



**RANCANG BANGUN APLIKASI *MULTI ATTRIBUTE DECISION MAKING* (MADM) MENGGUNAKAN ALGORITMA *SIMPLE ADDITIVE WEIGHTING* (SAW)**

**SKRIPSI**

Oleh

**Triyas Tuti Febriani**

**NIM 112410101034**

**Program Studi Sistem Informasi**

**Universitas Jember**

**2015**



**RANCANG BANGUN APLIKASI *MULTI ATTRIBUTE DECISION*  
*MAKING* (MADM) MENGGUNAKAN ALGORITMA *SIMPLE ADDITIVE*  
*WEIGHTING* (SAW)**

**SKRIPSI**

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

Oleh

**Triyas Tuti Febriani**

**NIM 112410101034**

**Program Studi Sistem Informasi**

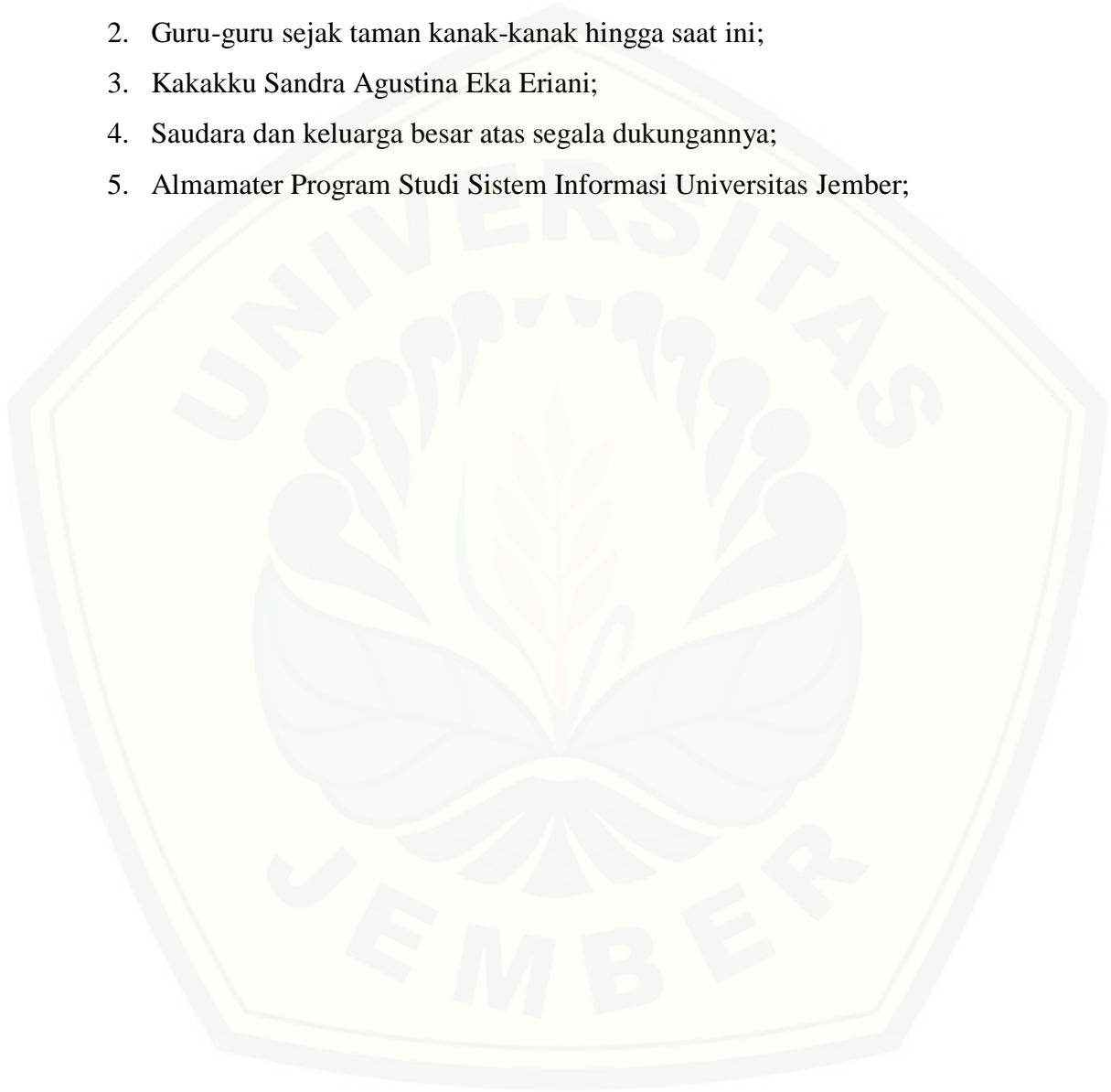
**Universitas Jember**

**2015**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Ibunda Ribut Eko Mujiati dan Ayahanda Erfan Hanaqi;
2. Guru-guru sejak taman kanak-kanak hingga saat ini;
3. Kakakku Sandra Agustina Eka Eriani;
4. Saudara dan keluarga besar atas segala dukungannya;
5. Almamater Program Studi Sistem Informasi Universitas Jember;



## MOTO

“Dalam hidup nyata, dan dalam perjuangan yang tak mudah, kita bukan tokoh dalam dongeng dan mitos, yang gagah berani dan penuh sifat kepahlawanan. Kita yang bukan tokoh mitos yang punya anak istri dan keluarga, mengenal rasa takut.

Tapi bahwa meskipun takut kita jalan terus dan berani melompati pagar batas kekuatan tadi, mungkin disitu harga kita ditetapkan.”

(KH. Abdurrahman Wahid)

“Kau akan berhasil dalam setiap pelajaran, dan kau harus percaya akan berhasil, dan berhasillah kau. Anggap semua pelajaran mudah, dan semua akan jadi mudah. Jangan takut pada pelajaran apa pun, karena ketakutan itu sendiri kebodohan awal yang akan membodohkan semua”

(Pramoedya Ananta Toer)

*“Knowledge is king and that’s all you need to be a real modern rebel”*

(I Gede Ari Astina)

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

nama : Triyas Tuti Febriani

NIM : 112410101034

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Rancang Bangun Aplikasi *Multi Attribute Decision Making* (MADM) Menggunakan Algoritma *Simple Additive Weighting* (SAW)” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan 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 mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 30 Agustus 2015

Yang menyatakan,

Triyas Tuti Febriani

NIM 112410101034

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “**Rancang Bangun Aplikasi *Multi Attribute Decision Making (MADM) Menggunakan Algoritma Simple Additive Weighting (SAW)***”, telah diuji dan disahkan pada:

Hari, tanggal : 10 Desember 2015

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori, ST., M.Kom.

NIP. 196811131994121001

Yanuar Nurdiansyah, ST., M.Cs.

NIP. 198201012010121004

**SKRIPSI**

**RANCANG BANGUN APLIKASI *MULTI ATTRIBUTE DECISION*  
*MAKING* (MADM) MENGGUNAKAN ALGORITMA *SIMPLE ADDITIVE*  
*WEIGHTING* (SAW)**

Oleh

Triyas Tuti Febriani

NIM 112410101034

Pembimbing

Dosen Pembimbing Utama : Dr. Saiful Bukhori, ST., M.Kom.

Dosen Pembimbing Pendamping : Yanuar Nurdiansyah, ST., M.Cs.

**PENGESAHAN**

Karya ilmiah Skripsi berjudul “**Rancang Bangun Aplikasi *Multi Attribute Decision Making (MADM) Menggunakan Algoritma Simple Additive Weighting (SAW)***” telah diuji dan disahkan pada:

hari, tanggal : 10 Desember 2015

tempat : Program Studi Sistem informasi.

Tim Penguji:

Penguji I,

Penguji II,

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

NIP. 196704201992011001

M. Arief Hidayat, S.Kom., M.Kom.

NIP. 198101232010121003

Mengesahkan

Ketua,

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

NIP. 196704201992011001



## RINGKASAN

**Rancang Bangun Aplikasi *Multi Attribute Decision Making* (MADM) Menggunakan Algoritma *Simple Additive Weighting* (SAW);** Triyas Tuti Febriani, 112410101034; 2015; 120 halaman; Program Studi Sistem Informasi Universitas Jember.

*Multi Attribute Decision Making* (MADM) merupakan salah satu kategori dalam sistem penunjang keputusan dengan banyak kriteria. MADM mencakup pemilihan alternatif terbaik dari pilihan alternatif yang tersedia dengan mempertimbangkan banyak atribut. Penelitian ini bertujuan untuk merancang dan membangun perangkat lunak yang mengimplementasikan metode *Simple Additive Weighting* (SAW), yang merupakan salah satu metode yang terdapat dalam MADM. Metode ini dipilih karena merupakan metode yang paling banyak digunakan dalam sistem penunjang keputusan. Metode ini juga memiliki langkah-langkah penyelesaian yang sederhana, sehingga sumber daya yang dibutuhkan relatif rendah dan mudah dipahami oleh pengguna yang asing terhadap sistem penunjang keputusan. Tujuan penelitian ini adalah untuk membangun aplikasi penunjang keputusan yang menerapkan tahapan dalam metode SAW untuk menginisiasi penunjang keputusan dan menggunakannya untuk memperoleh sebuah rekomendasi keputusan berupa ranking nilai dari alternatif serta analisa sensitivitas bobot dan nilai preferensi yang diperoleh dari proses perhitungan.

Penelitian dilakukan dengan mengeksplorasi kajian-kajian dan teori menyangkut topik dari penelitian-penelitian sebelumnya. Hasil yang diperoleh akan diintegrasikan sehingga memungkinkan untuk membangun sebuah perangkat lunak. Untuk mengetahui apakah perangkat lunak yang dihasilkan telah berjalan sesuai alur kerja yang diharapkan, terutama dalam kalkulasi menggunakan formula-formula yang ada, dibutuhkan sebuah data sampel yang diperoleh dari penelitian lain dan sampel dari perhitungan manual dengan nilai acak.

Pembuatan penunjang keputusan dengan aplikasi ini dibagi dalam 3 tahap, dimana tiap tahap terdiri dari beberapa langkah. Tahap pertama adalah inisiasi

problem yang terdiri dari: (1) pemodelan problem, (2) definisi kriteria, (3) input perbandingan berpasangan, dan (4) perhitungan indeks konsistensi. Sedangkan tahap kedua adalah tahap perhitungan nilai preferensi alternatif menggunakan metode formula SAW, yang terdiri dari: (1) input nilai masing-masing alternatif dan (2) perankingan alternatif berdasarkan nilai terbesar. Tahap ketiga adalah tahap perhitungan analisa sensitivitas. Tahap ini merupakan proses tambahan yang bisa dilakukan atau tidak oleh pengguna sistem mengingat proses ini cukup membantu dengan adanya kemungkinan perubahan bobot kriteria. Pengguna juga dapat menyimpan struktur model problem penunjang keputusan yang telah diinisiasi kedalam basis data dan dapat mengelolanya agar sesuai dengan perubahan-perubahan yang dimungkinkan terjadi pada suatu struktur model penunjang keputusan.

Perangkat lunak yang dibangun dapat digunakan membuat penunjang keputusan dengan hasil akhir berupa ranking alternatif. Untuk penunjang keputusan berjenis seleksi, evaluasi, optimasi, atau jenis lain yang pemrosesan datanya berbeda dengan pemrosesan data dalam aplikasi ini (dengan hasil akhir berupa ranking alternatif), maka pengguna dapat mentransformasikannya dalam pengolahan data yang didukung oleh perangkat lunak ini.

## PRAKATA

Pujiyukur kehadiran Allah SWT atas limpahan rahmat serta karunia-Nya, sehingga penulis dapat menyelesaikan skripsi berjudul “**Rancang Bangun Aplikasi Multi Attribute Decision Making (MADM) Menggunakan Algoritma Simple Additive Weighting (SAW)**”. Skripsi ini disusun untuk memenuhi syarat menyelesaikan pendidikan pada Program Studi Sistem Informasi Universitas Jember.

Penulisan skripsi ini dapat diselesaikan atas bantuan berbagai pihak. Untuk itu penulis mengucapkan terima kasih kepada:

1. Prof. Drs. Slamin, M.Comp.Sc.,Ph.D, selaku Ketua Program Studi Sistem Informasi;
2. Dr. Saiful Bukhori, ST., M.Kom. selaku Pembimbing Utama , dan Yanuar Nurdiansyah, ST,. M.Cs. selaku Pembimbing Anggota;
3. Seluruh dosen dan staf karyawan di Program Studi Sistem Informasi Universitas Jember;

Penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna, maka dari itu saran yang bersifat membangun sangat diharapkan.

Jember,

Penulis

**DAFTAR ISI**

PERSEMBAHAN .....	ii
MOTO .....	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	v
SKRIPSI.....	vi
PENGESAHAN .....	vii
RINGKASAN .....	viii
PRAKATA.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR LAMPIRAN.....	xvi
BAB 1. PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Tujuan dan Manfaat.....	3
1.4 Batasan Masalah.....	4
BAB 2. TINJAUAN PUSTAKA .....	5
3.1 Penelitian Sebelumnya .....	5
3.2 Sistem Penunjang Pengambilan Keputusan .....	7
3.3 <i>Multi-Attribute Decision Making</i> (MADM).....	7
3.4 <i>Simple Additive Weighting</i> (SAW).....	8
3.5 Contoh Kasus Penggunaan Metode SAW .....	10
BAB 3. METODOLOGI PENELITIAN.....	15
3.1 Jenis Penelitian .....	15
3.2 Tahapan Penelitian .....	15
3.3 Teknik Pengumpulan Data .....	16
3.4 Tahap Analisis .....	16
3.5 Tahap Pengembangan Sistem.....	17
BAB 4. ANALISIS DAN PENGEMBANGAN SISTEM.....	19

4.1	Pengumpulan Data.....	19
4.2	Penerapan Metode <i>Simple Additive Weighting</i> (SAW).....	21
4.3	Pengembangan Sistem.....	22
4.3.1	<i>Statement Of Purpose</i> .....	22
4.3.2	Definisi Kebutuhan .....	22
4.3.3	<i>Business Process</i> .....	24
4.3.4	<i>Use Case Diagram</i> .....	24
4.3.5	Skenario.....	28
4.3.6	<i>Sequence Diagram</i> .....	33
4.3.7	<i>Activity Diagram</i> .....	40
4.3.8	<i>Class Diagram</i> .....	45
4.3.9	<i>Entity relationship Diagram</i> (ERD).....	47
4.4	Penulisan Kode Program .....	47
4.5	Pengujian .....	51
4.5.1	<i>White Box Testing</i> .....	51
4.5.2	<i>Black Box Testing</i> .....	67
4.6	Pemeliharaan .....	70
BAB 5. HASIL DAN PEMBAHASAN.....		71
5.1	Perangkat Lunak MADM Menggunakan Metode SAW .....	71
5.2	Implementasi Metode SAW .....	73
5.3	Pembahasan .....	78
5.3.1	Implementasi Metode SAW dalam Desain Sistem .....	79
5.3.2	Implementasi metode SAW dalam Kode program Aplikasi.....	79
5.4	Pengujian Perhitungan Sistem.....	83
5.5	Batasan Sistem.....	90
BAB 6. PENUTUP.....		93
6.1	Kesimpulan.....	93
6.2	Saran.....	94

Daftar Pustaka

LAMPIRAN

**DAFTAR TABEL**

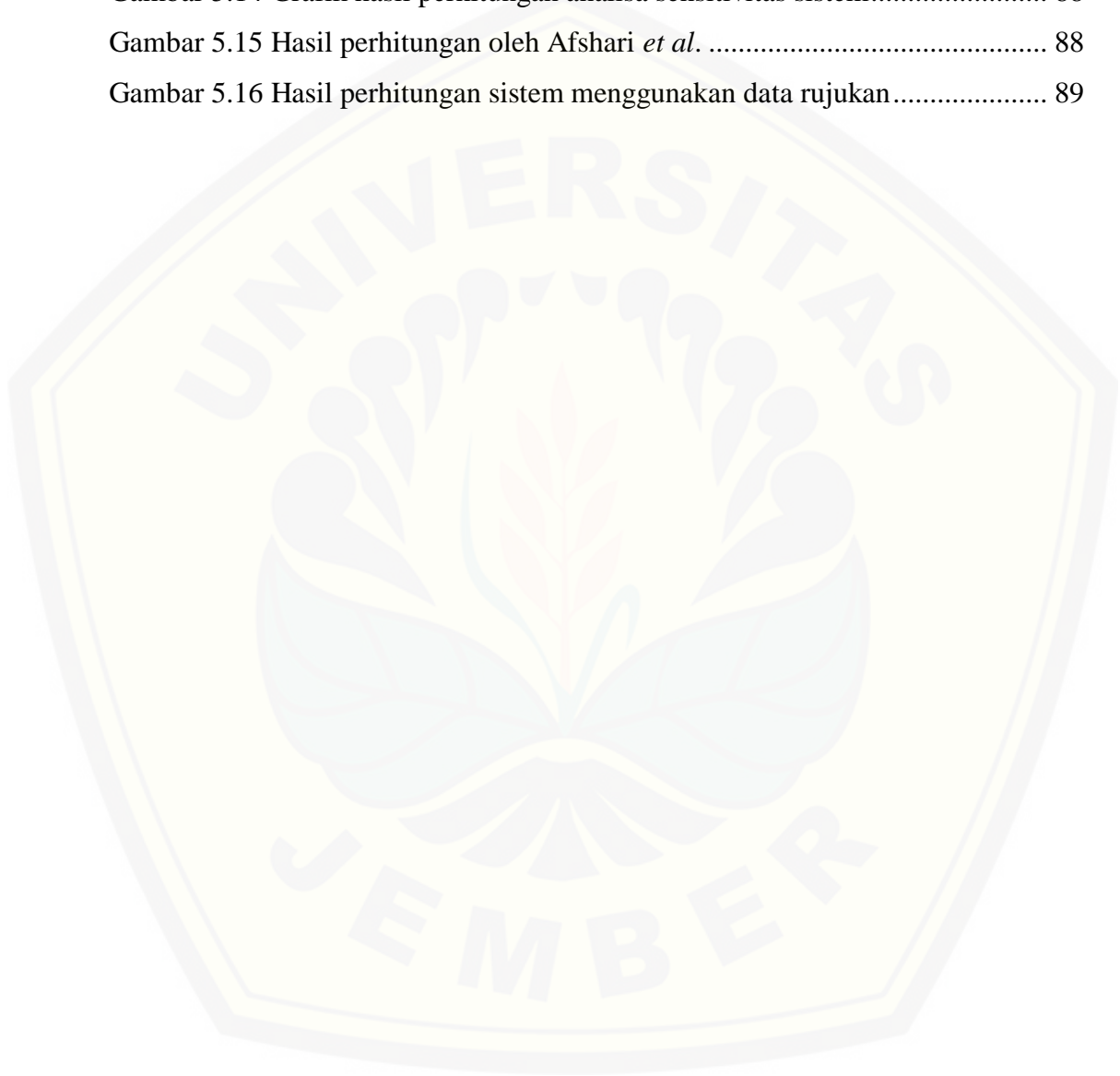
	Halaman
Tabel 2.1 Ketentuan penilaian verbal.....	11
Tabel 2.2 Matrik perbandingan berpasangan .....	11
Tabel 2.3 Normalisasi matrik perbandingan berpasangan .....	11
Tabel 2.4 Matrik Penilaian Alternatif .....	13
Tabel 2.5 Normalisasi matrik keputusan.....	13
Tabel 2.6 Nilai preferensi baru.....	14
Tabel 4.1 <i>Saaty's Random Index (RI)</i> .....	20
Tabel 4.3 Skenario pemodelan problem.....	29
Tabel 4.4 Skenario definisi kriteria.....	30
Tabel 4.5 Skenario input perbandingan berpasangan .....	31
Tabel 4.6 Skenario cek konsistensi bobot .....	32
Tabel 4.7 Kode program <i>class</i> penjumlahanTerbobot.java .....	48
Tabel 4.8 Kode program <i>class</i> matrikKeputusan .....	49
Tabel 4.9 Kode program <i>class</i> analisaSensisifitas .....	50
Tabel 4.10 <i>Listing program class</i> CRUDdata.....	51
Tabel 4.11 <i>Test case</i> fitur pengelolaan data .....	63
Tabel 4.12 Pengujian <i>black box</i> fitur Pengelolaan data .....	68
Tabel 5.1 Kode program <i>class</i> penjumlahanTerbobot.java .....	80
Tabel 5.2 Implementasi SAW dalam <i>class</i> controllerPenilaianAlternatif.java.....	80
Tabel 5.3 Hasil perhitungan manual bobot kriteria dan konsistensinya .....	84
Tabel 5.4 Hasil perhitungan manual nilai preferensi alternatif.....	84
Tabel 5.5 Hasil perhitungan manual analisa sensitivitas .....	86
Tabel 5.6 Batasan sistem.....	92



**DAFTAR GAMBAR**

	Halaman
Gambar 3.1 <i>Flow chart</i> sistem.....	17
Gambar 3.2 Tahapan model <i>waterfall</i> .....	18
Gambar 4.1 <i>Business process</i> .....	24
Gambar 4.2 <i>Use case diagram</i> .....	25
Gambar 4.3 <i>Sequence diagram</i> pemodelan problem .....	35
Gambar 4.4 <i>Sequence diagram</i> definisi kriteria.....	36
Gambar 4.5 <i>Sequence diagram</i> input perbandingan berpasangan .....	38
Gambar 4.6 <i>Sequence diagram</i> cek konsistensi bobot.....	39
Gambar 4.7 <i>Activity diagram</i> pemodelan problem .....	41
Gambar 4.8 <i>Activity diagram</i> definisi kriteria.....	42
Gambar 4.9 <i>Activity diagram</i> input perbandingan berpasangan .....	43
Gambar 4.10 <i>Activity diagram</i> cek konsistensi bobot.....	44
Gambar 4.11 <i>Class diagram</i> .....	46
Gambar 4.12 <i>Entity Relationship Diagram</i> (ERD).....	47
Gambar 4.13 Diagram alir mengambil data daftar SPK dan detail SPK dari <i>database</i> .....	58
Gambar 4.14 Diagram alir memuat detail data, hapus data, dan mengakses nilai <i>RI</i> .....	59
Gambar 4.15 Diagram alir perubahan data struktur problem tersimpan.....	60
Gambar 4.16 Diagram alir penyimpanan data .....	61
Gambar 5.1 Tampilan awal aplikasi.....	72
Gambar 5.2 Tampilan daftar data struktur model problem penunjang keputusan	72
Gambar 5.3 Tampilan jendela pemodelan problem .....	73
Gambar 5.4 Tampilan <i>form</i> isian definisi kriteria (1) .....	74
Gambar 5.5 Tampilan <i>form</i> isian definisi kriteria (2) .....	74
Gambar 5.6 Tampilan <i>form</i> perbandingan berpasangan .....	75
Gambar 5.7 Tampilan indeks konsistensi bobot dan bobot yang diperoleh .....	76
Gambar 5.8 Tampilan penilaian alternatif .....	76
Gambar 5.9 Tampilan jendela ranking alternatif .....	77

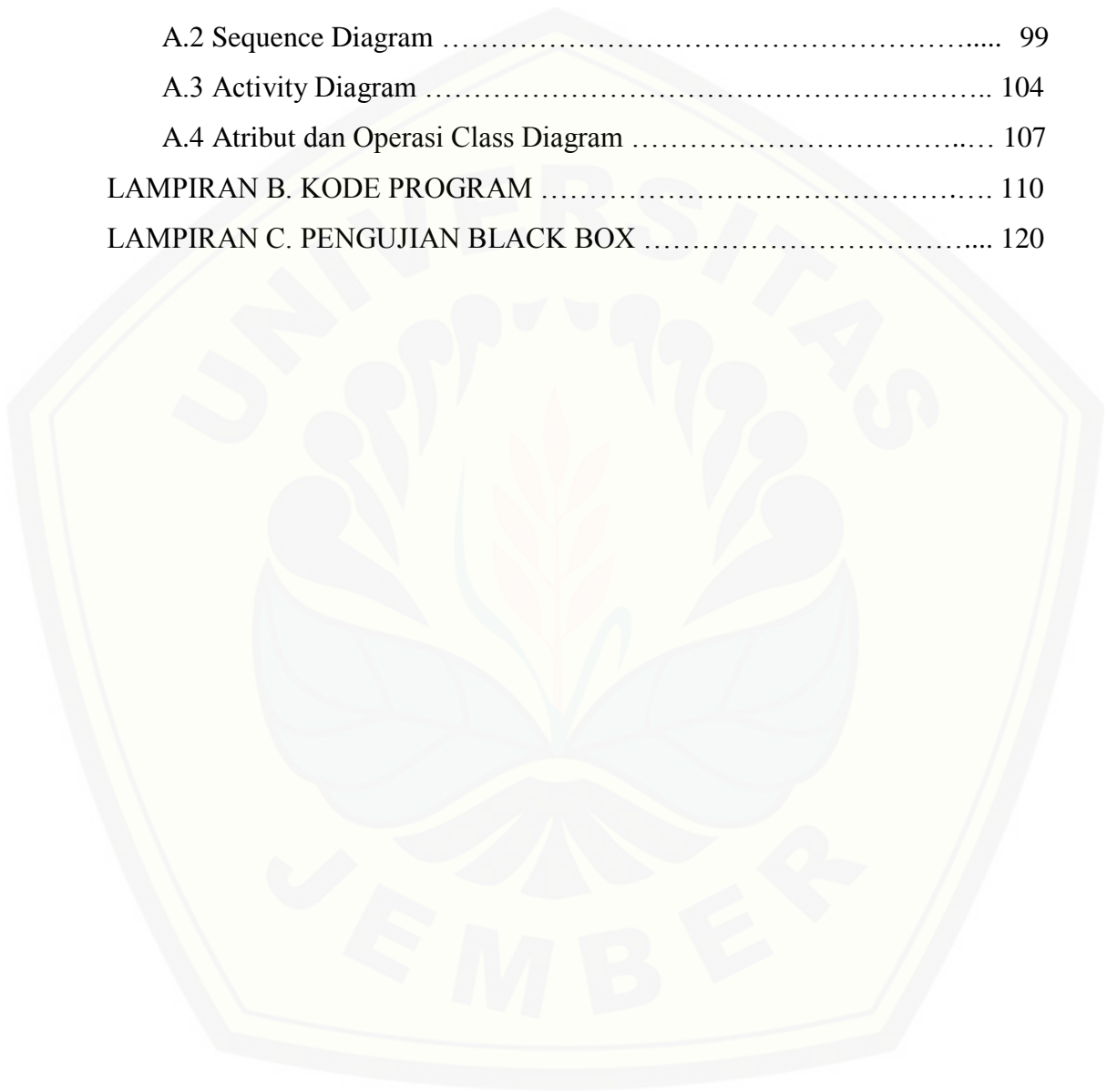
Gambar 5.10 Tampilan analisa sensitivitas.....	78
Gambar 5.11 Hasil perhitungan bobot kriteria dan konsistensinya oleh sistem ...	85
Gambar 5.12 Hasil perhitungan bobot kriteria dan konsistensinya oleh sistem ...	86
Gambar 5.13 Grafik analisa sensitivitas menggunakan perhitungan manual .....	87
Gambar 5.14 Grafik hasil perhitungan analisa sensitivitas sistem.....	88
Gambar 5.15 Hasil perhitungan oleh Afshari <i>et al.</i> ....	88
Gambar 5.16 Hasil perhitungan sistem menggunakan data rujukan.....	89





**DAFTAR LAMPIRAN**

LAMPIRAN A. PERANCANGAN .....	95
A.1 Skenario Sistem .....	95
A.2 Sequence Diagram .....	99
A.3 Activity Diagram .....	104
A.4 Atribut dan Operasi Class Diagram .....	107
LAMPIRAN B. KODE PROGRAM .....	110
LAMPIRAN C. PENGUJIAN BLACK BOX .....	120



## BAB 1. PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang penulisan skripsi, perumusan masalah penelitian, dan tujuan dan manfaat yang diharapkan atas penelitian yang dilakukan.

### 1.1 Latar Belakang

Pengambilan keputusan merupakan bagian dari kehidupan manusia. Setiap permasalahan yang dihadapi oleh manusia membutuhkan penentuan keputusan untuk menyelesaikan masalah tersebut. Keputusan diambil dari pemilihan satu atau lebih dari alternatif-alternatif yang tersedia. Untuk memilih alternatif dari permasalahan yang ada, dibutuhkan kriteria penilaian. Semakin banyak kriteria yang digunakan dalam keputusan akan semakin menyulitkan *Decision Maker* (DM) untuk menentukan alternatif mana yang akan dipilih.

Mobilitas yang tinggi seperti sekarang ini menuntut manusia untuk mengambil berbagai keputusan yang cepat namun tetap memperhitungkan kriteria yang ada dengan hasil yang semaksimal mungkin. Hal ini tentu sulit untuk dilakukan oleh manusia yang memiliki nilai subjektifitas tinggi dan memiliki batas kemampuan untuk mempertimbangkan berbagai kriteria dalam memutuskan satu masalah. Selain itu, permasalahan yang dihadapi juga sangat variatif. Sistem pendukung keputusan untuk satu objek akan tidak efektif mengingat problem pengambilan keputusan yang bermacam-macam. Untuk itulah dibutuhkan suatu sistem yang membantu manusia untuk membuat keputusan atas permasalahan-permasalahan dengan banyak kriteria.

Pengambilan keputusan dengan banyak kriteria dapat dilakukan dengan menggunakan *Multi-Criteria Decision Making* (MCDM). MCDM merupakan proses yang memungkinkan pengguna untuk membuat keputusan dengan keterlibatan banyak kriteria. MCDM menyediakan metode-metode penyelesaian

untuk masalah pengambilan keputusan dengan banyak kriteria. MCDM dapat dibagi kedalam dua kategori, yakni *Multi-Attribute Decision Making* dan *Multi-Objective Decision Making* (MODM). Hwang menjelaskan bahwa MADM merupakan pemilihan alternatif terbaik dari pilihan alternatif yang telah dideskripsikan sebelumnya, sedangkan MODM merupakan desain dari alternatif yang mengoptimalkan banyak objektif dari *Decision Maker* (DM) (Sun, Tanpa Tahun).

Problem pengambilan keputusan dengan single kriteria sangat intuitif dalam penyelesaiannya. Namun apabila dihadapkan dengan problem pengambilan keputusan dengan multi kriteria maka beberapa problem akan muncul, seperti pembobotan kriteria, konflik antar kriteria, kemungkinan perubahan bobot kriteria yang berpengaruh pada perubahan ranking alternatif dan beberapa problem lain yang membutuhkan metode tertentu untuk menyelesaikannya.

Salah satu metode yang terdapat dalam MADM adalah metode pengambilan keputusan *Simple Additive Weighting* (SAW). Metode ini menitikberatkan pada pembobotan kriteria. Metode ini juga memiliki formulasi untuk menanggulangi konflik antar kriteria yang biasa terjadi dalam pengambilan keputusan dengan banyak kriteria. Konflik antar kriteria yang dimaksud adalah dimana suatu penambahan nilai pada salah satu kriteria dapat menjadi deflasi pada kriteria lain (Memariani, 2009).

Terdapat berbagai perangkat lunak pengambilan keputusan yang serupa dengan perangkat lunak yang akan dibangun kali ini, yaitu antara lain: *Expert Choice* oleh *Expert Choice Inc.*, merupakan *software* yang menerapkan metode *Analytical Hierarchy Process*, *Criterium DecisionPlus* oleh *InfoHarvest Inc.* menerapkan metode AHP dan *Simple Multi-Attribute Rating Technique* (SMART), *Hiview3* oleh *Catalyze Ltd.* menerapkan metode *direct rating and swing weighting*. Sementara untuk metode SAW, aplikasi yang tersedia hanya untuk masalah-masalah yang spesifik seperti yang dijelaskan sebelumnya.

Berdasarkan latar belakang diatas, maka penulis tergerak untuk melakukan suatu penelitian untuk merancang dan membangun sistem pendukung pengambilan yang dapat digunakan untuk membantu pembuatan keputusan

berbagai permasalahan dengan banyak kriteria. Suatu sistem yang memungkinkan penggunaannya untuk mendefinisikan sendiri properti pengambilan keputusannya menggunakan metode SAW. Properti pengambilan keputusan yang dimaksud adalah, kriteria, alternatif, bobot kriteria, dan nilai alternatif tiap kriteria. Metode SAW dipilih karena metode ini merupakan metode yang paling sering digunakan (Afshari et al, 2010). Selain itu algoritma SAW sederhana sehingga pengguna yang tidak memiliki pemahaman yang mendalam mengenai sistem pengambilan keputusan, dapat mendefinisikan sendiri pengambilan keputusannya. Diharapkan pengguna dapat menerapkan langkah-langkah pengambilan keputusan dari metode SAW dengan menjalankan aplikasi yang dihasilkan dari penelitian ini.

## 1.2 Perumusan Masalah

Berdasar penjabaran diatas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana menerapkan langkah-langkah dalam algoritma *Simple Additive Weighting* kedalam sebuah aplikasi?
2. Bagaimana memberi rekomendasi keputusan kepada pengguna dan menganalisa perubahan rekomendasi keputusan yang disebabkan perubahan bobot salah satu atribut?
3. Bagaimana mengelola basis data problem pengambilan keputusan yang telah didefinisikan oleh pengguna?

## 1.3 Tujuan dan Manfaat

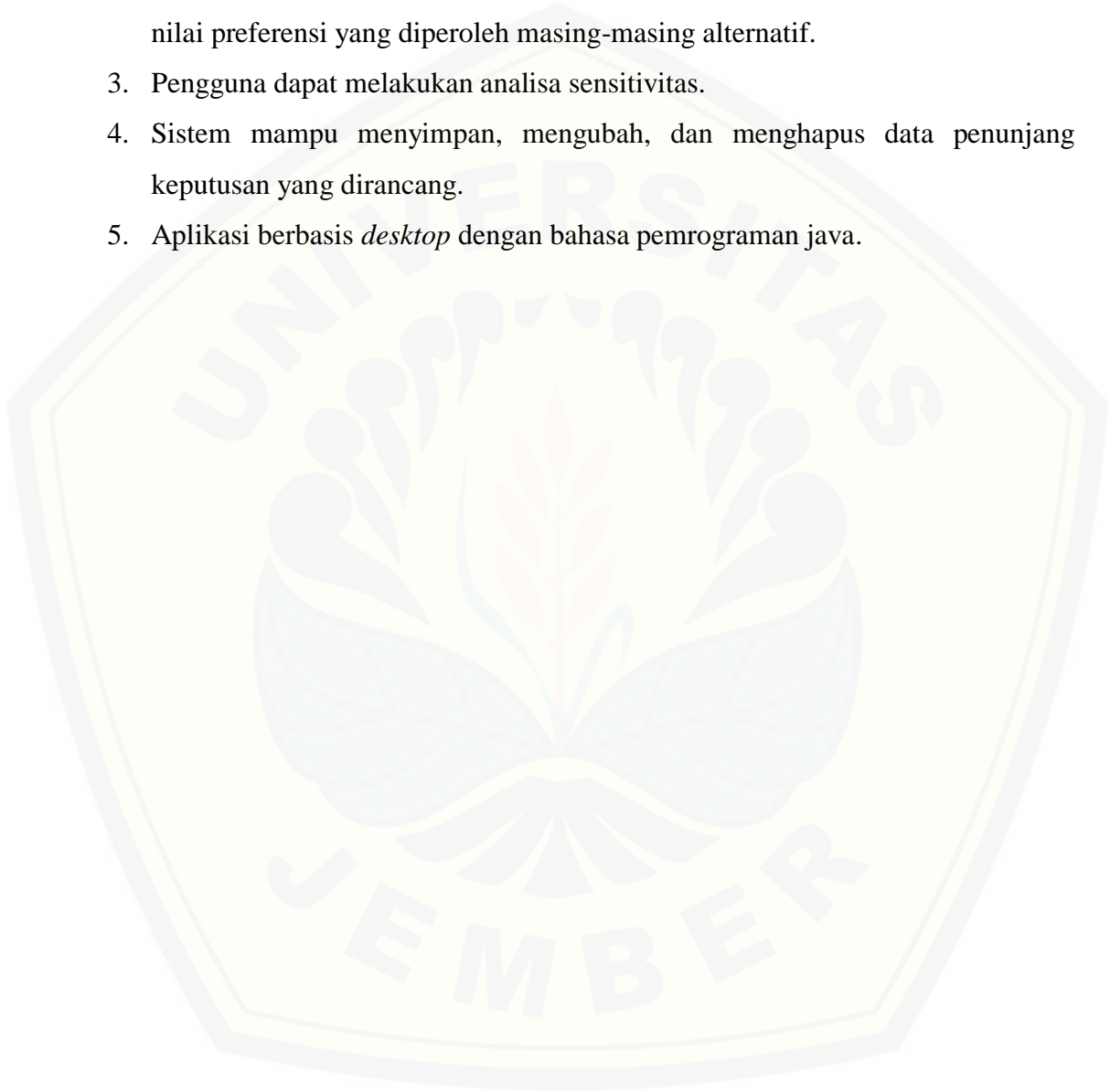
Penelitian kali ini memiliki tujuan sebagai berikut:

1. Merancang aplikasi yang dapat memandu pengguna untuk menerapkan algoritma *Simple Additive Weighting* dalam mendefinisikan pengambilan keputusan.
2. Membantu pengguna memperoleh rekomendasi keputusan dan menganalisanya.
3. Membantu pengguna menyimpan dan mengelola data model problem pengambilan keputusan dalam basis data.

#### 1.4 Batasan Masalah

Batasan masalah dari penelitian ini adalah:

1. Sistem hanya dapat digunakan untuk problem MADM dengan metode SAW.
2. Hasil akhir perhitungan oleh sistem berupa ranking keputusan berdasarkan nilai preferensi yang diperoleh masing-masing alternatif.
3. Pengguna dapat melakukan analisa sensitivitas.
4. Sistem mampu menyimpan, mengubah, dan menghapus data penunjang keputusan yang dirancang.
5. Aplikasi berbasis *desktop* dengan bahasa pemrograman java.



## BAB 2. TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai penelitian yang sebelumnya dilakukan untuk topic yang serupa. Selain itu teori yang diperoleh dari studi literatur juga dijelaskan sebagai dasar dari penelitian yang dilakukan.

### 3.1 Penelitian Sebelumnya

Penelitian mengenai penerapan metode MCDM ke dalam sebuah aplikasi telah dilakukan untuk beberapa metode MCDM. Salah satunya penelitian yang dilakukan oleh Ishizaka dan Labib (2009) mengenai penerapan algoritma *Analytic Hierarchy Process* (AHP) dalam aplikasi *Expert Choice*. Dalam jurnal tersebut mendiskusikan pemodelan problem, perbandingan berpasangan, skala pendapat, metode asal, indeks konsistensi, sintesis dari dua bobot, dan analisa sensitivitas. Jurnal ini menjelaskan implementasi metode AHP dalam aplikasi *Expert Choice* dengan menggunakan contoh permasalahan pemilihan mobil.

Peneliti menerangkan bahwa tantangan yang harus dihadapi oleh pengguna dalam menggunakan sistem dalam *Expert Choice* adalah proses pemilihan hirarki. Proses ini merupakan proses yang rumit namun sangat penting dan memiliki pengaruh besar pada hasil akhir. Disebutkan pula batasan metode AHP apabila dihadapkan dengan kriteria yang independen (tidak memiliki korelasi). Selain itu, karena kelemahan metode AHP adalah atribut yang digunakan terbatas, maka hal ini juga menjadi batasan dari sistem yang dirancang oleh mereka.

Penelitian kali ini, mengenai perancangan perangkat MADM yang menerapkan metode SAW, diharapkan aplikasi yang dihasilkan dapat memudahkan pengguna untuk merancang struktur model problem pemilihan keputusannya. Metode SAW dipilih selain karena tidak memiliki proses pemilihan



hirarki yang dinilai sulit dalam aplikasi *Expert Choice*, namun juga karena kesederhanaan dalam prosesnya yang memungkinkan pengguna akan lebih mudah menggunakan aplikasi pendukung keputusan. Selain itu, kesederhanaan proses juga diharapkan memiliki cakupan problem pengambilan keputusan yang lebih luas.

Penelitian lain yang berkaitan dengan metode SAW, yang dapat digunakan sebagai acuan bagi penelitian ini antara lain penelitian yang dilakukan oleh Lazim Abdullah dan Rabiatul Adawiyah (2014) telah melakukan penelitian tentang penggunaan metode SAW dan *Fuzzy SAW* (FSAW) dalam *multi-criteria decision making* selama periode 2003 hingga 2013. Penelitian ini menganalisa 19 jurnal yang menggunakan metode SAW dan FSAW. Sebanyak 52,63% dari jurnal-jurnal tersebut membahas meneliti mengenai aplikasi manajemen. Domain penelitian manajemen ini merupakan domain terbesar dalam penggunaan metode SAW dan FSAW. Mark Velasquez dan Patrick T. Hester (2013) melakukan analisis mengenai berbagai metode dalam MCDM. Kedua peneliti ini memaparkan *review*, menguji kelebihan dan kekurangan, dan menjelaskan bagaimana hubungan aplikasi dengan kekuatan serta kelemahan metode yang dibahas.

Selain dalam penelitian-penelitian diatas, metode SAW juga banyak digunakan untuk berbagai penelitian pengembangan sistem penunjang keputusan untuk masalah-masalah spesifik. Contohnya antara lain penelitian yang dilakukan oleh Afshari *et al.* (2010) yang menggunakan metode SAW untuk merancang sistem pendukung keputusan pemilihan personel dalam suatu organisasi; penelitian oleh Monica Louisa, seorang mahasiswa Fakultas Teknologi Informasi Jurusan Sistem informasi, Universitas Kristen Satya Wacana, yang menerapkan metode *simple additive weighting* pada penentuan tingkat kesejahteraan penduduk Propinsi Nusa Tenggara Timur; dll.

### 3.2 Sistem Penunjang Pengambilan Keputusan

Bisa juga disebut Sistem Pendukung Keputusan. Sistem penunjang pengambilan keputusan merupakan suatu ilmu dalam studi sistem informasi berbasis komputer yang membantu manusia atau pengguna untuk memperoleh rekomendasi keputusan yang berupa pilihan. Rohayani (2013) dalam jurnal penelitiannya mengutip pendapat Irfan Subakti yang menyebutkan bahwa sistem pendukung keputusan (*Decision Support System*) adalah suatu sistem interaktif berbasis komputer yang memanfaatkan data dan model untuk masalah semi terstruktur guna memperoleh rekomendasi keputusan. Dalam jurnal ini pula dikutip mengenai tujuan dari dibangunnya sistem pendukung keputusan, yaitu untuk membantu manajer mengambil keputusan, meningkatkan efektifitas pengambilan keputusan dan meningkatkan produktivitas.

Tzeng dan Huang (2011) mengutip beberapa rujukan mengenai proses umum dalam pengambilan keputusan. Dalam kutipannya menyebutkan bahwa proses pengambilan keputusan terdiri dari beberapa tahap penyelesaian, yaitu identifikasi problem, membangun preferensi, evaluasi alternatif, dan menentukan alternatif terbaik.

### 3.3 *Multi-Attribute Decision Making* (MADM)

*Multi-Attribute Decision Making* merupakan salah satu kategori dari *Multi-Criteria Decision Making*. Hwang (dalam Azar, 2000) berpendapat bahwa MADM merupakan pengambilan keputusan (seperti evaluasi, pemrioritasan, dan seleksi) dari alternatif yang tersedia yang dikarakteristikkan oleh banyak atribut yang biasanya terjadi konflik antar atribut. MADM juga dikenal sebagai cabang dari model *Operations Research* (OR) yang menangani masalah keputusan dari beberapa kriteria yang tersedia (Triantaphyllou *et al*, 1998).

Meskipun MADM memiliki metode penyelesaian yang bermacam-macam, tapi sebagian besar memiliki properti umum, yakni alternatif dan atribut (kriteria). Alternatif merepresentasikan pilihan aksi atau keputusan yang disediakan untuk



pembuat keputusan. Sedangkan kriteria/atribut merepresentasikan sudut pandang penilaian alternatif.

### 3.4 Simple Additive Weighting (SAW)

Algoritma *Simple Additive Weighting* merupakan salah satu algoritma yang sering digunakan dalam *Multi Attribute Decision Making* (MADM). Algoritma ini juga sering disebut sebagai penjumlahan berbobot karena ketergantungannya yang besar terhadap bobot yang diberikan pada setiap kriteria. Meskipun banyak digunakan untuk membantu membuat keputusan, algoritma ini memiliki nilai subjektifitas yang cukup tinggi.

Algoritma ini memiliki beberapa langkah dalam proses perhitungannya. Langkah-langkah tersebut antara lain :

1. Identifikasi problem.
2. Menentukan kriteria / atribut yang mempengaruhi pengambilan keputusan.
3. Menentukan bobot untuk masing-masing kriteria (pada penelitian kali ini proses pembobotan menggunakan perbandingan berpasangan).
4. Membuat matrik keputusan, yakni nilai tiap alternatif pada setiap keputusan.
5. Normalisasi matrik keputusan dengan formulasi :

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\max_i x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases} \quad \dots (1)$$

6. Menghitung nilai preferensi setiap alternatif dengan formulasi :

$$V_i = \sum_{j=1}^n w_j r_{ij} \quad \dots (2)$$

dengan:

V = nilai preferensi

w = bobot kriteria

r = nilai alternatif yang telah dinormalisasi

7. Mengambil alternatif dengan nilai preferensi terbesar sebagai rekomendasi keputusan.

Pada subbab sistem penunjang pengambilan keputusan telah dijelaskan tahapan proses pembuatan keputusan. Dalam SAW tahapan tersebut diimplementasikan melalui langkah-langkah penyelesaian diatas. Langkah nomor satu untuk identifikasi problem penunjang pengambilan keputusan. Langkah nomor dua dan tiga merupakan tahap membangun preferensi. Langkah nomor 4, 5, dan 6 merupakan tahap evaluasi terhadap alternatif. Sedangkan langkah ketujuh merupakan tahap pemilihan alternatif terbaik.

Peneliti menambahkan perhitungan analisa sensitivitas pada sistem yang akan dibangun. Perhitungan ini digunakan untuk mengetahui besar perubahan bobot kriteria dan nilai preferensi alternatif atas perubahan nilai bobot salah satu kriteria. analisa sensitivitas dapat memberikan kontribusi efektif dalam pembuatan keputusan yang akurat. Hal ini dikarenakan pengambilan keputusan dengan banyak atribut menggunakan data yang tidak stabil dan dapat berubah ewaktu-waktu (Memariani *et al*, 2009).

Formulasi yang digunakan untuk menghitung analisa sensitivitas bobot adalah formulasi yang dihasilkan dari penelitian yang dilakukan oleh Memariani *et al*. (2009) dalam jurnalnya. Formulasi tersebut ditunjukkan dalam persamaan (3). Peneliti menggunakan formulasi ini karena lebih mudah untuk diimplementasikan kedalam kode program dan telah diuji dan memiliki hasil yang sama dengan perhitungan biasa.

$$w'_j = \frac{1-w'_p}{1-w_p} \cdot w_j \quad \text{..... (3)}$$

Dimana :

$w'$  : bobot kriteria baru

$w$  : bobot awal

$j$  : indeks bobot yang dicari

$p$  : indeks bobot yang dirubah

Menggunakan persamaan (3) akan diperoleh vector bobot baru. Dengan bobot ini sensitifitas nilai preferensi alternatif dihitung menggunakan persamaan (2). Untuk mengetahui besar perubahan bobot kriteria dan nilai alternatif, akan dibandingkan nilai awal dengan nilai baru yang diperoleh dari perhitungan

sensitivitas. Analisa sensitivitas dilakukan dengan sedikit mengubah data input dengan tujuan untuk mencari akibat perubahan tersebut pada hasil akhir. Apabila ranking tidak berubah, maka ranking yang diperoleh dikatakan sehat.

### 3.5 Contoh Kasus Penggunaan Metode SAW

Pada subbab ini akan dijelaskan mengenai penggunaan metode SAW untuk menyelesaikan suatu kasus atau problem pengambilan keputusan. Salah satu problem yang akan digunakan sebagai contoh adalah problem pemilihan mahasiswa berprestasi. Berikut ini poin-poin implementasi algoritma SAW yang telah disebutkan pada subbab 3.4.

#### 1. Identifikasi problem

Pada tahap ini pembuat keputusan menentukan problem pengambilan keputusan yang akan dibuatkan sistemnya, dalam hal ini problem pemilihan mahasiswa berprestasi merupakan problem yang akan digunakan. Selain menentukan problem pengambilan keputusannya, pengambil keputusan pada tahap ini juga harus memperkirakan kriteria/atribut apa saja yang akan digunakan dan bagaimana penggunaan kriteria-kriteria tersebut dalam sistem penunjang keputusan.

#### 2. Menentukan kriteria

Kriteria yang digunakan dalam pemilihan mahasiswa berprestasi terdapat 3 (tiga) macam kriteria, yaitu (1) nilai IPK; (2) penghargaan bidang akademik dan non akademik; (3) keaktifan dalam organisasi. Ketiga kriteria ini akan dinilai secara berbeda, dimana kriteria pertama dinilai secara numerik dengan skala penilaian 0 hingga 4, kriteria kedua dinilai secara numerik, dan kriteria ketiga dinilai secara verbal. Tipe dari ketiga kriteria adalah benefit, yaitu semakin tinggi nilai pada kriteria tersebut maka akan semakin baik. Tabel 2.1 menunjukkan penilaian verbal dengan nilai numeriknya.

Tabel 2.1 Ketentuan penilaian verbal

Tingkat keaktifan	Score
Tidak aktif	1
Kurang aktif	2
Sedang	3
Cukup aktif	4
Aktif	5

### 3. Menentukan bobot

Proses pemberian bobot menggunakan matrik perbandingan berpasangan, yakni dengan membandingkan tingkat kepentingan antar satu kriteria dengan kriteria lain. Berikut proses pemberian bobot pada tiga kriteria yang dipakai dalam pemilihan mahasiswa berprestasi.

- a. Langkah pertama adalah menentukan nilai kepentingan antar kriteria dengan mengisi matrik perbandingan berpasangan.

Tabel 2.2 Matrik perbandingan berpasangan

	kr-1	kr-2	kr-3
kr-1	1,00	4,00	6,00
kr-2	0,25	1,00	3,00
kr-3	0,17	0,33	1,00
total kolom	1,42	5,33	10,00

Ket : kr-1 = nilai IPK;

kr-2 = penghargaan akademik dan non akademik;

kr-3 = keaktifan dalam organisasi.

- b. Setelah matrik terisi langkah selanjutnya adalah menormalisasi matrik perbandingan berpasangan dengan membagi nilai tiap sel dengan nilai total masing-masing kolom. Bobot kriteria diperoleh dengan mencari rata-rata pada masing-masing baris.

Tabel 2.3 Normalisasi matrik perbandingan berpasangan

	kr-1	kr-2	kr-3	rata-rata (bobot)
kr-1	0,706	0,750	0,600	0,685
kr-2	0,176	0,188	0,300	0,221
kr-3	0,118	0,063	0,100	0,093

- c. Apabila bobot telah ditemukan, dibutuhkan untuk mengetahui konsistensi bobot yang diperoleh. Konsistensi diketahui melalui *Consistency Ratio* (CR) dengan formulasi

$$CR = \frac{CI}{RI} \dots\dots\dots (4)$$

Dimana:

RI = *Random Index*, merupakan konstanta yang ditetapkan oleh beberapa peneliti. Pada penelitian kali ini yang digunakan adalah Saaty's RI yang tertera dalam Tabel 4.1.

CI = *Consistency Index*, yang diperoleh dengan formulasi

$$CI = \frac{\lambda_{max} - n}{n - 1} \dots\dots\dots (5)$$

Dengan n adalah jumlah kriteria, dan  $\lambda_{max}$  adalah rata-rata hasil pembagian vektor matrik perbandingan berpasangan dengan vektor bobot. Vektor matrik diperoleh dengan mengkalikan matrik perbandingan berpasangan yang belum dinormalisasi dengan vektor bobot.

1,00	4,00	6,00	x	0,685	=	2,13	/	0,685	=	3,11
0,25	1,00	3,00		0,221		0,67		0,221		3,04
0,17	0,33	1,00		0,093		0,28		0,093		3,01

$$\lambda_{max} = \frac{3,11 + 3,04 + 3,01}{3} = 3,054$$

$$CI = \frac{3,054 - 3}{3 - 1} = 0,027$$

$$CR = \frac{0,027}{0,52} = 0,052$$

- d. Melalui perhitungan diatas dapat diketahui bahwa bobot kriteria 'nilai IPK' adalah 0,685; bobot kriteria 'penghargaan akademik dan non akademik' adalah 0,221; sedangkan bobot kriteria 'keaktifan dalam organisasi' adalah 0.093. Apabila konsistensi rasio (CR) dapat diterima (tidak lebih dari 0,1) maka proses dapat dilanjutkan ke proses selanjutnya.

4. Penilaian alternatif dengan matrik keputusan

Mahasiswa yang akan diseleksi sebanyak 6 (enam) orang dan merupakan alternatif dari pengambilan keputusan ini. Masing-masing alternatif akan dinilai



pada masing-masing kriteria pada matrik keputusan seperti terlihat pada Tabel 2.4.

Tabel 2.4 Matrik Penilaian Alternatif

	<b>kr-1</b>	<b>kr-2</b>	<b>kr-3</b>
<b>Ana</b>	3,2	1	5
<b>Udin</b>	3,32	3	3
<b>Salma</b>	3,1	4	5
<b>Raka</b>	3,4	4	2
<b>Adi</b>	3,51	2	1
<b>Sulton</b>	3,3	6	2

#### 5. Normalisasi matrik keputusan

Matrik keputusan pada Tabel 2.4 selanjutnya akan dinormalisasi menggunakan formula pada persamaan (1). Karena kriteria yang digunakan pada pemilihan mahasiswa berprestasi ini seluruhnya bertipe benefit, maka persamaan pertama yang digunakan. Berikut hasil normalisasi tertera pada Tabel 2.5.

Tabel 2.5 Normalisasi matrik keputusan

	<b>kr-1</b>	<b>kr-2</b>	<b>kr-3</b>	<b>Nilai Preferensi</b>
<b>Ana</b>	0,970	4,000	0,833	1,628
<b>Udin</b>	1,006	1,333	0,500	1,031
<b>Salma</b>	0,939	1,000	0,833	0,943
<b>Raka</b>	1,030	1,000	0,333	0,959
<b>Adi</b>	1,064	2,000	0,167	1,187
<b>Sulton</b>	1,000	0,667	0,333	0,864

#### 6. Kalkulasi nilai preferensi alternatif

Nilai preferensi alternatif seperti tertera dalam Tabel 2.5 diperoleh dengan menjumlahkan seluruh hasil perkalian bobot kriteria dengan nilai alternatif pada kriteria tersebut.

#### 7. Menentukan rekomendasi keputusan

Rekomendasi keputusan adalah alternatif dengan nilai preferensi tertinggi. Berdasarkan Tabel 2.5, Ana merupakan kandidat terbaik untuk terpilih menjadi mahasiswa berprestasi, disusul Adi dan Udin.

#### 8. Analisa sensitivitas

Proses ini dilakukan untuk mengetahui kekuatan rekomendasi yang dihasilkan dengan merubah sedikit salah satu bobot kriteria yang digunakan. Seperti tertera pada Tabel 2.6, bobot kriteria pertama diubah menjadi 0,4 dimana bobot awalnya adalah 0,685. Dengan perubahan salah satu bobot ini, maka bobot kriteria lain juga akan ikut berubah. Menggunakan persamaan (3), bobot baru kriteria kedua dan ketiga akan diketahui.

$$w_2 = \left( \frac{1 - 0,4}{1 - 0,685} \right) \cdot 0,221 = 0,422$$

$$w_3 = \left( \frac{1 - 0,4}{1 - 0,685} \right) \cdot 0,093 = 0,178$$

Vektor bobot baru yang diperoleh akan digunakan untuk menghitung nilai preferensi alternatif baru seperti tertera paada Tabel 2.6.

Tabel 2.6 Nilai preferensi baru

	kr-1	kr-2	kr-3			Nilai Preferensi Baru
<b>Ana</b>	0,970	4,000	0,833			2,224
<b>Udin</b>	1,006	1,333	0,500		0,400	1,054
<b>Salma</b>	0,939	1,000	0,833	x	0,422	0,946
<b>Raka</b>	1,030	1,000	0,333		0,178	0,893
<b>Adi</b>	1,064	2,000	0,167			1,299
<b>Sulton</b>	1,000	0,667	0,333			0,741

Berdasarkan Tabel 2.6 dapat dilihat bahwa Ana masih menjadi alternatif terbaik, disusul Adi dan Udin. Dengan demikian rekomendasi yang dihasilkan sebelumnya bersifat kuat / sehat.

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

Bab ini menjelaskan tentang tahapan penelitian beserta metode-metode yang digunakan untuk mencapai tujuan dari penelitian. Terdapat 5 (lima) subbab dalam bab ini yang menjelaskan metodologi penelitian, yaitu jenis problem, tahapan penelitian, pengumpulan data, tahap analisa, dan tahap pengembangan sistem.

#### **3.1 Jenis Penelitian**

Berdasarkan Buku Pedoman Penulisan Karya Ilmiah Universitas Jember, jenis penelitian merupakan penegasan mengenai kategori penelitian yang akan dilakukan. Apabila dilihat dari sudut pandang jenis data yang digunakan, Penelitian kali ini merupakan penelitian bersifat kualitatif, yakni suatu penelitian yang menelaah secara deskriptif dengan tujuan untuk eksplorasi objek yang diteliti. Sedangkan apabila dilihat menurut ada/tidaknya perlakuan, penelitian ini merupakan penelitian observasional, yaitu menganalisa teori dan merancang sistem dari hasil analisa yang diperoleh.

Menurut sifatnya, penelitian bersifat rekayasa, yakni suatu penelitian yang menerapkan teori ilmu pengetahuan yang telah ada menjadi suatu rancangan yang bekerja sesuai dengan persyaratan tertentu. Dalam hal ini, teori yang akan diterapkan adalah teori mengenai sistem penunjang pengambilan keputusan dan pembuatan keputusan dengan banyak atribut. Peneliti mengeksplorasi teori dan membuat suatu rancangan sistem dengan batasan-batasan tertentu sesuai dengan kebutuhan.

#### **3.2 Tahapan Penelitian**

Penelitian ini diawali dengan identifikasi masalah yang akan diteliti. Dilanjutkan dengan pengumpulan data. Data yang dibutuhkan berupa berbagai macam kajian teori menyangkut topik bahasan, antara lain: jenis-jenis tipe data,



kategori dalam pengambilan keputusan, macam-macam pengambilan keputusan, dan lain sebagainya. Tidak semua data yang diperoleh akan diimplementasikan dalam rancangan sistem yang diteliti, melainkan dipilih sebagian data yang dapat diterapkan dalam rancangan. Tahap selanjutnya adalah perumusan masalah dan memberikan batasan terhadap masalah yang dirumuskan.

Apabila data yang dibutuhkan telah terkumpul, tahap selanjutnya adalah menganalisa sistem, lalu merancang sistem berdasarkan analisa yang dilakukan. Rancangan sistem nantinya akan diubah kedalam bentuk bahasa pemrograman untuk diperoleh sistem yang utuh.

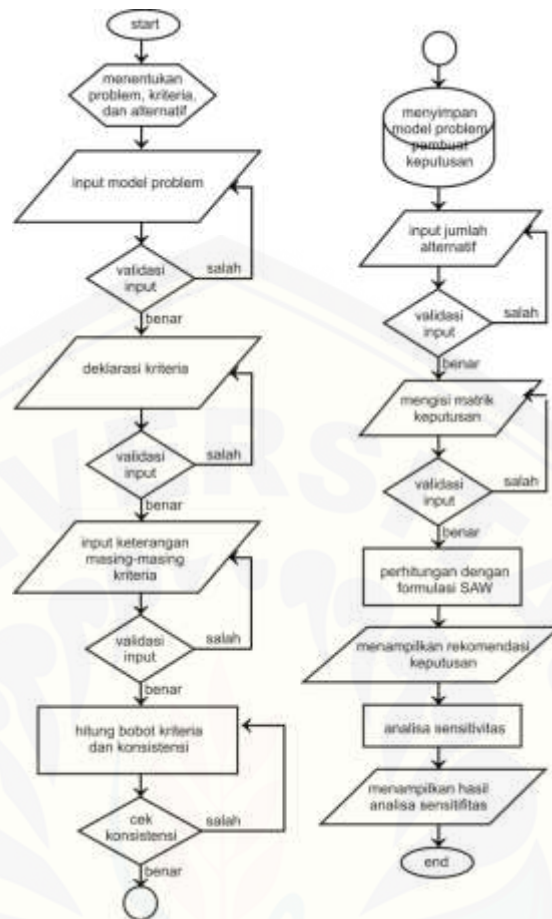
### **3.3 Teknik Pengumpulan Data**

Pengumpulan data dilakukan dengan studi literatur dari berbagai kajian yang telah dilakukan sebelumnya. Sebagian besar literatur yang digunakan sebagai rujukan dalam penelitian kali ini berupa jurnal-jurnal dengan bahasan yang disesuaikan dengan topik penelitian. Data lain yang digunakan pada penelitian ini adalah data sampel perhitungan dengan metode SAW untuk mengetahui apakah perhitungan yang dilakukan oleh sistem telah benar. Data sample ini diperoleh dari penelitian lain yang menggunakan metode SAW dalam perhitungannya. Selain itu juga menggunakan data hitung manual yang dilakukan oleh peneliti.

### **3.4 Tahap Analisis**

Tahap ini dilakukan setelah data yang dibutuhkan terkumpul. Data yang diperoleh selanjutnya akan dijadikan dasar analisa sistem yang akan dibangun. *Flow chart* hasil analisa sistem yang akan diimplementasikan dalam rancangan sistem terdapat dalam Gambar 3.1. gambar tersebut menunjukkan garis besar proses atau langkah-langkah dalam sistem yang akan dibangun.

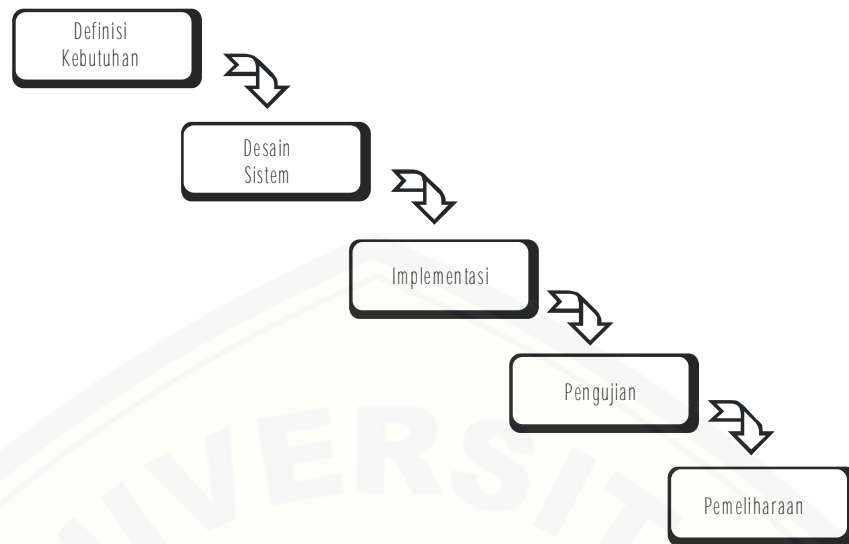
Gambar 3.1 menunjukkan langkah-langkah apa saja yang dilakukan pengguna di dalam sistem dan apa saja yang harus diketahui oleh pengguna untuk dapat menjalankan sistem dengan benar. Selain itu, Gambar 3.1 menunjukkan juga proses utama yang dilakukan oleh sistem atas tindakan yang diberikan oleh pengguna.

Gambar 3.1 *Flow chart* sistem

### 3.5 Tahap Pengembangan Sistem

Model pengembangan sistem yang dipakai untuk merancang sistem ini adalah model pengembangan metode *waterfall*. Tahapan perancangan dalam model ini antara lain, definisi kebutuhan, desain, implementasi, pengujian, dan pemeliharaan. Gambaran mengenai model ini diterangkan dalam Gambar 3.2.

Gambar 3.2 menunjukkan alur pengembangan sistem. Terdapat 6 (enam) tahapan dalam pengembangan sistem menurut model pengembangan *waterfall* ini. Tahap pertama adalah definisi kebutuhan, dimana pengembang menentukan apa saja yang dibutuhkan. Dilanjutkan dengan desain sistem, yakni menggambarkan sistem dengan diagram-diagram. Tahap ketiga adalah implementasi desain kedalam sebuah kode program. Program yang telah dibangun akan diuji dan diukur dalam tahap pengujian. Tahap terakhir adalah pemeliharaan sistem.



Gambar 3.2 Tahapan model *waterfall*

Pada penulisan ini akan disampaikan penjelasan dari tahap definisi kebutuhan hingga tahap pengujian. Tahap pemeliharaan merupakan tahap dimana sistem dijaga kemutakhiran fitur yang terdapat didalamnya, dan merupakan proses yang dilakukan sejak selesainya pembangunan sistem hingga saat yang tidak ditentukan selama peneliti merasa butuh untuk melakukan pembaharuan terhadap sistem

## BAB 5. HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil dari penelitian yang dilakukan yang berupa perangkat lunak untuk merancang dan menggunakan sistem penunjang keputusan dengan metode SAW. Selain memaparkan hasil penelitian, dalam bab ini juga dijelaskan mengenai implementasi metode yang digunakan di dalam perangkat lunak yang dihasilkan. Terdapat pula pembahasan mengenai fitur-fitur utama perangkat lunak sistem.

### 5.1 Perangkat Lunak MADM Menggunakan Metode SAW

Berdasarkan desain sistem yang telah dijelaskan pada bab sebelumnya, sistem ini memiliki satu aktor yang mempunyai hak akses penuh terhadap sistem, yaitu *user*. Dengan kata lain siapapun memiliki hak akses terhadap fitur-fitur yang terdapat dalam sistem.

Tampilan awal aplikasi seperti tertera dalam Gambar 5.1. Gambar 5.1 menunjukkan dua pilihan penggunaan perangkat. Tombol Gunakan Data Tersimpan merupakan pilihan untuk mengelola data sistem penunjang keputusan dimana struktur model problem telah diinisiasi dan disimpan dalam basis data. Pengelolaan ini mencakup lihat detail struktur model problem, menggunakan struktur model untuk mencari ranking dari alternatif keputusan, mengubah struktur model problem, dan menghapus struktur model problem. Tampilan jendela daftar data yang tersimpan jika *user* memilih tombol Gunakan Data Tersimpan tertera dalam Gambar 5.2. Penyimpanan data penunjang keputusan didalam basis data.



Gambar 5.1 Tampilan awal aplikasi



Gambar 5.2 Tampilan daftar data struktur model problem penunjang keputusan

Melalui tampilan di Gambar 5.2 *user* dapat memilih aksi pada salah satu data dalam daftar dengan mengklik sel data yang diinginkan. Sistem akan menampilkan *option dialog* dimana terdapat 4 (empat) pilihan aksi, yaitu lihat detail, gunakan, ubah, dan hapus. Tombol Selesai digunakan apabila *user* ingin kembali ke jendela awal Gambar 5.1.

Tabel data penunjang keputusan hanya menampilkan nama problem penunjang keputusan dan pembuat keputusan. Apabila *user* ingin mengetahui detail dari salah satu data, *user* memilih (klik) salah satu data, dan pilih lihat detail. Dengan melihat detail, maka *user* akan mengetahui informasi penting dari



salah satu penunjang keputusan tersebut. Melihat detail penunjang keputusan akan sangat membantu user apabila data penunjang keputusan yang tersimpan telah banyak.

Pengelolaan dapat membuat data tersimpan disesuaikan dengan perubahan-perubahan kondisi dari penunjang keputusan yang mungkin terjadi setelah user menginisiasi penunjang keputusan. Hal ini memudahkan pengguna apabila memiliki problem pemilihan keputusan yang dinamis.

## 5.2 Implementasi Metode SAW

Perancangan penunjang keputusan dalam perangkat lunak sistem ini dibagi menjadi dua kategori tahapan, yaitu inisiasi model struktur problem dan perhitungan preferensi. Untuk memperoleh satu sistem penunjang keputusan, user harus mengisi beberapa *form* inisiasi, seperti tertera dalam Gambar 5.3, Gambar 5.4, Gambar 5.5, Gambar 5.6 dan Gambar 5.7. Berdasarkan gambar-gambar tersebut, dapat diketahui data apa saja yang harus diketahui oleh *user* untuk menggunakan perangkat ini.

The screenshot shows a software window titled "preSAW - Pemodelan Problem". The interface is divided into a left sidebar and a main content area. The sidebar contains a menu with the following items: "Model Problem", "Definisi Masalah", "Perbandingan Berpasangan", "Indeks Konsistensi", "Perbandingan", "Penilaian Alternatif", and "Ranking Preferensi". The main content area is titled "Pemodelan Problem" and contains a form with the following fields and values:

- Problem:** Pemilihan mahasiswa beasiswa
- Keterangan Problem:** merupakan penunjang keputusan untuk memilih mahasiswa beasiswa berdasarkan kriteria yang ditentukan
- Jumlah Kriteria:** 6
- Pembuat Keputusan:** bagian akademik

On the right side of the main content area, there is a text box with the following text:

Pemodelan problem merupakan tahap awal berisi informasi dasar mengenai masalah keputusan.

- Dalam "Problem" diisi dengan info nama pembuat keputusan.
- Dalam "Keterangan Problem" diisi dengan detail problem.
- Dalam "Jumlah Kriteria" diisi dengan angka jumlah kriteria.

\*PERINGAT! Semakin besar jumlah kriteria, maka semakin besar pula sumber data yang dibutuhkan. Inisiasi pada tahap selanjutnya juga akan semakin sulit.

At the bottom of the window, there are three buttons: "Kembali" (grey), "Selanjutnya" (orange), and "Batalkan" (green).

Gambar 5.3 Tampilan jendela pemodelan problem

Melalui Gambar 5.3 dapat diketahui bahwa terdapat 4 (empat) informasi yang perlu diinputkan oleh *user*. Dari keempat informasi tersebut, informasi yang

paling penting disbanding dengan yang lain adalah informasi mengenai nama problem yang diinputkan ke kolom Problem, dan informasi mengenai jumlah kriteria yang diinputkan ke kolom Jumlah Kriteria. Meskipun demikian, *user* harus mengisi seluruh kolom isian agar dapat melanjutkan ke proses selanjutnya.

No.	Kriteria	Tipe Penilaian
1	IPK	
2	Memiliki catatan pelanggaran	Yes / No
3	persentase kehadiran	Numerik
4	total SKS yang telah ditempuh	Numerik
5	Koorganisasian	Verbal
6	Nilai perilaku	Verbal

Tentukan kriteria yang akan digunakan dalam membuat keputusan.  
 Kolom Tipe Penilaian diisi dengan tipe nilai yang akan digunakan pada masing-masing kriteria. Tipe "Numerik" digunakan apabila kriteria memiliki nilai angka. Tipe "verbal" digunakan apabila kriteria memiliki nilai verbal. Sedangkan tipe "Yes / No" digunakan apabila kriteria hanya akan diisi dengan dua pilihan tersebut (Ya dan Tidak).  
 Pilih tipe penilaian yang paling sesuai dengan kebutuhan Anda.

Sebelumnya Selanjutnya Batalkan

Gambar 5.4 Tampilan *form* isian definisi kriteria (1)

No.	Kriteria	Tipe Kriteria	Skala Nilai
1	IPK	Benefit	Numerik
2	Memiliki catatan pelanggaran	Cost	Ya-Tidak
3	persentase kehadiran	Benefit	Numerik
4	total SKS yang telah ditempuh	Benefit	Numerik
5	Koorganisasian	Benefit	Rendah - Tinggi
6	Nilai perilaku	Benefit	Baik - Buruk

Lebih lanjut mengenai detail kriteria yang akan digunakan:  
 Kolom "Tipe Kriteria" merupakan informasi jenis masing-masing kriteria. "Benefit" menunjukkan nilai kriteria yang semakin besar adalah yang lebih baik, sedangkan "Cost" menunjukkan sebaliknya, yaitu semakin kecil nilai pada kriteria tersebut, semakin akan semakin baik.  
 Kolom "Skala Nilai" menunjukkan informasi skala penilaian yang akan digunakan pada masing-masing kriteria. Pilihan yang terdapat dalam daftar berikut berkaitan dari Tipe Penilaian yang...

Sebelumnya Selanjutnya Batalkan

Gambar 5.5 Tampilan *form* isian definisi kriteria (2)

Kedua gambar diatas, Gambar 5.4 dan Gambar 5.5, menunjukkan terdapat dua *form* isian untuk mendefinisikan atribut/kriteria yang harus diisi oleh *user*. Form isian pertama untuk menentukan nama kriteria dan tipe penilaian untuk kriteria tersebut. Pemilihan tipe penilaian disesuaikan dengan kriteria. Sedangkan form isian kedua adalah untuk menentukan tipe kriteria (benefit atau cost) dan menentukan skala penilaian yang cocok untuk kriteria bersangkutan.

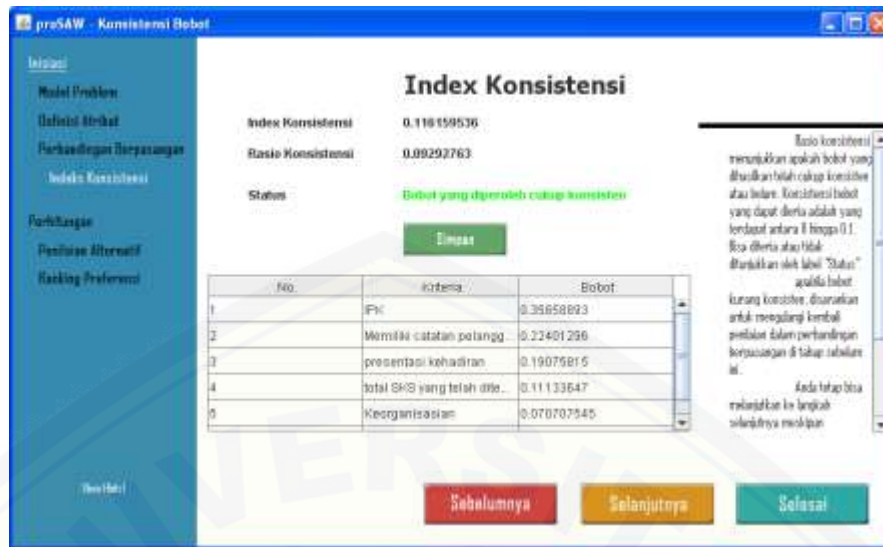
<Numerik>	K1-1	K1-2	K1-3	K1-4	K1-5	K1-6
K1-1	1	3	4	2	7	6
K1-2	0.33	1	2	3	5	3
K1-3	0.25	0.5	1	4	3	5
K1-4	0.5	0.33	0.25	1	2	3
K1-5	0.33	0.2	0.33	0.5	1	2
K1-6	0.16	0.33	0.2	0.33	0.5	1

dalam tabel berikut. Tabel ini digunakan untuk mencari bobot untuk setiap kriteria. Isi sel dalam tabel yang berada diatas diagonal angka 1. Isi dengan nilai tingkat kepentingan tiap kriteria dalam header vertikal (sebelah kiri) jika dibanding dengan masing-masing kriteria dalam header horizontal (atas).  
 Keterangan kriteria sebagai berikut:  
 K1-01 : PK  
 K1-11 : Memiliki catatan pelanggaran  
 K1-21 : presentasi kehadiran

Gambar 5.6 Tampilan *form* perbandingan berpasangan

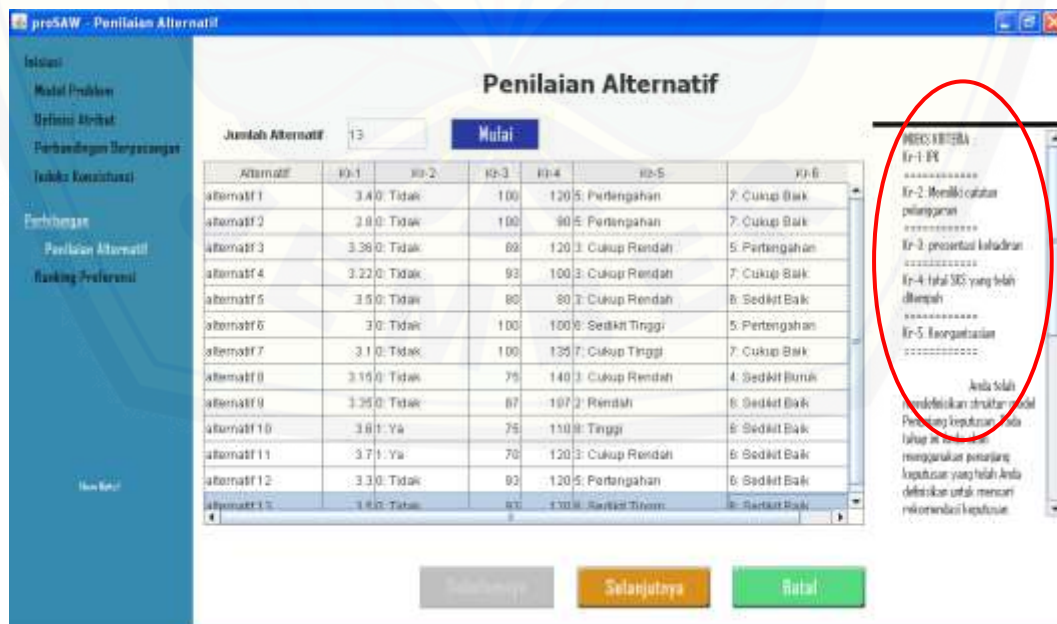
Terdapat 3 (tiga) pilihan input perbandingan berpasangan, yakni manual, numerik dan verbal input. Numerik dan verbal input, pilihan nilai disediakan oleh sistem. Sedangkan manual input nilai diinputkan oleh user melalui ketikan keyboard dengan hanya karakter numerik yang dapat diinputkan dengan bilangan riil. Pada Gambar 5.6 diatas menggunakan input numerik.





Gambar 5.7 Tampilan indeks konsistensi bobot dan bobot yang diperoleh

Pengguna dapat menyimpan struktur model problem yang telah diinisiasi apabila konsistensi bobot dapat diterima. Status penerimaan konsistensi bobot dapat dilihat di kolom Status yang terdapat diatas tombol Simpan. Sedangkan tabel dalam tampilan menunjukkan bobot yang diperoleh dari perbandingan berpasangan yang dilakukan sebelumnya.



Gambar 5.8 Tampilan penilaian alternatif

Gambar 5.8 diatas menunjukkan proses penilaian alternatif. Jumlah alternatif yang dipertimbangkan diinputkan di kolom isian jumlah alternatif yang terdapat di atas tabel isian. Dengan klik tombol mulai, sistem akan menampilkan tabel matrik keputusan dengan format isian pada tiap kolom tabel sesuai dengan tipe penilaian masing-masing kriteria. Keterangan nama kriteria terdapat dalam *text area* di sebelah kanan tabel.

The screenshot shows a software window titled 'preSAW - Rekomendasi Keputusan(Ranking Alternatif)'. The main area is titled 'Rekomendasi Keputusan' and contains a table with the following data:

Ranking	Alternatif	Nilai
1	alternatif 5	0.69195746
2	alternatif 13	0.6529734
3	alternatif 12	0.844135
4	alternatif 4	0.63282895
5	alternatif 10	0.62825966
6	alternatif 3	0.59889185
7	alternatif 9	0.59642434
8	alternatif 11	0.58406746
9	alternatif 8	0.5713257
10	alternatif 7	0.49862003
11	alternatif 1	0.4809432
12	alternatif 6	0.47703887
13	alternatif 2	0.46037882

Below the table are three buttons: 'Sebelumnya', 'Selesai', and 'Sensitivitas'. To the right of the table is a text area with the following text:

Tabel menunjukkan hasil perhitungan. Alternatif dengan nilai terbesar merupakan alternatif terbaik menurut perhitungan yang dilakukan. Kriteria yang ditandingkan tidak disesuaikan urutannya berdasarkan besar nilai hasil yang diperoleh oleh alternatif tersebut. Lakukan analisa sensitivitas untuk mengetahui perubahan vektor bobot dan vektor alternatif apabila salah satu bobot kriteria diubah.

Gambar 5.9 Tampilan jendela ranking alternatif

Setelah keseluruhan proses yang telah dilakukan, hasil akhirnya berupa ranking alternatif keputusan. Berdasarkan contoh di Gambar 5.9, alternatif dengan nilai tertinggi adalah alternatif 5 dengan nilai 0,69 disusul dengan alternatif 13 pada urutan kedua dengan nilai 0,65.

Tampilan pada Gambar 5.9 merupakan akhir dari proses penentuan alternatif keputusan. *User* dapat menghentikan proses pada tahap ini atau melanjutkan proses dengan melakukan analisa sensitivitas dengan mengklik tombol Sensitivitas. Gambar 5.10 merupakan tampilan jendela analisa sensitivitas.

Analisa sensitivitas digunakan untuk mengetahui besar perubahan vektor bobot dan preferensi alternatif oleh perubahan salah satu bobot kriteria.



Gambar 5.10 Tampilan analisa sensitivitas

Terdapat dua sensitivitas yang dapat dihitung oleh pengguna, yaitu sensitivitas bobot dan sensitivitas alternatif. Sebelum menampilkan grafik sensitivitas, *user* harus mengisi isian yang terdapat dibawah grafik seperti dalam Gambar 5.10. Pada contoh diatas, bobot kriteria IPK yang sebelumnya sebesar 0,4 diubah menjadi 0,2. Dengan mengklik tombol Sensitivitas, grafik perubahan bobot dan nilai alternatif (nilai sebelumnya dan nilai baru) akan ditampilkan. Terdapat dua pilihan tampilan grafik yang dapat dipilih oleh *user*, yaitu grafik garis dan grafik batang.

### 5.3 Pembahasan

Subbab ini memberikan penjabaran mengenai kesesuaian tujuan penelitian dengan desain dan hasil akhir penelitian yaitu perangkat lunak MADM yang mengimplementasikan metode SAW. Konsentrasi pembahasan dalam subbab ini adalah implementasi SAW dalam desain dan penulisan kode program aplikasi yang dibangun.

### 5.3.1 Implementasi Metode SAW dalam Desain Sistem

Desain sistem terdiri dari beberapa diagram seperti tertera dalam bab Analisis dan Pengembangan Sistem. Pada *use case diagram*, Gambar 4.2, dapat diketahui implementasi metode SAW terdapat dalam proses Perhitungan dengan SAW dimana memiliki dua proses dalam *usecase* pada Gambar 4.2, yaitu penilaian alternatif dan lihat ranking alternatif. Dari kedua proses tersebut terdapat proses perhitungan pencarian nilai preferensi alternatif dengan formulasi SAW, yaitu penjumlahan terbobot seperti tertera pada persamaan 2.

*Sequence diagram* yang menggambarkan proses perhitungan nilai preferensi alternatif terdapat dalam Lampiran A.2 nomor satu. Dalam diagram tersebut tertulis pesan untuk melakukan normalisasi matrik keputusan dan menghitung penjumlahan terbobot dari matrik yang telah dinormalisasi.

Pada *class diagram*, Gambar 4.11 yang menunjukkan elasi antar class dalam sistem dan Lampiran A.4 yang menunjukkan atribut dan operasi yang terdapat pada masing-masing class sistem, dapat diketahui terdapat dua *class* yang menangani perhitungan nilai preferensi alternatif. Kedua class tersebut adalah *class* matrikKeputusan.java dan *class* penjumlahanTerbobot.java.

### 5.3.2 Implementasi metode SAW dalam Kode program Aplikasi

Kode program menunjukkan implementasi yang lebih detail mengenai implementasi metode SAW dalam sistem. Detail yang dimaksud adalah dapat diketahui melalui urutan eksekusi program kesesuaian dengan algoritma SAW. Tabel 5.1 menunjukkan kode program class penjumlahanTerbobot.java yang terdapat dalam *package* model. Sedangkan Tabel 5.2 adalah kode program class controllerPenilaianAlternatif.java yang mengatur keseluruhan urutan eksekusi program untuk menentukan nilai preferensi alternatif.



Tabel 5.1 Kode program *class* penjumlahanTerbobot.java

```

1 package model;
2
3 public class penjumlahanTerbobot extends modelProblem{
4     private float [] hasil;
5
6     public void setPenjumlahanTerbotot (float [] bobot){
7         int jmlAlternatif = modelProblem.getJmlAlternatif();
8         byte jmlKriteria = modelProblem.getJmlKriteria();
9         float [][] matrikKepNormal = modelProblem.getMatrikKepNormal();
10
11         float [] bobots;
12         if (bobot == null){
13             bobots = modelProblem.getBobot();
14         }else{
15             bobots = bobot;
16         }
17         hasil = new float [jmlAlternatif];
18
19         byte baris=0, kolom;
20         while (baris<jmlAlternatif){
21             kolom=0;
22             hasil[baris]=0;
23             while (kolom<jmlKriteria){
24                 hasil[baris]=hasil[baris]+(bobots[kolom]
25                     * matrikKepNormal [baris][kolom]);
26                 kolom++;
27             }
28             baris++;
29         }
30     }
31     public float [] getPreferensi (){
32         return hasil;
33     }
34 }

```

Tabel 5.1 diatas menunjukkan implementasi SAW dalam aplikasi sistem, yakni terdapat dalam kode program baris ke 24 hingga 25. Kode program pada baris tersebut mengimplementasikan persamaan (2) untuk menentukan nilai preferensi alternatif.

Tabel 5.2 Implementasi SAW dalam *class* controllerPenilaianAlternatif.java

```

1     public void cekInputanKosong (){
2         if ( view.getAlternatif() != null ||
3             view.getMatrikKeputusan() != null){
4             error = false;
5         }else{
6             error = true;
7             JOptionPane.showMessageDialog(view, "Isian kurang lengkap! "

```

```

8      +". \nPindah seleksi bila telah diisi.", "KESALAHAN",
9      JOptionPane.INFORMATION_MESSAGE);
10     view.jTable1.requestFocus();
11     }
12 }
13
14 class listenerTombolSelanjutnya implements ActionListener {
15
16     public void actionPerformed(ActionEvent e) {
17         cekInputanKosong();
18
19         if (error == false){
20             boolean [] tipekuantitatif = new boolean [modelProblem.getJmlKriteria()];
21             byte i, j;
22             for (i=0;i<modelProblem.getJmlKriteria();i++){
23                 if(modelProblem.getTipeNilaiKriteria()[i].equalsIgnoreCase("kuantitatif")){
24                     tipekuantitatif [i]=true;
25                 }else{
26                     tipekuantitatif[i]=false;
27                 }
28             }
29             buatBaruSAW1.setTipe(tipekuantitatif);
30
31             modelProblem.setPenjumlahanTerbobot
32                 (view.getMatrikKeputusan(), view.getAlternatif());
33
34             normalisasi.setNormalisasiMatrikKeputusan();
35
36             hitung.setPenjumlahanTerbobot(null);
37             model.setRankingAlternatif(hitung.getPreferensi());
38             String [] header = {"Ranking", "Alternatif", "Nilai"};
39             Object [][] data = new Object [modelProblem.getJmlAlternatif()]
40 [modelProblem.getJmlKriteria()];
41             i =0;
42             setSortedPreference(view.getAlternatif(), modelProblem.getRankingAlternatif());
43             while (i<modelProblem.getJmlAlternatif()){
44                 data [i][0]=i+1;
45                 data [i][1]=getRankingAlt()[i];
46                 data [i][2]=getRanking()[i];
47                 i++;
48             }
49             DefaultTableModel md = new DefaultTableModel(data, header){
50                 @Override
51                 public boolean isCellEditable (int rowIndex, int columnIndex){
52                     return false;
53                 }
54             };
55
56             final JTable table = new JTable (md);
57             table.getTableHeader().setReorderingAllowed(false);
58             table.setEditingRow(data.length+2);
59             table.getColumnModel().getColumn(0).setPreferredWidth(20);
60             table.getColumnModel().getColumn(1).setPreferredWidth(300);

```



```
61 table.getColumnModel().getColumn(2).setPreferredWidth(30);
62 //=====
63 hitung.setPenjumlahanTerbotot(null);
64 model.setRankingAlternatif(hitung.getPreferensi());
65 //=====
66 viewSelanjutnya.setTable(table);
67 viewSelanjutnya.setVisibleHint(model.getStatHint());
68 viewSelanjutnya.setLocation(view.getLocation());
69 viewSelanjutnya.setExtendedState(view.getExtendedState());
70 viewSelanjutnya.setVisible(true);
71 view.setVisible(false);
72 }else{
73     JOptionPane.showMessageDialog(view, "Kesalahan tidak diketahui",
74         "KESALAHAN", JOptionPane.WARNING_MESSAGE);
75 }
76 }
77 }
78
79 private float [] hasil;
80 private void setSortedPreference (String [] namaAlt, float [] nilai){
81     float [] ranking = new float [model.getJmlAlternatif()];
82     hasil = nilai;
83     rankingAlt = namaAlt;
84
85     int index;
86     int smallestIndex;
87     int minIndex;
88     float temp;
89     String tempAlt;
90
91     for (index = 0; index < ranking.length - 1; index++) {
92         smallestIndex = index;
93         for (minIndex = index + 1;
94             minIndex < ranking.length; minIndex++) {
95             if (hasil[minIndex] > hasil[smallestIndex]) {
96                 smallestIndex = minIndex;
97             }
98         }
99
100         temp = hasil[smallestIndex];
101         tempAlt = rankingAlt[smallestIndex];
102
103         hasil[smallestIndex] = hasil[index];
104         rankingAlt [smallestIndex] = rankingAlt [index];
105
106         hasil[index] = temp;
107         rankingAlt [index] = tempAlt;
108     }
109 }
110 }
111 private String [] getRankingAlt (){
112     return rankingAlt;
113 }
```

114	private float [] getRanking (){
115	return hasil;
116	}
117	}

Melalui Tabel 5.2 dapat diketahui langkah langkah penyelesaian perhitungan nilai preferensi hingga mnyajikan ranking alternatif dalam sebuah tabel. Sebelum melakukan perhitungan sistem melakukan pengecekan terhadap kelengkapan input user. Pengecekan inputuan ini terdapat dalam kode baris 1 – 12. Setelah menyimpan nilai yang diinputkan user kedalam variabel, sistem melakukan perhitungan seperti tertera dalam kode baris 34 – 37. Hasil dari perhitungan (nilai preferensi alternatif) dimasukkan dalam variabel array dalam class modelProblem.java (kode baris 37). Setelah nilai masing-masing alternatif diketahui, selanjutnya sistem melakukan pengurutan alternatif berdasarkan nilai yang diperoleh. Proses ini dilakukan oleh kode program baris ke 80 – 110 *method* setSortedPreference (). *Method* tersebut dipanggil pada kode program baris 42 sebelum pembuatan tabel untuk menampilkan ranking alternatif.

*User interface* implementasi SAW dalam aplikasi sistem terdapat dalam Gambar 5.8 dan Gambar 5.9. Gambar 5.8 merupakan penilaian alternatif dengan mengisi matrik keputusan. Nilai yang diinputkan tersebut diproses seperti dijelaskan pada paragraf diatas. Tabel yang berisi ranking alternatif akan ditampilkan dalam tampilan pada Gambar 5.9.

#### 5.4 Pengujian Perhitungan Sistem

Pengujian ini ditujukan untuk mengetahui ketepatan perhitungan aplikasi sistem. Data perhitungan yng digunakan adalah manual perhitungan oleh peneliti dan data perbandingan dari jurnal milik Afshari (2010). Untuk data yang diambil dari jurnal lain, peneliti akan membuat data baru dengan informasi yang diperoleh dari jurnal yang digunakan sebagai data sampel. Setelah hasil diperoleh, akan dicocokkan dengan hasil perhitungan yang terdapat dalam jurnal yang dirujuk. Peneliti hanya menggunakan data perbandingan berpasangan dari jurnal yang diteriti oleh Rubiyatun *et al.*, karena data matrik keputusan tidak dicantumkan secara keseluruhan oleh penulis.

Pengujian pertama menggunakan data perhitungan manual yang dilakukan oleh peneliti. Perhitungan yang akan diuji ketepatannya adalah perhitungan bobot, consistency index, consistency ratio, nilai preferensi, dan analisa sensitivitas. Hasil perhitungan manual perbandingan berpasangan beserta konsistensi bobot seperti tertera dalam Tabel 5.3. Tabel 5.3 menunjukkan hasil nilai preferensi perhitungan manual. Sedangkan untuk hasil perhitungan menggunakan aplikasi sistem tertera pada Gambar 5.11 dan Gambar 5.12. Peneliti menggunakan random value dalam mengisi matrik perbandingan berpasangan maupun matrik keputusan. Hal ini dilakukan untuk mengetahui apakah sistem telah berjalan sesuai dengan yang diharapkan atau belum.

Tabel 5.3 Hasil perhitungan manual bobot kriteria dan konsistensinya

Kriteria	Bobot	Consistency Index (CI)	Consistency Ratio (CR)	Status
kriteria1	0.459	0,180	0,202	Bobor tidak konsisten
kriteria2	0.313			
kriteria3	0.162			
kriteria4	0.066			

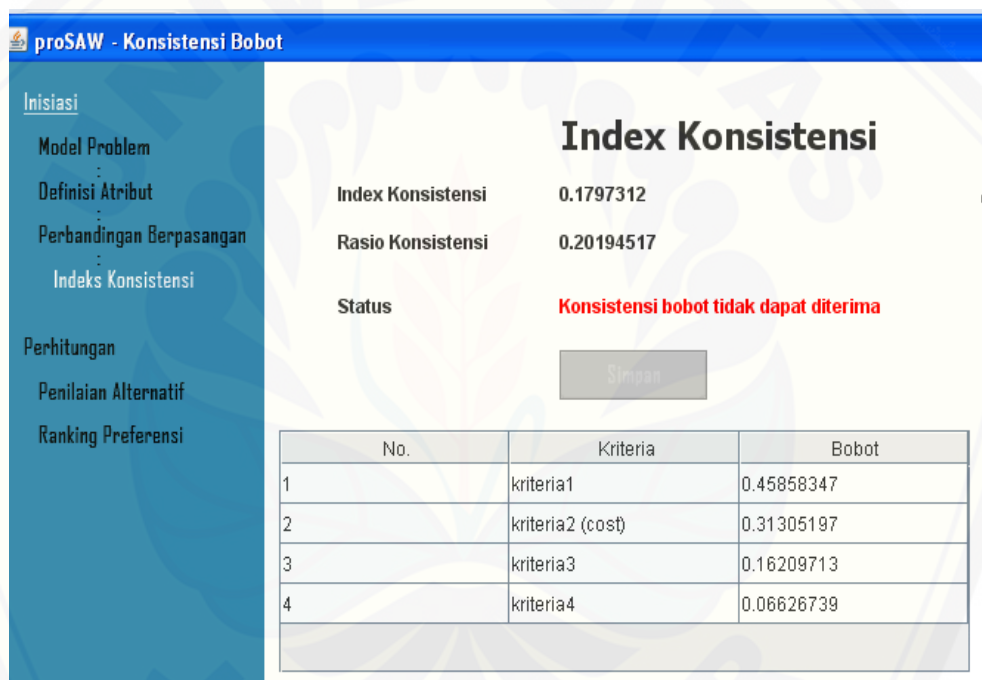
Tabel 5.4 Hasil perhitungan manual nilai preferensi alternatif

Alternatif	Nilai Preferensi
alternatif1	0.541
alternatif2	0.486
alternatif3	0.714
alternatif4	0.869
alternatif5	0.779
alternatif6	0.892

Tabel 5.3 menunjukkan konsistensi bobot dari nilai yang diberikan dalam matrik perbandingan berpasangan melebihi batas yang diperbolehkan sehingga bobot yang diperoleh dari perhitungan perbandingan berpasangan tidak konsisten.

Karena peneliti hanya ingin mengetahui ketepatan perhitungan yang dilakukan sistem, maka peneliti tetap menggunakan bobot yang dihasilkan meskipun konsistensinya tidak dapat diterima.

Setelah memperoleh bobot tiap kriteria, peneliti selanjutnya menggunakan bobot tersebut untuk menghitung nilai preferensi alternatif. Hasil yang diperoleh dari perhitungan manual tertera dalam Tabel 5.4. Berdasarkan Tabel 5.4 diatas, dapat diketahui alternatif dengan nilai terbesar adalah alternatif6 disusul alternatif4, alternatif5, alternatif3, alternatif1, dan alternatif dengan nilai terendah adalah alternatif2.



**Index Konsistensi**

Index Konsistensi: 0.1797312

Rasio Konsistensi: 0.20194517

Status: **Konsistensi bobot tidak dapat diterima**

No.	Kriteria	Bobot
1	kriteria1	0.45858347
2	kriteria2 (cost)	0.31305197
3	kriteria3	0.16209713
4	kriteria4	0.06626739

Gambar 5.11 Hasil perhitungan bobot kriteria dan konsistensinya oleh sistem

Gambar 5.11 menunjukkan bahwa hasil perhitungan bobot yang dilakukan sistem sama dengan bobot yang diperoleh dari perhitungan manual. Kesesuaian hasil perhitungan juga terdapat pada nilai indeks konsistensi (CI), rasio konsistensi (CR) dan status konsistensi bobot. dengan demikian dapat diketahui bahwa perhitungan yang dilakukan oleh sistem telah berjalan sesuai dengan yang diharapkan.



Ranking	Alternatif	Nilai
1	alternatif6	0.89193517
2	alternatif4	0.8689761
3	alternatif5	0.7785941
4	alternatif3	0.7138376
5	alternatif1	0.5410735
6	alternatif2	0.48628563

Gambar 5.12 Hasil perhitungan bobot kriteria dan konsistensinya oleh sistem

Nilai preferensi alternatif yang dihasilkan sistem berdasarkan proses perhitungan yang terjadi di dalamnya sama dengan nilai yang dihasilkan dari perhitungan manual. Peringkat pertama adalah alternatif6, lalu alternatif4, alternatif5, alternatif3, alternatif1, dan alternatif dengan nilai terendah adalah alternatif2. Nilai dari tabel pada Gambar 5.12 besarnya sama dengan nilai dalam Tabel 5.4.

Perhitungan lain yang juga dilakukan oleh sistem adalah perhitungan analisa sensitivitas. Perhitungan ini untuk mencari vector bobot baru apabila salah satu bobot kriteria diubah nilainya. Selain vector bobot baru, nilai alternatif baru juga akan dikalkulasi untuk mengetahui pengaruh perubahan bobot terhadap nilai preferensi. Pada Tabel 5.5 ditampilkan perhitungan manual analisa sensitivitas yang dilakukan oleh peneliti. Analisa sensitivitas bobot dihitung menggunakan persamaan (3). Sedangkan analisa sensitivitas nilai preferensi dihitung menggunakan persamaan (2).

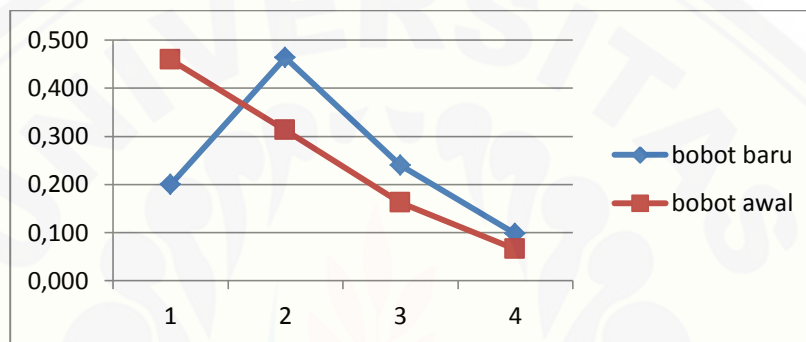
Tabel 5.5 Hasil perhitungan manual analisa sensitivitas

Kriteria	Bobot baru	Alternatif	Nilai baru
Kriteria1	0.200	Alternatif1	0.663
Riteria2	0.463	Alternatif2	0.514
Kriteria3	0.240	Alternatif3	0.782

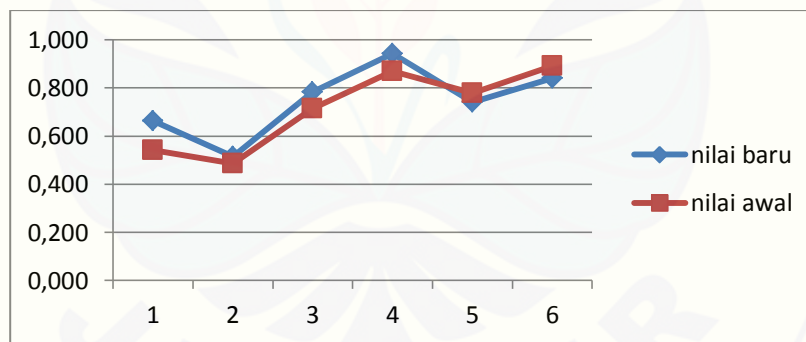


Kriteria4	0.098	Alternatif4	0.943
		Alternatif5	0.741
		Alternatif6	0.840

Sistem menampilkan hasil analisa sensitivitas menggunakan grafik. Gambar 5.13 merupakan nilai sensitivitas perhitungan manual dari Tabel 5.5 yang diubah dalam bentuk grafik. Dengan demikian penyesuaian hasil perhitungan akan lebih mudah.



(a)



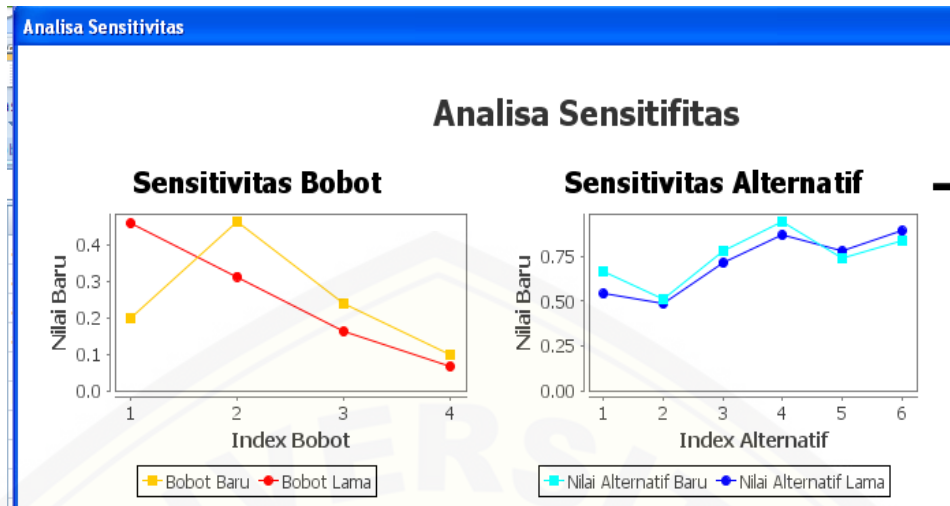
(b)

(a) Sensitivitas bobot; (b) Sensitivitas nilai preferensi

Gambar 5.13 Grafik analisa sensitivitas menggunakan perhitungan manual

Gambar 5.14 berikut merupakan grafik hasil perhitungan analisa sensitivitas bobot kriteria dan nilai alternatif berdasarkan perhitungan yang dilakukan oleh sistem. Dengan membandingkan grafik dalam Gambar 5.13 dengan Gambar 5.14 dapat diketahui ketepatan perhitungan yang dilakukan oleh sistem.





Gambar 5.14 Grafik hasil perhitungan analisa sensitivitas sistem

Pola grafik yang terdapat dalam Gambar 5.14 telah sesuai dengan pola grafik yang terdapat dalam Gambar 5.13. Selain itu nilai dalam sumbu-sumbu juga telah sesuai antarra kedua gambar tersebut. Dengan demikian telah diketahui bahwa perhitungan yang dilakukan oleh sistem telah akurat dan sesuai dengan apa yang diharapkan.

Selanjutnya sistem diuji dengan membandingkan data dari jurnal yang diteriliti oleh Afshari *et al.* (2010) dengan perhitungan dalam sistem. Gambar 5.15 merupakan *capture* hasil perhitungan yang dilakukan oleh Afshari *et al.* untuk bobot kriteria dan nilai alternatif.

C1	C2	C3	C4	C5	C6	C7
0.283	0.162	0.162	0.07	0.085	0.162	0.076

P1	0.553181
P2	0.713468
P3	0.837488
P4	0.51466
P5	0.579524

(a) Bobot Kriteria; (b) Nilai preferensi Alternatif

Gambar 5.15 Hasil perhitungan oleh Afshari *et al.*

Menggunakan nilai perbandingan berpasangan dan nilai matrik keputusan yang dicantumkan dalam jurnal yang dirujuk, peneliti memasukkan nilai tersebut

ke dalam sistem. Hasil dari proses perhitungan sistem dapat dilihat pada Gambar 5.16.

### Index Konsistensi

Index Konsistensi 0.028987408

Rasio Konsistensi 0.021472154

Status **Bobot yang diperoleh cukup konsisten**

Simpan

No.	Kriteria	Bobot
1	c1	0.2819766
2	c2	0.16133614
3	c3	0.16133614
4	c4	0.06976943
5	c5	0.08847345

(a)

### Rekomendasi Keputusan

Ranking	Alternatif	Nilai
1	p3	0.8381629
2	p2	0.7137211
3	p5	0.58092904
4	p1	0.55264443
5	p4	0.51491076

(b)

(a) Hasil perhitungan bobot oleh sistem; (b) Hasil perhitungan nilai oleh sistem

Gambar 5.16 Hasil perhitungan sistem menggunakan data rujukan

Hasil perhitungan sistem pada Gambar 5.17 menunjukkan sedikit perbedaan dengan jurnal rujukan. Perbedaan antar dua vektor bobot yang diperoleh tergolong kecil. Rata-rata selisih antara dua *vector* yang dihasilkan adalah 0,001. Perbedaan ini terjadi karena pembulatan angka desimal yang berbeda. Baik saat normalisasi data, maupun saat perhitungan konsistensi. Perbedaan nilai konsistensi rasio lebih besar daripada perbedaan vektor bobot kriteria. Penyebabnya selain karena bobot yang diperoleh sudah sedikit berbeda, juga disebabkan oleh nilai *random index (RI)* yang digunakan berbeda. Pada jurnal rujukan *RI* untuk 7 (tujuh) kriteria sebesar 1,32, sedangkan *RI* untuk 7 (tujuh) kriteria yang digunakan oleh sistem sebesar

1.35. hal tersebut menyebabkan selisih nilai  $CR$  antara dua perhitungan berbeda yaitu 0.031 untuk hasil dalam jurnal rujukan dan 0.021 untuk hasil perhitungan sistem.

Perbedaan vektor bobot menyebabkan perbedaan nilai preferensi alternatif. Apabila dibandingkan terdapat perbedaan nilai dalam Gambar 5.16 dengan gambar 5.17 . Meskipun demikian ranking alternatif yang diperoleh masih sama. Hal ini menunjukkan bahwa perbedaan perhitungan masih bisa ditoleransi karena selisihnya tidak terlalu besar dan hasil akhir (ranking alternatif) sama.

## 5.5 Batasan Sistem

Sistem yang dibangun oleh peneliti memiliki cakupan yang luas. Tidak keseluruhan data atau teori yang diperoleh dapat diimplementasikan kedalam sistem. Terdapat batasan-batasan yang melekat pada sistem yang dibangun. Batasan-batasan ini disebabkan oleh beberapa hal, seperti kelemahan metode yang digunakan, kemampuan aplikasi berdasarkan kode program, dan mempertimbangkan faktor kemudahan bagi pengguna. Beberapa batasan akan dibahas dalam subbab ini.

Pertama mengenai proses yang sangat penting dalam sistem, yaitu pemodelan struktur (inisiasi problem). Kontribusi inisiasi problem pada hasil akhir sangat besar. Apabila seseorang berhadapan dengan problem pengambilan keputusan dengan jumlah kriteria yang besar, maka hal ini akan menyulitkan bagi pengambil keputusan. Hal ini disebabkan karena semakin banyak kriteria yang terlibat, akan sulit mendapatkan bobot yang konsisten. Selain itu proses inisiasi semakin rumit apabila jumlah kriteria yang digunakan semakin besar. Maka dari itu peneliti memberlakukan batasan jumlah kriteria yang dapat digunakan.

Suatu problem pengambilan keputusan dapat terdiri dari lebih dari satu pembuat keputusan dan lebih dari satu level kriteria (hirarki) yang digunakan. Sistem yang diteliti kali ini hanya dapat digunakan untuk *single dimensional problems*. Yakni hanya melibatkan satu *decision maker* dan atribut yang terlibat tidak berupa hirarki.

Sistem hanya menggunakan nilai positif pada seluruh penilaiannya. Beberapa pendapat mengatakan bahwa penilaian dengan secara verbal memancing ambiguitas. Untuk mengatasi masalah ini peneliti mencantumkan nilai numerik untuk masing masing skala verbal sehingga pembuat keputusan dapat menyesuaikan persepsinya dengan standar nilai sistem. Standar skala nilai yang digunakan oleh peneliti adalah skala Saaty dimana penyajian bentuk verbalnya terdiri dari beberapa macam. Pilihan skala verbal ini dimaksudkan agar pembuat keputusan dapat menyesuaikan penilaian dengan kriteria yang digunakan. Sedangkan penilaian secara numerik diinputkan manual oleh pengguna dimana tidak memiliki fitur pembatasan besar angka yang dapat diinputkan selain batasan variabel yang digunakan dalam kode program. Oleh karena itu, kebijaksanaan dan ketelitian pengguna dalam menggunakan aplikasi diperlukan apabila berhadapan dengan kriteria dengan penilaian secara numerik.

Batasan lain yang ditemukan oleh peneliti terdapat pada konsistensi bobot. Ishizaka dan Labib (2009) mengutip beberapa jurnal penelitian yang mengkritisi perhitungan konsistensi rasio. Hal ini disebabkan karena perhitungan tersebut memperbolehkan penilaian yang bertentangan dalam matrik atau menolak penilaian yang beralasan jika nilainya melebihi 0,1. Meskipun penilaian dalam matrik diberikan oleh seorang ahli, tetap akan ditolak jika nilainya melebihi batasan yang diperbolehkan. Peneliti mengatasi hal ini dengan memungkinkan pembuat keputusan untuk tetap menggunakan bobot yang diperoleh meskipun konsistensinya melebihi batas yang ditentukan apabila pembuat keputusan telah merasa puas dengan bobot yang dihasilkan.

Mengingat berbagai jenis sistem penunjang keputusan dengan tujuan yang berbeda, berikut akan dituliskan macam-macam batasan dalam sistem dalam Tabel 5.6.

Tabel 5.6 Batasan sistem

Proses utama	Penjumlahan terbobot
Struktur problem	Banyak kriteria dan alternatif
Konsep	<i>Scoring model</i>
Hasil akhir	Ranking alternatif
Jenis input penilaian	Matrik keputusan
Dimensi problem	<i>Single dimensional</i>
Konsistensi bobot	YA
Data nilai	<i>Numeric value, verbal value, binary value</i>
Analisa Sensitivitas	Bobot dan nilai preferensi
Jumlah atribut	Max 99 atribut
Jumlah alternatif	Max 127 alternatif

Melalui Tabel 5.6 dapat diketahui problem pengambilan seperti apa yang dapat diproses dalam sistem yang dibangun. Jika pembuat keputusan ingin menggunakan aplikasi sistem untuk problem pengambilan keputusan yang memiliki kebutuhan berbeda dengan kemampuan sistem, maka apabila memungkinkan pembuat keputusan harus mentransformasikan problem pengambilan keputusannya kedalam bentuk yang disesuaikan dengan kemampuan sistem.

Keakuratan rekomendasi keputusan yang dihasilkan tergantung pada keakuratan penilaian pembuat keputusan dalam memberikan nilai, baik pada perbandingan berpasangan untuk menentukan bobot kriteria maupun penilaian alternatif pada masing-masing kriteria. Sedangkan keakuratan proses dalam sistem telah diuji dan sesuai dengan teori yang digunakan.



## BAB 6. PENUTUP

Bab ini merupakan bagian akhir yang berisi kesimpulan dan saran. Kesimpulan berisi keseluruhan informasi yang terdapat dalam penelitian yang dilakukan. Sedangkan saran merupakan masukan peneliti untuk penelitian yang dilakukan selanjutnya untuk topik yang serupa.

### 6.1 Kesimpulan

Kesimpulan dari penulisan hasil penelitian ini sebagai berikut:

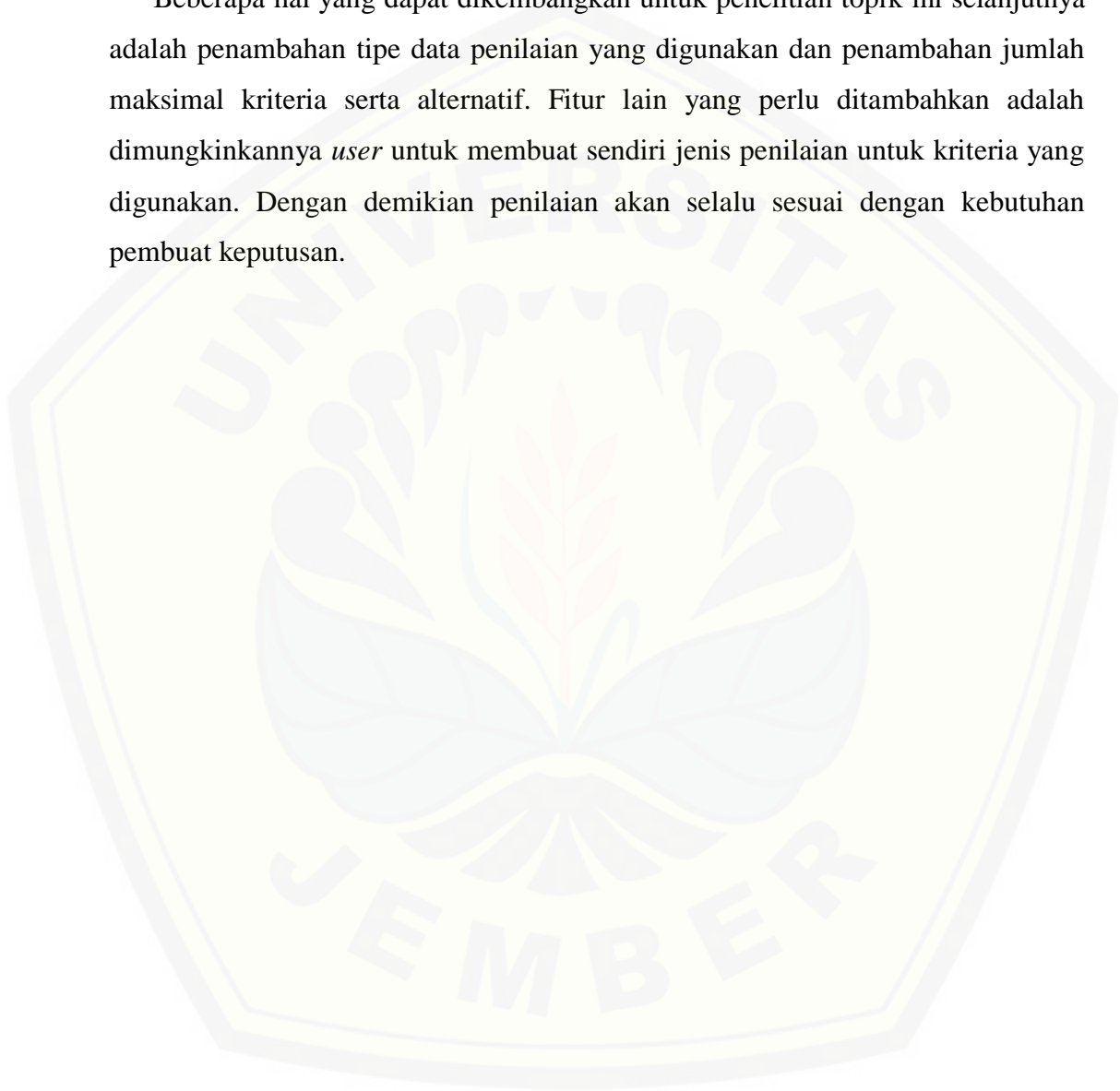
1. Hasil dari penelitian ini adalah sebuah perangkat lunak penunjang keputusan dengan banyak kriteria dan alternatif menggunakan metode SAW. Langkah-langkah dalam metode ini diterapkan dalam beberapa tahap pembuatan data dalam sistem. Tahapan tersebut antara lain pemodelan problem, definisi kriteria, perbandingan berpasangan, pengecekan indeks konsistensi, penilaian alternatif (matrik keputusan), dan menampilkan ranking keputusan.
2. Rekomendasi keputusan berupa ranking dari alternatif. Ranking berdasarkan nilai terbesar alternatif yang diperoleh dari penjumlahan terbobot. Sedangkan analisa perubahan nilai yang disebabkan oleh perubahan bobot salah satu atribut/kriteria disajikan dalam bentuk grafik yang menunjukkan nilai sebelum terjadi perubahan dan nilai baru setelah salah satu bobot atribut diubah. Menurut pengujian yang telah dilakukan, hasil perhitungan yang dihasilkan sistem sama dengan nilai perhitungan yang dilakukan secara manual.
3. Pengguna / pembuat keputusan dapat menyimpan struktur model problem penunjang keputusan kedalam database dan mengelolanya. Pengelolaan ini mencakup melihat detail struktur model problem, menggunakan model problem penunjang keputusan tersimpan untuk mencari alternatif terbaik, mengubah struktur model problem, dan menghapus struktur model problem tersimpan. Pengelolaan ini ditujukan agar struktur model problem yang



tersimpan dapat disesuaikan dengan perubahan-perubahan yang mungkin terjadi.

## 6.2 Saran

Beberapa hal yang dapat dikembangkan untuk penelitian topik ini selanjutnya adalah penambahan tipe data penilaian yang digunakan dan penambahan jumlah maksimal kriteria serta alternatif. Fitur lain yang perlu ditambahkan adalah dimungkinkannya *user* untuk membuat sendiri jenis penilaian untuk kriteria yang digunakan. Dengan demikian penilaian akan selalu sesuai dengan kebutuhan pembuat keputusan.



**Daftar Pustaka**

- Abdullah , Lazim, dan Adawiyah ,C.W. Rabiatul. 2014. Simple Additive Weighting Methods of Multi criteria Decision Making and Applications: A Decade Review. *International Journal of Information Processing and Management (IJIPM)*, 5 (1): 39-49.
- Afshari, Alireza, Mojahed, Majid, dan Yusuff, Mohd. 2010. Simple Additive Weighting approach to Personnel Selection problem. *International Journal of Innovation, Management and Technology*, 1 (5): 511-515.
- Azar, Fred S. 2000. Multiattribute Decision-Making: Use of Tree Scoring Methods to Compare the Performance of Imaging Techniques for Breast Cancer Detection. WWW repository [jurnal on line]. [http://repository.upenn.edu/cis\\_reports/119](http://repository.upenn.edu/cis_reports/119). [diakses pada 9 April 2015].
- Franco, L. Alberto, dan Montibeller, Gilberto. 2009. Problem Structuring for Multi-Criteria Decision Analysis Interventions. *Operational Research Group, Department of Management London School of Economics and Political Science*, 09 -115
- Gayatri, Vyas, dan S. Misal Chetan. 2013. Comparative Study of Different Multi-criteria Decision-making Methods. *IJACTE*, 2 (4): 9-12.
- Ishizaka, Alessio, dan Labib, Ashraf. 2009. Analytic Hierarchy Process and Expert Choice: Benefits and limitation. *OR Insight* , 22 (4): 201 – 220.
- Kaho, Monica Louisa Ratu, Tanaamah, Andeka Rocky, dan Wowor, Alz Danny. (Tanpa Tahun). Penerapan Metode Simple Additive Weighting Pada

Penentuan Tingkat Kesejahteraan Penduduk Propinsi Nusa Tenggara Timur. Salatiga: Universitas Kristen Satya Wacana.

Levine, P., dan Pomerol, J. Ch.. 1986. PRIAM, an interactive program for choosing among multiple attribute alternatives. *European Journal of Operational Research*, 25: 272-280.

Memariani, Azizollah, Amini, Abbas, dan Alinezhad, Alireza. 2009. Sensitivity Analysis of Simple Additive Weighting Method (SAW): The Results of Change in the Weight of One Attribute on the Final Ranking of Alternatives. *Journal of Industrial Engineering*, 4: 13- 18.

Podvezko, Valentinas. 2011. The Comparative Analysis of MCDA Methods SAW and COPRAS. *Inzinerine Ekonomika-Engineering Economics*, 22 (2): 134-146.

Rohayani, Hetty. 2013. Analisis Sistem Pendukung Keputusan Dalam Memilih Program Studi Menggunakan Metode Logika Fuzzy. *Jurnal Sistem Informasi (JSI)*, 5 (1): 530-539.

Rubiyatun, Winarno, Bowo, dan Sulistijowati, Sri. 2012. Simulasi Seleksi Mahasiswa Baru Jalur Undangan dengan Menggunakan Metode *Simple Additive Weighting*. Seminar Nasional Matematika, 162-167.

Saaty, Thomas L.. 2008. Relative Measurement and Its Generalization in Decision Making Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors The Analytic Hierarchy/Network Process. *RACSAM*, 102 (2): 251–318.

Sun, Xiaoqian, dan Li, Yongchang. (Tanpa Tahun). An Intelligent Multi-Criteria Decision Support System for Systems Design. *American Institute of Aeronautics and Astronautics*.

Thor, Jureen, Ding, Siew-Hong, dan Kamaruddin, Shahrul. Comparison of Multi Criteria Decision Making Methods From The Maintenance Alternative Selection Perspective. *The International Journal Of Engineering And Science*. 2 (6):27-34.

Triantaphyllou, Shu, Sanchez, Ray. 1998. Multi-Criteria Decision Making: An Operations Research Approach. *Encyclopedia of Electrical and Electronics Engineering*, 15: 175-186.

Tzeng, Gwo-Hshiung, dan Huang, Jih-Jeng. 2011. *Multiple Attribute Decision Making: Methods and applications*. CRC Press: Amerika

Velasquez, Mark, dan Hester, Patrick T.. 2013. An Analysis of Multi-Criteria Decision Making Methods. *International Journal of Operations Research* 10 (2): 56-66.

## LAMPIRAN

### LAMPIRAN A. PERANCANGAN SISTEM

#### A. 1 Skenario sistem

##### 1. Skenario Perhitungan dengan SAW

<i>Use case</i>	Perhitungan dengan SAW
<i>Actor</i>	<i>user</i>
<i>Pre-Condition</i>	<ul style="list-style-type: none"> <li>- <i>User</i> telah melakukan inisiasi model problem</li> <li>- <i>User</i> mengklik tombol “Selanjutnya” di jendela Indeks Konsistensi</li> <li>- Sistem menampilkan jendela Penilaian Alternatif</li> </ul>
<i>Post-Condition</i>	<ul style="list-style-type: none"> <li>- <i>User</i> memperoleh ranking alternatif</li> </ul>
Deskripsi	Proses ini mencakup pengisian nilai masing-masing alternatif untuk tiap kriteria

#### SKENARIO UTAMA : PENILAIAN ALTERNATIF

<i>User</i>	<i>System</i>
1. Mengisi jumlah alternatif yang digunakan pada kolom isian Jumlah Alternatif	
2. Klik tombol “Mulai”	
	3. Menampilkan tabel isian penilaian alternatif
4. Mengisi tabel isian dengan nama/identitas alternatif (kolom “Alternatif”) dan nilai masing-masing sesuai dengan format isian yang disediakan	
5. Klik tombol “Selanjutnya”	
	6. Cek inputan <i>user</i>
	7. Menyimpan matrik keputusan ke dalam variabel
	8. Mengkalkulasi nilai preferensi tiap alternatif
	9. Mengurutkan alternatif berdasarkan perolehan nilai preferensi
10. Jika memilih tombol “Batal”	
	11. Menampilkan dialog konfirmasi pembatalan pembuatan data baru

#### ALTERNATIF 1: ISIAN TIDAK LENGKAP

<i>User</i>	<i>System</i>
	7a. Menampilkan pesan isian belum lengkap



ALTERNATIF 2: JUMLAH ALTERNATIF LEBIH DARI 127 ATAU KURANG DARI 1	
<i>User</i>	<i>System</i>
	3b. Menampilkan pesan kesalahan

SKENARIO UTAMA : LIHAT RANKING ALTERNATIF	
<i>User</i>	<i>System</i>
1. Melihat ranking alternatif dalam tabel	
2. Klik tombol “Sensitifitas”	
	3. Menampilkan jendela Analisa Sensitifitas
4. Jika memilih tombol “Selesai”	
	5. Menampilkan jendela awal
6. Jika memilih tombol “Sebelumnya”	
	7. Menampilkan jendela penilaian alternatif

1. Skenario Kelola Penunjang Keputusan

<i>Use case</i>	Kelola Penunjang Keputusan Tersimpan
<i>Actor</i>	<i>User</i>
<i>Pre-Condition</i>	<ul style="list-style-type: none"> <li>- <i>User</i> membuka aplikasi proSAW</li> <li>- <i>User</i> memilih tombol “Gunakan Data Tersimpan”</li> <li>- Sistem menampilkan jendela Daftar Penunjang Keputusan</li> </ul>
<i>Post-Condition</i>	Pengguna menggunakan penunjang keputusan tersimpan atau data telah diperbarui
Deskripsi	Pengelolaan data penunjang keputusan dilakukan agar data tersimpan dapat disesuaikan dengan kebutuhan pengguna.

SKENARIO UTAMA : KELOLA PENUNJANG KEPUTUSAN TERSIMPAN	
<i>User</i>	<i>System</i>
1. Memilih salah satu data dalam tabel daftar penunjang keputusan	
	2. Menampilkan <i>option dialog</i> aksi
3. Klik “Gunakan”	
	4. Mengambil data penunjang keputusan yang dipilih dari basis data
	5. Memسukkan nilai yang diperoleh kedalam variabel
	6. Menampilkan jendela Penilaian Alternatif
7. Jika memilih tombol “Detail”	
	8. Mengambil data penunjang keputusan yang dipilih dari basis data
	9. Memسukkan nilai yang diperoleh kedalam

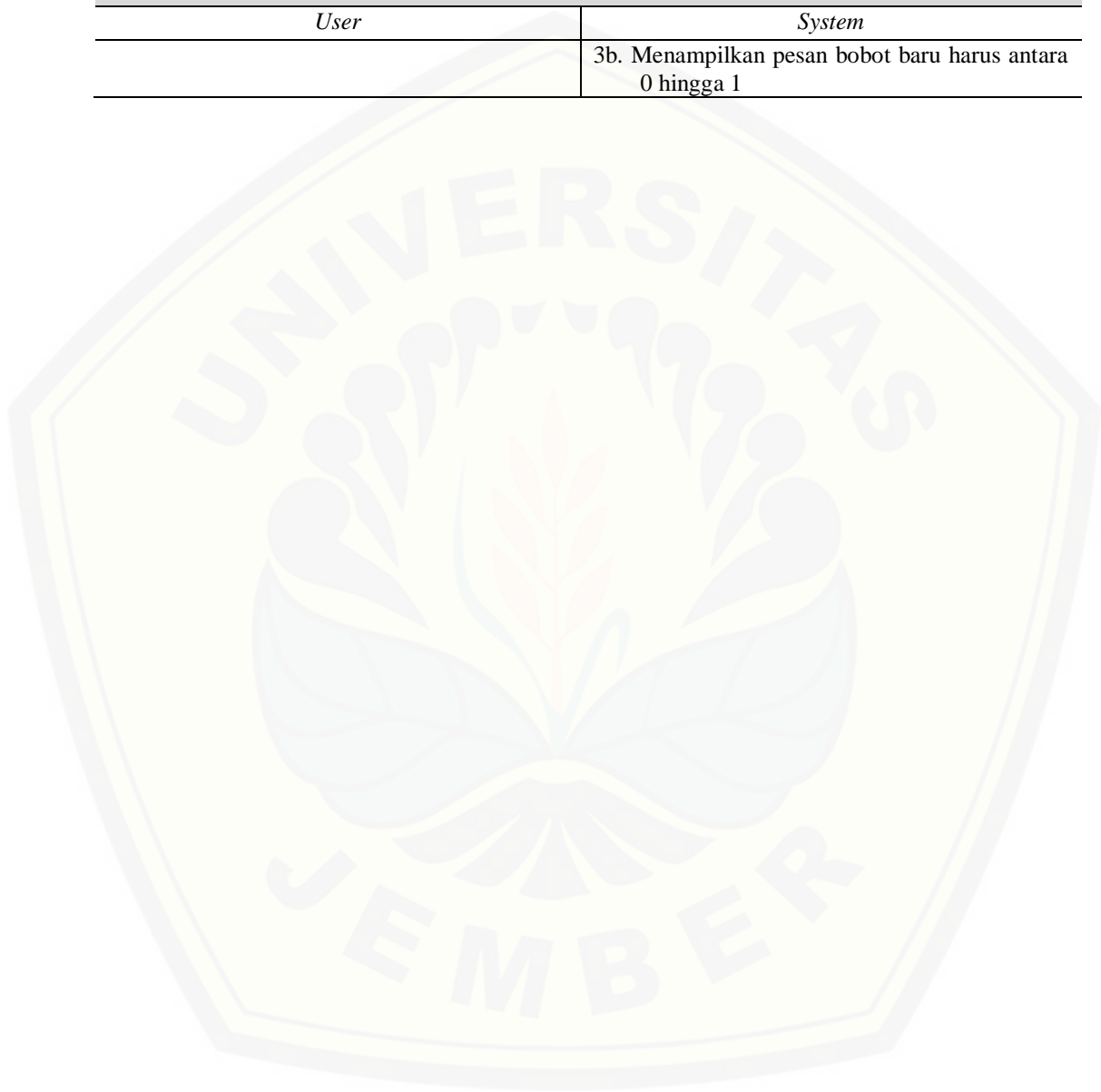
	variabel
	10. Menampilkan jendela Detail Data
11. Jika memilih tombol “Edit”	
	12. Mengambil data penunjang keputusan yang dipilih dari basis data
	13. Memsukkan nilai yang diperoleh kedalam variabel
	14. Menampilkan jendela Pemodelan Problem
15. Jika memilih tombol “Hapus”	
	16. Menampilkan dialog konfirmasi
17. Jika memilih tombol “Yes”	
	18. Menghapus data terpilih dari basis data
<b>ALTERNATIF 1: GAGAL AKSES DATA</b>	
<i>User</i>	<i>System</i>
	4a. Menampilkan pesan gagal
<b>ALTERNATIF 2: BATAL MENGHPUS DATA</b>	
<i>User</i>	<i>System</i>
17b. Klik “No” pada dialog konfirmasi	

## 2. Skenario Analisa Sensitifitas

<i>Use case</i>	Analisa Sensitifitas
<i>Actor</i>	<i>User</i>
<i>Pre-Condition</i>	<ul style="list-style-type: none"> <li>- <i>User</i> telah memperoleh ranking keputusan</li> <li>- <i>User</i> mengklik tombol “Sensitifitas” di jendela penilaian alternatif</li> <li>- Sistem menampilkan jendela analisa sensitifitas</li> </ul>
<i>Post-Condition</i>	<ul style="list-style-type: none"> <li>- Pengguna mengetahui sensitifitas bobot kriteria dan sensitifitas nilai preferensi alternatif</li> </ul>
Deskripsi	Proses ini menunjukkan seberapa besar perubahan bobot kriteria dan nilai preferensi akibat perubahan salah satu bobot kriteria.

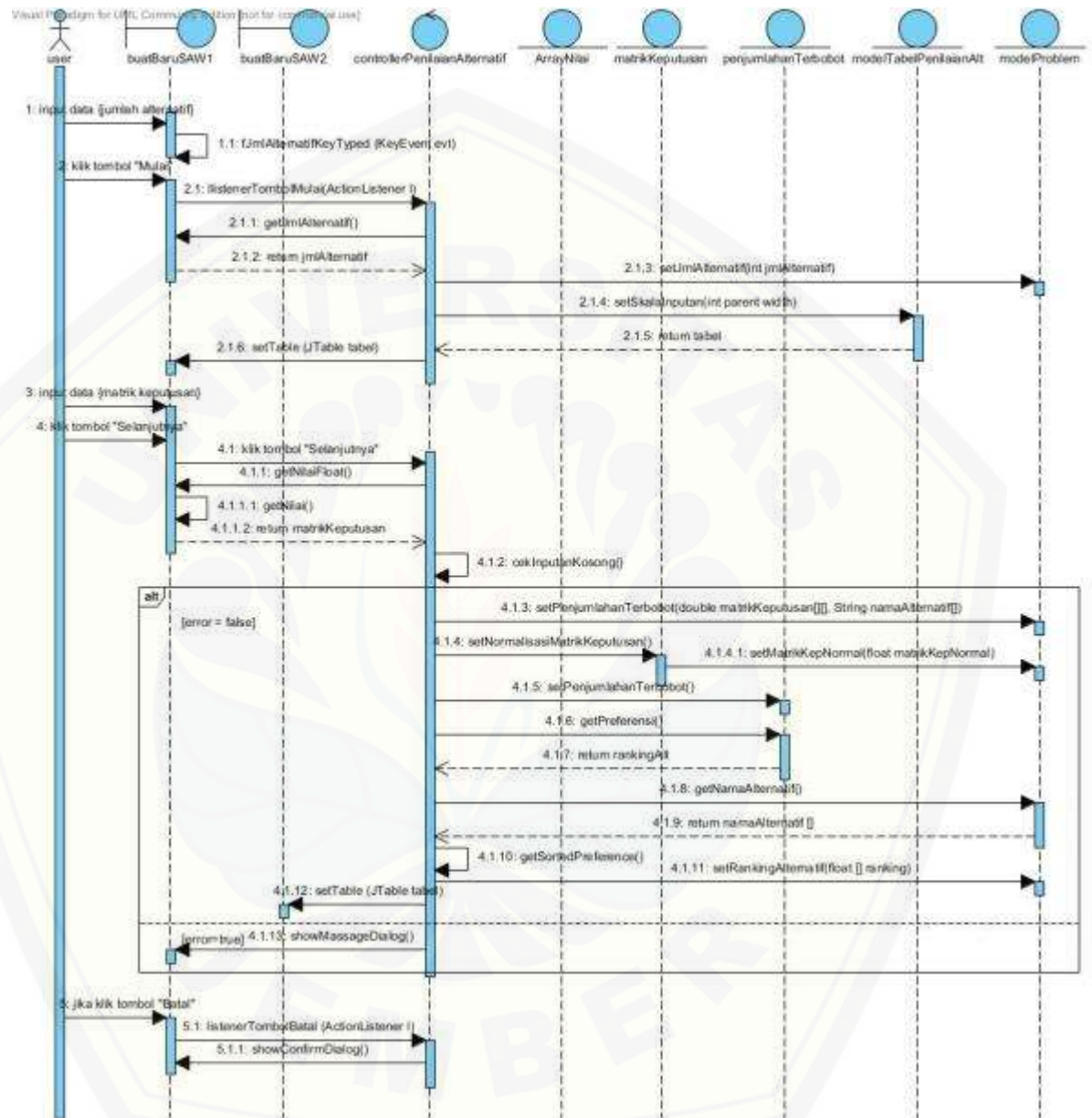
<b>SKENARIO UTAMA : ANALISA SENSITIFITAS</b>	
<i>User</i>	<i>System</i>
1. Memilih kriteria yang akan dirubah bobotnya. Isi bobot baru kriteria tersebut. Pilih sensitifitas apa yang ingin diketahui.	
2. Klik tombol “Sensitifitas”	
	3. Cek inputan <i>user</i>
	4. Mengkalkulasi besar perubahan bobot lain dan perubahan nilai akhir alternatif

5. Jika memilih tombol "Selesai"	
	6. Menutup jendela Analisa Sensitifitas
ALTERNATIF 1: ISIAN TIDAK LENGKAP	
<i>User</i>	<i>System</i>
	6a. Menampilkan pesan isian belum lengkap
ALTERNATIF 2: PERUBAHAN BOBOT LEBIH BESAR DARI 1	
<i>User</i>	<i>System</i>
	3b. Menampilkan pesan bobot baru harus antara 0 hingga 1

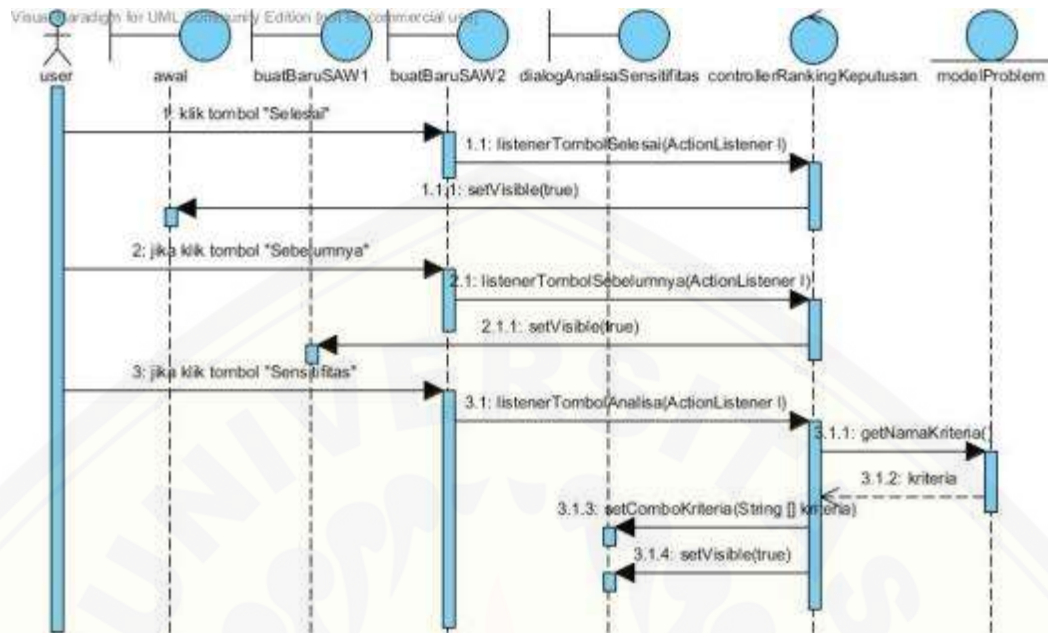


A. 2 Sequence Diagram

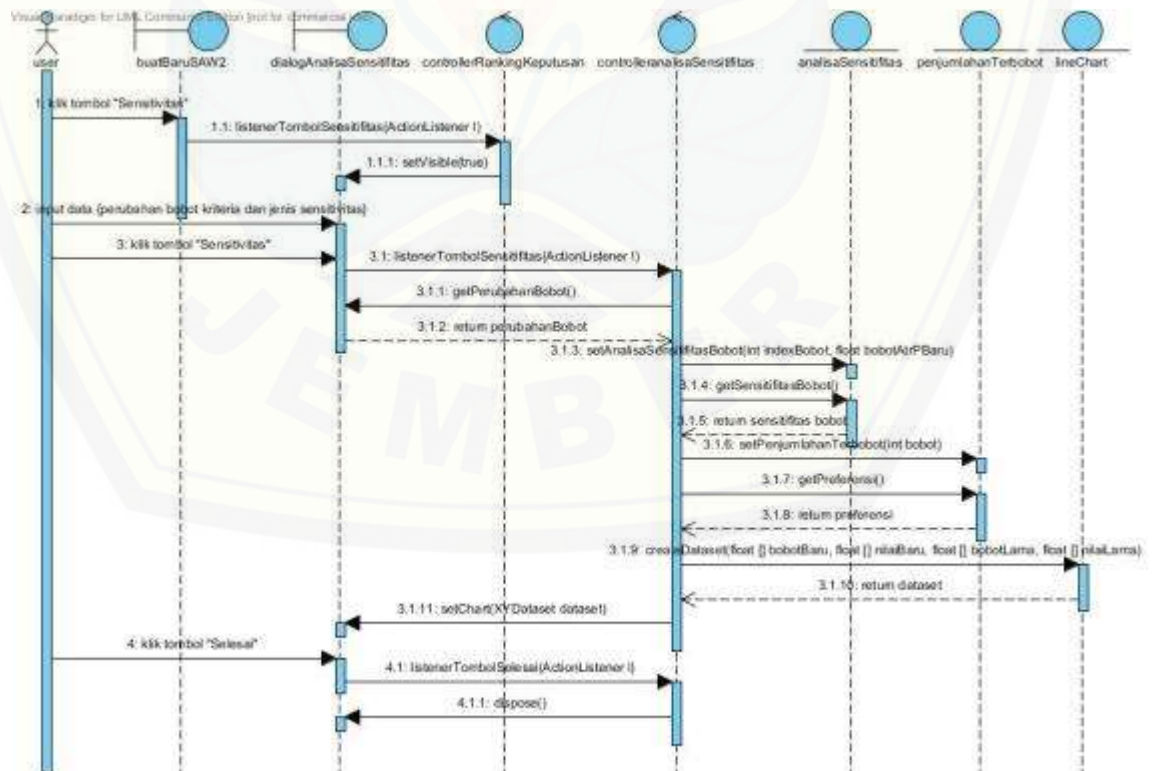
1. Sequence Diagram Penilaian Alternatif



2. Sequence Diagram Lihat Ranking Alternatif

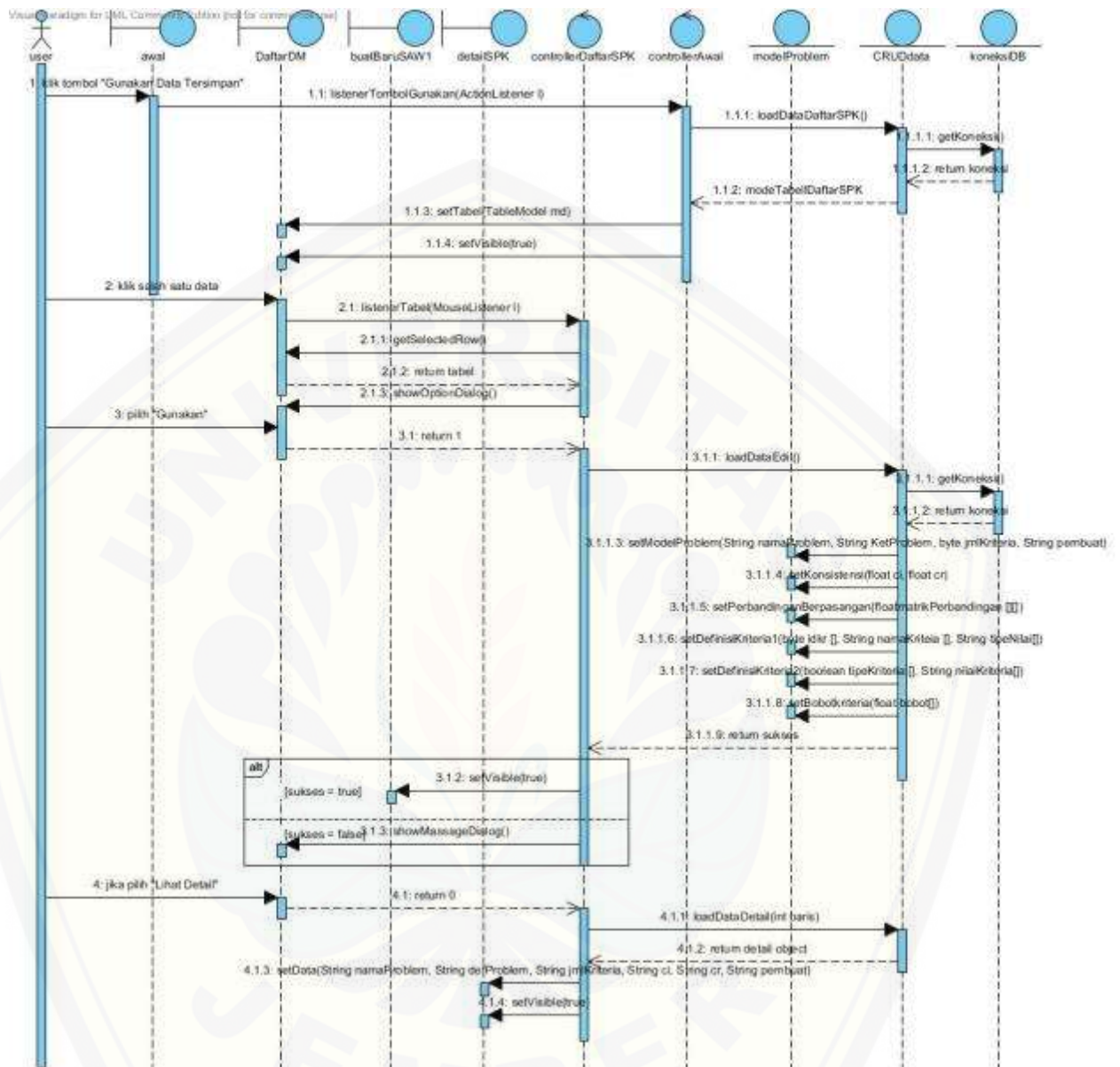


3. Sequence Diagram Analisa Sensitivitas

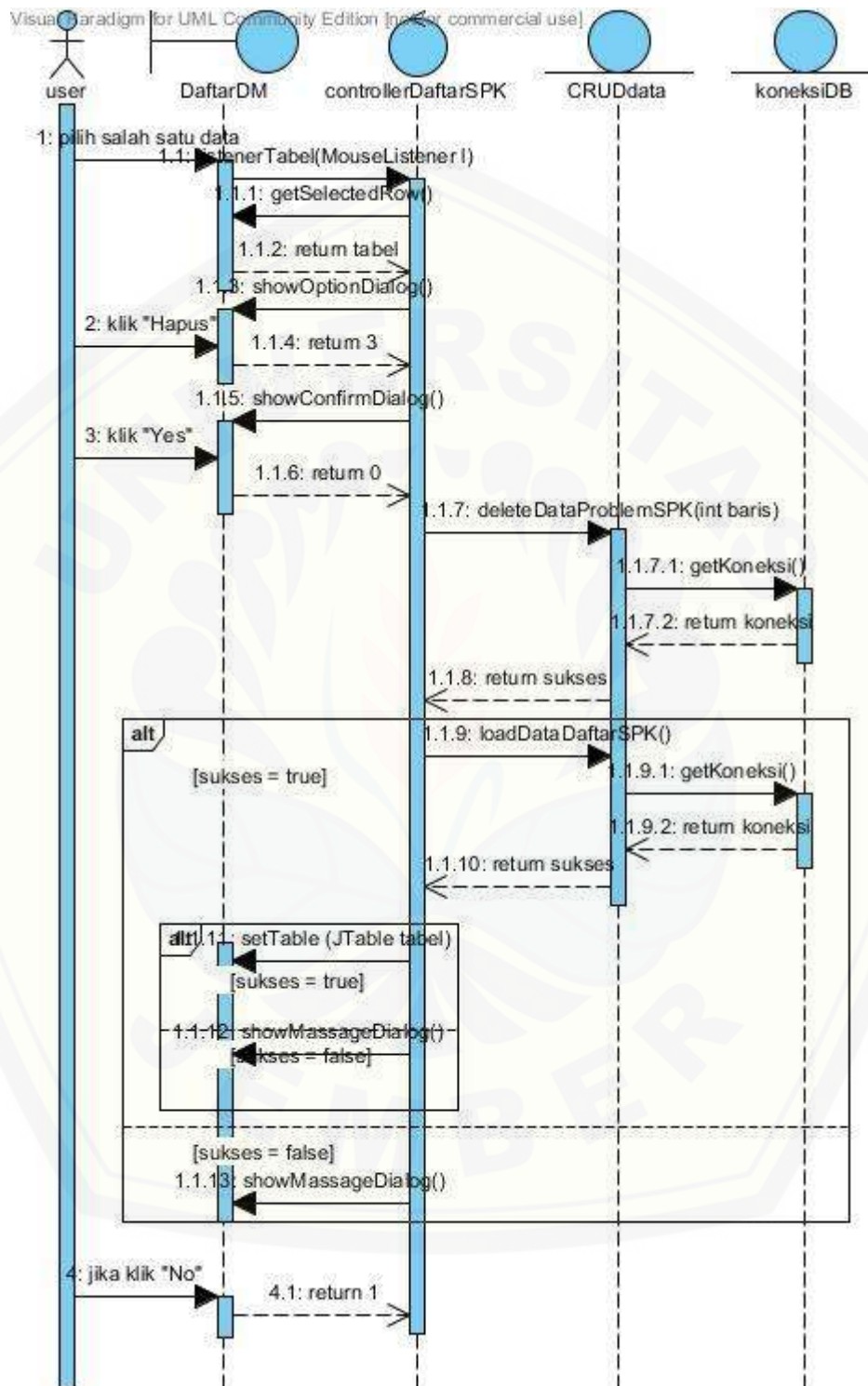




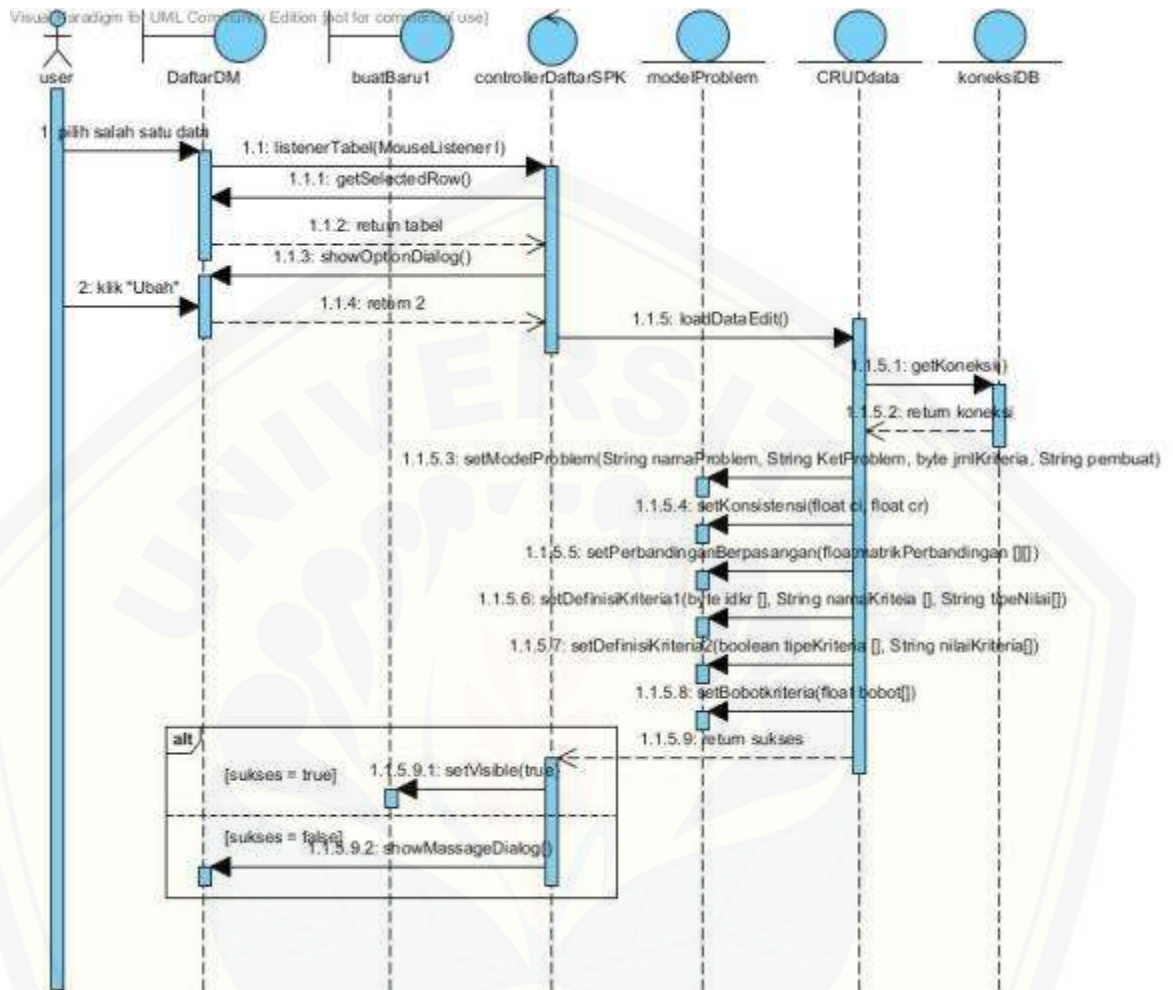
4. Sequence diagram: Gunakan SPK Tersimpan



5. Sequence Diagram: Hapus SPK Tersimpan

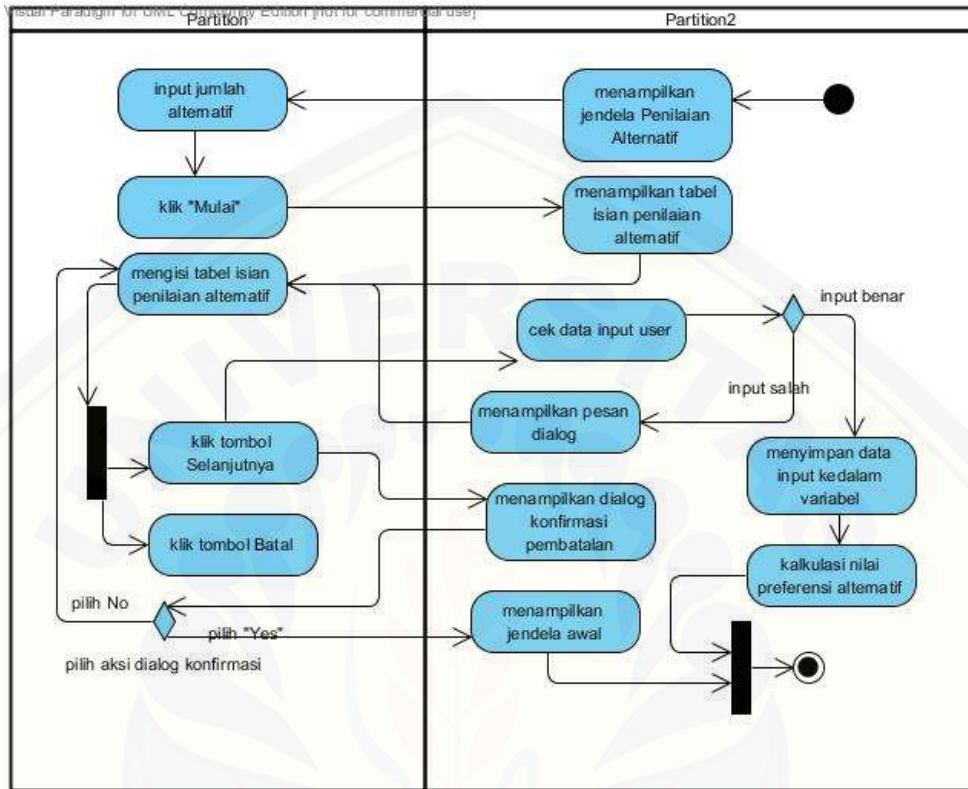


6. Sequence Diagram : Ubah SPK tersimpan

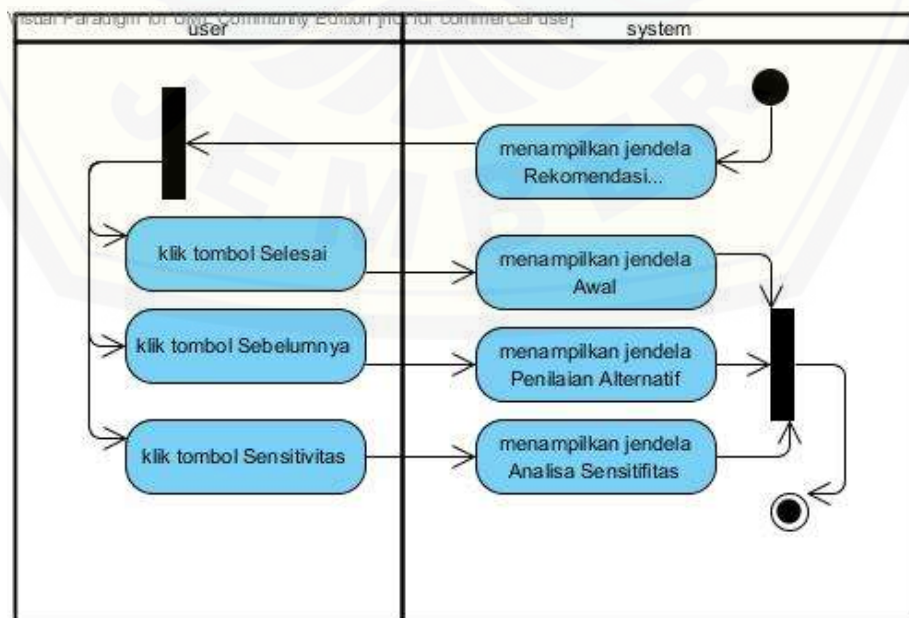


A. 3 Activity Diagram

1. Activity : Penilaian Alternatif

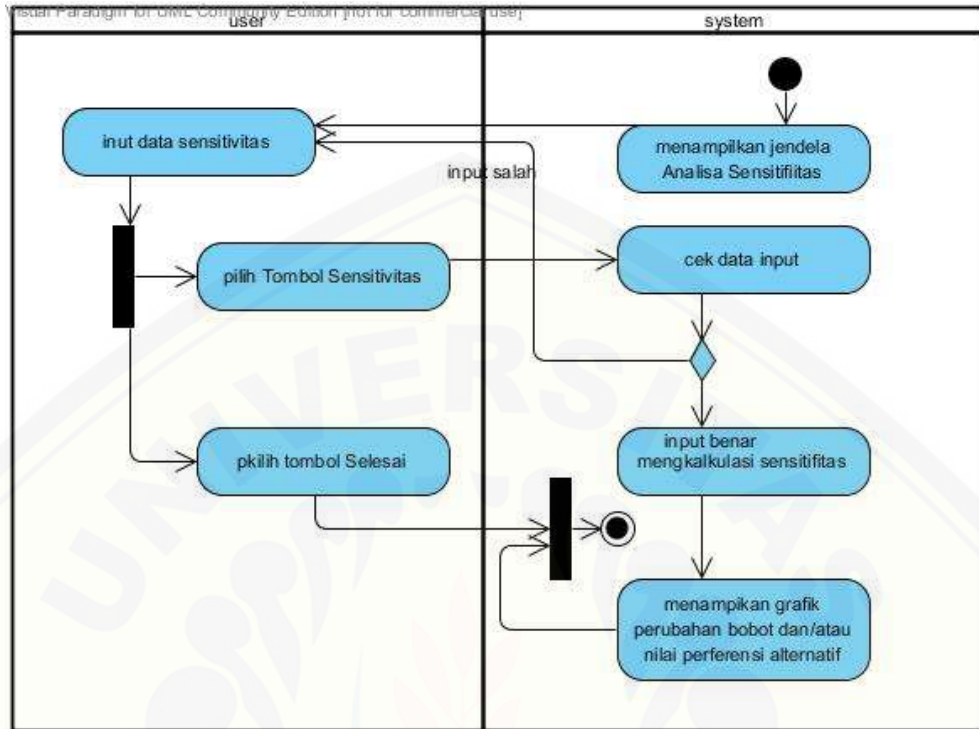


2. Activity : Lihat Ranking Alternatif

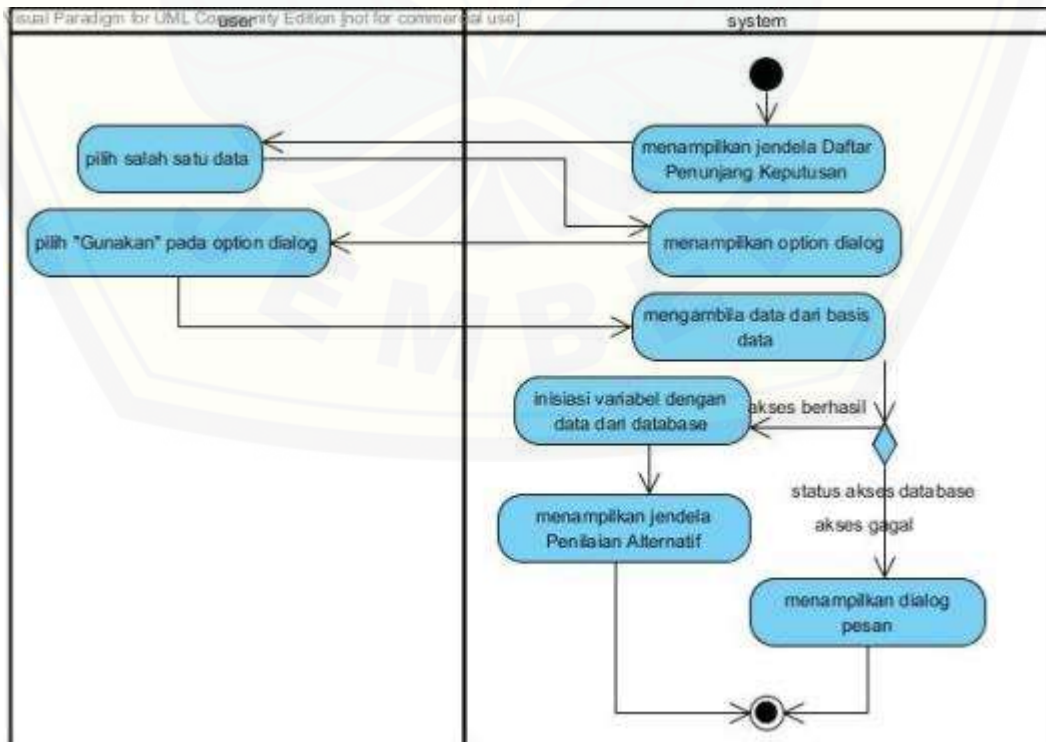




3. Activity : Analisa Sensitivitas

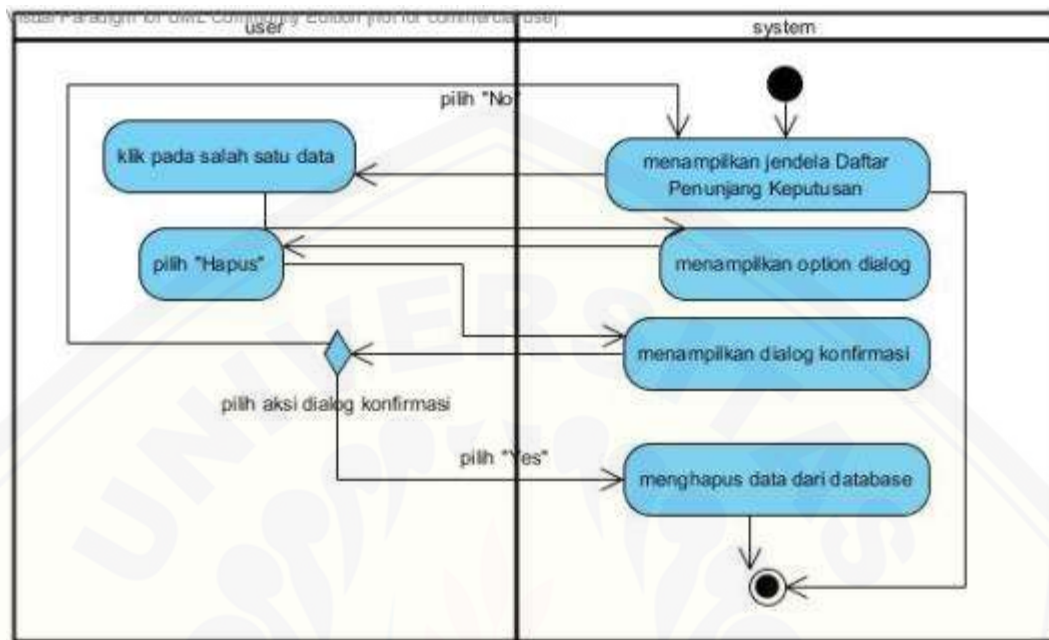


4. Activity : Gunakan SPK Tersimpan

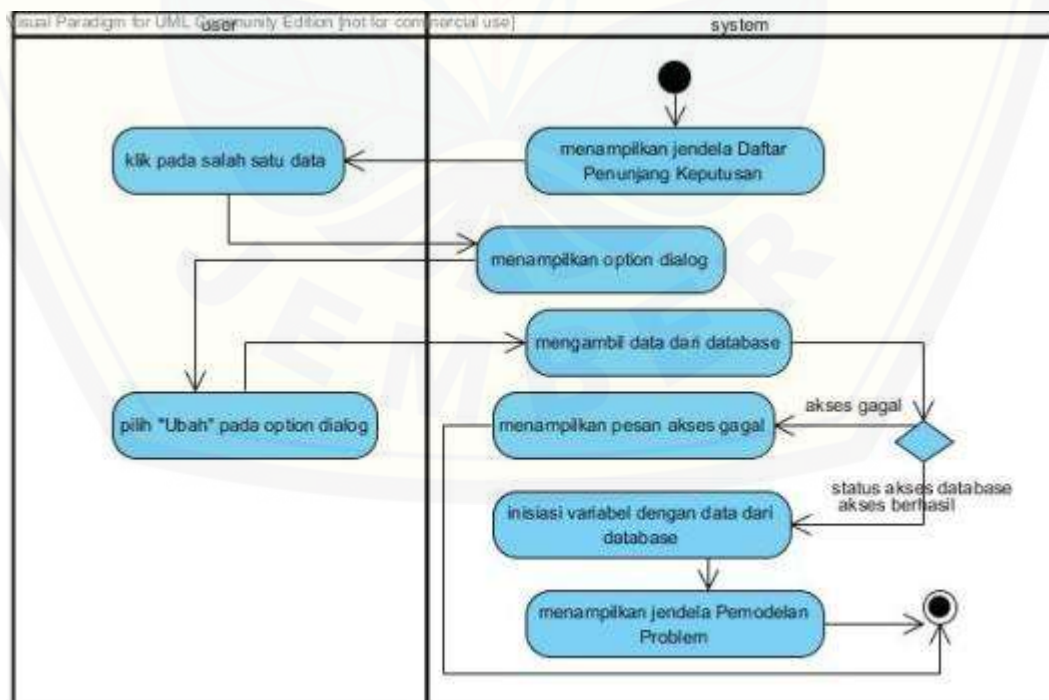




5. Activity : Hapus SPK Tersimpan

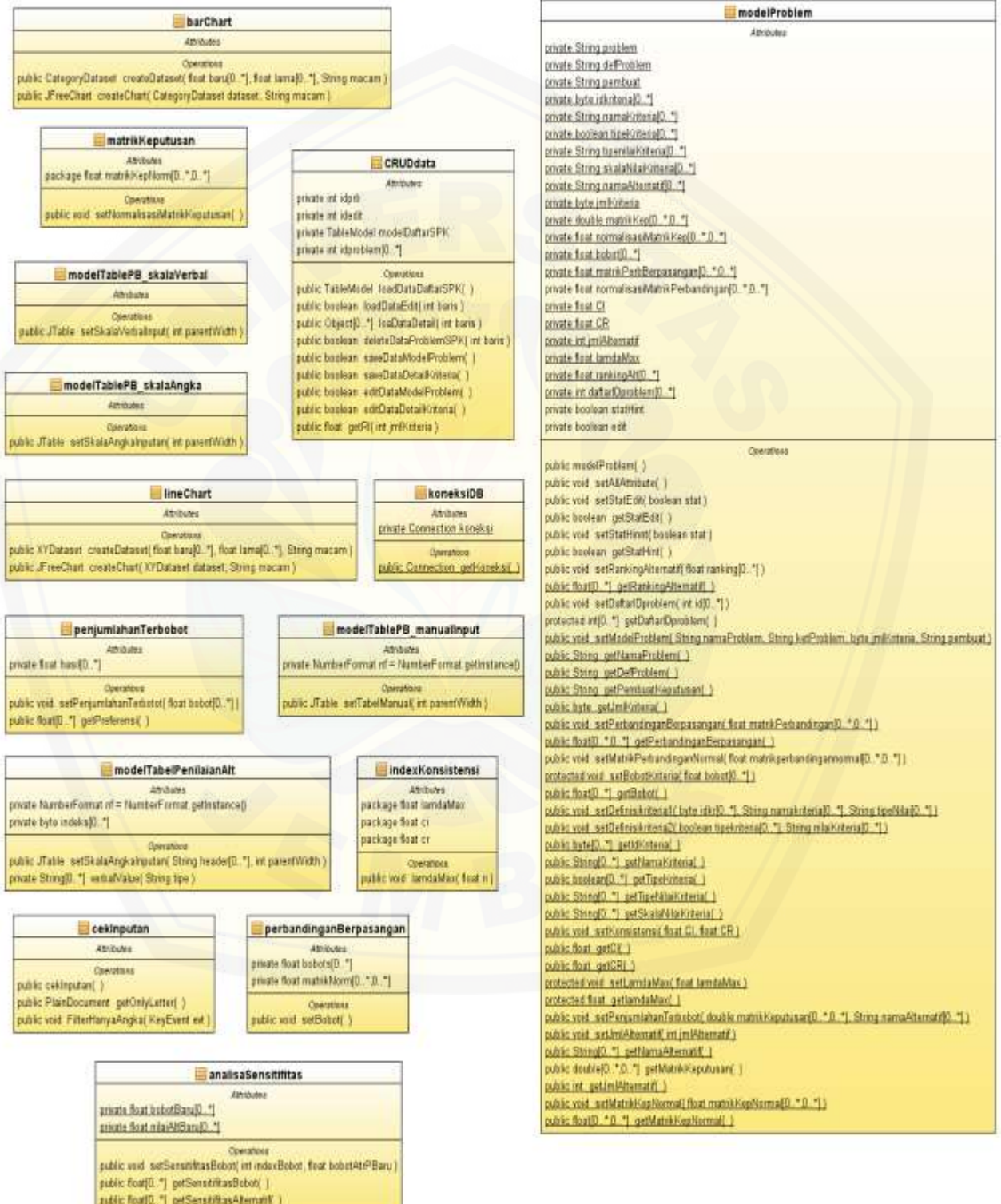


6. Activity : Ubah SPK Tersimpan



A. 4 Atribut dan Operasi Class Diagram

1. Atribut dan Operasi *Package Model*

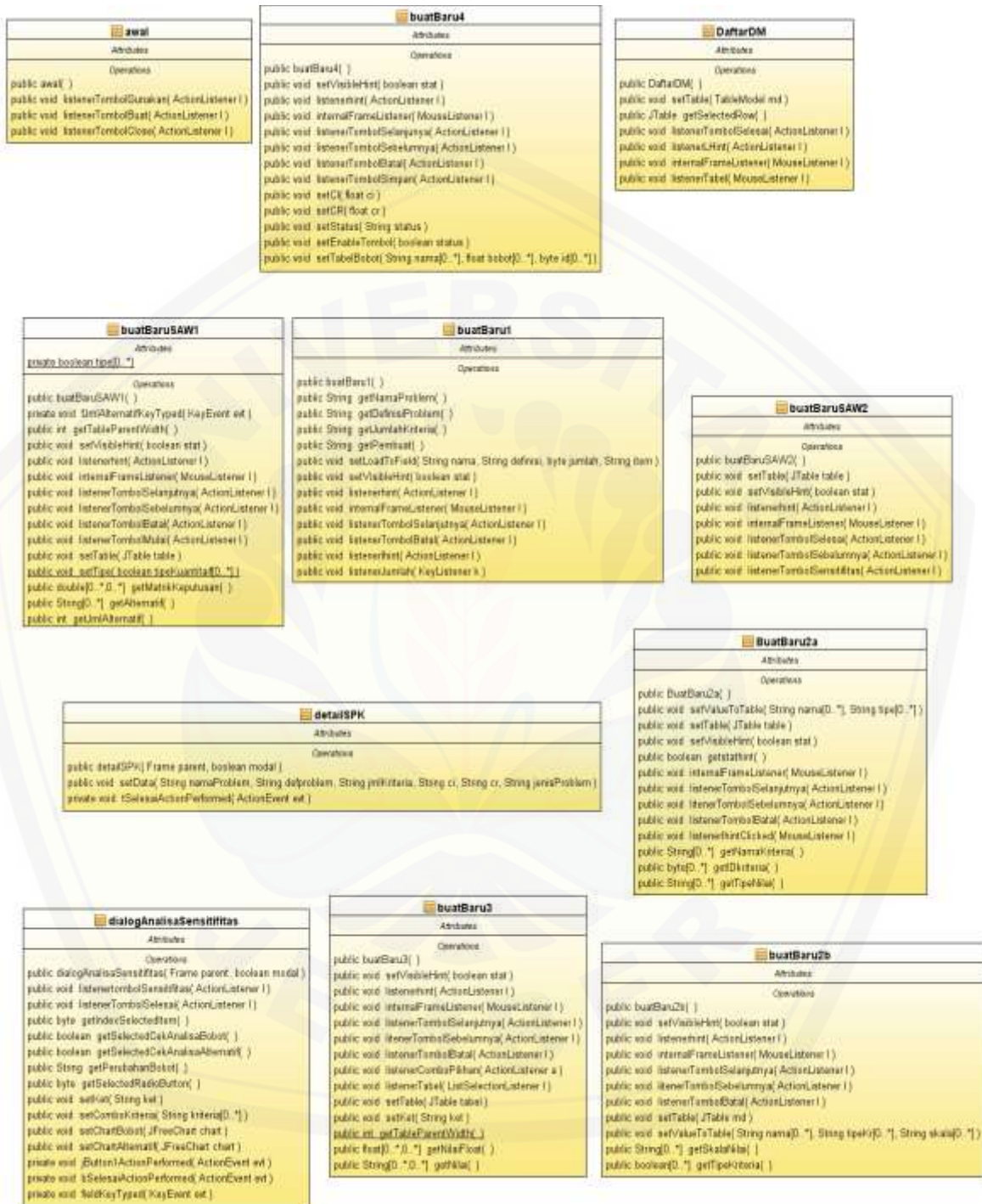


## 2. Atribut dan Operasi Package Controller





### 3. Atribut dan Operasi Package View



**LAMPIRAN B. KODE PROGRAM****1. Kode Program *Controller* Penyimpanan Data  
(controllerIndeksKonsistensi.java)**

```
package controler;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import javax.swing.JOptionPane;
import model.CRUDdata;
import model.modelProblem;
import view.awal;
import view.buatBaru3;
import view.buatBaru4;
import view.buatBaruSAW1;

public class controllerIndeksKonsistensi {
    private buatBaru4 view;
    private buatBaru3 viewSebelumnya;
    private buatBaruSAW1 viewSelanjutnya;
    private modelProblem model;
    private awal vawal;
    private CRUDdata db;
    private boolean tersimpan = false;

    class listenerhint implements ActionListener {

        public void actionPerformed(ActionEvent e) {
            view.jPanel3.setVisible(true);
            model.setStatHinnt(true);
        }
    }

    class internalframelistener implements MouseListener{

        public void mouseClicked (MouseEvent e) {
            view.jPanel3.setVisible(false);
            model.setStatHinnt(false);
        }
        public void mousePressed(MouseEvent e) {}

        public void mouseReleased(MouseEvent e) {}

        public void mouseEntered(MouseEvent e) {}

        public void mouseExited(MouseEvent e) {}
    }
    class listenerTombolSelanjutnya implements ActionListener{

        public void actionPerformed(ActionEvent e) {
            if(tersimpan==false){
```



```

        byte a =(byte) JOptionPane.showConfirmDialog(null, "Apakah Anda yakin melanjutkan tanpa
        menyimpan data?"
        , "Option", JOptionPane.YES_NO_OPTION);
        view.requestFocus();

        if (a == 0){
            viewSelanjutnya.setVisibleHint(model.getStatHint());
            viewSelanjutnya.setExtendedState(view.getExtendedState());
            viewSelanjutnya.setVisible(true);
            view.requestFocus();
            view.setVisible(false);
        }else{
        }
    }else{
        viewSelanjutnya.setVisibleHint(model.getStatHint());
        viewSelanjutnya.setLocation(view.getLocation());
//        viewSelanjutnya.setSize(view.getSize());
        viewSelanjutnya.setExtendedState(view.getExtendedState());
        viewSelanjutnya.setVisible(true);
        view.setVisible(false);
    }
}
}

class listenerTombolSebelumnya implements ActionListener{

    public void actionPerformed(ActionEvent e) {
        viewSebelumnya.setExtendedState(view.getExtendedState());
        modelProblem.setKonsistensi(0, 0);
        viewSebelumnya.setVisible(true);
        view.setVisible(false);
    }
}

class listenerTombolBatal implements ActionListener{

    public void actionPerformed(ActionEvent e) {
        byte a =(byte) JOptionPane.showConfirmDialog(null, "Kembali ke jendela awal?"
        , "Konfirmasi", JOptionPane.YES_NO_OPTION);
        view.requestFocus();
        if (a==0){
            vawal.setVisible(true);
            view.setVisible(false);
            model.setAllAttribute();
        }else{
        }
    }
}

class listenerTombolSimpan implements ActionListener{

    public void actionPerformed(ActionEvent e) {
        boolean sukses1, sukses2;
        if (model.getStatEdit()==true){
            sukses1 = db.editDataModelProblem(/*modelProblem.getNamaProblem(),

```

```

        modelProblem.getDefProblem(), modelProblem.getJmlKriteria(),
        modelProblem.getPerbandinganBerpasangan(),
        modelProblem.getCI(), modelProblem.getCR(),
        modelProblem.getPembuatKeputusan(*));
    if (sukses1==true){
    sukses2 = db.editDataDetailKriteria(/*modelProblem.getNamaKriteria(),
        modelProblem.getTipeKriteria(), modelProblem.getTipeNilaiKriteria(),
        modelProblem.getSkalaNilaiKriteria(), modelProblem.getBobot(*));
    if (sukses2 == true ){
        JOptionPane.showMessageDialog(null, "Data berhasil ditambahkan");
        view.tSimpan.setEnabled(false);
    }else{
        JOptionPane.showMessageDialog(null, "Penyimpanan gagal!");
    }
    }else{
        JOptionPane.showMessageDialog(null, "Penyimpanan gagal!");
    }
    }else{
    sukses1 = db.saveDataModelProblem();
    if (sukses1 == true){
        sukses2 = db.saveDataDetailKriteria(/*model.getNamaKriteria(), model.getTipeKriteria(),
            model.getTipeNilaiKriteria(), model.getSkalaNilaiKriteria(),
            model.getBobot(*));
        if ( sukses2 == true){
            JOptionPane.showMessageDialog(null, "Data berhasil ditambahkan");
            view.tSimpan.setEnabled(false);
        }else{
            JOptionPane.showMessageDialog(null, "Penyimpanan gagal!");
        }
    }else{
        JOptionPane.showMessageDialog(null, "Penyimpanan gagal!");
    }
    }
    tersimpan=true;
    }
}
}

public controllerIndeksKonsistensi (buatBaruSAW1 vsel, buatBaru4 v, buatBaru3 vseb
    , modelProblem m, CRUDdata db, awal a){
    this.db = db;
    this.model = m;
    this.vawal= a;
    this.view = v;
    this.viewSebelumnya = vseb;
    this.viewSelanjutnya = vsel;

    view.listenerTombolBatal(new listenerTombolBatal());
    view.listenerTombolSebelumnya(new listenerTombolSebelumnya());
    view.listenerTombolSelanjutnya(new listenerTombolSelanjutnya());
    view.listenerTombolSimpan(new listenerTombolSimpan());
    view.listenerhint(new listenerhint());
    view.internalFrameListener(new internalframelistener ());
}
}
}

```

## 2. Kode Program *Controller* Pengelolaan Data Tersimpan (controllerDaftarSPK.java)

```
package controler;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.IOException;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import model.CRUDdata;
import model.modelProblem;
import view.DaftarDM;
import view.awal;
import view.buatBaru1;
import view.buatBaruSAW1;
import view.detailSPK;

public class controllerDaftarSPK {
    private DaftarDM view;
    private detailSPK detail;
    private CRUDdata db;
    private modelProblem model;
    private buatBaru1 baru1;
    private buatBaruSAW1 buatsaw;
    private awal awal ;

    class listenerhint implements ActionListener {

        public void actionPerformed(ActionEvent e) {
            view.jPanel3.setVisible(true);
            model.setStatHinnt(true);
        }
    }

    class internalframelistener implements MouseListener{

        public void mouseClicked (MouseEvent e) {
            view.jPanel3.setVisible(false);
            model.setStatHinnt(false);
        }
        public void mousePressed(MouseEvent e) {}
        public void mouseReleased(MouseEvent e) {}
        public void mouseEntered(MouseEvent e) {}
        public void mouseExited(MouseEvent e) {}
    }

    class listenerTable implements MouseListener{

        public void mouseClicked(MouseEvent e) {
```

```
JTable table = view.getSelectedRow();
int baris =table.getSelectedRow();
String [] obj = {"Lihat Detail","Ubah","Gunakan","Hapus"};
int i = JOptionPane.showOptionDialog(null, "Pilih aksi untuk data Pendukung Keputusan terpilih :",
    "judul", JOptionPane.DEFAULT_OPTION,
    JOptionPane.PLAIN_MESSAGE, null, obj , null);
view.requestFocus();

boolean sukses = true;
if (i==0){
    try {
        Object [] op = db.loadDatadetail(baris);
        System.out.println("opo 0"+op[0]);
        detail.setData(op[0].toString(), op[1].toString(),
            op[2].toString(), op[3].toString(), op[4].toString(),
            model.getPembuatKeputusan());
        detail.setVisible(true);
        view.requestFocus();
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(null, "Terjadi kesalahan saat load data!",
            "Kesalahan", JOptionPane.WARNING_MESSAGE);
        System.out.println(ex.toString());
    }
} else if (i==1){
    try {
        sukses = db.loadDataEdit(baris);
        if (sukses == true){
            model.setStatEdit(true);
            baru1.setVisibleHint(model.getStatHint());
            baru1.setLoadToField(model.getNamaProblem(),
                model.getDefProblem(), model.getJmlKriteria(),
                model.getPembuatKeputusan());
            baru1.setVisible(true);
            view.setVisible(false);
        } else {
            JOptionPane.showMessageDialog(null, "Terjadi kesalahan saat load data!",
                "Kesalahan", JOptionPane.WARNING_MESSAGE);
        }
    } catch ( IOException | ClassNotFoundException ex) {
        System.out.println(ex.toString());
    }
} else if (i==2){
    try {
        sukses = db.loadDataEdit(baris);
        if (sukses == true){
            buatsaw.setVisibleHint(model.getStatHint());
            buatsaw.setVisible(true);
            view.setVisible(false);
        } else{
            JOptionPane.showMessageDialog(null, "Terjadi kesalahan saat load data!",
                "Kesalahan", JOptionPane.WARNING_MESSAGE);
        }
    } catch ( IOException | ClassNotFoundException ex) {
```

```
        System.out.println(ex.toString());
    }
} else if (i==3){
    byte ya = (byte) JOptionPane.showConfirmDialog(null, "Apakah Anda yakin akan menghapus
data?",
        "Konfirmasi",JOptionPane.YES_OPTION);
    if (ya == 0){
        sukses = db.deleteDataProblemSPK(baris);
        if (sukses ==true){
            JOptionPane.showMessageDialog(null, "Data berhasil dihapus.",
                "Informasi", JOptionPane.INFORMATION_MESSAGE);
            view.setTable(db.loadDataDaftarSPK());
        }else{
            JOptionPane.showMessageDialog(null, "Terjadi kesalahan saat penghapusan data!",
                "KESALAHAN", JOptionPane.WARNING_MESSAGE);
        }
    }else{
    }
} else{
}
}

public void mousePressed(MouseEvent e) {}
public void mouseReleased(MouseEvent e) {}
public void mouseEntered(MouseEvent e) {}
public void mouseExited(MouseEvent e) {}
}

class listenerTombolSelesai implements ActionListener {

    public void actionPerformed(ActionEvent e) {
        model.setDaftarIDproblem(null);
        awal.setVisible(true);
        view.setVisible(false);
    }
}

public controllerDaftarSPK (awal a,DaftarDM dm, detailSPK detail, buatBaru1 b1,
    buatBaruSAW1 b2, CRUDdata db, modelProblem model){
    this.baru1 = b1;
    this.buatsaw=b2;
    this.db=db;
    this.detail=detail;
    this.model=model;
    this.view=dm;
    this.awal=a;

    view.listenerTabel(new listenerTable ());
    view.listenerLHint(new listenerhint());
    view.internalFrameListener(new internalframelistener ());
    view.listenerTombolSelesai(new listenerTombolSelesai ());
}
}
```



## 3. Kode Program Perhitungan Bobot Kriteria (perbandinganBerpasangan.java)

```
package model;

public class perbandinganBerpasangan extends modelProblem{
    private float [] bobots ;
    private float[][] matrikNorm;

    public void setBobot (){
        byte baris, kolom = 0;
        byte dimensi = modelProblem.getJmlKriteria();
        float [][] matrik = modelProblem.getPerbandinganBerpasangan();

        float [] jmlKolom = new float [dimensi];
        matrikNorm = new float [dimensi][dimensi];
        bobots = new float[dimensi];

        //jumlah setiap kolom
        while (kolom<dimensi){
            baris=0;
            while (baris<dimensi){
                jmlKolom[kolom] = jmlKolom[kolom] + matrik[baris][kolom];
                baris++;
            }
            kolom++;
        }
        baris=0;
        //normalisasi: membagi nilai maing-masing matrik dengan jumla kolom yg bersangkutan
        while (baris<dimensi){
            kolom=0;
            while (kolom<dimensi){
                matrikNorm[baris][kolom]=matrik[baris][kolom]
                    /jmlKolom[kolom];
                kolom++;
            }
            baris++;
        }
        //=====
        super.setMatrikPerbandinganNormal(matrikNorm);
        //=====
        //mencari bobot dg mencari rata rata tiap baris dari matrik yang telah dinormalisasi
        baris =0;
        while (baris<dimensi){
            float jmlbaris=0;
            kolom = 0;
            while (kolom<dimensi){
                jmlbaris = matrikNorm[baris][kolom]+jmlbaris;
                kolom++;
            }
            bobots [baris]=jmlbaris/dimensi;
            System.out.println("bobot ke "+baris+"= "+bobots[baris]);
            baris++;
        }
    }
}
```

```

//=====
modelProblem.setBobotKriteria(bobots);
//=====
}
}

```

#### 4. Kode Program Perhitungan Konsistensi Bobot (indeksKonsistensi.java)

```

package model;

public class indexKonsistensi extends modelProblem{
    float lamdaMax, ci, cr;

    public void lamdaMax (float ri){
        byte dimensi = modelProblem.getJmlKriteria();
        float [][] perbBerpasangan = modelProblem.getPerbandinganBerpasangan();
        float [] bobot = super.getBobot();

        byte baris, kolom;
        float [] vektor= new float [dimensi];
        float lamda;

        //mencari vektor prioritas dg mengalikan matrik dengan matrik bobot
        baris=0;
        while (baris<dimensi){
            kolom=0;
            vektor[baris]=0;
            while (kolom<dimensi){
                vektor[baris]= vektor[baris]+
                    (perbBerpasangan [baris][kolom]*bobot[kolom]);
                kolom++;
            }
            System.out.println("vektor baris =" +vektor[baris]);
            baris++;
        }
        //membagi vektor prioritas dengan bobot
        baris=0;
        lamda=0;
        while (baris<dimensi){
            lamda= lamda+(vektor[baris]/bobot[baris]);
            baris++;
        }
        lamdaMax=lamda/dimensi;

        ci = (lamdaMax-dimensi)/(dimensi-1);
        cr=ci/ri;
        //=====
        modelProblem.setKonsistensi(ci, cr);
        //=====
    }
}

```

## 5. Kode Program Perhitungan Normalisasi Matrik Keputusan

(matrikKeputusan.java)

```
package model;

public class matrikKeputusan extends modelProblem{
    float [][] matrikKepNorm ;

    public void setNormalisasiMatrikKeputusan (){
        byte jmlKriteria = modelProblem.getJmlKriteria();
        boolean [] tipeKriteria = modelProblem.getTipeKriteria();
        int jmlAlternatif = modelProblem.getJmlAlternatif();
        double [][] matrikKeputusan = super.getMatrikKeputusan();

        matrikKepNorm = new float [jmlAlternatif][jmlKriteria];
        double [] pembagi=new double[jmlKriteria];

        //mencari nilai min cost criteria
        byte koloms =0;
        byte baris ;
        byte indek;

        while (koloms< jmlKriteria){
            baris =1;
            indek = 0;
            if (tipeKriteria[koloms]==false){
                while (baris<jmlAlternatif){
                    if(matrikKeputusan[baris][koloms]
                        < matrikKeputusan[indek][koloms]){
                        indek=baris;
                    }
                    baris++;
                }
            } else {
                while (baris<jmlAlternatif){
                    if(matrikKeputusan [baris][koloms]
                        > matrikKeputusan [indek][koloms]){
                        indek=baris;
                    }
                    baris++;
                }
            }
            pembagi[koloms]= matrikKeputusan[indek][koloms];
            koloms++;
        }

        koloms=0;
        while (koloms<jmlKriteria){
            baris=0;
            if (tipeKriteria [koloms]==false){
                while (baris<jmlAlternatif){
                    if (matrikKeputusan [baris][koloms]==0){
                        matrikKepNorm [baris][koloms]=0;
                    }
                }
            }
        }
    }
}
```

```

    }else{
        matrikKepNorm [baris][koloms] = (float) (pembagi[koloms]
            /matrikKeputusan [baris][koloms]);
    }
    baris++;
}
}else {
    while (baris<jmlAlternatif){
        if(matrikKeputusan [baris][koloms]==0){
            matrikKepNorm [baris][koloms]=0;
        }else{
            matrikKepNorm [baris][koloms] = (float) (matrikKeputusan [baris][koloms]
                /pembagi[koloms]);
        }
        baris++;
    }
    koloms++;
}
//=====
modelProblem.setMatrikKepNormal(matrikKepNorm);
//=====
}
}

```

## 6. Kode program Perhitungan Analisa Sensitivitas (analisaSensitifitas.java)

```

package model;

public class analisaSensitifitas extends modelProblem{
    private static float [] bobotBaru ;
    private static float [] nilaiAltBaru ;

    public void setSensitifitasBobot (int indexBobot, float bobotAtrPBaru ){
        float [] bobotlama = modelProblem.getBobot();

        bobotBaru= new float [ bobotlama.length];
        byte index=0;
        while (index<bobotBaru.length){
            if (index == indexBobot){
                bobotBaru[index] = bobotAtrPBaru;
            }else{
                bobotBaru[index]= ((1- bobotAtrPBaru )/ (1-bobotlama[indexBobot]))
                    *bobotlama[index];
            }
            index++;
        }
    }
    public float [] getSensitifitasBobot(){
        return bobotBaru;
    }
    public float [] getSensitifitasAlternatif (){
        return nilaiAltBaru;
    }
}

```

**LAMPIRAN C. PENGUJIAN *BLACK BOX* SISTEM**

No.	Fitur	Kasus	Hasil	Keterangan
5	Membuat / merancang SPK baru	<ul style="list-style-type: none"> <li>Klik tombol Buat Data Baru pada jendela awal</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan form isian Pemodelan Problem</li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ ] Gagal</li> </ul>
6	Inisiasi model problem	<ul style="list-style-type: none"> <li>Mengisi form isian dalam jendela Pemodelan Problem lalu klik tombol Selanjutnya</li> <li>Ketika klik tombol Batal</li> <li>Klik <i>Yes</i> pada dialog konfirmasi</li> <li>Apabila klik <i>No</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> <li>Apabila isian kurang lengkap</li> <li>Apabila mengisi kolom dengan karakter yang tidak sesuai</li> <li>Apabila jumlah kriteria yang diinputkan melebihi batas yang diperbolehkan sistem</li> <li>Apabila jumlah kriteria lebih dari 15</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan jendela Definisi Kriteria (1)</li> <li>Menampilkan dialog konfirmasi pembatalan pembuatan data baru</li> <li><i>Reset</i> isian dan menampilkan jendela awal</li> <li>Tidak memproses pembatalan</li> <li>Menampilkan <i>message box</i></li> <li>Mengabaikan isian yang tidak sesuai</li> <li>Menampilkan <i>message box</i></li> <li>Menampilkan dialog konfirmasi</li> </ul>	<ul style="list-style-type: none"> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ ] Gagal</li> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ ] Gagal</li> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ ] Gagal</li> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> <li>[ ] Gagal</li> <li>[ <input checked="" type="checkbox"/> ] Berhasil</li> </ul>



			[ ] Gagal			
7	Definisi kriteria	<ul style="list-style-type: none"> <li>Mengisi <i>form</i> isian Definisi Kriteria (1) dan klik tombol Selanjutnya</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan jendela Definisi kriteria (2)</li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>		
		<ul style="list-style-type: none"> <li>Mengisi <i>form</i> isian Definisi kriteria (2)</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan jendela Perbandingan Berpasangan</li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>		
		<ul style="list-style-type: none"> <li>Ketika klik tombol batal selama pendefinisian kriteria</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>		
		<ul style="list-style-type: none"> <li>Klik <i>Yes</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li><i>Reset</i> isian dan menampilkan jendela awal</li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>		
		<ul style="list-style-type: none"> <li>Apabila klik <i>No</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li>Tidak memproses pembatalan</li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>		
		<ul style="list-style-type: none"> <li>Apabila isian kurang lengkap</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan <i>message box</i></li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>		
		<ul style="list-style-type: none"> <li>Ketika klik tombol Sebelumnya di jendela Definisi Kriteria (2)</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan jendela Definisi Kriteria (1)</li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>		
		<ul style="list-style-type: none"> <li>Ketika klik tombol Sebelumnya di jendela Definisi Kriteria 1</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan jendela Pemodelan Problem</li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>		
		8	Melakukan perbandingan berpasangan	<ul style="list-style-type: none"> <li>Memilih macam input penilaian</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan tabel isian matrik perbandingan berpasangan sesuai dengan pilihan</li> </ul>	<ul style="list-style-type: none"> <li>[ √] Berhasil</li> <li>[ ] Gagal</li> </ul>

	<ul style="list-style-type: none"> <li>Mengisi matrik perbandingan berpasangan dan klik tombol selanjutnya</li> </ul>	<ul style="list-style-type: none"> <li>Mengkalkulasi bobot masing masing kriteria dan konsistensi bobot lalu menampilkan jendela Indeks konsistensi</li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Berhasil</li> <li><input type="checkbox"/> Gagal</li> </ul>	
	<ul style="list-style-type: none"> <li>Ketika klik tombol Batal</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Berhasil</li> <li><input type="checkbox"/> Gagal</li> </ul>	
	<ul style="list-style-type: none"> <li>Klik <i>Yes</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li><i>Reset</i> isian dan menampilkan jendela awal</li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Berhasil</li> <li><input type="checkbox"/> Gagal</li> </ul>	
	<ul style="list-style-type: none"> <li>Apabila klik <i>No</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li>Tidak memproses pembatalan</li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Berhasil</li> <li><input type="checkbox"/> Gagal</li> </ul>	
	<ul style="list-style-type: none"> <li>Apabila isian kurang lengkap</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan <i>message box</i></li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Berhasil</li> <li><input type="checkbox"/> Gagal</li> </ul>	
	<ul style="list-style-type: none"> <li>Apabila karakter yang dimasukkan dalam tabel isian matrik perbandingan berpasangan tipe penilaian manual input tidak sesuai</li> </ul>	<ul style="list-style-type: none"> <li>Merubah dengan nilai default</li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Berhasil</li> <li><input type="checkbox"/> Gagal</li> </ul>	
	<ul style="list-style-type: none"> <li>Ketika klik tombol Sebelumnya</li> </ul>	<ul style="list-style-type: none"> <li>Menampilkan jendela Definisi Kriteria (2)</li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Berhasil</li> <li><input type="checkbox"/> Gagal</li> </ul>	
9	Cek konsistensi bobot	<ul style="list-style-type: none"> <li>Ketika konsistensi bobot dapat diterima</li> </ul>	<ul style="list-style-type: none"> <li><i>Enable</i> tombol Simpan</li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Berhasil</li> <li><input type="checkbox"/> Gagal</li> </ul>

	<ul style="list-style-type: none"> <li>• Klik tombol Simpan</li> </ul>	<ul style="list-style-type: none"> <li>• Menyimpan data dalam database</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Ketika klik tombol Selanjutnya tanpa melakukan penyimpanan</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan dialog konfirmasi</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Ketika klik tombol Selesai</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan jendela awal</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Ketika klik tombol Sebelumnya</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan jendela Perbandingan Berpasangan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
10	Menalari alternatif pada tiap kriteria (matrik keputusan)	<ul style="list-style-type: none"> <li>• Menginputkan jumlah alternatif lalu klik tombol Mulai</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan tabel isian matrik keputusan sesuai inputan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	<ul style="list-style-type: none"> <li>• Mengisi matrik keputusan dan klik tombol Selanjutnya</li> </ul>	<ul style="list-style-type: none"> <li>• Mengkalkulasi nilai preferensi alternatif, mengurutkan berdasar nilai terbesar, menampilkan jendela Ranking Keputusan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Ketika klik tombol Batal</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Klik <i>Yes</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Reset</i> isian dan menampilkan jendela awal</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Apabila klik <i>No</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li>• Tidak memproses pembatalan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	

	<ul style="list-style-type: none"> <li>• Ketika isian tidak lengkap</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>message dialog</i></li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Apabila jumlah alternatif yang diinputkan melebihi batas yang diperbolehkan sistem</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan <i>message dialog</i></li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Apabila mengisi kolom isian jumlah kriteria dengan karakter yang tidak sesuai</li> </ul>	<ul style="list-style-type: none"> <li>• Mengabaikan inputan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Apabila mengisi matrik keputusan pada kolom numerik dengan karakter bukan angka atau titik</li> </ul>	<ul style="list-style-type: none"> <li>• Mengubah inputan dengan <i>default value</i></li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
11	Melihat ranking alternatif keputusan	<ul style="list-style-type: none"> <li>• Melihar nilai preferensi dan ranking alternatif berdasarkan nilai terbesar</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan hasil dari proses perhitungan dalam tabel berdasarkan urutan nilai terbesar</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	<ul style="list-style-type: none"> <li>• Ketika klik tombol Sensitivitas</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan jendela analisa sensitivitas</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Ketika klik tombol Batal</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan dialog konfirmasi pembatalan penilaian alternatif</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Klik <i>Yes</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li>• <i>Reset</i> isian dan menampilkan jendela awal</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Apabila klik <i>No</i> pada dialog konfirmasi pembatalan pembuatan data baru</li> </ul>	<ul style="list-style-type: none"> <li>• Tidak memproses pembatalan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	

	<ul style="list-style-type: none"> <li>• Ketika klik tombol Sebelumnya</li> </ul>	<ul style="list-style-type: none"> <li>• Menampilkan jendela Penilaian Alternatif</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
12	Analisa sensitivitas bobot dan alternatif	<ul style="list-style-type: none"> <li>• Input informasi analisa sensitivitas pada isian yang tersedialalu klik tombol Sensitivitas</li> </ul>	<ul style="list-style-type: none"> <li>• Mengkalkulasi sensitivitas dan menampilkan hasil dalam bentuk grafik</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal
	<ul style="list-style-type: none"> <li>• Ketika klik tombol Selesai</li> </ul>	<ul style="list-style-type: none"> <li>• Menutup jendela analisa sensitivitas</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Apabila input karakter pada kolom Perubahan Bobot tidak sesuai</li> </ul>	<ul style="list-style-type: none"> <li>• Mengabaikan inputan</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• apabila bobot baru yang diinputkan nilainya tidak sesuai dengan yang diperbolehkan sistem</li> </ul>	<ul style="list-style-type: none"> <li>• menampilkan pesan dialog</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Apabila check list pilihan perhitungan belum terisi</li> </ul>	<ul style="list-style-type: none"> <li>• menampilkan pesan dialog</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	
	<ul style="list-style-type: none"> <li>• Apabila pilihan sajian grafik belum terisi</li> </ul>	<ul style="list-style-type: none"> <li>• menampilkan pesan dialog</li> </ul>	<input checked="" type="checkbox"/> Berhasil <input type="checkbox"/> Gagal	

Penguji,

( \_\_\_\_\_ )