



**SISTEM PENUNJANG KEPUTUSAN REKOMENDASI MENU
MAKANAN BERDASARKAN TINGKAT KEBUTUHAN KALORI
TUBUH MENGGUNAKAN METODE *WEIGHTED SUM MODEL* (WSM)**

SKRIPSI

Oleh

Galuh Kusumastuti

NIM 142410101076

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2018



**SISTEM PENUNJANG KEPUTUSAN REKOMENDASI MENU
MAKANAN BERDASARKAN TINGKAT KEBUTUHAN KALORI
TUBUH MENGGUNAKAN METODE *WEIGHTED SUM MODEL* (WSM)**

SKRIPSI

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

HALAMAN JUDUL

Oleh

Galuh Kusumastuti

NIM 142410101076

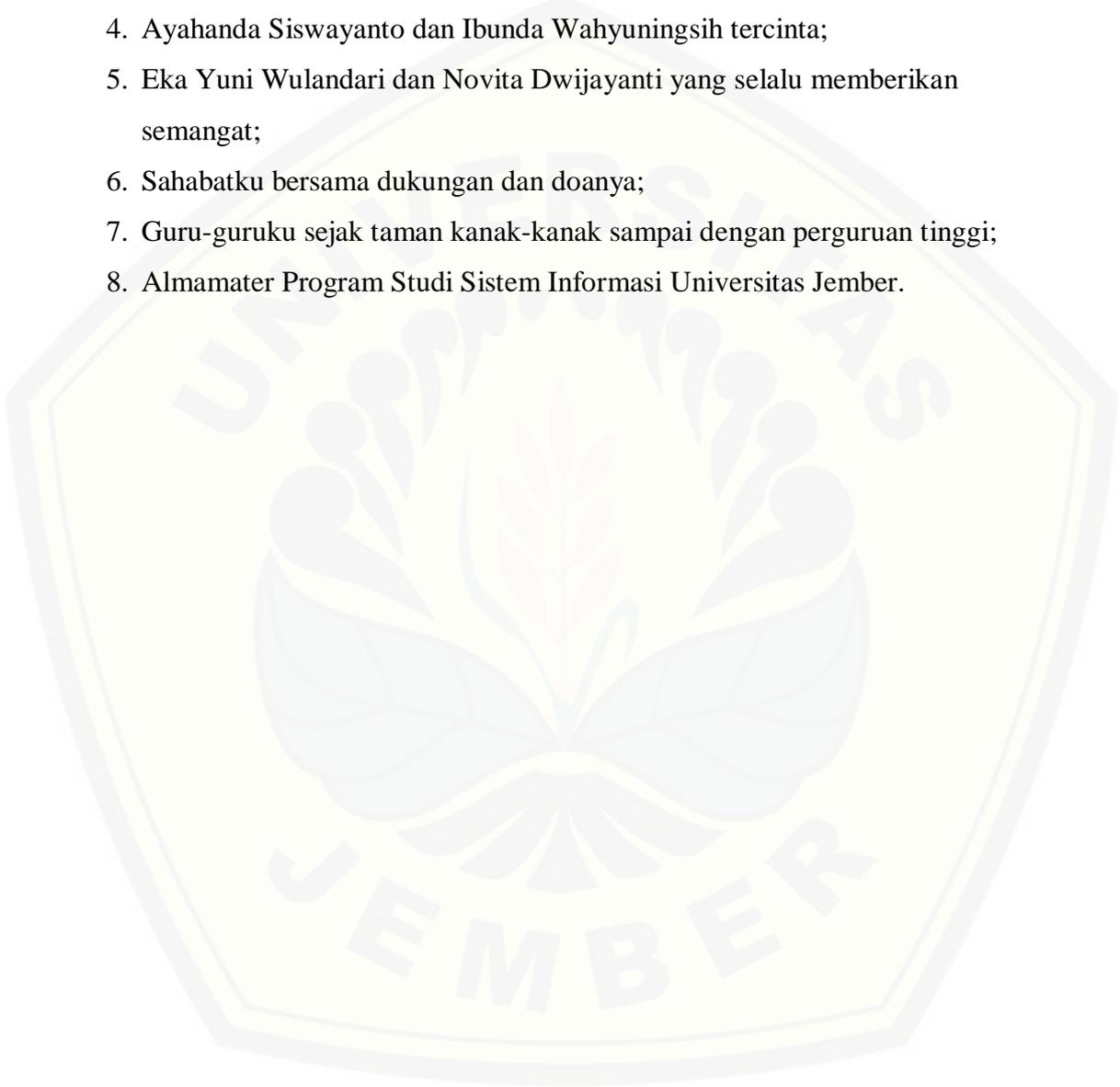
**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS JEMBER**

2018

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda Sutjipto dan (Almh) Ibunda Siswayanti tercinta;
2. Kakak perempuan Rika Krismayani beserta keluarga;
3. Dimas Arifianto yang selalu memberikan dukungan dan semangat;
4. Ayahanda Siswayanto dan Ibunda Wahyuningsih tercinta;
5. Eka Yuni Wulandari dan Novita Dwijayanti yang selalu memberikan semangat;
6. Sahabatku bersama dukungan dan doanya;
7. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
8. Almamater Program Studi Sistem Informasi Universitas Jember.



MOTO

“Cukuplah Allah menjadi penolong kami dan Allah adalah sebaik-baik pelindung”.

(QS. Ali ‘Imron: 173)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Galuh Kusumastuti

NIM : 142410101076

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori Tubuh Menggunakan Metode *Weighted Sum Model* (WSM)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, 16 Juli 2018

Yang menyatakan,

Galuh Kusumastuti

NIM 142410101076

SKRIPSI

**SISTEM PENUNJANG KEPUTUSAN REKOMENDASI MENU
MAKANAN BERDASARKAN TINGKAT KEBUTUHAN KALORI
TUBUH MENGGUNAKAN METODE *WEIGHTED SUM MODEL* (WSM)**

Oleh

Galuh Kusumastuti

NIM 142410101076

Pembimbing :

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

Dosen Pembimbing Pemdamping : Januar Adi Putra, S.Kom, M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori Tubuh Menggunakan Metode *Weighted Sum Model* (WSM)”, telah diuji dan disahkan pada:

hari, tanggal : Senin, 16 Juli 2018

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Drs. Antonius Cahya P, M.App., Sc., Ph.D. Januar Adi Putra, S.Kom, M.Kom.

NIP 196909281993021001

NIP 760017015

PENGESAHAN PENGUJI

Skripsi berjudul “Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori Tubuh Menggunakan Metode *Weighted Sum Model* (WSM)”, telah diuji dan disahkan pada:

hari, tanggal : Senin, 16 Juli 2018

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji:

Penguji I,

Penguji II,

M. Arief Hidayat, S.Kom., M.Kom
NIP. 198101232010121003

Gayatri Dwi Santika, S. SI., M.Kom
NIP. 760017013

Mengesahkan
Dekan Fakultas Ilmu Komputer,

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

RINGKASAN

Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori Tubuh Menggunakan Metode *Weighted Sum Model*; Galuh Kusumastuti, 142410101076; 2018, 143 halaman; Program Studi Sistem Informasi Universitas Jember.

Manusia melakukan aktivitas setiap hari. Aktivitas yang dilakukan berupa aktivitas fisik dan nonfisik. Aktivitas fisik adalah setiap gerakan tubuh yang dihasilkan oleh otot rangka yang memerlukan pengeluaran energi. Dalam melakukan aktivitas kita membutuhkan kalori yang sesuai dengan kebutuhan kalori tubuh. Kebutuhan kalori tubuh setiap orang berbeda-beda, hal ini dipengaruhi oleh aktivitas dan kondisi fisik (tinggi badan, berat badan, jenis kelamin dan usia). Kebutuhan kalori tubuh dapat dihitung dengan cara mengakumulasi beban kerja dan metabolisme tubuh. Setelah mendapatkan kebutuhan kalori tubuh, maka kita dapat membandingkan antara kalori yang dibutuhkan tubuh dengan kalori yang terdapat pada makanan. Perbandingan tersebut menghasilkan makanan yang sesuai dengan kebutuhan kalori tubuh. Akan tetapi, kebiasaan yang dilakukan oleh setiap orang yaitu mengkonsumsi makanan yang hanya membuat kenyang saja tidak mengetahui apakah makanan yang dikonsumsi sesuai dengan kebutuhan kalori tubuh atau tidak. Berdasarkan permasalahan tersebut, maka perlu adanya suatu sistem yang dapat merekomendasi menu makanan yang sesuai dengan kebutuhan kalori tubuh, sehingga dapat membantu pihak masyarakat dalam memilih makanan yang tepat dan sesuai dengan kebutuhan kalori.

Rekomendasi menu makanan ini menggunakan sistem penunjang keputusan dengan menggunakan metode *Weighted Sum Model*. Model keputusan tersebut akan menentukan suatu rekomendasi menu makanan. Rekomendasi tersebut dengan mempertimbangkan tiga kriteria yaitu karbohidrat, protein dan lemak. Hasil dari penelitian ini adalah sistem penunjang keputusan rekomendasi menu makanan.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori Tubuh Menggunakan Metode *Weighted Sum Model* (WSM)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

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

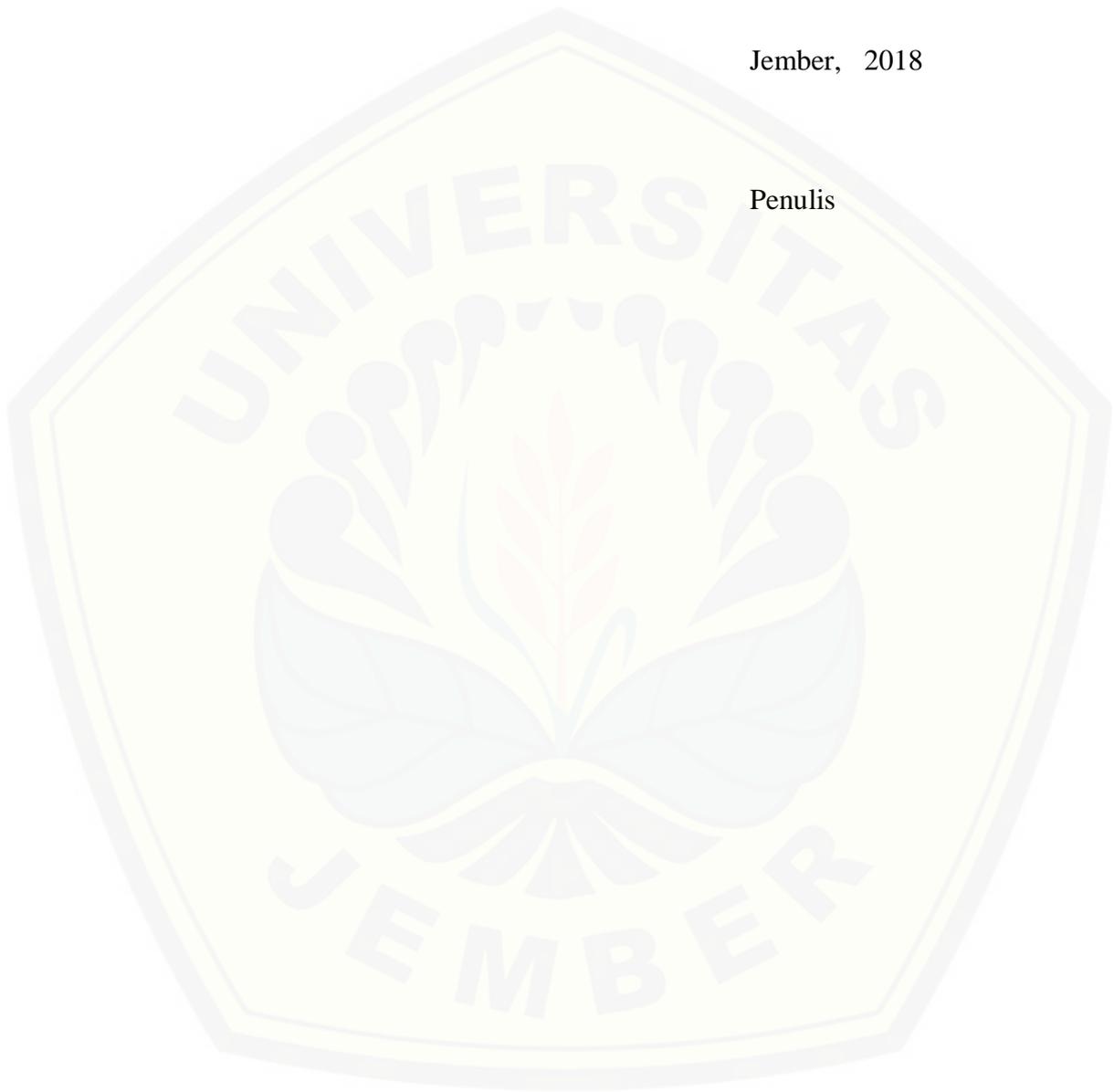
1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Drs. Antonius Cahya P, M.App. Sc., Ph.D., selaku Dosen Pembimbing Utama dan Januar Adi Putra, S.Kom, M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. M. Arief Hidayat, S.Kom, M.Kom. sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa;
4. Ayahanda Sutjipto dan (Almh) Ibunda Siswayanti tercinta;
5. Kakak perempuan Rika Krismayani beserta keluarga;
6. Dimas Arifianto yang selalu memberikan dukungan, motivasi, dan semangat;
7. Ayahanda Siswayanto dan Ibunda Wahyuningsih tercinta;
8. Eka Yuni Wulandari dan Novita Dwijayanti yang selalu memberikan semangat;
9. Sahabatku Grup Bermanfaat yang selalu menemani dan memberikan semangat serta doa Laily Monica, Ayu Permata, Gheseila Atika Sire, Shofiana Nur Aini;
10. Teman-teman seperjuangan SENSATION angkatan 2014 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
11. Kelompok KKN 04 Banyuputih yang telah menjadi keluarga kecil bagi penulis selama KKN.
12. Keluarga besar Himpunan Mahasiswa Sistem Informasi (HIMASIF) periode 2015-2016

13. Semua pihak yang tidak dapat disebutkan satu-persatu.

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

Jember, 2018

Penulis



DAFTAR ISI

HALAMAN JUDUL	ii
PERSEMBAHAN.....	iii
MOTO.....	iv
PERNYATAAN	v
PENGESAHAN PEMBIMBING	vii
PENGESAHAN PENGUJI.....	viii
RINGKASAN.....	ix
PRAKATA	x
DAFTAR ISI	xii
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Batasan Masalah	3
1.5 Sistematika Penulisan.....	3
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terdahulu	5
2.2 Beban Kerja	6
2.3 Kebutuhan Kalori Tubuh.....	8
2.4 Sistem Pendukung Keputusan	11
2.5 Metode <i>Weighted Sum Model</i>	12
BAB 3 METODOLOGI PENELITIAN	14

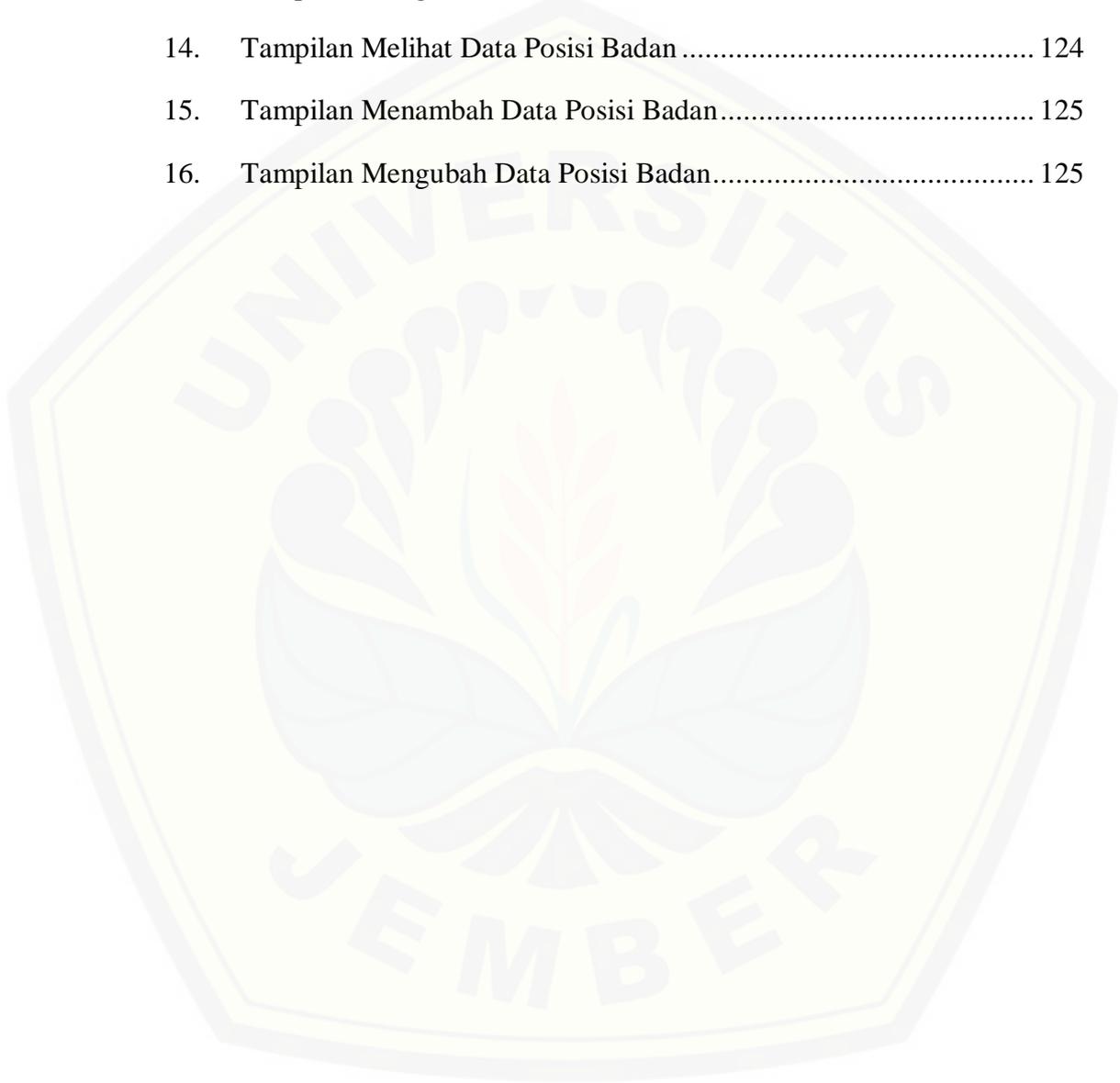
3.1 Jenis Penelitian	14
3.2 Pengembangan Sistem.....	14
3.2.1. Analisis Kebutuhan	15
3.2.2. Analisis Data	15
3.2.3. Desain Sistem.....	15
3.2.4. Implementasi	16
3.2.5. Pengujian Pengembangan Sistem.....	17
3.2.6. Pemeliharaan	17
3.3 Analisis Tahap Penerapan Metode.....	17
BAB 4 PENGEMBANGAN SISTEM.....	19
4.1 Deskripsi Umum Sistem.....	19
4.2 Analisis Kebutuhan Data dan Sistem.....	19
4.2.1 Kebutuhan Fungsional	20
4.2.2 Kebutuhan Non-Fungsional	21
4.3 Desain Sistem	21
4.3.1 Business Process.....	21
4.3.2 <i>Usecase Diagram</i>	22
4.3.3 Skenario	25
4.3.4 <i>Sequence Diagram</i>	32
4.3.5 <i>Activity Diagram</i>	41
4.3.6 <i>Class Diagram</i>	47
4.3.7 <i>Entity Relationship Diagram (ERD)</i>	49
4.4 Implementasi.....	51
4.5 Pengujian Sistem.....	54
BAB 5 HASIL DAN PEMBAHASAN	59
5.1 Hasil Perhitungan Weighted Sum Model.....	60

5.1.1	Data Pengguna.....	60
5.1.2	Data Makanan	60
5.1.3	Bobot Kriteria.....	61
5.1.4	Perhitungan Manual Metode <i>Weighted Sum Model</i>	62
5.1.5	Perhitungan Sistem Metode <i>Weighted Sum Model</i>	72
5.2	Hasil Implementasi Coding Pada Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori	74
5.3.1.	Tampilan Login	74
5.3.2.	Tampilan Menambah Data Pengguna Baru	75
5.3.3.	Tampilan Melihat Data Pengguna	76
5.3.4.	Tampilan Menambah Data Pengguna.....	76
5.3.5.	Tampilan Mengubah Data Pengguna	77
5.3.6.	Tampilan Melihat Data Makanan.....	77
5.3.7.	Tampilan Menambah Data Makanan	77
5.3.8.	Tampilan Mengubah Data Makanan	77
5.3.9.	Tampilan Lihat Rekomendasi Menu Makanan	78
5.3.10	Tampilan Melihat Data Kombinasi Makanan	78
5.3.11.	Tampilan Menambah Data Kombinasi Makanan	79
5.3.12.	Tampilan Mengubah Data kombinasi Makanan.....	79
5.3.13.	Tampilan Lihat Laporan Rekomendasi	79
5.3.14.	Tampilan Melihat Data Aktivitas	80
5.3.15.	Tampilan Menambah Data Aktivitas	80
5.3.16.	Tampilan Mengubah Data Aktivitas	80
5.3.17.	Tampilan Menambah Data Aktivitas	80
5.3.18.	Tampilan Melihat Data Posisi Badan.....	81
5.3.19.	Tampilan Menambah Data Posisi Badan	81

5.3.20. Tampilan Mengubah Data Posisi Badan	81
5.3 Hasil Pembahasan Metode Weighted Sum Model pada Sistem Penunjang Keputusan Rekomendasi Makanan.....	82
5.2.1. Kelebihan Sistem.....	83
5.2.2. Kelemahan Sistem.....	84
BAB 6 PENUTUP	85
6.1. Kesimpulan.....	85
6.2. Saran.....	86
DAFTAR PUSTAKA	87
LAMPIRAN	88
LAMPIRAN A. SKENARIO	88
1. Skenario Masuk.....	88
2. Skenario Menambah Data Pengguna Baru	89
3. Skenario Mengelola Data Pengguna	90
4. Mengelola Data Makanan	92
5. Mengelola Data Kombinasi Makanan	93
6. Skenario Melihat Laporan Rekomendasi Menu Makanan	95
7. Mengelola Data Aktivitas	96
8. Menambah Data Aktivitas	98
9. Mengelola Data Posisi Badan	99
10. Skenario Keluar.....	102
LAMPIRAN B. <i>SEQUENCE</i> DIAGRAM.....	102
1. <i>Sequence</i> Diagram Masuk.....	102
2. <i>Sequence</i> Diagram Menambah Data Pengguna Baru.....	102
3. <i>Sequence</i> Diagram Mengelola Data Pengguna	103
4. <i>Sequence</i> Diagram Mengelola Data Makanan	104

5.	<i>Sequence</i> Diagram Mengelola Data Kombinasi Makanan	105
6.	<i>Sequence</i> Diagram Melihat Laporan Rekomendasi Menu Makanan	105
7.	<i>Sequence</i> Diagram mengelola Data Aktivitas.....	106
8.	<i>Sequence</i> Diagram Menambah Data Aktivitas	107
9.	<i>Sequence</i> Diagram Mengelola Data Posisi Badan	108
10.	<i>Sequence</i> Diagram Keluar.....	109
LAMPIRAN C. <i>ACTIVITY</i> DIAGRAM		110
1.	<i>Activity</i> Diagram Masuk	110
2.	<i>Activity</i> Diagram Menambah Pengguna Baru	111
3.	<i>Activity</i> Diagram Mengelola Data Pengguna	112
4.	<i>Activity</i> Diagram Mengelola Data Makanan.....	113
5.	<i>Activity</i> Diagram Mengelola Data Kombinasi Makanan	114
6.	<i>Activity</i> Diagram Melihat Laporan Rekomendasi Menu Makanan ...	115
7.	<i>Activity</i> Diagram Mengelola Data Aktivitas	115
8.	<i>Activity</i> Diagram Menambah Data Aktivitas	116
9.	<i>Activity</i> Diagram Mengelola Data Posisi Badan	117
10.	<i>Activity</i> Diagram Keluar	117
LAMPIRAN D. TAMPILAN SISTEM.....		118
1.	Tampilan Melihat Data Pengguna	118
2.	Tampilan Menambah Data Pengguna.....	118
3.	Tampilan Mengubah Data Pengguna	119
4.	Tampilan Melihat Data Makanan.....	120
5.	Tampilan Menambah Data Makanan	120
6.	Tampilan Mengubah Data Makanan	121
7.	Tampilan Melihat Data Kombinasi Makanan	121
8.	Tampilan Menambah Data Kombinasi Makanan.....	122

9.	Tampilan Mengubah Data Kombinasi Makanan	122
10.	Tampilan Melihat Laporan Rekomendasi Menu Makanan	123
11.	Tampilan Melihat Data Aktivitas	123
12.	Tampilan Menambah Data Aktivitas.....	123
13.	Tampilan Mengubah Data Aktivitas	124
14.	Tampilan Melihat Data Posisi Badan	124
15.	Tampilan Menambah Data Posisi Badan.....	125
16.	Tampilan Mengubah Data Posisi Badan.....	125



DAFTAR GAMBAR

Gambar 2. 1 Teknik Pemberian Rekomendasi	12
Gambar 3. 1 Model <i>Waterfall</i>	14
Gambar 3. 2 Analisis Tahap Penerapan Metode	18
Gambar 4. 1 Business Process.....	22
Gambar 4. 2 Usecase Diagram	23
Gambar 4.3 Sequence Diagram Melihat Hasil Rekomendasi Menu Makanan	36
Gambar 4. 4 Activity Diagram Melihat Hasil Rekomendasi Menu Makanan	44
Gambar 4. 5 Class Diagram	48
Gambar 4. 6 Entity Relationship Diagram	50
Gambar 4. 7 Kode Program c_rekom	53
Gambar 4. 8 Kode Program m_aktivitas.....	53
Gambar 4. 9 Kode Program M_kombinasi	54
Gambar 4. 10 Kode Program M_rekom.....	54
Gambar 5. 1 Formulir Data Kontekstual.....	72
Gambar 5. 2 Hasil Analisa Kalori.....	73
Gambar 5. 3 Rekomendasi Menu Makanan A	74
Gambar 5. 5 Tampilan Login	75
Gambar 5. 6 Tampilan Menambah Data Pengguna Baru	76
Gambar 5. 7 Tampilan Hasil Rekomendasi Menu Makanan	78
Gambar 5. 4 Hasil Rekomendasi A	83

DAFTAR TABEL

Tabel 2. 1 Perkiraan Beban Kerja Menurut Kebutuhan Energi.....	8
Tabel 2. 2 Faktor Aktivitas.....	10
Tabel 3. 1 <i>Requirements Testing</i>	17
Tabel 4. 1 Data Kontekstual.....	20
Tabel 4. 2 Data Makanan	20
Tabel 4. 3 Definsi Aktor.....	24
Tabel 4. 4 Usecase	24
Tabel 4. 5 Skenario Melihat Hasil Rekomendasi Menu Makanan	28
Tabel 4. 6 Haril Requirements Testing	55
Tabel 5.1 Data Kontekstual Pengguna.....	60
Tabel 5.2 Daftar Alternatif Makanan.....	61
Tabel 5.3 Bobot Kriteria	62
Tabel 5.4 Nilai Performa Karbohidrat	63
Tabel 5.5 Nilai Performa Protein.....	64
Tabel 5.6 Nilai Performa Lemak	65
Tabel 5.7 Perhitungan Nilai WSM	67
Tabel 5.8 Hasil Rekomendasi A	68
Tabel 5.9 Nilai Performa Kalori B	69
Tabel 5.10 Perhitungan Nilai WSM B	70
Tabel 5.11 Hasil Rekomendasi Menu Makanan B	71

BAB 1 PENDAHULUAN

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

1.1 Latar Belakang

Manusia melakukan aktivitas setiap hari. Aktivitas yang dilakukan berupa aktivitas fisik dan nonfisik. Aktivitas fisik adalah setiap gerakan tubuh yang dihasilkan oleh otot rangka yang memerlukan pengeluaran energi. Kurangnya aktivitas fisik merupakan faktor resiko independen untuk penyakit kronis, dan secara keseluruhan diperkirakan menyebabkan kematian secara global (WHO, 2010). Dalam melakukan aktivitas membutuhkan asupan energi yang seimbang. Asupan energi diperoleh dari makanan yang kita konsumsi, karena hal tersebut merupakan kebutuhan terpenting untuk setiap orang guna kelangsungan hidupnya juga menyongkong tubuh dalam melakukan berbagai aktivitas. Makanan yang harus dikonsumsi adalah makanan yang seimbang. Makanan yang seimbang merupakan makanan yang sesuai dengan jumlah kebutuhan kalori tubuh. Kebutuhan kalori tubuh setiap orang berbeda-beda, hal ini dipengaruhi oleh aktivitas dan kondisi fisik (tinggi badan, berat badan, jenis kelamin dan usia). Oleh karena itu, untuk memenuhi kebutuhan kalori perlu adanya pola makan yang tepat dengan mempertimbangkan kandungan karbohidrat, protein dan lemak agar tidak terjadi permasalahan kesehatan.

Permasalahan kesehatan yang kerap terjadi salah satunya adalah masalah tentang gizi. Masalah gizi dibagi menjadi dua bagian yakni kurang gizi dan kelebihan gizi. Kurang gizi merupakan masalah kesehatan akibat dari tubuh tidak mendapat asupan nutrisi yang cukup untuk melakukan aktivitas, sedangkan kelebihan gizi adalah masalah kesehatan yang diakibatkan oleh kelebihan energi yang diperoleh dari mengonsumsi makanan melebihi jumlah kebutuhan energi saat melakukan aktivitas. Obesitas merupakan salah satu contoh masalah kelebihan gizi yang disebabkan oleh ketidak seimbangan antara jumlah energi yang masuk dengan yang dibutuhkan oleh tubuh untuk berbagai fungsi biologis seperti pertumbuhan

fisik, perkembangan, aktivitas, pemeliharaan kesehatan (Jahari, 2004). Agar tidak mengalami masalah kesehatan, maka kita harus mengkonsumsi makanan dengan gizi yang seimbang dengan jumlah yang sesuai dengan kebutuhan kalori tubuh.

Kebutuhan kalori tubuh dapat dihitung dengan cara mengakumulasi beban kerja dan metabolisme tubuh. Setelah mendapatkan kebutuhan kalori tubuh, maka kita dapat membandingkan antara kalori yang dibutuhkan tubuh dengan kalori yang terdapat pada makanan. Perbandingan tersebut menghasilkan makanan yang sesuai dengan kebutuhan kalori tubuh. Akan tetapi, kebiasaan yang dilakukan oleh setiap orang yaitu mengkonsumsi makanan yang hanya membuat kenyang saja tidak mengetahui apakah makanan yang dikonsumsi sesuai dengan kebutuhan kalori tubuh atau tidak. Oleh karena itu, dalam upaya mengatasi masalah tersebut maka perlu adanya suatu sistem penunjang keputusan rekomendasi makanan untuk memudahkan masyarakat dalam memilih makanan yang tepat dan sesuai dengan kebutuhan karbohidrat, protein dan lemak.

Sistem penunjang keputusan rekomendasi makanan ini menerapkan penggunaan *Multi Criteria Decision Making* dengan menggunakan metode *Weighted Sum Model*. Memilih atau memprioritaskan alternatif dari satu set alternatif yang tersedia dengan kategori yang banyak, biasanya kriteria yang saling bertentangan sering disebut sebagai multi kriteria pengambilan keputusan (MCDM) (Yeh & Deng, 1997). Dalam teori pengambilan keputusan, *Weighted Sum Model* adalah salah satu yang paling dikenal di MCDM (*Multi Criteria Decision Making*) dan juga metode yang paling sederhana untuk mengevaluasi alternatif berdasarkan beberapa kriteria (Ammar, Amelia, & Muhammad, 2016). Kriteria pemilihan yang digunakan pada penelitian ini ada tiga, yaitu karbohidrat, protein dan lemak. Diharapkan dengan adanya sistem penunjang keputusan rekomendasi makanan ini dapat membantu masyarakat yang kesulitan menentukan makanan yang harus dikonsumsi sesuai dengan kebutuhan kalori dalam satu hari.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang ada maka rumusan masalah yang akan dibahas dalam penelitian ini adalah :

1. Bagaimana cara menerapkan metode *Weighted Sum Model* (WSM) dalam rekomendasi menu makanan berdasarkan tingkat kebutuhan kalori tubuh?
2. Bagaimana membangun sistem penunjang keputusan rekomendasi menu makanan berdasarkan kebutuhan kalori tubuh menggunakan metode *Weighted Sum Model* ?

1.3 Tujuan

Tujuan dari penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Menerapkan *Weighted Sum Model* (WSM) dalam aplikasi sistem rekomendasi menu makanan berdasarkan tingkat kebutuhan kalori tubuh.
2. Mengimplementasikan sistem penunjang keputusan pada sistem rekomendasi menu makanan berdasarkan tingkat kebutuhan karbohidrat, protein dan lemak.

1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat sebagai parameter pengerjaan penelitian ini di antaranya sebagai berikut :

1. Objek penelitian ini adalah anak-anak, remaja, dan orang dewasa.
2. Penelitian ini diperuntukkan bagi orang yang sehat dan tidak menjalani terapi diet.
3. Kriteria dalam pemilihan menu makanan ini adalah karbohidrat, protein, dan lemak.
4. Menu makanan dikombinasikan sendiri oleh peneliti.
5. Makanan yang direkomendasikan untuk satu kali makan.
6. Perhitungan kalori menggunakan aktivitas yang akan dilakukan oleh pengguna.
7. Aplikasi ini dibangun berbasis web.

1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

1. Pendahuluan

Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan, dan manfaat, ruang lingkup studi dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

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

4. Pengembangan Sistem

Bab ini menjelaskan tentang tahap pengembangan Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori Tubuh.

5. Hasil dan Pembahasan

Bab ini menjelaskan mengenai hasil dan pembahasan dari penelitian yang telah dilakukan.

6. Penutup

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

BAB 2 TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku, *literature*, jurnal, dan internet.

2.1 Penelitian Terdahulu

Penelitian sebelumnya mengenai penerapan *weighted sum model* dalam penentuan peserta jaminan kesehatan masyarakat. Jaminan kesehatan masyarakat merupakan salah satu program pemerintah dalam hal layanan pengobatan pada puskesmas untuk membantu rakyat dengan berpenghasilan rendah. Indikator pemerintah dalam memberikan layanan ini terdiri dari sepuluh kriteria. Pada penerapannya memiliki kendala dalam memutuskan siapa peserta yang mendapatkan pelayanan jamkesmas tersebut. Oleh karena itu, peneliti memilih menerapkan *Multi-Criteria Decision Making* dengan metode *Weighted Sum Model* yang mampu mengatasi kendala yang dihadapi oleh pemerintah. Dengan menerapkan WSM peneliti meyakini bahwa metode tersebut tergolong mudah, cepat, dan mampu mendapatkan hasil yang terbaik (Mesran & Surya Darma Nasution, 2017).

Penelitian selanjutnya mengenai implementasi metode *k-nearest neighbor* untuk menentukan menu makanan sehat dan bergizi (Afandie, Cholissodin, & Supianto, 2013). Dalam penelitian ini inputan data pada proses awal menentukan pilihan menu yang sesuai terhadap kebutuhan gizi harian pasien dengan mendapatkan informasi kebutuhan gizi harian pasien dengan parameter yang digunakan yaitu tinggi badan, berat badan, usia, jenis kelamin, dan *physical activity*. Proses klasifikasi dilakukan dengan menggunakan metode *k-nearest neighbor* dengan training data dilakukan terhadap 50 data pasien dan 30 paket menu makanan. Inputan data dapat diterapkan dalam penulisan ini.

Penelitian selanjutnya yang dilakukan oleh Ammar Naufal, Amelia kurniawati, dan Muhammad Azani hasibuan, dengan judul *Decision support System of SMB Telkom University Roadshow Location Prioritization with Weighted Sum Model Method*. Pada jurnal tersebut, terdapat tiga kriteria yang menentukan SMA

yang sesuai untuk dikunjungi pada acara roadshow, antara lain: jumlah siswa yang mendaftar pada sekolah tersebut (diidentifikasi sebagai minat siswa), jumlah siswa yang mengundurkan diri (diidentifikasi sebagai komitmen SMA), dan jumlah siswa yang mendapatkan beasiswa unggulan (diidentifikasi sebagai prestasi SMA). Ketiga kriteria pembobotan ini ditentukan oleh manajer pendaftaran, dimana setiap SMA memiliki nilai performanya sendiri dari setiap kriteria. Kriteria ini memiliki bobot dan skor performanya akan menjadi faktor penentu dalam proses perhitungan menggunakan *Weighted Sum Model* (WSM). Proses penghitungan akan berikan hasilnya berupa potensi skor masing-masing sekolah yang akan diurutkan dari yang tertinggi sampai yang terendah (Ammar, Amelia, & Muhammad, 2016). Keterkaitan dengan penelitian yang penulis lakukan yaitu kesamaan permasalahan tentang sistem keputusan yang dilakukan pembobotan kriteria dengan alternatif yang banyak dan solusi yang diberikan oleh penulis menggunakan metode *Weighted Sum Model*.

2.2 Beban Kerja

Beban kerja merupakan beban yang dialami oleh tenaga kerja sebagai akibat pekerjaan yang dilakukan olehnya. Standar penilaian beban kerja berdasarkan tingkat kebutuhan kalori menurut pengeluaran energi menetapkan prinsip penilaian, peralatan penilaian dan prosedur kerja penilaian, pengukuran berat badan, pengamatan aktivitas tenaga kerja dan perhitungan beban kerja tenaga kerja ditempat kerja.

Beban kerja fisik selalu berkaitan dengan pergerakan otot. Salah satu kebutuhan umum dalam pergerakan otot adalah oksigen yang dibawa oleh darah ke otot untuk pembakaran zat dalam menghasilkan energi, dan satuan energi adalah kalori. Sedangkan menghitung kalori adalah menghitung asupan energi. Energi diperoleh dari makanan yang mengandung karbohidrat dan protein.

Dalam penerapannya beban yang dialami oleh tenaga kerja dapat dikategorikan menjadi tiga bagian yaitu (SNI, 7269:2009):

- a. Kerja ringan merupakan pekerjaan yang membutuhkan kalori untuk pengeluaran energi sebesar 100 kkal per jam sampai dengan 200 kkal per jam.

- b. Kerja sedang merupakan pekerjaan yang membutuhkan kalori untuk pengeluaran energi lebih besar dari 200 kkal per jam sampai dengan 350 kkal per jam.
- c. Kerja berat merupakan pekerjaan yang membutuhkan kalori untuk pengeluaran energi lebih besar dari 350 kkal per jam sampai dengan 500 kkal per jam.

Perhitungan beban kerja dihitung dengan memperhatikan komponen – komponen sebagai berikut :

1. Rerata Beban Kerja

Langkah pertama yaitu dengan menghitung rata – rata beban kerja. Perkiraan nilai beban kerja menurut kebutuhan energy dapat dilihat pada Tabel 2.1. Rata – rata beban kerja dihitung dengan menggunakan Persamaan 1.

$$\text{Rerata BK} = \frac{(BK1 \times T1) + (BK2 \times T2) + \dots + (BK_n \times T_n)}{(T1 + T2 + \dots + T_n)} \times 60 \text{ kkal per jam} \quad (1)$$

Keterangan :

BK = beban kerja per jam,

$BK1, BK2, \dots, BK_n$ = beban kerja sesuai aktivitas tenaga kerja 1,2, ... n,

T = waktu dalam satuan menit,

$T1, T2, \dots, T_n$ = waktu sesuai aktivitas tenaga kerja 1,2, ... n,

2. Metabolisme Basal

Langkah kedua yaitu dengan menghitung metabolisme basal tubuh. Metabolisme basal dapat dihitung berdasarkan jenis kelamin. Perhitungan metabolisme basal jenis kelamin laki-laki dapat menggunakan Persamaan 2, sedangkan untuk jenis kelamin perempuan menggunakan Persamaan 3.

$$\text{MB Laki – Laki} = \text{Berat Badan} \times 1 \text{ kkal per jam} \quad (2)$$

$$\text{MB Perempuan} = \text{Berat Badan} \times 0.9 \text{ kkal per jam} \quad (3)$$

3. Total Beban Kerja

Total beban kerja didapat dengan menjumlahkan rata-rata beban kerja dengan metabolisme basal tubuh yang dapat dilihat pada Persamaan 4.

$$\text{Total BK} = \text{Rerata BK} + \text{MB} \quad (4)$$

Tabel 2. 1 Perkiraan Beban Kerja Menurut Kebutuhan Energi

No	Pekerjaan	Posisi Badan			
		1	2	3	4
		Duduk (0.3)	Berdiri (0.6)	Berjalan (3.0)	Berjalan Mendaki (3.8)
1	Pekerjaan dengan tangan				
	Kategori I (contoh : menulis, merajut) (0.30)	0.60	0.90	3.30	4.10
	Kategori II (contoh : menyetrika) (0.70)	1.00	1.30	3.70	4.50
	Kategori III (contoh : mengetik) (1.10)	1.40	1.70	4.10	4.90
2	Pekerjaan dengan satu tangan				
	Kategori I (contoh : menyapu lantai) (0.90)	1.20	1.50	3.90	4.70
	Kategori II (contoh : menggergaji) (1.60)	1.90	2.20	4.60	5.40
	Kategori III (contoh : memukul paku) (1.10)	2.60	2.90	5.30	6.10
3	Pekerjaan dengan dua lengan				
	Kategori I (contoh : menambal logam, mengemas barang) (1.25)	1.55	1.85	4.25	5.05
	Kategori II (contoh : memompa) (2.25)	2.55	2.85	5.25	6.05
	Kategori III (contoh : mendorong) (3.25)	3.55	3.85	6.25	7.05
4	Pekerjaan dengan menggunakan gerakan tangan				
	Kategori I (contoh : pekerjaan administrasi) (3.75)	4.05	4.35	6.75	7.55
	Kategori II (contoh : mengepel) (8.75)	9.05	9.35	11.75	12.55
	Kategori III (contoh : menggali lobang, menebang pohon) (13.75)	14.05	14.35	16.75	17.55

(Sumber : SNI 7269:2009)

Pada Tabel 2.1 merupakan perkiraan beban kerja menurut kebutuhan energi dimana terbagi menjadi empat golongan pekerjaan berdasarkan posisi badan yaitu, pekerjaan dengan tangan, satu tangan, dua lengan, dan gerakan tangan. Golongan pekerjaan memiliki kategori dan rasio aktivitas pada masing-masing golongan pekerjaan yang sesuai dengan posisi badan saat melakukan pekerjaan tersebut.

2.3 Kebutuhan Kalori Tubuh

Kalori merupakan suatu istilah yang digunakan untuk menyatakan banyaknya energi yang dikeluarkan atau dibutuhkan. Energi yang dikeluarkan selama aktifitas fisik dan terkandung dalam makanan diukur dalam kilokalori (kcal). Kebutuhan kalori setiap orang berbeda bergantung pada energi yang digunakan untuk

beraktivitas setiap hari. Kebutuhan energi dihitung dengan memperhatikan komponen – komponen berikut (Nugroho & Ferdiana, 2014):

1. Metabolisme basal (*Basal Metabolic Rate*)

Metabolisme basal merupakan energi minimal yang digunakan saat tubuh beristirahat. Perhitungan metabolisme basal ini dapat dilakukan pada pagi hari saja, apabila dilakukan pada siang hari tidak akan berpengaruh karena metabolisme basal bergantung pada usia, jenis kelamin, berat badan, dan tinggi badan. Oleh karena itu metabolisme setiap orang berbeda-beda. Besarnya energi basal pada pria dapat dilihat pada Persamaan 5, sedangkan untuk wanita dapat dilihat pada Persamaan 6. Dimana angka 66,4730 dan 655,0955 merupakan angka prediksi untuk menghitung metabolisme basal menurut Harris-Benedict

$$h = 66,4730 + 13,7516w + 5,003s - 6,7550a \quad (5)$$

$$h = 655,0955 + 9,5634w + 1,849s - 4,6756a \quad (6)$$

Keterangan :

- h = energi basal (kkal),
- w = berat tubuh (kilogram),
- s = tinggi badan (centimeter),
- a = umur

2. *Specific Dynamic Action* (SDA)

SDA merupakan penggunaan energi ebagai akibat dari makanan. Energi ini digunakan untuk mengolah makanan dalam tubuh. Besarnya SDA diperoleh dari perhitungan dengan Persamaan 7, dimana SDA (10% BMR).

$$sda = 0,1 \times h \quad (7)$$

Keterangan :

- sda = energi SDA (kkal),
- h = energi basal (kkal),

3. Aktivitas Fisik

Aktivitas fisik merupakan aktivitas rutin yang dilakukan sehari-hari. Untuk mengetahui besarnya kalori yang dibutuhkan maka kebutuhan energy harian dapat dihitung dengan Persamaan 8.

$$kal = af \times (h + sda) \quad (8)$$

Keterangan :

kal = kebutuhan kalori harian (kkal),

af = faktor aktivitas,

h = energi basal (kkal),

sda = energi SDA (kkal)

Besarnya faktor aktivitas tergantung dari berat atau ringanya aktivitas yang dilakukan. Tabel 2.2 berisi daftar aktivitas berdasarkan berat atau ringanya kegiatan.

Tabel 2. 2 Faktor Aktivitas

Aktivitas	Pria	Wanita
Ringan	1,5	1,5
Sedang	1,8	1,7
Berat	2,1	1,8

(Sumber : Nugroho & Ferdiana, 2014)

4. Karbohidrat

Karbohidrat memegang peranan penting dalam tubuh karena merupakan sumber kebutuhan utama bagi manusia. Nilai energi karbohidrat adalah 4 kkal per gram (Almatsier, 2009). Untuk memelihara kesehatan WHO menganjurkan 55–75 % konsumsi energi total yang berasal dari karbohidrat kompleks dan yang paling banyak hanya 10% berasal dari gula sederhana. Untuk mendapatkan jumlah karbohidrat yang akan dikonsumsi oleh tubuh dapat dihitung dengan persamaan 9 (Almatsier, 2009).

$$Karbohidrat = \frac{Kal \times 0.6}{4} \quad (9)$$

5. Protein

Protein adalah bagian dari semua sel hidup dan merupakan bagian tubuh terbesar dari tubuh sesudah air. Bahan makanan hewani merupakan sumber protein yang baik, dalam jumlah maupun mutu seperti telur, daging, ikan, dan lainnya. Untuk mendapatkan jumlah protein yang akan dikonsumsi oleh tubuh dapat dihitung dengan persamaan 10. (Almatsier, 2009)

$$Protein = \frac{Kal \times 0.15}{4} \quad (10)$$

6. Lemak

Istilah lemak meliputi senyawa heterogen termasuk lemak dan minyak yang umum dikenal dalam makanan malam, folsofida, sterol dan ikatan lain sejenis yang terdapat didalam makanan dan tubuh manusia. Fungsi lemak adalah sebagai sumber energi, sebagai sumber asam lemak esensial alat angkut vitamin larut lemak, menghemat protein, memberi rasa kenyang, dan kelezatan. (Almatsier, 2009)

$$Lemak = \frac{Kal \times 0.25}{9} \quad (11)$$

2.4 Sistem Pendukung Keputusan

Pendapat beberapa ahli bahwa Sistem Pendukung Keputusan atau *Decision Support System* (DSS) dibuat untuk meningkatkan proses dan kualitas hasil pengambilan keputusan, dimana DSS dapat memadukan data dan pengetahuan untuk meningkatkan efektivitas dan efisiensi dalam proses pengambilan keputusan tersebut, disamping itu Sistem Pendukung Keputusan juga memberdayakan resources individu secara intelek dengan kemampuan komputer untuk meningkatkan kualitas keputusan dan berhubungan dengan manajemen pengambilan keputusan serta berhubungan dengan masalah-masalah yang semi terstruktur (Marthin & Abdul, 2010).

Decision Support System (DSS) atau lebih dikenal Sistem Pendukung Keputusan (SPK) merupakan sistem berbasis komputer yang adaptif, fleksibel, dan interaktif yang digunakan untuk memecahkan masalah-masalah tidak terstruktur sehingga meningkatkan nilai keputusan yang diambil (Afandie, Cholissodin, & Supianto, 2013).

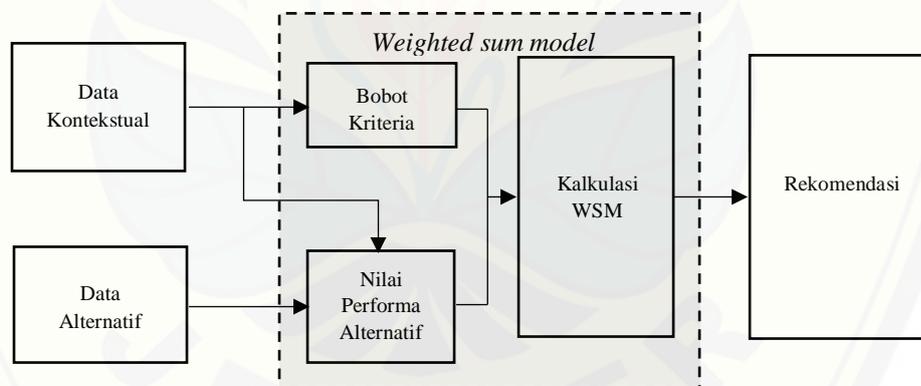
2.5 Metode *Weighted Sum Model*

Weighted Sum Model (WSM) merupakan salah satu *metode Multi Criteria Decision Making*. *Weighted Sum Model* (WSM) menerapkan pendekatan yang paling umum digunakan dalam menentukan alternatif terbaik, terutama dalam masalah pada dimensi tunggal. Metode ini memberikan satu set alternatif pilihan berdasarkan beberapa kriteria pengambilan keputusan. Formula perhitungan Metode *Weighted Sum Model* ditunjukkan pada Persamaan 12:

$$A_i^{SWM-Score} = \sum_{j=1}^n w_j a_{ij}$$

$$\text{for } i = 1, 2, 3, \dots, m \quad (12)$$

$A_i^{SWM-Score}$ merupakan skor WSM dari setiap alternative A_i . w_j merupakan bobot relatif dari kriteria C_j sedangkan a_{ij} merupakan nilai performa dari alternatif A_i dengan kriteria C_j . Alternatif terbaik adalah alternatif dengan nilai $A_i^{SWM-Score}$ terbesar atau maksimal. Konsep pemberian rekomendasi diambil dari penelitian (Nugroho & Ferdiana, 2014) dapat dilihat pada gambar 2.1.



Gambar 2. 1 Teknik Pemeberian Rekomendasi

Dalam penerapan sistem penunjang keputusan rekomendasi menu makanan, alternatif A_i dapat berupa menu makanan. Sedangkan bobot kriteria berupa preferensi pada suatu kriteria keputusan rekomendasi menu makanan. Dalam perhitungan WSM, komponen yang mempengaruhi nilai WSM adalah bobot kriteria dan nilai performa setiap menu makanan untuk setiap kriteria. Perhitungan nilai performa kalori untuk setiap menu makanan, menggunakan Persamaan 13.

$$a_i^{karbohidrat} = \left(1 - \frac{k_i}{\sum_{j=1}^n k_j}\right) \times 100 \quad (13)$$

Dengan,

$a_i^{karbohidrat}$ = nilai performa kriteria karbohidrat untuk alternatif A_i

k_i = besar selisih karbohidrat alternatif A_i dengan kebutuhan karbohidrat pengguna

k_i dapat dihitung dengan mencari nilai absolut dari selisih antara kalori dalam makanan dengan kalori yang dibutuhkan tubuh dengan menggunakan Persamaan 14.

$$k_i = |k_i^{Makanan} - k_i^{butuh}| \quad (14)$$

Perhitungan nilai performa protein untuk setiap menu makanan menggunakan Persaman 15.

$$a_i^{protein} = \left(1 - \frac{p_i}{\sum_{j=1}^n p_j}\right) \times 100 \quad (15)$$

Dengan,

$a_i^{protein}$ = nilai performa kriteria protein untuk alternatif A_i

p_i = protein untuk alternatif A_i

Perhitungan nilai performa karbohidrat untuk setiap menu makanan menggunakan Persaman 16.

$$a_i^{Lemak} = \left(1 - \frac{L_i}{\sum_{j=1}^n L_j}\right) \times 100 \quad (16)$$

Dengan,

a_i^{Lemak} = nilai performa kriteria lemak untuk alternatif A_i

L_i = Lemak untuk alternatif A_i

Setelah bobot kriteria dan nilai perfoma didapatkan, perhitungan WSM dapat dilakukan menggunakan Persamaan 12. Perhitungan WSM menghasilkan nilai WSM, dimana nilai terbesar akan menjadi rekomendasi menu makanan.

BAB 3 METODOLOGI PENELITIAN

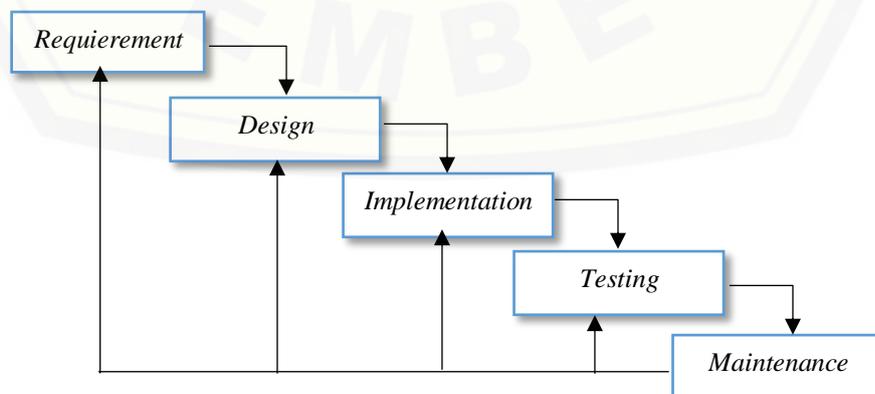
Bab ini menjelaskan tentang gambaran tahapan yang sistematis yang dilakukan untuk menganalisa data untuk menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang jenis penelitian, tahap pengembangan sistem serta tahapan dari penelitian.

3.1 Jenis Penelitian

Jenis penelitian yang akan dilakukan merupakan penelitian pengembangan sistem. Metode pengembangan sistem digunakan untuk menghasilkan suatu produk baru. Produk pada penelitian ini adalah sistem penunjang keputusan rekomendasi menu makanan berdasarkan tingkat kebutuhan kalori tubuh. Penelitian pengembangan dilakukan untuk memudahkan masyarakat mendapatkan rekomendasi menu makanan yang tepat dengan kebutuhan karbohidrat, protein dan lemak.

3.2 Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini adalah *System Development Life Cycle (SDLC)* dengan model *Waterfall*. Model *waterfall* merupakan model yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, kode, test, dan pemeliharaan (Pressman, 2002). Gambaran model *waterfall* dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Model *Waterfall*

3.2.1. Analisis Kebutuhan

Tahap pertama pada proses perancangan perangkat lunak adalah analisis kebutuhan. Analisis kebutuhan merupakan langkah menganalisis kebutuhan-kebutuhan sistem informasi yang akan dibangun. Kebutuhan sistem informasi dibagi menjadi 2 yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan yang didapat berasal dari data yang telah dikumpulkan dan telah diolah yang berhubungan dengan rekomendasi menu makanan yang melibatkan kalori. Data-data tersebut kemudian dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non-fungsional.

Untuk menentukan kebutuhan fungsional dan non-fungsional, maka dilakukan pengumpulan data:

1. Melakukan studi *literature*, jurnal, media maupun internet mengenai pembangunan sistem informasi penunjang keputusan rekomendasi menu makanan seperti data kriteria yakni kalori dan harga, dan data kalori makanan.
2. Penyebaran kuisisioner, penyebaran kuisisioner dilakukan untuk mendapatkan data kontekstual. Informasi kontekstual yang diperlukan yakni tanggal lahir, jenis kelamin, berat badan, tinggi badan.

3.2.2. Analisis Data

Tahap analisis data dimulai dengan cara menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Data yang dibutuhkan yaitu data kontekstual pengguna (umur, berat badan, tinggi badan, aktivitas) dan data makanan. Langkah selanjutnya adalah menghitung nilai performa dari setiap kriteria alternatif menu makanan untuk di proses menggunakan metode *Weighted Sum Model*. Data yang dihasilkan berupa nilai WSM menu makanan tertinggi, dimana nilai 10 menu makanan tertinggi akan terpilih sebagai rekomendasi menu makanan bagi pengguna sistem informasi. sehingga informasi yang dihasilkan lebih baik. Selanjutnya sistem penunjang keputusan akan mengeluarkan hasil dari rekomendasi menu makanan.

3.2.3. Desain Sistem

Pada penelitian ini, pembuatan desain sistem pada penelitian ini menggunakan *Unified Modeling Language* (UML) yang dirancang dengan konsep

Object Oriented programming (OOP). Pemodelan UML yang digunakan adalah sebagai berikut:

1. *Business Process*

Business Process adalah Proses Bisnis menyediakan metode universal untuk menggambarkan aspek operasional bisnis.

2. *Use Case Diagram*

Use Case Diagram digunakan untuk menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem.

3. *Scenario Diagram*

Scenario Diagram menjelaskan alur sistem dari fitur yang ada di job specification dan job description yang ada pada diagram use case.

4. *Activity Diagram*

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing aktivitas berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas berakhir.

5. *Sequence Diagram*

Sequence Diagram merupakan rangkaian pesan yang dikirim antar objek dan interaksi antar objek.

6. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi class serta hubungan antar class, sehingga memudahkan dalam proses pengkodean.

7. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram menjelaskan hubungan antar data dalam basis data berdasarkan data yang mempunyai hubungan antar relasi.

3.2.4. Implementasi

Pada tahap ini desain yang telah dibuatkan akan diimplementasikan dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

- a. Implementasi kode program (*coding*) menggunakan bahasa php (*Page Hypertext Preprocessor*) dan menggunakan sublime sebagai IDE-nya.
- b. Manajemen basis data menggunakan *Database Management System (DBMS)* MySQL.

- c. Menggunakan aplikasi Xampp untuk menjalankan program *localhost*.

3.2.5. Pengujian Pengembangan Sistem

Pada tahap ini dilakukan uji coba terhadap sistem yang telah dibangun. Kegiatan pengujian yang dilakukan bertujuan untuk menguji spesifikasi sistem yang telah dibangun menggunakan pengujian *black box testing*. *Black Box Testing* merupakan cara pengujian perangkat lunak yang memeriksa kesesuaian fitur dari sebuah sistem dengan kebutuhan fungsionalnya (Rouf, 2012). *Black Box* dapat menemukan kesalahan dalam beberapa kategori, yaitu fungsi fungsi yang salah atau hilang, kesalahan interface, dan kesalahan struktur data

Dalam melakukan *black box testing* penulis akan menggunakan tipe *requirement testing*. *Requirements testing* adalah pengujian aplikasi yang melakukan pengujian dengan cara melakukan penyocokan interface dan penerapannya terhadap kebutuhan fungsionalitas sistem. Tabel *requirements testing* ditunjukkan pada tabel 3.1.

Tabel 3. 1 *Requirements Testing*

Keterangan <i>Usecase</i>	Flow	Hasil yang Diharapkan	Hasil Sebenarnya	Kesimpulan	
				Berhasil	Tidak
Nama fitur Deskripsi	Main Flow : Alternatif Flow 1 :				

3.2.6. Pemeliharaan

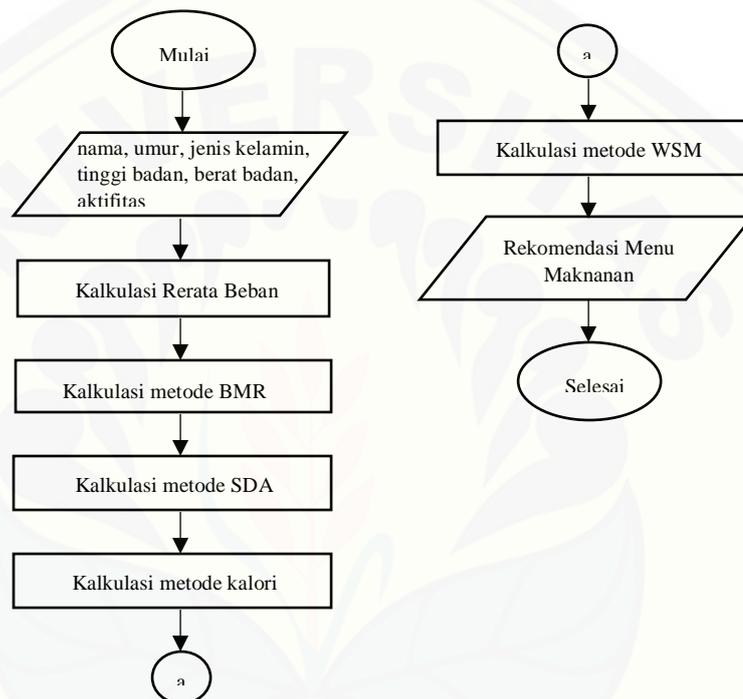
Pemeliharaan dilakukan untuk mengatasi masalah ketika sistem sudah digunakan oleh pengguna/*user*. Ketika *user* menemukan bug pada sistem, maka *user* akan konfirmasi ke *developer* untuk mendapat penanganan dari pengembang sistem.

3.3 Analisis Tahap Penerapan Metode

Penelitian ini menghasilkan rekomendasi menu makanan yang berdasarkan tingkat kebutuhan kalori dari pengguna dengan menerapkan metode *Weighted Sum Model* yang mengolah data. Proses pertama data kontekstual akan menjadi inputan bagi proses perhitungan beban kerja kemudian dilanjutkan dengan menghitung

kalori dengan menggunakan metode BMR, SDA, dan Aktivitas. Output dari dari tahapan analisa data ini berupa kalori yang dibutuhkan yang sesuai untuk pengguna.

Proses selanjutnya adalah melakukan analisis data dengan menggunakan metode *Weighted Sum Model*. Tahapan ini dilakukan berdasarkan kriteria kalori, protein dan karbohidrat sehingga menghasilkan rekomendasi menu makanan yang sesuai dengan kebutuhan perseorangan tersebut. Penerapan metode *Weighted Sum Model* pada rekomendasi menu makanan dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Analisis Tahap Penerapan Metode

BAB 4 PENGEMBANGAN SISTEM

Bab ini akan membahas tentang Pengembangan Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori Tubuh Menggunakan Metode *Weighted Sum Model*. Tahap pengembangan sistem dilaksanakan berdasarkan mode waterfall, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, dan pengujian sistem.

4.1 Deskripsi Umum Sistem

Sistem yang akan dibangun pada penelitian ini adalah Sistem Penunjang Keputusan Rekomendasi Menu Makanan berdasarkan Tingkat Kebutuhan Kalori Tubuh Menggunakan *Weighted Sum Model* (WSM). Sistem ini mampu merekomendasikan menu makanan yang harus dikonsumsi. Data yang dibutuhkan dalam sistem ini yaitu data kontekstual dari pengguna (berat badan, umur dan tinggi), data aktivitas, dan data makanan. Sistem ini memiliki dua hak akses, yaitu admin dan pengguna. Setiap masing-masing hak akses memiliki peran yang berbeda dalam menggunakan sistem. Sistem ini dapat melakukan rekomendasi menu makanan dengan kebutuhan pengguna. Pengguna memasukkan data kontekstual dan admin memasukkan data makanan ke sistem penunjang keputusan yang selanjutnya sistem ini akan melakukan rekomendasi. Penerapan *Weighted Sum Model* WSM digunakan untuk mendukung keputusan supaya berhasil dalam merekomendasikan menu makanan. Selain melakukan keputusan rekomendasi menu makanan, sistem ini juga dapat melakukan pencatatan hasil rekomendasi untuk pengguna.

4.2 Analisis Kebutuhan Data dan Sistem

Proses analisis kebutuhan data dilakukan dengan melakukan observasi pada web www.organisasi.org untuk mendapatkan data kalori, protein dan karbohidrat makanan dan melakukan penyebaran kuisioner kepada semua kalangan masyarakat untuk mendapatkan data kontekstual pengguna. Data kontekstual pengguna dan

data makanan yang didapatkan melalui proses pencarian data terdapat pada Tabel 4.1 dan Tabel 4.2.

Tabel 4. 1 Data Kontekstual

No	Data	Keterangan
1	Nama Pengguna	Data Atribut
2	Tanggal Lahir	Parameter
3	Berat Badan	Parameter
4	Tinggi Badan	Parameter
5	Aktivitas Yang Akan Dilakukan	Parameter

Tabel 4. 2 Data Makanan

No	Data	Keterangan
1	Nama Makanan	Data Atribut
2	Jenis Makanan	Parameter
3	Kalori Makanan	Parameter
4	Harga Makanan	Parameter

Data makanan dan data kontekstual di atas selanjutnya digunakan dalam proses analisis kebutuhan sistem. Tahap analisis kebutuhan sistem merupakan tahapan yang penting dalam pengembangan sebuah sistem informasi. Seluruh kebutuhan didefinisikan pada tahap kebutuhan fungsional dan kebutuhan non-fungsional.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses yang akan dilakukan oleh sistem. Kebutuhan fungsional sistem penunjang keputusan rekomendasi menu makanan antara lain:

- a. Sistem mampu menambah data pengguna baru
- b. Sistem mampu mengelola data pengguna (*view, insert, edit, delete*)
- c. Sistem mampu mengelola data makanan (*view, insert, edit, delete*)
- d. Sistem mampu mengelola data kombinasi makanan (*view, insert, edit, delete*)
- e. Sistem mampu mengelola data aktivitas (*view, insert, edit, delete*)
- f. Sistem mampu mengelola data posisi badan saat melakukan aktivitas (*view, insert, edit, delete*)

- g. Sistem mampu melihat hasil rekomendasi menu makanan
- h. Sistem mampu menampilkan laporan hasil merekomendasikan menu makanan.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional berisi kebutuhan yang menjadi pendukung pada sistem penunjang keputusan rekomendasi menu makanan. Kebutuhan non-fungsional sistem penunjang keputusan rekomendasi menu makanan antara lain :

- a. Sistem penunjang keputusan menggunakan autentikasi yang menggunakan *username* dan *password*.
- b. *User interface* yang *user-friendly* sehingga memudahkan pengguna dalam menggunakan sistem penunjang keputusan.
- c. Sistem penunjang keputusan berbasis web

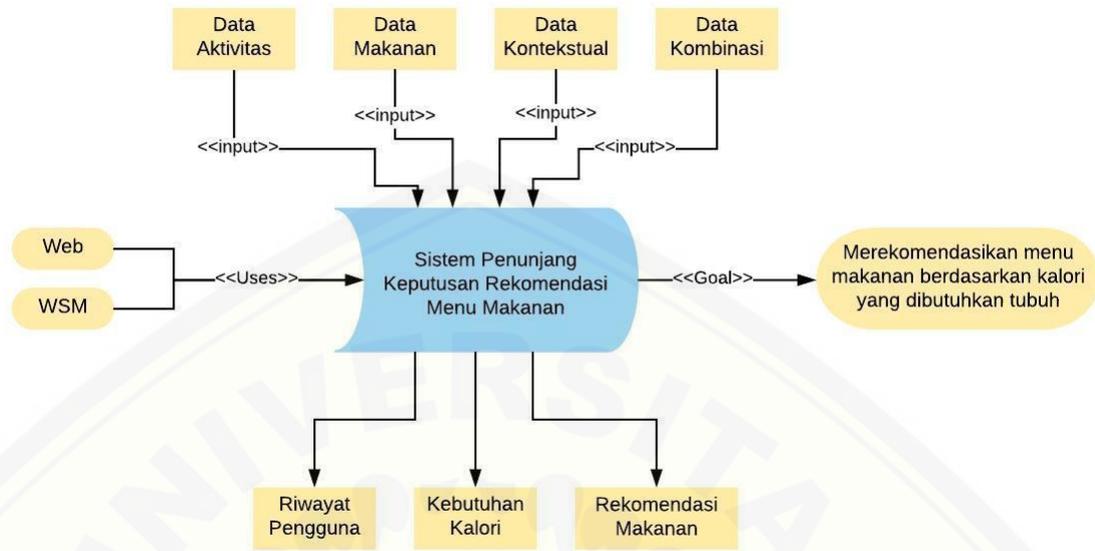
4.3 Desain Sistem

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

4.3.1 Business Process

Business process merupakan model yang menggambarkan proses yang terjadi dalam sistem informasi mulai dari *input*, *output*, *uses*, dan *goal* dari sistem informasi yang dibuat. Sistem penunjang keputusan rekomendasi menu makanan ini menggunakan platform *website* dan metode *Weighted Sum Model* dalam melakukan keputusan rekomendasi menu makanan. Sistem penunjang keputusan ini menggunakan data kontekstual, data makanan, data kalori makanan, dan data aktivitas sebagai *input* ke sistem penunjang keputusan. *Input* tersebut nantinya akan menghasilkan *output* berupa laporan hasil rekomendasi, kebutuhan kalori, dan hasil rekomendasi, sedangkan *goal* dari sistem penunjang keputusan ini adalah merekomendasikan menu makanan berdasarkan kalori yang dibutuhkan oleh tubuh.

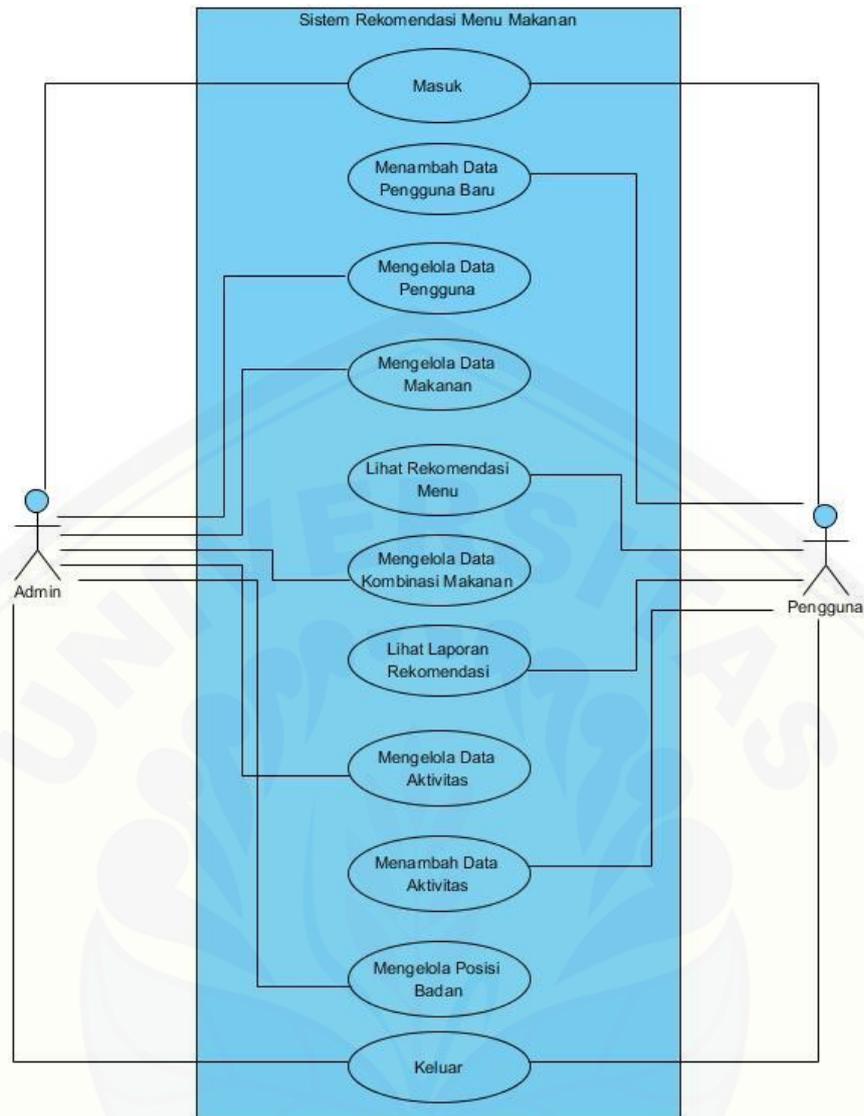
Business process dari sistem penunjang keputusan rekomendasi menu makanan ini dapat dilihat pada Gambar 4.1.



Gambar 4. 1 *Business Process*

4.3.2 *Usecase Diagram*

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



Gambar 4. 2 Usecase Diagram

Penjelasan tentang definisi aktor dan definisi *usecase diagram* pada gambar akan dijelaskan di bawah ini.

1. Definisi aktor

Definisi aktor yaitu penjelasan mengenai aktor-aktor sebagai pengguna dari sistem penunjang keputusan rekomendasi menu makanan. Terdapat 2 Aktor seperti yang dijelaskan pada Tabel 4.3.

Tabel 4. 3 Definsi Aktor

No.	Aktor	Deskripsi
1.	Pengguna	Aktor Pengguna pada sistem ini memiliki hak akses untuk: <ol style="list-style-type: none"> 1. Menambah Data Pengguna Baru 2. Melihat Hasil Rekomendasi Menu Makanan 3. Melihat Laporan Rekomendasi Menu Makanan 4. Menambah Data Aktivitas
2.	Admin	Aktor Admin pada sistem ini memiliki hak akses untuk: <ol style="list-style-type: none"> 1. Mengelola Data Pengguna 2. Mengelola Data Kombinasi Menu Makanan 3. Mengelola Data Makanan 4. Mengelola Data Aktivitas 5. Mengelola Data Posisi Badan

2. Definisi *Usecase*

Definisi *usecase* yaitu penjelasan tentang fitur-fitur yang tersedia dalam sistem penunjang keputusan rekomendasi menu makanan. Terdapat 11 *usecase* pada Tabel 4.4.

Tabel 4. 4 *Usecase*

No.	<i>Usecase</i>	Deskripsi
1	Masuk	Menggambarkan proses autentifikasi pengguna yang digunakan untuk masuk ke sistem penunjang keputusan.
2	Menambah Data Pengguna Baru	Menggambarkan proses pendaftaran data pengguna baru.
3	Mengelola Data Pengguna	Menggambarkan proses mengelola data pengguna yang meliputi untuk melihat, menambah, mengubah, dan menghapus data pengguna.

4	Mengelola Data Makanan	Menggambarkan proses mengelola data makanan yang meliputi untuk melihat, menambah, mengubah, dan menghapus data makanan.
5	Melihat Hasil Rekomendasi Menu Makanan	Menggambarkan proses melihat hasil rekomendasi menu makanan.
6	Mengelola Data Kombinasi Menu Makanan	Menggambarkan proses mengelola data kombinasi menu makanan yang meliputi untuk melihat, menambah, mengubah, dan menghapus data kombinasi menu makanan.
7	Melihat Laporan Rekomendasi Menu Makanan	Menggambarkan melihat laporan hasil rekomendasi menu makanan.
8	Mengelola Data Aktivitas	Menggambarkan proses mengelola data aktivitas yang meliputi untuk melihat, menambah, mengubah data aktivitas.
9	Menambah Data Aktivitas	Menggambarkan proses untuk menambah aktivitas apabila aktivitas yang dilakukan tidak ada dalam daftar aktivitas yang tersedia.
10	Mengelola Data Posisi Badan	Menggambarkan proses untuk menambah dan mengubah data posisi badan.
11	Keluar	Menggambarkan proses <i>user</i> untuk keluar dari sistem.

4.3.3 Skenario

Skenario merupakan penjabaran alur kerja sistem informasi yang terdapat pada *usecase diagram*. Skenario sistem penunjang keputusan rekomendasi menu makanan adalah sebagai berikut.

4.3.3.1 Skenario Masuk

Skenario masuk merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor akan masuk ke dalam sistem. Hanya *user* terdaftar saja yang dapat mengakses sistem tersebut. Terdapat dua *user* dalam sistem penunjang keputusan ini yaitu Admin dan Pengguna. Sistem ini memiliki reaksi yang berbeda untuk masing-masing aktor sehingga aktor dapat masuk ke sistem dan menggunakannya. Skenario masuk dapat dilihat pada Lampiran A Tabel 1.

4.3.3.2 Skenario Menambah Data Pengguna Baru

Skenario menambah data pengguna baru merupakan alur yang menjelaskan aksi dan reaksi sistem pada saat aktor menambah data pengguna baru. Aktor yang dapat menjalankan skenario ini adalah pengguna. Proses yang dilakukan oleh aktor agar dapat menambah data pengguna baru adalah pertama aktor memilih menu *sign up* selanjutnya sistem akan menampilkan halaman pendaftaran pengguna baru dengan menampilkan form data pengguna baru, kemudian aktor mengisi form data pengguna baru lalu klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form data pengguna baru jika sudah benar maka data akan tersimpan pada database. Skenario menambah data pengguna dapat dilihat pada Lampiran A Tabel 2.

4.3.3.3 Skenario Mengelola Data Pengguna

Skenario mengelola data pengguna merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data pengguna. Aktor yang dapat menjalankan skenario ini adalah admin. Aktor dapat melihat, menambah, mengubah, dan menghapus data pengguna. Skenario pertama melihat data pengguna, maka yang harus dilakukan oleh aktor adalah klik menu pengguna selanjutnya sistem akan menampilkan daftar pengguna dalam tabel. Skenario kedua menambah data pengguna, pertama klik menu pengguna pada halaman utama, selanjutnya sistem akan menampilkan daftar tabel yang berisi data pengguna, lalu klik tombol tambah pengguna, kemudian sistem akan menampilkan halaman form tambah pengguna, lalu aktor mengisi form data pengguna selanjutnya klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form data

pengguna jika sudah benar maka data akan tersimpan pada database. Skenario ketiga mengubah data pengguna, klik menu pengguna maka sistem akan menampilkan halaman daftar tabel pengguna, kemudian klik tombol edit pada salah satu data pengguna yang akan diubah maka sistem akan menampilkan form dengan data pengguna yang dipilih, selanjutnya aktor mengubah data dan klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form edit data pengguna jika sudah benar maka data akan tersimpan pada database. Skenario mengelola data pengguna dapat dilihat pada Lampiran A Tabel 3.

4.3.3.4 Skenario Mengelola Data Makanan

Skenario mengelola data makanan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data makanan. Aktor yang dapat menjalankan skenario ini adalah admin. Aktor dapat melihat, menambah, mengubah, dan menghapus data makanan. Skenario pertama melihat data makanan, maka yang harus dilakukan oleh aktor adalah klik menu makanan selanjutnya sistem akan menampilkan daftar makanan dalam tabel. Skenario kedua menambah data makanan, pertama klik menu makanan pada halaman utama, selanjutnya sistem akan menampilkan daftar tabel yang berisi data makanan, lalu klik tombol tambah makanan, kemudian sistem akan menampilkan halaman form tambah makanan, lalu aktor mengisi form data makanan selanjutnya klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form data makanan jika sudah benar maka data akan tersimpan pada database. Skenario ketiga mengubah data makanan, klik menu makanan maka sistem akan menampilkan halaman daftar tabel makanan, kemudian klik tombol edit pada salah satu data makanan yang akan diubah maka sistem akan menampilkan form dengan data makanan yang dipilih, selanjutnya aktor mengubah data dan klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form edit data makanan jika sudah benar maka data akan tersimpan pada database. Skenario mengelola data makanan dapat dilihat pada Lampiran A Tabel 4.

4.3.3.5 Skenario Melihat Hasil Rekomendasi Menu Makanan

Skenario melihat hasil rekomendasi menu makanan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor akan melihat hasil rekomendasi. Aktor yang dapat menjalankan skenario ini adalah pengguna. Skenario melihat hasil rekomendasi adalah pertama klik menu rekomendasi maka sistem akan menampilkan form data kontekstual, selanjutnya aktor mengisi form lalu klik tombol analisa, kemudian sistem akan menampilkan hasil dari perhitungan kalori, protein, karbohidat dan sepuluh rekomendasi menu makanan dalam tabel. Skenario melihat hasil rekomendasi menu makanan dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Skenario Melihat Hasil Rekomendasi Menu Makanan

Nomor Usecase	USC-05
Nama	Melihat Hasil Rekomendasi Menu Makanan
Aktor	Pengguna
Deskripsi Singkat	Aktor melihat data hasil rekomendasi menu makanan
Prekondisi	Aktor memilih menu rekomendasi
Prakondisi	Aktor berhasil melihat data rekomendasi menu makanan
Flow of Events	
Skenario Normal : Melihat Hasil Rekomendasi Menu Makanan	
1. Klik menu rekomendasi	
	2. Menampilkan halaman rekomendasi menu makanan yang meliputi : <ul style="list-style-type: none"> • <i>Form</i> Tambah data Personal berupa: <ul style="list-style-type: none"> - Berat badan - Tinggi badan - Aktivitas yang akan dikerjakan - Tombol tambah aktivitas - Tombol analisa untuk perhitungan
3. Mengisi <i>form</i> tambah data personal, kemudian klik tombol Analisa	
	4. Memproses perhitungan peramalan dengan metode <i>Weighted Sum Model</i>

	kemudian menyimpan ke dalam database
	5. Menampilkan hasil peramalan.
Skenario Alternatif : Isian form kosong	
4. Klik tombol analisa	
	5. Menampilkan span “ <i>Please fill out this field</i> ”

4.3.3.6 Skenario Mengelola Data Kombinasi Menu Makanan

Skenario mengelola data kombinasi makanan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data kombinasi makanan. Aktor yang dapat menjalankan skenario ini adalah admin. Aktor dapat melihat, menambah, mengubah, dan menghapus data kombinasi makanan. Skenario pertama melihat data kombinasi makanan, maka yang harus dilakukan oleh aktor adalah klik menu kombinasi makanan selanjutnya sistem akan menampilkan daftar kombinasi makanan dalam tabel. Skenario kedua menambah data kombinasi makanan, pertama klik menu kombinasi makanan pada halaman utama, selanjutnya sistem akan menampilkan daftar tabel yang berisi data kombinasi makanan, lalu klik tombol tambah kombinasi makanan, kemudian sistem akan menampilkan halaman form tambah kombinasi makanan, lalu aktor mengisi form data kombinasi makanan selanjutnya klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form data kombinasi makanan jika sudah benar maka data akan tersimpan pada database. Skenario ketiga mengubah data kombinasi makanan, klik menu kombinasi makanan maka sistem akan menampilkan halaman daftar tabel kombinasi makanan, kemudian klik tombol edit pada salah satu data kombinasi makanan yang akan diubah maka sistem akan menampilkan form dengan data kombinasi makanan yang dipilih, selanjutnya aktor mengubah data dan klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form edit data kombinasi makanan jika sudah benar maka data akan tersimpan pada database. Skenario mengelola data kombinasi makanan dapat dilihat pada Lampiran A Tabel 5.

4.3.3.7 Skenario Melihat Laporan Hasil Rekomendasi Menu Makanan

Skenario melihat laporan hasil rekomendasi menu makanan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor akan melihat laporan hasil rekomendasi menu makanan. Aktor yang dapat menjalankan skenario ini adalah pengguna. Skenario melihat laporan hasil rekomendasi menu makanan adalah klik menu riwayat maka sistem akan menampilkan laporan hasil rekomendasi yang telah dilakukan oleh aktor. Skenario melihat laporan hasil rekomendasi menu makanan dapat dilihat pada lampiran A Tabel 6.

4.3.3.8 Skenario Mengelola Data Aktivitas

Skenario mengelola data aktivitas merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data aktivitas. Aktor yang dapat menjalankan skenario ini adalah admin. Aktor dapat melihat, menambah, mengubah, dan menghapus data aktivitas. Skenario pertama melihat data aktivitas, maka yang harus dilakukan oleh aktor adalah klik menu aktivitas selanjutnya sistem akan menampilkan daftar aktivitas dalam tabel. Skenario kedua menambah data aktivitas, pertama klik menu aktivitas pada halaman utama, selanjutnya sistem akan menampilkan daftar tabel yang berisi data aktivitas, lalu klik tombol tambah aktivitas, kemudian sistem akan menampilkan halaman form tambah aktivitas, lalu aktor mengisi form data aktivitas selanjutnya klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form data aktivitas jika sudah benar maka data akan tersimpan pada database. Skenario ketiga mengubah data aktivitas, klik menu aktivitas maka sistem akan menampilkan halaman daftar tabel aktivitas, kemudian klik tombol edit pada salah satu data aktivitas yang akan diubah maka sistem akan menampilkan form dengan data aktivitas yang dipilih, selanjutnya aktor mengubah data dan klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form edit data aktivitas jika sudah benar maka data akan tersimpan pada database. Skenario mengelola data aktivitas dapat dilihat pada Lampiran A Tabel 7.

4.3.3.9 Skenario Menambah Data Aktivitas

Skenario menambah data aktivitas merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor menambah data aktivitas apabila aktivitas yang diinginkan tidak tersedia pada sistem. Aktor yang dapat menjalankan skenario ini adalah admin. Skenario aktor menambah data aktivitas adalah pertama klik menu rekomendasi maka sistem akan menampilkan halaman form data kontekstual, lalu klik tombol tambah aktivitas, selanjutnya sistem menampilkan halaman form tambah aktivitas, lalu aktor mengisi form dilanjutkan dengan klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form tambah data aktivitas jika sudah benar maka data akan tersimpan pada database. Skenario menambah data aktivitas dapat dilihat pada lampiran A Tabel 8.

4.3.3.10 Skenario Mengelola Data Posisi Badan

Skenario mengelola data posisi badan merupakan alur yang menjelaskan aksi aktor dan reaksi sistem pada saat aktor mengelola data posisi badan. Aktor yang dapat menjalankan skenario ini adalah admin. Aktor dapat melihat, menambah, mengubah, dan menghapus data posisi badan. Skenario pertama melihat data posisi badan, maka yang harus dilakukan oleh aktor adalah klik menu posisi badan selanjutnya sistem akan menampilkan daftar posisi badan dalam tabel. Skenario kedua menambah data posisi badan, pertama klik menu posisi badan pada halaman utama, selanjutnya sistem akan menampilkan daftar tabel yang berisi data aktivitas, lalu klik tombol tambah posisi badan, kemudian sistem akan menampilkan halaman form tambah posisi badan, lalu aktor mengisi form data posisi badan selanjutnya klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan menampilkan pesan dan mengembalikan pada halaman form data posisi badan jika sudah benar maka data akan tersimpan pada database. Skenario ketiga mengubah data posisi badan, klik menu posisi badan maka sistem akan menampilkan halaman daftar tabel posisi badan, kemudian klik tombol edit pada salah satu data posisi badan yang akan diubah maka sistem akan menampilkan form dengan data posisi badan yang dipilih, selanjutnya aktor mengubah data dan klik tombol simpan, kemudian sistem akan mengecek form jika terdapat isian kosong pada form maka sistem akan

menampilkan pesan dan mengembalikan pada halaman form edit data posisi badan jika sudah benar maka data akan tersimpan pada database. Skenario mengelola data posisi badan dapat dilihat pada Lampiran A Tabel 9.

4.3.3.11 Skenario Keluar

Skenario keluar merupakan alur dari aksi aktor dan reaksi sistem jika akan keluar sistem. Aktor dapat menjalankan skenario keluar dengan cara klik tombol *logout* pada halaman aktif dari aktor. Penjelasan urutan aksi aktor dan reaksi sistem pada skenario utama dan skenario alternatif use case keluar terdapat pada lampiran A Tabel 10

4.3.4 Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan aliran informasi yang terjadi antar objek. *Sequence* diagram merupakan desain perancangan yang mendekati proses pengkodean, maka dari itu di dalam *sequence* diagram bahasa yang dituliskan merupakan bahasa pengkodean, dimana bahasa pengkodean tersebut menjelaskan jalannya alur dalam fitur secara berurutan. *Sequence* diagram dalam sistem informasi ini antara lain sebagai berikut.

4.3.4.1 Sequence Diagram Masuk

Pada *sequence* diagram masuk terdapat beberapa interaksi antar kelas yang digunakan. Kelas yang saling terhubung adalah kelas tampilan antara dua pengguna yaitu Pengguna dan Admin. Proses dari aliran *sequence* diagram diawali ketika aktor melakukan perintah klik tombol login pada *v_login*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* *login()* yang terdapat pada *controller* *c_login* dimana *controller* akan melanjutkan memanggil *method* pada model *M_user* untuk melakukan pengecekan data user, kemudian kelas model mengembalikan data user tersebut. Apabila data yang dikembalikan dari model *M_user* menuju ke *controller* *c_login* terdapat kesalahan maka akan dikembalikan ke *view* *v_login* untuk mengulang proses *login*, apabila menurut model *M_user* sudah benar maka akan diteruskan ke *view* sesuai dengan aktor yang masuk ke sistem. Penggambaran *sequence* diagram masuk digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang dapat dilihat pada Lampiran B Gambar 1.

4.3.4.2 Sequence Diagram Menambah Pengguna Baru

Pada *sequence* diagram menambah pengguna baru terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram menambah pengguna baru diawali ketika pengguna melakukan perintah klik tombol *sign up* pada *view* *v_login*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* *signUp()* yang terdapat pada *controller* *c_home* yang mengirim perintah untuk menuju ke *view* *v_signUp*. Pengguna melakukan pengisian data kemudian menekan tombol *sign up* di *view* *v_signUp* dilanjutkan dengan mengirim data pada kelas *controller* *c_login* diteruskan menuju kelas *M_user* dengan *method* *simpan_data* melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat, apabila isian benar maka dilanjutkan menuju *view* *v_login*. Penggambaran *sequence* diagram menambah pengguna baru digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang dapat dilihat pada lampiran B Gambar 2.

4.3.4.3 Sequence Diagram Mengelola Pengguna

Pada *sequence* diagram mengelola pengguna terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram mengelola pengguna diawali ketika admin melakukan perintah klik menu pengguna pada kelas *view* *v_admin_menu*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* *pengguna()* yang terdapat pada kelas *controller* *c_home*, kemudian mengambil data pada kelas *M_user* dengan *method* *data_user()*, hasilnya akan menampilkan semua data pengguna pada tabel. Selanjutnya proses menambah pengguna, pertama aktor melakukan perintah dengan menekan tombol tambah pengguna pada kelas *view* *v_admin_user*, kemudian *view* akan mengirim perintah dengan memanggil *method* *tambah_user()* pada kelas *controller* *c_user* yang dapat menampilkan halaman tambah user di kelas *view* *v_admin_inputUser*. Selanjutnya aktor akan mengisi formulir tambah pengguna, jika sudah maka akan menekan tombol simpan yang mengirim perintah dengan memanggil *method* *simpan_data* pada kelas *controller* *c_user*, dilanjutkan dengan mengirim perintah menyimpan data pada kelas *M_user* dengan *method* *simpan_data*. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah

pengguna pada kelas *view* *v_admin_inputUser*, apabila isian benar maka dilanjutkan menuju *view* *v_admin_user*. Langkah selanjutnya apabila terdapat kesalahan pada data pengguna dan admin ingin mengubah data pengguna tersebut, maka admin akan mengirim pesan dengan menekan tombol *update* pada kelas *view* *v_admin_user* kemudian perintah tersebut akan memanggil *method* *update()* pada kelas *controller* *c_user*, selanjutnya dari kelas *controller* memanggil *method* *get_user* pada kelas model *M_user* yang akan menampilkan formulir data pengguna yang dipilih pada kelas *view* *v_admin_editUser*. Setelah melakukan perubahan, maka aktor akan mengirim pesan dengan menekan tombol simpan pada kelas *view* *v_admin_editPengguna* yang kemudian memanggil *method* *ubah_data* pada kelas *controller* *c_user*, dari kelas *controller* *c_user* dilanjutkan dengan penyimpanan pada kelas model *M_user*. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah pengguna pada kelas *view* *v_admin_editUser*, apabila isian benar maka dilanjutkan menuju *view* *v_admin_user*. Penggambaran *sequence* diagram mengelola pengguna digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang dapat dilihat pada lampiran B Gambar 3.

4.3.4.4 *Sequence* Diagram Mengelola Makanan

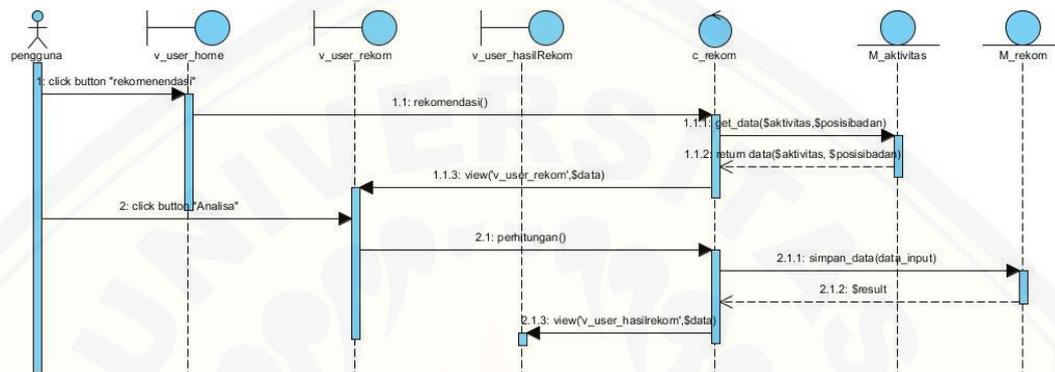
Pada *sequence* diagram mengelola makanan terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram mengelola pengguna diawali ketika admin melakukan perintah klik menu makanan pada kelas *view* *v_admin_menu*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* *menu_admin()* yang terdapat pada kelas *controller* *c_home*, kemudian mengambil data makanan pada kelas *M_user* dengan *method* *data_makanan()*, hasilnya akan menampilkan semua data makanan pada tabel. Selanjutnya proses menambah makanan, pertama aktor melakukan perintah dengan menekan tombol tambah makanan pada kelas *view* *v_admin_menu*, kemudian *view* akan mengirim perintah dengan memanggil *method* *input_menu()* pada kelas *controller* *c_menu* yang dapat menampilkan halaman tambah makanan di kelas *view* *v_admin_inputMenu*. Selanjutnya aktor akan mengisi formulir tambah makanan, jika sudah maka akan menekan tombol simpan yang mengirim perintah dengan memanggil *method*

simpan_data pada kelas *controller* c_menu, dilanjutkan dengan mengirim perintah menyimpan data pada kelas M_menu dengan *method* *simpan_data*. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah pengguna pada kelas *view* v_admin_inputMenu, apabila isian benar maka dilanjutkan menuju *view* v_admin_menu. Langkah selanjutnya apabila terdapat kesalahan pada data makanan dan aktor ingin mengubah data makanan tersebut, maka aktor akan mengirim pesan dengan menekan tombol *update* pada kelas *view* v_admin_menu kemudian perintah tersebut akan memanggil *method* *update()* pada kelas *controller* c_menu, selanjutnya dari kelas *controller* memanggil *method* *get_makanan* pada kelas model M_menu yang akan menampilkan formulir data makanan yang dipilih pada kelas *view* v_admin_editMenu. Setelah melakukan perubahan, maka aktor akan mengirim pesan dengan menekan tombol *simpan* pada kelas *view* v_admin_editMenu yang kemudian memanggil *method* *ubah_data* pada kelas *controller* c_menu, dari kelas *controller* c_menu dilanjutkan dengan penyimpanan pada kelas model M_menu. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah makanan pada kelas *view* v_admin_editMenu, apabila isian benar maka dilanjutkan menuju *view* v_admin_menu. Penggambaran *sequence* diagram mengelola pengguna digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang dapat dilihat pada lampiran B Gambar 4.

4.3.4.5 *Sequence* Diagram Melihat Hasil Rekomendasi Menu Makanan

Pada *sequence* diagram melihat hasil rekomendasi menu makanan terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram melihat hasil rekomendasi menu makanan diawali ketika aktor melakukan perintah klik menu rekomendasi pada kelas *view* v_user_home, selanjutnya *view* akan mengirim perintah dengan memanggil *method* *rekomendasi()* yang terdapat pada kelas *controller* c_rekom, kemudian mengambil data yang dibutuhkan pada kelas M_aktivitas dengan *method* *get_data()*, setelah data terambil maka diteruskan dengan menampilkan halaman formulir rekomendasi. Selanjutnya aktor melakukan pengisian formulir, jika sudah diisi maka aktor akan mengirim pesan dengan

menekan tombol analisa pada kelas *view* *v_user_rekom*, kemudian memanggil *method* *perhitungan()* pada kelas *controller* *c_rekom*. Setelah melakukan perhitungan, maka hasil dari perhitungan akan disimpan pada model *M_rekom*, setelah tersimpan hasil dari perhitungan dikirim pada kelas *controller* *c_rekom* untuk ditampilkan pada kelas *view* *v_user_hasilRekom*. Penggambaran *sequence* diagram melihat hasil rekomendasi menu makanan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang dapat dilihat pada Gambar 6.



Gambar 4.3 *Sequence* Diagram Melihat Hasil Rekomendasi Menu Makanan

4.3.4.6 *Sequence* Diagram Mengelola Data Kombinasi Menu Makanan

Pada *sequence* diagram mengelola data kombinasi menu makanan terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram mengelola data kombinasi menu makanan diawali ketika aktor melakukan perintah klik menu kombinasi makanan pada kelas *view* *v_admin_menuKombinasi*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* *menu_adminKombinasi()* yang terdapat pada kelas *controller* *c_home*, kemudian mengambil data makanan pada kelas *M_user* dengan *method* *data_kombinasi()*, hasilnya akan menampilkan semua data kombinasi makanan pada tabel. Selanjutnya proses menambah kombinasi makanan, pertama aktor melakukan perintah dengan menekan tombol tambah kombinasi makanan pada kelas *view* *v_admin_menuKombinasi*, kemudian *view* akan mengirim perintah dengan memanggil *method* *input_menuKombinasi()* pada kelas *controller* *c_kombinasi* yang dapat menampilkan halaman tambah kombinasi makanan di kelas *view* *v_admin_inputMenuKombinasi*. Selanjutnya aktor akan mengisi formulir tambah kombinasi makanan, jika sudah maka akan menekan tombol simpan yang mengirim perintah dengan memanggil *method* *simpan_data*

pada kelas *controller* *c_kombinasi*, dilanjutkan dengan mengirim perintah menyimpan data pada kelas *M_kombinasi* dengan *method* *simpan_data*. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah pengguna pada kelas *view* *v_admin_inputMenuKombinasi*, apabila isian benar maka dilanjutkan menuju *view* *v_admin_menuKombinasi*. Langkah selanjutnya apabila terdapat kesalahan pada data makanan dan aktor ingin mengubah data kombinasi makanan tersebut, maka aktor akan mengirim pesan dengan menekan tombol *update* pada kelas *view* *v_admin_menuKombinasi* kemudian perintah tersebut akan memanggil *method* *update()* pada kelas *controller* *c_kombinasi*, selanjutnya dari kelas *controller* memanggil *method* *get_kombinasi* pada kelas model *M_kombinasi* yang akan menampilkan formulir data kombinasi makanan yang dipilih pada kelas *view* *v_admin_editKombinasi*. Setelah melakukan perubahan, maka aktor akan mengirim pesan dengan menekan tombol simpan pada kelas *view* *v_admin_editKombinasi* yang kemudian memanggil *method* *ubah_data* pada kelas *controller* *c_kombinasi*, dari kelas *controller* *c_kombinasi* dilanjutkan dengan penyimpanan pada kelas model *M_kombinasi*. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah makanan pada kelas *view* *v_admin_editKombinasi*, apabila isian benar maka dilanjutkan menuju *view* *v_admin_menuKombinasi*. Penggambaran *sequence* diagram mengelola data kombinasi makanan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang dapat dilihat pada lampiran B Gambar 5.

4.3.4.7 *Sequence* Diagram Melihat Laporan Rekomendasi Menu Makanan

Pada *sequence* diagram melihat laporan rekomendasi menu makanan terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram melihat laporan rekomendasi menu makanan diawali ketika aktor melakukan perintah klik menu riwayat pada kelas *view* *v_user_home*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* *rekap()* yang terdapat pada kelas *controller* *c_rekap*, kemudian mengambil data rekapitulasi rekomendasi menu makanan pada kelas *M_rekap* dengan *method* *data_rekom()*, hasilnya akan menampilkan semua

data rekapitulasi rekomendasi menu makanan pada tabel yang terdapat pada kelas *view* *v_user_rekap*. Penggambaran *sequence* diagram melihat laporan rekomendasi menu makanan digunakan untuk menjelaskan fungsi yang akan dibuat seperti yang ditunjukkan pada lampiran B Gambar 6.

4.3.4.8 *Sequence* Diagram Mengelola Data Aktivitas

Pada *sequence* diagram mengelola data aktivitas terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram mengelola data aktivitas diawali ketika aktor melakukan perintah klik menu aktivitas pada kelas *view* *v_admin_menu*, selanjutnya *view* akan mengirim perintah dengan memanggil *method* *aktivitas()* yang terdapat pada kelas *controller* *c_aktivitas*, kemudian mengambil data aktivitas pada kelas *M_aktivitas* dengan *method* *data_aktivitas()*, hasilnya akan menampilkan semua data aktivitas pada tabel. Selanjutnya proses menambah aktivitas, pertama aktor melakukan perintah dengan menekan tombol tambah aktivitas pada kelas *view* *v_admin_aktivitas*, kemudian *view* akan mengirim perintah dengan memanggil *method* *tambah_aktivitas()* pada kelas *controller* *c_aktivitas* yang dapat menampilkan halaman tambah makanan di kelas *view* *v_admin_tambahAktivitas*. Selanjutnya aktor akan mengisi formulir tambah aktivitas, jika sudah maka akan menekan tombol simpan yang mengirim perintah dengan memanggil *method* *simpan_data* pada kelas *controller* *c_aktiivitas*, dilanjutkan dengan mengirim perintah menyimpan data pada kelas *M_aktivitas* dengan *method* *simpan_data*. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah pengguna pada kelas *view* *v_admin_tambahAktivitas*, apabila isian benar maka dilanjutkan menuju *view* *v_admin_aktivitas*. Langkah selanjutnya apabila terdapat kesalahan pada data aktivitas dan aktor ingin mengubah data aktivitas tersebut, maka aktor akan mengirim pesan dengan menekan tombol *update* pada kelas *view* *v_admin_aktivitas* kemudian perintah tersebut akan memanggil *method* *update()* pada kelas *controller* *c_aktivitas*, selanjutnya dari kelas *controller* memanggil *method* *get_aktivitas* pada kelas model *M_aktivitas* yang akan menampilkan formulir data aktivitas yang dipilih pada kelas *view* *v_admin_editAktivitas*. Setelah melakukan perubahan, maka aktor akan mengirim pesan dengan menekan tombol simpan pada kelas *view*

v_admin_editAktivitas yang kemudian memanggil *method* ubah_data pada kelas *controller* c_aktivitas, dari kelas *controller* c_aktivitas dilanjutkan dengan penyimpanan pada kelas model M_aktivitas. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah makanan pada kelas *view* v_admin_editAktivitas, apabila isian benar maka dilanjutkan menuju *view* v_admin_aktivitas. Penggambaran *sequence* diagram mengelola data aktivitas digunakan untuk menjelaskan fungsi yang akan dibuat seperti yang ditunjukkan pada lampiran B Gambar 7.

4.3.4.9 Sequence Diagram Menambah Data Aktivitas

Pada *sequence* diagram menambah data aktivitas terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram menambah data aktivitas diawali ketika aktor melakukan perintah dengan menekan tombol tambah aktivitas pada kelas *view* v_user_rekom, kemudian *view* akan mengirim perintah dengan memanggil *method* tambah_aktivitas1() pada kelas *controller* c_aktivitas yang dapat menampilkan halaman tambah aktivitas di kelas *view* v_user_inputAktivitas. Selanjutnya aktor akan mengisi formulir tambah aktivitas, jika sudah maka akan menekan tombol simpan yang mengirim perintah dengan memanggil *method* simpan_data pada kelas *controller* c_aktiivitas, dilanjutkan dengan mengirim perintah menyimpan data pada kelas M_aktivitas dengan *method* simpan_data. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah pengguna pada kelas *view* v_user_inputAktivitas, apabila isian benar maka dilanjutkan menuju *view* v_user_rekom. Penggambaran *sequence* diagram menambah data aktivitas digunakan untuk menjelaskan fungsi yang akan dibuat seperti yang ditunjukkan pada lampiran B Gambar 8.

4.3.4.10 Sequence Diagram Mengelola Data Posisi Badan

Pada *sequence* diagram mengelola data posisi badan terdapat interaksi dari beberapa kelas. Proses dari aliran *sequence* diagram mengelola data posisi badan diawali ketika aktor melakukan perintah klik menu posisi badan pada kelas *view* v_admin_menu, selanjutnya *view* akan mengirim perintah dengan memanggil *method* posisibadan() yang terdapat pada kelas *controller* c_posisibadan, kemudian

mengambil data posisi badan pada kelas *M_posisibadan* dengan *method* *data_posisibadan()*, hasilnya akan menampilkan semua data posisi badan pada tabel. Selanjutnya proses menambah posisi badan, pertama aktor melakukan perintah dengan menekan tombol tambah posisi badan pada kelas *view* *v_admin_posisibadan*, kemudian *view* akan mengirim perintah dengan memanggil *method* *tambah_posisibadan()* pada kelas *controller* *c_posisibadan* yang dapat menampilkan halaman tambah makanan di kelas *view* *v_admin_inputPosisibadan*. Selanjutnya aktor akan mengisi formulir tambah posisi badan, jika sudah maka akan menekan tombol simpan yang mengirim perintah dengan memanggil *method* *simpan_data* pada kelas *controller* *c_posisibadan*, dilanjutkan dengan mengirim perintah menyimpan data pada kelas *M_posisibadan* dengan *method* *simpan_data*. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah pengguna pada kelas *view* *v_admin_inputPosisibadan*, apabila isian benar maka dilanjutkan menuju *view* *v_admin_posisibadan*. Langkah selanjutnya apabila terdapat kesalahan pada data posisi badan dan aktor ingin mengubah data posisi badan tersebut, maka aktor akan mengirim pesan dengan menekan tombol *update* pada kelas *view* *v_admin_posisibadan* kemudian perintah tersebut akan memanggil *method* *update()* pada kelas *controller* *c_posisibadan*, selanjutnya dari kelas *controller* memanggil *method* *get_posisibadan* pada kelas model *M_posisibadan* yang akan menampilkan formulir data posisi badan yang dipilih pada kelas *view* *v_admin_editPosisibadan*. Setelah melakukan perubahan, maka aktor akan mengirim pesan dengan menekan tombol simpan pada kelas *view* *v_admin_editPosisibadan* yang kemudian memanggil *method* *ubah_data* pada kelas *controller* *c_posisibadan*, dari kelas *controller* *c_posisibadan* dilanjutkan dengan penyimpanan pada kelas model *M_posisibadan*. Selanjutnya melakukan pengecekan apabila terdapat isian yang kosong maka sistem akan menampilkan pesan singkat dan mengembalikan pada halaman pengisian formulir tambah makanan pada kelas *view* *v_admin_editPosisibadan*, apabila isian benar maka dilanjutkan menuju *view* *v_admin_posisibadan*. Selanjutnya apabila aktor ingin mengaktifkan suatu posisi badan, pertama aktor akan mengirim pesan dengan menekan tombol aktifkan pada salah satu posisi badan pada kelas *view*

v_admin_posisibadan, kemudian perintah tersebut akan memanggil *method* aktifkan() pada kelas *controller* c_posisibadan dilanjutkan dengan penyimpanan perubahan pada model M_posisibadan, jika aktivasi posisi badan salah maka akan menampilkan pesan singkat pada kelas *view* v_admin_posisibadan, jika aktivasi benar maka posisi badan akan aktif pada kelas *view* v_admin_posisibadan. Kemudian apabila aktor ingin menonaktifkan suatu posisi badan, pertama aktor akan mengirim pesan dengan menekan tombol nonaktifkan pada salah satu posisi badan pada kelas *view* v_admin_posisibadan, kemudian perintah tersebut akan memanggil *method* nonaktifkan() pada kelas *controller* c_posisibadan dilanjutkan dengan penyimpanan perubahan pada model M_posisibadan, jika nonaktifasi posisi badan salah maka akan menampilkan pesan singkat pada kelas *view* v_admin_posisibadan, jika nonaktifasi benar maka posisi badan akan non-aktif pada kelas *view* v_admin_posisibadan. Penggambaran *sequence* diagram mengelola posisi badan digunakan untuk menjelaskan fungsi yang akan dibuat seperti yang ditunjukkan pada lampiran B Gambar 9.

4.3.4.11 *Sequence* Diagram Keluar

Pada *sequence* diagram keluar merupakan langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk melakukan proses keluar dari sistem. Proses dari aliran *sequence* diagram keluar diawali ketika aktor melakukan perintah klik *logout* pada kelas *view* yang aktif, selanjutnya *view* akan mengirim perintah dengan memanggil *method* logout() yang terdapat pada kelas *controller* c_login, selanjutnya akan menampilkan halaman login. Interaksi antar kelas dan *function* yang digunakan dalam proses pengkodean selanjutnya dapat dilihat pada lampiran B Gambar 10.

4.3.5 *Activity* Diagram

Activity diagram berfungsi untuk menggambarkan alur aktivitas yang akan dijalankan Sistem Penunjang Keputusan Rekomendasi Menu Makanan dengan user dalam bentuk diagram aktivitas.

4.3.5.1 *Activity* Diagram Masuk

Activity diagram masuk menggambarkan aksi aktor dan reaksi sistem ketika pengguna akan masuk ke sistem. Aktor membuka sistem informasi kemudian

sistem menampilkan halaman masuk, setelah itu aktor mengisi *username* dan *password* kemudian klik tombol login, selanjutnya sistem akan menampilkan pesan jika ada kesalahan saat aktor mengisi *username* atau *password*, jika tidak ada kesalahan maka sistem dapat masuk ke halaman sesuai dengan hak akses dari pengguna. *Activity* diagram masuk selengkapnya dapat dilihat pada lampiran C Gambar 1.

4.3.5.2 *Activity* Diagram Menambah Pengguna Baru

Activity diagram menambah pengguna baru menggambarkan aksi aktor dan reaksi sistem ketika pengguna akan menambah pengguna baru. Aktor memilih menu *Sign Up* kemudian sistem menampilkan halaman pendaftaran pengguna baru, kemudian aktor mengisi form pendaftaran pengguna baru dan klik tombol sign up selanjutnya sistem akan menampilkan pesan jika terdapat form isian yang kosong, jika benar maka data akan tersimpan ke database dan menampilkan halaman login. *Activity* diagram menambah pengguna baru dapat dilihat selengkapnya pada lampiran C Gambar 2.

4.3.5.3 *Activity* Diagram Mengelola Data Pengguna

Activity diagram mengelola data pengguna menggambarkan aksi user dan reaksi sistem ketika user akan mengelola data pengguna. Pertama aktor memilih menu pengguna kemudian sistem akan menampilkan halaman pengguna. Pada halaman pengguna aktor dapat melakukan penambahan pengguna atau mengubah data pengguna yang sudah ada. Jika aktor ingin menambah pengguna maka aktor memilih tombol tambah pengguna kemudian sistem akan menampilkan form tambah data pengguna, selanjutnya aktor mengisi form tambah data pengguna jika sudah selesai maka klik tombol simpan. Setelah itu maka sistem akan menampilkan pesan jika terdapat isian form yang kosong dan mengembalikan ke halaman form tambah pengguna, jika tidak ada isian yang kosong maka sistem akan menyimpan data yang sudah diisi oleh aktor. Selanjutnya, jika aktor ingin mengubah data pengguna maka aktor harus memilih tombol edit pada salah satu data pengguna yang akan diubah. Kemudian sistem akan menampilkan form data yang dipilih oleh aktor, selanjutnya aktor mengubah data pada form lalu klik simpan jika selesai mengubah data. Reaksi sistem berikutnya memeriksa data yang kosong jika terdapat data yang kosong maka sistem akan menampilkan pesan dan

mengembalikan ke halaman form data yang dipilih, jika tidak ada isian yang kosong maka sistem akan menyimpan perubahan data tersebut ke dalam database dan menampilkan halaman pengguna. Activity diagram mengelola data pengguna dapat dilihat selengkapnya pada lampiran C Gambar 3.

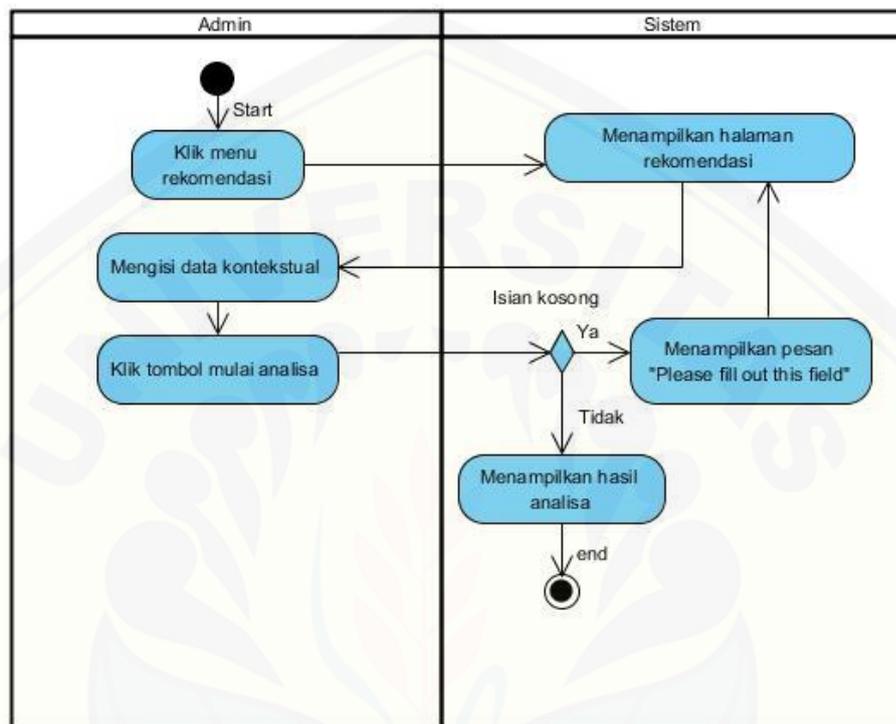
4.3.5.4 Activity Diagram Mengelola Data Makanan

Activity diagram mengelola data makanan menggambarkan aksi user dan reaksi sistem ketika user akan mengelola data makanan. Pada diagram mengelola data makanan aktor dapat melakukan penambahan data makanan atau mengubah data makanan. Pertama jika aktor menambah data makanan, klik menu makanan kemudian pilih tombol tambah maka sistem akan menampilkan form tambah makanan. Selanjutnya aktor mengisi form tambah makanan kemudian klik simpan, reaksi sistem berikutnya memeriksa data yang kosong, jika terdapat data yang kosong maka sistem akan menampilkan pesan dan mengembalikan pada halaman form tambah makanan, jika tidak ada isian yang kosong maka sistem akan menyimpan ke database dan menampilkan halaman data makanan. Kedua jika aktor ingin mengubah data makanan, klik menu makanan kemudian pilih tombol edit pada data makanan yang ingin diubah selanjutnya sistem menampilkan form edit data makanan kemudian aktor mengubah data makanan lalu klik tombol simpan. Reaksi sistem selanjutnya memeriksa data yang kosong jika terdapat data yang kosong maka sistem akan menampilkan pesan dan mengembalikan ke halaman form data yang dipilih, jika tidak ada isian yang kosong maka sistem akan menyimpan perubahan data tersebut ke dalam database dan menampilkan halaman pengguna. Activity diagram mengelola data makanan selengkapnya dapat dilihat pada lampiran C Gambar 4.

4.3.5.5 Activity Diagram Melihat Hasil Rekomendasi Menu Makanan

Activity diagram melihat hasil rekomendasi menu makanan menggambarkan aksi user dan reaksi sistem ketika user akan melihat hasil rekomendasi menu makanan. Pada *activity* diagram melihat hasil rekomendasi menu makanan pertama yang dilakukan oleh aktor klik menu rekomendasi kemudian sistem akan menampilkan halaman rekomendasi, setelah itu aktor mengisi formulir data kontekstual, kemudian klik tombol analisa dan selanjutnya rekasi dari sistem akan mengecek apabila terdapat isian formulir yang kosong jika

terdapat isian yang kosong maka sistem akan menampilkan pesan dan mengembalikan pada halaman rekomendasi, jika sudah benar maka sistem akan menampilkan hasil rekomendasi menu makanan yang sesuai dengan perhitungan yang sudah dilakukan. *Activity* diagram melihat hasil rekomendasi menu makanan dapat dilihat pada Gambar 4.4.



Gambar 4. 4 *Activity* Diagram Melihat Hasil Rekomendasi Menu Makanan

4.3.5.6 *Activity* Diagram Mengelola Data Kombinasi Menu Makanan

Activity diagram mengelola data kombinasi makanan menggambarkan aksi user dan reaksi sistem ketika user akan mengelola data kombinasi makanan. Pada diagram mengelola data kombinasi makanan aktor dapat menambah atau mengubah data kombinasi makanan. Pertama jika aktor menambah data kombinasi makanan, klik menu kombinasi makanan kemudian pilih tombol tambah maka sistem akan menampilkan form tambah makanan. Selanjutnya aktor mengisi form tambah makanan kemudian klik simpan, reaksi sistem berikutnya memeriksa data yang kosong, jika terdapat data yang kosong maka sistem akan menampilkan pesan dan mengembalikan pada halaman form tambah makanan, jika tidak ada isian yang kosong maka sistem akan menyimpan ke database dan menampilkan halaman data

makanan. Kedua jika aktor ingin mengubah data kombinasi makanan, klik menu kombinasi makanan kemudian pilih tombol edit pada data makanan yang ingin diubah selanjutnya sistem menampilkan form edit data makanan kemudian aktor mengubah data makanan lalu klik tombol simpan. Reaksi sistem selanjutnya memeriksa data yang kosong jika terdapat data yang kosong maka sistem akan menampilkan pesan dan mengembalikan ke halaman form data yang dipilih, jika tidak ada isian yang kosong maka sistem akan menyimpan perubahan data tersebut ke dalam database dan menampilkan halaman pengguna. Activity diagram mengelola data makanan selengkapnya dapat dilihat pada lampiran C Gambar 5.

4.3.5.7 Activity Diagram Melihat Laporan Hasil Rekomendasi Menu Makanan

Activity diagram melihat laporan hasil rekomendasi menu makanan menggambarkan aksi user dan reaksi sistem ketika user akan melihat laporan hasil rekomendasi menu makanan dilihat pada lampiran C Gambar 6.

4.3.5.8 Activity Diagram Mengelola Data Aktivitas

Activity diagram mengelola data aktivitas menggambarkan aksi user dan reaksi sistem ketika user akan mengelola data aktivitas. Pada diagram mengelola data aktivitas aktor dapat menambah atau mengubah data aktivitas. Pertama jika aktor menambah data aktivitas, klik menu aktivitas kemudian pilih tombol tambah maka sistem akan menampilkan form tambah aktivitas. Selanjutnya aktor mengisi form tambah aktivitas kemudian klik simpan, reaksi sistem berikutnya memeriksa data yang kosong, jika terdapat data yang kosong maka sistem akan menampilkan pesan dan mengembalikan pada halaman form tambah aktivitas, jika tidak ada isian yang kosong maka sistem akan menyimpan ke database dan menampilkan halaman data aktivitas. Kedua jika aktor ingin mengubah data aktivitas, klik menu aktivitas kemudian pilih tombol edit pada data aktivitas yang ingin diubah selanjutnya sistem menampilkan form edit data aktivitas kemudian aktor mengubah data aktivitas lalu klik tombol simpan. Reaksi sistem selanjutnya memeriksa data yang kosong jika terdapat data yang kosong maka sistem akan menampilkan pesan dan mengembalikan ke halaman form data yang dipilih, jika tidak ada isian yang kosong maka sistem akan menyimpan perubahan data tersebut ke dalam database dan menampilkan halaman pengguna. Activity diagram mengelola data aktivitas selengkapnya dapat dilihat pada lampiran C Gambar 7.

4.3.5.9 Activity Diagram Menambah Data Aktivitas

Activity diagram menambah data aktivitas menggambarkan aksi aktor dan reaksi sistem ketika pengguna akan menambah data aktivitas. Aktor memilih menu rekomendasi kemudian aktor dapat memilih tombol tambah aktivitas jika aktivitas yang diinginkan tidak ada pada pilihan dari sistem kemudian sistem menampilkan halaman form tambah data aktivitas, kemudian aktor mengisi form data aktivitas dan klik tombol simpan selanjutnya sistem akan menampilkan pesan jika terdapat form isian yang kosong, jika benar maka data akan tersimpan ke database dan menampilkan halaman login. Activity diagram menambah data aktivitas dapat dilihat selengkapnya pada lampiran C Gambar 8.

4.3.5.10 Activity Diagram Mengelola Data Posisi Badan

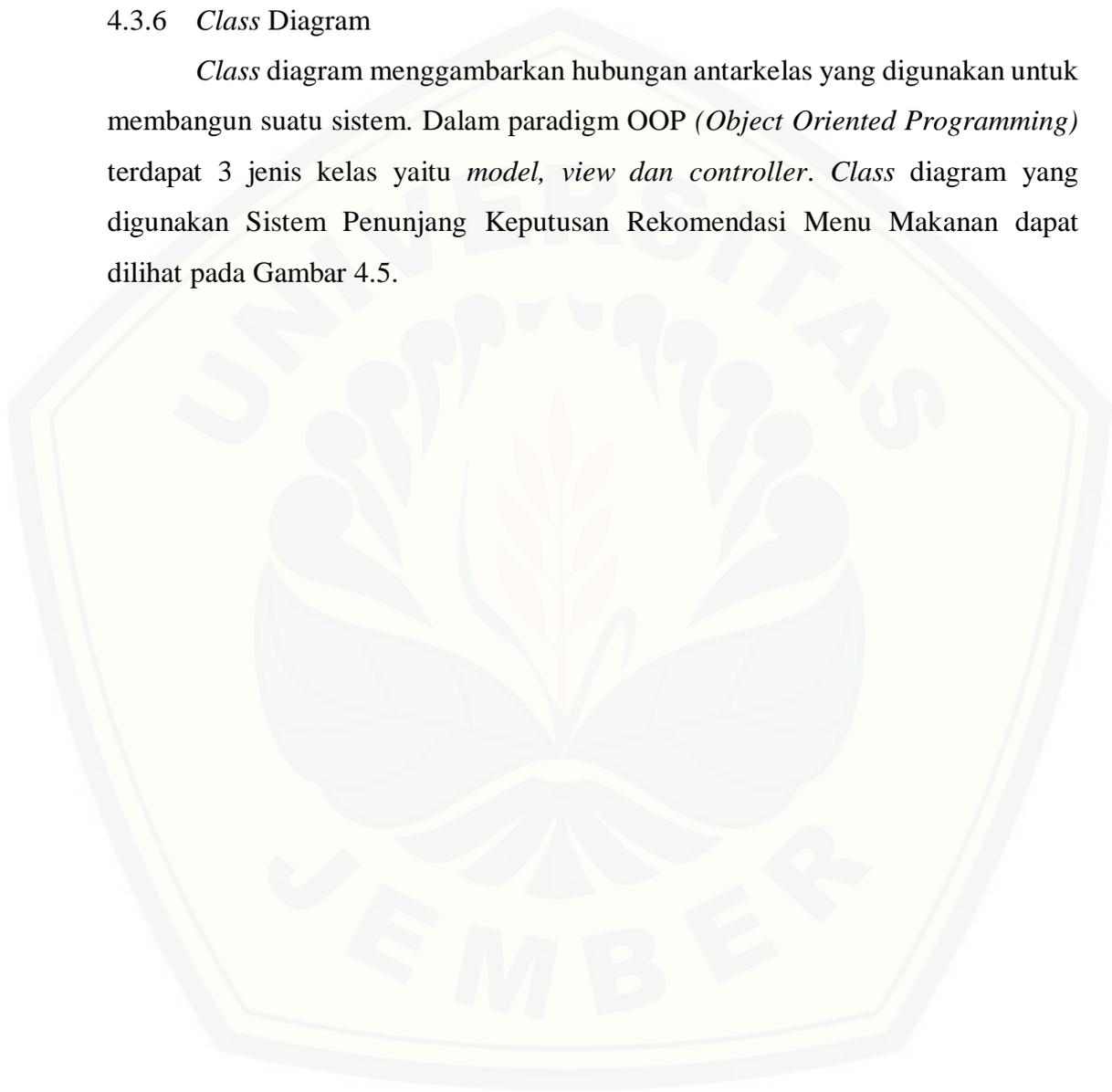
Activity diagram mengelola data posisi badan menggambarkan aksi user dan reaksi sistem ketika user akan mengelola data posisi badan. Pada diagram mengelola data posisi badan aktor dapat menambah atau mengubah data posisi badan. Pertama jika aktor menambah data posisi badan, klik menu posisi badan kemudian pilih tombol tambah maka sistem akan menampilkan form tambah posisi badan. Selanjutnya aktor mengisi form tambah posisi badan kemudian klik simpan, reaksi sistem berikutnya memeriksa data yang kosong, jika terdapat data yang kosong maka sistem akan menampilkan pesan dan mengembalikan pada halaman form tambah posisi badan, jika tidak ada isian yang kosong maka sistem akan menyimpan ke database dan menampilkan halaman data posisi badan. Kedua jika aktor ingin mengubah data posisi badan, klik menu posisi badan kemudian pilih tombol edit pada data posisi badan yang ingin diubah selanjutnya sistem menampilkan form edit data posisi badan kemudian aktor mengubah data posisi badan lalu klik tombol simpan. Reaksi sistem selanjutnya memeriksa data yang kosong jika terdapat data yang kosong maka sistem akan menampilkan pesan dan mengembalikan ke halaman form data yang dipilih, jika tidak ada isian yang kosong maka sistem akan menyimpan perubahan data tersebut ke dalam database dan menampilkan halaman pengguna. Activity diagram mengelola data posisi badan selengkapnya dapat dilihat pada lampiran C Gambar 9.

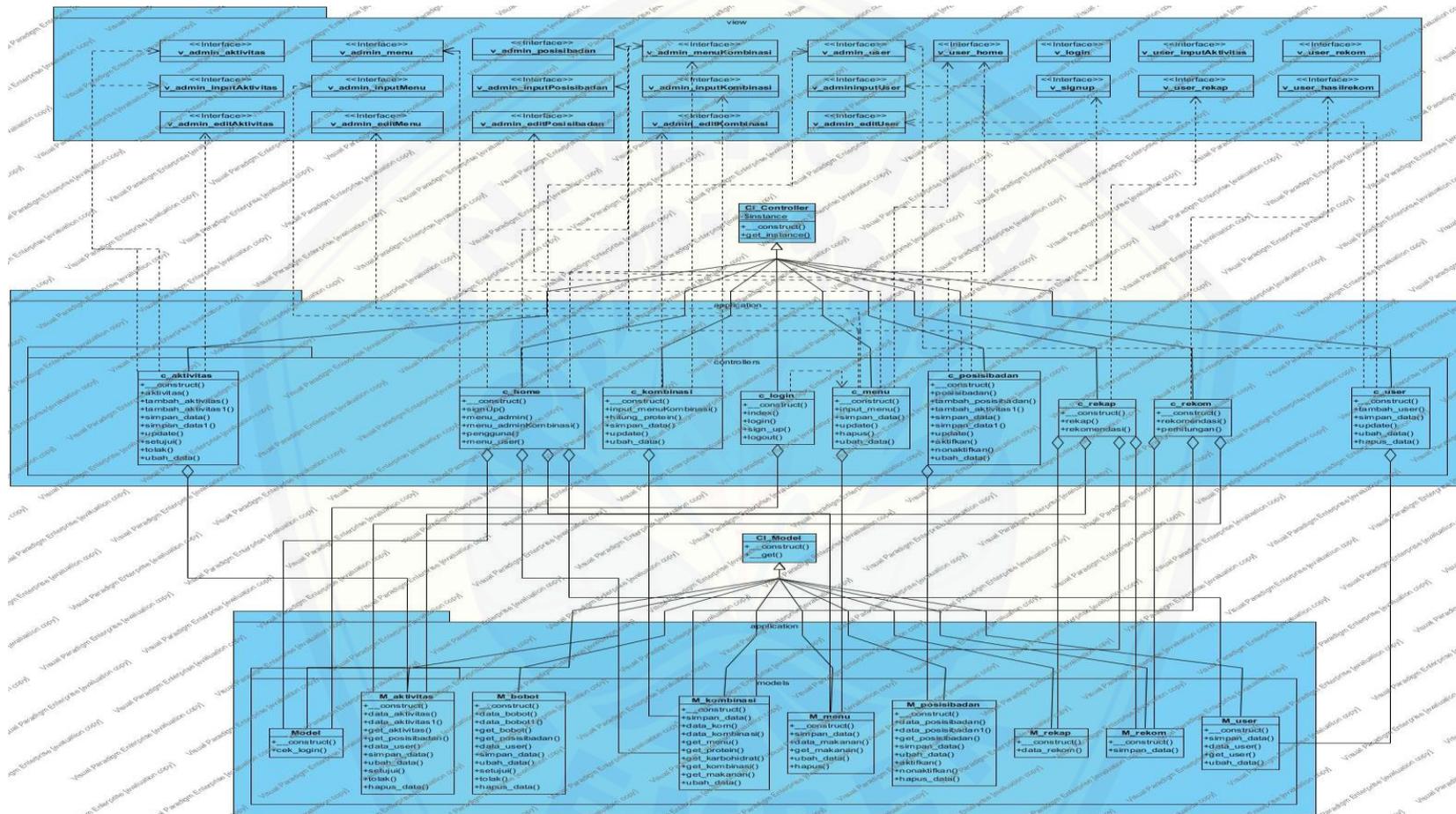
4.3.5.11 Activity Diagram Keluar

Activity diagram menambah keluar menggambarkan aksi aktor dan reaksi sistem ketika pengguna akan keluar dari sistem. Aktor memilih tombol *logout* pada halaman yang sedang aktif, kemudian sistem menampilkan halaman *login*. *Activity* diagram keluar dapat dilihat selengkapnya pada lampiran C Gambar 10.

4.3.6 Class Diagram

Class diagram menggambarkan hubungan antarkelas yang digunakan untuk membangun suatu sistem. Dalam paradig *OOP (Object Oriented Programming)* terdapat 3 jenis kelas yaitu *model, view dan controller*. *Class* diagram yang digunakan Sistem Penunjang Keputusan Rekomendasi Menu Makanan dapat dilihat pada Gambar 4.5.





Gambar 4. 5 Class Diagram

4.3.7 *Entity Relationship Diagram (ERD)*

ERD merupakan gambaran komponen dan struktur database yang digunakan dalam pembangunan sistem. ERD yang digunakan Sistem Penunjang Keputusan Rekomendasi Menu Makanan dapat dilihat pada Gambar 4.6.





Gambar 4. 6 Entity Relationship Diagram

4.4 Implementasi

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

Pada proses pengkodean mengimplementasikan metode *Weighted Sum Model* pada fitur rekomendasi. Fitur tersebut terdapat pada view di *database* serta kelas *c_rekom*, *M_aktivitas*, *M_kombinasi*, *M_rekom*.

1. Kelas *c_rekom*

Kelas *c_rekom* berfungsi untuk mengatur jalannya data antara model dengan tampilan. Pada kelas tersebut terdapat proses untuk menampilkan hasil rekomendasi metode *Weighted Sum Model* yang telah dibuat pada table view di *database*. Penulisan kode program fitur rekomendasi pada kelas *c_rekom* dapat dilihat pada Gambar 4.7.

```

1 public function perhitungan() {
2     $faktor_aktivitas = $this->input->post('faktor_aktivitas');
3     $posisibadan = $this->input->post('posisibadan');
4     $waktu = $this->input->post('waktu');
5     // $harga = $this->input->post('harga');
6     // $protein = $this->input->post('protein');
7     // $karbo = $this->input->post('karbo');
8     $cek = $this->input->post('bobot');
9     $berat_badan = $this->input->post('berat_badan');
10    $tinggi_badan = $this->input->post('tinggi_badan');
11    if (isset($posisibadan)) {
12        $a=0;
13        foreach ($posisibadan as $row){
14            if ($row == 0) { unset($posisibadan[$a]); }
15            $a++;
16        }
17    }
18
19    if (isset($waktu)) {
20        $a=0;
21        foreach ($waktu as $row){
22            if ($row == 0) { unset($waktu[$a]); }

```

```
23         $a++;
24     }
25 }
...
...
...
293 $wsm2 = array_msort($wsm, array('wsm'=>SORT_DESC));
294 $umur = $year_diff;
295 $data_user[0]['umur'] = $year_diff;
296 $data_user[0]['kal'] = $kal;
297 $data_user[0]['tinggi_badan'] = $tinggi_badan;
298 $data_user[0]['berat_badan'] = $berat_badan;
299 if ($data_user[0]['jenis_kelamin'] == 1) {
300     $data_user[0]['jenis_kelamin'] = "Laki Laki";
301 }else{
302     $data_user[0]['jenis_kelamin'] = "Perempuan";
303 }
304 $id_user = $data_user[0]['id_user'];
305 $id_kombinasi_makanan = 0;
306
307 $sterbesar = 0;
308 foreach ($wsm2 as $key) {
309     if ($key['wsm'] > $sterbesar) {
310         $sterbesar = $key['wsm'];
311         $id_kombinasi_makanan = $key['id_makanan'];
312     }
313 }
314 $tanggal_analisa = date('Y-m-d');
315 $kebutuhan_kalori = $kal;
316 $kebutuhan_karbohidrat = $kebutuhan_karbo;
317 $data_input = array(
318     'id_user' => $id_user,
319     'id_kombinasi_makanan' => $id_kombinasi_makanan,
320     'kebutuhan_kalori' => $kebutuhan_kalori,
321     'kebutuhan_protein' => $kebutuhan_protein,
322     'kebutuhan_karbohidrat' => $kebutuhan_karbohidrat,
323     'tinggi_badan' => $tinggi_badan,
324     'berat_badan' => $berat_badan,
325     'umur' => $umur,
326     'tanggal_analisa' => $tanggal_analisa);
327 )
328 $logic = $this->M_rekom->simpan_data($data_input);
329
330 $data = array(
331     'kebutuhan_kalori' => $kal,
332     'kebutuhan_protein' => $kebutuhan_protein,
333     'kebutuhan_karbohidrat' => $kebutuhan_karbohidrat,
334     'user' => $data_user,
```

335	'data_wsm'=>\$wsm2
336);
337	\$this->load->view('v_user_hasilrekom',\$data);

Gambar 4. 7 Kode Program c_rekom

Pada Gambar 4.7 merupakan potongan kode program dari kelas c_rekom. Pada kode program baris 1 – 292 merupakan fungsi proses perhitungan kalori. Baris 293 – 327 merupakan fungsi untuk merekomendasikan menu makanan. Baris 314 – 328 merupakan fungsi proses menyimpan hasil rekomendasi ke database. Baris 330-337 merupakan fungsi menampilkan hasil rekomendasi.

2. Kelas m_aktivitas

Kelas m_aktivitas merupakan kelas yang berfungsi sebagai perantara interaksi antara database dan sistem. Penulisan kode program kelas m_aktivitas dapat dilihat pada Gambar 4.8 berikut.

1	Class M_aktivitas extends CI_Model {
2	function get_aktivitas(\$id) {
3	return \$this->db->query('SELECT * FROM aktivitas where
4	id_a ktivitas = '.\$id.'')->result_array();
5	}
6	function get_posisibadan(\$status) {
7	\$this->db->where('status',\$status);
8	return \$query = \$this->db->get('posisibadan')->result_array();
9	}

Gambar 4. 8 Kode Program m_aktivitas

Pada Gambar 4.8 merupakan potongan kode program dari kelas m_aktivitas. Pada baris 1 class m_aktivitas merupakan anak class dari class CI_Model. Baris 2 – 5 merupakan fungsi untuk mendapatkan data aktivitas dari tabel aktivitas pada database. Baris 6 – 9 merupakan fungsi untuk mendapatkan data posisi badan dari tabel posisi badan pada database.

3. Kelas m_kombinasi

Kelas m_kombinasi merupakan kelas yang berfungsi sebagai perantara interaksi antara database dan sistem. Penulisan kode program kelas m_aktivitas dapat dilihat pada Gambar 4.9 berikut.

1	class M_kombinasi extends CI_Model {
2	function simpan_data(\$data) {
3	return \$this->db->insert('kombinasi_makanan', \$data);
4	}
5	function data_kom() {
6	return \$this->db->query("SELECT * FROM kombinasi_makanan")-
7	>result_array();
8	}

Gambar 4. 9 Kode Program M_kombinasi

Pada Gambar 4.9 merupakan potongan kode program dari kelas m_kombinasi. Pada baris 1 class m_kombinasi merupakan anak class dari class CI_Model. Baris 2 – 4 merupakan fungsi untuk menyimpan data kombinasi makanan pada database. Baris 5 – 8 merupakan fungsi untuk mendapatkan data kombinasi makanan dari tabel kombinasi makanana pada database.

4. Kelas m_rekom

Kelas m_rekom merupakan kelas yang berfungsi sebagai perantara interaksi antara database dan sistem. Penulisan kode program kelas m_rekom dapat dilihat pada Gambar 4. 10 berikut.

1	Class M_rekom extends CI_Model {
2	function simpan_data(\$data) {
3	return \$this->db->insert('rekomendasi', \$data);
4	}
5	}

Gambar 4. 10 Kode Program M_rekom

Pada Tabel 4.8 merupakan potongan kode program dari kelas m_kombinasi. Pada baris 1 class m_kombinasi merupakan anak class dari class CI_Model. Baris 2 – 5 merupakan fungsi untuk menyimpan data rekomendasi menu makanan pada database.

4.5 Pengujian Sistem

Pengujian sistem merupakan salah satu tahapan metode waterfall. Tahap tersebut dilakukan setelah melakukan proses kode program. Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *blackbox* menggunakan *requirement testing*. Pengujian *blackbox* merupakan pengujian yang dilakukan oleh pengguna dengan menjalankan program

secara langsung dan menganalisis *input* dan *output* yang dihasilkan aplikasi. Berikut adalah hasil Requirements testing terdapat pada Tabel 4.6.

Tabel 4. 6 Hasil Requirements Testing

1.No.	2. Fitur	3. Aksi	4. Hasil	5. Ket	
1.	Masuk	Mengisi form username lalu klik tombol Masuk	Masuk berhasil dan user masuk ke dalam sistem informasi sesuai hak akses pengguna	Berhasil	
		Klik tombol Masuk namun username atau password kosong	Menampilkan span "Please fill out this field"	Berhasil	
		Klik tombol Masuk namun username atau password tidak sesuai	Menampilkan span "These credentials do not match our records"	Berhasil	
2.	Menambah Data Pengguna Baru	Klik menu sign up	Menampilkan halaman berisi form sign up/pendaftaran pengguna	Berhasil	
			Mengisi form lalu klik sign up	Menyimpan data ke database dan menampilkan halaman login	Berhasil
			Klik sign up jika terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
3.	Melihat Data Pengguna	Klik menu pengguna	Menampilkan halaman berisi tabel data pengguna, tombol tambah pengguna dan tombol edit	Berhasil	

Dilanjutkan

Lanjutan

1.	2.	3.	4.	5.
		Klik tombol simpan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil

4.	Mengubah Data Pengguna	Klik menu pengguna	Menampilkan halaman berisi tabel pengguna, tombol tambah pengguna, dan tombol edit	Berhasil
		Klik tombol edit pada salah satu data pengguna di tabel pengguna	Menampilkan halaman form edit data pengguna dan tombol simpan	Berhasil
		Klik tombol simpan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
5.	Melihat Data Makanan	Klik menu makanan	Menampilkan halaman berisi tabel daftar makanan, tombol edit, dan tombol tambah makanan	Berhasil
6.	Menambah Data Makanan	Klik menu makanan, lalu klik tombol tambah makanan	Menampilkan formulir tambah makanan dan tombol simpan	Berhasil
		Mengisi form tambah makanan lalu klik simpan	Menyimpan data ke database dan menampilkan halaman lihat data makanan	Berhasil
		Klik tombol tambahkan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
7.	Mengubah Data Makanan	Klik menu makanan	Menampilkan halaman berisi tabel makanan, tombol tambah makanan, dan tombol edit	Berhasil

Dilanjutkan

Lanjutan

1.	2.	3.	4.	5.
----	----	----	----	----

		Klik tombol simpan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
		Klik tombol simpan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
8.	Melihat Rekomendasi Menu Makanan	Klik menu Rekomendasi	Menampilkan halaman form rekomendasi	Berhasil
		Mengisi form rekomendasi lalu klik analisa	Menampilkan data pengguna dalam formulir dan hasil rekomendasi menu makanan dalam tabel	Berhasil
9.	Melihat Data Kombinasi Makanan	Klik menu kombinasi makanan	Menampilkan halaman berisi tabel daftar kombinasi makanan, tombol edit, dan tombol tambah kombinasi makanan	Berhasil
10.	Menambah Data Kombinasi Makanan	Klik menu kombinasi makanan, lalu klik tombol tambah kombinasi makanan	Menampilkan formulir tambah kombinasi makanan dan tombol simpan	Berhasil
		Mengisi form tambah kombinasi makanan lalu klik simpan	Menyimpan data ke database dan menampilkan halaman lihat data kombinasi makanan	Berhasil
		Klik tombol tambahkan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil

Dilanjutkan

Lanjutan

1.	2.	3.	4.	5.
----	----	----	----	----

11.	Mengubah Data Kombinasi Makanan	Klik menu kombinasi makanan	Menampilkan halaman berisi tabel kombinasi makanan, tombol tambah kombinasi makanan, dan tombol edit	Berhasil
		Klik tombol edit pada salah satu data kombinasi makanan di tabel makanan	Menampilkan halaman form edit data kombinasi makanan dan tombol simpan	Berhasil
		Klik tombol simpan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
12.	Melihat Laporan Rekomendasi Menu Makanan	Klik menu riwayat	Menampilkan halaman tabel daftar rekomendasi menu makanan	Berhasil
13.	Melihat Data Aktivitas	Klik menu aktivitas	Menampilkan halaman berisi tabel daftar aktivitas, tombol edit, dan tombol tambah aktivitas	Berhasil
11.	Mengubah Data Kombinasi Makanan	Klik menu kombinasi makanan	Menampilkan halaman berisi tabel kombinasi makanan, tombol tambah kombinasi makanan, dan tombol edit	Berhasil
14.	Menambah Data Aktivitas	Klik menu aktivitas, lalu klik tombol tambah aktivitas	Menampilkan formulir tambah aktivitas dan tombol simpan	Berhasil
		Mengisi form tambah aktivitas lalu klik simpan	Menyimpan data ke database dan menampilkan halaman lihat data aktivitas	Berhasil

Dilanjutkan

Lanjutan

1.	2.	3.	4.	5.
----	----	----	----	----

		Klik tombol tambahkan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
15.	Mengubah Data Aktivitas	Klik menu aktivitas	Menampilkan halaman berisi tabel aktivitas , tombol tambah aktivitas , dan tombol edit	Berhasil
		Klik tombol edit pada salah satu data aktivitas di tabel aktivitas	Menampilkan halaman form edit data aktivitas dan tombol simpan	Berhasil
		Klik tombol simpan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
16.	Melihat Data Posisi Badan	Klik menu posisi badan	Menampilkan halaman berisi tabel daftar posisi badan , tombol edit, dan tombol aktifasi	Berhasil
17.	Menambah Data Posisi Badan	Klik menu posisi badan, lalu klik tombol tambah posisi badan	Menampilkan formulir tambah posisi badan dan tombol simpan	Berhasil
		Mengisi form tambah posisi badan lalu klik simpan	Menyimpan data ke database dan menampilkan halaman lihat data posisi badan	Berhasil
		Klik tombol tambahkan namun terdapat form isian yang kosong	Menampilkan span "Please fill out this field"	Berhasil
18.	Mengubah atau mengaktifkan posisi badana	Klik menu posisi badan, lalu klik tombol aktifasi pada salah satu data posisi badan di tabel	Menampilkan halaman tabel daftar posisi badan, lalu posisi badan akan aktif	Berhasil
19.	Keluar	Klik menu log out	Menampilkan halaman login	Berhasil

BAB 5 HASIL DAN PEMBAHASAN

BAB 6 PENUTUP

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

6.1. Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan adalah sebagai berikut:

1. Dari hasil penelitian yang telah dipaparkan diatas, cara menerapkan metode *Weighted Sum Model* pada sistem rekomendasi makanan adalah pertama terdapat data inputan berupa data kontekstual (tanggal lahir, berat badan, tinggi badan, dan aktivitas) dan data alternatif makanan. Selanjutnya menentukan kriteria dan nilai bobot dari masing-masing kriteria, dalam rekomendasi menu makanan terdapat tiga kriteria pemilihan yaitu kalori, protein dan karbohidrat. Sedangkan untuk pemberian nilai bobot pada kriteria rekomendasi ini terdapat tiga golongan dengan pemeberian nilai ekstrim yaitu, pertama menitikberatkan pada kriteria kalori, kedua menitikberatkan pada kriteria protein dan ketiga menitikberatkan pada kriteria karbohidrat. Kemudian masuk pada perhitungan *weighted sum model* yaitu, pertama dengan cara menghitung nilai performa kriteria dari masing-masing alternatif makanan. Setelah menemukan nilai performa kriteria dari masing-masing alternatif makanan, selanjutnya menghitung nilai WSM dari masing alternatif makanan, sepuluh nilai WSM terbesar adalah hasil rekomendasi menu makanan. Cara penerapan manual WSM dapat dilihat pada Tabel 5.1 – 5.7
2. Impelementasi Metode *Weighted Sum Model* pada Sistem Penunjang Keputusan Rekomendasi Menu Makanan Berdasarkan Tingkat Kebutuhan Kalori mampu menjadi website untuk membantu proses rekomenndasi makanan dengan memperhitungkan kalori. Proses membangun sistem terdiri dari observasi dan studi literatur untuk mendapatkan hasil berupa data makanan, data kriteria serta nilai bobot dari masing-masing kriteria yang digunakan sebagai acuan rekomendasi menu makanan. Kemudian tahap implementasi *coding* dimana data kontekstual dari pengguna (tanggal lahir, berat badan, tinggi badan, dan aktivitas) dan data alternatif makanan

merupakan data inputan pada sistem. Selanjutnya menerapkan metode *weighted sum model* pada sistem dapat dilihat pada Gambar 5.1 - 5.8, hasil rekomendasi menu makanan ditampilkan dalam bentuk tabel rekomendasi. Sistem yang dibangun dapat membantu pengguna dalam proses rekomendasi menu makanan sesuai dengan kebutuhan kalori harian pengguna dan mengolah nilai hingga menghasilkan nilai akhir yang sama dengan penerapan perhitungan manual metode *weighted sum model* sebagai acuan untuk proses rekomendasi.

6.2.Saran

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

1. Dalam penelitian ini membutuhkan pengembangan lebih lanjut dengan menambahkan kriteria pada proses rekomendasi menu makanan.
2. Implementasi metode *Weighted Sum Model* pada sistem penunjang keputusan rekomendasi menu makanan ini alangkah baiknya jika berbasis android untuk memudahkan pengguna mengakses sistem tersebut.

DAFTAR PUSTAKA

- Afandie, M. N., Cholissodin, I., & Supianto, A. A. (2013). Implementasi Metode K-Nearest Neighbor Untuk Pendukung Keputusan Menu Makanan Sehat Dan Bergizi. *Thesis*.
- Almatsier, S. (2009). *Prinsip Dasar Ilmu Gizi*. Jakarta: Gramedia Pustaka Utama.
- Ammar, N., Amelia, K., & Muhammad, A. H. (2016). Decision Support System of SMB Telkom University Roadshow Location Prioritization with Weighted Sum Model Method. *ICIMECE*, 107-108.
- Jahari, A. (2004). *Penilaian Status Gizi Berdasarkan Antropometri*. Bogor: Pulitbang Gizi dan Makanan.
- Marthin, A. T., & Abdul, S. (2010). Sistem Pendukung Keputusan Kelaikan Terbang Pada Helicopter Model Bell 205 A-1 Pusat Penerbangan TNI AD. *Jurnal Teknologi Informasi*, 94-101.
- Mesran, S., & Surya Darma Nasution, A. P. (2017). Penerapan Weighted Sum Model (WSM) Dalam Penentuan Jaminan Kesehatan Masyarakat. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, vol 2, ISSN: 2527-5771/EISSN: 2549-7839.
- Nugroho, R. A., & Ferdiana, R. (2014). Teknik Pemberian Rekomendasi Menu Makanan dengan Pendekatan Contextual Model dan Multi-Criteria Decision Making. *CITEE*, 2085-6350.
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi*. Yogyakarta: Andi.
- Rouf, A. (2012). *Pengujian Perangkat Lunak Dengan Menggunakan Whitebox dan Blackbox*. Semarang: STMIK HIMSYA.
- SNI. (7269:2009). Penilaian Beban Kerja Berdasarkan Tingkat Kebutuhan Kalori Menurut Pengeluaran energi. *Standar Nasional Indonesia (SNI)* (hal. 2-5). Jakarta: Badan Standarisasi Nasional.
- WHO. (2010). Physical Activity. *In Guide To Community Preventive Service*.
- Yeh, C.-H., & Deng, H. (1997). An Algorithm for Fuzzy Multi-Criteria Decision Making. *IEEE*, 1564-1565.

LAMPIRAN

LAMPIRAN A. SKENARIO

1. Skenario Masuk

Tabel 1 Skenario *Use Case* Masuk Sistem

Nomor Usecase	USC-01
Nama	Masuk
Aktor	Admin / Pengguna
Deskripsi Singkat	Aktor membuka halaman Masuk untuk autentifikasi hak akses sistem informasi
Prekondisi	Aktor membuka halaman Masuk
Prakondisi	Aktor berhasil masuk ke sistem informasi
Flow of Events	
Skenario Normal : Masuk	
1. Membuka sistem informasi pada <i>browser</i>	
	2. Menampilkan halaman Masuk yang berisi <ul style="list-style-type: none"> • Form yang berisi atribut: <ul style="list-style-type: none"> - <i>Username</i> - <i>Password</i> • Tombol Masuk
3. Mengisi form Masuk	
4. Klik tombol Masuk	
	5. Menampilkan halaman Dashboard untuk hak akses user
Skenario Alternatif : <i>Username</i> atau <i>Password</i> kosong	
4. Klik tombol Masuk	
	5. Menampilkan span " <i>Please fill out this field</i> "
Skenario Alternatif : <i>Username</i> dan <i>password</i> tidak sesuai	
4. Klik tombol Masuk	
	5. Menampilkan span " <i>These credentials do not match our records</i> "

2. Skenario Menambah Data Pengguna Baru

Tabel 2 Skenario Menambah Data Pengguna Baru

Nomor Usecase	USC-02
Nama	Menambah Data Pengguna Baru
Aktor	Pengguna
Deskripsi Singkat	Aktor menambah data pengguna baru
Prekondisi	Aktor melihat menu <i>sign up</i> pada halaman masuk
Prakondisi	Aktor berhasil menambah data pengguna
Flow of Events	
Skenario Normal : Menambah Data Pengguna Baru	
1. Klik menu <i>sign up</i>	
	2. Menampilkan halaman <i>sign up</i> yang berisi <ul style="list-style-type: none"> • Form yang berisi atribut: <ul style="list-style-type: none"> - Nama Lengkap - <i>Username</i> - <i>Password</i> - Tanggal Lahir - Jenis Kelamin - Alamat - No. Hp • Tombol <i>Sign Up</i>
3. Mengisi form tambah data pengguna	
4. Klik tombol <i>Sign Up</i>	
	5. Menampilkan halaman Masuk yang berisi <ul style="list-style-type: none"> • Form yang berisi atribut: <ul style="list-style-type: none"> - <i>Username</i> - <i>Password</i> • Tombol Masuk
Skenario Alternatif : Isian form kosong	
4. Klik tombol <i>Sign Up</i>	
	5. Menampilkan span " <i>Please fill out this field</i> "

3. Skenario Mengelola Data Pengguna

Tabel 3 Skenario Mengelola Data Pengguna

Nomor Usecase	USC-03
Nama	Mengelola Data Pengguna
Aktor	Admin
Deskripsi Singkat	Aktor mengelola data pengguna
Prekondisi	Aktor memilih menu pengguna
Prakondisi	Aktor berhasil melihat, menambah, dan mengubah data pengguna
Flow of Events	
Skenario Normal : Menambah Data Pengguna	
1. Klik menu pengguna	
	<p>2. Menampilkan halaman tabel daftar pengguna</p> <ul style="list-style-type: none"> • Tabel data pengguna yang berisi: <ul style="list-style-type: none"> - Id - Nama - <i>Username</i> - <i>Password</i> - Tanggal Lahir - Jenis Kelamin - Alamat - No. Hp - Status - Aksi berupa tombol <i>edit</i> • Tombol Tambah Pengguna
3. Klik tombol tambah pengguna	
	<p>4. Menampilkan form tambah data pengguna yang berisi:</p> <ul style="list-style-type: none"> - Nama lengkap - Username - Password - Tanggal lahir - Jenis kelamin - Alamat - No. Hp

<p>5. Mengisi form tambah data pengguna, kemudian klik tombol simpan</p>	<p>6. Menyimpan data ke dalam database</p> <p>7. Menampilkan halaman pengguna dengan data baru yang telah ditambahkan ke tabel</p>
<p>Skenario Alternatif : Isian form kosong</p>	
<p>5. Klik tombol simpan</p>	<p>6. Menampilkan span “<i>Please fill out this field</i>”</p>
<p>Skenario Normal : Mengubah Data Pengguna</p>	
<p>1. Klik menu pengguna</p>	
	<p>2. Menampilkan halaman tabel daftar pengguna</p> <ul style="list-style-type: none"> • Tabel data pengguna yang berisi: <ul style="list-style-type: none"> - Id - Nama - <i>Username</i> - <i>Password</i> - Tanggal Lahir - Jenis Kelamin - Alamat - No. Hp - Status - Aksi berupa tombol <i>edit</i> • Tombol Tambah Pengguna
<p>3. Klik tombol edit pada salah satu data yang akan diubah</p>	
<p>5. Mengubah data pada form pengguna, kemudian klik tombol simpan</p>	<p>4. Menampilkan halaman data pengguna dengan form data pengguna yang akan diubah.</p>
	<p>6. Menyimpan perubahan data ke dalam database</p>
	<p>7. Menampilkan halaman pengguna dengan data yang telah diubah</p>
<p>Skenario Alternatif : Isian form kosong</p>	

5. Klik tombol simpan	6. Menampilkan span “ <i>Please fill out this field</i> ”
-----------------------	---

4. Mengelola Data Makanan

Tabel 4 Skenario Mengelola Data Makanan

Nomor Usecase	USC-04
Nama	Mengelola Data Makanan
Aktor	Admin
Deskripsi Singkat	Aktor mengelola data makanan
Prekondisi	Aktor memilih menu makanan
Prakondisi	Aktor berhasil melihat, menambah, dan mengubah data makanan
Flow of Events	
Skenario Normal : Menambah Data Makanan	
1. Klik menu makanan	2. Menampilkan halaman tabel daftar makanan <ul style="list-style-type: none"> • Tabel data makanan yang berisi: <ul style="list-style-type: none"> - Id - Nama makanan - Kandungan kalori - Jenis - Keterangan - Aksi berupa tombol <i>edit</i> • Tombol Tambah Menu
3. Klik tombol tambah menu	4. Menampilkan form tambah data makanan yang berisi: <ul style="list-style-type: none"> - Nama makanan - Kandungan kalori - Jenis - Keterangan
5. Mengisi form tambah data makanan, kemudian klik tombol simpan	6. Menyimpan data ke dalam database

	7. Menampilkan halaman menu makanan dengan data baru yang telah ditambahkan ke tabel
Skenario Alternatif : Isian form kosong	
5. Klik tombol simpan	6. Menampilkan span “ <i>Please fill out this field</i> ”
Skenario Normal : Mengubah Data Makanan	
1. Klik menu makanan	
	2. Menampilkan halaman tabel daftar makanan <ul style="list-style-type: none"> • Tabel data makanan yang berisi: <ul style="list-style-type: none"> - Id - Nama makanan - Kandungan kalori - Jenis - Keterangan - Aksi berupa tombol <i>edit</i> • Tombol Tambah Makanan
3. Klik tombol edit pada salah satu data yang akan diubah	
5. Mengubah data pada form makanan, kemudian klik tombol simpan	4. Menampilkan halaman data makanan dengan form data makanan yang akan diubah.
	6. Menyimpan perubahan data ke dalam database
	7. Menampilkan halaman makanan dengan data yang telah diubah
Skenario Alternatif : Isian form kosong	
5. Klik tombol simpan	
	6. Menampilkan span “ <i>Please fill out this field</i> ”

5. Mengelola Data Kombinasi Makanan

Tabel 5 Skenario Mengelola Data Kombinasi Makanan

Nomor Usecase	USC-05
---------------	--------

Nama	Mengelola Data Kombinasi
Aktor	Admin
Deskripsi Singkat	Aktor mengelola data kombinasi makanan
Prekondisi	Aktor memilih menu kombinasi makanan
Prakondisi	Aktor berhasil melihat, menambah, dan mengubah data kombinasi makanan
Flow of Events	
Skenario Normal : Menambah Data Kombinasi Makanan	
1. Klik menu kombinasi makanan	2. Menampilkan halaman tabel daftar kombinasi makanan <ul style="list-style-type: none"> • Tabel data pengguna yang berisi: <ul style="list-style-type: none"> - Id - Menu makanan - Total kalori - Aksi berupa tombol <i>edit</i> • Tombol Tambah Kombinasi Makanan
3. Klik tombol tambah kombinasi makanan	4. Menampilkan form tambah data kombinasi makanan yang berisi: <ul style="list-style-type: none"> - Makanan pokok - Sayur - Lauk pauk - Total kalori
5. Mengisi form tambah data kombinasi makanan, kemudian klik tombol simpan	
	6. Menyimpan data ke dalam database
	7. Menampilkan halaman kombinasi makanan dengan data baru yang telah ditambahkan ke tabel
Skenario Alternatif : Isian form kosong	
7. Klik tombol simpan	
	8. Menampilkan span " <i>Please fill out this field</i> "
Skenario Normal : Mengubah Data Pengguna	

1. Klik menu kombinasi makanan	
	2. Menampilkan halaman tabel daftar kombinasi makanan <ul style="list-style-type: none"> • Tabel data kombinasi makanan yang berisi: <ul style="list-style-type: none"> - Id - Menu makanan - Total kalori - Aksi berupa tombol <i>edit</i> • Tombol Tambah Pengguna
3. Klik tombol edit pada salah satu data yang akan diubah	4. Menampilkan halaman data kombinasi makanan dengan form data kombinasi makanan yang akan diubah.
5. Mengubah data pada form kombinasi makanan, kemudian klik tombol simpan	
	6. Menyimpan perubahan data ke dalam database
	7. Menampilkan halaman kombinasi makanan dengan data yang telah diubah
Skenario Alternatif : Isian form kosong	
5. Klik tombol simpan	6. Menampilkan span " <i>Please fill out this field</i> "

6. Skenario Melihat Laporan Rekomendasi Menu Makanan

Tabel 6 Skenario Melihat Laporan Rekomendasi Makanan

Nomor Usecase	USC-06
Nama	Melihat Laporan Rekomendasi Menu Makanan
Aktor	Pengguna
Deskripsi Singkat	Aktor melihat laporan rekomendasi menu makanan

Prekondisi	Aktor memilih menu riwayat
Prakondisi	Aktor berhasil melihat laporan rekomendasi menu makanan
Flow of Events	
Skenario Normal : Melihat Laporan Rekomendasi Menu Makanan	
2. Klik menu riwayat	
	<p>3. Menampilkan halaman riwayat yang berisi</p> <ul style="list-style-type: none"> • Tabel yang berisi atribut: <ul style="list-style-type: none"> - Id rekap - Tanggal - Jumlah Kalori - Hasil Rekomendasi - Kalori Makanan

7. Mengelola Data Aktivitas

Tabel 7 Skenario Mengelola Data Aktivitas

Nomor Usecase	USC-07
Nama	Mengelola Data Aktivitas
Aktor	Admin
Deskripsi Singkat	Aktor mengelola data aktivitas
Prekondisi	Aktor memilih menu aktivitas
Prakondisi	Aktor berhasil melihat, menambah, dan mengubah data aktivitas
Flow of Events	
Skenario Normal : Menambah Data Aktivitas	
5. Klik menu aktivitas	
	<p>6. Menampilkan halaman tabel daftar aktivitas</p> <ul style="list-style-type: none"> • Tabel data makanan yang berisi: <ul style="list-style-type: none"> - Id - Nama aktivitas - Faktor aktivitas - Keterangan - Status - Aksi berupa tombol <i>edit</i> dan tombol tolak

	<ul style="list-style-type: none"> • Tombol Tambah Aktivitas
7. Klik tombol tambah aktivitas	
	<p>8. Menampilkan form tambah data makanan yang berisi:</p> <ul style="list-style-type: none"> - Nama aktivitas - Faktor aktivitas - Keterangan
8. Mengisi form tambah data makanan, kemudian klik tombol simpan	9. Menyimpan data ke dalam database
<p>Skenario Alternatif : Isian form kosong</p> <p>7. Klik tombol simpan</p> <p>Skenario Normal : Mengubah Data aktivitas</p>	<p>10. Menampilkan halaman menu makanan dengan data baru yang telah ditambahkan ke tabel</p> <p>8. Menampilkan span "<i>Please fill out this field</i>"</p>
5. Klik menu aktivitas	
	<p>6. Menampilkan halaman tabel daftar makanan</p> <ul style="list-style-type: none"> • Tabel data makanan yang berisi: <ul style="list-style-type: none"> - Id - Nama aktivitas - Faktor aktivitas - Keterangan - Status - Aksi berupa tombol <i>edit</i> dan tombol tolak
7. Klik tombol edit pada salah satu data yang akan diubah	<p>8. Menampilkan halaman data aktivitas dengan form data aktivitas yang akan diubah yang berisi :</p> <ul style="list-style-type: none"> - Id aktivitas - Aktivitas - Faktor aktivitas

8. Mengubah data pada form aktivitas, kemudian klik tombol simpan	- Keterangan
	9. Menyimpan perubahan data ke dalam database
	10. Menampilkan halaman aktivitas dengan data yang telah diubah
Skenario Alternatif : Isian form kosong	
7. Klik tombol simpan	
	8. Menampilkan span " <i>Please fill out this field</i> "
Skenario Normal : Menolak Data aktivitas	
1. Klik menu aktivitas	2. Menampilkan halaman tabel daftar makanan <ul style="list-style-type: none"> • Tabel data makanan yang berisi: <ul style="list-style-type: none"> - Id - Nama aktivitas - Faktor aktivitas - Keterangan - Status
4. Klik tombol tolak pada salah satu data aktivitas	3. Aksi berupa tombol <i>edit</i> dan tombol tolak
	5. Maka tombol tolak akan berubah menjadi tombol setuju

8. Menambah Data Aktivitas

Tabel 8 Skenario Menambah Data Aktivitas

Nomor Usecase	USC-08
Nama	Menambah Data Aktivitas
Aktor	Pengguna
Deskripsi Singkat	Aktor menambah data aktivitas
Prekondisi	Aktor memilih menu rekomendasi
Prakondisi	Aktor berhasil menambah data aktivitas

Flow of Events	
Skenario Normal : Menambah Data Aktivitas	
1. Klik menu rekomendasi	
	2. Menampilkan halaman rekomendasi yang berisi: <ul style="list-style-type: none"> • <i>Form</i> Tambah data Personal berupa: <ul style="list-style-type: none"> - Berat badan - Tinggi badan - Aktivitas yang akan dikerjakan - Tombol tambah aktivitas • Tombol analisa untuk perhitungan
3. Klik tombol tambah aktivitas	
	6. Menampilkan form tambah data aktivitas yang berisi: <ul style="list-style-type: none"> - Nama aktivitas - Faktor aktivitas - Keterangan
11. Mengisi form tambah data aktivitas, kemudian klik tombol simpan	
	12. Menyimpan data ke dalam database
	13. Menampilkan halaman menu rekomendasi dengan data baru yang telah ditambahkan ke tabel
Skenario Alternatif : Isian form kosong	
9. Klik tombol simpan	
	10. Menampilkan span " <i>Please fill out this field</i> "

9. Mengelola Data Posisi Badan

Tabel 9 Skenario Mengelola Data Posisi Badan

Nomor Usecase	USC-09
Nama	Mengelola Data Posisi Badan
Aktor	Admin
Deskripsi Singkat	Aktor mengelola data Posisi Badan
Prekondisi	Aktor memilih menu Posisi Badan

Prakondisi	Aktor berhasil melihat, menambah, dan mengubah data Posisi Badan
Flow of Events	
Skenario Normal : Menambah Data Posisi Badan	
9. Klik menu Posisi Badan	
	<p>10. Menampilkan halaman tabel daftar Posisi Badan</p> <ul style="list-style-type: none"> • Tabel data makanan yang berisi: <ul style="list-style-type: none"> - Id - Posisi Badan - Faktor aktivitas - Keterangan - Status - Aksi berupa tombol <i>edit</i> dan tombol non-aktifkan • Tombol Tambah Posisi Badan
11. Klik tombol tambah Posisi Badan	
	<p>12. Menampilkan form tambah data Posisi Badan yang berisi:</p> <ul style="list-style-type: none"> - Posisi Badan - Faktor aktivitas - Keterangan
14. Mengisi form tambah data Posisi Badan, kemudian klik tombol simpan	
	15. Menyimpan data ke dalam database
	16. Menampilkan halaman menu Posisi Badan dengan data baru yang telah ditambahkan ke tabel
Skenario Alternatif : Isian form kosong	
11. Klik tombol simpan	12. Menampilkan span " <i>Please fill out this field</i> "
Skenario Normal : Mengubah Data Posisi Badan	
9. Klik menu Posisi Badan	
	<p>10. Menampilkan halaman tabel daftar Posisi Badan</p> <ul style="list-style-type: none"> • Tabel data Posisi Badan yang berisi: <ul style="list-style-type: none"> - Id Posisi Badan

	<ul style="list-style-type: none"> - Posisi Badan - Faktor aktivitas - Keterangan - Aksi berupa tombol <i>edit</i> dan tombol non-aktifkan
11. Klik tombol edit pada salah satu data yang akan diubah	
	<p>12. Menampilkan halaman data aktivitas dengan form data aktivitas yang akan diubah yang berisi :</p> <ul style="list-style-type: none"> - Id aktivitas - Aktivitas - Faktor aktivitas - Keterangan
11. Mengubah data pada form posisi badan, kemudian klik tombol simpan	
	12. Menyimpan perubahan data ke dalam database
	13. Menampilkan halaman posisi badan dengan data yang telah diubah
Skenario Alternatif : Isian form kosong	
9. Klik tombol simpan	
	10. Menampilkan span " <i>Please fill out this field</i> "
Skenario Normal : Mengaktifkan Data Posisi Badan	
7. Klik menu posisi badan	
	<p>13. Menampilkan halaman tabel daftar Posisi Badan</p> <ul style="list-style-type: none"> • Tabel data Posisi Badan yang berisi: <ul style="list-style-type: none"> - Id Posisi Badan - Posisi Badan - Faktor aktivitas - Keterangan - Aksi berupa tombol <i>edit</i> dan tombol non-aktifkan
8. Klik tombol aktif pada salah satu data posisi badan	
	9. Maka tombol non-aktif akan berubah menjadi tombol aktif

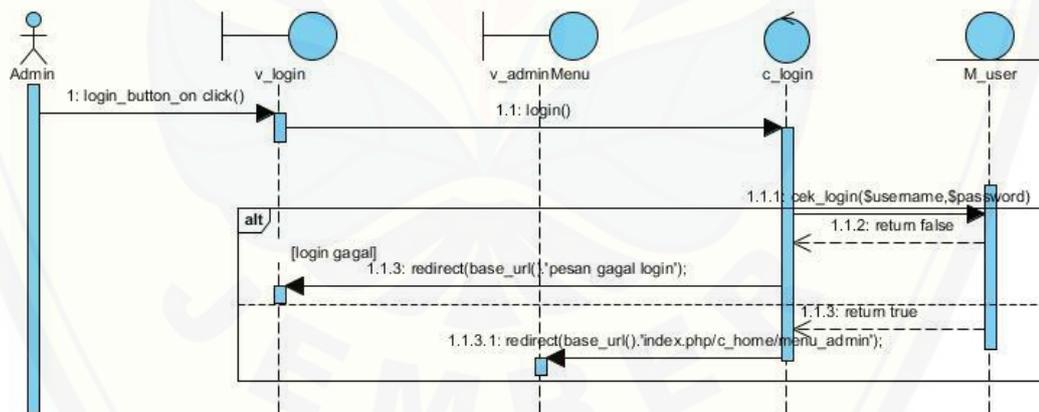
10. Skenario Keluar

Tabel 10 Skenario Keluar

Nomor Usecase	USC-10
Nama	Keluar
Aktor	Admin / Pengguna
Deskripsi Singkat	Aktor keluar dari sistem informasi
Prekondisi	Aktor menggunakan sistem informasi
Prakondisi	Aktor keluar dari sistem informasi
Flow of Events	
Skenario Normal : Keluar	
1. Klik tombol Keluar	2. Menampilkan halaman Masuk

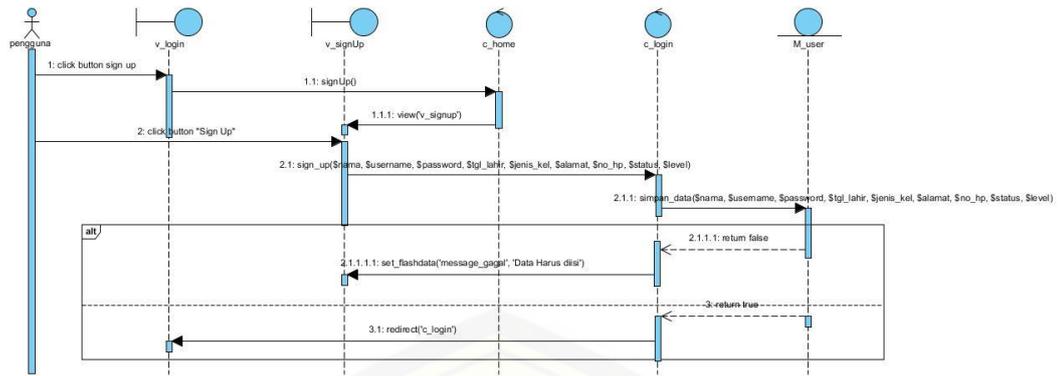
LAMPIRAN B. SEQUENCE DIAGRAM

1. Sequence Diagram Masuk



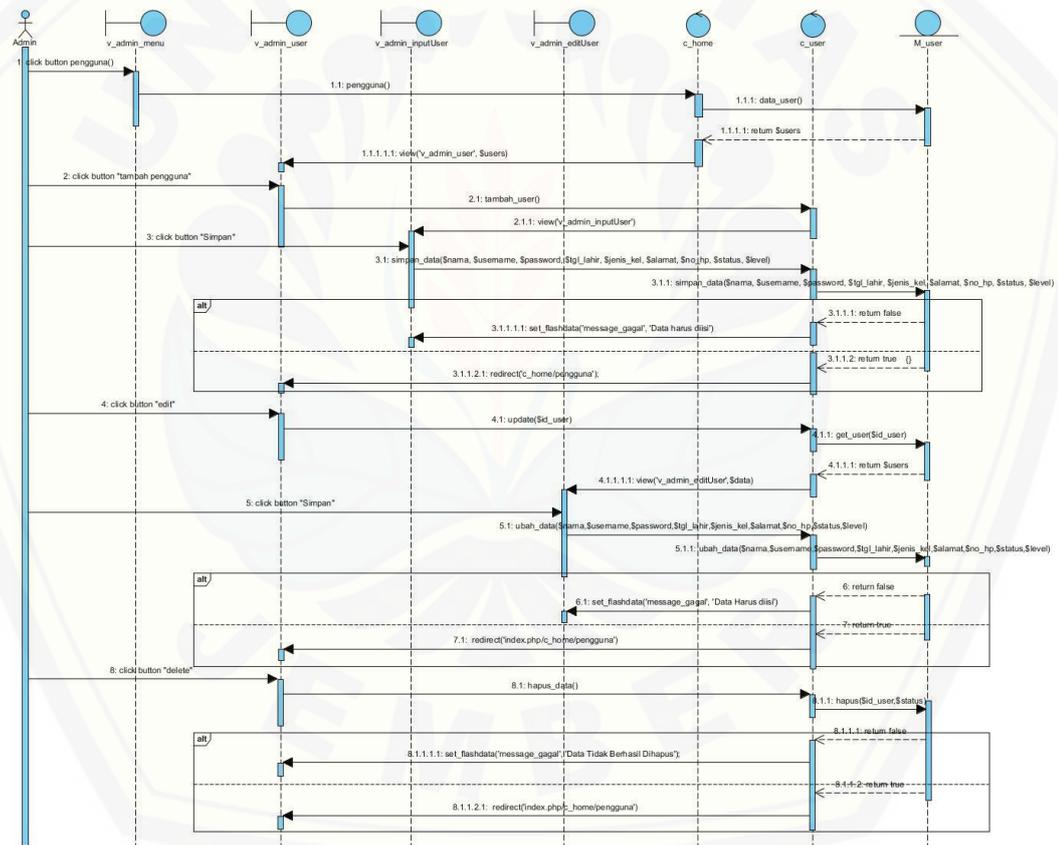
Gambar 1 Sequence Diagram Masuk

2. Sequence Diagram Menambah Data Pengguna Baru



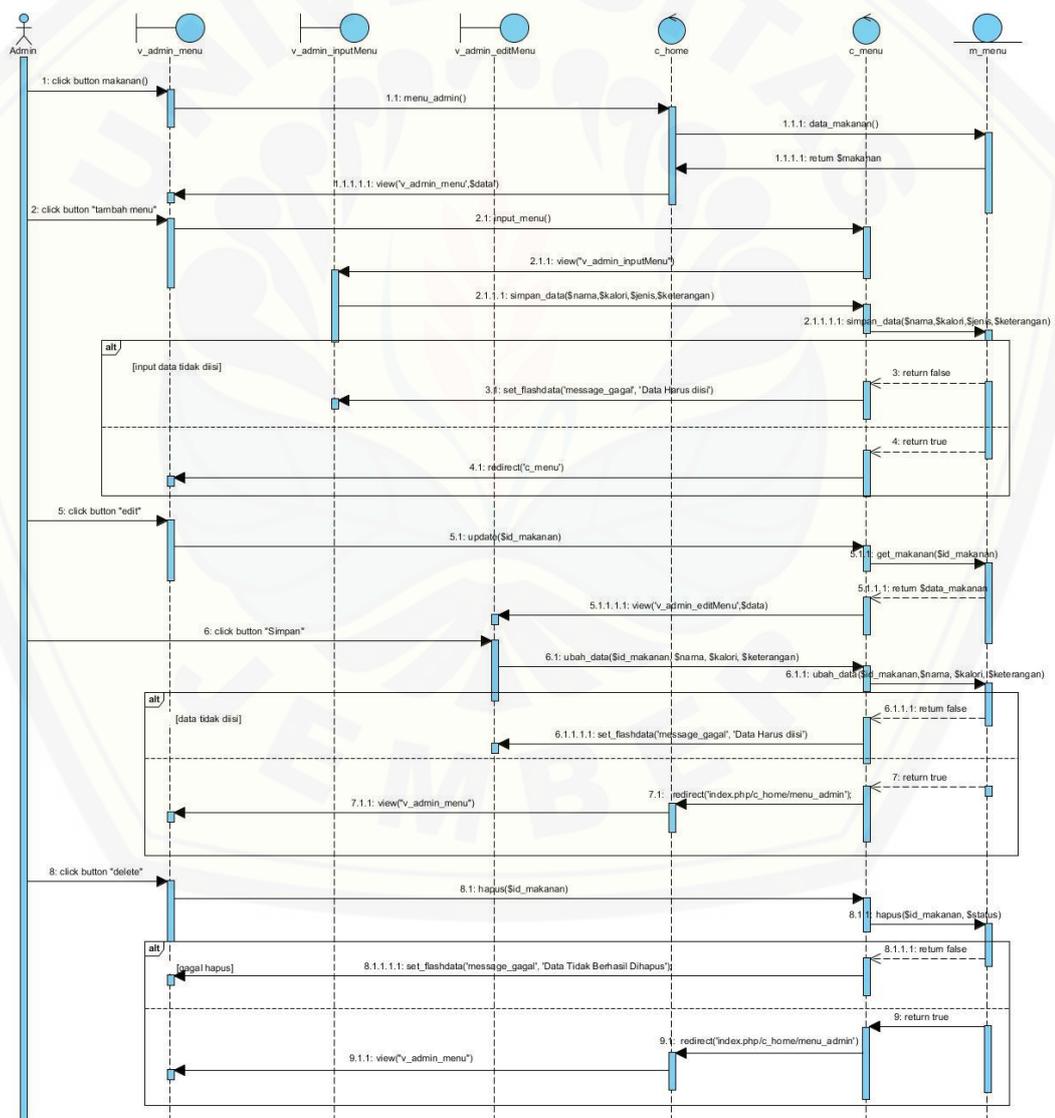
Gambar 2 Sequence Diagram Menambah Pengguna Baru

3. Sequence Diagram Mengelola Data Pengguna



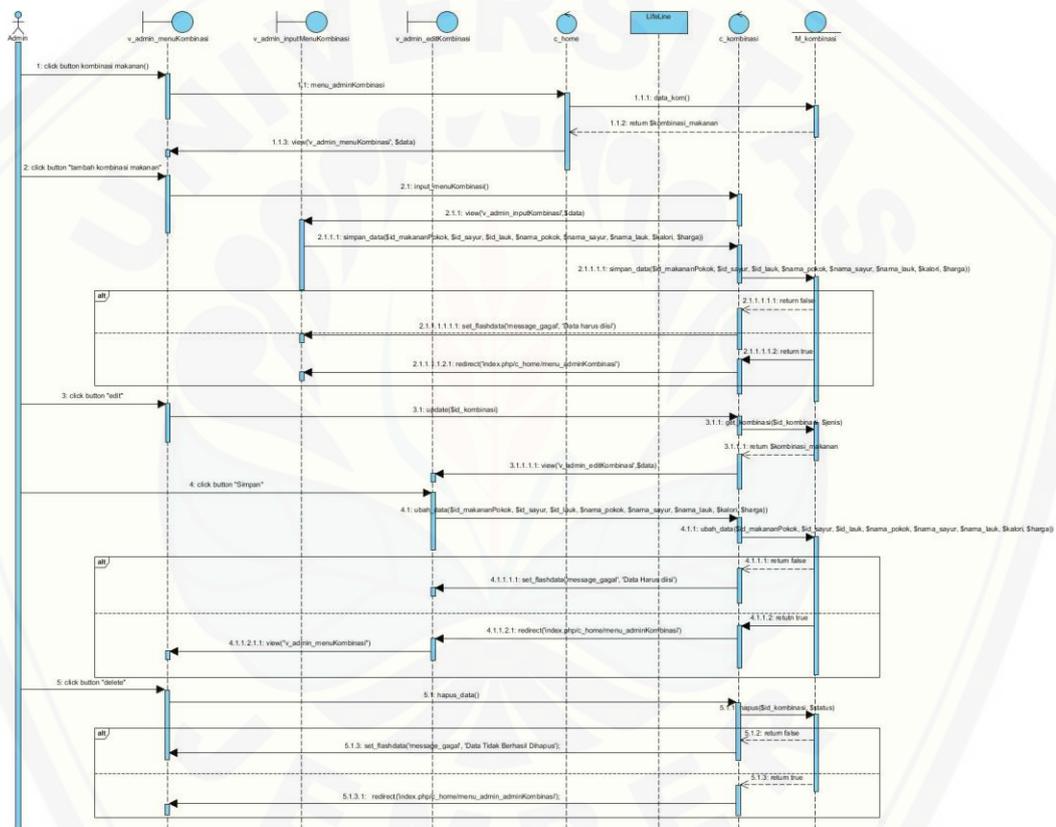
Gambar 3 Sequence Diagram Mengelola Data Pengguna

4. Sequence Diagram Mengelola Data Makanan



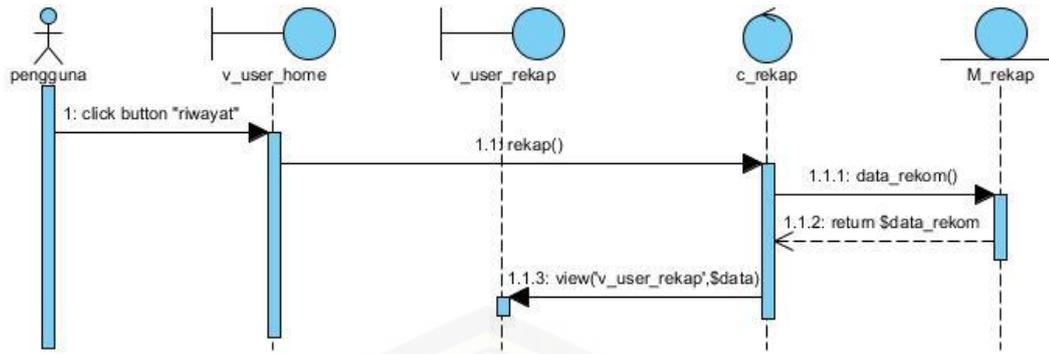
Gambar 4 Sequence Diagram Mengelola Data Makanan

5. Sequence Diagram Mengelola Data Kombinasi Makanan



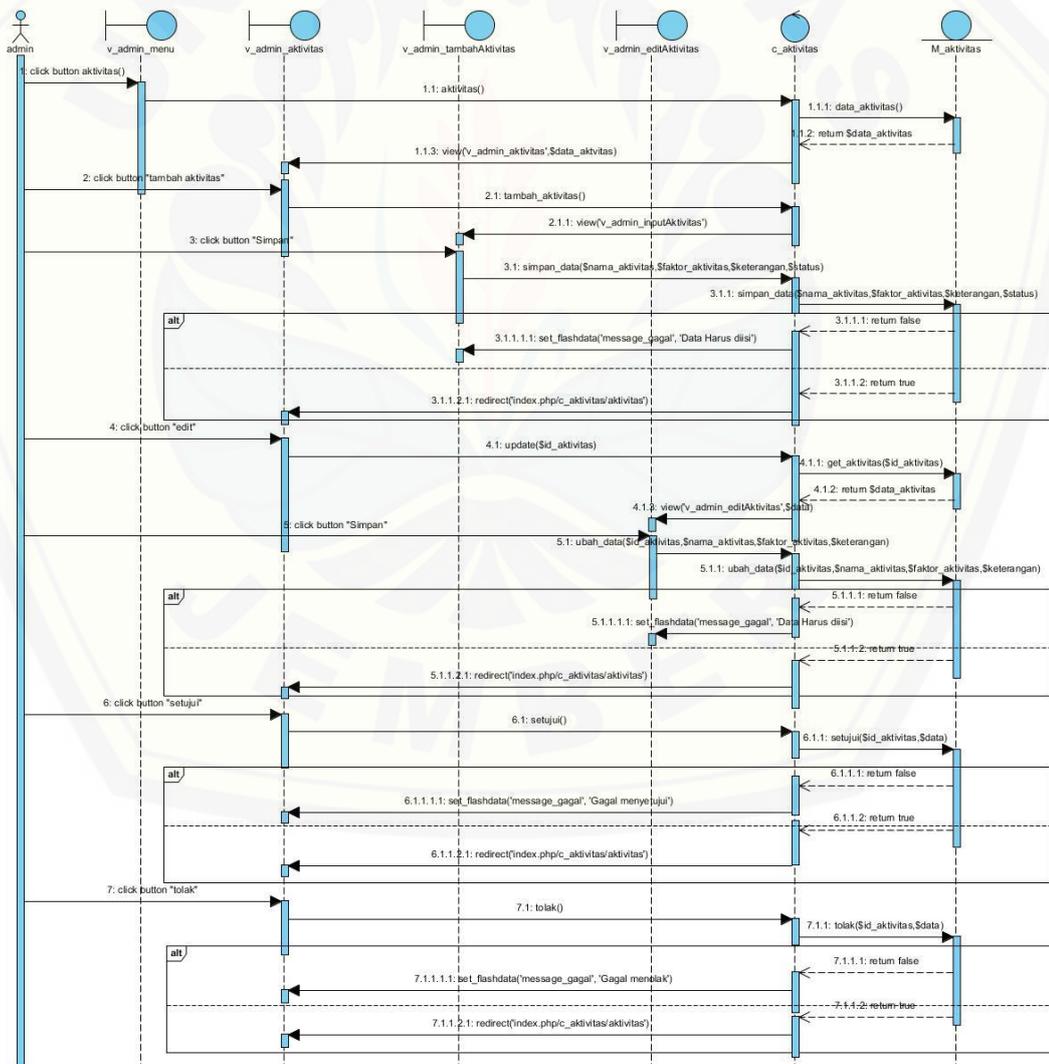
Gambar 5 Sequence Diagram Mengelola Kombinasi Makanan

6. Sequence Diagram Melihat Laporan Rekomendasi Menu Makanan



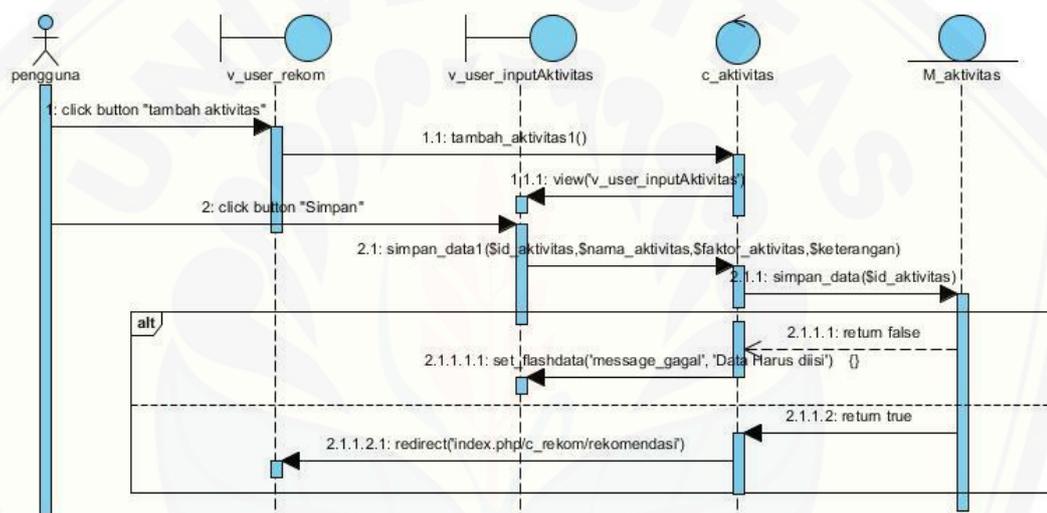
Gambar 6 Melihat Laporan Rekomendasi Menu Makanan

7. Sequence Diagram mengelola Data Aktivitas



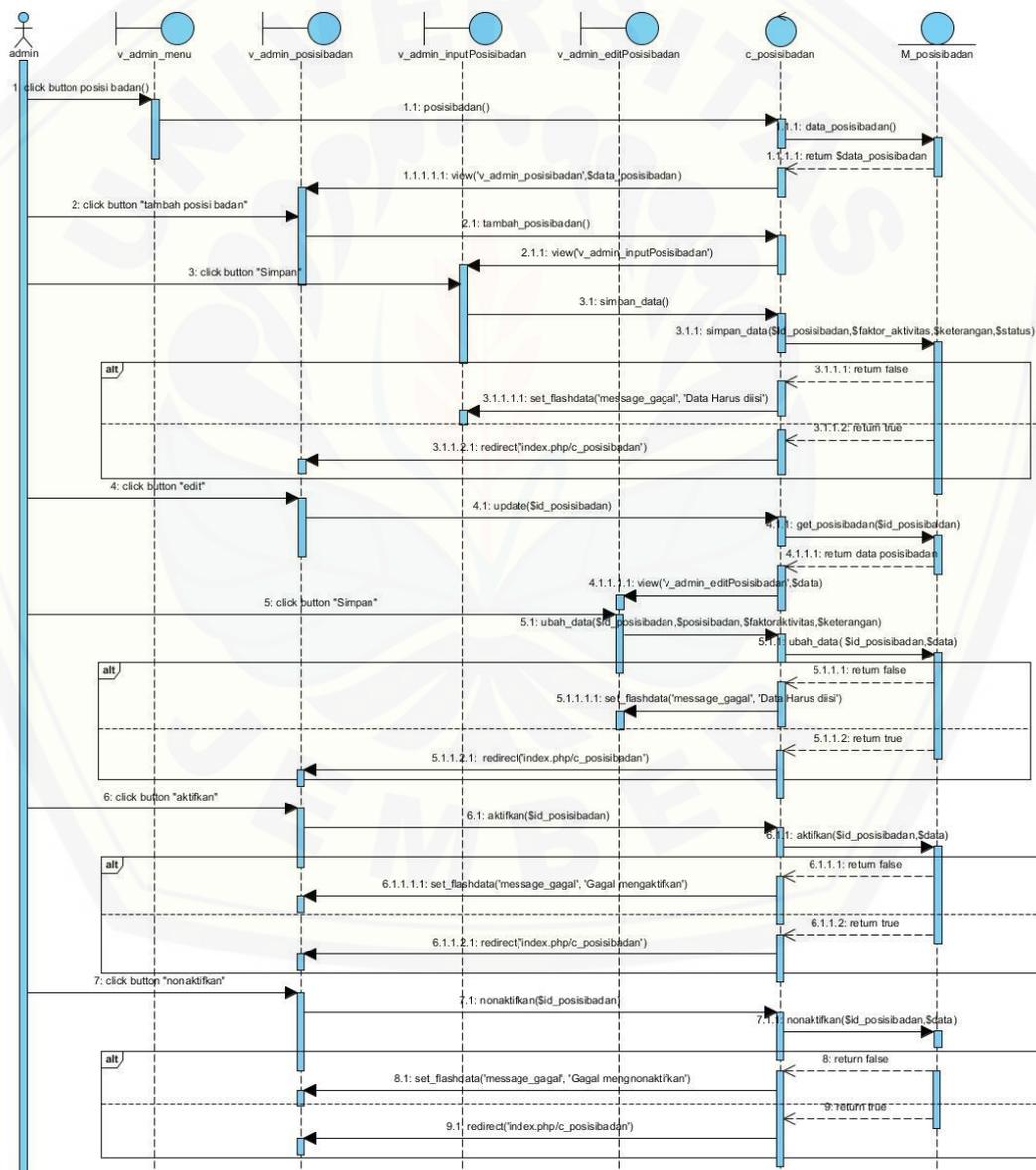
Gambar 7 Sequence Diagram mengelola Data Aktivitas

8. Sequence Diagram Menambah Data Aktivitas



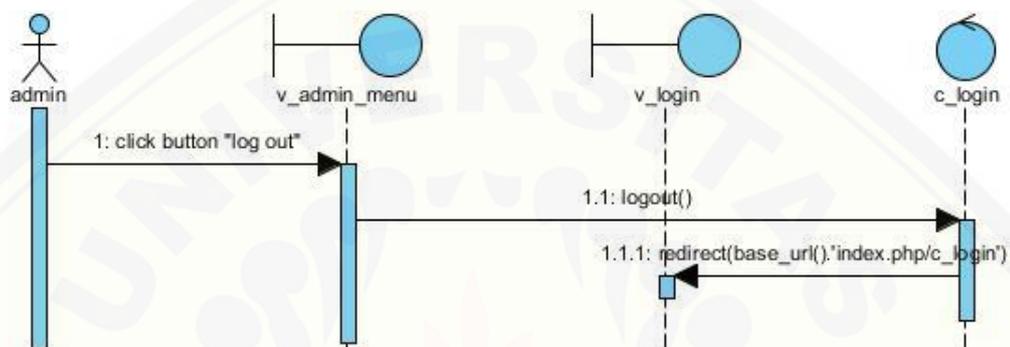
Gambar 8 Sequence Diagram Menambah Data Aktivitas

9. Sequence Diagram Mengelola Data Posisi Badan



Gambar 9 Sequence Diagram Mengelola Data Posisi Badan

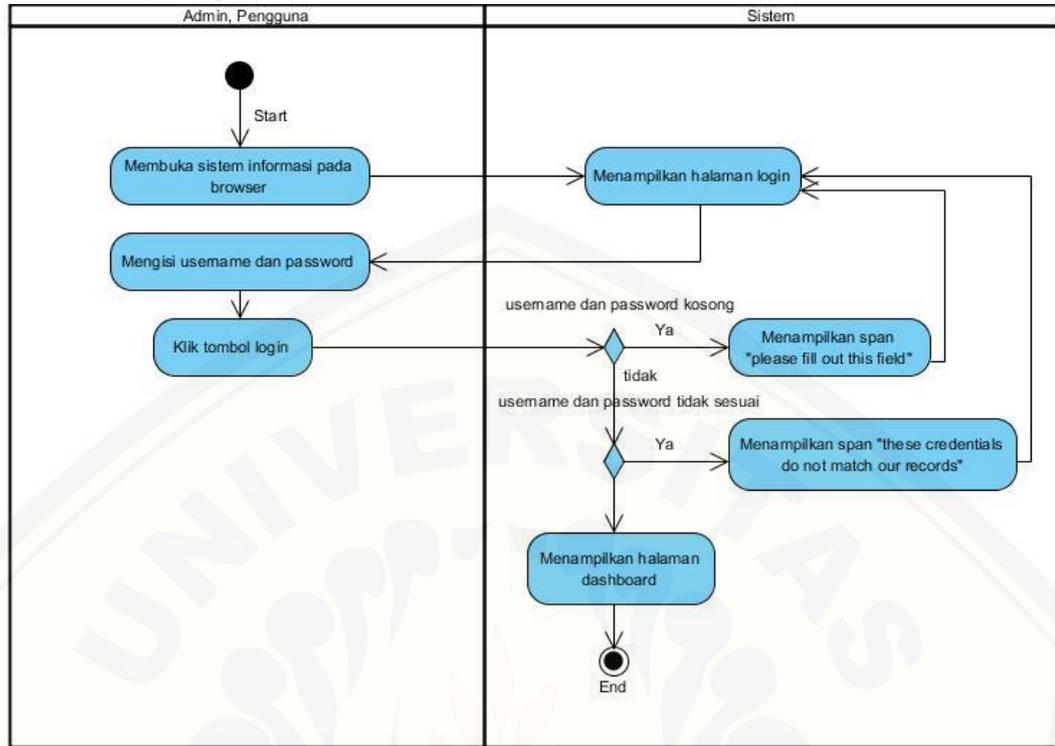
10. Sequence Diagram Keluar



Gambar 10 Sequence Diagram Keluar

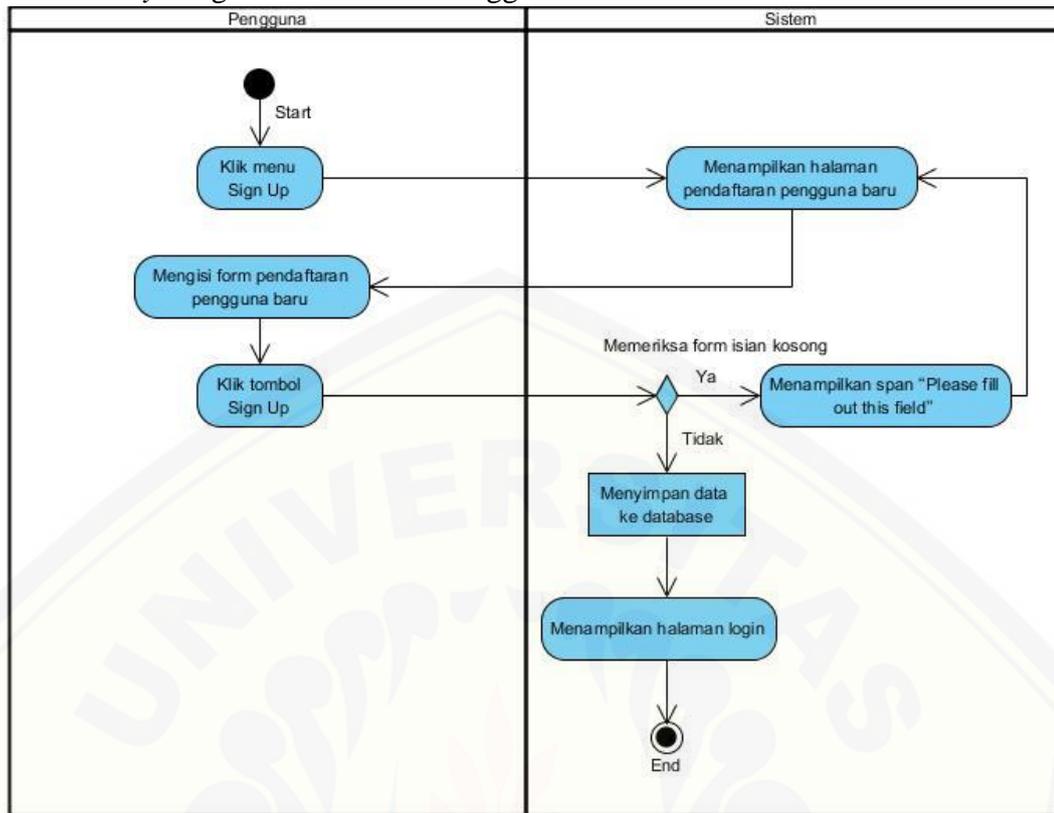
LAMPIRAN C. ACTIVITY DIAGRAM

1. Activity Diagram Masuk



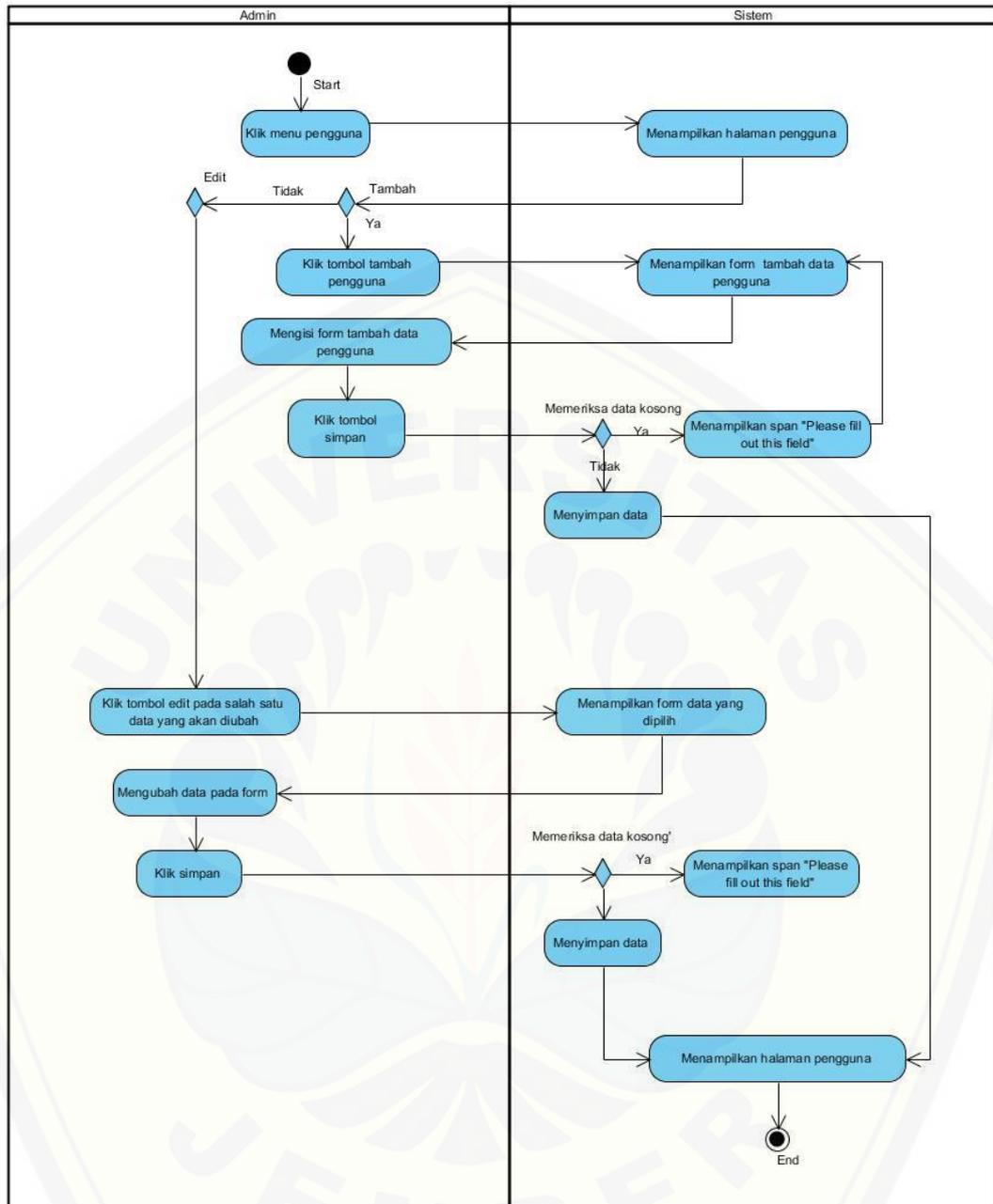
Gambar 11 Activity Diagram Masuk

2. Activity Diagram Menambah Pengguna Baru



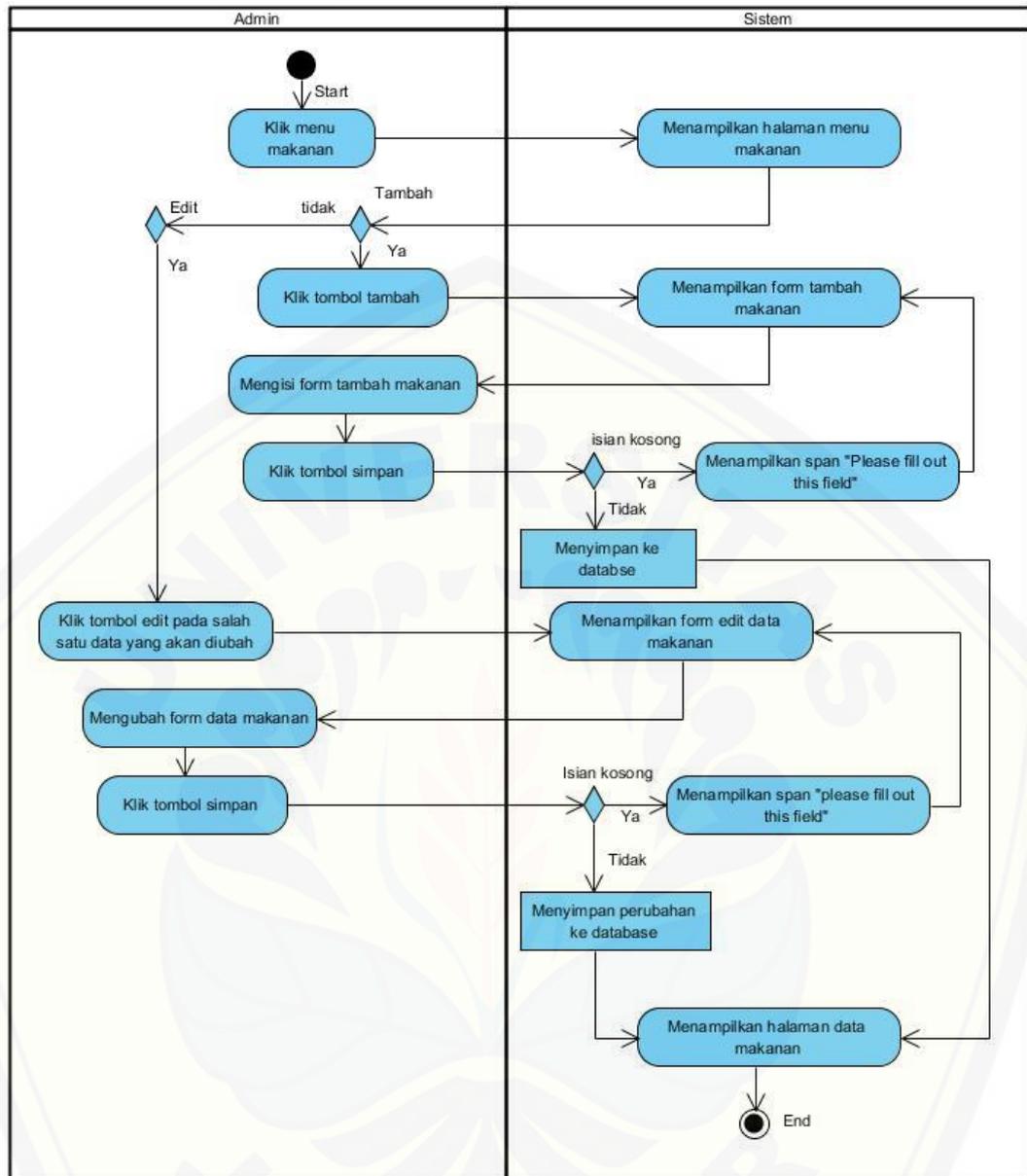
Gambar 12 Activity Diagram Menambah Pengguna Baru

3. Activity Diagram Mengelola Data Pengguna



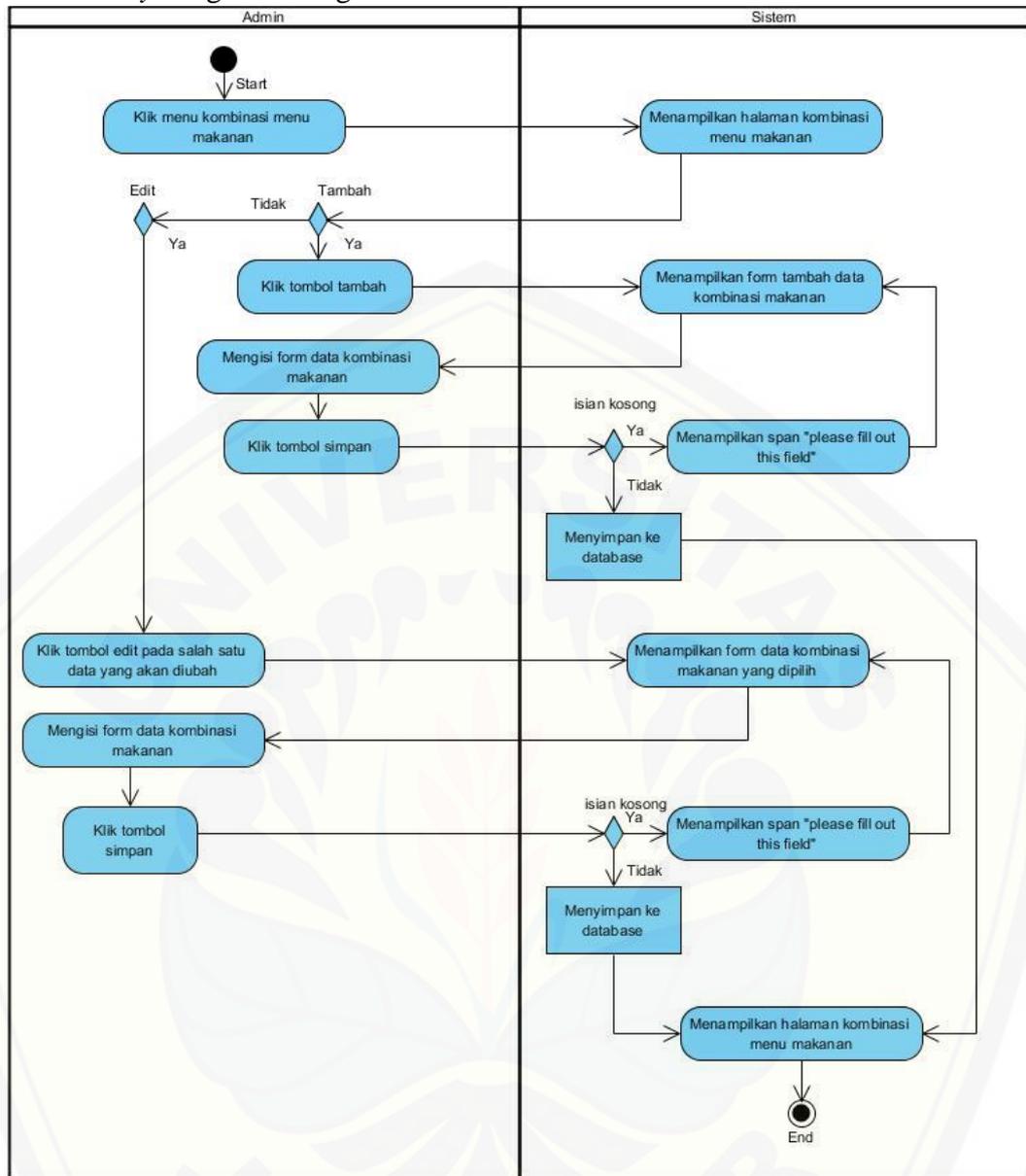
Gambar 13 Activity Diagram Mengelola Data Pengguna

4. Activity Diagram Mengelola Data Makanan



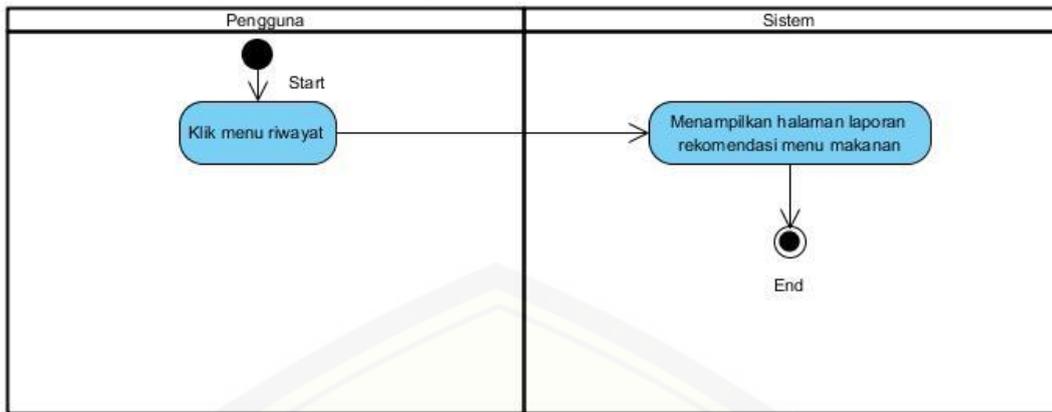
Gambar 14 Activity Diagram Mengelola Data Makanan

5. Activity Diagram Mengelola Data Kombinasi Makanan



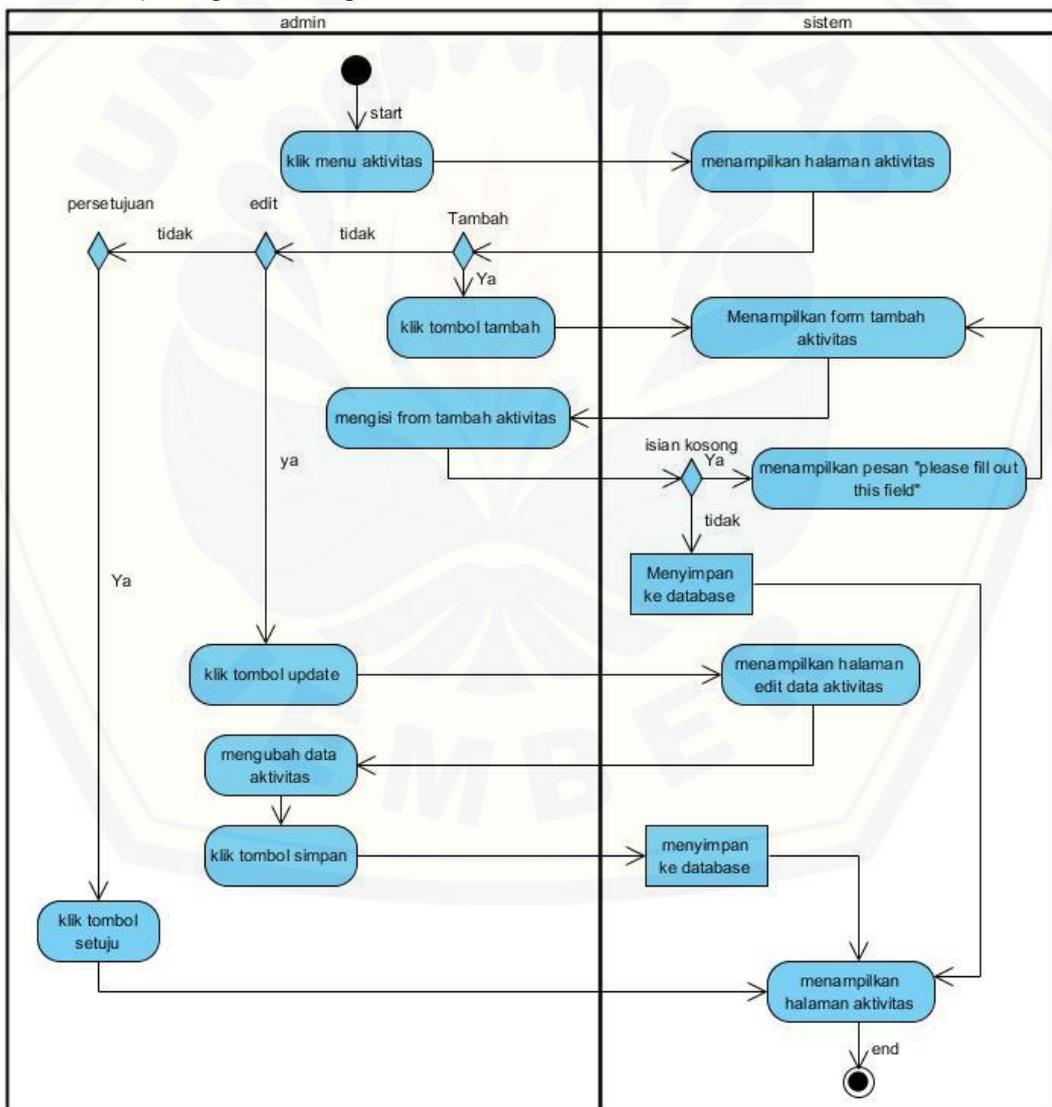
Gambar 15 Activity Diagram Mengelola Data Kombinasi Makanan

6. Activity Diagram Melihat Laporan Rekomendasi Menu Makanan



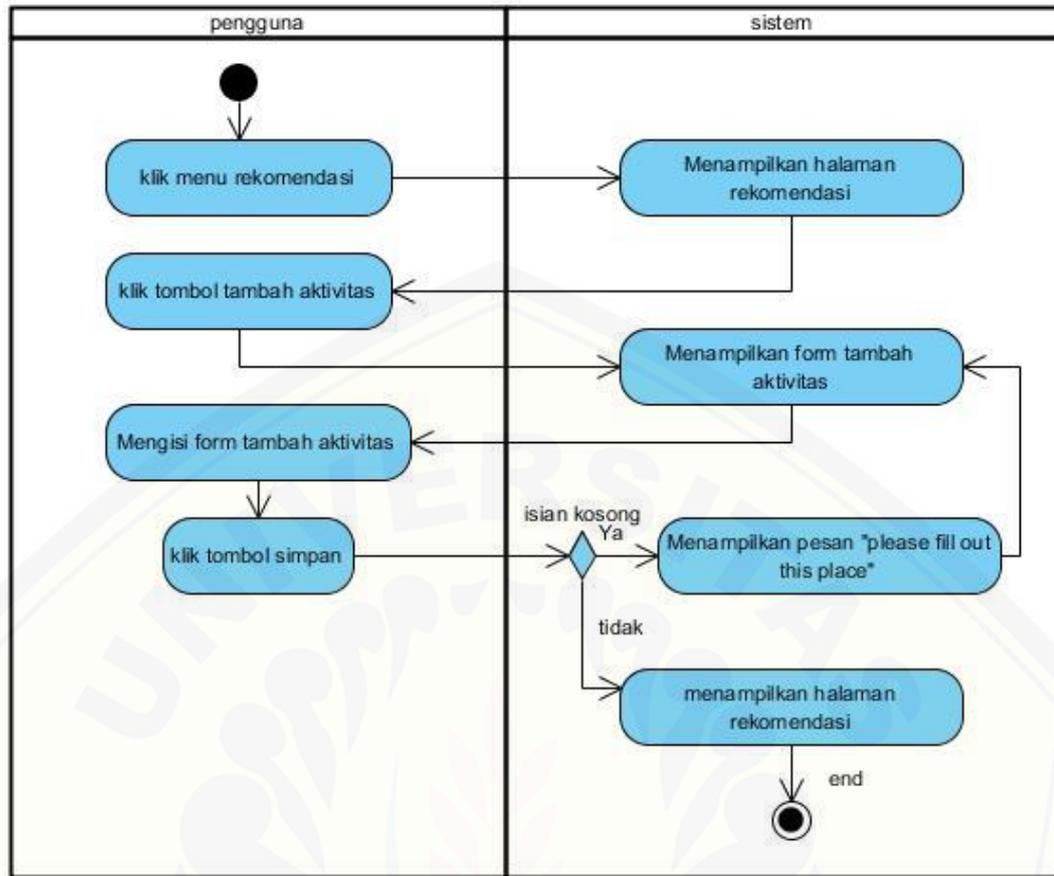
Gambar 16 Activity Diagram Melihat Laporan Rekomendasi Menu Makanan

7. Activity Diagram Mengelola Data Aktivitas



Gambar 17 Activity Diagram Mengelola Data Aktivitas

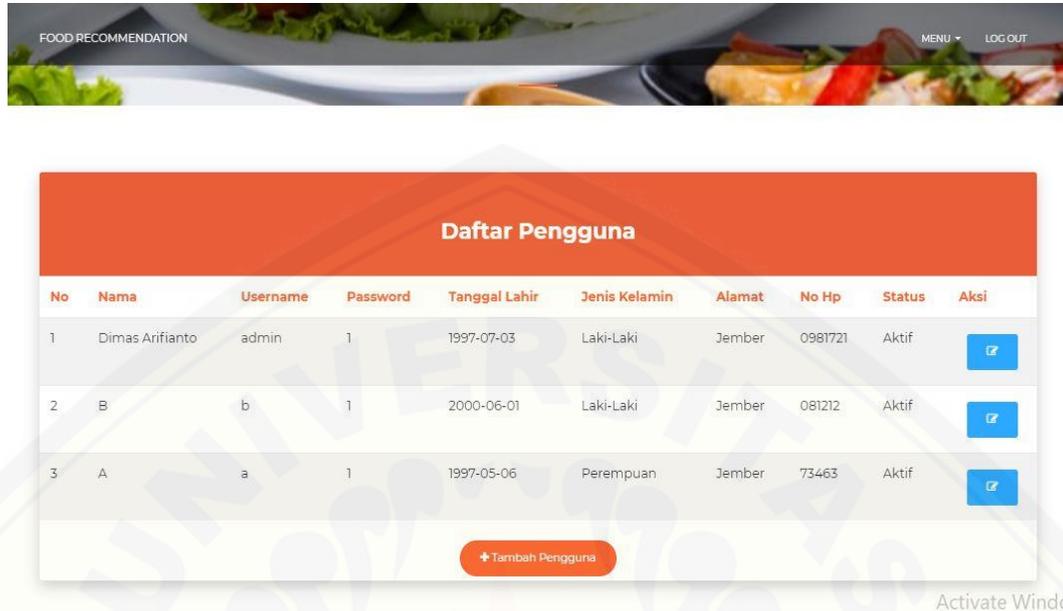
8. Activity Diagram Menambah Data Aktivitas



Gambar 18 Activity Diagram Menambah Data Aktivitas

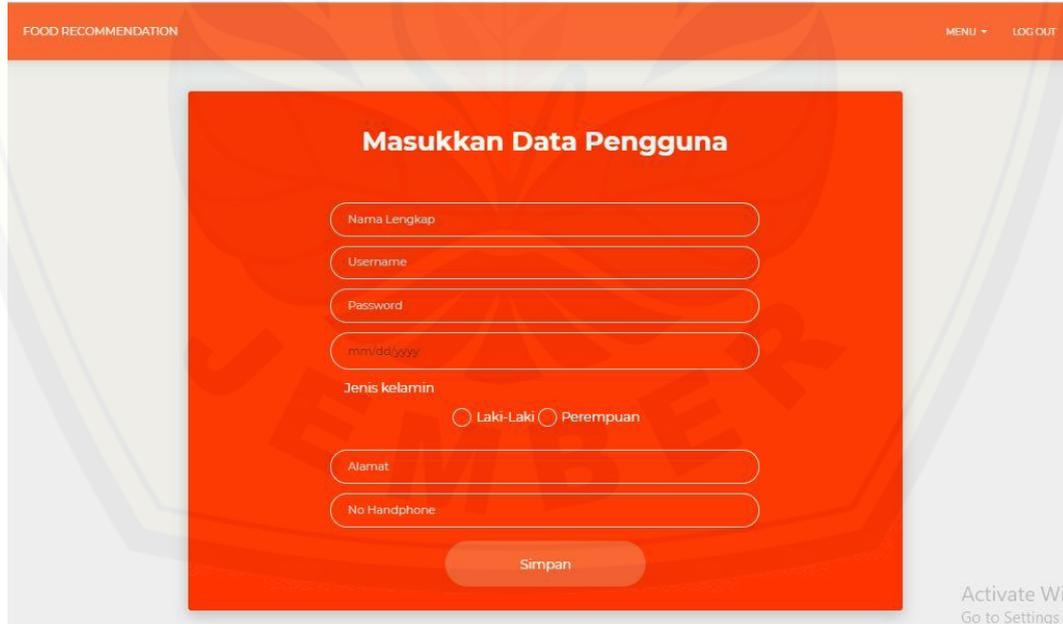
LAMPIRAN D. TAMPILAN SISTEM

1. Tampilan Melihat Data Pengguna



Gambar 21 Tampilan Melihat Data Pengguna

2. Tampilan Menambah Data Pengguna



Gambar 22 Tampilan Menambah Data Pengguna

3. Tampilan Mengubah Data Pengguna

Ubah Data Pengguna

MENU LOG OUT

Id User
1

Nama Lengkap
Dimas Arifianto

Username
admin

Password
1

Tanggal Lahir
07/03/1997

Jenis kelamin
 Laki-Laki Perempuan

Alamat
Jember

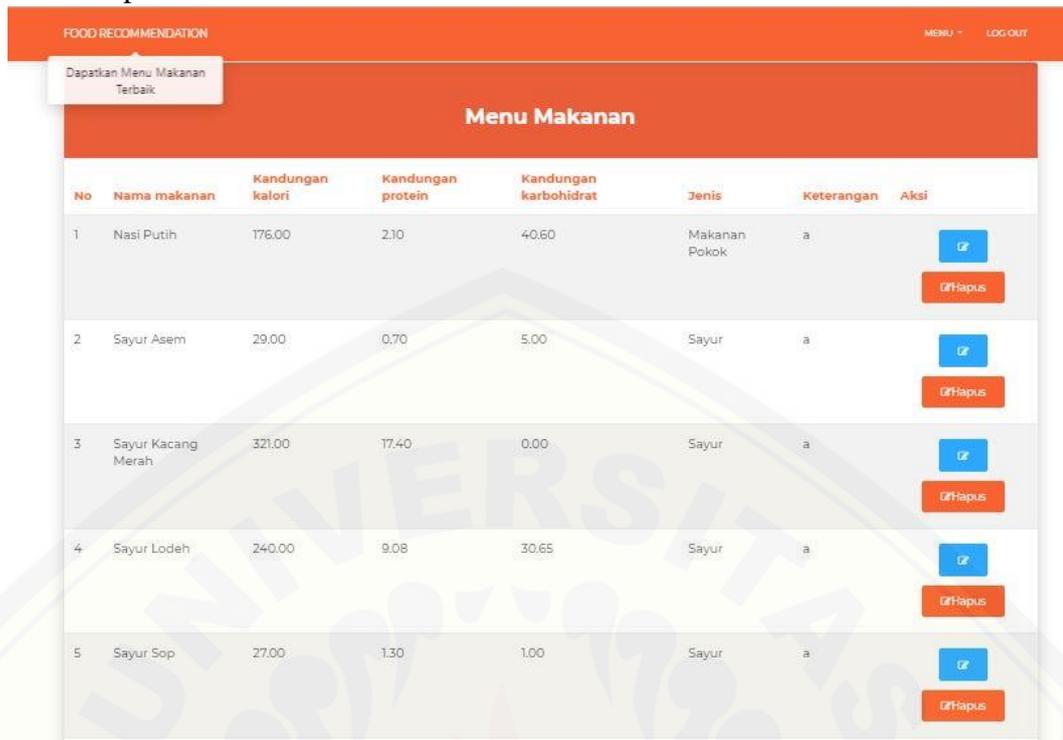
No HP
0981721

Status
 Aktif Non Aktif

Simpan

Gambar 23 Tampilan Mengubah Data Pengguna

4. Tampilan Melihat Data Makanan



FOOD RECOMMENDATION MENU LOG OUT

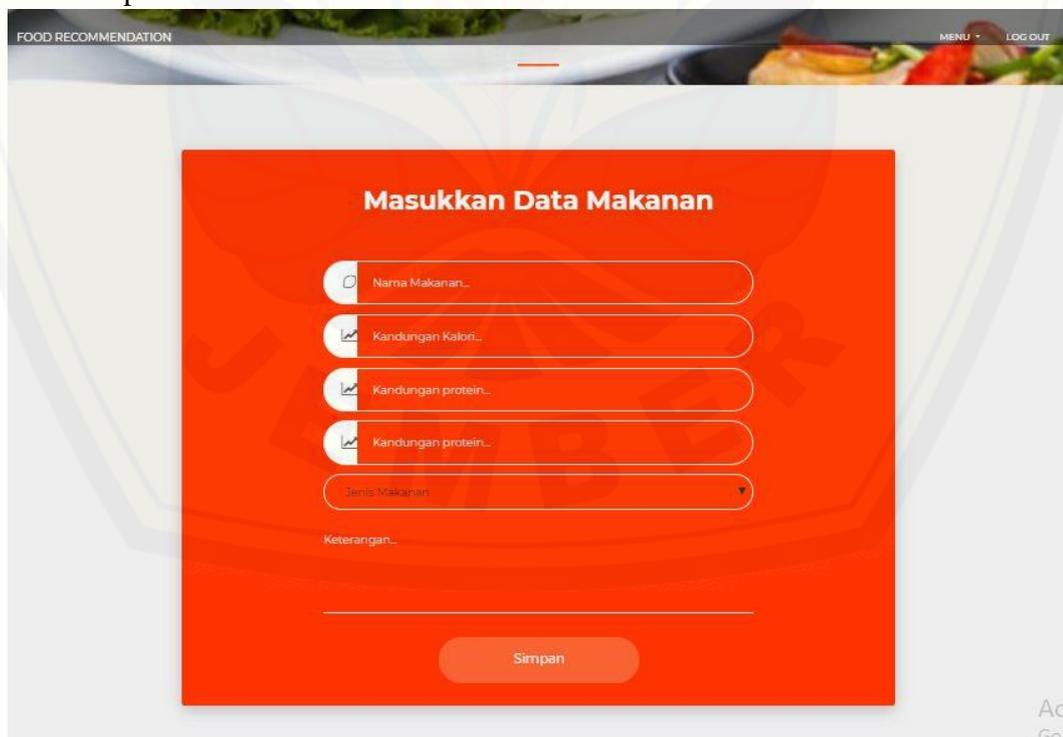
Dapatkan Menu Makanan Terbaik

Menu Makanan

No	Nama makanan	Kandungan kalori	Kandungan protein	Kandungan karbohidrat	Jenis	Keterangan	Aksi
1	Nasi Putih	176.00	2.10	40.60	Makanan Pokok	a	Edit Hapus
2	Sayur Asem	29.00	0.70	5.00	Sayur	a	Edit Hapus
3	Sayur Kacang Merah	321.00	17.40	0.00	Sayur	a	Edit Hapus
4	Sayur Lodeh	240.00	9.08	30.65	Sayur	a	Edit Hapus
5	Sayur Sop	27.00	1.30	1.00	Sayur	a	Edit Hapus

Gambar 24 Tampilan Melihat Data Makanan

5. Tampilan Menambah Data Makanan



FOOD RECOMMENDATION MENU LOG OUT

Masukkan Data Makanan

Gambar 25 Tampilan Menambah Data Makanan

6. Tampilan Mengubah Data Makanan

FOOD RECOMMENDATION MENU LOG OUT

Ubah Data Makanan

ID Makanan:

Nama Makanan:

Kandungan Kalori:

Kandungan Protein:

Kandungan Karbohidrat:

Jenis Makanan:

Keterangan:

Simpan

Gambar 26 Tampilan Mengubah Data Makanan

7. Tampilan Melihat Data Kombinasi Makanan

FOOD RECOMMENDATION MENU LOG OUT

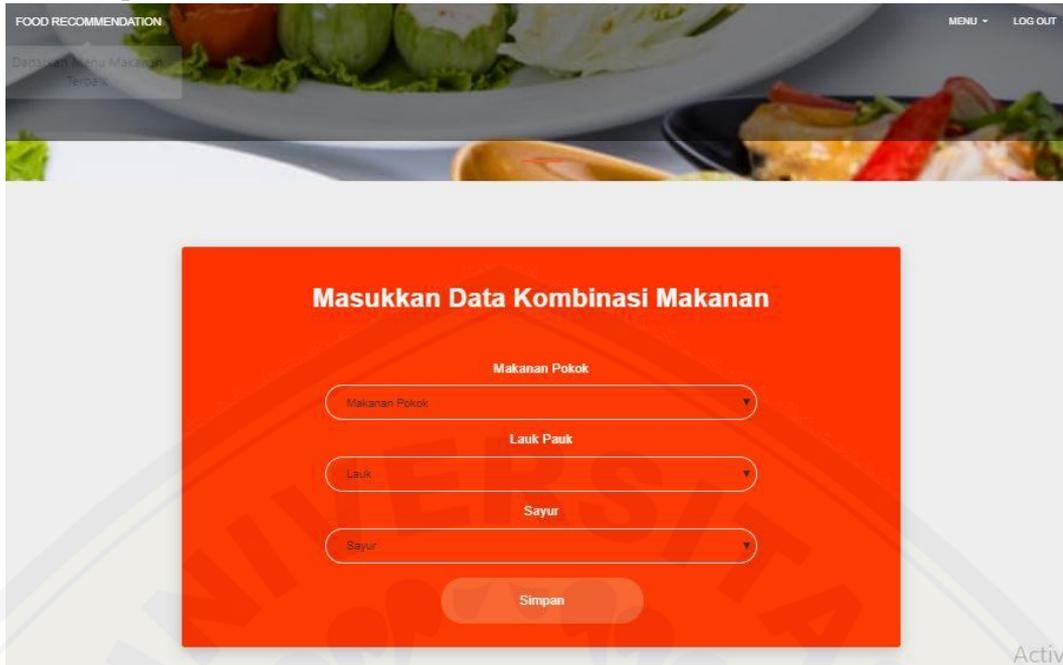
Kombinasi Makanan

No	Menu Makanan	Total Kalori	Total Protein	Total Karbohidrat	Aksi
1	Nasi Putih, Mujair Goreng, Sayur Asem	621.00	49.70	45.60	<input type="button" value="✓"/>
2	Nasi Putih, Ikan Tongkol, Sayur Asem	322.00	26.00	45.60	<input type="button" value="✓"/>
3	Nasi Putih, Ikan Mas Goreng, Sayur Asem	387.00	20.90	45.60	<input type="button" value="✓"/>
4	Nasi Putih, Telur Ayam Dadar, Sayur Asem	456.00	19.10	47.00	<input type="button" value="✓"/>
5	Nasi Putih, Tahu Goreng, Sayur Asem	333.00	8.40	46.80	<input type="button" value="✓"/>
6	Nasi Putih, Tempe Goreng, Sayur Asem	261.00	6.72	47.20	<input type="button" value="✓"/>
7	Nasi Putih, Bandeng Goreng, Sayur Asem	334.00	22.80	45.60	<input type="button" value="✓"/>
8	Nasi Putih, Bawal Goreng, Sayur Asem	301.00	21.80	45.60	<input type="button" value="✓"/>

Acti
Go to

Gambar 27 Tampilan Melihat Data Kombinasi Makanan

8. Tampilan Menambah Data Kombinasi Makanan



FOOD RECOMMENDATION MENU LOG OUT

Dapatkan Menu Makanan Terbaik

Masukkan Data Kombinasi Makanan

Makanan Pokok

Makanan Pokok

Lauk Pauk

Lauk

Sayur

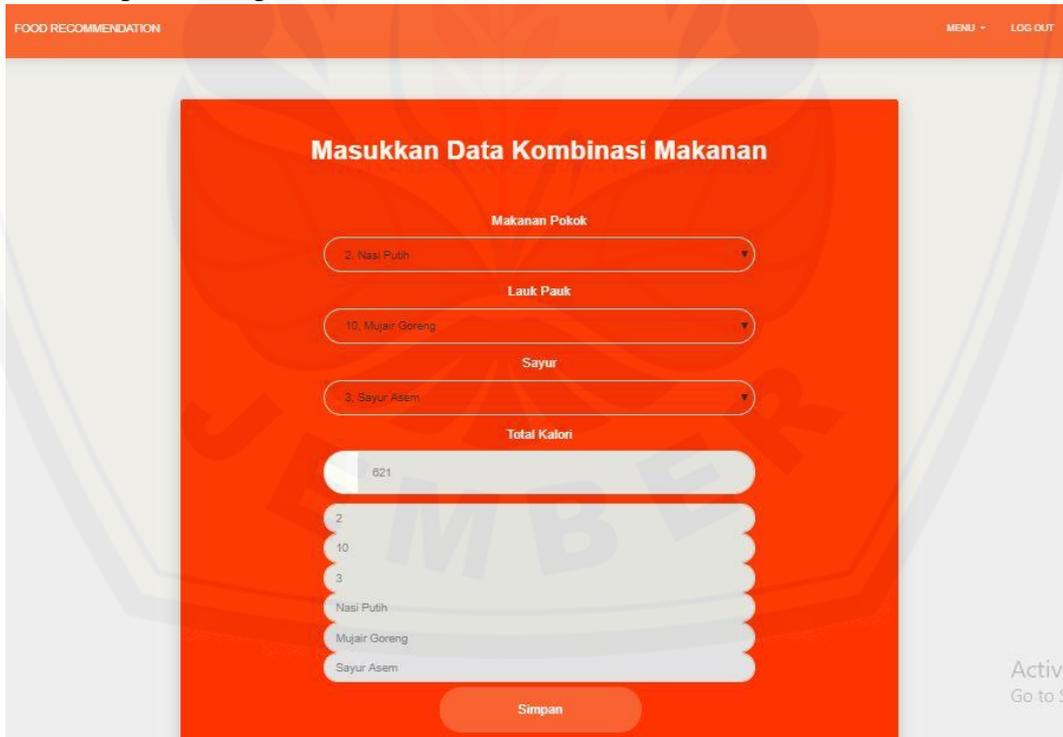
Sayur

Simpan

Active

Gambar 28 Tampilan Menambah Data Kombinasi Makanan

9. Tampilan Mengubah Data Kombinasi Makanan



FOOD RECOMMENDATION MENU LOG OUT

Masukkan Data Kombinasi Makanan

Makanan Pokok

2 Nasi Putih

Lauk Pauk

10 Mujair Goreng

Sayur

3 Sayur Asem

Total Kalori

621

2
10
3

Nasi Putih
Mujair Goreng
Sayur Asem

Simpan

Active
Go to S

Gambar 29 Tampilan Mengubah Data Kombinasi Makanan

10. Tampilan Melihat Laporan Rekomendasi Menu Makanan

11. Tampilan Melihat Data Aktivitas

No	Nama Aktivitas	Faktor Aktivitas	Keterangan	Status	Aksi
1	menulis	0.3	aktifasi	Valid	Update Tolak
2	menyetrika	0.7	b	Valid	Update Tolak
3	merajut	0.3	c	Valid	Update Tolak
4	mengetik	1.1	s	Valid	Update Tolak
5	menyapu lantai	0.9	w	Valid	Update Tolak
6	menggergaji	1.6	a	Valid	Update Tolak
7	memukul paku	1.1	w	Valid	Update Tolak
8	mengemas barang	1.25	w	Valid	Update Tolak

Gambar 30 Tampilan Melihat Data Aktivitas

12. Tampilan Menambah Data Aktivitas

Masukkan Data Aktivitas

Nama Aktivitas

Faktor Aktivitas

Deskripsi Aktivitas

Simpan

Gambar 31 Tampilan Menambah Data Aktivitas

13. Tampilan Mengubah Data Aktivitas

FOOD RECOMMENDATION MENU - LOG OUT

Daftar Menu Makanan Terbaik

Ubah Data Aktivitas

Id Aktivitas: 2

Aktivitas: menulis

Faktor Aktivitas: 0.5

Keterangan: aktifitas

Simpan

Gambar 32 Tampilan Mengubah Data Aktivitas

14. Tampilan Melihat Data Posisi Badan

FOOD RECOMMENDATION MENU - LOG OUT

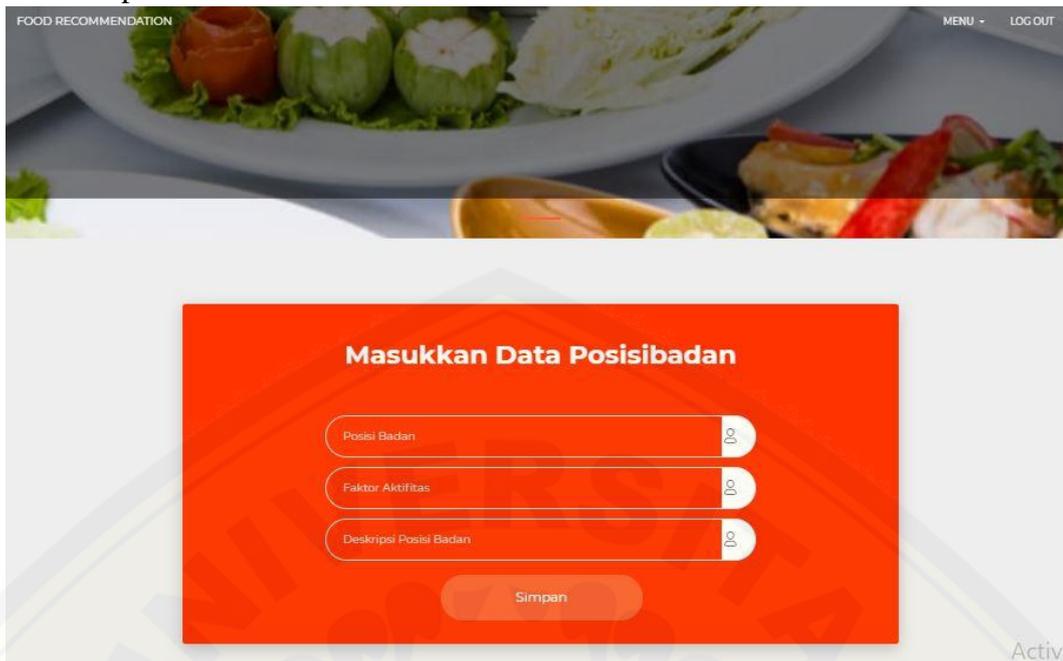
Daftar Posisi Badan

No	Posisi Badan	Faktor Aktivitas	Keterangan	Status	Aksi
1	duduk	0.3	posisi badan a	Aktif	Update Non Aktifkan
2	berdiri	0.6	a	Aktif	Update Non Aktifkan
3	berjalan	3	a	Aktif	Update Non Aktifkan
4	berjalan mendaki	3.8	a	Aktif	Update Non Aktifkan

+ Tambah Posisi Badan

Gambar 33 Tampilan Melihat Data Posisi Badan

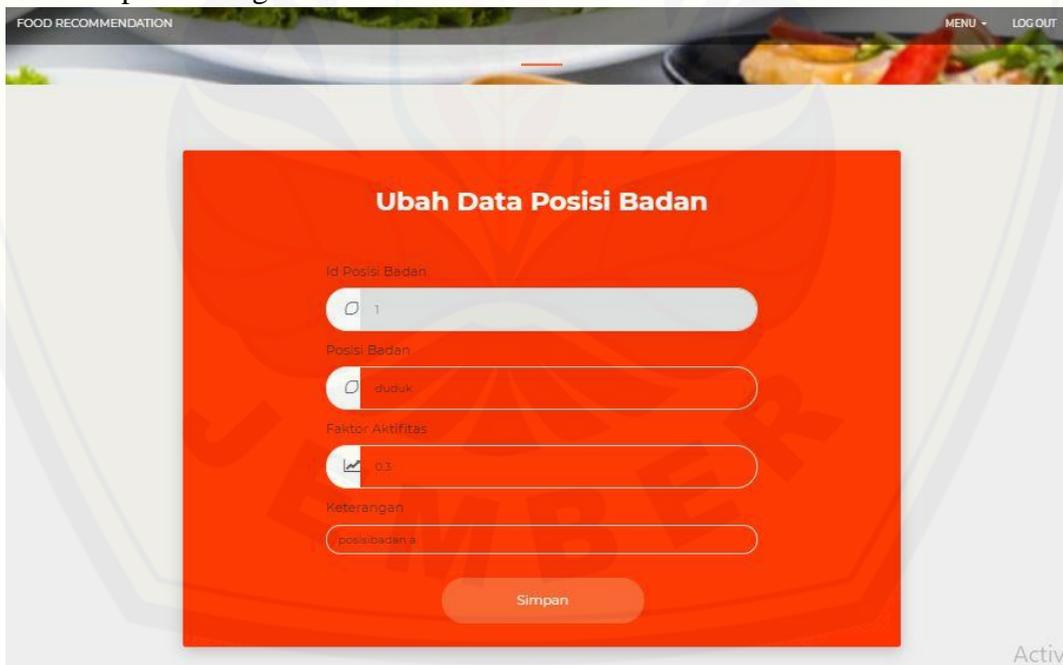
15. Tampilan Menambah Data Posisi Badan



The screenshot shows a web application interface for adding body position data. At the top, there is a header with 'FOOD RECOMMENDATION' on the left and 'MENU - LOG OUT' on the right. Below the header is a banner image of a plate of food. The main content area features a red modal box with the title 'Masukkan Data Posisi Badan'. Inside the modal, there are three input fields: 'Posisi Badan', 'Faktor Aktifitas', and 'Deskripsi Posisi Badan', each with a clear button (X) on the right. Below the input fields is a 'Simpan' (Save) button. The background of the page is light gray with a faint watermark of the Universitas Jember logo.

Gambar 34 Tampilan Menambah Data Posisi Badan

16. Tampilan Mengubah Data Posisi Badan



The screenshot shows a web application interface for editing body position data. At the top, there is a header with 'FOOD RECOMMENDATION' on the left and 'MENU - LOG OUT' on the right. Below the header is a banner image of a plate of food. The main content area features a red modal box with the title 'Ubah Data Posisi Badan'. Inside the modal, there are four input fields: 'Id Posisi: Badan' (containing '1'), 'Posisi: Badan' (containing 'duduk'), 'Faktor Aktifitas' (containing '0.3'), and 'Keterangan' (containing 'posisi badan a'). Each input field has a clear button (X) on the left. Below the input fields is a 'Simpan' (Save) button. The background of the page is light gray with a faint watermark of the Universitas Jember logo.

Gambar 35 Tampilan Mengubah Data Posisi Badan