**TUGAS AKHIR**

# IMPLEMENTASI METODE SIMPLE ADDITIVE WEIGHTING (SAW) DALAM SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA

**(Studi Kasus : BPBD Lombok Barat)**

**Oleh :**

**AHMAD MAULANA ARI SEPTIAN**
**F1D 015 004**

**PROGRAM STUDI TEKNIK INFORMATIKA**
**FAKULTAS TEKNIK**
**UNIVERSITAS MATARAM**
**2020**

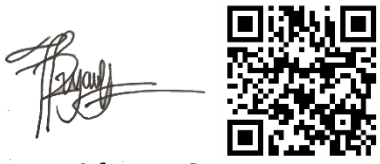
**TUGAS AKHIR**

**IMPLEMENTASI METODE SIMPLE ADDITIVE WEIGHTING (SAW) DALAM SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA**
**(Studi Kasus : BPBD Lombok Barat)**

**Oleh**
**AHMAD MAULANA ARI SEPTIAN**
**F1D 015 004**

Telah diperiksa dan disetujui oleh Tim Pembimbing:

1. Pembimbing Utama

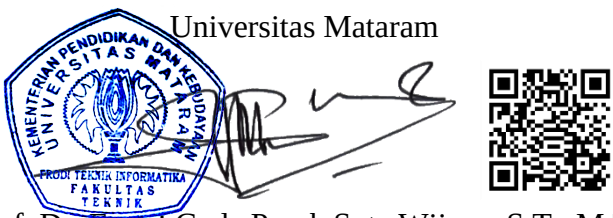


Royana Afwani. S.T., MT. Tanggal : 4 / Juni / 2020
NIP. 19850707 201404 2 001

2. Pembimbing Pendamping

Moh. Ali Albar, S.T., M.Eng. Tanggal : 4 / Juni / 2020
NIP. 19831125 201504 1 002

Mengetahui
Ketua Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram




Prof. Dr. Eng. I Gede Pasek Suta Wijaya, S.T., M.T.
NIP. 19731130 200003 1 001

**TUGAS AKHIR**

**IMPLEMENTASI METODE SIMPLE ADDITIVE WEIGHTING (SAW) DALAM SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA**
**(Studi Kasus : BPBD Lombok Barat)**

**Oleh**
**AHMAD MAULANA ARI SEPTIAN**
**F1D 015 004**

Telah dipertahankan di depan Dewan Penguji
Pada tanggal 1 Juni 2020
dan dinyatakan telah memenuhi syarat mencapai derajat Sarjana S-1
Program Studi Teknik Informatika

Susunan Tim Penguji:

1. Penguji I

Ariyan Zubaidi, S.Kom., M.T. Tanggal: 4/05/2020

NIP. 19860913 201504 1 001

2. Penguji II

Nadiyasari Agitha, S.Kom., M.MT. Tanggal: 4/05/2020
NIP. 19860813 201803 2 001

3. Penguji III

Fitri Bimantoro, S.T., M.Kom. Tanggal: 4/05/2020
NIP. 19860622 201504 1 002

Mataram, 6 Juni 2020
Dekan Fakultas Teknik
Universitas Mataram

Akmaluddin, S.T., M.Sc.(Eng.), Ph.D.
NIP. 19681231 199412 1 001

## PERNYATAAN KEASLIAN TUGAS AKHIR

Saya yang bertanda tangan di bawah ini bahwa dalam Skripsi ini tidak terdapat karya yang pernah di ajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Mataram, Juni 2020

Ahmad Maulana Ari Septian

# KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Bismillahirrahmaanirrahiim segala puji penulis panjatkan kepada Allah Subhaanahu Wa Ta'ala atas berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan penyusunan Tugas Akhir ini. Tanpa belas kasih dan pertolongan-Nya penulis tidak akan mampu untuk menyeleasikan Tugas Akhir ini dengan baik. Tidak lupa shalawat beserta salam selalu tercurahkan kepada baginda Nabi Muhammad Shallallahu 'Alaihi Wasallam sebagai suri tauladan manusia. Segala rasa syukur penulis haturkan kepara Allah atas nikmat sehat dan sempat-Nya , sehingga penulisa dapat mennyelesaikan Tugas Akhir dengan judul "Implementasi Metode *Simple Additive Weighting* (Saw) Dalam Sistem Pendukung Keputusan Penentuan Penerima Bantuan Korban Bencana Alam Gempa (Studi Kasus: BPBD Lombok Barat)".

Dalam penulisan Tugas Akhir ini penulis tentu menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna dan masih terdapat kesalahan dan kekurangan didalamnya. Oleh karena itu, penulis harapkan kritik serta saran dari pembaca untuk Tugas Akhir ini, agar kedepannya Tugas Akhir ini menjadi lebih baik lagi. Penulis juga mengucapkan banyak terimakasih kepada semua pihak yang telah membantu penulis sehingga Tugas Akhir ini dapat selesai.

Akhir kata penulis berharap Tugas Akhir ini bisa bermanfaat bagi pembaca. Terimakasih.

Mataram,  Juni 2020

Ahmad Maulana Ari Septian

## UCAPAN TERIMAKASIH

Tugas Akhir ini dapat diselesaikan berkat bimbingan dan dukungan ilmiah maupun materil dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terimakasih yang setulus-tulusnya kepada:

1. Kedua Orang Tua Penulis, selaku pemberi dukungan utama yang selalu memberikan do'a dan dukungan baik moril maupun materil yang tidak putus-putus kepada penulis sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir ini dengan baik.
2. Keluarga penulis yang telah memberikan dukungan dan doanya selama penulis menjalani perkuliahan sampai dengan pengerjaan Tugas Akhir ini dapat diselesaikan dengan baik.
3. Ibu Royana Afwani. S.T., MT. selaku dosen pembimbing utama yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir ini sehingga dapat penulis selesaikan dengan baik.
4. Bapak Moh. Ali Albar, S.T., M.Eng. selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir ini sehingga dapat penulis selesaikan dengan baik.
5. Bapak dan Ibu Dosen penguji, atas pemberian kritik dan saran yang bersifat membangun sangat diperlukan dalam penyelesaian Tugas Akhir ini.
6. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan do'a dan dukungan baik moril maupun materil sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir dengan baik.

Semoga Allah SWT selalu memberikan rahmat dan hidayah-Nya dan memberikan imbalan yang setimpal atas bantuan yang diberikan kepada penulis**.** Aamiin.

# DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR ..... iv
KATA PENGANTAR ..... v
UCAPAN TERIMAKASIH ..... vi
DAFTAR ISI ..... vii
DAFTAR GAMBAR ..... ix
DAFTAR TABEL ..... xi
ABSTRAK ..... xiii
BAB I ..... 1
1.1. Latar Belakang ..... 1
1.2. Rumusan Masalah ..... 3
1.3. Batasan Masalah ..... 3
1.4. Tujuan ..... 3
1.5. Manfaat ..... 4
1.6. Sistematika Penulisan ..... 4
BAB II ..... 6
2.1. Tinjauan Pustaka ..... 6
2.2. Dasar Teori ..... 8
2.2.1. Sistem Informasi ..... 8
2.2.2. Sistem Pendukung Keputusan ..... 9
2.2.3. *Website* ..... 9
2.2.4. Pedoman Penerimaan Bantuan Korban Bencana Alam Gempa Lombok ..... 10
2.2.5. UML ..... 13
2.2.6. *Framework CodeIgniter* ..... 18
2.2.7. ERD ..... 19
2.2.8. MySQL ..... 20
2.2.9. *Fuzzy Multiple Atribute Decission Making* (FMADM) ..... 20
2.2.10. *Simple Additive Weighting* (SAW) ..... 21
BAB III ..... 23
3.1. Alat dan Bahan ..... 23
3.2. Alur Penelitian ..... 24
3.3. Studi Literatur ..... 24
3.4. Analisis Sistem ..... 25
3.4.1. Analisis Sistem Yang Berjalan ..... 25
3.4.2. Analisis Sistem Yang Akan Dibangun ..... 27

3.4.3. Perhitungan manual yang sudah dilakukan ..... 29
3.5. Metode Simple Additive Weighting (SAW) ..... 32
3.6. Perancangan Sistem ..... 39
3.5.1. *Use Case Diagram* ..... 39
3.5.2. *Class Diagram* ..... 40
3.5.3. *Sequence Diagram* ..... 41
3.5.4. *Activity Diagram* ..... 44
3.5.5. *Entity Relationship Diagram* ..... 48
3.5.6. Desain Tampilan Sistem ..... 50
3.7. Pembuatan Sistem ..... 53
3.8. Pengujian Sistem ..... 53
3.7.1. Metode *White Box* ..... 53
3.7.2. Metode *Black Box* ..... 53
BAB IV ..... 55
4.1 Implementasi Sistem ..... 55
4.1.1 Implementasi *Database* ..... 55
4.1.2 Implementasi *Class* ..... 57
4.1.3 Implementasi *Interface* ..... 61
4.2 Pengujian Sistem ..... 71
4.2.1 Pengujian *White Box* ..... 71
4.2.2 Pengujian *Black Box* ..... 73
4.2.3 Pengujian Perbandingan Perhitungan SAW Manual dengan SAW pada Sistem 78
BAB V ..... 80
5.1 Kesimpulan ..... 80
5.2 Saran ..... 80
DAFTAR PUSTAKA ..... 82

# DAFTAR GAMBAR

Gambar 2.1 Fishbone Diagram ..... 8
Gambar 2.2 Arsitektur MVC ..... 18
Gambar 3.1 Diagram alir penelitian ..... 24
Gambar 3.2 Flowmap sistem yang sedang berjalan ..... 26
Gambar 3.3 Flowmap sistem yang akan dibangun ..... 28
Gambar 3.4 Use case diagram ..... 40
Gambar 3.5 Class diagram ..... 41
Gambar 3.6 Sequence diagram admin *login* ..... 42
Gambar 3.7 Sequence diagram tim survei *login* ..... 42
Gambar 3.8 Sequence diagram memasukkan data korban ..... 43
Gambar 3.9 Sequence diagram admin mengedit data korban ..... 43
Gambar 3.10 Sequence diagram admin menghapus data korban ..... 44
Gambar 3.11 Activity diagram *login* admin ..... 45
Gambar 3.12 Activity diagram *login* tim survei ..... 46
Gambar 3.13 Activity diagram memasukkan data korban ..... 46
Gambar 3.14 Activity diagram edit data korban ..... 47
Gambar 3.15 Activity diagram menghapus data korban ..... 48
Gambar 3.16 Entity relationship diagram ..... 49
Gambar 3.17 Desain halaman *login* ..... 53
Gambar 3.18 Desain halaman home ..... 53
Gambar 3.19 Desain halaman input data ..... 54
Gambar 3.20 Desain halaman data kriteria ..... 54
Gambar 3.21 Desain halaman hasil SAW ..... 55
Gambar 4.1 Implementasi *database.* ..... 55
Gambar 4.2 Implementasi tabel *users.* ..... 55
Gambar 4.3 Implementasi tabel korban. ..... 56
Gambar 4.4 Implementasi tabel kriteria. ..... 56
Gambar 4.5 Implementasi tabel detail kriteria ..... 57
Gambar 4.6 Implementasi *class controllers.* ..... 57
Gambar 4.7 Contoh *class controller login.* ..... 58
Gambar 4.8 Implementasi metode SAW. ..... 59
Gambar 4.9 Implementasi *class models.* ..... 59

Gambar 4.10 Contoh *class models user*..... 60
Gambar 4.11 Implementasi *class views.*..... 60
Gambar 4.12 Contoh *class views form login*..... 61
Gambar 4.13 Implementasi halaman awal..... 62
Gambar 4.14 Implementasi halaman *login.*..... 62
Gambar 4.15 Implementasi halaman *dashboard*..... 63
Gambar 4.16 Implementasi halaman data korban. ..... 63
Gambar 4.17 Implementasi tampilan tambah data korban. ..... 64
Gambar 4.18 Implementasi halaman edit data korban. ..... 64
Gambar 4.19 Implementasi halaman data kriteria. ..... 65
Gambar 4.20 Implementasi tampilan tambah data kriteria. ..... 65
Gambar 4.21 Implementasi halaman edit data kriteria. ..... 66
Gambar 4.22 Implementasi halaman detail data kriteria. ..... 66
Gambar 4.23 Implementasi halaman tambah data detail kriteria. ..... 67
Gambar 4.24 Implementasi halaman edit data detail kriteria. ..... 67
Gambar 4.25 Implementasi halaman data *user*..... 68
Gambar 4.26 Implementasi tampilan tambah *user*..... 68
Gambar 4.27 Implementasi halaman edit data *user*..... 69
Gambar 4.28 Implementasi halaman langkah SAW..... 69
Gambar 4.29 Implementasi halaman *profile*..... 70
Gambar 4.30 *Basis Path* Algoritma SAW..... 71

# DAFTAR TABEL

Tabel 2.1 Panduan tingkat kerusakan aset ..... 11
Tabel 2.2 Panduan bobot kerusakan rumah/gedung ..... 11
Tabel 2.3 Simbol UseCase Diagram ..... 14
Tabel 2.4 Simbol-simbol activity digram ..... 15
Tabel 2.5 Simbol class diagram ..... 16
Tabel 2.6 Simbol-simbol sequence diagram ..... 17
Tabel 2.7 Simbol ERD ..... 19
Tabel 3.1 Pembobotan Kriteria ..... 29
Tabel 3.2 Hasil perhitungan ..... 31
Tabel 3.3 Penentuan bobot setiap kriteria ..... 32
Tabel 3.4 kriteria 1 (C1) ..... 33
Tabel 3.5 kriteria 2 (C2) ..... 33
Tabel 3.6 kriteria 3 (C3) ..... 33
Tabel 3.7 kriteria 4 (C4) ..... 33
Tabel 3.8 kriteria 5 (C5) ..... 33
Tabel 3.9 kriteria 6 (C6) ..... 33
Tabel 3.10 kriteria 7 (C7) ..... 33
Tabel 3.11 kriteria 8 (C8) ..... 33
Tabel 3.12 kriteria 9 (C9) ..... 33
Tabel 3.13 kriteria 10 (C10) ..... 34
Tabel 3.14 kriteria 11 (C11) ..... 34
Tabel 3.15 kriteria 12 (C12) ..... 34
Tabel 3.16 kriteria 13 (C13) ..... 34
Tabel 3.17 kriteria 14 (C14) ..... 34
Tabel 3.18 kriteria 15 (C15) ..... 34
Tabel 3.19 kriteria 16 (C16) ..... 34
Tabel 3.20 kriteria 17 (C17) ..... 34
Tabel 3.21 kriteria 18 (C18) ..... 34
Tabel 3.22 kriteria 19 (C19) ..... 34
Tabel 3.23 kriteria 20 (C20) ..... 35
Tabel 3.24 kriteria 21 (C21) ..... 35
Tabel 3.25 kriteria 22 (C22) ..... 35

Tabel 3.26 kriteria 23 (C23) ..... 35
Tabel 3.27 Nilai Bobot Kriteria ($W_i$) ..... 36
Tabel 3.28 Rating kecocokan alternatif dan kriteria ..... 37
Tabel 3.29 Hasil perangkingan ..... 39
Tabel 3.30 Entitas *user* ..... 49
Tabel 3.31 Entitas korban ..... 49
Tabel 3.32 Entitas kriteria ..... 49
Tabel 3.33 Entitas detailkriteria ..... 50
Tabel 3.34 Standar penilaian UAT ..... 54
Tabel 4.1 *Path* algoritma SAW ..... 72
Tabel 4.2 Pengujian algoritma SAW ..... 72
Tabel 4.3 Pengujian fungsional fitur *user admin* ..... 73
Tabel 4.4 Pengujian fungsional fitur *user team survey* ..... 74
Tabel 4.5 Hasil pengujian UAT dasi sisi admin ..... 75
Tabel 4.6 Hasil pengujian UAT dari sisi team survey ..... 76
Tabel 4.7 Perbandingan nilai SAW sistem dan SAW manual ..... 77

# ABSTRAK

Bencana alam gempa merupakan salah satu bencana alam yang mengakibatkan kerugian materi maupun non materi bagi daerah dan masyarakat yang terdampak. Pemerintah melalui Lembaga pemerintah yakni Badan Penanggulangan Bencana Daerah (BPBD) melakukan kegiatan pemberian bantuan bagi korban terdampak berdasarkan beberapa kriteria kerusakan rumah yang dialami yakni rusak ringan, sedang, dan berat. Dalam menentukan kriteria dibutuhkan data-data kerusakan rumah korban yang dalam pendataanya masih dilakukan secara manual yaitu mengisi berkas form kerusakan rumah yang dialami korban sehingga cukup membutuhkan waktu untuk melakukan pendataan dan penentuan kriterianya. oleh karena itu dalam penilitian ini akan dibangun sebuah Sistem Pendukung Keputusan (SPK) berbasis *website* yang dalam pengembangan sistemnya menggunakan *framework codeigniter* dengan bahasa pemrograman PHP dan *database mysql*. Dalam sistem ini perhitunganya menggunakan metode SAW karena metode ini cukup sesuai dengan topik permasalahan pada penelitian yang dilakukan. Hasil akhir dari pengembangan sistem ini ialah keputusan kategori kerusakan rumah korban bencana alam gempa dengan kategori rusak ringan, rusak sedang atau rusak berat.

**Kata kunci:** Sistem Pendukung Keputusan, *Simple Additive Weighting*, *Website*.

# BAB I

# PENDAHULUAN

## 1.1. Latar Belakang

Indonesia adalah salah satu negara yang terletak antara pertemuan sirkum mediterania dan rangkaian sirkum pasifik dengan proses pembentukan gunung berapi yang masih berlangsung. Oleh sebab itu hampir setiap tahun, Indonesia dilanda bencana alam salah satunya bencana alam gempa bumi baik dengan disertai tsunami maupun tidak [1]. Dalam catatan bencana gempa besar yang pernah terjadi di Indonesia diantaranya ialah gempa besar yang pernah terjadi pada bulan Agustus tahun 2018 lalu di Pulau Lombok dengan kekuatan magnitudo 7.0 skala richter yang menyebabkan kerusakan rumah warga untuk wilayah Lombok Barat sejumlah 13.026 rumah rusak berat, 12.188 rumah rusak sedang dan 44.554 rumah rusak ringan [2]. Bencana alam gempa bumi ini sangat berdampak bagi masyarakat yang menjadi korban dan mengakibatkan kerugian materi maupun non-materi, tidak luput juga banyaknya nyawa manusia yang terenggut dan dampak skala besarnya mengakibatkan kelumpuhan aktivitas ekonomi masyarakat sehingga membuat masyarakat yang menjadi korban bencana alam gempa ini menderita dalam segi jiwa maupun materi.

Badan Nasional Penanggulangan Bencana (BNPB) dan Badan Penanggulangan Bencana Daerah (BPBD) ialah lembaga negara dalam mengurus bencana alam yang terjadi baik dalam skala nasional dan skala daerah. Badan Nasional ini akan selalu terkait dalam melakukan kegiatan rehabilitasi dan rekontruksi pasca bencana gempa. Salah satu bentuk kegiatannya adalah pemberian bantuan bagi para korban terdampak bencana alam gempa. Untuk metode penyaluran bantuan saat ini ialah sesuai dengan Instruksi Presiden (Inpres) No 5 tahun 2018 mengenai penyaluran bantunan keuangan terhadap korban bencana alam gempa, dalam instruksi tersebut faktor penentunya berdasarkan pada tingkat kerusakan bangunan rumah korban. Ada beberapa variabel yang digunakan sebagai tolak ukur untuk tingkat kerusakan bangunan yang terjadi diantaranya ialah pondasi bangunan, kolom/ring balok bangunan, rangka atap (kuda-kuda) bangunan, penutup atap/genteng, langit-langit/platfon, dinding, dan lantai yang setiap variabel tersebut memiliki bobot tersendiri untuk dihitung dalam menentukan keputusan tingkat kerusakan yang dialami.

Sesuai dengan hasil wawancara ke pihak BPBD dan sesuai dengan pelatihan untuk masyarakat yang telah diadakan oleh pihak BPBD mengenai sistem penilaian tingkat kerusakan dari bangunan rumah korban, datanya masih belum sepenuhnya terdigitalisasi. Dalam penyimpanan datanya masih berupa tumpukan kertas dan untuk pengumpulan data korban beserta kerusakan rumah atau gedung yang dialami masih dilakukan secara manual oleh tim survei yang turun langsung ke lokasi bencana dan dibantu oleh aparat dari desa setempat. Sementara dalam menentukan penerima bantuan korban bencana alam gempa diperlukan keputusan yang tepat dan cepat sehingga dapat memenuhi kebutuhan korban terdampak gempa dalam memulihkan kehidupan sosial mereka pasca gempa terjadi.

Sistem pendukung keputusan (SPK) merupakan sebuah sistem yang dibangun dengan tujuan mendukung pengambilan keputusan dengan melihat kriteria atau variabel yang diperhitungkan dalam menentukan keputusan akhirnya. Pada penelitian ini digunakan metode *Simple Additive Weighting* (SAW) karena berdasarkan penelian-penelitian sebelumnya yang menggunakan metode SAW, metode ini cukup sesuai dengan topik permasalahan pada penelitian yang dilakukan. Penilaiannya berdasarkan kriteria dan bobot dari masing-masing kriteria yang sudah ditentukan sehingga dari hasil perhitungan bobot dari setiap kriteria yang ada pada penelitian ini diantaranya kriteria dari pondasi bangunan, kriteria dari kolom/ring balok bangunan, kriteria dari rangka atap (kuda-kuda) bangunan, kriteria dari penutup atap/genteng, kriteria dari langit-langit/platfon, kriteria dari dinding, dan kriteria dari lantai akan diperoleh hasil yang menjadi keputusan akhirnya. Selain itu pada beberapa penelitian sebelumnya menyebutkan kelebihan metode SAW ialah lebih efisien karena waktu yang dibutuhkan dalam perhitungan lebih singkat, serta untuk hasil dari beberapa penelitian lainnya yang menggunakan metode SAW menyimpulkan bahwa metode SAW mampu memberikan alternatif atas permasalahan yang dihadapi dan mampu berjalan sesuai yang diharapkan[3].

Berdasarkan pemaparan dan permasalahan tersebut dalam penelitian ini akan dikembangkan sebuah sistem pendukung keputusan untuk penentuan penerima bantuan korban bencana alam gempa berbasis *website* yang bisa digunakan dalam pendataan para korban terdampak bencana gempa dan untuk metode pendukung keputusanya

menggunakan metode *Simple Additive Weighting* (SAW) sebagai metode alternatif penentuan penerima bantuan korban bencana alam gempa.

## 1.2. Rumusan Masalah

Berdasarkan pada latar belakang yang telah diuraikan maka perumusan masalah yang didapat pada tugas akhir ini yakni

a. Bagaimana cara membangun sistem pendukung keputusan untuk penentuan penerima bantuan korban bencana alam gempa berbasis *website* studi kasus BPBD Lombok Barat?
b. Bagaimana mengimplementasikan metode *Simple Additive Weighting* (SAW) sebagai metode alternatif pemberian bantuan untuk korban bencana alam gempa?

## 1.3. Batasan Masalah

Dari penjabaran permasalahan diatas dapat diambil kesimpulan untuk batasan-batasan masalah dalam pembangunan sistem ini yaitu :

a. Pembangunan aplikasi pendukung keputusan penentuan penerima bantuan korban bencana alam gempa berbasis *website* ini ditujukan untuk lembaga yang menangani masalah pendataan korban bencana alam gempa dengan studi kasus di BPBD Lombok Barat.
b. Pembangunan aplikasi berbasis *website* ini menggunakan *framework CodeIgniter* (CI) dengan Bahasa pemrograman PHP dan pengolahan databasenya menggunakan MySQL.
c. Administrator dari aplikasi ini ialah lembaga penanggulangan bencana gempa yaitu BPBD Lombok Barat.
d. Pengguna dari aplikasi ini ialah *team* yang dibentuk oleh Lembaga BPBD Lombok Barat dalam melakukan pendataan korban bencana alam gempa.
e. Hasil akhir berupa keputusan kategori kerusakan bangunan rumah korban yang terbagi menjadi tiga, yaitu: rusak ringan, rusak sedang, rusak berat.

## 1.4. Tujuan

Adapun tujuan yang akan dicapai dari tugas akhir ini yakni :

a. Membangun sistem pendukung keputusan untuk penentuan penerima bantuan korban bencana alam gempa berbasis *website* untuk membantu dalam pendataan dan penentuan penerima bantuan korban bencana alam gempa.
b. Mengimplementasikan metode *Simple Additive Weighting* (SAW) sebagai metode alternatif pemberian bantuan dengan parameter kriteria tingkat kerusakan bangunan rumah korban.

## 1.5. Manfaat

Manfaat yang diperoleh dari tugas akhir ini adalah :

1. Dapat membantu pihak BPBD Lombok Barat dalam menentukan para penerima bantuan korban bencana alam gempa dengan melihat variabel-variabel yang mempengaruhi keputusan.
2. Dapat menggunakan aplikasi sistem pendukung keputusan dalam penentuan tingkat kerusakan dari rumah korban bencana alam gempa.
3. Dapat menerapkan metode *Simple Additive Weighting* (SAW) dalam penentuan kriteria bantuan bagi penerima bantuan korban bencana alam gempa.
4. Dapat membuat data yang masih berupa kertas menjadi data digital dalam pendataan maupun penyimpanan data korban bencana alam gempa.

## 1.6. Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut :

1. Bab I Pendahuluan
   Bab ini membahas dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, dan sistematika penulisan laporan tugas akhir.
2. Bab II Tinjauan Pustaka dan Dasar Teori
   Bab ini membahas teori-teori yang mendukung dan berhubungan dengan topik penelitian.

3. Bab III Metodologi Penelitian
   Bab ini membahas tentang metodologi yang digunakan dalam pengembangan sistem pada penelitian ini.

4. Bab IV Hasil dan Pembahasan

   Bab ini akan disajikan hasil dari penelitian yang dilakukan serta pembahasanya.

5. Bab V Kesimpulan dan Saran

   Pada bab terakhir ini akan diberikan kesimpulan beserta saran dari hasil penelitian yang telah dilakukan.

# BAB II

# TINJAUAN PUSTAKA DAN DASAR TEORI

## 2.1. Tinjauan Pustaka

Penelitian dengan menggunakan metode *Simple Additive Weighting* (SAW) sebagai metode dalam sistem pendukung keputusan telah dilakukan oleh beberapa peneiliti dalam beberapa kasus diantaranya sebagai berikut :

Penelitian yang berjudul "Sistem Pendukung Keputusan Kelayakan Pemberian Kredit Pada Nasabah Dengan Metode Simple Additive Weighting (Studi Kasus : PT. BPRS AL WASHLIYAH MEDAN)". Penelitian ini menentukan layak atau tidaknya kredit yang akan diberikan pada calon nasabah dengan melihat beberapa kriteria penilaian. Metode yang digunakan dalam pengambilan keputusanya ialah metode *Simple Additive Weighting* (SAW). Variabel yang digunakan untuk diperhitungkan dalam pengambilan keputusan pada penelitian ini diantaranya pekerjaan nasabah, penghasilan nasabah, jaminan, jumlah tanggungan dan status rumah nasabah. Dalam pengembangan aplikasinya penelitian ini menggunakan aplikasi berbasis *website* dengan bahasa pemgrogaran PHP murni dan pengolahan datanya menggunakan MySQL. Dan hasil output dari penelitian ini berupa hasil seleksi calon nasabah berupa rekomendasi nama calon nasabah yang diurutkan berdasarkan perangkingan [4].

Selain itu juga dalam penelitian dengan judul "Sistem Pendukung Keputusan Rekomendasi Penerima Bantuan Siswa Miskin (BSM) dengan Metode *Simple Additive Weighting* (SAW) (Studi Kasus : SMPN 2 Tarabintang)" menyimpulkan bahwa penelitian ini mampu merancang suatu sistem pendukung keputusan menggunakan bahasa pemrograman *Visual Basic 6.0* dengan menggunakan database MySQL sedangkan untuk metode dalam pengambilan keputusanya menggunakan metode *Simple Additive Weighting* (SAW) dan variabel yang digunakan sebagai inputannya ialah pekerjaan dan penghasilan orang tua siswa sehingga output dari penelitian ini ialah rekomendasi nama-nama siswa miskin yang berhak menerima BSM [5].

Dalam penilitian yang berjudul "Implementasi Metode Logika *Fuzzy Simple Additive Weighting* (SAW) Dalam Pencarian Rumah Kos Terbaik Di Sekitar Universitas Mataram Berbasis *Website*" adalah penelitian yang membangun sistem untuk membantu

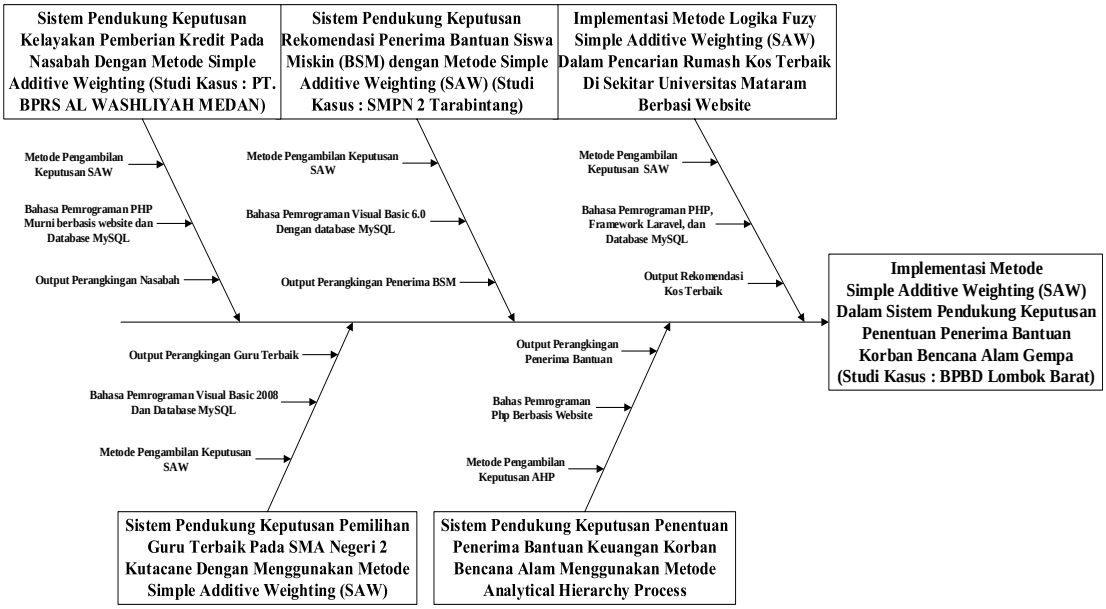
dalam pencarian rumah kos terbaik di sekitar Universitas Mataram. Metode yang digunakan dalam penelitian ini ialah metode *Simple Additive Weighting* (SAW) dengan variabel yang digunakan untuk perhitungan dalam pengambilan keputusanya ialah jarak, harga, dan fasilitas. Dalam Pengembangan sistemnya yang berbasis *website* penelitian ini menggunakan bahasa pemgrograman PHP dengan *framework Laravel* dan perancangan *database*-nya menggunakan MySQL. Untuk hasil luaran dari penelitian ini ialah berupa rekomendasi kos terbaik disekitar Universitas Mataram [6].

Berdasarkan penelitian yang berjudul "Sistem Pendukung Keputusan Pemilihan Guru Terbaik Pada SMA Negeri 2 Kutacane Dengan Menggunakan Metode *Simple Additive Weighting* (SAW)" Sistem ini dikembangkan dengan menggunakan bahasa pemrograman visual basic 2008 dan database MySQL. Untuk metode yang digunakan dalam pengambilan keputusanya menggunakan metode *Simple Additive Weighting* (SAW) dengan variabel penilaian yang digunakan ialah kedisiplinan, prestasi, prakarsa, tanggung jawab, dan menjaga nama baik sebagai pendidik. *Output* atau keluaran dari sistem ini berupa daftar nama-nama guru terbaik berdasarkan perangkingan [3].

Pada penelitian dengan judul "Sistem Pendukung Keputusan Penentuan Penerima Bantuan Keuangan Korban Bencana Alam Menggunakan *Metode Analytical Hierarchy Process*" yang menyimpulkan bahwa penelitian ini berhasil menciptakan suatu sistem pendukung keputusan untuk penentuan penerima bantuan keuangan bencana alam menggunakan metode AHP berbasis *website* sebagai alternatif pemberian bantuan untuk korban bencana alam dengan kriteria kategori perhitungan berdasarkan kategori kerusakan, kategori keluarga dan jumlah anggota rumah tangga korban. Dalam pengembangan sistemnya menggunakan bahasa pemrograman PHP murni. Kesimpulan hasil luaran dari penilitan ini ialah berupa rekomendasi nama para penerima bantuan bencana [1].

Berdasarkan tinjauan pustaka diatas, maka pada penelitian ini akan dibahas bagaimana cara membangun sistem pendukung keputusan dalam kasus penentuan penerima bantuan korban bencana alam gempa dengan menggunakan metode pendukung keputusan *Simple Additive Weighting* (SAW) dalam pengambilan keputusanya dan dalam pengambangan sistem yang berbasis *website* menggunakan bahasa pemrograman PHP dengan *framework CodeIgniter* dan perancangan *database* menggunakan MySQL.

Untuk lebih jelasnya mengenai tinjauan pustaka yang digunakan dalam penelitian ini dapat dilihat pada Gambar 2.1.




Gambar 2.1 Fishbone Diagram

## 2.2. Dasar Teori

Berikut beberapa dasar teori tentang konsep-konsep yang digunakan dalam pembuatan sistem pendukung keputusan penerima bantuan korban bencana gempa beserta materi-materi yang mendukung penyelesaian penelitian ini sebagai berikut.

### 2.2.1.Sistem Informasi

Sistem informasi adalah sistem yang disusun oleh satu atau beberapa sistem pengolahan data meliputi data, informasi, pengetahuan, dan kebijaksanaan. Dalam sistem informasi terdapat proses atau aktivitas aktivitas yang akan mendukung aktivitas usaha dalam sistem organisasi. Proses dari sistem informasi antara lain mengumpulkan data (data *gathering*), mengelolah data yang tersimpan, menyebarkan informasi. Di dalam sistem informasi, manusia berinteraksi dengan manusia, manusia dengan komputer, dan komputer berinteraksi dengan komputer lain dimana data, informasi dan pengetahuan mengalir di bawah media elektronik jaringan komputer. Keberadaan sistem informasi diperlukan dalam organisasi untuk mendampingi proses-proses bisnis dari organisasi [6].

### 2.2.2.Sistem Pendukung Keputusan

Sistem pendukung keputusan (SPK) merupakan sebuah sistem yang dibangun dengan tujuan mendukung pengambilan keputusan pada permasalahan tidak terstruktur maupun semi-terstruktur. Sebuah sistem pendukung keputusan dapat mendukung analisis *ad-hoc* data dan pemodelan keputusan, yang berorientasi pada perencanaan masa depan yang digunakan pada waktu yang tidak biasa dan rentang waktu yang tidak dapat ditentukan. Sistem pendukung keputusan merupakan sistem berbasis komputer yang mengandung interaksi tiga komponen yaitu, bahasa, pengetahuan serta pengolahan masalah. Beberapa tujuan dari sebuah sistem pendukung keputusan, diantaranya [7]:

a. Membantu pengambilan keputusan untuk pemasalahan semi terstruktur.
b. Sebagai pemberi dukungan atas pertimbangan manajer dalam pengambilan keputusan.
c. Untuk meningkatkan efektifitas keputusan yang diambil.
d. Mempercepat pengambilan keputusan, karena menggunakan proses komputasi.
e. Meningkatkan kualitas keputusan yang diambil.

### 2.2.3.*Website*

*Website* adalah kumpulan halaman-halaman yang digunakan untuk mempublikasikan informasi berupa teks, gambar dan program multimedia lainnya berupa animasi (gambar gerak, tulisan gerak), suara dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait antara satu *page* dengan *page* yang lain yang sering disebut sebagai *hyperlink*. *Website* biasanya terangkum dalam sebuah domain atau sub domain yang tempatnya berada di dalam *World Wide Web* (WWW) di internet. Sebuah *Web Page* adalah dokumen yang ditulis dalam format HTML yang hampir selalu bisa diakses melalui HTTP, yaitu protocol yang bisa menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

Halaman-halaman dari *website* akan bisa diakses melaui sebuah URL yang bisa disebut *Homepage*. URL ini mengatur halaman-halaman situs untuk menjadi sebuah hirarki, meskipun *hyperlink* yang ada di halaman tersebut mengatur para pembaca dan memberitahu mereka susunan keseluruhan dan bagaimana arus informasi ini berjalan. Beberapa *website* membutuhkan subskripsi (data masukan) agar para *user* bisa

mengakses sebagian atau keseluruhan isi *web* tersebut. Contohnya, ada beberapa situs-situs bisnis, situs-situs email gratisan yang membutuhkan subskripsi agar kita bisa mengakses situs tersebut [8].

### 2.2.4.Pedoman Penerimaan Bantuan Korban Bencana Alam Gempa Lombok

Dalam proses penyaluran bantuan korban bencana alam gempa khususnya di wilayah kabupaten Lombok Barat, BPBD Lombok Barat mengacu pada Instruksi Presiden (Inpres) no 5 Tahun 2018 yang di instruksikan kepada kepala Badan Nasional Penanggulangan Bencana (BNPB) tentang bantuan keuangan kepada korban bencana alam gempa. Penentuan bantuan didasarkan hanya berdasarkan pada faktor kerusakan saja, tidak memandang faktor ekonomi atau faktor-faktor yang lainya. Ada tiga kategori kerusakan, yaitu :

a. Rumah rusak ringan

   Secara struktur masih dalam kondisi baik, dikatakan baik jika sebagian komponen bangunan mengalami kerusakan yang tidak mempengaruhi kekuatan struktur bangunan dan masih bisa bertahan jika terjadi kembali bencana alam gempa. Untuk rumah yang rusak ringan akan mendapatkan bantuan minimal sebesar Rp.10.000.000,00.

b. Rumah rusak sedang

   Rumah rusak yang bangunannya masih berdiri, kerusakan tidak mempengaruhi kekuatan struktur bangunan tapi diperkikaran aman apabila terjadi kembali bencana gempa. Untuk rumah yang rusak sedang akan mendapatkan bantuan minimal sebesar Rp.25.000.000,00.

c. Rumah rusak berat

   Rumah yang benar-benar roboh/hancur atau bangunan yang tidak layak huni akibat bencana alam gempa yang terjadi karena strukturnya sudah rusak dan tidak dapat dipertahankan. Untuk rumah yang rusak berat akan mendapatkan bantuan minimal sebesar Rp.50.000.000,00.

Untuk panduan lebih detail mengenai tingkat kerusakan yang dialami bangunan rumah korban dapat dilihat pada Tabel 2.1 dan Tabel 2.2 sebagai berikut.

Tabel 2.1 Panduan tingkat kerusakan aset

| Klasifikasi kerusakan | Persentase tingkat kerusakan | Kategori tingkat kerusakan |
|---|---|---|
| Rusak Ringan | ≤ 30 % | Terdapat kerusakan namun masih berfungsi |
| Rusak Sedang | 31 - 70 % | Terdapat kerusakan masih berfungsi dan bisa diperbaiki |
| Rusak Berat | ≥ 71 % | Terdapat kerusakan fisik secara keseluruhan, sehingga tidak dapat berfungsi sama sekali |

Pada Tabel 2.1 dan 2.2 merupakan parameter tingkat kerusakan aset yang sudah ditetapkan oleh Badan Penanggulanan Bencana Nasional (BNPB) sebagai Lembaga Negara yang bertanggung jawab dalam menangani bencana alam sekala Nasional yang kemudian digunakan oleh BPBD sebagai panduan dalam menentukan tingkat kerusakan.

Tabel 2.2 Panduan bobot kerusakan rumah/gedung

| Komponen | Detail komponen | Visual detail kerusakan | Jumlah komponen | Bobot kerusakan |
|---|---|---|---|---|
| Struktur | Pondasi/ Sloof | Retak rambut dapat terlihat jelas pada permukaan beton (lebar retakan kira-kira 0,2 – 1 mm) | 1<br>2<br>≥3 | 3<br>4<br>5 |
| | | Retakan yang sangat jelas (lebar retakan kira-kira 1 – 2 mm) | 1<br>2<br>≥3 | 3<br>4<br>5 |
| | | Kehancuran beton yang sangat nyata dengan tulangan beton terlihat | 1<br>2<br>≥3 | 5<br>8<br>11 |
| | | Penurunan dan atau kemiringan lantai dapat terlihat | 1<br>2<br>≥3 | 11<br>14<br>15 |
| | | Pondasi hancur | 1<br>2<br>≥3 | 11<br>14<br>15 |
| | Kolom/Ring Balok | Retak rambut dapat terlihat jelas pada permukaan beton (lebar retakan kira-kira 0,2 – 1 mm) | 1<br>2<br>≥3 | 7<br>9<br>11 |
| | | Retakan yang sangat jelas (lebar retakan kira-kira 1 – 2 mm) | 1<br>2<br>≥3 | 7<br>9<br>11 |

| Komponen | Detail komponen | Visual detail kerusakan | Jumlah komponen | Bobot kerusakan |
|---|---|---|---|---|
| | | Kehancuran beton yang sangat nyata dengan tulangan beton terlihat | 1<br>2<br>≥3 | 12<br>19<br>25 |
| | | Tulang tertekuk | 1<br>2<br>≥3 | 26<br>32<br>35 |
| | | Deformasi vertikal/miring pada kolom dapat terlihat | 1<br>2<br>≥3 | 26<br>32<br>35 |
| | Rangka Atap(Kuda-kuda) | Reng hancur sebagian atau seluruh | 1<br>2<br>≥3 | 3<br>4<br>5 |
| | | Usuk hancur sebagian atau seluruh | 1<br>2<br>≥3 | 5<br>8<br>11 |
| | | Kuda-kuda roboh sebagian atau seluruh | 1<br>2<br>≥3 | 11<br>14<br>15 |
| Non Struktur | Penutup Atap/Genteng | Berjatuhan menyebar sebagian atau seluruh | 1<br>≥2 | 4<br>5 |
| | Langit-langit/Plafon | Langit-langit terlepas/jatuh menyebar sebagian atau seluruh | ≤2<br>≥3 | 1<br>2 |
| | | Langit-langit terlepas/jatuh dan rangka langit-langit rusak menyebar sebagian atau seluruh | 1<br>≥2 | 4<br>5 |
| | Dinding | Retak rambut pada plesteran (lebar retak < 0,2 mm) | 1<br>2<br>≥3 | 3<br>4<br>5 |
| | | Plesteran terkelupas | 1<br>2<br>≥3 | 3<br>4<br>5 |
| | | Retakan dapat terlihat jelas, tembus pada pasangan bata/batako (lebar retakan kira-kira > 0,3 mm) | 1<br>2<br>≥3 | 5<br>8<br>11 |
| | | Dinding mengalami displacement horisontal/miring | 1<br>2<br>≥3 | 11<br>14<br>15 |
| | | Dinding jatuh sebagian atau total | 1<br>2<br>≥3 | 11<br>14<br>15 |
| | Lantai | Retak rambut menyebar sebagian atau seluruh | 1<br>≥3 | 2<br>3 |

| Komponen | Detail komponen | Visual detail kerusakan | Jumlah komponen | Bobot kerusakan |
|---|---|---|---|---|
| | | Retak besar/ terbelah/ | 1 | 8 |
| | | hancur menyebar sebagian | 2 | 9 |
| | | atau seluruh | ≥3 | 10 |

### 2.2.5.UML

*Unified Modelling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti *C++*, *Java*, *C#* atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*) [9].

#### 2.2.5.1.*UseCase Diagram*

*Usecase diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *usecase* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Usecase* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [9]. Simbol-simbol *usecase diagram* dapat dilihat pada Tabel 2.3.

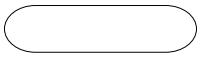
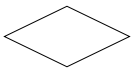
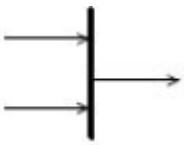
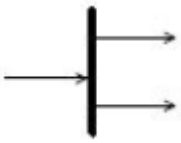
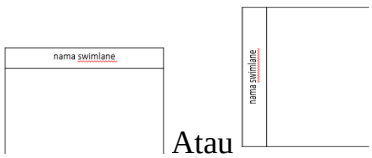
Tabel 2.3 Simbol UseCase Diagram

| Simbol | Nama | Keterangan |
|---|---|---|
| | *Actor* | *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. |
| | *Use Case* | Gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. |
|  | *Association* | Menghubungkan *link* antar *element*. |
|  | *Include* | Kejadian yang harus terpenuhi agar sebuah *event* dapat terjadi, di mana pada kondisi ini sebuah *usecase* adalah bagian dari *usecase* lainnya. |
|  | *Extends* | Relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* yang ditambahkan dapat berdiri sendiri. |
|  | *Generalization* | Disebut juga *inheritance* (pewarisan). |

### 2.2.5.2.*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* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah aksi dan sebagian besar transisi dibangkitkan oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum [9]. Simbol-simbol pada diagram aktivitas dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-simbol activity digram

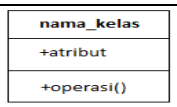
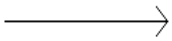
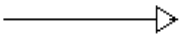
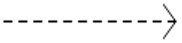
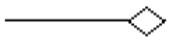
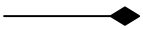
| Simbol | Deskripsi |
|---|---|
| Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| *Decision*  | Asosiasi *decision* di mana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan / *Join*  | Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu. |
| Percabangan / *Fork*  | Asosiasi percabangan di mana satu aktivitas bagi menjadi beberapa aktivitas. |
| Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas harus memiliki status akhir. |
| *Swimlane*   Atau | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

### 2.2.5.3.*Class Diagram*

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakaninti ari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti

*containment*, pewarisan, asosiasi, dan lain-lain [9]. Pada *class diagram* terdapat beberapa simbol dan beberapa cara penulisan diantaranya dapat dilihat pada Tabel 2.5

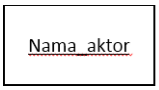
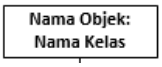
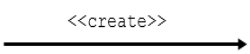
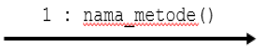
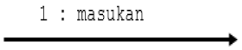
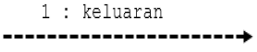
Tabel 2.5 Simbol class diagram

| Simbol | Nama | Deskripsi |
|---|---|---|
|  | *Class* | Kelas pada struktur sistem. |
|  | Asosiasi / *Association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*. |
|  | Asosiasi berarah / *Directed association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*. |
|  | Generalisasi | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
|  | Kebergantungan / *dependency* | Relasi antar kelas dengan makna kebergantungan antar kelas. |
|  | Agregasi / *Aggregation* | Relasi antar kelas dengan makna semua-bagian (*whole-part*) |
|  | *Computition* | Bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi *whole* dibuat. |

### 2.2.5.4.*Sequence Diagram*

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu [9]. Simbol-simbol yang terdapat pada *sequence diagram* dapat dilihat pada Tabel 2.6.

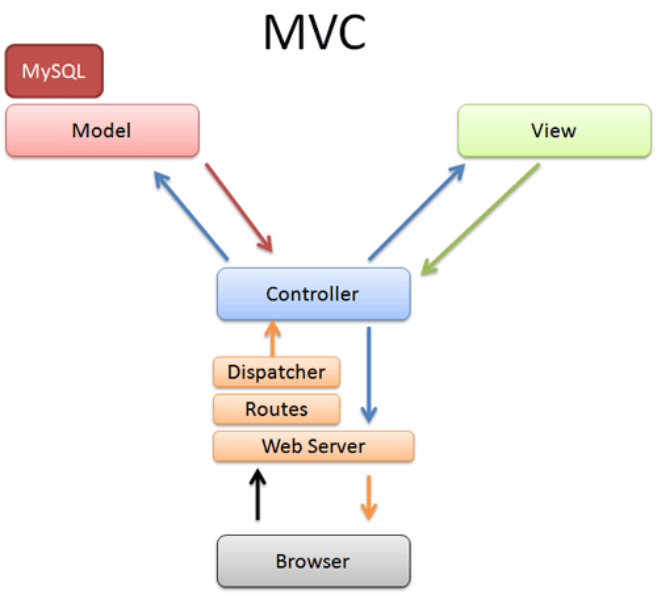
Tabel 2.6 Simbol-simbol sequence diagram

| Simbol | Deskripsi |
|---|---|
| *Lifeline*  | Menyatakan kehidupan suatu objek. |
| Aktor   | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di bawa frase nama aktor. |
| Objek   | Menyatakan objek yang berinteraksi pesan. |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya |
| Pesan tipe *create*   | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |
| Pesan tipe *call*   | Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek ain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. |
| Pesan tipe *send*   | Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi. |
| Pesan tipe *return*   | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |

### 2.2.6.*Framework CodeIgniter*

*Framework* adalah kumpulan instruksi-instruksi yang di kumpulkan dalam *class* dan *function-function* dengan fungsi masing-masing untuk memudahkan *developer* dalam pemanggilannya tanpa harus menuliskan *syntax* program yang sama berulang-ulang. hal ini memiliki kegunaan untuk menghemat waktu dan mencegah penulisan *syntax* secara berulang-ulang agar tercipta nya *source code* yang bersih dan terstruktur.

*CodeIgniter* adalah sebuah *framework* php yang bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*). *codeigniter* bersifat *free* alias tidak berbayar jika anda menggunakannya. *Framework codeigniter* di buat dengan tujuan sama seperti *framework* lainnya yaitu untuk memudahkan *developer* atau *programmer* dalam membangun sebuah aplikasi berbasis *web* tanpa harus membuat nya dari awal [9]. Arsiterktur MVC dapat dilihat pada Gambar 2.2 .




Gambar 2.2 Arsitektur MVC

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu *model, view dan controller*.

a. *Model*

*Model* merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi *database*. seperti misalnya mengambil data dari *database*, meng-*input* dan

pengolahan *database* lainnya. semua instruksi yang berhubung dengan pengolahan *database* di letakkan di dalam *model*.

b. *View*

*View* merupakan bagian yang menangani halaman *user interface* atau halaman yang muncul pada *user*. tampilan dari *user interface* dikumpulkan pada *view* untuk memisahkannya dengan *controller* dan *model* sehingga memudahkan *web designer* dalam melakukan pengembangan tampilan halaman *website*.

c. *Controller*

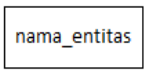
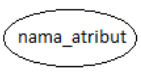
*Controller* merupakan kumpulan instruksi aksi yang menghubungkan model dan *view*, jadi *user* tidak akan berhubungan dengan *model* secara langsung, intinya dari *view* kemudian *controller* yang mengolah instruksi.

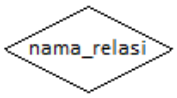
### 2.2.7.ERD

ERD (*Entity Relation Diagram)* merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data.ERD (*Entity Relation Diagram)* berguna untuk menggambarkan gambaran dari dunia nyata yang akan diterapkan pada suatu *database* sebuah sistem. ERD melihat objek nyata dapat sebagai sebuah entitas - entitas yang memiliki relasi antara entitas yang satu ataupun yang lain. Dengan ERD sendiri dapat membantu mengurangi kesalahan-kesalahan dalam melakukan perancangan *database* dari gambaran dunia nyata dan struktur *database* seperti redudansi data, hubungan – hubungan antara entitas dan lain sebagainya [10].

Pada ERD terdapat beberapa simbol dengan cara penulisan dapat dilihat pada Tabel 2.7.

Tabel 2.7 Simbol ERD

| Simbol | Nama | Keterangan |
|---|---|---|
|  | Entitas / *Entitiy* | Merupakan data inti yang akan disimpan; bakal tabel pada basis data; biasanya mengacu pada benda yang terlibat dalam aplikasi yang akan dibuat. |
|  | Atribut | *Field* atau kolom data yang butuh disimpan dalam satu entitas. |

| Simbol | Nama | Keterangan |
|---|---|---|
| | Atribut kunci primer | *Field* atau kolom data yang butuh disimpan dalam suatu entitas sebagai kunci akses *record* yang diinginkan. |
|  | Relasi | Penghubung antar entitas; biasanya diawali dengan kata kerja. |
|  | Asosiasi / *Association* | Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. |

### 2.2.8.MySQL

SQL (*Structured Query Language*) adalah sebuah bahasa dipergunakan untuk mengakses data dalam basis data relasional. SQL juga dapat diartikan sebagai antar muka standar untuk sistem manajemen basis data relasional, termasuk sistem yang beroperasi pada komputer pribadi. SQL memungkinkan seorang pengguna untuk mengakses informasi tanpa mengetahui dimana lokasinya atau bagaimana informasi tersebut disusun.

MySQL merupakan DBMS *open source* (kode programnya dapat di-*download* gratis) dan *freeware* (gratis dipakai), walau gratis MySQL tidak bisa dikatakan DBMS tidak bagus karena gratis, hal ini terbukti dengan adanya sebuah operator telpon selular yang menggunakan DBMS ini dan memiliki sistem yang andal dengan optomasi. Sampai saat ini MySQL masih menjadi DBMS *open source* paling banyak digunakan di dunia. MySQL terkenal dengan kecepatan dalam pengaksesan datanya. MySQL saat ini dimiliki oleh perusahaan *Oracle* [10].

### 2.2.9.*Fuzzy Multiple Atribute Decission Making* (FMADM)

*Fuzzy Multiple Attribute Decision Making* (FMADM) ialah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari FMADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan. Pada dasarnya, ada tiga pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif, pendekatan obyektif dan pendekatan integrasi antara subyektif dan

obyektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perankingan alternatif bisa ditentukan secara bebas. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan. Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah FMADM yaitu:

a. *Simple Additive Weighting* (SAW).
b. *Weighted Product* (WP).
c. ELECTRE.
d. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS).
e. *Analytic Hierarchy Process* (AHP).

### 2.2.10. *Simple Additive Weighting* (SAW)

Metode SAW ini sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Tahapan dalam metode SAW adalah sebagai berikut [11] :

a. Menentukan kriteria yang dijadikan acuan pengambilan keputusan.
b. Menentukan rating kecocokan setiap alternatif pada setiap kriteria
c. Membuat matriks keputusan berdasarkan kriteria, kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut sehingga diperoleh matriks ternormalisasi R.
d. Hasil akhir diperoleh dari proses perangkingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan *vector* bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik sebagai solusi.

Formula untuk normalisasi pada metode SAW adalah sebagai berikut:

$$r_{ij} = \begin{cases} \dfrac{x_{ij}}{Max\ x_{ij}} & \text{jika j adalah atribut keuntungan (benefit)} \\ \dfrac{Min\ x_{ij}}{x_{ij}} & \text{jika j adalah atribut biaya (cost)} \end{cases}$$

Keterangan :

$r_{ij}$ : Rating kinerja ternormalisasi

$Max_{ij}$ : Nilai maksimum dari setiap baris dan kolom

$Min_{ij}$ : Nilai minimum dari setiap baris dan kolom

$X_{ij}$ : Baris dan kolom dalam matriks

$r_{ij}$ merupakan rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i=1,2,…,m dan j=1,2,…,n.

Nilai preferensi untuk setiap alternatif (Vi) :

$$v_i = \sum_{j=1}^{n} w_j \, r_{ij}$$

Keterangan :

$V_i$ : nilai akhir dari alternatif

$w_j$ : bobot yang telah ditentukan

$r_{ij}$ : normalisasi matriks

Nilai $V_i$ yang lebih besar menandakan bahwa alternatif $A_i$ lebih tepat untuk dipilih.

# BAB III

# METODOLOGI PENELITIAN

## 3.1. Alat dan Bahan

Alat dan bahan yang dibutuhkan pada penelitian yang akan dilakukan berupa *Software* dan *Hardware* serta data-data dan informasi penunjang selama kegiatan penelitian dilakukan.

1. Alat Penelitian

   Alat-alat baik berupa *software* maupun *hardware* yang akan digunakan dalam pengerjaan penelitian ini diantaranya:

   a. Laptop ACER Intel® Core™ i3-7130U dengan RAM 4 GB , digunakan untuk pendukung utama dalam pengerjaan penelitian baik dalam pembuatan laporan dan pembangunan sistem.
   b. Sistem operasi *Windows* 10 *Home* 64-bit , digunakan untuk menjalankan beberapa *software* yang diperlukan dalam pengerjaan penelitian sebagai pendukung pembuatan sistem.
   c. *Text Editor Sublime* ,digunakan untuk membuat *code* program.
   d. *Microsoft office word* 2019 ,digunakan untuk membuat laporan dari penelitian yang dilakukan.
   e. *Microsoft office visio* 2019 ,digunakan untuk membuat diagram UML dari sistem.
   f. *Framework CodeIgniter* ,digunakan untuk pembuatan sistem berupa *website* agar mempermudah dalam pembangunan sistem.
   g. XAMPP ,digunakan sebagai *server local* pembuatan dan pengelolaan database sistem.

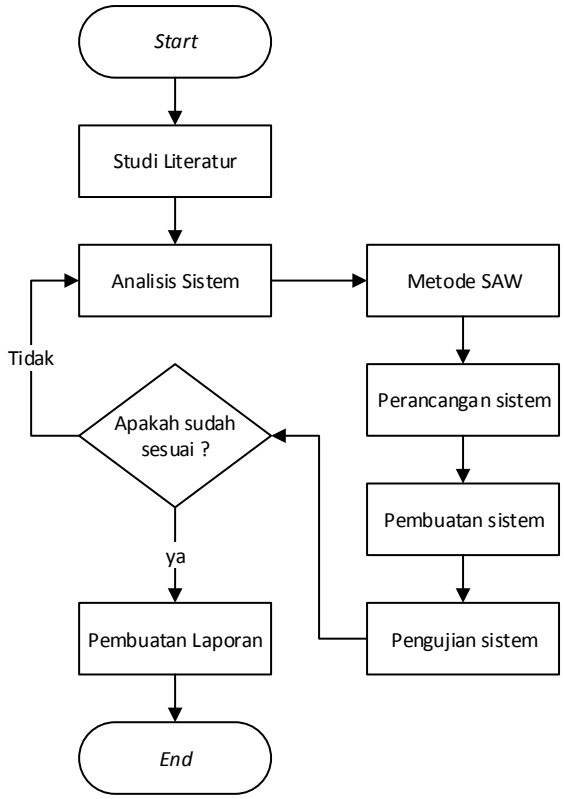
2. Bahan Penelitian

   Bahan-bahan yang dibutuhkan dalam pengerjaan penelitian ini diantaranya ialah data pribadi dari masyarakat yang menjadi korban terdampak gempa yang diperoleh dari Kantor Desa Duman dan data kerusakan beserta panduan tingkat kerusakan aset akibat bencana alam gempa yang diperoleh dari BPBD Lombok Barat.

## 3.2. Alur Penelitian

Penelitian ini tentang pengimplementasian metode pendukung keputusan dalam menentukan kategori bantuan bagi para korban bencana alam gempa Lombok. Sistem ini akan digunakan untuk membantu lembaga penanggulangan bencana yakni BPBD dalam mendata dan menentukan kategori bantuan bagi para korban bencana alam gempa.

Ada beberapa tahapan yang akan dilakukan dalam pengerjaan penelitian ini yang dapat dilihat pada diagram alir penelitan pada Gambar 3.1 berikut.




Gambar 3.1 Diagram alir penelitian.

## 3.3. Studi Literatur

Studi literatur dilakukan bertujuan untuk mencari informasi dan mempelajari lebih dalam teori-teori yang berhubungan dengan pembangunan sebuah sistem pendukung keputusan dalam menentukan suatu keputusan yang diangkat dalam tugas akhir ini ialah keputusan dalam menentukan kategori bantuan bagi para korban bencana alam gempa. Untuk literatur yang digunakan diataranya ialah berupa penelitian terdahulu mengenai sistem pendukung keputusan baik dalam kasus maupun metode yang sama, selain itu

literatur yang digunakan berupa dasar-dasar teori yang terkait baik dalam bentuk jurnal maupun buku.

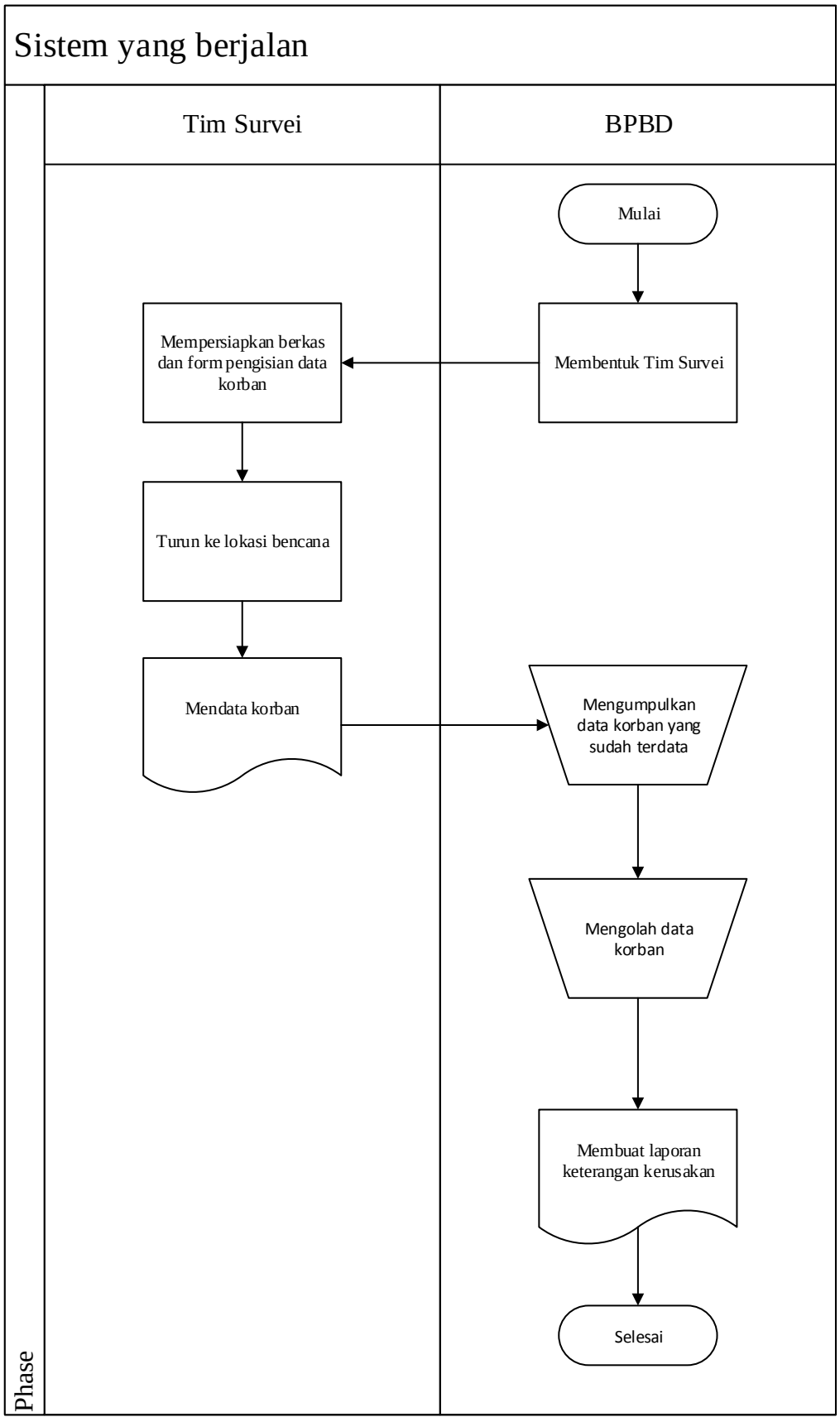
### 3.4. Analisis Sistem

Analisis sistem ialah penguraian terhadap suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dalam sistem sehingga mengetahui kebutuhan yang diharapkan untuk dilakukan penanggulangan, perbaikan, atau juga pengembangan [12].

#### 3.4.1.Analisis Sistem Yang Berjalan

Analisis sistem yang berjalan dilakukan bertujuan untuk mengetahui bagaimana alur kerja dari sistem yang sudah berjalan dilapangan. Pada penelitian ini mengangkat kasus mengenai penentuan penerima bantuan korban bencana alam gempa yang telah terjadi beberapa waktu lalu yang lebih khususnya pada bagian pendataan para korban bencana oleh lembaga terkait yakni BPBD yang berperan dalam mengolah data-data dari korban bencana. BPBD dalam melakukan pendataan bekerja sama dengan lembaga lainnya yakni Dinas Pekerjaan Umum (PU) yang dibantu oleh pihak dari desa membentuk tim kaji teknis atau tim survei untuk turun langsung ke lokasi bencana guna melakukan survei sekaligus pendataan terhadap masyarakat yang menjadi korban terdampak bencana, pendataan yang dilakukan mencangkup data pribadi dan data kerusakan dari rumah korban. Untuk lebih jelas alur dari sistem yang berjalan sebagai berikut:

1. BPBD membentuk tim kaji teknis atau tim survei yang akan turun ke lokasi bencana gempa untuk melakukan pendataan korban.
2. BPBD dan Tim survei mempersiapkan berkas-berkas berupa form isian untuk digunakan dalam melakukan pendataan korban.
3. Tim survei turun ke lokasi bencana untuk melakukan pendataan dengan mengisi berkas form isian sesuai dengan yang terjadi dilapangan.
4. Tim survei membawa data-data para korban berupa berkas-berkas form isian ke BPBD untuk diolah kembali.
5. BPBD mengolah data korban yang sudah terkumpul guna memberikan rekomendasi untuk bisa dikeluarkan surat keterangan rusak ringan, rusak sedang, dan rusak berat dengan membentuk tim verifikasi.




Gambar 3.2 *Flowmap* sistem yang sedang berjalan.

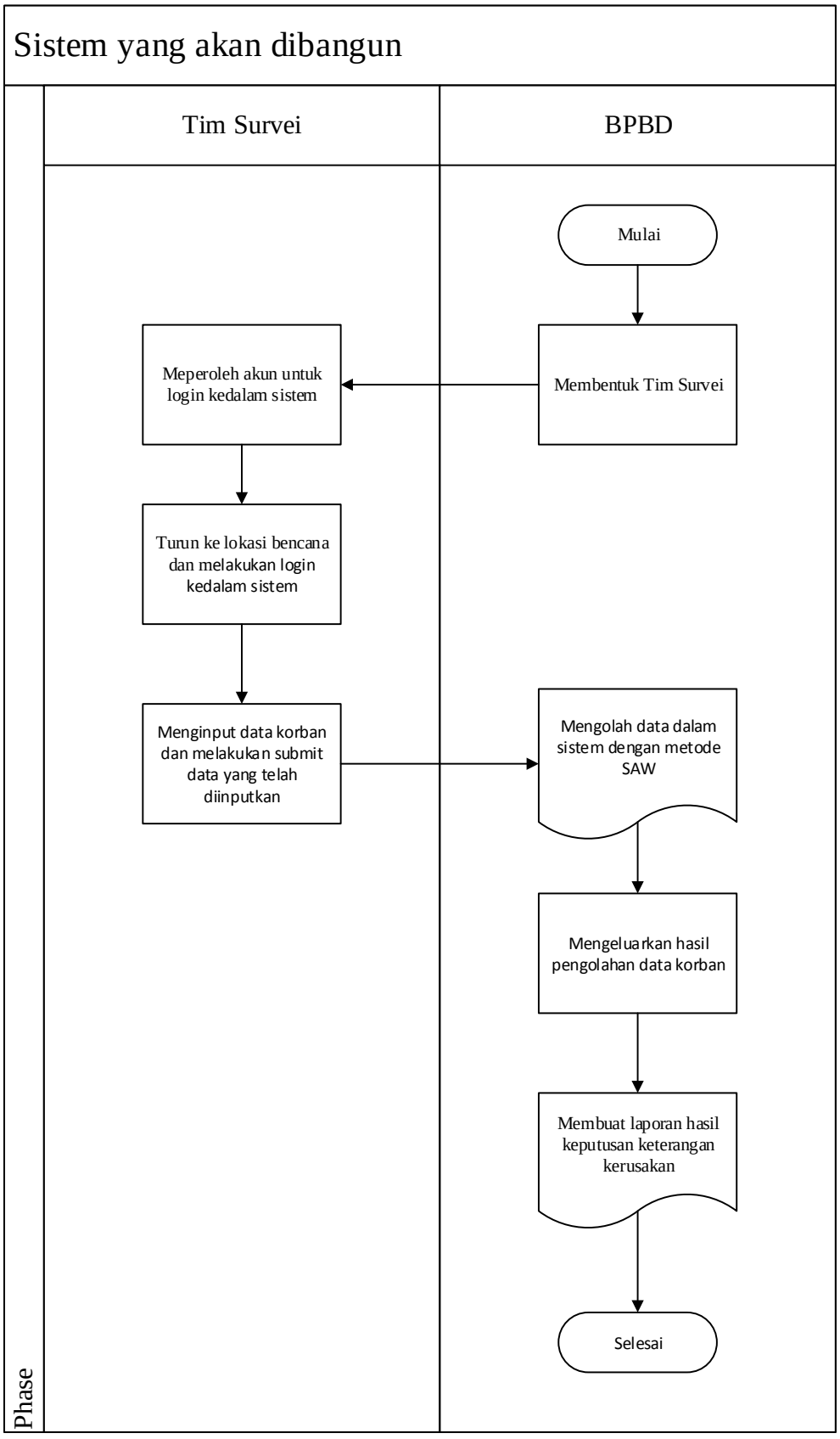
### 3.4.2.Analisis Sistem Yang Akan Dibangun

Analisis sistem yang akan dibangun dilakukan guna mengetahui dan menjelaskan cara kerja dari sistem yang akan dibangun. Sesuai dengan yang sudah dijelaskan pada latar belakang yakni dalam penelitian ini akan dibangun sebuah sistem yang diharapkan dapat membantu pihak BPBD dan tim survei dalam mendata para korban bencana beserta data kerusakan rumah atau bangunan tiap korban yang kemudian dari data-data yang diperoleh tersebut bisa dilakukan perhitungan dengan menggunakan metode pengambilan keputusan yang ditentukan yakni metode *Simple Additive Weighting* (SAW) untuk pengambilan keputusan kategori tingkat kurasakan yang dialami sehingga dapat dilakukan penyaluran bantuan oleh tim penyalur sesegera mungkin agar dapat memenuhi kebutuhan para korban terdampak bencana.

Pemilihan metode *Simple Additive Weighting* (SAW) sebagai metode pendukung keputusan dalam penelitian ini karena metode ini termasuk dalam *Fuzzy Multiple Attribute Decision Making* (FMADM) yang sebagaimana dijelaskan pada bab sebelumnya dimana FMADM ialah metode yang dapat digunakan dalam pengambilan keputusan dengan banyak kriteria sebagaimana dalam kasus pada penelitian ini ada dua puluh tiga kriteria yang memiliki sub-kritertia dengan bobot tersendiri.

Untuk alur kerja dari sistem yang akan dibangun ini ialah sebagai berikut:

1. BPBD membentuk tim kaji teknis atau tim survei, kemudian melakukan persiapan turun ke lokasi bencana untuk melakukan pendataan korban bencana.
2. BPBD dan Tim survei mempersiapkan sistem untuk digunakan dalam melakukan pendataan korban dan diberikan akun masing-masing untuk bisa melakukan *login* kedalam sistem sebagai *user admin* , *user* tim survei dan *user* Kabag.
3. Tim survei turun ke lokasi bencana untuk melakukan pendataan dengan mengakses *website* dan melakukan *login* sebelum bisa mengisi halaman *form input* data.
4. Tim survei melakukan input data korban pada halaman *form input* data korban sesuai dengan yang terjadi kemudian men-*submit* data ke dalam sistem.
5. Setelah menekan tombol *submit* data langsung diolah dengan metode pendukung keputusan SAW.
6. Data yang sudah diolah menggunakan metode SAW akan memberikan hasil keputusan yang langsung terkirim dan tersimpan ke *database*.




Gambar 3.3 *Flowmap* sistem yang akan dibangun.

### 3.4.3.Perhitungan manual yang sudah dilakukan

Untuk perhitungan manual yang sudah dilakukan dalam kasus penentuan penerima bantuan korban bencana alam gempa saat ini yaitu sebagai berikut.

1. Langkah pertama, setiap alternative ($A_i$) diisi bobot dari setiap ktriterianya ($C_i$) sesuai dengan data dan panduan pembobotan yang dibahas pada bab sebelumnya.
   Diketahui 4 data korban gempa :
   A1 : Ramedan
   A2 : H. Mahyudi
   A3 : Nuriah
   A4 : Mahyudin
   yang dimana setiap masing-masing korban gempa memiliki bobot kriteria kerusakan berbeda seperti terlihat pada Tabel 3.1 berikut.

Tabel 3.1 Pembobotan Kriteria.

| | **Kriteria** | **Keterangan kerusakan** | **A1** | **A2** | **A3** | **A4** |
|---|---|---|---|---|---|---|
| Pondasi/Slof | C1 | Retak rambut dapat terlihat jelas pada permukaan beton (lebar retakan kira-kira 0,2 – 1 mm) | 4 | 0 | 0 | 4 |
| | C2 | Retakan yang sangat jelas (lebar retakan kira-kira 1 – 2 mm) | 3 | 3 | 5 | 5 |
| | C3 | Kehancuran beton yang sangat nyata dengan tulangan beton terlihat | 0 | 8 | 8 | 5 |
| | C4 | Penurunan dan atau kemiringan lantai dapat terlihat | 0 | 11 | 0 | 0 |
| | C5 | Pondasi hancur | 0 | 0 | 11 | 0 |
| Kolom/Ring Balok | C6 | Retak rambut dapat terlihat jelas pada permukaan beton (lebar retakan kira-kira 0,2 – 1 mm) | 0 | 11 | 11 | 0 |
| | C7 | Retakan yang sangat jelas (lebar retakan kira-kira 1 – 2 mm) | 7 | 0 | 0 | 11 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | C8 | Kehancuran beton yang sangat nyata dengan tulangan beton terlihat | 0 | 0 | 12 | 19 |
| | C9 | Tulang tertekuk | 0 | 0 | 26 | 0 |
| | C10 | Deformasi vertikal/miring pada kolom dapat terlihat | 0 | 26 | 0 | 0 |
| Rangka Atap(Kuda-kuda) | C11 | Reng hancur sebagian atau seluruh | 0 | 4 | 3 | 5 |
| | C12 | Usuk hancur sebagian atau seluruh | 5 | 5 | 0 | 11 |
| | C13 | Kuda-kuda roboh sebagian atau seluruh | 0 | 0 | 14 | 0 |
| Genteng | C14 | Berjatuhan menyebar sebagian atau seluruh | 0 | 4 | 5 | 5 |
| Plafon | C15 | Langit-langit terlepas/jatuh menyebar sebagian atau seluruh | 1 | 2 | 0 | 0 |
| | C16 | Langit-langit terlepas/jatuh dan rangka langit-langit rusak menyebar sebagian atau seluruh | 0 | 0 | 4 | 4 |
| Dinding | C17 | Retak rambut pada plesteran (lebar retak < 0,2 mm) | 4 | 0 | 5 | 0 |
| | C18 | Plesteran terkelupas | 4 | 5 | 3 | 5 |
| | C19 | Retakan dapat terlihat jelas, tembus pada pasangan bata/batako (lebar retakan kira-kira > 0,3 mm) | 0 | 0 | 0 | 8 |
| | C20 | Dinding mengalami displacement horisontal/miring | 0 | 0 | 11 | 0 |
| | C21 | Dinding jatuh sebagian atau total | 0 | 0 | 11 | 0 |
| Lantai | C22 | Retak rambut menyebar sebagian atau seluruh | 3 | 3 | 0 | 0 |
| | C23 | Retak besar/ terbelah/ hancur menyebar sebagian atau seluruh | 0 | 8 | 10 | 9 |
| | **Jumlah** | | 31 | 90 | 139 | 91 |

Pada Tabel 3.1 diatas merupakan tabel pengisian bobot untuk setiap kriteria kerusakan (Ci) yang dialami oleh masing-masing korban gempa (Ai) berdasarkan panduan pembobotan pada Tabel 2.2.

2. Langkah kedua, menjumlahkan bobot dari setiap kriteria ($C_i$) pada masing-masing alternatif ($A_i$) untuk memperoleh bobot akhir kerusakan.

   A1 = C1+C2+C3+C4+C5+C6+7+C8+C9+C10+C11+C12+C13+C14+C15+C16+ C17+C18+ C19+C20+C21+C22+C23
   =4+3+0+0+0+0+7+0+0+0+0+5+0+0+1+0+4+4+0+0+0+3+0= 31

   A2 = C1+C2+C3+C4+C5+C6+7+C8+C9+C10+C11+C12+C13+C14+C15+C16+ C17+C18+ C19+C20+C21+C22+C23
   =0+3+8+11+0+11+0+0+0+26+4+5+0+4+2+0+0+5+0+0+0+3+8= 90

   A3 = C1+C2+C3+C4+C5+C6+7+C8+C9+C10+C11+C12+C13+C14+C15+C16+ C17+C18+ C19+C20+C21+C22+C23
   = 0+5+8+0+11+11+0+12+26+0+3+0+14+5+0+4+5+3+0+11+11+0+10= 139

   A4 = C1+C2+C3+C4+C5+C6+7+C8+C9+C10+C11+C12+C13+C14+C15+C16+ C17+C18+ C19+C20+C21+C22+C23
   =4+5+5+0+0+0+11+19+0+0+5+11+0+5+0+4+0+5+8+0+0+0+9= 91

   sehingga setelah dilakukan penjumlahan semua kriteria ($C_i$) dari masing-masing alternatif ($A_j$) kemudian ditentukan persentasenya dengan membagi hasil penjumlahan kriteria ($A_j$) atas total jumlah bobot maksimal keseluruhan kriteria, $\frac{31}{275}$ = 0.11 dan seterusnya sehingga masing-masing alternatif didapatkan hasil akhir seperti pada Tabel 3.2 berikut.

Tabel 3.2 Hasil perhitungan.

| Alternatif | Hasil | Persentase | Keterangan |
|---|---|---|---|
| A1 | 31 | 11% | Rusak ringan |
| A2 | 90 | 33% | Rusak sedang |
| A3 | 139 | 51% | Rusak sedang |
| A4 | 91 | 33% | Rusak sedang |

dari Tabel 3.2 dapat diambil kesimpulan bahwa korban gempa yang mengalami rusak ringan berdasarkan persentase dari hasil perhitungannya ialah A1, kemudian yang mengalami rusak sedang ialah A2, A3 dan A4.

## 3.5. Metode Simple Additive Weighting (SAW)

Dalam sistem ini digunakan metode pengambilan keputusan *Simple Additive Weighting* (SAW) sebagai metode pengambilan keputusan penentuan penerima korban bencana alam gempa, berikut langkah-langkah yang dilakukan dalam perhitungan untuk mendapatkan hasil keputusan yang terbaik:

1. Langkah pertama ialah pemberian bobot maksimal untuk setiap kriteria (Ci) dapat dilihat pada tabel 3.3.

Tabel 3.3 Penentuan bobot setiap kriteria.

| Bagian | Kriteria | Keterangan | Bobot |
|---|---|---|---|
| Pondasi | C1 | Retak rambut dapat terlihat jelas pada permukaan beton (lebar retakan kira-kira 0,2 – 1 mm) | 5 |
| | C2 | Retakan yang sangat jelas (lebar retakan kira-kira 1 – 2 mm) | 5 |
| | C3 | Kehancuran beton yang sangat nyata dengan tulangan beton terlihat | 11 |
| | C4 | Penurunan dan atau kemiringan lantai dapat terlihat | 15 |
| | C5 | Pondasi hancur | 15 |
| Kolom | C6 | Retak rambut dapat terlihat jelas pada permukaan beton (lebar retakan kira-kira 0,2 – 1 mm) | 11 |
| | C7 | Retakan yang sangat jelas (lebar retakan kira-kira 1 – 2 mm) | 11 |
| | C8 | Kehancuran beton yang sangat nyata dengan tulangan beton terlihat | 25 |
| | C9 | Tulang tertekuk | 35 |
| | C10 | Deformasi vertikal/miring pada kolom dapat terlihat | 35 |
| Rangka Atap | C11 | Reng hancur sebagian atau seluruh | 5 |
| | C12 | Usuk hancur sebagian atau seluruh | 11 |

| | | | |
|---|---|---|---|
| | C13 | Kuda-kuda roboh sebagian atau seluruh | 15 |
| Penutup atap | C14 | Berjatuhan menyebar sebagian atau seluruh | 5 |
| Plafon | C15 | Langit-langit terlepas/jatuh menyebar sebagian atau seluruh | 2 |
| | C16 | Langit-langit terlepas/jatuh dan rangka langit-langit rusak menyebar sebagian atau seluruh | 5 |
| Dinding | C17 | Retak rambut pada plesteran (lebar retak < 0,2 mm) | 5 |
| | C18 | Plesteran terkelupas | 5 |
| | C19 | Retakan dapat terlihat jelas, tembus pada pasangan bata/batako (lebar retakan kira-kira > 0,3 mm) | 11 |
| | C20 | Dinding mengalami displacement horisontal/miring | 15 |
| | C21 | Dinding jatuh sebagian atau total | 15 |
| Lantai | C22 | Retak rambut menyebar sebagian atau seluruh | 3 |
| | C23 | Retak besar/ terbelah/ hancur menyebar sebagian atau seluruh | 10 |

Dari kriteria diatas dapat dibuat suatu tingkatan kriteria yang masing-masing kriteria diatas memiliki nilai masing-masing yang dapat dilihat pada tabel 3.4 sampai tabel 3.26.

Tabel 3.4 kriteria 1 (C1).

| Jumlah komponen | nilai |
|---|---|
| 1 | 3 |
| 2 | 4 |
| ≥3 | 5 |

Tabel 3.5 kriteria 2 (C2).

| Jumlah komponen | nilai |
|---|---|
| 1 | 3 |
| 2 | 4 |
| ≥3 | 5 |

Tabel 3.6 kriteria 3 (C3).

| Jumlah komponen | nilai |
|---|---|
| 1 | 5 |
| 2 | 8 |
| ≥3 | 11 |

Tabel 3.7 kriteria 4 (C4).

| Jumlah komponen | nilai |
|---|---|
| 1 | 11 |
| 2 | 14 |
| ≥3 | 15 |

Tabel 3.8 kriteria 5 (C5).

| Jumlah komponen | nilai |
|---|---|
| 1 | 11 |
| 2 | 14 |
| ≥3 | 15 |

Tabel 3.9 kriteria 6 (C6).

| Jumlah komponen | nilai |
|---|---|
| 1 | 7 |
| 2 | 9 |
| ≥3 | 11 |

Tabel 3.10 kriteria 7 (C7).

| Jumlah komponen | nilai |
|---|---|
| 1 | 7 |
| 2 | 9 |
| ≥3 | 11 |

Tabel 3.11 kriteria 8 (C8).

| Jumlah komponen | nilai |
|---|---|
| 1 | 12 |
| 2 | 19 |
| ≥3 | 25 |

Tabel 3.12 kriteria 9 (C9).

| Jumlah komponen | nilai |
|---|---|
| 1 | 26 |
| 2 | 32 |
| ≥3 | 35 |

Tabel 3.13 kriteria 10 (C10).

| Jumlah komponen | nilai |
|---|---|
| 1 | 26 |
| 2 | 32 |
| ≥3 | 35 |

Tabel 3.14 kriteria 11 (C11).

| Jumlah komponen | nilai |
|---|---|
| 1 | 3 |
| 2 | 4 |
| ≥3 | 5 |

Tabel 3.15 kriteria 12 (C12).

| Jumlah komponen | nilai |
|---|---|
| 1 | 5 |
| 2 | 8 |
| ≥3 | 11 |

Tabel 3.16 kriteria 13 (C13).

| Jumlah komponen | nilai |
|---|---|
| 1 | 11 |
| 2 | 14 |
| ≥3 | 15 |

Tabel 3.17 kriteria 14 (C14).

| Jumlah komponen | nilai |
|---|---|
| 1 | 4 |
| ≥2 | 5 |

Tabel 3.18 kriteria 15 (C15).

| Jumlah komponen | nilai |
|---|---|
| ≤2 | 1 |
| ≥3 | 2 |

Tabel 3.19 kriteria 16 (C16).

| Jumlah komponen | nilai |
|---|---|
| 1 | 4 |
| ≥2 | 5 |

Tabel 3.20 kriteria 17 (C17).

| Jumlah komponen | nilai |
|---|---|
| 1 | 3 |
| 2 | 4 |
| ≥3 | 5 |

Tabel 3.21 kriteria 18 (C18).

| Jumlah komponen | nilai |
|---|---|
| 1 | 3 |
| 2 | 4 |
| ≥3 | 5 |

Tabel 3.22 kriteria 19 (C19).

| Jumlah komponen | nilai |
|---|---|
| 1 | 5 |
| 2 | 8 |
| ≥3 | 11 |

Tabel 3.23 kriteria 20 (C20).

| Jumlah komponen | nilai |
|---|---|
| 1 | 11 |
| 2 | 14 |
| ≥3 | 15 |

Tabel 3.24 kriteria 21 (C21).

| Jumlah komponen | nilai |
|---|---|
| 1 | 11 |
| 2 | 14 |
| ≥3 | 15 |

Tabel 3.25 kriteria 22 (C22).

| Jumlah komponen | nilai |
|---|---|
| 1 | 2 |
| ≥3 | 3 |

Tabel 3.26 kriteria 23 (C23).

| Jumlah komponen | nilai |
|---|---|
| 1 | 8 |
| 2 | 9 |
| ≥3 | 10 |

Pada Tabel 3.4 sampai dengan 3.26 merupakan detail penilain bobot sesuai jumlah komponen yang rusak dari setiap keriteria kerusakan yang ada.

Untuk perhitungan nilai perbaikan bobot yang ditentukan ($W_j$) berdasarkan pada ketentuan nilai bobot yang telah ditetapkan dari setiap kriteria pada tabel 3.3 ialah sebagai berikut.

$$W_1 = \frac{5}{5+5+11+15+15+11+11+25+35+35+5+11+15+5+2+5+5+5+11+15+15+3+10} = 0.0182$$

Dan seterusnnya dilakukan perhitungan untuk setiap masing-masing bobotnya sehingga diperoleh nilai bobot perbaikan yang dapat dilihat pada tabel 3.27.

Tabel 3.27 Nilai Bobot Kriteria ($W_i$).

| Kriteria | Bobot ($W_i$) |
|---|---|
| C1 | 0.0182 |
| C2 | 0.0182 |
| C3 | 0.04 |
| C4 | 0.0545 |
| C5 | 0.0545 |
| C6 | 0.04 |
| C7 | 0.04 |
| C8 | 0.0909 |
| C9 | 0.1273 |
| C10 | 0.1273 |
| C11 | 0.0182 |
| C12 | 0.04 |
| C13 | 0.0545 |
| C14 | 0.0182 |
| C15 | 0.0073 |
| C16 | 0.0182 |
| C17 | 0.0182 |
| C18 | 0.0182 |
| C19 | 0.04 |
| C20 | 0.0545 |
| C21 | 0.0545 |
| C22 | 0.0109 |
| C23 | 0.0364 |

Pada Tabel 3.27 merupakan tabel perbaikan nilai bobot dari bobot yang ditentukan sebelumnya pada tabel 3.3, hasil nilai perbaikan bobot inilah yang akan digunakan sebagai nilai ($W_i$) dalam perhitungan dengan menggunakan metode SAW pada langkah selanjutnya.

2. Langkah kedua ialah menentukan rating kecocokan setiap aternatif dan kriteria.

   Pada langkah ini digunakan empat alternatif data yang diperoleh dari lembaga yang dapat dilihat pada tabel 3.28.

Tabel 3.28 Rating kecocokan alternatif dan kriteria.

| Ai | Kriteria(Ci) | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | C13 | C14 | C15 | C16 | C17 | C18 | C19 | C20 | C21 | C22 | C23 |
| A1 | 4 | 3 | 0 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 1 | 0 | 4 | 4 | 0 | 0 | 0 | 3 | 0 |
| A2 | 0 | 3 | 8 | 11 | 0 | 11 | 0 | 0 | 0 | 26 | 4 | 5 | 0 | 4 | 2 | 0 | 0 | 5 | 0 | 0 | 0 | 3 | 8 |
| A3 | 0 | 5 | 8 | 0 | 11 | 11 | 0 | 12 | 26 | 0 | 3 | 0 | 14 | 5 | 0 | 4 | 5 | 3 | 0 | 11 | 11 | 0 | 10 |
| A4 | 4 | 5 | 5 | 0 | 0 | 0 | 11 | 19 | 0 | 0 | 5 | 11 | 0 | 5 | 0 | 4 | 0 | 5 | 8 | 0 | 0 | 0 | 9 |

3. Langkah ketiga ialah pembentukan matriks keputusan atau normalisasi dari matriks keputusan ($R_{ij}$)

   Berdasarkan pada tabel 3.28 rating kecocokan alternatif dan kriteria dibentuk sebuah matriks keputusan X.

$$X=\begin{Bmatrix} 4 & 3 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 1 & 0 & 4 & 4 & 0 & 0 & 0 & 3 & 0 \\ 0 & 3 & 8 & 11 & 0 & 11 & 0 & 0 & 0 & 26 & 4 & 5 & 0 & 4 & 2 & 0 & 0 & 5 & 0 & 0 & 0 & 3 & 8 \\ 0 & 5 & 8 & 0 & 11 & 11 & 0 & 12 & 26 & 0 & 3 & 0 & 14 & 5 & 0 & 4 & 5 & 3 & 0 & 11 & 11 & 0 & 10 \\ 4 & 5 & 5 & 0 & 0 & 0 & 11 & 19 & 0 & 0 & 5 & 11 & 0 & 5 & 0 & 4 & 0 & 5 & 8 & 0 & 0 & 0 & 9 \end{Bmatrix}$$

   Dari matriks keputusan X dilakukan normalisasi menjadi matriks ternormalisasi ($R_{rij}$) dengan perhitungan sebagai berikut.

$$R_{11} = \frac{x_{11}}{Max\ x_{11}} = \frac{4}{Max\ \{\ 4\ \ 0\ \ 0\ \ 4\ \}} = 1$$

Dan seterusnya dilakukan perhitungan yang sama untuk *index* $R_{ij}$ selanjutnya sehingga diperoleh matriks ternormalisasi ($R_{ij}$) sebagai berikut.

$$R_{ij} = \begin{Bmatrix} 1 & 0.6 & 0 & 0 & 0 & 0 & 0.6364 & 0 & 0 & 0 & 0 & 0.4545 & 0 & 0 & 0.5 & 0 & 0.8 & 0.8 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0.6 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0.8 & 0.4545 & 0 & 0.8 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0.8 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0.6316 & 1 & 0 & 0.6 & 0 & 1 & 1 & 0 & 1 & 1 & 0.6 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0.625 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0.9 \end{Bmatrix}$$

4. Langkah keempat ialah proses perangkingan.

Pada langkah ini dilakukan perhitungan dari setiap hasil normalisasi kriteria yang telah diperoleh untuk dikalikan dengan bobot dari setiap kriteria yang telah ditentukan sehingga akan diperoleh hasil nilai perangkingan dari setiap alternatif ($V_i$) yang ada, perhitungan yang dilakukan sebagai berikut.

$V_i = (R_{ij})*(W_i)$

$V_1$ = (1)*(0.0182) + (0.6)*(0.0182) + (0)*(0.04) + (0)*(0. 0545) + (0)*(0. 0545) + (0)*(0.04) + (0.6364)*(0.04) + (0)*(0.0909) + (0)*(0.1273) + (0)*(0.1273) + (0)*(0.0182) + (0.4545)*(0.04) + (0)*(0.0545) + (0)*(0.0182) + (0.5)*(0.0073) + (0)*(0.0182) + (0.8)*(0.0182) + (0.8)*(0.0182) + (0)*( 0.04) + (0)*(0.0545) + (0)*(0.0545) + (1)*(0.0109) + (0)*(0.0364) = 0.1164

Dan seterusnya dilakukan perhitungan yang sama untuk Vi selanjutnya sampai didapatkan hasil perangkingan yang dapat dilihat pada Tabel 3.29 berikut.

Tabel 3.29 Hasil perangkingan.

| Alternatif | Nilai | keterangan |
|---|---|---|
| A1 | 0.1164 | Rusak ringan |
| A2 | 0.3855 | Rusak sedang |
| A3 | 0.6137 | Rusak sedang |
| A4 | 0.3778 | Rusak sedang |

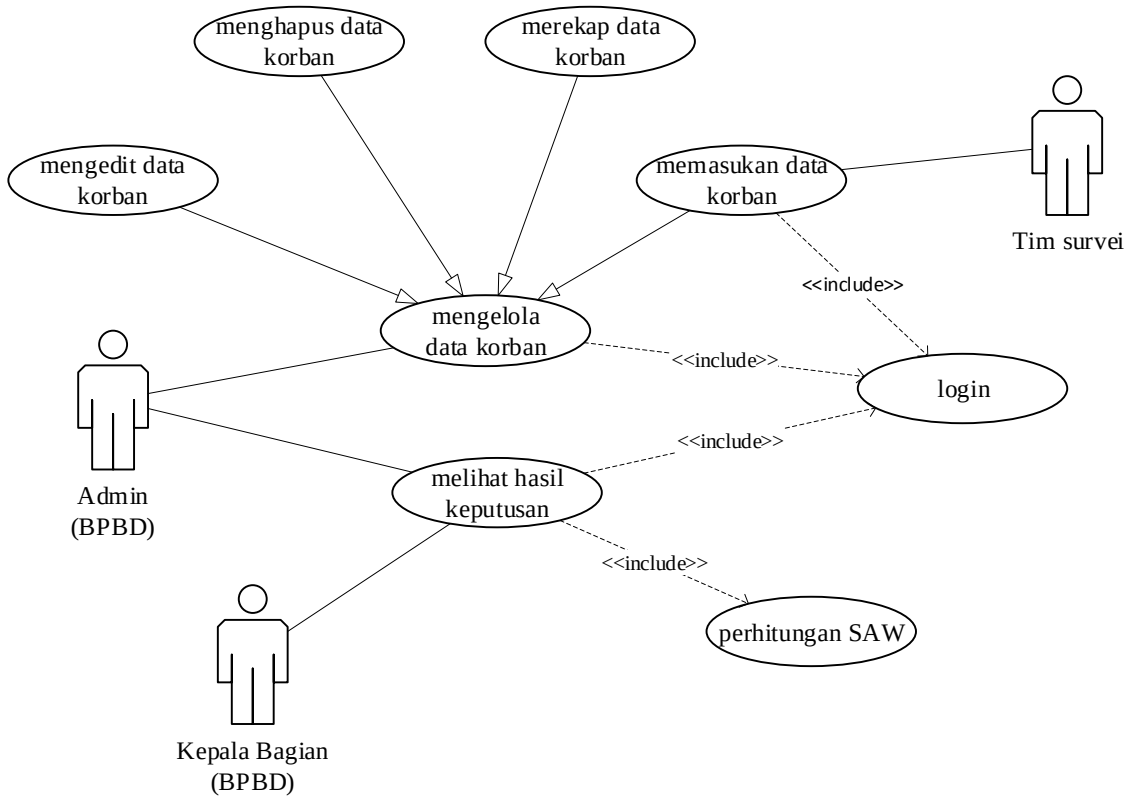
dari hasil perangkingan tersebut diperoleh data hasil korban gempa yang mengalami rusak ringan ialah A1 , kemudian korban gempa yang mengalami rusak sedang A2,A3 dan A4.

## 3.6. Perancangan Sistem

Setelah tahapan analisis sistem dan mengetahui kebutuhan dari sistem yang akan dibangun selanjutnya ialah tahapan perancangan dari sistem, pada tahapan perancangan ini akan dibuat desain dari sistem yang memberikan rancangan bagaimana *usecase* diagram guna menjelaskan interaksi antara *user* dari sistem dengan sistem itu sendiri, selanjutnya memberikan rancangan bagaimana *class diagram* dari sistem yang menjelaskan struktur dan deskripsi *class package* dan objek dari sitem yang akan dibangun, memberikan rancangan *sequence diagram* yang menjelaskan interaksi antar objeck di dalam dan sekitar sistem, memberikan rancangan *activity diagram* yang menjelaskan alir aktivitas dalam sistem, dan memberikan rancangan ERD dari sistem yang akan dibangun.

### 3.5.1.*Use Case Diagram*

Untuk penggambaran *use case diagram* dapat dilihat pada gambar 3.4.




Gambar 3.4 Use case diagram.

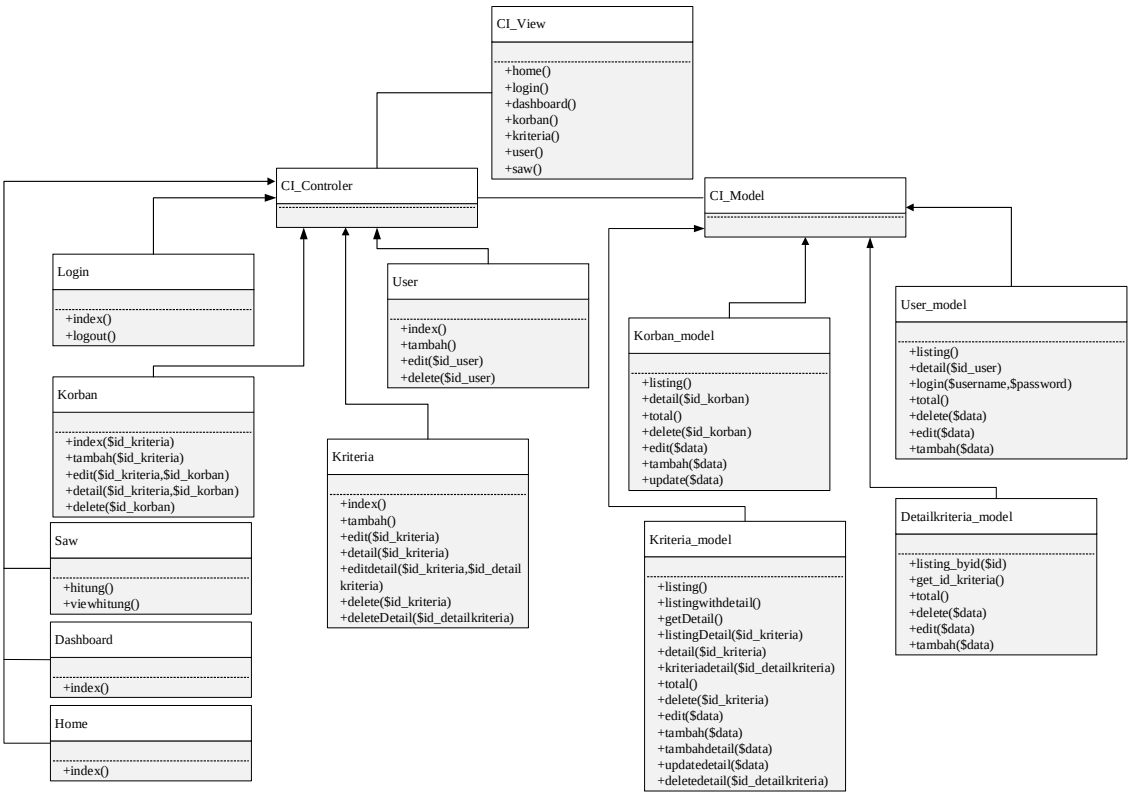
Pada Gambar 3.4 dapat diketahui bahwa terdapat tiga *user* yang akan menggunakan sistem yakni *user admin*, *user* tim survei dan *user* Kabag, hal-hal yang dapat dilakukan oleh masing-masing *user* diantaranya :

1. *Admin* , merupakan *user* yang dapat melakukan memasukan , mengedit, menghapus, dan merekap data para korban yang telah di-*input*-kan oleh tim survei, sebelum itu *admin* harus terlebih dahulu melakukan *login*, kemudian *admin* juga bisa melakukan melihat hasil keputusan kategori penerima bantuan.
2. Tim survei, merupakan *user* yang hanya dapat melakukan memasukan data para korban bencana alam gempa dengan melakukan *login* terlebih dahulu.
3. Kepala bagian, merupakan *user* yang hanya dapat melihat data hasil dari perhitungan data korban yang telah di-*input*-kan ke dalam sistem.

### 3.5.2.*Class Diagram*

Dalam perancangan sistem ini menggunakan pola arsitektur *Model View Controller* (MVC). Sehingga dilakukan perancangan kelas-kelas yang ada pada sistem yang terdiri dari

kelas *Model* , *View* , dan *Controller* yang saling berelasi satu sama lain seperti terlihat pada gambar 3.5.




Gambar 3.5 *Class diagram*.

Pada Gambar 3.5 dapat dilihat terdapat beberapa kelas yang dirancang terdiri dari kelas *view*, *controller* dan *model* , untuk kelas *controller* ada 7 kelas diantaranya *Login*, *Korban*, *Kriteria*, *User* ,*Saw*, *Dashboard* dan *Home*. Untuk kelas *model* ada 4 diantaranya *Korban_model*, *Kriteria_model*, *Detailkriteria_model* dan *User_model*.

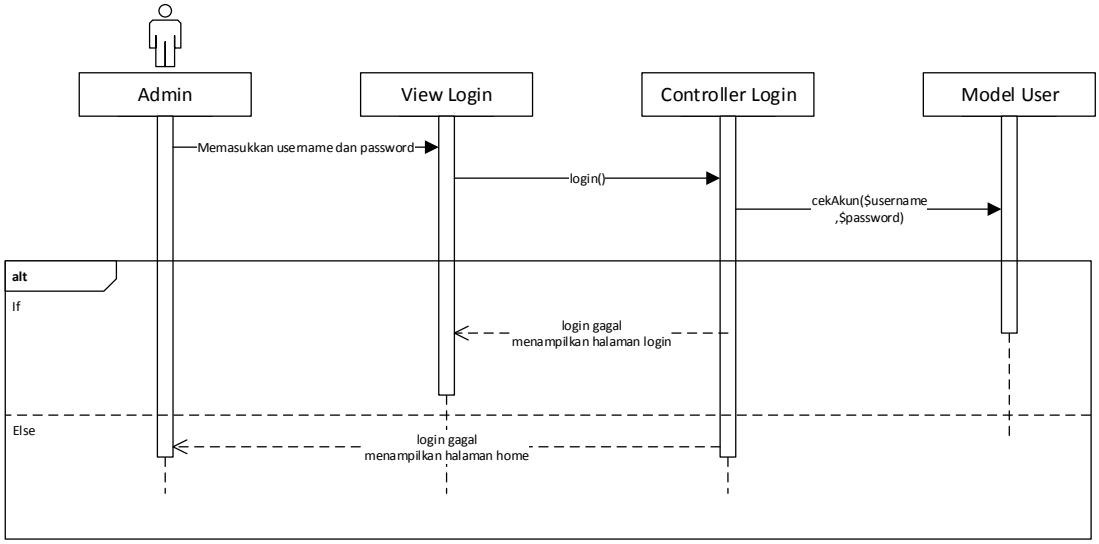
### 3.5.3.*Sequence Diagram*

Untuk perancangan *sequence diagram* dari sistem yang akan dibangun adalah sebagai berikut.

*1. Sequence diagram admin login*

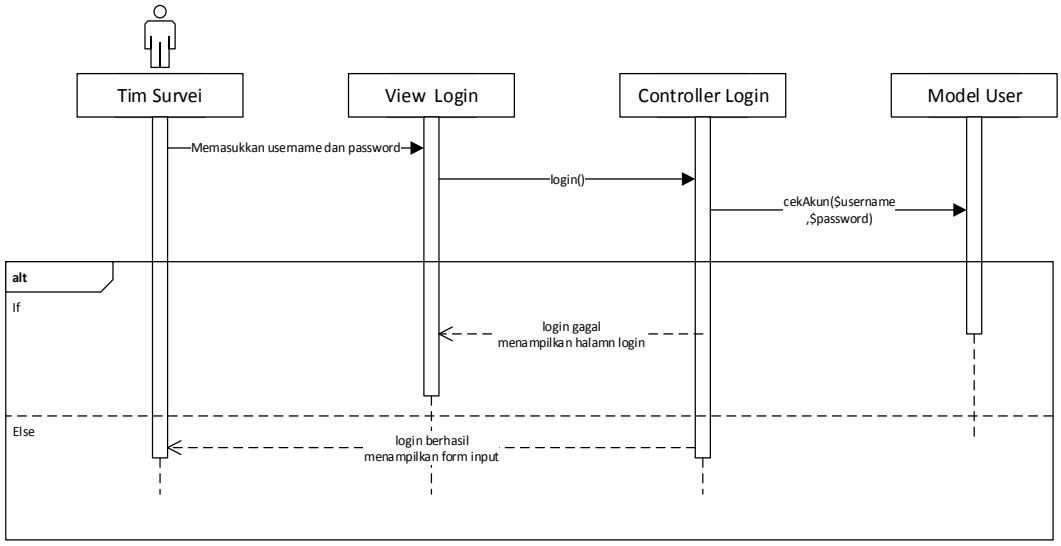
Pada Gambar 3.6 merupakan *sequence diagram* dari proses *login admin*. Proses bermula dari *view login*, *admin* memasukkan *username* dan *password* lalu memanggil fungsi `index()` pada *controller Login*. Kemudian memanggil fungsi `login()` pada *model User_model*. Dari *model* mengirimkan nilai *login* berhasil atau gagal, bila gagal akan

kembali ditampilkan *form login* dan apabila berhasil *controller* menampilkan halaman *dashboard*.




Gambar 3.6 *Sequence diagram admin login.*

2. *Sequence diagram* tim survei *login*

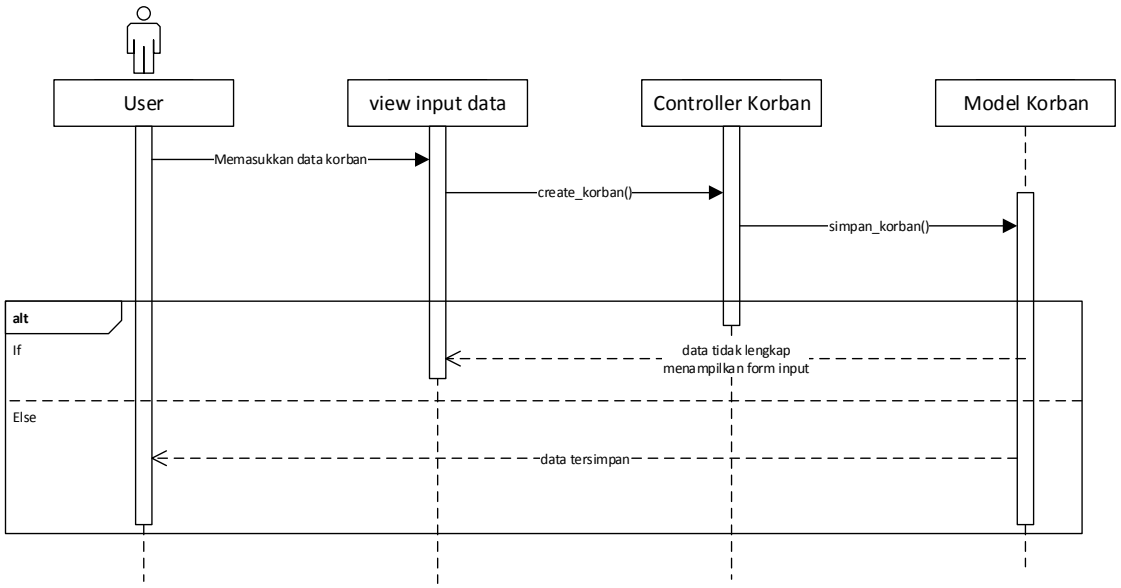



Gambar 3.7 *Sequence diagram tim survei login*.

Gambar 3.7 merupakan *sequence diagram* dari proses *login* tim survei. Proses bermula dari *view login*, tim survei memasukkan *username* dan *password* lalu memanggil fungsi `index()` pada *controller Login*. Kemudian memanggil fungsi `login()` pada *model User_model*. Dari *model* mengirimkan nilai *login* berhasil atau gagal, bila gagal akan kembali ditampilkan *view login* dan apabila berhasil *controller* menampilkan halaman *input* data.

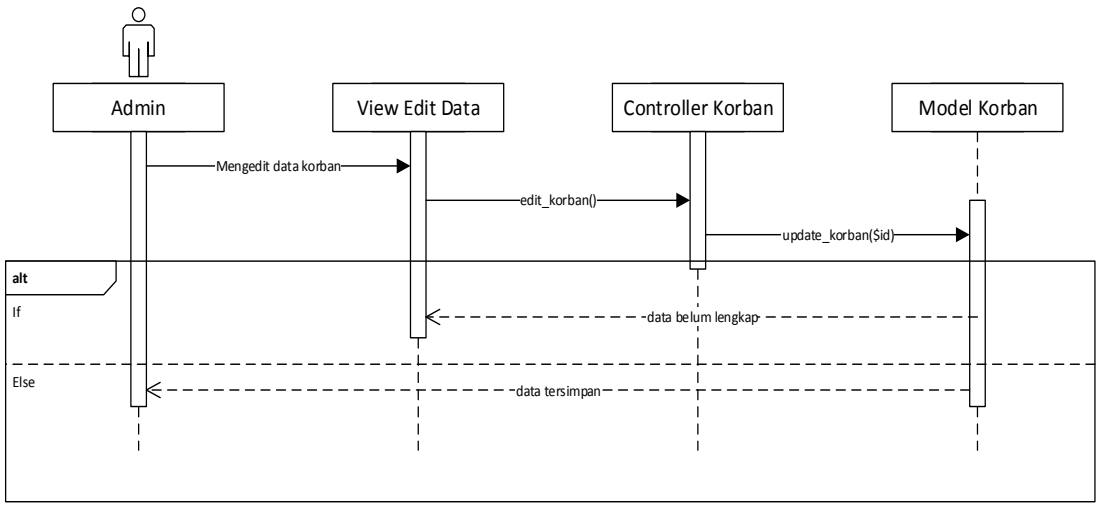
3. *Sequence diagram* memasukkan data korban

Pada Gambar 3.8 merupakan *sequence diagram* dari proses *user admin* maupun *user* tim survei memasukkan data korban. Proses bermula dari *form input*, *admin* memasukkan data korban lalu memanggil fungsi `tambah()` pada *controller Korban*. Kemudian memanggil fungsi `tambah()` pada *model Korban_model*. Dari *model* mengirimkan data ke *database* untuk disimpan.




Gambar 3.8 *Sequence diagram* memasukkan data korban.

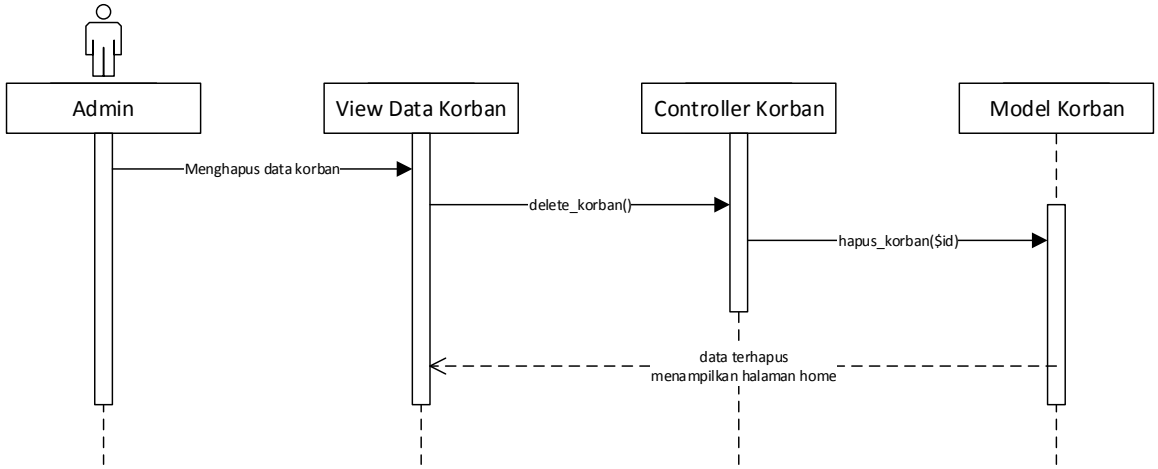
4. *Sequence diagram admin* mengedit data korban




Gambar 3.9 *Sequence diagram admin* mengedit data korban.

Pada Gambar 3.9 merupakan *sequence diagram* dari proses *admin* mengedit data korban. Proses bermula dari *form edit*, *admin* mengedit data korban lalu memanggil fungsi `edit()` pada *controller Korban*. Kemudian memanggil fungsi `update($id_korban)` pada *model Korban_model*. Dari *model* mengirimkan data yang sudah diedit ke *database* untuk disimpan.

5. *Sequance diagram admin* menghapus data korban




Gambar 3.10 *Sequence diagram admin* menghapus data korban.

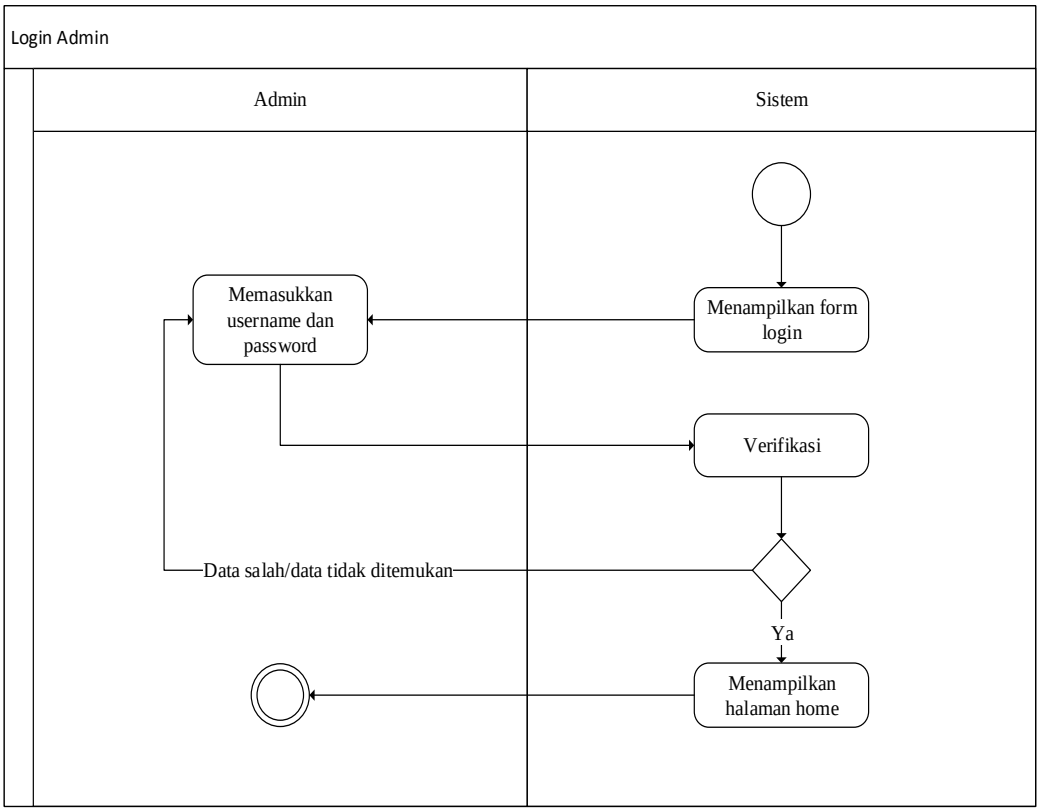
Gambar 3.10 merupakan *sequence diagram* dari proses menghapus data korban. Proses bermula dari *view data korban* , *admin* menghapus data korban lalu memanggil fungsi `delete()` pada *controller Korban*. Kemudian memanggil fungsi `delete($data)` pada *model Korban_model*. Dari *model* mengirimkan *query* untuk menghapus data berdasarkan id data pada *database* lalu data terhapus dan kembali menampilkan halaman *home.*

### 3.5.4.*Activity Diagram*

Berikut merupakan perancangan *activity diagram* dari sistem yang akan dibangun yang dimana berfungsi menggambarkan aliran kerja atau aktivitas dari sistem.

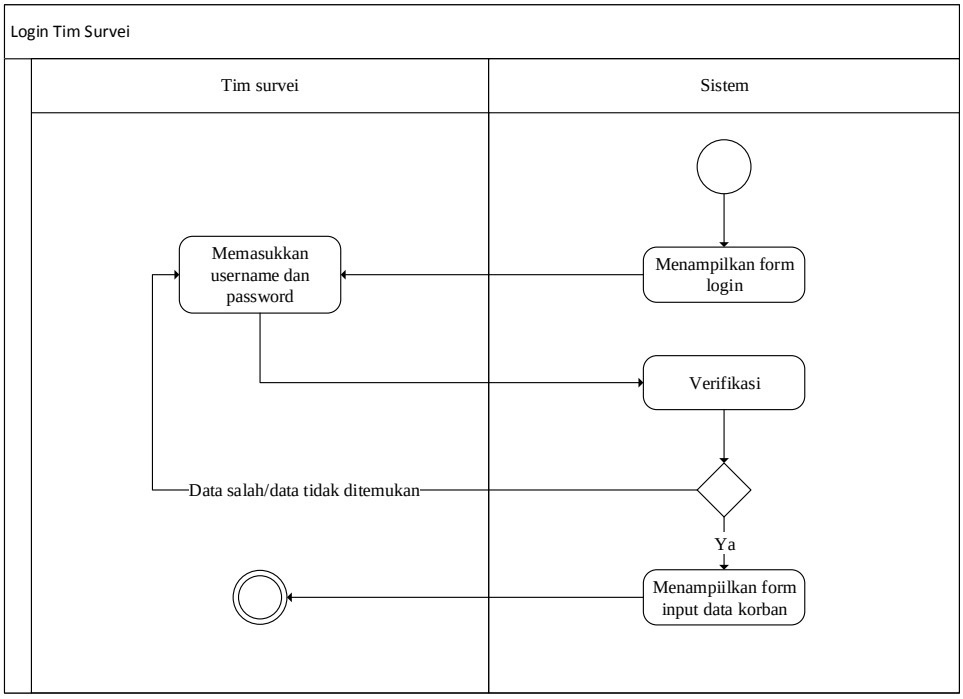
1. Proses *login admin*

Gambar 3.11 merupakan proses *login admin* yang dimulai dari sistem menampilkan *form login*. Kemudian *admin* memasukkan *username* dan *password* setelah itu sistem memverifikasi data yang telah dimasukkan apabila data ada dan valid maka sistem menampilkan halaman *home* apabila data tidak ditemukan atau data yang diisi salah maka *admin* harus memasukkan ulang *username* dan *password* yang sesuai.




Gambar 3.11 *Activity diagram login admin.*

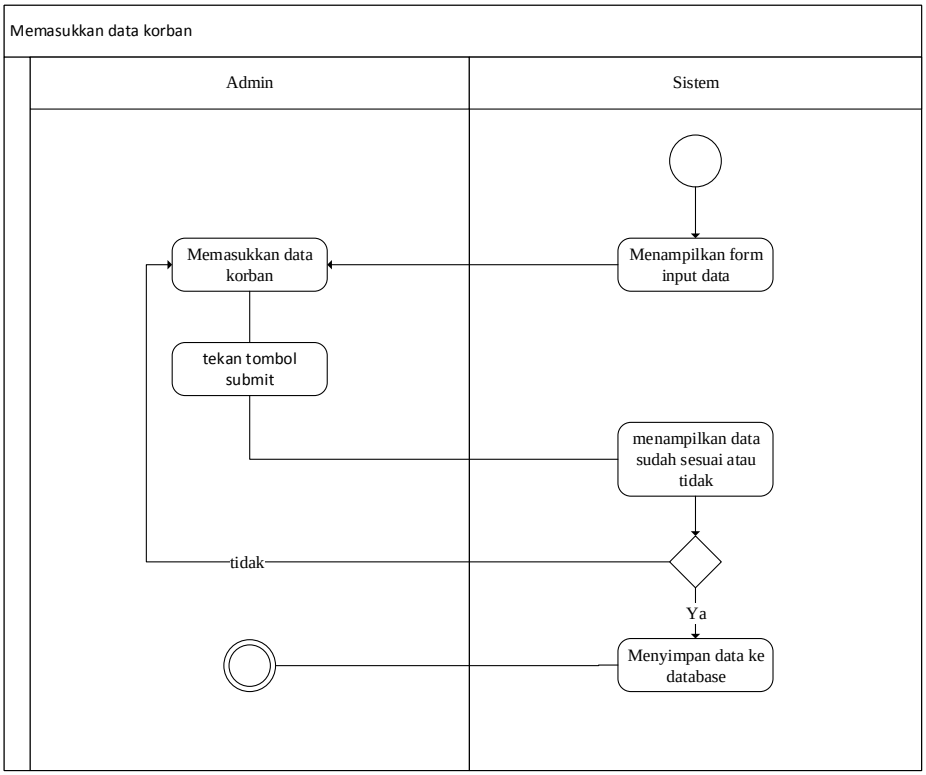
2. Proses *login* tim survei




Gambar 3.12 *Activity diagram login* tim survei.

Pada Gambar 3.12 merupakan proses *login* tim survei yang dimulai dari sistem menampilkan form *login*. Kemudian tim survei memasukkan *user*name dan password setelah itu sistem memverifikasi data yang telah dimasukkan apabila data ada dan valid maka sistem menampilkan form input apabila data tidak ditemukan atau data yang diisi salah maka tim survei harus memasukkan ulang *user*name dan password yang sesuai.

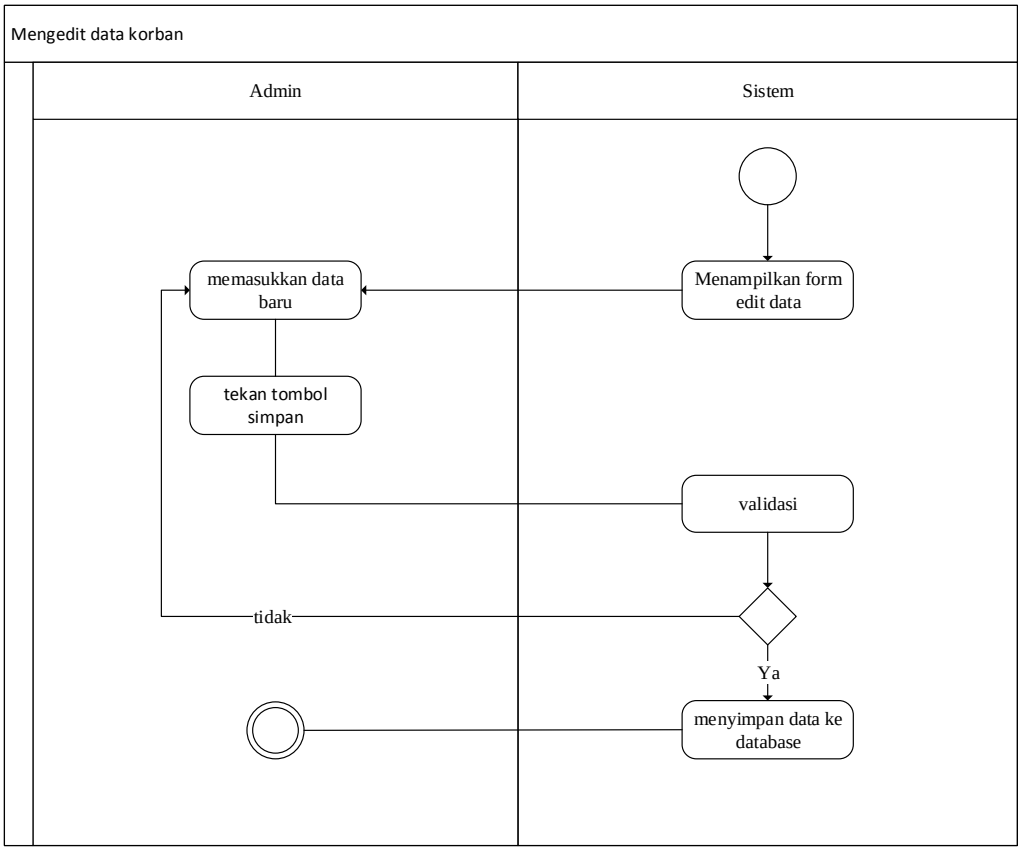
3. Proses memasukkan data korban




Gambar 3.13 Activity diagram memasukkan data korban.

Pada Gambar 3.13 merupakan proses memasukkan data korban. Proses ini berawal dari sistem menampilkan *form input* lalu *user* memasukkan data korban sesuai yang ada pada *form*. Data divalidasi, apabila data sudah benar dan lengkap maka data disimpan ke dalam *database* jika data tidak lengkap atau data salah *user* harus kembali mengisi data korban yang sesuai.

4. Proses edit data korban

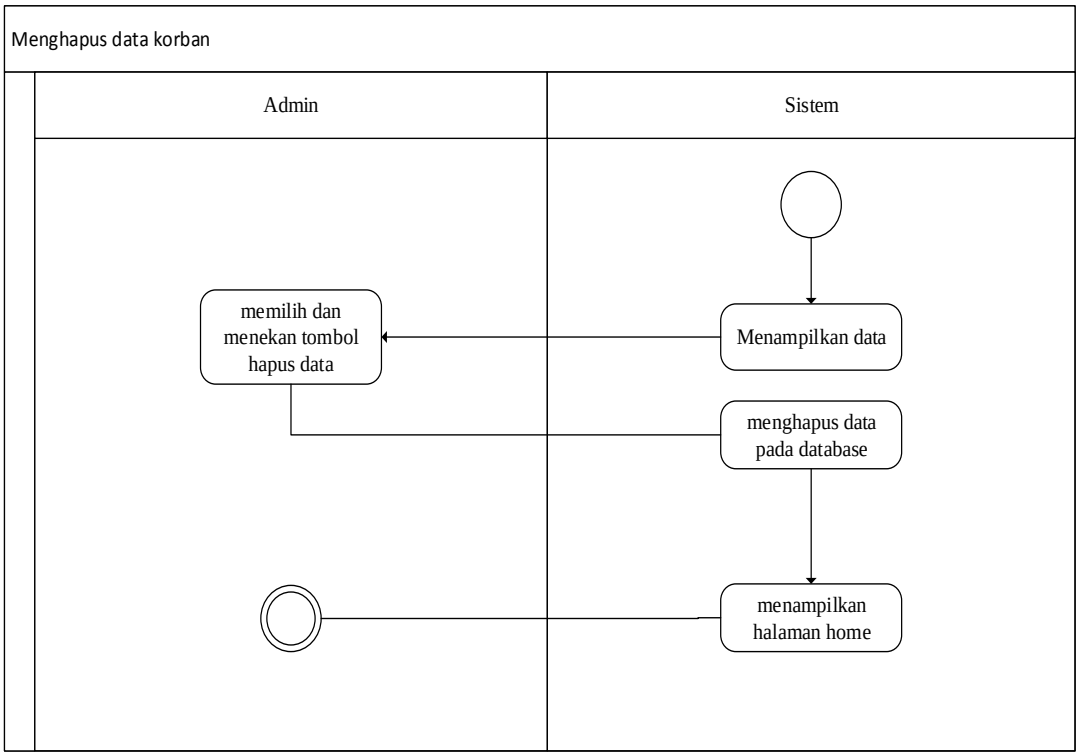



Gambar 3.14 *Activity diagram* edit data korban.

Gambar 3.14 merupakan proses *edit* data. Proses ini berawal dari sistem menampilkan *form edit* data korban kemudian *admin* mengisi data yang ingin di-*edit*. Data di validasi, apabila data yang telah dimasukkan telah benar dan lengkap maka data disimpan ke *database* dan sistem menampilkan halaman *home* jika data tidak lengkap maka *admin* kembali ke tahap memasukkan data baru.

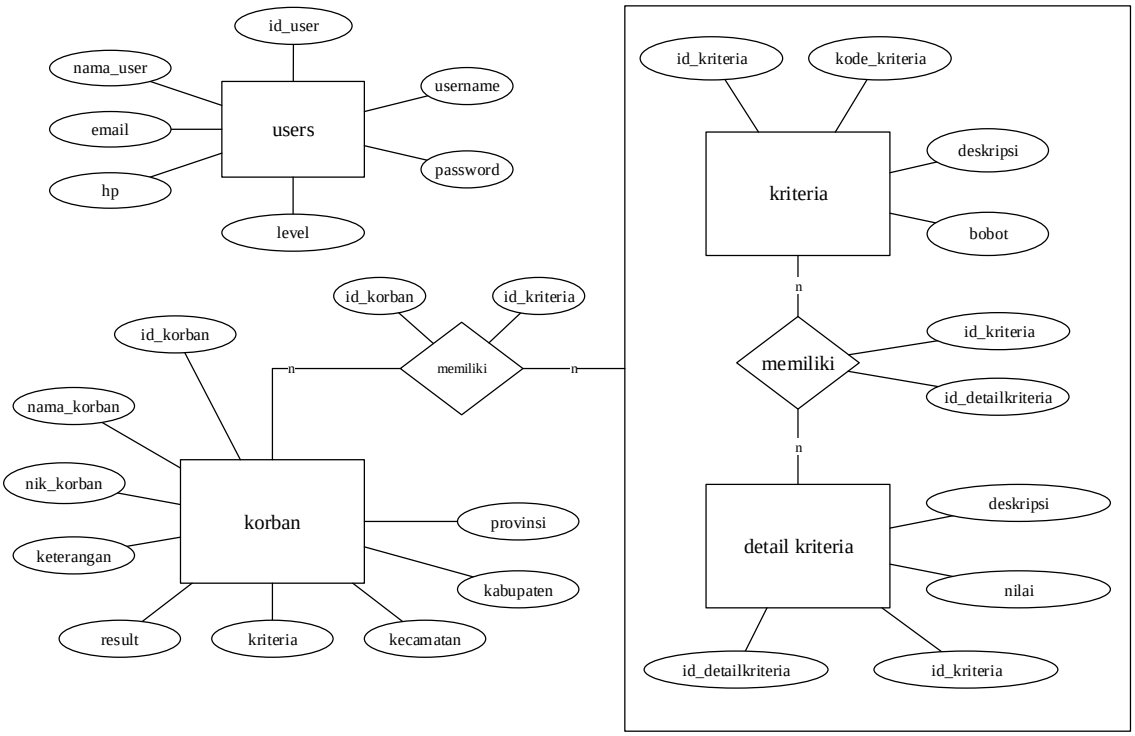
5. Proses hapus data korban

Pada gambar 3.15 merupakan proses hapus data korban. Proses hapus data korban berawal dari sistem menampilkan halaman *home* kemudian *user* mengklik hapus data pada data korban yang akan dihapus. Data akan terhapus dari *database* dan sistem kembali menampilkan halaman *home*.




Gambar 3.15 *Activity diagram* menghapus data korban.

### 3.5.5.*Entity Relationship Diagram*




Gambar 3.16 *Entity relationship diagram.*

*Entity Relation Diagram* (ERD) merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel data dalam sistem. Didalam sistem penentuan penerima korban bencana alam gempa ini dirancang ERD yang terdiri dari empat entitas yaitu *users*, korban, kriteria dan detailkriteria.

Detail atribut dari setiap entitas dan relasi ERD dapat dilihat pada Tabel 3.1 sampai dengan 3.6.

Tabel 3.30 Entitas *users*.

| Field | Type | Null | Default |
|---|---|---|---|
| id_*user* | int (11) | No | None |
| nama_*user* | Varchar (50) | No | None |
| email | Varchar(50) | No | None |
| hp | Varchar(15) | No | None |
| *user*name | Varchar (32) | No | None |
| password | Varchar(64) | No | None |
| level | Varchar(20) | No | None |

Tabel 3.31 Entitas korban.

| Field | Type | Null | Default |
|---|---|---|---|
| id_korban | inst (11) | No | None |
| nama_korban | Varchar (50) | No | None |
| nik_korban | Varchar (50) | No | None |
| provinsi | Varchar (50) | No | None |
| kabupaten | Varchar(50) | No | None |
| kecamatan | Varchar (50) | No | None |
| kriteria | text | No | None |
| result | float | No | None |
| keterangan | Varchar(32) | No | None |

Tabel 3.32 Entitas kriteria.

| Field | Type | Null | Default |
|---|---|---|---|
| id_kriteria | int (11) | No | None |
| Kode_kriteria | Varchar(20) | No | None |

| | | | |
|---|---|---|---|
| deskripsi | text | No | None |
| bobot | float | No | None |

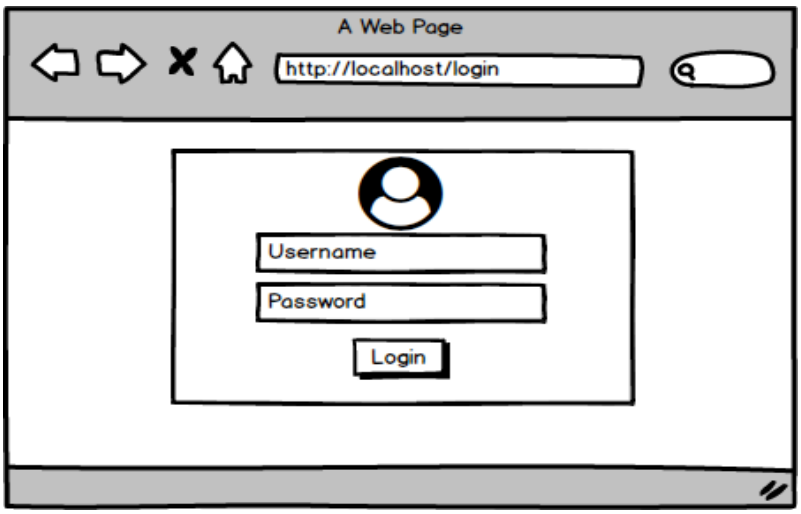
Tabel 3.33 Entitas detailkriteria.

| Field | Type | Null | Default |
|---|---|---|---|
| id_detailkriteria | int (11) | No | None |
| deskripsi | Varchar(255) | No | None |
| bobot | float | No | None |
| id_kriteria | Int(11) | No | None |

### 3.5.6.Desain Tampilan Sistem

Berikut merupakan tampilan desain dari sistem pendukung keputusan penentuan korban bencana alam gempa :

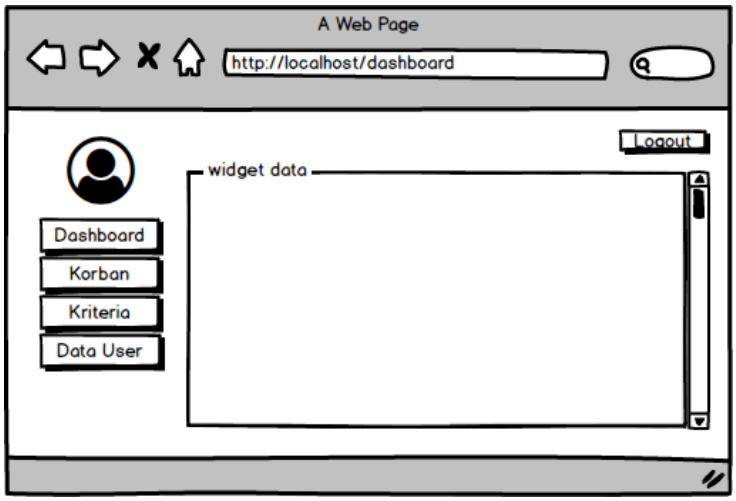
1. Halaman *login*




Gambar 3.17 Desain halaman login.

Gambar 3.17 merupakan desain dari halaman *login*. Halaman *login* merupakan halaman yang digunakan timsur dan *admin* (lembaga) untuk masuk ke dalam sistem. Pada *form* tersebut terdapat kotak isian *username* dan *password* , beserta tombol *Login.*

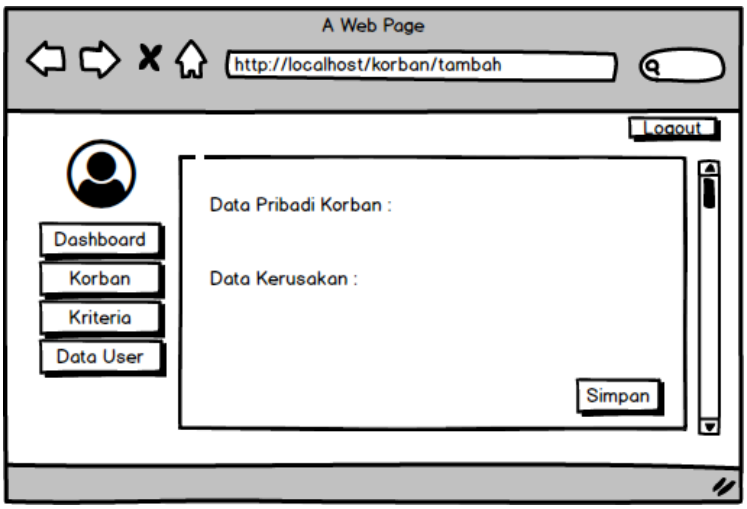
2. Halaman *dashboard*




Gambar 3.18 Desain halaman dashboard .

Gambar 3.18 merupakan desain dari halaman *dashboard*. Halaman *dashboard* merupakan halaman yang akan muncul setelah *admin* melakukan *login* dan sudah masuk ke dalam sistem. Terdapat *sidebar* menu *dashboard*, data korban, data kriteria dan data *user*. Pada halaman *dashboard* ini ditampilkan data korban yang telah di-*input* oleh tim survei dalam bentuk *widget* keterangan total data.

3. Halaman *input* data

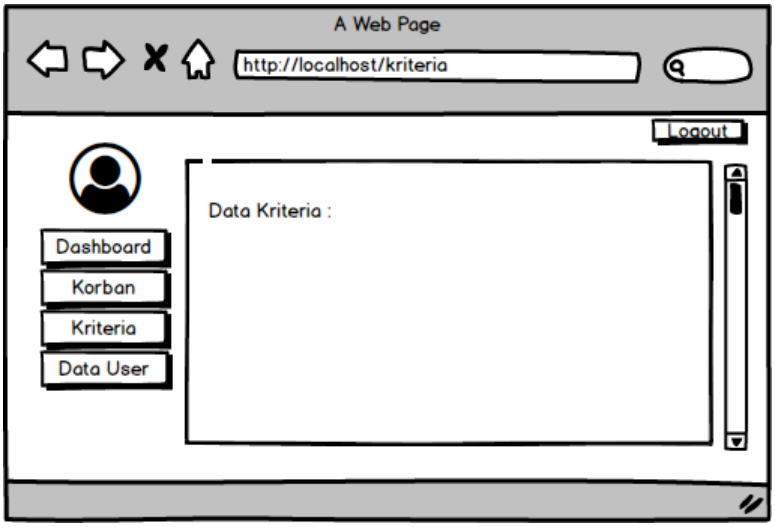



Gambar 3.19 Desain halaman input data.

Gambar 3.19 merupakan desain dari halaman *input* data. Halaman *input* data merupakan halaman yang akan muncul setelah tim survei melakukan *login* kedalam sistem. Halaman ini

terdapat sebuah form terdiri dari data pribadi dan data kerusakan yang akan di isi oleh tim survei dan tombol *simpan* untuk menyimpan data korban ke *database*.

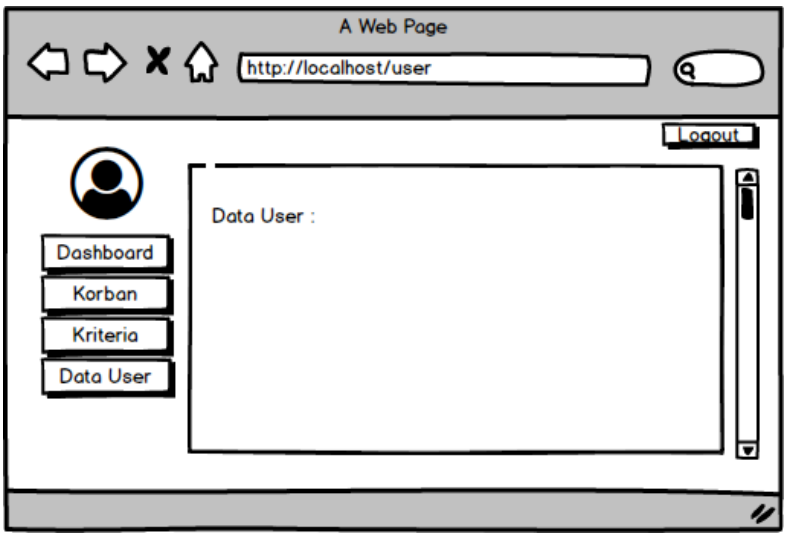
4. Halaman data kriteria




Gambar 3.20 Desain halaman data kriteria.

Gambar 3.20 merupakan desain dari halaman data kriteria. Halaman ini merupakan halaman yang menampilkan data kriteria yang digunakan dalam perhitungan keputusan dan akan ditampilkan dalam bentuk tabel data yang bisa dilakukan edit dan hapus.

5. Halaman data *user*




Gambar 3.21 Desain halaman data *user*.

Gambar 3.21 merupakan desain dari halaman data *user*. Halaman ini merupakan halaman yang menampilkan data *user* yang bisa digunakan untuk melalukan *login* kedalam sistem baik menjadi *user admin* maupun *user* tim survei.

## 3.7. Pembuatan Sistem

Tahapan pembuatan sistem adalah proses implementasi rancangan sistem yang sudah dilakukan ke dalam bentuk bahasa pemrograman (*coding*). Pada proses pembuatan sistem ini menggunakan *framework codeigniter* dan bahasa pemrograman PHP.

## 3.8. Pengujian Sistem

Untuk pengujian sistem akan dilakukan dengan 2 teknik, antara lain:

### 3.7.1. Metode *White Box*

*White Box testing* disebut juga sebagai *glass box testing* adalah sebuah cara pengujian yang menggunakan struktur kontrol yang dideskripsikan sebagai komponen perangkat lunak untuk memperoleh uji kasus. Dengan menggunakan *white-box testing*, penguji dapat memperoleh hasil uji kasus[13]. Sekenario pengujian sistem dengan menggunakan metode *white box* ini nantinya akan digunakan untuk menguji kesesuaian algoritma perhitungan menggunakan metode perhitungan *Simple Additive Weighting* (SAW) yang ada pada sistem ini dalam menentukan kriteria kerusakan rumah dari penerima bantuan korban bencana gempa.

### 3.7.2. Metode *Black Box*

Pengujian *black box* adalah pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinsikan tanpa mengetahui struktur kode dari sistem.

1. Pengujian fungsional oleh *developer*

Pada pengujian dilakukan dengan melihat setiap fitur dari sistem apakah telah berhasil berjalan sesuai dengan fungsinya atau tidak, kemudian melihat fungsi utama sistem yaitu hasil dari penentuan kriteria kerusakan rumah penerima bantuan korban bencana alam gempa apakah telah sesuai atau tidak dengan kriteria-kriteria yang telah di-*input*-kan. Yang membedakan pengujian ini dengan pengujian UAT ialah selama proses pengembangan sistem dan dilakukan oleh *developer*.

2. Pengujian *User Acceptance Test* (UAT) oleh *End User*

*User Acceptance Test* ialah suatu teknik pengujian untuk memverifikasi bahwa solusi yang dibuat dalam sistem sudah sesuai dengan kebutuhan pengguna. Pengujian UAT ini

termasuk pengujian *black box* karena dilakukan pengujian tanpa melihat struktur kode dari sistem. Untuk perbedaannya dengan pengujian *black box* pada penelitian ini pengujian UAT dilakukan oleh *end user* dari sistem yakni lembaga BPBD. UAT ini melibatkan responden dari lembaga BPBD yang akan menguji sistem untuk memberikan penilaian subjektif dengan kuisioner[14]. Adapun skala penilaian UAT yang digunakan dapat dilihat pada Tabel 3.34 sebagai berikut.

Tabel 3.34 Skala penilaian UAT.

| **Nilai UAT** | **Keterangan** |
|---|---|
| 5 | Sangat Sesuai |
| 4 | Sesuai |
| 3 | Kurang Sesuai |
| 2 | Tidak sesuai |
| 1 | Tidak Tau |

Pada Tabel 3.34 diatas merupakan standar penilaian dari metode UAT yang akan dilakukan dalam bentuk pertanyaan kuisioner diberikan secara langsung kepada aktor dalam sistem yakni Kepala Bidang dari BPBD, *admin* dan *team survey* dari BPBD dalam bentuk selembaran kertas yang akan menjadi bukti bahwa sistem sudah diterima dan sudah memenuhi kebutuhan.

# BAB IV

# HASIL DAN PEMBAHASAN

## 4.1 Implementasi Sistem

Pada sub bab ini, akan dibahas hasil implementasi dari perancangan pada bab sebelumnya mengenai "Implementasi Metode Simple Additive Weighting pada Sistem Pendukung Keputusan Penentuan Penerimaan Bantuan Korban Bencana Alam Gempa (Studi Kasus : BPBD Lombok Barat)". Hasil yang akan dibahas pada sub bab ini ialah berupa hasil implementasi *database*, implementasi *class* MVC dari *framework codeigniter* yang digunakan dan implementasi *interface* setiap fitur dari sistem beserta *source code*.

### 4.1.1 Implementasi *Database*

Gambar 4.1 Implementasi *database.*

Gambar 4.1 merupakan implementasi *database* yang diberi nama *database* db_gempa, terdapat 4 (empat) tabel terdiri dari tabel detailkriteria, tabel korban, tabel kriteria dan tabel *users*. Berikut penjelasan dari masing masing tabel:

1. Tabel *Users*

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | id_user | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | nama_user | varchar(50) | latin1_swedish_ci | | No | None | | |
| 3 | email | varchar(50) | latin1_swedish_ci | | No | None | | |
| 4 | hp | varchar(15) | latin1_swedish_ci | | Yes | NULL | | |
| 5 | username | varchar(32) | latin1_swedish_ci | | No | None | | |
| 6 | password | varchar(64) | latin1_swedish_ci | | No | None | | |
| 7 | akses_level | varchar(20) | latin1_swedish_ci | | No | None | | |

Gambar 4.2 Implementasi tabel *users.*

Gambar 4.2 merupakan implementasi tabel *users* yang berisi data-data *user* yang bisa masuk atau mengakses kedalam sistem, data-data *user* terdiri dari 7 atribut diantaranya atribut id_*user* sebagai *primary key* dengan tipe data *integer* dan panjang data 11, kemudian atribut nama_*user* dengan tipe data *varchar* dan panjang data 50, atribut *email* dengan tipe data *varchar* dan panjang data 50, atribut hp dengan tipe data

*varchar* dan panjang data 15, atribut *username* dengan tipe data *varchar* dan panjang data 32, atribut *password* dengan tipe data *varchar* dan panjang data 64, dan untuk atribut akses level dengan tipe data *varchar* dan panjang data 20.

2. Tabel Korban

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | id_korban | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | nama_korban | varchar(50) | latin1_swedish_ci | | No | None | | |
| 3 | nik_korban | varchar(50) | latin1_swedish_ci | | No | None | | |
| 4 | provinsi | varchar(50) | latin1_swedish_ci | | No | None | | |
| 5 | kabupaten | varchar(50) | latin1_swedish_ci | | No | None | | |
| 6 | kecamatan | varchar(50) | latin1_swedish_ci | | No | None | | |
| 7 | kriteria | text | latin1_swedish_ci | | No | None | | |
| 8 | result | float | | | Yes | NULL | | |
| 9 | keterangan | varchar(32) | latin1_swedish_ci | | No | None | | |
| 10 | tanggal_update | timestamp | | on update CURRENT_TIMESTAMP | No | CURRENT_TIMESTAMP | | ON UPDATE CURRENT_TIMESTAMP |

Gambar 4.3 Implementasi tabel korban.

Gambar 4.3 merupakan implementasi dari tabel korban yang berisi data-data korban berupa data pribadi dan data kerusakan bangunan atau rumah yang dialami korban, data-data korban pada tabel ini terdiri dari 9 (sembilan) atribut yaitu atribut id_korban sebagai *primary key* memiliki tipe data *integer* dengan panjang 11, atribut nama_korban memiliki tipe data *varchar* dengan panjang 50, atribut nik_korban memiliki tipe data *varchar* dengan panjang 50, atribut provinsi memiliki tipe data *varchar* dengan panjang 50, atribut kabupaten memiliki tipe data *varchar* dengan panjang 50, atribut kecamatan memiliki tipe data varchar dengan panjang 50, atribut kriteria memiliki tipe data text, atribut result memiliki tipe data float dan atribut keterangan memiliki tipe data *varchar* dengan panjang 32.

3. Tabel Kriteria

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | id_kriteria | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | kode_kriteria | varchar(20) | latin1_swedish_ci | | No | None | | |
| 3 | deskripsi | text | latin1_swedish_ci | | No | None | | |
| 4 | bobot | float | | | No | None | | |

Gambar 4.4 Implementasi tabel kriteria.

Gambar 4.4 merupakan implementasi dari tabel kriteria yang berisi data-data kriteria kerusakan yang digunakan sebagai tolak ukur dalam penentuan kategori kerusakan bangunan atau rumah korban gempa. Tabel kriteria memiliki 4 atribut diantaranya yaitu atribut id_kriteria sebagai *primary key* memiliki tipe data *integer*

dengan panjang data 11, atribut kode_kriteria memiliki tipe data *varchar* dengan panjang data 20, atribut deskripsi dengan tipe data *text* dan atribut bobot dengan tipe data *float*.

4. Tabel Detail Kriteria

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | id_detailkriteria | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | deskripsi | varchar(255) | latin1_swedish_ci | | No | None | | |
| 3 | nilai | float | | | No | None | | |
| 4 | id_kriteria | int(11) | | | No | None | | |

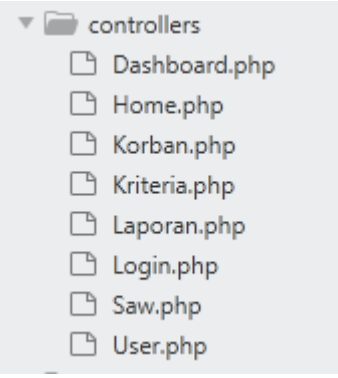
Gambar 4.5 Implementasi tabel detail kriteria.

Gambar 4.5 merupakan implementasi dari tabel detail kriteria. Tabel kriteria memiliki 4 atribut diantaranya atribut id_detailkriteria sebagai *primary key* memiliki tipe data *integer* dengan panjang data 11, atribut deskripsi memilki tipe data *varchar* dengan panjang data 255, atribut nilai dengan tipe data *float* dan atribut id_kriteria memiliki tipe data *integer* dengan panjang data 11 sebagai *foreign key*.

### 4.1.2 Implementasi *Class*

Selanjutnya akan dibahas implementasi *class* yang digunakan sesuai dengan struktur *class framework CodeIgniter* yakni dengan struktur MVC (Model View Controller). *Class-class* ini merupakan *class* yang berisi *source code* dengan menggunakan bahasa pemrograman PHP untuk mengimplementasikan sistem sesuai dengan perancangan yang dilakukan pada sub bab sebelumnya. Berikut beberapa struktur *class* yang diimplementasikan dalam sistem.

1. *Class Controllers*




Gambar 4.6 Implementasi *class controllers.*

Gambar 4.6 merupakan *folder* dari *class controllers* terdiri dari beberapa *file* berformat PHP yang berisi *source code* implementasi *controller* dari struktur MVC

*framework codeigniter* yang digunakan, *controllers* berfungsi untuk memproses data dalam sistem dan berfungsi menghubungkan antara *class models* dan *class views*. Beberapa *class controllers* yang ada berdasarkan Gambar 4.6 diantaranya *controller dashboard* memproses data dan *view dashboard*, *controller home* memproses data dan *view* beranda, controller korban memproses data dan *view* korban, controller kritreria memproses data dan *view* kriteria, *controller* laporan memproses data dan *view* laporan, *controller login* memproses data dan *view login*, controller saw memproses data dan *view* hitung dengan metode saw, dan *controller user* memproses data dan *view user*. Salah satu contoh class controller bisa dilihat pada gambar 4.7.

```
class Login extends CI_Controller {
        public function index() {
            $username = $this->input->post('username');
            $password = $this->input->post('password');
            $this->form_validation->set_rules('username',
'Username', 'required');
            $this->form_validation->set_rules('password',
'Password', 'required');
            if($this->form_validation->run())
            {
                  $this->my_login->login($username,$password);
            }
            $data = array( 'tittle' => "HALAMAN LOGIN" );
            $this->load->view("login/index", $data, FALSE);}
```

Gambar 4.7 Contoh *class controller login.*

Gambar 4.7 merupakan salah satu contoh *class controller* yang implementasikan ke dalam sistem, pada gambar tersebut terdapan *source code* untuk menampilkan *view form login* dan melakukan pengecekan *username* dan *password* yang di-*input*-kan *user* untuk *login* kedalam sistem. Salah satu ciri dari *class controller* pada *framework codeigniter* ialah ber-*extends* CI_Controller.

Berikut merupakan implementasi *class controller* untuk metode *saw* lebih jelasnya bisa dilihat pada Gambar 4.8.

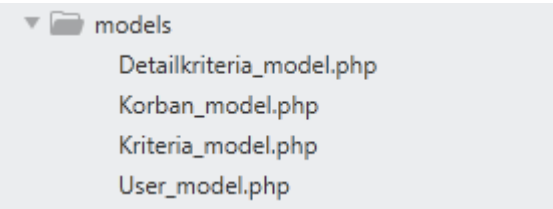
```
$tabel = array();
foreach($korban as $r){
 $no = 0;
 $detailkriteria = json_decode($r->kriteria,true);
 foreach($kriteria as $k){
 $tabel[$r->id_korban][] = $detailkriteria[$no]['nilai'];
        $no++;
    }
}
$step2 = array();
foreach($tabel as $t=>$val){
    $no = 0;
```

```
    foreach($kriteria as $k){
        $step2[$no][] = $tabel[$t][$no];
        $no++;
    }
}
$max = array();
foreach($step2 as $s=>$val){
    $max[$s] = max($val);
}
$step3 = array();
foreach($tabel as $r=>$val){
    $no = 0;
    foreach($val as $v){
        $step3[$r][]=@ round($v/$max[$no],4);
        $no++;
    }
}
$bobot = array();
foreach($kriteria as $k){
    $bobot[] = $k->bobot;
}
$result = array();
foreach($step3 as $k=>$v){
    $i = 0;
    foreach($v as $n){
        if (!isset($result[$k])){
            $result[$k] = "";
        }
        $result[$k]= $result[$k]+($n*$bobot[$i]);
        $++;
    }
```

Gambar 4.8 Implementasi metode SAW.

Gambar 4.8 merupakan implementasi code dari metode SAW pada sistem, digunakan untuk mencari nilai perhitungan dari setiap bobot kriteria yang diinputkan yang hasil akhirnya akan dikategorikan berdasarkan skala nilai dari masing-masing kategori kerusakan.

2. *Class Models*




Gambar 4.9 Implementasi *class models.*

Gambar 4.9 merupakan *folder models* terdiri dari beberapa *file* berformat PHP yang berisi *source code* implementasi *model* dari struktur MVC *framework codeigniter*, *model* berfungsi mengelola data dari *database*. Beberapa *class models* berdasarkan pada Gambar 4.9 diantaranya *class* Detailkriteria_model berfungsi menglola data detail

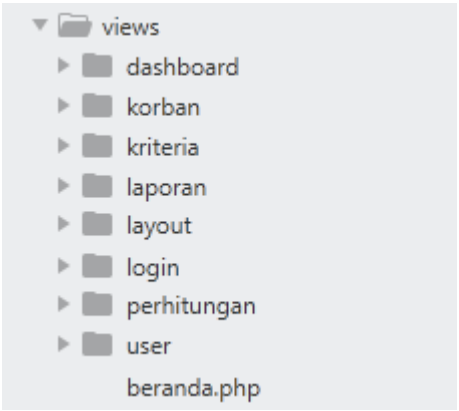
kriteria pada *database*, *class* Korban_model berfungsi mengelola data korban pada *database*, *class* Kriteria_model berfungsi mengelola data kriteria pada *database* dan *class User*_model berfungsi mengelola data *user* pada *database*. Salah satu contoh *class models* bisa dilihat pada Gambar 4.10 berikut.

```
public function login($username,$password)
{
      $this->db->select('*');
      $this->db->from('users');
      $this->db->where(array( 'username'  => $username,
                                          'password'  =>
SHA1($password))
                              );
      $this->db->order_by('id_user', 'desc');
      $query = $this->db->get();
      return $query->row();
}
```

Gambar 4.10 Contoh *class models user*.

Gambar 4.10 merupakan contoh *class models* yang diimplementasikan ke dalam sistem untuk mengolah data *user* pada *database*. Bisa dilihat pada Gambar 4.10 terdapat *source code* untuk mengambil data *user* dari *database* berdasarkan *username* dan *password* yang di-*input*-kan *user* ketika melakukan *login* pada *form login*.

3. *Class Views*




Gambar 4.11 Implementasi *class views.*

Gambar 4.11 merupakan *folder views* terdiri dari beberapa *folder class views* yang berisi beberapa *file* berformat PHP berfungsi mengatur tampilan atau *interface* dari masing-masing fitur yang ada dalam sistem. Beberapa *class* views yang ada berdasarkan Gambar 4.11 diantaranya sebagai berikut *view* dashboard berfungsi mengelola halaman *dashboard*, *view* korban berfungsi mengelola halaman korban, *view* kriteria berfungsi mengelola halaman kriteria, *view* laporan berfungsi menglola tampilan laporan, *view*

*layout* berfungsi mengelola tampilan *layout* dari setiap halaman, *view login* berfungsi mengelola halaman *login*, *view* perhitungan berfungsi mengelola halaman langkah perhitungan, *view user* berfungsi mengola halaman *user* dan *view* beranda berfungsi mengelola tampilan beranda. Salah satu contoh *class views* bisa dilihat pada gambar 4.14 berikut.

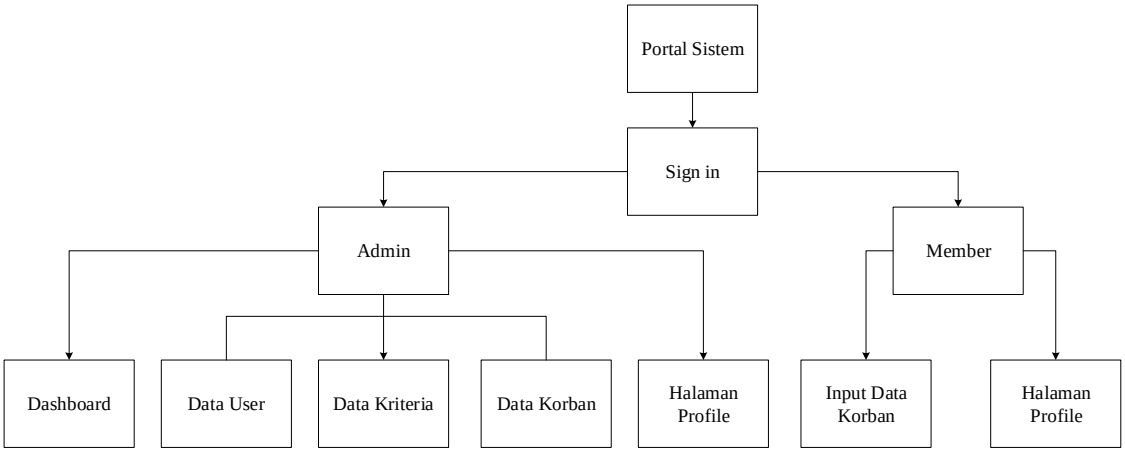
```
<div class="input-group mb-3">
  <input type="text" class="form-control" placeholder="Usernames"
name="username" required="">
     <div class="input-group-append">
         <div class="input-group-text">
         <span class="fas fa-user"></span>
         </div>
      </div>
</div>

<div class="input-group mb-3">
      <input type="password" class="form-control"
placeholder="Password" name="password">
            <div class="input-group-append">
                  <div class="input-group-text">
                  <span class="fas fa-lock"></span>
                  </div>
            </div>
</div>

<div class="row">
      <div class="col-8"></div>
            <div class="col-4">
            <button type="submit" class="btn btn-primary btn-block
float-right">Login</button>
            </div>
</div>
```

Gambar 4.12 Contoh *class views form login*.

Gambar 4.12 merupakan contoh potongan sebagian *source code* dari *view form login* yang pada gambar tersebut terdapat *input box* atau kotak isian *username*, *input box* atau kotak isian *password* dan tombol *login* yang ketika *user* mengklik tombol *login* tersebut akan menjalankan fungsi *login* pada *class controller login* untuk melakukan pengecekan *username* dan *password* yang di *input*-kan *user* pada halaman *form login* tersebut.

### 4.1.3 Implementasi *Interface*

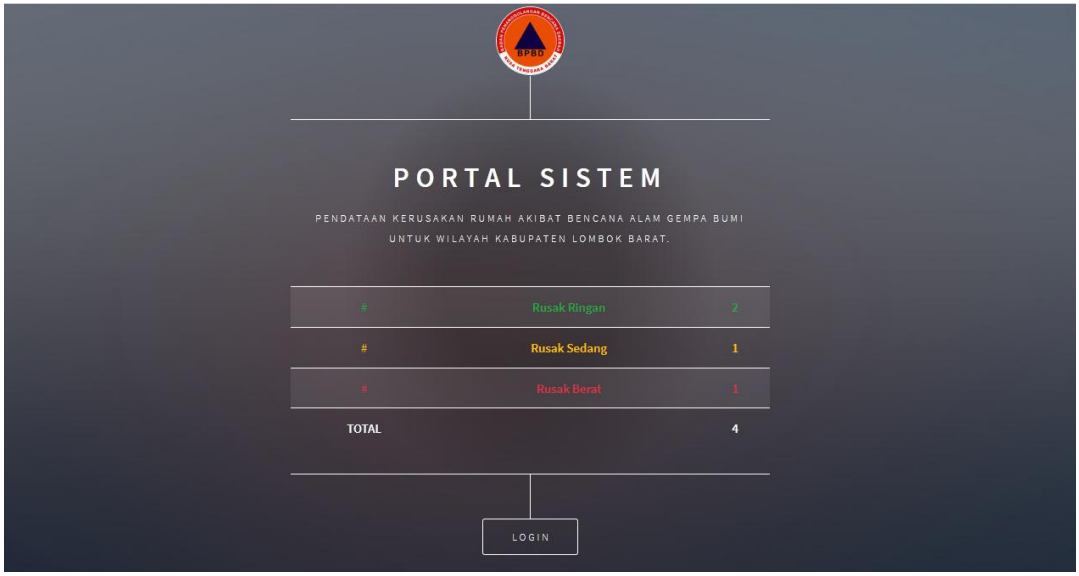
*Interface* atau tampilan sistem merupakan media interaksi antara pengguna atau *user* dengan sistem, *interface* dirancang atau dibangun menjadi dua bagian yakni tampilan yang bisa di akses bersama oleh admin dan *team* survei dan tampilan yang hanya bisa di akses oleh admin atau *team* survei.




Gambar 4.13 *Site map* sistem.

.Berikut beberapa implementasi interface dari sistem untuk sisi *admin* maupun sisi *team survey*.

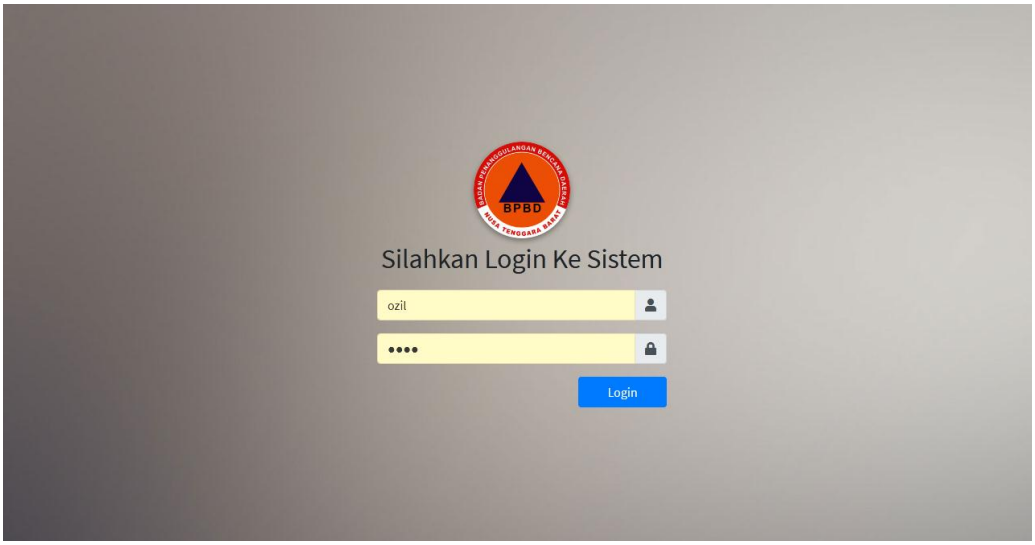
1. Halaman awal




Gambar 4.14 Implementasi halaman awal.

Gambar 4.13 merupakan implementasi halaman awal. Terdapat tampilan awal dari sistem yang berisi deskripsi singkat mengenai sistem, informasi kesimpulan hasil akhir yang diolah oleh sistem dan terdapat tombol *login* untuk *user* bisa masuk kedalam sistem. Halaman ini bisa diakses bersama oleh *user admin* dan *user team survei*.

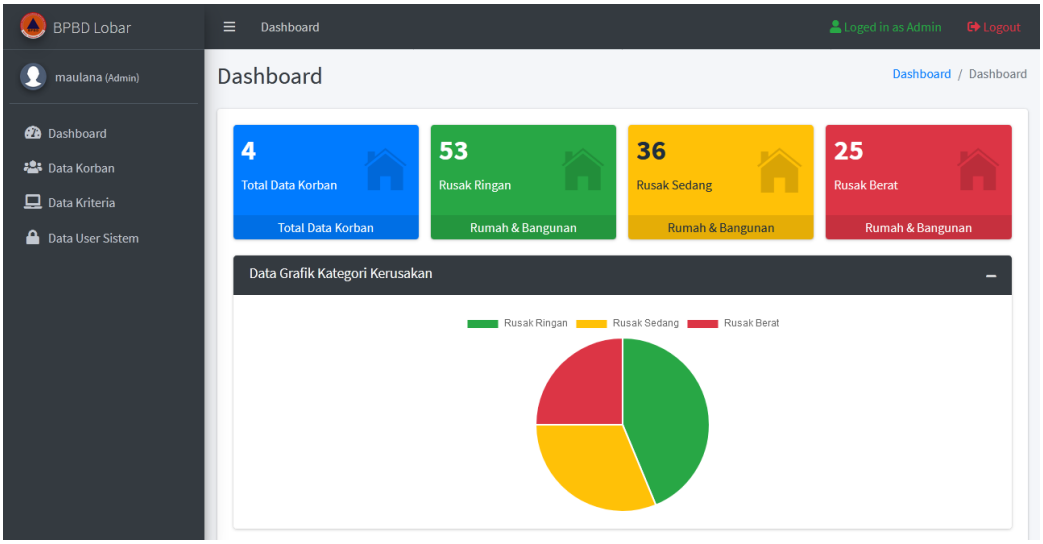
2. Halaman *login*




Gambar 4.15 Implementasi halaman *login.*

Gambar 4.14 merupakan implementasi halaman *login* yang digunakan untuk bisa masuk kedalam sistem dengan menggunakan *user*name dan password yang telah diberikan oleh pihak BPBD sebagai Admin dari sistem. Pada halama ini terdapat form dengan kotak isian *user*name dan password, dan tombol *login*. Halaman ini bisa diakses bersama oleh *user admin* dan *user team survei*.

3. Halaman dashboard

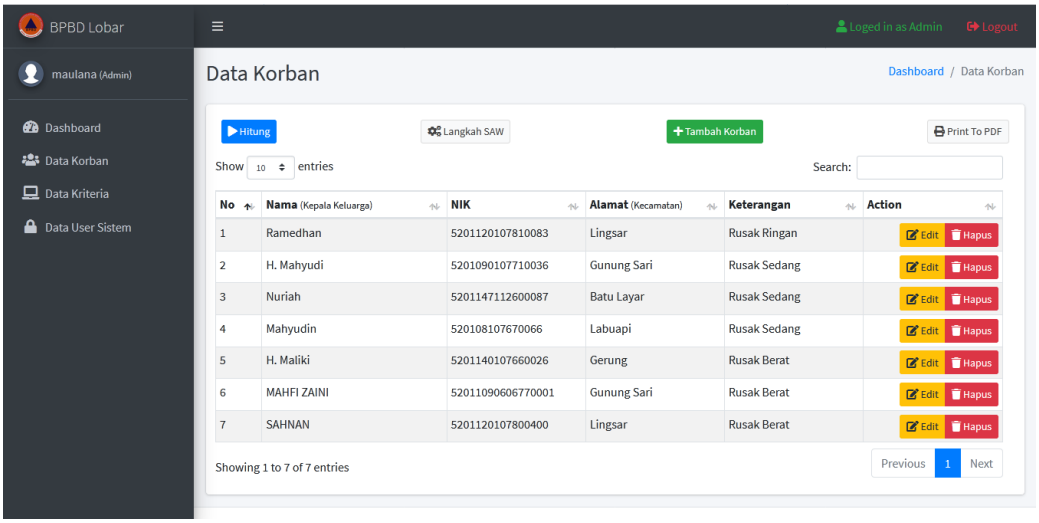



Gambar 4.16 Implementasi halaman *dashboard*.

Gambar 4.15 merupakan implementasi halaman *dashboard*. Pada halaman *dashboard* ini terdapat *widget* data diantaranya total keseluruhan data korban, total data rusak ringan, total data rusak sedang dan total data rusak berat, beserta data grafik untuk

memudahkan melihat perbandingan dari masing-masing data. Halaman ini hanya bisa diakses oleh *user admin*.

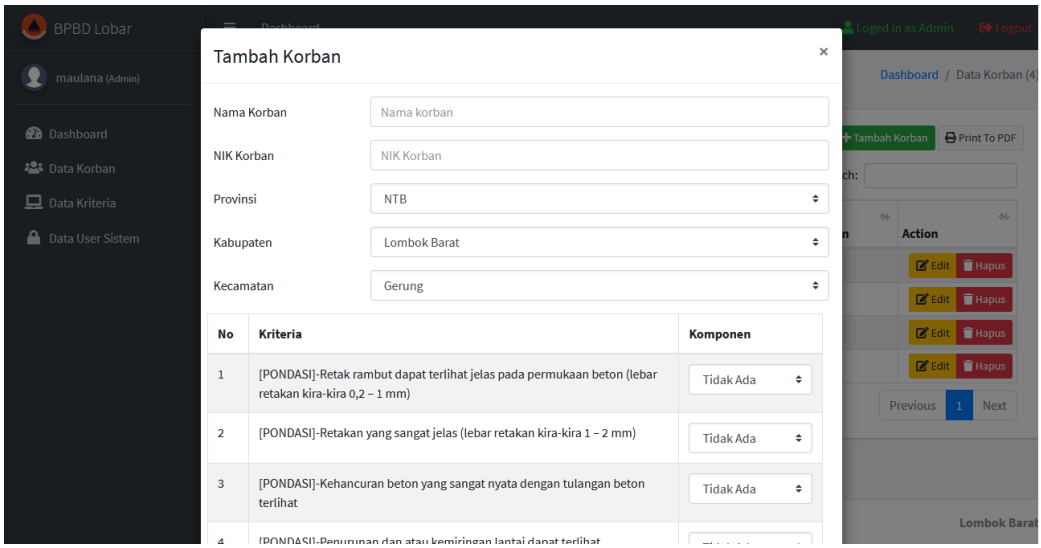
4. Halaman data korban




Gambar 4.17 Implementasi halaman data korban.

Gambar 4.16 merupakan implementasi halaman data korban. Pada halaman ini terdapat tabel yang berisi data para korban beserta aksi yang bisa dilakukan yakni mengedit dan menghapus data. Terdapat pula beberapa tombol diantaranya tombol hitung digunakan untuk melakukan perhitungan data, tombol langkah saw berfungsi untuk berpindah ke halaman langkah saw, tombol tambah korban berfungsi untuk menambah data korban baru dan tombol print to pdf berfungsi untuk mencetak laporan ke dalam bentuk file berformat pdf.

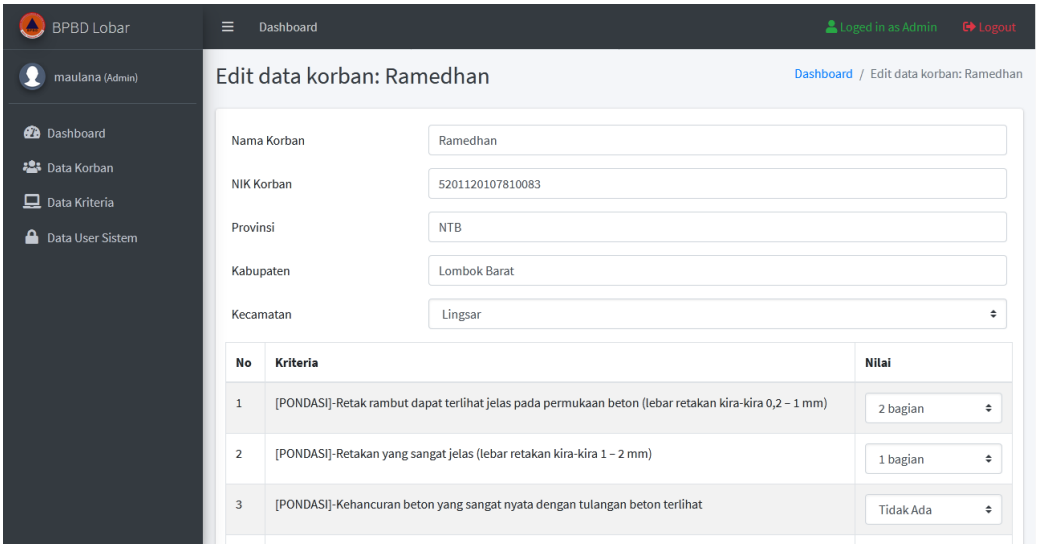
5. Tampilan tambah data korban




Gambar 4.18 Implementasi tampilan tambah data korban.

Gambar 4.17 merupakan implementasi tampilan tambah data korban. Terdapat form isian berupa beberapa kotak isian menyangkut data pribadi korban, tabel isian kriteria kerusakan rumah atau bangunan yang dialami korban dan tombol simpan yang berfungsi untuk menyimpan data kedalam *database*.

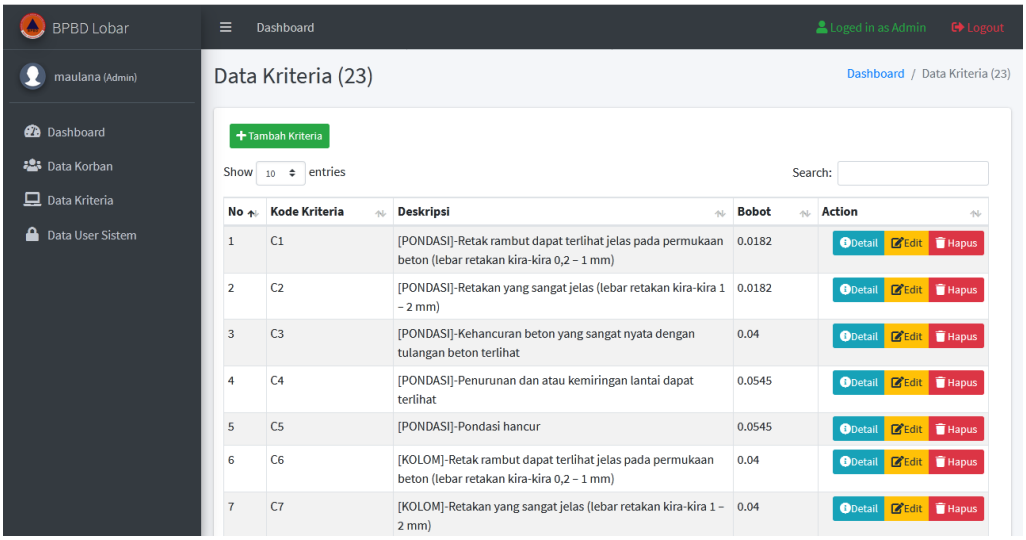
6. Halaman edit data korban




Gambar 4.19 Implementasi halaman edit data korban.

Gambar 4.18 merupakan implementasi halaman edit data korban. Pada halaman ini terdapat form isian sama seperti pada tampilan tambah data korban, yang membedakan ialah pada halaman ini mengambil data yang hendak di edit pada database kemudian di simpan kembali untuk diperbaharui.

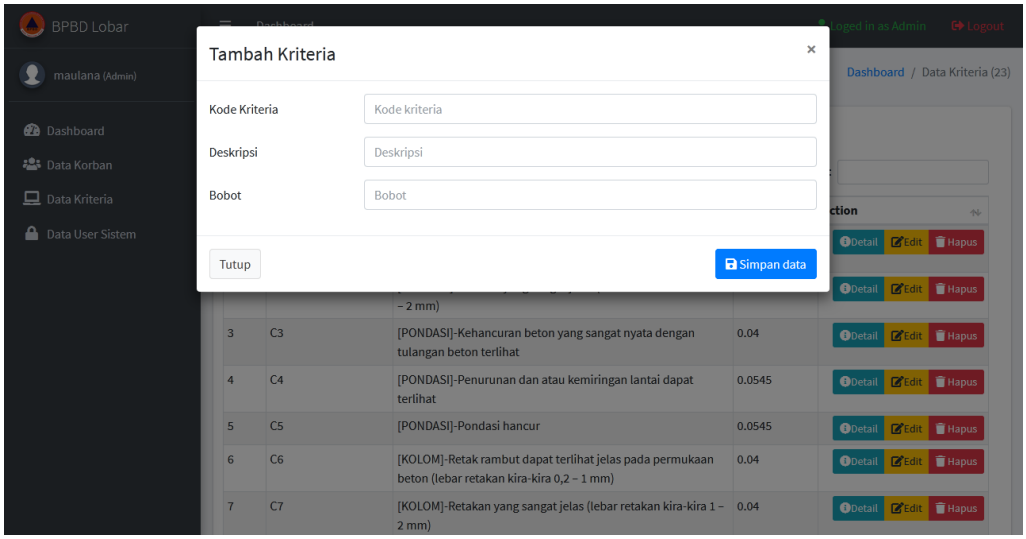
7. Halaman data kriteria




Gambar 4.20 Implementasi halaman data kriteria.

Gambar 4.19 merupakan implementasi halaman data kriteria. Pada halaman ini terdapat tabel berisikan data kriteria aksi berupa melihat detail data kriteria, mengedit data kriteria dan menghapus data kriteria. Terdapat pula tombol tambah kriteria untuk menambah data kriteria yang akan digunakan sebagai acuan dalam perhitungan.

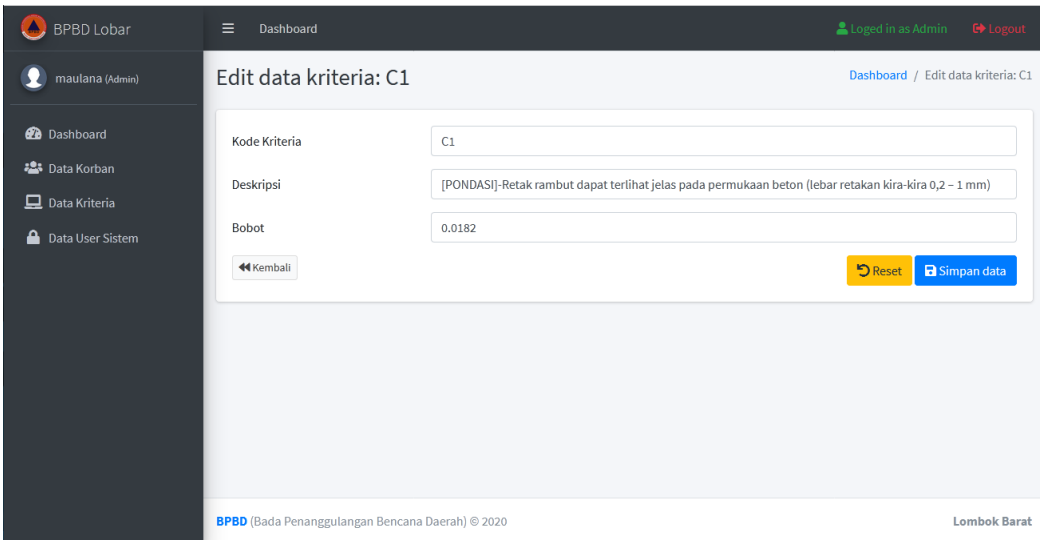
8. Tampilan tambah data kriteria




Gambar 4.21 Implementasi tampilan tambah data kriteria.

Gambar 4.20 merupakan implementasi tampilan tambah data kriteria. Terdapat *form* isian dengan beberapa kotak isian yang harus diisi diantaranya kode kriteria, deskripsi dan bobot. Kemudian terdapat dua tombol diantaranya tombol tutup untuk menutup tampilan tambah data kriteria dan tombol simpan data berfungsi untuk menyimpan data kriteria ke dalam *database*.

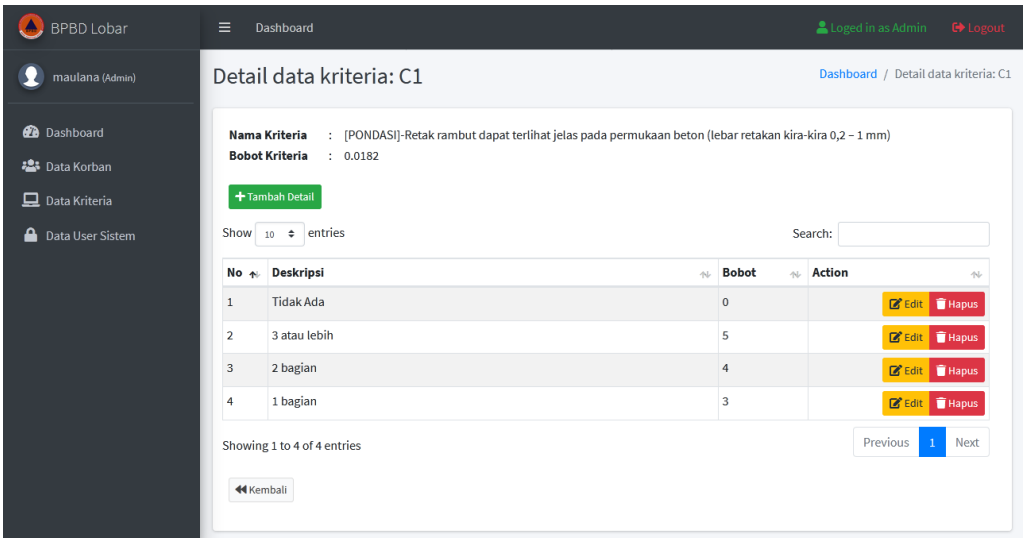
9. Halaman edit data kriteria




Gambar 4.22 Implementasi halaman edit data kriteria.

Gambar 4.21 merupakan implementasi halaman edit data kriteria. Terdapat form isian yang sama seperti form tambah data kriteria, yang membedakan ialah *form* ini mengambil data yang hendak diedit pada *database*.

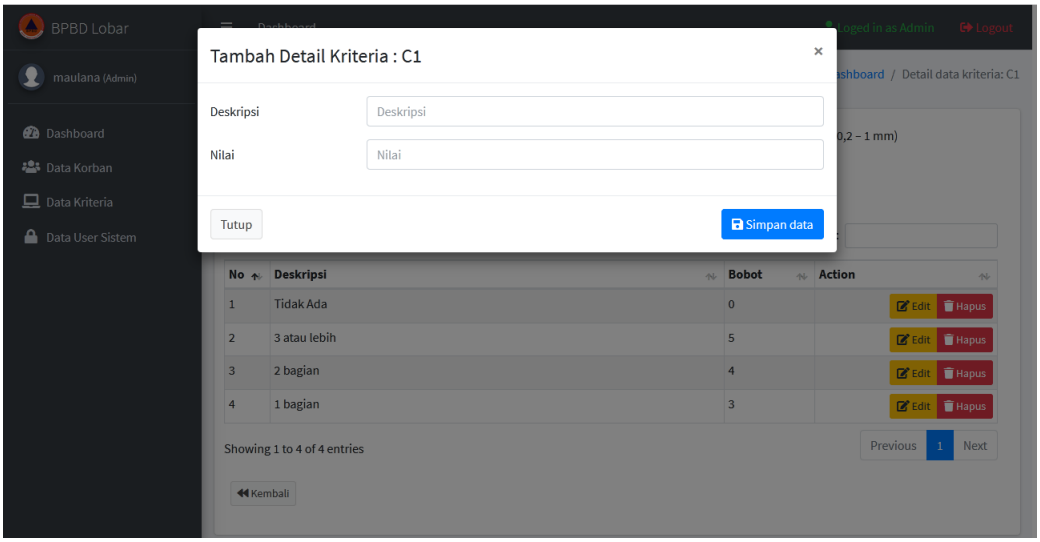
10. Halaman detail data kriteria




Gambar 4.23 Implementasi halaman detail data kriteria.

Gambar 4.22 merupakan implementasi halaman detail data kriteria. Terdapat tabel yang berisikan detail data kriteria beserta aksi yang dapat dilakukan diantaranya yaitu mengedit detail dan menghapus detail. Dan terdapat tombol tambah detail yang berfungsi untuk menambah data detail dari kriteria.

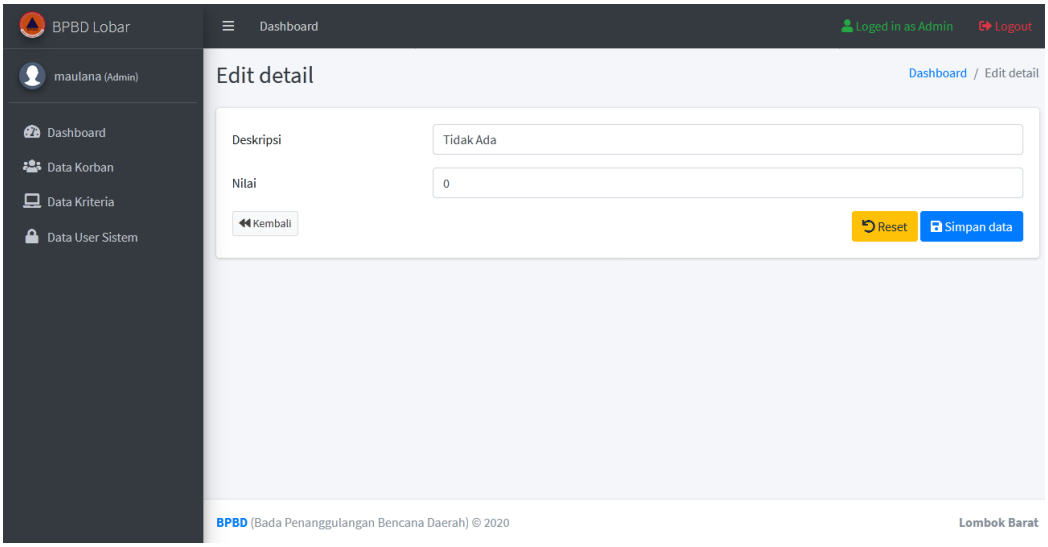
11. Tampilan tambah data detail kriteria




Gambar 4.24 Implementasi halaman tambah data detail kriteria.

Gambar 4.23 merupakan implementasi tampilan tambah detail. Terdapat *form* untuk mengisi detail dari data kriteria berupa deskripsi dan nilai dari detail krirteria.

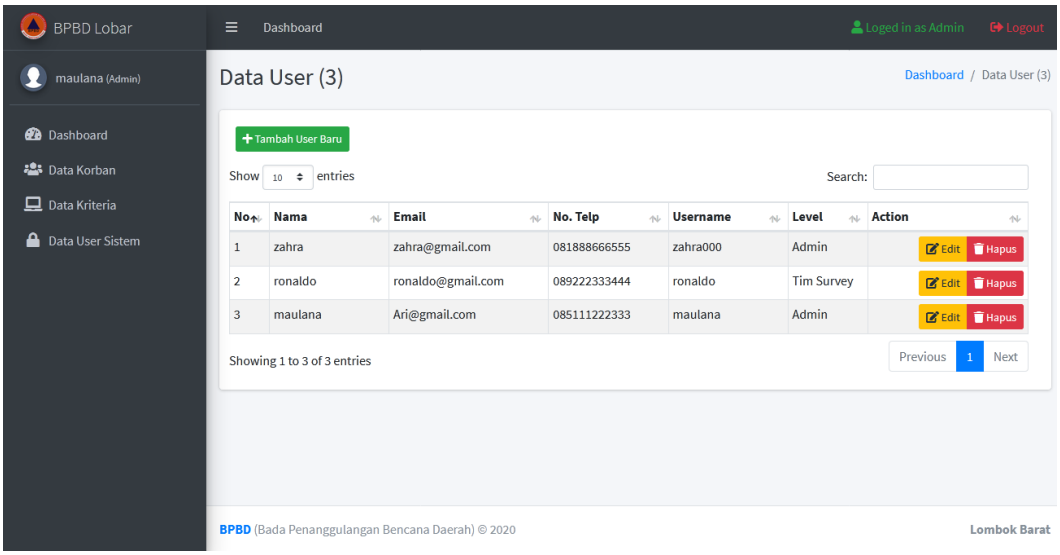
12. Halaman edit data detail kriteria




Gambar 4.25 Implementasi halaman edit data detail kriteria.

Gambar 4.24 merupakan implementasi halaman edit data detail kriteria. Pada halaman ini terdapat form yang sama dengan tampilan tambah data detail namun yang membedakan ialah pada halaman ini mengambil data yang sudah ada pada *database* untuk diedit dan perbaharui.

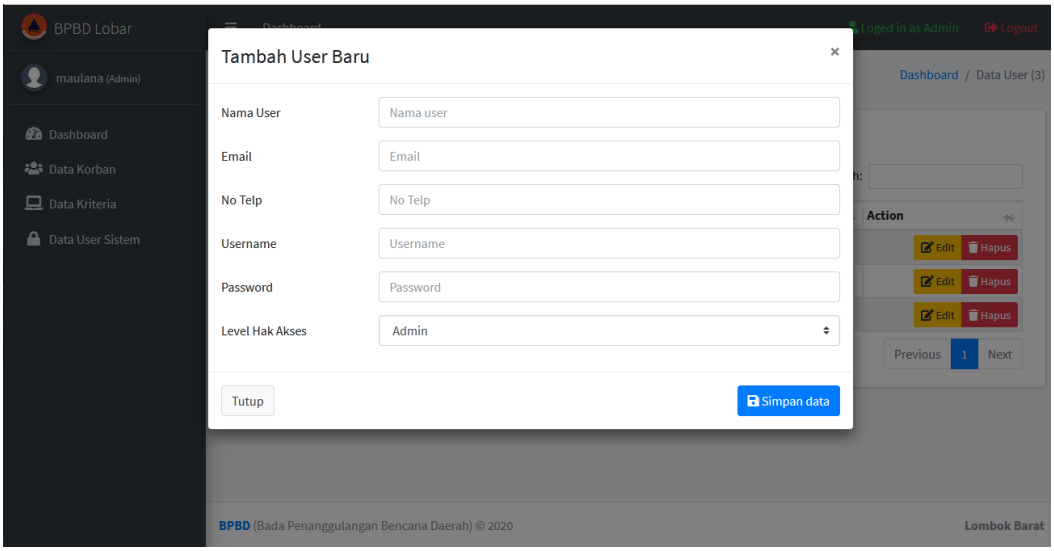
13. Halaman data *user*




Gambar 4.26 Implementasi halaman data *user*.

Gambar 4.25 merupakan implementasi halaman data *user*. Pada halaman ini terdapat tabel data *user* dengan aksi mengedi data *user* dan menghapus data *user*. Terdapat juga tombol tambah *user* untuk menambah *user* baru.

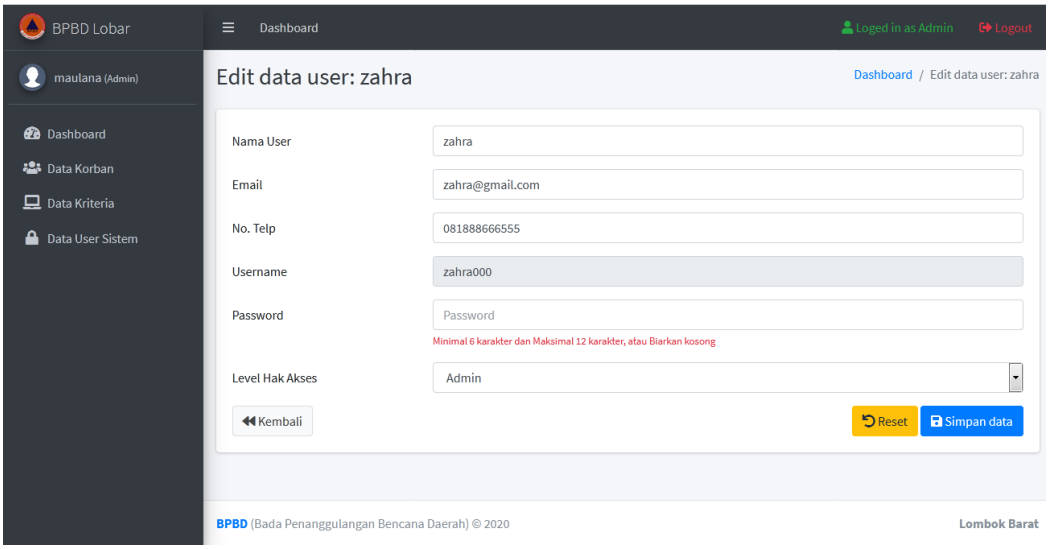
14. Tampilan tambah *user*




Gambar 4.27 Implementasi tampilan tambah *user*.

Gambar 4.26 merupakan implementasi tampilan tambah *user* baru. Pada tampilan ini terdapat *form* isian data *user* baru yang akan ditambahkan beserta tombol simpan untuk mennyimpan data kedalam *database*.

15. Halaman edit data *user*

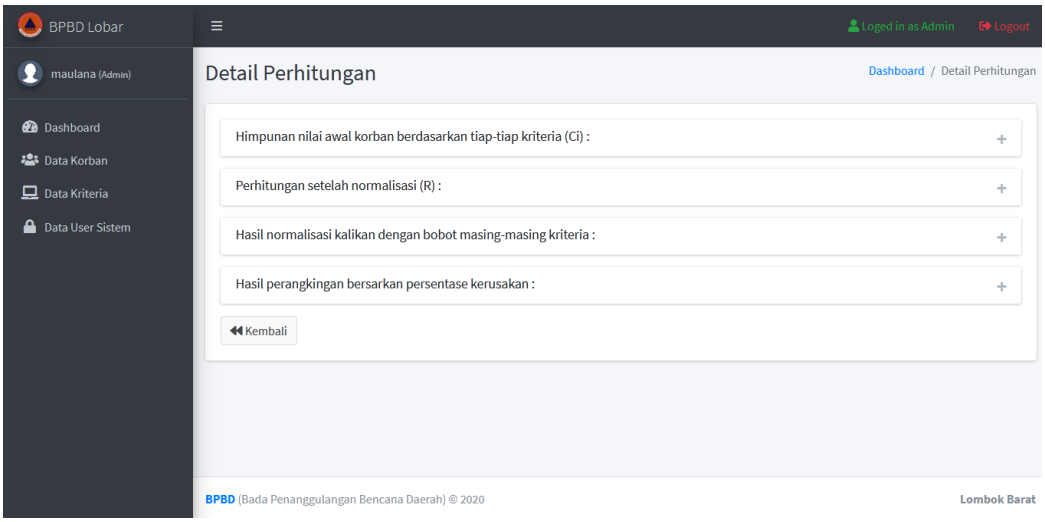



Gambar 4.28 Implementasi halaman edit data *user*.

Gambar 4.27 merupakan implementasi halaman edit data *user*. Halaman ini sama seperti halaman tambah data *user* baru, namun yang membedakan ialah halaman ini

mengambil terlebih dahulu data *user* yang akan diedit pada *database* untuk selanjutnya diperbaharui.

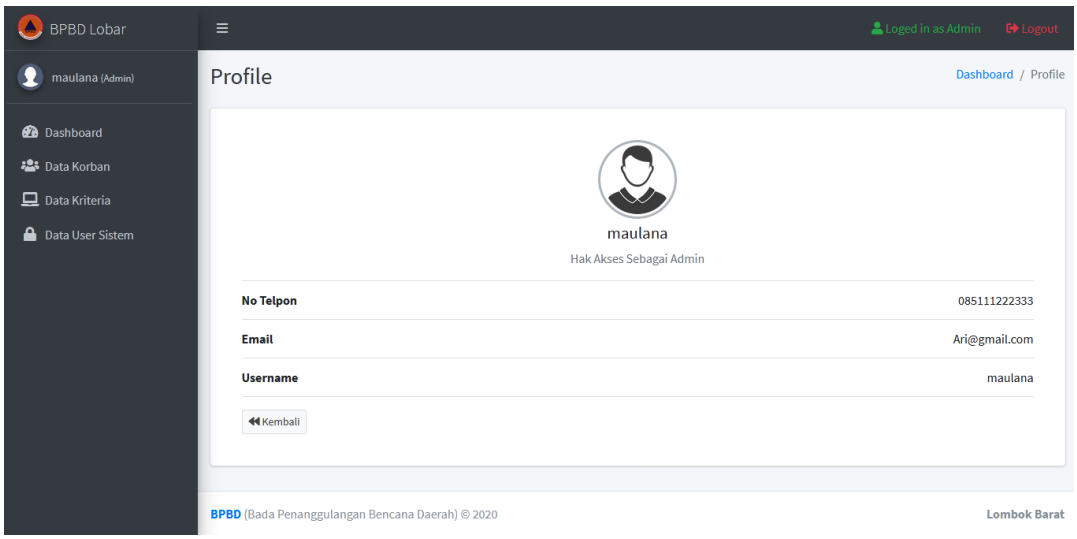
16. Halaman langkah SAW




Gambar 4.29 Implementasi halaman langkah SAW.

Gambar 4.28 merupakan implementasi halaman langkah perhitungan saw. Pada halaman ini terdapat pemaparan langkah-langkah perhitungan data yang dilakukan oleh sistem sesuai dengan metode yang di implementasikan yakni metode *simple additive weighting* (SAW).

17. Halaman *profile*




Gambar 4.30 Implementasi halaman *profile*.

Gambar 4.29 merupakan implementasi halaman *profile user* yang sedang *login*. Pada halaman ini ditampilkan data-data dari *user* yang sedang melakukan *login* ke dalam sistem.

## 4.2 Pengujian Sistem

Untuk hasil pengujian sistem dilakukan dengan metode pengujian yakni pengujian *white box* dan *black box*.

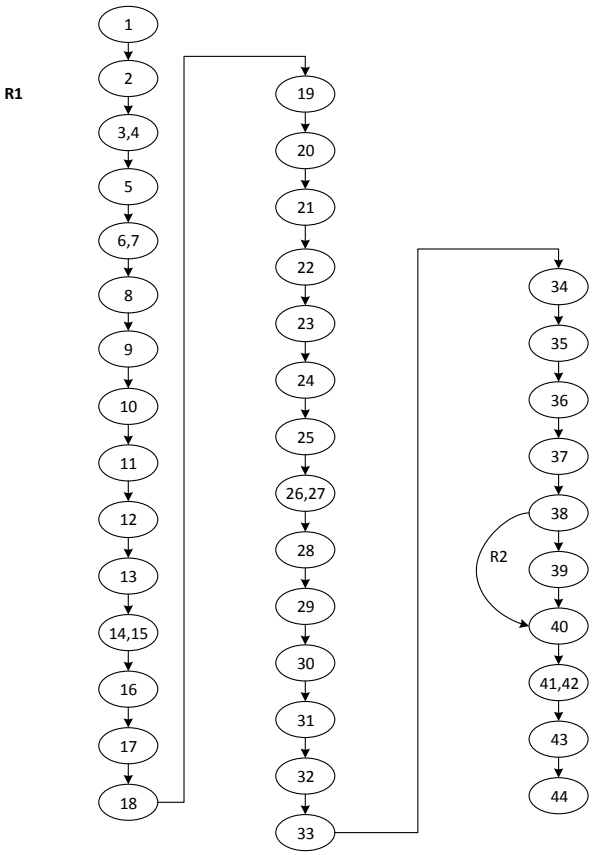
### 4.2.1 Pengujian *White Box*

Untuk pengujian *white box* pada sistem ini akan dilakukan pada bagian algoritma *Simple Additive Weighting* (SAW). Dalam melakukan pengujian menggunakan metode *white box* dilakukan terlebih dahulu pengubahan *code* algoritma *simple additive weighting* (SAW) menjadi *basis path*[15][16].

```
1   $tabel = array();
2   foreach($korban as $r){
3    $no = 0;
4    $detailkriteria = json_decode($r->kriteria,true);
5    foreach($kriteria as $k){
6    $tabel[$r->id_korban][] = $detailkriteria[$no]['nilai'];
7           $no++;
8       }
9   }
10  $step2 = array();
11  foreach($tabel as $t=>$val){
12      $no = 0;
13      foreach($kriteria as $k){
14          $step2[$no][] = $tabel[$t][$no];
15          $no++;
16      }
17  }
18  $max = array();
19  foreach($step2 as $s=>$val){
20      $max[$s] = max($val);
21  }
22  $step3 = array();
23  foreach($tabel as $r=>$val){
24      $no = 0;
25      foreach($val as $v){
26          $step3[$r][]=@ round($v/$max[$no],4);
27          $no++;
28      }
29  }
30  $bobot = array();
31  foreach($kriteria as $k){
32      $bobot[] = $k->bobot;
33  }
34  $result = array();
35  foreach($step3 as $k=>$v){
36      $i = 0;
37      foreach($v as $n){
```

```
38          if (!isset($result[$k])){
39              $result[$k] = "";
40          }
41          $result[$k]= $result[$k]+($n*$bobot[$i]);
42          $i++;
43      }
44  }
```

Berdasarkan *code* algoritma SAW diatas akan diubah kedalam bentuk *basis path* algoritma SAW sebagaimana bisa dilihat pada Gambar 4.30 berikut.




Gambar 4.31 *Basis Path* Algoritma SAW.

Keterangan:

= Menggambarkan kondisi

= Menggambarkan aksi

Dari Gambar 4.31 dapat dihitung *cyclomatic complexity* sebagai berikut:

V(G) = E – N + 2

V(G) = 39 – 39 + 2

V(G) = 2

Dimana:

E = jumlah *edge* pada *basis path*

N = jumlah *node* pada *basis path*

Jadi, berdasarkan hasil perhitungan *cyclomatic complexity* diatas, maka didapatkan 2 *path* atau jalur algoritma SAW dalam menentukan kriteria kerusakan yang diantaranya bisa dilihat pada tabel 4.1 berikut.

Tabel 4.1 *Path* algoritma SAW.

| | |
|---|---|
| *Path* 1 | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28 ,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44 |
| *Path* 2 | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28 ,29,30,31,32,33,34,35,36,37,38,40,41,42,43,44 |

Selanjutnya dilakukan pengujian *path* algoritma SAW yang bisa dilihat lebih jelasnya pada tabel 4.2 sebagai berikut.

Tabel 4.2 Pengujian algoritma SAW.

| Path | Kasus Uji | Hasil Yang Diharapkan | Hasil Uji | Keterangan |
|---|---|---|---|---|
| 1 | Menginputkan setiap nilai dari kriteria kerusakan | -Melakukan perulangan untuk kriteria keputusan<br>-melakukan perulangan unutuk rating kecocokan<br>-melakukan perulangan untuk matrix ternormalisasi<br>-melakukan perulangan untuk perangkingan | Hasil perangkingan ditemukan | [✓] Alur terlewati<br>[ ] Alur tidak terlewati |
| 2 | Variabel `$result` tidak berisi | Mengosongkan nilai dari variabel `$result` | Variabel `$result` kosong | [✓] Alur terlewati<br>[ ] Alur tidak terlewati |

### 4.2.2 Pengujian *Black Box*

Untuk pengujian Black Box dibagi menjadi 2 pengujian yakni pengujian oleh *developer* dan pengujian *user acceptance test* (UAT) oleh *end user*.

1. Pengujian fungsional oleh *developer*

Pengujian ini dilakukan pada setiap fungsi dari fitur-fitur yang terdapat dalam sistem yang terbagi menjadi dua yakni fitur-fitur sebagai *user admin* dan sebagai *user team survey*. Untuk lebih jelasnya akan dijabarkan masing-masing dari pengujian *black box* sebagai berikut.

a. Fungsional fitur *user admin*

Untuk fitur-fitur *user admin* terdapat 23 fitur yang akan dilakukan pengujian seperti terlihat pada Tabel 4.3 berikut.

Tabel 4.3 Pengujian fungsional fitur *user admin*

| **No** | **Fungsi** | **Hasil yang diharapkan** | **Hasil pengujian** | |
|---|---|---|---|---|
| | | | **Sesuai** | **Tidak** |
| 1 | *Login* sebagai *admin* | Bisa *login* sebagai *admin* | ✓ | |
| 2 | Menampilkan halaman *dashboard* | Mampu Menampilkan halaman *dashboard* | ✓ | |
| 3 | Menampilkan data korban | Mampu menampilkan data korban | ✓ | |
| 4 | Melakukan perhitungan data | Mampu melakukan perhitungan data | ✓ | |
| 5 | Menambahkan data korban | Mampu menambahkan data korban | ✓ | |
| 6 | Mengubah data korban | Mampu mengubah data korban | ✓ | |
| 7 | Menghapus data korban | Mampu menghapus data korban | ✓ | |
| 8 | Menampilkan data kriteria | Mampu menampilkan data kriteria | ✓ | |
| 9 | Menambah data kriteria | Mampu menambah data kriteria | ✓ | |
| 10 | Mengubah data kriteria | Mampu mengubah data kriteria | ✓ | |
| 11 | Menghapus data kriteria | Mampu menghapus data kriteria | ✓ | |
| 12 | Menampilkan detail data kriteria | Mampu menampilkan detail data kriteria | ✓ | |
| 13 | Menambah detail data kriteria | Mampu menambah detail data kriteria | ✓ | |
| 14 | Mengubah detail data kriteria | Mampu mengubah detail data kriteria | ✓ | |
| 15 | Menghapus detail data kriteria | Mampu menghapus detail data kriteria | ✓ | |
| 16 | Menampilkan data *user* | Mampu menampilkan data *user* | ✓ | |
| 17 | Menambah data *user* | Mampu menambah data *user* | ✓ | |

| | | | | |
|---|---|---|---|---|
| 18 | Mengubah data *user* | Mampu mengubah data *user* | ✓ | |
| 19 | Menghapus data *user* | Mampu menghapus data *user* | ✓ | |
| 20 | Menampilkan detail langkah perhitungan | Mampu menampilkan detail langkah perhitungan | ✓ | |
| 21 | Menampilkan *profile user* | Mampu menampilkan *profile user* yang sedang *login* | ✓ | |
| 22 | Mencetak laporan | Mampu mencetak laporan | ✓ | |
| 23 | *Logout* | Mampu melakukan *logout* | ✓ | |

b. Fungsional fitur *user team survey*

Untuk pengujian fitur-fitur *user team survey* terdapat 4 fitur seperti yang bisa dilihat pada Tabel 4.4 sebagai berikut.

Tabel 4.4 Pengujian fungsional fitur *user team survey.*

| **No** | **Fungsi** | **Hasil yang diharapkan** | **Hasil pengujian** | |
|---|---|---|---|---|
| | | | **Sesuai** | **Tidak** |
| 1 | *Login* sebagai *user team survey* | Bisa *login* sebagai *user team survey* | ✓ | |
| 2 | Melakukan *input* data korban | Mampu melakukan *input* data korban | ✓ | |
| 3 | Menampilkan *profile* | Bisa menampilkan *profile* | ✓ | |
| 4 | *Logout* | Bisa *logout* | ✓ | |

2. Pengujian User Acceptance Test (UAT) oleh *End User*

Untuk pengujian dengan menggunakan metode UAT ini dilakukan dengan mencari responden dari pihak terkait yakni *end user* dari lembaga BPBD untuk mencoba menggunakan sistem lalu memberikan penilaian terhadap sistem dalam bentuk kuisioner. Pengujian dengan metode UAT ini dilakukan untuk mengetahui bagaimana karakteristik atau kualitas sistem dari sisi *user*.

Skenario dari pengujian ini yakni 3 orang dari lembaga BPBD yang bertanggung jawab dalam hal mendata maupun mengolah data kerusakan rumah akibat gempa yakni kepala bidang Rehabilitasi dan Rekonstruksi (RR), kepala bidang Darurat dan Logistik (DarLog) dan staf BPBD Lombok Barat akan diminta untuk menjalankan sistem kemudian akan diberikan pertanyaan mengenai sistem dengan mengisi kuisioner yang telah diberikan. Untuk dokumentasi pengujian UAT yang telah dilakukan dapat dilihat pada lampiran 1 dan lampiran 2.

Hasil dari pengujian *user acceptance test* (UAT) berdasarkan skala penilaian UAT pada Tabel 3.34 dapat dilihat pada penjabaran sebagai berikut:

1. Pengujian sistem dari sisi *admin*

Tabel 4.5 Hasil pengujian UAT dasi sisi *admin*.

| No | Pernyataan | Skala penilaian | | | | |
|---|---|---|---|---|---|---|
| | | TT | TS | KS | S | SS |
| 1 | Apakah tampilan sistem nyaman dilihat ? | - | - | - | 1 | 2 |
| 2 | Apakah sistem nyaman digunakan ? | - | - | - | 2 | 1 |
| 3 | Apakah penggunaan menu atau fitur menu dalam sistem mudah dimengerti ? | - | - | - | 2 | 1 |
| 4 | Apakah sistem ini dapat membantu kinerja *user* ? | - | - | - | - | 3 |
| 5 | Apakah sistem dapat mempercepat pekerjaan *user* ? | - | - | - | 1 | 2 |
| 6 | Apakah informasi yang disediakan sistem bermanfaat bagi *user* ? | - | - | - | 2 | 1 |
| 7 | Apakah sistem ini sudah sesuai dengan kebutuhan *user* ? | - | - | - | 1 | 2 |
| 8 | Apakah sistem dapat mempermudah dalam pengelolaan data korban sesuai yg dibutuhkan? | - | - | - | 1 | 2 |
| 9 | Apakah tidak terdapat *error* saat sistem dijalankan? | - | - | - | 2 | 1 |
| 10 | Apakah tidak terdapat menu yang tidak berjalan saat sistem dijalankan? | - | - | - | 2 | 1 |
| 11 | Apakah sistem ini bersifat *user friendly* (mudah digunakan) ? | - | - | - | 1 | 2 |
| 12 | Apakah sistem ini menghasilkan laporan data kerusakan yang dialami korban ? | - | - | - | 1 | 2 |
| 13 | Apakah laporan yang dihasilkan sudah sesuai dengan yang diharapkan ? | - | - | - | 1 | 2 |

Berdasarkan hasil pernyataan yang telah diperoleh dari para responden, maka dapat diketahui nilai dari *user acceptance test* (UAT) sebagai berikut:

Tidak tahu (TT) = 0
Tidak Setuju (TS) = 0
Kurang Setuju (KS) = 0
Setuju (S) = 17
Sangat Setuju (SS) = 22

Kemudian jumlah nilai yang diperoleh dikalikan dengan skala penilaian UAT pada Tabel 3.34 pada bab sebelumnya dan dibagi dengan total jumlah nilai dari semua kuisioner.

$$\text{Hasil pengujian } = \frac{0+0+0+(4x17)+(5x22)}{195} x\ 100\ \%$$

$$= \frac{178}{195} x\ 100\ \%$$

$$= 91.2\ \%$$

Berdasarkan hasil perhitungan nilai *user acceptance test* diatas, maka dapat dinyatakan bahwa sistem yang telah dibuat sudah memperoleh persentase 91.2 % yang bermakna bahwa sistem dari sisi *admin* sudah baik untuk digunakan dalam pendataan dan pengolahan data korban bencana alam gempa.

2. Pengujian sistem dari sisi *team survey*

Tabel 4.6 Hasil pengujian UAT dari sisi *team survey*.

| **No** | **Pernyataan** | **Skala penilaian** | | | | |
|---|---|---|---|---|---|---|
| | | **TT** | **TS** | **KS** | **S** | **SS** |
| 1 | Apakah tampilan sistem nyaman dilihat ? | - | - | - | 2 | 1 |
| 2 | Apakah sistem nyaman digunakan ? | - | - | - | 1 | 2 |
| 3 | Apakah penggunaan menu atau fitur menu dalam sistem mudah dimengerti ? | - | - | - | 2 | 1 |
| 4 | Apakah sistem ini dapat membantu kinerja *user* ? | - | - | - | 1 | 2 |
| 5 | Apakah sistem dapat mempercepat pekerjaan *user* ? | - | - | - | 2 | 1 |
| 6 | Apakah informasi yang disediakan sistem bermanfaat bagi *user* ? | - | - | - | 2 | 1 |
| 7 | Apakah sistem ini sudah sesuai dengan kebutuhan *user* ? | - | - | - | 1 | 2 |
| 8 | Apakah sistem dapat mempermudah dalam pengelolaan data korban sesuai yg dibutuhkan? | - | - | - | 2 | 1 |
| 9 | Apakah tidak terdapat *error* saat sistem dijalankan? | - | - | - | 2 | 1 |
| 10 | Apakah tidak terdapat menu yang tidak berjalan saat sistem dijalankan? | - | - | - | 2 | 1 |
| 11 | Apakah sistem ini bersifat *user friendly* (mudah digunakan) ? | - | - | - | 1 | 2 |

Berdasarkan hasil pernyataan yang telah diperoleh dari para responden, maka dapat diketahui nilai dari *user acceptance test* (UAT) sebagai berikut:

Tidak tahu (TT) = 0

Tidak Setuju (TS) = 0

Kurang Setuju (KS) = 0

Setuju (S) = 18

Sangat Setuju (SS) = 15

Kemudian jumlah nilai yang diperoleh dikalikan dengan skala penilaian UAT pada Tabel 3.34 pada bab sebelumnya dan dibagi dengan total jumlah nilai dari semua pernyataan.

$$\text{Hasil pengujian} = \frac{0+0+0+(4x18)+(5x15)}{165} x\ 100\ \%$$

$$= \frac{147}{165} x\ 100\ \%$$

$$= 89\ \%$$

Berdasarkan hasil perhitungan nilai *user acceptance test* diatas, maka dapat dinyatakan bahwa sistem yang telah dibuat sudah memperoleh persentase 89 % yang bermakna bahwa sistem dari sisi *team survey* sudah cukup baik digunakan walau masih ada beberapa fitur yang perlu ditambahkan untuk dapat digunakan secara maksimal[14], [17].

### 4.2.3 Pengujian Perbandingan Perhitungan SAW Manual dengan SAW pada Sistem

Berikut ini merupakan perbandingan nilai dari hasil perhitungan SAW manual dengan nilai dari hasil perhitungan SAW pada sistem dapat lebih jelasnnya dilihat pada tabel 4.7.

Tabel 4.7 Perbandingan nilai SAW sistem dan SAW manual.

| No | NIK | Nama | Nilai (sistem) | Nilai (manual) | Kesimpulan |
|---|---|---|---|---|---|
| 1 | 5201090107710036 | H. Mahyudi | 0.38554 | 0.38554 | Sama |
| 2 | 5201147112600087 | Nuriah | 0.586754 | 0.586754 | Sama |
| 3 | 520108107670066 | Mahyudin | 0.359672 | 0.359672 | Sama |
| 4 | 5201091304870002 | M. Safri | 0.805673 | 0.805673 | Sama |
| 5 | 5201142302910003 | Zahrul M | 0.907546 | 0.907546 | Sama |
| 6 | 5201033112450003 | Amaq Munasih | 0.878023 | 0.878023 | Sama |
| 7 | 5201081805630001 | Baharudin | 0.218316 | 0.218316 | Sama |
| 8 | 5201154502900001 | Devi Hediana | 0.211018 | 0.211018 | Sama |
| 9 | 5201131404750001 | Rahman | 0.275689 | 0.275689 | Sama |
| 10 | 5201083112890058 | Maesur Haris | 0.328624 | 0.328624 | Sama |

Dari Tabel 4.7 diatas dapat diambil kesimpulan perbandingan hasil dari perhitungan sistem dengan perhitungan manual yang telah diperoleh menyatakan bahwa dari 10 data sampel yang dinilai kebenarannya didapatkan kesamaan sebesar $\frac{10}{10}$ x 100% sama dengan

100% oleh karena itu diambil kesimpulan perbandingan hasil keputusan dari perhitungan manual maupun sistem sepenuhnya sama.

# BAB V

# PENUTUP

## 5.1 Kesimpulan

Setelah dilakukan penelitian mengenai "Implementasi Metode Simple Additive Weighting (Saw) Dalam Sistem Pendukung Keputusan Penentuan Penerima Bantuan Korban Bencana Alam Gempa (Studi Kasus: BPBD Lombok Barat)" dapat diambil beberapa kesimpulan sebagai berikut:

1. Pegawai yang bertindak sebagai *user admin* maupun *team survey* dilembaga BPBD dapat melakukan pendataan korban bencana alam gempa secara online menggunakan sistem yang telah dibuat.
2. Dari hasil pengujian *white box* yang telah dilakukan dapat disimpulkan bahwa algoritma *simple additive weighting* yang digunakan dalam menentukan kriteria kerusakan rumah korban bencana alam gempa sudah berajalan dan dapat menghasilkan nilai sesuai yang diharapkan.
3. Dari hasil pengujian *black box* yang telah dilakukan menjelaskan bahwa setiap fungsi dari masing-masing fitur yang ada pada *user* admin dan *user* team survey berhasil berjalan sesuai dengan yang direncanakan.
4. Dari hasil pengujian UAT menunjukkan untuk dari sisi admin dan team survey diperoleh 91.2 % dan 89 % yang menyatakan bahwa sistem sudah baik untuk digunakan dalam pendataan dan pengolahan data korban dengan saran penambahan beberapa fitur yang perlu ditambahkan demi untuk dapat digunakan secara maksimal.
5. Dari hasil pengujian perbandingan nilai perhitungan SAW manual dengan nilai perhitungan SAW pada sistem didapatkan persentase kesamaan hasil sebesar 100%.

## 5.2 Saran

Setelah dilakukan penelitian mengenai "Implementasi Metode Simple Additive Weighting (Saw) Dalam Sistem Pendukung Keputusan Penentuan Penerima Bantuan Korban Bencana Alam Gempa (Studi Kasus: BPBD Lombok Barat)", peneliti mengambil beberapa point saran sebagai berikut:

1 Dalam pengembangan kedepannya sistem ini diharapkan bisa digunakan dalam kasus yang berbeda dengan mengganti metode yang digunakan.

2 Untuk fitur dalam sistem ini kedepannya peneliti bisa menambahkan fitur yang dapat memudahkan komunikasi antara team survey dengan admin seperti halnya fitur chat.

3 Dalam penelitian ini masih diperlukan pengembangan lanjutan kedepannya seperti halnya penambahan fitur lapor bagi masyarakat yang belum terdata.

## DAFTAR PUSTAKA


[1] M. Mustakim and E. Apriyanto, "Sistem Pendukung Keputusan Penentuan Penerima Bantuan Keuangan Korban Bencana Alam Menggunakan Metode Analytical Hierarchy Process," *Semin. Nas. Apl. Teknol. Inf. (SN ATI)*, pp. 1–6, 2014.

[2] M. H. Pradono, "KAJIAN KERENTANAN BANGUNAN PASCA GEMPA LOMBOK 5 AGUSTUS 2018," *J. Alami J. Teknol. Reduksi Risiko Bencana*, vol. 2, no. 2, p. 82, 2018, doi: 10.29122/alami.v2i2.3109.

[3] S. Gunawan, "Sistem Pendukung Keputusan Pemilihan Guru Terbaik Pada Sma Negeri 2 Kutacane Dengan Menggunakan Metode Simple Additive Weighting (Saw)," *Pelita Inform. Budi Darma*, vol. 9, no. 3, pp. 143–148, 2015.

[4] M. Yusup, "Kredit Pada Nasabah Dengan Metode Simple Additive Weighting ( Studi Kasus Pt . Bprs Al Washliyah Medan )," *Pelita Inform. Budi Darma*, vol. 9, no. 2, pp. 182–188, 2015.

[5] O. Tp and B. Butar, "Sistem Pendukung Keputusan Rekomendasi Penerima Bantuan Siswa Miskin (BSM) Dengan Metode Simple Additive Weighting (SAW) (Studi Kasus : SMPN2 Tarabintang)," *Pelita Inform. Budi Darma*, vol. 10, no. April, p. 3, 2015, [Online]. Available: http://www.stmik-budidarma.ac.id//.

[6] B. N. Aini, I. B. K. Widiartha, and R. Afwani, "Implementasi Metode Logika Fuzzy Simple Additive Weighting (SAW) Dalam Pencarian Rumah Kos Terbaik Di Sekitar Universitas Mataram Berbasis Website," *J. Comput. Sci. Informatics Eng.*, vol. 1, no. 1, p. 41, 2018, doi: 10.29303/jcosine.v1i1.71.

[7] Yakub, *Pengantar Sistem Informasi*. Yoyakarta: Graha Ilmu, 2012.

[8] H. Heri, *Cara Instan Menguasai Pemrograman Website Secara Otodidak*. Jakarta Barat: Agogos Publishing, 2011.

[9] Widodo, *Menggunakan UML*. Bandung: Informatika, 2011.

[10] R. Budi, *Belajar Otodidak Membuat Database Menggunakan MySQL*. Bandung: Informatika, 2006.

[11] A. C. Febryanti, I. Darmawan, and R. Andreswari, "Pemodelan Sistem Pendukung

Keputusan Pemilihan Bidang Peminatan Menggunakan Metode Simple Additive Weighting Studi Kasus : Program Studi Sistem Informasi Universitas Telkom," vol. 4, no. 2, pp. 3114–3121, 2017.

[12] Jogiyanto, *Analisis dan Desain Sistem Informasi*. Yogyakarta: ANDI OFFSET, 1990.

[13] Pressman, *Software Engineering A practitioners's Approach Sixth Edition*. Singapore: McGraw-Hill International Edition, 2004.

[14] D. W. Utomo, D. Kurniawan, and Y. P. Astuti, "Teknik Pengujian Perangkat Lunak Dalam Evaluasi Sistem Layanan Mandiri Pemantauan Haji Pada Kementerian Agama Provinsi Jawa Tengah," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 2, pp. 731–746, 2018, doi: 10.24176/simet.v9i2.2289.

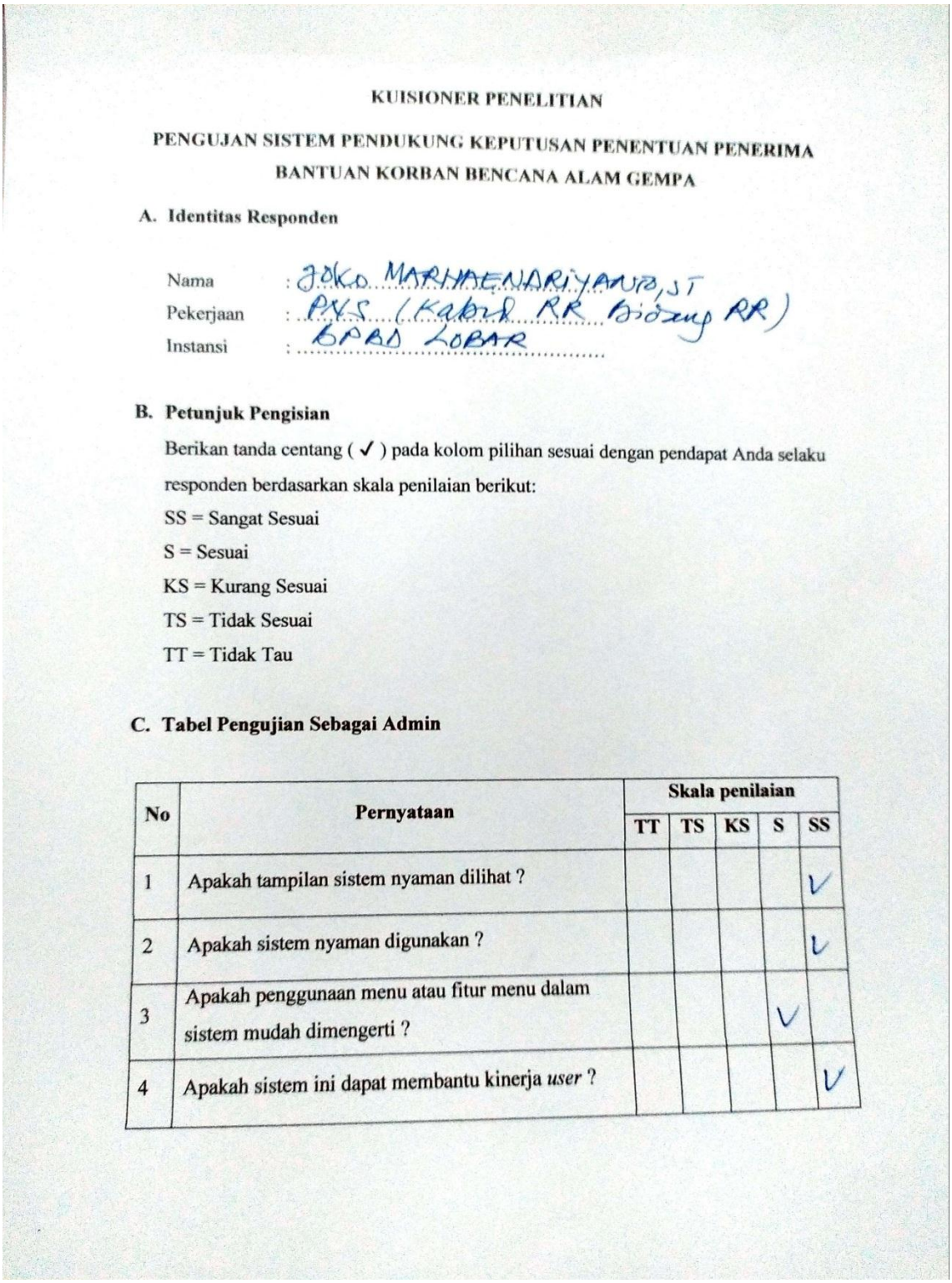
[15] M. Nuris, "White box testing pada sistem penilaian pembelajaran," *Skripsi Tek. Inform.*, pp. 1–102, 2015.

[16] W. Fauzan, "Pembangunan aplikasi game puzzle And The Solver," *Skripsi Tek. Inform.*, pp. 1–96, 2013.

[17] V. V. Wang, A. S. Sukamto, and E. E. Pratama, "Sistem Pendukung Keputusan Seleksi Mahasiswa Penerima Beasiswa BBP-PPA dengan Metode TOPSIS pada Fakultas Teknik UNTAN," *J. Sist. dan Teknol. Inf.*, vol. 7, no. 2, p. 105, 2019, doi: 10.26418/justin.v7i2.29656.

**LAMPIRAN**

Lampiran 1. Pengujian UAT sistem untuk sisi admin.



KUISIONER PENELITIAN

PENGUJAN SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA

**A. Identitas Responden**

Nama : JOKO MARHAENARIYANTO, ST
Pekerjaan : PNS (Kabid RR Bidang RR)
Instansi : BPBD LOBAR

**B. Petunjuk Pengisian**

Berikan tanda centang ( ✓ ) pada kolom pilihan sesuai dengan pendapat Anda selaku responden berdasarkan skala penilaian berikut:

SS = Sangat Sesuai
S = Sesuai
KS = Kurang Sesuai
TS = Tidak Sesuai
TT = Tidak Tau

**C. Tabel Pengujian Sebagai Admin**

| No | Pernyataan | Skala penilaian | | | | |
|---|---|---|---|---|---|---|
| | | TT | TS | KS | S | SS |
| 1 | Apakah tampilan sistem nyaman dilihat ? | | | | | ✓ |
| 2 | Apakah sistem nyaman digunakan ? | | | | | ✓ |
| 3 | Apakah penggunaan menu atau fitur menu dalam sistem mudah dimengerti ? | | | | ✓ | |
| 4 | Apakah sistem ini dapat membantu kinerja *user* ? | | | | | ✓ |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | Apakah sistem dapat mempercepat pekerjaan *user* ? | | | | | ✓ |
| 6 | Apakah informasi yang disediakan sistem bermanfaat bagi *user* ? | | | | | ✓ |
| 7 | Apakah sistem ini sudah sesuai dengan kebutuhan *user* ? | | | | | ✓ |
| 8 | Apakah sistem dapat mempermudah dalam pengelolaan data korban sesuai yg dibutuhkan? | | | | | ✓ |
| 9 | Apakah tidak terdapat *error* saat sistem dijalankan? | | | | | ✓ |
| 10 | Apakah tidak terdapat menu yang tidak berjalan saat sistem dijalankan? | | | | | ✓ |
| 11 | Apakah sistem ini bersifat *user friendly* (mudah digunakan) ? | | | | | ✓ |
| 12 | Apakah sistem ini menghasilkan laporan data kerusakan yang dialami korban ? | | | | | ✓ |
| 13 | Apakah laporan yang dihasilkan sudah sesuai dengan yang diharapkan ? | | | | | ✓ |

Lombok Barat, 15 Mei 2020

Responden

(Joko. M.)

## KUISIONER PENELITIAN

## PENGUJIAN SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA

**A. Identitas Responden**

Nama : M. Zammul Hadi
Pekerjaan : Staf
Instansi : BPBD

**B. Petunjuk Pengisian**

Berikan tanda centang ( ✓ ) pada kolom pilihan sesuai dengan pendapat Anda selaku responden berdasarkan skala penilaian berikut:

SS = Sangat Sesuai
S = Sesuai
KS = Kurang Sesuai
TS = Tidak Sesuai
TT = Tidak Tau

**C. Tabel Pengujian Sebagai Admin**

| No | Pernyataan | Skala penilaian | | | | |
|---|---|---|---|---|---|---|
| | | TT | TS | KS | S | SS |
| 1 | Apakah tampilan sistem nyaman dilihat ? | | | | ✓ | |
| 2 | Apakah sistem nyaman digunakan ? | | | | ✓ | |
| 3 | Apakah penggunaan menu atau fitur menu dalam sistem mudah dimengerti ? | | | | | ✓ |
| 4 | Apakah sistem ini dapat membantu kinerja *user* ? | | | | | ✓ |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | Apakah sistem dapat mempercepat pekerjaan *user* ? | | | | | ✓ |
| 6 | Apakah informasi yang disediakan sistem bermanfaat bagi *user* ? | | | | ✓ | |
| 7 | Apakah sistem ini sudah sesuai dengan kebutuhan *user* ? | | | | ✓ | |
| 8 | Apakah sistem dapat mempermudah dalam pengelolaan data korban sesuai yg dibutuhkan? | | | | ✓ | |
| 9 | Apakah tidak terdapat *error* saat sistem dijalankan? | | | | ✓ | |
| 10 | Apakah tidak terdapat menu yang tidak berjalan saat sistem dijalankan? | | | | ✓ | |
| 11 | Apakah sistem ini bersifat *user friendly* (mudah digunakan) ? | | | | ✓ | |
| 12 | Apakah sistem ini menghasilkan laporan data kerusakan yang dialami korban ? | | | | ✓ | |
| 13 | Apakah laporan yang dihasilkan sudah sesuai dengan yang diharapkan ? | | | | ✓ | |

Lombok Barat,....15-Mei-....2020

Responden

(.....)

**KUISIONER PENELITIAN**

**PENGUJAN SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA**

**A. Identitas Responden**

Nama : ...TOHRI...
Pekerjaan : ...PNS...BPBD...LOBAR...Bid...BARLOG
Instansi : ...B.P.B.D...LOBAR...

**B. Petunjuk Pengisian**

Berikan tanda centang ( ✓ ) pada kolom pilihan sesuai dengan pendapat Anda selaku responden berdasarkan skala penilaian berikut:

SS = Sangat Sesuai
S = Sesuai
KS = Kurang Sesuai
TS = Tidak Sesuai
TT = Tidak Tau

**C. Tabel Pengujian Sebagai Admin**

| No | Pernyataan | Skala penilaian | | | | |
|---|---|---|---|---|---|---|
| | | TT | TS | KS | S | SS |
| 1 | Apakah tampilan sistem nyaman dilihat ? | | | | | ✓ |
| 2 | Apakah sistem nyaman digunakan ? | | | | ✓ | |
| 3 | Apakah penggunaan menu atau fitur menu dalam sistem mudah dimengerti ? | | | | ✓ | |
| 4 | Apakah sistem ini dapat membantu kinerja *user* ? | | | | | ✓ |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | Apakah sistem dapat mempercepat pekerjaan *user* ? | | | | ✓ | |
| 6 | Apakah informasi yang disediakan sistem bermanfaat bagi *user* ? | | | | ✓ | |
| 7 | Apakah sistem ini sudah sesuai dengan kebutuhan *user* ? | | | | | ✓ |
| 8 | Apakah sistem dapat mempermudah dalam pengelolaan data korban sesuai yg dibutuhkan? | | | | | ✓ |
| 9 | Apakah tidak terdapat *error* saat sistem dijalankan? | | | | ✓ | |
| 10 | Apakah tidak terdapat menu yang tidak berjalan saat sistem dijalankan? | | | | ✓ | |
| 11 | Apakah sistem ini bersifat *user friendly* (mudah digunakan) ? | | | | | ✓ |
| 12 | Apakah sistem ini menghasilkan laporan data kerusakan yang dialami korban ? | | | | | ✓ |
| 13 | Apakah laporan yang dihasilkan sudah sesuai dengan yang diharapkan ? | | | | | ✓ |

Lombok Barat, 15-MEI-2020

Responden

(.....)

Lampiran 2. Pengujian UAT sistem untuk sisi team survey.

**KUISIONER PENELITIAN**

**PENGUJAN SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA**

**A. Identitas Responden**

Nama : JOKO MARHAENDRIYANTO, ST
Pekerjaan : KABID RR
Instansi : BPBD LOBAR

**B. Petunjuk Pengisian**

Berikan tanda centang ( ✓ ) pada kolom pilihan sesuai dengan pendapat Anda selaku responden berdasarkan skala penilaian berikut:

SS = Sangat Sesuai
S = Sesuai
KS = Kurang Sesuai
TS = Tidak Sesuai
TT = Tidak Tau

**C. Tabel Pengujian Sebagai Team Survey**

| No | Pernyataan | Skala penilaian | | | | |
|---|---|---|---|---|---|---|
| | | TT | TS | KS | S | SS |
| 1 | Apakah tampilan sistem nyaman dilihat ? | | | | | ✓ |
| 2 | Apakah sistem nyaman digunakan ? | | | | | ✓ |
| 3 | Apakah penggunaan menu atau fitur menu dalam sistem mudah dimengerti ? | | | | | ✓ |
| 4 | Apakah sistem ini dapat membantu kinerja *user* ? | | | | | ✓ |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | Apakah sistem dapat mempercepat pekerjaan *user* ? | | | | | ✓ |
| 6 | Apakah informasi yang disediakan sistem bermanfaat bagi *user* ? | | | | | ✓ |
| 7 | Apakah sistem ini sudah sesuai dengan kebutuhan *user* ? | | | | | ✓ |
| 8 | Apakah sistem dapat mempermudah dalam pengelolaan data korban sesuai yg dibutuhkan? | | | | | ✓ |
| 9 | Apakah tidak terdapat *error* saat sistem dijalankan? | | | | | ✓ |
| 10 | Apakah tidak terdapat menu yang tidak berjalan saat sistem dijalankan? | | | | | ✓ |
| 11 | Apakah sistem ini bersifat *user friendly* (mudah digunakan) ? | | | | | ✓ |

Lombok Barat, 15 Mei 2020

Responden

(JOKO. M)

**KUISIONER PENELITIAN**

**PENGUJIAN SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA**

**A. Identitas Responden**

Nama : M. Zaimul Hadi

Pekerjaan : Staf

Instansi : BPBD Lobar

**B. Petunjuk Pengisian**

Berikan tanda centang ( ✓ ) pada kolom pilihan sesuai dengan pendapat Anda selaku responden berdasarkan skala penilaian berikut:

SS = Sangat Sesuai

S = Sesuai

KS = Kurang Sesuai

TS = Tidak Sesuai

TT = Tidak Tau

**C. Tabel Pengujian Sebagai Team Survey**

| No | Pernyataan | Skala penilaian | | | | |
|---|---|---|---|---|---|---|
| | | TT | TS | KS | S | SS |
| 1 | Apakah tampilan sistem nyaman dilihat ? | | | | ✓ | |
| 2 | Apakah sistem nyaman digunakan ? | | | | ✓ | |
| 3 | Apakah penggunaan menu atau fitur menu dalam sistem mudah dimengerti ? | | | | ✓ | |
| 4 | Apakah sistem ini dapat membantu kinerja *user* ? | | | | ✓ | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | Apakah sistem dapat mempercepat pekerjaan *user* ? | | | | ✓ | |
| 6 | Apakah informasi yang disediakan sistem bermanfaat bagi *user* ? | | | | ✓ | |
| 7 | Apakah sistem ini sudah sesuai dengan kebutuhan *user* ? | | | | ✓ | |
| 8 | Apakah sistem dapat mempermudah dalam pengelolaan data korban sesuai yg dibutuhkan? | | | | ✓ | |
| 9 | Apakah tidak terdapat *error* saat sistem dijalankan? | | | | ✓ | |
| 10 | Apakah tidak terdapat menu yang tidak berjalan saat sistem dijalankan? | | | | ✓ | |
| 11 | Apakah sistem ini bersifat *user friendly* (mudah digunakan) ? | | | | ✓ | |

Lombok Barat, 15 Mei 2020

Responden

(.....)

**KUISIONER PENELITIAN**

**PENGUJAN SISTEM PENDUKUNG KEPUTUSAN PENENTUAN PENERIMA BANTUAN KORBAN BENCANA ALAM GEMPA**

**A. Identitas Responden**

Nama : TOHRI

Pekerjaan : PNS BPBD LOBAR Bid. DARLOG

Instansi : BPBD LOBAR

**B. Petunjuk Pengisian**

Berikan tanda centang ( ✓ ) pada kolom pilihan sesuai dengan pendapat Anda selaku responden berdasarkan skala penilaian berikut:

SS = Sangat Sesuai

S = Sesuai

KS = Kurang Sesuai

TS = Tidak Sesuai

TT = Tidak Tau

**C. Tabel Pengujian Sebagai Team Survey**

| No | Pernyataan | Skala penilaian | | | | |
|---|---|---|---|---|---|---|
| | | TT | TS | KS | S | SS |
| 1 | Apakah tampilan sistem nyaman dilihat ? | | | | ✓ | |
| 2 | Apakah sistem nyaman digunakan ? | | | | | ✓ |
| 3 | Apakah penggunaan menu atau fitur menu dalam sistem mudah dimengerti ? | | | | ✓ | |
| 4 | Apakah sistem ini dapat membantu kinerja *user* ? | | | | | ✓ |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | Apakah sistem dapat mempercepat pekerjaan *user* ? | | | | ✓ | |
| 6 | Apakah informasi yang disediakan sistem bermanfaat bagi *user* ? | | | | ✓ | |
| 7 | Apakah sistem ini sudah sesuai dengan kebutuhan *user* ? | | | | | ✓ |
| 8 | Apakah sistem dapat mempermudah dalam pengelolaan data korban sesuai yg dibutuhkan? | | | | ✓ | |
| 9 | Apakah tidak terdapat *error* saat sistem dijalankan? | | | | ✓ | |
| 10 | Apakah tidak terdapat menu yang tidak berjalan saat sistem dijalankan? | | | | ✓ | |
| 11 | Apakah sistem ini bersifat *user friendly* (mudah digunakan) ? | | | | | ✓ |

Lombok Barat, 15 Mei 2020

Responden

(.....)