



**SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PRIORITAS
PENDISTRIBUSIAN AIR BERSIH PADA DAERAH KRISIS AIR
DI WILAYAH KABUPATEN BONDOWOSO MENGGUNAKAN
METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)**

SKRIPSI

oleh :

Sofy Indra Wahyudi

112410101091

PROGRAM ILMU KOMPUTER

UNIVERSITAS JEMBER

2017



**SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PRIORITAS
PENDISTRIBUSIAN AIR BERSIH PADA DAERAH KRISIS AIR
DI WILAYAH KABUPATEN BONDOWOSO MENGGUNAKAN
METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)**

SKRIPSI

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

oleh :

Sofy Indra Wahyudi

112410101091

PROGRAM ILMU KOMPUTER

UNIVERSITAS JEMBER

2017

PERSEMBAHAN

Dengan bangga dan kerendahan hati, skripsi ini saya persembahkan untuk :

1. Allah SWT;
2. Nabi Muhammad SAW;
3. Kedua Orang Tua, Ayah Djuali Hadi Siswanto dan Ibu Halimatus Sadiyah;
4. Seluruh keluarga besar Hamzah dan Sanafi;
5. Guru-guru sejak taman kanak-kanak hingga perguruan tinggi;
6. Seluruh teman-teman yang selalu memberikan bantuan dan dukungan;
7. Untuk perempuan yang sedang dalam rahasia-Nya;
8. Almamater tercinta Program Studi Sistem Informasi Universitas Jember.

MOTTO

“Great men are not born great, they grow great...”

- Mario Puzo -



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Sofy Indra Wahyudi

NIM : 112410101091

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Pendukung Keputusan Penentuan Prioritas Pendistribusian Air Bersih Pada Daerah Krisis Air Di Wilayah Kabupaten Bondowoso Menggunakan Metode *Simple Additive Weighting* (SAW)”, adalah benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 8 Desember 2017

Yang menyatakan,

Sofy Indra Wahyudi

NIM. 112410101091

PENGESAHAN PEMBIMBING

Skripsi berjudul “**Sistem Pendukung Keputusan Penentuan Prioritas Pendistribusian Air Bersih Pada Daerah Krisis Air Di Wilayah Kabupaten Bondowoso Menggunakan Metode *Simple Additive Weighting* (SAW)**”, telah diuji dan disahkan pada :

Hari, tanggal : Jum’at, 8 Desember 2017

Tempat : Program Ilmu Komputer Universitas Jember

Disetujui oleh :

Pembimbing I,

Pembimbing II,

Windi Eka Yulia Retnani, S.Kom., MT
NIP. 198403052010122002

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

SKRIPSI

**SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PRIORITAS
PENDISTRIBUSIAN AIR BERSIH PADA DAERAH KRISIS AIR
DI WILAYAH KABUPATEN BONDOWOSO MENGGUNAKAN
METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)**

oleh :

Sofy Indra Wahyudi

112410101091

Pembimbing :

Dosen Pembimbing Utama : Windi Eka Yulia Retnani, S.Kom., MT.

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

PENGESAHAN

Skripsi berjudul “**Sistem Pendukung Keputusan Penentuan Prioritas Pendistribusian Air Bersih Pada Daerah Krisis Air Di Wilayah Kabupaten Bondowoso Menggunakan Metode *Simple Additive Weighting* (SAW)**”, telah diuji dan disahkan pada :

Hari, tanggal : Jum’at, 8 Desember 2017

Tempat : Program Ilmu Komputer Universitas Jember

Tim Penguji :

Penguji I,

Penguji II,

Prof. Dr. Saiful Bukhori, ST., M.Kom
NIP. 196811131994121001

Yanuar Nurdiansyah, ST., M.Cs
NIP. 198201012010121004

Mengesahkan
Ketua Program Studi,

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

RINGKASAN

“Sistem Pendukung Keputusan Penentuan Prioritas Pendistribusian Air Bersih Pada Daerah Krisis Air Di Wilayah Kabupaten Bondowoso Menggunakan Metode *Simple Additive Weighting* (SAW)”; Sofy Indra Wahyudi, 112410101091; 2017; 132 halaman; Program Ilmu Komputer Universitas Jember.

Kekeringan merupakan salah satu jenis bencana alam yang terjadi secara perlahan (*slow onset disaster*), berlangsung lama sampai hujan tiba, berdampak sangat luas, dan bersifat lintas sektor (ekonomi, sosial, kesehatan, pendidikan, dan lain-lain). Jawa Timur khususnya Kabupaten Bondowoso rentan terhadap bencana kekeringan. Di Kabupaten Bondowoso tercatat 10 kecamatan yang teridentifikasi mengalami bencana kekeringan. Pemerintah Daerah telah mengupayakan beberapa solusi diantara membuat waduk, hujan buatan, serta mendistribusikan air bersih kepada seluruh kecamatan yang mengalami kekeringan. Pendistribusian air bersih yang tidak tepat sasaran inilah yang memicu tidak meratanya pembagian air bersih pada setiap daerah krisis.

Pembuatan sistem penunjang keputusan pendistribusian air bersih pada daerah yang mengalami krisis air bersih dengan menggunakan metode *Simple Additive Weighting* (SAW). Kelebihan dari metode SAW dibanding dengan model pengambil keputusan lainnya terletak pada kemampuannya untuk melakukan penilaian secara lebih tepat karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan, selain itu SAW juga dapat menyeleksi alternatif terbaik dari sejumlah alternatif yang ada karena adanya proses perankingan setelah menentukan bobot untuk setiap atribut. Tentu sangat relevan untuk pengambilan keputusan pendistribusian air bersih ke lokasi krisis air. Sistem ini dimaksudkan untuk membantu mengatasi masalah yang terjadi, dan sistem ini lebih bersifat memberikan dukungan atau pertimbangan bagi pihak penyeleksi atau pihak berwenang, dengan menggunakan tampilan web yang sangat *user friendly*.

PRAKATA

Alhamdulillah, puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “**Sistem Pendukung Keputusan Penentuan Prioritas Pendistribusian Air Bersih Pada Daerah Krisis Air Di Wilayah Kabupaten Bondowoso Menggunakan Metode *Simple Additive Weighting* (SAW)**”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Ilmu Komputer Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih yang tidak terhingga kepada :

1. Kedua orang tua penulis, Ayahanda Djuali Hadi Siswanto dan Ibunda Halimatus Sadiyah tercinta, tiada kata yang bisa menggantikan segala kasih sayang, usaha, do'a serta dukungan kepada penulis disetiap waktu;
2. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
3. Windi Eka Yulia Retnani, S.Kom., MT., selaku Dosen Pembimbing Utama dan Oktalia Juwita S.Kom., M.MT, selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
4. Muhamad Arief Hidayat S.Kom., M.Kom., selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
5. Prof. Dr. Saiful Bukhori, ST., M.Kom, selaku Dosen Penguji I dan Yanuar Nurdiansyah ST., M.Cs selaku Dosen Penguji II;
6. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Ilmu Komputer Universitas Jember yang memberikan bantuan selama perkuliahan;
7. Saefuddin Suhri, S.Pd, selaku pembimbing penelitian di kantor Badan Penanggulangan Bencana Daerah Kabupaten Bondowoso beserta staf;

8. Keluarga besar di Bondowoso dan Malang, Muhammad Subhan, Ratna Dewi, Sevinna Artrinita, Salva Radityal Balya, Ifa Febrianti, Ina Yulianti, Dimas Nuansa, Shellavie Erlyana, Arif Wicaksono, Ratih Permatasari;
9. Keluarga 3E SPASA, Agung Laksono, Avieka Joeniarsa, Dino Eka, Diky Purwo, Ismad Sulhi, Jibril, Lalu Rizky, Reza Yomi, Rhiza Fajar terima kasih atas semua bantuannya;
10. Teman “KRT 2G”, Adip Hafifi, Anang Syaiful Rohmat, Annas Fathir, Danang Mahaldi, Prisan Bastio, Rois Aminullah, sukses untuk kalian semua;
11. NEFOTION 2011, Alif Yanuar, Amris Faisal, Arie Fandi, Andre Bhaskoro, Dany Fairuz, Egit Megantara, Guntur Gibran, Resha Muhammad, Rizka Amiruddin, Yogi Ardiansyah, dan semua yang belum disebutkan terima kasih atas segala bantuan, kenangan dan kebersamaan selama ini;
12. Semua pemilik warung makan, tambal ban, tukang parkir, percetakan dan foto kopi, serta seniman jalanan yang menghibur dikala jenuh akan tugas kuliah, terima kasih kalian telah menginspirasi dan berjasa untuk karya ini;
13. Semua pihak yang telah membantu terselesaikannya skripsi ini baik pikiran, tenaga, materi, maupun yang lainnya demi kelancaran penulisan skripsi ini.

Penulis menyadari bahwa karya ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, Desember 2017

Penulis

DAFTAR ISI

PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING.....	v
SKRIPSI.....	vi
RINGKASAN	viii
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
BAB 1. PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	4
1.3. Tujuan.....	5
1.4. Batasan Masalah	5
3.1. Sistematika Penulisan	6
BAB 2. TINJAUAN PUSTAKA	7
2.1 Sistem Pendukung Keputusan	7
2.2 Bencana Kekeringan.....	8
2.3 Metode <i>Simple Additive Weighting</i>	9
2.3.1 Pengertian Metode <i>Simple Additive Weighting</i> (SAW).....	9
2.3.2 Langkah-langkah Metode <i>Simple Additive Weighting</i> (SAW).....	10
2.4 Model <i>Waterfall</i>	10
BAB 3. METODOLOGI PENELITIAN.....	17
3.1 Jenis Penelitian	17
3.2 Tempat dan Waktu Penelitian	17
3.3 Tahapan Penelitian	17

3.3.1 Teknik Pengumpulan Data	18
3.3.2 Tahap Analisis Data.....	19
3.3.3 Tahap Pengembangan Sistem.....	20
BAB 4. PENGEMBANGAN SISTEM.....	21
4.1 Analisis Kebutuhan Sistem.....	21
4.1.1 Kebutuhan Fungsional.....	21
4.1.2 Kebutuhan Non-Fungsional.....	22
4.2 Deskripsi Umum Sistem.....	22
4.2.1 SOP (<i>Statement of Purpose</i>) Sistem	22
4.2.2 Fungsi Sistem	23
4.3 Desain Sistem	23
4.3.1 <i>Business Process</i>	24
4.3.2 <i>Use Case Diagram</i>	25
4.3.3 <i>Skenario</i>	28
4.3.4 <i>Activity Diagram</i>	31
4.3.5 <i>Sequence Diagram</i>	32
4.3.6 <i>Class Diagram</i>	34
4.3.7 <i>Entity Relationship Diagram (ERD)</i>	36
4.4 Penulisan Kode Program	36
4.5 Pengujian Sistem	37
4.5.1 Metode <i>White Box</i>	37
4.5.2 Metode <i>Black Box</i>	40
BAB 5. HASIL DAN PEMBAHASAN.....	41
5.1 Pembahasan Sistem	41
5.1.1. Pembahasan Implementasi Metode SAW.....	41
5.1.2. Pembahasan Hasil Pengujian.....	53
5.2 Sistem Pendukung Keputusan Penentuan Prioritas Pendistribusian Air ...	54
5.2.1 Tampilan <i>Login</i>	54

5.2.2 Tampilan Utama Sistem	55
5.2.3 Tampilan Fitur Survei.....	55
5.2.4 Tampilan Fitur Mengelola Data Survei Detail	56
5.2.5 Tampilan Fitur Analisis Prioritas Distribusi Air	56
BAB 6. PENUTUP	58
6.1 Kesimpulan.....	58
6.2 Saran	59
DAFTAR PUSTAKA	60
LAMPIRAN	62
1. Lampiran <i>Skenario</i>	62
2. Lampiran <i>Activity Diagram</i>	85
3. Lampiran <i>Sequence Diagram</i>	91
4. Lampiran Implementasi <i>Coding</i>	96
5. Lampiran Pengujian <i>Black Box</i>	113

DAFTAR TABEL

Tabel 2.1 Intensitas Kekeringan.....	8
Tabel 4.1 Penjelasan <i>Use Case</i>	25
Tabel 4.2 Skenario Menambah Data Survei	28
Tabel 4.3 Skenario Menganalisis Prioritas Distribusi Air Bersih.....	29
Tabel 4.4 Test Case Analisis Prioritas	40
Tabel 5.1 Data Kriteria.....	41
Tabel 5.2 Data Subkriteria	42
Tabel 5.3 Jarak Tempuh Ke Sumber Air	42
Tabel 5.4 Standar Kebutuhan Air Rumah Tangga.....	43
Tabel 5.5 Jenis Partikel Tanah	43
Tabel 5.6 Kriteria Distribusi Curah Hujan.....	44
Tabel 5.7 Tingkat Elevasi Air Sumur	44
Tabel 5.8 Tabel Sampel Alternatif	45
Tabel 5.9 Penilaian Alternatif	45
Tabel 5.10 Hasil Normalisasi Matrik	49
Tabel 5.11 Perhitungan Perangkingan Awal.....	50
Tabel 5.12 Hasil Akhir Perhitungan Ranging	51
Tabel 5.13 Perbandingan Hasil Nilai Perangkingan	52

DAFTAR GAMBAR

Gambar 2.1 Alur Model Waterfall (Pressman, 2009).....	11
Gambar 2.2 Contoh Listing Program	14
Gambar 2.3 Contoh Diagram Alir (Pressman, 2005).....	14
Gambar 3.1 Diagram Alir Penelitian (Sumber : Hasil Analisis, 2016).....	18
Gambar 3.2 Flowchart Metode SAW pada sistem	19
Gambar 4.1 Business Process Sistem.....	24
Gambar 4.2 Use Case Diagram	25
Gambar 4.3 Activity Diagram Menganalisis Prioritas Distribusi Air Bersih.....	31
Gambar 4.4 Activity Diagram Mengelola Data Survei Detail Surveyor	32
Gambar 4.5 Sequence Diagram Menganalisis Prioritas Distribusi Air Bersih.....	33
Gambar 4.6 Sequence Diagram Menginput Data Survei Detail oleh Surveyor	34
Gambar 4.7 Class Diagram Sistem	35
Gambar 4.8 ERD Sistem.....	36
Gambar 4.9 Listing Program Fitur Analisis Prioritas.....	38
Gambar 4.10 Diagram Alir Fitur Analisis Prioritas.....	38
Gambar 5.1 Hasil Nilai Perangkingan Sistem	52
Gambar 5.2 Tampilan halaman login	54
Gambar 5.3 Tampilan halaman utama sistem	55
Gambar 5.4 Tampilan halaman menu Survei.....	55
Gambar 5.5 Tampilan fitur tambah data survei detail.....	56
Gambar 5.6 Tampilan fitur analisis prioritas pendistribusian air.....	57
Gambar 5.7 Tampilan hasil analisis prioritas	57

BAB 1. PENDAHULUAN

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

1.1. Latar Belakang

Air bersih merupakan suatu kebutuhan utama dan tidak dapat diganti dalam kehidupan manusia di alam semesta ini, oleh karena itu ketersediaannya harus tetap terjamin dalam waktu, kuantitas maupun kualitasnya. Kebutuhan akan air bersih ini menjadi masalah hampir di semua negara. Permasalahan ini disebabkan karena ketidakseimbangan antara permintaan dan persediaan air bersih tersebut. Permintaan terus bertambah sedangkan persediaan air bersih cenderung berkurang karena berkurangnya debit sumber air baku seperti mata air, sungai, danau dan air tanah sebagai akibat dari degradasi lingkungan serta kurangnya perawatan. Menurut Badan Meteorologi Klimatologi dan Geofisika (BMKG) memasuki bulan April sebanyak 102 wilayah zona musim atau 29.8% dari seluruh wilayah zona musim, akan memasuki musim kemarau. Pada bulan Mei sebanyak 99 wilayah atau 28.9%, dan pada bulan Juni sebanyak 84 wilayah atau 24.6% dari seluruh wilayah zona musim. Tentu memasuki musim kemarau akan menyebabkan kekeringan di beberapa wilayah Indonesia (Badan Meteorologi Klimatologi dan Geofisika, 2015).

Kekeringan merupakan salah satu jenis bencana alam yang terjadi secara perlahan (*slow onset disaster*), berlangsung lama sampai hujan tiba, berdampak sangat luas, dan bersifat lintas sektor (ekonomi, sosial, kesehatan, pendidikan, dan lain-lain). Kekeringan adalah keadaan kekurangan pasokan air pada suatu daerah dalam masa yang berkepanjangan (beberapa bulan hingga bertahun-tahun). Biasanya kejadian ini muncul bila suatu wilayah secara terus-menerus mengalami curah hujan di bawah rata-rata. Musim kemarau yang panjang akan menyebabkan kekeringan karena cadangan air tanah akan habis akibat penguapan (evaporasi), transpirasi, ataupun penggunaan lain oleh manusia.

Menurut Rahmato. D (1991) kekeringan didefinisikan sebagai pengurangan persediaan air atau kelembaban yang bersifat sementara secara signifikan di bawah normal atau volume yang diharapkan untuk jangka waktu khusus. Dampak kekeringan muncul sebagai akibat dari kekurangannya air, atau perbedaan-perbedaan antara permintaan dan persediaan air. Kekeringan dapat menjadi bencana alam apabila mulai menyebabkan suatu wilayah kehilangan sumber pendapatan akibat gangguan pada pertanian dan ekosistem yang ditimbulkannya. Dampak ekonomi dan ekologi kekeringan merupakan suatu proses sehingga batasan kekeringan dalam setiap bidang dapat berbeda-beda. Namun demikian, suatu kekeringan yang singkat tetapi intensif dapat pula menyebabkan kerusakan yang signifikan.

Berdasarkan data terakhir dari Badan Nasional Penanggulangan Bencana (BNPB), sebanyak 12 Provinsi mengalami kekeringan, serta 77 kabupaten atau kota dan 526 kecamatan yang ada di Indonesia mengalami bencana kekeringan. Menurut hasil pantauan, wilayah yang tidak turun hujan selama dua bulan terakhir itu yakni, adalah Jawa Timur dan khususnya Kabupaten Bondowoso rentan terhadap bencana kekeringan. Menurut Badan Penanggulangan Bencana Daerah (BPBD) di Kabupaten Bondowoso tercatat 7 kecamatan yang teridentifikasi mengalami bencana kekeringan. Dan bencana kekeringan ini terus berlanjut dari tahun ke tahun. Pemerintah Daerah telah mengupayakan beberapa solusi diantara membuat waduk, hujan buatan, serta mendistribusikan air bersih kepada seluruh kecamatan yang mengalami kekeringan (Friska, 2015). Namun pada kenyataannya, pendistribusian air bersih tidak berjalan baik. Ada beberapa kecamatan yang sebenarnya persediaan air bersih masih cukup, malah mendapatkan bantuan juga. Oleh sebab itu untuk mengurangi kesalahan dalam pendistribusian air bersih, BPBD perlu selektif dalam menentukan lokasi yang akan menerima bantuan air bersih. Bagi masyarakat di Bondowoso yang mengalami kekurangan pasokan air bersih, tentu akan sangat dirugikan apabila mereka tidak mendapatkan jatah pasokan air bersih yang cukup.

Penelitian sebelumnya mengenai pendistribusian air bersih dengan judul Penentuan Distribusi Air Bersih di Kabupaten X menggunakan Metode SAW (*Simple Additive Weighting*), menyatakan perhitungan nilai menggunakan tahapan dan langkah-langkah proses yang sama dan jelas. Nilai setiap kecamatan dipelakukan secara adil sesuai dengan ketentuan atau ketetapan yang sama seperti bobot kriteria, dan persamaan perhitungan. Berdasarkan hasil perhitungan matriks normalisasi dengan bobot setiap kriteria maka didapat nilai setiap kecamatan, yaitu Kecamatan A (29), Kecamatan B (28.75), Kecamatan C (28.2). Maka nilai terbesar ada pada Kecamatan A yaitu kecamatan yang terpilih sebagai alternatif terbaik, dengan kata lain Kecamatan A merupakan daerah yang layak diprioritaskan untuk pendistribusian air bersih (Merry Agustina & A. Mutatkin Bakti, 2015).

Penelitian lain yang berjudul Sistem Pendukung Penentuan Keputusan Distribusi Pengadaan Air Bersih Dengan Metode *Technique for order Preference by Similarity to Ideal Solution* (TOPSIS) (Studi Kasus : Kabupaten Sragen), menyatakan bahwa metode TOPSIS dapat diterapkan untuk menentukan urutan kecamatan yang akan mendapat bantuan pengadaan air bersih. Dimana dalam menentukan pengalokasian distribusi air bersih, ada beberapa kriteria yang digunakan antara lain jumlah keluarga dan jumlah dukuh yang membutuhkan air bersih disetiap daerah yang sesuai dengan proposal yang diajukan tiap kecamatan/kelurahan, jumlah keseluruhan jiwa per daerah, serta jumlah tangki yang akan di distribusikan. Penelitian ini melakukan pengujian secara manual menggunakan *Microsoft Excel* dan sistem yang telah dibuat, dan memberikan hasil yang sama dalam perhitungannya. Kekurangan pada penelitian ini, meskipun hasil alternatif dengan ranking tertinggi merupakan solusi yang terbaik, namun belum tentu ranking tertinggi tersebut adalah yang terdekat dari solusi ideal. Sehingga perlu dilakukan perhitungan kembali untuk memastikannya dan hal ini tentu akan menghambat proses pendistribusian air bersih. Metode TOPSIS kurang baik jika digunakan dalam mendapatkan bobot yang memperhitungkan hubungan antara kriteria. Walaupun dapat dilakukan dengan

perbandingan berpasangan, tetapi membutuhkan matriks dan perhitungan yang lebih rumit (Henrykus Bagus Handoko, 2016).

Berdasarkan uraian diatas maka dibutuhkan suatu sistem yang dapat membantu dalam menentukan prioritas pendistribusian air bersih yang dapat menunjukan tempat atau daerah untuk mendapatkan skala prioritas pendistribusian air bersih di Bondowoso. Oleh karena itu, peneliti akan merancang sistem penunjang keputusan pendistribusian air bersih pada daerah yang mengalami krisis air bersih dengan menggunakan metode *Simple Additive Weighting* (SAW). Kelebihan dari metode SAW dibanding dengan model pengambil keputusan lainnya terletak pada kemampuannya untuk melakukan penilaian secara lebih tepat karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan, selain itu SAW juga dapat menyeleksi alternatif terbaik dari sejumlah alternatif yang ada karena adanya proses perangkingan setelah menentukan bobot untuk setiap atribut (Kusumadewi, 2006). Asumsi yang mendasari metode SAW adalah setiap atribut bersifat independen, jadi tidak akan saling mempengaruhi atribut lain. Skoring dengan metode ini diperoleh dengan menambahkan kontribusi dari setiap atribut (Kahraman, 2008). Adanya beberapa kesamaan kriteria yang digunakan pada penelitian sebelumnya dengan yang terjadi pada daerah kekeringan di Kabupaten Bondowoso, maka penelitian ini tentu sangat berguna untuk pengambilan keputusan pendistribusian air bersih ke lokasi krisis air yang ada pada Kabupaten Bondowoso.

Sistem ini dimaksudkan untuk membantu mengatasi masalah yang terjadi, dan sistem ini lebih bersifat memberikan dukungan atau pertimbangan bagi pihak penyeleksi atau pihak berwenang dan membantu pihak penyeleksi atau pihak berwenang dalam pengambilan keputusan sehingga dapat meningkatkan efektivitas dan efisiensi dari proses pengambilan keputusan itu sendiri.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dirumuskan beberapa permasalahan sebagai berikut :

1. Bagaimana merancang dan membangun sebuah sistem pendukung keputusan penentuan prioritas pendistribusian air bersih pada daerah krisis air di wilayah Kabupaten Bondowoso menggunakan metode *Simple Additive Weighting* (SAW) yang *user friendly* ?.
2. Bagaimana menerapkan metode *Simple Additive Weighting* (SAW) pada proses pendukung keputusan penentuan prioritas pendistribusian air bersih pada daerah krisis air di wilayah Kabupaten Bondowoso ?.

1.3. Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Merancang dan membangun sebuah sistem pendukung keputusan penentuan prioritas pendistribusian air bersih pada daerah krisis air di wilayah Kabupaten Bondowoso menggunakan metode *Simple Additive Weighting* (SAW) yang *user friendly*.
2. Menerapkan metode *Simple Additive Weighting* (SAW) pada proses pendukung keputusan penentuan prioritas pendistribusian air bersih pada daerah krisis air di wilayah Kabupaten Bondowoso.

1.4. Batasan Masalah

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Penentuan lokasi pendistribusian air bersih ditentukan oleh para pengambil keputusan, sistem hanya memberikan rekomendasi lokasi pendistribusian air bersih yang sesuai dengan kriteria.
2. Bobot kriteria pada sistem ini resmi dari Badan Penanggulangan Bencana Daerah yang mengacu pada buku pedoman tata cara penyusunan program pendistribusian air bersih dan dihitung menggunakan metode *Simple Additive Weighting* (SAW).
3. Sistem ini menentukan lokasi pendistribusian air bersih hanya dalam lingkup Kabupaten Bondowoso.

3.1. Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut :

BAB 1. PENDAHULUAN

Bab ini menjelaskan latar belakang, perumusan masalah, tujuan dan sistematika penulisan.

BAB 2. TINJAUAN PUSTAKA

Bab ini berisi tentang kajian materi, penelitian terdahulu dan informasi apa saja yang digunakan dalam penelitian ini. Dimulai dari kajian pustaka mengenai penelitian terdahulu hingga model *waterfall*.

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode penelitian yang akan digunakan. Mulai dari tahap studi literatur, pengumpulan data, analisis, dan perancangan sistem.

BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan tentang analisis dan perancangan sistem yang dikembangkan.

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang telah dilakukan dengan memaparkan hasil penelitian dan hasil percobaan pengimplementasian sistem.

BAB 6. PENUTUP

Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan teori-teori dan pustaka yang digunakan dalam penelitian. Teori yang dibahas meliputi sistem pendukung keputusan, bencana kekeringan, metode *Simple Additive Weighting*, model *Waterfall*.

2.1 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) merupakan penggabungan beberapa sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah-masalah semi struktur (Morton, 1978). Menurut Turban (2001), SPK dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK adalah :

1. SPK memperluas kemampuan pengambil keputusan dalam memproses data/informasi bagi pemakainya,
2. SPK membantu pengambil keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur,
3. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan,
4. Walaupun suatu SPK mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun dia dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan.

Dari pengertian diatas dapat dijelaskan bahwa Sistem Pendukung Keputusan merupakan alat yang digunakan untuk membantu pengambilan keputusan, dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat.

2.2 Bencana Kekeringan

Kekeringan adalah merupakan salah satu bencana yang sulit dicegah dan datang berulang dan terjadi secara perlahan (*slow onset disaster*). Secara umum pengertian kekeringan adalah hubungan antara ketersediaan air yang jauh di bawah kebutuhan air baik untuk kebutuhan hidup, pertanian, kegiatan ekonomi dan lingkungan. Kekeringan merupakan salah satu fenomena yang terjadi sebagai dampak penyimpangan iklim global seperti *El Nino* dan Osilasi Selatan. Dewasa ini bencana kekeringan semakin sering terjadi bukan saja pada periode tahun-tahun *El Nino*, tetapi juga pada periode tahun dalam keadaan kondisi normal.

Menurut Badan Penanggulangan Bencana Daerah (BPBD) Bondowoso, ada beberapa wilayah kecamatan di Bondowoso mengalami kekeringan sosial ekonomi (Friska, 2015). Kekeringan sosial ekonomi terjadi berhubungan dengan berkurangnya pasokan komoditi yang bernilai ekonomi dari kebutuhan normal sebagai akibat dari terjadinya kekeringan meteorologis, pertanian dan hidrologis (Shelia B. Red, 1995). Intensitas kekeringan berdasarkan definisi hidrologis dapat dilihat pada Tabel 2.1.

Tabel 2.1 Intensitas Kekeringan

No.	Kategori	Ketersediaan Air (L/orang/hari)	Pemenuhan Kebutuhan Untuk	Jarak ke Sumber Air (Km)
1.	Kering Langka Terbatas	> 30 < 60	Minum, masak, cuci alat masak/makan, mandi terbatas	0,1 – 0,5
2.	Kering Langka	> 10 < 30	Minum, masak, cuci alat masak/makan, mandi	0,5 - 3
3.	Kering Kritis	< 30	Minum, masak	> 3

2.3 Metode *Simple Additive Weighting*

Pada sub bab ini dijelaskan tentang metode *Simple Additive Weighting* (SAW) diantaranya, pengertian metode *Simple Additive Weighting* (SAW), langkah-langkah metode *Simple Additive Weighting* (SAW).

2.3.1 Pengertian Metode *Simple Additive Weighting* (SAW)

Menurut Kusumadewi (2006), metode SAW sering juga dikenal dengan istilah metode penjumlahan terbobot. Konsep dasar SAW adalah mencari penjumlahan terbobot dari kinerja setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat dibandingkan dengan semua rating alternatif yang ada.

$$r_{ij} \begin{cases} \frac{x_{ij}}{\max x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases} \dots\dots \text{persamaan 1}$$

Keterangan :

r_{ij} = nilai rating kinerja normalisasi

X_{ij} = nilai yang dimiliki dari setiap kriteria

Max_{ij} = nilai terbesar dari setiap kriteria

Min_{ij} = nilai terkecil dari setiap kriteria

Atribut keuntungan (*benefit*) = nilai terbesar adalah terbaik

Atribut biaya (*cost*) = nilai terkecil adalah terbaik

Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j : $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai berikut:

$$V_i = \sum_{j=1}^n w_j r_{ij} \dots\dots \text{persamaan 2}$$

Keterangan :

V_i = nilai preferensi

w_j = bobot rangking

r_{ij} = rating kinerja ternormalisasi

Nilai V_i lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

2.3.2 Langkah-langkah Metode *Simple Additive Weighting* (SAW)

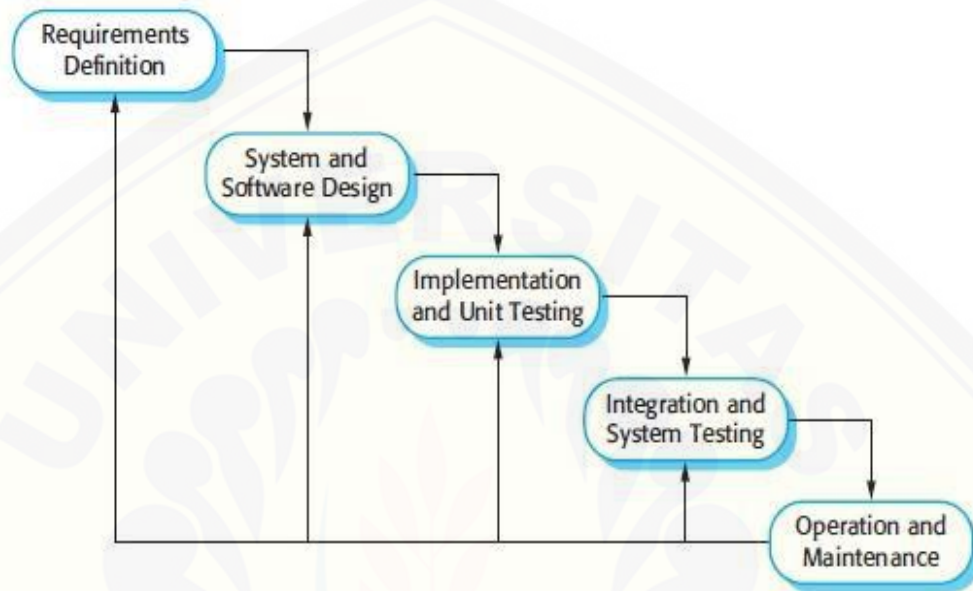
Menurut Kusumadewi (2006), dalam jurnal Implementasi Metode *Simple Additive Weighting* Dalam Sistem Informasi Lowongan Kerja Berbasis Web untuk Rekomendasi Pencari Kerja Terbaik, langkah-langkah metode SAW :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (C), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R dengan menggunakan rumus persamaan 1.
4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik dengan menggunakan rumus persamaan 2.

2.4 Model *Waterfall*

Pembuatan sistem informasi pada penelitian ini menggunakan model *waterfall*. Metode ini merupakan metode yang tepat untuk membangun sebuah perangkat lunak yang tidak terlalu besar dan sumber daya manusia yang terlibat dalam jumlah yang terbatas. Menurut Sommerville (2011), tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall model*, yaitu *requirement analysis and definition, system and software design, implementation and unit testing, integration and system testing*, dan *operation and maintenance*. Alur *life cycle* pengembangan perangkat lunak pada model *waterfall*

diselesaikan pertahapan dan berurutan. Alur dari model *waterfall* dapat dilihat pada Gambar 2.1 berikut :



Gambar 2.1 Alur Model Waterfall (Pressman, 2009)

Keterangan dari skema di atas adalah :

1. *Requirements Definition* (Analisis Kebutuhan)

Tahap ini menganalisis kebutuhan yang akan digunakan dalam pembuatan aplikasi. Meliputi pengumpulan data kebutuhan fungsional dan non-fungsional dari aplikasi yang akan kita bangun.

2. *System and Software Design*

Jika proses analisis kebutuhan telah diketahui maka proses selanjutnya adalah pada tahapan desain sistem. Proses pendesainan sistem dari aplikasi yang akan kita bangun yaitu dengan menggunakan *Unified Modeling Language* (UML). Penggunaan UML karena sudah menggunakan konsep *Object Oriented Design* yang tentunya akan sangat memudahkan *developer* untuk membangun sebuah sistem. UML diagram yang akan dibuat antara lain :

a. *Business Process*

Business Proses digunakan untuk menggambarkan inputan data yang dibutuhkan sistem, output dari sistem serta tujuan dari pembuatan sistem.

b. *Use case Diagram*

Use case merupakan model yang menggambarkan fungsi atau tugas yang dilakukan oleh *user*, baik manusia maupun mesin/komputer. *Use case* model ini dapat digunakan untuk menggambarkan *job spesifcation* dan *job description*, serta keterkaitan antar *job*.

c. *Scenario*

Scenario diagram digunakan untuk menjelaskan atau menceritakan fitur atau isi yang ada di *use case* diagram. *Scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu event tertentu.

d. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* mempunyai fungsi yang sama dengan skenario namun diimplementasikan dalam diagram alir.

e. *Sequence Diagram*

Sequence diagram digunakan untuk menunjukkan rangkaian pesan yang dikirim antar *object* juga interaksi antar *object*.

f. *Class Diagram*

Class diagram digunakan untuk menggambarkan struktur statis class dalam sistem. *Class Diagram* dibuat untuk memudahkan dalam proses penulisan kode program.

g. *Entity Relationship Diagram (ERD)*

Merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

3. *Implementation and Unit Testing*

Tahap ini mengimplementasikan desain yang telah dibuat menjadi sebuah aplikasi berbasis web. Beberapa hal yang dilakukan dalam tahap implementasi adalah menulis kode program (*coding*) menggunakan bahasa pemrograman *Hypertext Markup Language (HTML)* dan *Hypertext Preprocessor (PHP)*. *HTML* dan *PHP* merupakan bahasa pemrograman utama dalam pembangunan aplikasi berbasis web. Manajemen basis data yang digunakan dalam pembangunan aplikasi adalah *DBMS MySQL*.

4. *Integration and System Testing*

Tahap pengujian yaitu uji coba terhadap sistem yang dibuat. Pengujian wajib dilakukan untuk menguji apakah sistem ini sudah sesuai dengan kebutuhan dari user atau belum. Dan apakah masih ada kesalahan maupun kelemahan terhadap sistem yang kami bangun tersebut. Diharapkan proses pengujian/testing dapat menyempurnakan sistem yang kami buat. Pengujian perangkat lunak menggunakan dua metode yakni :

1. *White Box Testing*

Merupakan cara pengujian dengan melihat modul untuk yang telah dibuat dengan program – program yang ada. Dan menganalisa apakah terjadi kesalahan atau tidak pada penulisan kode program. Pengujian ini dilakukan oleh pembuat program (*developer*). Menurut Pressman (2005), pengujian *white box* merupakan teknik pengujian jalur dasar yang digunakan untuk menentukan kompleksitas logis dengan menentukan rangkaian dasar jalur eksekusinya. Tahapan teknik pengujian jalur dasar meliputi dari mulai *listing program*, grafik alir, kompleksitas siklomatik, jalur program independen dan pengujian *basis set*.

a. *Listing Program*

Merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program. Contoh penerapan tahapan ini dapat dilihat pada Gambar 2.2.

```

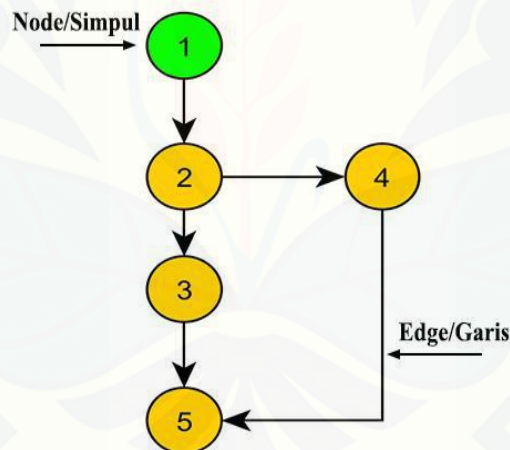
4 class Analisis extends CI_Controller {
5
6     function __construct()
7     {
8         parent::__construct();
9         $this->load->model('Survey_model');
10        $this->load->model('Survey_detail_model');
11        $this->load->model('Analisis_model');
12        if (!$this->ion_auth->logged_in() || !$this->ion_auth->is_admin())
13        {
14            redirect('auth/login', 'refresh');
15        }
16    }
17 }

```

Gambar 2.2 Contoh Listing Program

b. Grafik Alir

Menurut Pressman (2005), grafik alir merupakan sebuah notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari *listing program*. Grafik alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program. Contoh penggambaran diagram alir dapat dilihat pada Gambar 2.3.



Gambar 2.3 Contoh Diagram Alir (Pressman, 2005)

c. Kompleksitas Siklomatik

Kompleksitas Siklomatik merupakan matrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis per

suatu program (Pressman, 2005). Rumus yang digunakan untuk menghitung kompleksitas siklomatik yaitu:

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

Keterangan :

$V(G)$: Kompleksitas Siklomatik

E : Jumlah *Edge*

N : Jumlah *Node*

Berdasarkan grafik alir yang ada pada tahapan kedua diketahui jumlah *edge* adalah 5 dan jumlah *node* adalah 5, sehingga dapat dihitung kompleksitas siklomatik $V(G) = E - N + 2 = 5 - 5 + 2 = 2$. Jadi jumlah jalur independen adalah 2 jalur.

d. Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi (Pressman, 2005).

e. Pengujian *Basis Set*

Pada bagian ini diberikan contoh data yang akan memaksa pelaksanaan jalur di *basis set*. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati *basis set* yang tersedia. Sistem telah memenuhi syarat kelayakan *software* jika salah satu jalur yang dieksekusi setidaknya satu kali. Dari tahap sebelumnya telah diketahui 2 *basis set* Jika kemudian diuji dengan memasukkan data panjang = 5 dan lebar 3, maka basis set jalur yang digunakan adalah 1-2-4-5. Dapat dilihat bahwa jalur telah dieksekusi satu kali. Berdasarkan ketentuan tersebut dari segi kelayakan *software*, sistem ini telah memenuhi syarat.

2. *Black Box Testing*

Black Box Testing adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal atau kerja.

Pengetahuan khusus dari kode aplikasi atau struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Metode ini memfokuskan pada keperluan fungsionalitas dari *software* (Wildan Agissa, 2013). Pada pengujian *black box* ini, aplikasi yang dibangun pada penelitian ini akan diuji dengan mengujikan langsung *running* aplikasi dan melakukan kegiatan pengujian dengan menganalisis proses *input* dan *output* yang dihasilkan aplikasi. Dalam metode *black box* juga dilakukan pengujian dengan cara memasukkan data normal dan data salah, dari penginputan ini nantinya akan dilakukan analisis terhadap reaksi yang muncul pada sistem.

3. *Operation and Maintenance*

Tahap ini merupakan tahap akhir dari perancangan model *waterfall*. Penerapan program ini dilakukan setelah program yang dibuat oleh penulis selesai dan telah melalui tahap pengujian terlebih dahulu. Selanjutnya dilakukan proses *maintenance* atau perawatan mengatasi masalah *bugs* yang muncul setelah aplikasi diserahkan pada *user*. Apabila terdapat *bugs*, *user* langsung dapat mengkonfirmasi kepada *developer* untuk segera ditangani oleh *developer*.

BAB 3. METODOLOGI PENELITIAN

Pada bab ini dijelaskan tentang metode-metode yang digunakan selama penelitian dilakukan, seperti jenis penelitian, studi literatur, data dan sumber data penelitian, serta tahapan analisis hingga model perancangan sistem.

3.1 Jenis Penelitian

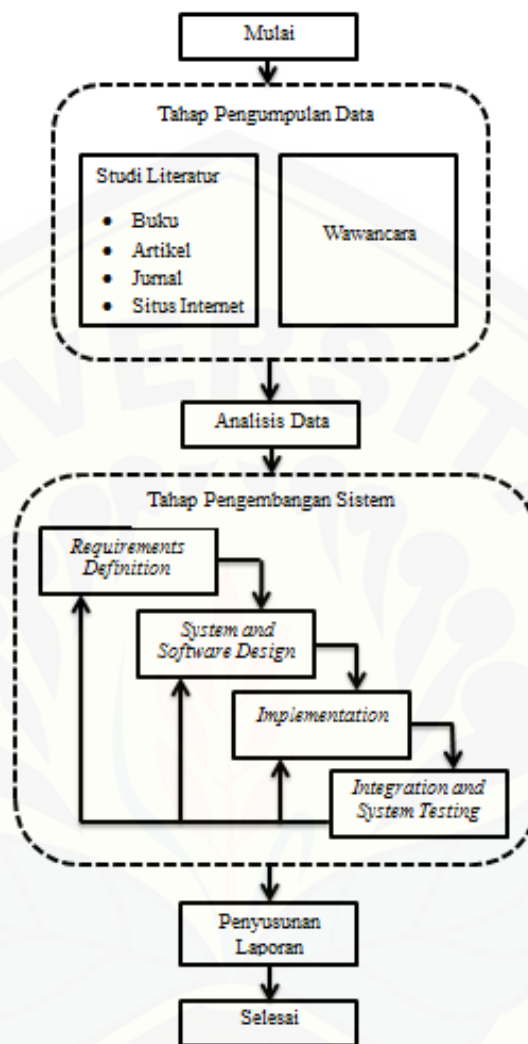
Penelitian yang dilakukan merupakan penelitian pengembangan (*Development Research*). Penelitian pengembangan ini dilakukan untuk membangun sebuah sistem informasi pendistribusian air bersih. Dengan mengadopsi metode *waterfall*, ditambah pengumpulan dan analisis data dalam bentuk angka serta *flow of document* (FOD). Kemudian melakukan wawancara atau observasi kepada pihak penyeleksi atau pihak berwenang yaitu Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Bondowoso, sebagai aktor pengambil keputusan, untuk menentukan lokasi prioritas pendistribusian air bersih pada daerah yang krisis air. Tahapan selanjutnya adalah mulai membangun sistem guna memperoleh sebuah sistem yang baik, dari segi efektifitas dan efisiensi.

3.2 Tempat dan Waktu Penelitian

Tempat dan waktu yang dilakukan dalam penelitian ini yaitu, Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Bondowoso dan wilayah yang mengalami krisis air, waktu yang dilakukan selama 2 (dua) bulan, penelitian ini di mulai pada bulan Oktober 2017 sampai dengan bulan November 2017. Kemudian penelitian juga dilakukan di Laboratorium Program Studi Sistem Informasi (PSSI) yang akan dilanjutkan dengan tahap pembangunan sistem tersebut.

3.3 Tahapan Penelitian

Dalam tahap penelitian dilakukan dengan beberapa tahapan diantaranya pengumpulan data, dengan tahap analisis dan tahap perancangan sistem. Tahap ini dilakukan dengan diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian (Sumber : Hasil Analisis, 2016)

3.3.1 Teknik Pengumpulan Data

Proses untuk mendapatkan data yang dibutuhkan untuk membangun sistem pendukung keputusan penentuan prioritas pendistribusian air bersih menggunakan teknik pengumpulan data sebagai berikut:

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan data dan informasi yang diperlukan untuk proses perancangan sistem. Data dan informasi dapat diperoleh dari lokasi penelitian

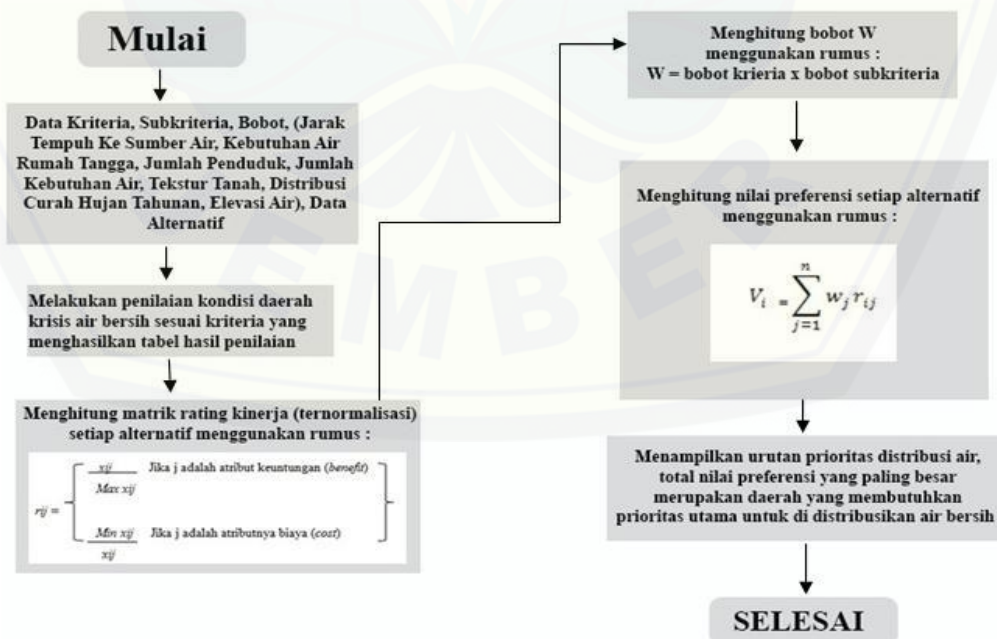
yaitu di Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Bondowoso dan wilayah yang mengalami krisis air. Selain itu, studi literatur juga dapat diperoleh dari *paper*, jurnal ilmiah, serta buku-buku referensi yang berkaitan dengan penelitian.

2. Wawancara

Metode *interview* atau wawancara yaitu mencari data secara langsung dari pihak yang bersangkutan atau *interview* secara langsung. Dari hasil wawancara di daerah krisis air ada beberapa faktor yang menyebabkan krisis air bersih mulai dari ketersediaan air bersih, jarak ke sumber air, kebutuhan air sehari-hari serta curah hujan. Penulis dalam penelitian ini melakukan wawancara secara langsung kepada pihak Badan Penanggulangan Bencana Daerah (BPBD) Kabupaten Bondowoso beserta para warga yang berada di wilayah krisis air.

3.3.2 Tahap Analisis Data

Tahap analisis dilakukan setelah melakukan pengumpulan data yang diperoleh dari studi literatur dan wawancara mengenai penentuan prioritas pendistribusian air bersih. Data yang diperoleh akan dianalisa dengan metode *Simple Additive Weighting* (SAW). Alur dari proses metode SAW dapat dilihat pada Gambar 3.2



Gambar 3.2 Flowchart Metode SAW pada sistem

Tahap pertama alur metode yaitu menentukan kriteria, subkriteria, bobot. Kemudian dilanjutkan dengan melakukan penilaian kondisi daerah krisis air bersih sesuai dengan kriteria, penilaian kondisi daerah krisis air bersih dilakukan oleh surveyor. Dilanjutkan dengan menghitung matriks rating kinerja tiap alternatif. Setelah mendapatkan hasil dari penghitungan rating kinerja dilanjutkan dengan menghitung bobot W , dengan menghitung nilai preferensi. Nilai preferensi yang paling paling besar merupakan daerah yang membutuhkan prioritas untuk didistribusikan pertama.

3.3.3 Tahap Pengembangan Sistem

Setelah tahap pengumpulan data selesai, selanjutnya data akan dianalisis menggunakan metode *Simple Additive Weighting* (SAW). Kemudian akan dilanjutkan tahap pengembangan sistem dengan menggunakan konsep berbasis objek dengan pemodelan *Unified Modelling Language* (UML). Pemodelan UML yang digunakan pada penelitian ini antara lain, *Business Process*, *Use case Diagram*, *Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class Diagram* dan *Entity Relationship Diagram* (ERD).

Setelah tahap perancangan selesai, dilanjutkan dengan tahap implementasi menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) dengan bantuan *framework Code Igniter* (CI) dan manajemen basisdata menggunakan *DBMS MySQL*. Hasil perancangan dan implementasi kemudian akan ditesting menggunakan *White Box* dan *Black Box*. Ketika ada perbaikan dalam sistem maka proses dari tahapan pengembangan sistem harus diulang sesuai dengan model pengembangan sistem pada penelitian ini yaitu model *waterfall*.

BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan dan perancangan hingga tahap penulisan kode program (*coding*) dan pengujian aplikasi yang digunakan dalam proses pengembangan atau pembangunan sistem pendukung keputusan penentuan prioritas pendistribusian air bersih menggunakan metode *Simple Additive Weighting* (SAW).

4.1 Analisis Kebutuhan Sistem

Berdasarkan metode pengembangan sistem model *waterfall*, tahapan awal yang dilakukan adalah tahapan analisis. Tahapan analisis ini dilakukan terhadap objek penelitian untuk memperoleh kebutuhan-kebutuhan dari sistem yang dibangun, baik berupa kebutuhan fungsional maupun kebutuhan nonfungsional. Dimana hasil analisa tersebut sangat mempengaruhi fungsionalitas sistem yang dibangun untuk dapat digunakan sesuai dengan fungsi dan kebutuhan pengguna. Kebutuhan fungsional dan nonfungsional diperoleh dari pengumpulan data berupa wawancara terhadap pihak terkait yaitu BPBD Kabupaten Bondowoso dan beberapa studi literatur dengan tujuan mendapatkan semua informasi untuk membangun sistem pendukung keputusan penentuan prioritas pendistribusian air bersih menggunakan metode *Simple Additive Weighting* (SAW).

4.1.1 Kebutuhan Fungsional

Pengumpulan Kebutuhan fungsional merupakan hal pokok yang harus dapat dilakukan oleh sistem dalam menerima masukan untuk diproses sehingga menghasilkan keluaran. Kebutuhan fungsional dari Sistem Pendukung Keputusan Penentuan Prioritas Pendistribusian Air Bersih antara lain :

1. Sistem dapat mengelola data *user* untuk pemberian hak akses.
2. Sistem dapat mengelola data daerah krisis sebagai data alternatif untuk mencari alternatif daerah prioritas.

3. Sistem dapat mengelola dan menampilkan hasil analisis prioritas daerah krisis air berupa tabel penilaian alternatif, hasil normalisasi matriks dan urutan prioritas.
4. Sistem dapat mengelola, menyimpan dan memanajemen data area daerah krisis.
5. Sistem mampu menampilkan hasil analisis prioritas pendistribusian air menggunakan metode *Simple Additive Weighting (SAW)*.
6. Sistem *login* dengan *username* dan *password*.

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktivitas sistem sesuai dengan kebutuhan fungsional yang telah disusun. Kebutuhan non-fungsional dari Sistem Pendukung Keputusan Penentuan Prioritas Pendistribusian Air Bersih antara lain :

1. Sistem berbasis *website*.
2. Sistem memiliki batasan hak akses pengguna dengan menggunakan *username* dan *password*.
3. Sistem memiliki tampilan yang menarik dan mudah dimengerti bagi *user*.
4. Sistem mudah diakses pada beberapa sistem operasi karena sistem berbasis web.

4.2 Deskripsi Umum Sistem

Deskripsi umum dari sistem informasi implementasi SAW untuk sistem pendukung keputusan penentuan prioritas pendistribusian air bersih yang dibangun dalam penelitian ini akan dijelaskan lebih detail pada SOP (*Statement of Purpose*) sistem dan fungsi sistem.

4.2.1 SOP (*Statement of Purpose*) Sistem

Pada sistem perangkaan daerah krisis ini memerlukan beberapa data, antara lain nilai kriteria dan nilai sub kriteria. Lalu sistem ini dapat menghitung nilai bobot dari setiap kriteria, kemudian dilanjutkan dengan proses perangkaan yang akan menghitung nilai alternatif yang telah dipilih. Selanjutnya sistem ini akan menampilkan alternatif terbaik berupa nilai yang paling besar.

Sistem dapat mengelola data kriteria, data sub kriteria serta data dari surveyor. Sistem ini dapat melakukan prioritas pengambilan keputusan yaitu pemberian nilai alternatif terbaik menggunakan metode SAW sehingga dapat mempermudah dalam pencarian informasi dan menentukan pemilihan daerah krisis yang paling utama untuk didahulukan mendapat bantuan air bersih.

4.2.2 Fungsi Sistem

Fungsi utama dari sistem yang dikembangkan ini terletak pada fitur *login* yang dapat menentukan hak akses dari setiap pengguna dari sistem ini sendiri. Selanjutnya sistem akan menyajikan tampilan sistem yang sesuai dengan hak akses dari pengguna yang meliputi:

a. Admin

Administrator merupakan *user* yang terdaftar dan dapat melakukan *login* ke dalam sistem sebagai admin. Admin memiliki hak akses penuh terhadap manajemen sistem seperti menambahkan, merubah, menghapus, memperbaharui input data kriteria serta melakukan konfirmasi dari inputan user. Admin bertanggung jawab menentukan proses survei telah selesai atau masih dalam proses.

b. Surveyor

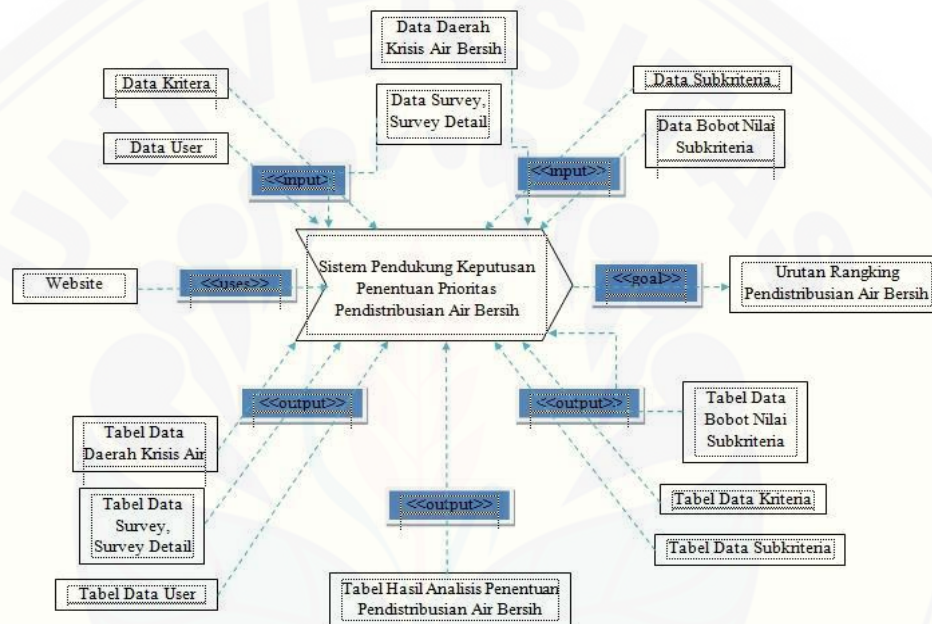
Surveyor merupakan *user* yang terdaftar dan dapat melakukan *login* kedalam sistem. Selanjutnya surveyor akan masuk kedalam halaman *web* sesuai hak akses yang diberikan. Surveyor dapat melakukan tambah data survei yang selanjutnya akan masuk *queue* yang nantinya akan dikonfirmasi oleh admin.

4.3 Desain Sistem

Desain sistem adalah tahapan untuk memodelkan sistem informasi penunjang keputusan yang akan dibuat. Dokumen desain sistem yang akan dibuat meliputi *Business Process*, *Use Case Diagram*, *Use Case Skenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *Entity Relationship Diagrams (ERD)*.

4.3.1 Business Process

Business Process merupakan model atau diagram yang menggambarkan sebuah proses lengkap dengan *resources* dan *information* yang dibutuhkan, event yang mendorong terjadinya proses dan *goal* yang dituju. *Business Process* digunakan untuk menggambarkan inputan data yang dibutuhkan sistem, *output* dari sistem serta tujuan dari pembuatan sistem. *Business Process* sistem dapat dilihat pada Gambar 4.1.

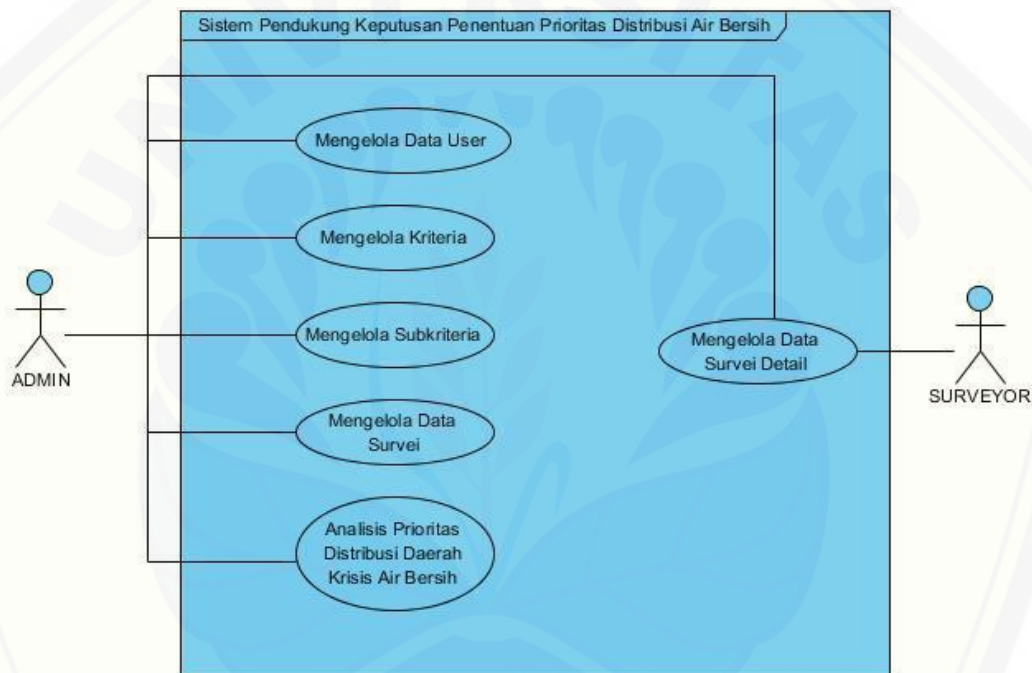


Gambar 4.1 *Business Process* Sistem

Berdasarkan gambar 4.1 sistem membutuhkan enam inputan data dan menghasilkan tujuh *output*. Data yang diperlukan sebagai inputan yaitu data *user*, data kriteria, data daerah krisis air bersih, data survei dan survei detail, data subkriteria, dan data bobot nilai subkriteria. Sedangkan data *output* sebagai hasil dari pengolahan sistem yaitu tabel data daerah krisis air, tabel data survei dan survei detail, tabel data *user*, tabel data bobot nilai subkriteria, tabel data kriteria, tabel data subkriteria dan tabel hasil analisis penentuan pendistribusian air bersih. Tujuan dari sistem ini adalah urutan rangking prioritas pendistribusian air bersih.

4.3.2 Use Case Diagram

Use Case Diagram adalah rangkaian/uraian kelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use Case Diagram* berfungsi untuk menggambarkan fitur apa saja yang akan dijalankan pada *software* yang akan dibuat. *Use Case Diagram* sistem pendukung keputusan penentuan prioritas penentuan pendistribusian air bersih dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Use Case Diagram*

Definisi *use case* merupakan penjelasan dari setiap *use case* yang merupakan fitur fitur dari sistem. Penjelasan dari definisi *use case* sistem pendukung keputusan penentuan prioritas pendistribusian air bersih dapat dilihat pada Tabel 4.1.

Tabel 4.1 Penjelasan *Use Case*

No	Use Case	Deskripsi
1	Mengelola Data User	
	Tambah <i>User</i>	<i>Use Case</i> yang menggambarkan proses menambahkan <i>user</i> ke sistem.

No	Use Case	Deskripsi
	<i>Update User</i>	<i>Use Case</i> yang menggambarkan proses mengubah <i>user</i> yang disimpan dalam sistem.
	<i>Detail User</i>	<i>Use Case</i> yang menggambarkan proses melihat <i>user</i> dalam sistem.
	<i>Delete User</i>	<i>Use Case</i> yang menggambarkan proses menghapus data <i>user</i> dari sistem.
2	Mengelola Data Kriteria	
	Tambah data kriteria	<i>Use Case</i> yang menggambarkan proses menambahkan data kriteria kedalam sistem oleh admin
	<i>Update</i> data kriteria	<i>Use Case</i> yang menggambarkan proses mengubah data kriteria yang disimpan dalam sistem.
	<i>Detail</i> data kriteria	<i>Use Case</i> yang menggambarkan proses melihat data kriteria dalam sistem.
	<i>Delete</i> data kriteria	<i>Use Case</i> yang menggambarkan proses menghapus data kriteria dari sistem.
3	Mengelola Data Subkriteria	
	Tambah data subkriteria	<i>Use Case</i> yang menggambarkan proses menambahkan data subkriteria kedalam sistem oleh admin.
	<i>Detail</i> data subkriteria	<i>Use Case</i> yang menggambarkan proses melihat data subkriteria dalam sistem.
	<i>Update</i> data subkriteria	<i>Use Case</i> yang menggambarkan proses mengubah data subkriteria yang disimpan dalam sistem.
	<i>Delete</i> data subkriteria	<i>Use Case</i> yang menggambarkan proses menghapus data subkriteria dari sistem.

No	Use Case	Deskripsi
4	Mengelola Data Survei	
	Tambah data survei	<i>Use Case</i> yang menggambarkan proses menambahkan data survei kedalam sistem oleh admin.
	<i>Proses</i> data survei	<i>Use Case</i> yang menggambarkan penentuan status survei masih dalam proses atau telah selesai yang disimpan dalam sistem.
	<i>Update</i> data survei	<i>Use Case</i> yang menggambarkan proses mengubah data survei yang disimpan dalam sistem.
	<i>Delete</i> data survei	<i>Use Case</i> yang menggambarkan proses menghapus data survei dari sistem oleh admin.
5	Analisis Prioritas Distribusi Air Bersih	
	Analisis prioritas distribusi air bersih	<i>Use Case</i> yang menggambarkan proses penghitungan nilai masing-masing kondisi daerah krisis air. Kemudian ditampilkan sesuai dengan prioritas.
6	Mengelola Data Survei Detail	
	Tambah data survei detail admin	<i>Use Case</i> yang menggambarkan proses menambahkan data survei detail kedalam sistem oleh admin.
	Hapus data survei detail admin	<i>Use Case</i> yang menggambarkan proses menghapus data survei detail dari sistem oleh admin.
	Tambah data survei detail surveyor	<i>Use Case</i> yang menggambarkan proses menambahkan data survei detail kedalam sistem oleh surveyor.

4.3.3 Skenario Diagram

Skenario diagram digunakan untuk menjelaskan fitur yang ada di *use case* diagram. Skenario menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu *event* tertentu. *Skenario* pada sistem prioritas pendistribusian air bersih untuk menambah data survei dapat dilihat pada Tabel 4.2.

Tabel 4.2 Skenario Menambah Data Survei

Nomor <i>Use Case</i>	UC-06
Nama	Mengelola Data Survei Detail
Aktor	Surveyor
<i>Pre Condition</i>	Surveyor harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Surveyor berhasil mengelola data survei detail.
SKENARIO NORMAL MENGELOLA DATA SURVEI DETAIL	
TAMBAH DATA SURVEI	
Aktor	Sistem
1. Klik menu Survei.	
	2. Menampilkan halaman data survei tersedia, tombol Mulai Survei.
3. Klik tombol <i>Mulai Survei</i> .	
	4. Menampilkan halaman <i>Form Data Survei</i> yang berisi form dengan kolom Pilih Desa, Lokasi Kekeringan, Jarak Tempuh ke Sumber Air, Distribusi Cyrah Huja Tahunan, Elevasi Air, Tekstur Tanah, Jumlah Penduduk, Jumlah Kebuahan Air, tombol Tambah,

	tombol Cancel.
5. Mengisi <i>Form Data Survei</i> .	
6. Klik tombol <i>Tambah</i> .	
	7. Menyimpan data baru ke database 8. Menampilkan halaman data survei tersedia, tombol Mulai Survei.
SKENARIO ALTERNATIF MENGELOLA DATA SURVEI DETAIL	
DATA TIDAK LENGKAP	
Aktor	Sistem
6a. Klik tombol <i>Tambah</i> .	
	7a. Menampilkan <i>warning</i> “ <i>This field is required</i> ”.
SKENARIO ALTERNATIF MENGELOLA DATA SURVEI DETAIL	
TOMBOL CANCEL PADA DATA SURVEI	
Aktor	Sistem
6b. Klik tombol <i>Cancel</i> .	
	7b. Menampilkan halaman data survei detail <i>list</i> , tombol Mulai Survei.

Skenario pada sistem prioritas pendistribusian air bersih analisis prioritas pendistribusian dapat dilihat pada Tabel 4.3.

Tabel 4.3 Skenario Menganalisis Prioritas Distribusi Air Bersih

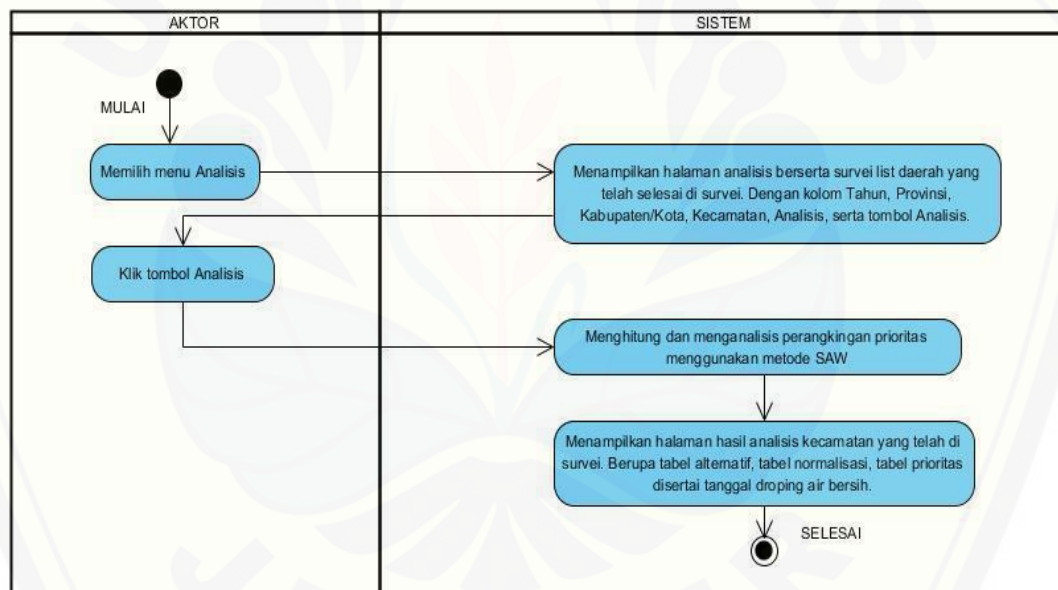
Nomor <i>Use Case</i>	UC-05
Nama	Analisis Prioritas Distribusi Air Bersih
Aktor	Admin

<i>Pre Condition</i>	Admin harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Admin berhasil menganalisis prioritas.
SKENARIO NORMAL ANALISIS PRIORITAS DISTRIBUSI	
ANALISIS HASIL SURVEI	
Aktor	Sistem
1. Memilih menu <i>Analisis</i> .	
	2. Menampilkan halaman analisis beserta survei <i>list</i> daerah yang telah selesai di survei. Dengan kolom Tahun, Provinsi, Kabupaten/Kota, Kecamatan, serta tombol Analisis.
3. Klik tombol <i>Analisis</i> .	
	4. Menghitung dan menganalisis perangkingan prioritas menggunakan metode <i>Simple Additive Weighting</i> (SAW)
	5. Menampilkan halaman hasil analisis kecamatan yang telah di survei. Berupa tabel alternatif, tabel normalisasi, tabel prioritas disertai tanggal droping air bersih.

4.3.4 Activity Diagram

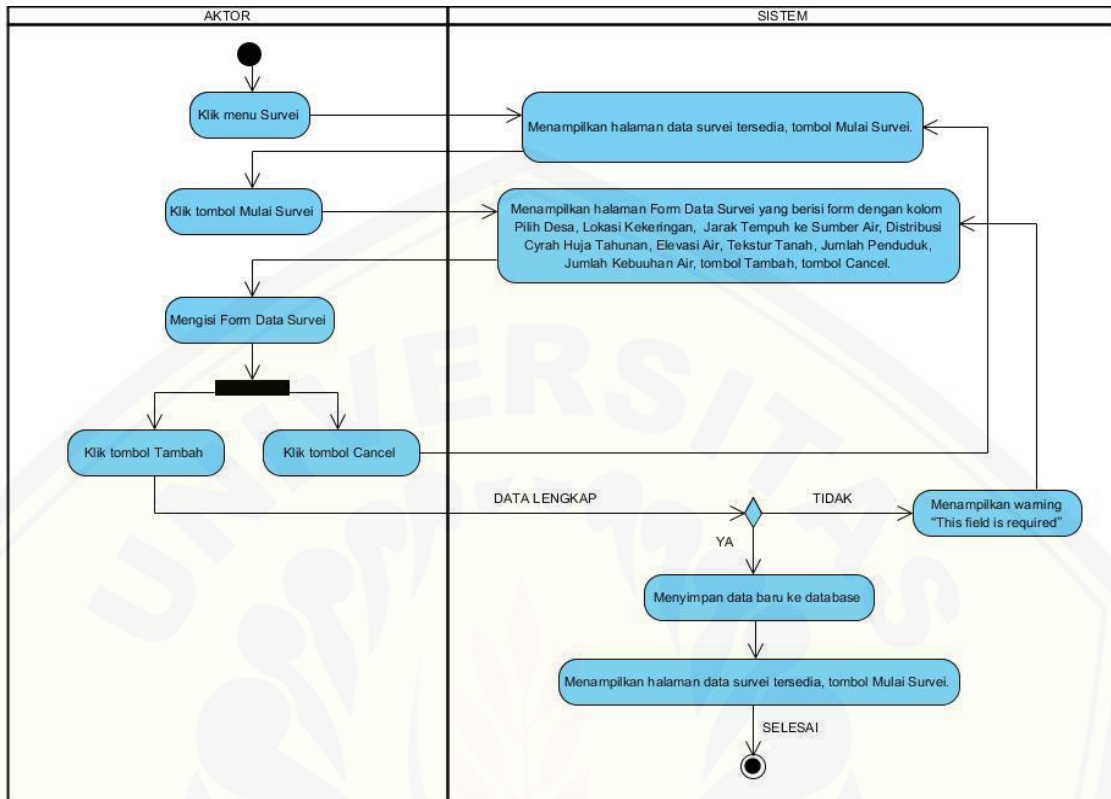
Activity diagram menggambarkan aktivitas aktor dan sistem yang saling berhubungan dalam suatu aktivitas atau *event*. *Activity Diagram* menggambarkan berbagai alur aktivitas dalam sistem yang dirancang berawal *decision*. *Activity diagram* harus sesuai dengan skenario sistem yang telah dirancang. Sistem memberikan respon pada aktivitas yang dilakukan aktor.

Activity diagram menganalisis prioritas distribusi air bersih menggambarkan alur aktivitas pada fitur analisis prioritas. Fitur analisis prioritas distribusi air bersih memiliki aktivitas utama, yaitu menganalisis prioritas distribusi. Detail urutan aktivitas dalam menganalisis prioritas distribusi ditunjukkan pada Gambar 4.3.



Gambar 4.3 Activity Diagram Menganalisis Prioritas Distribusi Air Bersih

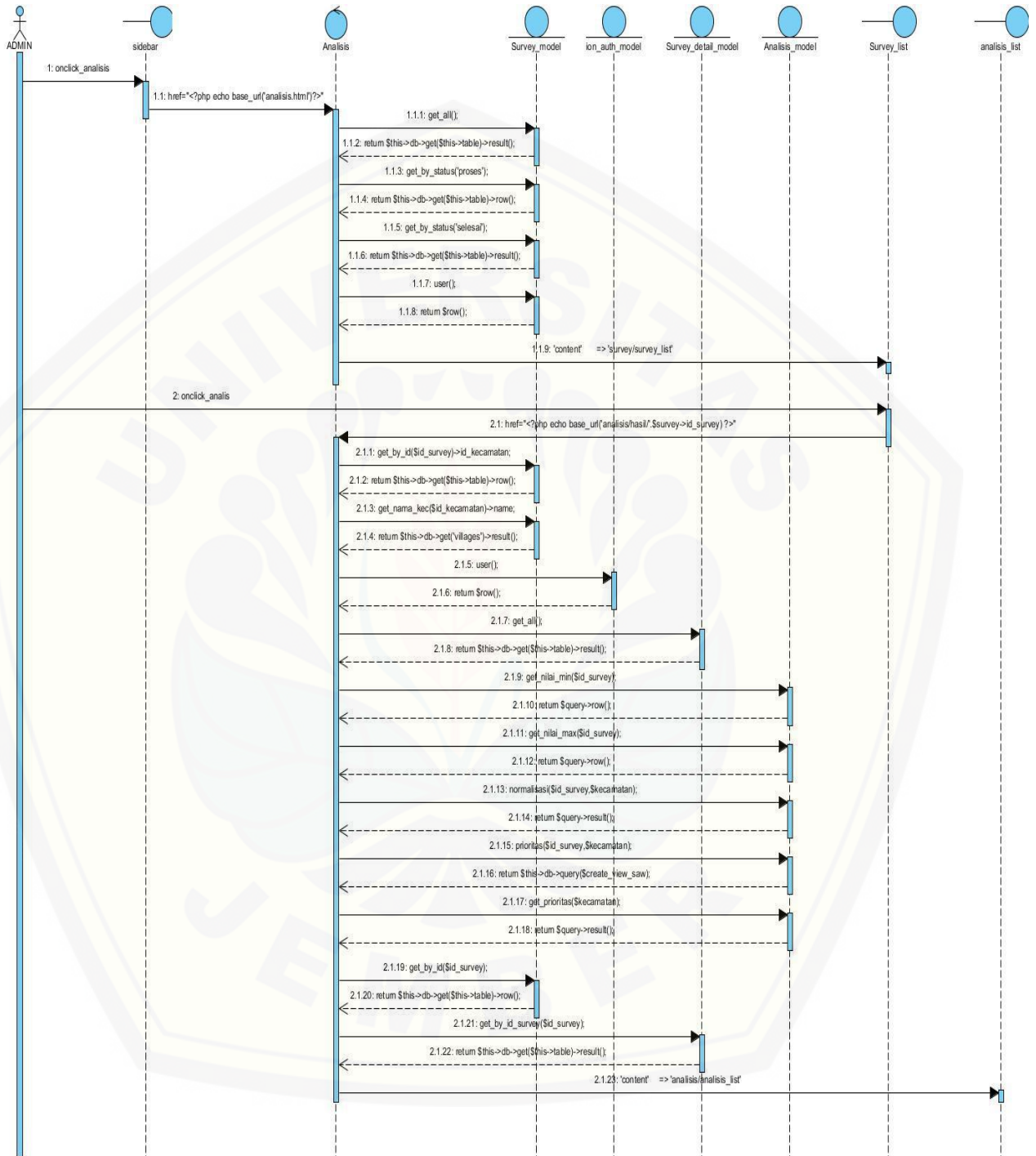
Activity diagram mengelola data survei detail untuk surveyor menggambarkan alur aktivitas pada fitur survei untuk surveyor. Fitur ini memiliki aktivitas utama yaitu menginputkan data dari daerah yang telah di survei. Yang nanti hasil survei akan di proses oleh admin. Detail urutan aktifitas dalam mengelola data survei dapat dilihat pada Gambar 4.4.



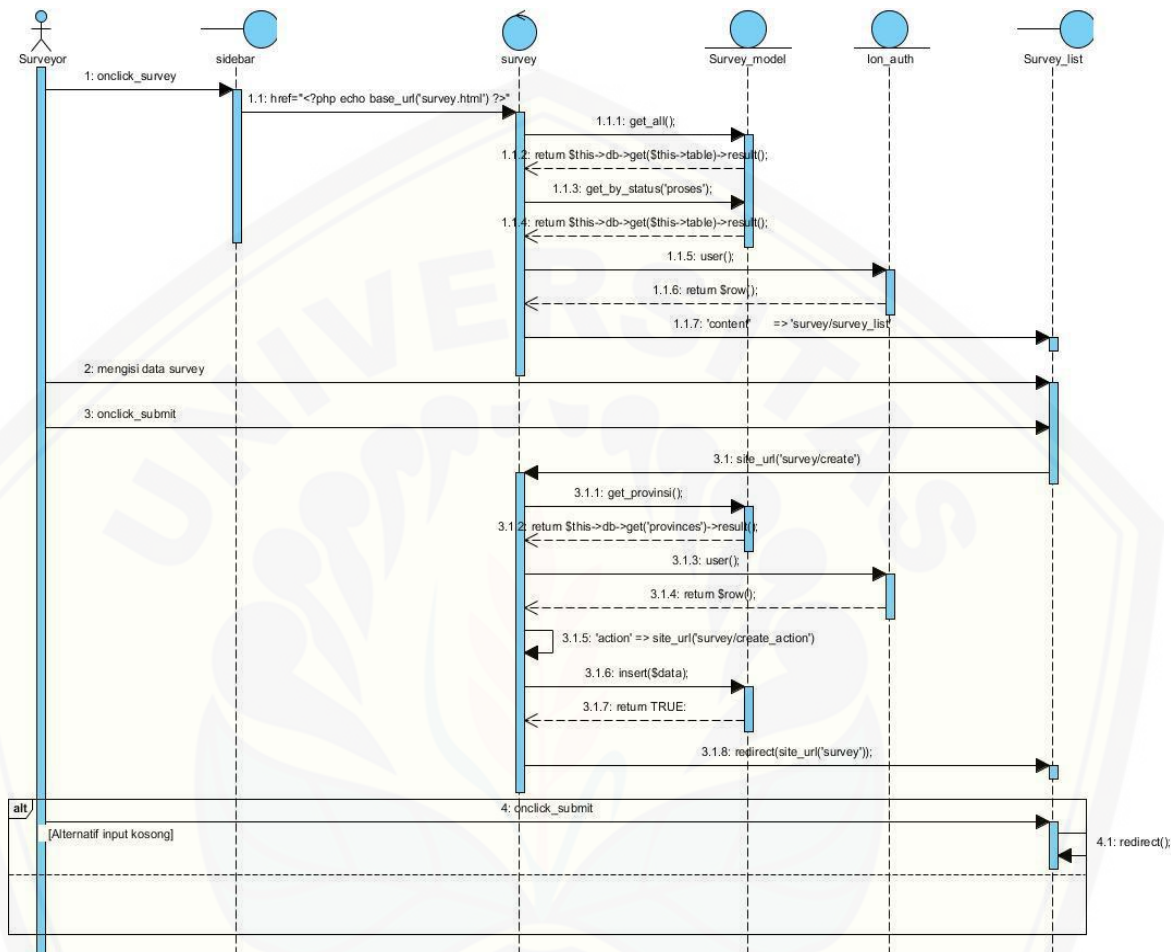
Gambar 4.4 Activity Diagram Mengelola Data Survei Detail Surveyor

4.3.5 Sequence Diagram

Sequence diagram pada sistem pendukung keputusan penentuan prioritas pendistribusian air bersih ini digunakan untuk menggambarkan interaksi dari masing-masing komponen pada satu fungsi. Interaksi tersebut dilakukan oleh *user* pada sistem. Di dalam sistem sendiri juga terdapat interaksi yaitu antara *view*, *controller* dan *model*. Pada proses perancangan ini setiap fitur akan digambarkan ke dalam *sequence diagram*. Detail *sequence diagram* analisis penentuan prioritas pendistribusian air bersih dan *sequence diagram* menginput data survei oleh surveyor ditunjukkan pada Gambar 4.5 dan Gambar 4.6.



Gambar 4.5 Sequence Diagram Menganalisis Prioritas Distribusi Air Bersih



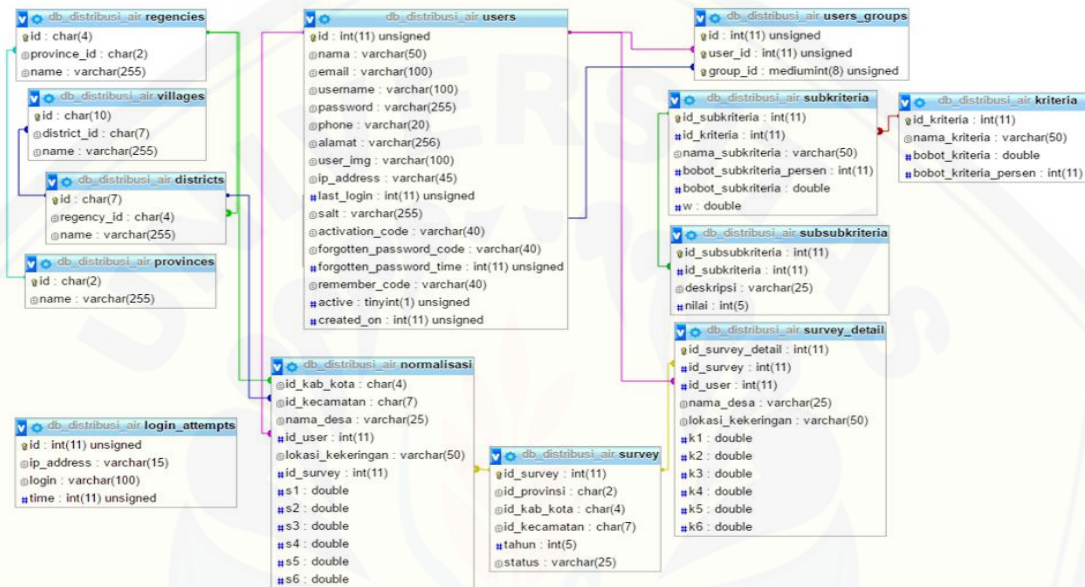
Gambar 4.6 Sequence Diagram Menginput Data Survei Detail oleh Surveyor

4.3.6 Class Diagram

Class diagram menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Selain itu *class diagram* juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dirancang sehingga bagaimana caranya setiap *class* saling berkolaborasi untuk mencapai sebuah tujuan. *Class diagram* sistem dapat dilihat pada Gambar 4.7.

4.3.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) pada aplikasi pendukung keputusan penentuan prioritas pendistribusian air bersih pada daerah krisis, menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD aplikasi ditunjukkan pada Gambar 4.8.



Gambar 4.8 ERD Sistem

Pada gambar 4.8 menjelaskan hubungan antar tabel dalam *database* sistem. Dalam ERD sistem menggunakan tiga belas tabel yang semuanya memiliki relasi. Relasi yang digunakan adalah relasi *one to many* atau relasi satu ke banyak dan relasi *many to many*. Objek-objek dalam *database* sistem telah melalui proses normalisasi.

4.4 Penulisan Kode Program

Setelah tahap desain selesai, tahap selanjutnya dalam penelitian ini yaitu tahap pengimplementasian desain perancangan kedalam bahasa pemrograman. Desain sistem dari semua fitur sistem yang telah dibuat menggunakan pemodelan UML akan diimplementasikan kedalam kode program. Penulisan kode program menggunakan bahasa pemrograman *PHP*, *HTML*, dan *CSS* dengan *framework Code Igniter (CI)* dan menggunakan *database My-SQL*. Sistem pendukung keputusan penentuan prioritas

pendistribusian air bersih menggunakan metode *Simple Additive Weighting* (SAW) ini dibangun menggunakan *framework Code Igniter* (CI).

4.5 Pengujian Sistem

Pada penelitian ini penulis menggunakan dua metode pengujian sistem yaitu *Black Box Testing* dan *White Box Testing*. Berikut adalah hasil pengujian sistem :

4.5.1 Metode *White Box Testing*

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi kode program. Pengujian ini bertujuan sebagai evaluasi apakah sistem berhasil menjalankan fungsi-fungsi, inputan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem itu sendiri. Pengujian dengan metode *white box* dilakukan oleh penulis dengan cara menghitung *independent path* yaitu dengan menggunakan suatu pengukuran kuantitatif yang bertahap mulai dari *listing program*, pembuatan diagram alir, *cyclomatic complexity*. Tahapan-tahapan pengujian dengan metode *white box* ini akan diterapkan pada fitur yang dinilai dapat mewakili sistem ini yaitu fitur analisis prioritas distribusi air bersih. Tahapan pengujian jalur dasar meliputi:

a. *Listing Program*

Listing program merupakan baris-baris kode yang nantinya akan diuji. Setiap langkah dari kode-kode yang ada diberi nomor baik menjalankan *statement* biasa atau penggunaan kondisi dalam program. *Listing program* sistem SAW untuk analisis prioritas distribusi air bersih dapat dilihat pada Gambar 4.9.

```

40     public function hasil($id_survey)
41     {
42         $id_kecamatan = $this->Survey_model->get_by_id($id_survey)->id_kecamatan;
43         $kecamatan = $this->Survey_model->get_nama_kec($id_kecamatan)->name;
44         $user = $this->ion_auth->user()->row();
45
46         $survey_detail = $this->Survey_detail_model->get_all();
47
48         $nilai_min = $this->Analisis_model->get_nilai_min($id_survey);
49         $nilai_max = $this->Analisis_model->get_nilai_max($id_survey);
50
51         $normalisasi = $this->Analisis_model->normalisasi($id_survey,$id_kecamatan);
52
53         $this->Analisis_model->prioritas($id_survey,$id_kecamatan);
54         $hasil_analisis = $this->Analisis_model->get_prioritas($id_kecamatan);
55
56         $survey = $this->Survey_model->get_by_id($id_survey);
57         $survey_detail_by_id = $this->Survey_detail_model->get_by_id_survey($id_survey);
58         $begin = new DateTime( $survey->tgl_drop );
59         $interval = new DateInterval("P2D"); // 1 month
60         $occurrences = count($survey_detail_by_id);
61         $period = new DatePeriod($begin,$interval,$occurrences);
62         foreach($period as $dt){
63             $sarr_date[] = $dt->format("Y-m-d");
64         }
65         //print_r($sarr_date);

```

```

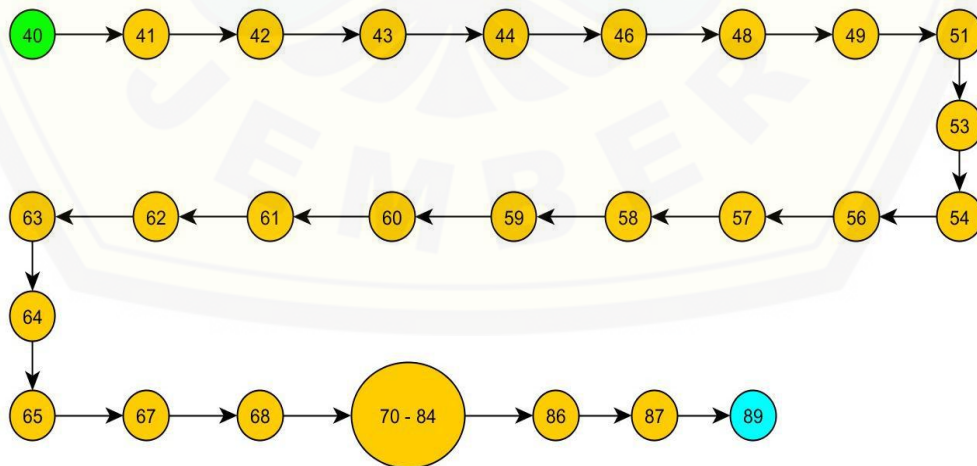
67     $this->breadcrumbs->push('Analisis', '/analisis');
68     $this->breadcrumbs->push('Hasil Kecamatan '.$kecamatan, '/survey_detail');
69
70     $data = array(
71         'title' => 'Hasil Analisis Kecamatan '.$kecamatan ,
72         'content' => 'analisis/analisis_list',
73         'breadcrumbs' => $this->breadcrumbs->show(),
74         'user' => $user ,
75
76         'survey_detail_data' => $survey_detail_by_id ,
77         'nilai_min' => $nilai_min,
78         'nilai_max' => $nilai_max,
79
80         'normalisasi' => $normalisasi,
81         'hasil_analisis' => $hasil_analisis,
82
83         'tgl_drop' => $arr_date ,
84     );
85
86     $this->load->view('layout/layout', $data);
87 }
88
89 }
    
```

Gambar 4.9 Listing Program Fitur Analisis Prioritas

Dari gambar 4.9 dapat dilihat fitur analisis prioritas distribusi air bersih yang dituliskan dengan bahasa pemrograman *PHP* dari baris nomor 40 sampai baris nomor 89. Kode pada baris ini digunakan untuk proses menghitung nilai alternatif sampai hasil prioritas daerah krisis air yang harus mendapat distribusi air bersih.

b. Diagram Alir

Diagram alir merupakan notasi sederhana yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari *listing program*. Diagram alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program. Diagram alir analisis prioritas distribusi air bersih dapat dilihat pada Gambar 4.10.



Gambar 4.10 Diagram Alir Fitur Analisis Prioritas

Pada gambar 4.10 dinotasikan alur proses fitur menganalisis prioritas air bersih. Notasi *node elips* untuk proses normal seperti nomor 42. Setiap notasi *node* memiliki *edge* (garis) masuk atau pun keluar yang sesuai dengan jalannya program.

c. Kompleksitas Siklomatik (*Cyclomatic Complexity*)

Kompleksitas siklomatik merupakan matrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jumlah jalur independen dalam basis set suatu program. Kompleksitas analisis prioritas distribusi air bersih berdasarkan diagram alir sebagai berikut :

$$CC = EDGE - NODE + 2$$

$$CC = 26 - 27 + 2$$

$$CC = 1$$

Fitur analisis prioritas distribusi air bersih memiliki nilai cc sebesar 1 karena pada fitur ini tidak terdapat percabangan. Sehingga nilai cc sebesar 1 mampu menjalankan fitur analisis prioritas distribusi air bersih dengan inputan *user* dan *output* yang sesuai dengan hasil perhitungan manual.

d. Jalur Program Independen

Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi. Jalur 1 = 40 - 41 - 51 - 53 - 54 - 63 - 64 - 65 - 86 - 87 - 89.

e. Pengujian *Basis Set*

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

lunak jika salah satu jalur yang dieksekusi setidaknya satu kali. Pengujian basis set analisis prioritas dapat dilihat pada Tabel 4.4.

Tabel 4.4 *Test Case* Analisis Prioritas

Jalur 1	
<i>Test Case</i>	Jika data bobot kriteria dan subkriteria ada di database, ambil semua data bobot kriteria dan subkriteria sesuai data inputan kriteria.
Target yang diharapkan	Menampilkan hasil analisis prioritas
Hasil Pengujian	Benar
<i>Path/Jalur</i>	40 - 41 - 51 - 53 - 54 - 63 - 64 - 65 - 86 - 87 - 89

4.5.2 Metode *Black Box*

Pengujian *black box* berfungsi untuk menguji sistem pendukung keputusan penentuan prioritas pendistribusian air bersih dari segi spesifikasi fungsional sistem, dengan tujuan mengetahui apakah fungsi-fungsi, inputan, dan keluaran sistem sesuai dengan spesifikasi yang dibutuhkan oleh pengguna.

BAB 6. PENUTUP

Pada bab ini merupakan bagian akhir didalam penulisan skripsi, berisi tentang kesimpulan dan saran. Kesimpulan yang ditulis merupakan kesimpulan dari hasil penelitian yang telah dilakukan dan saran diharapkan dapat digunakan sebagai acuan untuk penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Sistem pendukung keputusan penentuan prioritas pendistribusian air bersih pada daerah krisis air di wilayah Kabupaten Bondowoso menggunakan metode *Simple Additive Weighting* (SAW) dirancang dengan menggunakan model *waterfall*. Tahap pengembangan sistem meliputi, analisis data yang dilakukan melalui wawancara, dan studi literatur. Desain sistem, penulisan kode program dengan menerapkan metode *Simple Additive Weighting* (SAW) untuk proses menghitung nilai hasil penjumlahan dari bobot kriteria, subkriteria, dan normalisasi. Pengujian *white box* yang menghasilkan nilai kompleksitas siklomatik sebesar 1, sehingga baris kode metode pada sistem ini terbilang memiliki tingkat kompleksitas yang tidak terlalu tinggi dan dapat dilakukan proses *maintenance* dengan mudah jika terjadi kesalahan. Sistem pendukung keputusan penentuan prioritas pendistribusian air bersih memiliki dua hak akses yaitu admin dan surveyor. Peran utama admin dalam sistem ini yaitu menganalisis urutan prioritas pendistribusian air bersih dari hasil survei kondisi masing-masing wilayah kecamatan yang diinputkan oleh surveyor.
2. Penerapan metode *Simple Additive Weighting* (SAW) pada sistem prioritas pendistribusian air bersih ini memerlukan dua hak akses yaitu admin dan surveyor. Hasil analisis pendistribusian air bersih ditampilkan dalam bentuk urutan prioritas, jadi urutan yang pertama merupakan wilayah yang benar-benar membutuhkan distribusi air bersih. Pada penelitian ini, contoh yang dianalisis

yaitu survei pada Kecamatan Botolinggo, dan hasil yang didapat menunjukkan 6 desa membutuhkan distribusi air bersih, dengan urutan prioritas desa yaitu, desa Klekean, desa Lumutan, desa Lanas, desa Gayam, desa Sumber Canting, dan terakhir desa Gayam Lor. Urutan prioritas didapatkan dari hasil perhitungan sistem yang menggunakan metode *Simple Additive Weighting*, karena didalam sistem akan diurutkan secara otomatis dari perhitungan nilai tertinggi hingga yang paling rendah. Dimana untuk wilayah dengan hasil perhitungan tertinggi akan mendapat urutan prioritas pertama dan untuk selanjutnya menyesuaikan menurut hasil perhitungan masing-masing.

3. Dengan adanya sistem pendukung keputusan penentuan prioritas pendistribusian air bersih menggunakan metode *Simple Additive Weighting* (SAW) dapat membantu instansi terkait yaitu BPBD Kabupaten Bondowoso untuk menentukan prioritas pendistribusian air bersih dengan lebih tepat dan terjadwal, karena sebelum menggunakan sistem ini pihak terkait melakukan distribusi air bersih dengan menunggu laporan dari tiap kecamatan atau desa yang meminta pasokan air bersih ketika masuk musim kemarau tiba.

6.2 Saran

Beberapa saran dan masukan berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu :

1. Melakukan penerapan metode lain untuk penentuan prioritas pendistribusian air bersih agar dapat dilakukan perbandingan antar metode.
2. Dapat ditambahkan fitur untuk manajemen anggaran proyek pendistribusian air bersih sehingga dapat menghitung rancangan pembiayaan suatu proyek.
3. Pengembangan lebih lanjut sistem ini dapat dilakukan dengan membuat aplikasi berbasis *mobile* agar lebih memudahkan petugas lapangan dalam melakukan survei.

DAFTAR PUSTAKA

- Agissa, Wildan. (2013). *White Box and Black Box testing*. Diambil kembali Agustus 14, 2015, dari <http://bangwildan.web.id/berita-176-white-box-testing--black-box-testing.html>
- Agustina, Merry & Bakti, A. Mutatkin. (2015). *Penentuan Distribusi Air Bersih di Kabupaten X menggunakan Metode SAW (Simple Additive Weighting)*. Jurnal Seminar Nasional Teknologi Informasi, Komunikasi dan Aplikasinya Volume 03, 187 -188.
- BMKG, Humas (2015, Oktober 21). *PRAKIRAAN MUSIM KEMARAU 2015*. Diambil kembali September 11, 2015, dari http://bmkg.go.id/BMKG_pusat/Sestama/Humas/PRAKIRAAN_MUSIM_KEMARAU_2015.bmkg
- Handoko, Henrycus Bagus (2016). *Sistem Pendukung Penentuan Keputusan Distribusi Pengadaan Air Bersih Dengan Metode Technique for order Preference by Similarity to Ideal Solution (TOPSIS) (Studi Kasus Kabupaten Sragen)*. Skripsi. Program Studi Teknik Informatika. Fakultas Sains dan Teknologi. Universitas Sanata Dharma Yogyakarta.
- Kahraman, C. (2008). *MULTI-CRITERIA DECISION MAKING*. In *Fuzzy Multi-Criteria Decision Making* (pp. 1–18).
<https://doi.org/10.1016/j.prevetmed.2010.06.010>
- Kalia, Friska (2015, Juni 05). *Memasuki Kemarau, 10 Kecamatan Di Bondowoso Beresiko Kekeringan*. Diambil kembali Agustus 19, 2015, dari http://www.rri.co.id/http://portalkbr.com/062015/memasuki_kemarau__10_kecamatan_di_bondowoso_beresiko_kekeringan/71779.html
- Kusumadewi, S. (2006). *Fuzzy Multi Attribute Decision Making*. Yogyakarta: Graha Ilmu.
- Morton, P. G. (1978). *Decision Support Systems: An Organizational Perspective*. Reading MA: Addison-Wesley.

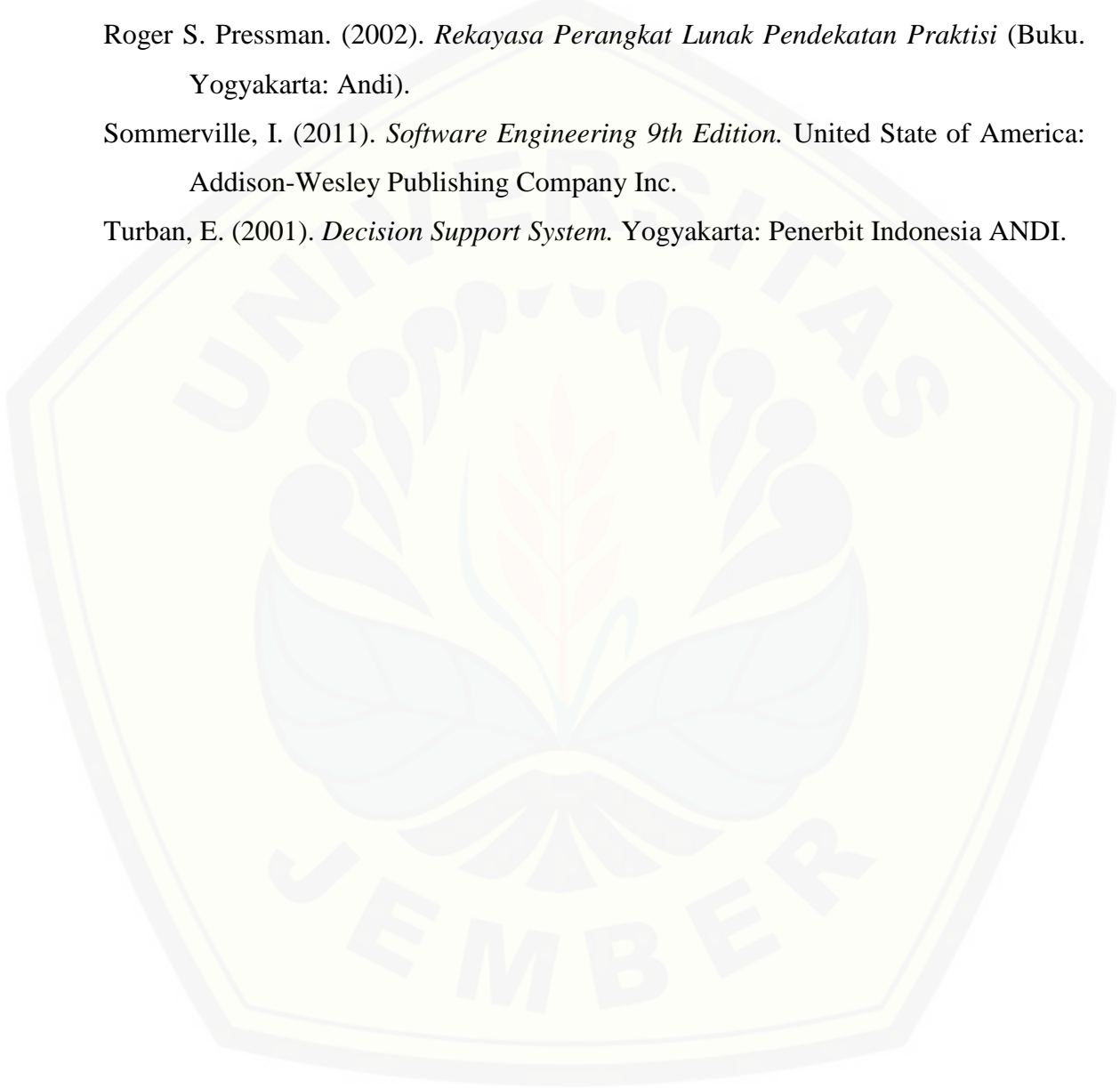
Rahmato, D., *Famine and Survival Strategis: A case Study from Northwest Ethiopia*,
Uppsala: Sacndinavian Institute for African Studies, 1991.

Red, Shelia B. 1992. *Pengantar Tentang Bahaya*. Edisi Ke 3, UNDP dan DMTP.

Roger S. Pressman. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi* (Buku.
Yogyakarta: Andi).

Sommerville, I. (2011). *Software Engineering 9th Edition*. United State of America:
Addison-Wesley Publishing Company Inc.

Turban, E. (2001). *Decision Support System*. Yogyakarta: Penerbit Indonesia ANDI.



LAMPIRAN

1. Lampiran Skenario

1.1. Skenario Mengelola Data User

Nomor <i>Use Case</i>	UC-01
Nama	Mengelola Data User
Aktor	Admin
<i>Pre Condition</i>	Admin harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Admin berhasil mengelola data user.
SKENARIO NORMAL MENGELOLA DATA USER	
TAMBAH USER	
Aktor	Sistem
6. Memilih menu <i>Kelola User</i> .	
	7. Menampilkan halaman kelola user yang berisi tabel user dengan kolom Nama, Email, Username, Phone, Alamat, User Image, Last Login, Level, tombol Tambah, tombol Detail, tombol Update dan tombol Delete.
8. Klik tombol Tambah.	
	9. Menampilkan halaman Form Tambah User yang berisi form tambah user dengan kolom Nama, No Telepon, Alamat, User Image, User Level, Email, Username,

	<p>Password, Konfirmasi Password, tombol Tambah dan tombol Cancel.</p>
10. Mengisi Form Tambah User.	
11. Klik tombol Tambah.	
	<p>12. Menampilkan halaman kelola user yang berisi tabel user dengan kolom Nama, Email, Username, Phone, Alamat, User Image, Last Login, Level, tombol Tambah, tombol Detail, tombol Update dan tombol Delete.</p>
SKENARIO ALTERNATIF MENGELOLA DATA USER	
TOMBOL CANCEL PADA TAMBAH USER	
6a. Klik tombol Cancel.	
	<p>7a. Menampilkan halaman kelola user yang berisi tabel user dengan kolom Nama, Email, Username, Phone, Alamat, User Image, Last Login, Level, tombol Tambah, tombol Detail, tombol Update dan tombol Delete.</p>
SKENARIO ALTERNATIF MENGELOLA DATA USER	
DATA TIDAK LENGKAP	
6b. Klik tombol <i>Tambah</i> .	
	<p>7b. Menampilkan <i>warning</i> “<i>This field is required</i>”.</p>
SKENARIO NORMAL MENGELOLA DATA USER	

TOMBOL UPDATE	
Aktor	Sistem
1. Memilih menu <i>Kelola User</i> .	
	2. Menampilkan halaman kelola user yang berisi tabel user dengan kolom Nama, Email, Username, Phone, Alamat, User Image, Last Login, Level, tombol Tambah, tombol Detail, tombol Update dan tombol Delete.
3. Klik tombol <i>Update</i> .	
	4. Menampilkan form update user yang berisi tabel user dengan kolom Nama, No Telepon, Alamat, User Level, Email, Username, User Image, tombol Update, tombol Cancel, tombol Ubah Password.
5. Mengubah isi dataform user.	
6. Klik tombol <i>Update</i> .	
	7. Menampilkan halaman kelola user yang berisi tabel user dengan kolom Nama, Email, Username, Phone, Alamat, User Image, Last Login, Level, tombol Tambah, tombol Detail, tombol Update dan tombol

	Delete.
SKENARIO ALTERNATIF MENGELOLA DATA USER	
TOMBOL CANCEL PADA UPDATE USER	
6a. Klik tombol Cancel.	
	7a. Menampilkan halaman kelola user yang berisi tabel user dengan kolom Nama, Email, Username, Phone, Alamat, User Image, Last Login, Level, tombol Tambah, tombol Detail, tombol Update dan tombol Delete.
SKENARIO NORMAL MENGELOLA DATA USER	
TOMBOL DELETE	
1. Klik tombol <i>Delete</i> .	
	2. Menampilkan pop up berupa tulisan <i>Are You Sure ?</i> . Tombol Ok dan tombol Cancel.
SKENARIO NORMAL MENGELOLA DATA USER	
TOMBOL DETAIL	
1. Klik tombol <i>Detail</i> .	
	2. Menampilkan halaman user detail yang berisi tabel user dengan kolom Nama, Email, Username, Phone, Alamat, User Image, Last Login, Active, tombol Back.

1.2.Skenario Mengelola Data Kriteria

Nomor <i>Use Case</i>	UC-02
Nama	Mengelola Data Kriteria
Aktor	Admin
<i>Pre Condition</i>	Admin harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Admin berhasil mengelola data kriteria.
SKENARIO NORMAL MENGELOLA DATA KRITERIA	
TAMBAH KRITERIA	
Aktor	Sistem
4. Memilih menu Kriteria.	
	5. Menampilkan halaman data kriteria yang berisi tabel kriteria <i>list</i> dengan kolom No, Nama Kriteria, Bobot, Bobot Desimal, Subkriteria, tombol <i>Detail</i> , tombol <i>Update</i> dan tombol <i>Delete</i> .
6. Klik tombol <i>Tambah</i> .	
	7. Menampilkan halaman <i>Form Tambah</i> yang berisi <i>form</i> tambah kriteria dengan kolom Nama, Bobot, Bobot Desimal, tombol <i>Tambah</i> dan tombol <i>Cancel</i> .
8. Mengisi <i>Form Tambah Kriteria</i> .	
9. Klik tombol <i>Tambah</i> .	
	10. Menyimpan data baru ke database

	11. Menampilkan halaman data kriteria yang berisi tabel kriteria <i>list</i> dengan kolom No, Nama Kriteria, Bobot, Bobot Desimal, Subkriteria, tombol <i>Detail</i> , tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGELOLA DATA KRITERIA	
TOMBOL CANCEL PADA TAMBAH KRITERIA	
Aktor	Sistem
1a. Klik tombol <i>Cancel</i> .	
	2a. Menampilkan halaman data kriteria yang berisi tabel kriteria <i>list</i> dengan kolom No, Nama Kriteria, Bobot, Bobot Desimal, Subkriteria, tombol <i>Detail</i> , tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO NORMAL MENGELOLA DATA KRITERIA	
UPDATE KRITERIA	
Aktor	Sistem
1. Klik tombol <i>Update</i> .	
	2. Menampilkan halaman <i>Form Update Kriteria</i> yang berisi <i>form</i> dengan kolom Nama Kriteria, Bobot, Bobot Desimal, tombol <i>Update</i> dan tombol <i>Cancel</i> .
3. Mengisi <i>Form Update Kriteria</i> .	
4. Klik tombol <i>Update</i> .	

	5. Menyimpan data baru ke database
	6. Menampilkan halaman data kriteria yang berisi tabel kriteria <i>list</i> dengan kolom No, Nama Kriteria, Bobot, Bobot Desimal, Subkriteria, tombol <i>Detail</i> , tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGELOLA DATA KRITERIA	
TOMBOL CANCEL PADA FORM UPDATE KRITERIA	
Aktor	Sistem
1a. Klik tombol <i>Cancel</i> .	
	2a. Menampilkan halaman data kriteria yang berisi tabel kriteria <i>list</i> dengan kolom No, Nama Kriteria, Bobot, Bobot Desimal, Subkriteria, tombol <i>Detail</i> , tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO NORMAL MENGELOLA DATA KRITERIA	
DETAIL KRITERIA	
Aktor	Sistem
1. Klik tombol <i>Detail</i> .	
	2. Menampilkan halaman Kriteria Detail, yang berisi kolom Nama Kriteria, Bobot Kriteria, Bobot Kriteria Persen, dan tombol <i>Back</i> .
SKENARIO NORMAL MENGELOLA DATA KRITERIA	

DELETE KRITERIA	
Aktor	Sistem
1. Klik tombol <i>Delete</i> .	
	2. Menampilkan pop up berupa tulisan <i>Are You Sure ?</i> . Tombol Ok dan tombol Cancel.

1.3.Skenario Mengelola Data Subkriteria

Nomor <i>Use Case</i>	UC-03
Nama	Mengelola Data Subkriteria
Aktor	Admin
<i>Pre Condition</i>	Admin harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Admin berhasil mengelola data subkriteria.
SKENARIO NORMAL MENGELOLA DATA SUBKRITERIA	
TAMBAH DATA SUBKRITERIA	
Aktor	Sistem
12. Klik tombol <i>Subkriteria</i> .	
	13. Menampilkan halaman data subkriteria <i>list</i> yang berisi Kriteria, Bobot Kriteria, Nama Subkriteria, Bobot Subkriteria, W (Bobot Kriteria x Bobot Subkriteria), Sub Sub Kriteria, dan tombol <i>Detail</i> .
14. Klik tombol <i>Detail</i> .	

	15. Menampilkan halaman data subsubkriteria <i>list</i> yang berisi Subkriteria, Deskripsi, Nilai, tombol Tambah, tombol <i>Detail</i> , tombol <i>Update</i> , tombol <i>Delete</i> .
16. Klik tombol <i>Tambah</i> .	
	17. Menampilkan halaman <i>Form Tambah Subsubkriteria</i> yang berisi <i>form</i> dengan kolom Subkriteria, Deskripsi, Nilai, tombol <i>Tambah</i> dan tombol <i>Cancel</i> .
18. Mengisi <i>Form Tambah SubsubKriteria</i>	
19. Klik tombol <i>Tambah</i> .	
	20. Menyimpan data baru ke database.
	21. Menampilkan halaman data subsubkriteria <i>list</i> yang berisi Subkriteria, Deskripsi, Nilai, tombol Tambah, tombol <i>Detail</i> , tombol <i>Update</i> , tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGELOLA DATA SUBKRITERIA	
DATA TIDAK LENGKAP	
Aktor	Sistem
5a. Klik tombol <i>Tambah</i> .	
	6b. Menampilkan <i>warning</i> “ <i>This field is required</i> ”.
SKENARIO ALTERNATIF MENGELOLA DATA SUBKRITERIA	

TOMBOL CANCEL PADA FORM TAMBAH SUBKRITERIA	
Aktor	Sistem
5b. Klik tombol <i>Cancel</i> .	
	6b. Menampilkan halaman data subsubkriteria <i>list</i> yang berisi Subkriteria, Deskripsi, Nilai, tombol Tambah, tombol <i>Detail</i> , tombol <i>Update</i> , tombol <i>Delete</i> .
SKENARIO NORMAL MENGELOLA DATA SUBKRITERIA	
DETAIL DATA SUBKRITERIA	
Aktor	Sistem
1. Klik tombol <i>Subkriteria</i> .	
	2. Menampilkan halaman data subkriteria <i>list</i> yang berisi Kriteria, Bobot Kriteria, Nama Subkriteria, Bobot Subkriteria, W (Bobot Kriteria x Bobot Subkriteria), Sub Sub Kriteria, dan tombol <i>Detail</i> .
3. Klik tombol <i>Detail</i> .	
	4. Menampilkan halaman data subsubkriteria yang berisi Subkriteria, Deskripsi, Nilai, tombol Tambah, tombol <i>Detail</i> .
SKENARIO NORMAL MENGELOLA DATA SUBKRITERIA	
UPDATE DATA SUBKRITERIA	
Aktor	Sistem

1. Klik tombol <i>Update</i> .	
	2. Menampilkan halaman <i>Form Update Subsubkriteria</i> yang berisi <i>form</i> dengan kolom Subkriteria, Deskripsi, Nilai, tombol <i>Update</i> dan tombol <i>Cancel</i> .
3. Mengisi <i>Form Update SubsubKriteria</i>	
4. Klik tombol <i>Update</i> .	
	5. Menyimpan data baru ke database.
	6. Menampilkan halaman data subsubkriteria <i>list</i> yang berisi Subkriteria, Deskripsi, Nilai, tombol Tambah, tombol <i>Detail</i> , tombol <i>Update</i> , tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGELOLA DATA SUBKRITERIA	
DATA TIDAK LENGKAP	
Aktor	Sistem
5a. Klik tombol <i>Update</i> .	
	6b. Menampilkan <i>warning</i> “ <i>This field is required</i> ”.
SKENARIO ALTERNATIF MENGELOLA DATA SUBKRITERIA	
TOMBOL CANCEL PADA FORM TAMBAH SUBKRITERIA	
Aktor	Sistem
5b. Klik tombol <i>Cancel</i> .	

	6b. Menampilkan halaman data subsubkriteria <i>list</i> yang berisi Subkriteria, Deskripsi, Nilai, tombol Tambah, tombol <i>Detail</i> , tombol <i>Update</i> , tombol <i>Delete</i> .
SKENARIO NORMAL MENGELOLA DATA SUBKRITERIA	
DELETE DATA SUBKRITERIA	
Aktor	Sistem
3. Klik tombol <i>Delete</i> .	
	4. Menampilkan pop up berupa tulisan <i>Are You Sure ?</i> . Tombol Ok dan tombol Cancel.

1.4.Skenario Mengelola Data Survei

Nomor <i>Use Case</i>	UC-04
Nama	Mengelola Data Survei
Aktor	Admin
<i>Pre Condition</i>	Admin harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Admin berhasil mengelola data survei.
SKENARIO NORMAL MENGELOLA DATA SURVEI	
TAMBAH DATA SURVEI	
Aktor	Sistem
22. Memilih <i>Survei</i> .	
	23. Menampilkan halaman survei <i>list</i> dengan kolom Tahun, Provinsi,

	Kabupaten/Kota, Kecamatan, Status, Tanggal Dropping, tombol Tambah, tombol Proses, tombol <i>Update</i> dan tombol <i>Delete</i> .
24. Klik tombol <i>Tambah</i> .	
	25. Menampilkan halaman <i>Form Tambah Survei</i> yang berisi <i>form</i> tambah Provinsi, Kabupaten/Kota, Kecamatan, Tahun, tombol <i>Tambah</i> dan tombol <i>Cancel</i> .
26. Mengisi <i>Form Tambah Survei</i> .	
27. Klik tombol <i>Tambah</i> .	
	28. Menyimpan data baru ke database. 29. Menampilkan halaman survei <i>list</i> dengan kolom Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Status, Tanggal Dropping, tombol Tambah, tombol Proses, tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGELOLA DATA SURVEI	
TOMBOL CANCEL PADA DATA SURVEI	
Aktor	Sistem
6a. Klik tombol <i>Cancel</i> .	
	7a. Menampilkan halaman survei <i>list</i> dengan kolom Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Status, Tanggal Dropping, tombol

	Tambah, tombol Proses, tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGELOLA DATA SURVEI	
DATA TIDAK LENGKAP	
Aktor	Sistem
6b. Klik tombol <i>Tambah</i> .	
	7b. Menampilkan <i>warning</i> “ <i>This field is required</i> ”.
SKENARIO NORMAL MENGELOLA DATA SURVEI	
PROSES DATA SURVEI	
Aktor	Sistem
1. Klik tombol <i>Proses</i> .	
	2. Menampilkan <i>pop up</i> berupa tulisan Status Survei, dengan tombol pilihan <i>drop down</i> proses atau selesai, serta input tanggal dropping air.
3. Mengisi <i>Form Pop Up Proses</i> .	
4. Klik tombol <i>Simpan</i> .	
	5. Menyimpan data baru ke database.
	6. Menampilkan <i>pop up</i> status survei, tombol <i>drop down</i> proses atau selesai, serta tanggal dropping air. Tombol Simpan dan tombol <i>close</i> .

SKENARIO ALTERNATIF MENGELOLA DATA SURVEI	
DATA TIDAK LENGKAP	
Aktor	Sistem
4a. Klik tombol <i>Simpan</i> .	
	5b. Menampilkan <i>pop up</i> "Please fill out this field".
SKENARIO ALTERNATIF MENGELOLA DATA SURVEI	
TOMBOL CLOSE PADA FORM POP UP PROSES	
Aktor	Sistem
6a. Klik tombol <i>Close</i> .	
	7b. Menampilkan halaman survei <i>list</i> dengan kolom Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Status, Tanggal Droping, tombol Tambah, tombol Proses, tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO NORMAL MENGELOLA DATA SURVEI	
UPDATE DATA SURVEI	
Aktor	Sistem
8. Klik tombol <i>Update</i> .	
	9. Menampilkan <i>form update survei</i> yang berisi <i>form</i> dengan kolom Provinsi, Kabupaten/Kota, Kecamatan, Tahun, tombol <i>Update</i> dan tombol <i>Cancel</i> .

10. Mengubah isi data <i>form survei</i> .	
11. Klik tombol <i>Update</i> .	
	12. Menyimpan data baru ke database.
	13. Menampilkan halaman survei <i>list</i> dengan kolom Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Status, Tanggal Dropped, tombol Tambah, tombol Proses, tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGELOLA DATA SURVEI	
TOMBOL CANCEL PADA UPDATE SURVEI	
Aktor	Sistem
4a. Klik tombol <i>Cancel</i> .	
	4b. Menampilkan halaman survei <i>list</i> dengan kolom Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Status, Tanggal Dropped, tombol Tambah, tombol Proses, tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO NORMAL MENGELOLA DATA SURVEI	
DELETE DATA SURVEI	
Aktor	Sistem
1. Klik tombol <i>Delete</i> .	
	2. Menampilkan pop up berupa tulisan <i>Are You Sure ?</i> . Tombol Ok dan tombol Cancel.

1.5.Skenario Mengelola Data Survei Detail

Nomor <i>Use Case</i>	UC-06
Nama	Mengelola Data Survei Detail
Aktor	Admin
<i>Pre Condition</i>	Admin harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Admin berhasil mengelola data survei detail.
SKENARIO NORMAL MENGELOLA DATA SURVEI DETAIL	
TAMBAH DATA SURVEI DETAIL	
Aktor	Sistem
30. Klik menu <i>Survei</i> .	
	31. Menampilkan halaman data survei <i>list</i> yang Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Kecamatan, Status, Tanggal Dropped, tombol <i>Tambah</i> , tombol <i>Update</i> , dan tombol <i>Delete</i> .
32. Klik tombol <i>Tambah</i> .	
	4. Menampilkan halaman <i>Form Tambah Survei</i> yang berisi kolom form Provinsi, Kabupaten/Kota, Kecamatan, Tahun serta tombol <i>Tambah</i> , tombol <i>Cancel</i> .
9. Mengisi <i>Form Tambah Survei</i> .	
10. Klik tombol <i>Update</i> .	
	11. Menyimpan data baru ke database

	12. Menampilkan halaman data survei <i>list</i> yang Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Kecamatan, Status, Tanggal Droping, tombol <i>Tambah</i> , tombol <i>Update</i> , dan tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGELOLA DATA SURVEI DETAIL	
DATA TIDAK LENGKAP	
Aktor	Sistem
8a. Klik tombol <i>Update</i> .	
	9a. Menampilkan <i>warning</i> “ <i>This field is required</i> ”.
SKENARIO ALTERNATIF MENGELOLA DATA SURVEI DETAIL	
TOMBOL CANCEL PADA DATA SURVEI	
Aktor	Sistem
8b. Klik tombol <i>Cancel</i> .	
	9b. Menampilkan halaman data survei <i>list</i> yang Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Kecamatan, Status, Tanggal Droping, tombol <i>Tambah</i> , tombol <i>Update</i> , dan tombol <i>Delete</i> .
SKENARIO NORMAL MENGELOLA DATA SURVEI DETAIL	
DELETE DATA SURVEI	
Aktor	Sistem
33. Klik menu <i>Survei</i> .	

	34. Menampilkan halaman data survei <i>list</i> yang Tahun, Provinsi, Kabupaten/Kota, Kecamatan, Kecamatan, Status, Tanggal Droping, tombol <i>Tambah</i> , tombol <i>Update</i> , dan tombol <i>Delete</i> .
35. Klik tombol <i>Delete</i> .	
	4. Menampilkan pop up berupa tulisan <i>Are You Sure ?</i> . Tombol <i>Ok</i> dan tombol <i>Cancel</i> .

1.6.Skenario Mengubah Nama Data Kriteria

Nomor <i>Use Case</i>	UC-07
Nama	Mengubah Nama Data Kriteria
Aktor	Admin
<i>Pre Condition</i>	Admin harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Admin berhasil mengubah nama data kriteria.
SKENARIO NORMAL MENGUBAH NAMA DATA KRITERIA	
UBAH NAMA DATA KRITERIA	
Aktor	Sistem
1. Memilih menu <i>Kriteria</i> .	
	2. Menampilkan halaman data kriteria yang berisi tabel kriteria <i>list</i> dengan kolom No, Nama Kriteria, Bobot, Bobot Desimal, Subkriteria, tombol

	<i>Detail</i> , tombol <i>Update</i> dan tombol <i>Delete</i> .
3. Klik tombol <i>Update</i> .	
	4. Menampilkan halaman <i>Form Update Kriteria</i> yang berisi <i>form</i> dengan kolom Nama Kriteria, Bobot, Bobot Desimal, tombol <i>Update</i> dan tombol <i>Cancel</i> .
5. Mengisi <i>Form Update Kriteria</i> .	
6. Klik tombol <i>Update</i> .	
	7. Menyimpan data baru ke database. 8. Menampilkan halaman data kriteria yang berisi tabel kriteria <i>list</i> dengan kolom No, Nama Kriteria, Bobot, Bobot Desimal, Subkriteria, tombol <i>Detail</i> , tombol <i>Update</i> dan tombol <i>Delete</i> .
SKENARIO ALTERNATIF MENGUBAH NAMA DATA KRITERIA	
DATA TIDAK LENGKAP	
Aktor	Sistem
6a. Klik tombol <i>Update</i> .	
	7a. Menampilkan <i>warning</i> “ <i>This field is required</i> ”.
SKENARIO ALTERNATIF MENGUBAH NAMA DATA KRITERIA	
TOMBOL CANCEL PADA UPDATE KRITERIA	

Aktor	Sistem
6b. Klik tombol <i>Cancel</i> .	
	7b. Menampilkan halaman data kriteria yang berisi tabel kriteria <i>list</i> dengan kolom No, Nama Kriteria, Bobot, Bobot Desimal, Subkriteria, tombol <i>Detail</i> , tombol <i>Update</i> dan tombol <i>Delete</i> .

1.7.Skenario Mengubah Data Subkriteria

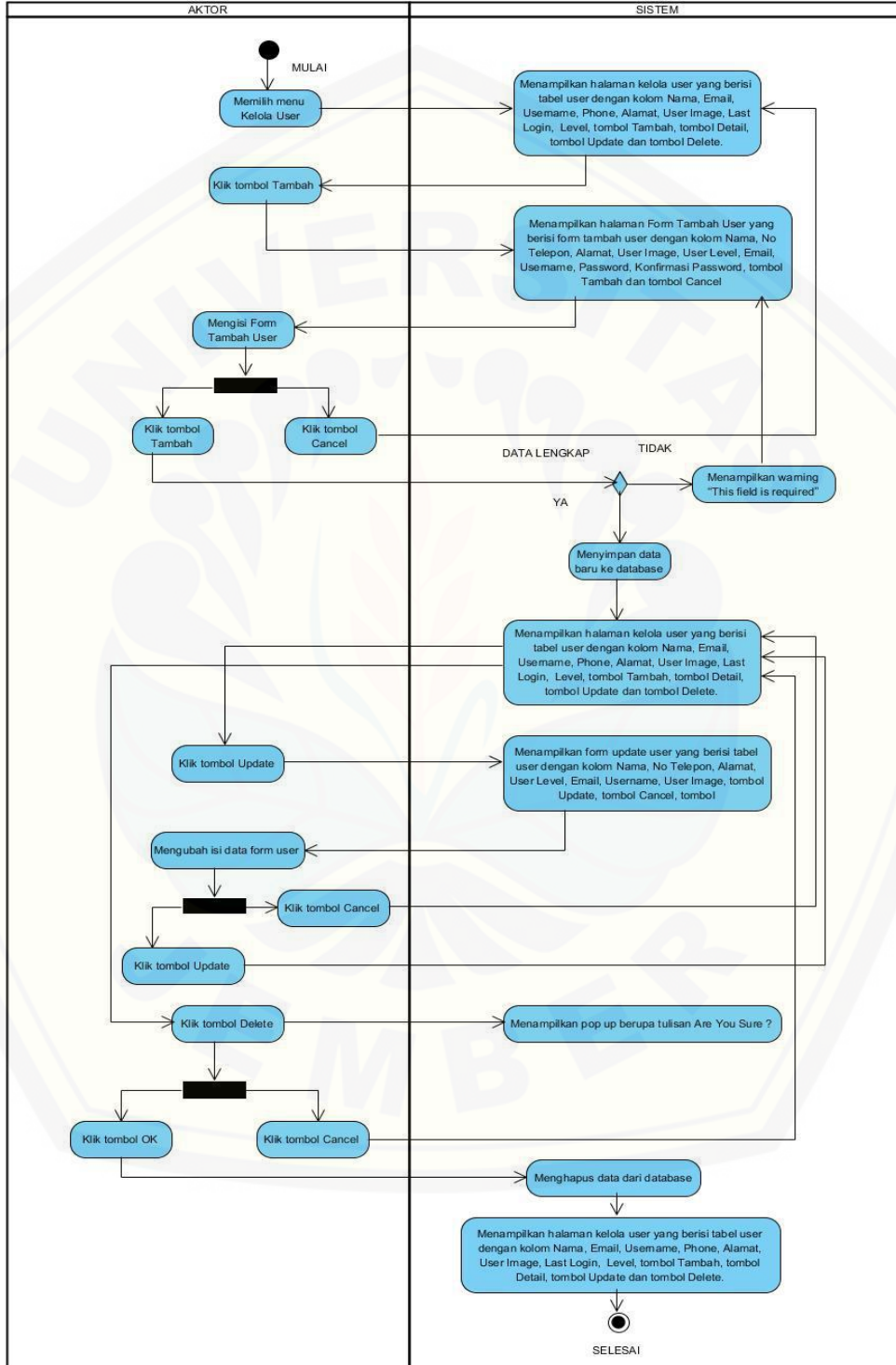
Nomor <i>Use Case</i>	UC-08
Nama	Mengubah Data Subkriteria
Aktor	Admin
<i>Pre Condition</i>	Admin harus sudah <i>login</i> sistem.
<i>Post Condition</i>	Admin berhasil mengubah data subkriteria.
SKENARIO NORMAL MENGUBAH DATA SUBKRITERIA	
UBAH DATA SUBKRITERIA	
Aktor	Sistem
36. Klik menu <i>Subkriteria</i> .	
	37. Menampilkan halaman data subkriteria <i>list</i> yang berisi Kriteria, Bobot Kriteria, Nama Subkriteria, Bobot Subkriteria, W (Bobot Kriteria x Bobot Subkriteria), Subsub Kriteria, serta tombol

	Detail.
38. Klik tombol <i>Detail</i> .	
	4. Menampilkan halaman data subsubkriteria <i>list</i> yang berisi subkriteria, deskripsi, nilai, serta tombol Update, tombol detail, tombol Delete.
13. Klik tombol <i>Update</i> .	
	14. Menampilkan halaman <i>Form Update Subsubkriteria</i> yang berisi <i>form</i> dengan kolom Subkriteria, Deskripsi, Nilai, tombol <i>Update</i> dan tombol <i>Cancel</i> .
15. Mengisi <i>Form Update Subsubkriteria</i> .	
16. Klik tombol <i>Update</i> .	
	17. Menyimpan data baru ke database. 18. Menampilkan halaman data subsubkriteria <i>list</i> yang berisi subkriteria, deskripsi, nilai, serta tombol Update, tombol detail, tombol Delete.
SKENARIO ALTERNATIF MENGUBAH DATA SUBKRITERIA	
DATA TIDAK LENGKAP	
Aktor	Sistem
8a. Klik tombol <i>Update</i> .	
	9a. Menampilkan <i>warning</i> “ <i>This field is required</i> ”.

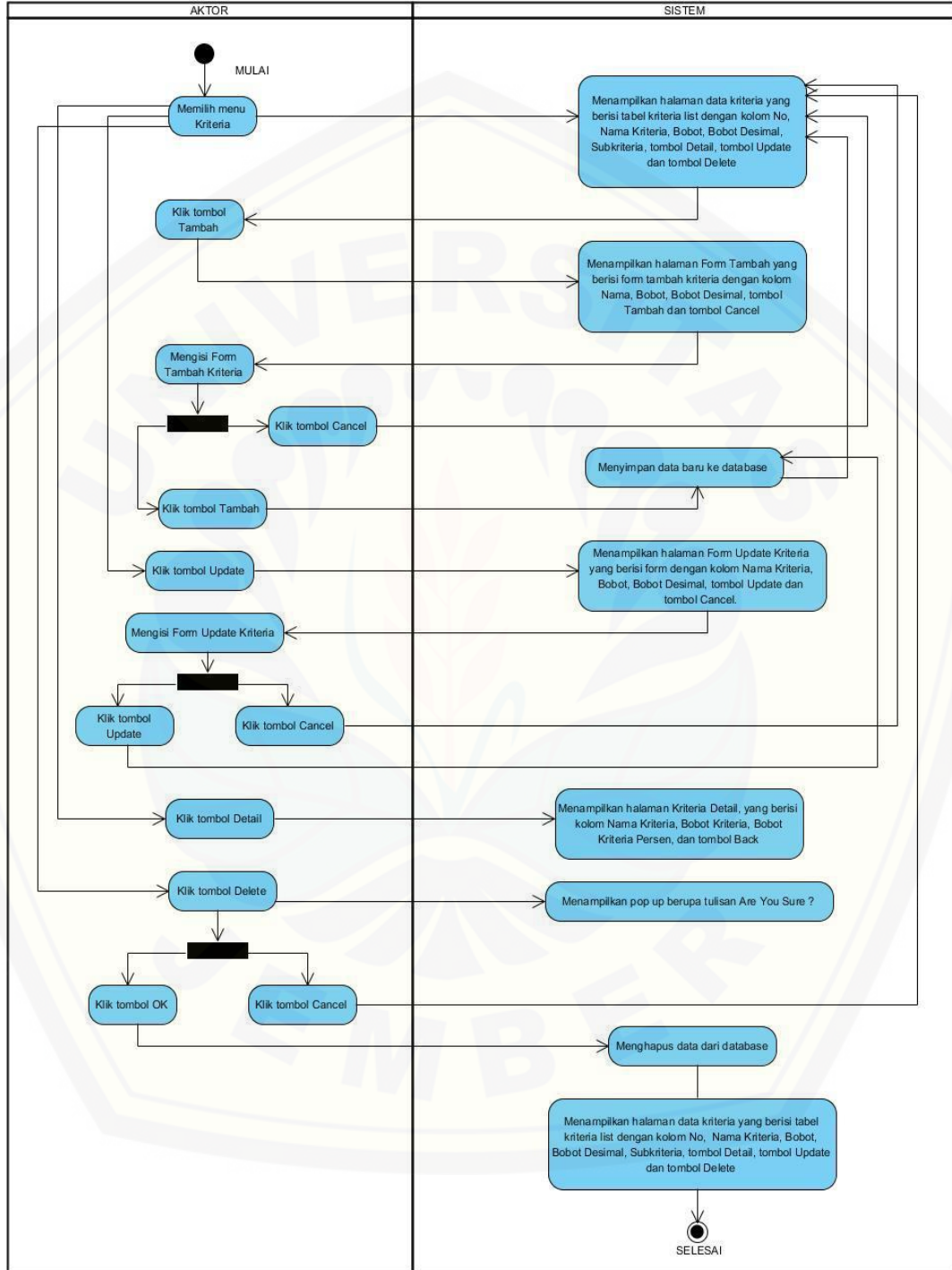
SKENARIO ALTERNATIF MENGUBAH DATA SUBKRITERIA	
DELETE SUBSUBKRITERIA	
Aktor	Sistem
8b. Klik tombol <i>Cancel</i> .	
	9b. Menampilkan halaman <i>Form Update Subsubkriteria</i> yang berisi <i>form</i> dengan kolom Subkriteria, Deskripsi, Nilai, tombol <i>Update</i> dan tombol <i>Cancel</i> .

2. Lampiran Activity Diagram

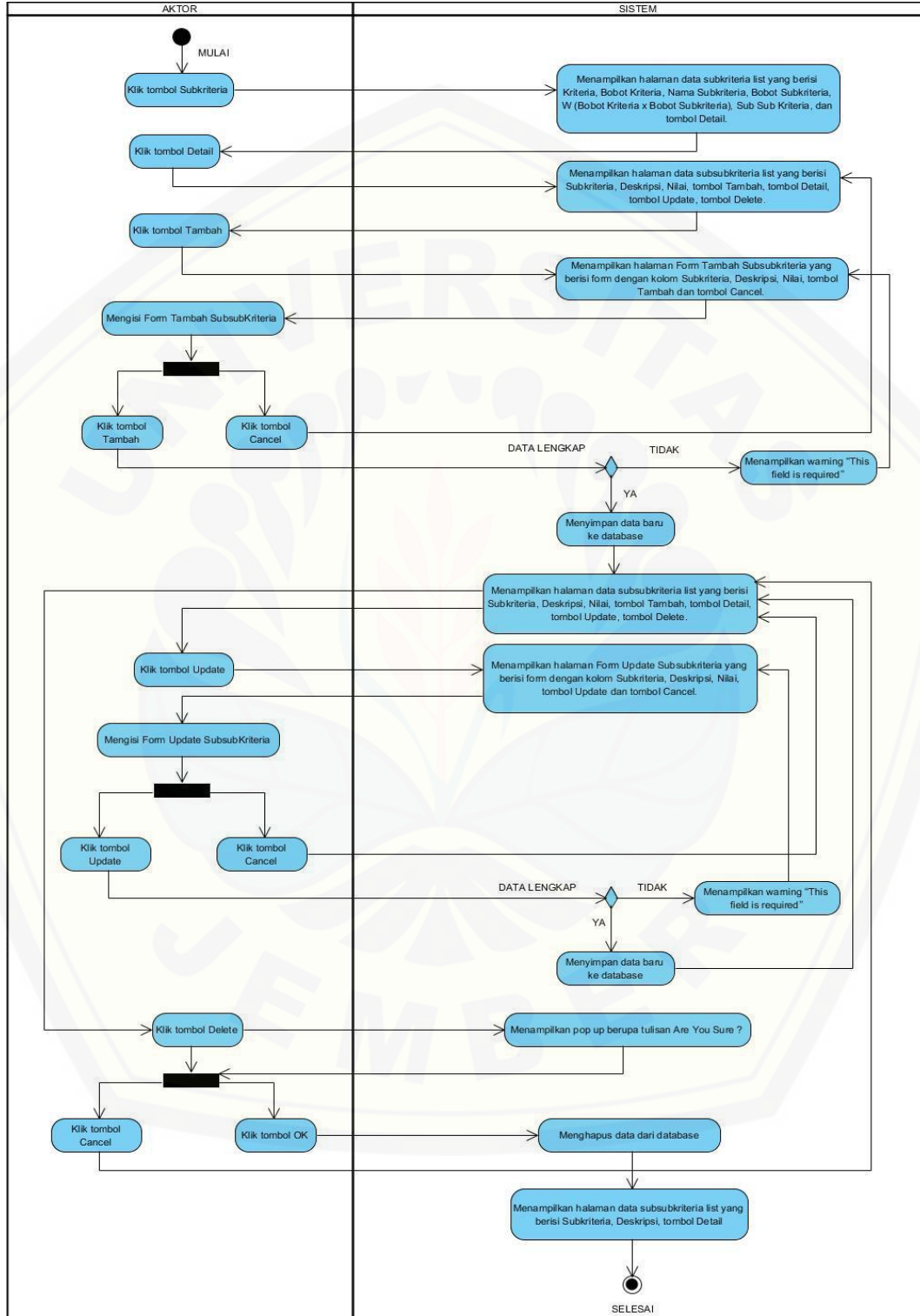
2.1. Activity Diagram Mengelola Data User



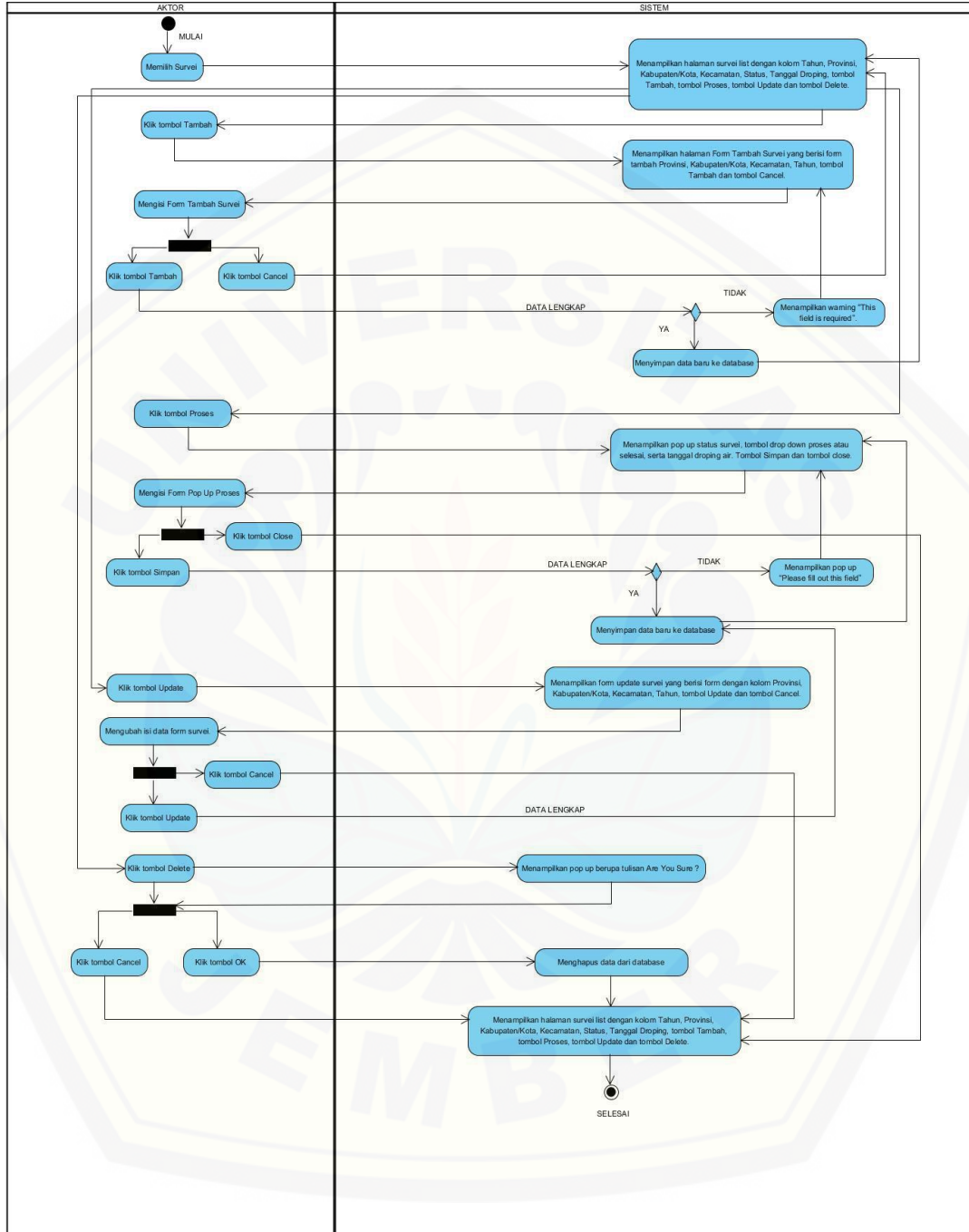
2.2. Activity Diagram Mengelola Data Kriteria



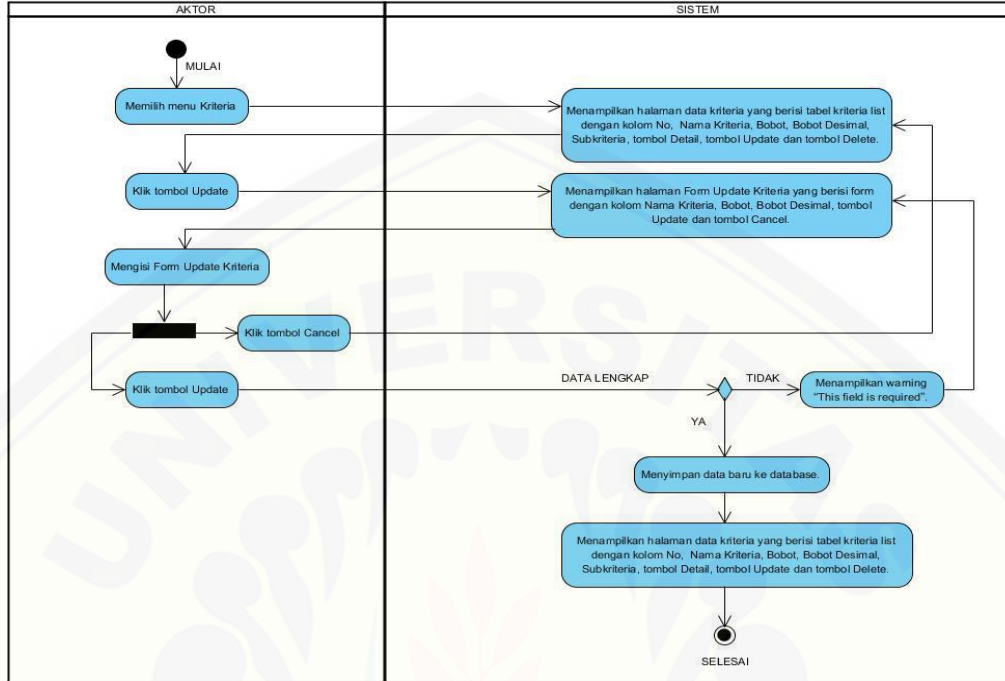
2.3. Activity Diagram Mengelola Data Subkriteria



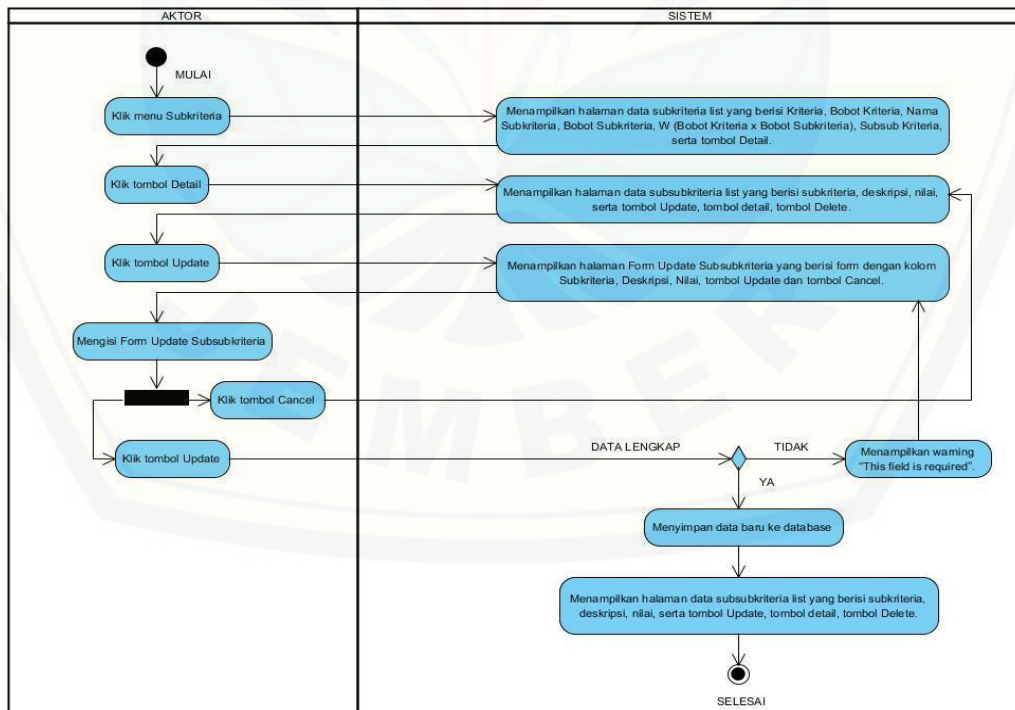
2.4. Activity Diagram Mengelola Data Survei



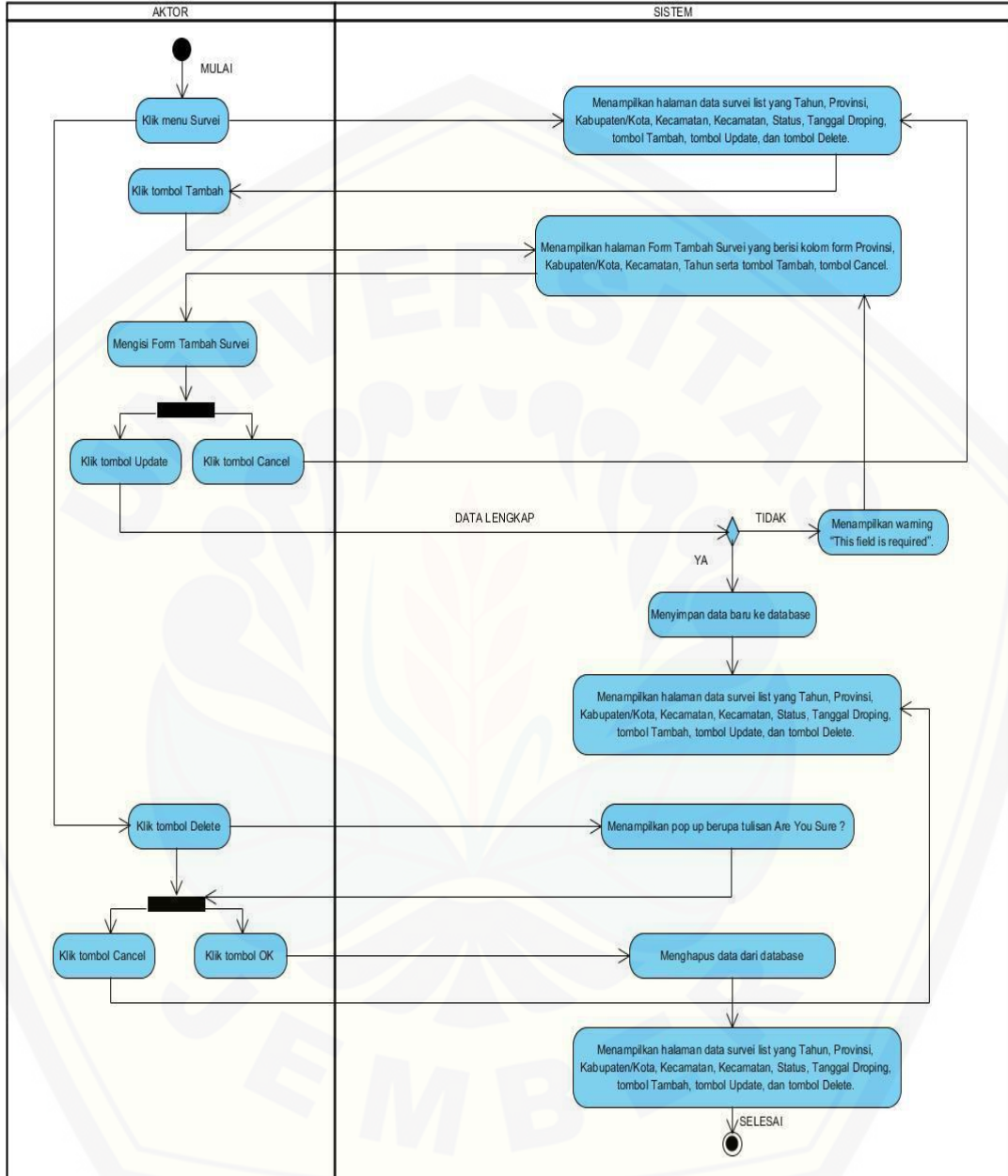
2.5. Activity Diagram Mengubah Nama Data Kriteria



2.6. Activity Diagram Mengubah Data Bobot Nilai Subkriteria

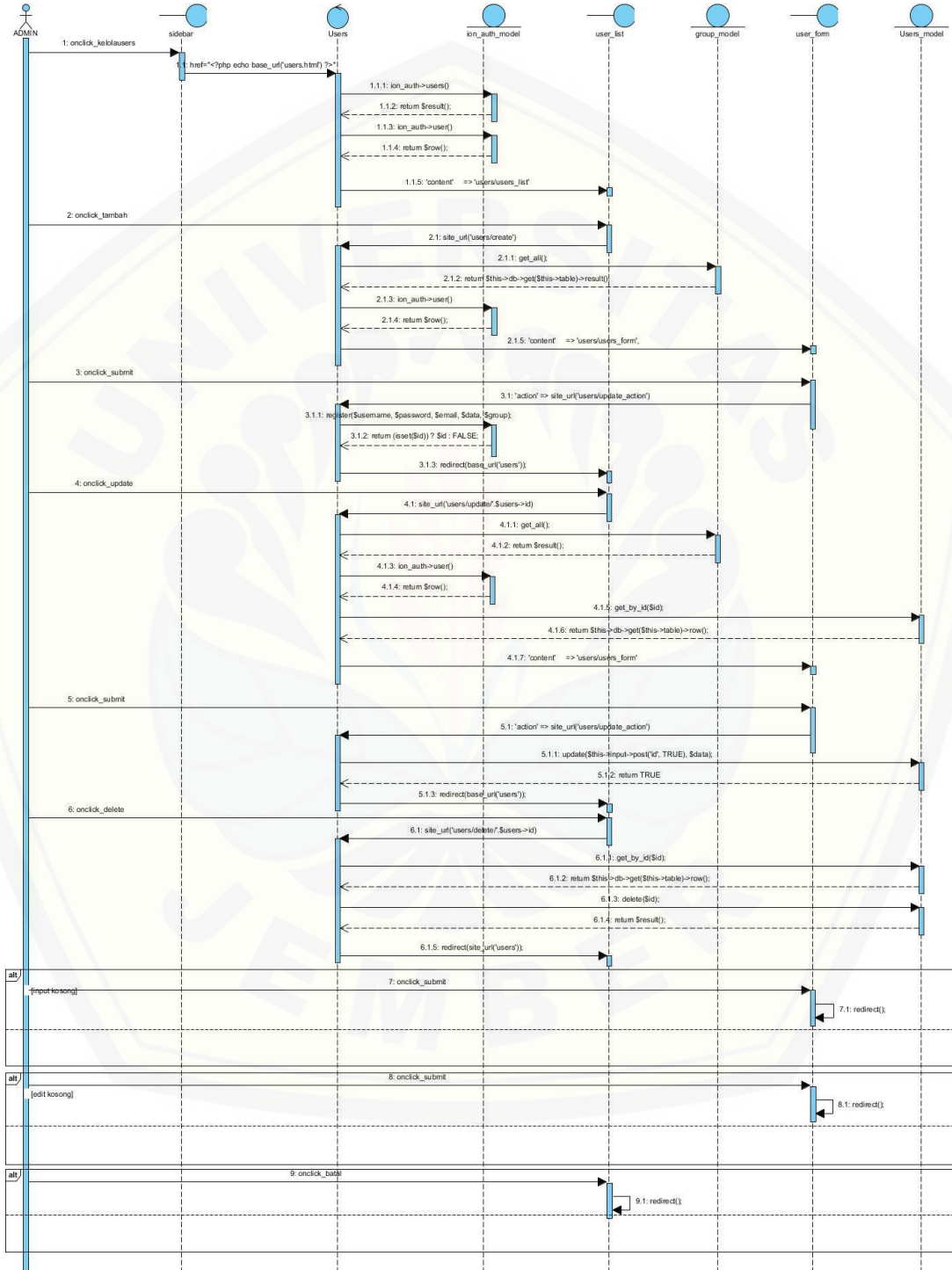


2.7. Activity Diagram Mengelola Data Survei Detail

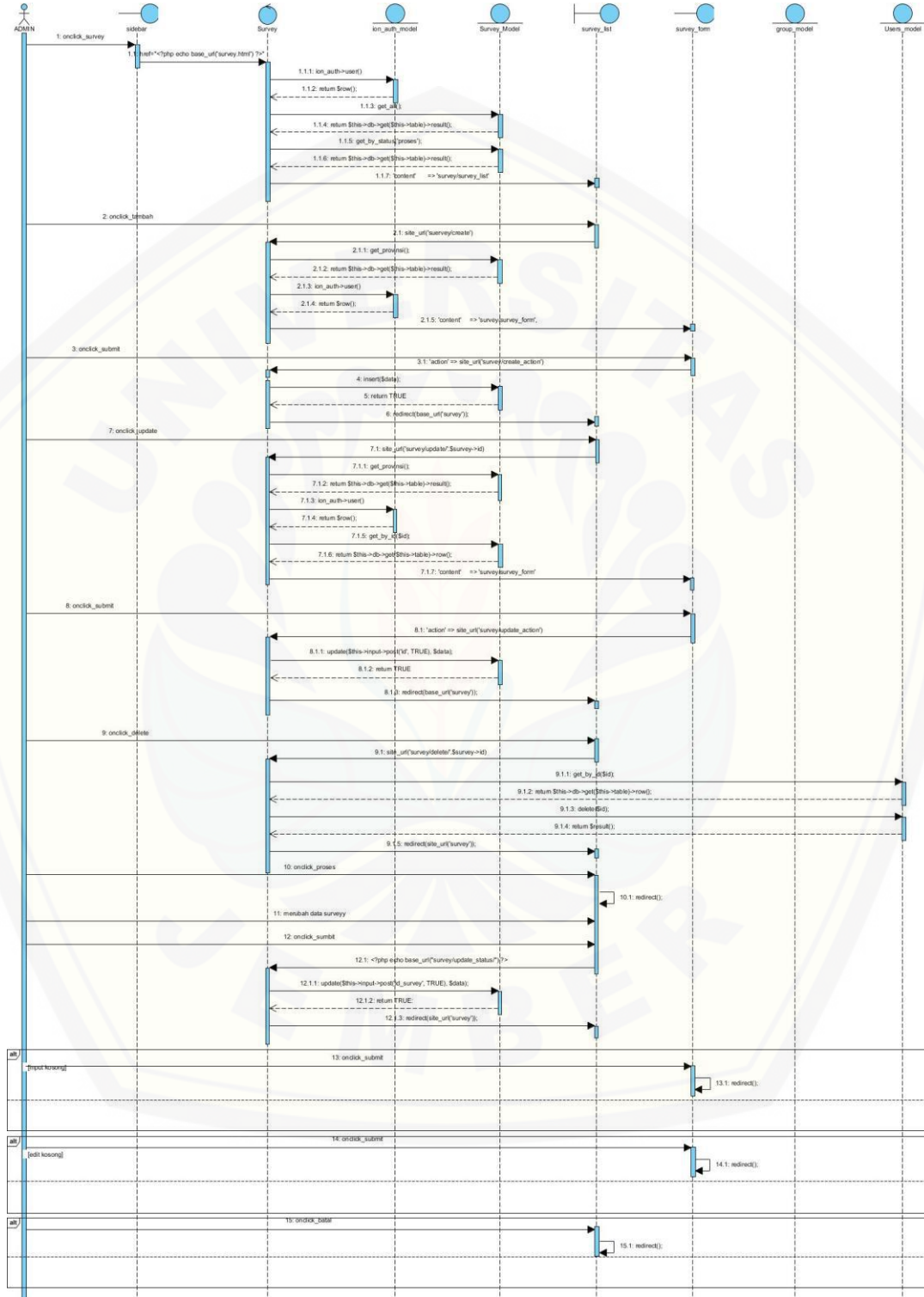


3. Lampiran Sequence Diagram

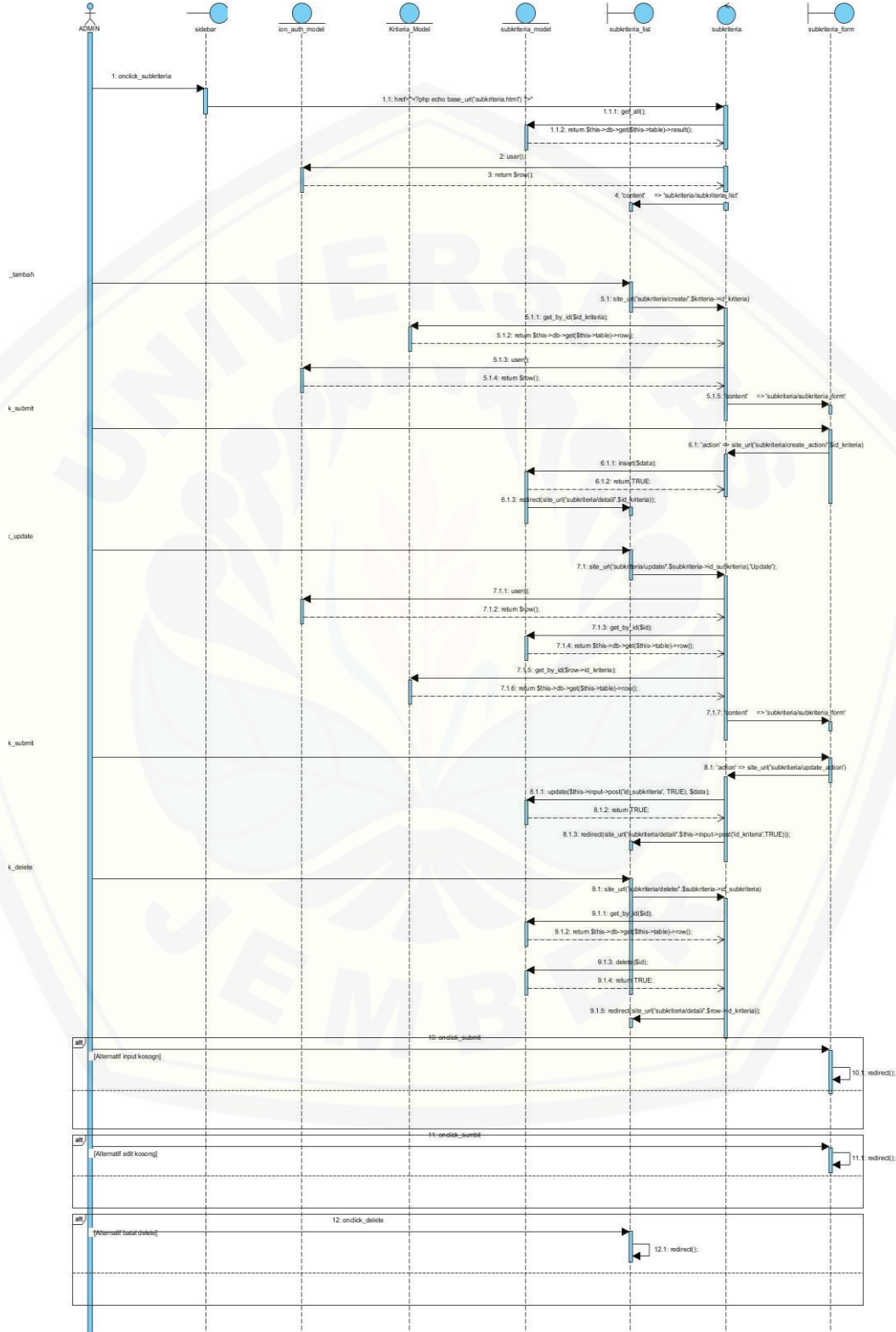
3.1. Sequence Diagram Mengelola Data User



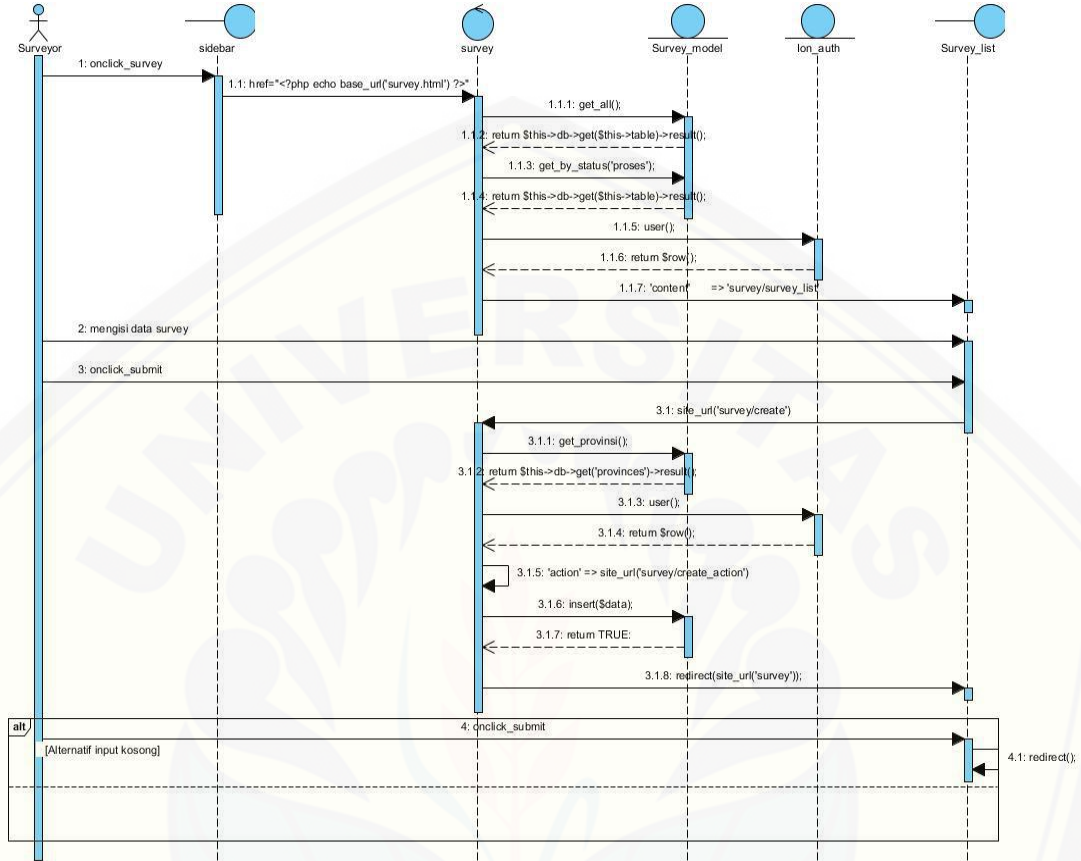
3.3. Sequence Diagram Mengelola Data Survei



3.4. Sequence Diagram Mengelola Subkriteria



3.5. Sequence Diagram Mengelola Data Survei Detail



4. Lampiran Implementasi Coding

4.1. Controller Login

```

1 |<?php defined('BASEPATH') OR exit('No direct script access allowed');
2
3 |class Auth extends CI_Controller {
4
5 |    public function __construct()
6 |    {
7 |        parent::__construct();
8 |        $this->load->database();
9 |        $this->load->helper(array('url','language'));
10
11 |        $this->form_validation->set_error_delimiters($this->config->item('error_start_delimiter', 'ion_auth'), $this->config->item('
12 |            error_end_delimiter', 'ion_auth'));
13
14 |        $this->lang->load('auth');
15 |    }
16
17 |    // redirect if needed, otherwise display the user list
18 |    public function index()
19 |    {
20 |        if (!$this->ion_auth->logged_in())
21 |        {
22 |            // redirect them to the login page
23 |            redirect('auth/login', 'refresh');
24 |        }
25 |        else
26 |        {
27 |            redirect('dashboard', 'refresh');
28 |        }
29 |    }
30
31 |    // log the user in
32 |    public function login()
33 |    {
34 |        if ($this->ion_auth->logged_in())
35 |        {
36 |            // redirect them to the login page
37 |            redirect('dashboard', 'refresh');
38 |        }
39 |
40 |        $this->data['title'] = $this->lang->line('login_heading');
41
42 |        //validate form input
43 |        $this->form_validation->set_rules('identity', str_replace(':', '', $this->lang->line('login_identity_label')), 'required');
44 |        $this->form_validation->set_rules('password', str_replace(':', '', $this->lang->line('login_password_label')), 'required');
45
46 |        if ($this->form_validation->run() == true)
47 |        {
48 |            // check to see if the user is logging in
49 |            // check for "remember me"
50 |            $remember = (bool) $this->input->post('remember');
51
52 |            if ($this->ion_auth->login($this->input->post('identity'), $this->input->post('password'), $remember))
53 |            {
54 |                //if the login is successful
55 |                //redirect them back to the home page
56 |                $this->session->set_flashdata('message', $this->ion_auth->messages());
57 |                redirect('dashboard', 'refresh');
58 |            }
59 |            else
60 |            {
61 |                // if the login was un-successful
62 |                // redirect them back to the login page
63 |                $this->session->set_flashdata('message', $this->ion_auth->errors());
64 |                redirect('auth/login', 'refresh'); // use redirects instead of loading views for compatibility with MY_Controller
65 |                libraries
66 |            }
67 |        }
68 |        else
69 |        {

```

```

69 // the user is not logging in so display the login page
70 // set the flash data error message if there is one
71 $this->data['message'] = (validation_errors()) ? validation_errors() : $this->session->flashdata('message');
72
73 $this->data['identity'] = array('name' => 'identity',
74     'id' => 'identity',
75     'type' => 'text',
76     'class' => 'form-control',
77     'placeholder' => 'Username',
78     'value' => $this->form_validation->set_value('identity'),
79 );
80 $this->data['password'] = array('name' => 'password',
81     'id' => 'password',
82     'type' => 'password',
83     'class' => 'form-control',
84     'placeholder' => 'Password',
85 );
86
87 $this->_render_page('auth/login', $this->data);
88 }
89 }
90
91 // log the user out
92 public function logout()
93 {
94     $this->data['title'] = "Logout";
95
96     // log the user out
97     $logout = $this->ion_auth->logout();
98
99     // redirect them to the login page
100     $this->session->set_flashdata('message', $this->ion_auth->messages());
101     redirect('auth/login', 'refresh');
102 }
103
104 // change password
105 public function change_password()
106 {
107     $this->form_validation->set_rules('old', $this->lang->line('change_password_validation_old_password_label'), 'required');
108     $this->form_validation->set_rules('new', $this->lang->line('change_password_validation_new_password_label'), 'required|min_length[' . $this->config->item('min_password_length', 'ion_auth') . ']|max_length[' . $this->config->item('max_password_length', 'ion_auth') . ']|matches[new_confirm]');
109     $this->form_validation->set_rules('new_confirm', $this->lang->line('change_password_validation_new_password_confirm_label'), 'required');
110
111     if (!$this->ion_auth->logged_in())
112     {
113         redirect('auth/login', 'refresh');
114     }
115
116     $user = $this->ion_auth->user()->row();
117
118     if ($this->form_validation->run() == false)
119     {
120         // display the form
121         // set the flash data error message if there is one
122         $this->data['message'] = (validation_errors()) ? validation_errors() : $this->session->flashdata('message');
123
124         $this->data['min_password_length'] = $this->config->item('min_password_length', 'ion_auth');
125         $this->data['old_password'] = array(
126             'name' => 'old',
127             'id' => 'old',
128             'type' => 'password',
129         );
130         $this->data['new_password'] = array(
131             'name' => 'new',
132             'id' => 'new',
133             'type' => 'password',
134             'pattern' => '^{' . $this->data['min_password_length'] . '}.*$',
135         );
136         $this->data['new_password_confirm'] = array(
137             'name' => 'new_confirm',
138             'id' => 'new_confirm',
139             'type' => 'password',
140             'pattern' => '^{' . $this->data['min_password_length'] . '}.*$',
141         );
142         $this->data['user_id'] = array(
143             'name' => 'user_id',
144             'id' => 'user_id',
145             'type' => 'hidden',
146             'value' => $user->id,
147         );
148
149         // render
150         $this->_render_page('auth/change_password', $this->data);
151     }
152     else
153     {

```



```

154     $identity = $this->session->userdata('identity');
155
156     $change = $this->ion_auth->change_password($identity, $this->input->post('old'), $this->input->post('new'));
157
158     if ($change)
159     {
160         //if the password was successfully changed
161         $this->session->set_flashdata('message', $this->ion_auth->messages());
162         $this->logout();
163     }
164     else
165     {
166         $this->session->set_flashdata('message', $this->ion_auth->errors());
167         redirect('auth/change_password', 'refresh');
168     }
169 }
170 }
171
172 // forgot password
173 public function forgot_password()
174 {
175     // setting validation rules by checking whether identity is username or email
176     if($this->config->item('identity', 'ion_auth') != 'email' )
177     {
178         $this->form_validation->set_rules('identity', $this->lang->line('forgot_password_identity_label'), 'required');
179     }
180     else
181     {
182         $this->form_validation->set_rules('identity', $this->lang->line('forgot_password_validation_email_label'), '
            required|valid_email');
183     }
184     if ($this->form_validation->run() == false)
185     {
186         $this->data['type'] = $this->config->item('identity', 'ion_auth');
187         // setup the input
188         $this->data['identity'] = array('name' => 'identity',
189             'id' => 'identity',
190         );
191
192         if ( $this->config->item('identity', 'ion_auth') != 'email' ){
193             $this->data['identity_label'] = $this->lang->line('forgot_password_identity_label');
194         }
195         else
196         {
197             $this->data['identity_label'] = $this->lang->line('forgot_password_email_identity_label');
198         }
199
200         // set any errors and display the form
201         $this->data['message'] = (validation_errors() ? validation_errors() : $this->session->flashdata('message'));
202         $this->_render_page('auth/forgot_password', $this->data);
203     }
204     else
205     {
206         $identity_column = $this->config->item('identity', 'ion_auth');
207         $identity = $this->ion_auth->where($identity_column, $this->input->post('identity'))->users()->row();
208
209         if(empty($identity)) {
210
211             if($this->config->item('identity', 'ion_auth') != 'email')
212             {
213                 $this->ion_auth->set_error('forgot_password_identity_not_found');
214             }
215             else
216             {
217                 $this->ion_auth->set_error('forgot_password_email_not_found');
218             }
219
220             $this->session->set_flashdata('message', $this->ion_auth->errors());
221             redirect("auth/forgot_password", 'refresh');
222         }
223     }
224 }
225
226 // run the forgotten password method to email an activation code to the user
227 $forgotten = $this->ion_auth->forgotten_password($identity->{$this->config->item('identity', 'ion_auth')});
228
229 if ($forgotten)
230 {
231     // if there were no errors
232     $this->session->set_flashdata('message', $this->ion_auth->messages());
233     redirect("auth/login", 'refresh'); //we should display a confirmation page here instead of the login page
234 }
235 else
236 {
237     $this->session->set_flashdata('message', $this->ion_auth->errors());
238     redirect("auth/forgot_password", 'refresh');
239 }
240 }
241 }

```

4.2. Controller Dashboard

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Dashboard extends CI_Controller {
5
6     public function __construct()
7     {
8         parent::__construct();
9         if (!$this->ion_auth->logged_in())
10        {
11            // redirect them to the login page
12            redirect('auth/login', 'refresh');
13        }
14    }
15
16    public function index()
17    {
18        $user = $this->ion_auth->user()->row();
19        $this->breadcrumbs->push('Dashboard', '/dashboard');
20        $data = array(
21            'title' => 'Dashboard',
22            'content' => 'dashboard/dashboard',
23            'breadcrumbs' => $this->breadcrumbs->show(),
24            'user' => $user,
25        );
26
27        $this->load->view('layout/layout', $data);
28    }
29 }
30

```

4.3. Controller Kriteria

```

1 <?php
2
3 if (!defined('BASEPATH')) exit('No direct script access allowed');
4
5 class Kriteria extends CI_Controller
6 {
7     function __construct()
8     {
9         parent::__construct();
10        $this->load->model('Kriteria_model');
11        if (!$this->ion_auth->logged_in() || !$this->ion_auth->is_admin())
12        {
13            redirect('auth/login', 'refresh');
14        }
15    }
16
17    public function index()
18    {
19        $kriteria = $this->Kriteria_model->get_all();
20        $user = $this->ion_auth->user()->row();
21        $this->breadcrumbs->push('Kriteria', '/kriteria');
22
23        $data = array(
24            'title' => 'Kriteria',
25            'content' => 'kriteria/kriteria_list',
26            'breadcrumbs' => $this->breadcrumbs->show(),
27            'user' => $user,
28
29            'kriteria_data' => $kriteria
30        );
31
32        $this->load->view('layout/layout', $data);
33    }
34
35    public function read($id)
36    {
37        $user = $this->ion_auth->user()->row();
38        $this->breadcrumbs->push('Kriteria', '/kriteria');
39        $this->breadcrumbs->push('detail', '/kriteria/read');
40        $row = $this->Kriteria_model->get_by_id($id);
41        if ($row) {
42            $data = array(
43                'title' => 'Kriteria',
44                'content' => 'kriteria/kriteria_read',
45                'breadcrumbs' => $this->breadcrumbs->show(),
46                'user' => $user,
47
48                'id_kriteria' => $row->id_kriteria,
49                'nama_kriteria' => $row->nama_kriteria,
50                'bobot_kriteria' => $row->bobot_kriteria,
51                'bobot_kriteria_persen' => $row->bobot_kriteria_persen,
52            );
53            $this->load->view('layout/layout', $data);
54        } else {
55            $this->session->set_flashdata('message', 'Record Not Found');
56            redirect(site_url('kriteria'));
57        }
58    }
59 }

```

```

59
60 public function create()
61 {
62     $user = $this->ion_auth->user()->row();
63     $this->breadcrumbs->push('Kriteria', '/kriteria');
64     $this->breadcrumbs->push('tambah', '/kriteria/create');
65     $data = array(
66         'title' => 'Kriteria',
67         'content' => 'kriteria/kriteria_form',
68         'breadcrumbs' => $this->breadcrumbs->show(),
69         'user' => $user,
70
71         'button' => 'Tambah',
72         'action' => site_url('kriteria/create_action'),
73         'id_kriteria' => set_value('id_kriteria'),
74         'nama_kriteria' => set_value('nama_kriteria'),
75         'bobot_kriteria' => set_value('bobot_kriteria'),
76         'bobot_kriteria_persen' => set_value('bobot_kriteria_persen'),
77     );
78     $this->load->view('layout/layout', $data);
79 }
80
81 public function create_action()
82 {
83     $this->_rules();
84
85     if ($this->form_validation->run() == FALSE) {
86         $this->create();
87     } else {
88         $data = array(
89             'nama_kriteria' => $this->input->post('nama_kriteria', TRUE),
90             'bobot_kriteria' => $this->input->post('bobot_kriteria', TRUE),
91             'bobot_kriteria_persen' => $this->input->post('bobot_kriteria_persen', TRUE),
92         );
93
94         $this->Kriteria_model->insert($data);
95         $this->session->set_flashdata('message', 'Create Record Success');
96         redirect(site_url('kriteria'));
97     }
98 }
99
100 public function update($id)
101 {
102     $user = $this->ion_auth->user()->row();
103     $this->breadcrumbs->push('Kriteria', '/kriteria');
104     $this->breadcrumbs->push('update', '/kriteria/update');
105
106     $row = $this->Kriteria_model->get_by_id($id);
107     if ($row) {
108         $data = array(
109             'title' => 'Kriteria',
110             'content' => 'kriteria/kriteria_form',
111             'breadcrumbs' => $this->breadcrumbs->show(),
112             'user' => $user,
113
114             'button' => 'Update',
115             'action' => site_url('kriteria/update_action'),
116             'id_kriteria' => set_value('id_kriteria', $row->id_kriteria),
117             'nama_kriteria' => set_value('nama_kriteria', $row->nama_kriteria),
118             'bobot_kriteria' => set_value('bobot_kriteria', $row->bobot_kriteria),
119             'bobot_kriteria_persen' => set_value('bobot_kriteria_persen', $row->bobot_kriteria_persen),
120         );
121         $this->load->view('layout/layout', $data);
122     } else {
123         $this->session->set_flashdata('message', 'Record Not Found');
124         redirect(site_url('kriteria'));
125     }
126 }
127
128 public function update_action()
129 {
130     $this->_rules();
131
132     if ($this->form_validation->run() == FALSE) {
133         $this->update($this->input->post('id_kriteria', TRUE));
134     } else {
135         $data = array(
136             'nama_kriteria' => $this->input->post('nama_kriteria', TRUE),
137             'bobot_kriteria' => $this->input->post('bobot_kriteria', TRUE),
138             'bobot_kriteria_persen' => $this->input->post('bobot_kriteria_persen', TRUE),
139         );
140
141         $this->Kriteria_model->update($this->input->post('id_kriteria', TRUE), $data);
142         $this->session->set_flashdata('message', 'Update Record Success');
143         redirect(site_url('kriteria'));
144     }
145 }
146
147 public function delete($id)
148 {
149     $row = $this->Kriteria_model->get_by_id($id);
150
151     if ($row) {
152         $this->Kriteria_model->delete($id);
153         $this->session->set_flashdata('message', 'Delete Record Success');
154         redirect(site_url('kriteria'));
155     } else {
156         $this->session->set_flashdata('message', 'Record Not Found');
157         redirect(site_url('kriteria'));
158     }
159 }

```

4.4. Controller Analysis

```

1 |<?php
2 |defined('BASEPATH') OR exit('No direct script access allowed');
3 |
4 |class Analisis extends CI_Controller {
5 |
6 |    function __construct()
7 |    {
8 |        parent::__construct();
9 |        $this->load->model('Survey_model');
10 |        $this->load->model('Survey_detail_model');
11 |        $this->load->model('Analisis_model');
12 |        if (!$this->ion_auth->logged_in() || !$this->ion_auth->is_admin())
13 |        {
14 |            redirect('auth/login', 'refresh');
15 |        }
16 |    }
17 |
18 |    public function index()
19 |    {
20 |        $survey = $this->Survey_model->get_all();
21 |        $survey_proses = $this->Survey_model->get_by_status('proses');
22 |        $survey_selesai = $this->Survey_model->get_by_status('selesai');
23 |        $user = $this->ion_auth->user()->row();
24 |        $this->breadcrumbs->push('Analisis', '/analisis');
25 |
26 |        $data = array(
27 |            'title' => 'Analisis',
28 |            'content' => 'survey/survey_list',
29 |            'breadcrumbs' => $this->breadcrumbs->show(),
30 |            'user' => $user,
31 |            'survey_proses' => $survey_proses,
32 |            'survey_selesai' => $survey_selesai,
33 |            'survey_data' => $survey,
34 |            'mtd' => 'analisis',
35 |        );
36 |
37 |        $this->load->view('layout/layout', $data);
38 |    }
39 |
40 |    {
41 |
42 |        $id_kecamatan = $this->Survey_model->get_by_id($id_survey)->id_kecamatan;
43 |        $kecamatan = $this->Survey_model->get_nama_kec($id_kecamatan)->name;
44 |        $user = $this->ion_auth->user()->row();
45 |
46 |        $survey_detail = $this->Survey_detail_model->get_all();
47 |
48 |        $nilai_min = $this->Analisis_model->get_nilai_min($id_survey);
49 |        $nilai_max = $this->Analisis_model->get_nilai_max($id_survey);
50 |
51 |        $normalisasi = $this->Analisis_model->normalisasi($id_survey,$id_kecamatan);
52 |
53 |        $this->Analisis_model->prioritas($id_survey,$id_kecamatan);
54 |        $hasil_analisis = $this->Analisis_model->get_prioritas($id_kecamatan);
55 |
56 |        $survey = $this->Survey_model->get_by_id($id_survey);
57 |        $survey_detail_by_id = $this->Survey_detail_model->get_by_id_survey($id_survey);
58 |        $begin = new DateTime($survey->tgl_drop);
59 |        $interval = new DateInterval("P2D"); // 1 month
60 |        $occurrences = count($survey_detail_by_id);
61 |        $period = new DatePeriod($begin,$interval,$occurrences);
62 |        foreach($period as $dt){
63 |            $arr_date[] = $dt->format("Y-m-d");
64 |        }
65 |        //print_r($arr_date);
66 |
67 |        $this->breadcrumbs->push('Analisis', '/analisis');
68 |        $this->breadcrumbs->push('Hasil Kecamatan '.$kecamatan, '/survey_detail');
69 |
70 |        $kapasitas = array('7500','4500','3000','1500');
71 |        //print_r($kapasitas);
72 |
73 |        $data = array(
74 |            'title' => 'Hasil Analisis Kecamatan '.$kecamatan,
75 |            'content' => 'analisis/analisis_list',
76 |            'breadcrumbs' => $this->breadcrumbs->show(),
77 |            'user' => $user,
78 |
79 |            'survey_detail_data' => $survey_detail_by_id,
80 |            'nilai_min' => $nilai_min,
81 |            'nilai_max' => $nilai_max,
82 |
83 |            'normalisasi' => $normalisasi,
84 |            'hasil_analisis' => $hasil_analisis,
85 |            'kapasitas' => $kapasitas,
86 |            'tgl_drop' => $arr_date,
87 |        );
88 |
89 |        $this->load->view('layout/layout', $data);
90 |    }
91 |
92 | }

```

4.5. Controller Subkriteria

```

1 <?php
2
3 if (!defined('BASEPATH')) exit('No direct script access allowed');
4
5 class Subkriteria extends CI_Controller
6 {
7     function __construct()
8     {
9         parent::__construct();
10        $this->load->model('Subkriteria_model');
11        $this->load->model('Kriteria_model');
12        if (!$this->ion_auth->logged_in() || !$this->ion_auth->is_admin())
13        {
14            redirect('auth/login', 'refresh');
15        }
16    }
17
18    public function index()
19    {
20        $subkriteria = $this->Subkriteria_model->get_all();
21        $user = $this->ion_auth->user()->row();
22        $this->breadcrumbs->push('Subkriteria', '/subkriteria');
23
24        $data = array(
25            'title' => 'Subkriteria',
26            'content' => 'subkriteria/subkriteria_list',
27            'breadcrumbs' => $this->breadcrumbs->show(),
28            'user' => $user,
29            'kriteria' => '',
30            'subkriteria_data' => $subkriteria
31        );
32
33        $this->load->view('layout/layout', $data);
34    }
35
36    public function detail($id_kriteria)
37    {
38        $kriteria = $this->Kriteria_model->get_by_id($id_kriteria);
39        $subkriteria = $this->Subkriteria_model->get_by_id_kriteria($id_kriteria);
40        $user = $this->ion_auth->user()->row();
41        $this->breadcrumbs->push('Kriteria', '/kriteria');
42        $this->breadcrumbs->push($kriteria->nama_kriteria, '/subkriteria');
43
44        $data = array(
45            'title' => 'Subkriteria',
46            'content' => 'subkriteria/subkriteria_list',
47            'breadcrumbs' => $this->breadcrumbs->show(),
48            'user' => $user,
49            'kriteria' => $kriteria,
50
51            'subkriteria_data' => $subkriteria
52        );
53
54        $this->load->view('layout/layout', $data);
55    }
56
57    public function read($id)
58    {
59        $user = $this->ion_auth->user()->row();
60        $this->breadcrumbs->push('Subkriteria', '/subkriteria');
61        $this->breadcrumbs->push('detail', '/subkriteria/read');
62        $row = $this->Subkriteria_model->get_by_id($id);
63        if ($row) {
64            $kriteria = $this->Kriteria_model->get_by_id($row->id_kriteria);
65            $data = array(
66                'title' => 'Subkriteria',
67                'content' => 'subkriteria/subkriteria_read',
68                'breadcrumbs' => $this->breadcrumbs->show(),
69                'user' => $user,
70                'kriteria' => $kriteria,
71
72                'id_subkriteria' => $row->id_subkriteria,
73                'id_kriteria' => $row->id_kriteria,
74                'nama_subkriteria' => $row->nama_subkriteria,
75                'bobot_subkriteria_persen' => $row->bobot_subkriteria_persen,
76                'bobot_subkriteria' => $row->bobot_subkriteria,
77                'w' => $row->w,
78            );
79            $this->load->view('layout/layout', $data);
80        } else {
81            $this->session->set_flashdata('message', 'Record Not Found');
82            redirect(site_url('subkriteria'));
83        }
84    }
85
86 }

```

```

85 public function create($id_kriteria='')
86 {
87     $kriteria = $this->Kriteria_model->get_by_id($id_kriteria);
88     $user = $this->ion_auth->user()->row();
89     $this->breadcrumbs->push('Kriteria', '/kriteria');
90     $this->breadcrumbs->push($kriteria->nama_kriteria, '/subkriteria/detail/'.$kriteria->id_kriteria);
91     $this->breadcrumbs->push('tambah', '/subkriteria/create');
92     $data = array(
93         'title' => 'Subkriteria',
94         'content' => 'subkriteria/subkriteria_form',
95         'breadcrumbs' => $this->breadcrumbs->show(),
96         'user' => $user,
97         'kriteria' => $kriteria,
98
99         'button' => 'Tambah',
100        'action' => site_url('subkriteria/create_action/'.$id_kriteria),
101        'id_subkriteria' => set_value('id_subkriteria'),
102        'id_kriteria' => $kriteria->id_kriteria,
103        'nama_subkriteria' => set_value('nama_subkriteria'),
104        'bobot_subkriteria_persen' => set_value('bobot_subkriteria_persen'),
105        'bobot_subkriteria' => set_value('bobot_subkriteria'),
106        'w' => set_value('w'),
107    );
108    $this->load->view('layout/layout', $data);
109 }
110 public function create_action($id_kriteria='')
111 {
112     $this->_rules();
113
114     if ($this->form_validation->run() == FALSE) {
115         $this->create($id_kriteria);
116     } else {
117         $data = array(
118             'id_kriteria' => $this->input->post('id_kriteria', TRUE),
119             'nama_subkriteria' => $this->input->post('nama_subkriteria', TRUE),
120             'bobot_subkriteria_persen' => $this->input->post('bobot_subkriteria_persen', TRUE),
121             'bobot_subkriteria' => $this->input->post('bobot_subkriteria', TRUE),
122             'w' => $this->input->post('w', TRUE),
123         );
124
125         $this->Subkriteria_model->insert($data);
126         $this->session->set_flashdata('message', 'Create Record Success');
127         redirect(site_url('subkriteria/detail/'.$id_kriteria));
128     }
129 }
130 public function update($id)
131 {
132     $user = $this->ion_auth->user()->row();
133     $this->breadcrumbs->push('Subkriteria', '/subkriteria');
134     $this->breadcrumbs->push('update', '/subkriteria/update');
135
136     $row = $this->Subkriteria_model->get_by_id($id);
137     if ($row) {
138         $kriteria = $this->Kriteria_model->get_by_id($row->id_kriteria);
139         $data = array(
140             'title' => 'Subkriteria',
141             'content' => 'subkriteria/subkriteria_form',
142             'breadcrumbs' => $this->breadcrumbs->show(),
143             'user' => $user,
144             'kriteria' => $kriteria,
145
146             'button' => 'Update',
147             'action' => site_url('subkriteria/update_action'),
148             'id_subkriteria' => set_value('id_subkriteria', $row->id_subkriteria),
149             'id_kriteria' => set_value('id_kriteria', $row->id_kriteria),
150             'nama_subkriteria' => set_value('nama_subkriteria', $row->nama_subkriteria),
151             'bobot_subkriteria_persen' => set_value('bobot_subkriteria_persen', $row->bobot_subkriteria_persen),
152             'bobot_subkriteria' => set_value('bobot_subkriteria', $row->bobot_subkriteria),
153             'w' => set_value('w', $row->w),
154         );
155         $this->load->view('layout/layout', $data);
156     } else {
157         $this->session->set_flashdata('message', 'Record Not Found');
158         redirect(site_url('subkriteria/detail/'.$row->id_kriteria));
159     }
160 }
161 }
162 }

```

```

164 public function update_action()
165 {
166     $this->_rules();
167
168     if ($this->form_validation->run() == FALSE) {
169         $this->update($this->input->post('id_subkriteria', TRUE));
170     } else {
171         $data = array(
172             'id_kriteria' => $this->input->post('id_kriteria',TRUE),
173             'nama_subkriteria' => $this->input->post('nama_subkriteria',TRUE),
174             'bobot_subkriteria_persen' => $this->input->post('bobot_subkriteria_persen',TRUE),
175             'bobot_subkriteria' => $this->input->post('bobot_subkriteria',TRUE),
176             'w' => $this->input->post('w',TRUE),
177         );
178
179         $this->Subkriteria_model->update($this->input->post('id_subkriteria', TRUE), $data);
180         $this->session->set_flashdata('message', 'Update Record Success');
181         redirect(site_url('subkriteria/detail/'.$this->input->post('id_kriteria',TRUE)));
182     }
183 }
184
185 public function delete($id)
186 {
187     $row = $this->Subkriteria_model->get_by_id($id);
188
189     if ($row) {
190         $this->Subkriteria_model->delete($id);
191         $this->session->set_flashdata('message', 'Delete Record Success');
192         redirect(site_url('subkriteria'));
193     } else {
194         $this->session->set_flashdata('message', 'Record Not Found');
195         redirect(site_url('subkriteria/detail/'.$row->id_kriteria));
196     }
197 }
198
199 public function _rules()
200 {
201     $this->form_validation->set_rules('id_kriteria', 'id kriteria', 'trim|required');
202     $this->form_validation->set_rules('nama_subkriteria', 'nama subkriteria', 'trim|required');
203     $this->form_validation->set_rules('bobot_subkriteria_persen', 'bobot subkriteria persen', 'trim|required');
204     $this->form_validation->set_rules('bobot_subkriteria', 'bobot subkriteria', 'trim|required|numeric');
205     $this->form_validation->set_rules('w', 'w', 'trim|required|numeric');
206
207     $this->form_validation->set_rules('id_subkriteria', 'id_subkriteria', 'trim');
208     $this->form_validation->set_error_delimiters('<span class="text-danger">', '</span>');
209 }

```

4.6. Controller Survei

```

1 <?php
2
3 if (!defined('BASEPATH')) exit('No direct script access allowed');
4
5 class Survey extends CI_Controller
6 {
7     function __construct()
8     {
9         parent::__construct();
10        $this->load->model('Survey_model');
11        $this->load->model('Survey_detail_model');
12        if (!$this->ion_auth->logged_in())
13        {
14            redirect('auth/login', 'refresh');
15        }
16    }
17
18    public function index()
19    {
20        $survey = $this->Survey_model->get_all();
21        $survey_proses = $this->Survey_model->get_by_status('proses');
22        $user = $this->ion_auth->user()->row();
23        $this->breadcrumbs->push('Survey', '/survey');
24
25        $data = array(
26            'title' => 'Survey',
27            'breadcrumbs' => $this->breadcrumbs->show(),
28            'content' => 'survey/survey_list',
29            'user' => $user,
30            'survey_proses' => $survey_proses,
31            'survey_data' => $survey,
32            'mtd' => 'list',
33            'analisis' => 0,
34        );
35
36        $this->load->view('layout/layout', $data);
37    }

```

```

39 public function read($id)
40 {
41     $user = $this->ion_auth->user()->row();
42     $this->breadcrumbs->push('Survey', '/survey');
43     $this->breadcrumbs->push('detail', '/survey/read');
44     $row = $this->Survey_model->get_by_id($id);
45     if ($row) {
46         $data = array(
47             'title' => 'Survey',
48             'content' => 'survey/survey_read',
49             'breadcrumbs' => $this->breadcrumbs->show(),
50             'user' => $user,
51
52             'id_survey' => $row->id_survey,
53             'id_provinsi' => $row->id_provinsi,
54             'kab_kota' => $row->id_kab_kota,
55             'kecamatan' => $row->id_kecamatan,
56             'tahun' => $row->tahun,
57             'status' => $row->status,
58         );
59         $this->load->view('layout/layout', $data);
60     } else {
61         $this->session->set_flashdata('message', 'Record Not Found');
62         redirect(site_url('survey'));
63     }
64 }
65
66 public function create()
67 {
68     if (!$this->ion_auth->is_admin())
69     {
70         redirect('auth/login', 'refresh');
71     }
72
73     $provinsi = $this->Survey_model->get_provinsi();
74     $user = $this->ion_auth->user()->row();
75     $this->breadcrumbs->push('Survey', '/survey');
76     $this->breadcrumbs->push('tambah', '/survey/create');
77     $data = array(
78         'title' => 'Survey',
79         'content' => 'survey/survey_form',
80         'breadcrumbs' => $this->breadcrumbs->show(),
81         'user' => $user,
82         'provinsi' => $provinsi,
83
84         'button' => 'Tambah',
85         'action' => site_url('survey/create_action'),
86         'id_survey' => set_value('id_survey'),
87         'id_provinsi' => set_value('id_provinsi'),
88         'kab_kota' => set_value('kab_kota'),
89         'kecamatan' => set_value('kecamatan'),
90         'tahun' => set_value('tahun'),
91         'status' => set_value('status'),
92     );
93     $this->load->view('layout/layout', $data);
94 }
95
96 public function create_action()
97 {
98     if (!$this->ion_auth->is_admin())
99     {
100         redirect('auth/login', 'refresh');
101     }
102
103     $this->_rules();
104
105     if ($this->form_validation->run() == FALSE) {
106         $this->create();
107     } else {
108         $data = array(
109             'id_provinsi' => $this->input->post('id_provinsi', TRUE),
110             'id_kab_kota' => $this->input->post('kab_kota', TRUE),
111             'id_kecamatan' => $this->input->post('kecamatan', TRUE),
112             'tahun' => $this->input->post('tahun', TRUE),
113         );
114
115         $this->Survey_model->insert($data);
116         $this->session->set_flashdata('message', 'Create Record Success');
117         redirect(site_url('survey'));
118     }
119 }

```



```

121 public function update($id)
122 {
123     if (!$this->ion_auth->is_admin())
124     {
125         redirect('auth/login', 'refresh');
126     }
127
128     $provinsi = $this->Survey_model->get_provinsi();
129     $user = $this->ion_auth->user()->row();
130     $this->breadcrumbs->push('Survey', '/survey');
131     $this->breadcrumbs->push('update', '/survey/update');
132
133     $row = $this->Survey_model->get_by_id($id);
134     if ($row) {
135         $data = array(
136             'title' => 'Survey',
137             'content' => 'survey/survey_form',
138             'breadcrumbs' => $this->breadcrumbs->show(),
139             'user' => $user,
140             'provinsi' => $provinsi,
141
142             'button' => 'Update',
143             'action' => site_url('survey/update_action'),
144             'id_survey' => set_value('id_survey', $row->id_survey),
145             'id_provinsi' => set_value('id_provinsi', $row->id_provinsi),
146             'kab_kota' => set_value('kab_kota', $row->id_kab_kota),
147             'kecamatan' => set_value('kecamatan', $row->id_kecamatan),
148             'tahun' => set_value('tahun', $row->tahun),
149         );
150         $this->load->view('layout/layout', $data);
151     } else {
152         $this->session->set_flashdata('message', 'Record Not Found');
153         redirect(site_url('survey'));
154     }
155 }
156
157 public function update_action()
158 {
159     if (!$this->ion_auth->is_admin())
160     {
161         redirect('auth/login', 'refresh');
162     }
163
164     $this->_rules();
165
166     if ($this->form_validation->run() == FALSE) {
167         $this->update($this->input->post('id_survey', TRUE));
168     } else {
169         $data = array(
170             'id_provinsi' => $this->input->post('id_provinsi', TRUE),
171             'id_kab_kota' => $this->input->post('kab_kota', TRUE),
172             'id_kecamatan' => $this->input->post('kecamatan', TRUE),
173             'tahun' => $this->input->post('tahun', TRUE),
174         );
175
176         $this->Survey_model->update($this->input->post('id_survey', TRUE), $data);
177         $this->session->set_flashdata('message', 'Update Record Success');
178         redirect(site_url('survey'));
179     }
180 }
181
182 public function update_status()
183 {
184     $data = array(
185         'status' => $this->input->post('status', TRUE),
186         'tgl_drop' => $this->input->post('tgl_drop', TRUE)
187     );
188     print_r($data);
189     $this->Survey_model->update($this->input->post('id_survey', TRUE), $data);
190     $this->session->set_flashdata('message', 'Update status survey berhasil');
191     redirect(site_url('survey'));
192 }
193
194 public function delete($id)
195 {
196     if (!$this->ion_auth->is_admin())
197     {
198         redirect('auth/login', 'refresh');
199     }
200
201     $row = $this->Survey_model->get_by_id($id);
202
203     if ($row) {
204         $this->Survey_model->delete($id);
205         $this->session->set_flashdata('message', 'Delete Record Success');
206         redirect(site_url('survey'));
207     } else {
208         $this->session->set_flashdata('message', 'Record Not Found');
209         redirect(site_url('survey'));
210     }
211 }

```

```

213 public function _rules()
214 {
215     $this->form_validation->set_rules('id_provinsi', 'id provinsi', 'trim|required');
216     $this->form_validation->set_rules('kab_kota', 'kab kota', 'trim|required');
217     $this->form_validation->set_rules('kecamatan', 'kecamatan', 'trim|required');
218     $this->form_validation->set_rules('tahun', 'tahun', 'trim|required');
219
220     $this->form_validation->set_rules('id_survey', 'id survey', 'trim');
221     $this->form_validation->set_error_delimiters('<span class="text-danger">', '</span>');
222 }
223
224 public function get_kabupaten($id_prov)
225 {
226     $kab = $this->Survey_model->get_kab_by_id_prov($id_prov);
227     header('Content-Type: application/json');
228     echo json_encode($kab);
229 }
230
231 public function get_kecamatan($kab)
232 {
233     $kec = $this->Survey_model->get_kec_by_kab($kab);
234     header('Content-Type: application/json');
235     echo json_encode($kec);
236 }

```

4.7. Controller Survei Detail

```

1 <?php
2
3 if (!defined('BASEPATH')) exit('No direct script access allowed');
4
5 class Survey_detail extends CI_Controller
6 {
7     function __construct()
8     {
9         parent::__construct();
10        $this->load->model('Survey_detail_model');
11        $this->load->model('Survey_model');
12        if (!$this->ion_auth->logged_in())
13        {
14            redirect('auth/login', 'refresh');
15        }
16    }
17
18    public function index()
19    {
20
21        $survey_detail = $this->Survey_detail_model->get_all();
22        $user = $this->ion_auth->user()->row();
23        $this->breadcrumbs->push('Survey_detail', '/survey_detail');
24
25        $data = array(
26            'title' => 'Survey_detail',
27            'content' => 'survey_detail/survey_detail_list',
28            'breadcrumbs' => $this->breadcrumbs->show(),
29            'user' => $user,
30
31            'survey_detail_data' => $survey_detail
32        );
33
34        $this->load->view('layout/layout', $data);
35    }
36
37    public function read($id)
38    {
39        $user = $this->ion_auth->user()->row();
40        $this->breadcrumbs->push('Survey_detail', '/survey_detail');
41        $this->breadcrumbs->push('detail', '/survey_detail/read');
42        $row = $this->Survey_detail_model->get_by_id($id);
43        if ($row) {
44            $data = array(
45                'title' => 'Survey_detail',
46                'content' => 'survey_detail/survey_detail_read',
47                'breadcrumbs' => $this->breadcrumbs->show(),
48                'user' => $user,
49
50                'id_survey_detail' => $row->id_survey_detail,
51                'id_survey' => $row->id_survey,
52                'id_user' => $row->id_user,
53                'nama_desa' => $row->nama_desa,
54                'lokasi_kekeringan' => $row->lokasi_kekeringan,
55                'k1' => $row->k1,
56                'k2' => $row->k2,
57                'k3' => $row->k3,
58                'k4' => $row->k4,
59                'k5' => $row->k5,
60                'k6' => $row->k6,
61            );
62            $this->load->view('layout/layout', $data);
63        } else {
64            $this->session->set_flashdata('message', 'Record Not Found');
65            redirect(site_url('survey_detail'));
66        }
67    }
68 }

```

```

69 public function create($id_survey='')
70 {
71     $survey = $this->Survey_model->get_by_id($id_survey);
72     $user = $this->ion_auth->user()->row();
73     $kab_kota = $this->Survey_model->get_nama_kab($survey->id_kab_kota)->name;
74     $kecamatan = $this->Survey_model->get_nama_kec($survey->id_kecamatan)->name;
75     $desa = $this->Survey_model->get_desa_by_kec_id_survey($survey->id_kecamatan,$id_survey);
76
77     if ($this->ion_auth->is_admin()) {
78         $this->breadcrumbs->push('Survey detail', '/survey_detail');
79         $this->breadcrumbs->push('tambah', '/survey_detail/create');
80     } else {
81         $this->breadcrumbs->push('Survey', '/survey');
82         $this->breadcrumbs->push($kab_kota, ' '.$kecamatan, '/survey_detail/create/'.$id_survey);
83         $this->breadcrumbs->push('Form Data Survey', '/survey_detail/create');
84     }
85
86     $data = array(
87         'title' => 'Survey detail',
88         'content' => 'survey_detail/survey_detail_form',
89         'breadcrumbs' => $this->breadcrumbs->show(),
90         'user' => $user,
91         'survey' => $survey,
92         'desa' => $desa,
93
94         'button' => 'Tambah',
95         'action' => site_url('survey_detail/create_action/'.$id_survey),
96         'id_survey_detail' => set_value('id_survey_detail'),
97         'id_survey' => $id_survey,
98         'nama_desa' => set_value('nama_desa'),
99         'lokasi_kekeringan' => set_value('lokasi_kekeringan'),
100         'k1' => set_value('k1'),
101         'k2' => set_value('k2'),
102         'k3' => set_value('k3'),
103         'k4' => set_value('k4'),
104         'k5' => set_value('k5'),
105         'k6' => set_value('k6'),
106     );
107     $this->load->view('layout/layout', $data);
108 }
109 public function create_action($id_survey='')
110 {
111     $this->_rules();
112
113     if ($this->form_validation->run() == FALSE) {
114         $this->create($id_survey);
115     } else {
116         $user = $this->ion_auth->user()->row();
117         $data = array(
118             'id_survey' => $this->input->post('id_survey',TRUE),
119             'id_user' => $user->id,
120             'nama_desa' => $this->input->post('nama_desa',TRUE),
121             'lokasi_kekeringan' => $this->input->post('rt',TRUE)." ".$this->input->post('rw',TRUE),
122             'k1' => $this->input->post('k1',TRUE),
123             'k2' => $this->input->post('k2',TRUE),
124             'k3' => $this->input->post('k3',TRUE),
125             'k4' => $this->input->post('k4',TRUE),
126             'k5' => $this->input->post('k5',TRUE),
127             'k6' => $this->input->post('k6',TRUE),
128         );
129
130         $this->Survey_detail_model->insert($data);
131         $this->session->set_flashdata('message', 'Create Record Success');
132         redirect(site_url('survey'));
133     }
134 }
135
136 public function update($id)
137 {
138     $user = $this->ion_auth->user()->row();
139     $this->breadcrumbs->push('Survey detail', '/survey_detail');
140     $this->breadcrumbs->push('update', '/survey_detail/update');
141
142     $row = $this->Survey_detail_model->get_by_id($id);
143     if ($row) {
144         $data = array(
145             'title' => 'Survey detail',
146             'content' => 'survey_detail/survey_detail_form',
147             'breadcrumbs' => $this->breadcrumbs->show(),
148             'user' => $user,
149
150             'button' => 'Update',
151             'action' => site_url('survey_detail/update_action'),
152             'id_survey_detail' => set_value('id_survey_detail', $row->id_survey_detail),
153             'id_survey' => set_value('id_survey', $row->id_survey),
154             'id_user' => set_value('id_user', $row->id_user),
155             'nama_desa' => set_value('nama_desa', $row->nama_desa),
156             'lokasi_kekeringan' => set_value('lokasi_kekeringan', $row->lokasi_kekeringan),
157             'k1' => set_value('k1', $row->k1),
158             'k2' => set_value('k2', $row->k2),
159             'k3' => set_value('k3', $row->k3),
160             'k4' => set_value('k4', $row->k4),
161             'k5' => set_value('k5', $row->k5),
162             'k6' => set_value('k6', $row->k6),
163         );
164     }

```

```

165     $this->load->view('layout/layout', $data);
166 } else {
167     $this->session->set_flashdata('message', 'Record Not Found');
168     redirect(site_url('survey_detail'));
169 }
170 }
171
172 public function update_action()
173 {
174     $this->_rules();
175
176     if ($this->form_validation->run() == FALSE) {
177         $this->update($this->input->post('id_survey_detail', TRUE));
178     } else {
179         $data = array(
180             'id_survey' => $this->input->post('id_survey', TRUE),
181             'id_user' => $this->input->post('id_user', TRUE),
182             'nama_desa' => $this->input->post('nama_desa', TRUE),
183             'lokasi_kekeringan' => $this->input->post('lokasi_kekeringan', TRUE),
184             'k1' => $this->input->post('k1', TRUE),
185             'k2' => $this->input->post('k2', TRUE),
186             'k3' => $this->input->post('k3', TRUE),
187             'k4' => $this->input->post('k4', TRUE),
188             'k5' => $this->input->post('k5', TRUE),
189             'k6' => $this->input->post('k6', TRUE),
190         );
191
192         $this->Survey_detail_model->update($this->input->post('id_survey_detail', TRUE), $data);
193         $this->session->set_flashdata('message', 'Update Record Success');
194         redirect(site_url('survey_detail'));
195     }
196 }
197
198 public function delete($id)
199 {
200     $row = $this->Survey_detail_model->get_by_id($id);
201
202     if ($row) {
203         $this->Survey_detail_model->delete($id);
204         $this->session->set_flashdata('message', 'Delete Record Success');
205         redirect(site_url('survey_detail'));
206     } else {
207         $this->session->set_flashdata('message', 'Record Not Found');
208         redirect(site_url('survey_detail'));
209     }
210 }
211
212 public function _rules()
213 {
214     $this->form_validation->set_rules('id_survey', 'id survey', 'trim|required');
215     $this->form_validation->set_rules('nama_desa', 'nama desa', 'trim|required');
216     //$this->form_validation->set_rules('lokasi_kekeringan', 'lokasi kekeringan', 'trim|required');
217     $this->form_validation->set_rules('k1', 'k1', 'trim|required');
218     $this->form_validation->set_rules('k2', 'k2', 'trim|required');
219     $this->form_validation->set_rules('k3', 'k3', 'trim|required');
220     $this->form_validation->set_rules('k4', 'k4', 'trim|required');
221     $this->form_validation->set_rules('k5', 'k5', 'trim|required');
222     $this->form_validation->set_rules('k6', 'k6', 'trim|required');
223
224     $this->form_validation->set_rules('id_survey_detail', 'id_survey_detail', 'trim');
225     $this->form_validation->set_error_delimiters('<span class="text-danger">', '</span>');
226 }

```

4.8. Controller Users

```

1 <?php
2
3 if (!defined('BASEPATH')) exit('No direct script access allowed');
4
5 class Users extends CI_Controller
6 {
7     function __construct()
8     {
9         parent::__construct();
10        $this->load->model('Users_model');
11        $this->load->model('Groups_model');
12        if (!$this->ion_auth->logged_in() || !$this->ion_auth->is_admin())
13        {
14            redirect('auth/login', 'refresh');
15        }
16    }
17
18    public function index()
19    {
20        $users = $this->ion_auth->users()->result();
21        $user = $this->ion_auth->user()->row();
22        $this->breadcrumbs->push('Users', '/users');
23    }

```

```

24     $data = array(
25         'title'     => 'Users' ,
26         'content'  => 'users/users_list',
27         'breadcrumbs' => $this->breadcrumbs->show(),
28         'user'     => $user ,
29
30         'users_data' => $users
31     );
32
33     $this->load->view('layout/layout', $data);
34 }
35 public function read($id)
36 {
37     $user = $this->ion_auth->user()->row();
38     $this->breadcrumbs->push('Users', '/users');
39     $this->breadcrumbs->push('detail', '/users/read');
40     $row = $this->Users_model->get_by_id($id);
41     if ($row) {
42         $data = array(
43             'title'     => 'Users' ,
44             'content'  => 'users/users_read',
45             'breadcrumbs' => $this->breadcrumbs->show(),
46             'user'     => $user ,
47
48             'id' => $row->id,
49             'name' => $row->name,
50             'email' => $row->email,
51             'username' => $row->username,
52             'password' => $row->password,
53             'phone' => $row->phone,
54             'alamat' => $row->alamat,
55             'user_img' => $row->user_img,
56             'ip_address' => $row->ip_address,
57             'last_login' => $row->last_login,
58             'salt' => $row->salt,
59             'activation_code' => $row->activation_code,
60             'forgotten_password_code' => $row->forgotten_password_code,
61             'forgotten_password_time' => $row->forgotten_password_time,
62             'remember_code' => $row->remember_code,
63             'active' => $row->active,
64             'created_on' => $row->created_on,
65         );
66         $this->load->view('layout/layout', $data);
67     } else {
68         $this->session->set_flashdata('message', 'Record Not Found');
69         redirect(site_url('users'));
70     }
71 }
72
73 public function create()
74 {
75     $grup = $this->Groups_model->get_all();
76     $user = $this->ion_auth->user()->row();
77     $this->breadcrumbs->push('Users', '/users');
78     $this->breadcrumbs->push('tambah', '/users/create');
79     $data = array(
80         'title'     => 'Users' ,
81         'content'  => 'users/users_form',
82         'breadcrumbs' => $this->breadcrumbs->show(),
83         'user'     => $user ,
84         'grup'    => $grup ,
85
86         'groupss' => set_value('groupss') ,
87         'button' => 'Tambah',
88         'action' => site_url('users/create_action'),
89         'id' => set_value('id'),
90         'name' => set_value('name'),
91         'email' => set_value('email'),
92         'username' => set_value('username'),
93         'password' => set_value('password'),
94         'phone' => set_value('phone'),
95         'alamat' => set_value('alamat'),
96         'user_img' => set_value('user_img'),
97     );
98     $this->load->view('layout/layout', $data);
99 }
100
101 public function create_action()
102 {
103     $this->form_validation->set_rules('name', 'name', 'trim|required');
104     $this->form_validation->set_rules('email', 'email', 'trim|required|valid_emails|valid_email|is_unique[users.email]');
105     $this->form_validation->set_rules('username', 'username', 'trim|required|is_unique[users.username]');
106     $this->form_validation->set_rules('password', 'password', 'trim|required');
107     $this->form_validation->set_rules('password2', 'Konfirmasi Password', 'trim|matches[password]');
108     $this->form_validation->set_rules('phone', 'phone', 'trim|required');
109     $this->form_validation->set_rules('alamat', 'alamat', 'trim|required');
110     $this->form_validation->set_rules('user_img', 'User Image', 'callback_image_upload');
111     $this->form_validation->set_rules('group_id', 'Level User', 'trim|required');
112     $this->form_validation->set_rules('id', 'id', 'trim');
113     $this->form_validation->set_error_delimiters('<span class="text-danger">', '</span>');
114 }
115

```

```

116     if ($this->form_validation->run() == FALSE) {
117         $this->create();
118     } else {
119         $CI =& get_instance();
120         $upload_data = $CI->upload->data();
121         $img = $upload_data['file_name'];
122         $data = array(
123             'nama' => $this->input->post('name',TRUE),
124             'phone' => $this->input->post('phone',TRUE),
125             'alamat' => $this->input->post('alamat',TRUE),
126             'user_img' => $img,
127         );
128
129         $username = $this->input->post('username',TRUE);
130         $password = $this->input->post('password',TRUE);
131         $email = $this->input->post('email',TRUE);
132         $group = array($this->input->post('group_id'));
133
134         $this->ion_auth->register($username, $password, $email, $data, $group);
135
136         $this->session->set_flashdata('message', 'Create Record Success');
137         redirect(base_url('users'));
138     }
139 }
140
141 public function update($id)
142 {
143     $group = $this->Groups_model->get_all();
144     $user = $this->ion_auth->user()->row();
145     $this->breadcrumbs->push('Users', '/users');
146     $this->breadcrumbs->push('update', '/users/update');
147
148     $row = $this->Users_model->get_by_id($id);
149     if ($row) {
150         $data = array(
151             'title' => 'Users',
152             'content' => 'users/users_form',
153             'breadcrumbs' => $this->breadcrumbs->show(),
154             'user' => $user,
155             'grup' => $group,
156
157             'button' => 'Update',
158             'action' => site_url('users/update_action'),
159             'id' => set_value('id', $row->id),
160             'name' => set_value('name', $row->name),
161             'email' => set_value('email', $row->email),
162             'username' => set_value('username', $row->username),
163             'password' => set_value('password', $row->password),
164             'phone' => set_value('phone', $row->phone),
165             'alamat' => set_value('alamat', $row->alamat),
166             'groupsss' => set_value('groupsss', $row->name),
167             'user_img' => set_value('user_img', $row->user_img),
168         );
169         $this->load->view('layout/layout', $data);
170     } else {
171         $this->session->set_flashdata('message', 'Record Not Found');
172         redirect(site_url('users'));
173     }
174 }
175
176 public function update_action()
177 {
178     $this->form_validation->set_rules('name', 'name', 'trim|required');
179     $this->form_validation->set_rules('email', 'email', 'trim|required');
180     $this->form_validation->set_rules('username', 'username', 'trim|required');
181     $this->form_validation->set_rules('phone', 'phone', 'trim|required');
182     $this->form_validation->set_rules('alamat', 'alamat', 'trim|required');
183     $this->form_validation->set_error_delimiters('<span class="text-danger">', '</span>');
184
185     if ($this->form_validation->run() == FALSE) {
186         $this->update($this->input->post('id', TRUE));
187     } else {
188         $data = array(
189             'nama' => $this->input->post('name',TRUE),
190             'email' => $this->input->post('email',TRUE),
191             'username' => $this->input->post('username',TRUE),
192             'phone' => $this->input->post('phone',TRUE),
193             'alamat' => $this->input->post('alamat',TRUE),
194         );
195
196         $this->Users_model->update($this->input->post('id', TRUE), $data);
197         $this->session->set_flashdata('message', 'Update Record Success');
198         redirect(site_url('users'));
199     }
200 }
201
202 public function delete($id)
203 {
204     $row = $this->Users_model->get_by_id($id);
205
206     if ($row) {
207         $this->Users_model->delete($id);
208         $this->session->set_flashdata('message', 'Delete Record Success');
209         redirect(site_url('users'));
210     } else {
211         $this->session->set_flashdata('message', 'Record Not Found');
212         redirect(site_url('users'));
213     }
214 }

```

```

216 public function ubah_foto($id)
217 {
218     $this->form_validation->set_rules('user_img', 'User Image', 'callback_image_upload');
219     if ($this->form_validation->run() == FALSE) {
220         $this->update($id);
221     } else {
222         $CI =& get_instance();
223         $upload_data = $CI->upload->data();
224         $gambar = $upload_data['file_name'];
225         $data = array(
226             'user_img' => $gambar,
227         );
228         $row = $this->Users_model->get_by_id($id);
229         unlink('./images/users/'.$row->user_img);
230         $this->Users_model->update($id,$data);
231         if ($this->ion_auth->is_admin()) {
232             redirect(site_url('users/update/'.$id));
233         } else {
234             redirect('home/update_akun/'.$id,'refresh');
235         }
236     }
237 }
238
239 }
240 public function ubah_password($id)
241 {
242     $password = $this->input->post('password',TRUE);
243     $this->form_validation->set_rules('password', 'password', 'trim|required');
244     if ($this->form_validation->run() == FALSE) {
245         $this->update($id);
246     } else {
247         $data = array(
248             'password' => $password,
249         );
250         $this->ion_auth->update($id,$data);
251         $this->session->set_flashdata('message', 'Password Berhasil dirubah');
252         if ($this->ion_auth->is_admin()) {
253             redirect(site_url('users/update/'.$id));
254         } else {
255             redirect('home/update_akun/'.$id,'refresh');
256         }
257     }
258 }
259
260 public function image_upload(){
261     if($_FILES['user_img']['size'] != 0){
262         $upload_dir = './images/users/';
263         if (!is_dir($upload_dir)) {
264             mkdir($upload_dir);
265         }
266         $config['upload_path'] = $upload_dir;
267         $config['allowed_types'] = 'gif|jpg|png|jpeg';
268         $config['file_name'] = 'usr_img_'.substr(md5(rand()),0,7);
269         $config['overwrite'] = true;
270         $config['max_size'] = '51200';
271
272         $this->load->library('upload', $config);
273         if (!$this->upload->do_upload('user_img')){
274             $this->form_validation->set_message('image_upload', $this->upload->display_errors());
275             return false;
276         }
277         else{
278             $this->upload_data['file'] = $this->upload->data();
279             return true;
280         }
281     }
282     else{
283         $this->form_validation->set_message('image_upload', "No file selected");
284         return false;
285     }
286 }

```

5. Lampiran Hasil Pengujian *Black Box*

5.1. Hasil *Black Box Testing*

No.	Menu	Fungsi	Kasus	Hasil	Ket.
1.	Login	Menu ini berfungsi sebagai keamanan sistem. memilah hak akses <i>user</i> yang dapat menggunakan sistem ini.	Ketika <i>user</i> memasukan <i>username</i> dan password dengan benar	Menampilkan halaman sesuai level <i>user</i>	Berhasil
2	Dashboard	Menampilkan halaman dashboard	Ketika <i>user</i> telah melakukan login, sistem akan menampilkan halaman dashboard	Menampilkan halaman dashboard	Berhasil
3.	Kelola <i>User</i>	Untuk melihat data <i>user</i> , menambah <i>user</i> , ubah data <i>user</i> , dan menghapus data <i>user</i>	Ketika <i>admin</i> klik menu kelola <i>user</i> menampilkan halaman kelola data <i>user</i>	Menampilkan halaman kelola data <i>user</i>	Berhasil
			Ketika <i>admin</i> klik tombol " <i>Tambah</i> "	Menampilkan form tambah data <i>user</i>	Berhasil
			Ketika <i>admin</i> klik tombol " <i>Delete</i> "	Menghapus data <i>user</i>	Berhasil

			Ketika <i>admin</i> klik tombol “ <i>Update</i> ”	Menampilkan form halaman <i>update</i> data <i>user</i>	Berhasil
			Ketika <i>admin</i> klik tombol “ <i>Detail</i> ”	Menampilkan halaman <i>detail</i> data <i>user</i>	Berhasil
4.	Analisis Prioritas	Melakukan analisis prioritas distribusi air bersih	Ketika <i>user</i> klik menu Analisis Prioritas	Menampilkan halaman form analisis	Berhasil
			Ketika <i>user</i> klik tombol “ <i>Analisis</i> ”	Menampilkan halaman hasil analisis	Berhasil
5.	Kelola Data Survei	Untuk melihat data survei, menambah data survei, dan menghapus data survei	Ketika <i>admin</i> klik tombol “ <i>Tambah</i> ”	Menampilkan halaman form tambah daerah survei	Berhasil
			Ketika <i>surveyor</i> klik tombol “ <i>Mulai Survei</i> ”	Menampilkan halaman form daerah yang akan di survei	Berhasil
			Ketika <i>admin</i> klik tombol “ <i>Delete</i> ”	Menampilkan pop up untuk menghapus daerah survei	Berhasil
			Ketika <i>admin</i> klik tombol “ <i>Selesai</i> ”	Menampilkan pop up untuk penentuan status daerah survei dan tanggal dropping air bersih	Berhasil

6.	Kelola Data Kriteria	Untuk melihat data kriteria, menambah data kriteria, ubah data kriteria, dan menghapus data kriteria	Ketika <i>admin</i> klik menu Kriteria menampilkan halaman kelola data kriteria	Menampilkan halaman kelola data kriteria	Berhasil
			Ketika <i>admin</i> klik tombol “ <i>Tambah</i> ”	Menampilkan form tambah data kriteria	Berhasil
			Ketika <i>admin</i> klik tombol “ <i>Delete</i> ”	Menghapus data kriteria	Berhasil
			Ketika <i>admin</i> klik tombol “ <i>Update</i> ”	Menampilkan form halaman <i>update</i> data kriteria	Berhasil
			Ketika <i>admin</i> klik tombol “ <i>Detail</i> ”	Menampilkan halaman <i>detail</i> data kriteria	Berhasil
7.	Kelola Subkriteria	Untuk melihat data subkriteria, menambah data subkriteria, ubah data subkriteria, dan menghapus data subkriteria	Ketika <i>admin</i> klik menu Subkriteria menampilkan halaman kelola data subkriteria	Menampilkan halaman kelola data subkriteria	Berhasil
			Ketika <i>admin</i> klik tombol “ <i>Detail</i> ”	Menampilkan halaman <i>detail</i> data subkriteria	Berhasil
			Ketika <i>admin</i> klik tombol	Menampilkan form tambah	Berhasil

			<i>"Tambah"</i>	data subkriteria	
			Ketika <i>admin</i> klik tombol <i>"Delete"</i>	Menghapus data subkriteria	Berhasil
			Ketika <i>admin</i> klik tombol <i>"Update"</i>	Menampilkan form halaman <i>update</i> data subkriteria	Berhasil