPesan($id){
    $data['nama'] = $this->sess['nama'];
    $data['pesan'] = $this->pesan->viewPesan($id)[0];

    if ($data['pesan']['status_pesan'] == 0) {
        $status = 1;
        $update = $this->pesan->updateStatus($id,$status);
        $this->load->view('user/v_lihat',$data);
    }else{
        $this->load->view('user/v_lihat',$data);
    }
}

public function accept($id){
    $iduser = $this->sess['id_user'];
    $status = 2;
    $pesan = $this->pesan->viewPesan($id)[0];
    $kejuaraan = $pesan['id_kejuaraan'];
    $update = $this->pesan->updateStatus($id,$status);
    $update = $this->user->updateUndangan($iduser,$kejuaraan);
    pesan('sukses', 'Terima Kasih atas partisipasinya');
    redirect(base_url(). 'User/undangan/');
}
```

Dilanjutkan

Lanjutan

```

public function decline($id){
    $iduser = $this->sess['id_user'];
    $status = 3;
    $kejuaraan = 0;
    $update = $this->pesan->updateStatus($id,$status);
    $update = $this->user->updateUndangan($iduser,$kejuaraan);
    pesan('gagal', 'Oke ditunggu partisipasinya');
    redirect(base_url(). 'User/undangan/');
}

public function ranking(){
    $data['nama'] = $this->sess['nama'];
    $data['kejuaraan'] = $this->dKejuaraan->dataEvent($this->sess['id_user']);
    // dd($data);
    $this->load->view('user/v_ranking',$data);
}

public function view_perhitungan($id_kejuaraan){
    $data['nama'] = $this->sess['nama'];
    $idkejuaraan = $id_kejuaraan;
    $posisi = $this->sess['id_posisi'];
    $id = $this->sess['id_user'];

    $data['kejuaraan'] = $this->kejuaraan->detailKejuaraan($idkejuaraan)[0];
    $data['pemain'] = $this->kejuaraan->userKejuaraan($idkejuaraan,$posisi);
    $data['pos'][$posisi] = $this->dKejuaraan-
    >maksPos($id_kejuaraan,$posisi);

    foreach ($data['pemain'] as $keyPemain => $pemain) {
        foreach ($data['pos'] as $keyPos => $pos) {
            if($pemain['id_posisi'] == $keyPos){
                $data['hasil'][$keyPemain]['id_detail'] =
                $pemain['id_detail'];
                $data['hasil'][$keyPemain]['posisi'] =
                $pemain['posisi'];
                $data['hasil'][$keyPemain]['nama'] =
                $pemain['nama'];
                $data['hasil'][$keyPemain]['id_user'] =
                $pemain['id_user'];
                $data['hasil'][$keyPemain]['id_posisi'] =
                $pemain['id_posisi'];
                $data['hasil'][$keyPemain]['nama'] =
                $pemain['nama'];
            }
        }
    }
}

```

Dilanjutkan

```

lanjutan
//NORMALISASI
round($pemain['dribbling']      $data['hasil'][$keyPemain]['nDribling']      =
                                / $pos['maksDrib'],4);
round($pemain['passing']        $data['hasil'][$keyPemain]['nPass']      =
                                / $pos['maksPass'],4);
round($pemain['shooting']       $data['hasil'][$keyPemain]['nShoot']     =
                                / $pos['maksShoot'],4);
round($pemain['ketahanan_fisik'] $data['hasil'][$keyPemain]['nFisik']     =
                                / $pos['maksFisik'],4);
                                $data['hasil'][$keyPemain]['nIq']
= round($pemain['basket_iq']    / $pos['maksIq'],4);
                                $data['hasil'][$keyPemain]['nTB']
= round($pemain['tinggi_badan'] / $pos['maksTB'],4);
                                $data['hasil'][$keyPemain]['nBB']
= round($pemain['berat_badan'] / $pos['maksBB'],4);

//PERKALIAN KE NILAI KRITERIA
round($data['hasil'][$keyPemain]['nDribling'] * $pemain['k_dribbling'],4);
                                $data['hasil'][$keyPemain]['skor_Pass'] =
round($data['hasil'][$keyPemain]['nPass'] * $pemain['k_passing'],4);
                                $data['hasil'][$keyPemain]['skor_Shoot'] =
round($data['hasil'][$keyPemain]['nShoot'] * $pemain['k_shooting'],4);
                                $data['hasil'][$keyPemain]['skor_Fisik'] =
round($data['hasil'][$keyPemain]['nFisik'] * $pemain['k_ketahanan_fisik'],4);
                                $data['hasil'][$keyPemain]['skor_Iq']
= round($data['hasil'][$keyPemain]['nIq'] *
$pemain['k_basket_iq'],4);
                                $data['hasil'][$keyPemain]['skor_TB']
= round($data['hasil'][$keyPemain]['nTB'] *
$pemain['k_tinggi_badan'],4);
                                $data['hasil'][$keyPemain]['skor_BB']
= round($data['hasil'][$keyPemain]['nBB'] *
$pemain['k_berat_badan'],4);

//JUMLAH AKHIR
                                $data['hasil'][$keyPemain]['jumlah'] =
round($data['hasil'][$keyPemain]['skor_Drib'] +
$data['hasil'][$keyPemain]['skor_Pass'] + $data['hasil'][$keyPemain]['skor_Shoot']
+ $data['hasil'][$keyPemain]['skor_Fisik'] + $data['hasil'][$keyPemain]['skor_Iq']
+ $data['hasil'][$keyPemain]['skor_TB'] +
$data['hasil'][$keyPemain]['skor_BB'],4);

```

Dilanjutkan

Lanjutan

```

        }
    }
}

$data['lolosSeleksi'] = $this->dKejuaraan-
>view_lolos($id_kejuaraan,$posisi);

$data['kriteria'] = $this->kriteria->dataKriteria();
$data['kriteria1'] = $this->kriteria->dataKriteria1($posisi)[0];
// dd($data);
$this->load->view('user/v_perhitungan',$data);
}
}

```

D.5 Kode Program *class models/m_detailKejuaraan*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class m_detailKejuaraan extends CI_Model {

    private $tbName;

    public function __construct()
    {
        parent::__construct();

        $this->tbName = 'detail_kejuaraan';
        $this->primary = 'id_detail';
    }

    public function maksPos($idkejuaraan,$posisi){
        $this->db->select('user.id_posisi, kriteria.posisi, max(dribbling) as
maksDrib, max(passing) as maksPass, max(shooting) as maksShoot,
max(ketahanan_fisik) as maksFisik, max(basket_iq) as maksIq,
max(tinggi_badan) as maksTB, max(berat_badan) as maksBB');

        $this->db->from('detail_kejuaraan');
        $this->db->join('user','user.id_user = detail_kejuaraan.id_user');
        $this->db->join('kriteria','user.id_posisi = kriteria.id_kriteria');
        $this->db->where('id_kejuaraan',$idkejuaraan);
        $this->db->where('id_posisi',$posisi);
    }
}

```

Dilanjutkan

Lanjutan

```
$maksPos_1 = $this->db->get();
return $maksPos_1->row_array();
}

public function updateSkor($id,$jumlah){
    return $this->db->query("UPDATE detail_kejuaraan SET skor =
'$jumlah', status = 0 WHERE id_detail = '$id'");
}

public function updateStatus($id,$status){
    return $this->db->query("UPDATE detail_kejuaraan SET status =
'$status' WHERE id_detail = '$id'");
}

public function lolosSeleksi($id,$posisi){

    $this->db->select('*');
    $this->db->from('detail_kejuaraan');
    $this->db->join('kejuaraan','kejuaraan.id_kejuaraan =
detail_kejuaraan.id_kejuaraan');
    $this->db->join('user','user.id_user = detail_kejuaraan.id_user');
    $this->db->join('kriteria','kriteria.id_kriteria = user.id_posisi');
    $this->db->where('detail_kejuaraan.id_kejuaraan',$id);
    $this->db->where('user.id_posisi',$posisi);
    $this->db->order_by('detail_kejuaraan.skor','desc');
    $this->db->LIMIT(3);

    $lolosSeleksi = $this->db->get();
    return $lolosSeleksi->result_array();
}

public function tidakLolos($id){
    $this->db->select('id_detail');
    $this->db->from('detail_kejuaraan');
    $this->db->join('kejuaraan','kejuaraan.id_kejuaraan =
detail_kejuaraan.id_kejuaraan');
    $this->db->join('user','user.id_user = detail_kejuaraan.id_user');
    $this->db->join('kriteria','kriteria.id_kriteria = user.id_posisi');
    $this->db->where('detail_kejuaraan.id_kejuaraan',$id);
    $this->db->where('detail_kejuaraan.status',0);

    $lolosSeleksi = $this->db->get();
    return $lolosSeleksi->result_array();
}
```

Dilanjutkan

Lanjutan

```

public function view_lolos($id,$posisi){

    $this->db-
>select('user.id_user,nama,posisi,skor,detail_kejuaraan.status');
    $this->db->from('detail_kejuaraan');
    $this->db->join('kejuaraan','kejuaraan.id_kejuaraan =
detail_kejuaraan.id_kejuaraan');
    $this->db->join('user','user.id_user = detail_kejuaraan.id_user');
    $this->db->join('kriteria','kriteria.id_kriteria = user.id_posisi');
    $this->db->where('detail_kejuaraan.id_kejuaraan',$id);
    $this->db->where('user.id_posisi',$posisi);
    $this->db->order_by('detail_kejuaraan.skor','desc');
    // $this->db->LIMIT(3);

    $lolosSeleksi = $this->db->get();
    return $lolosSeleksi->result_array();
}

public function detailSkor($id_detail){
    $this->db->select('*');
    $this->db->from('detail_kejuaraan');
    $this->db->join('user','user.id_user = detail_kejuaraan.id_user');
    $this->db->where('id_detail',$id_detail);

    $detailSkor = $this->db->get();
    return $detailSkor->result_array();
}

public function update_detailSkor($data,$id_detail){
    $this->db->where($this->primary, $id_detail);
    return $this->db->update($this->tbName, $data);
}

public function cekStatus($id,$idkejuaraan){
    $this->db-
>select('kejuaraan.id_kejuaraan,kejuaraan.nama_kejuaraan,id_user,id_detail,detail
_kejuaraan.status');
    $this->db->from('detail_kejuaraan');
    $this->db->join('kejuaraan','kejuaraan.id_kejuaraan =
detail_kejuaraan.id_kejuaraan');
    $this->db->where('id_user',$id);
    $this->db->where('detail_kejuaraan.id_kejuaraan',$idkejuaraan);
    $this->db->order_by('detail_kejuaraan.id_kejuaraan','desc');
    $this->db->order_by('id_detail','desc');
}

```

Dilanjutkan

```

Lanjutan
        $this->db->limit(1);
        // $query = "SELECT id_kejuaraan, id_user, id_detail status from
detail_kejuaraan WHERE id_user = 21 ORDER BY id_kejuaraan desc, id_detail
desc";

        $cekStatus = $this->db->get();
        return $cekStatus->result_array();
    }

    public function uStatus($id,$angka){
        return $this->db->query("UPDATE detail_kejuaraan SET status =
'$angka' WHERE id_detail ='$id'");
    }

    public function dataEvent($id){
        $this->db-
>select('kejuaraan.id_kejuaraan,kejuaraan.nama_kejuaraan,id_user,id_detail,keju
araan.status');
        $this->db->from('detail_kejuaraan');
        $this->db->join('kejuaraan','kejuaraan.id_kejuaraan =
detail_kejuaraan.id_kejuaraan');
        $this->db->where('id_user',$id);
        $this->db->order_by('kejuaraan.status','desc');

        $dataEvent = $this->db->get();
        return $dataEvent->result_array();
    }

    public function statusTdkLolos($id){
        return $this->db->query("UPDATE detail_kejuaraan SET status =
3 WHERE id_detail ='$id'");
    }
}

```

D.6 Kode Program *class models/m_kejuaraan*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class m_kejuaraan extends CI_Model {

    private $tbName;

```

Dilanjutkan

Lanjutan

```
public function __construct()
{
    parent::__construct();

    $this->tbName = 'kejuaraan';
    $this->primary = 'id_kejuaraan';
}

public function insert($data){
    return $this->db->insert($this->tbName, $data);
}

public function insertPemain($data){
    return $this->db->insert('detail_kejuaraan',$data);
}

public function namaKejuaraan($id){
    $this->db->select('nama_kejuaraan,maks_umur,gender');
    $this->db->from('kejuaraan');
    $this->db->where('id_kejuaraan',$id);

    $namaKejuaraan = $this->db->get();
    return $namaKejuaraan->result_array();
}

public function dataKejuaraan(){
    $this->db->select('kejuaraan.*,count(detail_kejuaraan.id_kejuaraan) as total');
    $this->db->from('kejuaraan');
    $this->db->join('detail_kejuaraan','detail_kejuaraan.id_kejuaraan =
kejuaraan.id_kejuaraan','left');
    $this->db->group_by('kejuaraan.id_kejuaraan');
    $this->db->order_by('kejuaraan.status', 'desc');
    $this->db->order_by('kejuaraan.id_kejuaraan');

    $dataKejuaraan = $this->db->get();
    return $dataKejuaraan->result_array();
}

public function detailKejuaraan($idkejuaraan){
    $this->db->select('*');
    $this->db->from('kejuaraan');
    $this->db->where('id_kejuaraan',$idkejuaraan);
}
```

Dilanjutkan

Lanjutan

```

        $detailKejuaraan = $this->db->get();
        return $detailKejuaraan->result_array();
    }

    public function pemainKejuaraan($idkejuaraan){
        $this->db->select('*');
        $this->db->from('kejuaraan');
        $this->db->join('detail_kejuaraan','detail_kejuaraan.id_kejuaraan =
kejuaraan.id_kejuaraan');
        $this->db->join('user','detail_kejuaraan.id_user = user.id_user');
        $this->db->join('kriteria','kriteria.id_kriteria = user.id_posisi');
        $this->db->where('detail_kejuaraan.id_kejuaraan',$idkejuaraan);
        $this->db->order_by('id_posisi','asc');

        $pemainKejuaraan = $this->db->get();
        return $pemainKejuaraan->result_array();
    }

    public function daftarPemain($maksumur,$idkejuaraan){
        $query = ('SELECT
*,TIMESTAMPDIFF(YEAR,tanggal_lahir,CURDATE()) as umur FROM
`user` WHERE YEAR(CURDATE()) - YEAR(tanggal_lahir) <= ' . $maksumur. '
and status_pemain = 0 and status_undangan = ' . $idkejuaraan. ' and id_user NOT
IN ((SELECT id_user FROM detail_kejuaraan WHERE id_kejuaraan
= ' . $idkejuaraan. '))););
// $query ('SELECT
id_user,TIMESTAMPDIFF(YEAR,tanggal_lahir,CURDATE()) as umur FROM
`user` WHERE TIMESTAMPDIFF(YEAR,tanggal_lahir,CURDATE()) < 20 and
id_user NOT IN ((SELECT id_user FROM detail_kejuaraan WHERE
id_kejuaraan = 2))););

        $daftarPemain = $this->db->query($query);
        return $daftarPemain->result_array();
    }

    public function editKejuaraan($id_kejuaraan){
        $this->db->select('*');
        $this->db->from('kejuaraan');
        $this->db->where('id_kejuaraan',$id_kejuaraan);

        $editKejuaraan = $this->db->get();
        return $editKejuaraan->result_array();
    }

```

Dilanjutkan

```

Lanjutan
    public function update_kejuaraan($data,$id){
        $this->db->where($this->primary, $id);
        return $this->db->update($this->tbName, $data);
    }

    public function cekPosisi($id){
        $this->db->select('id_posisi');
        $this->db->from('user');
        $this->db->where('id_user',$id);

        $cek_posisi = $this->db->get();
        return $cek_posisi->result_array();
    }

    public function statusKejuaraan($idkejuaraan){
        return $this->db->query("UPDATE kejuaraan SET status = 1
WHERE id_kejuaraan ='$idkejuaraan");
    }

    public function userKejuaraan($idkejuaraan,$posisi){
        $this->db->select('*');
        $this->db->from('kejuaraan');
        $this->db->join('detail_kejuaraan','detail_kejuaraan.id_kejuaraan =
kejuaraan.id_kejuaraan');
        $this->db->join('user','detail_kejuaraan.id_user = user.id_user');
        $this->db->join('kriteria','kriteria.id_kriteria = user.id_posisi');
        $this->db->where('detail_kejuaraan.id_kejuaraan',$idkejuaraan);
        $this->db->where('user.id_posisi',$posisi);
        $this->db->order_by('id_posisi','asc');

        $userKejuaraan = $this->db->get();
        return $userKejuaraan->result_array();
    }
}

```

D.7 Kode Program *class models/m_kriteria*

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class m_kriteria extends CI_Model {

```

Dilanjutkan

Lanjutan

```
private $tbName;

public function __construct()
{
    parent::__construct();

    $this->tbName = 'kriteria';
    $this->primary = 'id_kriteria';
}

public function dataKriteria(){
    $this->db->select('*');
    $this->db->from('kriteria');

    $dataKriteria = $this->db->get();
    return $dataKriteria->result_array();
}

public function dataKriteria1($id_kriteria){
    $this->db->select('*');
    $this->db->from('kriteria');
    $this->db->where('id_kriteria',$id_kriteria);
    $dataKriteria = $this->db->get();
    return $dataKriteria->result_array();
}

public function detailKriteria($id_kriteria){
    $this->db->select('*');
    $this->db->from('kriteria');
    $this->db->where('id_kriteria',$id_kriteria);

    $detailKriteria = $this->db->get();
    return $detailKriteria->result_array();
}

public function updateKriteria($id_kriteria,$data){
    $this->db->where($this->primary, $id_kriteria);
    return $this->db->update($this->tbName, $data);
}
}
```

D.8 Kode Program *class models/m_pesan*

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class m_pesan extends CI_Model {

    private $tbName;

    public function __construct()
    {
        parent::__construct();

        $this->tbName = 'pesan';
        $this->primary = 'id_pesan';
    }

    public function inputUndangan($id,$id_kejuaraan,$pesan){
        return $this->db->query("INSERT INTO pesan (`id_pesan`,
`id_user`, `id_kejuaraan`, `tanggal_pesan`, `isi_pesan`, `status_pesan`) VALUES
(NULL, '$id', '$id_kejuaraan', CURRENT_TIMESTAMP, '$pesan', '0')");
    }

    public function pesanMasuk($id){
        $this->db->select('*');
        $this->db->from('pesan');
        $this->db->where('id_user',$id);
        $this->db->order_by('id_pesan','desc');

        $pesanMasuk = $this->db->get();
        return $pesanMasuk->result_array();
    }

    public function viewPesan($id){
        $this->db->select('*');
        $this->db->from('pesan');
        $this->db->where('id_pesan',$id);

        $viewPesan = $this->db->get();
        return $viewPesan->result_array();
    }

    public function updateStatus($id,$status){
        return $this->db->query("UPDATE pesan SET status_pesan =
Dilanjutkan
```



```
Lanjutan
'status' WHERE id_pesan ='$id''');
    }
}
}
```

D.9 Kode Program *class models/m_user*

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class m_user extends CI_Model {

    private $tbName;

    public function __construct()
    {
        parent::__construct();

        $this->tbName = 'user';
        $this->primary = 'id_user';
    }

    public function login($username, $password)
    {
        $this->db->select("id_user, count(id_user) as jumlah, level");
        $this->db->from("user");
        $this->db->where("username",$username);
        $this->db->where("password",$password);

        $login = $this->db->get();
        return $login->result_array();
    }

    public function cekusername($username)
    {
        $this->db->select('count(id_user) as jumlah');
        $this->db->from('user');
        $this->db->where('username',$username);

        $cekusername = $this->db->get();
        return $cekusername->result_array();
    }
}
```

Dilanjutkan

Lanjutan

```
}

    public function namaPemain($id)
    {
        $this->db->select('nama,username,password');

$this->db->from('user');
        $this->db->where('id_user',$id);

        $namaPemain = $this->db->get();
        return $namaPemain->result_array();
    }

    public function emailUser($id)
    {
        $this->db->select('email');
        $this->db->from('user');
        $this->db->where('id_user',$id);

        $emailUser = $this->db->get();
        return $emailUser->result_array();
    }

    public function insert($data){
        return $this->db->insert($this->tbName, $data);
    }

    public function dataPemain(){
        $this->db->select("id_user,username,password,nama,gender,kota_lahir,tanggal_lahir,posisi,
TIMESTAMPDIFF(YEAR,tanggal_lahir,CURDATE()) as umur");
        $this->db->from("user");
        $this->db->join('kriteria','user.id_posisi = kriteria.id_kriteria');
        $this->db->where("level",1);

        $dataPemain = $this->db->get();
        return $dataPemain->result_array();
    }

    public function updateUndangan($id,$kejuaraan){
```

Dilanjutkan

Lanjutan

```
return $this->db->query("UPDATE user SET status_undangan = '$kejuaraan'
WHERE id_user ='$id'");
}

public function edit($id_user){
    $this->db->select("*",id_user");
    $this->db->from("user");
    $this->db->where("id_user",$id_user);
    $edit = $this->db->get();
    return $edit->result_array();
}

public function update($data, $id){
    $this->db->where($this->primary, $id);
return $this->db->update($this->tbName, $data);
}

public function cekData($username,$password){
    $this->db->select("id_user,nama,id_posisi,status_pesan");
    $this->db->from("user");
    $this->db->where("username",$username);
    $this->db->where("password",$password);

    $cekData = $this->db->get();
    return $cekData->result_array();
}

public function updateStatus($id){
    return $this->db->query("UPDATE user SET status_pemain = 1
WHERE id_user ='$id'");
}

public function updateStatusBebas($id,$kejuaraan){
    return $this->db->query("UPDATE user SET status_pemain = 0,
status_pesan = '$kejuaraan' WHERE id_user ='$id'");
}

public function status_pesan($id){
    return $this->db->query("UPDATE user SET status_pesan = 0
WHERE id_user ='$id'");
}
```

Dilanjutkan

Lanjutan

```

    public function set_StatusUndangan($idkejuaraan){

return $this->db->query("UPDATE user SET status_undangan = 0 WHERE
status_undangan = '$idkejuaraan'");
    }

    public function undangPemain($umur){
        $this->db-
>select("id_user,username,password,nama,kota_lahir,tanggal_lahir,posisi,TIMES
TAMPDIFF(YEAR,tanggal_lahir,CURDATE()) as umur");

$this->db->from("user");
        $this->db->join('kriteria','user.id_posisi = kriteria.id_kriteria');
        $this->db->where("level",1);
        $this->db->where("status_undangan",0);
        $this->db-
>where("TIMESTAMPDIFF(YEAR,tanggal_lahir,CURDATE()) <=", $umur);

        $dataPemain = $this->db->get();
        return $dataPemain->result_array();
    }

    public function undangPemain2($umur,$id_kejuaraan,$gender){
        $query = ('SELECT
id_user,nama,tanggal_lahir,TIMESTAMPDIFF(YEAR,tanggal_lahir,CURDATE(
)) as umur,posisi FROM `user` u JOIN `kriteria` k on u.id_posisi = k.id_kriteria
WHERE TIMESTAMPDIFF(YEAR,tanggal_lahir,CURDATE()) <= '.$umur.'
and gender = '.$gender.' and id_user not in ((SELECT id_user FROM `pesan`
WHERE id_kejuaraan = '.$id_kejuaraan.'));
        // $query = ('SELECT
id_user,tanggal_lahir,TIMESTAMPDIFF(YEAR,tanggal_lahir,CURDATE()) as
umur,posisi FROM `user` u JOIN `kriteria` k on u.id_posisi = k.id_kriteria
WHERE YEAR(CURDATE()) - YEAR(tanggal_lahir) <= '.$maksumur.' and
status_pemain = 0 and status_undangan = '.$idkejuaraan.' and id_user NOT IN
((SELECT id_user FROM detail_kejuaraan WHERE id_kejuaraan
='.$idkejuaraan.'));

        $undangPemain2 = $this->db->query($query);
        return $undangPemain2->result_array();
    }
}

```

LAMPIRAN E. PENGUJIAN BLACK BOX

No	Fitur	Aksi	Hasil	Kesimpulan
1	Login	Mengisi kolom username dan password lalu klik tombol login	<i>Login</i> berhasil dan menampilkan halaman <i>dashboard</i>	[√] Berhasil [] Gagal
		Kolom <i>username</i> atau <i>password</i> tidak sesuai	Menampilkan pesan “ <i>username/password</i> salah “	[√] Berhasil [] Gagal
2	<i>View Dashboard Admin</i>	Mengisi kolom <i>username</i> dan <i>password Admin</i> lalu klik tombol <i>login</i>	<i>Login</i> berhasil dan menampilkan halaman <i>dashboard Admin</i>	[√] Berhasil [] Gagal
		Kolom <i>username</i> atau <i>password Admin</i> tidak sesuai	Menampilkan pesan “ <i>username/password</i> salah “	[√] Berhasil [] Gagal
3	<i>View Dashboard Admin</i>	Mengisi kolom <i>username</i> dan <i>password User</i> lalu klik tombol <i>login</i>	<i>Login</i> berhasil dan menampilkan halaman <i>dashboard User</i>	[√] Berhasil [] Gagal
		Kolom <i>username</i> atau <i>password User</i> tidak sesuai	Menampilkan pesan “ <i>username/password</i> salah “	[√] Berhasil [] Gagal
4	Mengelola Biodata Pemain	Klik tombol tambah	Menampilkan form tambah pemain	[√] Berhasil [] Gagal
		Mengisi form tambah pemain yang berisi <i>username</i> , <i>password</i> , nama, alamat, kota lahir, <i>e-mail</i> , tanggal lahir, posisi, jenis kelamin dan klik <i>submit</i>	Input data berhasil dan menampilkan pesan “sukses berhasil menambahkan data”	[√] Berhasil [] Gagal
		Form tambah pemain yang berisi <i>username</i> , <i>password</i> , nama, alamat, kota lahir, <i>e-mail</i> , tanggal lahir, posisi, jenis kelamin kosong	Menampilkan pesan “please fill out this field”	[√] Berhasil [] Gagal

Dilanjutkan

Lanjutan

		Klik tombol <i>action</i> ubah	Menampilkan form ubah data pemain	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi form ubah data yang berisi <i>password</i> , nama, alamat, kota lahir, <i>e-mail</i> , tanggal lahir, posisi, jenis kelamin dan klik submit	Ubah data berhasil dan menampilkan pesan “sukses berhasil merubah data”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		form ubah data yang berisi <i>password</i> , nama, alamat, kota lahir, <i>e-mail</i> , tanggal lahir, posisi, jenis kelamin kosong	Menampilkan pesan “please fill out this field”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
5	Data Kejuaraan	Klik tombol tambah	Menampilkan form tambah kejuaraan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi form tambah kejuaraan yang berisi nama kejuaraan, maksimal umur, jenis kelamin dan klik <i>submit</i>	Input data berhasil dan menampilkan pesan ”sukses berhasil menambahkan data”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol <i>action</i> edit	Menampilkan form ubah data kejuaraan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Mengisi form ubah data kejuaraan yang berisi maksimal umur, jenis kelamin dan klik <i>submit</i>	Ubah data berhasil dan menampilkan pesan “sukses berhasil merubah data”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Form ubah data kejuaraan yang berisi maksimal umur, jenis kelamin kosong	Menampilkan pesan “please fill out this field”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
6	Mengelola Nilai Seleksi	Klik tombol info pada daftar kejuaraan	Menampilkan halaman data pemain seleksi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol tambah	Menampilkan form tambah pemain seleksi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Dilanjutkan

Lanjutan

		Mengisi form tambah pemain seleksi yang berisi <i>dribbling, passing, shooting</i> , ketahanan fisik, basket IQ, tinggi badan, berat badan lalu klik submit	Input data berhasil dan menampilkan pesan” sukses berhasil menambahkan data”	[√] Berhasil [] Gagal
		Form tambah pemain seleksi yang berisi <i>dribbling, passing, shooting</i> , ketahanan fisik, basket IQ, tinggi badan, berat badan kosong	Menampilkan pesan “please fill out this field”	[√] Berhasil [] Gagal
		Klik tombol edit	Menampilkan form ubah nilai pemain	[√] Berhasil [] Gagal
		Mengisi form ubah nilai pemain seleksi yang berisi <i>dribbling, passing, shooting</i> , ketahanan fisik, basket IQ, tinggi badan, berat badan dan klik submit	Ubah data berhasil dan menampilkan pesan “sukses mengubah data”	[√] Berhasil [] Gagal
		Form ubah nilai pemain seleksi yang berisi <i>dribbling, passing, shooting</i> , ketahanan fisik, basket IQ, tinggi badan, berat badan kosong	Menampilkan pesan “please fill out this field”	[√] Berhasil [] Gagal
7	Mengelola Nilai Bobot	Klik menu kriteria	Menampilkan halaman data kriteria SAW	[√] Berhasil [] Gagal
		Klik action edit	Menampilkan halaman form ubah kriteria	[√] Berhasil [] Gagal
		Mengisi form ubah kriteria yang berisi <i>dribbling, passing, shooting</i> , ketahanan fisik, basket IQ, tinggi badan, berat badan dan klik submit	Berhasil merubah data dan menampilkan pesan “ sukses berhasil merubah data”	[√] Berhasil [] Gagal

Dilanjutkan

Lanjutan

		Form ubah kriteria yang berisi <i>dribbling</i> , <i>passing</i> , <i>shooting</i> , ketahanan fisik, basket IQ, tinggi badan, berat badan kosong	Menampilkan pesan “please fill out this field”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Form ubah kriteria yang berisi <i>dribbling</i> , <i>passing</i> , <i>shooting</i> , ketahanan fisik, basket IQ, tinggi badan, berat badan lebih besar dari 100%	Menampilkan pesan “perhatian total bobot harus = 100”	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
8	View Perhitungan SAW	Klik menu kejuaraan	Menampilkan halaman data kejuaraan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol <i>action</i> perhitungan	Menampilkan halaman perhitungan seleksi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol tahap 2	Menampilkan halaman skor setiap pemain	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol tahap 3	Menampilkan halaman nilai maksimal	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol tahap 4	Menampilkan halaman normalisasi 1	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol tahap 5	Menampilkan halaman nilai preferensi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik tombol tahap 6	Menampilkan halaman ranking	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
9	Kirim Undangan Seleksi	Klik menu kejuaraan	Menampilkan halaman daftar kejuaraan	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
		Klik <i>action</i> info	Menampilkan data pemain seleksi	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal

Dilanjutkan

Lanjutan

		Klik undang pemain	Menampilkan halaman undang pemain	[√] Berhasil [] Gagal
		Klik <i>action</i> undang	Menampilkan pesan “sukses undangan telah dikirim”	[√] Berhasil [] Gagal
10	Terima Undangan Seleksi	Klik menu undangan pada <i>dashboard user</i>	Menampilkan halaman pesan masuk	[√] Berhasil [] Gagal
		Klik <i>action</i> lihat	Menampilkan isi pesan	[√] Berhasil [] Gagal
		Klik tombol terima	Menampilkan pesan “sukses terima kasih atas partisipasinya”	[√] Berhasil [] Gagal
		Klik tombol tolak	Menampilkan pesan “perhatian ok ditunggu partisipasinya”	[√] Berhasil [] Gagal
11	View Ranking	Klik menu ranking pada <i>dashboard user</i>	Menampilkan halaman daftar seleksi yang diikuti	[√] Berhasil [] Gagal
		Klik <i>action</i> info	Menampilkan ranking	[√] Berhasil [] Gagal
12	Logout	Klik tombol <i>logout</i>	Menampilkan halaman login	[√] Berhasil [] Gagal

LAMPIRAN D. TRANSKRIP WAWANCARA

Narasumber : Arif Angga Yudha (Perbasi Jember)

Jabatan : Kepala Bidang Kepelatiham Perbasi Jember

- Apa saja Prestasi Perbasi Jember ?
“Untuk prestasi paling membanggakan adalah salah satu atlet Jember Sandy Febiansyah terpilih menjadi pemain ini di Timnas Indonesia pada kejuaraan Sea Games 2015, dan mendapat medali perak, Selanjutnya untuk prestasi regional Tim Jember Putra meraih medali perak dalam kejuaraan PORPROV 2013.”
- Bagaimana perkembangan basket di Jember?
“Kalau disini pada tahun 2016 hampir setiap bulan terdapat satu agenda kompetisi yang berjalan di Jember. Dibandingkan tahun 2012 yang setiap tiga bulan hanya ada satu kompetisi. Khusus untuk kompetisi tingkat Sekolah Dasar (SD) pada Tahun 2016 terdapat empat kompetisi yang berlangsung, di bandingkan tahun 2012 yang tidak ada kompetisi”
- Untuk meningkatkan prestasi bola basket di Jember, apa yang dilakukan oleh Perbasi Jember ?
“Perbasi Jember mempunyai progam-progam kerja untuk meningkatkan kualitas perbasketan di Jember. Salah satu progam kerja Perbasi Jember adalah pembinaan atlet bola basket mulai dari usia dini. Bentuk kegiatan yang dilakukan adalah mengadakan turnamen antar pelajar dan pemusatan latihan/training camp (TC) untuk mendapatkan atlet-atlet terbaik Jember yang akan dikirim untuk mengikuti kejuaraan di dalam kota maupun luar kota mewakili Kabupaten Jember”
- Bagaimana mekanisme seleksi di Perbasi Jember?
“Program seleksi Perbasi Jember dipimpin dan dinilai berdasarkan pengamatan pelatih. Biasanya coaching staff terdiri dari 2 orang pelatih. Atlet-atlet terbaik Jember akan dipilih melalui proses seleksi dan dikirim ke kejuaraan, Mekanisme seleksi yang dilakukan yaitu menilai kemampuan dasar-dasar dalam permainan bola basket seperti: drible, passing, shooting, fisik, dll Setiap posisi dalam bermain basket mempunyai prioritas penilaian yang akan di nilai sesuai dengan materi yang diberikan. Setiap posisi akan dipilih 3 pemain terbaik. Total luaran yang di hasilkan dalam program seleksi ini adalah 15 orang pemain terbaik mewakili 1 tim. Proses seleksi mengandalkan sistem manual”

- Apa yang dibutuhkan Perbasi Jember sekarang ?
“Perbasi Jember membutuhkan suatu sistem yang dapat memudahkan pelatih menentukan pemain yang cocok untuk tim Jember, mungkin sistem yang terkomputerisasi”
- Bagaimana mekanisme seleksi yang dibutuhkan Perbasi Jember?
“Kriteria setiap posisi akan ditentukan penilaiannya sesuai dengan bobot masing-masing kriteria, kriteria setiap posisi mempunyai bobot yang berbeda”
- Apa saja kriteria seleksi yang dibutuhkan pelatih di Jember?
“Untuk kriteria seleksi meliputi dribbling, passing, shooting, ketahanan fisik, basketball IQ, tinggi badan, dan berat badan”
- Berapa bobot masing-masing kriteria tersebut dalam setiap posisi?
*“Untuk posisi point guard bobot kriteria seleksi dribbling 30%, passing 20%, shooting 10%, ketahanan fisik 20%, basketball IQ 20%, tinggi badan 5%, berat badan 5%
Untuk posisi shooting guard bobot kriteria seleksi dribbling 10%, passing 10%, shooting 40%, ketahanan fisik 10%, basketball IQ 20%, tinggi badan 5%, berat badan 5%
Untuk posisi small forward bobot kriteria seleksi dribbling 20%, passing 10%, shooting 20%, ketahanan fisik 10%, basketball IQ 20%, tinggi badan 10%, berat badan 10%
Untuk posisi power forward bobot kriteria seleksi dribbling 5%, passing 5%, shooting 10%, ketahanan fisik 10%, basketball IQ 20%, tinggi badan 30%, berat badan 20%
Untuk posisi center bobot kriteria seleksi dribbling 5%, passing 5%, shooting 5%, ketahanan fisik 10%, basketball IQ 20%, tinggi badan 5%, berat badan 5%”*
- Bagaimana anda menentukan nilai bobot tersebut ?
“Ini adalah penjelasan tugas utama posisi dalam bermain basket. Point guard mempunyai tugas memimpin penyerangan, maka seorang point guard harus memiliki kemampuan membawa bola (Dribbling) diatas rata-rata dan harus memiliki distribusi bola (Passing) yang baik. Shooting Guard mempunyai tugas menembak (shooting) tiga angka, maka seorang shooting guard harus mempunyai tembakan yang baik. Small forward merupakan seorang allrounder yaitu pemain yang mampu mencetak angka dari semua kesempatan menyerang, baik itu shooting dan penetrasi kedalam pertahanan lawan (drive), maka seorang small forward harus mempunyai kemampuan

dribbling dan shooting sama baiknya. Power forward bertugas menangkap bola pantul (rebound) yang gagal masuk ke ring, terutama saat bertahan, maka seorang power forward harus bertubuh tinggi, semakin tinggi pemain, maka semakin mudah untuk melakukan rebound. Biasanya seorang power forward juga memiliki tembakan dua angka yang baik. Posisi center harus ditempati oleh pemain yang bertubuh paling tinggi dan besar, karena tugas dari seorang center menjadi pertahanan terakhir dalam bermain bola basket”

- Apa saja kebutuhan pelatih untuk diimplementasikan ke sistem ?
“kalau menurut saya, pertama sistem harus dapat mencatat seluruh database pemain yang ada di Jember, selanjutnya dapat menginputkan data kejuaraan apa saja yang nantinya akan diikuti, lalu dapat mengelola nilai seleksi pemain itu yang penting”

