

**SISTEM REKOMENDASI KULINER DI MALANG
DENGAN METODE TOPSIS
(Technique for Order Preference by Similarity to Ideal Solution)**

SKRIPSI



Disusun oleh :

**Sigit Setiawan
08.18.107**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN

**SISTEM REKOMENDASI KULINER DI MALANG
DENGAN METODE TOPSIS
(Technique for Order Preference by Similarity to Ideal Solution)**

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Informatika Strata Satu (S-1)*



**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

SISTEM REKOMENDASI KULINER DI MALANG
DENGAN METODE TOPSIS
(Technique for Order Preference by Similarity to Ideal Solution)

Sigit Setiawan (08.18.107)

Program Studi Teknik informatika S-1,
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
Email: sigit2@gmail.com

Dosen Pembimbing : 1. Joseph Dedy Irawan, ST, MT
2. Ali Mahmudi, B.Eng, Phd

Abstrak

Banyak informasi kuliner pada saat ini membuat banyak masyarakat harus memilih mana informasi kuliner yang baik dan terjamin dikarenakan informasi yang ada belum tentu benar adanya. Disamping itu kebutuhan informasi kuliner dari setiap masyarakat berbeda-beda, tergantung dari selera, kesenangan dan minat tiap-tiap individu. Oleh sebab itu, perlu dibuatlah sebuah sistem rekomendasi informasi kuliner yang sudah tervalidasi dan terjamin kebenarannya.

Sistem rekomendasi merupakan suatu sistem yang bertujuan memperkirakan informasi yang menarik bagi penggunanya dan merekomendasikan suatu item dalam membuat suatu keputusan yang diinginkan oleh pengguna. Dalam penyelesaian masalah pada penelitian ini menggunakan metode TOPSIS karena metode ini berkonsep mendapatkan alternatif terbaik yang memiliki jarak terpendek dari solusi ideal negatif dan jarak terpanjang dari solusi ideal positif. Metode TOPSIS memiliki konsep sederhana dan mudah dipahami, komputasinya efisien dan mempunyai kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana.

Pengujian sistem rekomendasi ini, sebanyak 65% user merasa dapat terbantu dalam mencari dan mendapatkan tempat kuliner yang ada. Kecepatan penghitungan metode TOPSIS ini dipengaruhi jumlah alternatif dan kriteria yang ada dalam database.

Kata kunci: sistem rekomendasi, MADM, MCDM, TOPSIS, kuliner

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT. Karena atas rahmat dan hidayah-Nya yang telah dilimpahkan, sehingga kami dapat menyelesaikan Laporan Skripsi yang berjudul “SISTEM REKOMENDASI KULINER DI MALANG DENGAN METODE TOPSIS (Technique for Order Preference by Similarity to Ideal Solution)” ini dengan baik dan lancar.

Laporan Skripsi ini merupakan salah satu persyaratan akademik dalam menyelesaikan Jenjang Strata 1 Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.

Selesaiannya penyusunan skripsi ini berkat bantuan dari berbagai pihak oleh karena itu, pada kesempatan ini penyusun mengucapkan terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. **Tuhan Yang Maha Esa**, yang selalu memberikan kesehatan bagi penyusun sehingga dapat menyelesaikan Laporan Skripsi ini dengan baik.
2. **Kedua Orang Tua**, serta keluarga penyusun yang telah memberikan dorongan baik secara moril maupun materiel untuk menyelesaikan Laporan Skripsi ini.
3. **Bapak Ir. Soeparno Djiwo, MT.**, selaku Rektor Institut Teknologi Nasional Malang.
4. **Bapak Ir. Anang Subardi, MT.**, selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
5. **Bapak Joseph Dedy Irawan, ST., MT.**, selaku Ketua Program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang sekaligus dosen pembimbing I.
6. **Bapak Sonny Prasctio, ST, MT.**, selaku Sekretaris Program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang.
7. **Bapak Ali Mahmudi, B.eng, Phd.**, selaku dosen pembimbing II sekaligus Kepala Lab. RPL yang senantiasa membimbing dan mengarahkan penyusun selama menjadi anggota lab.
8. **Ibu Sandy Nataly Mantja, S.Kom.**, atas dukungan dan bimbingannya kepada penyusun selama studi.
9. **Ibu Karina Auliasari, ST, M.Eng.**, atas bimbingan, dukungan dan arahnya kepada penyusun selama studi dan menjadi anggota lab.

10. Ibu Febriana Santi Wahyuni, S.kom, M.Kom., atas dukungan dan bimbingannya kepada penyusun selama studi.
11. Seluruh dosen dan staf di Program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang.
12. Budi, Adi, Nuril, Ganda, Agil, Indra, ayong, asep, bonek, bos gembul, big booss aswin, gung eka, Novan, didik, rengga, herdi, indra mbek, Cak Ridwan, Asenk, penghuni kontrakan anak Lombok semua, seluruh anggota IT Club 99% Insting 1% otak, sahabat-sahabat seperjuangan bukan hanya di kuliah, tapi juga diseluruh keseharian penyusun selama di Malang, terima kasih atas segalanya.
13. Seluruh anggota Lab. yang ada di Jurusan Informatika baik yang sudah wisuda duluan maupun yang masih aktif, terutama Lab. RPL (zen, chelsy, krisna, syarif, rusda, zea, eko, ferrys, dan siapa saja yang penyusun lupa namanya). Penyusun ucapkan terima kasih banyak atas kerjasama dan pengalaman yang diberikan selama ini kepada penyusun.
14. Teman-teman seperjuangan angkatan 2008, 2009, dan 2010 Jurusan Informatika, terima kasih atas sharing-sharingnya dan pengalamannya selama ini.
15. Mas anang, Mas satrio, Mas rino, Mas angga, Mas agus gajul, Mas arifin, Mas deni, Mas sikun, Mas wahyu, Seluruh warga SH terate di ITN, UNMUH, UIN, dan pelatih Cabang yang mengenalkan penyusun dengan SH Terate, seluruh saudara-saudara saya dari UNMUH (mas abdi, mas heri, mas guru ali),KARLOS,UB,UM,UIN, atas dukungan dan semangatnya penyusun ucapkan terima kasih banyak.
16. Serta semua pihak yang telah membantu secara langsung maupun tidak langsung sehingga penulis dapat menyelesaikan laporan ini.

Penyusun menyadari laporan ini masih belum sempurna, oleh karena itu penyusun mengharap kritik dan saran serta penilaian yang bersifat membangun dari semua pihak guna sempurnanya laporan ini.

Akhir kata penyusun mohon maaf yang sebesar-besarnya bilamana dalam penyusunan laporan ini terdapat kekurangan serta kesalahan. Semoga Laporan Skripsi ini bermanfaat bagi kita semua.

Malang, 12 Februari 2013

Sigit Setiawan

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAK	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	v
DAFTAR TABEL.....	vii
DAFTAR GAMBAR.....	viii
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	3
1.4 Manfaat Penelitian.....	3
1.5 Batasan Masalah.....	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI.....	6
2.1 Sistem Rekomendasi	6
2.2 Multi Criteria Decesion Making (MCDM)	6
2.3 PHP (PHP Hypertext Preprocessor)	12
2.4 Framework.....	15
2.5 Codeigniter	16
2.6 MVC (Model View Controller).....	17
2.7 MySQL	19
2.8 SQL	20
2.9 PHPMyAdmin.....	23
BAB III ANALISIS DAN PERANCANGAN	25
3.1 Metodologi Penelitian	25
3.2 Metode TOPSIS	29
3.3 Hak Akses.....	32
3.4 Diagram Konteks.....	33
3.5 Data Flow Diagram (DFD).....	34
3.6 Entity Relationship Diagram (ERD)	37

3.7 Pengembangan Perangkat Lunak	38
BAB IV IMPLEMENTASI DAN PENGUJIAN	50
4.1 Implementasi Nilai Kecocokan	50
4.2 Implementasi Metode TOPSIS.....	50
4.3 Implementasi Komponen Aplikasi.....	54
4.4 Implementasi Interface	55
4.5 Hasil Implementasi.....	56
4.6 Pengujian.....	63
BAB V PENUTUP	68
5.1 Kesimpulan.....	68
5.2 Saran	68
DAFTAR PUSTAKA	69
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 Tabel Perbedaan antara MADM dan MODM	7
Tabel 3.1 Tabel Nilai Kecocokan dan Tingkat Kepentingan ^[4]	27
Tabel 3.2 Tabel analisa kriteria dalam penelitian	27
Tabel 3.3 Tabel bobot kriteria pada setiap alternatif	30
Tabel 3.4 Tabel Perencanaan pengembangan aplikasi.....	38
Tabel 3.5 Daftar kebutuhan fungsional.....	39
Tabel 3.6 Daftar kebutuhan non fungsional.....	39
Tabel 3.7 Daftar kebutuhan informasi	40
Tabel 3.8 Tabel admin	41
Tabel 3.9 Tabel atribut	41
Tabel 3.10 Tabel atribut detail	41
Tabel 3.11 Tabel kategori	42
Tabel 3.12 Tabel bobot	42
Tabel 3.13 Tabel kecocokan	42
Tabel 3.14 Tabel kriteria.....	43
Tabel 3.15 Tabel komentar	43
Tabel 3.16 Tabel menu.....	43
Tabel 3.17 Tabel pesan	44
Tabel 3.18 View atribut	44
Tabel 3.19 View kecocokan.....	45
Tabel 3.20 View kriteria	45
Tabel 4.1 Tabel nilai kecocokan	50
Tabel 4.2 Implementasi komponen aplikasi	54
Tabel 4.3 Daftar hasil implementasi <i>interface</i>	55
Tabel 4.4 Hasil pengujian terhadap beberapa <i>browser</i>	63
Tabel 4.5 Rencana dan bentuk pengujian disisi <i>user</i>	64
Tabel 4.6 Rencana dan bentuk pengujian disisi admin.....	64
Tabel 4.7 Rekap hasil pengujian blackbox pada sisi user.....	66
Tabel 4.8 Rekap hasil pengujian blackbox pada sisi admin	66
Tabel 4.9 Hasil pengujian terhadap user.....	67
Tabel 4.10 Hasil pengujian kecepatan perhitungan metode TOPSIS	67

DAFTAR GAMBAR

Gambar 2.1 Perbandingan PHP biasa dan codeigniter	18
Gambar 2.2 Flow Codeigniter.....	19
Gambar 2.3 Tampilan antarmuka PHPMyAdmin.....	24
Gambar 3.1 Desain Penelitian.....	25
Gambar 3.2 Flowchart metode TOPSIS	29
Gambar 3.3 Diagram Konteks	33
Gambar 3.4 DFD level 1 aplikasi sistem rekomendasi kuliner di Malang	34
Gambar 3.5 DFD level 2 sub proses perekomendasi kuliner.....	36
Gambar 3.6 Diagram relasi antar tabel	37
Gambar 3.7 Diagram keterhubungan antara M-V-C	40
Gambar 3.8 Layout halaman home	46
Gambar 3.9 Layout halaman rekomendasi	46
Gambar 3.10 Layout halaman login.....	47
Gambar 3.11 Layout halaman administrasi	47
Gambar 3.12. Layout halaman TOPSIS.....	48
Gambar 3.13 Layout halaman pesan dan komentar	48
Gambar 3.14 Layout halaman gallery.....	49
Gambar 3.15 Layout halaman setting	49
Gambar 4.1 Halaman home	56
Gambar 4.2 Halaman rekomendasi	57
Gambar 4.3 Halaman hasil rekomendasi	57
Gambar 4.4 Halaman about	58
Gambar 4.5 Halaman login.....	58
Gambar 4.6 Halaman dashboard.....	59
Gambar 4.7 Halaman header web.....	59
Gambar 4.8 Halaman tabel kriteria	60
Gambar 4.9 Halaman table atribut	60
Gambar 4.10 Halaman tabel kecocokan	61
Gambar 4.11 Halaman setting.....	61
Gambar 4.12 Halaman galery	62
Gambar 4.13 Halaman pesan	62
Gambar 4.14 Halaman komentar	63

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kuliner merupakan salah satu syarat utama pendukung berdirinya suatu tempat wisata, bahkan kuliner dapat menjadi daya tarik tersendiri suatu kawasan, daerah atau kota itu melebihi keindahan tempat wisatanya. Keanekaragaman kuliner yang ada saat ini baik di Indonesia maupun mancanegara semakin berkembang seiring dengan perkembangan jaman. Adanya nilai seni dalam dunia kuliner ini membuat makanan tidak lagi sekedar sebagai pemenuhan kebutuhan pokok, tetapi juga merupakan suatu karya seni dan gaya hidup. Ditambah lagi dengan selera tiap-tiap individu yang berbeda, di mulai dari masakan yang sederhana sampai individu-individu atau kelompok yang membuat atau menciptakan resep kuliner yang begitu khas. Hal ini mendorong peningkatan daya kreativitas dalam membuat kuliner yang enak dihidangkan, dan tidak membosankan serta nikmat untuk disantap.

Malang merupakan kota terbesar nomor dua setelah Surabaya di Jawa Timur. Malang juga kota yang indah sekaligus kaya akan sejarah, wisata kuliner, tujuan wisata, reruntuhan candi kuno, dan peninggalan budaya Jawa. Sehingga hal ini menarik wisatawan untuk datang ke kota tersebut. Ini di buktikan dengan masuknya kota Malang ke daftar 10 wisata tujuan wisata baru di Asia^[3].

Seiring perkembangan jaman, kota Malang juga semakin berkembang, begitupun dengan kulinernya. Ini dikarenakan Malang merupakan kota yang konsumtif karena setiap tahunnya banyak para pelajar dari berbagai kota, pulau bahkan luar indonesia datang ke Malang untuk mencari pendidikan yang lebih baik dari kota lain (karena di Malang terdapat universitas negeri maupun swasta yang cukup terkenal). Sehingga menyebabkan banyak para wirausahawan melirik bisnis di kota tersebut, salah satunya kuliner. Hal ini menyebabkan semakin banyaknya jenis, variasi dan tempat-tempat kuliner di Malang.

Semakin banyaknya informasi kuliner pada saat ini membuat masyarakat kebingungan untuk memilih informasi kuliner yang baik dan terjamin kebenarannya, terlebih lagi informasi yang ada belum tentu benar adanya. Apalagi kebutuhan akan informasi kuliner tersebut dari setiap masyarakat berbeda-beda,

tergantung dari selera, kesenangan dan minat tiap-tiap individu. Oleh sebab itu, perlu dibuatlah sebuah sistem yang dapat merekomendasikan informasi kuliner yang sudah tervalidasi dan terjamin kebenarannya.

Permasalahan yang dijabarkan atau diutarakan diatas merupakan permasalahan *Multi Attribute Decision Making (MADM)*. MADM merupakan salah satu model MCDM (*Multiple Criteria Decision Making*). MCDM adalah suatu metode pengambilan keputusan untuk mendapatkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. MADM biasanya digunakan untuk melakukan penilaian dan pilihan terhadap beberapa alternatif dalam jumlah yang terbatas (ruang diskrit). Dalam menyelesaikan masalah MADM ada beberapa metode yang dapat digunakan, salah satunya TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*).

Metode TOPSIS dipilih dikarenakan didasarkan pada konsep dimana alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif, namun juga memiliki jarak terpanjang dari solusi ideal negatif. Solusi ideal positif diartikan dengan solusi yang memaksimalkan atribut keuntungan (profit) dan meminimalkan atribut biaya (*cost*), sedang solusi ideal negatif diartikan dengan solusi yang meminimalkan atribut keuntungan (profit) dan memaksimalkan biaya (*cost*). Pendekatan TOPSIS ini juga memiliki beberapa kelebihan yaitu (1) Logikanya sederhana dan mudah dipahami, (2) komputasinya efisien, (3) alternatif yang terpilih merupakan model matematika sederhana.

Hal inilah yang melatar belakangi dilakukannya penelitian ini, yaitu: **“System Rekomendasi kuliner di Malang dengan Metode TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*)”**. Penelitian ini dilakukan dengan harapan dapat membantu para pecinta kuliner agar mendapatkan informasi kuliner di Malang sesuai yang diinginkan.

1.2. Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah bagaimana membangun sebuah sistem rekomendasi kuliner di Malang menggunakan metode TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*).

1.3. Tujuan

Tujuan dari penelitian ini adalah:

1. Membuat aplikasi sistem rekomendasi kuliner dengan menggunakan metode TOPSIS.
2. Mengetahui efektifitas dan efisiensi sistem rekomendasi dengan menggunakan metode TOPSIS.

1.4. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah agar sistem rekomendasi kuliner dengan menggunakan metode TOPSIS diharapkan dapat memudahkan *user* untuk mendapatkan informasi kuliner yang sesuai.

1.5. Batasan Masalah

Agar pembahasan dalam penulisan ini lebih terarah dan mencegah adanya pembahasan yang terlalu kompleks, maka penulis membuat batasan masalah yang akan dijadikan pedoman dalam pelaksanaan penelitian ini yaitu:

1. Alternatif yang digunakan merupakan tempat-tempat kuliner yang ada di Malang.
2. Sistem yang dihasilkan adalah sistem berbasis web dengan menggunakan metode *Multi Attribute Decision Making* (MADM) dengan metode penyelesaian TOPSIS dalam merekomendasikan kuliner di Malang.
3. Perancangan sistem menggunakan bahasa pemrograman PHP 5.2.6 dengan menggunakan *framework codeigniter* dan database MySQL 5.0.51b.
4. Data kuliner diperoleh dari website <http://www.halomalang.com>.

1.6. Metodologi Penelitian

1. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan bahan-bahan referensi baik dari buku, artikel, paper, jurnal, makalah, maupun situs-situs internet. Studi literatur yang dilakukan terkait dengan Sistem Rekomendasi, serta data tempat-tempat kuliner di Kota Malang, Jawa Timur.

2. Analisis Permasalahan

Pada tahap ini dilakukan analisis terhadap hasil studi literatur untuk mengetahui dan mendapatkan pemahaman mengenai metode TOPSIS pada Sistem Rekomendasi.

3. Perancangan Sistem

Pada tahap ini dilakukan perancangan arsitektur, perancangan data dan perancangan antarmuka. Proses perancangan dilakukan berdasarkan hasil analisis studi literatur yang telah didapatkan. Pada penelitian ini, model proses pengembangan perangkat lunak yang digunakan adalah *Web Engineering Process* karena perangkat lunak yang dibangun berbasis web. Perangkat lunak terbagi kedalam dua buah sistem utama. Sistem-sistem tersebut adalah :

a. Sistem *administrator*

Sistem ini merupakan sistem pengolahan data untuk perekomendasi, dimana admin dapat menambah, mengganti, mengedit, mengunggah foto untuk di tampilkan pada user. Sistem ini akan diproteksi dengan *token*, untuk melindungi halaman dari pengaksesan pihak yang tidak bertanggung jawab.

b. Sistem *user*

Sistem ini merupakan sistem rekomendasi kuliner yang berhubungan langsung dengan user. Sistem ini berjalan di dalam sebuah aplikasi berbasis web sehingga dapat diakses oleh pembaca dimana pun berada.

4. Implementasi Sistem

Pada tahap ini dilakukan proses implementasi yang merupakan tahap membangun sistem dan pengkodean program dengan menggunakan bahasa pemrograman PHP dengan menggunakan *framework codeigniter* yang telah dipilih yang sesuai dengan analisis dan perancangan yang sudah dilakukan. Implementasi terdiri atas pembuatan web portal yang dapat melakukan perhitungan dengan menggunakan metode TOPSIS, serta dapat merekomendasikan kuliner yang sekiranya disenangi oleh *user*.

5. Pengujian

Pada tahap ini dilakukan proses pengujian dan percobaan terhadap sistem yang telah dibuat serta memastikan program yang dibuat berjalan seperti yang diharapkan.

6. Penyusunan Laporan

Tahap terakhir merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan serta hasil implementasi dari metode TOPSIS pada sistem rekomendasi yang telah dibuat.

1.7. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan penelitian ini sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi pembahasan masalah secara umum meliputi 1.1 latar belakang, 1.2 rumusan masalah, 1.3 tujuan penelitian, 1.4 manfaat penelitian, 1.5 batasan masalah, 1.6 metode penelitian, dan 1.7 sistematika penulisan.

Bab II Landasan Teori

Bab ini berisi landasan teori yang berfungsi sebagai sumber dan alat dalam memahami permasalahan yang berkaitan dengan sistem rekomendasi menggunakan metode TOPSIS.

Bab III Analisis dan Perancangan

Bab ini berisi tentang penjelasan dari desain penelitian, metode yang digunakan dalam proses penelitian, serta alat dan bahan yang digunakan dalam penelitian dan perancangan dari sistem.

Bab IV Implementasi dan Pengujian

Bab ini berisi tentang implementasi penelitian dan pengujian dari sistem.

Bab V Penutup

Bab ini berisikan uraian tentang kesimpulan dan saran yang dapat digunakan untuk penelitian selanjutnya.

BAB II LANDASAN TEORI

2.1 Sistem Rekomendasi

Konsep sistem rekomendasi telah digunakan secara luas oleh hampir semua area bisnis dimana seorang konsumen memerlukan informasi untuk membuat suatu keputusan. Sistem rekomendasi bertindak sebagai pemandu keputusan personal untuk pengguna, membantu mereka dalam pengambilan keputusan tentang hal yang berkaitan dengan selera pribadi⁷¹.

Sistem rekomendasi merupakan *software tools* (web), teknik, alat personalisasi yang menyediakan pengguna sebuah informasi daftar item-item yang sesuai dengan keinginan masing-masing pengguna. Sistem rekomendasi menyimpulkan preferensi pengguna dengan menganalisis ketersediaan data pengguna, informasi tentang pengguna dan lingkungannya. Oleh karena itu, menyatakan sistem rekomendasi akan menawarkan kemungkinan dari penyaringan informasi personal sehingga hanya informasi yang sesuai dengan kebutuhan dan preferensi pengguna yang akan ditampilkan pada sistem dengan menggunakan sebuah teknik atau *model* rekomendasi.

2.2 Multi Criteria Decision Making (MCDM)

Multi Criteria Decision Making (MCDM) adalah suatu metode pengambilan keputusan untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Kriteria biasanya berupa ukuran-ukuran, aturan – aturan atau standar yang digunakan dalam pengambilan keputusan. Berdasarkan tujuannya MCDM dapat dibagi menjadi 2 *model* : *Multi Attribute Decision Making* (MADM) dan *Multi Objective Decision Making* (MODM). Seringkali MCDM dan MADM digunakan untuk menyelesaikan masalah-masalah dalam ruang diskrit. Oleh karena itu, pada MADM biasanya digunakan untuk melakukan penilaian atau seleksi terhadap beberapa alternatif dalam jumlah yang terbatas. Sedangkan MODM digunakan untuk menyelesaikan masalah-masalah pada ruang kontinyu (seperti permasalahan pada pemrograman matematis). Secara umum dapat dikatakan bahwa, MADM menyeleksi alternatif

terbaik dari sejumlah alternatif, sedangkan MODM merancang alternatif terbaik. Perbedaan mendasar terlihat pada tabel 2.1 :

Tabel 2.1 Perbedaan antara MADM dan MODM

	MADM	MODM
Kriteria (didefinisikan oleh)	Atribut	Tujuan
Tujuan	Implisit	Eksplisit
Atribut	Eksplisit	Implisit
Alternatif	Diskrit, dalam jumlah terbatas	Kontinyu, dalam jumlah tak terbatas
Kegunaan	Seleksi	Desain

Ada beberapa fitur umum yang akan digunakan dalam MCDM yaitu:

1. Alternatif, merupakan obyek-obyek yang berbeda dan memiliki kesempatan yang sama untuk dipilih oleh pengambil keputusan.
2. Atribut, atribut disebut juga sebagai karakteristik komponen, atau kriteria keputusan. Meskipun pada kebanyakan kriteria bersifat satu level, namun tidak menutup kemungkinan adanya sub kriteria yang berhubungan dengan kriteria yang telah diberikan.
3. Konflik antar kriteria, beberapa kriteria biasanya mempunyai konflik antara satu dengan yang lainnya, misalnya kriteria keuntungan akan mengalami konflik dengan kriteria biaya.
4. Bobot keputusan, bobot keputusan menunjukkan kepentingan relatif dari setiap kriteria, $W = (w_1, w_2, \dots, w_n)$. Pada MCDM akan dicari bobot kepentingan dari setiap kriteria.
5. Matriks keputusan, suatu matrik keputusan X yang berukuran $m \times n$, berisi elemen – elemen x_{ij} , yang merepresentasikan rating dari alternatif $A_i (i=1,2,\dots,m)$ terhadap kriteria $C_j (j=1,2,\dots,n)$ [4].

2.2.1 Klasifikasi Metode MCDM (*Multi Criteria Decision Making*)

Ada beberapa cara dalam mengklasifikasi metode MCDM. Menurut tipe data yang digunakan, MCDM dapat dibagi berdasarkan tipe *deterministic*, *stokastik* atau *fuzzy*. Menurut jumlah pengambil keputusan yang terlibat dalam proses pengambil keputusan. MCDM dapat dibagi berdasarkan pengambil

keputusan satu orang, atau pengambil keputusan dalam bentuk grup (kelompok)^[4].

2.2.2 Klasifikasi Solusi MCDM (*Multi Criteria Decision Making*)

Masalah MDCM tidak selalu memberikan solusi unik, perbedaan tipe bisa jadi akan memberikan perbedaan solusi.

1. Solusi ideal, kriteria atau atribut dapat dibagi menjadi dua kategori, yaitu kriteria yang nilainya akan dimaksimumkan (kategori nilai keuntungan), dan kriteria yang nilainya akan diminimumkan (kategori kriteria biaya). Solusi ideal akan memaksimumkan semua kriteria keuntungan dan meminimumkan semua kriteria biaya.
2. Solusi *non-dominated*, solusi ini sering juga dikenal dengan nama solusi *pareto-optimal*. Solusi *feasible* MDCM dikatakan *non-dominated* jika tidak ada solusi *feasible* lain yang menghasilkan perbaikan terhadap suatu atribut tanpa menyebabkan degenerasi pada atribut lainnya.
3. Solusi yang memuaskan, adalah himpunan bagian dari solusi-solusi *feasible* dimana setiap alternatif melampaui semua kriteria yang diharapkan.
4. Solusi yang lebih disukai, adalah solusi *non-dominated* yang paling banyak memuaskan pengambil keputusan^[4].

2.2.3 Konsep Dasar *Multi Attribute Decision Making* (MADM)

Pada dasarnya, proses MADM dilakukan melalui 3 tahap, yaitu: penyusunan komponen-komponen situasi, analisis, dan sintesis informasi pada tahap penyusunan komponen-komponen situasi, akan dibentuk tabel taksiran yang berisi identifikasi alternatif dan spesifikasi tujuan, kriteria dan atribut. Salah satu cara untuk menspesifikasikan tujuan situasi $\{O_i, i=1, \dots, l\}$ adalah dengan cara mendaftar konsekuensi-konsekuensi yang mungkin dari alternatif yang telah teridentifikasi $\{A_i, i=1, \dots, n\}$. Selain itu juga disusun atribut-atribut yang akan digunakan $\{a_k, k=1, \dots, m\}$.

Tahap analisis dilakukan melalui 2 langkah. Pertama, mendatangkan taksiran dari besaran yang potensial, kemungkinan dan ketidakpastian yang berhubungan dengan dampak-dampak yang mungkin pada setiap alternatif. Kedua, meliputi pemilihan dari preferensi pengambil keputusan untuk setiap nilai, dan ketidakpedulian terhadap resiko yang timbul. Pada langkah pertama,

beberapa metode menggunakan fungsi distribusi $p_i(x)$ yang menyatakan probabilitas kumpulan atribut a_k terhadap setiap alternatif A_j . Konsekuensi juga dapat ditentukan secara langsung dari agregasi sederhana yang dilakukan pada informasi terbaik yang tersedia. Demikian pula, ada beberapa cara untuk menentukan preferensi pengambil keputusan pada setiap konsekuensi yang dapat dilakukan pada langkah kedua. Metode yang paling sederhana adalah untuk menurunkan bobot atribut dan kriteria adalah dengan fungsi utilitas atau penjumlahan terbobot.

Secara umum, *model multi attribute decision making* dapat didefinisikan sebagai berikut:

Misalkan $A = \{a_i \mid i=1, \dots, n\}$ adalah himpunan alternatif-alternatif keputusan dan $C = \{c_j \mid j=1, \dots, m\}$ adalah himpunan tujuan yang diharapkan, maka akan ditentukan alternatif x^0 yang memiliki derajat harapan tertinggi terhadap tujuan-tujuan yang relevan c_j .

Sebagian besar pendekatan MADM dilakukan melalui 2 langkah yaitu: pertama, melakukan agregasi terhadap keputusan-keputusan yang tanggap terhadap semua tujuan pada setiap alternatif. Kedua, melakukan perankingan alternatif-alternatif keputusan tersebut berdasarkan hasil agregasi keputusan.

Dengan demikian, bisa dikatakan bahwa, masalah *multi attribute decision making* (MADM) adalah mengevaluasi m alternatif A_i ($i=1, 2, \dots, m$) terhadap sekumpulan atribut atau kriteria C_j ($j=1, 2, \dots, n$), dimana setiap atribut saling tidak bergantung satu dengan yang lainnya. Matrik keputusan setiap alternatif terhadap setiap atribut X , diberikan sebagai:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \quad \text{Persamaan (2.1)}$$

Dimana x_{ij} merupakan rating kinerja alternatif ke- i terhadap atribut ke- j . nilai bobot yang menunjukkan tingkat kepentingan relatif setiap atribut, diberikan sebagai, W :

$$W = \{w_1, w_2, \dots, w_n\}$$

Rating kinerja (X), dan nilai bobot (W) merupakan nilai utama yang merepresentasikan preferensi absolut dari pengambil keputusan. Masalah MADM diakhiri dengan proses perankingan untuk mendapatkan alternatif terbaik yang diperoleh berdasarkan nilai keseluruhan preferensi yang diberikan^[4].

2.2.4 Metode-Metode Penyelesaian Masalah MADM (*Multi Attribute Decision Making*)

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah MADM, antara lain:

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)^[4]

2.2.5 TOPSIS (*Technique For Order Preference by Similarity to Ideal Solution*)

TOPSIS adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang tahun 1981. TOPSIS didasarkan pada konsep dimana alternatif yang terpilih atau terbaik tidak hanya mempunyai jarak terdekat (terpendek) dari solusi ideal positif, namun juga memiliki jarak terjauh (terpanjang) dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *Euclidean* untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi ideal negatif terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut.

TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa dicapai. Metode ini banyak digunakan pada beberapa *model* MADM untuk menyelesaikan masalah pengambilan keputusan secara praktis. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien, dan memiliki kemampuan mengukur kinerja

relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana^[4].

2.2.5.1 Prosedur Metode TOPSIS (*Technique For Order Preference by Similarity to Ideal Solution*)

Secara umum, prosedur TOPSIS mengikuti langkah-langkah sebagai berikut:

1. Membuat matriks keputusan yang ternormalisasi
2. Membuat matriks keputusan yang ternormalisasi terbobot
3. Menentukan matriks solusi ideal positif dan matriks solusi ideal negatif
4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif
5. Menentukan nilai preferensi untuk setiap alternatif
 - a. Decision matrix D mengacu terhadap m alternatif yang akan dievaluasi berdasarkan kriteria yang didefinisikan sebagai berikut:

$$D = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \quad \text{Persamaan (2.2)}$$

- b. Dengan x_{ij} menyatakan performansi dari perhitungan untuk alternatif ke- i terhadap atribut ke- j ^[4].

2.2.5.2 Langkah-langkah Metode TOPSIS (*Technique For Order Preference by Similarity to Ideal Solution*)

1. Membangun matrik ternormalisasi
 Elemen r_{ij} hasil dari normalisasi *decision matrix* R dengan metode *Euclidean length of a vector* adalah:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}; \text{ dengan } i=1,2,\dots,m; \text{ dan } j=1,2,\dots,n \quad \text{Persamaan (2.3)}$$

2. Membangun matrik ternormalisasi terbobot
 Solusi ideal positif A^+ dan solusi ideal negatif A^- dapat ditentukan berdasarkan rating bobot ternormalisasi (y_{ij}) sebagai :

$$y_{ij} = w_i r_{ij}; \quad \text{dengan } i=1,2,\dots,m; \text{ dan } j=1,2,\dots,n \quad \text{Persamaan (2.4)}$$

3. Menentukan matriks solusi ideal positif dan matrik solusi ideal negatif

Solusi ideal positif (A^+) dihitung berdasarkan:

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+) \quad \text{Persamaan (2.5)}$$

dengan

$$y_j^+ = \begin{cases} \max_i y_{ij} & ; \text{jika } j \text{ adalah atribut keuntungan} \\ \min_i y_{ij} & ; \text{jika } j \text{ adalah atribut biaya} \end{cases} \quad \text{Persamaan (2.6)}$$

Solusi ideal positif (A^+) dihitung berdasarkan:

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+) \quad \text{Persamaan (2.7)}$$

dengan

$$y_j^- = \begin{cases} \min_i y_{ij} & ; \text{jika } j \text{ adalah atribut keuntungan} \\ \max_i y_{ij} & ; \text{jika } j \text{ adalah atribut biaya} \end{cases} \quad \text{Persamaan (2.8)}$$

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif

Jarak antara alternatif A_i dengan solusi ideal positif dirumuskan sebagai:

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i^+ - y_{ij})^2} ; i=1,2,\dots,m \quad \text{Persamaan (2.9)}$$

Jarak antara alternatif A_i dengan solusi ideal negatif dirumuskan sebagai:

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_i^-)^2} ; i=1,2,\dots,m \quad \text{Persamaan (2.10)}$$

5. Menentukan nilai preferensi untuk setiap alternatif

Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai^[4]:

$$V_i = \frac{D_i^-}{D_i^+ + D_i^-} ; i=1,2,\dots,m \quad \text{Persamaan (2.11)}$$

2.3 PHP (PHP Hypertext Preprocessor)

PHP singkatan dari PHP *Hypertext Preprocessor*, merupakan scripting yang dibundel dengan HTML, yang dijalankan di sisi server. Sebagian besar perintahnya berasal C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP juga dimaksudkan untuk mengganti teknologi lama seperti CGI (*Common Gateway Interface*).

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. *Script* yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi visitor, dan menampilkan jumlah pengunjung dari suatu website. Dan

sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan *script* buatan Rasmus Lerdorf, hingga akhirnya rasmus mulai membuat sebuah *tool/script*, bernama *Personal Home Page* (PHP).

Kebutuhan PHP sebagai *tool* yang serba guna membuat Lerdorf melanjutkan untuk mengembangkan PHP hingga menjadi suatu bahasa tersendiri yang mungkin dapat mengkonversikan data yang di *inputkan* melalui Form HTML menjadi suatu variabel, yang dapat dimanfaatkan oleh sistem lainnya. Untuk merealisasikannya, akhirnya Lerdorf mencoba mengembangkan PHP menggunakan bahasa C ketimbang menggunakan Perl. Tahun 1997, PHP versi 2.0 di rilis, dengan nama *Personal Home Page Form Interpreter* (PHP-FI). PHP Semakin populer, dan semakin diminati oleh *programmer* web dunia.

Rasmus Lerdorf benar-benar menjadikan PHP sangat populer, dan banyak sekali *Team Developer* yang ikut bergabung dengan Lerdorf untuk mengembangkan PHP hingga menjadi seperti sekarang. Hingga akhirnya dirilis versi ke 3-nya, pada Juni 1998, dan tercatat lebih dari 50.000 *programmer* menggunakan PHP dalam membuat website dinamis.

Pengembangan demi pengembangan terus berlanjut, ratusan fungsi ditambahkan sebagai fitur dari bahasa PHP, dan di awal tahun 1999, netcraft mencatat, ditemukan 1.000.000 situs di dunia telah menggunakan PHP. Ini membuktikan bahwa PHP merupakan bahasa yang paling populer digunakan oleh dunia web *development*. Hal ini mengagetkan para developernya termasuk Rasmus sendiri, dan tentunya sangat diluar dugaan sang pembuatnya. Kemudian Zeev Suraski dan Andi Gutschman selaku *core developer* (*programmer* inti) mencoba untuk menulis ulang PHP Parser, dan diintegrasikan dengan menggunakan *Zend scripting engine*, dan mengubah jalan alur operasi PHP. Dan semua fitur baru tersebut dirilis dalam PHP 4.

Pada tanggal 13 Juli 2004, evolusi PHP, PHP telah mengalami banyak sekali perbaikan disegala sisi, dan wajar jika netcraft mengumumkan PHP sebagai bahasa web populer didunia, karena tercatat 19 juta domain telah menggunakan PHP sebagai *server side* scriptingnya. PHP saat ini telah Mendukung XML dan Web Services, Mendukung SQLite. Tercatat lebih dari 19 juta domain telah menggunakan PHP sebagai server scriptingnya. Benar-benar PHP sangat mengejutkan.

Yang menjadikan PHP berbeda dengan HTML adalah proses dari PHP itu sendiri. HTML merupakan bahasa statis yang apabila kita ingin merubah konten/isinya maka yang harus dilakukan pertama kali nya adalah, membuka file-nya terlebih dahulu, kemudian menambahkan isi kedalam file tersebut. Beda halnya dengan PHP. Contohnya pada CMS wordpress atau Joomla yang dibangun dengan PHP, ketika akan menambahkan konten kedalam website, kita tinggal masuk kedalam halaman admin, kemudian pilih *new* artikel untuk membuat halaman/*content* baru. Artinya hal ini, seorang *user* tidak berhubungan langsung dengan scriptnya. Sehingga seorang pemula sekalipun dapat menggunakan aplikasi seperti itu.

Keunggulan PHP :

1. Gratis

PHP merupakan salah satu dari begitu banyak aplikasi *open source*. Aplikasi *open source* adalah aplikasi yang berlisensi GPL (*General Public License*) yang diperuntukkan secara bebas digunakan oleh Masyarakat Internasional dan sifatnya gratis, *open source* berarti kode sumbernya terbuka, dimana seorang pengguna, maupun pengembang, dapat mengkostumisasi, dan mengembangkan kode tersebut secara bebas. *Open source* memungkinkan para *programmer* dunia membuat aplikasi tersebut mengalami perkembangan yang sangat pesat. Oleh sebab itulah PHP begitu sangat populer pada saat ini.

2. Cross platform

Artinya dapat di gunakan di berbagai sistem operasi, mulai dari linux, windows, mac OS dan OS yang lain.

3. Mendukung banyak database

Sistem *database* yang dapat didukung oleh PHP diantara Mysql, MS-SQL, ORACLE, PostgreSQL, Sybase, ODBC, SQLite, dan lain-lain.

4. Mendukung Komunikasi banyak Protokol

PHP juga mendukung protokol komunikasi dengan layanan seperti HTTP, IMAP, POP3, SNMP, NNTP, FTP, dan lain-lain.

5. On The Fly

PHP sudah mendukung *on the fly*, artinya dengan php anda dapat membuat document text, Word, Excel, PDF, menciptakan image dan flash, juga menciptakan file-file seperti zip, XML, dan banyak lagi^[2].

2.4 Framework

Framework adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan, bahkan isu-isu kompleks yang ada. Sebuah *framework* telah berisi kumpulan arsitektur / konsep-konsep yang dapat mempermudah dalam pemecahan sebuah permasalahan. *Framework* bukanlah peralatan/*tools* untuk memecahkan sebuah masalah, tetapi sebagai alat bantu. *Framework* hanya menjadi sebuah konstruksi dasar menopang sebuah konsep atau sistem yang bersifat *essential suport* (penting tapi bukan komponen utama).

Framework banyak digunakan karena kemudahan yang ditawarkan. Didalam sebuah *framework* biasanya sudah tersedia struktur aplikasi yang baik, yaitu :

1. *standart coding*, merupakan sebuah standar yang harus diikuti oleh *programmer* untuk menulis *code*
2. *best practice*, merupakan kumpulan-kumpulan *action* yang telah teruji oleh para *expert*
3. *design pattern*, merupakan teknik-teknik yang menjadi *best practice*
4. *common function*, merupakan fungsi-fungsi atau *library* yang telah umum digunakan dalam pengembangan sebuah sistem

Dengan menggunakan *framework* kita dapat langsung fokus kepada *business proses* yang dihadapi tanpa harus berfikir banyak masalah struktur aplikasi, standart coding dan lain-lain.

Dengan memanfaatkan *design pattern* dan *common function* yang telah ada di dalam *framework* maka hal tersebut dapat mempercepat proses pengembangan aplikasi. Kita tidak perlu membuat sesuatu fungsionalitas yang bersifat umum. Tanpa disadari ketika kita membangun sebuah aplikasi yang banyak melibatkan banyak fungsionalitas yang telah dibangun itu ternyata sama atau berulang-ulang. Dengan pengelompokkan itulah kita dapat mempercepat pengembangan aplikasi.

Selain kemudahan dan kecepatan dalam membangun sistem, dengan menggunakan *framework* tertentu kita juga dapat menyeragamkan cara kita mengimplementasikan kode program. Dengan *framework* kita akan dipaksa untuk patuh kepada sebuah kesepakatan. Selain itu juga akan memudahkan pengembang lain untuk mempelajari dan mengubah aplikasi yang telah dibuat apabila kode yang dihasilkan konsisten dan patuh pada sebuah aturan tertentu¹¹.

2.5 Codeigniter

Codeigniter adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi php dinamis. *Codeigniter* dikembangkan oleh Rick Ellis. Tujuan utama pengembangan *codeigniter* adalah untuk membantu developer untuk mengerjakan aplikasi lebih cepat daripada menulis semua *code* dari awal. *Codeigniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengembangan. *Codeigniter* diperkenalkan kepada publik pada tanggal 28 februari 2006.

Codeigniter sendiri dibangun menggunakan konsep *Model-View-Controller development pattern*. *Codeigniter* sendiri merupakan salah satu *framework* tercepat dibandingkan dengan *framework* lainnya. Pada acara frOSCon (agustus 2008), pembuat php Rasmus Lerdorf mengatakan dia menyukai *codeigniter* karena dia lebih ringan dan cepat dibandingkan *framework* lainnya (*"because it is faster, lighter and least the least like a framework"*).

Codeigniter sangat ringan, terstruktur, mudah dipelajari, dokumentasi lengkap dan dukungan luar biasa dari forum *codeigniter*. Selain itu *codeigniter* juga memiliki fitur-fitur lainnya yang sangat bermanfaat, antara lain:

1. Gratis. *Codeigniter* dilisensikan dibawah lisensi Apache/BSD *style open source*, ini berarti kita dapat menggunakannya sesuai dengan keinginan kita.
2. Menggunakan *pattern* MVC. Dengan menggunakan *pattern* MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas.
3. Pustaka yang lengkap. *Codeigniter* dilengkapi dengan berbagai pustaka siap pakai untuk berbagai kebutuhan, misalnya koneksi *database*, *email*, *session*, *cookies*, *security*, manipulasi gambar, dan lain-lain.
4. *URL friendly*. URL yang dihasilkan sangat *url friendly*. Pada *codeigniter* diminimalisasi penggunaan `$_GET` dan digantikan dengan URL.
5. Kemudahan. Kemudahan dalam mempelajari, membuat *library* dan *helper*, memodifikasi serta meng-intregasikan *library* dan *helper*.

Jika membandingkan antara *codeigniter* dengan *framework-framework* lainnya maka beberapa poin yang membuat *codeigniter* unggul adalah:

1. Kecepatan. Berdasarkan hasil *bechmark codeigniter* merupakan salah satu *framework* PHP tercepat yang ada saat ini.
-

2. Mudah dimodifikasi dan beradaptasi. Sangat mudah memodifikasi *behavior framework* ini. Tidak membutuhkan *server requirement* yang macam-macam serta mudah mengadopsi *library* lainnya.
3. Dokumentasi lengkap dan jelas. *Codeigniter* menyediakan sebuah panduan yang lengkap dan semua informasi yang dibutuhkan tentang *codeigniter* ada disana.
4. *Learning curve* rendah. *Codeigniter* sangat mudah dipelajari. Jika sebuah *framework* sangat sulit dipelajari maka akan beresiko untuk memperlambat anda *men-development project anda*^[1].

2.6 MVC (*Model View Controller*)

MVC adalah konsep dasar yang harus diketahui sebelum mengenal *codeigniter*. MVC singkatan dari *Model View Controller*. MVC sebenarnya adalah *pattern / teknik pemrograman* yang memisahkan *bisnis logic* (alur pikir), *data logic* (penyimpanan data) dan *presentation logic* (antarmuka aplikasi) atau secara sederhana memisahkan antara desain, data dan proses. Adapun komponen-komponen MVC antara lain:

1. *Model*

Model berhubungan dengan data dan interaksi ke *database* atau *webservice*. *Model* juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun *webservice*. Biasanya didalam *model* akan berisi *class* dan fungsi untuk mengambil, melakukan *update* dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian *Model* biasanya akan berhubungan dengan perintah-perintah *query SQL*.

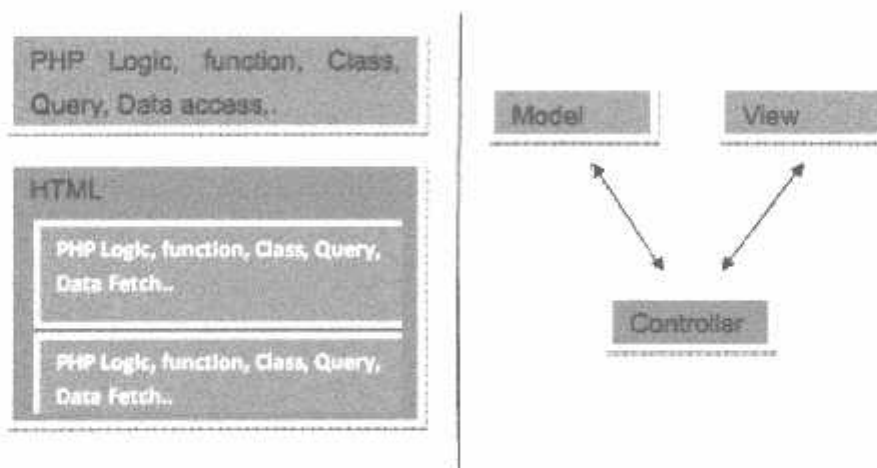
2. *View*

View berhubungan dengan segala sesuatu yang akan ditampilkan ke *end-user*. Bisa berupa halaman web, rss, javascript dan lain-lain. Kita harus menghinsari *adanya* logika atau pemrosesan data di *view*. Didalam *view* hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. *View* dapat dikatakan sebagai halaman website yang dibuat dengan menggunakan HTML dan bantuan CSS atau JavaScript. Didalam *view* jangan pernah ada kode untuk

melakukan koneksi ke basisdata. *View* hanya dikhususkan untuk menampilkan data-data hasil dari *model* dan *controller*.

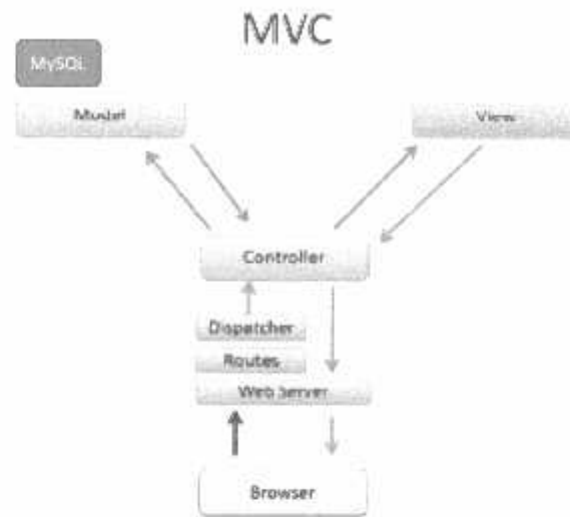
3. Controller

Controller bertindak sebagai penghubung data dan *view*. Didalam *controller* inilah terdapat *class-class* dan fungsi-fungsi yang memproses permintaan dari *view* ke dalam struktur sata di dalam *Model*. *Controller* juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada *model*. Tugas *controller* adalah menyediakan berbagai variabel yang akan ditampilkan di *view*, memanggil *model* untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/*error*, mengerjakan proses logika dari aplikasi serta melakukan validasi ata cek terhadap *input*.



Gambar 2.1 Perbandingan PHP biasa dan *codeigniter*

Perhatikan gambar 2.1. Pada eksekusi PHP, biasanya kita akan me-load semua *library* dan fungsi yang dibutuhkan kemudian digabungkan ke dalam HTML untuk dieksekusi oleh PHP. Untuk kasus sederhana cara tersebut masih baik-baik saja, tetapi ketika aplikasi tersebut menjadi kompleks / rumit maka kita akan sulit memeliharanya jika tidak didukung oleh arsitektur *software* yang bagus. Hal tersebut bisa terjadi disebabkan oleh *code* yang sama namun dibuat berulang-ulang, kode tidak konsisten dan lain-lain.



Gambar 2.2 *Flow Codeigniter*

Jika dipetakan, alur kerja *codeigniter* akan tampak seperti gambar 2.2. *Browser* berinteraksi melalui *controller*. *Controller* lah yang akan menerima dan membalas semua *request* dari *browser*. Untuk data maka *controller* akan meminta ke *model* dan untuk UI / *template* akan meminta ke *view*. Jadi “Otak” dari aplikasi ada di *controller*, “Muka” aplikasi ada di *view* dan “Data” ada di *model*. Ketika *Browser* meminta sebuah halaman web maka *router* akan mencari *controller* mana yang harus menangani *request* tersebut. Setelah itu barulah si *controller* menggunakan *model* untuk mengakses data dan *view* untuk menampilkan data tersebut^[1].

2.7 MySQL

MySQL adalah *database management system* (DBMS) untuk *relational databases* (karena itu MySQL juga termasuk RDMS). *Database* sendiri dalam istilah sederhana merupakan kumpulan data yang terkait, baik teks, angka atau file biner yang disimpan dan dikelola oleh DBMS (Larry,2008:xiv). *Database* digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

Untuk mengelola *database* diperlukan suatu perangkat lunak yang disebut DBMS (*Database Management System*). DBMS merupakan suatu sistem perangkat lunak yang memungkinkan *user* untuk membuat, mengelola, mengontrol, dan mengakses *database* secara praktis dan efisien. Dengan DBMS, *user* akan lebih mudah mengontrol dan memanipulasi data yang ada.

RDBMS atau *Relationship Database Management System* merupakan salah satu jenis DBMS yang mendukung adanya relationship atau hubungan antar tabel. Di samping RDBMS, terdapat jenis DBMS lain, misalnya Hierarchy DBMS, Object Oriented DBMS, dan sebagainya.

Salah satu keuntungan MySQL adalah MySQL merupakan perangkat lunak *open source*. *Open source* berarti dapat digunakan dan dimodifikasi oleh siapa saja. Semua orang dapat mengunduh MySQL dari internet dan menggunakannya secara gratis. Untuk administrasi *database*, seperti pembuatan *database*, pembuatan tabel, dan sebagainya, dapat digunakan aplikasi berbasis web seperti PHPMyAdmin. MySQL juga mendukung berbagai tipe data, multi *user*, mempunyai *security* yang baik, fleksibel dengan berbagai pemrograman, dan dukungan dari komunitas yang banyak^[8].

Untuk mempermudah administrasi server MySQL, biasanya digunakan beberapa tools, diantaranya :

1. Mysql command line *client*
2. Mysql front
3. PHPMyAdmin
4. SQLYog
5. MySQL Administrator dan MySQL *Query browser*
6. MySQL Workbench

2.8 SQL (*Structured Query Language*)

SQL (*Structured Query Language*) merupakan bahasa *query* yang digunakan untuk mengakses *database* relasional. SQL sekarang sudah menjadi bahasa *database* yang standar dan hampir semua sistem *database* memahaminya. SQL terdiri dari berbagai jenis *statement*. Semuanya didesain agar memungkinkan untuk dapat berhubungan dengan *database* secara interaktif.

Penggunaan SQL pada DBMS (*Database Management System*) sudah cukup luas. SQL dapat dipakai oleh berbagai kalangan, misalnya DBA (*Database Administrator*), *programmer* ataupun pengguna. Hal ini disebabkan oleh hal-hal berikut:

1. SQL sebagai bahasa administrasi *database*

Dalam hal ini SQL dipakai oleh DBA untuk menciptakan serta mengendalikan pengaksesan *database*.

2. SQL sebagai bahasa *query* interaktif

Pengguna dapat memberikan perintah-perintah untuk mengakses *database* yang sesuai dengan kebutuhannya.

3. SQL sebagai bahasa pemrograman *database*

Pemrogram dapat menggunakan perintah-perintah SQL. dalam aplikasi yang dibuat. Dalam hal ini adalah *embedded SQL*.

4. SQL sebagai bahasa *client/server*

SQL juga dipakai untuk mengimplementasikan sistem *client/server*. Sebuah *client* dapat menjalankan suatu aplikasi yang mengakses *database*. Dalam hal ini sistem operasi antara *server* dan *client* bisa berbeda.

SQL memiliki pernyataan-pernyataan yang dapat dikelompokkan menjadi 5 bagian, yaitu DDL, DML, DCL, *Transactional*, dan pengendali programatik.

1. DDL (*Data Definition Language*)

DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut *database*, tabel, atribut (kolom), batasan-batasan terhadap suatu atribut serta hubungan antartabel. Yang termasuk kelompok DDL ini adalah:

- a. CREATE, untuk menciptakan table atau index.
- b. DROP, untuk menghapus table atau index.
- c. ALTER, untuk mengubah struktur table.

2. DML (*Data Manipulation Language*)

Adalah kelompok perintah yang berfungsi untuk memanipulasi data, misalnya untuk pengambilan, penyisipan pengubahan dan penghapusan data. Yang termasuk DML adalah:

- a. SELECT, untuk memilih data.
- b. UPDATE, untuk mengubah data.
- c. DELETE, untuk menghapus data.
- d. INSERT, untuk memasukkan data.

3. DCL (*Data Control Language*)

Berisi perintah-perintah untuk mengendalikan pengaksesan data. Yang termasuk DCL adalah:

- a. GRANT, untuk memberikan kendali pada pengaksesan data.
 - b. REVOKE, untuk mencabut hak kendali pengaksesan data.
-

- c. LOCK TABLES & UNLOCK TABLES, untuk mengunci dan membuka kunci tabel.

4. Transactional

Adalah perintah-perintah yang berfungsi untuk mengendalikan pengekseskuan transaksi. Yang termasuk kelompok ini adalah:

- a. COMMIT, untuk menyetujui rangkaian perintah yang berhubungan erat dan telah berhasil dilakukan.
- b. ROLLBACK, untuk membatalkan transaksi yang dilakukan karena adanya kesalahan pada salah satu rangkain perintah.

5. Pengendali Programatik

Mencakup pernyataan-pernyataan yang berhubungan dengan pemanfaatan SQL dalam bahasa lain (SQL yang dilekatkan). Yang termasuk dalam kelompok ini adalah:

- a. CLOSE untuk menutup kursor.
- b. DECLARE untuk mendeklarasikan kursor.
- c. FETCH untuk mengambil nilai baris berikutnya.
- d. OPEN untuk membuka kursor⁸¹.

2.11.1 Join Tabel

Join merupakan sebuah operasi yang digunakan untuk mendapatkan data gabungan dari dua buah tabel atau lebih. Operasi tersebut digunakan dalam perintah SELECT, dan biasanya dipakai untuk memperoleh detil data dari tabel-tabel yang saling terkait (memiliki relasi). Terdapat 3 jenis operasi join yaitu :

1. INNER

Merupakan jenis operator default dari parameter join, sehingga apabila parameter join tidak dilakukan oleh kata kunci lain akan dianggap sebagai inner join. operator inner hanya akan menghasilkan *record* yang saling berelasi dalam sebuah kumpulan table akibatnya jika terdapat *record* yang tidak memiliki referensi, relasi, *record* tersebut tidak akan ditampilkan di *query*.

2. OUTER

Operator outer dibagi menjadi :

- a. *left – outer*, untuk *record* yang berelasi dari table sebelah kiri
- b. *right – outer*, untuk *record* yang berelasi dari table sebelah kanan
- c. *full – outer*, untuk menampilkan seluruh *record*

3. CROSS

Cross akan membentuk kumpulan data yang sangat besar dengan mengkombinasikan semua kemungkinan relasi.

2.11.2 View

View merupakan tabel virtual dan merupakan snapshot (cuplikan) dari satu atau beberapa tabel. *View* digunakan agar data-data yang diperlihatkan dari satu atau beberapa tabel tersebut berdasarkan keperluannya saja.

View juga digunakan untuk melindungi agar tabel tidak secara langsung dapat diakses oleh pengguna dari penambahan, perubahan maupun penghapusan data.

Sehingga bisa dikatakan bahwa : “*View* hampir sama dengan tabel biasa, hanya saja ia tidak bisa di ubah, tambah dan hapus”^[8].

2.9 PHPMyAdmin

PHPMyAdmin adalah web aplikasi *client* untuk *database server* MySQL. PHPMyAdmin mendukung berbagai operasi MySQL, diantaranya: mengelola *database*, tabel, *fields*, *relations*, *users*, *permissions*, dan lain-lain. Ratschiller mengerjakan sebuah program bernama MySQL-Webadmin dengan bebas, yang merupakan produk dari Petrus Kuppelwieser, yang telah berhenti mengembangkannya pada saat itu. Ratschiller menulis kode baru untuk *phpmyadmin*, dan ditingkatkan pada konsep dari proyek Kuppelwieser. Ratschiller meninggalkan proyek *phpMyAdmin* pada tahun 2001. Sekarang, Sebuah tim dari delapan pengembang yang dipimpin oleh Oliver Muller meneruskan pengembangan *phpMyAdmin* di SourceForge.net. Beberapa pengembang diantaranya :

1. Marc Delisle
 2. Michal Cihar
 3. Sebastian Mendel
-

4. Herman van Rink

Berikut ini merupakan beberapa fitur-fitur dalam PHPMyAdmin:

1. Antarmuka berbasis web
2. Banyak Mendukung fitur MySQL:
 - a. Menelusuri dan menghapus *database*, *tables*, *views*, *fields* dan *indexes*
 - b. Membuat, menyalin, menghapus, merubah nama dan mengedit *databases*, *tables*, *fields*, dan *indexes*
 - c. Memelihara server, *databases*, *tables*, dengan server konfigurasi
 - d. Mengeksekusi, mengedit dan menandai pernyataan SQL, bahkan *batch-queries*
 - e. Mengelola *User* dan *Previlages*
 - f. Mengelola *Stored procedures* dan *triggers*
3. *Import* data dari CSV dan SQL
4. *Export* data ke berbagai format: CSV, SQL, XML, PDF, ISO / IEC 26300 – OpenDocument Text and Spreadsheet, Word, LaTeX dan lain-lain
5. Dapat Mengadministrasi beberapa server
6. Membuat grafik PDF dari tampilan *database*
7. Membuat kompleks *query* menggunakan *Query by example* (QBE)
8. Mendukung lebih dari 50 bahasa, termasuk bahasa indonesia

Tampilan antarmuka PHPMyAdmin dapat dilihat pada gambar 2.7^[2]:



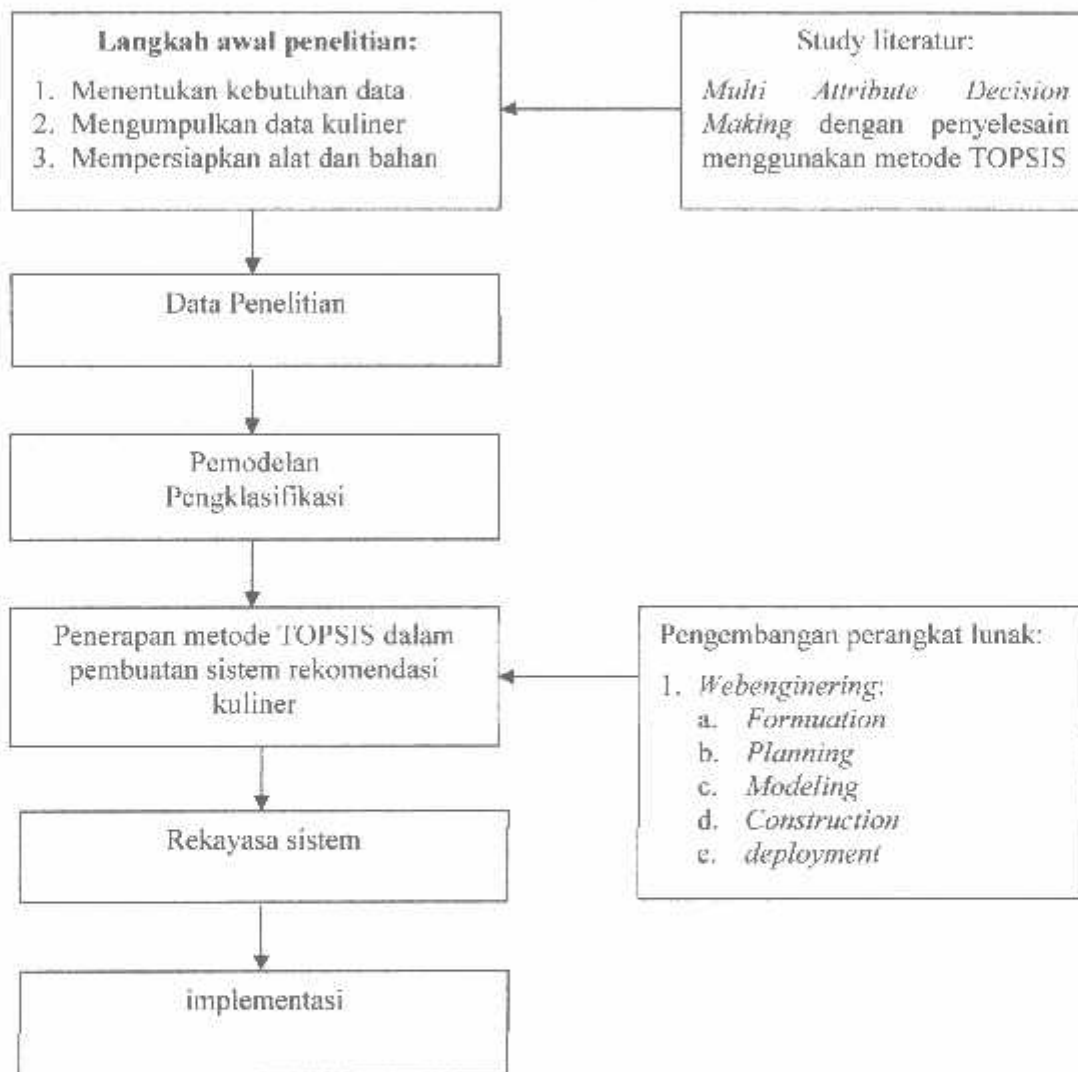
Gambar 2.3 Tampilan antarmuka PHPMyAdmin

BAB III
ANALISIS DAN PERANCANGAN

3.1 Metodologi Penelitian

3.1.1 Desain Penelitian

Gambar 3.1 merupakan desain penelitian yang akan digunakan pada sistem rekomendasi kuliner di Malang dengan menggunakan metode TOPSIS.



Gambar 3.1 Desain Penelitian

Tahapan atau gambaran yang akan dilakukan dalam melakukan penelitian dinamakan dengan desain penelitian, dibutuhkan desain penelitian untuk memudahkan peneliti dalam melakukan penelitian. Berikut adalah tahapan penelitian yang dilakukan oleh penulis :

1. Langkah Awal Penelitian

Perumusan masalah dimana perumusan masalah penelitian ini adalah sistem rekomendasi kuliner di Malang dengan menggunakan metode TOPSIS. Untuk memahami permasalahan, dilakukan studi literatur mengenai metode TOPSIS tersebut. Selanjutnya:

a. Menentukan kebutuhan data yang digunakan

Data yang di maksud adalah data kuliner di Malang , yaitu tempat, alamat, peta, menu makanan, dan harga.

b. Mengumpulkan data yang dibutuhkan, data yang sudah ditentukan di atas kemudian dikumpulkan untuk diproses.

c. Mempersiapkan alat dan bahan penelitian

Yang dimaksud alat disini adalah alat yang digunakan untuk membuat sistem rekomendasi kuliner sedangkan yang dimaksud dengan bahan adalah data yang dikumpulkan untuk selanjutnya diolah dan dijadikan program. Alat dan bahan ini akan dibahas pada Poin berikutnya.

2. Setelah diperoleh data penelitian kemudian data tersebut diproses melalui pengembangan perangkat lunak, dengan menggunakan model proses *Web Engineering Process*.

3.1.2 Fokus Penelitian

Penelitian ini memfokuskan pada tingkat akurasi prediksi yang dihasilkan oleh sistem, dimana proses pembuatan prediksi dilakukan dengan menghitung nilai jarak terpendek dan terpanjang dari solusi ideal negatif dan positif (persamaan 2.9 dan persamaan 2.10).

Untuk menentukan alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif, namun juga memiliki jarak terpanjang dari solusi ideal negatif (persamaan 2.11).

3.1.3 Alat dan Bahan Penelitian

1. Alat penelitian

Penelitian ini menggunakan seperangkat sistem komputer beserta perangkat lunak pendukungnya, yaitu :

a. Satu unit komputer dengan spesifikasi minimum:

i. *Processor* Intel Pentium 4 2.0 Ghz

- ii. RAM 1 GB
 - iii. *Harddisk* 30 GB
 - iv. Monitor dengan resolusi 1024 x 800 pixel, 24 bit *color*
 - v. Mouse + keyboard
- b. Software:
- i. *Sistem Operasi* windows XP SP 2
 - ii. Appserv 2.5.10
 - iii. *Web browser* google chrome
 - iv. *Framework* codeigniter
 - v. Editor sublime text 2 version 2.0.1

2. Bahan penelitian

Data kuliner berasal dari situs halomalang.com. Lalu data di rating sendiri oleh admin pada sistem, dengan ketentuan tertentu. Dalam merating/memberi nilai kecocokan setiap atribut dan tingkat kepentingan setiap kriteria dinilai dengan 1 sampai 5, dapat dilihat pada tabel 3.1.

Tabel 3.1 Tabel nilai kecocokan dan tingkat kepentingan^[4]

Nilai	Atribut	Kriteria
1	Sangat buruk	Sangat rendah
2	Buruk	Rendah
3	Cukup	Cukup
4	Baik	Tinggi
5	Sangat baik	Sangat tinggi

Untuk kriteria yang digunakan dalam penelitian ini, didapat dari hasil analisa angket yang diberikan ke 20 orang. Untuk detailnya dapat dilihat pada tabel 3.2:

Tabel 3.2 Tabel analisa kriteria dalam penelitian

Kriteria	Bobot	Status kriteria
Rasa	4	Keuntungan
Harga	4	Biaya
Pelayanan	3	Keuntungan
Suasana	3	Keuntungan
Keamanan	2	Keuntungan
Kebersihan	4	Keuntungan

3.1.4 Tahap Penelitian

Dalam pelaksanaannya, tahapan yang dilaksanakan dalam penelitian ini adalah sebagai berikut :

1. Langkah awal penelitian

Pernyataan masalah penelitian ini adalah sistem rekomendasi kuliner di Malang dengan metode *TOPSIS*. Untuk memahami permasalahan, dilakukan studi literatur mengenai metode *TOPSIS* dan penerapan metode tersebut.

Selanjutnya mengumpulkan data kuliner di Malang dan menentukan kriteria apa saja yang akan dijadikan acuan dalam pengambilan keputusan untuk digunakan dalam sistem, serta menyiapkan perangkat penelitian.

2. Pemberian nilai kecocokan/kepentingan

Pada Tahap ini data alternatif dan kriteria yang sudah ada diberi nilai kecocokan dan nilai kepentingan menurut ketentuan pada tabel 3.1.

3. Membuat matrik keputusan yang ternormalisasi

Tahap ini merupakan tahap pembuatan matrik ternormalisasi dari data alternatif dan kriteria yang sudah diberi nilai kecocokan dan nilai kepentingan dengan menggunakan persamaan 2.3

4. Membuat matrik keputusan yang ternormalisasi terbobot

Pada Tahap ini, matrik keputusan yang dihasilkan dari langkah sebelumnya dikalikan dengan bobot preferensi yang ditentukan oleh *user* dengan menggunakan persamaan 2.4.

5. Menentukan matrik solusi ideal positif dan solusi ideal negatif

Tahap ini merupakan tahap menentukan solusi ideal positif dan solusi ideal negatif dari matrik ternormalisasi terbobot dengan menggunakan persamaan 2.6 dan 2.8

6. Menentukan jarak antara nilai setiap alternatif dengan matrik solusi ideal positif dan solusi ideal negatif

Tahap ini merupakan tahap menentukan jarak dari setiap alternatif dengan matrik solusi ideal positif dan solusi ideal negatif dengan menggunakan persamaan 2.9 dan 2.10

7. Pembuatan rating alternatif

Tahap ini selanjutnya menentukan nilai preferensi dari setiap alternatif yang ada dengan menggunakan persamaan 2.11. Selanjutnya alternatif akan diurutkan secara *descending*.

keputusan. Selanjutnya matrik keputusan yang terbentuk dinormalisasi dengan menggunakan persamaan 2.3.

Langkah selanjutnya, mengalikan matrik keputusan yang ternormalisasi dengan bobot preferensi / bobot kriteria yang ada. Setelah itu, menentukan matrik solusi ideal positif dan matrik solusi ideal negatif dengan menggunakan persamaan 2.6 dan persamaan 2.8. Kemudian menentukan jarak dari setiap alternatif dengan matrik solusi ideal positif dan matrik solusi ideal negatif dengan menggunakan persamaan 2.9 dan persamaan 2.10.

Terakhir, menentukan nilai preferensi dari setiap alternatif yang ada dengan menggunakan persamaan 2.11.

3.2.2 Contoh Perhitungan

Untuk memahami lebih jelas tentang *flowchart* metode TOPSIS seperti pada gambar 3.2, dapat dilihat berikut ini tentang perhitungan secara real tahap-tahap metode TOPSIS dari data sampel pada *database*.

Tabel 3.3 Tabel bobot kriteria pada setiap alternatif

Alternatif	Kriteria					
	Rasa	Harga	Pelayanan	Suasana	Keamanan	Kebersihan
Mie iblis	3	3	3	3	3	4
Mie setan	4	3	3	2	3	3
Mie galau	3	3	2	2	3	3

Dari tabel 3.3, dapat dibentuk matrik keputusan X sebagai berikut:

$$X = \begin{bmatrix} 3 & 3 & 3 & 3 & 3 & 4 \\ 4 & 3 & 3 & 2 & 3 & 3 \\ 3 & 3 & 2 & 2 & 3 & 3 \end{bmatrix}$$

Dimana *user* memberikan bobot preferensi (tingkat kepentingan kriteria) sebagai berikut:

$$W = (4 \ 4 \ 3 \ 3 \ 2 \ 3)$$

1. Normalisasi matrik keputusan

Untuk membuat matrik ternormalisasi dengan menggunakan persamaan 2.3. berikut contoh perhitungannya:

$$|X_1| = \sqrt{3^2 + 4^2 + 3^2} = 5,8309 \quad \Rightarrow \quad r_{11} = \frac{3}{5,8309} = 0,5145$$

$$r_{21} = \frac{4}{5,8309} = 0,6860$$

$$r_{31} = \frac{3}{5,8309} = 0,5145$$

Dan seterusnya hingga didapat matrik ternormalisasi R:

$$R = \begin{bmatrix} 0,5145 & 0,5773 & 0,6396 & 0,7276 & 0,5773 & 0,6860 \\ 0,6860 & 0,5773 & 0,6396 & 0,4850 & 0,5773 & 0,5145 \\ 0,5145 & 0,5773 & 0,4264 & 0,4850 & 0,5773 & 0,5145 \end{bmatrix}$$

2. Membuat matrik ternormalisasi terbobot

Matrik ternormalisasi terbobot didapat dengan cara mengalikan bobot preferensi / bobot kriteria dengan matrik ternormalisasi. Berikut contoh perhitungannya:

$$\begin{aligned} y_{11} &= 4 * 0,5145 = 2,0580 & y_{21} &= 4 * 0,6860 = 2,7440 \\ y_{12} &= 4 * 0,5773 = 2,3092 & y_{22} &= 4 * 0,5773 = 2,3092 \\ y_{13} &= 3 * 0,6396 = 1,9188 & y_{23} &= 3 * 0,6396 = 1,9188 \\ y_{14} &= 3 * 0,7276 = 2,1828 & y_{24} &= 3 * 0,4850 = 1,4550 \\ y_{15} &= 2 * 0,5773 = 1,1546 & y_{25} &= 2 * 0,5773 = 1,1546 \\ y_{16} &= 3 * 0,6860 = 2,0580 & y_{26} &= 3 * 0,5145 = 1,5435 \end{aligned}$$

Dan seterusnya sehingga didapat matrik ternormalisasi terbobot Y:

$$Y = \begin{bmatrix} 2,0580 & 2,3092 & 1,9188 & 2,1828 & 1,1546 & 2,0580 \\ 2,7440 & 2,3092 & 1,9188 & 1,4550 & 1,1546 & 1,5435 \\ 2,0580 & 2,3092 & 1,2792 & 1,4550 & 1,1546 & 1,5435 \end{bmatrix}$$

3. Menentukan matrik solusi ideal positif dan negatif

Untuk menentukan matrik solusi ideal positif dan negatif digunakan persamaan 2.6 dan 2.8. Berikut contoh perhitungannya:

Sebagai contoh untuk menentukan matrik solusi ideal positif kriteria rasa, kriteria rasa berdasarkan analisa merupakan kriteria keuntungan maka:

$$y_1^+ = \max(2,0580; 2,7440; 2,0580) = 2,7440$$

Sedangkan untuk kriteria biaya contohnya kriteria harga, maka:

$$y_2^+ = \min(2,3092; 2,3092; 2,3092) = 2,3092$$

Dan seterusnya sehingga didapat matrik A⁺:

$$A^+ = \{2,7440; 2,3092; 1,9188; 2,1828; 1,1546; 2,0580\}$$

Sedangkan untuk menentukan matrik solusi ideal negative kriteria rasa, maka:

$$y_1^- = \min(2,0580; 2,7440; 2,0580) = 2,0580$$

Untuk kriteria harga, maka:

$$y_2^- = \max(2,3092; 2,3092; 2,3092) = 2,3092$$

Dan seterusnya sehingga didapat matrik A⁻:

$$A^- = \{2,0580; 2,3092; 1,2792; 1,4550; 1,1546; 2,0580\}$$

4. Menentukan jarak setiap alternatif dengan matrik solusi ideal

Untuk menentukan jarak setiap alternatif dengan matrik solusi ideal positif dan negatif menggunakan persamaan 2.9 dan 2.10. Berikut contoh perhitungannya:

Untuk jarak alternatif dengan matrik solusi ideal positif adalah:

$$D_1^+ = \sqrt{\frac{(2,7440 - 2,0580)^2 + (2,3092 - 2,3092)^2 + (1,9188 - 1,9188)^2 + (2,1828 - 2,1828)^2 + (1,1546 - 1,1546)^2 + (2,0580 - 2,0580)^2}{6}} = 0,4705$$

Dan seterusnya sehingga didapat matrik $D^+ = \{0,4705; 0,6588; 1,5409\}$

Sedangkan untuk jarak alternatif dengan matrik solusi ideal negatif:

$$D_1^- = \sqrt{\frac{(2,0580 - 2,0580)^2 + (2,3092 - 2,3092)^2 + (1,9188 - 1,2792)^2 + (2,1828 - 1,4550)^2 + (1,1546 - 1,1546)^2 + (2,0580 - 1,5435)^2}{6}} = 1,2033$$

Dan seterusnya sehingga didapat matrik $D^- = \{1,2033; 0,8795; 0\}$

5. Menentukan nilai preferensi setiap alternatif

Nilai preferensi setiap alternatif didapat dengan cara persamaan 2.11. Berikut contoh perhitungannya:

$$V_1 = \frac{1,2033}{1,2033 + 0,4705} = 0,7189 \qquad V_2 = \frac{0,8795}{0,8795 + 0,6588} = 0,5717$$

$$V_3 = \frac{0}{0 + 1,5409} = 0$$

Nilai V_i yang lebih besar menunjukkan bahwa alternatif A_i lebih dipilih. Sehingga dari nilai V diatas, dapat dilihat bahwa V_1 (mie iblis) memiliki nilai terbesar sehingga dapat disimpulkan bahwa alternatif pertama yang akan lebih dipilih.

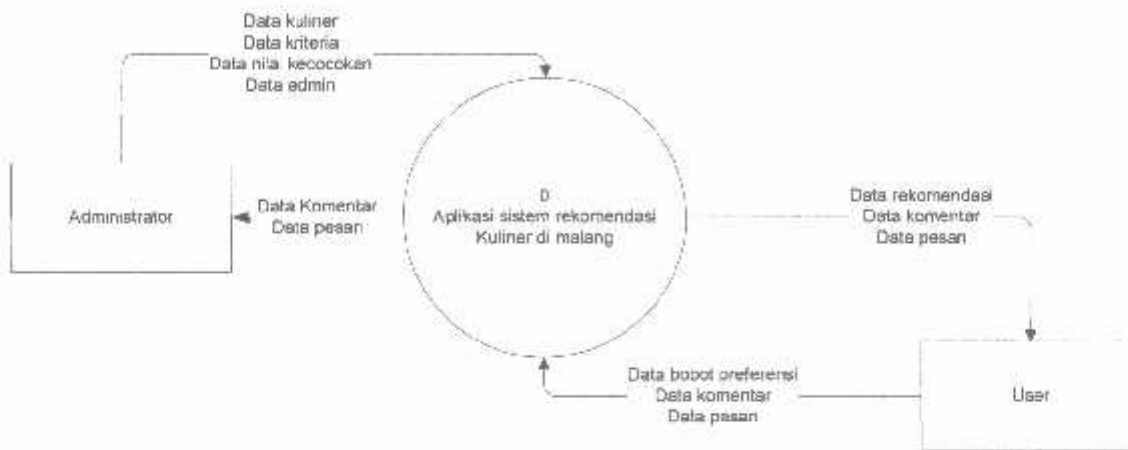
3.3 Hak Akses

Sistem pada aplikasi ini hanya memiliki satu hak akses saja yaitu *Administrator* pada halaman administrasi. Adapun hal-hal yang dapat dilakukan *Administrtor* pada sistem ini adalah sebagai berikut:

1. Menambah, meng-edit atau menghapus data kuliner yang tersimpan dalam sistem basis data
2. Menambah, meng-edit atau menghapus data *user* yang dapat hak ases sistem
3. Menambah, meng-edit atau menghapus data atau ketentuan-ketentuan yang digunakan dalam perhitungan metode TOPSIS

3.4 Diagram Konteks

Diagram konteks adalah rancangan diagram alur paling awal (atau diagram alur secara umum/garis besar) dalam pembuatan suatu aplikasi maupun sistem informasi. Ciri-ciri dari *diagram* ini adalah terdapat satu proses utama yang menggambarkan satu aplikasi yang dikembangkan dan memiliki nomor proses 0. Diagram konteks pada aplikasi ini dapat dilihat dalam gambar 3.3.



Gambar 3.3 Diagram Konteks

Dari diagram konteks dalam gambar 3.3, dapat dijabarkan penjelasan lebih detail mengenai entitas, proses dan data yang digunakan sebagai berikut:

1. Entitas

Merupakan pelaku-pelaku yang berperan dalam kegiatan atau aktifitas yang dilakukan oleh sistem/aplikasi. Dalam aplikasi ini terdapat 2 entitas, yaitu:

- a. Administrator.
- b. User.

2. Proses

Merupakan kegiatan-kegiatan yang dilakukan dalam sebuah sistem. Namun pada diagram konteks biasanya satu proses utama dinamakan dengan nama aplikasi atau sistem tersebut. Proses utama dalam aplikasi ini adalah:

- a. Aplikasi sistem rekomendasi kuliner di Malang.

3. Data

Merupakan sebuah informasi yang belum diolah atau informasi mentah. Dalam aplikasi ini terdapat beberapa data yang digunakan, yaitu:

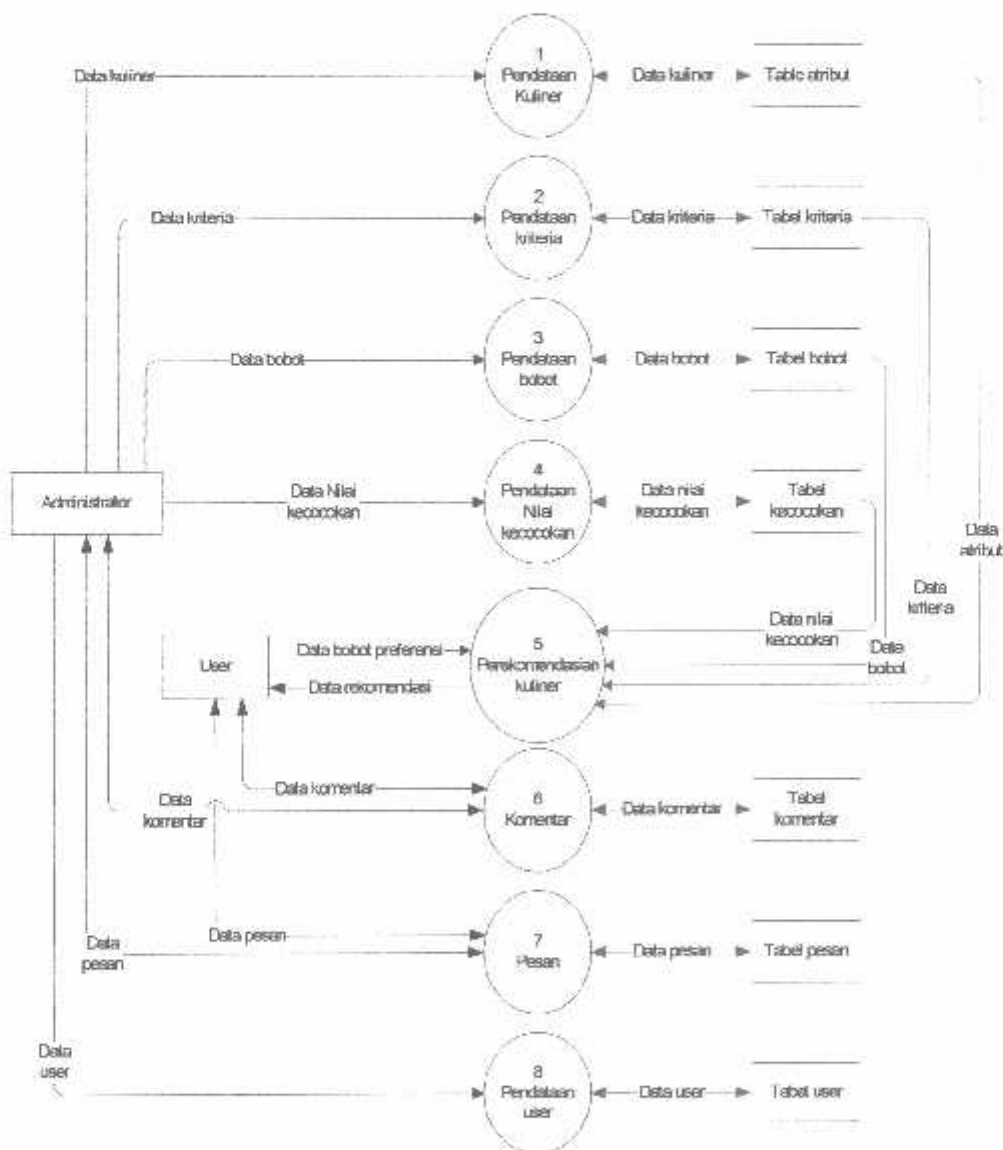
- a. Data kuliner

- b. Data kriteria
- c. Data nilai kecocokan
- d. Data admin
- e. Data komentar
- f. Data pesan
- g. Data rekomendasi

3.5 Data Flow Diagram (DFD)

1. Level 1

DFD level 1 merupakan penjabaran yang lebih detail dari diagram konteks suatu sistem. DFD level 1 pada sistem ini dapat dilihat pada gambar 3.4.



Gambar 3.4 DFD level 1 aplikasi sistem rekomendasi kuliner di Malang

Dalam gambar 3.4, dapat dijabarkan *input* dan *output* data dari masing-masing entitas dan proses sebagai berikut:

a. Administrator

Input: Data komentr, Data pesan

Output: Data kuliner, Data kriteria, Data bobot, Data nilai kecocokan, Data komentar, Data pesan, Data *user*

b. *User*

Input: Data komentar, Data pesan, Data rekomendasi

Output: Data bobot preferensi, Data komentar, Data pesan

c. Proses pendataan kuliner

Input: Data kuliner

Output: Data kuliner

d. Proses pendataan kriteria

Input: Data kriteria

Output: Data kriteria

e. Proses pendataan bobot

Input: Data bobot

Output: Data bobot

f. Proses pendataan nilai kecocokan

Input: Data nilai kecocokan

Output: Data nilai kecocokan

g. Proses perekomendasiian kuliner

Input: Data bobot preferensi, Data atribut, Data kriteria, Data bobot, Data nilai kecocokan

Output: Data rekomendasi

h. Proses komentar

Input: Data komentar

Output: Data komentar

i. Proses pesan

Input: Data pesan

Output: Data pesan

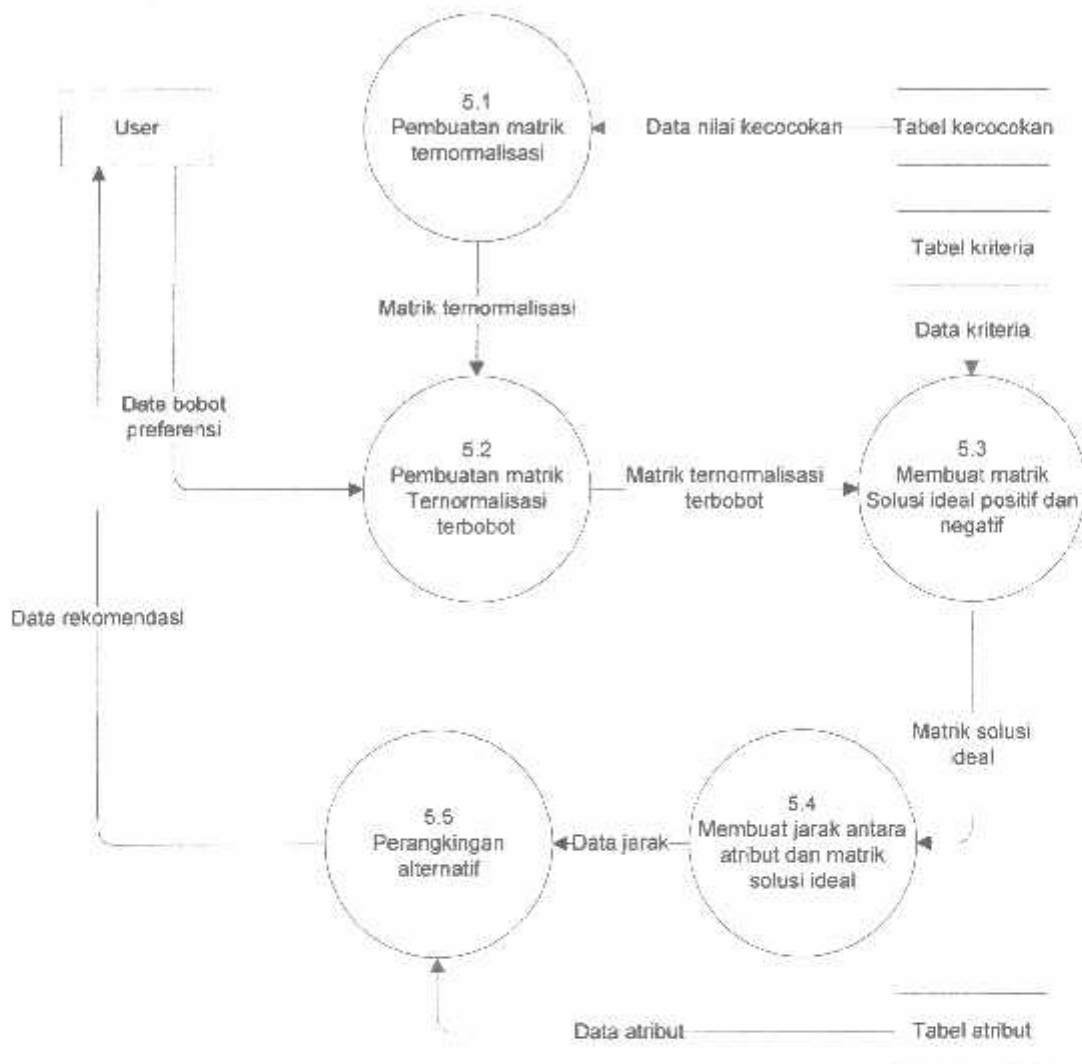
j. Proses pendataan *user*

Input: Data *user*

Output: Data *user*

2. Level 2

DFD level 2 merupakan penjabaran lebih detail dari suatu proses yang terdapat pada DFD level 1. DFD level 2 sub proses rekomendasi kuliner ditunjukkan dalam gambar 3.5.



Gambar 3.5 DFD level 2 sub proses rekomendasi kuliner

Dalam gambar 3.5, dapat dijabarkan *input* dan *output* data dari masing-masing entitas dan proses sebagai berikut:

a. User

Input: Data rekomendasi

Output: Data bobot preferensi

b. Pembuatan matrik ternormalisasi

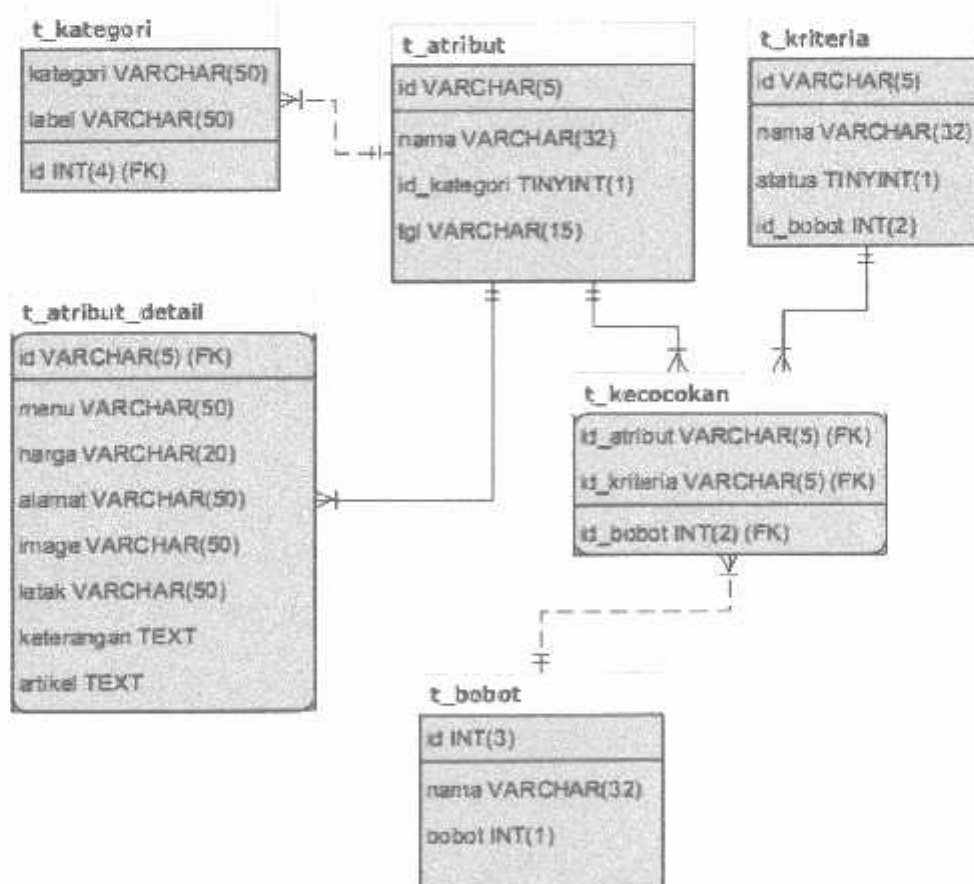
Input: Data nilai kecocokan

Output: Data matrik ternormalisasi

- c. Pembuatan matrik ternormalisasi terbobot
 - Input: Data matrik ternormalisasi
 - Output: Data matrik ternormalisasi terbobot
- d. Membuat matrik solusi ideal positif dan negatif
 - Input: Data kriteria, Data matrik ternormalisasi terbobot
 - Output: Data matrik solusi ideal
- e. Membuat jarak antara atribut dan matrik solusi ideal
 - Input: Data matrik solusi ideal
 - Output: Data jarak
- f. Perangkingan alternatif
 - Input: Data atribut, Data jarak
 - Output: Data rekomendasi

3.6 Entity Relationship Diagram (ERD)

Diagram relasi antar tabel pada sistem ini dapat digambarkan seperti gambar 3.6.



Gambar 3.6 Diagram relasi antar tabel

3.7 Pengembangan Perangkat Lunak

3.7.1 Formulation

1. Deskripsi umum

Sistem rekomendasi kuliner ini merupakan suatu aplikasi berbasis *web* (WebApp) yang memungkinkan pengguna untuk mendapatkan rekomendasi kuliner yang mungkin akan disukainya.

Secara umum tujuan dari sistem ini adalah memberikan informasi detail kepada pengguna / *user* tentang suatu kuliner, baik itu tempat, harga rasa, dan sebagainya.

2. Batasan aplikasi

Berikut ini adalah batasan dari sistem yang akan dikembangkan adalah:

- a. Karena sistem ini rencananya akan dihosting di jaringan internet, maka pengguna mutlak harus memiliki akses ke jaringan internet untuk mengakses sistem ini.
- b. Dalam sekali melakukan rekomendasi, sistem maksimal akan memberikan sebanyak 20 kuliner yang direkomendasikan.

3.7.2 Planning

Perencanaan pengembangan aplikasi ini dapat dilihat pada tabel 3.4.

Tabel 3.4 Perencanaan pengembangan aplikasi

No.	Versi	Keterangan
1	Versi 1.0	Aplikasi ini belum mempunyai <i>interface</i> khusus jika diakses lewat <i>mobile device</i>
2	Versi 2.0	Aplikasi ini sudah memiliki <i>interface</i> khusus jika diakses lewat <i>mobile device</i>

3.7.3 Modelling

Spesifikasi kebutuhan dari aplikasi sistem rekomendasi kuliner ini terbagi menjadi 2, yaitu kebutuhan fungsional dan kebutuhan non fungsional.

1. Kebutuhan fungsional

Kebutuhan fungsional pada aplikasi dapat dilihat pada tabel 3.5 berikut:

Tabel 3.5 Daftar kebutuhan fungsional

No.	Deskripsi
1	Sistem memberikan daftar rekomendasi kuliner yang mungkin disukai oleh <i>user</i>
2	Sistem menyediakan fasilitas bagi <i>user</i> untuk memberikan daftar rekomendasi kuliner berdasarkan kata kunci atau kriteria tertentu dan berdasarkan tingkat kepentingan yang disukai <i>user</i> sendiri-sendiri
3	Sistem menyediakan fasilitas bagi <i>user</i> untuk melihat informasi kuliner
4	Sistem menyediakan fasilitas bagi <i>user</i> untuk memberi komentar tentang kuliner
5	Sistem menyediakan fasilitas bagi <i>user</i> untuk mengirim pesan untuk admin

2. Kebutuhan non fungsional

Kebutuhan non fungsional pada aplikasi ini dapat dilihat pada tabel 3.6 berikut:

Tabel 3.6 Daftar kebutuhan non fungsional

No.	Aspek	Deskripsi
1	<i>Availability</i>	Sistem tersedia untuk pengguna selama 24 jam dalam sehari, 7 hari dalam seminggu, dan 365 hari dalam setahun dengan toleransi 5%
2	<i>Response Time</i>	Sistem memberikan respon terhadap aksi pengguna kurang dari 5 detik
3	<i>Scalability</i>	Sistem dapat diakses oleh multiple pengguna secara bersamaan
4	<i>Security</i>	Sistem mampu menolak dan mencegah segala akses ke sistem yang dilakukan secara tidak sah
5	<i>Usability</i>	Sistem mempunyai antar muka yang mudah dipahami sehingga mudah digunakan
6	<i>Portability</i>	Sistem dapat di- <i>install</i> pada web server dengan versi php > 5 dan mysql > 5.0.45 dan juga sistem mampu dioperasikan pada platform sistem operasi apapun dengan menggunakan browser versi terbaru

3. Kebutuhan informasi

Data kebutuhan informasi pada aplikasi ini dapat dilihat pada tabel 3.7 berikut ini:

Tabel 3.7 Daftar kebutuhan informasi

No.	Informasi yang dibutuhkan	Tujuan	Frekwensi	Format
1	Data informasi kuliner	Untuk mengisi data pada bagian detail informasi	Digunakan pada saat pertama <i>insert</i> data	Text
2	Data nilai kecocokan	Digunakan pada proses rekomendasi	Setiap membuat rekomendasi	Angka 1-5 dengan iterasi 1
3	Data nilai tingkat kepentingan	Digunakan pada proses rekomendasi	Setiap membuat rekomendasi	Angka 1-5 dengan iterasi 1

4. Desain arsitektur aplikasi

Dalam pengembangan aplikasi ini, digunakan *design pattern* MVC (*Model-View-Controller*) dimana dengan menggunakan *design pattern* ini aplikasi dipisahkan menjadi beberapa bagian utama, yaitu :

a. Model

Berisi seluruh logic pemrosesan objek data yang digunakan pada aplikasi (CRUD) serta akses terhadap eksternal data atau sumber informasi lainnya.

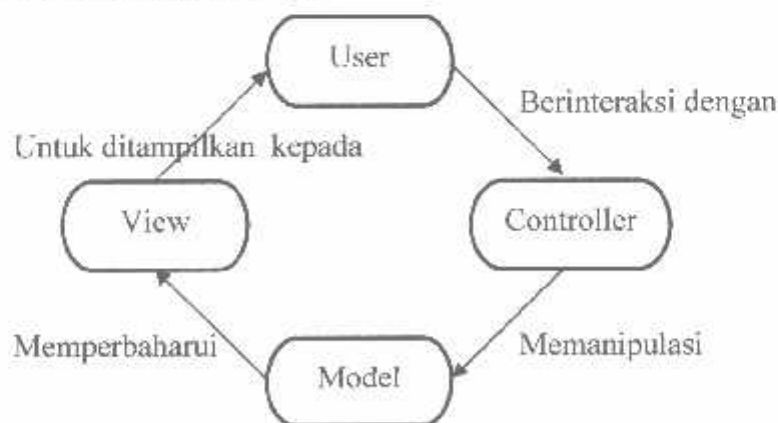
b. View

Tampilan yang dipanggil oleh controller untuk disajikan terhadap pengguna.

c. Controller

Bagian ini menjembatani dan mengatur akses antara model dan view serta memproses HTTP request dari setiap pengguna.

Ilustrasi keterhubungan antara bagian-bagian yang telah dijelaskan diatas dapat dilihat pada gambar 3.7



Gambar 3.7 Diagram keterhubungan antara M-V-C

5. Desain database

Struktur fisik data dalam database yang digunakan pada sistem ini ada 13. Terdiri dari 11 tabel dan 3 view, yaitu:

a. Tabel admin

Tabel admin berfungsi untuk menyimpan data akun-akun administrator. Struktur tabel admin dapat ditunjukkan dalam tabel 3.8.

Tabel 3.8 Tabel admin

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>id</u> (PK)	Int(2)	Not Null, Autoincrement
user	Varchar(32)	Not Null
pass	Varchar(32)	Not Null
email	Varchar(32)	Not Null
last_login	Varchar(32)	Null

b. Tabel atribut

Tabel ini menyimpan data umum dari tempat2 kuliner yang ada di Malang. Struktur tabel atribut dapat ditunjukkan dalam tabel 3.9.

Tabel 3.9 Tabel atribut

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>id</u> (PK)	Varchar(5)	Not Null
Nama	Varchar(32)	Not Null
Id_kategori	Tinyint(1)	Not Null, Default 2
tgl	Varchar(15)	Not Null

c. Tabel atribut detail

Tabel ini menyimpan detail data dari tempat kuliner yang ada pada tabel atribut. Struktur tabel atribut detail dapat ditunjukkan dalam tabel 3.10.

Tabel 3.10 Tabel atribut detail

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>id</u> (PK)	Varchar(5)	Not Null
Menu	Varchar(50)	Not Null
Harga	Varchar(20)	Not Null
Alamat	Varchar(50)	Not Null

Image	Varchar(50)	Not Null
Letak	Varchar(50)	Not Null
Keterangan	Text	Not Null
Artikel	Text	Null

d. Tabel kategori

Tabel ini menyimpan data jenis-jenis kuliner yang ada. Struktur tabel kategori dapat ditunjukkan dalam tabel 3.11.

Tabel 3.11 Tabel kategori

Kolom	Tipe Data(Panjang Data)	Deskripsi
id	Int(4)	Not Null
<u>Kategori(PK)</u>	Varchar(50)	Not Null
<u>Label(PK)</u>	Varchar(50)	Not Null

e. Tabel bobot

Tabel ini menyimpan data bobot/tingkat kepentingan yang digunakan dalam penghitungan metode TOPSIS. Struktur tabel bobot dapat ditunjukkan dalam tabel 3.12.

Tabel 3.12 Tabel bobot

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>Id(PK)</u>	Int(3)	Not Null, Autoincrement
Nama	Varchar(32)	Not Null
Bobot	Int(1)	Not Null

f. Tabel kecocokan

Tabel ini menyimpan data bobot setiap kriteria dari setiap atribut yang ada, dimana nantinya akan digunakan dalam penghitungan metode TOPSIS dalam mencari atribut yang akan direkomendasikan. Struktur tabel kecocokan dapat ditunjukkan dalam tabel 3.13.

Tabel 3.13 Tabel kecocokan

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>Id_atribut(PK)</u>	Varchar(5)	Not Null
<u>Id_kriteria(PK)</u>	Varchar(5)	Not Null
Id_bobot	Int(2)	Not Null

g. Tabel kriteria

Tabel ini menyimpan data kriteria yang dijadikan acuan dalam pengambilan keputusan pada sistem ini. Struktur tabel kriteria dapat ditunjukkan dalam tabel 3.14.

Tabel 3.14 Tabel kriteria

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>Id (PK)</u>	Varchar(5)	Not Null
Nama	Varchar(32)	Not Null
status	tinyint(1)	Not Null
Id_bobot	Int(2)	Not Null

h. Tabel komentar

Tabel ini berisi komentar-komentar dari *user* tentang sistem maupun tentang tempat kuliner yang direkomendasikan. Struktur tabel komentar dapat ditunjukkan dalam tabel 3.15.

Tabel 3.15 Tabel komentar

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>Id (PK)</u>	Varchar(10)	Not Null, Autoincrement
Nama	Varchar(50)	Not Null
Email	Varchar(50)	Not Null
url	Varchar(50)	Null
Komentar	Text	Not Null
tgl	Varchar(15)	Not Null

i. Tabel menu

Tabel ini berisi menu-menu yang akan digunakan dalam halaman administrasi untuk admin. Struktur tabel menu dapat ditunjukkan dalam tabel 3.16.

Tabel 3.16 Tabel menu

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>Id (PK)</u>	Int(2)	Not Null
Class	Varchar(50)	Null
Icon	Varchar(50)	Not Null

menu	Varchar(50)	Not Null
Link	Text	Not Null
Submenu	Text	Not Null
Link_submenu	Text	Not Null

j. Tabel pesan

Tabel ini menyimpan data pesan dari *user*/pengguna khusus untuk *admin*. Struktur tabel pesan dapat ditunjukkan dalam tabel 3.17.

Tabel 3.17 Tabel pesan

Kolom	Tipe Data(Panjang Data)	Deskripsi
<u>Id (PK)</u>	Varchar(5)	Not Null, Autoincrement
Nama	Varchar(50)	Not Null
Email	Varchar(50)	Not Null
Website	Varchar(50)	Not Null
Pesan	Text	Not Null
Tgl	Varchar(15)	Not Null
Status	Tinyint(1)	Not Null, Default 1
Id_reply	Varchar(5)	Null

k. View atribut

View ini digunakan untuk melihat data atribut yang merupakan hasil join dari tabel atribut, tabel atribut detail dan tabel kategori. Struktur view atribut dapat ditunjukkan dalam tabel 3.18.

Tabel 3.18 View atribut

Kolom	Tipe Data(Panjang Data)	Deskripsi
id	Varchar(5)	Not Null
Nama	Varchar(32)	Not Null
Menu	Varchar(50)	Not Null
Harga	Varchar(20)	Not Null
Alamat	Varchar(50)	Not Null
Image	Varchar(50)	Not Null
Letak	Varchar(50)	Not Null
Keterangan	Text	Not Null

Label	Varchar(50)	Not Null
tgl	Varchar(15)	Not Null

l. View kecocokan

View ini digunakan untuk melihat nilai kecocokan yang diberikan pada setiap atribut disetiap kriteria yang ada. View ini merupakan hasil join dari tabel kecocokan, tabel bobot, tabel atribut, dan tabel kriteria. Struktur view kecocokan dapat ditunjukkan dalam tabel 3.19.

Tabel 3.19 View kecocokan

Kolom	Tipe Data(Panjang Data)	Deskripsi
Id_atribut	Varchar(5)	Not Null
Id_kriteria	Varchar(5)	Not Null
Atribut	Varchar(32)	Not Null
Kriteria	Varchar(32)	Not Null
Bobot	Int(1)	Not Null
Status_kriteria	Tinyint(1)	Not Null

m. View kriteria

View ini digunakan untuk melihat bobot preferensi/tingkat kepentingan yang digunakan untuk acuan dalam pengambilan keputusan dengan metode TOPSIS. Struktur view kriteria dapat ditunjukkan dalam tabel 3.20.

Tabel 3.20 View kriteria

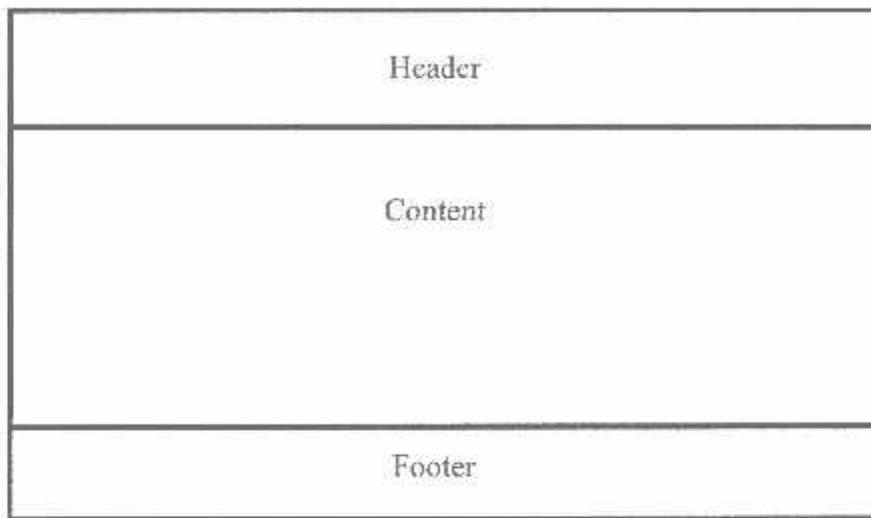
Kolom	Tipe Data(Panjang Data)	Deskripsi
id	Varchar(5)	Not Null
Nama	Varchar(32)	Not Null
Status	Tinyint(1)	Not Null
Nama_bobot	Varchar(32)	Not Null
Bobot	Int(1)	Not Null

6. Desain antarmuka aplikasi

Desain Layout dalam aplikasi ini terbagi menjadi 2 bagian, yaitu: *layout* untuk halaman *user* dan *layout* untuk halaman administrasi.

1. *Layout* halaman *home* untuk *user*

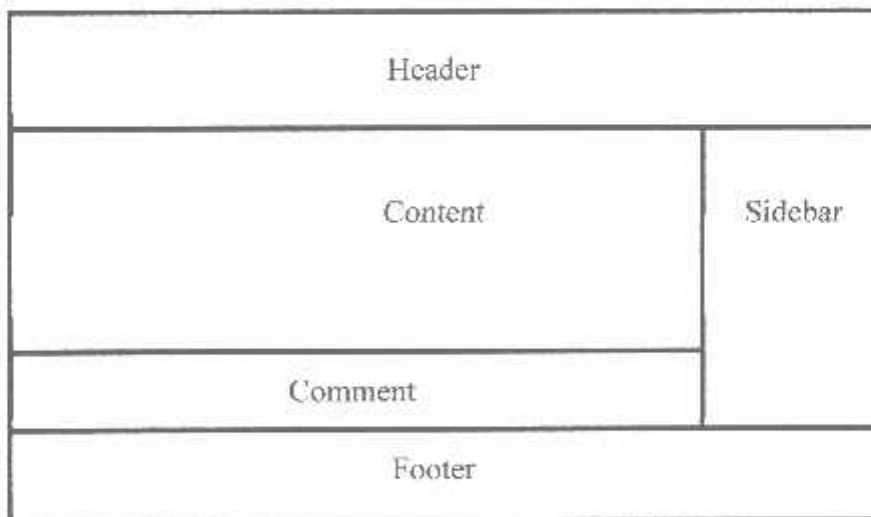
Halaman *home* merupakan halaman utama yang digunakan untuk berinteraksi secara langsung dengan pengguna. Desain halaman *home* dapat dilihat dalam gambar 3.8.



Gambar 3.8 *Layout* halaman *home*

2. *Layout* halaman rekomendasi untuk *user*

Halaman rekomendasi merupakan halaman yang digunakan pengguna untuk mendapatkan rekomendasi kuliner. Desain halaman rekomendasi dapat dilihat dalam gambar 3.9.



Gambar 3.9 *Layout* halaman rekomendasi

3. *Layout* halaman login administrator

Merupakan halaman login untuk dapat masuk ke dalam halaman administrasi. *Input* yang dibutuhkan adalah *username* dan *password*. Desain halaman login dapat dilihat dalam gambar 3.10.

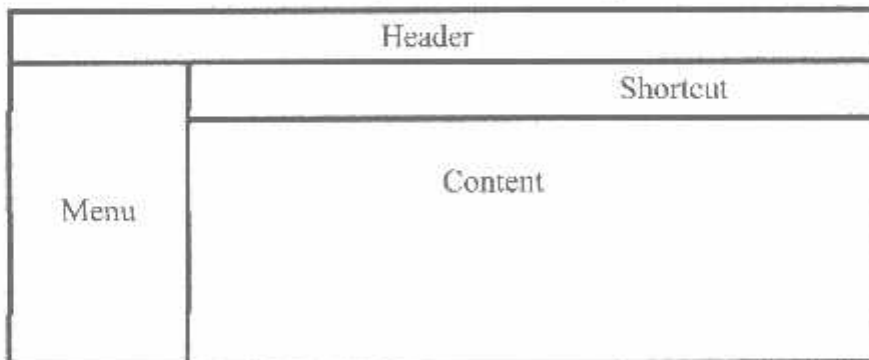
Gambar 3.10 *Layout* halaman login

4. *Layout* halaman administrasi

Halaman administrasi ini terbagi menjadi 4 section, yaitu *header*, menu, *shortcut* dan halaman isi. Berikut penjabaran lebih detailnya:

- i. *Header*, yang berisi akun *user* dan logout.
- ii. Menu, yang berisi menu utama dari halaman administrasi, yaitu dashboard, content, topsis, gallery dan setting
- iii. *Shortcut*, yang berisi *shortcut* dari menu: *Home*, pesan dan komentar
- iv. Halaman isi, halaman ini merupakan halaman untuk menampilkan isi dari masing-masing menu, baik menu utama maupun *shortcut*.

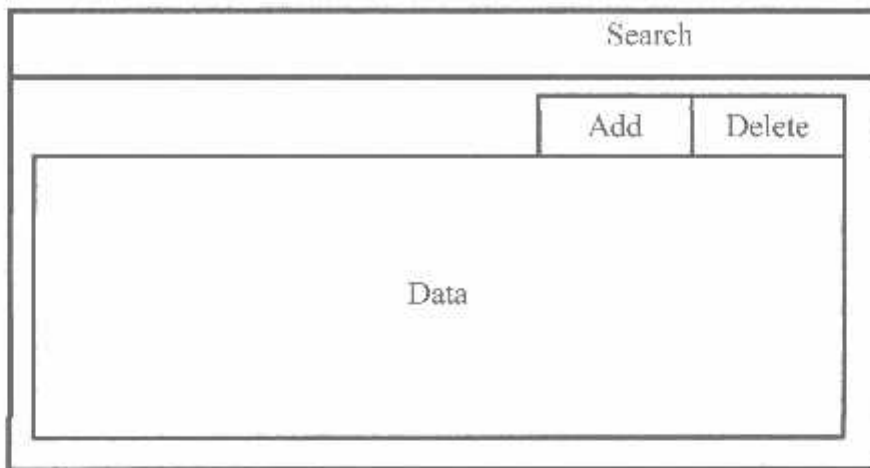
Desain halaman administrasi dapat dilihat dalam gambar 3.11



Gambar 3.11 *Layout* halaman administrasi

5. *Layout* halaman TOPSIS untuk admin

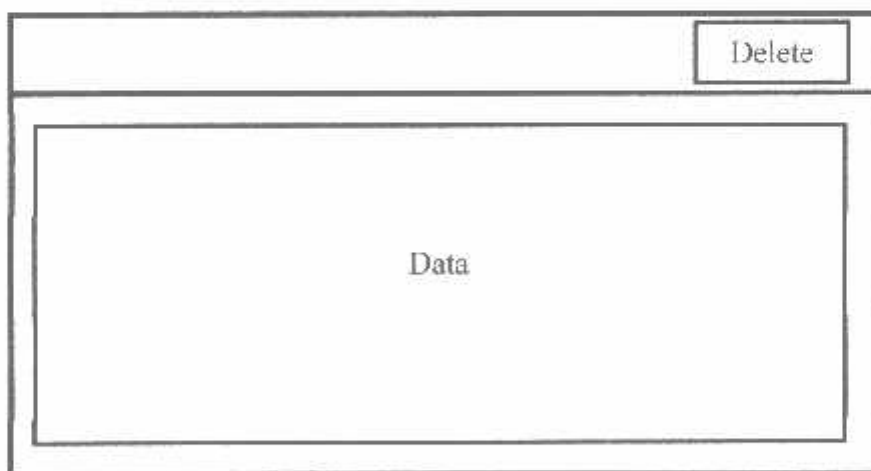
Halaman ini berfungsi untuk menambah, menghapus dan mengubah data-data yang nantinya digunakan untuk penghitungan metode TOPSIS. Desain dapat dilihat pada gambar 3.12.



Gambar 3.12 Layout halaman TOPSIS

6. *Layout* halaman pesan dan komentar untuk admin

Halaman ini berfungsi untuk administrasi (membalas dan menghapus) pesan dan komentar yang dikirim *user* ke admin. Desain dapat dilihat pada gambar 3.13.



Gambar 3.13 Layout halaman pesan dan komentar

7. *Layout* halaman galeri dan header untuk admin

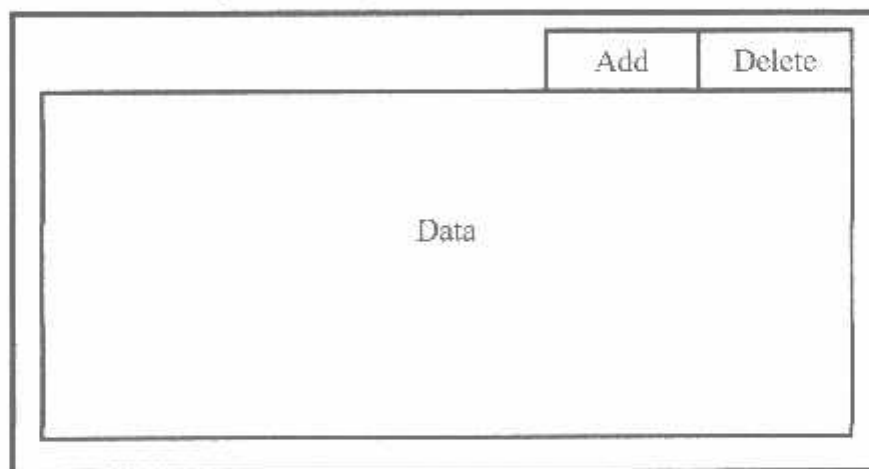
Halaman ini digunakan untuk administrasi gambar semua gambar yang ada di aplikasi. Desain untuk halaman ini dapat dilihat pada gambar 3.14



Gambar 3.14 Layout halaman gallery

8. *Layout* halaman setting untuk admin

Halaman ini berfungsi untuk menambah, menghapus dan merubah *user* admin, serta menentukan bobot dan criteria yang ada dalam sistem. Desain pada halaman ini dapat dilihat pada gambar 3.15.



Gambar 3.15 Layout halaman setting

3.7.4 *Construction*

Tahap ini terbagi menjadi dua, yaitu tahap implementasi dan tahap pengujian. Untuk sub bab ini akan dibahas secara detail pada BAB IV.

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Nilai Kecocokan

Pada intinya data yang dibutuhkan pada penghitungan dalam sistem rekomendasi ini disimpan pada tabel kecocokan. Tabel ini berisi nilai kecocokan yang di miliki setiap alternatif / atribut dan juga nilai tingkat kepentingan setiap kriteria yang ada.

Pada tabel 4.1 berisi data nilai kecocokan dari setiap alternatif n terhadap setiap kriteria m dengan rentang nominal rating 1-5, dimana nilai rating 1 berarti alternatif dinilai sangat buruk dan rating 5 dinilai alternatif sangat baik, seperti yang sudah dijelaskan pada bab sebelumnya.

Berikut adalah contoh sebagian data yang digunakan dalam penelitian ini:

Tabel 4.1 Tabel nilai kecocokan

Id_atribut	Id_kriteria	Id_bobot
a1	c2	3
a1	c1	3
a2	c1	4
a2	c2	2
a3	c1	3

Pada implementasinya dalam memudahkan pembacaan data pada tabel nilai kecocokan dan relasinya ini, dibuatlah sebuah *view* dalam *database* dengan SQL sebagai berikut:

```
create view vw_kecocokan as select a.id_atribut, a.id_kriteria, c.nama  
as atribut, d.nama as kriteria, b.bobot, d.status as status_kriteria from  
't_kecocokan' a join 't_bobot' b on a.id_bobot=b.id join 't_atribut' c on  
a.id_atribut = c.id join 't_kriteria' d on a.id_kriteria = d.id
```

4.2 Implementasi Metode TOPSIS

Implementasi penghitungan metode TOPSIS didalam aplikasi ini masing-masing adalah sebagai berikut:

1. Membangun matrik ternormalisasi

Untuk membangun matrik yang ternormalisasi ini menggunakan persamaan 2.3 seperti yang disebutkan pada bab 2. Implementasi dari persamaan 2.3 kedalam salah satu kelas TOPSIS dapat dilihat dibawah ini:

```
function matrik_normalisasi() {
    $baris = $this->get_baris();
    $kolom = $this->get_kolom();
    $mat_normalisasi = array();

    for($i=1;$i<=$baris;$i++) {
        for($j=1;$j<=$kolom;$j++) {
            $this->db->select($this->t_kecocokan['field']['bobot'],' as bobot');
            $this->db->where($this->t_kecocokan['field']['id_kriteria'],'c{$j}");
            $query = $this->db->get($this->t_kecocokan['name']);

            $xTmp = 0;
            foreach ($query->result() as $value) {
                $xTmp = $xTmp+pow($value->bobot,2);
            }

            $x{$j} = besqrt($xTmp,4);
            $this->db->select($this->t_kecocokan['field']['bobot']);
            $this->db->where($this->t_kecocokan['field']['id_kriteria'],'c{$j}");
            if(empty($this->alternatif_filter)) {
                $this->db->where($this->t_kecocokan['field']['id_atribut'],
                    $this->alternatif_filter[$i]);
            } else {
                $this->db->where($this->t_kecocokan['field']['id_atribut'],'a{$i}");
            }

            $query_ = $this->db->get($this->t_kecocokan['name']);

            foreach ($query_>result() as $val) {
                @$mat_normalisasi{$i}{$j} = bcddiv($val->bobot, $x{$j},4);
            }
        }
    }

    return $mat_normalisasi;
}
```

2. Membangun matrik ternormalisasi terbobot

Untuk membangun matrik yang ternormalisasi terbobot digunakan persamaan 2.4. Implementasi dari persamaan 2.4 didalam kelas TOPSIS adalah sebagai berikut:

```

function matrik_normalisasi_terbobot() {
  $mat_normal = $this->matrik_normalisasi();
  for($i=1;$i<=$this->baris;$i++) {
    for($j=1;$j<=$this->kolom;$j++) {
      @$mY{$i}{$j} = bcmul($this->bobot_p[$j-1],
        $mat_normal[$i][$j],4);
    }
  }
  return $mY;
}

```

3. Menentukan matriks solusi ideal positif dan matrik solusi ideal negatif

Untuk menentukan matrik solusi ideal positif dan matrik solusi ideal negative digunakan persamaan 2.6 dan persamaan 2.8. Implementasi dari persamaan 2.6 dan persamaan 2.8 adalah sebagai berikut:

```

function get_solusi_ideal_positif() {
  $st = $this->get_status_kriteria();
  $data = $this->transpose_matrik();
  for($i=1;$i<=$this->kolom;$i++) {
    for($j=1;$j<=$this->baris;$j++) {
      if($st[$j]==1) { //default code $st[$j]==1
        $max{$i} = max($data[$i]);
      } else {
        $max{$i} = min($data[$i]);
      }
    }
  }
  return $max;
}

function get_solusi_ideal_negatif() {
  $st = $this->get_status_kriteria();
  $data = $this->transpose_matrik();
  for($i=1;$i<=$this->kolom;$i++) {
    for($j=1;$j<=$this->baris;$j++) {
      if($st[$j]==1) {
        $min{$i} = min($data[$i]);
      } else {
        $min{$i} = max($data[$i]);
      }
    }
  }
  return $min;
}

```

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif

Jarak antara setiap alternatif dengan solusi ideal dapat diperoleh dengan persamaan 2.9 dan persamaan 2.10. implementasi dari persamaan 2.9 dan persamaan 2.10 adalah sebagai berikut:

```
function get_jarak_tribut_negatif() {
    $data = $this->matrik_normalisasi_terbobot();
    $solusi_min = $this->get_solusi_ideal_negatif();

    for($i=1;$i<=$this->baris;$i++) {
        $dMinTmp{$i} = 0;
        for($j=1;$j<=$this->kolom;$j++) {
            $dMinTmp{$i} = $dMinTmp{$i}+pow(($data[$i][$j] -
            $solusi_min[$j]),2);
        }
        $dMin{$i} = bcsqrt($dMinTmp{$i},4);
    }

    return $dMin;
}

function get_jarak_tribut_positif() {
    $data = $this->matrik_normalisasi_terbobot();
    $solusi_max = $this->get_solusi_ideal_positif();

    for($i=1;$i<=$this->baris;$i++) {
        $dPlusTmp{$i} = 0;
        for($j=1;$j<=$this->kolom;$j++) {
            $dPlusTmp{$i} = $dPlusTmp{$i}+pow(($data[$i][$j]-
            $solusi_max[$j]),2);
        }
        $dPlus{$i} = bcsqrt($dPlusTmp{$i},4);
    }

    return $dPlus;
}
```

5. Menentukan nilai preferensi untuk setiap alternatif

Dalam proses ini, setelah nilai preferensi didapat dengan menggunakan persamaan 2.11 kemudian dilakukan pengurutan berdasarkan nilai yang tertinggi yang kemudian hasilnya dijadikan acuan dalam perekomendasi. Implementasi persamaan 2.11 dan dalam mendapatkan rekomendasi alternatif adalah sebagai berikut:

```

function get_ranking() {
    $dMin = $this->get_jarak_ atribut_negatif();
    $dPlus = $this->get_jarak_ atribut_positif();

    $rankTmp{$this->baris} = array();
    for($i=1;$i<=$this->baris;$i++) {
        for($j=1;$j<=$this->kolom;$j++) {
            $v{$i} = @bcddiv($dMin[$i], ($dPlus[$i]+$dMin[$i]),4);
        }

        $dt = $this->get_detail_ atribut($this->alternatif_filter[$i]);

        foreach ($dt->result() as $value) {
            $rankTmp{$i} = array('id_ atribut' => $this->alternatif_filter[$i],
                                'nama' => $this->nama_ atribut[$i],
                                'menu' => $value->menu,
                                'harga' => $value->harga,
                                'alamat' => $value->alamat,
                                'gambar' => $value->image,
                                'letak' => $value->letak,
                                'keterangan' => $value->keterangan,
                                'value' => (float) $v[$i]);
        }
    }

    asort($rankTmp);

    $C = $this->bubbleSort($rankTmp);

    return $C;
}

```

4.3 Implementasi Komponen Aplikasi

Daftar kelas-kelas yang telah diimplementasikan pada aplikasi ini dapat dilihat pada tabel 4.2.

Tabel 4.2 Implementasi komponen aplikasi

No.	Nama Kelas	Nama File Fisik	Keterangan
1	Mod_admin	mod_admin.php	Model untuk halaman administrasi
2	Mod_interface	mod_interface.php	Model untuk halaman interface user
3	Mod_login	mod_login.php	Model untuk login menjadi admin
4	TOPSIS	TOPSIS.php	Library untuk penerapan metode TOPSIS dan perkomendasi

5	Authorization	authorization.php	Library untuk autentifikasi login menjadi admin
6	Home	home.php	Controller untuk berinteraksi antara user dan aplikasi
7	Cpanel	cpanel.php	Controller untuk berinteraksi antara admin dan aplikasi
8	Login	login.php	Controller untuk melakukan login menjadi admin
9	Upload	upload.php	Controller untuk melakukan upload image ke aplikasi

4.4 Implementasi *Interface*

Berdasarkan rancangan *interface* yang dibahas pada bab dan poin sebelumnya maka dapat dilihat daftar *interface* yang sudah diimplementasikan pada tabel 4.7

Tabel 4.7 Daftar hasil implementasi *interface*

No.	Nama Kelas (Controller)	Nama File Fisik (View)	Keterangan
1	Home	header.php portal.php footer.php sidebar.php content_index.php content_rekomendasi.php content_hasil_rekomendasi.php content_about.php content_detail atribut.php comment.php content_404.php	Interface buat user
2	Login	login.php dashboard.php	Untuk login menjadi admin
3	Cpanel	header.php menu.php shortcut.php footer.php dashboard.php content_dashboard.php content_kriteria.php content_atribut.php content_kecocokan.php content_setting.php content_email.php content_komen.php content_profile.php content_galery.php modal_admin.php modal_album.php	Interface untuk admin, modal merupakan form untuk penambahan, perubahan, serta penghapusan data/informasi yang ada di database

	modal_atribut.php modal_bobot.php modal_kecocokan.php modal_kriteria.php modal_rep_message.php modal_delete.php	
--	--	--

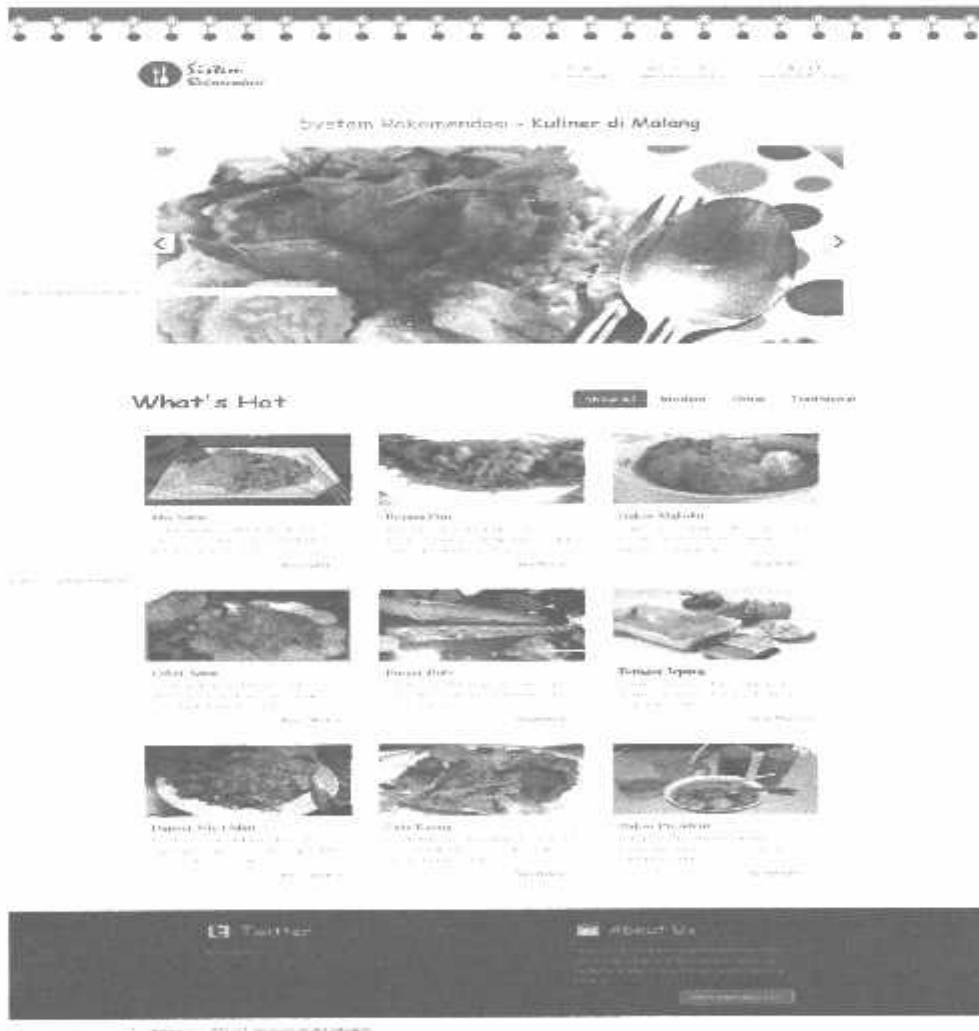
File-file *view* pada tabel 4.7 ini dipanggil oleh kelas *controller* terkait untuk ditampilkan menjadi antarmuka untuk *user* maupun admin.

4.5 Hasil Implementasi

Setelah membuat rancangan dan melakukan proses implementasi, didapatkanlah hasil dari aplikasi sebagai berikut:

1. Tampilan halaman *home* untuk *user*

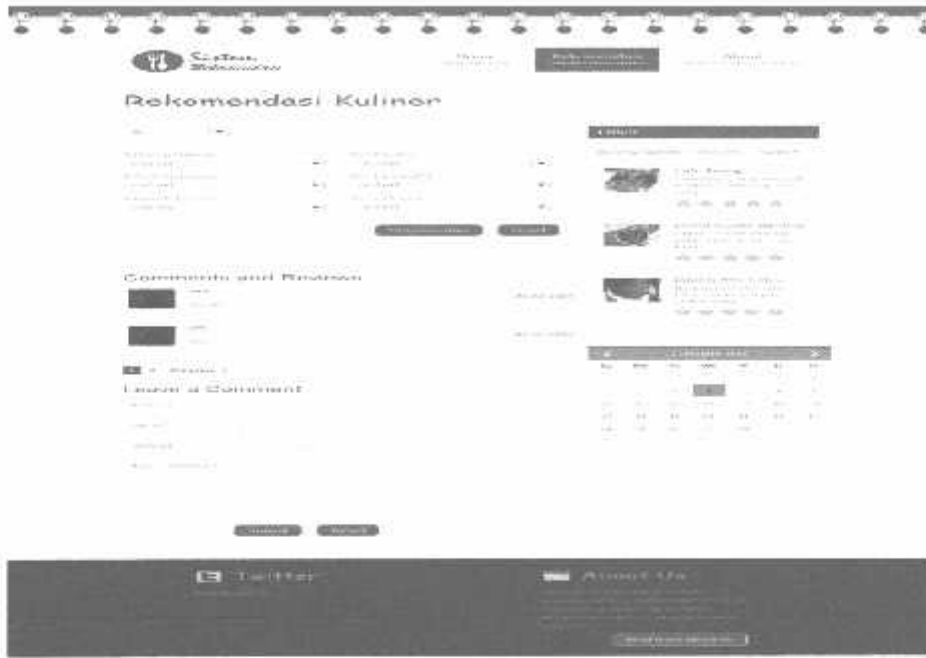
Pada saat pertama kali *user*, menjalankan aplikasi ini akan nampak halaman awal seperti pada gambar 4.1.



Gambar 4.1 Halaman *home*

2. Tampilan halaman rekomendasi untuk *user*

Halaman ini digunakan oleh *user* untuk mendapatkan rekomendasi. Hasil implementasi pada halaman ini ditunjukkan pada gambar 4.2.



Gambar 4.2 Halaman rekomendasi

3. Tampilan halaman hasil rekomendasi untuk *user*

Hasil implementasi pada halaman ini ditunjukkan pada gambar 4.3.



Gambar 4.3 Halaman hasil rekomendasi

4. Tampilan halaman about

Halaman ini berisi keterangan tentang aplikasi yang dibuat. Hasil implementasi pada halaman ini ditunjukkan pada gambar 4.4.



Gambar 4.4 Halaman about

5. Tampilan halaman login untuk admin

Pada saat admin akan masuk ke halaman administrasi maka akan nampak halaman ini, yang menyuruh admin untuk melakukan autentikasi. Hasil implementasi pada halaman ini dapat dilihat pada gambar 4.5.



Gambar 4.5 Halaman login

6. Tampilan halaman *dashboard* untuk admin

Hasil implementasi dari halaman ini dapat dilihat pada gambar 4.6.



Gambar 4.6 Halaman *dashboard*

7. Tampilan Halaman *header web* untuk admin

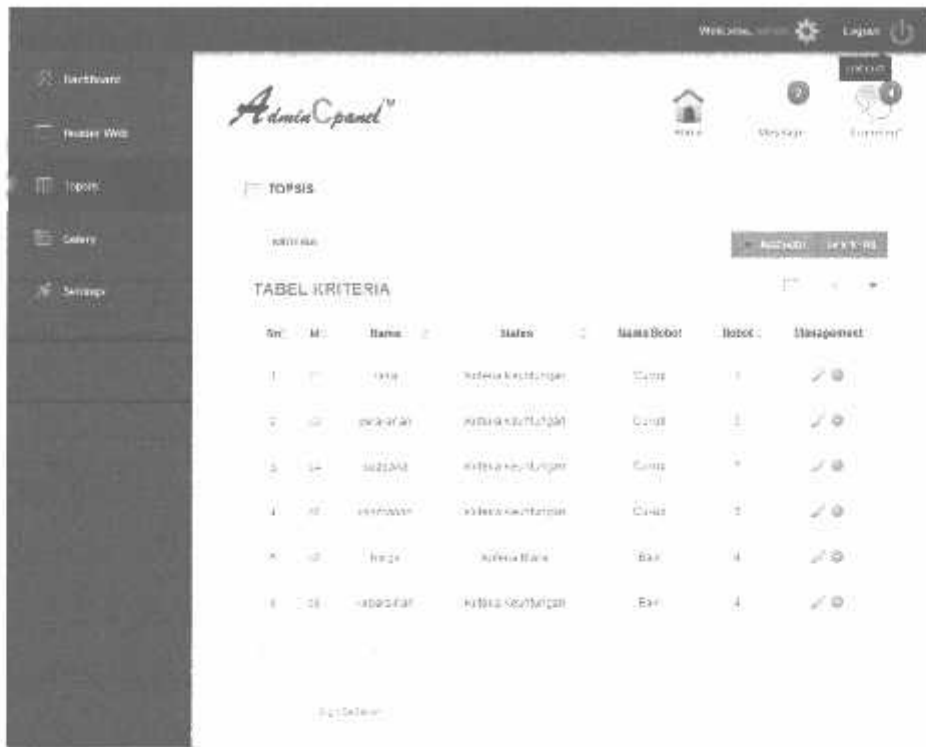
Hasil implementasi dari halaman ini dapat dilihat pada gambar 4.7



















Gambar 4.7 Halaman *header web*

8. Tampilan halaman menu TOPSIS untuk admin

Hasil implementasi untuk halaman ini dapat dilihat sebagai berikut:

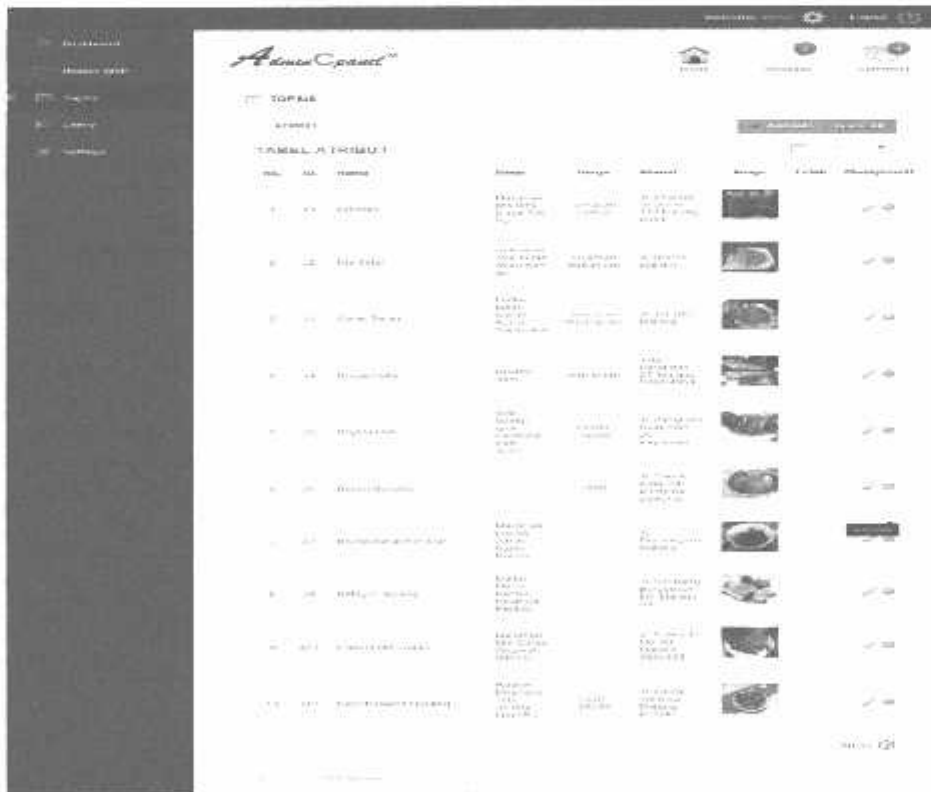


The screenshot shows the Admin Panel interface for the TOPSIS application. The main content area displays a table titled "TABEL KRITERIA" (Criteria Table). The table has 8 columns: No., M., Nama, Status, Nama Bobot, bobot, and Manajemen. The data rows are as follows:














No.	M.	Nama	Status	Nama Bobot	bobot	Manajemen
1	01	1919	Notifikasi Kelengkapan	Catatan	1	 
2	02	1919	Notifikasi Kelengkapan	Catatan	2	 
3	03	1919	Notifikasi Kelengkapan	Catatan	3	 
4	04	1919	Notifikasi Kelengkapan	Catatan	4	 
5	05	1919	Notifikasi Kelengkapan	Catatan	5	 
6	06	1919	Notifikasi Kelengkapan	Catatan	6	 
7	07	1919	Notifikasi Kelengkapan	Catatan	7	 
8	08	1919	Notifikasi Kelengkapan	Catatan	8	 

Gambar 4.8 Halaman tabel kriteria

Untuk Halaman tabel atribut dapat dilihat pada gambar 4.9



The screenshot shows the Admin Panel interface for the TOPSIS application, displaying a table titled "TABEL ATRIBUT" (Attribute Table). The table has 8 columns: No., M., Nama, Status, Nama Bobot, bobot, and Manajemen. The data rows are as follows:

No.	M.	Nama	Status	Nama Bobot	bobot	Manajemen
1	01	1919	Notifikasi Kelengkapan	Catatan	1	 
2	02	1919	Notifikasi Kelengkapan	Catatan	2	 
3	03	1919	Notifikasi Kelengkapan	Catatan	3	 
4	04	1919	Notifikasi Kelengkapan	Catatan	4	 
5	05	1919	Notifikasi Kelengkapan	Catatan	5	 
6	06	1919	Notifikasi Kelengkapan	Catatan	6	 
7	07	1919	Notifikasi Kelengkapan	Catatan	7	 
8	08	1919	Notifikasi Kelengkapan	Catatan	8	 
9	09	1919	Notifikasi Kelengkapan	Catatan	9	 
10	10	1919	Notifikasi Kelengkapan	Catatan	10	 

Gambar 4.9 Halaman table atribut

Sedangkan untuk halaman tabel kecocokan dapat dilihat pada gambar 4.10

The screenshot shows the 'TABEL KECECOKAN' page in the Admin Panel. The table contains the following data:

ID	Nama	Harga	Spesifikasi	Warna	Ketersediaan	Status
1	Kopi Susu	2	100ml	Putih	10	✓
2	Biskuit	1	100g	Brown	5	✓
3	Kue Kacang	3	150g	Brown	3	✓
4	Kue Coklat	2	100g	Brown	5	✓
5	Wafel Susu	1	100g	Putih	10	✓
6	Wafel Coklat	1	100g	Brown	10	✓
7	Pasta	10	500g	Putih	10	✓
8	Margarin	5	250g	Putih	5	✓
9	Minyak Goreng	3	500ml	Putih	5	✓

Gambar 4.10 Halaman tabel kecocokan

9. Tampilan halaman Setting untuk admin

Hasil implementasi dari halaman ini dapat dilihat pada gambar 4.11

The screenshot shows the 'User Admin' page in the Admin Panel. The table contains the following data:

No.	Username	Password	Email	Manajemen
1	admin	admin	admin@domain.com	✓
2	user	user	user@domain.com	✓

Below the table, there are sections for 'Kategori Produk' and 'Kategori Admin':

Kategori Produk

No.	Nama	Harga	Manajemen
1	Manajemen	1	✓
2	Produk	2	✓
3	Produk	1	✓
4	Produk	1	✓
5	Produk	1	✓

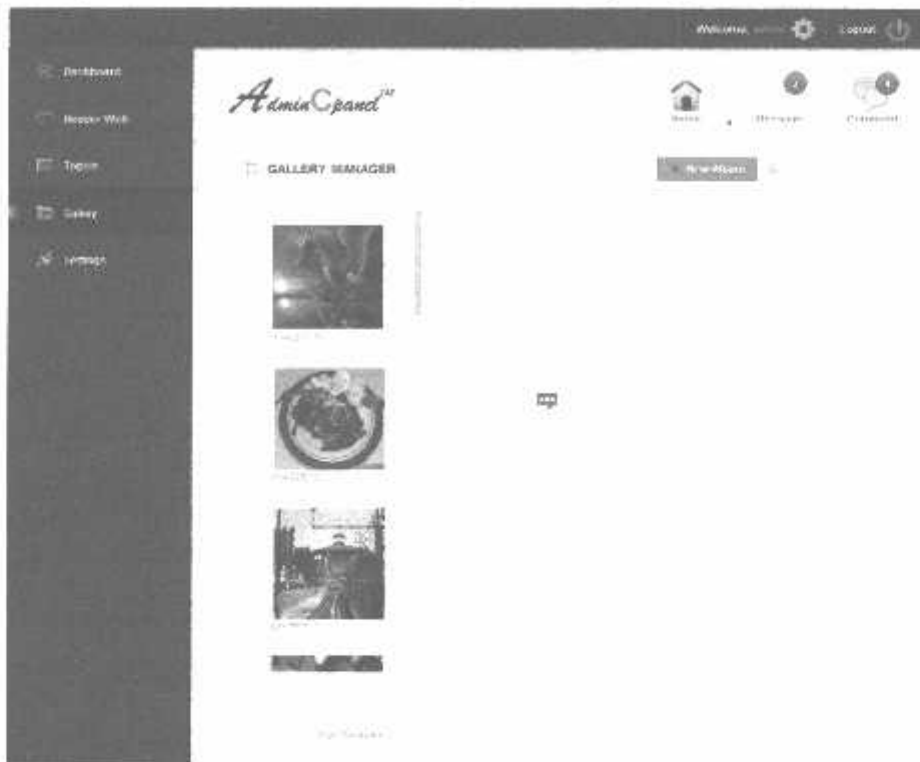
Kategori Admin

No.	Kategori	Label	Manajemen
1	Admin	Admin	✓
2	Admin	Other	✓
3	Admin	Admin	✓

Gambar 4.11 Halaman setting

10. Tampilan halaman gallery untuk admin

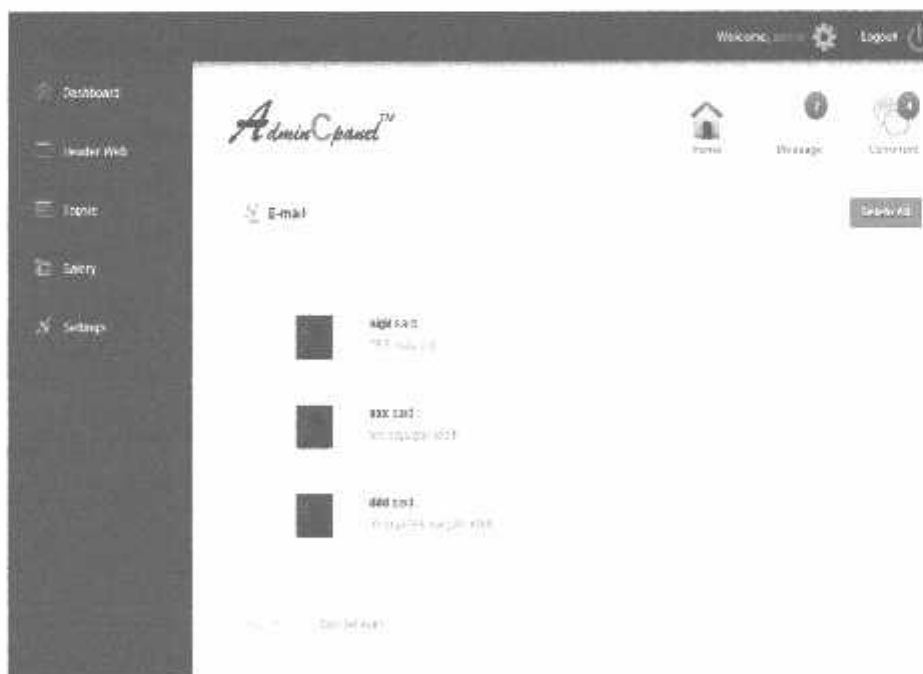
Hasil implementasi dari halaman ini dapat dilihat pada gambar 4.12



Gambar 4.12 Halaman gallery

11. Halaman pesan

Hasil implementasi dari halaman ini dapat dilihat pada gambar 4.13



Gambar 4.13 Halaman pesan

12. Halaman komentar

Hasil implementasi dari halaman ini dapat dilihat pada gambar 4.14



Gambar 4.14 Halaman komentar

4.6 Pengujian

Tahap pengujian adalah suatu proses untuk menguji aplikasi yang telah selesai dibuat. Hal ini bertujuan untuk menemukan kesalahan dan kemudian memperbaikinya. Berikut merupakan bentuk pengujian yang dilakukan:

1. Pengujian *browser*

Pengujian ini dilakukan untuk mengetahui *web browser* apa saja yang dapat didukung oleh aplikasi ini. Hasil pengujian dapat dilihat pada tabel 4.4.

Tabel 4.4 Hasil pengujian terhadap beberapa *browser*

Browser	Kompatibilitas		Keterangan
	Sistem	Antarmuka	
Google Chrome	Ya	Ya	
Mozilla Firefox	Ya	Ya	
Internet Explorer	Ya	Tidak	CSS 3 dan JS tidak berjalan
Opera	Ya	Ya	
Safari	Ya	Ya	
Maxthon	Ya	Tidak	CSS 3 dan JS kurang optimal
Avant Browser	Ya	Tidak	CSS 3 kurang optimal

2. Pengujian aplikasi dengan teknik *blackbox*

Pada pengujian aplikasi ini dilakukan dengan menggunakan teknik pengujian *blackbox*. Pada dasarnya teknik ujicoba *blackbox* memfokuskan keperluan fungsional dari aplikasi yang dibuat dengan mengabaikan internal atau komponen dari aplikasi tersebut.

Pengujian pada aplikasi ini terbagi menjadi 2, yaitu:

- Sisi user

Bentuk pengujian pada sisi *user* dapat dilihat pada tabel 4.5.

Tabel 4.5 Rencana dan bentuk pengujian disisi *user*

No.	Fungsi Yang Diuji	Skenario Pengujian	Data Uji Yang Digunakan	Kode Uji
1	Lihat Kuliner	Melihat kuliner serta detail kuliner	-	Uji-u1
2	Membuat rekomendasi	Mengakses menu rekomendasi	Data kategori, kata kunci, dan bobot preferensi	Uji-u2
3	Mengirim komentar	Melakukan pengiriman komentar	Data nama, email, website, dan komentar	Uji-u3
4	Mengirim pesan	Melakukan pengiriman pesan	Data nama, email, url, dan pesan	Uji-u4

- Sisi admin

Bentuk pengujian pada sisi admin dapat dilihat pada tabel 4.6.

Tabel 4.6 Rencana dan bentuk pengujian disisi admin

No.	Fungsi Yang Diuji	Skenario Pengujian	Data Uji Yang Digunakan	Kode Uji
1	Autentikasi	Melakukan login pada system	Data username dan password	Uji-a1
2	CRUD atribut	Input, update dan delete data atribut	Data id_data, nama, Menu, harga, alamat, gambar, letak dan artikel	Uji-a2
3	CRUD kategori	Input, update dan delete data kategori	Data id_data, kategori, dan label	Uji-a3
4	CRUD nilai kecocokan	Input, update dan delete data nilai kecocokan	Data id_atribut, id_kriteria, id_bobot	Uji-a4

5	CRUD admin	Input, update dan delete data admin	Data id_data, user, pass dan email	Uji-a5
6	CRUD kriteria	Input, update dan delete data kriteria	Data id_data, nama, status dan bobot	Uji-a6
7	CRUD bobot	Input, update dan delete data bobot	Data id_data, nama, dan bobot	Uji-a7
8	Administrasi galeri	Menambah dan menghapus directory dan mengupload gambar	Data nama, gambar	Uji-a8
9	Administrasi komentar	Membalas dan menghapus komen	Data nama, email, url dan komen	Uji-a9
10	Administrasi pesan	Membalas dan menghapus pesan	Data nama, email, website dan komen	Uji-a10
11	Administrasi Header web	Upload dan menghapus header web	Data nama, gambar	Uji-a11

Dan berikut salah satu hasil dari pengujian pada sisi *user* yang telah dilakukan:

Kode uji	Uji-u1 dan Uji-u2
Deskripsi pengujian	Menguji proses melihat dan membuat rekomendasi yang dihasilkan aplikasi
Tahapan pengujian	<ul style="list-style-type: none"> • Masuk menu rekomendasi • Pilih kategori kuliner yang akan direkomendasikan • Input kata kunci untuk memilih menu makanan tertentu yang akan direkomendasikan • Pilih tingkat kepentingan kriteria yang ada Rekomendasi
Input data	Kategori:modern Kata kunci:mi Bobot preferensi:Default
Hasil yang diharapkan	Aplikasi memberikan daftar rekomendasi kuliner berdasarkan algoritma yang digunakan
Hasil Nyata	Aplikasi memberikan daftar rekomendasi
Kesimpulan	Diterima

Untuk rekap hasil pengujian pada sisi *user* dapat dilihat pada tabel 4.7 berikut ini:

Tabel 4.7 Rekap hasil pengujian *blackbox* pada sisi *user*

Kode uji	Fungsi yang diuji	Hasil
Uji-u1	Lihat kuliner	Sesuai
Uji-u2	Membuat rekomendasi	Sesuai
Uji-u3	Mengirim komentar	Sesuai
Uji-u4	Mengirim pesan	Sesuai

Sedangkan untuk hasil salah satu pengujian pada sisi admin yang dilakukan sebagai berikut:

Kode uji	Uji-a1
Deskripsi pengujian	Menguji fungsi autentikasi dengan cara login ke sistem
Tahapan pengujian	<ul style="list-style-type: none"> • Inputkan username • Inputkan password submit
Input data	Username:admin Password:admin
Hasil yang diharapkan	Login sukses, sistem menset session untuk pengguna.
Hasil Nyata	Login sukses, sistem menset session untuk pengguna.
Kesimpulan	Diterima

Untuk rekap hasil pengujian pada sisi admin dapat dilihat pada tabel 4.8 berikut ini:

Tabel 4.8 Rekap hasil pengujian *blackbox* pada sisi admin

Kode uji	Fungsi yang diuji	Hasil
Uji-a1	Autentikasi	Sesuai
Uji-a2	CRUD atribut	Sesuai
Uji-a3	CRUD kategori	Sesuai
Uji-a4	CRUD nilai kecocokan	Sesuai
Uji-a5	CRUD admin	Sesuai
Uji-a6	CRUD kriteria	Sesuai

Uji-a7	CRUD bobot	Sesuai
Uji-a8	Administrasi galeri	Sesuai
Uji-a9	Administrasi komentar	Sesuai
Uji-a10	Administrasi pesan	Sesuai
Uji-a11	Administrasi header web	Sesuai

3. Pengujian *user*

Pengujian ini dilakukan terhadap 20 *user* untuk menggunakan aplikasi. Hasil dari pengujian ini dapat dilihat pada tabel 4.9.

Tabel 4.9 Hasil pengujian terhadap *user*

Kategori	Penilaian	Jumlah	Presentase
Manfaat	Membantu	13	65%
	Cukup	5	25%
	Tidak Membantu	2	10%
Tampilan <i>user friendly</i>	Ya	18	90%
	Tidak	2	10%

4. Pengujian kecepatan perhitungan metode TOPSIS

Pengujian ini dilakukan untuk mengetahui jumlah alternatif berpengaruh atau tidaknya dalam kecepatan perhitungan metode TOPSIS. Pengujian dilakukan pada komputer dengan processor AMD Turion X2 2.1 Ghz, RAM 1 Gb, Sistem Operasi XP SP 2, dengan *browser Google Chrome*. Hasil dari pengujian ini dapat dilihat pada tabel 4.10

Tabel 4.10 Hasil pengujian kecepatan perhitungan metode TOPSIS

Jumlah alternatif	Kecepatan perhitungan (detik)
3	0,2137
5	0,3784
7	0,5216
10	0,7364
13	0,9658

BAB V PENUTUP

5.1 Kesimpulan

Dari seluruh hasil pengujian penelitian yang dilaksanakan, dapat disimpulkan bahwa:

1. Sistem pada aplikasi ini dapat berfungsi dengan baik pada semua *browser* yang terkenal pada saat ini, yaitu: *google chrome*, *Mozilla firefox*, *opera*, *Internet Explorer*, *Maxthon*, *avant browser* dan *safari*. Sedangkan untuk *interface* / antarmuka berjalan dengan baik pada *google chrome*, *Mozilla firefox*, *opera* dan *safari*, sedangkan pada *maxthon* dan *avant browser*, CSS3 dan Javascript berjalan kurang optimal dan pada IE CSS3 dan Javascript tidak berjalan.
2. Berdasarkan hasil pengujian dengan metode *blackbox* pada sisi *user* dan admin, semua fungsionalitas dapat berjalan sesuai yang diinginkan.
3. Dari hasil pengujian *user*, sebanyak 65% *user* merasa aplikasi ini dapat membantu dalam mencari dan merekomendasikan tempat kuliner yang ada di Malang dan sebanyak 90% *user* merasa aplikasi ini memiliki tampilan / *interface* yang *user friendly* sehingga mudah mengoperasikannya.
4. Jumlah kriteria dan alternatif dalam *database* mempengaruhi kecepatan dalam perhitungan metode TOPSIS ini.
5. Dari hasil pengujian, metode TOPSIS ini terbukti efektif dalam memberikan rekomendasi kuliner karena metode ini dapat menyelesaikan masalah perekomendasi kuliner yang ada dan terbukti efisien karena *user* saat ingin mendapatkan rekomendasi kuliner, tidak harus memberikan rating beberapa kuliner yang ada dan *user* tidak perlu login ke sistem dan Inputan dari *user*, hanya berupa jenis kategori, kata kunci kuliner, dan tingkat kepentingan kriteria yang ada.

5.2 Saran

1. Penambahan peta dari tempat kuliner yang direkomendasikan. Sehingga *user* mendapat fasilitas *direct access* (penunjuk arah) ke tempat tujuan.
2. *User* dapat menambahkan rekomendasi tempat kuliner yang ada, sehingga data akan *ter-update* terus tanpa menunggu dari admin. Admin tinggal menerima atau menolak tambahan informasi dari *user*.

DAFTAR PUSTAKA

- [1] Daqiqil, Ibnu. 2011. *Framework Codeigniter sebuah panduan dan best practice*. <http://koder.web.id/>, diakses 30 april 2012
- [2] Dwiartara, Loka. 2011. *Menyelam & Menaklukan Samudra PHP*. www.ilmuwebsite.com, diakses 20 Pebruary 2012.
- [3] Indonesia, Wonderful. 2012. *Malang Masuk Daftar 10 Destinasi Wisata Baru di Asia*. <http://www.indonesia.travel/>, diakses 20 February 2012
- [4] Kusumadewi, Sri; Hartati, Sri; Harjoko, Agus; Wardoyo, Retantyo. 2006. *Fuzzy Multi-Attribute Decision Making (Fuzzy MADM)*. Graha Ilmu. Yogyakarta.
- [5] Ricci, Francesco; Rokach, Lior; Shapira, Bracha. 2011. *Introduction to Recommender System Handbooks*. Springer Science+Business Media.
- [6] Pressman, Roger S. 2004. *Software engineering: a practitioner's approach – fifth edition*. New York. McGraw-Hill in computer science.
- [7] Sharda, Nalin, 2010. *Tourism Informatics: visual travel recommender systems, social communities, and user interface design*. New York: Information Science Reference.
- [8] Solichin, Achmad. 2010. *MySQL 5 dari Pemula hingga Mahir*. <http://achmatim.net/download/21/>, diakses 18 Maret 2012.
- [9] Ullman Larry. 2008. *PHP 6 and MySQL 5 for Dinamic Web Sites*. Peachpit Press. Berkeley.
- [10] Wardana. 2010. *Menjadi master PHP dengan framework codeigniter*. PT Elex Media Komputindo. Jakarta.
- [11] Yunanto, Harry. 2011. *Dasar-dasar pembuatan website dengan codeigniter*. <http://iorme.net>, diakses 20 Maret 2012.

LAMPIRAN

Lampiran 1. Surat Pernyataan



INTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

SURAT PERNYATAAN

Nama : Sigit Setiawan
NIM : 08.18.107
Jurusan : Teknik Informatika S-1

Menyatakan bahwa karya skripsi saya yang berjudul:

**“Sistem Rekomendasi Kuliner Di Malang dengan Metode TOPSIS
(Technique for Order Preference by Similarity to Ideal Solution)”**

Adalah bukan merupakan karya tulis orang lain, baik sebagian maupun keseluruhan, kecuali dalam bentuk kuipan yang saya sebutkan sumbernya. Demikian surat pernyataan ini saya buat dengan sebenar-benarnya dan apabila pernyataan ini tidak benar saya bersedia mendapatkan sanksi akademis.

Malang, 27 Mei 2013



Sigit Setiawan

Lampiran 2. Formulir Bimbingan Skripsi



INTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Sigit Setiawan
NIM : 08.18.107
Judul Skripsi : Sistem Rekomendasi Kuliner Di Malang dengan Metode TOPSIS
(Technique for Order Preference by Similarity to Ideal Solution)

No	Tanggal	Uraian	Paraf
1	03-12-2012	Bab 1 sampai bab 2	
2	26-12-2012	Demo Program	
3	02-01-2013	Makalah	
4	07-01-2013	Acc Makalah	
5	22-01-2013	Bab 1 sampai bab 5	
6	12-02-2013	Acc bab 1 sampai bab 5	

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT
NIP.19740416 200501 1 002

Lampiran 3. Formulir Bimbingan Skripsi



INTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR BIMBINGAN SKRIPSI

Nama : Sigit Setiawan
NIM : 08.18.107
Judul Skripsi : Sistem Rekomendasi Kuliner Di Malang dengan Metode TOPSIS
(Technique for Order Preference by Similarity to Ideal Solution)

No	Tanggal	Uraian	Paraf
1	02-09-2012	Revisi bab 1 dan bab 2	
2	05-12-2012	Makalah	
3	11-12-2012	Perbaiki Makalah	
4	19-12-2012	Acc Makalah	
5	24-12-2012	Demo Program	
6	03-01-2013	Bab 3 dan bab 4	
7	16-01-2013	Bab 1 sampai bab 5	
8	23-01-2013	Perbaiki Bab 1 sampai bab 5	
9	05-02-2012	Acc Bab 1 sampai bab 5	

Dosen Pembimbing II

Ali Mahmudi, B.eng, Phd
NIP.P. 1031000429

Lampiran 4. Berita Acara Ujian Skripsi



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

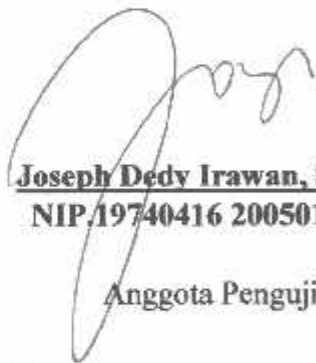
Nama : Sigit Setiawan
NIM : 08.18.107
Jurusan : Teknik Informatika S-1
Judul : Sistem Rekomendasi Kuliner Di Malang dengan Metode TOPSIS
(Technique for Order Preference by Similarity to Ideal Solution)

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 19 Pebruari 2013
Nilai : 89,9 (A)

Panitia Ujian Skripsi :

Ketua Majelis Penguji


Joseph Dedy Irawan, ST, MT
NIP.19740416 200501 1 002

Anggota Penguji :

Penguji Pertama

Penguji Kedua



Sandy Nataly Mantja, S.Kom
NIP.P. 1030800418



Suryo Adi Wibowo, ST
NIP.P. 1031000438

Lampiran 5. Formulir Perbaikan Skripsi



INTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : Sigit Setiawan
NIM : 08.18.107
Jurusan : Teknik Informatika S-1
Judul : Sistem Rekomendasi Kuliner Di Malang dengan Metode TOPSIS
(Technique for Order Preference by Similarity to Ideal Solution)

Penguji	Perbaikan	Tanda Tangan
Penguji I	1. Kata pengantar 2. Rumusan masalah 3. Tidak bolch pakai button tapi angka / huruf 4. Catatan kaki sebagai sumber tidak bolch diletakkan di judul bab 5. Landasan teori yang tidak perlu di hapus	
Penguji II	1. Penulisan 2. Harus online 3. Fungsional dari segi admin dan user harus berjalan 4. Kesimpulan harus dapat menjawab rumusan dan tujuan	

Anggota Penguji :

Penguji Pertama

Sandy Nataly Mantja, S.Kom
NIP.P. 1030800418

Penguji Kedua

Survo Adi Wibowo, ST
NIP.P. 1031000438

Mengetahui

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT
NIP. 19740416 200501 1 002

Dosen Pembimbing II

Ali Mahmudi, B.eng, Phd
NIP.P. 1031000429

Lampiran 6. Hasil Pengujian Aplikasi dengan teknik *blackbox* pada sisi *user*

Kode uji	Uji-u1 dan Uji-u2
Deskripsi pengujian	Menguji proses melihat dan membuat rekomendasi yang dihasilkan aplikasi
Tahapan pengujian	<ul style="list-style-type: none"> • Masuk menu rekomendasi • Pilih kategori kuliner yang akan direkomendasikan • Input kata kunci untuk memilih menu makanan tertentu yang akan direkomendasikan • Pilih tingkat kepentingan kriteria yang ada Rekomendasi
Input data	Kategori:modern Kata kunci:mi Bobot preferensi:Default
Hasil yang diharapkan	Aplikasi memberikan daftar rekomendasi kuliner berdasarkan algoritma yang digunakan
Hasil Nyata	Aplikasi memberikan daftar rekomendasi
Kesimpulan	Diterima

Kode uji	Uji-u3
Deskripsi pengujian	Menguji proses mengirim komentar pada aplikasi
Tahapan pengujian	Masuk menu rekomendasi Inputkan name, email, website, dan comment submit
Input data	Name:sanjaya Email:sanjaya@gmail.com Website: Null Comment: test komentar
Hasil yang diharapkan	Komentar masuk system, dan menampilkannya pada aplikasi
Hasil Nyata	Komentar masuk system, dan menampilkannya pada aplikasi
Kesimpulan	Diterima

Kode uji	Uji-u4
Deskripsi pengujian	Menguji proses mengirim pesan untuk admin pada aplikasi
Tahapan pengujian	Masuk menu admin Inputkan name, email, url, dan comment submit
Input data	Name:sanjaya

	Email: sanjaya@gmail.com Url: http://facebook.com/sanjaya.1922 Comment: test komentar
Hasil yang diharapkan	Pesan masuk system
Hasil Nyata	Pesan masuk system
Kesimpulan	Diterima

Lampiran 7. Hasil Pengujian Aplikasi dengan teknik *blackbox* pada sisi admin

Kode uji	Uji-a1
Deskripsi pengujian	Menguji fungsi autentikasi dengan cara login ke sistem
Tahapan pengujian	<ul style="list-style-type: none"> • Inputkan username • Inputkan password submit
Input data	Username:admin Password:admin
Hasil yang diharapkan	Login sukses, sistem menset session untuk pengguna.
Hasil Nyata	Login sukses, sistem menset session untuk pengguna.
Kesimpulan	Diterima

Kode uji	Uji-a2
Deskripsi pengujian	Menguji fungsi CRUD atribut untuk data
Tahapan pengujian	<ul style="list-style-type: none"> • id_data • nama • Menu • Harga • Alamat • Gambar • letak • artikel submit
Input data	id_data:a14 nama:tes Menu:tes Harga:7000 Alamat:tes juga Gambar: Letak: Artikel:lorem isum dolor si amet
Hasil yang diharapkan	Data masuk ke database
Hasil Nyata	Data masuk ke database
Kesimpulan	Diterima

Kode uji	Uji-a3
Deskripsi pengujian	Menguji fungsi CRUD kategori pada data
Tahapan pengujian	<ul style="list-style-type: none"> • id_data, • kategori

	<ul style="list-style-type: none"> • label submit
Input data	id_data:5 kategori:atribut label:test
Hasil yang diharapkan	Data kategori atribut masuk ke database
Hasil Nyata	Data kategori atribut masuk ke database
Kesimpulan	Diterima

Kode uji	Uji-a4
Deskripsi pengujian	Menguji fungsi CRUD nilai kecocokan pada data
Tahapan pengujian	<ul style="list-style-type: none"> • id_atribut, • id_kriteria • id_bobot submit
Input data	id_atribut:a3 id_kriteria:a4 id_bobot:3
Hasil yang diharapkan	Data nilai kecocokan untuk atribut masuk ke database
Hasil Nyata	Data nilai kecocokan untuk atribut masuk ke database
Kesimpulan	Diterima

Kode uji	Uji-a5
Deskripsi pengujian	Menguji fungsi CRUD admin pada sistem
Tahapan pengujian	<ul style="list-style-type: none"> • id_data • user • pass • email submit
Input data	id_data:3 user:xxx pass:xxx email:xxx@ymail.com
Hasil yang diharapkan	Data admin masuk ke database
Hasil Nyata	Data admin masuk ke database
Kesimpulan	Diterima

Kode uji	Uji-a6
Deskripsi pengujian	Menguji fungsi CRUD kriteria pada sistem
Tahapan pengujian	<ul style="list-style-type: none"> • id_data • nama, • status • bobot submit
Input data	id_data:7 nama:tempat status:1 bobot:3
Hasil yang diharapkan	Data kriteria masuk ke database
Hasil Nyata	Data kriteria masuk ke database
Kesimpulan	Diterima

Kode uji	Uji-a7
Deskripsi pengujian	Menguji fungsi CRUD bobot pada sistem
Tahapan pengujian	<ul style="list-style-type: none"> • id_data • nama, • bobot submit
Input data	id_data:6 nama:istimewa bobot:6
Hasil yang diharapkan	Data bobot masuk ke database
Hasil Nyata	Data bobot masuk ke database
Kesimpulan	Diterima

Kode uji	Uji-a8
Deskripsi pengujian	Menguji fungsi administrasi galeri pada sistem
Tahapan pengujian	<ul style="list-style-type: none"> • nama • Gambar submit
Input data	Nama:test Gambar:img001.jpg
Hasil yang diharapkan	Data gambar masuk ke sistem dan dapat dihapus
Hasil Nyata	Data gambar masuk ke sistem dan dapat dihapus
Kesimpulan	Diterima

Kode uji	Uji-a9
Deskripsi pengujian	Menguji fungsi administrasi komentar pada sistem
Tahapan pengujian	<ul style="list-style-type: none"> • Nama • Email • url • komen submit
Input data	Nama:sigit Email:sigit2@gmail.com url:- komen:test
Hasil yang diharapkan	Komentar dapat dibalas dan dihapus
Hasil Nyata	Komentar dapat dibalas dan dihapus
Kesimpulan	Diterima

Kode uji	Uji-a10
Deskripsi pengujian	Menguji fungsi administrasi pesan pada sistem
Tahapan pengujian	<ul style="list-style-type: none"> • Nama • Email • website • komen submit
Input data	Nama:sigit Email:sigit2@gmail.com website:- komen:test
Hasil yang diharapkan	Pesan dapat dibalas dan dapat dihapus
Hasil Nyata	Pesan dapat dibalas dan dapat dihapus
Kesimpulan	Diterima

Kode uji	Uji-a11
Deskripsi pengujian	Menguji fungsi administrasi header web pada sistem
Tahapan pengujian	<ul style="list-style-type: none"> • nama • Gambar submit
Input data	Nama:img001 Gambar:img001.jpg
Hasil yang diharapkan	Data gambar masuk ke sistem dan dapat dihapus
Hasil Nyata	Data gambar masuk ke sistem dan dapat dihapus
Kesimpulan	Diterima

Lampiran 8. Code Hasil Implementasi Metode TOPSIS

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access
allowed');
class Topsis {

    private $t_tribut;
    private $t_kriteria;
    private $t_kecocokan;
    private $t_bobot;

    private $bobot_p;
    private $baris;
    private $kolom;

    private $kategori_tribut;
    private $kata_kunci;

    private $alternatif_filter; //id alternatif yg terfilter
    private $status_kriteria;
    private $nama_tribut;

    function __construct() {
        $this->CI =& get_instance();
        $this->db = $this->CI->load->database('default',TRUE);

        $this->t_tribut = table_tribut();
        $this->t_kriteria = table_kriteria();
        $this->t_kecocokan = table_kecocokan();
        $this->t_bobot = table_bobot();
    }

    function init($kategori_tribut='', $kata_kunci='',
        $bobot_preferensi='') {

        if(isset($kategori_tribut)) {
            $this->kategori_tribut = $kategori_tribut;
        } else {
            $this->kategori_tribut = NULL;
        }

        $this->bobot_p = $bobot_preferensi;

        if(isset($kata_kunci)) {
            $this->kata_kunci = $kata_kunci;
        } else {
            $this->kata_kunci = NULL;
        }
    }

    function get_baris() {
        if(isset($this->kategori_tribut))
            $this->db->where_in('label', $this->kategori_tribut);

        if(isset($this->kata_kunci)) {
            $this->db->like('nama', $this->kata_kunci);
            $this->db->or_like('menu', $this->kata_kunci);
            $query = $this->db->get($this->t_tribut['name']);

            $this->baris = count($query->result_array());
        }
    }
}
```

```

        $this->alternatif_filter = array(); //untuk
        mengambil alternatif id yg ke filter
        $i=1;

        foreach ($query->result() as $value) {
            $this->alternatif_filter[$i] = $value->id; //id
            atribut
            $this->nama_atribut[$i] = $value->nama; //nama
            atribut
            $i++;
        }

    } else {
        $query = $this->db->get(
            $this->t_atribut['name']);
        $this->alternatif_filter = array(); //untuk
        mengambil alternatif id yg ke filter
        $i=1;

        foreach ($query->result() as $value) {
            $this->alternatif_filter[$i] = $value->id;
            //id atribut
            $this->nama_atribut[$i] = $value->nama;
            //nama atribut
            $i++;
        }

        $this->baris = $this->db->count_all(
            $this->t_atribut['name']);
    }

    return $this->baris;
}

function get_kolom() {
    $this->kolom = $this->db->count_all(
        $this->t_kriteria['name']);

    return $this->kolom;
}

//default bobot preferensi
function get_default_bobot_preferensi() {
    $this->db->select(
        $this->t_kriteria['field']['bobot'].' as bobot');
    $query = $this->db->get($this->t_kriteria['name']);

    $bobot = array();
    $i = 0;
    foreach ($query->result() as $value) {
        $bobot[$i] = $value->bobot;
        $i++;
    }

    return $bobot;
}

```

```

function matrik_normalisasi() {
    $baris = $this->get_baris();
    $kolom = $this->get_kolom();
    $mat_normalisasi = array();

    for($i=1;$i<=$baris;$i++) {
        for($j=1;$j<=$kolom;$j++) {
            $this->db->select(
                $this->t_kecocokan['field']['bobot'].' as bobot');
            $this->db->where(
                $this->t_kecocokan['field']['id_kriteria'],'c($j)');
            $query = $this->db->get($this->t_kecocokan['name']);

            $xTmp = 0;
            foreach ($query->result() as $value) {
                $xTmp = $xTmp+pow($value->bobot,2);
            }

            $x[$j] = bcsqrt($xTmp,4);

            $this->db->select(
                $this->t_kecocokan['field']['bobot']);
            $this->db->where(
                $this->t_kecocokan['field']['id_kriteria'],'c($j)');
            if(!empty($this->alternatif_filter)) {
                $this->db->where(
                    $this->t_kecocokan['field']['id_atribut'],
                    $this->alternatif_filter[$i]);
            } else {
                $this->db->where(
                    $this->t_kecocokan['field']['id_atribut'],'a($i)');
            }

            $query_ = $this->db->get(
                $this->t_kecocokan['name']);

            foreach ($query_>result() as $val) {
                @$mat_normalisasi[$i][$j] = bcdiv($val->bobot,
                    $x[$j],4);
            }
        }
    }

    return $mat_normalisasi;
}

function matrik_normalisasi_terbobot() {
    $mat_normal = $this->matrik_normalisasi();
    $mY = array();
    for($i=1;$i<=$this->baris;$i++) {
        for($j=1;$j<=$this->kolom;$j++) {
            @$mY[$i][$j] = bcmul($this->bobot_p[$j-1],
                $mat_normal[$i][$j],4);
        }
    }

    return $mY;
}

```

```

function transpose_matrik() {
    $arr = $this->matrik_normalisasi_terbobot();
    $arrTmp[$this->kolom][$this->baris] = array();
    for($i = 1;$i<=$this->baris;$i++) {
        for($j = 1;$j<=$this->kolom;$j++) {
            $$arrTmp[$j][$i] = $arr[$i][$j];
        }
    }
    return $arrTmp;
}

function get_status_kriteria() {
    $this->db->select($this->
>t_kriteria['field']|'status');
    $query = $this->db->get($this->
>t_kriteria['name']);
    $this->status_kriteria = array();
    $i=0;
    foreach ($query->result() as $value) {
        $this->status_kriteria[$i]= $value->status;
        $i++;
    }

    return $this->status_kriteria;
}

function get_solusi_ideal_positif() {
    $st = $this->get_status_kriteria();
    $data = $this->transpose_matrik();
    $max = array();
    for($i=1;$i<=$this->kolom;$i++) {
        for($j=1;$j<=$this->baris;$j++) {
            if(@$st[$j]=="1") { //default code
                $max[$i] = max($data[$i]);
            } else {
                $max[$i] = min($data[$i]);
            }
        }
    }
    return $max;
}

function get_solusi_ideal_negatif() {
    $st = $this->get_status_kriteria();
    $data = $this->transpose_matrik();

    $min = array();

    for($i=1;$i<=$this->kolom;$i++) {
        for($j=1;$j<=$this->baris;$j++) {
            if(@$st[$j]=="1") {
                $min[$i] = min($data[$i]);
            } else {
                $min[$i] = max($data[$i]);
            }
        }
    }
    return $min;
}

```

```

function get_ranking() {
    $dMin = $this->get_jarak_atribut_negatif();
    $dPlus = $this->get_jarak_atribut_positif();

    $rankTmp[$this->baris] = array();
    for($i=1;$i<=$this->baris;$i++) {
        for($j=1;$j<=$this->kolom;$j++) {
            $v[$i] = @bcdiv($dMin[$i],
($dPlus[$i]+$dMin[$i]),4);
        }

        $dt = $this->get_detail_atribut($this->
>alternatif_filter[$i]);

        foreach ($dt->result() as $value) {
            $rankTmp[$i] = array(
                'id_atribut' => $this->alternatif_filter[$i],
                'nama' => $this->nama_atribut[$i],
                'menu' => $value->menu,
                'harga' => $value->harga,
                'alamat' => $value->alamat,
                'gambar' => $value->image,
                'letak' => $value->letak,
                'artikel' => $value->artikel,
                'value' => (float) $v[$i]
            );
        }

        asort($rankTmp);

        $C = $this->bubbleSort($rankTmp);

        return $C;
    }
}

function get_ranking_limit($limit,$index_begin='') {
    if(!isset($index_begin)) {
        $offset = $index_begin;
    } else {
        $offset = 1;
    }

    $rank_limit = array();
    $rek = $this->get_ranking();

    for($i=$offset;$i<=($limit+($offset-1));$i++) {
        @$rank_limit[$i] = $rek[$i];
    }

    return array_filter($rank_limit);
}
?>

```