

**RANCANG BANGUN SISTEM REKOMENDASI KULINER
KOTA MALANG DENGAN BERBASIS WEB
MENGUNAKAN METODE *WEIGHTED PRODUCT***

SKRIPSI

REKAYASA PERANGKAT LUNAK

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Diajukan oleh

Mutiara Arinda Putri

NIM. 105060801111037

KEMENTERIAN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

PRORAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

MALANG

2014

LEMBAR PERSETUJUAN

**Rancang Bangun Sistem Rekomendasi Kuliner Kota Malang
dengan Berbasis *Web* Menggunakan Metode *Weighted Product***

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer



Disusun Oleh :

Mutiara Arinda Putri

NIM.105060801111037

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Denny Sagita R., S.Kom, M.KomBudi Darma Setiawan, S.Kom.,M.Cs

NIK. 851124 06 1 1 0250

NIK. 841015 06 1 1 0090



LEMBAR PENGESAHAN

**Rancang Bangun Sistem Rekomendasi Kuliner Kota Malang
dengan Berbasis *Web* Menggunakan Metode *Weighted Product***

SKRIPSI

KONSENTRASI REKAYASA PERANGKAT LUNAK

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Mutiara Arinda Putri

NIM. 105060801111037

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 04 Juli 2014

Penguji I

Penguji II

Issa Arwani, S.Kom., M.Sc.
NIP. 198309222012121003

Fajar Pradana, S.ST., M.Eng.
NIK. 87112116110371

Penguji III

Eriq Muhammad Adams J, S.T.,M.Kom.
NIK. 85041006110027

Mengetahui

Ketua Program Studi Teknik Informatika

Drs. Marji, M.T.
NIP. 196708011992031001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 10 Juli 2014

Mahasiswa,

Mutiara Arinda Putri

NIM 105060801111037



KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan Skripsi yang berjudul “**Rancang Bangun Sistem Rekomendasi Kuliner Kota Malang dengan Berbasis Web Menggunakan Metode *Weighted Product***”. Shalawat dan salam atas junjungan besar kita Nabi Muhammad S.A.W. beserta keluarga dan para sahabat sekalian. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.

Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih penulis yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan bantuan baik lahir maupun batin selama penulisan skripsi ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Bapak Denny Sagita Rusdianto, S.Kom., M.Kom. dan Budi Darma Setiawan, S.Kom., M.Cs. selaku dosen pembimbing yang selalu memberikan masukan dalam penyusunan skripsi.
2. Bapak Eriq Muhammad Adams, S.T., M.Kom. selaku dosen penasehat akademik yang selalu memberikan nasehat akademis kepada penulis selama menempuh masa studi.
3. Bapak Ir. Sutrisno, M.T., Bapak Ir. Heru Nurwasito, M.Kom., Bapak Himawat Aryadita, S.T., M.Sc., dan Bapak Edy Santoso, S.Si.,M.Kom., selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Bapak Drs. Marji, M.T. dan Bapak Issa Arwani, S.Kom., M.Sc. selaku Ketua dan Sekretaris Program Studi Teknik Informatika Universitas Brawijaya.
5. Seluruh Dosen PTIIK Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
6. Seluruh Civitas Akademika Informatika/Ilmu Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di

Informatika / Ilmu Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.

7. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga skripsi ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 10 Juli 2014

Penulis



ABSTRAK

MUTIARA ARINDA PUTRI. 2014. Rancang Bangun Sistem Rekomendasi Kuliner Kota Malang dengan Berbasis Web Menggunakan Metode *Weighted Product*. Skripsi Program Studi Informatika, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Pembimbing: Denny Sagita Rusdianto, S.Kom., M.Kom. dan Budi Darma Setiawan, S.Kom., M.Cs.

Banyaknya restoran baru yang bermunculan di Kota Malang, membuat tantangan tersendiri bagi konsumen dalam menentukan restoran yang mereka inginkan. Bahkan hal ini bisa menjadi sesuatu yang sulit dan membutuhkan banyak waktu karena konsumen dihadapkan oleh banyak pilihan restoran. Oleh sebab itu, sebuah sistem rekomendasi dibutuhkan untuk dapat membantu para konsumen dalam memilih restoran sesuai dengan keinginan mereka. Pada penelitian ini, sistem rekomendasi kuliner dikembangkan menggunakan teknik *multiple criteria decision* tepatnya dengan metode *weighted product* dan berbasis web responsif. Metode ini akan memberikan rekomendasi restoran berdasarkan pilihan kriteria yang diinputkan oleh pengguna. Selain itu, sistem rekomendasi ini juga akan memberikan informasi tentang restoran-restoran tersebut.

Berdasarkan pengujian yang telah dilakukan, keseluruhan fungsional sistem rekomendasi ini dapat berjalan baik sesuai dengan kebutuhan yang telah dibuat. Selain itu, *user acceptance testing* yang dilakukan pada 100 responden menyatakan bahwa sistem rekomendasi kuliner ini dapat digunakan dengan mudah dan berguna membantu mereka dalam memilih restoran.

Kata kunci: Restoran , Sistem Rekomendasi, *Weighted Product*

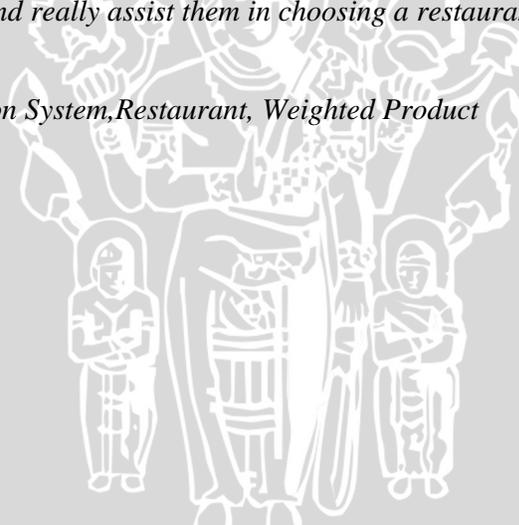
ABSTRACT

MUTIARA ARINDA PUTRI.2014. *Developing Web Based Malang Culinary Recommendation System Using Weighted Product Method. Supervisor:Denny Sagita Rusdianto, S.Kom., M.Kom. dan Budi Darma Setiawan, S.Kom., M.Cs.*

There are a lot of new restaurant in Malang City, make its own challenges for consumers in selecting a restaurant according what they need or want. Even this can be difficult and time-consuming because consumers are confronted by many choices of restaurants. Therefore, a recommendation system's needed to be able to assist consumers in selecting restaurants according to their needs. In this research, the culinary recommendation system is developed using multiple criteria decision technique, it is weighted product method. This method will let user to get recommendation from some criteria which is inserted by them. Beside, this recommendation system also provides information about those restaurant.

Based on the testing, the overall functional recommendation system is running well which is appropriate with requirements. In addition, user acceptance testing , which is conducted on 100 respondents revealed that this culinary recommendation system can be used easily and really assist them in choosing a restaurant.

Keywords : Recommendation System, Restaurant, Weighted Product



DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN.....	iii
PERNYATAAN ORISINALITAS SKRIPSI.....	iv
KATA PENGANTAR.....	v
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Kajian pustaka.....	5
2.2 Sistem Rekomendasi.....	7
2.3 Multi-Criteria Decision Making	8
2.3.1 Multi-attribute Decision Making	8
2.4 Metode Weighted Product (WP).....	9
2.5 Responsive Web Design	10
2.6 Codeigniter.....	11
2.6.1 Alur Proses Aplikasi.....	11
2.7 Rekayasa Perangkat Lunak	12

2.8	Software Process Model	13
2.9	Unified Modelling Language	15
2.9.1	Use Case Diagram	16
2.9.2	Activity Diagram	20
2.9.3	Class Diagram	20
2.9.4	Sequence Diagram	22
2.10	Pengujian (<i>Testing</i>)	24
2.10.1	Teknik Pengujian	25
BAB III METODOLOGI PENELITIAN		30
3.1	Requirement Specifications	30
3.1.1	Analisis Data	32
3.1.2	Daftar Kebutuhan	32
3.2	Component Analysis	35
3.3	Requirements Modifications	36
3.4	System Design with Reuse	38
3.4.1	Perancangan <i>Use Case</i>	38
3.4.2	Perancangan Aktivitas	39
3.4.3	Perancangan Kelas	39
3.4.4	Perancangan Interaksi	39
3.4.5	Perancangan Basis Data	39
3.4.6	Perancangan Algoritma	39
3.4.7	Perancangan Antarmuka	40
3.5	Development and Integration	40
3.6	System Validation	40
BAB IV PERANCANGAN DAN IMPLEMENTASI		42
4.1	Perancangan Algoritma	42

4.2	Perancangan <i>Use Case</i>	49
4.2.1	Identifikasi Aktor.....	49
4.3	Diagram Aktivitas.....	69
4.4	Diagram Kelas.....	74
4.5	Diagram Sekuensial (<i>Sequence Diagram</i>).....	98
4.6	Perancangan Basis Data.....	101
4.7	Perancangan Antarmuka.....	108
4.8	Implementasi Sistem.....	116
4.8.1	Implementasi Sistem.....	116
4.8.2	Spesifikasi Sistem.....	117
4.8.3	Batasan Implementasi.....	118
4.8.4	Implementasi Basis Data.....	118
4.8.5	Implementasi Prosedur Algoritma.....	120
4.8.6	Implementasi Antarmuka Aplikasi.....	121
BAB V	PENGUJIAN DAN ANALISIS.....	124
5.1	Pengujian.....	125
5.1.1	Pengujian <i>Basis Path</i>	125
5.1.2	Pengujian Validasi.....	134
5.1.3	Pengujian Kompabilitas <i>Device</i>	184
5.1.4	Pengujian Akurasi Sistem.....	187
5.1.5	UAT (User Acceptance Testing).....	191
5.2	Analisis.....	193
5.2.1	Analisis Pengujian <i>Basis Path</i>	194
5.2.2	Analisis Pengujian Validasi.....	194
5.2.3	Analisis Pengujian Kompatibilitas.....	194
5.2.4	Analisis Pengujian Akurasi Sistem.....	195

5.2.5	Analisis Pengujian UAT	195
BAB VI	PENUTUP	200
6.1	Kesimpulan	200
6.2	Saran	201
DAFTAR PUSTAKA	202
Lampiran 1. Kuesioner UAT	205



DAFTAR GAMBAR

Gambar 2. 1 Alur Proses Data pada *Codeigniter* 11

Gambar 2. 2 *Reuse-Oriented Software Engineering* 14

Gambar 2. 3 Diagram UML 16

Gambar 2. 4 Simbol Diagram Aktivitas 20

Gambar 2. 5 Transformasi flow chart (A) ke flow graph (B)..... 26

Gambar 2. 6 Hubungan *Cyclometric Complexity* dan Resiko..... 27

Gambar 3. 1 Alur Penelitian dengan *Reuse Oriented*..... 30

Gambar 4. 1 Diagram Perancangan..... 42

Gambar 4. 2 Diagram Alir Cara Kerja Sistem Rekomendasi Kuliner..... 44

Gambar 4. 3 *Use Case* Sistem Rekomendasi Kuliner Berbasis Web 50

Gambar 4. 4 *Use Case* Modul Komentar Sistem Rekomendasi Kuliner Berbasis Web . 51

Gambar 4. 5 *Use Case* Modul Fasilitas dan Modul Kategori Makanan..... 51

Gambar 4. 6 *Use Case* Modul Berita..... 52

Gambar 4. 7 *Use Case* Modul Promo 52

Gambar 4. 8 *Use Case* Modul Menu 53

Gambar 4. 9 *Use Case* Modul Akun Admin 53

Gambar 4. 10 *Use Case* Modul Akun dan Data Restoran 54

Gambar 4. 11 *Activity Diagram Register* 70

Gambar 4. 12 *Activity Diagram Menu* 71

Gambar 4. 13 *Acitivity Diagram Cari Restoran* 72

Gambar 4. 14 *Activity Diagram Cari Rekomendasi Restoran*..... 73

Gambar 4. 15 *Class Diagram* Kelas Login 74

Gambar 4. 16 *Class Diagram* Kelas Tipe 74

Gambar 4. 17 *Class Diagram* Kelas Fasilitas 75

Gambar 4. 18 *Class Diagram* Kelas Administrator 75

Gambar 4. 19 *Class Diagram* Kelas Main 75

Gambar 4. 20 *Class Diagram* Kelas Komentar 76

Gambar 4. 21 *Class Diagram* Kelas Akun 77

Gambar 4. 22	<i>Class Diagram</i> Kelas Menu	78
Gambar 4. 23	<i>Class Diagram</i> Kelas Berita	79
Gambar 4. 24	<i>Class Diagram</i> Kelas Promo	95
Gambar 4. 25	<i>Class Diagram</i> Model MVC	96
Gambar 4. 26	<i>Class Diagram</i> Controller MVC	97
Gambar 4. 27	<i>Sequence Diagram</i> Cari Rekomendasi	98
Gambar 4. 28	<i>Sequence Diagram</i> Register	99
Gambar 4. 29	<i>Sequence Diagram</i> Modul Kelola Fasilitas	100
Gambar 4. 30	Perancangan Basis Data	102
Gambar 4. 31	<i>Sitemap</i> Sistem Rekomendasi Kuliner Pengelola Rumah Makan	109
Gambar 4. 32	<i>Sitemap</i> Sistem Rekomendasi Kuliner <i>Administrator</i>	110
Gambar 4. 33	Perancangan Antarmuka Halaman Utama	111
Gambar 4. 34	Perancangan Antarmuka Halaman <i>Administrator</i>	112
Gambar 4. 35	Perancangan Antarmuka Halaman Pengelola Rumah Makan	113
Gambar 4. 36	Perancangan Antarmuka Halaman Promo	114
Gambar 4. 37	Perancangan Antarmuka Halaman Berita	115
Gambar 4. 38	Diagram Implementasi	116
Gambar 4. 39	Implementasi Basis Data	119
Gambar 4. 40	Implementasi Algoritma Dengan Fungsi Hitung	120
Gambar 4. 41	Implementasi Algoritma dengan Fungsi Hitung1()	120
Gambar 4. 42	Implementasi Algoritma untuk Menampilkan Rekomendasi	121
Gambar 4. 43	Antarmuka Halaman Utama Sistem Rekomendasi Kuliner	121
Gambar 4. 44	Antarmuka Halaman <i>Administrator</i>	122
Gambar 4. 45	Antarmuka Halaman Pengelola Restoran	122
Gambar 4. 46	Antarmuka Halaman Promo	122
Gambar 4. 47	Antarmuka Halaman Berita	123
Gambar 5. 1	Diagram Pengujian dan Analisis	124
Gambar 5. 2	Basis Path Fungsi Search()	126
Gambar 5. 3	Basis Path Fungsi hitung()	127
Gambar 5. 4	Basis Path Fungsi hitung1()	130
Gambar 5. 5	Basis Path Fungsi hitung2()	133

DAFTAR TABEL

Tabel 2. 1 Keterangan Simbol-simbol Use Case	17
Tabel 2. 2 Keterangan Simbol-Simbol Class Diagram	22
Tabel 2. 3 Simbol-simbol Diagram Sekuen	23
Tabel 3. 1 Daftar Kebutuhan Fungsional.....	32
Tabel 3. 2 Daftar Kebutuhan Non-Fungsional	34
Tabel 3. 3 Daftar Komponen	35
Tabel 3. 4 Integrasi Komponen dengan Kebutuhan Fungsional	36
Tabel 3. 5 Integrasi Komponen dengan Kebutuhan Non-Fungsional	38
Tabel 4. 1 Daftar Nilai Kriteria	45
Tabel 4. 2 Keterangan Bobot Kriteria	46
Tabel 4. 3 Data Rumah Makan Yang Dicari Pengguna	46
Tabel 4. 4 Data Rumah Makan yang Dicari	47
Tabel 4. 5 Hasil Normalisasi	48
Tabel 4. 6 Deskripsi Aktor	49
Tabel 4. 7 Use Case Register.....	54
Tabel 4. 8 Use Case Kelola Akun Admin	55
Tabel 4. 9 Use Case Kelola Kategori Fasilitas	56
Tabel 4. 10 <i>Use Case Kelola Kategori Makanan</i>	57
Tabel 4. 11 Use Case Reset Password.....	58
Tabel 4. 12 Kelola Akun dan Data Restoran	59
Tabel 4. 13 Use Case Kelola Menu Restoran.....	61
Tabel 4. 14 Use Case Kelola Promo Restoran.....	62
Tabel 4. 15 Use Case Kelola Promo Restoran.....	64
Tabel 4. 16 Use Case Kelola Berita.....	65
Tabel 4. 17 Use Case Cari Restoran	67
Tabel 4. 18 Use Case Cari Rekomendasi Restoran	68
Tabel 4. 19 Use Case Lihat Petunjuk Arah	69
Tabel 4. 20 Struktur Tabel Restaurant.....	103

Tabel 4. 21 Struktur Tabel Promo	104
Tabel 4. 22 Struktur Tabel Berita	104
Tabel 4. 23 Struktur Tabel Restaurant Facilities	105
Tabel 4. 24 Struktur Tabel Facilities	105
Tabel 4. 25 Struktur Tabel Cuisine	106
Tabel 4. 26 Struktur Tabel Menu	106
Tabel 4. 27 Struktur Tabel Komentar	106
Tabel 4. 28 Struktur Tabel Sub Komentar	107
Tabel 4. 29 Struktur Tabel Account	107
Tabel 4. 30 Struktur Tabel User Admin	108
Tabel 4. 31 Tabel Spesifikasi Perangkat Keras	117
Tabel 4. 32 Tabel Spesifikasi Perangkat Lunak	117
Tabel 5. 1 Uji Validitas Register	134
Tabel 5. 2 Uji Validitas Kelola Akun Admin	135
Tabel 5. 3 Uji Validitas Edit Akun	136
Tabel 5. 4 Uji Validitas Lihat Fasilitas	137
Tabel 5. 5 Tambah Kategori Fasilitas	138
Tabel 5. 6 Uji Validitas Lihat Kategori Makanan	140
Tabel 5. 7 Uji Validitas Tambah Kategori Makanan	141
Tabel 5. 8 Uji validitas Reset Password	142
Tabel 5. 9 Uji Validitas Lihat Data	144
Tabel 5. 10 Uji Validitas Edit Data	146
Tabel 5. 11 Uji Validitas Tambah Data	150
Tabel 5. 12 Uji Validitas Hapus Data	152
Tabel 5. 13 Uji Validitas Aktivasi	152
Tabel 5. 14 Uji Validitas Deaktivasi	153
Tabel 5. 15 Uji Validitas Lihat Menu	154
Tabel 5. 16 Uji Validitas Tambah Menu	156
Tabel 5. 17 Uji Validitas Hapus Menu	158
Tabel 5. 18 Uji Validitas lihat Promo	159
Tabel 5. 19 Uji Validitas Lihat Detail Promo	161

Tabel 5. 20 Uji Validitas Tambah Promo.....	163
Tabel 5. 21 Uji Validasi Edit Promo	164
Tabel 5. 22 Uji Validasi Hapus Promo.....	166
Tabel 5. 23 Uji Validitas Lihat Komentar	167
Tabel 5. 24 Uji Validitas Hapus Komentar	169
Tabel 5. 25 Uji Validitas Tambah Komentar	170
Tabel 5. 26 Uji Validitas Lihat Berita	173
Tabel 5. 27 Uji Validitas Lihat Detail Berita.....	175
Tabel 5. 28 Uji Validitas Tambah Berita.....	177
Tabel 5. 29 Uji Validitas Edit Berita	179
Tabel 5. 30 Uji Validitas Hapus Berita	180
Tabel 5. 31 Uji Validitas Cari Restoran	181
Tabel 5. 32 Uji Validitas Cari Rekomendasi Restoran.....	182
Tabel 5. 33 Uji Validitas Lihat Petunjuk Arah.....	183
Tabel 5. 34 Uji Validitas Kompabilitas Tablet Samsung.....	184
Tabel 5. 35 Pengujian Kompabilitas Smartphone Apple	185
Tabel 5. 36 Uji Kompatibilitas Ipad Mini	186
Tabel 5. 37 Uji Akurasi Sistem Rekomendasi.....	188
Tabel 5. 38 Demografi Responden	193
Tabel 5. 39 Hasi Pengujian Basis Path.....	194
Tabel 5. 40 Skor Ideal	195
Tabel 5. 41 Nilai Scale Rating.....	196
Tabel 5. 42 Distribusi Hasil UAT.....	196

BAB I PENDAHULUAN

1.1 Latar Belakang

Kota Malang adalah salah satu kota di Jawa Timur yang memiliki luas sekitar 110,06 km² dengan jumlah penduduk kurang lebih sekitar 820.243 jiwa (Sekilas Kota Malang,2011). Kota Malang sendiri merupakan kota terbesar kedua di Jawa Timur setelah Kota Surabaya (Sekilas Kota Malang,2011). Kota Malang dikenal sebagai kota pendidikan dan kota wisata karena memiliki sekitar 40 Perguruan Tinggi (negeri dan swasta) dan beragam objek wisata.

Kelebihan-kelebihan inilah yang membuat Kota Malang menjadi salah satu lokasi strategis bagi para *investor* untuk mengembangkan bisnisnya, salah satunya pada bidang industri makanan dan minuman. Berdasarkan data BPS Kota Malang tahun 2012, kegiatan industri makanan, minuman, dan tembakau menjadi penyumbang terbesar pada sektor perdagangan di Kota Malang dengan prosentase sebesar 31,74 %. Hal ini disebabkan oleh banyaknya jumlah restoran atau rumah makan di Kota Malang baik yang berskala besar maupun yang berskala kecil/mikro. Berdasarkan data Pemerintah Kota Malang pada tahun 2010 saja, sedikitnya terdapat 304 restorandan cafe yang tersebar di Kota Malang.

Akibatnya, memilih restoran dapat menjadi satu tantangan tersendiri bagi konsumen. Bahkan hal ini bisa menjadi sulit dan membutuhkan banyak waktu karena konsumen dihadapkan oleh banyak pilihan restoran. Oleh karena itu, dibutuhkan sebuah sistem yang dapat memberikan rekomendasi untuk membantu konsumen memilih restoran sesuai dengan apa yang mereka butuhkan atau yang mereka inginkan. Selain itu, sistem tersebut mampu menyediakan informasi tentang restoran tersebut dimana hal ini juga dapat menjadi pertimbangan lain dalam memilih restoran.

Sistem rekomendasi bukanlah hal yang baru saat ini. Banyak vendor-vendor *online*, seperti Amazon dan Netflix sudah mengimplementasikan sistem rekomendasi untuk membantu para pelanggan mereka. Ada beberapa pendekatan yang biasa digunakan dalam sistem rekomendasi saat ini, yaitu *:content-based filtering*, *collaborative filtering* dan *hybrid recommender*. Akan tetapi ketiga pendekatan tersebut

lebih terfokus pada preferensi dan analisis kemiripan *item* yang telah dinilai oleh para penggunanya, bukan berdasarkan atribut produk ataupun kriteria yang dibutuhkan dan diinginkan oleh konsumen.

Oleh sebab itu, metode MCDM (*Multi-criteria Decision Making*) seperti *weighted product method* (WPM) menawarkan jawaban untuk permasalahan di atas. MCDM (*Multi-criteria Decision Making*) merujuk pada pengambilan keputusan dari sejumlah alternatif tertentu berdasarkan beberapa kriteria (Kusumadewi, 2006). Pemilihan restoran melibatkan beberapa variabel yang berarti dapat dirumuskan sebagai permasalahan MCDM (Ahn,2006).

Berdasarkan penelitian "*Product Evaluation Using Entropy and Multicriteria Decision Making Methods*" *weighted product* merupakan salah satu metode digunakan untuk pemilihan produk laptop. *Weighted product* memberikan rekomendasi produk yang sama dengan dua metode lain yang diuji, yaitu : TOPSIS dan SAW. Ketiga metode ini memberikan *preferensi* produk yang lebih baik dibandingkan dengan alternatif produk lainnya. Selain itu, *weighted product* juga digunakan pada industri pemilihan robot pada penelitian "*A Comparative Study on The Ranking Performance of Some Multi-Criteria Decision Making Methods for Industrial Robot Selection*". Berdasarkan penelitian tersebut, *Weighted product* menjadi salah satu metode terbaik yang dapat digunakan untuk industri pemilihan robot. Penelitian "*Sistem Pendukung Keputusan Pemilihan Penerima Jamkesmas Menggunakan Metode Weighted Product*" juga menggunakan metode WPM untuk menyeleksi calon penerima Jamkesmas dan tingkat akurasi yang dihasilkan pun cukup baik, yaitu sebesar 86,6% .Oleh karena itu, *weighted product* dipilih sebagai metode rekomendasi untuk sistem ini.

Sistem rekomendasi kuliner berbasis web dengan metode *weighted product* ini, diharapkan dapat lebih membantu konsumen untuk memilih restoran sesuai dengan kebutuhan dan keinginannya. Selain itu, diharapkan sistem ini juga dapat memberikan informasi lebih banyak mengenai tempat kuliner di Kota Malang.

1.2 Rumusan Masalah

Berdasarkan paparan latar belakang tersebut, maka rumusan masalah dalam penelitian ini adalah bagaimana menganalisis kebutuhan, melakukan perancangan,

mengimplementasikan, dan menguji sebuah sistem rekomendasi kuliner di Kota Malang dengan berbasis web menggunakan metode *weighted product*.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Studi Kasus dilakukan di Kota Malang, Jawa Timur.
2. Kriteria yang digunakan dalam perhitungan rekomendasi kuliner adalah harga, lokasi, kualitas pelayanan yang merujuk pada ketersediaan fasilitas yang dibutuhkan oleh konsumen/calon pelanggan.
3. Model pengembangan sistem rekomendasi kuliner ini menggunakan model *reuse oriented software engineering*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah :

Untuk membangun sebuah sistem yang dapat memberikan rekomendasi restoran berdasarkan kriteria yang dibutuhkan oleh konsumen. Selain itu, sistem ini dapat memberikan informasi mengenai restoran yang ada di Kota Malang.

1.5 Manfaat Penelitian

Manfaat dari penelitian adalah untuk membangun sebuah sistem yang dapat memberikan rekomendasi restoran sesuai dengan kriteria yang diminta konsumen serta membantu pengelola restoran dalam mempromosikan usaha yang dimiliki secara *online*

1.6 Sistematika Penulisan

Penyusunan skripsi ini menggunakan kerangka pembahasan yang tersusun sebagai berikut :

1. BAB I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. BAB II Tinjauan Pustaka

Membahas teori-teori dan penelitian-penelitian yang mendukung dalam pengembangan dan perancangan sistem rekomendasi kuliner di Kota Malang dengan berbasis web menggunakan metode *weighted product*.

3. BAB III Metodologi Penelitian

Membahas tentang metode yang digunakan untuk mengembangkan sistem dengan model *reuse development*.

4. BAB IV Perancangan dan Implementasi

Membahas tentang analisis kebutuhan dan perancangan sistem rekomendasi kuliner di Kota Malang dengan berbasis web menggunakan metode *weighted product* kemudian mengimplementasikannya sesuai dengan rancangan yang telah dibuat.

5. BAB V Pengujian dan Analisis

Memuat tentang hasil pengujian dan analisis terhadap sistem yang telah direalisasikan.

6. Bab VI Penutup

Memuat kesimpulan yang diperoleh dari pengembangan dan pengujian sistem dalam skripsi ini serta saran – saran untuk penelitian lebih lanjut



BAB II

TINJAUAN PUSTAKA

Bab dua terdiri dari kajian pustaka dan dasar teori. Kajian pustaka merupakan pembahasan penelitian yang mendukung penelitian yang akan diusulkan. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan. Kajian pustaka pada penelitian ini adalah membahas enam penelitian yang berjudul '*Visualisasi Geografi untuk Sistem Pendukung Keputusan Wisata Kuliner menggunakan Metode rule of Thumb*', '*Ranking of Product Design Alternatives using Multi-criteria Decision Making Methods*', '*A Comparative Study On The Ranking Performance Of Some Multi-Criteria Decision-Making Methods For Industrial Robot Selection*', '*Vertical Handover Decision Schemes Using SAW And WPM For Network Selection In Heterogeneous Wireless Networks*', '*Product Evaluation Using Entropy and Multi Criteria Decision Making Methods*', dan '*Sistem Pendukung Keputusan Pemilihan Penerima Jamkesmas Menggunakan Metode Weighted Product*'. Pada penelitian ini, dasar teori yang diperlukan berdasarkan latar belakang dan rumusan masalah adalah pengertian dari sistem rekomendasi, pengertian dari *Multi-criteria Decision Making* (MCDM), konsep dasar *weighted product*, konsep dasar *responsive web design*, konsep dasar *code igniter* (CI), pengembangan dan rekayasa perangkat lunak, *software process model*, konsep dasar *Unified Modelling Language* yang dipakai pada rekayasa perangkat lunak, dan konsep pengujian perangkat lunak.

2.1 Kajian pustaka

Penelitian mengenai sistem pendukung keputusan pemilihan restoran atau wisata kuliner, pernah dilakukan sebelumnya. Hamdani (2012) membuat sistem pendukung keputusan wisata kuliner Kota Yogyakarta. Wisata kuliner diambil sebagai objek penelitian karena wisata kuliner menjadi hal yang sangat populer saat ini dan sudah menjadi tren gaya hidup masyarakat dalam melakukan wisata liburan di suatu daerah tertentu. Oleh karena itu, dibutuhkan sebuah sistem pendukung keputusan yang dapat membantu masyarakat untuk menentukan lokasi tempat kuliner yang mereka inginkan. Metode yang digunakan pada sistem

pendukung keputusan ini adalah metode *rule of thumb*. Metode ini memudahkan pengguna dalam memenuhi kebutuhan untuk melakukan wisata kuliner sesuai lokasi yang ditentukan oleh pengguna. Berdasarkan paparan di atas, maka penelitian yang diusulkan, mengangkat objek yang sama, yaitu : wisata kuliner. Akan tetapi, metode yang akan digunakan pada penelitian yang diusulkan menggunakan metode MCDM (*Multi-criteria Decision Making*), *weighted product*.

Metode MCDM (*Multi-criteria Decision Making*) telah diterapkan pada berbagai permasalahan. Gangurde dan Akarte (Gangurde & Akarte, 2011) melakukan pemilihan PDA (*Personal Data Assistant*) berdasarkan kebutuhan konsumen dengan berbagai pendekatan (metode). Metode-metode yang digunakan adalah metode-metode MCDM karena pemilihan PDA dianggap sebagai permasalahan *multi-criteria decision* yang melibatkan banyak atribut dan sudut pandang yang berbeda pada tiap konsumen. Alternatif PDA akan dievaluasi dengan *Simple Additive Weighting* (SAW), *Weighting Product Method* (WPM), TOPSIS Method, Modified TOPSIS Method, GRA Method, PROMETHEE Method, dan, VIKOR Method. Setelah dilakukan perhitungan dan perankingan, maka didapatkan kesimpulan bahwa alternatif yang dihasilkan oleh SAW dan WPM lebih baik jika dibandingkan dengan metode lainnya yang diujikan.

Athawale dan Chakraborty (Athawale & Chakraborty,2011) juga mengusulkan prosedur pemilihan robot dengan berbagai metode MCDM. MCDM digunakan karena dalam pemilihan robot melibatkan beberapa kriteria dan menghasilkan beberapa alternatif. Oleh karena itu, MCDM dapat digunakan untuk menyelesaikan permasalahan ini. Sepuluh pendekatan MCDM digunakan untuk pemilihan robot ini. Semua metode memberikan ranking alternatif robot yang hampir sama, tetapi berdasarkan *Z test* yang dilakukan, metode WPM (*Weighted Product Method*), TOPSIS, dan GRA memiliki nilai yang lebih tinggi, hampir mendekati angka 2. lebih baik dibandingkan dengan metode lainnya.

Hal yang sama juga diungkapkan oleh Savitha dan Chandrasekar (Savitha & Chandrasekar,2011) dan Dashore dkk (Dashore,Pawar,Sohani,&Verma,2013). Meskipun objek kedua jurnal ini berbeda, Savitha dan Chandrasekar (2011), melakukan perbandingan metode MCDM yang sesuai untuk memilih jaringan

terbaik saat terjadi *handover*. Sedangkan Dashore dkk (2013), melakukan perankingan produk laptop dengan metode SAW, WPM, TOPSIS, dan Modified TOPSIS.), hasil perankingan yang dilakukan pada kedua jurnal ini menyatakan bahwa metode WPM (*Weighted Product Method*) merupakan salah satu metode yang memberikan alternatif terbaik dibandingkan dengan metode lainnya.

Penelitian '*Sistem Pendukung Keputusan Pemilihan Penerima Jamkesmas Menggunakan Metode Weighted Product*' (Astriana, 2014) menggunakan metode *weighted product* untuk pemilihan penerimaan Jamkesmas dengan kriteria-kriteria tertentu dimana nilai kriteria tersebut ditentukan dari kuisioner *user*. Tingkat akurasi pada sistem ini cukup baik, yaitu sebesar 86,6 %.

Oleh karena itu, penelitian ini mengangkat menggunakan metode *weighted product* (WPM) sebagai perhitungan rekomendasi untuk pemilihan restoran.

2.2 Sistem Rekomendasi

Konsep sistem rekomendasi telah digunakan secara luas oleh hampir semua area bisnis dimana seorang konsumen memerlukan informasi untuk membuat suatu keputusan (Sharda, 2010). Sistem rekomendasi kuliner ini menggunakan konsep tersebut untuk membantu konsumen dalam menentukan rumah makan yang sesuai dengan keinginan konsumen.

Sistem rekomendasi merupakan model aplikasi dari hasil observasi terhadap keadaan dan keinginan pelanggan. Oleh karena itu sistem rekomendasi memerlukan model rekomendasi yang tepat agar yang direkomendasikan sesuai dengan keinginan pelanggan, serta mempermudah pelanggan mengambil keputusan yang tepat dalam menentukan produk yang akan digunakannya (McGinty & Smyth, 2006).

Sistem rekomendasi merupakan sebuah (web) alat personalisasi yang menyediakan pengguna sebuah informasi daftar item-item yang sesuai dengan keinginan masing-masing pengguna (Sebastia, Garcia, Onaindia, Guzman, 2009). Sistem rekomendasi menyimpulkan preferensi pengguna dengan menganalisis ketersediaan data pengguna, informasi tentang pengguna dan lingkungannya (Sebastia, Garcia, Onaindia, Guzman, 2009). Oleh karena itu, sistem rekomendasi akan menawarkan kemungkinan dari penyaringan informasi personal sehingga

hanya informasi yang sesuai dengan kebutuhan dan preferensi pengguna yang akan ditampilkan di sistem dengan menggunakan sebuah teknik atau model rekomendasi (Sebastia,Garcia,Onaindia,Guzman,2009).

2.3 Multi-Criteria Decision Making

Keputusan merupakan proses pemilihan alternatif terbaik dari banyak alternatif. Pengambilan keputusan terkadang melibatkan pengalaman. Tidak jarang pula *decision maker* (DM) mengambil keputusan dengan menggunakan insting atau intuisi sehingga menghasilkan keputusan yang tidak tepat. Oleh karena itu, untuk menghasilkan yang tepat DM harus memperoleh informasi sebanyak mungkin.

Metode MCDM ditujukan untuk pengambilan keputusan yang mengandung kriteria objek majemuk juga saling konfliktual dan memiliki ukuran yang tidak bisa saling dibandingkan (Ciptomulyono, 2000). MCDM selalu melibatkan lebih dari satu kriteria yang saling menimbulkan *trade off* keputusan dimana tingkat kepuasan dari suatu kriteria berakibat pada penurunan kepuasan kriteria lainnya. Dalam menilai tingkat kepentingan dalam *multiple criteria*, ada beberapa metode yang dapat digunakan. Teknik pada kategori ini melibatkan satu atau sekelompok orang yang pada umumnya terdiri dari para ahli, bisa juga untuk DM.

MCDM dianggap sebagai istilah untuk semua model dan teknik yang berhubungan dengan *Multiobjective Decision Making* (MODM) dan *Multi-attribute Decision Making* (MADM) (Tabucanon,1988) .

2.3.1 Multi-attribute Decision Making

Model *Multi-attribute Decision Making* (MADM) digunakan untuk menyelesaikan masalah-masalah dalam ruang diskret sehingga penilaian atau seleksi dilakukan terhadap beberapa alternatif dalam jumlah terbatas (Kusrini,2007). Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah MADM antara lain (Novita,2012):

1. *Simple Additive Weighting Method* (SAW)
2. *Weighted Product* (WP)
3. *ELECTRE*

4. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)
5. *Analytic Hierarchy Process* (AHP)

2.4 Metode Weighted Product (WP)

Metode WP merupakan salah satu metode penyelesaian yang ditawarkan untuk menyelesaikan masalah Multi Attribute Decision Making (MADM). Metode WP mirip dengan Metode Weighted Sum (WS), hanya saja metode WP terdapat perkalian dalam perhitungan matematikanya. Metode WP juga disebut analisis berdimensi karena struktur matematikanya menghilangkan satuan ukuran (Savitha & Chandrasekar, 2011).

Metode weighted product menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot yang bersangkutan. Preferensi A_i diberikan pada Rumus 2-1 (Kusrini, 2007)

$$S_i = \prod_{j=1}^n X_{ij}^{W_j}, \text{ dengan } i = 1, 2, \dots, m \quad (2-1)$$

dimana :

S : Preferensi alternatif dianalogikan sebagai vektor S

X : Nilai kriteria

W : Bobot kriteria/subkriteria

i : Alternatif

j : Kriteria

n : Banyaknya kriteria

dimana $\sum w_j = 1$. w_j adalah pangkat bernilai positif untuk atribut keuntungan, dan bernilai negatif untuk atribut biaya.

Preferensi relatif dari setiap alternatif, diberikan pada Rumus 2-2 :

$$V_i = \frac{\prod_{j=1}^n X_{ij}^{W_j}}{\prod_{j=1}^n (X_j^W)^{W_j}}, \text{ dengan } i = 1, 2, \dots, m. \quad (2-2)$$

dimana :

V : Preferensi alternatif dianalogikan sebagai vektor V

X : Nilai Kriteria

W : Bobot kriteria/subkriteria

- i : Alternatif
- j : Kriteria
- n : Banyaknya kriteria

Langkah – langkah menggunakan metode WP adalah sebagai berikut (Basyaib,2006):

1. Melakukan perbaikan bobot dengan menggunakan rumus 2-3.

$$W_j = \frac{w_j}{\sum w_j} \quad (2-3)$$

2. Mengalikan seluruh atribut bagi sebuah alternatif dengan bobot sebagai pangkat positif untuk atribut manfaat dan bobot berfungsi sebagai pangkat negatif pada atribut biaya.
3. Hasil perkalian dijumlahkan untuk menghasilkan nilai pada setiap alternatif.
4. Mencari nilai alternatif dengan melakukan langkah yang sama seperti langkah satu, hanya saja menggunakan nilai tertinggi untuk setiap atribut tertinggi untuk setiap atribut manfaat dan terendah untuk atribut biaya.
5. Membagi nilai V bagi setiap alternatif dengan nilai standar ($V(A^*)$) yang menghasilkan R.
6. Ditemukan urutan alternatif terbaik yang akan menjadi keputusan.

2.5 Responsive Web Design

Responsive Web Design (RWD) merupakan pendekatan dalam desain *web* yang bertujuan untuk membuat sebuah situs atau aplikasi berbasis *website* yang memberikan tampilan optimal, kemudahan membaca, dan navigasi dengan ukuran minimal, *panning*, dan *scrolling* pada berbagai macam *device* (dari *mobile phone* hingga *desktop* ataupun komputer) (Marcotte,2010). RWD terdiri dari tiga komponen utama, yaitu :

1. *A Flexible, grid-based layout*,
2. *Flexible images and media*, dan
3. *Media queries*, sebuah modul dari CSS3

Pada skripsi ini untuk memudahkan dalam membuat desain web responsif, maka digunakanlah sebuah *framework* yaitu : *Bootstrap*. *Bootstrap* merupakan

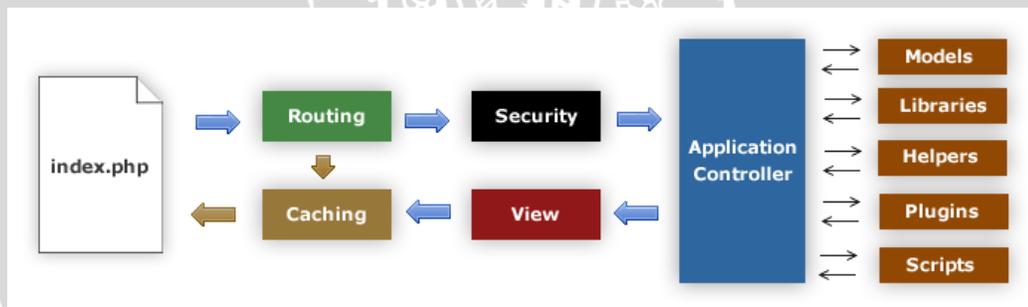
framework ataupun *tools* untuk membuat aplikasi *web* ataupun situs *web* responsif secara cepat, mudah, dan gratis (Alatas,2013).

2.6 Codeigniter

Codeigniter (CI) adalah sebuah kerangka atau *framework* pembangunan aplikasi atau lebih mudahnya disebut *toolkit*, untuk *developer* yang akan membangun sebuah aplikasi berbasis *web* dengan menggunakan PHP. Tujuan CI adalah untuk membangun sebuah aplikasi yang lebih cepat dibandingkan dengan menulis *source code* dari awal karena CI telah menyediakan banyak *library* untuk proses-proses yang sering digunakan pada aplikasi serta dengan kemudahan dalam menggunakan *library* tersebut serta kesederhanaan penggunaannya (Utama,2011).

2.6.1 Alur Proses Aplikasi

Gambar 2.1 merupakan ilustrasi alur proses data pada CI.



Gambar 2.1 Alur Proses Data pada *Codeigniter*
Sumber : (Utama,2011)

1. Index.php berfungsi sebagai pengendali awal dan menginisialisasi sumber daya utama yang dibutuhkan *codeigniter*.
2. *Routing* memeriksa paket HTTP *request* untuk menentukan aksi apa yang harus dilakukan sistem.
3. Jika *cache* tersedia, maka halaman langsung dikirim ke *browser*, eksekusi sistem yang normal akan dilewati.
4. *Security*, sebelum *application controller* dieksekusi, paket HTTP *request* dan semua data yang dikirimkan pengguna akan disaring terlebih dahulu oleh *security class*.

5. *Application Controller* menginisialisasi model, *library* utama, *helpers*, dan semua sumber daya yang dibutuhkan untuk setiap *request*.
6. Antarmuka aplikasi (*view*) yang sudah disiapkan dikirimkan ke *browser*. Jika *caching* diaktifkan, maka *view* akan disimpan sementara untuk *request* yang sama berikutnya.

2.6.2 Model, View, Controller, Libraries, Helper

Layaknya *framework* PHP pada umumnya, *codeigniter* menggunakan konsep MVC serta menyediakan banyak *library* dan *helper* untuk digunakan. Berikut penjelasan mengenai model, *view*, *controller*, *library*, dan *helper* (Utama,2011).

1. Model mempresentasikan struktur data. Biasanya *class* model akan berisi fungsi-fungsi untuk mengambil data, *insert* data, dan *update* data ke *database*.
2. *View* merupakan informasi/halaman yang ditampilkan ke pengguna. Sebuah *view* biasanya adalah sebuah *web page*, tapi di *codeigniter* *view* juga dapat berupa bagian-bagian halaman *web*, seperti *header* dan *footer*.
3. *Controller* berfungsi sebagai penghubung antara model dan *view* serta dengan sumber daya lain yang digunakan untuk memproses HTTP *request*. *Controller* juga biasanya berfungsi sebagai inti pemrosesan logik aplikasi.
4. *Libraries* adalah macam-macam *class* yang masing-masing mempunyai fungsi khusus yang dapat digunakan untuk mengembangkan aplikasi.
5. *Helper* berfungsi untuk melakukan tugas-tugas tertentu. Setiap *file helper* terdiri dari kumpulan fungsi (*function*). *Helper* tidak menggunakan format *object oriented* sehingga dapat digunakan dimanapun baik pada model, *view*, *controller*, dan *library*.

2.7 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah digunakan (Sommerville,2011). Pada definisi ini, ada dua istilah kunci :

1. ‘disiplin rekayasa’

Perekayasa membuat suatu alat bekerja. Perekayasa menerapkan teori, metode, dan alat bantu yang sesuai. Perekayasa menggunakannya dengan selektif dan selalu mencoba mencari solusi terhadap permasalahan, walaupun tidak ada teori atau metode yang mendukung. Perekayasa juga menyadari bahwa mereka harus bekerja dalam batasan organisasi dan keuangan, sehingga mereka berusaha mencari solusi dalam batasan – batasan ini.

2. ‘semua aspek produksi perangkat lunak’

Rekayasa perangkat lunak tidak hanya berhubungan dengan proses teknis dari pengembangan perangkat lunak tetapi juga dengan kegiatan seperti manajemen proyek perangkat lunak dan pengembangan alat bantu, metode dan teori untuk mendukung produksi perangkat lunak.

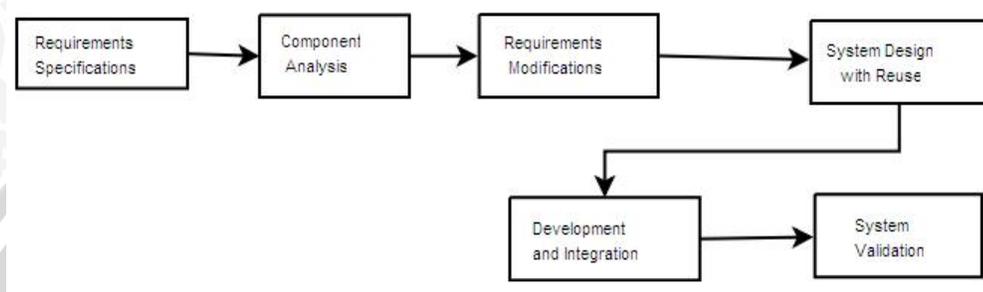
Perekayasa perangkat lunak memakai pendekatan yang sistematis dan terorganisir terhadap pekerjaan mereka karena cara ini seringkali paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Rekayasa ini mencakup masalah pemilihan metode yang paling sesuai untuk satu set keadaan dan pendekatan yang lebih kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan (Sommerville,2011).

2.8 Software Process Model

Model proses perangkat lunak adalah representasi yang sederhana dari proses perangkat lunak. Setiap model proses merupakan proses dari perspektif tertentu dan dengan demikian hanya menyediakan informasi parsial tentang proses tersebut, misalnya model proses aktivitas yang menunjukkan kegiatan dan urutan namun tidak menunjukkan peran orang yang terlibat dalam aktivitas ini. Sebuah model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat proyek dan aplikasi, metode, dan alat-alat yang akan digunakan, dan kontrol *deliverable* yang diperlukan. Beberapa model proses perangkat lunak adalah *waterfall model*, *incremental development*, dan *reuse-oriented software engineering* (Sommerville,2011).

Reuse-oriented software engineering adalah penggunaan kembali perangkat lunak yang telah ada. Hal ini sering terjadi ketika orang-orang yang

bekerja pada proyek mengetahui desain atau kode yang mirip dengan apa yang dibutuhkan. Mereka mencari dan memodifikasi sesuai kebutuhan lalu menggabungkan ke dalam sistem. Pendekatan *reuse-oriented software engineering* berfokus pada mengintegrasikan komponen pada sistem dibandingkan dengan mengembangkan dari awal (Sommerville,2011).



Gambar 2.2 *Reuse-Oriented Software Engineering*
Sumber : (Sommerville,2011)

Proses-proses yang terlibat pada *reuse-oriented software engineering* dapat dilihat pada gambar 2.2. Meskipun tahap awal spesifikasi kebutuhan dan tahap validasi sama dengan model lainnya, tetapi tahap-tahap pertengahan proses *reuse-oriented* berbeda (Sommerville,2011). Tahap-tahap tersebut adalah sebagai berikut :

1. Analisis komponen

Dari spesifikasi kebutuhan yang telah ditentukan, dicari komponen yang dapat mengimplementasikan spesifikasi tersebut. Biasanya tidak ada komposisi yang tepat dan komponen yang dapat digunakan hanya menyediakan beberapa fungsi yang diperlukan dari spesifikasi kebutuhan.

2. Modifikasi Kebutuhan

Setiap tahap ini kebutuhan dianalisis menggunakan informasi tentang komponen yang telah ditentukan. Kemudian dimodifikasi untuk mencerminkan komponen yang tersedia. Jika terdapat modifikasi yang tidak mungkin untuk diterapkan, proses analisis komponen dapat dilakukan kembali. .

3. Desain Sistem dengan *Reuse*

Selama fase ini, mendesain *framework* pada sistem atau *framework* yang tersedia akan digunakan kembali. Para desainer memperhitungkan komponen yang digunakan kembali dan mengatur *framework* untuk mengembangkannya.

4. Pengembangan dan Integrasi Perangkat Lunak

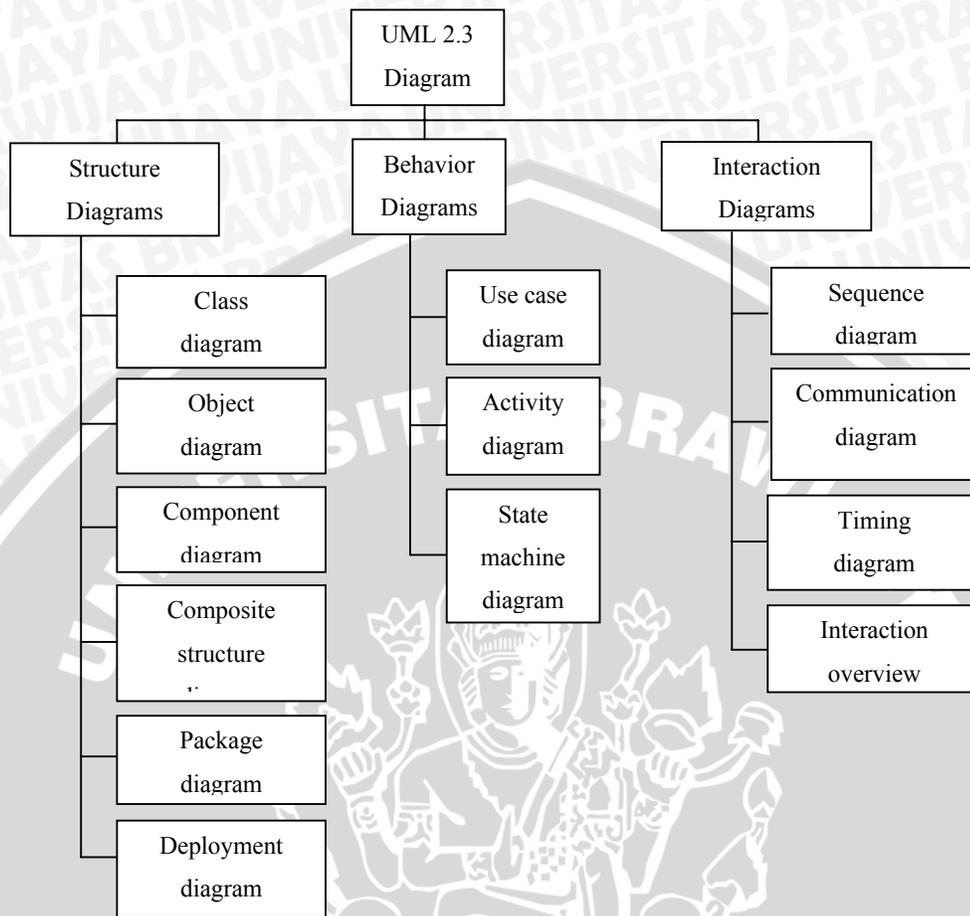
Perangkat lunak yang tidak dapat diperoleh secara eksternal maka akan dikembangkan dan komponen diintegrasikan untuk menciptakan sistem baru.

Reuse-oriented software engineering memiliki keuntungan yang jelas untuk mengurangi jumlah perangkat lunak yang akan dikembangkan sehingga mengurangi biaya dan resiko. Biasanya juga menyebabkan pengembangan perangkat lunak yang lebih cepat. Namun, kebutuhan yang tak terelakan dapat menyebabkan sistem tidak dapat memenuhi kebutuhan nyata pengguna (Sommerville,2011).

2.9 Unified Modelling Language

Unified Modelling Language(UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikkan, menggambarkan, membangun, dan mendokumentasikan dari sistem perangkat lunak (Rosa & Shalahudin,2011).

Structure diagrams yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.*Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.*Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem(Rosa & Shalahudin,2011).



Gambar 2. 3 Diagram UML
Sumber : (Rosa & Shalahudin,2011)

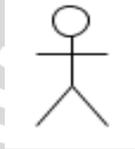
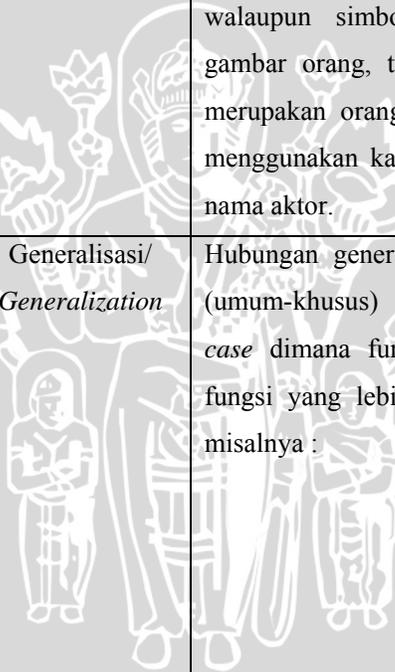
2.9.1 Use Case Diagram

Diagram *Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa & Shalahudin, 2011).

Syarat penamaan pada diagram *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Simbol aktor adalah gambar orang, namun aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor.

Tabel 2. 1 Keterangan Simbol-simbol Use Case

NO	GAMBAR	NAMA	KETERANGAN
1		Aktor / <i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2		Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya : Ubah data Mengelola Hapus data Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
3		Menggunakan / <i>include / uses</i>	Fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

	<p> <pre> <<include>> -----> <<uses>> </pre> </p>	<p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> :</p> <p>Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut :</p> <p style="text-align: center;">Validasi username</p> <p style="text-align: center;"> <pre> <<include>> </pre> </p> <p style="text-align: center;">Login</p> <p>Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut :</p> <p style="text-align: center;">Validasi user</p> <p style="text-align: center;"> <pre> <<include>> </pre> </p> <p style="text-align: center;">Ubah data</p> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
<p>4</p>	<p> <pre> -----> </pre> </p>	<p>Ekstensi/ <i>Extend</i></p> <p> <pre> <<extend>> </pre> </p> <p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan,</p>

			<p>misal :</p> <pre> graph TD A([Validasi username]) -.-> <<extend>> B([Validasi user]) C([Validasi idik jari]) -.-> <<extend>> B </pre> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
5		Asosiasi/ <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interksi dengan aktor.
6		<i>Sistem</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
7		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
8		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
9		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

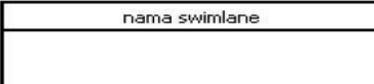
Sumber : (Rosa &Shalahudin,2011)

2.9.2 Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Digram aktivitas juga banyak digunakan untuk mendefenisikan hal-hal berikut:

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis system yang didefenisikan.
2. Urutan atau pengelompokkan tampilan dari sistem atau user intefasce dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan penjualan di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefenisikan kasus ujinya.

Gambar 2. 4 Simbol Diagram Aktivitas

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / decision 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / join 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
fork, 	digunakan utk menunjukkan kegiatan yg dilakukan secara paralel
join, 	digunakan utk menunjukkan kegiatan yg digabungkan

Sumber : (Rosa & Shalahudin,2011)

2.9.3 Class Diagram

Class diagram atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas

memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa & Shalahudin, 2011).

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas main

kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case*

kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4. Kelas yang diambil dari pendefinisian data

kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data

Jenis-jenis kelas dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan (Rosa & Shalahudin, 2011).

Cohesion dan *coupling* perlu diperhatikan dalam pendefinisian metode yang ada di dalam kelas. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain. *Coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah. Berikut adalah simbol-simbol yang ada pada kelas (Rosa & Shalahudin, 2011).

Tabel 2. 2 Keterangan Simbol-Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Kelas/ <i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2	 Nama_interface	Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4		Asosiasi berarah/ <i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5		Generalisasi/ <i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6		Kebergantungan <i>/ dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7		Agregasi/ <i>Aggregation</i>	Semua-bagian (<i>whole-part</i>).

Sumber : (Rosa &Shalahudin,2011)

2.9.4 Sequence Diagram

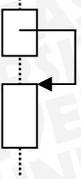
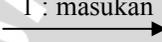
Sequence diagram atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Penggambaran diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa & Shalahudin, 2011).

Diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri. Banyaknya *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen.

Tabel 2. 3 Simbol-simbol Diagram Sekuen

NO	GAMBAR	NAMA	KETERANGAN
1	 <u>nama aktor</u>	Aktor/ <i>Actor</i> atau tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2		Garis hidup/ <i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3	<u>nama objek :</u> <u>nama kelas</u>	Objek/ <i>Object</i>	Menyatakan objek yang berinteraksi pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
5	<<create>>	Pesantipecreate	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
6	1 : nama_metode()	Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lin atau dirinya sendiri.

			 <p>1 : nama_metode()</p> <p>Arah panah mengarah pada objek yang memiliki operasi atau metode, karena ini memanggil operasi atau metode maka operasi atau metode yang dipanggil harus ada pada diagram kelas sesuai dengan objek yang berinteraksi.</p>
7	<p>1 : masukan</p> 	Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data masukan atau informasi ke objek lainnya, arah panah mengarah pada objek yang akan dikirim.
8	<p>1 : keluaran</p> 	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9	<p><<destroy>></p>	Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : (Rosa &Shalahudin,2011)

Penomoran pesan berdasarkan urutan interaksi pesan. Penggambaran letak pesan harus berurutan, pesan yang lebih atas dari lainnya adalah pesan yang berjalan terlebih dahulu(Rosa & Shalahudin,2011).

2.10 Pengujian (Testing)

Pengujian dimaksudkan untuk menunjukka bahwa program melakukan apa yang diharapkan untuk dilakukan dan menemukan kesalahan program sebelum

program benar-benar siap untuk digunakan. Proses pengujian mempunyai dua tujuan (Sommerville,2011):

1. Untuk mendemonstrasikan kepada *developer* dan pengguna bahwa perangkat lunak tersebut sudah memenuhi kebutuhan yang ada.
2. Untuk menemukan situasi ketika perangkat lunak melakukan kesalahan, hasil yang tidak diinginkan, atau tidak sesuai dengan kebutuhan yang diinginkan.

2.10.1 Teknik Pengujian

Pengujian perangkat lunak memerlukan perancangan kasus uji (*test case*) agar dapat menemukan kesalahan dalam waktu singkat dan usaha minimum. Berbagai macam metode perancangan kasus uji telah berevolusi. Metode-metode ini menyediakan *developer* pendekatan sistematis untuk pengujian. Terlebih lagi metode-metode ini menyediakan mekanisme yang dapat membantu memastikan kelengkapan dari pengujian dan menyediakan kemungkinan tertinggi untuk menemukan kesalahan-kesalahan dalam perangkat lunak (Pressman,2011). Teknik atau metode perancangan kasus uji yang digunakan adalah *black-box testing*.

2.10.1.1 White Box Testing

White box testing adalah teknik pengujian yang menguji berdasarkan jalur internal, struktur dan implementasi dari perangkat lunak yang sedang diuji. *White box testing* adalah teknik pengujian yang menggunakan struktur kontrol dari prosedur yang terdapat dalam perancangan untuk membuat kasus uji(Pressman,2011). Ada dua jenis pengujian yang termasuk *white box testing* yaitu basis path testing dan control structure testing.*White-box* dalam skripsi ini dilakukan dengan menggunakan basis path testing yang diusulkan pertama kali oleh Tom McCabe(Pressman,2011). *Black-Box Testing*

Black-box testing atau *behavioral testing* berfokus pada persyaratan fungsional perangkat lunak (Pressman,2011). Dengan demikian, pengujian *black-box* memungkinkan perencana perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk semua program. Pengujian *black-box* merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan.

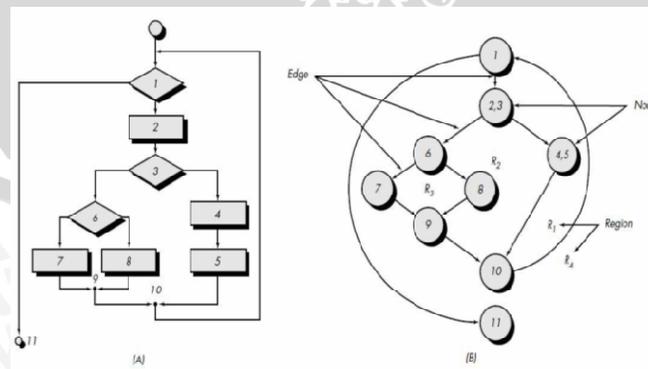
Pengujian *black-box* berusaha menemukan kesalahan dalam kategori berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *basis data* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi. Pada penelitian ini, pengujian yang menggunakan teknik *black box* adalah pengujian validasi sistem dan pengujian kompatibilitas.

1. Pengujian *Basis Path*

Sebelum metode basis path dapat digunakan, notasi sederhana untuk representasi aliran kontrol yang disebut diagram alir (flow graph) harus didefinisikan. Setiap representasi desain prosedural yang berupa flow chart dapat diterjemahkan ke dalam flow graph. Setelah flow graph didefinisikan maka harus ditentukan ukuran kompleksitasnya (cyclomatic complexity). Gambar 2.5 menunjukkan proses transformasi flow chart ke flow graph. Cyclomatic complexity adalah metrik perangkat lunak yang memberikan pengukuran kuantitatif terhadap kompleksitas logis suatu program.



Gambar 2.5 Transformasi flow chart (A) ke flow graph (B)
Sumber : (Pressman,2011)

Bila metrik ini digunakan dalam konteks metode pengujian basis path, maka nilai yang terhitung untuk cyclomatic complexity menentukan jumlah jalur independen (independent path) dalam basis set suatu program dan memberi batas atas bagi jumlah pengujian yang harus dilakukan untuk memastikan bahwa semua statement telah dieksekusi sedikitnya satu kali. Untuk menentukan cyclomatic complexity bisa dilakukan dengan beberapa cara, diantaranya (Pressman,2011):

1. Jumlah region pada flow graph sesuai dengan cyclomatic complexity.
2. Cyclomatic complexity $V(G)$, untuk grafik G adalah $V(G) = E - N + 2$, dimana E adalah jumlah edge, dan N adalah jumlah node.

$V(G) = P + 1$, dimana P adalah jumlah predicate node yaitu node yang merupakan kondisi (ada dua atau lebih edge akan keluar node ini).

Dari hasil nilai *cyclomatic complexity* terdapat hubungan dengan resiko dalam suatu prosedur. Gambar 2.6 akan menjelaskan hubungan *cyclomatic complexity* dengan resiko.

CC	Type of Procedure	Risk
1-4	A simple procedure	Low
5-10	A well structured and stable procedure	Low
11-20	A more complex procedure	Moderate
21-50	A complex procedure, alarming	High
>50	An error-prone, extremely troublesome, untestable procedure	Very high

Gambar 2. 6 Hubungan Cyclomatic Complexity dan Resiko
Sumber: (Cyclomatic Metrics,)

2. Pengujian Validasi

Pengujian validasi berfokus kepada sudut pandang pengguna dan data keluaran dari sistem yang dapat dikenali oleh pengguna. Pengujian validasi berhasil ketika fungsi dari perangkat lunak dalam perilakunya dapat dijelaskan dan sesuai dengan keinginan pengguna (Pressman,2011). Pengujian validasi merupakan pengujian untuk memenuhi kebutuhan fungsional dari sistem. Validasi perangkat lunak dicapai melalui sederetan pengujian *black box* yang memperlihatkan konformitas dengan persyaratan.

3. Pengujian Kompatibilitas

Pengujian kompatibilitas adalah pengujian dilakukan pada aplikasi untuk mengevaluasi kompatibilitas aplikasi dengan lingkungan komputasi (Pressman,2011). Pengujian kompatibilitas dilakukan dengan melihat kemampuan aplikasi untuk dapat ditampilkan dengan tidak mengurangi konten aplikasi pada lingkungan *platform* yang berbeda. Pengujian kompatibilitas menggunakan teknik *black box testing*. Pengujian kompatibilitas merupakan untuk memenuhi kebutuhan non-fungsional dari sistem.

2.10.1.2 Pengujian Akurasi

Pengujian akurasi dilakukan untuk mengetahui kebenaran dari algoritma tersebut. Pengujian dilakukan dengan mencocokkan hasil perhitungan manual dengan hasil rekomendasi sistem. Pengujian akurasi merupakan pengujian untuk memenuhi kebutuhan fungsional dari sistem. Perhitungan untuk pengujian akurasi dapat dijabarkan dengan *accuracy* sebagai berikut (Tan,Steinbach,& Kumar,2006):

$$accuracy = \frac{N_c}{N}$$

(2-4)

Dimana:

- N_c : Number of positive instances covered by rule
- N : Number of instances covered by rule

2.10.1.3 User Acceptance Testing

User Acceptance Testing (UAT) digunakan untuk mendapatkan konfirmasi atau kepastian melalui *trial* atau *review* dari pihak yang terkait sesuai dengan *requirement* yang telah disetujui sebelumnya. Tahap pengujian UAT terdiri dari dua faktor, yaitu : *perceived ease of use* dan *perceived usefulness*. Kedua pengujian tersebut menitik beratkan kepada kemudahan serta kenyamanan pengguna dalam menggunakan aplikasi yang telah disiapkan (Segar & Grover,1993).

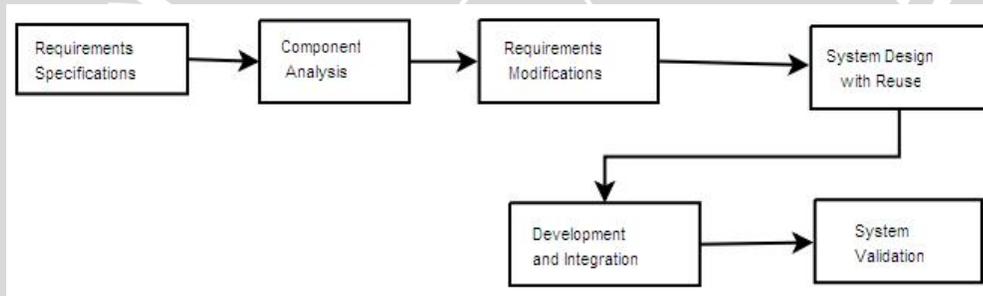
Perceived ease of use adalah suatu metode pengujian yang menekankan konsep kemudahan fungsi dari suatu perangkat lunak (Segar & Grover,1993). Jenis pengujian ini merujuk pada sejauh mana seseorang percaya bahwa menggunakan sistem tertentu akan menjadi jauh lebih mudah (Davis,1989). Sedangkan *perceived usefulness* merujuk pada suatu pengujian yang lebih menekankan kepada apakah aplikasi atau perangkat lunak yang telah dibuat akan berguna dalam penerapannya (Segar & Grover,1993). UAT merupakan pengujian untuk memenuhi kebutuhan non-fungsional dari sistem.

UNIVERSITAS BRAWIJAYA



BAB III METODOLOGI PENELITIAN

Bab ini akan menjelaskan tentang langkah-langkah yang dilakukan dalam melakukan penelitian berdasarkan tahap-tahap pada model *reuse development*. Peneliti menerapkan model *reuse* pada pengembangan perangkat lunak karena pemodelan ini membuat proses pengembangan lebih cepat. Hal ini disebabkan model ini menggunakan komponen yang sudah ada untuk dijadikan sebuah sistem yang baru. Kemudian, pemodelan *reuse* memungkinkan penggunaan tools yang tepat dapat mengurangi jumlah basis code yang dibutuhkan saat menulis program sehingga dapat meningkatkan efisiensi dan kualitas. Diagram alir langkah-langkah dalam penelitian ini ditunjukkan pada gambar 3.1.



Gambar 3.1 Alur Penelitian dengan *Reuse Oriented*
Sumber : [metodologi penelitian]

3.1 Requirement Specifications

Tahap *Requirement Specifications* atau spesifikasi kebutuhan merupakan tahap yang bertujuan untuk mendapatkan dan menjelaskan kebutuhan yang diperlukan dalam mengembangkan sistem ini. Tahap ini dapat disamakan dengan *requirement analysis* atau analisis kebutuhan pada proses perangkat lunak lainnya. Langkah awal pada tahap ini adalah melakukan studi literatur. Metode ini digunakan untuk mendapatkan dasar teori yang digunakan untuk pengembangan sistem. Sumber atau referensi yang digunakan meliputi buku, jurnal, laporan penelitian, artikel ilmiah, dan bantuan dari mesin pencari (*search engine*). Teori-teori tersebut adalah sebagai berikut :

1. Sistem Rekomendasi

2. Metode *Weighted Product* (WP)
3. *Responsive Web Design*
4. *Framework Codeigniter*
5. *Reuse-oriented Software Engineering*
6. UML (*Unified Modeling Language*)
 - a. *Use case Diagram*
 - b. *Activity Diagram*
 - c. *Class Diagram*
 - d. *Sequence Diagram*
7. Pengujian
 - a. *Pengujian Basis Path*
 - b. *Pengujian Validasi*
 - c. *Pengujian Akurasi Sistem Rekomendasi*
 - d. *Pengujian Kompatibilitas*
 - e. *User Acceptance Testing*
 - *Perceived Ease of Use*
 - *Perceived Usefulness*

Setelah tahap studi literatur, penentuan kebutuhan untuk membangun sistem rekomendasi dilakukan. Teknik yang digunakan untuk penentuan kebutuhan adalah teknik observasi. Observasi dilakukan pada *website review* restoran seperti *zomato*, *openrice*, dan *dine in*, serta aplikasi-aplikasi rekomendasi kuliner pada beberapa *platform mobile*. Selain itu, pengambilan data sampel juga dilakukan pada tahap *requirement specifications*. Data sampel meliputi data restoran atau rumah makan di Kota Malang. Data rumah makan seharusnya didapatkan dari para pengelola rumah makan yang menginputkan usahanya melalui sistem ini. Namun, untuk membantu perhitungan rekomendasi maka data tersebut didapatkan melalui dinas pariwisata Kota Malang, informasi restoran pada *website id.openrice.com*, dan observasi langsung di beberapa restoran di Kota Malang.

Setelah penentuan kebutuhan telah ditetapkan, kebutuhan-kebutuhan tersebut dikelompokkan dan dianalisis menggunakan *object oriented analysis* dengan bahasa pemodelan UML (*Unified Modelling Language*). Pengelompokkan

kebutuhan menjadi dua kategori, yaitu : kebutuhan fungsional dan kebutuhan non-fungsional.

3.1.1 Analisis Data

Analisis data bertujuan untuk mendapatkan struktur penyimpanan data yang dibutuhkan. Struktur penyimpanan data pada sistem ini disusun berdasarkan analisis data sebagai berikut:

1. Data rumah makan yang ada di Kota Malang.
2. Data pengelola rumah makan yang terdiri dari email, nama, nomor telepon, dan *password* untuk melakukan *login* di sistem
3. Data untuk perhitungan rekomendasi yang meliputi : jenis makanan, range harga, lokasi restoran, dan fasilitas tiap rumah makan di Kota Malang.
4. Data pelengkap restoran seperti : alamat website, jam buka restoran, jenis pembayaran, deskripsi singkat restoran, estimasi waktu tunggu, dll.
5. Data menu makanan dari tiap rumah makan di Kota Malang
6. Data berita terkait rumah makan di Kota Malang.
7. Data promo tiap rumah makan di Kota Malang.
8. Data komentar terhadap rumah makan di Kota Malang.
9. Data *administrator* yang terdiri dari *username*, *password*, nama *administrator*, dan *e-mail*.
10. Data kategori fasilitas dan jenis makanan yang terdiri dari id, dan nama kategori fasilitas atau jenis makanan.

3.1.2 Daftar Kebutuhan

Daftar kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Spesifikasi kebutuhan fungsional ditunjukkan pada tabel 3.1

Tabel 3.1 Daftar Kebutuhan Fungsional

SRS No	Tabel Kebutuhan	Aktor	Use Case
SRS_001_01	Sistem harus menyediakan fasilitas untuk melakukan proses registrasi bagi pengelola restoran.	Pengelola rumah makan	Register
SRS_002_01	Sistem harus menyediakan fasilitas untuk mengatur	<i>Administrator</i>	Kelola akun admin

	data akun administrator.		
SRS_002_02	Sistem harus menyediakan fungsi untuk mengatur data fasilitas.	<i>Administrator</i>	Kelola kategori fasilitas
SRS_002_03	Sistem harus menyediakan fungsi untuk mengolah data jenis makanan.	<i>Administrator</i>	Kelola kategori makanan
SRS_003_01	Sistem harus menyediakan fungsi reset password bagi pengelola rumah makan maupun <i>administrator</i> .	<i>Administrator</i> dan Pengelola restoran	Reset password
SRS_004_01	Sistem harus menyediakan fungsi untuk mengatur data pengelola dan data rumah makan yang ada pada sistem, mulai dari lihat info restoran, menambahkan restoran baru, memperbaiki informasi restoran, dsb.	<i>Administrator</i> ,peng elola restoran, dan konsumen/calon pelanggan.	Kelola akun dan data restoran
SRS_004_02	Sistem harus menyediakan fungsi untuk mengatur menu restoran mulai dari menambahkan menu hingga menghapusnya.	<i>Administrator</i> , pengelola restoran, dan konsumen/calon pelanggan	Kelola menu restoran
SRS_004_03	Sistem harus menyediakan fungsi untuk mengatur promo restoran mulai dari menambahkan promo, memperbaiki promo, dan menghapus promo.	<i>Administrator</i> , pengelola restoran, dan konsumen/calon pelanggan	Kelola promo restoran
SRS_004_04	Sistem juga harus menyediakan fasilitas mengolah data komentar mulai dari menambahkan	<i>Administrator</i> , pengelola restoran, dan konsumen/calon	Kelola komentar

	komentar dan menghapus komentar.	pelanggan	
SRS_004_05	Sistem juga harus menyediakan fasilitas mengolah berita mulai dari menambahkan berita, memperbarui berita, dan menghapus berita.	<i>Administrator</i> , pengelola restoran, dan konsumen/calon pelanggan	Kelola berita
SRS_005_01	Sistem harus menyediakan fasilitas untuk melakukan pencarian restoran berdasarkan nama restoran atau kategori makanan.	Konsumen/calon pelanggan	Cari restoran
SRS_005_02	Sistem harus menyediakan fasilitas untuk melakukan pencarian rekomendasi berdasarkan input user .	Konsumen/calon pelanggan	Cari rekomendasi restoran
SRS_005_03	Sistem harus menyediakan fasilitas untuk memberikan petunjuk arah ke restoran yang akan dituju.	Konsumen/calon pelanggan	Lihat petunjuk arah

Sumber: [Metodoogi Peneitian]

Setelah mendefinisikan kebutuhan fungsional sistem ini, maka kebutuhan non-fungsional juga perlu untuk didefinisikan. Kebutuhan non-fungsional merupakan kebutuhan pendukung untuk sistem yang akan dijalankan. Kebutuhan non-fungsional sistem rekomendasi ini dapat dilihat pada tabel 3.2.

Tabel 3. 2 Daftar Kebutuhan Non-Fungsional

Parameter	Deskripsi Kebutuhan
<i>Compatibility</i>	Sistem harus dapat dijalankan di berbagai lingkungan <i>platform</i> yang berbeda. Pada penelitian ini, <i>compatibility</i> dilakukan pada beberapa device yang berbeda.
<i>Usability</i>	Sistem harus dapat dioperasikan dengan mudah oleh pengguna.

<i>Accuracy</i>	Rekomendasi yang dihasilkan oleh sistem sudah sama dengan perhitungan manual dengan metode yang sama.
-----------------	---

Sumber:[Metodologi Penelitian]

3.2 Component Analysis

Component Analysis atau Analisis komponen dilakukan untuk mencari komponen-komponen yang digunakan untuk pengembangan sistem rekomendasi. Komponen-komponen yang diperlukan dalam mengimplementasikan sesuai spesifikasi kebutuhan sistem rekomendasi kuliner adalah *framework* dan API peta.

Framework untuk mengembangkan sistem rekomendasi menggunakan *design pattern* MVC (Model View Controller) yaitu : *Codeigniter*. Kemudian, untuk *front-end* sistem rekomendasi, menggunakan *framework* CSS dari twitter, yaitu : Bootstrap. Selain *framework*, API peta juga dibutuhkan pada pengembangan sistem rekomendasi kuliner untuk menambah fungsionalitas fitur navigasi di sistem rekomendasi ini. API yang digunakan adalah Google Map API.

Komponen-komponen yang akan digunakan dalam sistem rekomendasi ini dijelaskan pada tabel 3.3.

Tabel 3.3 Daftar Komponen

No	Nama Komponen	Fungsi Komponen
1	<i>Codeigniter</i> (CI)	<i>Code igniter</i> digunakan untuk mengimplementasikan keseluruhan sistem rekomendasi kuliner. <i>Codeigniter</i> menggunakan <i>design pattern</i> MVC (Model,View,Controller) yang nantinya akan mempercepat proses pengerjaan sistem rekomendasi kuliner
2	Bootstrap	<i>Bootstrap</i> akan digunakan sebagai <i>framework front-end</i> dari sistem rekomendasi kuliner.
3	Google Maps API	Google Maps API digunakan pada saat menampilkan lokasi rumah makan dan sebagai navigasi dari posisi konsumen ke lokasi restoran.

Sumber:[Metodologi Penelitian]

3.3 Requirements Modifications

Requirements Modififcations atau modifikasi kebutuhan merupakan tahap dimana setiap kebutuhan dianalisis menggunakan informasi tentang komponen-komponen yang telah ditentukan, apakah sudah dapat mengakomodasi seluruh kebutuhan yang dibutuhkan sistem. Jika semua komponen sudah dapat mengakomodasi keseluruhan sistem, maka modifikasi kebutuhan tidak perlu dilakukan. Akan tetapi, jika terdapat komponen yang tidak mampu mengakomodasi keseluruhan kebutuhan sistem, maka dapat dilakukan pemodifikasian kebutuhan dapat dilakukan dengan komponen yang sudah ditentukan dan analisis komponen dapat dilakukan kembali untuk mencari solusi tersebut.

Pada pengembangan sistem rekomendasi ini, modifikasi kebutuhan tidak dilakukan karena komponen-komponen yang ditentukan sudah dapat mengakomodasi seluruh kebutuhan sistem rekomendasi. Tabel 3.4 akan menjelaskan integrasi komponen dengan kebutuhan sistem.

Tabel 3. 4 Integrasi Komponen dengan Kebutuhan Fungsional

Tabel Kebutuhan	Komponen	Keterangan
Sistem harus menyediakan fasilitas untuk melakukan proses registrasi bagi pengelola restoran.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem harus menyediakan fasilitas untuk mengatur data akun administrator.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem harus menyediakan fungsi untuk mengatur data fasilitas.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem harus menyediakan fungsi untuk mengolah data jenis makanan.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem harus menyediakan fungsi reset password bagi pengelola rumah makan maupun <i>administrator</i> .	Codeigniter	Mampu mengakomodasi seluruh kebutuhan

Sistem harus menyediakan fungsi untuk mengatur data pengelola dan data rumah makan yang ada pada sistem, mulai dari lihat info restoran, menambahkan restoran baru, memperbarui informasi restoran, dsb.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem harus menyediakan fungsi untuk mengatur menu restoran mulai dari menambahkan menu hingga menghapusnya.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem harus menyediakan fungsi untuk mengatur promo promo restoran mulai dari menambahkan promo, memperbarui promo, dan menghapus promo.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem juga harus menyediakan fasilitas mengolah data komentar mulai dari menambahkan komentar dan menghapus komentar.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem juga harus menyediakan fasilitas mengolah berita mulai dari menambahkan berita, memperbarui berita, dan menghapus berita.	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem harus menyediakan fasilitas untuk melakukan pencarian restoran	Codeigniter	Mampu mengakomodasi seluruh kebutuhan

berdasarkan nama restoran atau kategori makanan.		
Sistem harus menyediakan fasilitas untuk melakukan pencarian rekomendasi berdasarkan input user .	Codeigniter	Mampu mengakomodasi seluruh kebutuhan
Sistem harus menyediakan fasilitas untuk memberikan petunjuk arah ke restoran yang akan dituju.	Codeigniter dan Google Map API	Mampu mengakomodasi seluruh kebutuhan

Sumber:[Metodologi Penelitian]

Tabel 3. 5 Integrasi Komponen dengan Kebutuhan Non-Fungsional

Parameter	Komponen	Keterangan
<i>Compatibility</i>	Bootstrap	Mampu mengakomodasi seluruh kebutuhan
<i>Usability</i>	Bootstrap	Mampu mengakomodasi seluruh kebutuhan
<i>Accuracy</i>	Codeigniter dan Google Map API	Mampu mengakomodasi seluruh kebutuhan

Sumber:[Metodologi Penelitian]

3.4 System Design with Reuse

Langkah selanjutnya adalah merancang *framework* berdasarkan *design pattern* yang digunakan. Tahap ini menggunakan *object oriented design* yaitu menggunakan pemodelan UML dengan bentuk diagram *use case*, diagram aktivitas, diagram kelas, diagram interaksi, perancangan basis data, perancangan algoritma dan perancangan antar muka.

3.4.1 Perancangan Use Case

Perancangan diagram *use case* bertujuan untuk menggambarkan fungsi-fungsi yang ada pada sistem dan interaksi antara aktor dengan sistem. Aktor atau pengguna dari sistem ini adalah pengelola rumah makan, *administrator*, dan

konsumen atau pelanggan. Aktivitas apa saja yang dapat dilakukan oleh aktor akan dijelaskan pada bab selanjutnya.

3.4.2 Perancangan Aktivitas

Perancangan diagram aktivitas bertujuan untuk menggambarkan alur kerja sistem secara berurutan dan menganalisis *use case* yang telah dibuat dengan penggambaran aktivitas yang dibutuhkan, penggambaran algoritma yang berurutan, dan pemodelan aplikasi dengan proses paralel.

3.4.3 Perancangan Kelas

Perancangan kelas-kelas dalam sistem dimodelkan dalam *class diagram*. Diagram kelas menggambarkan relasi dan komponen-komponen yang digunakan. Diagram kelas juga dapat digunakan untuk mengidentifikasi kelas-kelas serta paket-paket yang ada pada sistem.

3.4.4 Perancangan Interaksi

Perancangan interaksi antar objek dalam sistem yang telah diidentifikasi akan dimodelkan dalam *sequence diagram* dimana diagram tersebut akan menggambarkan interaksi antar objek yang disusun dalam urutan waktu.

3.4.5 Perancangan Basis Data

Perancangan basis data bertujuan untuk menggambarkan manajemen data yang akan digunakan sistem. Perancangan basis data terdiri dari dua data model, yaitu : *Conceptual (High Level) Data Model* dan *Physical (Low Level) Data Model*. *Conceptual data model* lebih merujuk kepada konsep yang berkaitan dengan pemakai terhadap data. Sedangkan *Physical data model* menerangkan detail bagaimana data disimpan di dalam komputer. Perancangan basis data dengan model *conceptual data model* akan digambarkan melalui *entity relationship diagram (ERD)*.

3.4.6 Perancangan Algoritma

Perancangan algoritma bertujuan untuk menjelaskan manualisasi metode yang akan diimplementasikan. Perancangan algoritma juga akan menjelaskan tahap-tahap perhitungan dengan metode *weighted product* yang disertai dengan studi kasus.

3.4.7 Perancangan Antarmuka

Kebutuhan-kebutuhan desain antarmuka untuk pembuatan sistem rekomendasi ini adalah sebagai berikut :

1. Sistem rekomendasi akan dikembangkan dengan tampilan antarmuka yang mudah dimengerti (*user friendly*) oleh pengguna.
2. Kompatibilitas sistem harus sesuai dengan *browser* dari beberapa *device*.

3.5 Development and Integration

Tahap ini merupakan tahap pengembangan sistem dan integrasi antara komponen-komponen yang sudah ditentukan. Pengembangan sistem ini adalah sebagai berikut :

1. Pembuatan sistem rekomendasi kuliner ini sesuai dengan perancangan sistem yang telah dibuat dengan menggunakan.
2. Pengembangan sistem dibuat dengan menggunakan bahasa PHP dengan komponen *framework Codeigniter* dan *bootstrap*.
3. Komponen API Peta diimplementasikan pada sistem dengan menggunakan Google Map API.
4. Penerapan perhitungan *weighted product* diimplementasikan menggunakan PHP berdasarkan rumus yang telah dipaparkan di bab sebelumnya.
5. Pembuatan basis data sesuai dengan perancangan basis data.
6. Data-data rumah makan di Kota Malang dimasukkan ke dalam *database*.
7. Pembuatan *interface* sistem dibuat dengan berbasis *web* responsif sesuai dengan perancangan yang telah dibuat.

3.6 System Validation

System Validation bertujuan untuk memastikan bahwa sistem yang dibuat telah berjalan sesuai dengan kebutuhan. Pengujian yang dilakukan untuk sistem rekomendasi ini meliputi : pengujian *basis path*, pengujian validasi sistem dengan menggunakan teknik *black box testing*, pengujian akurasi sistem rekomendasi dengan membandingkan hasil manualisasi dengan hasil rekomendasi sistem, pengujian kompatibilitas dengan menjalankan sistem pada berbagai *device*, dan

terakhir adalah UAT (*user acceptance testing*) yang berguna untuk mengetahui *feedback user* terhadap sistem yang telah dibuat.

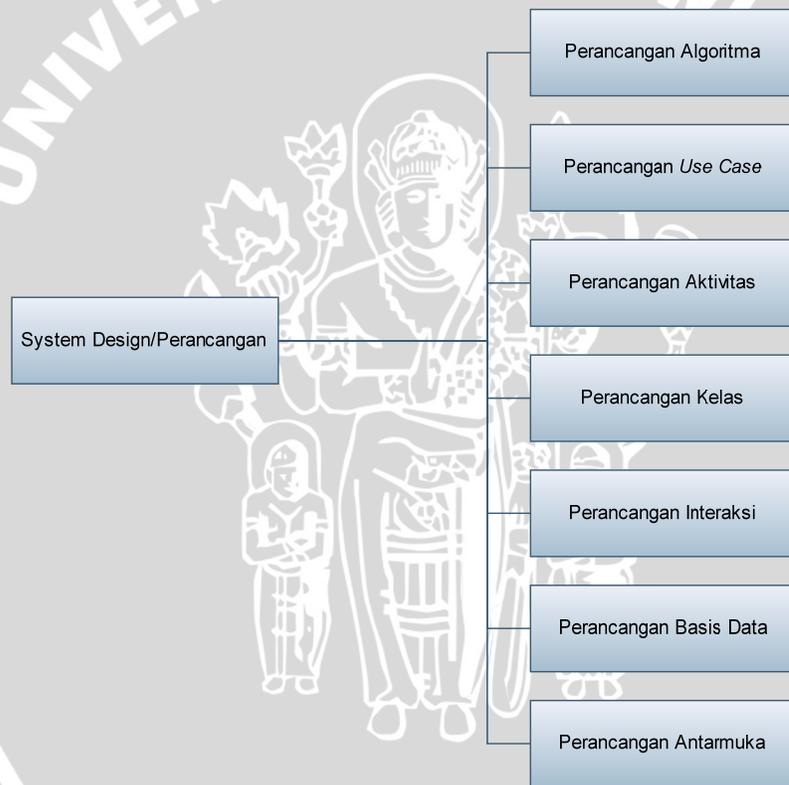
Kemudian, tahap terakhir dalam penelitian ini adalah menyimpulkan penelitian yang dilakukan untuk menjawab rumusan masalah yang sudah diterapkan pada awal penelitian. Setelah pengambilan kesimpulan, maka terdapat saran yang dimaksudkan untuk memperbaiki kesalahan dari sistem ataupun memberikan masukan untuk pengembangan sistem selanjutnya.

UNIVERSITAS BRAWIJAYA



BAB IV PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas mengenai perancangan perangkat lunak dengan komponen dan implementasinya. Perancangan yang telah dibuat, selanjutnya akan diimplementasikan ke dalam bahasa pemrograman yang dikehendaki. Proses perancangan meliputi perancangan algoritma, perancangan *class diagram*, *sequence diagram*, *activity diagram*, perancangan basis data, dan perancangan *interface*. Tahap-tahap perancangan yang dilakukan digambarkan pada gambar 4.1.



Gambar 4. 1 Diagram Perancangan
Sumber:[Perancangan]

4.1 Perancangan Algoritma

Sebelum membahas perhitungan rekomendasi dengan menggunakan *weighted product*, akan dijelaskan terlebih dahulu sistem yang akan dikembangkan. Sistem yang akan dibangun pada skripsi ini adalah sebuah sistem rekomendasi kuliner khususnya di Kota Malang dengan berbasis *web*

menggunakan metode *weighted product*. Sistem ini akan memberikan rekomendasi tempat kuliner sesuai dengan kriteria dan bobot yang diinputkan konsumen. Kriteria tersebut adalah jenis makanan yang diinginkan, harga makanan, lokasi kuliner, dan ketersediaan fasilitas.

Selain itu, Sistem ini juga akan memberikan informasi detail restoran. Berikut ini merupakan diagram alir proses rekomendasi pada sistem rekomendasi kuliner di Kota Malang dengan berbasis web menggunakan metode *weighted product* ditunjukkan pada gambar 4.2.

Pengelola restoran juga aktif terlibat pada sistem rekomendasi kuliner ini. Pengelola restoran dapat mendaftarkan restoran yang dimiliki sehingga sistem kuliner ini juga berfungsi sebagai media promosi dan mengetahui sejauh mana tingkat kepuasan konsumen terhadap restoran mereka. Fitur-fitur yang akan didapat oleh pengelola rumah makan akan dibahas pada subbab selanjutnya. Terakhir, agar sistem dapat berjalan dengan baik maka terdapat *administrator* yang akan memantau keadaan sistem rekomendasi ini

Berikut ini adalah kriteria yang digunakan pada sistem rekomendasi:

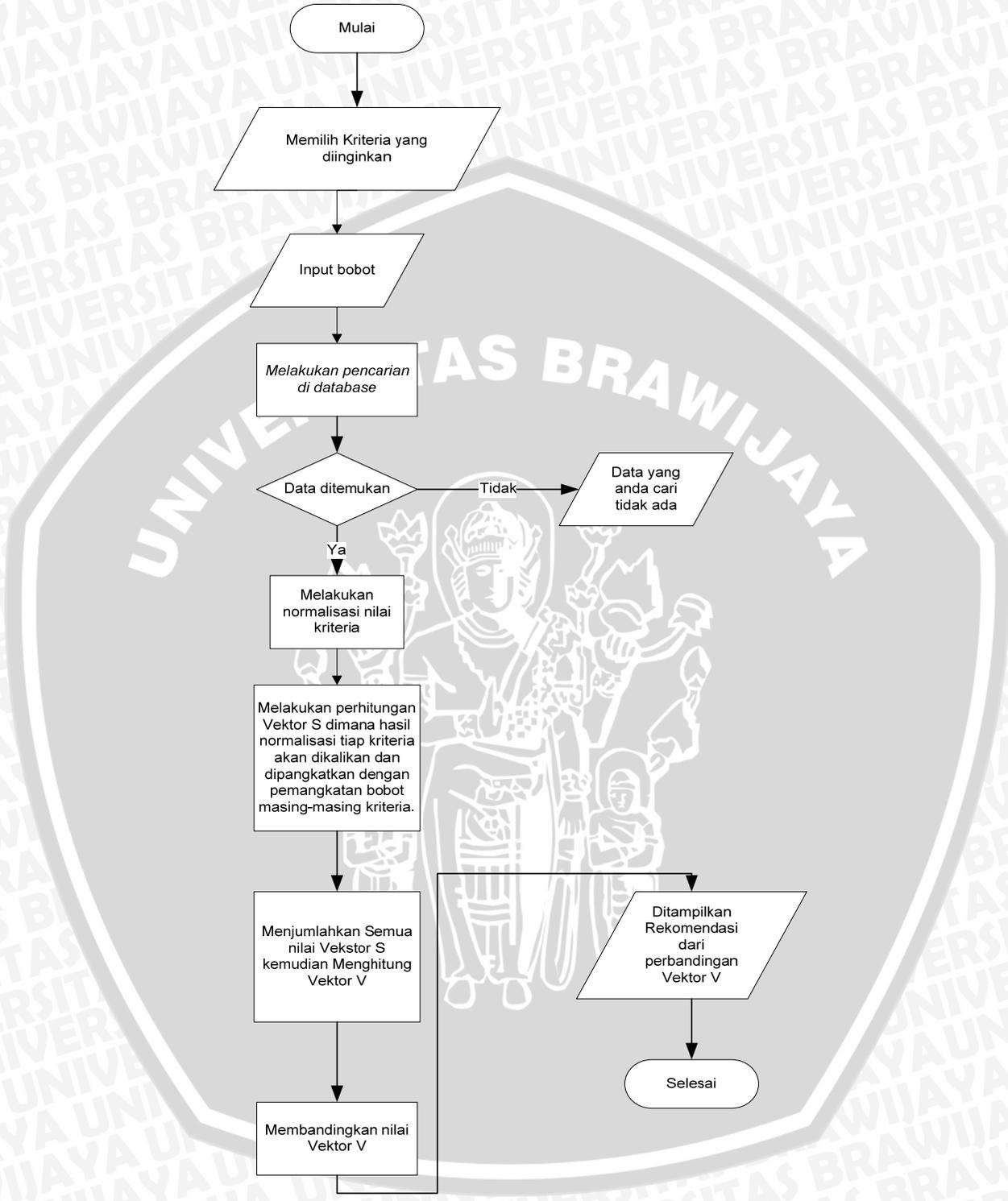
a. Jenis Makanan (*Type of Food*)

Jenis makanan merupakan salah satu faktor penting pada putusan konsumen dalam memilih restoran (Cullen, 2012).

b. Harga (*Price*)

Dari sudut pandang konsumen, harga adalah sesuatu yang diberikan oleh konsumen untuk memiliki barang atau jasa. Harga adalah faktor penting pada keputusan konsumen dalam membeli dan sering dianggap sebagai indikator kualitas (Mitra, 1995). Berikut adalah range yang digunakan pada sistem rekomendasi kuliner (didapatkan dari situs *review* restoran id.openrice.com).

- Sangat Murah □ 15.001 IDR
- Murah (15.001- 30.000 IDR)
- Sedang (30.001 – 50.000 IDR)
- Mahal (50.001 – 100.000 IDR)
- Sangat Mahal (□100.000 IDR)



Gambar 4. 2 Diagram Alir Cara Kerja Sistem Rekomendasi Kuliner dengan Metode *Weighted Product*
 Sumber : [Perancangan]

c. Lokasi (*Location of Restaurant*)

Lokasi restoran adalah kriteria/fitur yang dapat mempengaruhi perilaku dan kepuasan konsumen (Hyun, 2010). Range yang digunakan adalah sebagai berikut (didapatkan dari *Aplikasi untuk Panduan Wisata Kuliner dengan SAW pada Android Mobile*) :

- Dekat (< 1 km)
- Sedang (1-2 km)
- Jauh (> 2)

d. Kualitas Pelayanan (*Quality of Service*)

Kualitas pelayanan (*service quality*) merujuk pada putusan konsumen berdasarkan superiority produk dan lebih mengarah pada *attitude*. Kualitas pelayanan merupakan salah satu faktor penting dalam memilih restoran (Jamal,2009). Faktor yang mempengaruhi Kualitas service merujuk dari faktor ketersediaan fasilitas berdasarkan *Measuring Perceived Service Quality Using servqual: A Case Study of the Croatian Hotel Industry* dengan pilihan dari *website review restaurant*, www.dine.com, dan id.openrice.com,

a) Ketersediaan fasilitas

- Wi-fi
- Kamar Mandi
- Tempat Ibadah
- Parkir
- Delivery
- Smoking Area
- dll

Tabel 4. 1 Daftar Nilai Kriteria

Kriteria	Nilai	Keterangan
Range Harga	1	Any
	2	Sangat Murah
	3	Murah
	4	Sedang
	5	Mahal
	6	Sangat Mahal
Lokasi	1	Any
	2	Dekat
	3	Sedang

	4	Jauh
Ketersediaan Fasilitas	-	Jumlah Fasilitas yang dimiliki oleh rumah makan

Sumber : [Perancangan]

Studi Kasus :

Terdapat konsumen yang akan mencari rekomendasi rumah makan dengan kriteria jenis makanan japanese, dengan harga sedang, lokasi dekat, dan memiliki fasilitas *wi-fi* dan kamar mandi. Bobot(*) yang diberikan untuk masing-masing kriteria adalah sebagai berikut :

- W1 (Harga) = 5
- W2 (Lokasi) = 4
- W3 (Fasilitas) = 4

(*) : bobot yang diberikan adalah dari skala 1-5.

Tabel 4. 2 Keterangan Bobot Kriteria

Bobot	Keterangan
1	Sangat Penting
2	Penting
3	Cukup Penting
4	Kurang Penting
5	Tidak Penting

Sumber: [Perancangan]

Berikut adalah langkah-langkah melakukan perhitungan rekomendasi dengan *weighted product* :

1. Pencarian data pada *database* sesuai dengan permintaan *user*

Tabel 4. 3 Data Rumah Makan Yang Dicari Pengguna

Nama	C1	C3
Sugoitei	4	Wi-fi, kamar mandi, pembayaran dengan kartu kredit
Osaka Sushi	3	Kamar mandi
Hoka-Hoka bento	2	Kamar mandi
Saboten	3	Wi-fi, kamar mandi

Sumber:[Perancangan]

C1 = harga

C3 = Fasilitas

- Setelah itu, mencari jarak dari lokasi *user* ke rumah makan tersebut dengan menggunakan bantuan rumus *haversine*.

Tabel 4. 4 Data Rumah Makan yang Dicari

Nama	C1	C3	C2
Sugoitei	4	Wi-fi, kamar mandi, pembayaran dengan kartu kredit	4
Osaka Sushi	3	Kamar mandi	3
Hoka-Hoka bento	2	Kamar mandi	2
Saboten	3	Wi-fi, kamar mandi	4

Sumber: [Perancangan]

C2 = Jarak

- Sebelumnya akan dilakukan normalisasi data terlebih dahulu dengan rumus sebagai berikut.

C1, C2

$$= \frac{\text{nilai maksimal kriteria} - |\text{nilai kriteria yang dicari user} - \text{nilai yang dimiliki tempat makan}|}{\text{nilai maksimal kriteria}}$$

$$C3 = \frac{\text{Jumlah fasilitas yang cocok}}{\text{jumlah failitas yang diminta}}$$

Sehingga menjadi,

a. Harga

- $V1 = \frac{6-|4-4|}{6} = 1$
- $V2 = \frac{6-|4-3|}{6} = 0.8$
- $V3 = \frac{6-|4-4|}{6} = 1$
- $V14 = \frac{6-|4-3|}{6} = 0.8$

b. Lokasi

- $V1 = \frac{4-|2-4|}{4} = 0.33$
- $V1 = \frac{4-|2-3|}{4} = 0.67$

- $V1 = \frac{4-|2-2|}{4} = 1$
- $V1 = \frac{4-|2-4|}{4} = 0.33$

c. Fasilitas

- $V1 = \frac{2}{2} = 1$
- $V2 = \frac{1}{2} = 0.5$
- $V3 = \frac{1}{2} = 0.5$
- $V4 = \frac{2}{2} = 1$

Tabel 4. 5 Hasil Normalisasi

Nama	C1	C3	C2
Sugoitei	1	1	0.33
Osaka Sushi	0.8	0.5	0.67
Hoka-Hoka bento	1	0.5	1
Saboten	0.8	1	0.33

Sumber:[Perancangan]

4. Melakukan perbaikan bobot

$$W1 = \frac{5}{5 + 4 + 4} = \frac{5}{13} = 0,38$$

$$W2 = \frac{4}{5 + 4 + 4} = \frac{4}{13} = 0,30$$

$$W3 = \frac{4}{5 + 4 + 4} = \frac{4}{13} = 0,30$$

5. Langkah selanjutnya adalah menghitung vektor S, dimana data yang ada akan dikalikan tetapi sebelumnya dilakukan pemangkatan dengan bobot dari masing-masing kriteria.

$$S1 = 1^{0.38} \times 1^{0.30} \times 0.33^{0.30} = 0.71705$$

$$S2 = 0.8^{0.38} \times 0.5^{0.30} \times 0.67^{0.30} = 0.66174$$

$$S3 = 1^{0.38} \times 0.5^{0.30} \times 1^{0.30} = 0.81225$$

$$S4 = 0.8^{0.38} \times 1^{0.30} \times 0.33^{0.30} = 0.65876$$

6. Setelah masing-masing vektor S didapat nilainya, langkah selanjutnya adalah menjumlahkan seluruh S untuk menghitung vektor V. Langkahnya sebagai berikut.

$$V1 = \frac{0.71705}{2.8498} = 0.25161$$

$$V2 = \frac{0.66174}{2.8498} = 0.23220$$

$$V3 = \frac{0.81225}{2.8498} = 0.28502$$

$$V4 = \frac{0.65876}{2.84985} = 0.23115$$

Kemudian hasil perhitungan vektor V akan dibandingkan. Alternatif yang memiliki hasil V lebih besar maka akan berada pada posisi paling atas saat pencarian rekomendasi. Dari studi kasus di atas maka rekomendasi yang didapat adalah V3 (Hoka-hoka Bento), V1 (Sugoitei), V2 (Osaka Sushi), dan V4 (Saboten).

4.2 Perancangan Use Case

Kebutuhan fungsional yang telah dibuat akan dimodelkan dalam diagram *use case*. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Sebelum membuat *use case diagram*, maka identifikasi aktor dilakukan.

4.2.1 Identifikasi Aktor

Tahap ini bertujuan untuk mengetahui siapa saja aktor yang terlibat pada sistem yang akan dibuat. Pada tabel 4.6 memperlihatkan bahwa terdapat tiga aktor yang terlibat pada sistem.

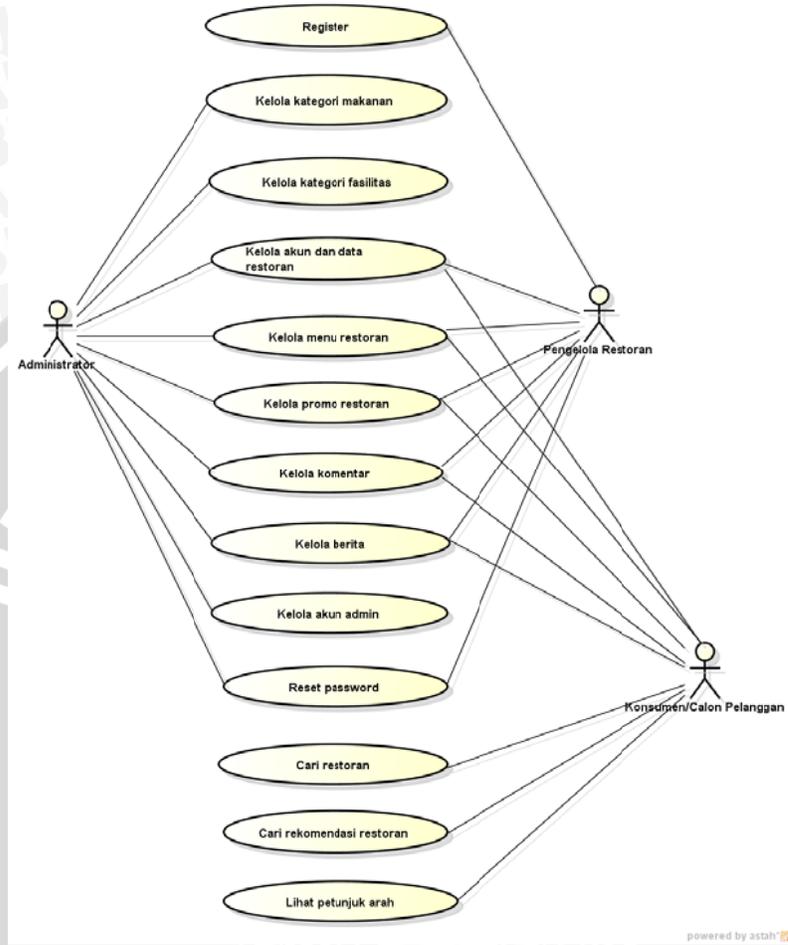
Tabel 4. 6 Deskripsi Aktor

Aktor	Deskripsi Aktor
Konsumen	Konsumen merupakan aktor yang menggunakan sistem rekomendasi kuliner tanpa harus melakukan proses autentifikasi.
Pengelola restoran	Pengelola Rumah makan merupakan aktor pengguna sistem rekomendasi dimana aktor tersebut merupakan orang yang bertanggung jawab atas restoran tersebut.
Administrator	Administrator merupakan aktor yang dapat memantau dan memelihara sistem.

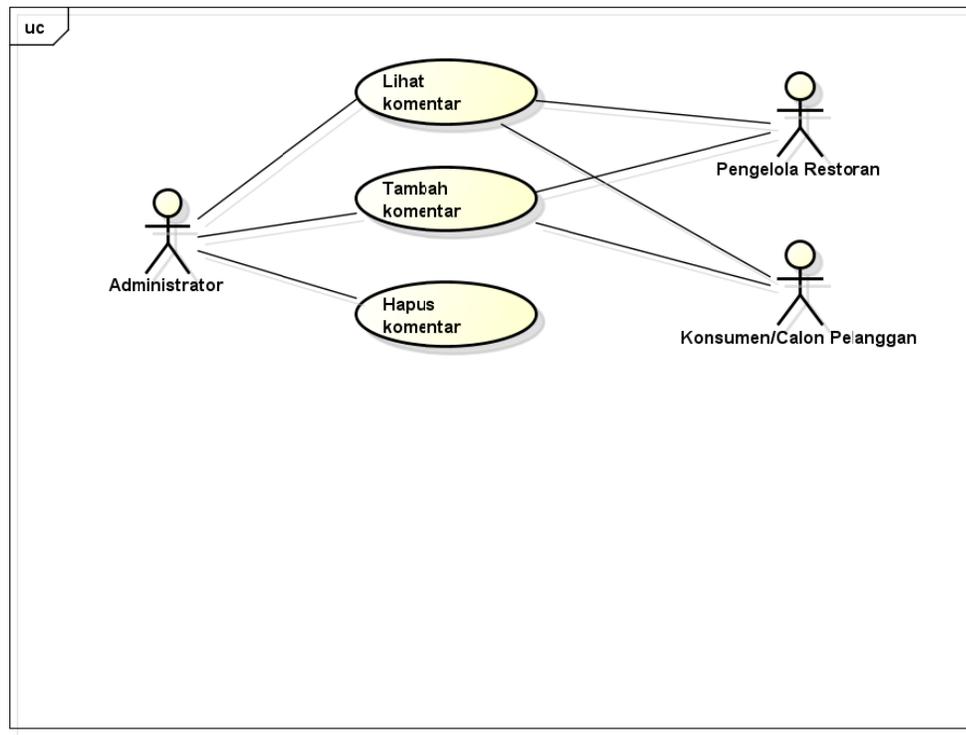
Sumber: [Perancangan]

. Secara keseluruhan sistem rekomendasi ini dibagi menjadi tiga, yaitu : *use case* untuk pengelola rumah makan, *use case* untuk administrator, dan *use case* untuk pelanggan/calon konsumen.

Perancangan *use case diagram* untuk sistem rekomendasi ini berdasarkan pada tabel 3.1 dan ditunjukkan pada gambar 4.3 sebagai berikut.

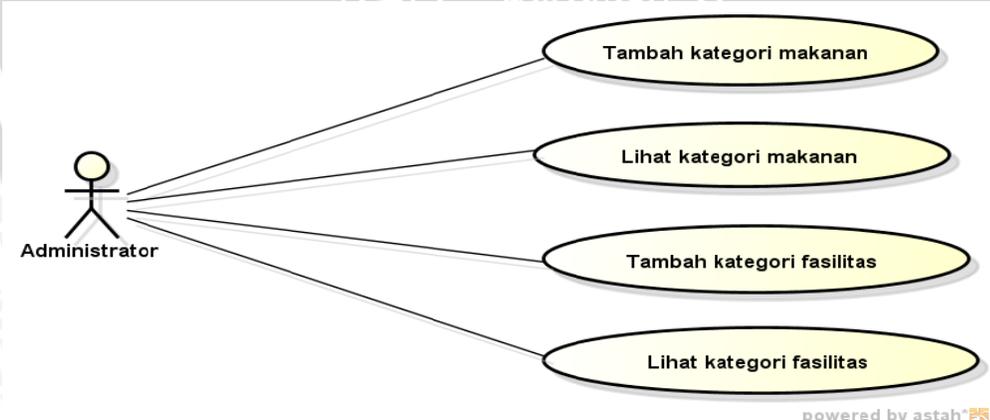


Gambar 4.3 *Use Case* Sistem Rekomendasi Kuliner Berbasis Web dengan Metode *Weighted Product*
Sumber: [Perancangan]



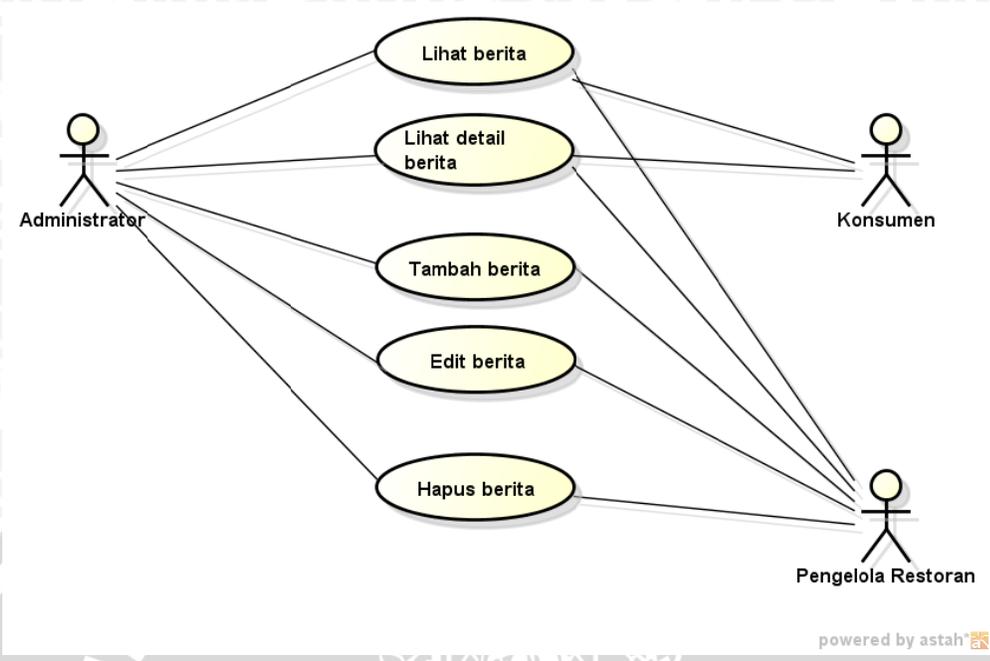
powered by astah

Gambar 4. 4 Use Case Modul Komentar Sistem Rekomendasi Kuliner Berbasis Web dengan Metode Weighted Product
Sumber:[Perancangan]



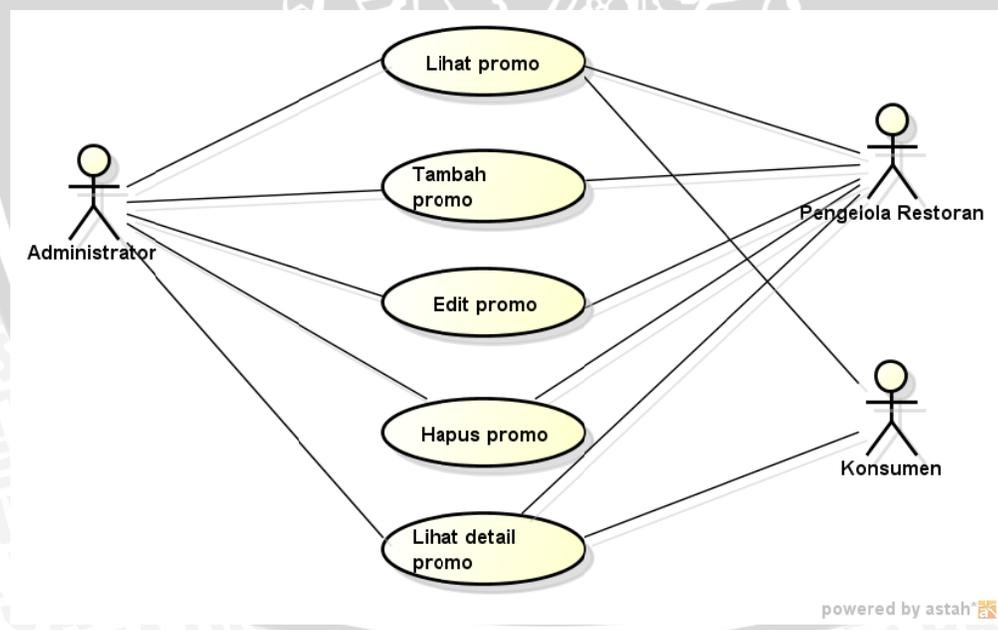
powered by astah

Gambar 4. 5 Use Case Modul Fasilitas dan Modul Kategori Makanan
Sumber:[Perancangan]



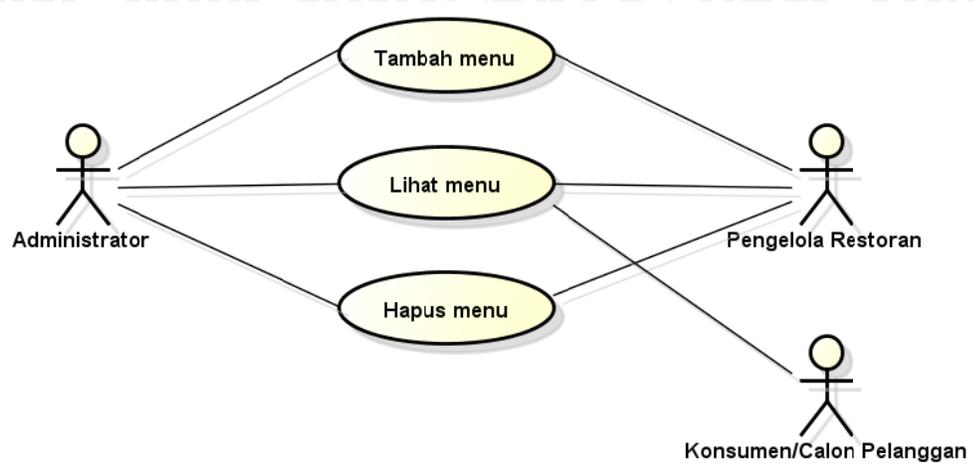
powered by astah*

Gambar 4. 6 Use Case Modul Berita
Sumber:[Perancangan]



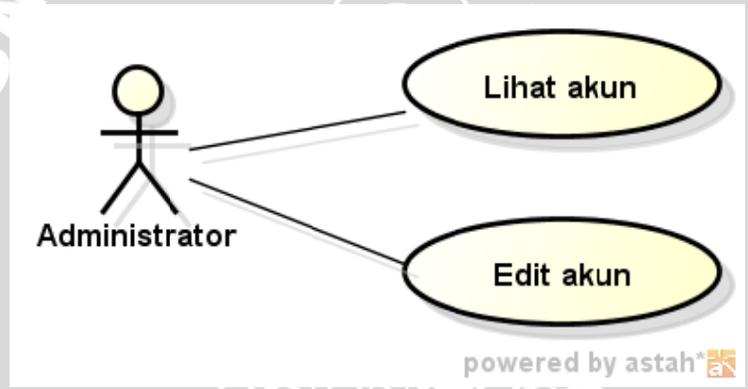
powered by astah*

Gambar 4. 7 Use Case Modul Promo
Sumber:[Perancangan]



powered by astah*

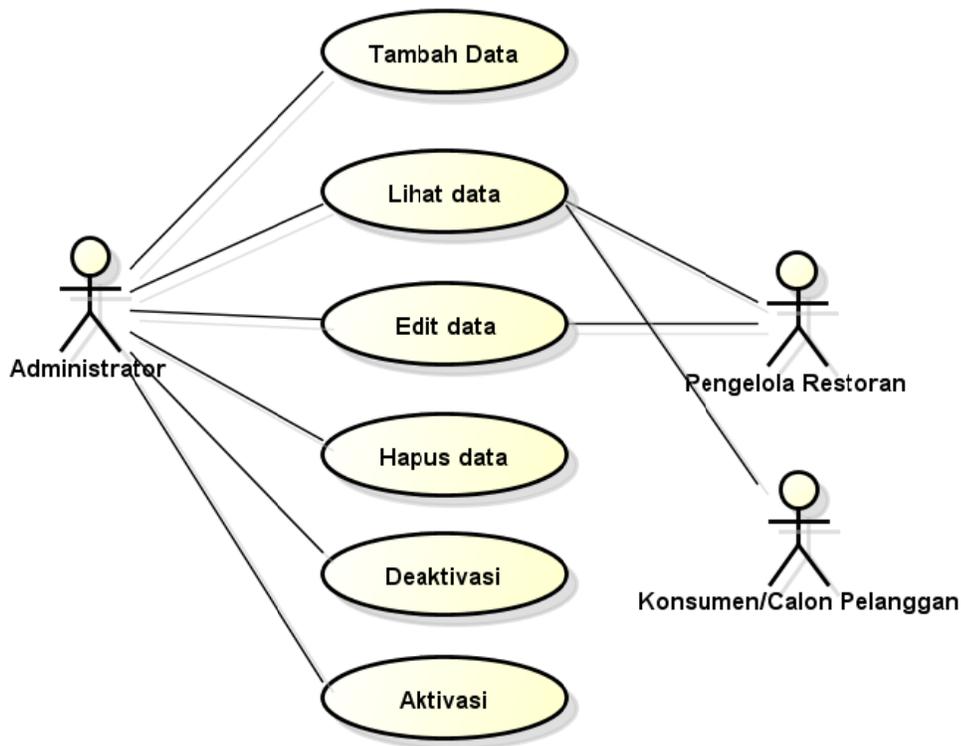
Gambar 4. 8 Use Case Modul Menu
Sumber:[Perancangan]



powered by astah*

Gambar 4. 9 Use Case Modul Akun Admin
Sumber:[Perancangan]





powered by astah

Gambar 4. 10 Use Case Modul Akun dan Data Restoran
Sumber:[Perancangan]

Agar lebih jelas, maka masing-masing *use case* pada gambar di atas akan dijabarkan pada tabel-tabel di bawah ini.

1. *Use case register*

Tabel 4. 7 Use Case Register

Nomor Use Case	SRS_001_01
Nama Use Case	Register
Aktor	Pengelola Restoran
Tujuan	Untuk melakukan registrasi akun
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses registrasi.
Skenario Utama	
Kondisi Awal	Sistem menampilkan halaman utama sistem rekomendasi kuliner.
Aksi dari Aktor	Tanggapan dari sistem
• Aktor menekan tombol menu	• Sistem menampilkan form

<p>pada halaman utama.</p> <ul style="list-style-type: none"> Aktor mengisikan data dan menekan tombol register. 	<p>registrasi.</p> <ul style="list-style-type: none"> Sistem menyimpan data ke database.
Skenario Alternatif 1	
<ul style="list-style-type: none"> Ada beberapa field yang belum terisi oleh pengelola (<i>required field</i>). 	<ul style="list-style-type: none"> Sistem akan menampilkan pesan error kepada pengelola rumah makan.
Skenario Alternatif 2	
<ul style="list-style-type: none"> Mengisi dengan data yang sama pada beberapa kolom yang <i>unique</i>, contoh email. 	<ul style="list-style-type: none"> Sistem akan menampilkan pesan error kepada pengelola rumah makan.
Kondisi Akhir	Sistem menampilkan pesan sukses dan menyimpannya ke database.

Sumber:[Perancangan]

2. Use case Kelola Akun Admin

Tabel 4. 8 Use Case Kelola Akun Admin

Nomor Use Case	SRS_002_01
Nama Use Case	Kelola akun admin
Aktor	<i>Administrator</i>
Tujuan	Untuk mengatur akun <i>administrator</i>
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>administrator</i> mengelola akunya.
Skenario Utama	
Kondisi Awal	<i>Administrator</i> sudah melakukan <i>login</i> dan menampilkan halaman awal (<i>dashboard</i>) <i>administrator</i> .
Aksi dari Aktor	Tanggapan dari sistem
<ul style="list-style-type: none"> <i>Administrator</i> memilih menu <i>Account</i> pada <i>navigation bar</i>. <i>Administrator</i> menekan tombol edit akun. 	<ul style="list-style-type: none"> Sistem akan menampilkan data <i>administrator</i> yang bersangkutan. Sistem menampilkan form edit.

<ul style="list-style-type: none"> Administrator memperbarui data 	<ul style="list-style-type: none"> Sistem melakukan perubahan pada <i>database</i>.
Skenario Alternatif 1	
<ul style="list-style-type: none"> Jika <i>administrator</i> tidak mengisi <i>required field</i>. 	<ul style="list-style-type: none"> Sistem akan menampilkan pesan error.
Kondisi Akhir	Sistem menampilkan data administrator.

3. Use case Kelola Kategori Fasilitas

Tabel 4.9 Use Case Kelola Kategori Fasilitas

Nomor Use Case	SRS_002_02	
Nama Use Case	Kelola kategori fasilitas	
Aktor	<i>Administrator</i>	
Tujuan	Untuk mengatur kategori fasilitas yang ada pada sistem	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>administrator</i> mengelola data kategori fasilitas.	
Skenario Utama		
Kondisi Awal	<i>Administrator</i> sudah melakukan <i>login</i> dan menampilkan halaman awal (<i>dashboard</i>) <i>administrator</i> .	
Aksi dari Aktor		Tanggapan dari sistem
<ul style="list-style-type: none"> <i>Administrator</i> memilih menu <i>others</i> di <i>sidebar</i>. <i>Administrator</i> memilih menu <i>Facilities</i>. <i>Administrator</i> menambahkan jenis fasilitas pada sistem dengan menekan tombol <i>add</i>. 		<ul style="list-style-type: none"> Sistem akan menampilkan <i>dropdown menu</i>. Sistem menampilkan tabel yang berisikan kategori-kategori fasilitas yang ada pada sistem. Sistem menampilkan <i>form</i> untuk menambahkan kategori.
Skenario Alternatif 1		
<ul style="list-style-type: none"> Jika fasilitas yang dimasukkan sudah ada. 		<ul style="list-style-type: none"> Sistem akan menampilkan pesan bahwa data yang diinputkan sudah ada.
Skenario Alternatif 2		

<ul style="list-style-type: none"> • Ada <i>field</i> yang tidak diisi oleh <i>administrator</i> saat menginputkan kategori fasilitas. 	<ul style="list-style-type: none"> • Sistem akan menampilkan pesan error dan gagal menyimpan ke database.
Skenario Alternatif 3	
<ul style="list-style-type: none"> • Id yang dimasukkan tidak sesuai dengan format yang ada pada sistem. 	<ul style="list-style-type: none"> • Sistem akan menampilkan pesan/notifikasi bahwa id fasilitas yang dimasukkan tidak sesuai dengan format yang ada.
Kondisi Akhir	Data fasilitas ditampilkan dan data fasilitas yang baru berhasil dimasukkan ke dalam <i>database</i> .

Sumber:[Perancangan]

4. *Use case* Kelola Kategori Makanan

Tabel 4. 10 *Use Case Kelola Kategori Makanan*

Nomor Use Case	SRS_002_03
Nama Use Case	Kelola kategori makanan
Aktor	<i>Administrator</i>
Tujuan	Untuk mengatur data kategori makanan
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana <i>administrator</i> mengelola data kategori makanan.
Skenario Utama	
Kondisi Awal	<i>Administrator</i> sudah melakukan <i>login</i> dan menampilkan halaman data kategori makanan.
Aksi dari Aktor	Tanggapan dari sistem
<ul style="list-style-type: none"> • <i>Administrator</i> memilih menu <i>cuisine</i> di <i>sidebar</i>. • <i>Administrator</i> menambahkan jenis makanan pada sistem dengan menekan tombol <i>add</i>. 	<ul style="list-style-type: none"> • Sistem akan menampilkan data jenis makanan yang ada pada sistem. • Sistem menerima masukan data dari <i>administrator</i>, kemudian menyimpannya ke <i>database</i> dengan <i>query insert</i>.

Skenario Alternatif 1	
<ul style="list-style-type: none"> Jika jenis makanan yang dimasukkan sudah ada. 	<ul style="list-style-type: none"> Sistem akan menampilkan pesan bahwa data yang diinputkan sudah ada.
Skenario Alternatif 2	
<ul style="list-style-type: none"> Ada <i>field</i> yang tidak diisi oleh <i>administrator</i> saat menginputkan kategori makanan. 	<ul style="list-style-type: none"> Sistem akan menampilkan pesan error dan gagal menyimpan ke database.
Skenario Alternatif 3	
<ul style="list-style-type: none"> Id yang dimasukkan tidak sesuai dengan format yang ada pada sistem. 	<ul style="list-style-type: none"> Sistem akan menampilkan pesan/notifikasi bahwa id fasilitas yang dimasukkan tidak sesuai dengan format yang ada.
Kondisi Akhir	Data jenis makanan ditampilkan dan data jenis makanan yang baru berhasil dimasukkan ke dalam <i>database</i> .

Sumber:[Perancangan]

5. Use case reset password

Tabel 4. 11 Use Case Reset Password

Nomor Use Case	SRS_003_01
Nama Use Case	<i>Reset password</i>
Aktor	<i>Administrator</i> dan pengelola restoran
Tujuan	Untuk melakukan <i>reset password</i>
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses <i>reset password</i> .
Skenario Utama	
Kondisi Awal	Aktor masuk ke halaman utama sistem dan menekan tombol login pada halaman utama sistem.
Aksi dari Aktor	Tanggapan dari sistem
<ul style="list-style-type: none"> Aktor memilih tombol <i>forgot password</i>. 	<ul style="list-style-type: none"> Sistem menampilkan form <i>reset password</i>.

<ul style="list-style-type: none"> Aktor mengisikan <i>email</i> dan role di sistem ini. 	<ul style="list-style-type: none"> <i>Password</i> akan dikirim ke email dan <i>password</i> diperbarui.
Skenario Alternatif 1	
<ul style="list-style-type: none"> <i>Email</i> tidak dapat ditemukan di <i>database</i>. 	<ul style="list-style-type: none"> Sistem akan menampilkan pesan error dan gagal melakukan reset password.
Kondisi Akhir	Sistem menampilkan notifikasi bahwa reset password berhasil dan password diperbarui.

6. Use case Kelola Akun Dan Data Restoran

Tabel 4. 12 Kelola Akun dan Data Restoran

Nomor Use Case	SRS_004_01
Nama Use Case	Kelola Akun Dan Data/Restoran Rumah makan
Aktor	<i>Administrator</i> , pengelola rumah makan, dan konsumen
Tujuan	Untuk melakukan pengolahan pada data rumah makan
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses pengolahan akun dan data .restoran
Skenario Utama	
Kondisi Awal	<ul style="list-style-type: none"> <i>Administrator</i> dan Pengelola restoran telah memasuki halaman akun setelah melakukan <i>login</i>. Konsumen memasuki halaman utama sistem rekomendasi.
Aksi dari Aktor	Tanggapan dari sistem
<ul style="list-style-type: none"> Aktor dapat melihat data restoran. Pengelola restoran dapat melihat data akun miliknya. Pengelola restoran dapat memperbarui data restoran yang dimiliki dengan menekan tombol <i>edit</i>. Pengelola restoran dapat 	<ul style="list-style-type: none"> Sistem menampilkan data restoran. Sistem menampilkan data akun pengelola restoran. Sistem menampilkan <i>form</i> untuk memperbarui data Sistem menampilkan <i>form</i> untuk memperbarui data

<p>memperbarui data dirinya menekan tombol <i>edit</i> pada <i>user profile</i>.</p> <ul style="list-style-type: none"> • <i>Administrator</i> dapat memperbarui data akun dengan menekan tombol <i>update</i>. • <i>Administrator</i> dapat memperbarui data restoran dengan menekan tombol <i>edit</i> sesudah menekan tombol <i>view</i> pada saat halaman akun pengelola ditampilkan. • <i>Administrator</i> dapat menambahkan akun dan restoran baru dengan menekan tombol <i>Add</i>. • <i>Administrator</i> dapat melihat data akun milik pengelola restoran. 	<ul style="list-style-type: none"> • Sistem menampilkan <i>form</i> untuk memperbarui data. • Sistem menampilkan <i>form</i> untuk memperbarui data. • Sistem menampilkan <i>form</i> untuk proses tambah akun dan restoran. • Sistem menampilkan data pengelola restoran.
<p>Skenario Alternatif 1</p>	
<ul style="list-style-type: none"> • Jika pada proses edit, ada <i>field (required)</i> yang tidak terisi. 	<ul style="list-style-type: none"> • Sistem akan menampilkan pesan bahwa <i>field</i> tersebut harus diisi.
<p>Skenario Alternatif 2</p>	
<ul style="list-style-type: none"> • Jika pada proses tambah akun dan restoran, ada <i>field (required)</i> yang tidak terisi. 	<ul style="list-style-type: none"> • Sistem akan menampilkan pesan bahwa <i>field</i> tersebut harus diisi.
<p>Skenario Alternatif 3</p>	
<ul style="list-style-type: none"> • Jika pada proses tambah akun dan data restoran dan proses edit, <i>field email</i> memiliki data yang sama 	<ul style="list-style-type: none"> • Sistem akan menampilkan pesan bahwa <i>email already exist</i> (email sudah ada) dan kembali



dengan yang ada pada <i>database</i> .	menampilkan halaman <i>edit</i> .
Kondisi Akhir	Sistem akan menampilkan halaman akun dan data restoran.

Sumber:[Perancangan]

7. Use case Kelola Menu Restoran

Tabel 4. 13 Use Case Kelola Menu Restoran

Use Case	SRS_004_02	
Nama use case	Kelola menu restoran	
Aktor	<i>Administrator</i> , pengelola restoran, Konsumen	
Tujuan	Untuk menampilkan, menambahkan, dan menghapus menu	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses menampilkan, menambahkan, dan menghapus menu	
Skenario Utama		
Kondisi Awal	<ul style="list-style-type: none"> • <i>Administrator</i> dan Pengelola restoran telah memasuki halaman <i>menu</i> setelah melakukan <i>login</i>. • Konsumen sudah memasuki halaman utama sistem. 	
Aksi dari Aktor		Tanggapan dari sistem
<ul style="list-style-type: none"> • Aktor dapat melihat <i>menu</i> • Pengelola restoran dapat menambahkan menu dengan menekan <i>icon add</i> • Pengelola restoran dapat menghapus menu dengan menekan <i>icon delete</i> • <i>Administrator</i> dapat menambahkan <i>menu</i> sesuai dengan akun yang dipilihnya 		<ul style="list-style-type: none"> • Sistem menampilkan <i>menu</i> yang ada pada sistem. • Sistem menampilkan <i>form upload</i> sesuai dengan akun pengelola restoran • Sistem akan menghapus data yang dipih. • Sistem menampilkan <i>form upload</i> sesuai dengan akun yang dipilih.

<ul style="list-style-type: none"> • <i>Administrator</i> dapat menghapus <i>menu</i> yang dipilihnya. • <i>Administrator</i> dapat menghapus semua data <i>menu</i> sesuai dengan akun yang dipilihnya. 	<ul style="list-style-type: none"> • Sistem akan melakukan proses penghapusan. • Sistem akan melakukan proses penghapusan.
Skenario Alternatif 1	
<ul style="list-style-type: none"> • Gambar <i>menu</i> yang diupload tidak sesuai dengan format yang ditentukan (.JPG, JPEG,.PNG). 	<ul style="list-style-type: none"> • Sistem akan menampilkan pesan bahwa gambar <i>menu</i> yang diupload tidak sesuai dengan format.
Skenario Alternatif 2	
<ul style="list-style-type: none"> • Jika <i>Administrator</i> dan Pengelola restoran tidak mengisi semua <i>field</i> yang dibutuhkan untuk proses tambah <i>menu</i>. 	<ul style="list-style-type: none"> • Sistem akan memunculkan pesan error.
Kondisi Akhir	.Sistem menampilkan halaman <i>menu</i>

Sumber:[Perancangan]

8. *Use case* Kelola Promo Restoran

Tabel 4. 14 Use Case Kelola Promo Restoran

Nomor Use Case	SRS_004_03
Nama Use Case	Kelola promo restoran
Aktor	<i>Administrator</i> , pengelola restoran, dan konsumen
Tujuan	Untuk melakukan pengolahan data promo
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses pengolahan pada data promo
Skenario Utama	
Kondisi Awal	<i>Administrator</i> dan pengelola restoran harus melalui proses autentifikasi terlebih dahulu. Sedangkan untuk konsumen/calon pelanggan sudah harus membuka

	halaman awal sistem.
Aksi dari Aktor	Tanggapan dari sistem
<ul style="list-style-type: none"> • Aktor dapat melihat promo restoran. • Aktor dapat melihat detail promo restoran. • Pengelola restoran dapat menambahkan promo baru dengan menekan tombol <i>add</i> • Pengelola restoran dapat menghapus promo dengan menekan tombol <i>delete</i> • Pengelola restoran dapat memperbaiki promo • <i>Administrator</i> dapat menambahkan promo baru dengan menekan tombol <i>add</i> sesuai dengan akun yang dipilih. • <i>Administrator</i> dapat menghapus promo dengan menekan tombol <i>delete</i> sesuai dengan akun yang dipilih. • <i>Administrator</i> dapat memperbaiki promo sesuai dengan akun yang dipilih. • <i>Administrator</i> dapat menghapus semua data promo sesuai dengan akun yang dipilih. 	<ul style="list-style-type: none"> • Sistem menampilkan promo restoran. • Sistem menampilkan detail promo restoran. • Sistem menampilkan <i>form</i> untuk menambahkan promo • Sistem akan menghapus data. • Sistem menampilkan <i>form</i> untuk perubahan data. • Sistem menampilkan <i>form</i> untuk menambahkan promo rumah makan. • Sistem akan menghapus data. • Sistem menampilkan <i>form</i> untuk perubahan data. • Sistem akan menghapus data.
Skenario Alternatif 1	
<ul style="list-style-type: none"> • Jika <i>Administrator</i> dan Pengelola restoran tidak mengisi semua <i>field</i> 	<ul style="list-style-type: none"> • Sistem akan memunculkan pesan error.

yang dibutuhkan untuk proses edit promo.	
Skenario Alternatif 2	
<ul style="list-style-type: none"> Jika <i>Administrator</i> dan Pengelola restoran tidak mengisi semua <i>field</i> yang dibutuhkan untuk proses tambah promo. 	<ul style="list-style-type: none"> Sistem akan memunculkan pesan error.
Kondisi Akhir	Sistem akan menampilkan halaman promo

Sumber:[Perancangan]

9. *Use case* Kelola Komentar

Tabel 4. 15 Use Case Kelola Promo Restoran

Nomor Use Case	SRS_004_04
Nama Use Case	Kelola komentar
Aktor	<i>Administrator</i> , pengelola restoran, dan konsumen
Tujuan	Untuk melakukan pengolahan pada data komentar
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses pengolahan pada data komentar
Skenario Utama	
Kondisi Awal	<i>Administrator</i> dan pengelola rumah makan harus melalui proses autentifikasi terlebih dahulu. Sedangkan untuk konsumen/calon pelanggan sudah harus membuka halaman awal sistem.
Aksi dari Aktor	Tanggapan dari sistem
<ul style="list-style-type: none"> Aktor dapat melihat komentar restoran. Pengelola restoran dapat membalas komentar. Konsumen menambahkan komentar. <i>Administrator</i> dapat menambahkan komentar sesuai 	<ul style="list-style-type: none"> Sistem menampilkan komentar restoran. Sistem akan menampilkan <i>form</i> untuk balas komentar. Sistem menampilkan <i>form</i> untuk menambahkan komentar. Sistem menampilkan <i>form</i> untuk menambahkan komentar.

<p>dengan akun yang dipilih.</p> <ul style="list-style-type: none"> • <i>Administrator</i> dapat menghapus data komentar sesuai dengan akun yang dipilih. • <i>Administrator</i> dapat menghapus semua data komentar sesuai dengan akun yang dipilih. 	<ul style="list-style-type: none"> • Sistem akan melakukan penghapusan data. • Sistem akan melakukan penghapusan data.
Skenario Alternatif 1	
<ul style="list-style-type: none"> • Jika aktor tidak mengisi semua <i>field</i> yang dibutuhkan untuk proses tambah komentar/balas komentar. 	<ul style="list-style-type: none"> • Sistem akan memunculkan pesan error.
Kondisi Akhir	Sistem akan menampilkan halaman komentar.

Sumber:[Perancangan]

10. *Use case* Kelola berita

Tabel 4. 16 Use Case Kelola Berita

Nomor Use Case	SRS_004_05
Nama Use Case	Kelola berita
Aktor	<i>Administrator</i> , pengelola restoran, dan konsumen
Tujuan	Untuk melakukan pengolahan pada data berita
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses pengolahan pada data berita
Skenario Utama	
Kondisi Awal	<i>Administrator</i> dan pengelola rumah makan harus melalui proses autentifikasi terlebih dahulu. Sedangkan untuk konsumen/calon pelanggan sudah harus membuka halaman awal sistem.
Aksi dari Aktor	Tanggapan dari system
<ul style="list-style-type: none"> • Aktor dapat melihat berita restoran. 	<ul style="list-style-type: none"> • Sistem menampilkan berita restoran.

<ul style="list-style-type: none"> • Aktor dapat melihat detail berita restoran/rumah makan. • Pengelola restoran dapat menambahkan berita baru dengan menekan tombol <i>add</i> • Pengelola restoran dapat menghapus berita dengan menekan tombol <i>delete</i> • Pengelola restoran dapat memperbaiki berita dengan menekan tombol edit • <i>Administrator</i> dapat menambahkan berita baru dengan menekan tombol <i>add</i> sesuai dengan akun yang dipilih. • <i>Administrator</i> dapat menghapus berita dengan menekan tombol <i>delete</i> sesuai dengan akun yang dipilih. • <i>Administrator</i> dapat menghapus semua berita dengan menekan tombol <i>delete</i> sesuai dengan akun yang dipilih. • <i>Administrator</i> dapat memperbaiki berita sesuai dengan akun yang dipilih. 	<ul style="list-style-type: none"> • Sistem menampilkan detail berita restoran. • Sistem menampilkan <i>form</i> untuk menambahkan berita • Sistem akan menghapus data. • Sistem menampilkan <i>form</i> untuk perubahan data. • Sistem menampilkan <i>form</i> untuk menambahkan berita • Sistem akan menghapus data. • Sistem akan menghapus data. • Sistem menampilkan <i>form</i> untuk perubahan data.
---	---

Skenario Alternatif 1

<ul style="list-style-type: none"> • Jika <i>Administrator</i> dan Pengelola restoran tidak mengisi semua <i>field</i> yang dibutuhkan untuk proses edit berita. 	<ul style="list-style-type: none"> • Sistem akan memunculkan pesan error.
---	--



Skenario Alternatif 2	
<ul style="list-style-type: none"> Jika <i>Administrator</i> dan Pengelola restoran tidak mengisi semua <i>field</i> yang dibutuhkan untuk proses tambah berita. 	<ul style="list-style-type: none"> Sistem akan memunculkan pesan error.
Kondisi Akhir	Sistem akan menampilkan halaman berita.

Sumber:[Perancangan]

11. *Use case* Cari Restoran

Tabel 4. 17 Use Case Cari Restoran

Nomor Use Case	SRS_005_01
Nama Use Case	Cari restoran
Aktor	Konsumen
Tujuan	Untuk melakukan pencarian rumah makan berdasarkan nama rumah makan dan jenis makanan yang ada pada sistem.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses pencarian rumah makan.
Skenario Utama	
Kondisi Awal	Aktor telah memasuki halaman utama sistem
Aksi dari Aktor	Tanggapan dari sistem
<ul style="list-style-type: none"> Konsumen/ memilih menu <i>search</i> pada <i>navigation bar</i> Konsumen melakukan pencarian dengan memasukkan nama rumah makan 	<ul style="list-style-type: none"> Sistem menampilkan form Sistem menampilkan restoran yang dicari. Kemudian ada dua pilihan yaitu menampilkan informasi rumah makan tersebut atau mendapatkan petunjuk arah menuju tempat tersebut.
Skenario alternatif 1	
<ul style="list-style-type: none"> Rumah makan yang dicari tidak ditemukan. 	<ul style="list-style-type: none"> Sistem menampilkan pesan bahwa data yang dicari tidak ada.

Kondisi Akhir	Sistem akan menampilkan hasil pencarian.
----------------------	--

Sumber:[Perancangan]

12. Use case Cari Rekomendasi Restoran

Tabel 4. 18 Use Case Cari Rekomendasi Restoran

Nomor Use Case	SRS_005_02
Nama Use Case	Cari rekomendasi restoran
Aktor	Konsumen
Tujuan	Untuk melakukan pencarian rumah makan berdasarkan kriteria dan bobot kriteria.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana aktor dapat melakukan proses pencarian rumah makan berdasarkan kriteria dan bobot kriteria.
Skenario Utama	
Kondisi Awal	Aktor telah membuka halaman utama sistem.
Aksi dari Aktor	Tanggapan dari sistem
<ul style="list-style-type: none"> Konsumen memilih menu <i>recommendation</i>. Konsumen mengisikan <i>form</i> sesuai dengan kebutuhannya. 	<ul style="list-style-type: none"> Sistem menampilkan <i>form</i> rekomendasi. Sistem akan melakukan perhitungan dan menampilkan hasil rekomendasi.
Skenario Alternatif 1	
<ul style="list-style-type: none"> Restoran dengan jenis makanan yang dicari tidak ada. 	<ul style="list-style-type: none"> Sistem akan menampilkan pesan/notifikasi bahwa restoran dengan jenis makanan tersebut tidak ada.
Kondisi Akhir	Sistem akan menampilkan hasil perhitungan rekomendasi dan sebuah peta untuk mendapatkan petunjuk arah ke tempat tersebut.

Sumber:[Perancangan]

13. Use case Lihat Petunjuk Arah

Tabel 4. 19 Use Case Lihat Petunjuk Arah

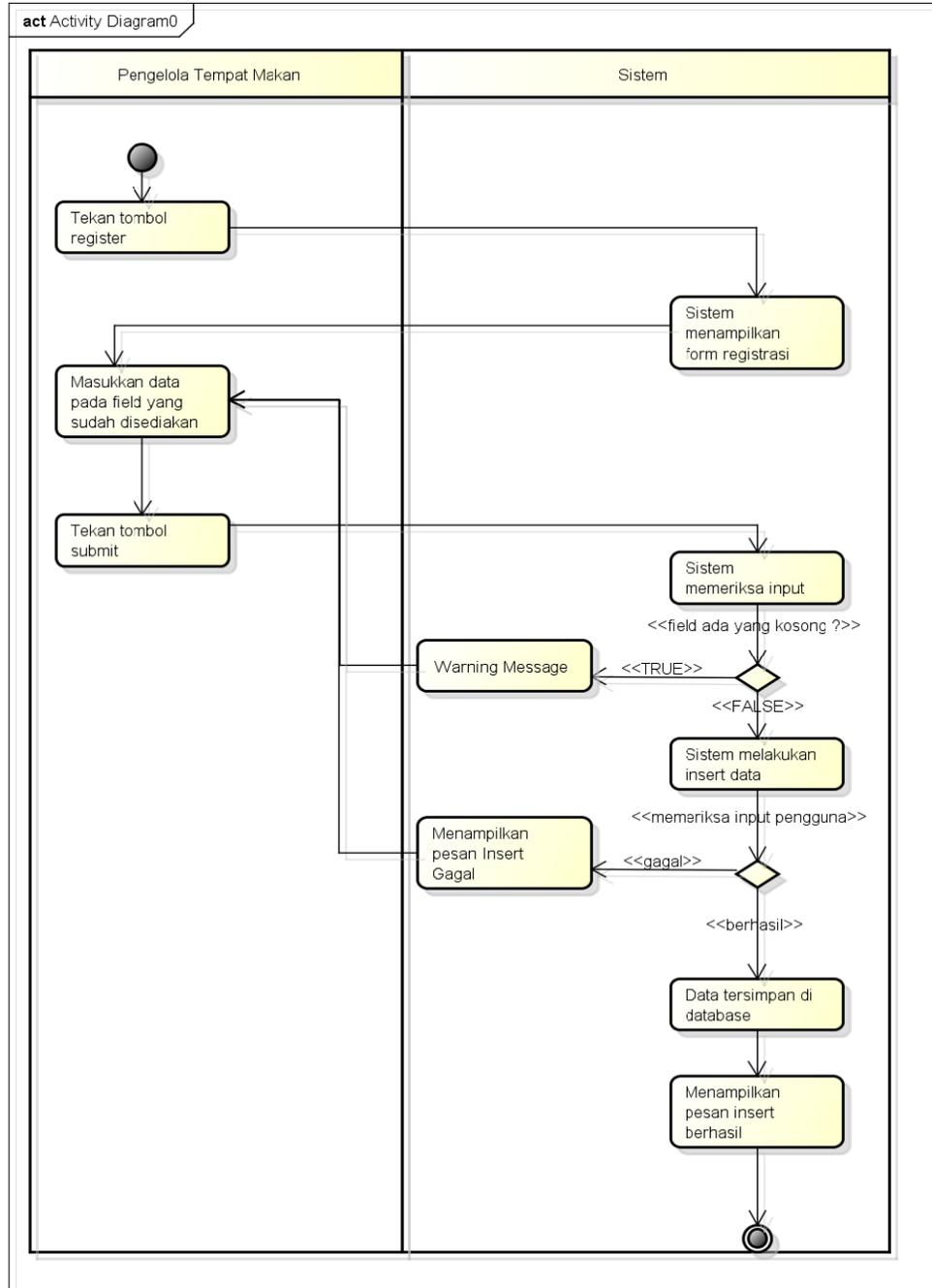
Nomor Use Case	SRS_005_03	
Nama Use Case	Lihat petunjuk arah	
Aktor	Konsumen	
Tujuan	Untuk mendapatkan arah menuju ke restoran tersebut dari tempat konsumen berada	
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana konsumen mendapatkan arah menuju ke restoran tersebut dari tempat konsumen/calon pelanggan berada	
Skenario Utama		
Kondisi Awal	Aktor telah memasuki halaman awal sistem	
	Aksi dari Aktor	Tanggapan dari sistem
	<ul style="list-style-type: none"> Konsumen telah melakukan pencarian dengan menekan menu <i>recommendation</i> pada <i>navbar</i>. Konsumen telah melakukan pencarian dengan menekan menu <i>search</i> pada <i>navbar</i>. 	<ul style="list-style-type: none"> Sistem menampilkan peta dan petunjuk arah menuju restoran tersebut, serta hasil rekomendasi restoran. Sistem menampilkan hasil pencarian dengan disertai tombol <i>view</i> dan <i>map</i>. Tombol <i>map</i> berfungsi untuk menampilkan peta dan petunjuk arah menuju restoran tersebut, serta hasil rekomendasi restoran
Kondisi Akhir	Sistem akan menampilkan peta dan petunjuk arah menuju restoran tersebut.	

Sumber:[Perancangan]

4.3 Diagram Aktivitas

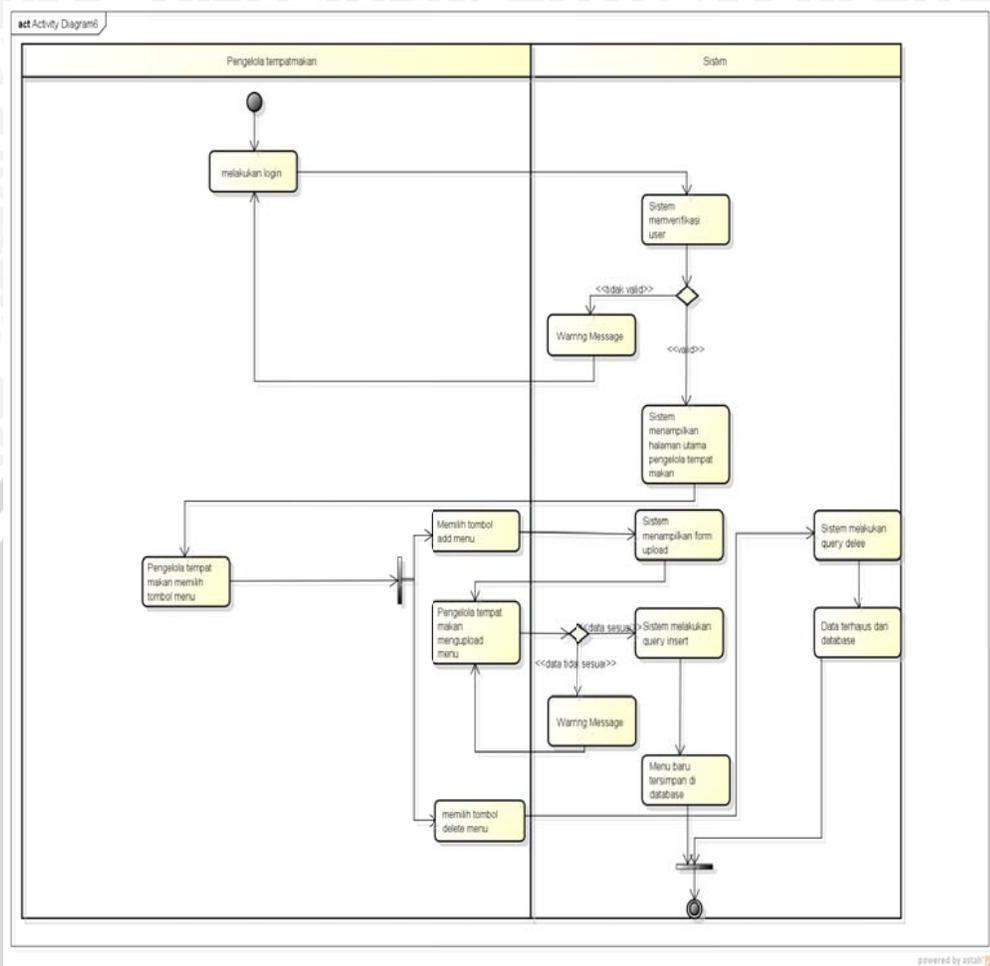
Diagram aktivitas digunakan untuk menggambarkan urutan aktivitas dari proses yang ada pada setiap *use case*. Berikut merupakan gambaran dari beberapa contoh diagram aktivitas yang ada pada sistem.

Gambar 4.11 terdapat diagram aktivitas *register* untuk pengelola rumah makan. Sedangkan aktivitas kelola menu digambarkan pada gambar 4.12.

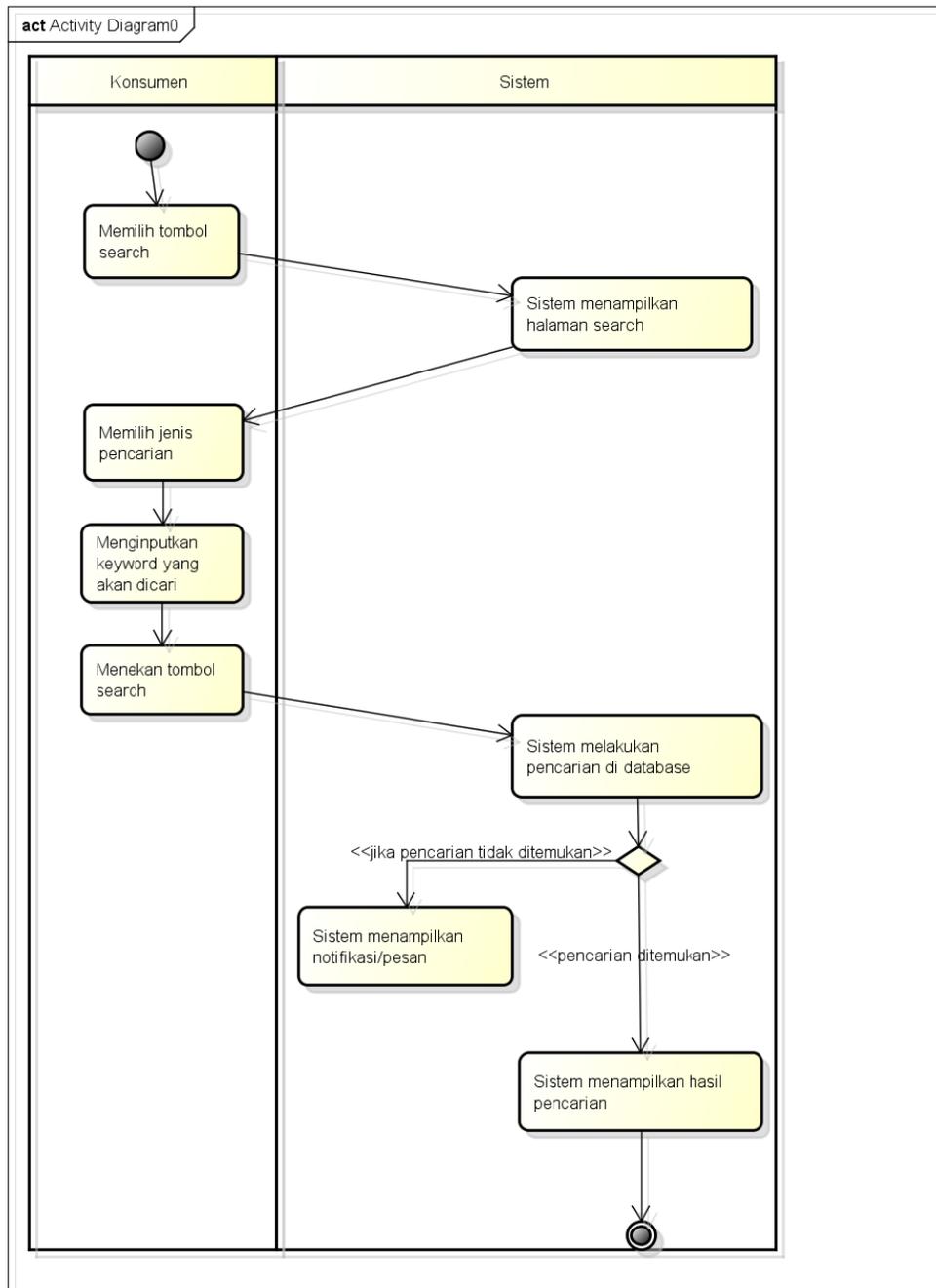


powered by astah

Gambar 4. 11Activity Diagram Register
Sumber:[Perancangan]

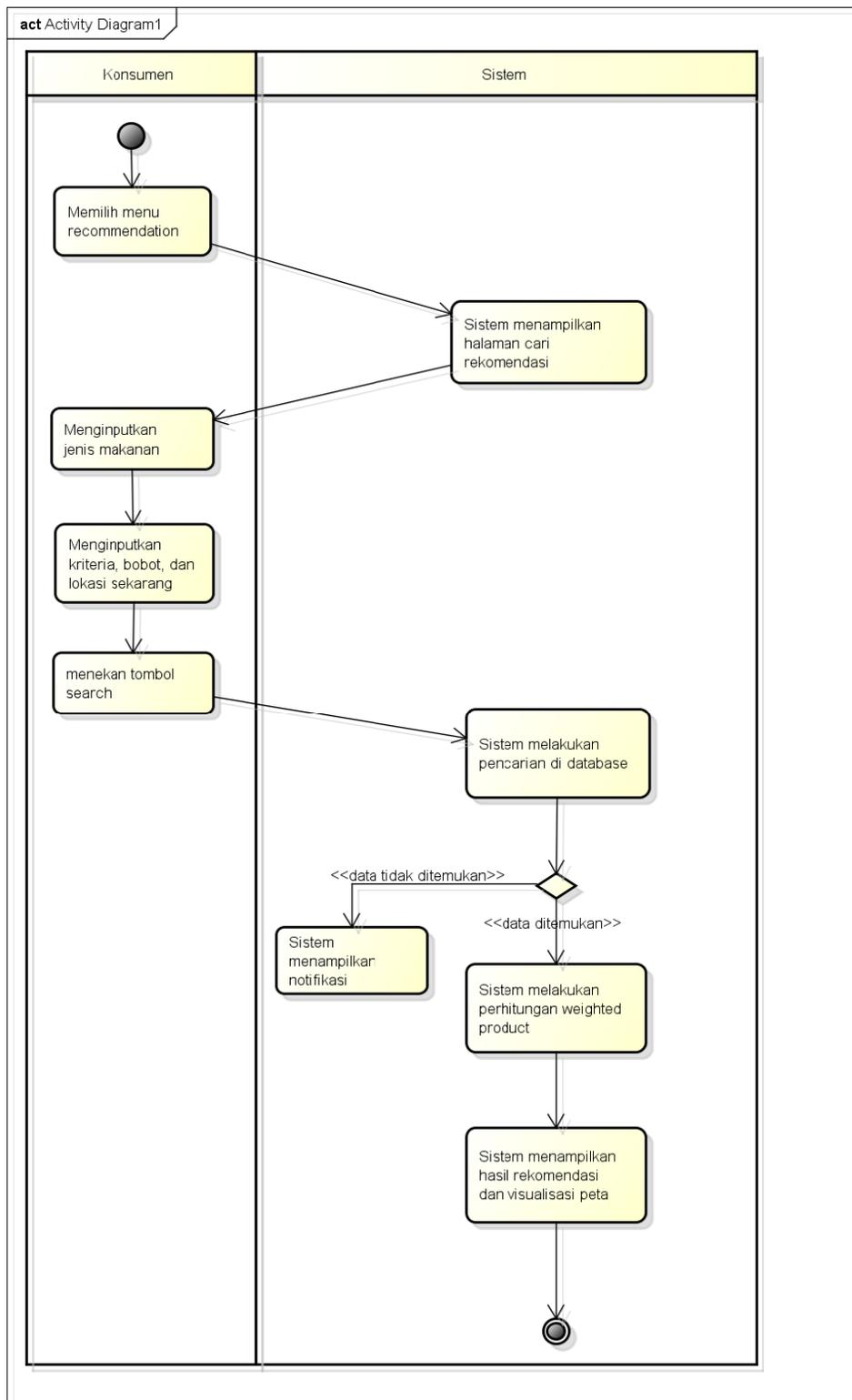


Gambar 4. 12Activity Diagram Menu
Sumber:[Perancangan]



powered by astah

Gambar 4. 13Activity Diagram Cari Restoran
Sumber:[Perancangan]

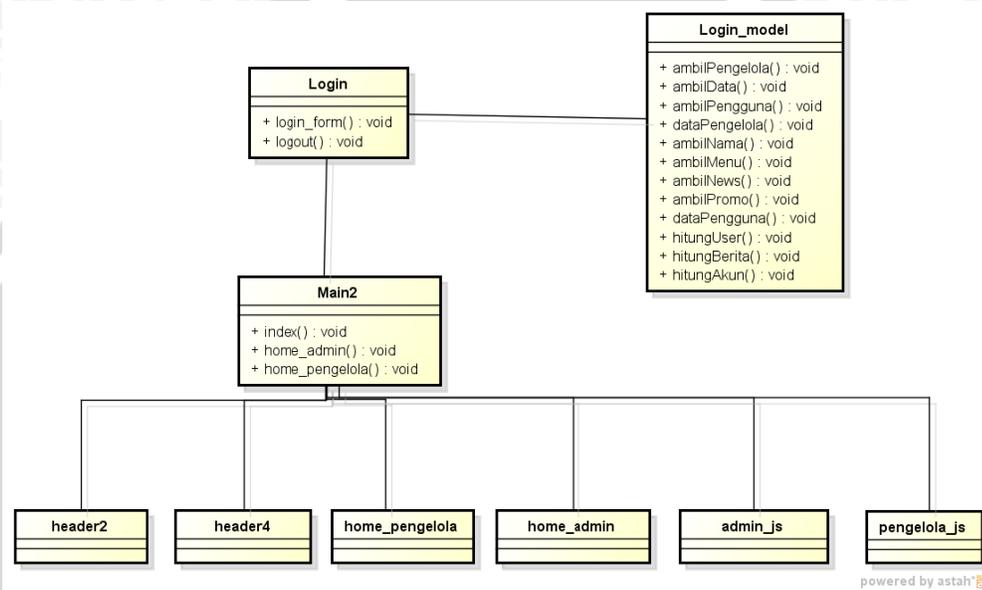


powered by astah

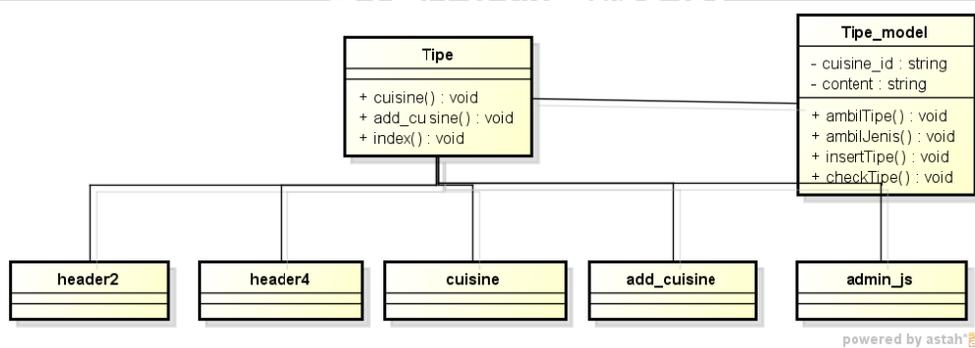
Gambar 4. 14Activity Diagram Cari Rekomendasi Restoran
Sumber:[Perancangan]

4.4 Diagram Kelas

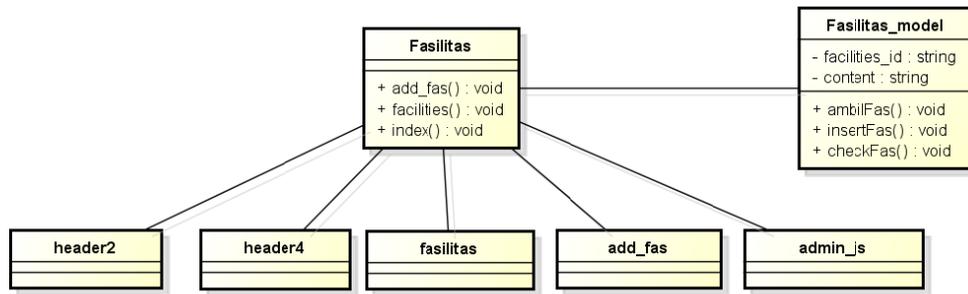
Diagram kelas memberikan gambaran tentang elemen-elemen *class* yang membentuk sebuah sistem perangkat lunak mulai dari atribut, method, dan hubungannya dengan *class* yang lain dalam sistem. *Class Diagram* pada sistem rekomendasi kuliner ini menggunakan *MVC pattern*.



Gambar 4. 15 Class Diagram Kelas Login
Sumber:[Perancangan]

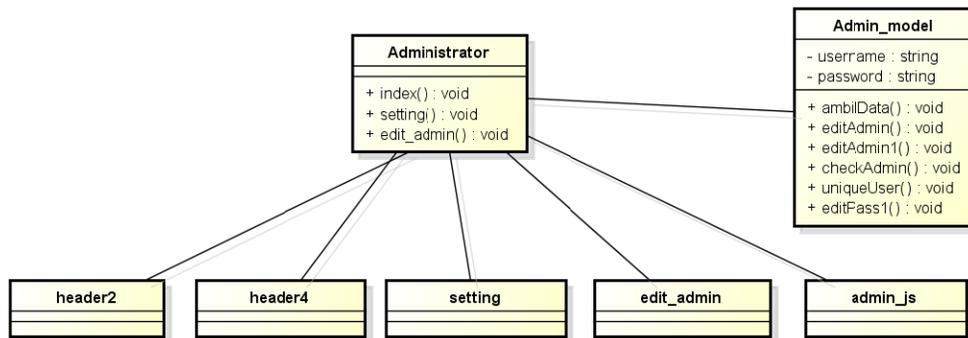


Gambar 4. 16 Class Diagram Kelas Tipe
Sumber:[Perancangan]



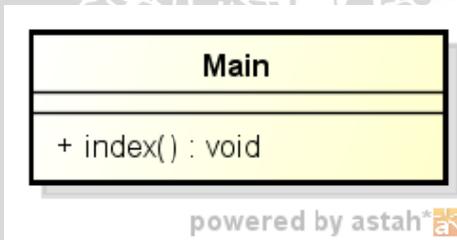
powered by astah

Gambar 4. 17Class Diagram Kelas Fasilitas
Sumber:[Perancangan]



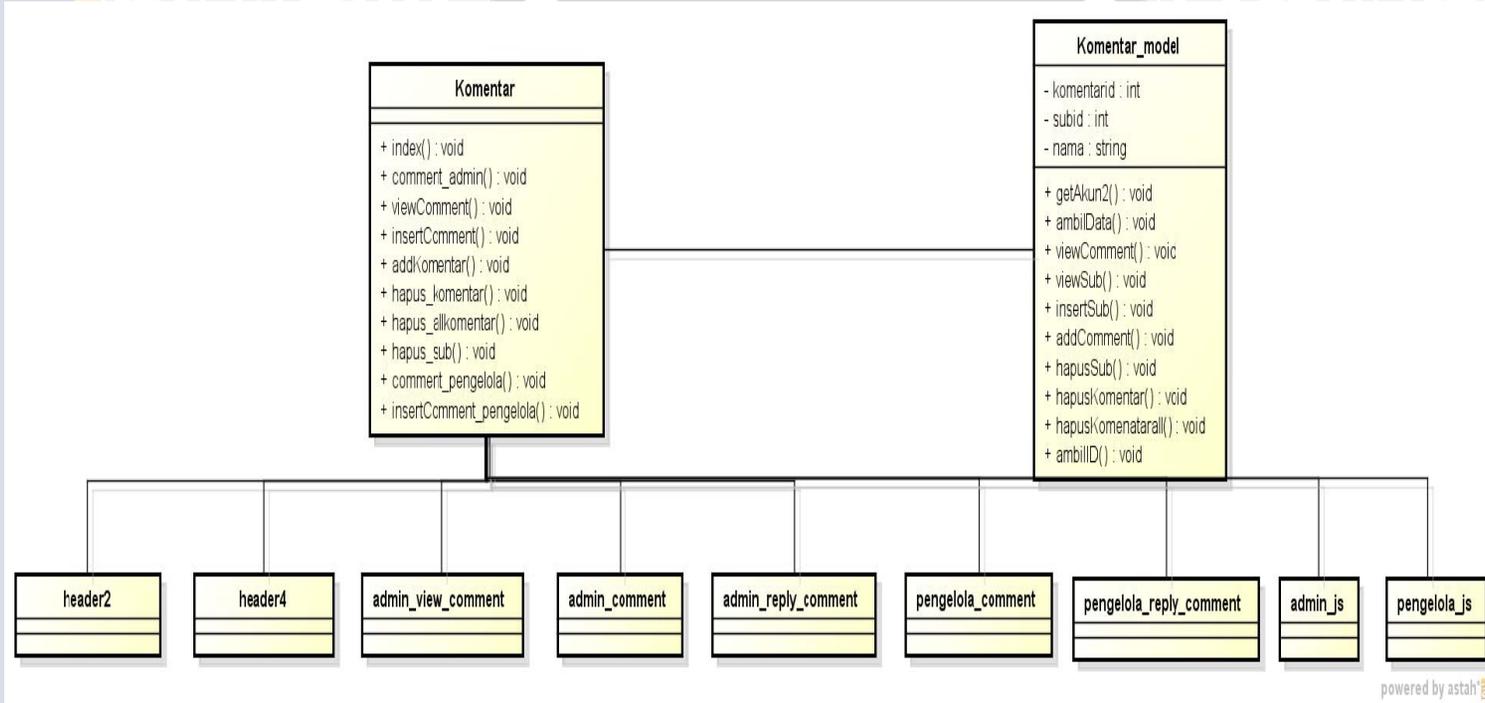
powered by astah

Gambar 4. 18Class Diagram Kelas Administrator
Sumber:[Perancangan]



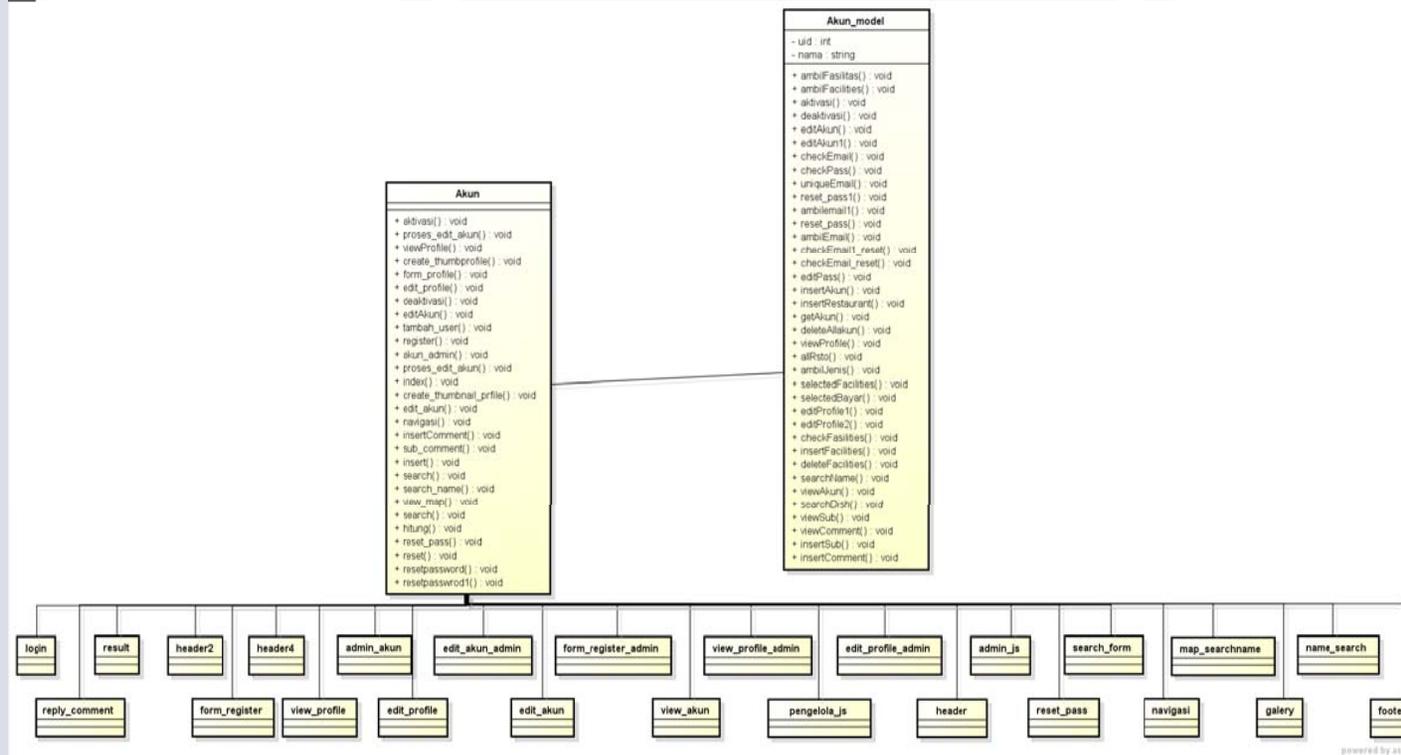
powered by astah

Gambar 4. 19Class Diagram Kelas Main
Sumber:[Perancangan]

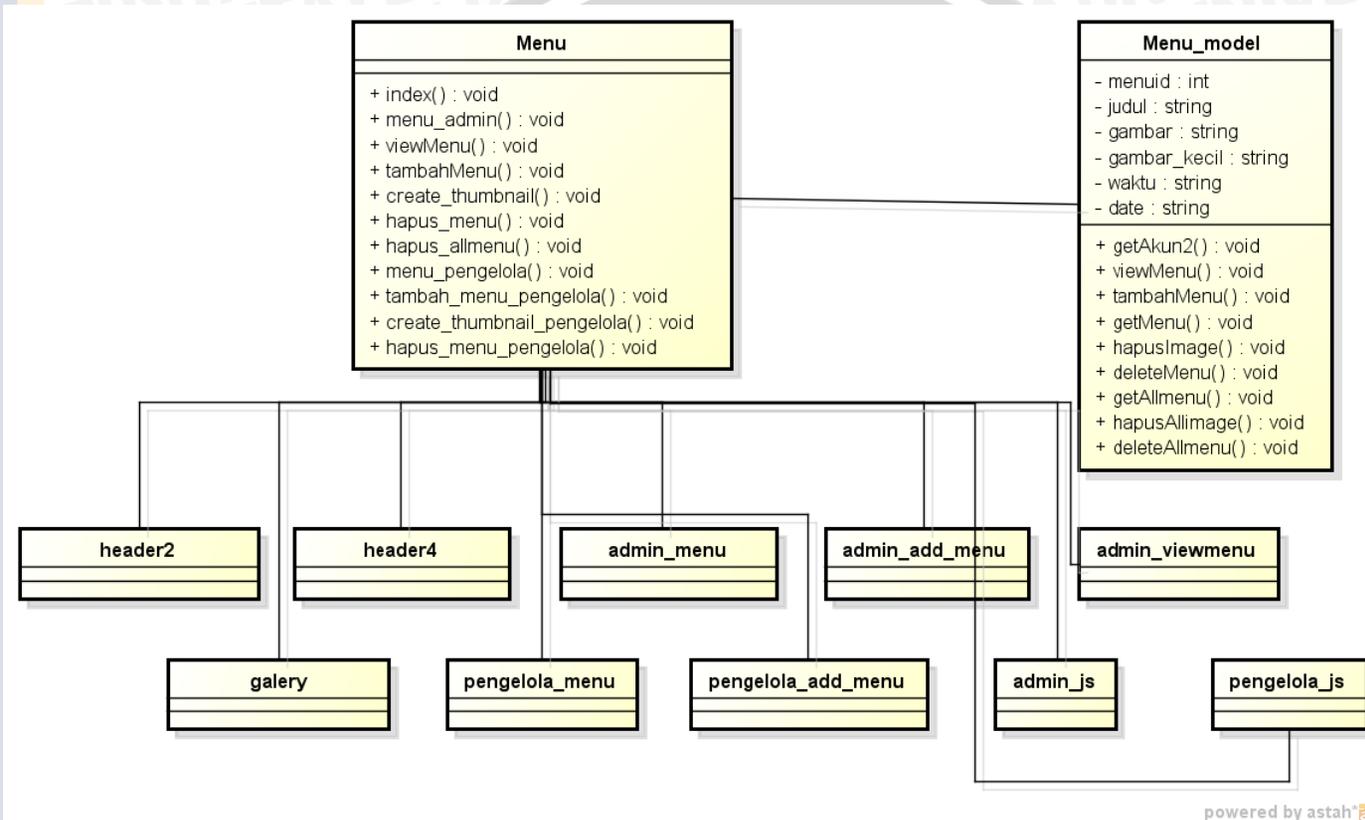


powered by astah

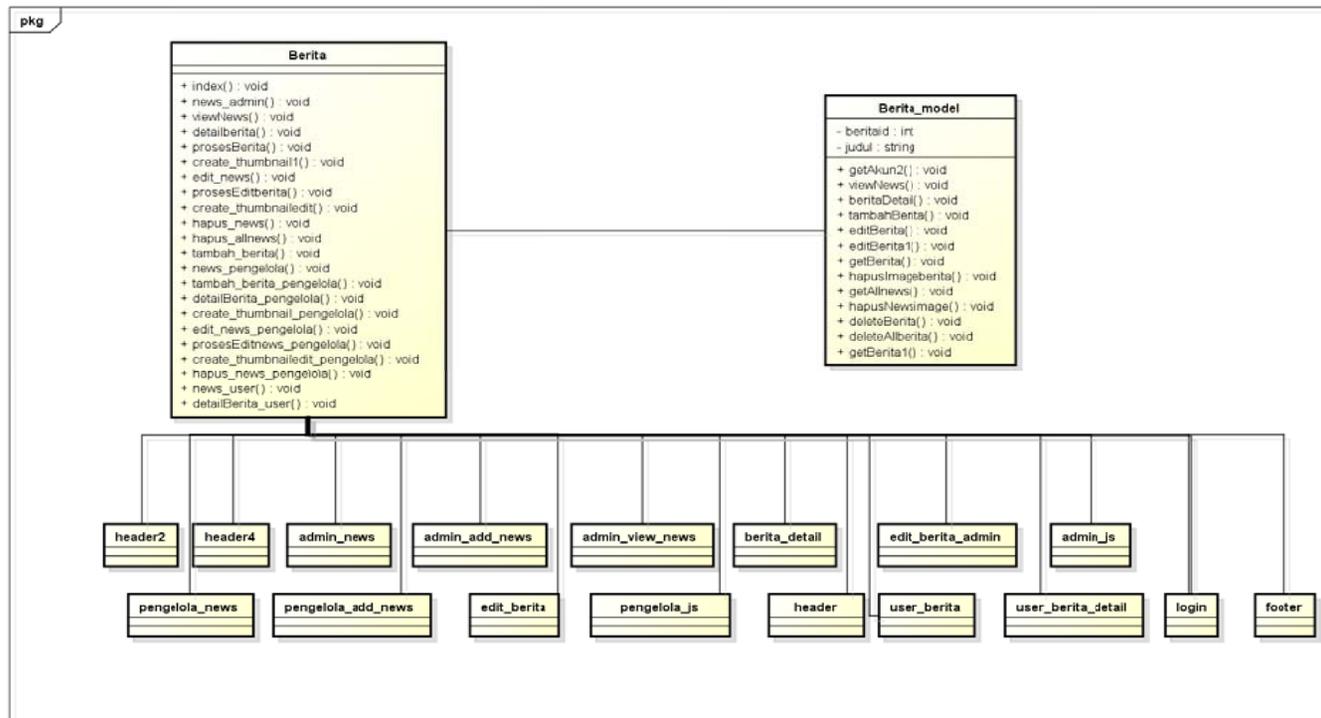
Gambar 4. 20Class Diagram Kelas Komentar
Sumber:[Perancangan]



Gambar 4. 21Class Diagram Kelas Akun
Sumber:[Perancangan]

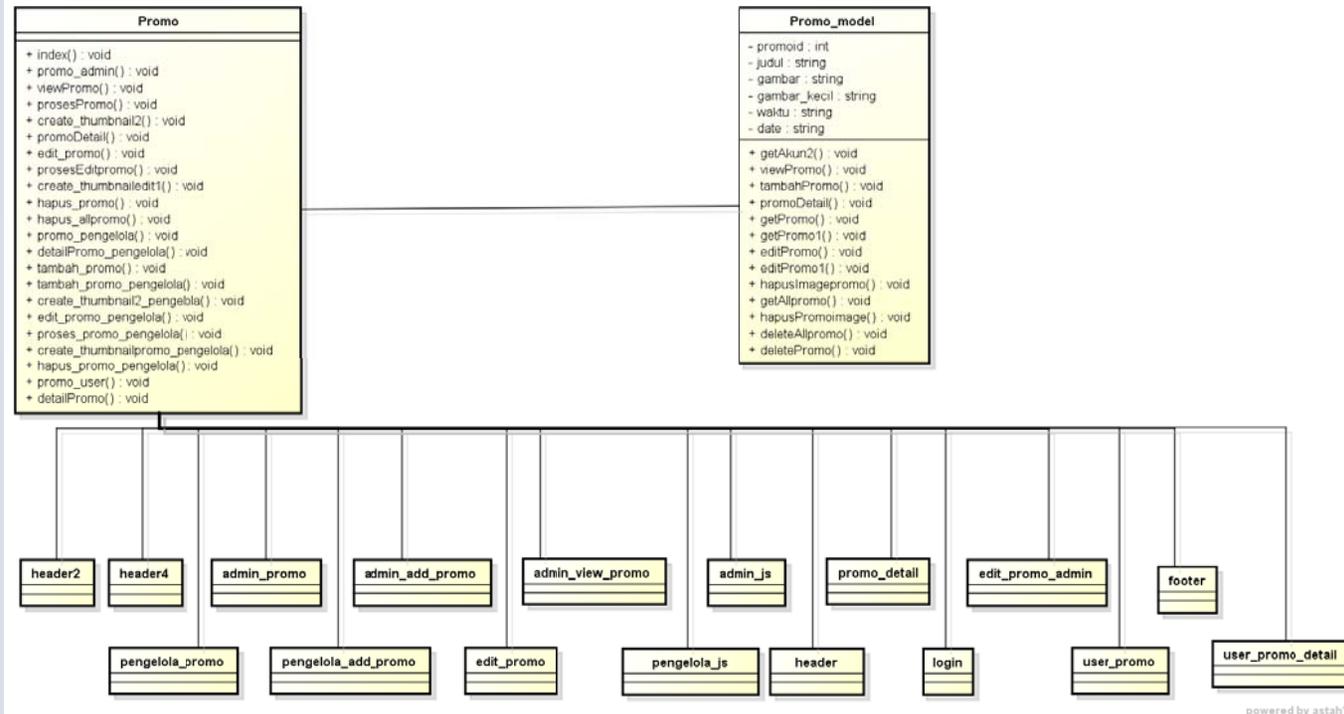


Gambar 4. 22Class Diagram Kelas Menu
 Sumber:[Perancangan]

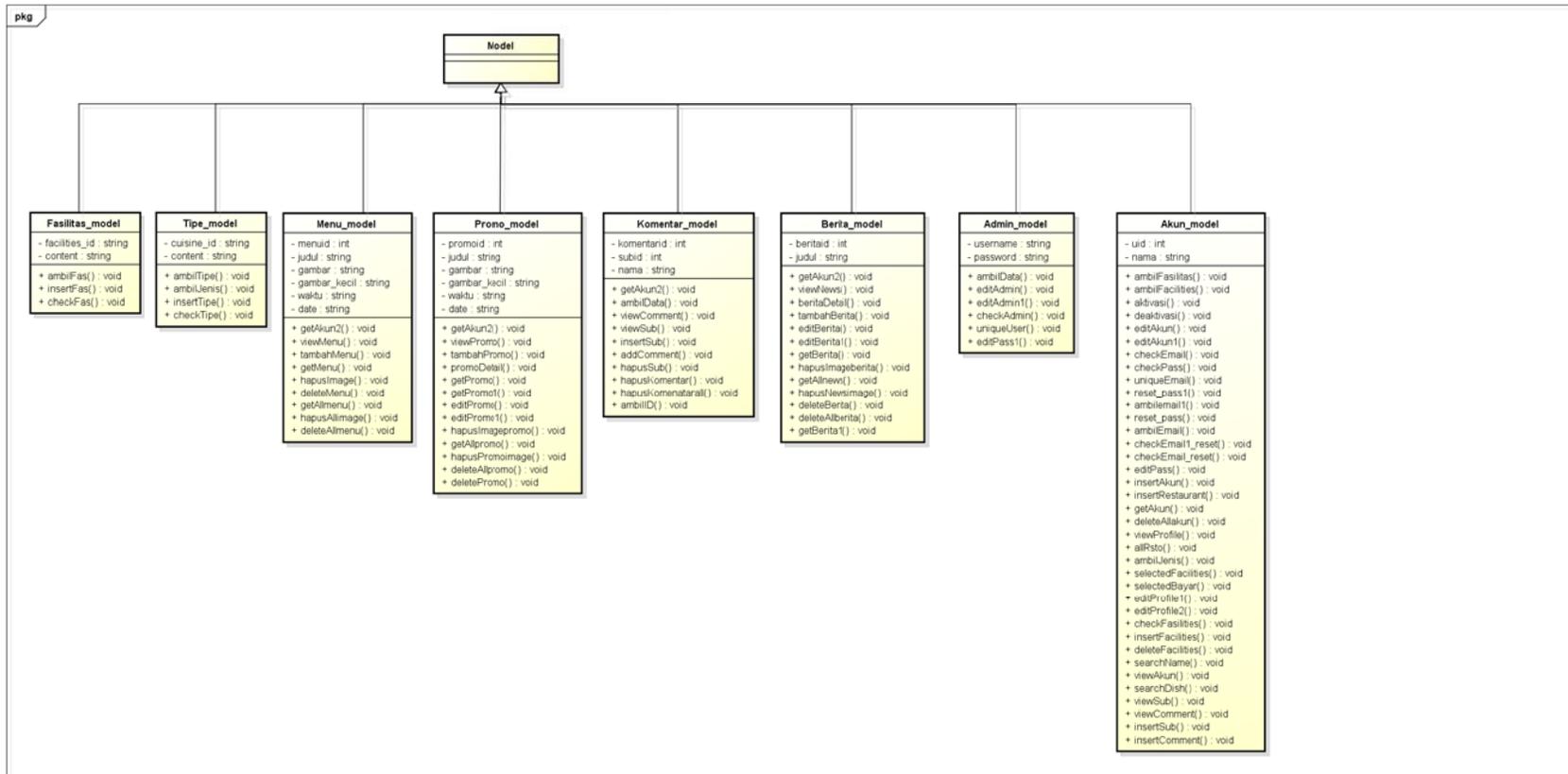


powered by astah

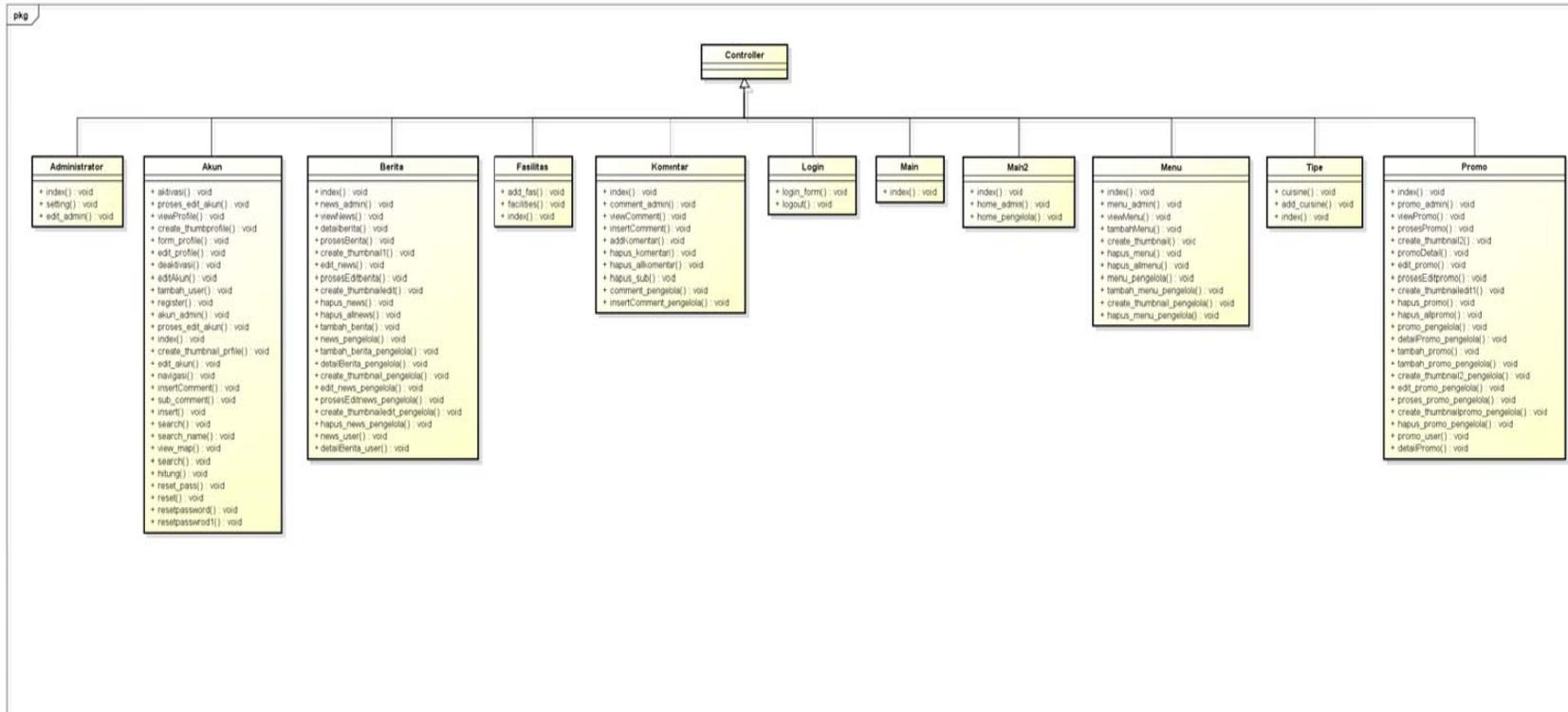
Gambar 4. 23 Class Diagram Kelas Berita
 Sumber:[Perancangan]



Gambar 4. 24Class Diagram Kelas Promo
 Sumber:[Perancangan]



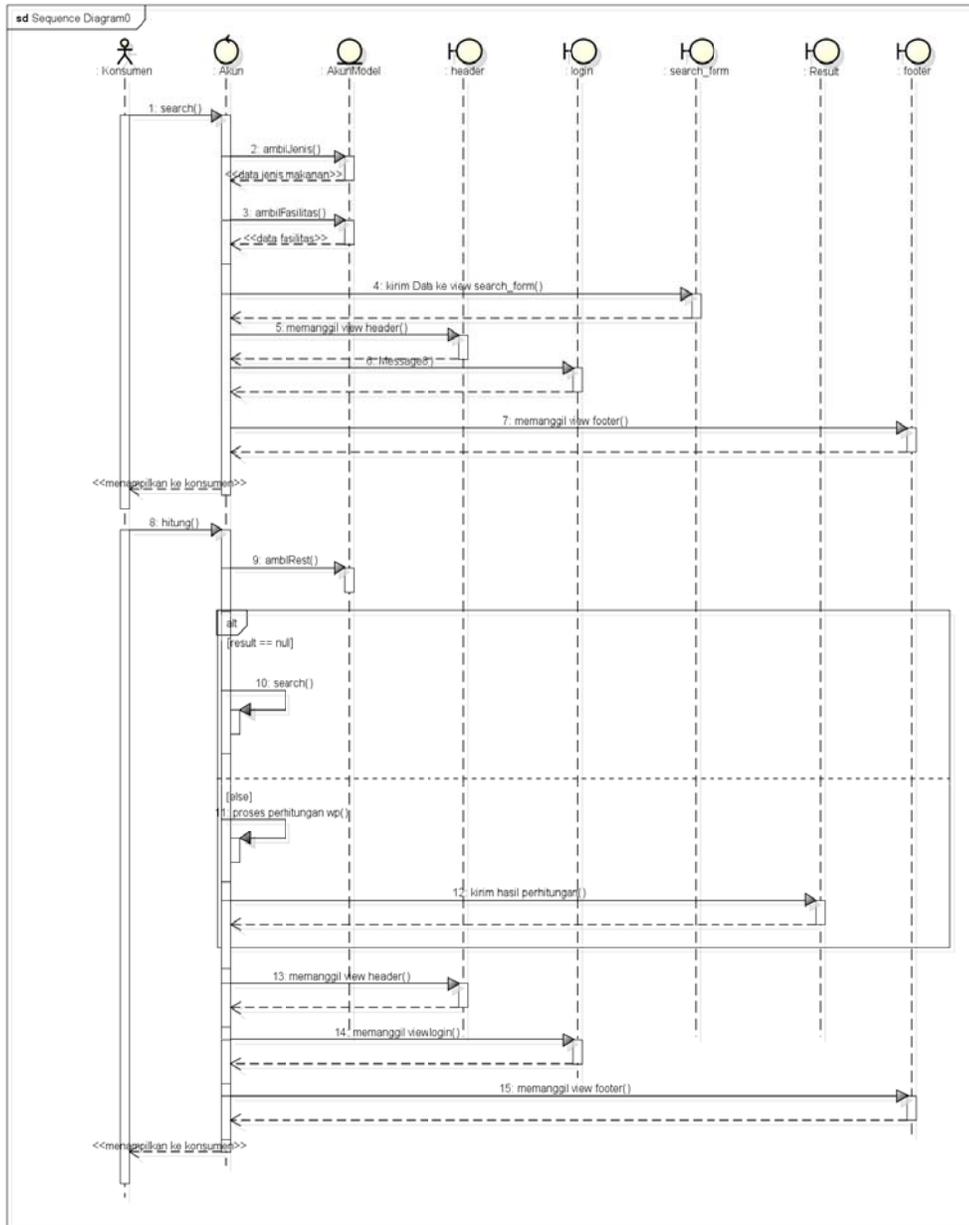
Gambar 4. 25Class Diagram Model MVC
 Sumber:[Perancangan]



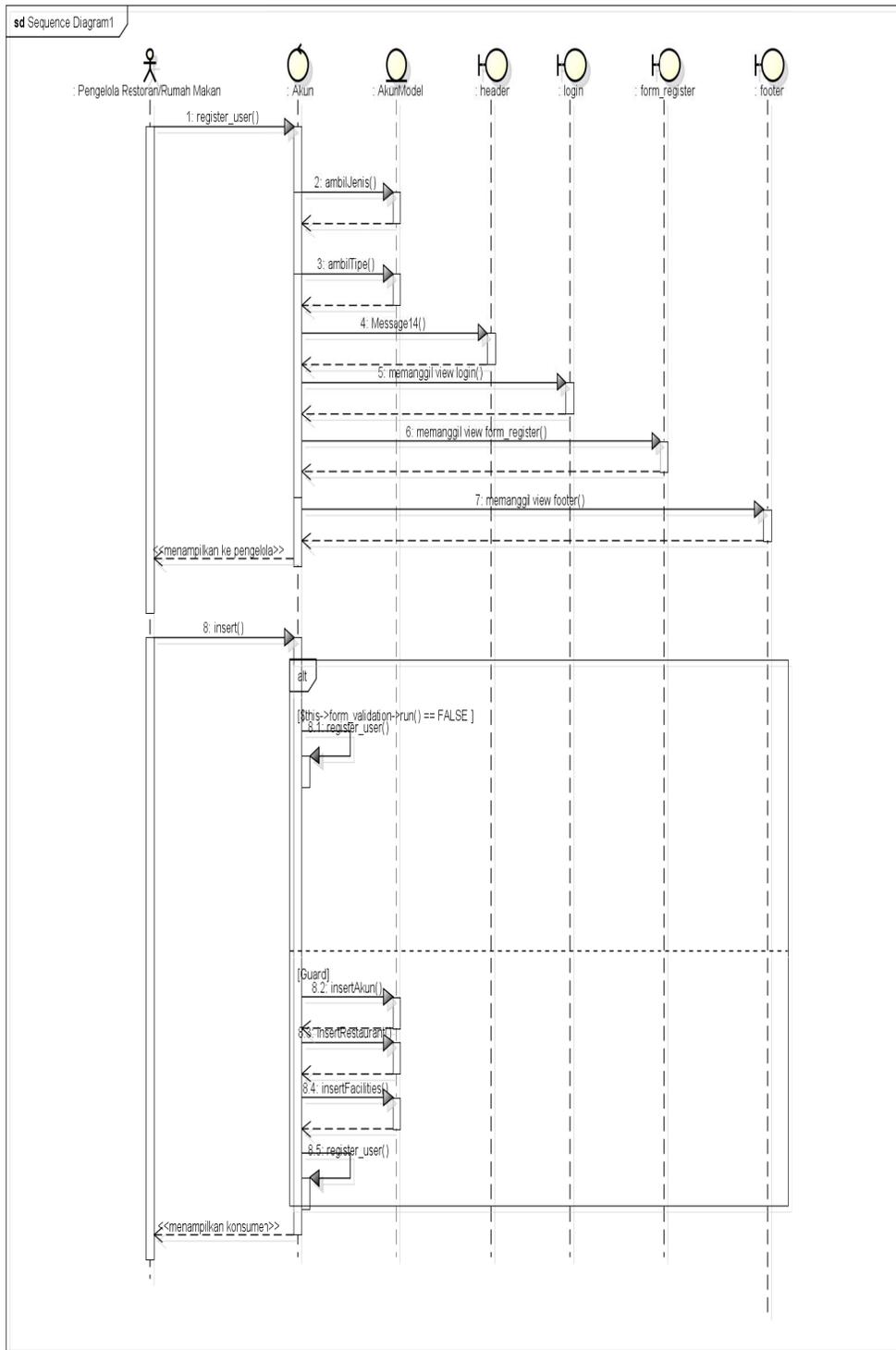
Gambar 4. 26Class Diagram Controller MVC
Sumber:[Perancangan]

4.5 Diagram Sekuensial (*Sequence Diagram*)

Diagram sekuensial digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan dari sebuah kejadian untuk menghasilkan keluaran tertentu. Berikut adalah beberapa diagram *sequence* untuk sistem rekomendasi.

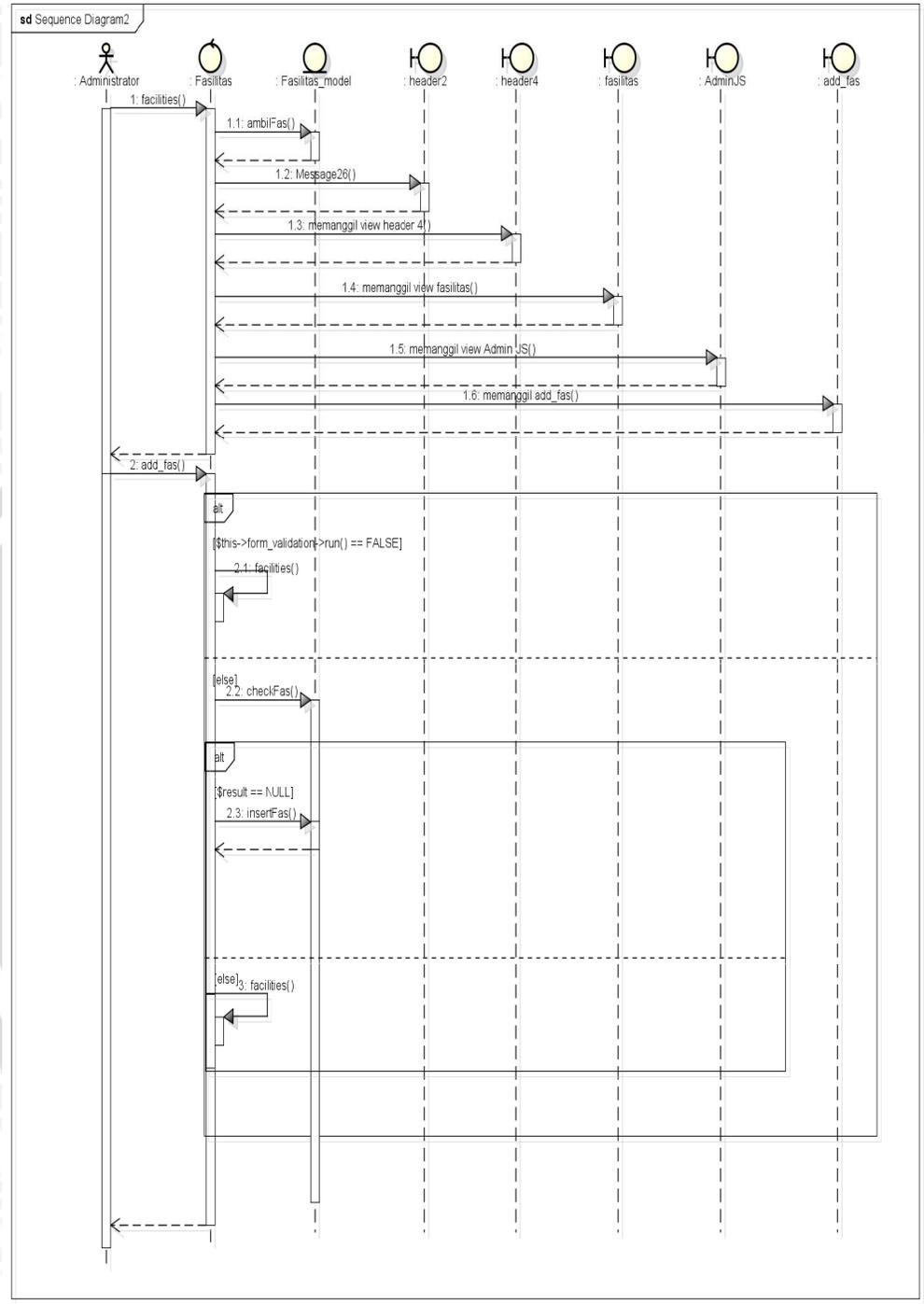


Gambar 4. 27 *Sequence Diagram* Cari Rekomendasi
 Sumber: [Perancangan]



powered by astah

Gambar 4. 28Sequence Diagram Register
Sumber:[Perancangan]



powered by astah

Gambar 4. 29 Sequence Diagram Modul Kelola Fasilitas
Sumber:[Perancangan]

4.6 Perancangan Basis Data

Perancangan basis data adalah perancangan manajemen data yang akan digunakan pada sistem. Basis data ini nantinya akan diatur oleh *software* yang disebut DBMS (*Database Management System*). Pada skripsi ini, perancangan basis data dimodelkan melalui *entity-relationship diagram* (ERD). ERD sistem rekomendasi ini ditunjukkan pada gambar 4.30.

Perancangan basis data dalam skripsi ini terdiri dari, yaitu: tabel *restaurant*, tabel komentar, tabel *promo*, tabel *menu*, tabel *account*, tabel *news*, tabel *restaurant_facilities*, tabel *facilities*, tabel *cuisine* dan *user_admin*. Berikut ini merupakan struktur tabel tabel serta keterangan masing masing tabel dan *field* yang ada pada *database*.

Entitas *restaurant* berisikan data-data utama rumah makan yang meliputi nama rumah makan, alamat rumah makan, lokasi rumah makan yang diambil dari *autocomplete google maps API*, nomor telepon, latitude, longitude, gambar rumah makan, *thumbnail* gambar rumah makan, tanggal daftar, dan deskripsi tempat, harga, kategori jenis makanan, website, uid, dan lain-lainya. Struktur tabel *restaurant* ditunjukkan pada Tabel 4.20 .

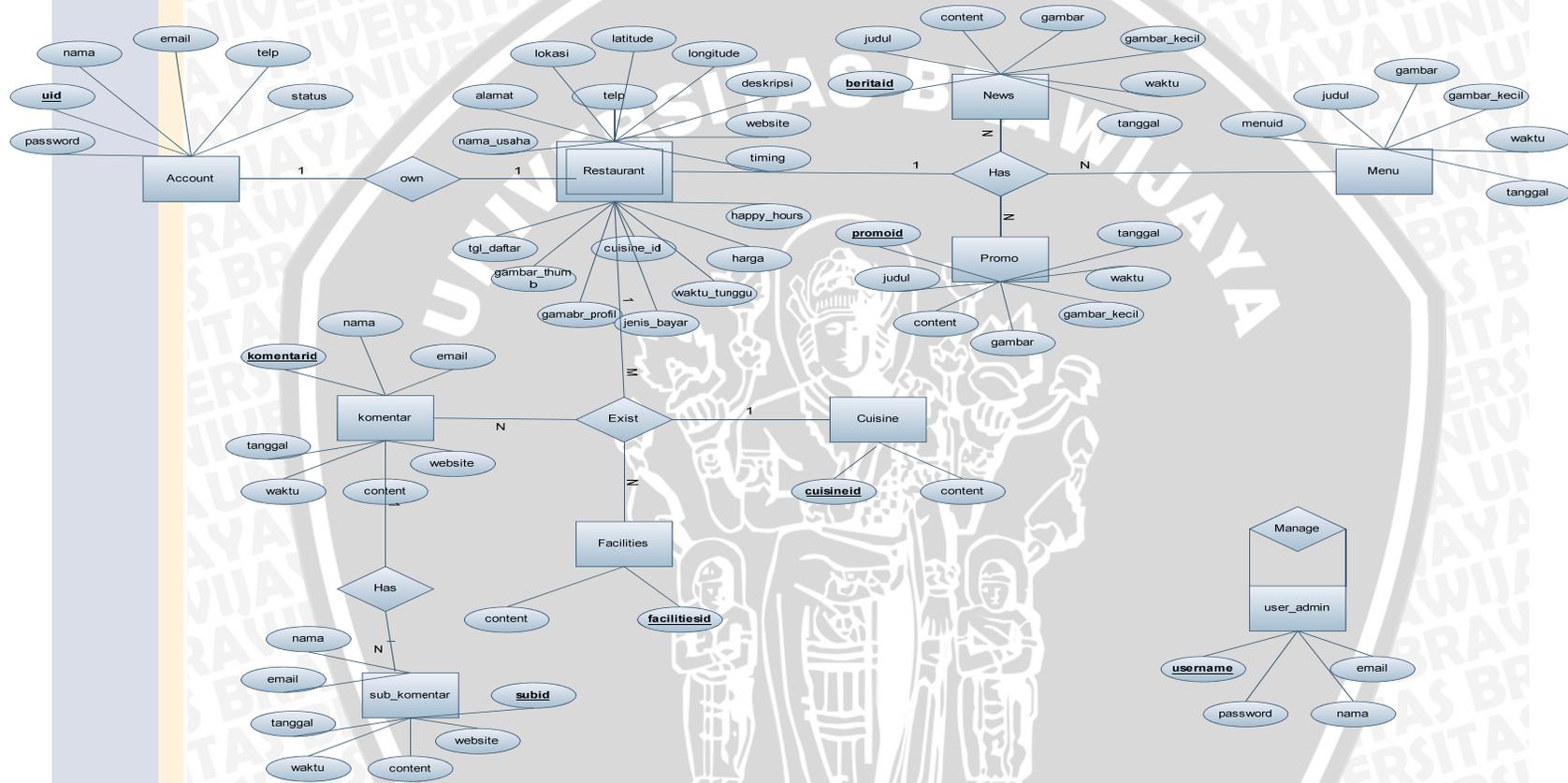
Tabel *promo* mempresentasikan data promo rumah makan. Tabel *promo* dapat dijelaskan pada tabel 4.21.

Tabel *news* mempresentasikan data berita dari rumah makan. Tabel *news* dapat dijelaskan pada tabel 4.22.

Tabel *restaurant_facilities* mempresentasikan data fasilitas yang dimiliki oleh rumah makan. Tabel *restaurant_facilities* dapat dijelaskan pada tabel 4.23.

Tabel *Cuisine* mempresentasikan data jenis makanan. Struktur tabel dapat dijelaskan pada tabel 4.25.

Tabel *menu* mempresentasikan data *menu* untuk rumah makan. Tabel *menu* terdiri dari *menuid*, *gambar_menu*, dan *uid*. Struktur tabel menu dapat dilihat pada tabel 4.26.



Gambar 4. 30 Perancangan Basis Data
Sumber:[Perancangan]

Tabel 4. 20 Struktur Tabel Restaurant

No	Nama Field	Tipe	Lebar	Keterangan
1	uid(fk)	Int	-	Id dari pengelola rumah makan
2	nama_usaha	Varchar	100	nama tempat usaha
3	Alamat_usaha	Text	50	Alamat rumah makan
4	Lokasi	Varchar	255	Alamat rumah makan yang diambil dari autocomplete google.
5	Telepon	Varchar	15	Nomor telepon rumah makan
6	Latitude	Double	-	Garis lintang yang akan digunakan untuk menampilkan peta lokasi
7	Longitude	Double	-	Garis bujur yang akan digunakan untuk menampilkan peta lokasi
8	gambar_profil	Varchar	255	Gambar restoran
9	gambar_thumb	Varchar	255	Gambar thumbnail dari kolom gambar_profil
10	tgl_daftar	Timestamp	-	Hari dan tanggal pengelola mendaftarkan rumah makan
11	deskripsi_tempat	Text		Deskripsi singkat restoran
12	cuisine_id	Varchar	15	Kode jenis makanan rumah makan
13	Harga	Int		Kode kategori harga rumah makan
14	Timing	Text		Keterangan hari dan jam buka rumah makan
15	website	Varchar	255	Website resmi yang dimiliki

				oleh rumah makan
16	waktu_tunggu	Int		Kode kategori waktu tunggu pemesanan di restoran tersebut
17	happy_hours	Text		Jam sibuk rumah makan tersebut
18	jenis_pembayaran	Varchar	50	Metode pembayaran yang disediakan restoran

Sumber: [Perancangan]

Tabel 4. 21 Struktur Tabel Promo

No	Nama Field	Tipe	Lebar	Keterangan
1	uid (FK)	Int	-	Id dari pengelola rumah makan
2	promoid (PK)	Int	-	Id untuk promo
3	Judul	Varchar	255	Judul promo
4	Content	Text	-	Isi dari promo
5	Waktu	Time	-	Waktu posting promo
6	Tanggal	Date	-	Tanggal posting promo
7	Gambar	Varchar	255	Gambar untuk promo
8	gambar_kecil	Varchar	255	Gambar thumbnail dari kolom gambar

Sumber: [Perancangan]

Tabel 4. 22 Struktur Tabel Berita

No	Nama Field	Tipe	Lebar	Keterangan
1	uid (FK)	Int	-	Id dari pengelola rumah makan
2	beritaid (PK)	Int	-	Id untuk berita
3	Judul	Varchar	255	Judul berita
4	Content	Text	-	Isi dari berita
5	Waktu	Time	-	Waktu berita
6	Tanggal	Date	-	Tanggal posting berita

7	Gambar	Varchar	255	Gambar untuk berita
8	gambar_kecil	Varchar	255	Gambar thumbnail dari kolom gambar

Sumber: [Perancangan]

Tabel 4. 23 Struktur Tabel Restaurant Facilities

No	Nama Field	Tipe	Lebar	Keterangan
1	uid (FK)	Int	-	Id dari pengelola rumah makan
2	facilities_id (FK)	Int	-	Id dari jenis fasilitas yang dimiliki.

Sumber: [Perancangan]

Tabel *facilities* mempresentasikan data jenis fasilitas yang dimiliki oleh rumah makan. Struktur tabel dapat dijelaskan pada tabel 4.24.

Tabel 4. 24 Struktur Tabel Facilities

No	Nama Field	Tipe	Lebar	Keterangan
1	facilities_id (pk)	Int	-	Id dari fasilitas
2	<i>Content</i>	Varchar	255	Keterangan dari jenis/kode fasilitas

Sumber: [Perancangan]

Tabel 4. 25 Struktur Tabel Cuisine

No	Nama Field	Tipe	Lebar	Keterangan
1	cuisine_id(pk)	Int	-	Id dari jenis makanan
2	Content	Varchar	255	Keterangan jenis makanan

Sumber: [Perancangan]

Tabel 4. 26 Struktur Tabel Menu

No	Nama Field	Tipe	Lebar	Keterangan
1	uid(fk)	Int	-	Id dari pengelola rumah makan
2	menuid (pk)	Int	-	Id untuk menu
3	Gambar	Varchar	255	Gambar untuk menu
4	Judul	Varchar	255	Judul untuk posting menu
5	gambar_kecil	Varchar	255	Gambar thumbnail dari kolom gambar
6	Waktu	Time	-	Waktu posting menu
7	Tanggal	Date	-	Tanggal posting menu

Sumber: [Perancangan]

Tabel komentar mempresentasikan data komentar untuk rumah makan yang ada pada database. Tabel ini terdiri dari uid, komentarid, nama, email, komentar, dan waktu. Struktur tabel komentar dapat dilihat pada tabel 4.27.

Tabel 4. 27 Struktur Tabel Komentar

No	Nama Field	Tipe	Lebar	Keterangan
1	uid (fk)	Int	-	Id dari pengelola rumah makan
2	komentarid (pk)	Int	-	Id untuk komentar
3	Nama	Varchar	50	Nama pemberi komentar
4	Email	Varchar	50	Alamat email pemberi komentar
5	Komentar	Text	-	Isi komentar

6	Waktu	Time	-	Waktu komentar disimpan di database
7	Tanggal	Date	-	Tanggal komentar disimpan di database
8	Website	Varchar	50	Alamat website (<i>optional</i>)

Sumber: [Perancangan]

Tabel *sub_komentar* merupakan tabel bantu yang digunakan untuk melakukan penyimpanan balasan dari komentar yang ada pada sistem. Tabel ini terdiri dari *uid*, *komentarid*, *nama*, *email*, *komentar*, dan *waktu*. Struktur tabel komentar dapat dilihat pada tabel 4.28.

Tabel 4. 28 Struktur Tabel Sub Komentar

No	Nama Field	Tipe	Lebar	Keterangan
1	<i>uid</i> (fk)	Int	-	Id dari pengelola rumah makan
2	<i>komentarid</i> (fk)	Int	-	Id untuk komentar
3	<i>subid</i> (pk)	Int		Id untuk <i>sub_komentar</i>
4	Nama	Varchar	50	Nama
5	Email	Varchar	50	Alamat email
6	Komentar	Text	-	Isi balasan komentar
7	Waktu	Time	-	Waktu data disimpan di database
8	Tanggal	Date	-	Tanggal data disimpan di database
9	<i>website</i>	Varchar	50	Alamat website (<i>optional</i>)

Sumber: [Perancangan]

Tabel *account* mempresentasikan data pengelola rumah makan berupa *uid* yang merupakan *primary key*, *password*, *nama* pengelola, *email* dan nomor telepon pengelola. Struktur tabel *user_pengelola* ditunjukkan pada tabel 4.29.

Tabel 4. 29 Struktur Tabel Account

No	Nama Field	Tipe	Lebar	Keterangan
----	------------	------	-------	------------

1	uid(pk)	Int	-	Id dari pengelola rumah makan.
2	password	Varchar	50	Password yang digunakan untuk autentifikasi
3	nama	Varchar	50	Nama pengelola
4	email	Varchar	50	Email dari pengelola rumah makan
5	telp	Varchar	25	Nomor telepon pengelola rumah makan

Sumber: [Perancangan]

Tabel user_admin mempresentasikan data administrator yang berwenang untuk memantau dan memelihara sistem. Struktur tabel user_admin dijelaskan pada tabel 4.30.

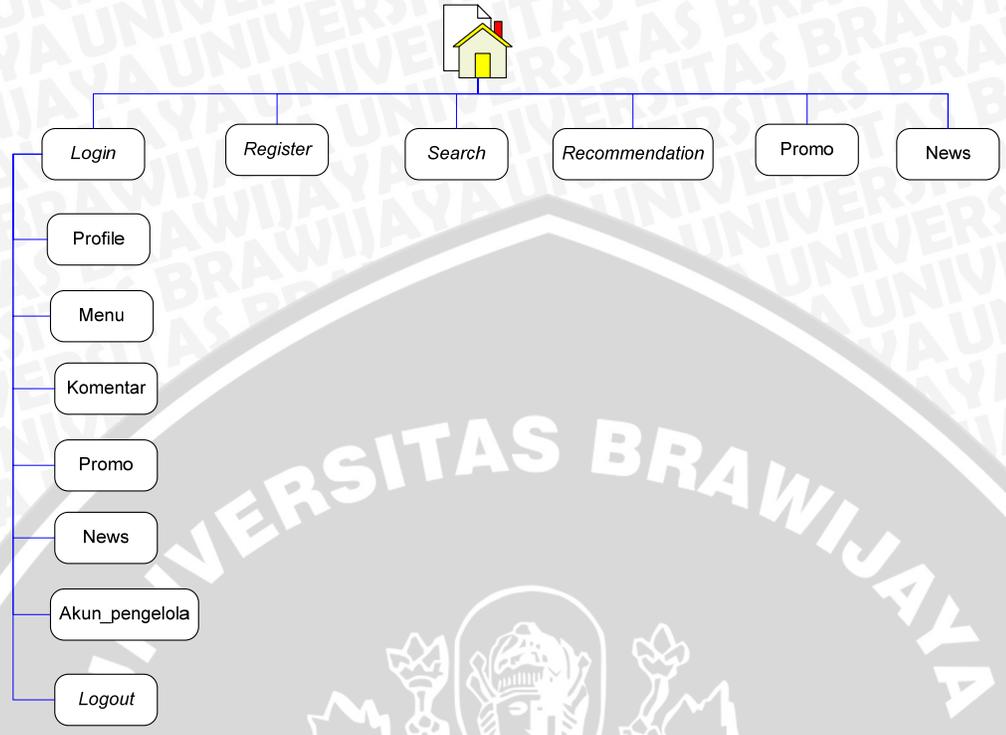
Tabel 4. 30 Struktur Tabel User Admin

No	Nama Field	Tipe	Lebar	Keterangan
1	username (pk)	Varchar	50	Username untuk administrator
2	passwords	Varchar	50	Password yang digunakan untuk autentifikasi
3	Nama	Varchar	50	Nama admin
4	Email	Varchar	50	Email admin

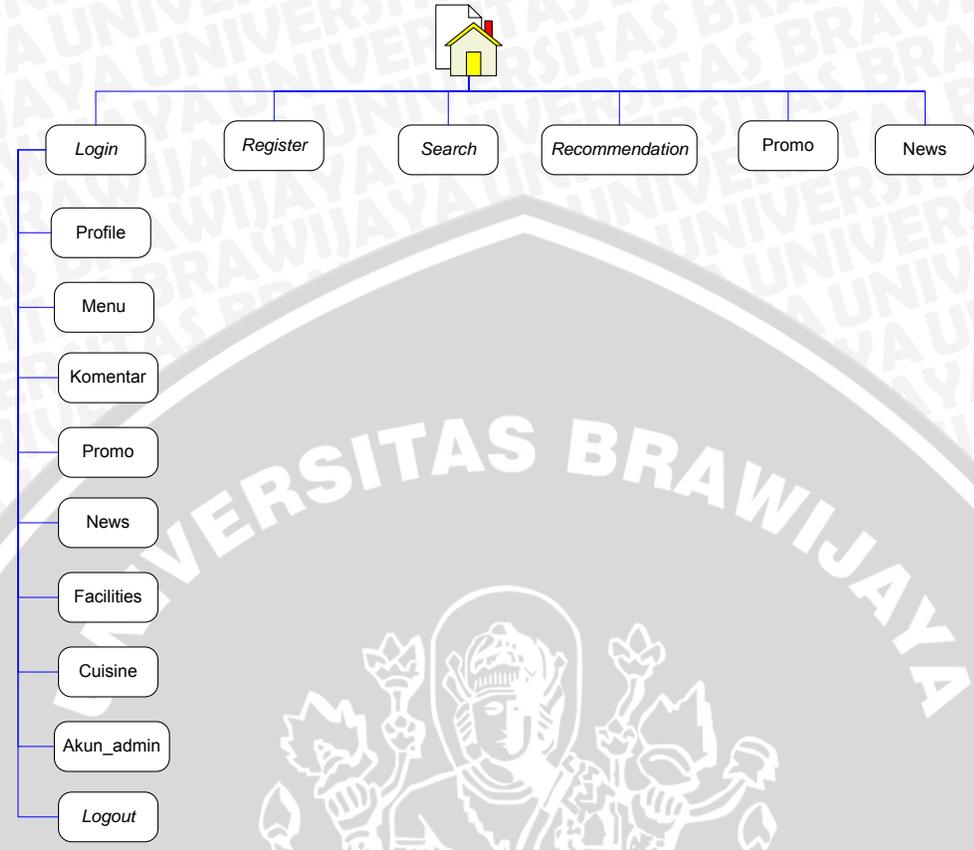
Sumber: [Perancangan]

4.7 Perancangan Antarmuka

Pada bagian ini akan dijelaskan tentang perancangan antarmuka sistem rekomendasi kuliner. Tujuan dari perancangan antarmuka adalah untuk memudahkan interaksi antara pengguna sistem dengan sistem itu sendiri. Perancangan antarmuka mewakili keadaan sebenarnya dari sistem yang dibangun. Gambar 4.29 menunjukkan *sitemap* dari sistem rekomendasi kuliner.

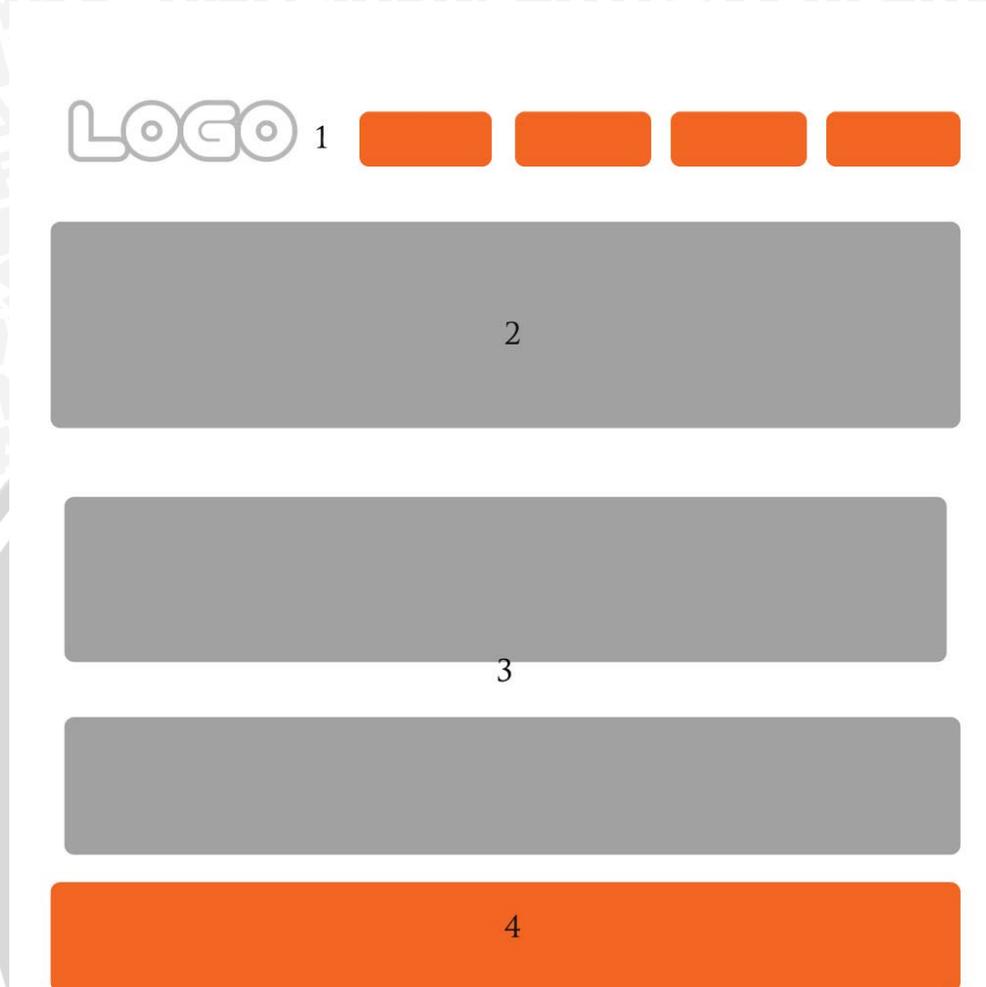


Gambar 4. 31Sitemap Sistem Rekomendasi Kuliner Pengelola Rumah Makan
Sumber:[Perancangan]



Gambar 4. 32Sitemap Sistem Rekomendasi Kuliner Administrator
Sumber:[Perancangan]

1. Perancangan Halaman Utama

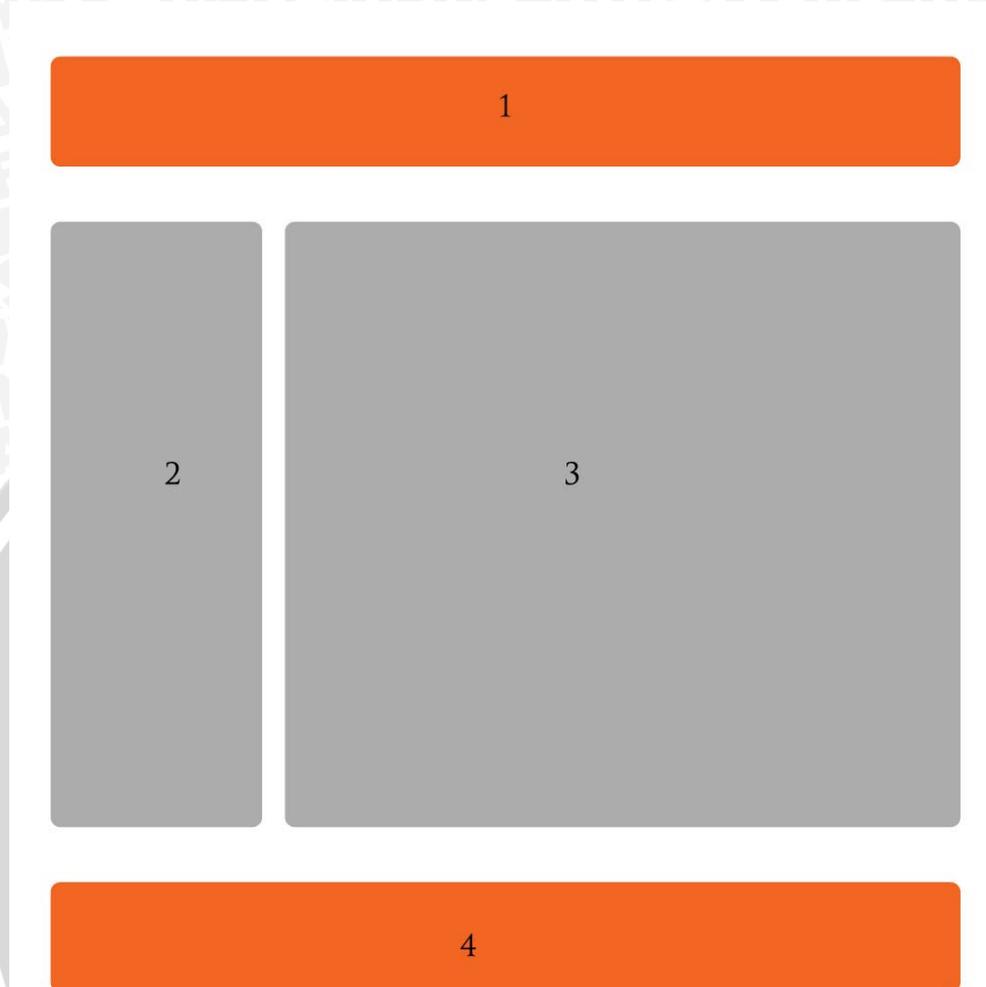


Gambar 4.33 Perancangan Antarmuka Halaman Utama
Sumber:[Perancangan]

Halaman utama menampilkan sekilas mengenai promo dan beberapa fitur yang ada pada sistem. Gambar 4.33 memiliki keterangan sebagai berikut :

1. *Navigation bar* berisikan identitas dari sistem rekomendasi kuliner ini dan tombol navigasi halaman utama.
2. *Carousel* menggunakan fungsi *carousel bootstrap*
3. *Content* yang berisikan penjelasan tentang apa saja yang dapat dilakukan sistem rekomendasi ini.
4. *Content*
5. *Footer* berisi hak cipta dan tahun dari sistem rekomendasi kuliner ini.

2. Perancangan Halaman *Administrator*

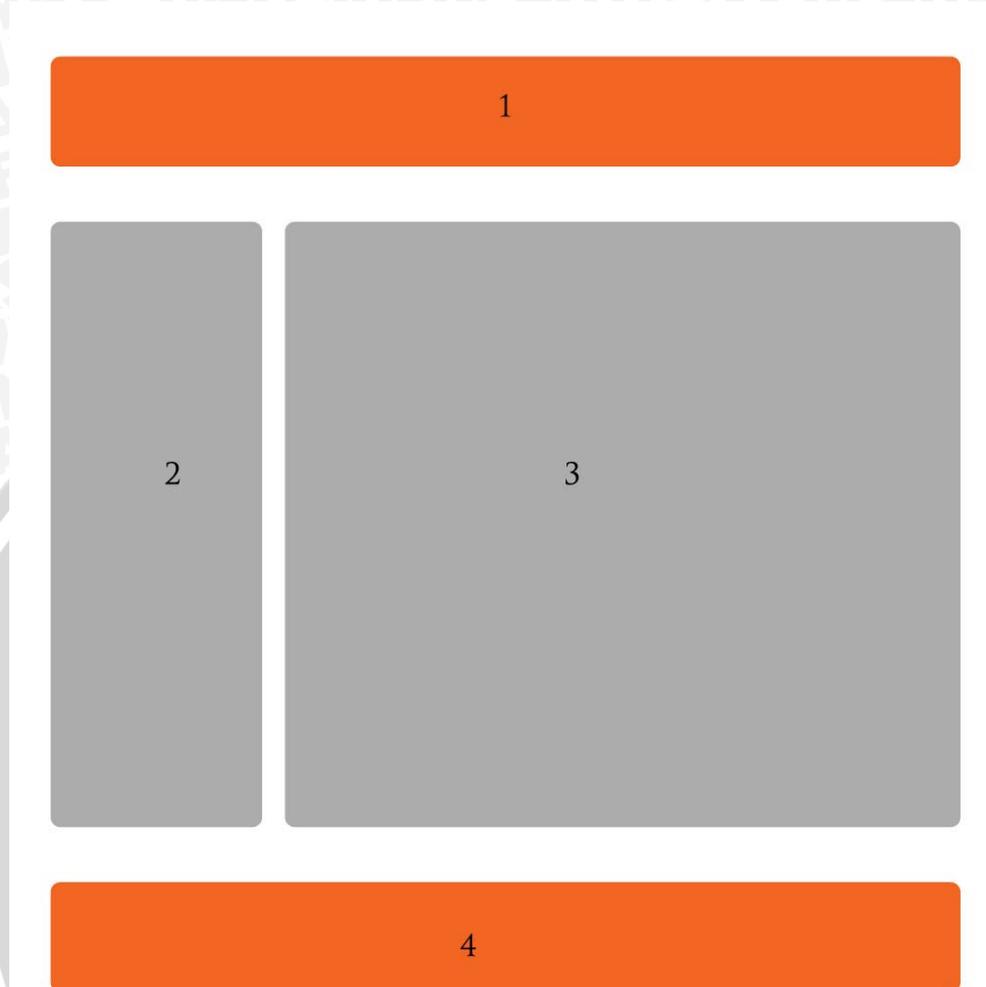


Gambar 4. 34 Perancangan Antarmuka Halaman *Administrator*
Sumber:[Perancangan]

Halaman *Administrator* menampilkan fitur-fitur yang ada pada halaman *administrator* setelah melalui proses autentifikasi. Gambar 4.34 memiliki keterangan sebagai berikut :

1. *Navigation bar* berisi identitas dari sistem rekomendasi kuliner ini dan tombol navigasi untuk *administrator*.
2. Berisikan *content* dari menu-menu yang ada pada navigasi dan *side bar*
3. *Sidebar* berisikan menu-menu apa saja yang dapat dilakukan oleh *administrator*.
4. *Footer* berisi hak cipta dan tahun dari sistem rekomendasi kuliner ini

3. Perancangan Halaman Pengelola Rumah makan

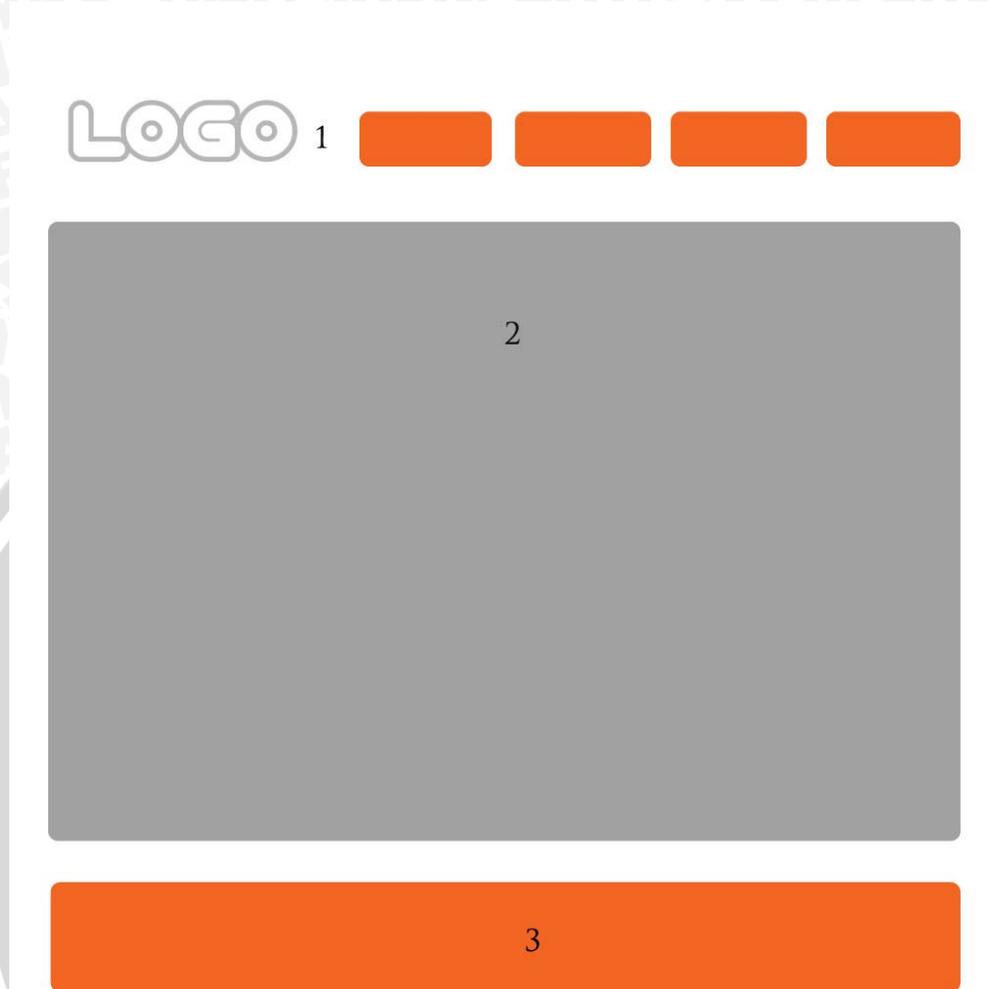


Gambar 4. 35 Perancangan Antarmuka Halaman Pengelola Rumah Makan
Sumber:[Perancangan]

Halaman pengelola rumah makan menampilkan fitur-fitur yang ada pada halaman pengelola rumah makan setelah melalui proses autentifikasi. Gambar 4.35 memiliki keterangan sebagai berikut :

1. *Navigation bar* berisi identitas dari sistem rekomendasi kuliner ini dan tombol navigasi untuk pengelola restoran.
2. Berisikan *content* dari menu-menu yang ada pada navigasi dan *side bar*
3. *Sidebar* berisikan menu-menu apa saja yang dapat dilakukan oleh pengelola restoran.
4. *Footer* berisi hak cipta dan tahun dari sistem rekomendasi kuliner ini.

4. Perancangan Halaman Promo

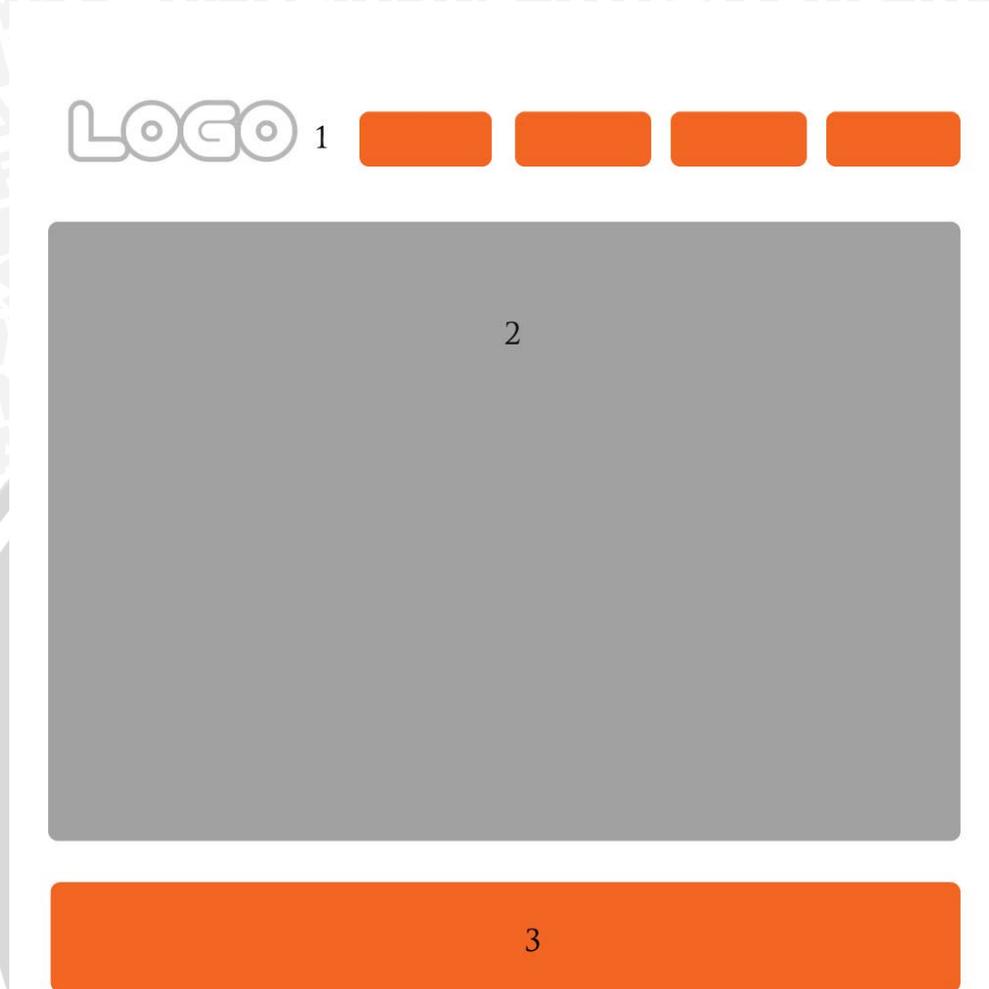


Gambar 4.36 Perancangan Antarmuka Halaman Promo
Sumber:[Perancangan]

Halaman Promo akan menampilkan semua promo/berita yang ada pada sistem. Gambar 4.36 memiliki keterangan sebagai berikut :

1. *Navigation bar* berisi identitas dari sistem rekomendasi kuliner ini dan tombol navigasi utama.
2. *Content* yang berisikan promo rumah makan
3. *Footer* berisi hak cipta dan tahun dari sistem rekomendasi kuliner ini

5. Perancangan Halaman Berita/News



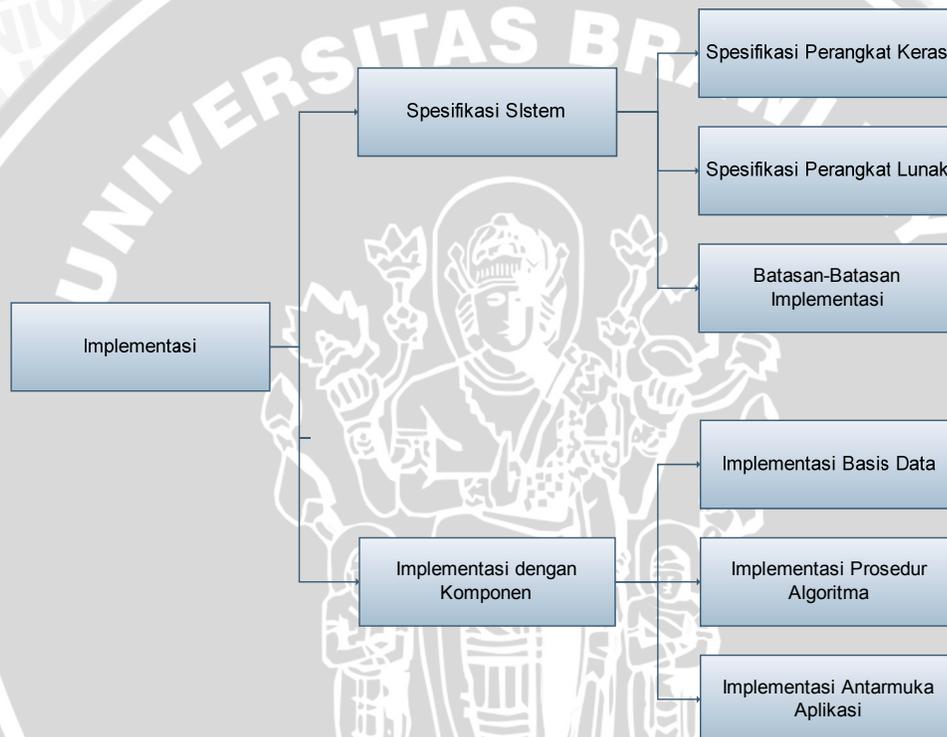
Gambar 4. 37 Perancangan Antarmuka Halaman Berita
Sumber:[Perancangan]

Halamanberita/news memiliki keterangan sebagai berikut :

1. *Navigation bar* berisi identitas dari sistem rekomendasi kuliner ini dan tombol navigasi utama.
2. *Content* yang berisikan berita terbaru rumah makan.
3. *Footer* berisi hak cipta dan tahun dari sistem rekomendasi kuliner ini.

4.8 Implementasi Sistem

Pada bab ini akan dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan perancangan perangkat lunak. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi database, implementasi prosedur algoritma, dan implementasi antarmuka perangkat lunak seperti pada gambar 4.38.



Gambar 4. 38 Diagram Implementasi
Sumber:[Implementasi]

4.8.1 Implementasi Sistem

Implementasi sistem akan membahas mengenai bagaimana implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dan analisis kebutuhan dan perancangan perangkat lunak. Sub bab ini terdiri dari penjelasan spesifikasi sistem, batasan-batasan implementasi, implementasi database, implementasi algoritma, dan implementasi antarmuka perangkat lunak.

4.8.2 Spesifikasi Sistem

Hasil dari analisis kebutuhan dan perancangan lunak yang telah dijelaskan pada tahap analisis kebutuhan dan tahap perancangan sistem menjadi dasar untuk dilakukan implementasi menjadi sebuah aplikasi *mobile web* sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

4.8.2.1 Spesifikasi Perangkat Keras

Dalam pengembangan sistem ini menggunakan sebuah komputer dengan spesifikasi *processor*, *memory*, dan *display* yang akan dijelaskan pada tabel 4.31.

Tabel 4. 31 Tabel Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>System Model</i>	HP Pavilion G4-1116TX Notebook
<i>Processor</i>	Intel Core i3-2350M 2.4 GHz
<i>Memory</i>	4 GB DDR3 RAM
<i>Display</i>	14.0 inch, 16:9, 1366x768 pixels

Sumber : [Implementasi]

4.8.2.2 Spesifikasi Perangkat Lunak

Sistem rekomendasi ini dikembangkan menggunakan sebuah *notebook* dengan spesifikasi perangkat lunak yang akan dijelaskan pada tabel 4.32 berikut.

Tabel 4. 32 Tabel Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 7 Home Basic
Bahasa Pemrograman	PHP, Javascript, CSS, HTML
<i>Framework</i>	<i>CodeIgniter</i> 2.1.0
<i>Template Interface Program</i>	<i>Bootstrap</i> 3
API	Google Map API v3
<i>Web Server</i>	XAMPP 1.7.7
<i>Database</i>	MySQL dengan <i>tool</i> Heidi SQL 8.3
<i>Browser</i>	Google Chrome 35.0.1916.114 m

Sumber : [Implementasi]

4.8.3 Batasan Implementasi

Pada implementasi perangkat lunak sistem rekomendasi terdapat batasan-batasan dalam prosesnya, yaitu :

1. Sistem rekomendasi kuliner ini dirancang untuk dapat dijalankan di berbagai *device* (desain responsif).
2. Sistem ini dikembangkan menggunakan Bahasa pemrograman PHP, Javascript, HTML, dan CSS dengan *frameworkcodeigniter*. dan *framework bootstrap* sebagai *interfacesistem*.
3. Metode rekomendasi yang digunakan adalah metode pembobotan *weighted product*.
4. Kriteria yang digunakan dalam perhitungan rekomendasi kuliner adalah harga, lokasi, dan ketersediaan fasilitas yang dibutuhkan oleh konsumen/calon pelanggan.
5. Peta yang digunakan petunjuk arah dalam sistem rekomendasi kuliner ini menggunakan *Google Map API*.

4.8.4 Implementasi Basis Data

Bab ini akan membahas bagaimana implementasi basis data berdasarkan perancangan basis data yang telah didapatkan pada analisis kebutuhan. Implementasi basis data dibuat sesuai dengan perancangan basis data yang telah dibuat.

4.8.5 Implementasi Prosedur Algoritma

Tahap ini akan menjelaskan bagaimana rancangan algoritma yang telah dibuat diimplementasikan dengan menggunakan bahasa pemrograman. Gambar 4.38 akan menunjukkan implementasi perhitungan rekomendasi dengan menggunakan metode *weighted product*.

```

1315     public function hitung()
1316     {
1317
1318
1319         $w1= $this->input->post('q1');
1320         $w3= $this->input->post('q3');
1321         $w4= $this->input->post('q4');
1322
1323         $jenis= $this->input->post('jenis_makanan');
1324         $harga = $this->input->post('harga');
1325         if($harga == "range0")
1326         {
1327             $price = 1;
1328             $w1 = 0;
1329         }
1330         else if($harga == "range1")
1331         {
1332             $price = 2;
1333         }
1334         else if($harga == "range2")
1335         {
1336             $price = 3;
1337         }
1338         else if($harga == "range3")
1339         {
1340             $price = 4;
1341         }
1342         else if($harga == "range4")
1343         {
1344             $price = 5;

```

Gambar 4. 40 Impementasi Algoritma Dengan Fungsi Hitung
Sumber : [Implementasi]

```

1420     function hitung1($jenis,$price,$jauh,$fixw1,$fixw2,$fixw3,$fixw4,$latitude,$longitude,$jenis,$lokasi_awal)
1421     {
1422         $data['query'] = $this->akun_model->ambilRest($jenis,$latitude,$longitude);
1423         if($data['query'] == null)
1424         {
1425             $this->session->set_flashdata('err_message', 'No Result Found');
1426             redirect('akun/search');
1427         }
1428         else
1429         {
1430             foreach($data['query'] as $row)
1431             {
1432                 if($row->distance < 1)
1433                 {
1434                     $jaraklokasi=2;
1435                 }
1436                 elseif(($row->distance >= 1) and ($row->distance <=2))
1437                 {
1438                     $jaraklokasi=3;
1439                 }
1440                 elseif($row->distance > 2)
1441                 {
1442                     $jaraklokasi=4;
1443                 }
1444             }

```

Gambar 4. 41 Implementasi Algoritma dengan Fungsi Hitung1()
Sumber:[Implementasi]

```

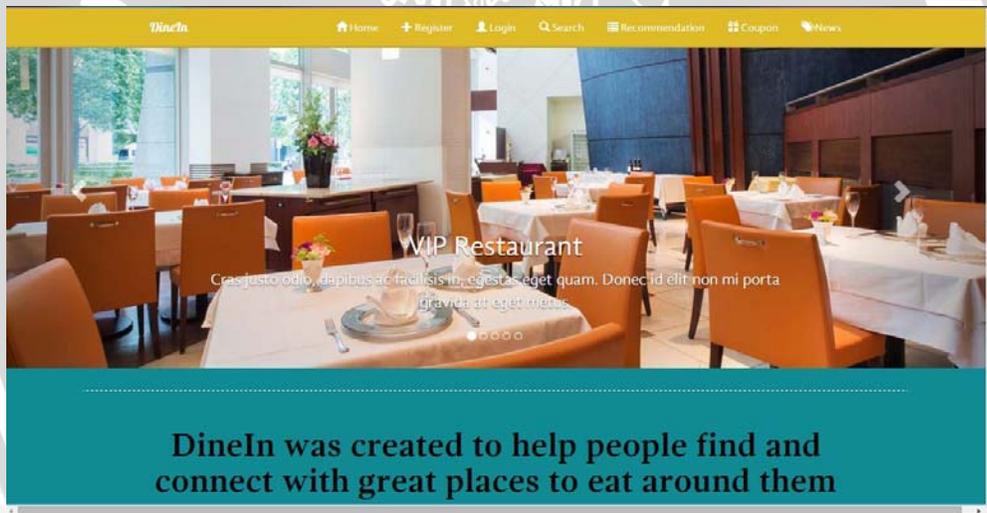
1565
1566     for($x=0;$x<$jum;$x++)
1567     {
1568         $vektor_v = $ta[$x]/$akhir;
1569         $v = $vektor_v;
1570         $nama = $nama_usaha[$x];
1571         $user_id = $uid[$x];
1572         $loc = $lokasi[$x];
1573         $data['item'][] = array('uid' => $user_id,
1574             'nama' => $nama,
1575             'lokasi' => $loc,
1576             'hasil' => $v
1577         );
1578     }
1579
1580
    
```

Gambar 4. 42 Implementasi Algoritma untuk Menampilkan Rekomendasi
 Sumber:[Implementasi]

4.8.6 Implementasi Antarmuka Aplikasi

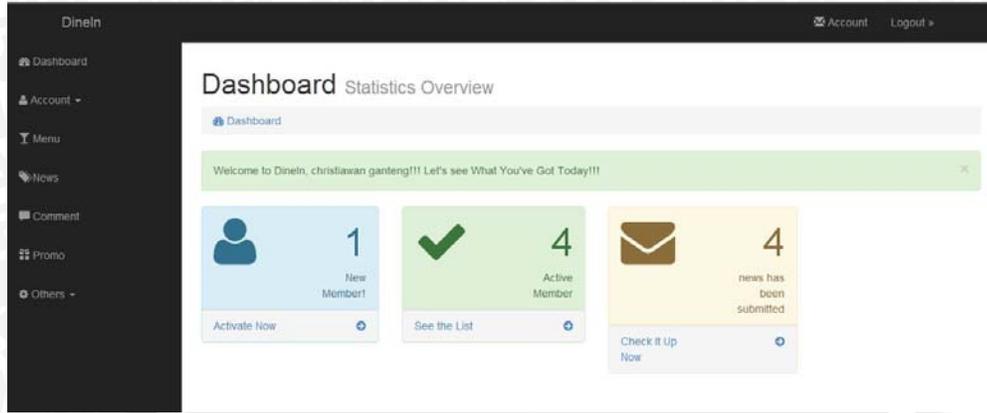
Tahap ini akan menjelaskan tentang implementasi antarmuka sistem rekomendasi kuliner sesuai dengan perancangan yang telah dibuat.

1. Tampilan Halaman Utama



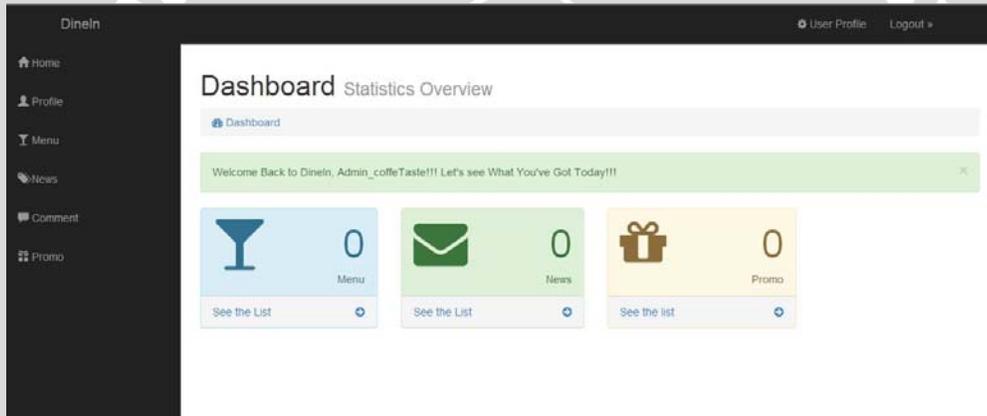
Gambar 4. 43 Antarmuka Halaman Utama Sistem Rekomendasi Kuliner
 Sumber:[Implementasi]

2. Tampilan Halaman Administrator



Gambar 4. 44 Antarmuka Halaman Administrator
Sumber:[Implementasi]

3. Tampilan Halaman Pengelola Restoran



Gambar 4. 45 Antarmuka Halaman Pengelola Restoran
Sumber:[Implementasi]

4. Tampilan Halaman Lihat Promo

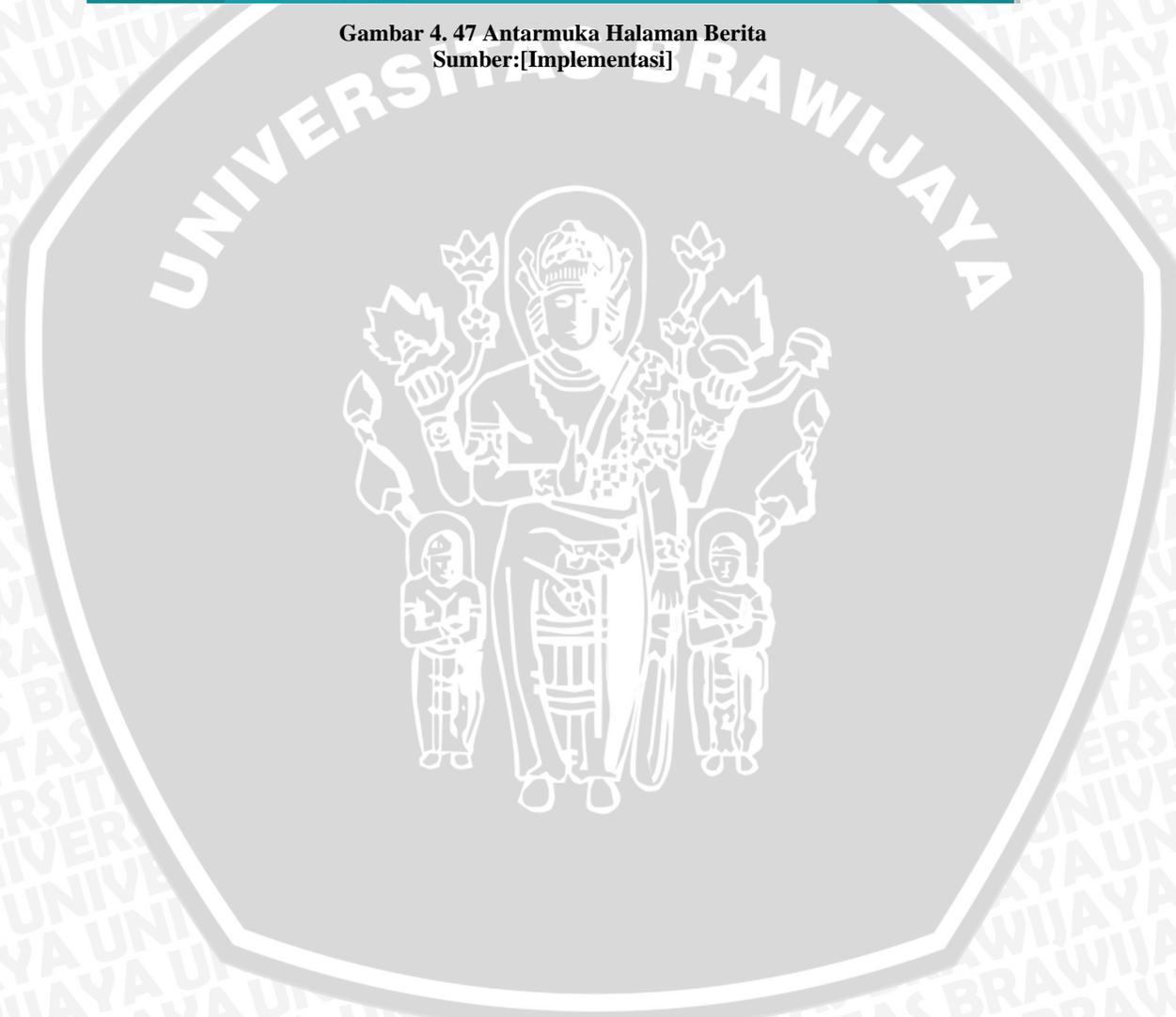


Gambar 4. 46Antarmuka Halaman Promo
Sumber:[Implementasi]

5. Tampilan Halaman Berita



Gambar 4. 47 Antarmuka Halaman Berita
Sumber:[Implementasi]



BAB V PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang proses pengujian dan analisis terhadap sistem rekomendasi kuliner yang dikembangkan. Proses pengujian dilakukan melalui enam tahapan, yaitu : pengujian *basis path*, pengujian validasi, pengujian kompatibilitas, pengujian akurasi sistem rekomendasi kuliner, dan *user acceptance testing*. Pada pengujian validasi dan kompatibilitas menggunakan teknik pengujian *black box testing*,



Gambar 5. 1 Diagram Pengujian dan Analisis
Sumber : Pengujian dan Analisis

5.1 Pengujian

Proses pengujian dilakukan melalui limatahap seperti yang telah digambarkan pada diagram yang digambarkan.

5.1.1 Pengujian *Basis Path*

Penulisan laporan skripsi ini hanya dicantumkan hasil pengujian *basis path* yang berhubungan dengan rekomendasi restoran.

5.1.1.1 Pengujian untuk fungsi Search()

Fungsi Search() merupakan fungsi untuk menampilkan form inputan kriteria-kriteria yang digunakan untuk perhitungan rekomendasi. Gambar 5.2 menunjukkan prosedur untuk fungsi Search().

NAMA ALGORITMA Search

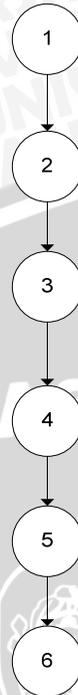
DESKRIPSI

Proses :

1. Memanggil fungsi ambilJenis pada akun_model
2. Memanggil fungsi ambilFasilitas pada akun_model
3. Memanggil view header
4. Memanggil view login
5. Memanggil view search_from
6. Memanggil view footer

Keluaran :

Menampilkan form untuk perhitungan rekomendasi



Gambar 5.2 Basis Path Fungsi Search()
Sumber: Pengujian dan Analisis

$$N(\text{Path}) = 5$$

$$P = 6$$

$$\text{Region} = 1$$

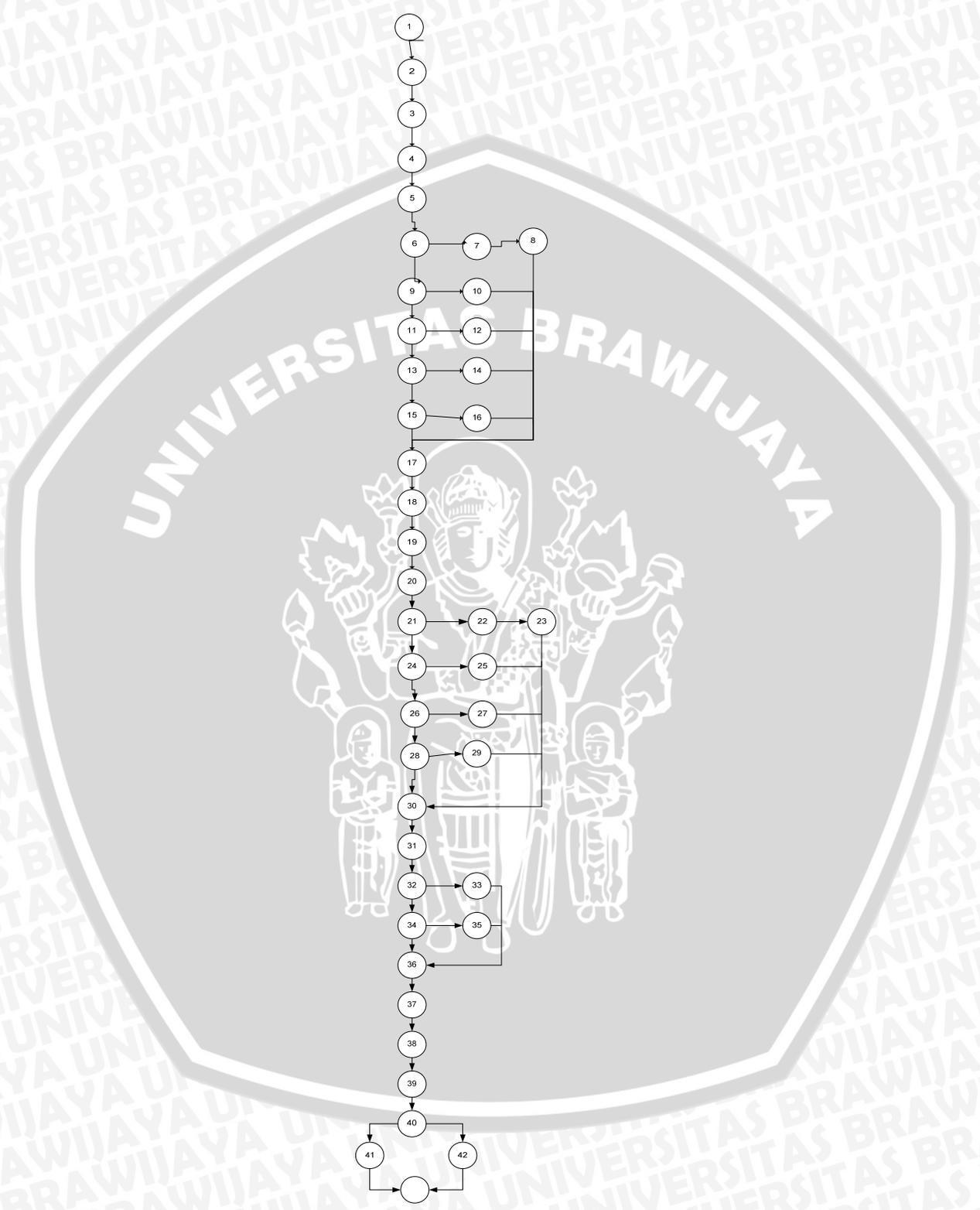
$$\text{Predikat} = 0$$

$$\begin{aligned} V &= n - P + 2 \\ &= 5 - 6 + 2 \\ &= 1 \end{aligned}$$

$$\begin{aligned} V &= \text{Predikat} + 1 \\ &= 0 + 1 \\ &= 1 \end{aligned}$$

5.1.1.2 Fungsi untuk fungsi hitung()

Fungsi hitung() merupakan fungsi untuk menampung semua inputan dari konsumen . Gambar 5.3 menunjukkan prosedur untuk fungsi hitung().



Gambar 5. 3Basis Path Fungsi hitung()
Sumber:[Pengujian dan Analisis]



$$N(\text{Path}) = 54$$

$$P = 43$$

$$\text{Region} = 13$$

$$\text{Predikat} = 12$$

$$\begin{aligned} V &= n - P + 2 \\ &= 54 - 43 + 2 \\ &= 13 \end{aligned}$$

$$\begin{aligned} V &= \text{Predikat} + 1 \\ &= 12 + 1 \\ &= 13 \end{aligned}$$

5.1.1.3 Fungsi hitung1()

Fungsi hitung1() merupakan fungsi menghitung rekomendasi jika kriteria fasilitas berisi value *any* dan bobot fasilitas menjadi 0. Gambar 5.4 menunjukkan pengujian *basis path* fungsi hitung1().

$$N(\text{Path}) = 42$$

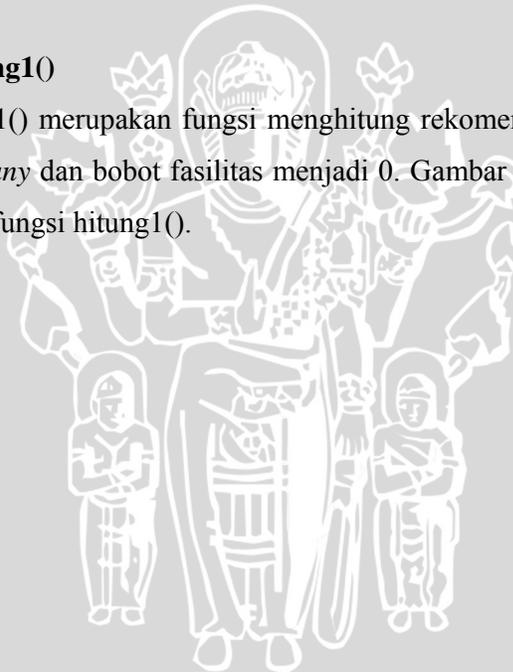
$$P = 37$$

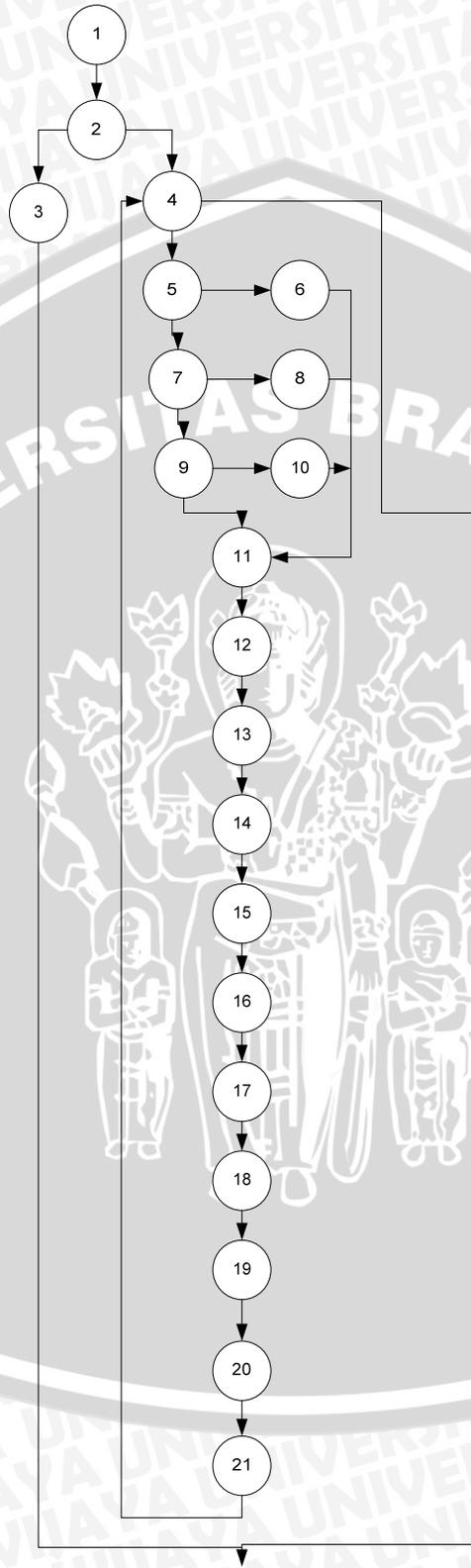
$$\text{Region} = 7$$

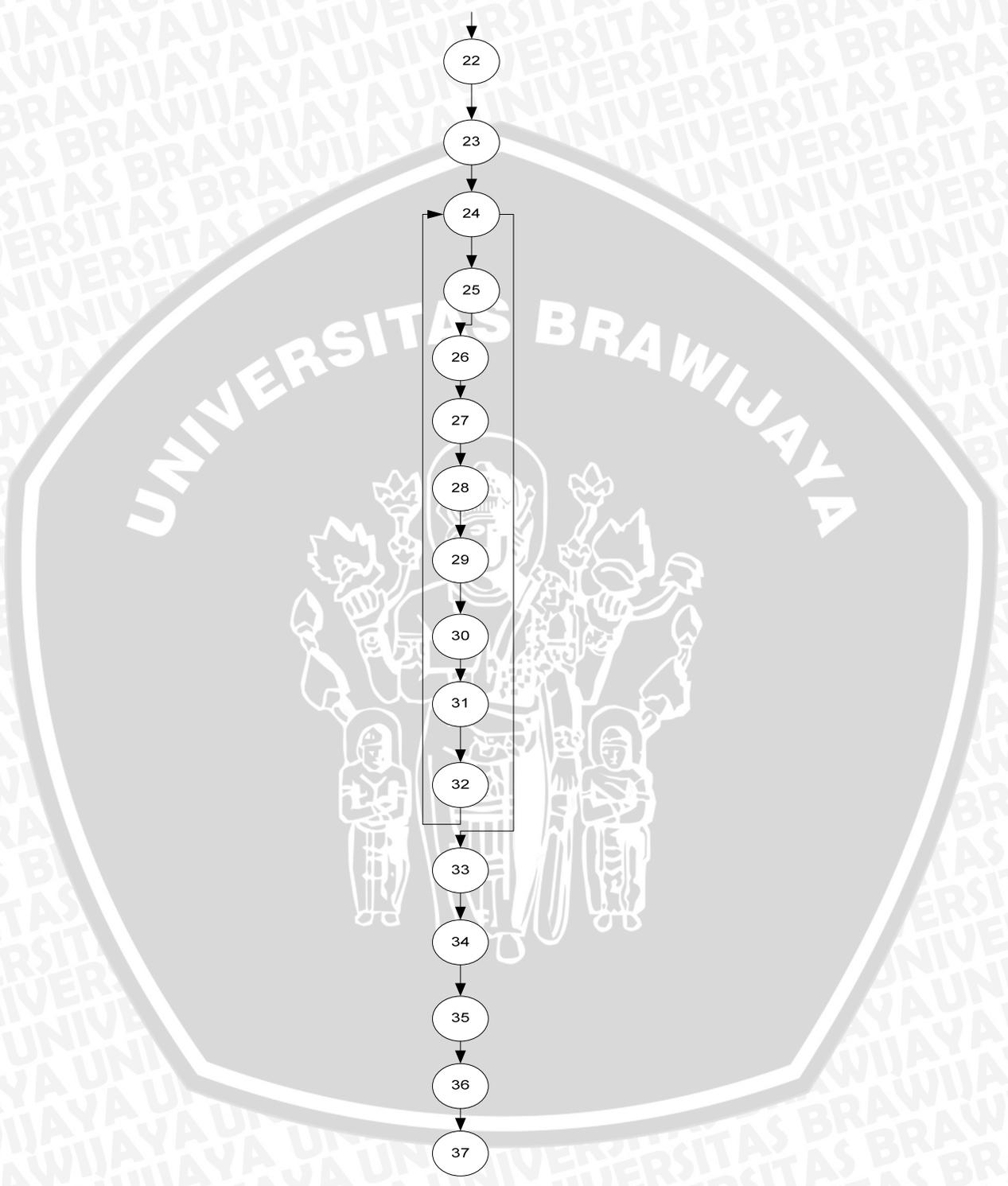
$$\text{Predikat} = 6$$

$$\begin{aligned} V &= n - P + 2 \\ &= 42 - 37 + 2 \\ &= 7 \end{aligned}$$

$$\begin{aligned} V &= \text{Predikat} + 1 \\ &= 6 + 1 \\ &= 7 \end{aligned}$$







Gambar 5. 4Basis Path Fungsi hitung1()
Sumber: [Pengujian dan Analisis]



5.1.1.4 Basis Path Fungsi hitung2()

Fungsi hitung2() merupakan fungsi menghitung rekomendasi jika kriteria fasilitas berisi value selain *any* dan memiliki nilai bobot. Gambar 5.5 menunjukkan pengujian *basis path* fungsi hitung2()

$$N(\text{Path}) = 51$$

$$P = 44$$

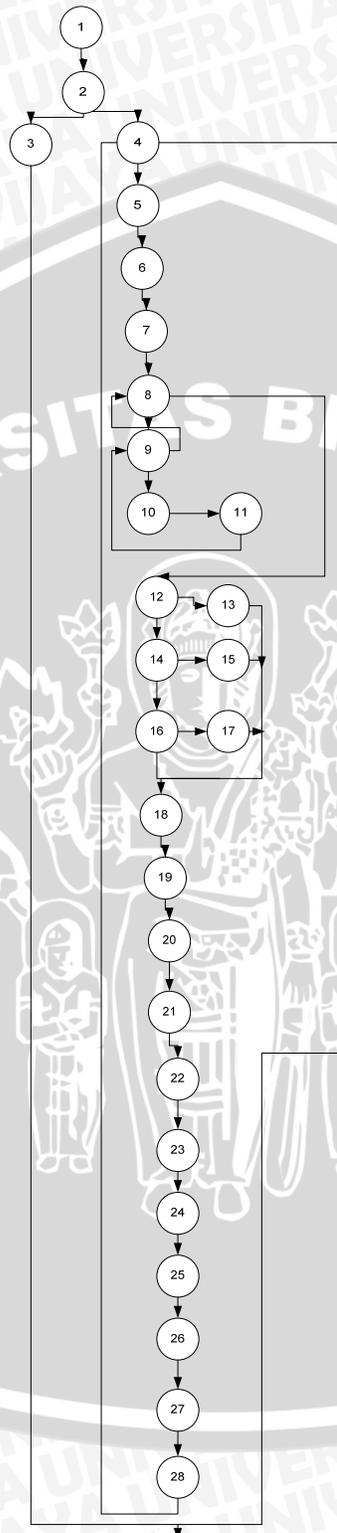
$$\text{Region} = 9$$

$$\text{Predikat} = 8$$

$$\begin{aligned} V &= n - P + 2 \\ &= 51 - 44 + 2 \\ &= 9 \end{aligned}$$

$$\begin{aligned} V &= \text{Predikat} + 1 \\ &= 8 + 1 \\ &= 9 \end{aligned}$$







Gambar 5. 5Basis Path Fungsi hitung2()
Sumber:[Pengujian dan Analisis]



5.1.2 Pengujian Validasi

Pengujian validasi digunakan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan. Daftar kebutuhan yang telah dirumuskan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *black box* karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan.

1. Uji Validitas Register

Tabel 5. 1 Uji Validitas Register

Nomor Kasus Uji	SRS_001_01
Nama Kasus Uji	<i>Register</i>
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu memenuhi kebutuhan fungsional dalam proses registrasi restoran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>register</i> pada menu <i>navbar</i>. 3. Sistem akan menampilkan <i>form</i> registrasi. 4. Aktor mengisi data pada <i>form</i> yang tersedia. 5. Aktor menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem dapat menyimpan data pengelola dan restoran yang diinputkan ke dalam sistem.
Hasil Yang Didapatkan	Sistem menampilkan <i>form</i> registrasi dengan baik dan dapat melakukan penyimpanan data sesuai dengan yang diharapkan.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Pada saat mengisi form register, pengelola restoran tidak mengisi beberapa <i>field required</i>.

	2. Tekan tombol <i>submit</i> .
Hasil Yang Diharapkan	Sistem dapat menampilkan pesan/notifikasi bahwa beberapa <i>field</i> yang harus diisi dan proses insert restoran gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan/notifikasi bahwa ada <i>field</i> yang harus diisi dan proses insert restoran gagal.
Status Validasi	Valid
Skenario Alternatif 2	
Prosedur Uji	<ol style="list-style-type: none"> 1. Pada saat mengisi form register, pengelola restoran mengisi alamat e-mail. 2. Tekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Ketika terdapat <i>email</i> yang sama pada database, sistem mampu menampilkan pesan/notifikasi bahwa <i>email</i> yang telah diinputkan oleh pengelola tersebut sudah ada pada <i>database</i> sistem.
Hasil Yang Didapatkan	sistem menampilkan pesan/notifikasi bahwa <i>email</i> yang telah diinputkan oleh pengelola tersebut sudah ada pada <i>database</i> sistem.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

2. Uji Validitas Kelola Akun Admin

Tabel 5. 2 Uji Validitas Kelola Akun Admin

Nomor Kasus Uji	SRS_002_01
Nama Kasus Uji	Lihat Akun
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem menyediakan kebutuhan <i>administrator</i> untuk melihat data <i>administrator</i> tersebut.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka halaman awal sistem.

	<ol style="list-style-type: none"> 2. Pilih menu <i>login</i> pada <i>navbar</i>. 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Administrator</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>administrator</i>.. 5. Sistem akan menampilkan halaman utama <i>administrator</i>. 6. Pilih menu <i>account</i> pada <i>navbar</i>. 7. Sistem menampilkan data <i>administrator</i> tersebut.
Hasil Yang Diharapkan	Sistem dapat menampilkan data <i>administrator</i> tersebut.
Hasil Yang Didapatkan	Sistem menampilkan data <i>administrator</i>
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5.3 Uji Validitas Edit Akun

Nomor Kasus Uji	SRS_002_01
Nama Kasus Uji	Edit Akun
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem menyediakan kebutuhan <i>administrator</i> untuk memperbarui data <i>administrator</i> tersebut.
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan data <i>administrator</i> tersebut. 2. <i>Administrator</i> menekan tombol <i>edit</i>. 3. Sistem menampilkan <i>form edit</i> akun. 4. <i>Adminisitrator</i> melakukan perubahan data pada <i>field</i> yang diinginkan. 5. <i>Administrator</i> menekan tombol <i>update</i>.
Hasil Yang Diharapkan	Sistem dapat memperbarui data <i>administrator</i> tersebut.
Hasil Yang Didapatkan	Sistem menyimpan data baru yang diinputkan

	oleh <i>administrator</i> tersebut.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Ketika <i>administrator</i> memperbarui data, <i>field email</i> dan nama tidak diisi. 2. Tekan tombol <i>update</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan pesan notifikasi bahwa <i>field</i> tersebut harus diisi dan proses <i>update</i> gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan notifikasi bahwa <i>field</i> tersebut harus diisi dan proses <i>update</i> gagal.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

3. Uji Validitas Kelola Kategori Fasilitas

Tabel 5. 4 Uji Validitas Lihat Fasilitas

Nomor Kasus Uji	SRS_002_02
Nama Kasus Uji	Lihat Kategori Fasilitas
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem dapat menyediakan kebutuhan <i>administrator</i> untuk melihat kategori fasilitas yang ada pada sistem.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i>. 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Administrator</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>administrator</i>. 5. Sistem akan menampilkan halaman utama <i>administrator</i>. 6. Pilih menu <i>others</i> pada <i>sidebar</i> dan pilih <i>facilities</i>.

	7. Sistem menampilkan data kategori fasilitas yang ada pada sistem.
Hasil Yang Diharapkan	Sistem dapat menampilkan semua kategori fasilitas yang ada pada sistem.
Hasil Yang Didapatkan	Sistem menampilkan semua kategori fasilitas yang ada pada sistem.
Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 5 Tambah Kategori Fasilitas

Nomor Kasus Uji	SRS_002_02
Nama Kasus Uji	Tambah Kategori Fasilitas
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem dapat menyediakan kebutuhan <i>administrator</i> untuk menambah kategori fasilitas yang ada pada sistem
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan data kategori fasilitas yang ada pada sistem. . 2. <i>Administrator</i> melakukan proses tambah kategori dengan menekan tombol <i>add facilities</i>. 3. Sistem menampilkan <i>form</i> tambah kategori. 4. <i>Administrator</i> mengisikan data pada <i>form</i> tambah kategori. 5. <i>Administrator</i> menekan tombol <i>submit</i>
Hasil Yang Diharapkan	Sistem dapat melakukan tambah kategori fasilitas ke dalam sistem.
Hasil yang didapatkan	Sistem menyimpan kategori fasilitas yang baru ditambahkan oleh <i>administrator</i> .
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>form</i> tambah kategori.

	<ol style="list-style-type: none"> 2. <i>Administrator</i> memasukkan data fasilitas yang sama dengan data fasilitas lainnya pada sistem. 3. <i>Administrator</i> menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan bahwa data yang dimasukkan sudah ada dan proses <i>insert</i> fasilitas gagal.
Hasil Yang Didapatkan	Sistem menampilkan notifikasi/pesan bahwa data yang dimasukkan sudah ada dan proses <i>insert</i> fasilitas gagal.
Status Validasi	Valid
Skenario Alternatif 2	
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>form</i> tambah kategori. 2. <i>Administrator</i> tidak mengisi semua <i>field</i> yang ada pada <i>form</i> tambah kategori fasilitas. 3. <i>Administrator</i> menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>insert</i> fasilitas gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>insert</i> fasilitas gagal.
Status Validasi	Valid
Skenario Alternatif 3	
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>form</i> tambah kategori. 2. <i>Administrator</i> memasukkan Id fasilitas tidak sesuai dengan format yang ada. 3. <i>Administrator</i> menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan kepada <i>administrator</i> bahwa ID yang dimasukkan tidak sesuai dan proses <i>insert</i>

	fasilitas baru gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan notifikasi/pesan bahwa data fasilitas yang baru berhasil ditambahkan dan proses <i>insert</i> fasilitas berhasil.
Status Validasi	Tidak Valid

Sumber : Pengujian dan Analisis

4. Uji Validitas Kelola Kategori Makanan

Tabel 5. 6 Uji Validitas Lihat Kategori Makanan

Nomor Kasus Uji	SRS_002_03
Nama Kasus Uji	Lihat Kategori Makanan
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu untuk menampilkan kategori makanan yang ada pada sistem.
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i> . 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Administrator</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>administrator</i>. 5. Sistem akan menampilkan halaman utama <i>administrator</i>. 6. Pilih menu <i>others</i> pada <i>sidebar</i> dan pilih <i>cuisine type</i>. 7. Sistem menampilkan data kategori fasilitas yang ada pada sistem. .
Hasil Yang Diharapkan	Sistem dapat menampilkan kategori jenis makanan yang ada pada sistem
Hasil Yang Didapatkan	Sistem menampilkan kategori jenis makanan yang ada pada sistem
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 7 Uji Validitas Tambah Kategori Makanan

Nomor Kasus Uji	SRS_002_03
Nama Kasus Uji	Tambah Kategori Makanan
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu untuk menambahkan kategori makanan yang ada pada sistem.
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan data kategori jenis makanan yang ada pada sistem. 2. <i>Administrator</i> menekan tombol <i>add cuisine</i>. 3. Sistem menampilkan <i>form</i> tambah kategori. 4. <i>Administrator</i> mengisikan data pada <i>form</i>. 5. <i>Administrator</i> menekan tombol <i>submit</i>
Hasil Yang Diharapkan	Sistem dapat melakukan tambah kategori fasilitas ke dalam sistem.
Hasil Yang Didapatkan	Sistem menyimpan kategori fasilitas yang baru ditambahkan oleh <i>administrator</i> .
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>form</i> tambah kategori. 2. <i>Administrator</i> memasukkan data jenis makanan yang sudah ada pada sistem. 3. <i>Administrator</i> menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan bahwa data yang dimasukkan sudah ada dan proses <i>insert</i> gagal.
Hasil Yang Didapatkan	Sistem menampilkan notifikasi/pesan bahwa data yang dimasukkan sudah ada dan proses <i>insert</i> gagal.
Status Validasi	Valid

Skenario Alternatif 2	
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>form</i> tambah kategori. 2. <i>Administrator</i> tidak mengisi semua <i>field</i> yang ada pada <i>form</i> tambah kategori jenis makanan. 3. <i>Administrator</i> menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>insert</i> gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>insert</i> gagal.
Status Validasi	Valid
Skenario Alternatif 3	
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan <i>form</i> tambah kategori. 2. <i>Administrator</i> memasukkan Id jenis makanan tidak sesuai dengan format yang ada. 3. <i>Administrator</i> menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan kepada <i>administrator</i> bahwa Id yang dimasukkan tidak sesuai dan proses <i>insert</i> gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan notifikasi/pesan bahwa jenis makanan yang baru berhasil ditambahkan dan proses <i>insert</i> berhasil.
Status Validasi	Tidak Valid

Sumber : Pengujian dan Analisis

5. Uji Validitas Reset Password

Tabel 5. 8 Uji validitas Reset Password

Nomor Kasus Uji	SRS_003_01
Nama Kasus Uji	Reset Password
Objek Uji	Kebutuhan Fungsional

Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu memenuhi kebutuhan <i>administrator</i> dan pengelola restoran untuk melakukan <i>reset password</i> .
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i>. 3. Sistem akan menampilkan <i>form login</i>. 4. Aktor memilih <i>forgot your password</i> 5. Sistem akan menampilkan <i>form reset password</i>. 6. Aktor mengisi data pada <i>form reset password</i>. 7. Aktor menekan tombol <i>reset password</i>.
Hasil Yang Diharapkan	Sistem dapat memberikan <i>password</i> baru ke alamat email aktor.
Hasil Yang Didapatkan	<i>Password</i> baru terkirim ke <i>email</i> aktor tersebut.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i>. 3. Sistem akan menampilkan <i>form login</i>. 4. Aktor memilih <i>forgot your password</i> 5. Sistem akan menampilkan <i>form reset password</i>. 6. Aktor mengisi <i>email</i> yang tidak terdaftar pada sistem. 7. Aktor menekan tombol <i>reset password</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan pesan bahwa <i>email</i> tidak ada pada sistem dan proses <i>reset password</i> gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan bahwa <i>email</i> tidak ada pada sistem dan proses <i>reset password</i> gagal.

Status Validasi	Valid
-----------------	-------

Sumber : Pengujian dan Analisis

6. Uji Validitas Kelola Akun dan Data Restoran

Tabel 5. 9 Uji Validitas Lihat Data

Nomor Kasus Uji	SRS_004_01
Nama Kasus Uji	Lihat data
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menampilkan data akun maupun data restoran.
Prosedur Uji	<p>Administrator</p> <ul style="list-style-type: none"> • Lihat Akun Pengelola <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i>. 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Administrator</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>administrator</i>. 5. Sistem akan menampilkan halaman utama <i>administrator</i>. 6. Pilih menu <i>account</i> pada <i>sidebar</i> dan tekan <i>list account</i> 7. Sistem menampilkan data akun pengelola restoran. • Lihat Data Restoran <ol style="list-style-type: none"> 1. Sistem menampilkan data akun pengelola restoran. 2. <i>Administrator</i> menekan tombol <i>view</i>. 3. Sistem menampilkan data restoran milik pengelola tersebut. <p>Pengelola Restoran</p> <ul style="list-style-type: none"> • Lihat Data Restoran <ol style="list-style-type: none"> 1. Buka halaman awal sistem.

	<ol style="list-style-type: none"> 2. Pilih menu <i>login</i> pada <i>navbar</i> . 3. Sistem akan menampilkan <i>form login</i>. 4. Pengelola mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai pengelola restoran. 5. Sistem akan menampilkan halaman utama pengelola restoran. 6. Pilih menu <i>profile</i> pada <i>sidebar</i> 7. Sistem menampilkan data restoran milik pengelola tersebut. <ul style="list-style-type: none"> • Lihat Akun Pengelola <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu login pada navbar . 3. Sistem akan menampilkan form login. 4. Pengelola mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai pengelola restoran. 5. Sistem akan menampilkan halaman utama pengelola restoran. 6. Pilih menu <i>user profile</i> pada navbar 7. Sistem menampilkan data akun pengelola tersebut. <p>Konsumen</p> <p>Skenario 1:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama sistem. 2. Konsumen memilih menu <i>search</i> pada navbar. 3. Konsumen melakukan pencarian restoran atau langsung memilih pada <i>list</i> yang ada pada halaman <i>search</i>. 4. Konsumen menekan tombol <i>view</i>. 5. Sistem menampilkan data restoran yang dipilih oleh konsumen.
--	---

	<p>Skenario II :</p> <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama sistem. 2. Konsumen memilih menu <i>recommendation</i> pada navbar. 3. Sistem menampilkan <i>form</i> untuk membantu konsumen mencari rekomendasi dengan beberapa kriteria. . 4. Konsumen memilih kriteria-kriteria yang diinginkan kemudian menekan tombol <i>search</i> 5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen 6. Konsumen menekan tombol <i>view</i>. 7. Sistem menampilkan data restoran yang dipilih oleh konsumen.
Hasil Yang Diharapkan	Sistem dapat menampilkan akun pengelola dan data restoran.
Hasil Yang Didapatkan	Sistem menampilkan akun pengelola restoran dan data restoran.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 10 Uji Validitas Edit Data

Nomor Kasus Uji	SRS_004_01
Nama Kasus Uji	Edit Data
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu memperbarui data akun maupun data restoran.

Prosedur Uji	Administrator
	<ul style="list-style-type: none"> • Edit Akun Pengelola 1. Sistem menampilkan data akun pengelola restoran. 2. <i>Administrator</i> menekan tombol <i>edit</i>. 3. Sistem menampilkan <i>form edit</i> untuk memperbarui data pengelola restoran. 4. <i>Administrator</i> melakukan perubahan data dan menekan tombol <i>update</i>. • Edit Data Restoran 1. Sistem menampilkan data akun pengelola restoran. 2. <i>Administrator</i> menekan tombol <i>view</i>. 3. Sistem menampilkan data restoran milik pengelola restoran yang dipilih <i>administrator</i>. 4. <i>Administrator</i> menekan tombol <i>edit</i>. 5. Sistem menampilkan <i>form edit</i>. 6. <i>Administrator</i> melakukan perubahan pada <i>field</i> yang diinginkan. 7. <i>Administrator</i> menekan tombol <i>update</i>.
	<p>Pengelola Restoran</p> <ul style="list-style-type: none"> • Edit Data Restoran 1. Sistem menampilkan halaman utama pengelola restoran 2. Pengelola restoran memilih tombol <i>edit</i>. 3. Sistem menampilkan <i>form edit</i>. 4. <i>Administrator</i> memperbarui data yang diinginkan. 5. <i>Administrator</i> menekan tombol <i>update</i>. • Edit Akun Pengelola Restoran 1. Pengelola restoran masuk ke halaman <i>user profile</i>. 2. Pengelola restoran memilih tombol <i>edit</i>.

	<ol style="list-style-type: none"> 3. Sistem menampilkan <i>form edit</i>. 4. Pengelola restoran memperbarui data yang diinginkan. 5. Pengelola restoran menekan tombol <i>update</i>.
Hasil Yang Diharapkan	Sistem dapat memperbarui data akun pengelola dan restoran.
Hasil Yang Didapatkan	Sistem menyimpan data pengelola yang baru.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<p>Administrator</p> <ul style="list-style-type: none"> • Edit Akun Pengelola <ol style="list-style-type: none"> 1. Sistem menampilkan data akun pengelola restoran. 2. <i>Administrator</i> menekan tombol <i>edit</i>. 3. Sistem menampilkan <i>form edit</i> untuk memperbarui data pengelola restoran. 4. <i>Administrator</i> melakukan perubahan data dengan mengosongkan beberapa <i>required field</i> dan menekan tombol <i>update</i>.



	<ul style="list-style-type: none"> • Edit Data Restoran <ol style="list-style-type: none"> 1. Sistem menampilkan data akun pengelola restoran. 2. Administrator menekan tombol <i>view</i>. 3. Sistem menampilkan data restoran milik pengelola restoran yang dipilih <i>administrator</i>. 4. <i>Administrator</i> menekan tombol <i>edit</i>. 5. Sistem menampilkan <i>form edit</i>. 6. <i>Administrator</i> melakukan perubahan data dengan mengosongkan beberapa <i>required field</i>. 7. <i>Administrator</i> menekan tombol <i>update</i>. <p>Pengelola Restoran</p> <ul style="list-style-type: none"> • Edit Data Restoran <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama pengelola restoran 2. Pengelola restoran memilih tombol <i>edit</i>. 3. Sistem menampilkan <i>form edit</i>. 4. <i>Administrator</i> melakukan perubahan data dengan mengosongkan beberapa <i>required field</i>. 5. <i>Administrator</i> menekan tombol <i>update</i>. <ul style="list-style-type: none"> • Edit Akun Pengelola Restoran <ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman <i>user profile</i>. 2. Pengelola restoran memilih tombol <i>edit</i>. 3. Sistem menampilkan <i>form edit</i>. 4. Pengelola restoran melakukan perubahan data dengan mengosongkan beberapa <i>required field</i>. 5. Pengelola restoran menekan tombol <i>update</i>.
--	---

Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>update</i> gagal.
Hasil Yang Didapatkan	Sistem menampilkan notifikasi/pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>update</i> gagal.
Status Validasi	Valid
Skenario Alternatif 2	
Prosedur Uji	<ol style="list-style-type: none"> 1. Ketika mengupdate data pengelola restoran, <i>field</i> email diisi dengan data <i>email</i> yang sudah ada pada <i>database</i> sistem. 2. Tekan tombol <i>update</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan bahwa alamat <i>email</i> sudah ada di <i>database</i> sistem dan proses update gagal.
Hasil Yang Didapatkan	Sistem menampilkan notifikasi/pesan bahwa alamat <i>email</i> sudah ada di <i>database</i> sistem dan proses update gagal.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 11 Uji Validitas Tambah Data

Nomor Kasus Uji	SRS_004_01
Nama Kasus Uji	Tambah Data
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menambah data akun maupun data restoran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem akan menampilkan halaman utama <i>administrator</i>. 2. Pilih menu <i>account</i> pada <i>sidebar</i> dan tekan <i>add new member</i>. 3. Sistem menampilkan <i>form</i> tambah member.

	4. <i>Administrator</i> mengisikan data dan menekan tombol <i>submit</i> .
Hasil Yang Diharapkan	Sistem dapat menyimpan data akun pengelola dan restoran yang baru.
Hasil Yang Didapatkan	Sistem menyimpan data pengelola yang baru.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem akan menampilkan halaman utama <i>administrator</i>. 2. Pilih menu <i>account</i> pada <i>sidebar</i> dan tekan <i>add new member</i>. 3. Sistem menampilkan <i>form</i> tambah member. 4. <i>Administrator</i> tidak mengisikan data pada <i>field required</i> dan menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>insert</i> gagal.
Hasil Yang Didapatkan	Sistem menampilkan notifikasi/pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>insert</i> gagal.
Status Validasi	Valid
Skenario Alternatif 2	
Prosedur Uji	<ol style="list-style-type: none"> 1. <i>Administrator</i> mengisikan data dan pada <i>field email</i>, <i>administrator</i> mengisikan data yang sudah ada pada <i>database</i> sistem. 2. <i>Administrator</i> menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan pesan.notifikasi bahwa alamat <i>email</i> yang dimasukkan sudah ada pada sistem.
Hasil Yang Didapatkan	Sistem menampilkan pesan.notifikasi bahwa alamat <i>email</i> yang dimasukkan sudah ada pada

	sistem.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 12 Uji Validitas Hapus Data

Nomor Kasus Uji	SRS_004_01
Nama Kasus Uji	Hapus Data
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menghapus data akun maupun data restoran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem akan menampilkan halaman utama <i>administrator</i>. 2. Pilih menu <i>account</i> pada <i>sidebar</i> dan tekan <i>list account</i>. 3. Sistem menampilkan data akun pengelola restoran. 4. <i>Administrator</i> menekan tombol <i>delete</i>.
Hasil Yang Diharapkan	Sistem dapat menghapus data akun pengelola dan restoran yang dipilih oleh <i>administrator</i> .
Hasil Yang Didapatkan	Sistem menghapus data pengelola yang dipilih oleh <i>administrator</i> .
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 13 Uji Validitas Aktivasi

Nomor Kasus Uji	SRS_004_01
Nama Kasus Uji	Aktivasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu mengkativasi akun pengelola restoran yang baru mendaftar pada sistem.
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem akan menampilkan halaman utama <i>administrator</i>.

	<ol style="list-style-type: none"> 2. Pilih menu <i>account</i> pada <i>sidebar</i> dan tekan <i>list account</i>. 3. Sistem menampilkan data akun pengelola restoran. 4. <i>Administrator</i> menekan tombol <i>activate</i>.
Hasil Yang Diharapkan	Sistem mampu mengkativasi akun pengelola restoran yang baru mendaftar pada sistem.
Hasil Yang Didapatkan	Status akun pengelola tersebut menjadi aktif dan pengelola tersebut akan mendapatkan notifikasi melalui email bahwa akun telah aktif dan siap digunakan.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 14 Uji Validitas Deaktivasi

Nomor Kasus Uji	SRS_004_01
Nama Kasus Uji	Deaktivasi
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu meng-nonaktifkan akun pengelola restoran.
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem akan menampilkan halaman utama <i>administrator</i>. 2. Pilih menu <i>account</i> pada <i>sidebar</i> dan tekan <i>list account</i>. 3. Sistem menampilkan data akun pengelola restoran. 4. <i>Administrator</i> menekan tombol <i>deactivate</i>.
Hasil Yang Diharapkan	Sistem mampu meng-nonaktifkan akun pengelola restoran. .
Hasil Yang Didapatkan	Status akun pengelola tersebut menjadi tidak aktif dan pengelola tersebut akan mendapatkan

	notifikasi melalui email dan akun tidak lagi bisa digunakan.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

7. Uji Validitas Kelola Menu Restoran

Tabel 5. 15 Uji Validitas Lihat Menu

Nomor Kasus Uji	SRS_004_02
Nama Kasus Uji	Lihat Menu
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menampilkan menu milik restoran.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i> . 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Administrator</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>administrator</i>. 5. Sistem akan menampilkan halaman utama <i>administrator</i>. 6. Pilih <i>menu</i> pada <i>sidebar</i> 7. Sistem menampilkan list akun pengelola restoran 8. <i>Administrator</i> menekan tombol <i>view</i>. 9. Sistem menampilkan menu restoran yang dipilih oleh <i>administrator</i> <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i> . 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Pengelola</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>pengelola restoran</i>.

5. Sistem akan menampilkan halaman utama pengelola restoran.
6. Pilih *menu* pada *sidebar*
7. Sistem menampilkan menu restoran milik pengelola tersebut.

Konsumen

Skenario 1:

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *search* pada navbar.
3. Konsumen melakukan pencarian restoran atau langsung memilih pada *list* yang ada pada halaman *search*.
4. Konsumen menekan tombol *view*.
5. Sistem menampilkan data restoran yang dipilih oleh konsumen.
6. Konsumen menekan *tab menu*.
7. Sistem menampilkan menu restoran yang dipilih konsumen.

Skenario II :

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *recommendation* pada navbar.
3. Sistem menampilkan *form* untuk membantu konsumen mencari rekomendasi dengan beberapa kriteria. .
4. Konsumen memilih kriteria-kriteria yang diinginkan kemudian menekan tombol *search*
5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen

	<ol style="list-style-type: none"> 6. Konsumen menekan tombol <i>view</i>. 7. Sistem menampilkan data restoran yang dipilih oleh konsumen. 8. Konsumen menekan <i>tab menu</i>. 9. Sistem menampilkan menu restoran yang dipilih konsumen.
Hasil Yang Diharapkan	Sistem mampu menampilkan menu restoran.
Hasil Yang Didapatkan	Sistem menampilkan menu restoran.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 16 Uji Validitas Tambah Menu

Nomor Kasus Uji	SRS_004_02
Nama Kasus Uji	Tambah Menu
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menambahkan menu baru pada sistem.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman menu. 2. <i>Administrator</i> menekan tombol <i>add</i> pada akun yang dipilih. 3. Sistem menampilkan <i>form</i> tambah menu untuk menambahkan menu. 4. <i>Administrator</i> menginputkan menu dan menekan tombol <i>submit</i>. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman menu. 2. Sistem menampilkan menu restoran milik pengelola tersebut. 3. Pengelola restoran memilih tombol <i>add</i>. 4. Sistem menampilkan <i>form</i> tambah menu. 5. Pengelola restoran menambahkan

	<p>menu.</p> <p>6. Pengelola restoran menekan tombol <i>submit</i>.</p>
Hasil Yang Diharapkan	Sistem dapat menyimpan menu restoran yang baru.
Hasil Yang Didapatkan	Menu restoran yang baru tersimpan ke dalam sistem.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Saat menambahkan menu, aktor mengupload gambar menu tidak sesuai dengan format (JPG, JPEG, PNG). 2. Tekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan bahwa gambar yang diupload tidak sesuai dengan format dan proses <i>insert</i> menu gagal.
Hasil Yang Didapatkan	Sistem menampilkan notifikasi/pesan bahwa gambar yang diupload tidak sesuai dengan format dan proses <i>insert</i> menu gagal.
Status Validasi	Valid
Skenario Alternatif 2	
Prosedur Uji	<ol style="list-style-type: none"> 1. Saat menambahkan menu, aktor tidak mengisi semua <i>field</i> pada <i>form</i> tambah menu. 2. Tekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan notifikasi/pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>insert</i> menu gagal.
Hasil Yang Didapatkan	Sistem menampilkan notifikasi/pesan bahwa <i>field</i> tersebut harus diisi dan proses <i>insert</i> menu gagal.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 17 Uji Validitas Hapus Menu

Nomor Kasus Uji	SRS_004_02
Nama Kasus Uji	Hapus Menu
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menghapus menu pada sistem.
Prosedur Uji	<p>Administrator</p> <ul style="list-style-type: none"> • Skenario I <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman menu. 2. <i>Administrator</i> menekan tombol <i>delete all</i> pada akun yang dipilih. • Skenario II <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman menu. 2. <i>Administrator</i> menekan tombol <i>view</i>. 3. Sistem menampilkan menu restoran milik akun pengelola yang dipilih <i>administrator</i>. 4. <i>Administrator</i> menekan tombol <i>delete</i> pada menu yang akan dihapus. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman menu. 2. Sistem menampilkan menu restoran milik pengelola tersebut. 3. Pengelola restoran memilih tombol <i>delete</i> pada menu yang akan dihapus.
Hasil Yang Diharapkan	Sistem dapat menghapus data menu restoran.
Hasil Yang Didapatkan	Menu restoran terhapus dari sistem.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

8. Uji Validitas Kelola Promo Restoran

Tabel 5. 18 Uji Validitas lihat Promo

Nomor Kasus Uji	SRS_004_03
Nama Kasus Uji	Lihat Promo
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menampilkan promomilik restoran.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i> . 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Administrator</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>administrator</i>. 5. Sistem akan menampilkan halaman utama <i>administrator</i>. 6. Pilih <i>promo</i> pada <i>sidebar</i> 7. Sistem menampilkan list akun pengelola restoran 8. <i>Administrator</i> menekan tombol <i>view</i>. 9. Sistem menampilkan promo dari restoran yang dipilih oleh <i>administrator</i>. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i> . 3. Sistem akan menampilkan <i>form login</i>. 4. Pengelola mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai pengelola restoran. 5. Sistem akan menampilkan halaman utama pengelola restoran. 6. Pilih <i>promo</i> pada <i>sidebar</i> 7. Sistem menampilkan promo restoran

milik pengelola tersebut.

Konsumen

Skenario 1:

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *search* pada navbar.
3. Konsumen melakukan pencarian restoran atau langsung memilih pada *list* yang ada pada halaman *search*.
4. Konsumen menekan tombol *view*.
5. Sistem menampilkan data restoran yang dipilih oleh konsumen.
6. Konsumen menekan *tab promo*.
7. Sistem menampilkan promo restoran yang dipilih konsumen.

Skenario II :

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *recommendation* pada navbar.
3. Sistem menampilkan *form* untuk membantu konsumen mencari rekomendasi dengan beberapa kriteria. .
4. Konsumen memilih kriteria-kriteria yang diinginkan kemudian menekan tombol *search*
5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen
6. Konsumen menekan tombol *view*.
7. Sistem menampilkan data restoran yang dipilih oleh konsumen.
8. Konsumen menekan *tab promo*.

	<p>9. Sistem menampilkan promo restoran yang dipilih konsumen.</p> <ul style="list-style-type: none"> • Skenario III <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama. 2. Konsumen memilih <i>coupon</i> pada navbar. 3. Sistem menampilkan semua promo yang ada pada sistem.
Hasil Yang Diharapkan	Sistem mampu menampilkan promo restoran.
Hasil Yang Didapatkan	Sistem menampilkan promo restoran.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 19 Uji Validitas Lihat Detail Promo

Nomor Kasus Uji	SRS_004_03
Nama Kasus Uji	Lihat Detail Promo
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menampilkan detail promo milik restoran.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. Pilih <i>promo</i> pada <i>sidebar</i> 2. Sistem menampilkan list akun pengelola restoran 3. <i>Administrator</i> menekan tombol <i>view</i>. 4. Sistem menampilkan promo dari restoran yang dipilih oleh <i>administrator</i>. 5. <i>Administrator</i> menekan tombol <i>read more</i>. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Pilih <i>promo</i> pada <i>sidebar</i> 2. Sistem menampilkan promo restoran milik pengelola tersebut.

	<p>3. <i>Administrator</i> menekan tombol <i>read more</i>.</p> <p>Konsumen</p> <p>Skenario 1:</p> <ol style="list-style-type: none">1. Sistem menampilkan halaman utama sistem.2. Konsumen memilih menu <i>search</i> pada navbar.3. Konsumen melakukan pencarian restoran atau langsung memilih pada <i>list</i> yang ada pada halaman <i>search</i>.4. Konsumen menekan tombol <i>view</i>.5. Sistem menampilkan data restoran yang dipilih oleh konsumen.6. Konsumen menekan <i>tab promo</i>.7. Sistem menampilkan promo restoran yang dipilih konsumen.8. Konsumen menekan tombol <i>read more</i>. <p>Skenario II :</p> <ol style="list-style-type: none">1. Sistem menampilkan halaman utama sistem.2. Konsumen memilih menu <i>recommendation</i> pada navbar.3. Sistem menampilkan <i>form</i> untuk membantu konsumen mencari rekomendasi dengan beberapa kriteria. .4. Konsumen memilih kriteria-kriteria yang diinginkan kemudian menekan tombol <i>search</i>5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen6. Konsumen menekan tombol <i>view</i>.7. Sistem menampilkan data restoran yang
--	--

	<p>dipilih oleh konsumen.</p> <ol style="list-style-type: none"> 8. Konsumen menekan <i>tab</i> promo. 9. Sistem menampilkan promo restoran yang dipilih konsumen. 10. Konsumen menekan tombol <i>read more</i>. <ul style="list-style-type: none"> • Skenario III <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama. 2. Konsumen memilih <i>coupon</i> pada navbar. 3. Sistem menampilkan semua promo yang ada pada sistem. 4. Konsumen menekan tombol <i>read more</i>.
Hasil Yang Diharapkan	Sistem mampu menampilkan detail spromo restoran.
Hasil Yang Didapatkan	Sistem menampilkan detail promo restoran.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 20 Uji Validitas Tambah Promo

Nomor Kasus Uji	SRS_004_03
Nama Kasus Uji	Tambah Promo
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menambahkan promo baru pada sistem.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman promo. 2. <i>Administrator</i> menekan tombol <i>add</i> pada akun yang dipilih. 3. Sistem menampilkan <i>form</i> tambah promo untuk menambahkan promo. 4. <i>Administrator</i> menginputkan promo dan menekan tombol <i>submit</i>. <p>Pengelola Restoran</p>

	<ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman promo. 2. Sistem menampilkan promo restoran milik pengelola tersebut. 3. Pengelola restoran memilih tombol <i>add promo</i>. 4. Sistem menampilkan <i>form</i> tambah promo. 5. Pengelola restoran menambahkan promos. 6. Pengelola restoran menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem dapat menyimpan promo restoran yang baru.
Hasil Yang Didapatkan	Promo restoran yang baru tersimpan ke dalam sistem.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Saat menambahkan promo, <i>field required</i> tidak diisi. 2. Aktor menekan tombol <i>submit</i>.
Hasil Yang Diharapkan	Sistem dapat menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses tambah promo gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses tambah promo gagal.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 21 Uji Validasi Edit Promo

Nomor Kasus Uji	SRS_004_03
Nama Kasus Uji	Edit Promo
Objek Uji	Kebutuhan Fungsional

Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu memperbarui data promo restoran.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman promo. 2. Sistem menampilkan data akun pengelola restoran. 3. <i>Administrator</i> menekan tombol <i>view</i> 4. Sistem menampilkan data promo milik pengelola yang dipilih oleh <i>administrator</i>. 5. <i>Administrator</i> menekan tombol <i>edit</i> 6. Sistem menampilkan <i>form edit</i> untuk memperbarui data promo yang dipilihnya. 7. <i>Administrator</i> melakukan perubahan data dan menekan tombol <i>update</i>. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman promo 2. Pengelola restoran memilih tombol <i>edit</i>. 3. Sistem menampilkan <i>form edit</i>. 4. Pengelola restoran memperbarui data yang diinginkan. 5. Pengelola restoran menekan tombol <i>update</i>.
Hasil Yang Diharapkan	Sistem dapat memperbarui data promo restoran.
Hasil Yang Didapatkan	Data promo restoran telah diperbarui.
Status Validasi	Valid
Skenario Alternatif 1	
Uji Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan form edit 2. Saat menambahkan promo, <i>field required</i> tidak diisi.

	3. Aktor menekan tombol <i>update</i> .
Hasil Yang Diharapkan	Sistem akan menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses edit promo gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses edit promo gagal.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 22 Uji Validasi Hapus Promo

Nomor Kasus Uji	SRS_004_03
Nama Kasus Uji	Hapus Promo
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menghapus promo pada sistem.
Prosedur Uji	<p>Administrator</p> <ul style="list-style-type: none"> • Skenario I <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman promo. 2. <i>Administrator</i> menekan tombol <i>delete all</i> pada akun yang dipilih. • Skenario II <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman promos. 2. <i>Administrator</i> menekan tombol <i>view</i>. 3. Sistem menampilkan promo restoran milik akun pengelola yang dipilih <i>administrator</i>. 4. <i>Administrator</i> menekan tombol <i>delete</i> pada promo yang akan dihapus. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman promo.

	<ol style="list-style-type: none"> 2. Sistem menampilkan promo restoran milik pengelola tersebut. 3. Pengelola restoran memilih tombol <i>delete</i> pada promo yang akan dihapus.
Hasil Yang Diharapkan	Sistem dapat menghapus data promo restoran.
Hasil Yang Didapatkan	Promo restoran terhapus dari sistem.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

9. Uji Validitas Kelola Komentar

Tabel 5. 23 Uji Validitas Lihat Komentar

Nomor Kasus Uji	SRS_004_04
Nama Kasus Uji	Lihat Komentar
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menampilkan komentar restoran.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i>. 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Administrator</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>administrator</i>. 5. Sistem akan menampilkan halaman utama <i>administrator</i>. 6. Pilih <i>comment</i> pada <i>sidebar</i> 7. Sistem menampilkan list akun pengelola restoran 8. <i>Administrator</i> menekan tombol <i>view</i>. 9. Sistem menampilkan komentar dari restoran yang dipilih oleh <i>administrator</i>.

Pengelola Restoran

1. Buka halaman awal sistem.
2. Pilih menu *login* pada *navbar* .
3. Sistem akan menampilkan *form login*.
4. Pengelola mengisikan *username*, *password*, dan memilih login sebagai pengelola restoran.
5. Sistem akan menampilkan halaman utama pengelola restoran.
6. Pilih komentarpada *sidebar*
7. Sistem menampilkan komentar milik pengelola tersebut.

Konsumen**Skenario 1:**

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *search* pada *navbar*.
3. Konsumen melakukan pencarian restoran atau langsung memilih pada *list* yang ada pada halaman *search*.
4. Konsumen menekan tombol *view*.
5. Sistem menampilkan data restoran dan komentar yang dipilih oleh konsumen.

Skenario II :

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *recommendation* pada *navbar*.
3. Sistem menampilkan *form* untuk membantu konsumen mencari rekomendasi dengan beberapa kriteria. .
4. Konsumen memilih kriteria-kriteria

	<p>yang diinginkan kemudian menekan tombol <i>search</i></p> <ol style="list-style-type: none"> 5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen 6. Konsumen menekan tombol <i>view</i>. 7. Sistem menampilkan data restoran dan data komentar yang dipilih oleh konsumen.
Hasil Yang Diharapkan	Sistem mampu menampilkan komentar milik restoran.
Hasil Yang Didapatkan	Sistem menampilkan komentar milik restoran.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 24 Uji Validitas Hapus Komentar

Nomor Kasus Uji	SRS_004_04
Nama Kasus Uji	Hapus Komentar
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menghapus komentar pada sistem.
Prosedur Uji	<p>Administrator</p> <ul style="list-style-type: none"> • Skenario I <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman komentar. 2. <i>Administrator</i> menekan tombol <i>delete all</i> pada akun yang dipilih. • Skenario II <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman komentar. 2. <i>Administrator</i> menekan tombol <i>view</i>. 3. Sistem menampilkan komentar milik akun pengelola yang dipilih <i>administrator</i>. 4. <i>Administrator</i> menekan tombol <i>delete</i>

	pada komentar yang akan dihapus.
Hasil Yang Diharapkan	Sistem dapat menghapus komentar restoran.
Hasil Yang Didapatkan	Komentar restoran terhapus dari sistem.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 25 Uji Validitas Tambah Komentar

Nomor Kasus Uji	SRS_004_04
Nama Kasus Uji	Tambah Komentar
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menambahkan komentar baru pada sistem.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman komentar. 2. <i>Administrator</i> menekan tombol <i>view</i> pada akun yang dipilih. 3. Sistem menampilkan komentar milik restoran yang dipilih oleh <i>administrator</i>. 4. <i>Administrator</i> menekan tombol <i>reply</i> atau <i>add comment</i>. 5. Sistem menampilkan <i>form</i> tambah komentar. 6. <i>Administrator</i> menginputkan komentar dan menekan tombol <i>submit</i> atau <i>send comment</i>. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman komentar. 2. Sistem menampilkan komentar restoran milik pengelola tersebut. 3. Pengelola restoran memilih tombol

reply.

4. Sistem menampilkan *form* tambah komentar.
5. Pengelola restoran menambahkan komentar.
6. Pengelola restoran menekan tombol *submit*.

Konsumen

Skenario 1:

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *search* pada navbar.
3. Konsumen melakukan pencarian restoran atau langsung memilih pada *list* yang ada pada halaman *search*.
4. Konsumen menekan tombol *view*.
5. Sistem menampilkan data restoran dan komentar yang dipilih oleh konsumen.
6. Konsumen menekan tombol *reply* atau *add comment*.
7. Sistem menampilkan *form* untuk menambahkan komentar.
8. Konsumen menambahkan komentar dan menekan tombol *submit* atau *send comment*.

Skenario II :

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *recommendation* pada navbar.
3. Sistem menampilkan *form* untuk membantu konsumen mencari

	<p>rekomendasi dengan beberapa kriteria. .</p> <ol style="list-style-type: none"> 4. Konsumen memilih kriteria-kriteria yang diinginkan kemudian menekan tombol <i>search</i> 5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen 6. Konsumen menekan tombol <i>view</i>. 7. Sistem menampilkan data restoran dan data komentar yang dipilih oleh konsumen. 8. Konsumen menekan tombol <i>reply</i> atau <i>add comment</i>. 9. Sistem menampilkan <i>form</i> untuk menambahkan komentar. 10. Konsumen menambahkan komentar dan menekan tombol <i>submit</i> atau <i>send comment</i>.
Hasil Yang Diharapkan	Sistem dapat menyimpan komentar restoran yang baru.
Hasil Yang Didapatkan	Komentar restoran yang baru tersimpan ke dalam sistem.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Saat proses menambahkan komentar, aktor tidak mengisi semua <i>field</i> yang ada pada form tambah komentar/balas komentar. 2. Tekan tombol <i>submit</i> atau <i>send comment</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses tambah komentar gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan/notifikasi bahwa

	<i>field</i> tersebut harus diisi dan proses tambah komentar gagal.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

10. Uji Validitas Kelola Berita

Tabel 5. 26 Uji Validitas Lihat Berita

Nomor Kasus Uji	SRS_004_05
Nama Kasus Uji	Lihat Berita
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menampilkan berita milik restoran.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i> . 3. Sistem akan menampilkan <i>form login</i>. 4. <i>Administrator</i> mengisikan <i>username</i>, <i>password</i>, dan memilih login sebagai <i>administrator</i>. 5. Sistem akan menampilkan halaman utama <i>administrator</i>. 6. Pilih <i>news</i> pada <i>sidebar</i> 7. Sistem menampilkan list akun pengelola restoran 8. <i>Administrator</i> menekan tombol <i>view</i>. 9. Sistem menampilkan berita dari restoran yang dipilih oleh <i>administrator</i>. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Buka halaman awal sistem. 2. Pilih menu <i>login</i> pada <i>navbar</i> . 3. Sistem akan menampilkan <i>form login</i>. 4. Pengelola mengisikan <i>username</i>,

password, dan memilih login sebagai pengelola restoran.

5. Sistem akan menampilkan halaman utama pengelola restoran.
6. Pilih *news* pada *sidebar*
7. Sistem menampilkan berita restoran milik pengelola tersebut.

Konsumen

Skenario 1:

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *search* pada navbar.
3. Konsumen melakukan pencarian restoran atau langsung memilih pada *list* yang ada pada halaman *search*.
4. Konsumen menekan tombol *view*.
5. Sistem menampilkan data restoran yang dipilih oleh konsumen.
6. Konsumen menekan *tab news*.
7. Sistem menampilkan berita restoran yang dipilih konsumen.

Skenario II :

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *recommendation* pada navbar.
3. Sistem menampilkan *form* untuk membantu konsumen mencari rekomendasi dengan beberapa kriteria. .
4. Konsumen memilih kriteria-kriteria yang diinginkan kemudian menekan tombol *search*

	<ol style="list-style-type: none"> 5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen 6. Konsumen menekan tombol <i>view</i>. 7. Sistem menampilkan data restoran yang dipilih oleh konsumen. 8. Konsumenmenekan <i>tab news</i>. 9. Sistem menampilkan berita restoran yang dipilih konsumen. <ul style="list-style-type: none"> • Skenario III <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama. 2. Konsumen memilih <i>news</i> pada navbar. 3. Sistem menampilkan semua berita yang ada pada sistem.
Hasil Yang Diharapkan	Sistem mampu menampilkan berita restoran.
Hasil Yang Didapatkan	Sistem menampilkan berita restoran.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 27 Uji Validitas Lihat Detail Berita

Nomor Kasus Uji	SRS_004_05
Nama Kasus Uji	Lihat Detail Berita
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menampilkan detail berita milik restoran.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. Pilih <i>news</i> pada <i>sidebar</i> 2. Sistem menampilkan list akun pengelola restoran 3. <i>Administrator</i> menekan tombol <i>view</i>. 4. Sistem menampilkan berita dari restoran yang dipilih oleh <i>administrator</i>. 5. <i>Administrator</i> menekan tombol <i>read</i>

more.

Pengelola Restoran

1. Pilih *news* pada *sidebar*
2. Sistem menampilkan berita restoran milik pengelola tersebut.
3. *Administrator* menekan tombol *read more.*

Konsumen

Skenario 1:

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *search* pada navbar.
3. Konsumen melakukan pencarian restoran atau langsung memilih pada *list* yang ada pada halaman *search*.
4. Konsumen menekan tombol *view*.
5. Sistem menampilkan data restoran yang dipilih oleh konsumen.
6. Konsumen menekan *tab news*.
7. Sistem menampilkan berita restoran yang dipilih konsumen.
8. Konsumen menekan tombol *read more.*

Skenario II :

1. Sistem menampilkan halaman utama sistem.
2. Konsumen memilih menu *recommendation* pada navbar.
3. Sistem menampilkan *form* untuk membantu konsumen mencari rekomendasi dengan beberapa kriteria. .
4. Konsumen memilih kriteria-kriteria

	<p>yang diinginkan kemudian menekan tombol <i>search</i></p> <ol style="list-style-type: none"> 5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen 6. Konsumen menekan tombol <i>view</i>. 7. Sistem menampilkan data restoran yang dipilih oleh konsumen. 8. Konsumenmenekan <i>tab news</i>. 9. Sistem menampilkan berita restoran yang dipilih konsumen. 10. Konsumen menekan tombol <i>read more</i>. <ul style="list-style-type: none"> • Skenario III <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama. 2. Konsumen memilih <i>news</i> pada navbar. 3. Sistem menampilkan semua berita yang ada pada sistem. 4. Konsumen menekan tombol <i>read more</i>.
Hasil Yang Diharapkan	Sistem mampu menampilkan detail sberita restoran.
Hasil Yang Didapatkan	Sistem menampilkan detail berita restoran.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 28 Uji Validitas Tambah Berita

Nomor Kasus Uji	SRS_004_05
Nama Kasus Uji	Tambah Berita
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menambahkan berita baru pada sistem.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman berita. 2. <i>Administrator</i> menekan tombol <i>add</i> pada akun yang dipilih. 3. Sistem menampilkan <i>form</i> tambah

	<p>berita untuk menambahkan berita.</p> <p>4. <i>Administrator</i> menginputkan berita dan menekan tombol <i>submit</i>.</p> <p>Pengelola Restoran</p> <p>1. Pengelola restoran masuk ke halaman berita.</p> <p>2. Sistem menampilkan berita restoran milik pengelola tersebut.</p> <p>3. Pengelola restoran memilih tombol <i>add news</i>.</p> <p>4. Sistem menampilkan <i>form</i> tambah berita.</p> <p>5. Pengelola restoran menambahkan berita.</p> <p>6. Pengelola restoran menekan tombol <i>submit</i>.</p>
Hasil Yang Diharapkan	Sistem dapat menyimpan berita restoran yang baru.
Hasil Yang Didapatkan	Berita restoran yang baru tersimpan ke dalam sistem.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<p>1. Sistem menampilkan form tambah berita.</p> <p>2. Saat menambahkan berita, <i>field required</i> tidak diisi.</p> <p>3. Aktor menekan tombol <i>submit</i>.</p>
Hasil Yang Diharapkan	Sistem akan menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses tambah berita gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses tambah

	berita gagal.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 29 Uji Validitas Edit Berita

Nomor Kasus Uji	SRS_004_05
Nama Kasus Uji	Edit Berita
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu memperbarui data berita restoran.
Prosedur Uji	<p>Administrator</p> <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman berita. 2. Sistem menampilkan data akun pengelola restoran. 3. <i>Administrator</i> menekan tombol <i>view</i> 4. Sistem menampilkan data berita milik pengelola yang dipilih oleh <i>administrator</i>. 5. <i>Administrator</i> menekan tombol <i>edit</i> 6. Sistem menampilkan <i>form edit</i> untuk memperbarui data berita yang dipilihnya. 7. <i>Administrator</i> melakukan perubahan data dan menekan tombol <i>update</i>. <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman berita 2. Pengelola restoran memilih tombol <i>edit</i>. 3. Sistem menampilkan <i>form edit</i>. 4. Pengelola restoran memperbarui data yang diinginkan. 5. Pengelola restoran menekan tombol <i>update</i>.
Hasil Yang Diharapkan	Sistem dapat memperbarui data berita restoran.

Hasil Yang Didapatkan	Data berita restoran telah diperbarui.
Status Validasi	Valid
Skenario Alternatif 1	
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan form edit berita. 2. Saat edit berita, <i>field required</i> tidak diisi. 3. Aktor menekan tombol <i>update</i>.
Hasil Yang Diharapkan	Sistem akan menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses edit berita gagal.
Hasil Yang Didapatkan	Sistem menampilkan pesan/notifikasi bahwa <i>field</i> tersebut harus diisi dan proses edit berita gagal.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 30 Uji Validitas Hapus Berita

Nomor Kasus Uji	SRS_004_05
Nama Kasus Uji	Hapus Berita
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menghapus berita pada sistem.
Prosedur Uji	<p>Administrator</p> <ul style="list-style-type: none"> • Skenario I <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman berita. 2. <i>Administrator</i> menekan tombol <i>delete all</i> pada akun yang dipilih. • Skenario II <ol style="list-style-type: none"> 1. <i>Administrator</i> masuk ke halaman berita. 2. <i>Administrator</i> menekan tombol <i>view</i>. 3. Sistem menampilkan berita restoran milik akun pengelola yang dipilih <i>administrator</i>. 4. <i>Administrator</i> menekan tombol <i>delete</i>

	<p>pada berita yang akan dihapus.</p> <p>Pengelola Restoran</p> <ol style="list-style-type: none"> 1. Pengelola restoran masuk ke halaman berita. 2. Sistem menampilkan berita restoran milik pengelola tersebut. 3. Pengelola restoran memilih tombol <i>delete</i> pada berita yang akan dihapus.
Hasil Yang Diharapkan	Sistem dapat menghapus data berita restoran.
Hasil Yang Didapatkan	Berita restoran terhapus dari sistem.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

11. Uji Validitas Cari Restoran

Tabel 5. 31 Uji Validitas Cari Restoran

Nomor Kasus Uji	SRS_005_01
Nama Kasus Uji	Cari Restoran
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu menampilkan restoran sesuai dengan apa yang dicari oleh konsumen.
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama. 2. Konsumen memilih <i>search</i> pada navbar. 3. Konsumen melakukan pencarian dengan menginputkan pilihan cari dan kata kunci.
Hasil Yang Diharapkan	Sistem dapat menampilkan data restoran yang dicari. Jika restoran yang dicari tidak ada, maka sistem akan menampilkan pesan/notifikasi bahwa data yang dicari tidak ada.
Hasil Yang Didapatkan	Sistem menampilkan data restoran yang dicari. Jika restoran yang dicari tidak ada, sistem menampilkan halaman kosong.

Status Validasi	Tidak Valid
------------------------	-------------

Sumber : Pengujian dan Analisis

Tabel 5. 32 Uji Validitas Cari Rekomendasi Restoran

Nomor Kasus Uji	SRS_005_02
Nama Kasus Uji	Cari Rekomendasi Restoran
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu memberikan rekomendasi restoran sesuai dengan kriteria yang diinputkan oleh konsumen.
Prosedur Uji	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama. 2. Konsumen memilih <i>recommendation</i> pada navbar. 3. Sistem menampilkan <i>form</i> untuk pencarian rekomendasi 4. Konsumen melakukan pencarian dengan menginputkan kriteria an bobot yang diinginkan. 5. Konsumen menekan tombol <i>search</i>
Hasil Yang Diharapkan	Sistem dapat memberikan rekomendasi restoran dengan disertai visualisasi peta. Ketika restoran dengan jenis makanan tersebut tidak ada maka sistem akan menampilkan pesan bahwa data yang dicari tidak ada.
Hasil Yang Didapatkan	Sistem menampilkan rekomendasi restoran dengan disertai visualisasi peta. Ketika restoran dengan jenis makanan tersebut tidak ada, sistem akan menampilkan pesan bahwa data yang dicari tidak ada.
Status Validasi	Valid

Sumber : Pengujian dan Analisis

Tabel 5. 33 Uji Validitas Lihat Petunjuk Arah

Nomor Kasus Uji	SRS_005_03
Nama Kasus Uji	Lihat Petunjuk Arah
Objek Uji	Kebutuhan Fungsional
Tujuan Pengujian	Pengujian dilakukan untuk memastikan sistem mampu memberikan visualisasi lokasi restoran melalui peta dan dapat memberikan arah dari lokasi konsumen ke lokasi restoran.
Prosedur Uji	<p>Skenario 1:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama sistem. 2. Konsumen memilih menu <i>search</i> pada navbar. 3. Konsumen melakukan pencarian restoran atau langsung memilih pada <i>list</i> yang ada pada halaman <i>search</i>. 4. Konsumen menekan tombol <i>map</i>. 5. Sistem menampilkan peta yang dilengkapi dengan fasilitas untuk mendapatkan petunjuk arah. 6. Konsumen mengisikan lokasi pada <i>field start</i> dan dapat memilih mode travel yang diinginkan. <p>Skenario II :</p> <ol style="list-style-type: none"> 1. Sistem menampilkan halaman utama sistem. 2. Konsumen memilih menu <i>recommendation</i> pada navbar. 3. Sistem menampilkan <i>form</i> untuk membantu konsumen mencari rekomendasi dengan beberapa kriteria. . 4. Konsumen memilih kriteria-kriteria yang diinginkan kemudian menekan tombol <i>search</i>

	<ol style="list-style-type: none"> 5. Sistem menampilkan hasil rekomendasi restoran berdasarkan kriteria konsumen yang disertai dengan peta dan fasilitas untuk mendapatkan petunjuk arah 6. Konsumen dapat memilih restoran yang akan dituju dan mode travel yang diinginkan
Hasil Yang Diharapkan	Sistem dapat menampilkan peta lokasi restoran yang disertai dengan fasilitas petunjuk arah.
Hasil Yang Didapatkan	Sistem menampilkan peta lokasi restoran yang disertai dengan fasilitas petunjuk arah..
Status Validasi	Valid

Sumber : Pengujian dan Analisis

5.1.3 Pengujian Kompabilitas *Device*

Pengujian kompatibilitas berguna untuk mengetahui kompabilitas antarmuka sistem. Teknik yang digunakan untuk pengujian ini adalah *black box testing*. Pengujian akan dilakukan pada beberapa device dengan berbagai resolusi layar menggunakan *emulator online quirkttools.com/screenfly*. Hal ini dikarenakan *screenfly* didukung oleh banyak resolusi layar hingga resolusi layar televisi.

1. Pengujian Kompabilitas pada Tablet *Samsung Galaxy Tab (600 x 1024)*

Tabel 5. 34 Uji Validitas Kompabilitas Tablet Samsung

Nama Kasus Uji	Pengujian kompatibilitas pada Tablet <i>Samsung Galaxy Tab</i> dengan resolusi 600 x 1024.
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fitur-fitur dan antarmuka pengguna pada tablet dengan resolusi layar sebesar 600 x 1024.
Prosedur Uji	Membuka setiap halaman sesuai spesifikasi kebutuhan.
Hasil Yang Diharapkan	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan perancangan antarmuka sistem.
Hasil Yang Didapatkan	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai implementasi antarmuka sistem pada tablet

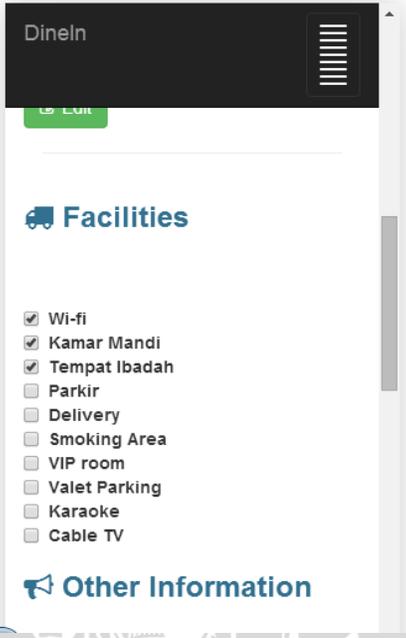
	dengan resolusi 600 x 1024.
Screenshot	
Status Validasi	Valid

Sumber : Pengujian dan Analisis

2. Pengujian Kompabilitas pada *Smartphone* Apple Iphone 5 (320 x 568)

Tabel 5. 35 Pengujian Kompabilitas *Smartphone* Apple

Nama Kasus Uji	Pengujian kompatibilitas pada <i>Smartphone</i> Apple Iphone 5 dengan resolusi (320 x 568)
Tujuan Pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fitur-fitur dan antarmuka pengguna pada <i>smartphone</i> dengan resolusi layar sebesar 320 x 568.
Prosedur Uji	Membuka setiap halaman sesuai spesifikasi kebutuhan.
Hasil Yang Diharapkan	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan perancangan antarmuka sistem.
Hasil Yang Didapatkan	Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai implementasi antarmuka sistem pada <i>smartphone</i> dengan resolusi 320 x 568.

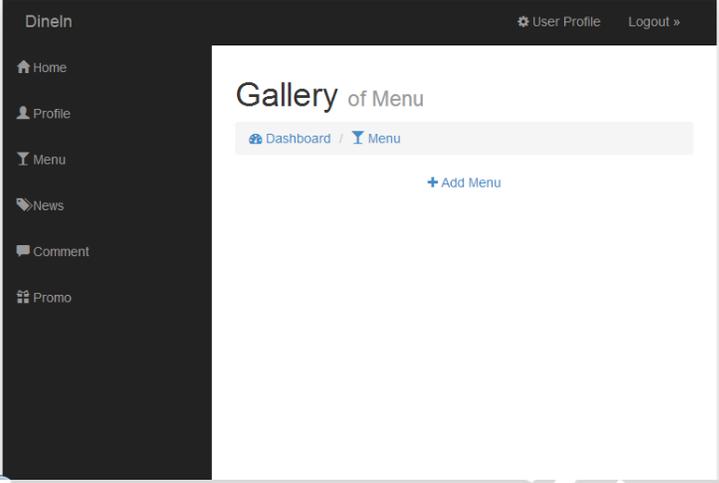
<p>Screenshot</p>	
<p>Status Validasi</p>	<p>Valid</p>

Sumber : Pengujian dan Analisis

3. Pengujian Kompabilitas pada TabletIpad Mini (768 x 1024)

Tabel 5. 36 Uji Kompatibilitas Ipad Mini

<p>Nama Kasus Uji</p>	<p>Pengujian kompatibilitas pada TabletIpad Mini dengan resolusi (768 x 1024)</p>
<p>Tujuan Pengujian</p>	<p>Untuk menguji validitas kinerja dari sistem dalam menyediakan fitur-fitur dan antarmuka pengguna pada <i>smartphone</i> dengan resolusi layar sebesar 768 x 1024.</p>
<p>Prosedur Uji</p>	<p>Membuka setiap halaman sesuai spesifikasi kebutuhan.</p>
<p>Hasil Yang Diharapkan</p>	<p>Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai dengan perancangan antarmuka sistem.</p>
<p>Hasil Yang Didapatkan</p>	<p>Sistem dapat menampilkan fitur-fitur dan antarmuka sesuai implementasi antarmuka sistem pada tabletdengan resolusi768 x 1024.</p>

Screenshot	
Status Validasi	Valid

Sumber : Pengujian dan Analisis

5.1.4 Pengujian Akurasi Sistem

Pengujian ini berfungsi untuk mengetahui apakah algoritma yang diimplementasikan sudah benar. Lima studi kasus berbeda akan diujikan pada sistem ini. Prosedur pengujiannya adalah menginputkan kriteria dan bobot yang nantinya akan menghasilkan rekomendasi restoran. Hasil rekomendasi tersebut kemudian dicocokkan kesesuaiannya dengan prediksi dari perhitungan manual dengan menggunakan rumus yang telah dibahas pada bab 2.

Hasil pengujian akurasi dari 5 kasus uji yang diuji adalah sebagai berikut:

Tabel 5. 37 Uji Akurasi Sistem Rekomendasi

Input	Hasil Perhitungan Manual	Hasil Sistem Rekomendasi	Keterangan
<ul style="list-style-type: none"> - Jenis : Bakso - Harga : (15.001-30.000) - Jarak/lokasi : Dekat - Fasilitas : Any - Lokasi saat ini : Jalan Mertojoyo Malang - Bobot Kepentingan : <ol style="list-style-type: none"> 1. Harga : Penting 2. Jarak/lokasi : sangat penting 	<ul style="list-style-type: none"> - Bakso Kota Cak Man/ Bakso Bakar Pahlawan Trip - Bakso Gong/ Bakso Priangan Mang Yayat/ Bakso Bakar Pak Man 	<ul style="list-style-type: none"> - Bakso Kota Cak Man Bakso Bakar Pahlawan Trip - Bakso Gong Bakso Priangan Mang Yayat Bakso Bakar Pak Man 	<p>P</p>
<ul style="list-style-type: none"> - Jenis : Japanese - Harga : (30.001-50.001) - Jarak/lokasi : any - Fasilitas : Any - Lokasi saat ini : Jalan Mertojoyo Malang - Bobot Kepentingan : <ol style="list-style-type: none"> 1. Harga : Sangat Penting 	<ul style="list-style-type: none"> - Hachi-Hachi Bistro - Sugoitei/ KL Suki Hoka-Hoka Bento/ Saboten Shokudo/ Japanese Bento 	<ul style="list-style-type: none"> - Hachi-Hachi Bistro - Sugoitei KL Suki Hoka-Hoka Bento Saboten Shokudo Japanese Bento 	<p>P</p>
<ul style="list-style-type: none"> - Jenis : Japanese - Harga : (< 15.000) - Jarak : Sedang - Fasilitas : Wi-fi 	<ul style="list-style-type: none"> - Sugoitei - Saboten Shokudo - Hachi hachi Bistro - KL Suki/ Japanese Bento/ 	<ul style="list-style-type: none"> - Sugoitei - Saboten Shokudo - Hachi hachi Bistro - KL Suki Japanese Bento 	<p>P</p>

<ul style="list-style-type: none"> - Lokasi saat ini : Jalan Raya Dieng Malang - Bobot Kepentingan <ol style="list-style-type: none"> 1. Harga : sangat penting 2. Lokasi : penting 3. Fasilitas : cukup penting 	<p>Hoka-Hoka Bento</p>	<p>Hoka-Hoka Bento</p>	
<ul style="list-style-type: none"> - Jenis : Indonesia - Harga : (50.001-100.000) - Jarak : Dekat - Fasilitas : wifi dan kamar mandi - Lokasi saat ini : Jalan Raya Soekarno-Hatta Malang - Bobot kepentingan <ol style="list-style-type: none"> 1. Harga : sangat penting 2. Lokasi : penting 3. Fasilitas : cukup penting 	<ul style="list-style-type: none"> - Taman Indie-River - Warung Lesehan Jogjakarta - Museum resto inggil/ Harmoni Resto/ Warung LeKo - Ayam Bakar Wong Solo/ Warna Warni 	<ul style="list-style-type: none"> - Taman Indie-River - Warung Lesehan Jogjakarta - Museum resto inggil Harmoni Resto Warung LeKo - Ayam Bakar Wong Solo Warna Warni 	<p>P</p>
<ul style="list-style-type: none"> - Jenis : Indonesia 	<ul style="list-style-type: none"> - Warung Lesehan Jogjakarta 	<ul style="list-style-type: none"> - Warung Lesehan Jogjakarta 	<p>P</p>

<ul style="list-style-type: none"> - Harga : (< 15.000) - Jarak : Dekat - Fasilitas : Kamar mandi - Lokasi saat ini : Jalan Raya Soekarno-Hatta Malang - Bobot Kepentingan <ol style="list-style-type: none"> 1. Harga : Sangat Penting 2. Lokasi : penting 3. Fasilitas : cukup penting 	<ul style="list-style-type: none"> - Ayam Bakar Wong Solo/ Museum Resto Inggil - Taman Indie-River / Harmoni Resto/ Warung LeKo/ Warna Warni 	<ul style="list-style-type: none"> - Ayam Bakar Wong Solo Museum Resto Inggil - Taman Indie-River Harmoni Resto Warung LeKo Warna Warni 	
--	--	---	--

Sumber : Pengujian dan Analisis

Keterangan :

- P: jika sistem mampu menghasilkan rekomendasi dari inputan konsumen dan hasilnya sama dengan hasil perhitungan manual.
- N: jika sistem tidak mampu menghasilkan rekomendasi dari inputan konsumen dan hasilnya sama dengan hasil perhitungan manual.

Jika P adalah positif, dan N adalah negatif, maka:

$$accuracy = \frac{N_c}{N}$$

$$= 5 / 5 = 100\%$$

Berdasarkan hasil pengujian akurasi dengan 5 uji kasus dihasilkan akurasi 100 %.

5.1.5 UAT (User Acceptance Testing)

Pengujian UAT dilakukan dengan melibatkan pengguna. Pengujian ini dilakukan dengan cara pengisian kuisioner dengan jumlah pertanyaan 10 butir dengan jumlah koresponden 100 orang. Pengujian UAT bertujuan untuk melihat tingkat penerimaan pengguna terhadap sistem yang telah dibuat. Ada dua variabel yang diukur dalam pengujian ini, yaitu : Kemudahan penggunaan sistem (*perceived ease of use*) dan pencapaian kegunaan sistem (*perceived of usefulness*).

1. Penentuan Metode Pengumpulan Data dan Penentuan Sampel

Pengumpulan data dilakukan dengan cara survei langsung terhadap responden. Hasil yang diperoleh dari metode tersebut adalah informasi yang berdasarkan jawaban-jawaban yang diajukan ke responden melalui kuesioner. Kuesioner yang dibagi berisikan pertanyaan-pertanyaan yang telah dirumuskan sehingga responden dapat memberikan informasi yang berupa jawaban dari pertanyaan tersebut.

Penentuan sampel dalam penelitian ini adalah dengan menggunakan teknik probability sampling. Metode pengambilan sampel ini merupakan suatu metode pemilihan dimana setiap anggota populasi mempunyai peluang yang sama untuk dipilih menjadi anggota sampel. Salah satu teknik probability sampling adalah simple random sampling. Pengambilan sampel metode ini dilakukan secara acak dan simple.

Populasi yang akan diambil untuk bahan penelitian adalah masyarakat Kota Malang. Menurut data Dispendukcapil (Dinas Kependudukan dan Pencatatan Sipil) kota Malang, jumlah penduduk Kota Malang yang berumur ≥ 17 tahun pada tahun 2013 sejumlah 567.650 orang. Dalam perhitungan jumlah sampel digunakan rumus slovin sebagai berikut :

$$\begin{aligned} n &= \frac{N}{1+Ne^2} \\ &= \frac{567.650}{1+567.650(10\%)^2} \\ &= \frac{567.650}{1+5.676,5} \\ &= 99,98 \approx 100 \text{ responden} \end{aligned}$$

Keterangan :

n = Ukuran sampel
N = Ukuran populasi

e = Persen kelonggaran ketidakteelitian karena kesalahan pengambilan sampel yang masih dapat ditolerir

maka jumlah responden yang akan diwanwancarai adalah sebanyak 100 responden.

2. Penyusunan Instrumen

Penyusunan instrumen untuk pengujian UAT, menggunakan kuisioner dengan skala Likert 5 respons, yaitu :

Sangat Setuju(SS) : 5

Setuju(S) : 4

Cukup Setuju(CS) : 3

Tidak Setuju(TS) : 2

Sangat Tidak Setuju(STS): 1

Pertanyaan pada kuisioner didasarkan pada model *usefulness* dan *ease of use*.

Berikut adalah daftar pertanyaan untuk poin *usefulness*.

Usefulness
Sistem Rekomendasi Kuliner ini berguna bagi saya. .
Sistem Rekomendasi Kuliner ini memudahkan saya dalam menentukan restoran yang saya butuhkan.
Sistem Rekomendasi Kuliner ini sangat efektif dalam menentukan restoran yang saya butuhkan.
Sistem Rekomendasi Kuliner ini dapat membantu saya dalam menentukan restoran yang saya butuhkan lebih cepat.
Sistem ini memungkinkan saya untuk mendapatkan informasi restoran yang saya butuhkan.

Sedangkan tabel menunjukkan daftar pertanyaan untuk poin *ease of use*.

Ease of Use
Pengoperasian/penggunaan sistem rekomendasi kuliner ini mudah untuk dipelajari.
Sistem rekomendasi kuliner ini mudah dioperasikan/digunakan.
Informasi yang ditampilkan pada sistem rekomendasi kuliner sudah jelas dan mudah dipahami.
Interaksi saya dengan sistem rekomendasi kuliner ini jelas dan mudah dimengerti.
Mudah untuk mengingat bagaimana mengoperasikan/menggunakan sistem rekomendasi kuliner ini.

3. Pengumpulan data

Pengumpulan data dilakukan setelah penyebaran kuesioner. Data yang telah dikumpulkan kemudian diolah sesuai dengan analisis data yang akan dilakukan. Analisis yang digunakan pada penelitian ini adalah analisis deskriptif.

Berikut ini adalah informasi demografi dari subjek penelitian ini.

Tabel 5. 38 Demografi Responden

Kriteria	Sub-kriteria	Jumlah
Jenis Kelamin	Laki-Laki	43
	Perempuan	57
Usia	≤ 20 tahun	21
	21-30 tahun	37
	31-40 tahun	24
	41- 50 tahun	10
	>50 tahun	8
Pekerjaan	PNS	20
	Karyawan Swasta	27
	Pelajar/Mahasiswa	47
	Lain-lain	6
Pendidikan Terakhir	SD dan sederajat	1
	SMP dan sederajat	9
	SMA dan sederajat	50
	Perguruan tinggi	40

Sumber : Pengujian dan Analisis

5.2 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian sistem rekomendasi kuliner yang telah dilakukan. Proses analisis yang dilakukan adalah analisis pengujian validasi, pengujian kompatibilitas, pengujian akurasi sistem, dan pengujian UAT.

5.2.1 Analisis Pengujian Basis Path

Proses analisis terhadap hasil pengujian basis path dilakukan dengan melihat hasil perhitungan nilai V pada setiap fungsi untuk melakukan rekomendasi. Berdasarkan hasil pengujian tersebut dapat disimpulkan bahwa :

Tabel 5. 39 Hasil Pengujian Basis Path

Fungsi	Cyclomatic Complexity	Resiko
search()	1	Rendah
hitung()	13	Menengah
hitung1()	7	Rendah
hitung2()	9	Rendah

Sumber: Pengujian dan Analisis

5.2.2 Analisis Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa implementasi dan fungsionalitas sistem rekomendasi kuliner Kota Malang dengan berbasis web menggunakan metode *weighted product* sudah memenuhi kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan. Akan tetapi, untuk skenario alternatif kelola fasilitas, jenis makanan, dan cari restoran masih belum sesuai dengan hasil yang diharapkan pada sistem. Oleh karena itu, masih diperlukan pengembangan dan perbaikan pada implementasi sistem rekomendasi kuliner agar dapat berjalan baik dan sesuai dengan seluruh kebutuhan yang telah dijabarkan pada tahap analisis kebutuhan.

5.2.3 Analisis Pengujian Kompatibilitas

Proses analisis terhadap hasil pengujian kompatibilitas yang dilakukan pada sistem rekomendasi kuliner terhadap beberapa jenis *device* yang berbeda dilakukan untuk melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan.

Berdasarkan hasil pengujian kompatibilitas dapat disimpulkan bahwa

1. Dalam pengujiannya, aplikasi dapat diuji menggunakan simulator *online*. Pada pengujian ini, simulator yang digunakan adalah *screenfly* dengan mengetikkan alamat *localhost* sistem pada emulator tersebut. *Screenfly* digunakan karena didukung oleh banyak resolusi layar hingga resolusi layar televisi.

2. Sistem rekomendasi mampu menghasilkan antarmuka yang dinamis dan responsive berdasarkan ukuran layar dari device yang digunakan. Hal ini disebabkan karena penggunaan bootstrap 3 yang membuat *interface* sistem rekomendasi kuliner menjadi responsif.

5.2.4 Analisis Pengujian Akurasi Sistem

Proses analisis terhadap hasil pengujian akurasi sistem yang dilakukan pada sistem rekomendasi kuliner dilakukan untuk melihat kesesuaian antara hasil rekomendasi sistem dengan perhitungan manualisasinya.

Berdasarkan hasil pengujian akurasi sistem dapat disimpulkan bahwa implementasi algoritma sistem rekomendasi kuliner ini sudah sesuai dengan perhitungan manualisasi yang ada dengan akurasi sebesar 100 %.

5.2.5 Analisis Pengujian UAT

Langkah-langkah untuk melakukan analisis pengujian UAT adalah melakukan perhitungan nilai dari skala likert yang telah didapatkan dari kuisioner. Adapun langkah-langkahnya :

- a. Menentukan skor ideal, merupakan skor yang digunakan untuk menghitung skor untuk menentukan rating scale dan jumlah seluruh jawaban. Untuk menghitung jumlah skor ideal (kriterium) dari seluruh item, digunakan rumus berikut, yaitu.

Skor Kriterium = Nilai skala x Jumlah responden

Sehingga didapatkan,

Tabel 5. 40 Skor Ideal

Rumus	Skala
$5 \times 100 = 500$	Sangat Setuju
$4 \times 100 = 400$	Setuju
$3 \times 100 = 300$	Cukup Setuju
$2 \times 100 = 200$	Tidak Setuju
$1 \times 100 = 100$	Sangat Tidak Setuju

Sumber : Pengujian dan Analisis

- b. Selanjutnya kita menentukan nilai rating scale dan jarak intervalnya dari nilai pada tabel di atas.

Skala+Likert+Scale

Dengan ketentuan sebagai berikut.

Tabel 5. 41 Nilai Scale Rating

Rumus	Skala
401-500	Sangat Setuju
301-400	Setuju
201-300	Cukup Setuju
101-200	Tidak Setuju
0-100	Sangat Tidak Setuju

Sumber : Pengujian dan Analisis

- c. Menghitung skor nilai untuk setiap item pernyataan, yaitu dengan cara mengalikan bobot nilai dengan jumlah frekuensi (jumlah jawaban responden setiap alternatif jawaban tiap item pernyataan).
- d. Kemudian hasil skor nilai tersebut dibentuk dalam tabel distribusi frekuensi sebagai berikut.

Tabel 5. 42 Distribusi Hasil UAT

No	Item Pernyataan	tanggapan Responden										N	Skor	Kategori
		SS (5)		S (4)		CS (3)		TS (2)		STS (1)				
		f	%	F	%	F	%	F	%	f	%			
1	item 1	30	30	38	38	27	27	4	4	1	1	100	392	setuju
2	item 2	27	27	32	32	33	33	8	8	0	0	100	378	setuju
3	item 3	23	23	41	41	34	34	2	2	0	0	100	385	setuju
4	item 4	14	14	43	43	37	37	6	6	0	0	100	365	setuju
5	item 5	17	17	40	40	32	32	10	10	1	1	100	362	setuju
Skor nilai variabel <i>ease of use</i>												1882		
Rata-rata variabel <i>ease of use</i>												376.4	setuju	
6	item 6	24	24	48	48	21	21	5	5	0	0	100	385	setuju
7	item 7	29	29	49	49	20	20	2	2	0	0	100	405	sangat setuju
8	item 8	21	21	53	53	21	21	5	5	0	0	100	390	setuju
9	item 9	27	27	40	40	31	31	2	2	0	0	100	392	setuju
10	item 10	30	30	41	41	27	27	0	0	0	0	100	395	setuju
skor nilai variabel <i>usefulness</i>												1967		
rata-rata variabel <i>usefulness</i>												393.4	setuju	

Sumber : Pengujian dan Analisis

Berdasarkan tabel di atas dapat dideskripsikan tanggapan responden terhadap item-item variabel *ease of use* dan *usefulness*.

1. Tanggapan responden terhadap pernyataan (item 1), yaitu : “Pengoperasian/penggunaan sistem rekomendasi kuliner ini mudah untuk dipelajari” sebagian besar responden menyatakan setuju (38 %) dengan skor nilai 392. Hal ini membenarkan bahwa responden setuju pengoperasian/penggunaan sistem rekomendasi kuliner mudah untuk dipelajari. Tempat kedua, ditempati oleh kategori setuju dengan prosentase sebesar 30 %. Selanjutnya, kategori cukup setuju dengan prosentase 27%, Tidak setuju dengan 4 %, dan terakhir sangat tidak setuju dengan prosentase 1 %. Beberapa responden dengan usia di atas 40 tahun, mengatakan alasan mereka tidak setuju dengan pernyataan di atas karena mereka belum terbiasa dengan sistem *online*.
2. Tanggapan responden terhadap pernyataan (item 2) “Sistem rekomendasi kuliner ini mudah untuk dioperasikan “, sebagian besar responden menyatakan cukup setuju (33 %) dengan skor nilai 378. Hal ini membenarkan bahwa responden setuju sistem rekomendasi kuliner ini mudah untuk dioperasikan. Namun, bagi beberapa responden dengan usia di atas 40 tahun mengatakan bahwa sistem rekomendasi ini memiliki banyak inputan yang harus diisi saat mencari restoran sehingga sistem ini menjadi agak sulit untuk digunakan. Oleh karena itu, beberapa dari mereka pun menyarankan agar sistem menyediakan dokumentasi manual penggunaan sistem rekomendasi.
3. Tanggapan responden terhadap pernyataan (item 3) “Informasi yang ditampilkan pada sistem rekomendasi kuliner sudah jelas dan mudah dipahami. “, sebagian besar responden menyatakan setuju (41 %) dengan skor nilai 385. Hal ini membenarkan bahwa responden setuju informasi yang ditampilkan pada sistem rekomendasi kuliner sudah jelas dan mudah dipahami. Beberapa responden mengungkapkan bahwa sistem rekomendasi sudah cukup informatif bagi responden.
4. Tanggapan responden terhadap pernyataan (item 4) “Interaksi saya dengan sistem rekomendasi kuliner ini jelas dan mudah dimengerti “, sebagian besar responden menyatakan setuju (43 %) dengan skor nilai 365. Hal ini

membenarkan bahwa responden setuju interaksi sistem rekomendasi kuliner sudah jelas dan mudah dimengerti .

5. Tanggapan responden terhadap pernyataan (item 5) “Mudah untuk mengingat bagaimana mengoperasikan/menggunakan sistem rekomendasi kuliner ini “, sebagian besar responden menyatakan setuju (40 %) dengan skor nilai 362. Hal ini membenarkan bahwa responden setuju jika mudah untuk mengingat bagaimana mengoperasikan/menggunakan sistem rekomendasi.
6. Tanggapan responden terhadap pernyataan (item 6) “Sistem Rekomendasi Kuliner ini berguna bagi saya. “, sebagian besar responden menyatakan setuju (48 %) dengan skor nilai 385. Hal ini membenarkan bahwa responden setuju jika sistem rekomendasi kuliner berguna bagi mereka. Masih sedikitnya informasi kuliner di Kota Malang menyebabkan banyak responden mengatakan bahwa sistem ini berguna bagi mereka untuk mencari informasi dan rekomendasi restoran di Kota Malang.
7. Tanggapan responden terhadap pernyataan (item 7) “Sistem Rekomendasi Kuliner ini memudahkan saya dalam menentukan restoran yang saya butuhkan “, sebagian besar responden menyatakan setuju (49 %) dengan skor nilai 405. Hal ini membenarkan bahwa responden sangat setuju jika sistem rekomendasi kuliner memudahkan mereka dalam menentukan restoran yang mereka butuhkan.
8. Tanggapan responden terhadap pernyataan (item 8) “Sistem Rekomendasi Kuliner ini sangat efektif dalam menentukan restoran yang saya butuhkan“, sebagian responden menyatakan setuju (53 %) dengan skor nilai 390. Hal ini membenarkan bahwa responden sangat setuju dengan pernyataan di atas. Beberapa responden mengatakan bahwa dengan adanya sistem ini mereka tidak perlu repot untuk bertanya kepada orang lain yang sudah pernah mencoba ke restoran tersebut. Akan tetapi, ada responden lain yang mengatakan bahwa mencari rekomendasi restoran ke orang lain lebih efektif daripada menggunakan sistem.
9. Tanggapan responden terhadap pernyataan (item 9) “Sistem Rekomendasi Kuliner ini dapat membantu saya dalam menentukan restoran yang saya butuhkan lebih cepat “, sebagian responden menyatakan setuju (40%) dengan

skor nilai 392. Hal ini membenarkan bahwa responden setuju dengan sistem rekomendasi kuliner dapat membantu mereka dalam menentukan restoran yang mereka butuhkan lebih cepat.

10. Tanggapan responden terhadap pernyataan (item 10) “Sistem ini memungkinkan saya untuk mendapatkan informasi restoran yang saya butuhkan“, sebagian responden menyatakan setuju (41%) dengan skor nilai 395. Hal ini membenarkan bahwa responden setuju dengan pernyataan di atas.

Berdasarkan perhitungan rata-rata untuk variabel *ease of use* dan variabel *usefulness*, dapat dinyatakan bahwa sistem rekomendasi kuliner ini dapat digunakan dengan mudah dan berguna bagi responden dalam mencari rekomendasi dan informasi restoran.



BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, dan pengujian yang dilakukan, maka dapat disimpulkan bahwa :

1. Perancangan sistem rekomendasi kuliner Kota Malang dengan berbasis *web* menggunakan metode *weighted product* telah dibuat sesuai spesifikasi kebutuhan yang telah dianalisis sebelumnya.
2. Analisis kebutuhan untuk sistem rekomendasi ini menggunakan metode *object oriented analysis* dengan pemodelan UML (*Unified Modeling language*).
3. Perancangan sistem rekomendasi ini menggunakan pemodelan UML, yang meliputi : *use case diagram*, *class diagram*, *sequence diagram*, dan *activity Diagram*. Selain itu, perancangan basis data, algoritma, dan antarmuka sistem juga dilakukan.
4. Sistem rekomendasi kuliner Kota Malang dengan berbasis *web* menggunakan metode *weighted product* diimplementasikan menggunakan *codeigniter* dan *bootstrap*.
5. Model pengembangan perangkat lunak dengan menggunakan model *reuse-oriented software engineering* sesuai dengan pengembangan sistem rekomendasi kuliner ini karena mempercepat proses pengembangan sistem rekomendasi.
6. Sistem rekomendas kuliner diuji menggunakan beberapa pengujian yang meliputi : pengujian *basis path*, pengujian validasi sistem, pengujian kompatibilitas, pengujian akurasi sistem rekomendasi, dan *user acceptance testing*.
7. Berdasarkan pengujian *basis path* menunjukkan bahwa sistem rekomendasi kuliner ini memiliki nilai *cyclomatic complexity* dengan tingkat resiko rendah dan menengah.
8. Walaupun ada beberapa skenario alternatif yang tidak valid pada pengujian validasi sistem rekomendasi kuliner ini, tetapi semua

kebutuhan utama sistem rekomendasi ini sudah dapat berjalan dengan baik dan sesuai dengan apa yang diharapkan lewat pengujian validasi.

9. Berdasarkan hasil uji akurasi sistem, implementasi metode *weighted product* sesuai dengan perhitungan manualisasi dan menghasilkan akurasi sebesar 100 %.
10. Pada pengujian kompatibilitas pada beberapa *device* berbeda, sistem rekomendasi kuliner dapat menampilkan fitur dan antarmuka yang sesuai dengan perancangan.
11. Berdasarkan hasil pengujian *user acceptance testing*, sistem rekomendasi dapat diterima oleh konsumen dengan rata-rata variabel *ease of use* 376,4 dan variabel *usefulness* dengan rata-rata variabel 393,4. Berdasarkan skala rating yang dibuat, nilai kedua variabel tersebut untuk sistem rekomendasi ini sudah dapat diterima. Hal ini berarti bahwa sistem rekomendasi kuliner yang telah dikembangkan mudah untuk digunakan dan berguna bagi pengguna.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem rekomendasi kuliner Kota Malang selanjutnya adalah :

1. Untuk pengembangan selanjutnya, sistem dapat dikembangkan dengan menambahkan fitur reservasi tempat dan fitur *delivery online*. Selain itu, fitur rating dari pengguna juga dapat ditambahkan pada sistem rekomendasi kuliner ini
2. Untuk pengembangan selanjutnya, kriteria-kriteria yang digunakan pada metode *weighted product* dapat ditambahkan lagi sehingga hasil rekomendasi akan lebih variatif.
3. Untuk pengembangan selanjutnya, metode yang digunakan untuk rekomendasi dapat digabungkan dengan metode MCDM lainnya agar hasil rekomendasi lebih baik.

DAFTAR PUSTAKA

- Ahn,B.A,. (2006).*Multiattribute Decision Aid with Extended ISMAUT.IEEE Transaction on Systems, Man, And Cybernetics Part A-System and Humans*,36(3),507-520.
- Athawale, V.,M. & Chakraborty,Shankar.(2011). *A Comparative Study on the ranking performance of some multi-criteria decision-making methods for industrial robot selection*. International Journal of Industrial Engineering Computation,2,831-850.
- Astriana,Litha.(2014).*Sistem Pendukung Keputusan Pemilihan Penerima Jamkesmas Menggunakan Metode Weighted Product*. (Skripsi S1, Program Teknologi Informasi dan Ilmu Komputer,Universitas Brawijaya,2014).
- Alatas,Husein. (2013). *Responsive Web Design dengan PHP dan Bootstrap*. Yogyakarta: Lokomedia.
- Basyaib,F.(2006). *Teori Pembuatan Keputusan*. Diakses 2 Desember 2013, dari <http://books.google.co.id/books?id=1oX1gq9ofjYC&printsec=frontcover&hl=id#v=onepage&q&f=false>.
- Ciptomulyono.(2000). *Handout Multi Criteria Decision Making*.Surabaya: ITS.
- Cullen,F.(2012).*Restaurant Selection in Dublin* (Thesis,School of Culinary Arts and Food Technology Dublin Institute of Technology,2012). Diakses dari google scholar.
- Dashore,K.,Pawar,S.,S.,Sohani,N.,Verma,D.S.(2013).*Product evaluation Using Entropy and Multi Criteria Decision Making Methods*. International Journal of engineering Trends and Technology,4(5),2183-2187.
- Davis,F.D. (1989). *Perceived usefulness, perceived ease of use, and user acceptance of information technology*. MIS Quarterly,pp 319–340.
- Gangurde,S.,R.,(2011).*Ranking of Product Design Alternatives Using Multi-criteria Decision Making Methods*. International Journal of Operational Research,pp 101-111.
- Hamdani. (2009).*Visualisasi Geografi untuk Sistem Pendukung Keputusan Wisata Kuliner Menggunakan Metode Rule of Thumb*. Tesis S2 Program Magister Ilmu Komputer Universitas Gajah Mada, Yogyakarta.
- Hyun,S.S. (2010). *Predictors of Relationship Quality and Loyalty in The Chain Restaurant Industry*.Cornell Hosp.Q.,51(2),251-657.
- Jamal,A & Anastasiadou,K.(2009).*Investigating The Effects of Service Quality Dimensions and Expertise on Loyalty*. Eur.J.Mark.,43(3),pp 398-420.

- Kusrini.(2007). *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta:Andi.
- Kusumadewi,Hartati,Sri.,Harjoko,Wardoyo,A.,Retantyo.(2006). *Fuzzy Multi-Attribute Decision Making*. Graha Ilmu, Yogyakarta.
- Marcotte,Ethan.(2010). *Responsive Web Design*. Diakses 2 Januari 2014, dari <http://alistapart.com/article/responsive-web-design>.
- McGinty, L. dan Smyth,B. (2006). *Adaptive selection : analysis of critiquing and preference based feed back in conversational recommender systems*. International Journal of Electronic Commerce, 11(2), pp 35–57.
- Mitra,A.(1995).*Price Cue Utilization in Product Evaluations The Moderating Role of Motivation and Attribute Information*. Journal of Business Research,33(3),pp 187-195.
- Novita.(2012). *Sistem Pendukung Keputusan untuk Proses Penentuan Rumah Tangga Miskin Menggunakan Metode WP*. (Skripsi S1, Program Teknologi Informasi dan Ilmu Komputer,Universitas Brawijaya,2012).
- Pressman, Roger S. (2010). *Software Engineering : A Practitioner's Approach (7th ed.)*.Boston: McGraw Hill.
- Rosa,A.,S.& Shalahudin, M. (2011). *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Savitha,K. & Chandrasekar, C. (2011).*Vertical Handover Decision Schemes Using SAW And WPM For Network Selection in Heterogeneous Wireless Networks*. Global Journal of Computer Science And Technology,11(9),pp 20-24.
- Sebastia,L.,Garcia,L.,Onaindia,E.,Guzman,C.(2009).*E-tourism: A Tourist Recommendation and Planning application*.International Journal on Artificial Intelligence Tools,18(5),pp 717-738.
- Segars, A. H.; Grover, V. (1993), *Re-examining perceived ease of use and usefulness: A confirmatory factor analysis*, MIS Quarterly 17:.pp 517–525.
- Sekilas Kota Malang. (2014). *Sekilas Kota Malang*. Jawa Timur: Pemerintah Kota Malang.
- Sharda,N.(2010).*Building Visual Travel Recommender Systems and Tourism Communities for Effective User Experience*. In Sharda, N. (Ed.), Tourism Informatics: Visual Travel Recommender Systems, Social Communities, and User Interface Design. (pp. 1-22). DOI:10.4018/978-1-60566-818-5.ch001. Diakses 20 September 2013, dari academia.edu.

Sommerville, Ian. (2011). *Software Engineering* (9th ed.). Boston: Pearson/Addison-Wesley.

Tabucanon, Mario T. (1988). *Multiple Criteria Decision Making in industry*. Netherlands: Elsevier science publishing company. Inc.

Tan, Pang-Ning., Steinbach, M., Kumar, V. (2006). *Introduction to Data Mining*. United States: Addison-Wesley Company.

Utama, Canra. (2011). *Codeigniter framework*. Bandung: Universitas Pasundan.

UNIVERSITAS BRAWIJAYA



Lampiran 1. Kuesioner UAT

KUESIONER

**User Acceptance Testing Sistem Rekomendasi
Kuliner Di Kota Malang Dengan Berbasis Web
Menggunakan *Weighted Product***

Responden Yth.

Saya Mutiara Arinda Putri adalah mahasiswa Jurusan Informatika, Program Teknologi Informasi dan Ilmu Komputer , Universitas Brawijaya, yang sedang mengadakan pengujian kepada pengguna (*user acceptance testing*) terhadap sistem yang saya buat, yaitu : “Sistem Rekomendasi Kuliner Di Kota Malang Dengan Berbasis Web Menggunakan *Weighted Product*”. Pengujian ini dimaksudkan untuk mengetahui sejauh mana sistem yang dibuat sudah dapat diterima oleh pengguna. Oleh karena itu, sangat diharapkan partisipasi Anda dalam mengisi kuesioner ini. Terimakasih atas kerja sama dan perhatiannya.

**BAGIAN 1
DATA IDENTITAS RESPONDEN**

Pada bagian ini anda diminta untuk menjawab satu dari pilihan jawaban yang tersedia dengan memberikan tanda silang (X).

1. Berapa usia anda saat ini?

a. ≤ 20 tahun	d. 41- 50 tahun
b. 21-30 tahun	e. >50tahun
c. 31-40 tahun	
2. Jenis kelamin anda?

a. Perempuan	b. Laki-laki
--------------	--------------
3. Apa pekerjaan anda?

a. PNS	c. Pelajar/ Mahasiswa
b. Karyawan Swasta	d. Lain-lain.....
4. Apa pendidikan terakhir anda?

a. SD dan Sederajat	c. SMA dan Sederajat
b. SMP dan Seerajat	d. Perguruan Tinggi

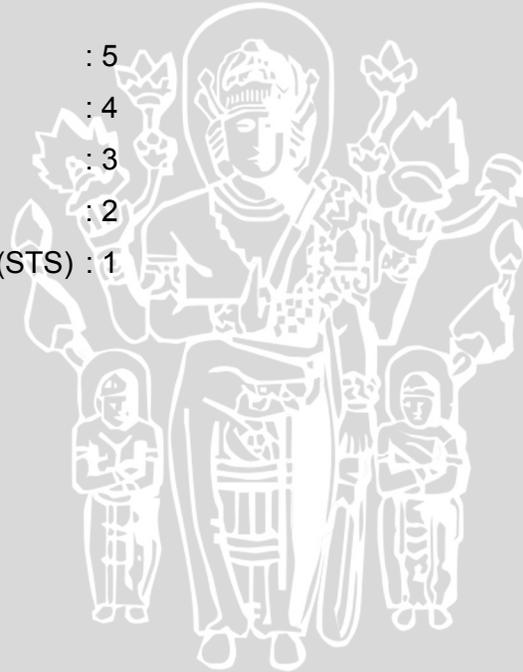
BAGIAN 2

Petunjuk Pengisian:

- Jawablah pertanyaan ini dengan jujur dan benar.
- Bacalah terlebih dahulu pertanyaan dengan cermat sebelum anda memulai jawabannya.
- Pilihlah salah satu jawaban yang tersedia dengan memberitanda silang (X) atau (\surd).
- Atas kesediaan anda untuk mengisi kuesioner ini terlebih dahulu saya ucapkan terima kasih.

Keterangan Skor:

- | | |
|-----------------------------|-----|
| a. Sangat Setuju(SS) | : 5 |
| b. Setuju(S) | : 4 |
| c. Cukup Setuju(CS) | : 3 |
| d. Tidak Setuju(TS) | : 2 |
| e. Sangat Tidak Setuju(STS) | : 1 |



No	Pertanyaan	Pilihan jawaban anda					Keterangan
		SS (5)	S (4)	CS (3)	TS (2)	STS (1)	
1	Pengoperasian/penggunaan sistem rekomendasi kuliner ini mudah untuk dipelajari.						
2	Sistem rekomendasi kuliner ini mudah dioperasikan/digunakan.						
3	Informasi yang ditampilkan pada sistem rekomendasi kuliner sudah jelas dan mudah dipahami.						
4	Interaksi saya dengan sistem rekomendasi kuliner ini jelas dan mudah dimengerti.						
5	Mudah untuk mengingat bagaimana mengoperasikan/menggunakan sistem rekomendasi kuliner ini.						
6	Sistem Rekomendasi Kuliner ini berguna bagi saya. .						
7	Sistem Rekomendasi Kuliner ini memudahkan saya dalam menentukan restoran yang saya butuhkan.						
8	Sistem Rekomendasi Kuliner ini sangat efektif dalam menentukan restoran yang saya butuhkan.						

9	Sistem Rekomendasi Kuliner ini dapat membantu saya dalam menentukan restoran yang saya butuhkan lebih cepat.													
10	Sistem ini memungkinkan saya untuk mendapatkan informasi restoran yang saya butuhkan.													

