



**PENGEMBANGAN SISTEM REKOMENDASI PRODUK
PERAWATAN KULIT BERBASIS WEB MENGGUNAKAN
METODE AHP**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Jojo Yeanesy Sinaga
NIM: 15515020111169



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2020**

PENGESAHAN

PENGEMBANGAN SISTEM REKOMENDASI PRODUK PERAWATAN KULIT BERBASIS
WEB MENGGUNAKAN METODE AHP

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Jojo Yeanesy Sinaga
NIM: 155150201111169

Skripsi ini telah diuji dan dinyatakan lulus pada
4 juni 2020

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Digitally signed by
faiz_amalia@ub.ac.id
DN:
cn=faiz_amalia@ub.ac.id
e=faiz_amalia@ub.ac.id
Reason: I approved this
document
Location:
Date: 2020-09-29
11:13+07:00

Faizatul Amalia, S.Pd., M.Pd.
NIK: 201309 860821 2 001

Dosen Pembimbing 2



Edy Santoso, S.Si., M.Kom.
NIP: 19740414 200312 1 004

Mengetahui
Ketua Jurusan Teknik Informatika



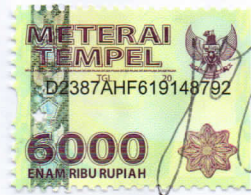
Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 4 Juni 2020



Jojo Yeanesy Sinaga

NIM: 155150201111169



ABSTRAK

Jojo Yeanesy Sinaga, Pengembangan Sistem Rekomendasi Produk Perawatan Kulit Berbasis Web Menggunakan Metode AHP.

Pembimbing: Faizatul Amalia, S.Pd., M.Pd. dan Edy Santoso, S.Si., M.Kom.

Kulit adalah bagian terluar dari tubuh manusia yang paling mudah diamati dan merupakan salah satu organ terbesar yang dimiliki manusia. Kulit memiliki beberapa fungsi diantaranya adalah mengatur suhu tubuh, melindungi jaringan yang ada di bawahnya dan sebagai indra peraba. Untuk menjalankan fungsi-fungsi tersebut, kulit harus dijaga dan dirawat dengan benar. Terutama pada kulit wajah. Selain dapat menarik perhatian, kulit wajah yang indah dan sehat juga dapat meningkatkan kepercayaan diri. Pemilihan produk perawatan kulit wajah yang tepat adalah kunci utama untuk mendapatkan hasil yang efektif. Namun, banyaknya produk perawatan kulit wajah yang dijual dapat mengakibatkan kesalahan dalam memilih dan memakai produk. Hal ini dapat mengakibatkan munculnya masalah kulit dan bisa menjadi lebih parah. Kulit bisa menjadi iritasi dan dehidrasi. Berdasarkan permasalahan tersebut, dikembangkanlah sebuah sistem rekomendasi produk perawatan kulit wajah berdasarkan tipe kulit dan permasalahan kulit. Untuk mendapatkan rekomendasi, metode yang digunakan adalah *Analytical Hierarchy Process* (AHP). Proses pengembangan sistem dilakukan dengan menggunakan metode *waterfall* dengan pendekatan Object-Oriented (OO) dan berbasis web. Pengujian pada kebutuhan fungsional dilakukan dengan pengujian unit, integrasi, dan validasi. Hasil yang didapat adalah 100% valid pada seluruh kasus yang diuji. Pengujian pada kebutuhan non fungsional yang dilakukan dengan pengujian *compatibility*. Hasil yang didapat menunjukkan bahwa sistem dapat diakses dari berbagai *browser* dan sistem operasi yang berbeda. Pengujian metode rekomendasi yang dilakukan adalah dengan pengujian akurasi dan menghasilkan nilai 66.67%. Hasil yang didapat menunjukkan sistem dapat berfungsi dengan baik sesuai dengan metode *Analitycal Hierarcy Process* (AHP).

Kata kunci: produk perawatan kulit wajah, rekomendasi, metode AHP.

ABSTRACT

Jojo Yeanesy Sinaga, Development of Web-Based Skin Care Product Recommendation System Using AHP Method.

Supervisors: Faizatul Amalia, S.Pd., M.Pd. dan Edy Santoso, S.Si., M.Kom.

Skin is the outermost part of the human body and one of the largest organs in humas. Skin has several functions including regulating body temperature, protecting the tissue underneath and as a sense of touch. To carry out these functions, the skin must be properly guarded and cared for. Especially on facial skin. Besides being able to attract attention, beautiful and healthy facial skin can also increase self-confidence. The selection of the right facial skin care products is the main key to getting effective results. However, the large number of facial skin care products being sold can result in mistakes in choosing and using the product. This can lead to skin problems and they can get worse. Skin can become irritated and dehydrated. Based on these problems, a system of recommendations for skin care products was developed based on skin types and web-based skin problems. To get recommendations, the method used is Analytical Hierarchy Process (AHP). The system development process is carried out using the waterfall method with an Object-Oriented (OO) approach and web-based. Testing on functional requirements is carried out by unit testing and validation. The results obtained are 100% valid in all tested cases. Testing on non-functional requirements is carried out with compatibility testing. The results obtained indicate that the system can be accessed from various browsers and different operating systems. Testing the recommendation method performed with accuracy testing produces a value of 66.67%. The results obtained indicate the system can function properly according to the Analytical Hierarchy Process (AHP) method.

Keyword: facial skin care product, recommendation, AHP method.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Landasan Teori	6
2.2.1 Perawatan Kulit	6
2.2.2 Sistem Rekomendasi	8
2.2.3 CodeIgniter	8
2.2.4 Konsep MVC	8
2.2.5 Metode AHP	9
2.2.6 <i>Waterfall</i> Model	12
2.2.7 UML	13
2.2.8 Bahasa Pemrograman	15
2.2.9 Pengujian Perangkat Lunak	16
BAB 3 METODOLOGI PENELITIAN	19
3.1 Studi Literatur	19



3.2	Rekayasa Kebutuhan.....	20
3.3	Perancangan Sistem.....	20
3.4	Implementasi Sistem.....	20
3.5	Pengujian Sistem.....	21
3.6	Kesimpulan dan Saran.....	21
BAB 4 REKAYASA KEBUTUHAN.....		22
4.1	Rekayasa Kebutuhan.....	22
4.1.1	Deskripsi Sistem.....	22
4.1.2	Identifikasi Aktor.....	22
4.2	Spesifikasi Kebutuhan.....	22
4.2.1	Aturan Penomoran.....	22
4.2.2	Kebutuhan Fungsional.....	23
4.2.3	Kebutuhan Non-Fungsional.....	29
4.3	Pemodelan Kebutuhan.....	30
4.3.1	<i>Use Case Diagram</i>	30
4.3.2	<i>Use Case Scenario</i>	30
BAB 5 PERANCANGAN DAN IMPLEMENTASI.....		47
5.1	Perancangan.....	47
5.1.1	Perancangan Arsitektur Sistem.....	47
5.1.2	Perancangan <i>Sequence Diagram</i>	47
5.1.3	Perancangan <i>Class Diagram</i>	49
5.1.4	Perancangan Data.....	51
5.1.5	Perancangan Komponen.....	51
5.1.6	Perancangan Antarmuka.....	52
5.2	Implementasi.....	58
5.2.1	Spesifikasi Sistem.....	58
5.2.2	Batasan Implementasi.....	59
5.2.3	Implementasi Data.....	59
5.2.4	Implementasi Kode Program.....	63
5.2.5	Implementasi Antarmuka.....	66
BAB 6 PENGUJIAN.....		71
6.1	Pengujian Unit (Fungsional).....	71



6.1.1 Method create() Klas C_Register	71
6.1.2 Method tambah_produk() Klas C_Admin	72
6.1.3 Method simpanProfile() Klas C_Pengguna	74
6.2 Pengujian Integrasi (Fungsional)	76
6.2.1 Method tambah_produk() Klas C_Admin	76
6.3 Pengujian Validasi (Fungsional)	80
6.3.1 Validasi Login	80
6.3.2 Validasi Registrasi Akun	81
6.3.3 Validasi Logout	84
6.3.4 Pengujian Validasi Melihat Data Diri	84
6.3.5 Validasi Mengubah Data Diri	84
6.3.6 Validasi Melihat Halaman Rekomendasi	85
6.3.7 Validasi Mengisi Data Rekomendasi	85
6.3.8 Validasi Mengisi Nilai Perbandingan	87
6.3.9 Validasi Melihat Daftar Rekomendasi	88
6.3.10 Validasi Menulis Ulasan Produk	89
6.3.11 Validasi Menghapus Ulasan Produk	90
6.3.12 Validasi Menulis Ulasan Artikel	91
6.3.13 Validasi Menghapus Ulasan Artikel	92
6.3.14 Validasi Menyukai Artikel	92
6.3.15 Validasi Membatalkan Menyukai Artikel	92
6.3.16 Validasi Melihat Daftar Pengguna	93
6.3.17 Validasi Menambah Data Produk	93
6.3.18 Validasi Melihat Daftar Produk	95
6.3.19 Validasi Mengubah Data Produk	95
6.3.20 Validasi Menghapus Data Produk	96
6.3.21 Validasi Menambah Data Artikel	96
6.3.22 Validasi Melihat Daftar Artikel	98
6.3.23 Validasi Mengubah Data Artikel	98
6.3.24 Validasi Menghapus Data Artikel	99
6.3.25 Validasi Melihat Produk	99
6.3.26 Validasi Melihat Artikel	100



6.3.27 Validasi Mencari Produk.....	100
6.3.28 Validasi Mencari Artikel.....	101
6.3.29 Validasi Melihat Halaman <i>About Us</i>	102
6.3.30 Validasi Melihat Halaman FAQ.....	103
6.3.31 Validasi Melihat Data Diri <i>Member</i> lain.....	103
6.4 Pengujian <i>Compatibility</i> (Non-Fungsional).....	104
6.5 Pengujian Akurasi.....	104
BAB 7 PENUTUP.....	107
7.1 Kesimpulan.....	107
7.2 Saran.....	108
DAFTAR REFERENSI.....	109
LAMPIRAN A.....	111
LAMPIRAN B.....	117



DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	5
Tabel 2.2 Nilai Skala Perbandingan Berpasangan.....	10
Tabel 2.3 Notasi <i>Use Case Diagram</i>	13
Tabel 2.4 Notasi <i>Sequence Diagram</i>	14
Tabel 2.5 Notasi <i>Class Diagram</i>	15
Tabel 4.1 Identifikasi Aktor.....	22
Tabel 4.2 Daftar Kebutuhan Fungsional.....	23
Tabel 4.3 Daftar Kebutuhan Non-Fungsional	30
Tabel 4.4 <i>Login</i>	30
Tabel 4.5 Registrasi Akun.....	31
Tabel 4.6 <i>Logout</i>	32
Tabel 4.7 Melihat Data Diri	32
Tabel 4.8 Mengubah Data Diri	32
Tabel 4.9 Melihat Halaman Rekomendasi	33
Tabel 4.10 Mengisi Data Rekomendasi.....	33
Tabel 4.11 Mengisi Nilai Perbandingan	34
Tabel 4.12 Melihat Daftar Rekomendasi	35
Tabel 4.13 Menulis Ulasan Produk	35
Tabel 4.14 Menghapus Ulasan Produk.....	36
Tabel 4.15 Menulis Ulasan Artikel	36
Tabel 4.16 Menghapus Ulasan Artikel	36
Tabel 4.17 Menyukai Artikel.....	37
Tabel 4.18 Membatalkan Menyukai Artikel.....	37
Tabel 4.19 Melihat Daftar Pengguna	38
Tabel 4.20 Menambah Data Produk	38
Tabel 4.21 Melihat Daftar Produk.....	39
Tabel 4.22 Mengubah Data Produk	39
Tabel 4.23 Menghapus Data Produk.....	40
Tabel 4.24 Menambah Artikel	40
Tabel 4.25 Melihat Daftar Produk.....	41



Tabel 4.26 Mengubah Artikel.....	41
Tabel 4.27 Menghapus Artikel.....	42
Tabel 4.28 Melihat Produk.....	42
Tabel 4.29 Melihat Artikel.....	43
Tabel 4.30 Mencari Produk.....	43
Tabel 4.31 Mencari Artikel.....	43
Tabel 4.32 Melihat Halaman About Us.....	44
Tabel 4.33 Melihat Halaman FAQ.....	44
Tabel 4.34 Melihat Data Diri <i>Member</i> lain.....	45
Tabel 5.1 Algoritme <i>Method</i> create.....	51
Tabel 5.2 Algoritme <i>Method</i> tambah_produk.....	52
Tabel 5.3 Algoritme <i>Method</i> simpanProfile.....	52
Tabel 5.4 Penjelasan Antarmuka Halaman <i>login</i>	53
Tabel 5.5 Penjelasan Antarmuka Halaman Utama Sistem.....	54
Tabel 5.6 Penjelasan Antarmuka Halaman <i>Find Recommendation</i>	55
Tabel 5.7 Penjelasan Antarmuka Halaman Utama Admin.....	56
Tabel 5.8 Penjelasan Antarmuka Tambah Produk.....	57
Tabel 5.9 Spesifikasi Perangkat Keras.....	58
Tabel 5.10 Spesifikasi Perangkat Lunak.....	59
Tabel 5.11 Implementasi Basis Data Tabel Admin.....	60
Tabel 5.12 Implementasi Basis Data Tabel Pengguna.....	60
Tabel 5.13 Implementasi Basis Data Tabel Artikel.....	61
Tabel 5.14 Implementasi Basis Data Tabel Komentar.....	61
Tabel 5.15 Implementasi Basis Data Tabel Suka Artikel.....	62
Tabel 5.16 Implementasi Basis Data Tabel Produk.....	62
Tabel 5.17 Implementasi Basis Data Tabel Komentar.....	63
Tabel 5.18 Kode Program <i>Method</i> <i>create()</i> Klas C_Register.....	64
Tabel 5.19 Kode Program <i>Method</i> <i>tambah_produk()</i> Klas C_Admin.....	64
Tabel 5.20 Kode Program <i>Method</i> <i>simpanProfile()</i> Klas C_Pengguna.....	65
Tabel 6.1 Algoritme <i>Method</i> <i>create()</i>	71
Tabel 6.2 Kasus Uji <i>Method</i> <i>create()</i>	72
Tabel 6.3 Algoritme <i>Method</i> <i>tambah_produk()</i>	72



Tabel 6.4 Kasus Uji <i>Method</i> tambah_produk()	74
Tabel 6.5 Algoritme <i>Method</i> simpanProfile()	74
Tabel 6.6 Kasus Uji <i>Method</i> simpanProfile()	75
Tabel 6.7 Algoritme <i>Method</i> tambah_produk()	76
Tabel 6.8 Algoritme <i>Method</i> create_product()	77
Tabel 6.9 Algoritme Gabungan <i>Method</i> tambah_produk dan create_product()	78
Tabel 6.10 Kasus Uji <i>Method</i> tambah_produk()	80
Tabel 6.11 Validasi <i>Login</i>	80
Tabel 6.12 Validasi <i>Login Alternative</i> Satu	81
Tabel 6.13 Validasi <i>Login Alternative</i> Dua	81
Tabel 6.14 Validasi Registrasi Akun	81
Tabel 6.15 Validasi Registrasi Akun <i>Alternative</i> Satu	82
Tabel 6.16 Validasi Registrasi Akun <i>Alternative</i> Dua	82
Tabel 6.17 Validasi Registrasi Akun <i>Alternative</i> Tiga	83
Tabel 6.18 Validasi <i>Logout</i>	84
Tabel 6.19 Validasi Melihat Data Diri	84
Tabel 6.20 Validasi Mengubah Data Diri	84
Tabel 6.21 Validasi Melihat Halaman Rekomendasi	85
Tabel 6.22 Validasi Mengisi Data Rekomendasi	85
Tabel 6.23 Validasi Mengisi Data Rekomendasi <i>Alternative</i> Dua	86
Tabel 6.24 Validasi Mengisi Data Rekomendasi <i>Alternative</i> Tiga	86
Tabel 6.25 Validasi Mengisi Data Rekomendasi <i>Alternative</i> Empat	87
Tabel 6.26 Validasi Mengisi Nilai Perbandingan	88
Tabel 6.27 Validasi Melihat Daftar Rekomendasi	88
Tabel 6.28 Pengujian Validasi Melihat Halaman Rekomendasi <i>Alternative</i> Satu	89
Tabel 6.29 Validasi Menulis Ulasan Produk	90
Tabel 6.30 Validasi Menulis Ulasan Produk <i>Alternative</i> Satu	90
Tabel 6.31 Validasi Menghapus Ulasan Produk	90
Tabel 6.32 Validasi Menulis Ulasan Artikel	91
Tabel 6.33 Validasi Menulis Ulasan Artikel <i>Alternative</i> Satu	91
Tabel 6.34 Validasi Menghapus Ulasan Artikel	92
Tabel 6.35 Validasi Menyukai Artikel	92



Tabel 6.36 Validasi Membatalkan Menyukai Artikel.....	93
Tabel 6.37 Validasi Melihat Daftar Pengguna.....	93
Tabel 6.38 Validasi Menambah Data Produk.....	93
Tabel 6.39 Validasi Menambah Data Produk <i>Alternative</i> Satu.....	94
Tabel 6.40 Validasi Melihat Daftar Produk.....	95
Tabel 6.41 Validasi Mengubah Data Produk.....	95
Tabel 6.42 Validasi Menghapus Data Produk.....	96
Tabel 6.43 Validasi Menambah Data Artikel.....	97
Tabel 6.44 Validasi Menambah Data Artikel <i>Alternative</i> Satu.....	97
Tabel 6.45 Validasi Melihat Daftar Artikel.....	98
Tabel 6.46 Validasi Mengubah Data Artikel.....	98
Tabel 6.47 Validasi Menghapus Data Artikel.....	99
Tabel 6.48 Validasi Melihat Produk.....	100
Tabel 6.49 Validasi Melihat Artikel.....	100
Tabel 6.50 Validasi Mencari Produk.....	100
Tabel 6.51 Validasi Mencari Produk <i>Alternative</i> Satu.....	101
Tabel 6.52 Validasi Mencari Artikel <i>Alternative</i> Dua.....	101
Tabel 6.53 Validasi Mencari Artikel.....	101
Tabel 6.54 Validasi Mencari Artikel <i>Alternative</i> Satu.....	102
Tabel 6.55 Validasi Mencari Artikel <i>Alternative</i> Dua.....	102
Tabel 6.56 Validasi Melihat Halaman <i>About Us</i>	103
Tabel 6.57 Validasi Melihat Halaman FAQ.....	103
Tabel 6.58 Validasi Melihat Data Diri <i>Member</i> lain.....	103
Tabel 6.59 Perbandingan Hasil Pakar dan Sistem.....	105
Tabel 6.60 Perbandingan Hasil Pakar dan Sistem.....	105



DAFTAR GAMBAR

Gambar 2.1 MVC.....	9
Gambar 2.2 Struktur Hirarki.....	9
Gambar 2.3 <i>Waterfall Model</i>	12
Gambar 2.4 Indikator Masalah <i>Compatibility SortSite</i>	18
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	19
Gambar 4.1 Aturan Penomoran.....	23
Gambar 4.2 <i>Use Case Diagram</i>	46
Gambar 5.1 Arsitektur Sistem Rekomendasi Pemilihan Produk Skincare.....	47
Gambar 5.2 <i>Sequence Diagram</i> Registrasi Akun.....	48
Gambar 5.3 <i>Sequence Diagram</i> Melihat Daftar Rekomendasi.....	48
Gambar 5.4 <i>Sequence Diagram</i> Mencari Produk.....	49
Gambar 5.5 Perancangan Umum <i>Class Diagram</i>	49
Gambar 5.6 Perancangan Klas <i>Cl_Controller</i>	50
Gambar 5.7 Perancangan Klas <i>Cl_Model</i>	50
Gambar 5.8 <i>Entity Relationship Diagram</i> (ERD).....	51
Gambar 5.9 Antarmuka Halaman <i>login</i>	53
Gambar 5.10 Antarmuka Halaman Utama Sistem.....	54
Gambar 5.11 Antarmuka Halaman <i>Find Recommendation</i>	55
Gambar 5.12 Antarmuka Halaman Utama Admin.....	56
Gambar 5.13 Antarmuka Halaman Tambah Produk.....	57
Gambar 5.14 Implementasi Basis Data.....	60
Gambar 5.15 Antarmuka Halaman <i>login</i>	67
Gambar 5.16 Antarmuka Halaman Utama Sistem.....	67
Gambar 5.17 Antarmuka Halaman Utama Admin.....	68
Gambar 5.18 Antarmuka Halaman <i>Find Recommendation</i>	69
Gambar 5.19 Antarmuka Halaman Tambah Produk.....	70
Gambar 6.1 <i>Flow Graph Method</i> create().....	72
Gambar 6.2 <i>Flow Graph Method</i> tambah_produk().....	73
Gambar 6.3 <i>Flow Graph Method</i> simpanProfile().....	75
Gambar 6.4 <i>Flow Graph Method</i> tambah_produk.....	77



Gambar 6.5 *Flow Graph Method* create_product()..... 78

Gambar 6.6 *Flow Graph* Gabungan *Method* tambah_produk() dan create_product()..... 79

Gambar 6.7 Hasil Pengujian *Compatibility* Menggunakan Aplikasi *Sortsite*..... 104

**DAFTAR LAMPIRAN**

LAMPIRAN A.....	111
A.1 Wawancara.....	111
A.1.1 Latar Belakang.....	111
A.1.2 Maksud dan Tujuan.....	111
A.1.3 Topik Wawancara.....	111
A.1.4 Waktu Pelaksanaan.....	111
A.1.5 Narasumber dan Pewawancara.....	111
A.1.6 Hasil Wawancara.....	112
A.2 Hasil Kuesioner.....	115
LAMPIRAN B.....	117
B.1 Contoh Perhitungan Manual AHP.....	117



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kulit adalah bagian terluar dari tubuh manusia yang paling mudah diamati dan merupakan salah satu organ terbesar yang dimiliki manusia. Kulit memiliki banyak fungsi. Fungsi kulit di antaranya adalah mengatur suhu tubuh, melindungi jaringan yang ada di bawahnya dari kerusakan akibat sinar matahari, tempat pembentukan perubahan pro-vitamin D menjadi vitamin D, dan sebagai indra peraba (Djuanda, 2007).

Untuk menjalankan fungsi-fungsinya, kulit harus dijaga dan dirawat dengan benar. Terutama pada bagian kulit wajah. Wajah adalah bagian tubuh yang pertama kali dilihat saat sedang berjumpa. Oleh karena itu, kulit wajah harus dirawat kesehatan dan keindahannya. Selain dapat menarik perhatian, kulit wajah yang sehat dan indah juga dapat meningkatkan kepercayaan diri.

Menurut Irawati (2013), kulit wajah manusia dikelompokkan menjadi lima tipe. Tipe kulit normal, tipe kulit berminyak, tipe kulit kering, tipe kulit kombinasi, dan tipe kulit *sensitive*. Karena memiliki struktur dan karakteristik yang berbeda, perawatan kulit dibedakan menjadi 2 (dua) bagian. Perawatan kulit sehari-hari (secara sederhana) dan perawatan kulit secara *periodic* (secara lengkap). Perawatan kulit sehari-hari dapat dilakukan sendiri pada pagi hari dan malam hari (sebelum tidur). Perawatan kulit sehari-hari di antaranya adalah pembersihan (*cleansing*), penyegaran (*toning*), dan pelembapan (*moisturizing*). Perawatan kulit secara *periodic* dapat dilakukan 1 (satu) atau 2 (dua) bulan sekali. Perawatan kulit secara *periodic* dilakukan oleh ahli kecantikan di klinik kecantikan. Perawatan kulit secara *periodic* di antaranya adalah pijatan (*massage*), pengelupasan sel tanduk (*skin peeling*), dan pengeluaran komedo (Tresna, 2010). Berdasarkan data dari hasil survey kuesioner yang disebar, perawatan kulit sehari-hari (sederhana) dengan menggunakan produk lebih dipilih daripada perawatan kulit secara *periodic* (secara bertahap). Selain karena biaya yang terjangkau, produk perawatan kulit wajah juga dapat dengan mudah ditemukan.

Produk perawatan kulit wajah dapat dengan mudah ditemukan di berbagai pusat perbelanjaan atau *mall*. Contohnya di Guardian Indonesia, Century Healthcare, dan Watsons Indonesia. Produk yang tersedia pun beragam jenis dan mereknya. Ada produk *face wash*, produk *toner*, produk serum, produk *essence*, produk pelembap, produk masker, produk *sunscreen*, dan produk *eye cream*. Namun, beragamnya produk yang tersedia mengakibatkan kesalahan dalam pembelian dan pemakaian. Memakai produk yang tidak tepat dapat menyebabkan kulit menjadi kusam, iritasi, dehidrasi, atau berjerawat.

Berdasarkan data dari hasil wawancara dengan pakar, pemakaian produk perawatan kulit wajah yang tepat adalah dengan menyesuakannya dengan tipe kulit dan permasalahan kulit yang dialami. Urutan pemakaian juga harus diperhatikan agar produk dapat bekerja lebih efektif. Dokter kulit kosmetik di London, dr. Sam Bunting dalam artikel hellosehat.com juga berpendapat bahwa,



urutan pemakaian produk perawatan kulit yang salah dapat mengurangi efektivitas produk yang digunakan. Kunci utama dalam memakai produk perawatan kulit adalah menggunakan produk dengan tekstur dari yang paling ringan ke yang paling berat. Melakukan *cleansing* dengan menggunakan *face wash* adalah langkah pertama dilakukan sebelum melanjutkan tahap lainnya. Tahap kedua adalah mengaplikasikan *toner*. Tahap ketiga adalah mengaplikasikan serum. Tahap keempat adalah mengaplikasikan pelembap dan tahap terakhir adalah mengaplikasikan *sunscreen* (Andini, 2019).

Berdasarkan permasalahan di atas, dikembangkanlah sebuah sistem rekomendasi produk perawatan kulit wajah berdasarkan tipe kulit dan permasalahan kulit dengan menggunakan metode. Metode yang digunakan untuk menemukan rekomendasi adalah AHP (*Analytical Hierarchy Process*). Metode ini menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki (Saaty, 2000).

Sistem rekomendasi yang dikembangkan dalam penelitian ini adalah berbasis web. Sistem ini dapat digunakan di platform apapun selama terhubung dengan jaringan internet. Sistem ini tidak hanya bisa diakses melalui *personal computer* (PC) saja, tetapi juga dapat diakses melalui telepon gengam sehingga bisa digunakan kapan dan dimana saja.

Beberapa penelitian telah dilakukan dengan menggunakan metode AHP (*Analytical Hierarchy Process*). Salah satunya adalah penelitian yang dilakukan oleh Fajar Agustini (2018) dengan menggunakan metode AHP (*Analytical Hierarchy Process*). Metode tersebut digunakan untuk memilih kosmetik yang sesuai untuk siswi SMA. Hasil dari penelitian tersebut menempatkan Wardah sebagai urutan utama. Maka dari itu, penulis berharap dengan diimplementasikannya metode AHP (*Analytical Hierarchy Process*) dalam pemilihan produk perawatan kulit dapat membantu pengguna sistem menghindari kesalahan dalam membeli dan memakai produk.

1.2 Rumusan Masalah

Adapun rumusan masalah yang diperoleh dari penjelasan latar belakang yang telah disampaikan adalah sebagai berikut:

1. Bagaimana hasil analisis dan spesifikasi kebutuhan sistem rekomendasi produk perawatan kulit berbasis web menggunakan metode AHP?
2. Bagaimana hasil perancangan pada sistem rekomendasi produk perawatan kulit berbasis web menggunakan metode AHP?
3. Bagaimana hasil implementasi sistem rekomendasi produk perawatan kulit berbasis web menggunakan metode AHP?
4. Bagaimana hasil pengujian kebutuhan fungsional dan non fungsional pada sistem rekomendasi produk perawatan kulit berbasis web menggunakan metode AHP?



1.3 Tujuan

Pada rumusan masalah yang disampaikan di atas, adapun tujuan penelitian pengembangan sistem ini adalah sebagai berikut:

1. Untuk mengetahui hasil analisis dan spesifikasi kebutuhan sistem rekomendasi produk perawatan kulit berbasis web menggunakan metode AHP.
2. Untuk mengetahui hasil perancangan sistem rekomendasi produk perawatan kulit berbasis web menggunakan metode AHP.
3. Untuk mengetahui hasil implementasi sistem rekomendasi produk perawatan kulit berbasis web menggunakan metode AHP.
4. Untuk mengetahui hasil pengujian sistem rekomendasi produk perawatan kulit berbasis web menggunakan metode AHP.

1.4 Manfaat

1. Manfaat untuk pengguna sistem
Sistem dapat membantu para pengguna untuk mendapatkan rekomendasi produk-produk perawatan kulit yang sesuai dengan tipe kulit dan permasalahan kulit yang dialami.
2. Bagi Penulis
Mengaplikasikan ilmu yang didapat di bangku kuliah.

1.5 Batasan Masalah

Penelitian ini tentu memiliki batasan, sehingga batasan masalah yang diuraikan adalah sebagai berikut:

1. Sistem rekomendasi produk perawatan kulit ini dibangun pada platform *website*.
2. Metode pengembangan perangkat lunak menggunakan pengembangan model *waterfall*.
3. Sistem rekomendasi produk perawatan kulit dikembangkan menggunakan *framework* CodeIgniter.
4. Metode yang dipakai dalam penentuan rekomendasi adalah metode AHP.
5. Kriteria yang digunakan dalam penentuan rekomendasi adalah harga, kualitas, dan bahan.
6. Observasi dilakukan di Guardian Indonesia, Century Healthcare, dan Watsons Indonesia.

1.6 Sistematika Pembahasan

Sistematika pembahasan digunakan pada penelitian agar tujuan yang diharapkan tercapai adalah:

**BAB 1 : PENDAHULUAN**

Menjelaskan mengenai latar belakang yang melandasi dibuatnya penelitian, rumusan masalah, tujuan, manfaat, batasan masalah serta sistematika pembahasan penelitian.

BAB 2 : LANDASAN KEPUSTAKAAN

Menjelaskan mengenai teori-teori yang terkait atau relevan untuk penelitian ini. Serta membahas penelitian-penelitian yang sudah ada.

BAB 3 : METODOLOGI PENELITIAN

Menjelaskan alur penelitian yang dimulai dengan studi literatur sampai dengan kesimpulan beserta saran.

BAB 4 : REKAYASA KEBUTUHAN

Menjelaskan tentang kebutuhan yang didapatkan dari wawancara dan observasi. Analisis dilakukan untuk mengidentifikasi aktor, kebutuhan fungsionalitas, dan kebutuhan non-fungsionalitas yang ada pada sistem.

BAB 5 : PERANCANGAN DAN IMPLEMENTASI

Menjelaskan tentang perancangan dan implementasi sistem. Perancangan dilakukan berdasarkan analisis kebutuhan yang didapatkan melalui wawancara dan observasi. Sedangkan implementasi sistem dibuat berdasarkan perancangan sistem.

BAB 6 : PENGUJIAN

Menjelaskan tentang hasil dari pengujian yang telah dilakukan terhadap sistem yang telah dibuat.

BAB 7 : PENUTUP

Menjelaskan tentang kesimpulan yang menjawab rumusan masalah dan saran yang berguna untuk pembangunan sistem selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka menjabarkan beberapa penelitian yang telah dilakukan dan dijadikan sebagai referensi dalam penulisan skripsi ini.

Penelitian pertama yang dilakukan oleh I Dewa Ayu Eka Yuliani (2016) dengan judul "Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik dengan Metode *Analytical Hierarchy Process*". Penelitian ini bertujuan untuk membantu mengambil keputusan dalam memutuskan karyawan terbaik dalam jangka waktu tertentu. Kriteria yang digunakan dalam penelitian ini adalah kebersihan, keramah tamahan, ketepatan, perawatan peralatan, kualitas produk, dan kecepatan.

Penelitian selanjutnya dilakukan oleh Bhima Arya Tristya Nisswara (2018) dengan judul "Rekomendasi Pemilihan Paket *Personal Computer* Menggunakan Metode AHP-TOPSIS". Penelitian ini bertujuan untuk membantu memilih hal apa saja yang perlu diperhatikan dalam melakukan perakitan sebuah komputer. Kriteria yang digunakan adalah harga, *processor*, VGA, RAM, HDD, SSD, dan tipe RAM.

Penelitian dengan topik yang sama dilakukan juga oleh Ireneusz Miciula dan Joanna Nowakowska-Grunt (2019) dengan judul "*Using the AHP method to select an energy supplier for household in Poland*". Penelitian ini bertujuan untuk membantu pengambilan keputusan memilih pemasok energi untuk rumah tangga di Polandia. Kriteria yang digunakan dalam penelitian ini adalah tarif rumah, tarif siang dan malam, jumlah energi yang dihasilkan dari pembaharuan energi, dan saham.

Hasil dari setiap penelitian dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

Judul	Penulis	Metode	Hasil Penelitian
Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik dengan Metode <i>Analytical Hierarchy Process</i>	(Yuliani, 2016)	AHP (<i>Analytical Hierarchy Process</i>)	Sistem yang dapat membantu mengambil keputusan dalam pemilihan karyawan terbaik berdasarkan kriteria kebersihan, keramah tamahan, ketepatan, perawatan peralatan, kualitas produk, kecepatan.



Tabel 2.1 Kajian Pustaka (Lanjutan)

Judul	Penulis	Metode	Hasil Penelitian
Rekomendasi Pemilihan Paket <i>Personal Computer</i> Menggunakan Metode AHP-TOPSIS	(Niswara, 2018)	AHP-TOPSIS	Sistem yang dapat membantu memilih hal apa saja yang perlu diperhatikan dalam melakukan perakitan sebuah komputer berdasarkan kriteria harga, <i>processor</i> , VGA, RAM, HDD, SSD, dan tipe RAM.
<i>Using the AHP method to select an energy supplier for household in Poland</i>	(Miciula & Nowakowska-Grunt, 2019)	AHP (<i>Analytical Hierarchy Process</i>)	Sistem yang dapat membantu mengambil keputusan dalam pemilihan pemasok energi dalam rumah tangga di Polandia berdasarkan kriteria tarif rumah, tarif siang dan malam, jumlah energi yang dihasilkan dari pembaharuan energi, dan saham.

2.2 Landasan Teori

2.2.1 Perawatan Kulit

2.2.1.1 Kulit

Kulit adalah bagian terluar dari tubuh manusia yang paling mudah diamati dan merupakan salah satu organ terbesar. Fungsi kulit adalah untuk melindungi manusia dari lingkungan berbahaya dan mempertahankan lingkungan internal yang homeostatis. Kulit melindungi jaringan yang ada di bawahnya dari bahan kimia, sinar matahari, dan bakteri. Kulit dapat menjadi indikator penting dalam menentukan kesehatan. Bertindak sebagai isolator dan membran, kulit menjaga lingkungan di dalam tubuh pada untuk tetap berada di suhu yang konstan (Parrish, 1975).

Kulit memiliki struktur yang didalamnya terdapat sel-sel yang memiliki fungsi tersendiri serta bermacam-macam tipenya. Struktur kulit terdiri dari dua lapisan utama. Lapisan epidermis dan lapisan dermis. Lapisan epidermis adalah lapisan terluar yang mengalami kontak langsung dengan lingkungan luar tubuh. Pada lapisan epidermis, terdapat sel-sel yang terus membelah dan membentuk sel-sel



baru. Lapisan dermis adalah lapisan yang berada di bawah lapisan epidermis. Lapisan dermis lebih tebal daripada lapisan epidermis. Lapisan dermis adalah tempat melekatnya pembuluh darah, saraf, dan folikel rambut. Lapisan dermis menyediakan nutrisi dan oksigen. Lapisan dermis sebagian besar terdiri dari jaringan ikat (Penzer & Ersser, 2010).

Menurut Irawati (2013), kulit dikelompokkan menjadi lima tipe. Tipe kulit normal, tipe kulit berminyak, tipe kulit kering, tipe kulit kombinasi, dan tipe kulit *sensitive*. Tipe kulit normal adalah kulit dengan permukaan yang halus dan lembut. Kadar minyak (*sebum*) yang dikeluarkan seimbang. Tipe kulit berminyak adalah kulit yang aktif dan produktif mengeluarkan minyak (*sebum*). Tipe kulit kering adalah kulit yang memiliki tekstur yang terlihat dan cenderung terlihat keriput. Kadar minyak (*sebum*) yang dikeluarkan sangat rendah. Tipe kulit kombinasi adalah gabungan tipe kulit kering dan berminyak. Ada bagian tertentu yang mengeluarkan minyak (*sebum*), sedangkan di bagian lain tidak. Tipe kulit *sensitive* adalah kulit yang cepat bereaksi terhadap alergen, serta mudah iritasi dan terluka.

2.2.1.2 Perawatan Kulit

Perawatan kulit adalah aktivitas yang dilakukan untuk menjaga kesehatan kulit dan meningkatkan penampilan. Perawatan kulit dapat mencakup nutrisi bagi kulit, menghindari dampak dari paparan sinar matahari yang berlebihan, dan penggunaan pelembap yang tepat. Aktivitas yang dapat meningkatkan penampilan adalah dengan menggunakan kosmetik, melakukan exfoliasi untuk menghilangkan sel kulit mati, menggunakan *botox* untuk mengurangi kerutan, melakukan *filler* untuk mengisi bagian yang tidak rata pada kulit, melakukan *laser* untuk menghilangkan kerutan, melakukan *peeling* untuk menghilangkan sel kulit mati, dan melakukan terapi retinol untuk mengurangi garis-garis halus dan kerutan (Penzer & Ersser, 2010).

Kulit memiliki struktur dan karakteristik yang berbeda. Oleh karena itu, perawatan kulit dapat dibedakan menjadi perawatan kulit sehari-hari (secara sederhana) dan perawatan kulit secara *periodic* (secara lengkap). Perawatan kulit sehari-hari dapat dilakukan sendiri pada pagi hari dan malam hari (sebelum tidur). Perawatan kulit sehari-hari di antaranya adalah pembersihan (*cleansing*), penyegaran (*toning*), dan pelembapan (*moisturizing*). Perawatan kulit secara *periodic* dapat dilakukan 1 (satu) bulan sekali untuk usia di bawah 35 tahun dan 2 (dua) bulan sekali untuk usia di atas 35 tahun. Perawatan kulit secara *periodic* dilakukan oleh ahli kecantikan karena dalam perawatan ini memerlukan keterampilan. Perawatan kulit secara *periodic* di antaranya adalah pemijatan (*massage*), pengelupasan sel tanduk (*skin peeling*), dan pengeluaran komedo (Tresna, 2010).



2.2.2 Sistem Rekomendasi

2.2.2.1 Sistem

Menurut Edhy Sutanta (2003), sistem adalah kumpulan dari elemen-elemen yang ada dan saling bekerja sama antara satu dengan yang lain dalam membentuk suatu kesatuan untuk melakukan suatu tujuan tertentu.

Menurut Romney dan Steinbart (2015), sistem diartikan sebagai kumpulan yang saling berhubungan yang terdiri dari dua komponen atau lebih dan saling berinteraksi dalam mencapai tujuan tertentu.

Berdasarkan definisi yang dijelaskan diatas, dapat diambil kesimpulan bahwa sistem adalah kumpulan dari dua komponen atau lebih yang saling bekerja sama untuk mencapai suatu tujuan tertentu.

2.2.2.2 Sistem Rekomendasi

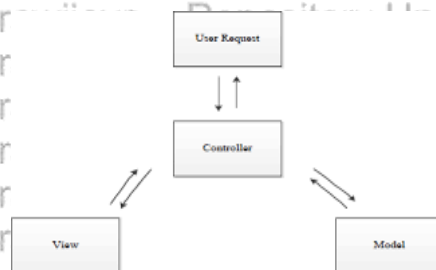
Sistem rekomendasi diimplementasikan sebagai sistem yang memberikan prediksi terhadap produk tertentu. Sistem rekomendasi merupakan suatu sistem untuk membantu para konsumen dalam menilai suatu produk dari informasi yang disarankan oleh pihak lain yang telah menggunakan sehingga lebih efisien dalam memilih produk tertentu (Kurniawan, 2016). Pada mekanisme sistem rekomendasi yaitu dilakukan dengan konsumen akan menerima data mengenai produk yang dipilih lalu memberikan *feedback* berupa saran atau rekomendasi untuk keperluan tersebut (Marthasari, Azhar and Puspitaningrum, 2015). Dengan diterapkannya sistem rekomendasi, maka dapat menambahkan kepercayaan konsumen terhadap produk yang diinginkan.

2.2.3 CodeIgniter

Menurut Upton & Blanco (2009), CodeIgniter adalah *framework* yang menerapkan pola MVC (*Model-View-Controller*) sehingga mempermudah dalam penggunaannya. Banyak keuntungan yang didapat dengan menggunakan CodeIgniter di antaranya adalah gratis, menggunakan pola MVC (*Model-View-Controller*), URL yang sederhana, dan memiliki paket *library* yang lengkap.

2.2.4 Konsep MVC

CodeIgniter adalah *framework* PHP yang menerapkan pola MVC (*Model-View-Controller*). Dengan MVC maka memungkinkan pemisahan antara data (*model*), tampilan (*view*), dan proses (*control*). Adapun alur program dengan menggunakan *framework* Codeigniter dapat dilihat pada Gambar 2.1.



Gambar 2.1 MVC

Sumber: (Hakim, 2010)

Gambar di atas menjelaskan bahwa ketika datang permintaan, maka akan ditangani oleh *controller*, kemudian *controller* akan memanggil *model* jika memang diperlukan operasi *database*. Kemudian *model* akan mengembalikan hasil ke *controller*. Selanjutnya *controller* akan memanggil *view* yang tepat dan mengkombinasikannya dengan hasil *query model*. Hasil akhir yang dilakukan oleh operasi di atas akan ditampilkan di *browser*.

2.2.5 Metode AHP

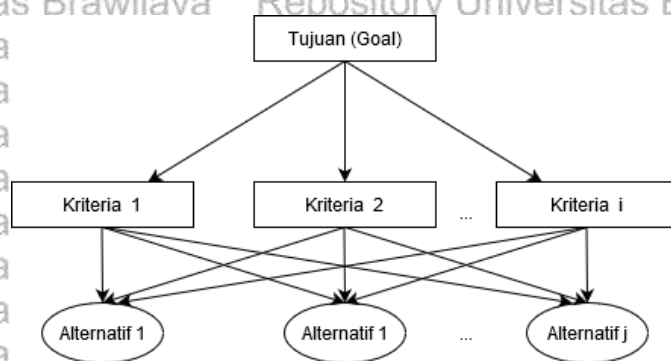
Metode dikembangkan awal tahun 1970-an oleh Thomas L. Saaty, seorang ahli matematika dari Universitas Pittsburg. AHP merupakan metode untuk membuat urutan *alternative* yang bertujuan untuk mendapatkan keputusan yang terbaik pada saat pengambilan keputusan.

2.2.5.1 Prinsip Dasar AHP

Menurut Syukron (2014), terdapat 3 prinsip dasar yang digunakan dalam AHP. Prinsip-prinsip tersebut adalah:

1. Penyusunan Hirarki

Prinsip ini memecah satu tujuan utuh menjadi unsur-unsur yang terpisah. Struktur hirarki dapat dilihat pada Gambar 2.2. Hirarki tertinggi adalah tujuan (*goal*) yang ingin dicapai. Tingkat kedua adalah kriteria. Kriteria merupakan aspek penting yang perlu dipertimbangkan dalam mengambil keputusan. Tingkat terendah adalah *alternative*. *Alternative* adalah pilihan keputusan dari penyelesaian masalah.



Gambar 2.2 Struktur Hirarki

Sumber: (Mulyono, 2004)



2. Penentuan Prioritas

Prinsip ini memberikan penilaian tentang kepentingan relatif dua elemen. Penilaian yang digunakan pada proses perbandingan kepentingan antar faktor yaitu berdasarkan skala nilai perbandingan berpasangan yang dikembangkan oleh Saaty. Skala nilai perbandingan berpasangan dapat dilihat pada Tabel 2.2.

Tabel 2.2 Nilai Skala Perbandingan Berpasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya.
3	Elemen yang satu sedikit lebih penting dari elemen lainnya.
5	Elemen yang satu lebih penting dari elemen lainnya.
7	Satu elemen jelas lebih mutlak penting dari pada elemen lainnya.
9	Satu elemen mutlak penting dari pada elemen lainnya.
2,4,6,8	Nilai - nilai di antara dua pertimbangan yang berdekatan.
Kebalikan	Jika untuk aktivitas i mendapatkan satu angka dibandingkan dengan aktivitas j, maka j mempunyai nilai kebalikannya dibandingkan i.

3. Konsistensi Logis

Prinsip ini menjamin semua elemen dikelompokkan secara logis dan diperingkatkan secara konsisten sesuai dengan kriteria yang logis.

2.2.5.2 Perhitungan AHP

Berikut merupakan perhitungan dengan metode AHP (Benrnadifta, et al., 2016):

1. Dekomposisi Masalah

Dekomposisi masalah adalah langkah penguraian secara sistematis dari tujuan yang telah ditetapkan ke dalam struktur yang menyusun rangkaian sistem hingga tujuan dapat dicapai secara rasional.

2. Menetapkan Prioritas Elemen

2.1. Langkah pertama dalam menentukan prioritas elemen dengan membuat perbandingan berpasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan.

2.2. Matriks perbandingan berpasangan diisi berdasarkan skala nilai perbandingan berpasangan yang dikembangkan oleh Saaty.

3. Sintesis

Untuk memperoleh prioritas secara keseluruhan maka pertimbangan-pertimbangan terhadap perbandingan berpasangan perlu disintesis. Dalam langkah ini, hal-hal yang dilakukan adalah:

3.1. Menjumlahkan nilai-nilai dari setiap kolom matriks.



3.2. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.

3.3. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.

4. Mengukur Konsistensi

Dalam pembuatan keputusan, tingkat konsistensi penting untuk diperhatikan karena tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah:

4.1. Mengalihkan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada elemen kedua dengan prioritas relatif elemen kedua dan seterusnya.

4.2. Menjumlahkan setiap baris.

4.3. Hasil dari penjumlahan baris dibagi elemen prioritas yang bersangkutan.

4.4. Menjumlahkan hasil bagi diatas dengan banyaknya elemen yang dihasilkan disebut λ_{max} seperti persamaan 2.1.

$$\lambda_{max} = \frac{\sum cv}{n} \quad (2.1)$$

Keterangan:

- λ_{max} : nilai eigen maksimum
- $\sum cv$: jumlah setiap baris
- n : ukuran matriks

5. Menghitung *Consistency Index* (CI)

Consistency Index (CI) adalah tingkat konsistensi dalam memberikan jawaban terhadap suatu elemen di dalam masalah. Rumus menghitung nilai *Consistency Index* (CI) dapat dilihat pada persamaan 2.2.

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (2.2)$$

Keterangan:

- CI : nilai *consistency index*
- λ_{max} : nilai eigen maksimum
- n : ukuran matriks

6. Menghitung *Consistency Ratio* (CR)

Consistency Ratio (CR) adalah hasil perbandingan antara *index*. Rumus menghitung nilai *Consistency Ratio* CR dapat dilihat pada persamaan 2.3.

$$CR = \frac{CI}{RI} \quad (2.3)$$

Keterangan:

- CR : nilai *consistency ratio*
- CI : nilai *consistency index*



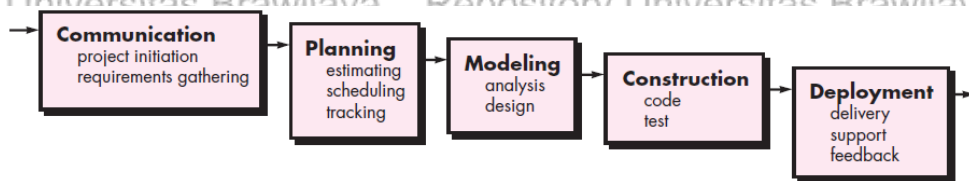
RI : nilai indeks konsisten random

7. Memeriksa *Consistency Hierarki*

Jika nilai *consistency ratio* (CR) lebih dari 0,1, maka penilaian data keputusan harus diperbaiki. Namun, jika nilai *consistency ratio* (CR) kurang atau sama dengan 0,1 maka hasil perhitungan bisa dinyatakan benar.

2.2.6 Waterfall Model

Model *waterfall* terkadang disebut siklus hidup klasik, menunjukkan pendekatan yang sistematis dan berurutan pada pengembangan perangkat lunak. Dimulai dengan menspesifikasikan kebutuhan pengguna dan dilanjutkan dengan perencanaan, pemodelan, konstruksi, serta penyerahan perangkat lunak ke para pengguna (Pressman, 2010).



Gambar 2.3 Waterfall Model

Sumber: (Pressman, 2010)

Tahap-tahap dari model *waterfall* yang menggambarkan inti dari aktivitas-aktivitas pengembangan adalah:

1. *Communication (requirements gathering)*

Pada tahap ini, layanan yang akan diberikan oleh sistem, batasan sistem, dan tujuan ditetapkan setelah melakukan konsultasi dengan pengguna sistem. Semua didefinisikan secara rinci dan dibuat sebagai spesifikasi dari sistem.

2. *Planning (estimating, scheduling, tracking)*

Pada tahap ini, dilakukanlah perencanaan. Perencanaan untuk pembuatan sistem melibatkan pengidentifikasian dan penjelasan dari abstraksi sistem dan hubungannya, perkiraan waktu pengerjaan, penjadwalan.

3. *Modelling (analysis and design)*

Pada tahap ini, dilakukan analisis dan perancangan dari sistem yang ingin dibuat berdasarkan perencanaan yang telah dibuat sebelumnya.

4. *Construction (coding and sistem test)*

Pada tahap ini, perencanaan yang telah dilakukan sebelumnya akan diubah menjadi sebuah pemrograman atau unit program. lalu dilakukan pengujian unit yang melibatkan verifikasi untuk memastikan apakah setiap unit memenuhi spesifikasi sistem. Setiap unit program dan program-program yang sudah ada diintegrasikan dan diuji sebagai satu keutuhan sistem untuk memastikan apakah kebutuhan sistem sudah terpenuhi. Setelah melakukan pengujian, sistem baru disebarakan ke pengguna.



5. *Deployment (delivery, support, feedback)*

Pada tahap ini, dilakukan instalasi terhadap sistem dan digunakan dalam prakteknya. *Maintenance* melibatkan koreksi terhadap *error* yang tidak ditemukan pada tahap sebelumnya, memperbaiki implementasi dari unit sistem dan meningkatkan layanan yang diberikan oleh sistem sebagai kebutuhan baru yang ditemukan. Serta memberikan bantuan kepada *user* yang mengalami kendala.


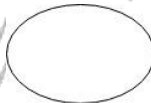
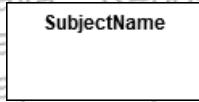

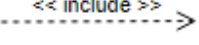
2.2.7 UML

Unified Modeling Language (UML) adalah sebuah rancang model yang memiliki berbagai macam notasi, di mana perancangan berorientasi objek akan menggunakan berbagai notasi tersebut (Sommerville, 2011). Berikut adalah beberapa bagian dari rancang model UML yang akan dijelaskan.

2.2.7.1 Use Case Diagram

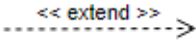
Use case diagram akan menjelaskan isi dari fungsionalitas sistem. Pada *use case diagram* sudah ditentukan atas "apa" yang akan dibuat oleh sistem dan bukan "bagaimana". Komponen-komponen yang terdapat di *use case diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Notasi Use Case Diagram

No.	Nama	Simbol	Deskripsi
1	<i>Actor</i>		Aktor adalah orang atau sistem lain yang berinteraksi dengan sistem.
2	<i>Use case</i>		Merepresentasikan bagian utama dari fungsionalitas sistem.
3	<i>Subject boundary</i>		Merepresentasikan ruang lingkup dari subjek. Contohnya adalah sebuah sistem atau proses bisnis.
4	<i>Association</i>		Penghubung aktor dengan <i>use case</i> yang digunakan untuk berinteraksi.
5	<i>Include</i>		Menampilkan hubungan satu <i>use case</i> yang digunakan oleh beberapa <i>use case</i> lain.



Tabel 2.3 Notasi Use Case Diagram (Lanjutan)


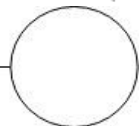

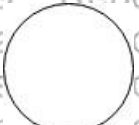


No.	Nama	Simbol	Deskripsi
6	<i>Extend</i>		Menampilkan hubungan suatu <i>use case</i> yang merupakan fungsionalitas tambahan dari <i>use case</i> lain.

Sumber: (Dennis, et al., 2012)


2.2.7.2 Sequence Diagram

Sequence diagram menjelaskan interaksi yang terjadi antara objek dengan objek di dalam sistem. Adapun komponen-komponen yang terdapat pada *sequence diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Notasi Sequence Diagram

No.	Nama	Simbol	Deskripsi
1	<i>Actor</i>		Aktor adalah pengguna sistem. Pengguna dapat berupa manusia, mesin, atau sistem lainnya.
2	<i>Boundary</i>		<i>Boundary</i> menggambarkan interaksi objek dengan aktor.
3	<i>Controller</i>		<i>Controller</i> menggambarkan objek antara <i>boundary</i> dan <i>entity</i> yang akan melakukan proses interaksi.
4	<i>Entity</i>		<i>Entity</i> menggambarkan objek yang mengatur data dan informasi pada sistem.
5	<i>Message</i>		<i>Message</i> menggambarkan komunikasi antara objek-objek.
6	<i>Return</i>		<i>Return</i> akan menampilkan pesan yang dikembalikan oleh <i>message</i> sebelumnya.

Tabel 2.4 Notasi *Sequence Diagram* (Lanjutan)

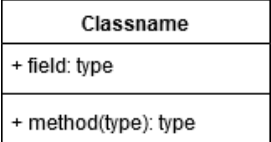

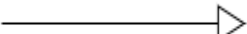

No.	Nama	Simbol	Deskripsi
7	<i>Self-message</i>		<i>Self-message</i> menggambarkan komunikasi yang kembali kedalam objek itu sendiri.

Sumber: (System, 2016)

2.2.7.3 *Class Diagram*

Class diagram menjelaskan struktur dan deskripsi *class*, *packed* dan objek beserta hubungan satu sama lain. Adapun komponen-komponen yang terdapat pada *class diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Notasi *Class Diagram*

No.	Nama	Simbol	Deskripsi
1	<i>Class</i>		<i>Class</i> menggambarkan orang, tempat, atau hal yang berhubungan dimana sistem memerlukan tempat menyimpan informasi.
2	<i>Association</i>		Merepresentasikan hubungan antara banyak klas atau klas itu sendiri.
3	<i>Generalization</i>		Merepresentasikan hubungan antara banyak klas dengan makna generalisasi-spesialisasi (umum-khusus).
4	<i>Aggregation</i>		Merepresentasikan hubungan antara banyak klas atau klas itu sendiri (<i>whole-part</i>).

Sumber: (Dennis, et al., 2012)

2.2.8 Bahasa Pemrograman

Menurut Munir (2011), bahasa pemrograman adalah bahasa komputer yang digunakan dalam menulis program. Berikut adalah bahasa pemrograman yang digunakan dalam penelitian ini adalah:

2.2.8.1 HTML

Hypertext Markup Language (HTML) adalah bahasa yang digunakan untuk menulis halaman web. Fungsi dari HTML adalah memanipulasi *browser* sehingga



dapat menampilkan informasi yang dapat dibaca oleh pengguna (Koesheryatin & Suryana, 2014).

Contoh dokumen HTML.

```
<html>
<head>
  <title></title>
</head>
<body>
  Isi
</body>
</html>
```

2.2.8.2 JavaScript

JavaScript adalah bahasa *script* yang berpedoman pada objek yang memperbolehkan pengguna untuk mengendalikan banyak aspek interaksi pengguna pada suatu dokumen HTML. Objek tersebut dapat berupa suatu *window*, *frame*, URL, dokumen, *form*, *button*, atau item yang lain (Koesheryatin & Suryana, 2014).

Contoh penulisan *script*.

```
<SCRIPT LANGUAGE="JavaScript">
//program Anda masukkan di sini
</SCRIPT>
```

2.2.8.3 PHP

PHP adalah singkatan dari *Hypertext Processor*. Menggunakan PHP dapat membantu pengembang merancang sebuah web dinamis dengan cepat. Struktur penulisan PHP dapat dituliskan berdiri sendiri atau disisipkan pada *script* HTML. (Connolly & Begg, 2010).

Contoh penulisan *script* PHP.

```
<html>
<head>
  <title></title>
</head>
<body>
  <?php
  //isi
  ?>
</body>
</html>
```

2.2.9 Pengujian Perangkat Lunak

Untuk meningkatkan kualitas perangkat lunak perlu dilakukan evaluasi guna mencari kesalahan (Bertolino, 2001). Berikut adalah merupakan pengujian perangkat lunak yaitu:



2.2.9.1 Pengujian Unit

Pengujian unit dilakukan dengan pendekatan *white-box testing*. Diperlukan kode program dalam pengujian ini (Sommerville, 2011). Metode yang digunakan dalam pengujian ini adalah dengan metode *basis path testing*. Tujuannya adalah untuk mendapatkan kompleksitas dari perancangan algoritme (Pressman, 2010).

Tahap-tahap dalam pengujian *basis path* adalah:

1. Mendefinisikan *flow graph*
2. Menentukan ukuran kompleksitas (*cyclomatic complexity*)
3. Menentukan *independent path*
4. Mendefinisikan kasus uji

2.2.9.2 Pengujian Validasi

Pengujian validasi dilakukan dengan pendekatan *black-box testing*. Pengujian validasi digunakan untuk mengetahui apakah semua fungsi yang ada dalam perangkat lunak telah berjalan sesuai dengan kebutuhan yang telah didefinisikan. Teknik pengujian validasi berfokus pada domain informasi dari perangkat lunak yaitu melakukan *test case* pada salah satu *use case* dengan memastikan bahwa domain input akan sesuai dengan domain *output*. Kesalahan yang terdapat pada sistem seperti kesalahan fungsi, antarmuka, dan *database* dapat ditemukan dengan menggunakan pengujian ini (Pressman, 2010).

2.2.9.3 Pengujian *Compatibility*

Pengujian *compatibility* dilakukan untuk memastikan apakah sistem yang dibangun dapat berjalan pada sistem operasi ataupun *browser* yang berbeda. Dalam aplikasi web, perbedaan sistem operasi atau *browser* terkadang dapat memberikan hasil yang tidak sama. Terdapat beberapa kasus dalam permasalahan kompatibilitas. Contohnya adalah saat pengunduhan tidak dapat dilakukan. Permasalahan seperti ini terjadi karena perbedaan *browser* yang dapat mengubah tata letak halaman atau *font style* yang dapat berubah sehingga tidak dapat dibaca. Dengan melakukan pengujian ini, pengembang akan dapat mengetahui masalah kompatibilitas dengan skala kecil sebelum perangkat lunak dirilis dan digunakan (Pressman, 2010).

Pengujian ini dilakukan dengan menggunakan aplikasi SortSite. SortSite adalah sebuah aplikasi yang digunakan sebagai alat pendeteksi *website* untuk melakukan pemeriksaan dari berbagai sisi, baik itu *accessibility*, *compatibility*, *privacy*, ataupun *usability*. Indikator masalah *compatibility* yang ada pada aplikasi SortSite dibagi menjadi 3 (tiga) yaitu *critical issue*, *major issue*, dan *minor issue*. *Critical issue* menunjukkan bahwa fungsionalitas atau konten yang tidak bisa digunakan/didukung pada suatu *browser*. *Major issue* menunjukkan adanya masalah utama pada performa atau tampilan pada suatu *browser*. *Minor issue* menunjukkan adanya masalah *minor*/kecil pada performa atau tampilan pada suatu *browser*. Kategori masalah *compatibility* yang menjadi indikator pada aplikasi SortSite ditunjukkan pada Gambar 2.4.



Key

- Missing content or functionality
- Major layout or performance problems
- Minor layout or performance problems

Gambar 2.4 Indikator Masalah *Compatibility SortSite*

2.2.9.4 Pengujian Akurasi

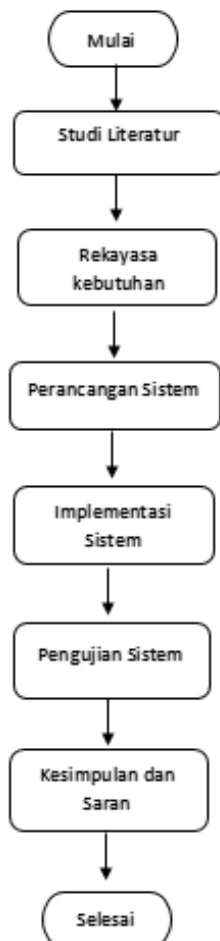
Pengujian akurasi dilakukan untuk mengetahui nilai akurasi yang dihasilkan dari sistem. Akurasi adalah hasil perhitungan semua nilai prediksi yang benar dibagi dengan keseluruhan data dan dinyatakan dalam bentuk persentase (Nugroho & Sari, 2018). Perhitungannya dapat dilihat pada persamaan 2.4.

$$\%Akurasi = \frac{\text{Jumlah data akurat}}{\text{Jumlah seluruh data}} \times 100\% \quad (2.4)$$



BAB 3 METODOLOGI PENELITIAN

Pada Gambar 3.1 digambarkan alur dalam melakukan penelitian ini. Penelitian ini merupakan salah satu bentuk penelitian implementatif.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Tahap pertama dalam melakukan penelitian adalah melakukan studi literatur. Studi literatur adalah mencari referensi-referensi teori yang sesuai dengan permasalahan yang diteliti untuk menjadi dasar landasan yang menunjang penelitian ini dalam menyusun kerangka pemikiran dari kasus atau permasalahan. Sumber literatur atau referensi dalam penelitian ini meliputi skripsi, jurnal, buku, dan artikel. Teori-teori yang digunakan untuk mendukung penelitian ini adalah:

1. Perawatan kulit
2. Sistem rekomendasi
3. *Waterfall model*
4. MVC (*Model-view-controller*)



5. AHP (*Analytic Hierarchy Process*)
6. Pengujian perangkat lunak

3.2 Rekayasa Kebutuhan

Tahap ini dilakukan untuk mencari dan mengidentifikasi kebutuhan sistem. Kebutuhan sistem didapatkan dengan menggunakan metode observasi, kuisisioner dan wawancara. Observasi dilakukan dengan cara mengamati langsung. Observasi dilakukan untuk mengumpulkan data produk yang akan digunakan di sistem. Kuisisioner dilakukan dengan memberi pertanyaan kepada responden. Wawancara dilakukan dengan cara tanya jawab langsung secara tatap muka. Wawancara dilakukan dengan pakar untuk mendapatkan data-data yang berkaitan dengan pemilihan dan pemakaian produk perawatan kulit.

3.3 Perancangan Sistem

Tahap ini dilakukan berdasarkan kebutuhan yang telah dikumpulkan. Perancangan sistem pada penelitian ini terdiri dari:

1. Perancangan Arsitektur
Perancangan arsitektur digunakan untuk menjelaskan komponen utama dari sistem dan hubungan antar komponen-komponen tersebut. Perancangan arsitektur menerapkan pola MVC (*Model-View-Controller*).
2. Perancangan *Sequence Diagram*
Perancangan ini digunakan untuk menjelaskan interaksi antar objek di dalam sistem dalam urutan waktu. Perancangan yang akan ditampilkan adalah sebanyak tiga sampel.
3. Perancangan *Class Diagram*
Perancangan ini digunakan untuk menjelaskan struktur dan deskripsi *class*, *packed* dan objek beserta hubungan satu sama lain. Perancangan yang akan ditampilkan adalah sebanyak tiga sampel.
4. Perancangan Data
Pada perancangan ini dilakukan suatu perancangan basis data yang dimodelkan dalam bentuk ERD (*Entity Relationship Diagram*).
5. Perancangan Komponen
Perancangan komponen digunakan untuk menjelaskan gambaran dari alur atau proses yang dibuat dalam bentuk algoritme. Perancangan yang akan ditampilkan adalah sebanyak tiga sampel.
6. Perancangan Antarmuka
Perancangan antarmuka dilakukan untuk menggambarkan antarmuka sistem yang akan dibangun. Perancangan yang akan ditampilkan adalah sebanyak lima sampel.

3.4 Implementasi Sistem

Implementasi sistem dilakukan setelah perancangan sistem. Pada penelitian ini, *framework* CodeIgneter digunakan dalam mengimplementasikan sistem. Hal



ini dikarenakan sistem berbasis web. Implementasi sistem pada penelitian ini terdiri dari:

1. Implementasi Antarmuka
Implementasi antarmuka menggunakan bahasa pemrograman PHP, CSS, dan JavaScript.
2. Implementasi Basis Data
Implementasi basis data menggunakan MySQL.

3.5 Pengujian Sistem

Untuk melakukan pengecekan fungsional dan non-fungsional sistem, dilakukan pengujian. Terdapat lima pengujian yang akan digunakan pada penelitian ini, yaitu:

1. Pengujian Unit
Pengujian unit dilakukan menggunakan pendekatan *white-box testing* dengan menggunakan metode *basis path testing*. Pada pengujian ini yang diuji adalah kode program berdasarkan algoritme yang berada di tiap-tiap *method* pada klas.
2. Pengujian Integrasi
Pengujian integrasi dilakukan menggunakan pendekatan *white-box testing* dengan menggunakan metode *basis path testing*. Pengujian ini dilakukan untuk mengetahui interaksi yang terjadi antar klas.
3. Pengujian Validasi
Pengujian validasi dilakukan menggunakan pendekatan *black-box testing*. Pengujian ini dilakukan untuk menguji fungsionalitas sistem. Apakah dalam kondisi normal atau dalam kondisi *alternative*, sistem dapat berjalan sesuai dengan parameter yang sudah ditentukan sebelumnya.
4. Pengujian *Compatibility*
Pengujian *compatibility* dilakukan untuk melihat apakah sistem dapat berjalan dengan baik pada berbagai *browser* dan sistem operasi yang berbeda. Pengujian ini menggunakan aplikasi yang bernama *sortsite*.
5. Pengujian Akurasi
Pengujian akurasi dilakukan untuk mengetahui nilai akurasi yang dihasilkan dari sistem. Pengujian ini membandingkan keakuratan keluaran sistem dengan data yang sebenarnya.

3.6 Kesimpulan dan Saran

Kesimpulan didapat berdasarkan hasil dari tahapan yang telah dilakukan sebelumnya yang dimulai dari rekayasa kebutuhan sampai dengan pengujian. Tahap ini dilakukan agar diperoleh inti dari keseluruhan penelitian yang telah dilakukan dan sekaligus menjawab permasalahan yang telah dirumuskan. Setelah mengambil kesimpulan, dituliskan saran untuk pertimbangan pengembangan sistem dalam penelitian selanjutnya.



BAB 4 REKAYASA KEBUTUHAN

4.1 Rekayasa Kebutuhan

Tahapan rekayasa bertujuan dalam mendefinisikan dan menspesifikasikan kebutuhan pada sistem. Kebutuhan didapatkan dari hasil wawancara, kuisisioner, dan observasi.

4.1.1 Deskripsi Sistem

Fungsi dari sistem yang dibangun adalah membantu pengguna dalam memilih produk perawatan kulit. Metode yang digunakan dalam proses pemilihan produk perawatan kulit adalah dengan metode AHP. AHP itu sendiri merupakan metode untuk membuat urutan *alternative* yang bertujuan untuk mendapatkan keputusan yang terbaik pada saat pengambilan keputusan.

4.1.2 Identifikasi Aktor

Tabel 4.1 menjelaskan aktor yang terlibat didalam sistem ini beserta penjelasan fungsinya.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi
Admin	Merupakan aktor yang memiliki peran untuk mengelola data produk dan data artikel.
Member	Merupakan aktor yang memiliki hak berpartisipasi dalam menggunakan fitur <i>find recommendation</i> , mengubah data diri, memberi <i>review</i> produk, memberi <i>like</i> dan komentar pada artikel.
Guest	Merupakan aktor yang menggunakan sistem secara umum.

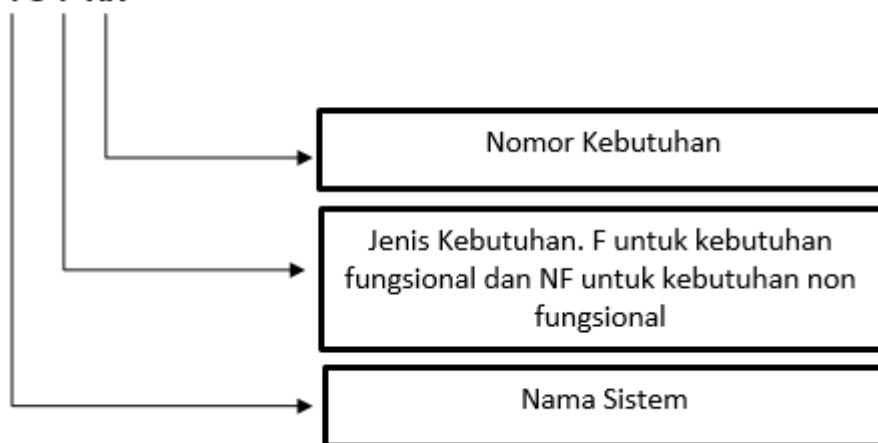
4.2 Spesifikasi Kebutuhan

4.2.1 Aturan Penomoran

Gambar 4.1 menjelaskan aturan penomoran yang digunakan untuk pengkodean setiap kebutuhan.



RPPS-F-XX



Gambar 4.1 Aturan Penomoran

4.2.2 Kebutuhan Fungsional

Pada Tabel 4.2 dijabarkan kebutuhan fungsional pada sistem.

Tabel 4.2 Daftar Kebutuhan Fungsional

No.	Kode	Nama Fungsi	Deskripsi	Aktor
1	RPPS-F-01	Login	Sistem harus mampu menyediakan fungsi <i>login</i> untuk mengizinkan <i>guest</i> menggunakan sistem sebagai <i>member</i> atau <i>admin</i> . 1.1. <i>Field</i> yang harus diisi dalam <i>form login</i> adalah <i>username</i> dan <i>password</i> .	<i>Guest</i>
2	RPPS-F-02	Registrasi Akun	Sistem harus mampu menyediakan fungsi registrasi jika <i>guest</i> ingin terdaftar ke dalam sistem dan menjadi <i>member</i> . 2.1. <i>Field</i> yang harus diisi dalam <i>form register</i> adalah <i>name</i> , <i>address</i> , <i>email</i> , <i>username</i> , dan <i>password</i> .	<i>Guest</i>



Tabel 4.2 Daftar Kebutuhan Fungsional (lanjutan)

No.	Kode	Nama Fungsi	Deskripsi	Aktor
			<p>2.2. Sistem menampilkan pesan peringatan jika terdapat <i>username</i> dan <i>email</i> yang sama.</p> <p>2.3. Sistem menampilkan pesan peringatan jika terdapat <i>username</i> dan <i>password</i> yang kurang dari jumlah minimal karakter.</p>	
3	RPPS-F-03	<i>Logout</i>	Sistem harus mampu menyediakan fungsi <i>logout</i> kepada admin dan <i>member</i> untuk dapat keluar dari sistem.	Admin dan <i>Member</i>
4	RPPS-F-04	Melihat Data Diri	<p>Sistem harus mampu menyediakan fungsi melihat data diri.</p> <p>4.1. Sistem menampilkan informasi data diri dari <i>member</i> yang terdiri dari nama, alamat, <i>email</i>, foto, usia, jenis kulit, permasalahan kulit.</p>	<i>Member</i>
5	RPPS-F-05	Mengubah Data Diri	<p>Sistem harus mampu menyediakan fungsi mengubah data diri.</p> <p>5.1. Sistem menyediakan <i>form</i> yang terdiri dari nama, alamat, <i>email</i>, foto, usia, jenis kulit, permasalahan kulit.</p> <p>5.2. Sistem menampilkan pesan konfirmasi sebelum perubahan dilakukan.</p>	<i>Member</i>
6	RPPS-F-06	Melihat Halaman Rekomendasi	Sistem harus mampu menyediakan fungsi melihat halaman rekomendasi.	<i>Member</i>



Tabel 4.2 Daftar Kebutuhan Fungsional (lanjutan)

No.	Kode	Nama Fungsi	Deskripsi	Aktor
			6.1. Sistem menampilkan halaman rekomendasi dalam bentuk <i>multiple page form</i> yang terdiri dari 3 (tiga) <i>page</i> . <i>Page</i> pertama adalah jenis kulit. <i>Page</i> kedua adalah permasalahan kulit, dan terakhir adalah kategori produk.	
7	RPPS-F-07	Mengisi Data Rekomendasi	Sistem harus mampu menyediakan fungsi mengisi data rekomendasi. 7.1. Sistem menyediakan pilihan untuk <i>member</i> . Pilihan pertama memilih tipe kulit. Pilihan kedua memilih permasalahan kulit. Pilihan ketiga memilih kategori produk yang dicari.	<i>Member</i>
8	RPPS-F-08	Mengisi Nilai Perbandingan	Sistem harus mampu menyediakan fungsi mengisi nilai perbandingan. 8.1. Sistem menyediakan pilihan untuk <i>member</i> . Jika <i>member</i> memilih kolom yang disebelah kiri, maka kriteria disebelah kiri memiliki derajat lebih tinggi daripada kolom yang disebelah kanan. Begitu pun sebaliknya.	<i>Member</i>
9	RPPS-F-09	Melihat Daftar Rekomendasi Produk	Sistem harus mampu menyediakan fungsi melihat halaman daftar rekomendasi.	<i>Member</i>



Tabel 4.2 Daftar Kebutuhan Fungsional (lanjutan)

No.	Kode	Nama Fungsi	Deskripsi	Aktor
			9.1. Sistem menampilkan daftar rekomendasi dalam bentuk Tabel yang terdiri dari <i>rating</i> , nama, <i>brand</i> , deskripsi, harga, dan tempat pembelian produk.	
10	RPPS-F-10	Menulis Ulasan Produk	Sistem harus mampu menyediakan fungsi menulis ulasan tentang produk. 10.1. Sistem menyediakan kolom untuk menulis ulasan.	Member
11	RPPS-F-11	Menghapus Ulasan Produk	Sistem harus mampu menyediakan fungsi menghapus ulasan tentang produk yang telah ditulis sebelumnya.	Member
12	RPPS-F-12	Menulis Ulasan Artikel	Sistem harus mampu menyediakan fungsi menulis ulasan tentang artikel. 12.1. Sistem menyediakan kolom untuk menulis ulasan.	Member
13	RPPS-F-13	Menghapus Ulasan Artikel	Sistem harus mampu menyediakan fungsi menghapus ulasan tentang asrtikel yang telah ditulis sebelumnya.	Member
14	RPPS-F-14	Menyukai Artikel	Sistem harus mampu menyediakan fungsi <i>like</i> pada artikel.	Member
15	RPPS-F-15	Membatalkan menyukai artikel	Sistem harus mampu menyediakan fungsi <i>unlike</i> pada artikel.	Member
16	RPPs-F-16	Melihat Daftar Pengguna	Sistem harus mampu menyediakan fungsi melihat halaman daftar pengguna.	Admin



Tabel 4.2 Daftar Kebutuhan Fungsional (lanjutan)

No.	Kode	Nama Fungsi	Deskripsi	Aktor
			16.1. Sistem menampilkan daftar pengguna dalam bentuk Tabel yang terdiri dari nama, alamat, <i>email</i> , dan <i>username</i> .	
17	RPPS-F-17	Menambah Data Produk	Sistem harus mampu menyediakan fungsi tambah data produk. 17.1. Sistem menyediakan halaman untuk menambah data produk yang terdiri dari <i>problem</i> , kategori, tipe kulit, <i>brand</i> produk, nama produk, deskripsi, kualitas, harga, bahan, tempat pembelian, dan gambar produk.	Admin
18	RPPS-F-18	Melihat Daftar Produk	Sistem harus mampu menyediakan fungsi melihat daftar produk. 18.1. Sistem menampilkan daftar produk dalam bentuk tabel yang terdiri dari <i>brand</i> , nama produk, dan gambar produk. 18.2. Pada Tabel daftar produk tersedia <i>field</i> pencarian.	Admin
19	RPPS-F-19	Mengubah Data Produk	Sistem harus mampu menyediakan fungsi ubah data produk. 19.1. Sistem menyediakan halaman untuk ubah data produk yang terdiri dari <i>problem</i> ,	Admin



Tabel 4.2 Daftar Kebutuhan Fungsional (lanjutan)

No.	Kode	Nama Fungsi	Deskripsi	Aktor
			kategori, tipe kulit, <i>brand</i> produk, nama produk, deskripsi, kualitas, harga, bahan, tempat pembelian, dan gambar produk.	
20	RPPS-F-20	Menghapus Data Produk	Sistem harus mampu menyediakan fungsi hapus data produk.	Admin
21	RPPS-F-21	Menambah Data Artikel	Sistem harus mampu menyediakan fungsi tambah data artikel. 21.1. Sistem menyediakan <i>form</i> untuk menambah data artikel yang terdiri dari judul artikel, sumber, URI, nama penulis, tanggal penulisan, isi artikel, dan gambar artikel.	Admin
22	RPPS-F-22	Melihat Daftar Artikel	Sistem harus mampu menyediakan fungsi melihat daftar artikel. 22.1. Sistem menampilkan daftar artikel dalam bentuk Tabel yang terdiri dari nama penulis, judul artikel, dan gambar artikel. 22.2. Pada tabel daftar artikel tersedia <i>field</i> pencarian.	Admin
23	RPPS-F-23	Mengubah Data Artikel	Sistem harus mampu menyediakan fungsi ubah data artikel. 23.1. Sistem menyediakan halaman untuk ubah data artikel yang terdiri	Admin



Tabel 4.2 Daftar Kebutuhan Fungsional (lanjutan)

No.	Kode	Nama Fungsi	Deskripsi	Aktor
			dari judul artikel, sumber, URI, nama penulis, tanggal penulisan, isi artikel, dan gambar artikel.	
24	RPPS-F-24	Menghapus Data Artikel	Sistem harus mampu menyediakan fungsi hapus data artikel.	Admin
25	RPPS-F-25	Melihat Produk	Sistem harus mampu menyediakan fungsi melihat halaman detail produk.	Member dan Guest
27	RPPS-F-27	Mencari Produk	Sistem harus mampu menyediakan fungsi mencari produk dengan memasukkan <i>keyword</i> . 27.1. Sistem menyediakan <i>field</i> pencarian.	Member dan Guest
28	RPPS-F-28	Mencari Artikel	Sistem harus mampu menyediakan fungsi mencari artikel dengan memasukkan <i>keyword</i> . 28.1. Sistem menyediakan <i>field</i> pencarian.	Member dan Guest
29	RPPS-F-29	Melihat Halaman <i>About Us</i>	Sistem harus mampu menyediakan fungsi melihat halaman <i>about us</i> .	Member dan Guest
30	RPPS-F-30	Melihat Halaman FAQ	Sistem harus mampu menyediakan fungsi melihat halaman FAQ.	Member dan Guest
31	RPPS-F-31	Melihat Data Diri <i>Member</i> lain	Sistem harus mampu menyediakan fungsi melihat data diri member lain.	Member dan Guest

4.2.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dapat dilihat pada Tabel 4.3.



Tabel 4.3 Daftar Kebutuhan Non-Fungsional

No.	Kode	Parameter	Deskripsi
1	RPPS-NF-01	<i>Compatibility</i>	Sistem dapat diakses melalui berbagai browser dan sistem operasi yang berbeda.

4.3 Pemodelan Kebutuhan

Pemodelan kebutuhan dilakukan dengan UML. Diagram yang digunakan adalah *use case diagram* dan *use case scenario*.

4.3.1 Use Case Diagram

Use case merupakan perilaku atau interaksi yang terjadi antara aktor dan sistem. *Use case diagram* dibuat berdasarkan dari kebutuhan fungsional dari sistem yang telah di definisikan sebelumnya. Gambar 4.2 menunjukkan ada terdapat 31 *use case* dan 3 aktor.

4.3.2 Use Case Scenario

Use case scenario menjelaskan secara rinci *use case* yang dilakukan oleh sistem. *Use case scenario* juga menggambarkan alur sebuah *use case* dalam mendapatkan tujuan fungsionalnya.

4.3.2.1. Login

Tabel 4.4 adalah penjelasan dari *use case scenario login*.

Tabel 4.4 Login

Login	
Objective	Melakukan <i>login</i> untuk mendapatkan hak akses sesuai perannya
Actor	<i>Guest</i>
Pre-Condition	Aktor ada di halaman utama
Main Flow	<ol style="list-style-type: none"> Memilih menu <i>login</i> pada menu <i>dropdown account</i> Halaman <i>login</i> ditampilkan Pada halaman login diisi <i>username</i> dan <i>password</i> pada <i>field</i> yang disediakan Tombol "<i>Login</i>" ditekan
Alternative Flow	<ol style="list-style-type: none"> Jika terdapat <i>field</i> yang belum diisi, maka akan muncul pesan peringatan pada <i>field</i> yang belum diisi untuk diisi menampilkan pesan "<i>wrong username or password</i>"



Tabel 4.4 Login (Lanjutan)

Login	
Post Condition	Berhasil melakukan <i>login</i> dan masuk ke halaman utama sistem

4.3.2.2. Registrasi Akun

Tabel 4.5 adalah penjelasan *use case scenario* registrasi akun.

Tabel 4.5 Registrasi Akun

Registrasi Akun	
Objective	Melakukan <i>register</i> jika belum pernah terdaftar ke dalam sistem
Actor	<i>Guest</i>
Pre-Condition	Aktor ada di halaman utama
Main Flow	<ol style="list-style-type: none"> Memilih menu <i>register</i> pada menu <i>dropdown account</i> Halaman <i>register</i> ditampilkan Pada halaman <i>register</i> diisi <i>name, address, email, username, dan password</i> pada <i>field</i> yang disediakan Tombol "<i>Register</i>" ditekan
Alternative Flow	<ol style="list-style-type: none"> Jika ada <i>field</i> yang belum terisi, maka sistem memberikan pesan peringatan pada <i>field</i> yang belum diisi untuk diisi Jika ada <i>username</i> dan <i>email</i> yang sama, maka sistem akan memberikan pesan peringatan untuk mengganti data dengan <i>username</i> dan <i>email</i> yang lain Jika ada <i>username</i> dan <i>password</i> yang kurang dari jumlah minimal karakter, maka sistem akan memberikan pesan peringatan untuk menambahkan karakternya
Post Condition	Sistem memproses data <i>register</i> ke dalam <i>database</i> dan menampilkan halaman <i>login</i>

4.3.2.3. Logout

Tabel 4.6 adalah penjelasan *use case scenario* *logout*.



Tabel 4.6 Logout

Logout	
Objective	Logout dapat dilakukan
Actor	Admin, Member
Pre-Condition	Login telah dilakukan
Main Flow	1. Memilih menu <i>logout</i> pada menu <i>dropdown account</i>
Alternative Flow	-
Post Condition	Berhasil melakukan logout dan masuk ke halaman utama sistem sebagai <i>guest</i>

4.3.2.4. Melihat Data Diri

Tabel 4.7 adalah penjelasan *use case scenario* melihat data diri.

Tabel 4.7 Melihat Data Diri

Melihat Data Diri	
Objective	Melihat informasi data diri
Actor	Member
Pre-Condition	Login telah dilakukan
Main Flow	1. Aktor menekan menu <i>profile</i> pada menu <i>dropdown account</i>
Alternative Flow	-
Post Condition	Halaman <i>my profile</i> ditampilkan

4.3.2.5. Mengubah Data Diri

Tabel 4.8 adalah penjelasan *use case scenario* mengubah data diri.

Tabel 4.8 Mengubah Data Diri

Mengubah Data Diri	
Objective	Dapat melakukan perubahan pada informasi data diri
Actor	Member
Pre-Condition	Berada di halaman <i>profile</i> .
Main Flow	1. Tombol " <i>Setting Profile</i> " dipilih 2. Sistem menampilkan halaman ubah data



Tabel 4.8 Mengubah Data Diri (Lanjutan)

Mengubah Data Diri	
	3. Pada halaman ubah data, dilakukan perubahan pada data-data yang diinginkan 4. Tombol “ <i>Save changes</i> ” ditekan 5. Sistem memproses data yang diubah
Alternative Flow	-
Post Condition	Perubahan berhasil dilakukan dan kembali ke halaman <i>my profile</i>

4.3.2.6. Melihat Halaman Rekomendasi

Tabel 4.9 merupakan *use case scenario* melihat halaman rekomendasi.

Tabel 4.9 Melihat Halaman Rekomendasi

Melihat Halaman Rekomendasi	
Objective	Halaman rekomendasi dapat dilihat
Actor	<i>Member</i>
Pre-Condition	Halaman utama sistem ditampilkan
Main Flow	1. Memilih menu <i>find recommendation</i>
Alternative Flow	-
Post Condition	Halaman <i>find recommendation</i> berhasil tampil

4.3.2.7. Mengisi Data Rekomendasi

Tabel 4.10 adalah penjelasan *use case scenario* mengisi data rekomendasi.

Tabel 4.10 Mengisi Data Rekomendasi

Mengisi Data Rekomendasi	
Objective	Aktor dapat mengisi data rekomendasi untuk mendapatkan rekomendasi
Actor	<i>Member</i>
Pre-Condition	halaman <i>find recommendation</i> ditampilkan
Main Flow	1. Memilih tipe kulit 2. Menekan tombol “ <i>Next</i> ” 3. Memilih <i>concern</i>



Tabel 4.10 Mengisi Data Rekomendasi (Lanjutan)

Mengisi Data Rekomendasi	
	4. Menekan tombol "Next" 5. Memilih kategori produk 6. Menekan tombol "Submit"
Alternative Flow	2.1. Jika tombol <i>next</i> ditekan tanpa memilih tipe kulit maka akan muncul pesan peringatan untuk memilih salah satu dari pilihan yang tersedia 4.1. Jika tombol <i>next</i> ditekan tanpa memilih <i>concern</i> maka akan muncul pesan peringatan untuk memilih salah satu dari pilihan yang tersedia 6.1. Jika tombol <i>submit</i> ditekan tanpa memilih kategori produk maka akan muncul pesan peringatan untuk memilih.
Post Condition	Berhasil mengisi data rekomendasi

4.3.2.8. Mengisi Nilai Perbandingan

Tabel 4.11 adalah penjelasan *use case scenario* mengisi nilai perbandingan.

Tabel 4.11 Mengisi Nilai Perbandingan

Mengisi Nilai Perbandingan	
Objective	Aktor dapat mengisi nilai perbandingan untuk mendapatkan rekomendasi
Actor	<i>Member</i>
Pre-Condition	Halaman <i>input AHP value</i> ditampilkan
Main Flow	1. Memilih nilai perbandingan
Alternative Flow	-
Post Condition	Nilai perbandingan berhasil diisi.

4.3.2.9. Melihat Daftar Rekomendasi

Tabel 4.12 adalah penjelasan *use case scenario* melihat daftar rekomendasi produk.



Tabel 4.12 Melihat Daftar Rekomendasi

Melihat Daftar Rekomendasi	
Objective	Aktor dapat melihat daftar-daftar produk yang direkomendasi sistem
Actor	<i>Member</i>
Pre-Condition	Aktor sudah mengisi semua data rekomendasi yang diperlukan di halaman <i>find recommendation</i> dan <i>input AHP value</i>
Main Flow	<ol style="list-style-type: none"> 1. Tombol “Submit” ditekan 2. Halaman hasil rekomendasi ditampilkan dengan daftar produk hasil dari rekomendasi sistem
Alternative Flow	1.1. Akan menampilkan pesan peringatan “Isi Isian Ini”, jika ada data yang kosong.
Post Condition	Daftar rekomendasi produk berhasil ditampilkan

4.3.2.10. Menulis Ulasan Produk

Tabel 4.13 adalah penjelasan *use case scenario* menulis ulasan produk.

Tabel 4.13 Menulis Ulasan Produk

Menulis Ulasan Produk	
Objective	Aktor dapat menulis ulasan tentang pengalaman menggunakan produk dan memberi <i>rating</i> ke produk
Actor	<i>Member</i>
Pre-Condition	Halaman detail produk ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Menulis ulasan tentang produk di kolom “Leave a Review” 2. Tombol “Submit” ditekan
Alternative Flow	2.1. Jika kolom ulasan tidak diisi dan <i>member</i> melakukan <i>submit</i> , maka muncul pesan peringatan “Isi Isian Ini”
Post Condition	Sistem menampilkan kembali halaman detail produk beserta ulasan yang telah dibuat

4.3.2.11. Menghapus Ulasan Produk

Tabel 4.14 adalah penjelasan *use case scenario* menghapus ulasan produk.



Tabel 4.14 Menghapus Ulasan Produk

Menghapus Ulasan Produk	
Objective	Ulasan produk dapat dihapus
Actor	<i>Member</i>
Pre-Condition	Halaman detail produk ditampilkan
Main Flow	1. Menekan tombol “Delete this review?” yang terdapat di kolom ulasan
Alternative Flow	-
Post Condition	Berhasil menghapus ulasan produk

4.3.2.12. Use Case Scenario Menulis Ulasan Artikel

Tabel 4.15 adalah penjelasan *use case scenario* menulis ulasan artikel.

Tabel 4.15 Menulis Ulasan Artikel

Menulis Ulasan Artikel	
Objective	Aktor dapat menulis ulasan tentang artikel
Actor	<i>Member</i>
Pre-Condition	Halaman detail artikel ditampilkan
Main Flow	1. Menulis ulasan tentang artikel yang telah dibaca di kolom “leave a Comment” 2. Tombol “Submit” ditekan
Alternative Flow	1.1. Jika kolom ulasan tidak diisi dan <i>member</i> melakukan <i>Submit</i> , maka akan muncul pesan peringatan “Isi Isian Ini”
Post Condition	Sistem menampilkan kembali halaman detail artikel beserta ulasan yang telah dibuat

4.3.2.13. Menghapus Ulasan Artikel

Tabel 4.16 adalah penjelasan *use case scenario* menghapus ulasan artikel.

Tabel 4.16 Menghapus Ulasan Artikel

Menghapus Ulasan Artikel	
Objective	Ulasan artikel dapat dihapus
Actor	<i>Member</i>
Pre-Condition	Halaman detail artikel ditampilkan



Tabel 4.16 Menghapus Ulasan Artikel (Lanjutan)

Menghapus Ulasan Artikel	
Main Flow	1. Menekan tombol " <i>Delete this comment?</i> " yang terdapat di kolom ulasan
Alternative Flow	-
Post Condition	Berhasil menghapus ulasan produk

4.3.2.14. Menyukai Artikel

Tabel 4.17 adalah penjelasan *use case scenario* menyukai artikel.

Tabel 4.17 Menyukai Artikel

Menyukai Artikel	
Objective	Aktor dapat memberi <i>like</i> pada artikel
Actor	<i>Member</i>
Pre-Condition	Halaman artikel ditampilkan
Main Flow	<ol style="list-style-type: none"> Memilih artikel Menekan tombol "<i>Read More</i>" Sitem menampilkan halaman detail artikel Aktor menekan tombol yang berbentuk <i>love</i> untuk memberi <i>like</i>
Alternative Flow	-
Post Condition	Sistem menampilkan kembali halaman detail produk beserta jumlah <i>like</i>

4.3.2.15. Membatalkan Menyukai Artikel

Tabel 4.18 adalah penjelasan *use case scenario* membatalkan menyukai artikel.

Tabel 4.18 Membatalkan Menyukai Artikel

Membatalkan Menyukai Artikel	
Objective	Aktor dapat membatalkan <i>like</i> pada artikel yang telah disukai sebelumnya
Actor	<i>Member</i>
Pre-Condition	Halaman detail artikel ditampilkan dan telah memberi <i>like</i>
Main Flow	<ol style="list-style-type: none"> Menekan lagi tombol berbentuk <i>love</i>



Tabel 4.18 Membatalkan Menyukai Artikel (Lanjutan)

Membatalkan Menyukai Artikel	
Alternative Flow	-
Post Condition	Berhasil membatalkan <i>like</i> dan kembali halaman detail produk beserta jumlah <i>like</i> yang telah berkurang

4.3.2.16. Melihat Daftar Pengguna

Tabel 4.19 adalah penjelasan *use case scenario* melihat daftar pengguna.

Tabel 4.19 Melihat Daftar Pengguna

Melihat Daftar Pengguna	
Objective	Dapat melihat daftar pengguna
Actor	Admin
Pre-Condition	Halaman utama admin ditampilkan
Main Flow	1. Memilih menu data pengguna pada <i>sidebar menu</i>
Alternative Flow	-
Post Condition	Berhasil masuk ke halaman daftar pengguna

4.3.2.17. Menambah Data Produk

Tabel 4.20 adalah penjelasan *use case scenario* menambah data produk.

Tabel 4.20 Menambah Data Produk

Menambah Data Produk	
Objective	Data produk dapat ditambah
Actor	Admin
Pre-Condition	Halaman utama admin ditampilkan
Main Flow	<ol style="list-style-type: none"> 1. Memilih menu data produk data <i>sidebar menu</i> 2. Menekan tombol "<i>add new</i>" 3. Sistem menampilkan halaman tambah data 4. Mengisi <i>field-field</i> yang tersedia 5. Tombol "<i>simpan</i>" ditekan 6. Sistem berhasil menyimpan data
Alternative Flow	5.1. Jika ada <i>field</i> yang kosong, sistem memberi pesan agar <i>field</i> yang kosong dapat segera diisi



Tabel 4.20 Menambah Data Produk (Lanjutan)

Menambah Data Produk	
Post Condition	Data berhasil tersimpan dan kembali ke halaman daftar produk

4.3.2.18. Melihat Daftar Produk

Tabel 4.21 adalah penjelasan *use case scenario* melihat daftar semua produk.

Tabel 4.21 Melihat Daftar Produk

Melihat Daftar Produk	
Objective	Daftar semua produk <i>skincare</i> dapat ditampilkan dengan berbagai kategori
Actor	Admin
Pre-Condition	Halaman utama admin ditampilkan
Main Flow	1. Memilih menu data produk pada <i>sidebar menu</i>
Alternative Flow	-
Post Condition	Halaman daftar produk tampil

4.3.2.19. Mengubah Data Produk

Tabel 4.22 adalah penjelasan *use case scenario* mengubah data produk.

Tabel 4.22 Mengubah Data Produk

Mengubah Data Produk	
Objective	Informasi produk dapat diubah
Actor	Admin
Pre-Condition	Halaman utama admin tampil
Main Flow	<ol style="list-style-type: none"> 1. Memilih menu data produk pada <i>sidebar menu</i> 2. Menekan tombol "Edit" 3. Mengubah data pada <i>field-field</i> yang tersedia 4. Menekan tombol "Save"
Alternative Flow	-
Post Condition	Data diubah dan kembali ke halaman data produk.



4.3.2.20. Menghapus Data Produk

Tabel 4.23 adalah penjelasan *use case scenario* menghapus data produk.

Tabel 4.23 Menghapus Data Produk

Menghapus Data Produk	
Objective	Data produk dapat dihapus
Actor	Admin
Pre-Condition	Halaman utama admin tampil
Main Flow	<ol style="list-style-type: none"> 1. Memilih menu data produk pada <i>sidebar menu</i> 2. Memilih produk yang ingin dihapus 3. Tombol "Hapus" ditekan 4. Pesan konfirmasi ditampilkan 5. Aktor memilih "Ya"
Alternative Flow	-
Post Condition	Data produk berhasil dihapus dan kembali ke halaman data produk

4.3.2.21. Menambah Data Artikel

Tabel 4.24 adalah penjelasan *use case scenario* menambah artikel.

Tabel 4.24 Menambah Artikel

Menambah Artikel	
Objective	Aktor dapat menambah artikel produk <i>skincare</i> atau sejenisnya yang berkaitan dengan <i>skincare</i>
Actor	Admin
Pre-Condition	Aktor berada pada halaman utama admin
Main Flow	<ol style="list-style-type: none"> 1. Memilih menu data artikel data <i>sidebar menu</i> 2. Halaman daftar artikel ditampilkan 3. Tombol "add new" ditekan 4. Halaman tambah data artikel ditampilkan 5. Mengisi <i>field-field</i> yang tersedia 6. Tombol "save" ditekan
Alternative Flow	5.1. Jika ada field yang kosong, maka pesan peringatan ditampilkan agar field yang kosong dapat segera diisi



Tabel 4.24 Menambah Artikel (Lanjutan)

Menambah Artikel	
Post Condition	Artikel berhasil tersimpan dan kembali ke halaman daftar artikel

4.3.2.22. Melihat Daftar Artikel

Tabel 4.25 adalah penjelasan *use case scenario* melihat daftar semua artikel.

Tabel 4.25 Melihat Daftar Produk

Melihat Daftar Produk	
Objective	Aktor dapat melihat daftar semua produk <i>skincare</i> dengan berbagai kategori
Actor	Admin
Pre-Condition	Halaman utama admin tampil
Main Flow	1. Aktor memilih menu data artikel pada <i>sidebar menu</i>
Alternative Flow	-
Post Condition	Halaman daftar artikel ditampilkan

4.3.2.23. Mengubah Data Artikel

Tabel 4.26 adalah penjelasan *use case scenario* mengubah data artikel.

Tabel 4.26 Mengubah Artikel

Mengubah Artikel	
Objective	Aktor dapat mengubah artikel jika terdapat kesalahan atau tambahan data
Actor	Admin
Pre-Condition	Halaman utama admin tampil
Main Flow	<ol style="list-style-type: none"> Memilih menu data artikel pada <i>sidebar menu</i> Menekan tombol "Edit" Mengubah data pada <i>field-field</i> yang tersedia Tombol "Save" ditekan
Alternative Flow	-
Post Condition	Data berhasil diubah dan masuk ke halaman data artikel



4.3.2.24. Menghapus Data Artikel

Tabel 4.27 adalah penjelasan *use case scenario* menghapus artikel *skincare*.

Tabel 4.27 Menghapus Artikel

Menghapus Artikel	
Objective	Artikel dapat dihapus.
Actor	Admin
Pre-Condition	Halaman utama admin tampil
Main Flow	<ol style="list-style-type: none"> 1. Memilih menu data artikel pada <i>sidebar menu</i> 2. Sistem menampilkan halaman artikel 3. Memilih artikel yang akan dihapus 4. Tombol “hapus” ditekan 5. Sistem memunculkan pesan peringatan 6. Memilih “Ya”
Alternative Flow	-
Post Condition	Artikel berhasil dihapus

4.3.2.25. Melihat Produk

Tabel 4.28 adalah penjelasan *use case scenario* melihat produk.

Tabel 4.28 Melihat Produk

Melihat Produk	
Objective	Aktor dapat melihat detail informasi dari produk yang dipilih
Actor	<i>Guest, Member</i>
Pre-Condition	Halaman produk tampil
Main Flow	<ol style="list-style-type: none"> 1. Pilih salah satu produk 2. Menekan tombol “<i>View detail</i>”
Alternative Flow	-
Post Condition	Berhasil masuk ke halaman detail produk

4.3.2.26. Melihat Artikel

Tabel 4.29 adalah penjelasan *use case scenario* untuk *use case* melihat artikel.



Tabel 4.29 Melihat Artikel

Melihat Detail Artikel	
Objective	Aktor dapat melihat detail artikel yang dipilih
Actor	<i>Guest, Member</i>
Pre-Condition	Halaman daftar artikel tampil
Main Flow	<ol style="list-style-type: none"> 1. Pilih artikel yang ingin dibaca 2. Tombol “<i>read more</i>” ditekan
Alternative Flow	-
Post Condition	Berhasil masuk ke halaman detail artikel

4.3.2.27. Mencari Produk

Tabel 4.30 adalah penjelasan *use case scenario* mencari produk.

Tabel 4.30 Mencari Produk

Mencari Produk	
Objective	Pencarian produk dapat dilakukan
Actor	<i>Member dan Guest</i>
Pre-Condition	halaman utama sistem tampil
Main Flow	<ol style="list-style-type: none"> 1. Mengisi <i>field</i> pencarian dengan <i>keyword</i> 2. Menekan <i>enter</i> pada <i>keyboard</i>
Alternative Flow	<ol style="list-style-type: none"> 2.1. Pesan peringatan “Isi isian ini” akan tampil jika <i>field</i> kosong 2.2. Jika ada <i>keyword</i> yang tidak ditemukan, maka pesan “Data Tidak Ditemukan” ditampilkan
Post Condition	Pencarian berhasil dan menampilkan halaman hasil pencarian

4.3.2.28. Mencari Artikel

Tabel 4.31 adalah penjelasan *use case scenario* mencari artikel.

Tabel 4.31 Mencari Artikel

Mencari Artikel	
Objective	Pencarian artikel dapat dilakukan
Actor	<i>Member dan Guest</i>



Tabel 4.31 Mencari Artikel (Lanjutan)

Mencari Artikel	
Pre-Condition	Aktor berada dalam halaman utama sistem
Main Flow	<ol style="list-style-type: none"> Memilih menu <i>Article</i> Halaman artikel ditampilkan Mengisi <i>field</i> pencarian dengan <i>keyword</i> Menekan <i>enter</i> pada <i>keyboard</i>
Alternative Flow	<ol style="list-style-type: none"> 4.1. Pesan peringatan “Isi isian ini” akan tampil jika <i>field</i> kosong 4.2. Jika ada <i>keyword</i> yang tidak ditemukan, maka pesan “Data Tidak Ditemukan” ditampilkan
Post Condition	Pencarian berhasil dan menampilkan halaman hasil pencarian

4.3.2.29. Melihat Halaman *About Us*

Tabel 4.32 adalah penjelasan *use case scenario* melihat halaman *about us*.

Tabel 4.32 Melihat Halaman *About Us*

Melihat Halaman <i>About Us</i>	
Objective	Halaman <i>about us</i> ditampilkan
Actor	<i>Member</i> dan <i>Guest</i>
Pre-Condition	halaman utama sistem tampil
Main Flow	1. Memilih menu <i>About Us</i>
Alternative Flow	-
Post Condition	Halaman <i>About Us</i> ditampilkan

4.3.2.30. Melihat Halaman FAQ

Tabel 4.33 adalah *use case scenario* melihat halaman FAQ.

Tabel 4.33 Melihat Halaman FAQ

Melihat Halaman FAQ	
Objective	Halaman FAQ dapat ditampilkan
Actor	<i>Member</i> dan <i>Guest</i>
Pre-Condition	Halaman utama sistem tampil



Tabel 4.33 Melihat Halaman FAQ (Lanjutan)

Melihat Halaman FAQ	
Main Flow	1. Memilih menu FAQ yang berada di <i>footer</i>
Alternative Flow	-
Post Condition	Halaman FAQ ditampilkan

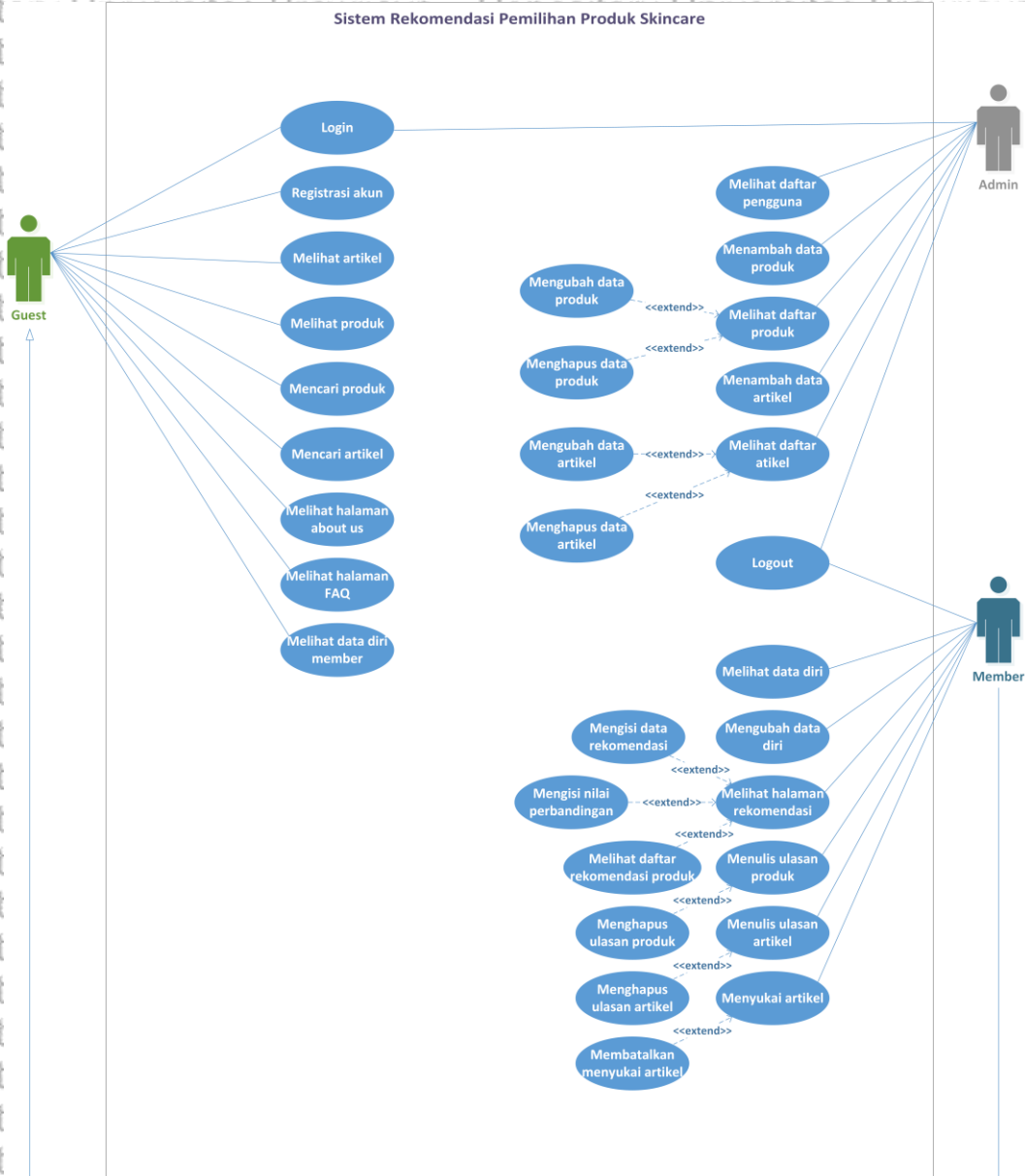
4.3.2.31. Melihat Data Diri *Member* lain

Tabel 4.34 adalah penjelasan *use case scenario* melihat data diri *member* lain.

Tabel 4.34 Melihat Data Diri *Member* lain

Melihat Data Diri <i>Member</i> lain	
Objective	Aktor dapat melihat informasi data diri <i>member</i> lain
Actor	<i>Member</i> dan <i>Guest</i>
Pre-Condition	Halaman detail produk atau artikel tampil
Main Flow	1. Memilih <i>member</i> yang akan dilihat datanya 2. Menekan nama dari <i>member</i> tersebut
Alternative Flow	-
Post Condition	Sistem menampilkan halaman profil dari <i>member</i> yang dipilih

Sistem Rekomendasi Pemilihan Produk Skincare



Gambar 4.2 Use Case Diagram

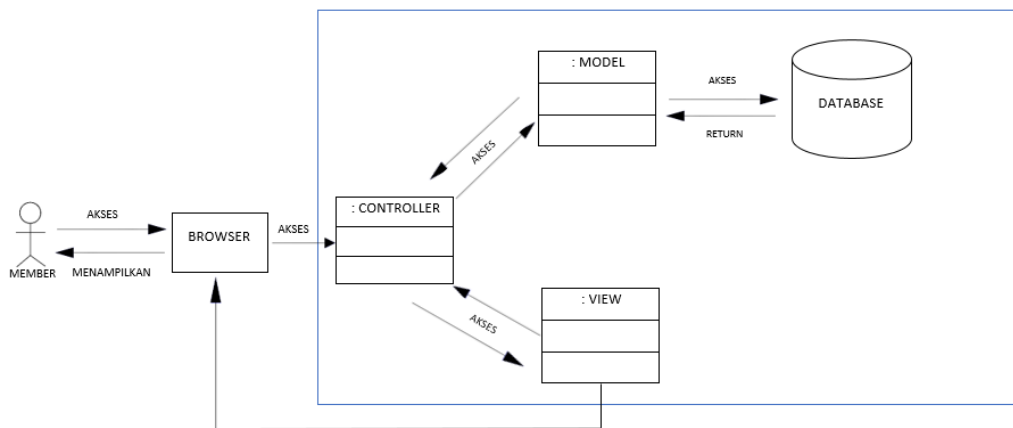
BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Tahap yang dilakukan setelah rekayasa kebutuhan adalah tahap perancangan. Perancangan ini bertujuan untuk memberikan gambaran sistem yang dibangun. Perancangan pada sistem dalam penelitian ini dibagi menjadi beberapa bagian di antaranya adalah perancangan arsitektur, perancangan *sequence diagram*, perancangan *class diagram*, perancangan komponen, perancangan data, dan perancangan antarmuka.

5.1.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem digunakan untuk menjelaskan komponen struktur utama dari sistem dan hubungan antar komponen-komponen tersebut. Arsitektur sistem yang akan dirancang dapat dilihat pada Gambar 5.1. Pada gambar tersebut, menjelaskan perancangan arsitektur sistem yang dibangun menggunakan *framework* CI dengan menggunakan pola MVC (*Model-View-Controller*).



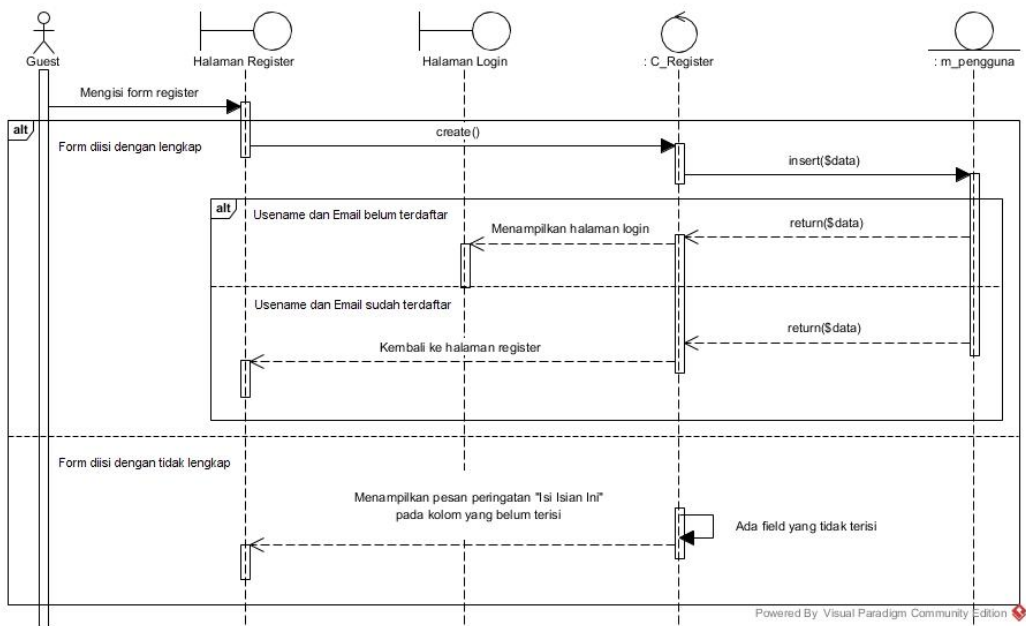
Gambar 5.1 Arsitektur Sistem Rekomendasi Pemilihan Produk Skincare

5.1.2 Perancangan *Sequence Diagram*

Sequence diagram dibuat berdasarkan *use case scenario* yang telah dibuat sebelumnya. *Sequence diagram* yang akan dibuat berjumlah tiga. *Sequence diagram* tersebut adalah registrasi akun, melihat daftar rekomendasi, dan mencari produk.

5.1.2.1 *Sequence Diagram* Registrasi Akun

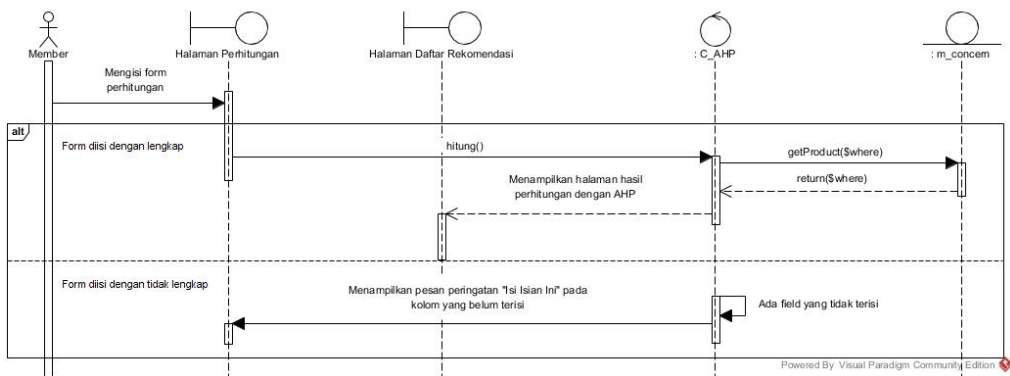
Gambar 5.2 adalah *sequence diagram* untuk registrasi akun. Objek yang terlibat adalah dua *boundary* yaitu halaman *register* dan halaman *login*, satu *control* yaitu C_Register dan satu *entity* yaitu m_pengguna.



Gambar 5.2 Sequence Diagram Registrasi Akun

5.1.2.2 Sequence Diagram Melihat Daftar Rekomendasi

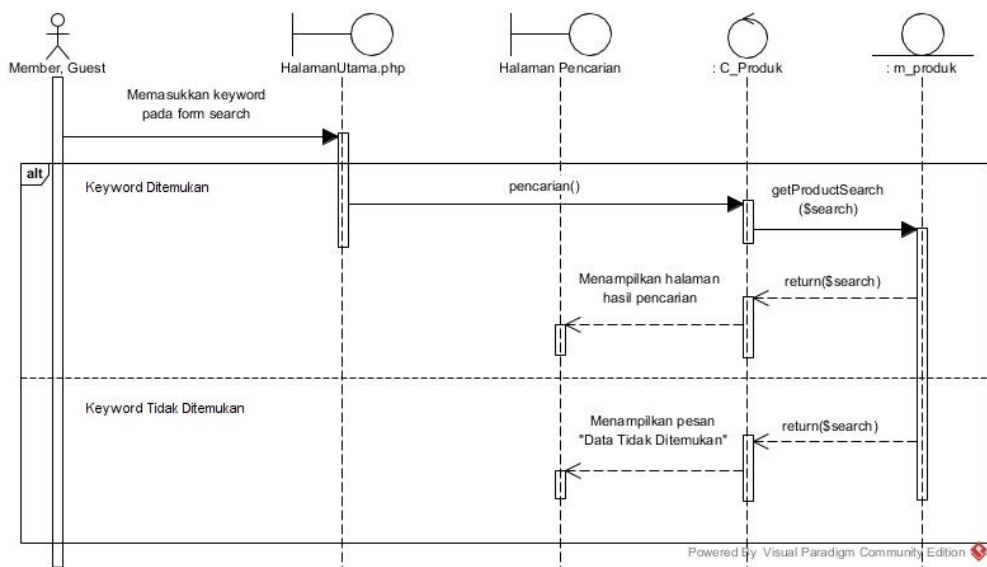
Gambar 5.3 adalah *sequence diagram* untuk melihat daftar rekomendasi. Objek yang terlibat adalah dua *boundary* yaitu halaman perhitungan dan halaman daftar rekomendasi, satu *control* yaitu C_AHP dan satu *entity* yaitu m_concern.



Gambar 5.3 Sequence Diagram Melihat Daftar Rekomendasi

5.1.2.3 Sequence Diagram Mencari Produk

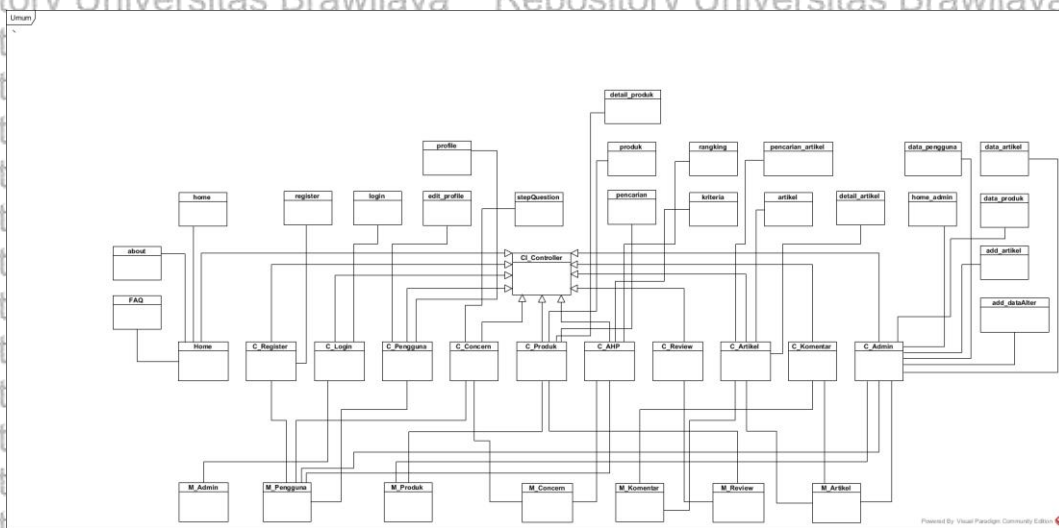
Gambar 5.4 adalah *sequence diagram* untuk mencari produk. Objek yang terlibat adalah dua *boundary* yaitu halaman utaman dan halaman pencarian, satu *control* yaitu C_Produk dan satu *entity* yaitu m_produk.



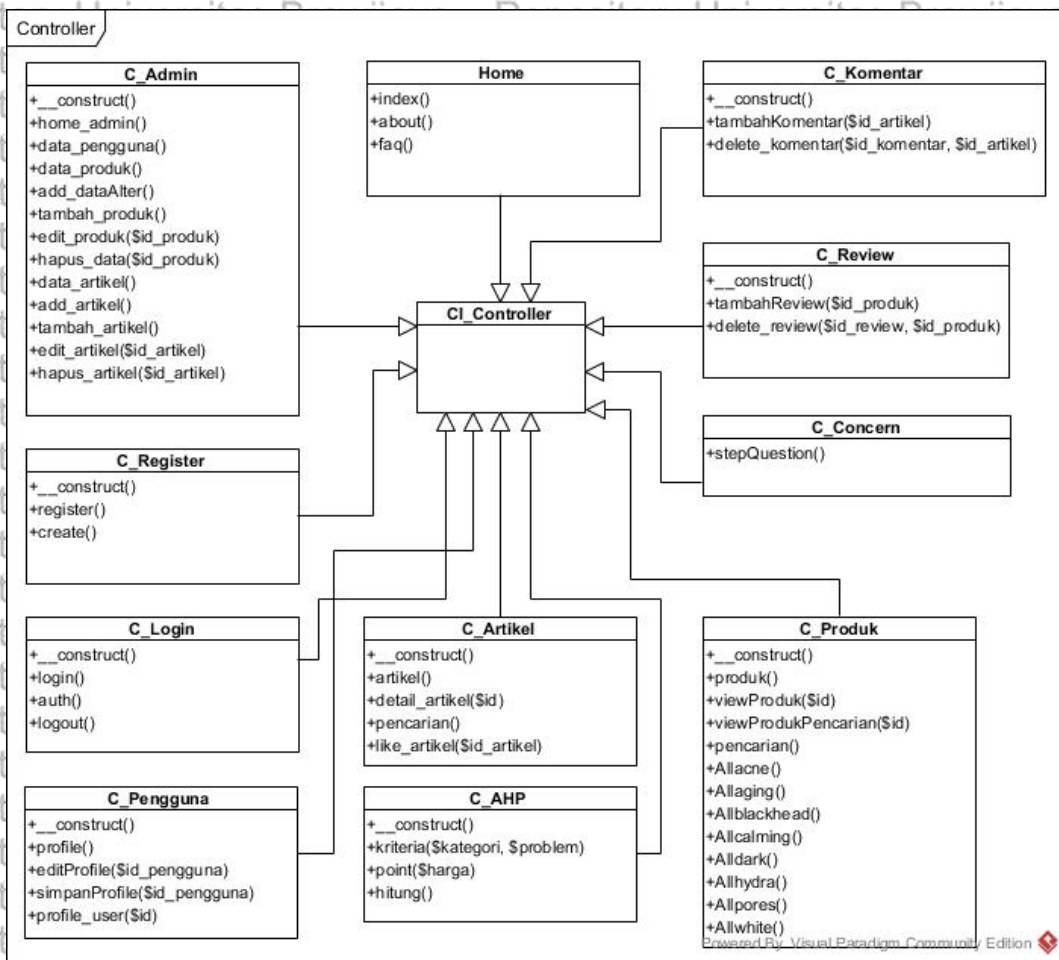
Gambar 5.4 Sequence Diagram Mencari Produk

5.1.3 Perancangan Class Diagram

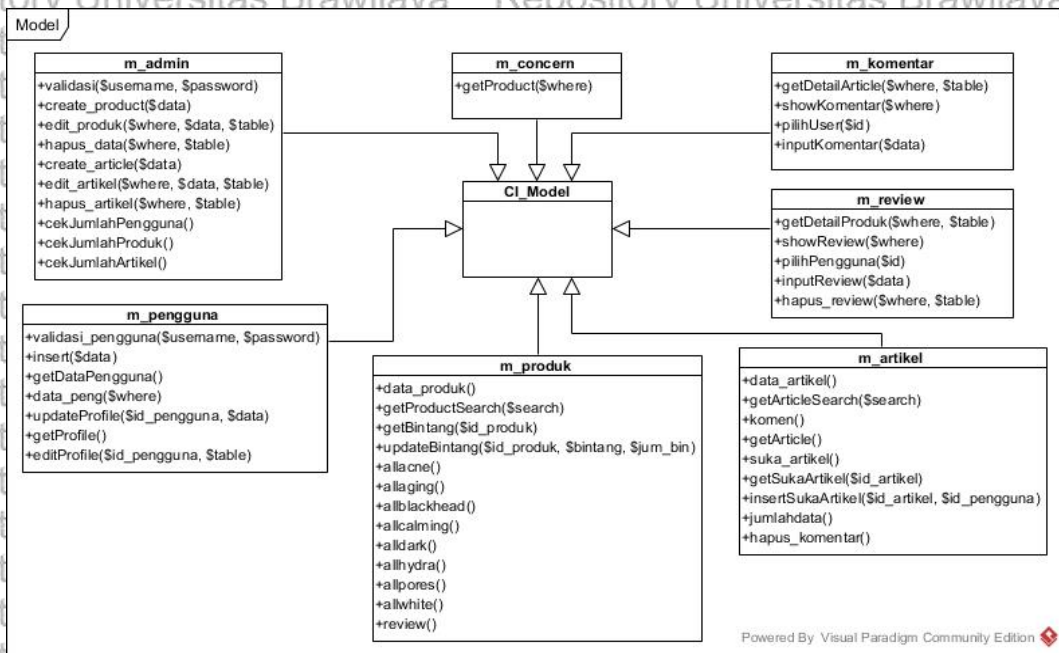
Perancangan *class diagram* secara umum dapat dilihat pada Gambar 5.5. *CI_Controller* dan *CI_Model* merupakan kelas utama. Perancangan kelas *CI_Controller* dan *CI_Model* dapat dilihat pada Gambar 5.6 dan Gambar 5.7. Kelas *CI_Controller* mempunyai kelas turunan yaitu *C_Admin*, *C_AHP*, *C_Artikel*, *C_Concern*, *C_Komentar*, *C_login*, *C_Pengguna*, *C_Produk*, *C_Register*, *C_Review*, *Home*. Kelas *CI_Model* mempunyai kelas turunan *m_admin*, *m_artikel*, *m_concern*, *m_komentar*, *m_pengguna*, *m_produk*, *m_review*. Hubungan antara kelas turunan *CI_Controller* dengan kelas turunan *CI_Model* adalah hubungan asosiasi.



Gambar 5.5 Perancangan Umum Class Diagram



Gambar 5.6 Perancangan Kelas CI_Controller

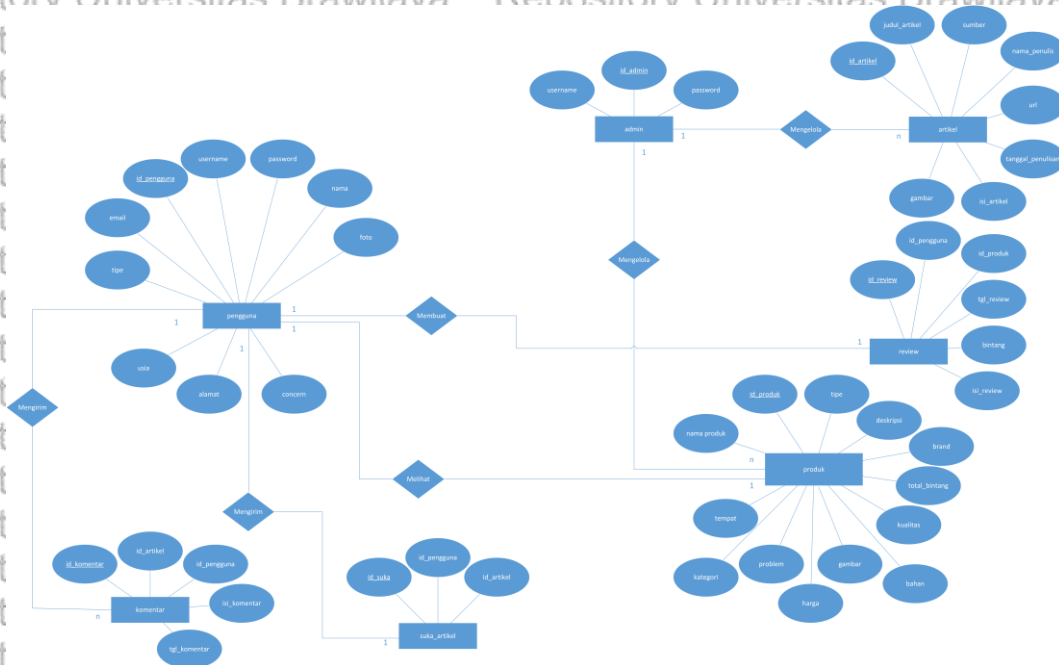


Gambar 5.7 Perancangan Kelas CI_Model



5.1.4 Perancangan Data

Perancangan data dimodelkan dalam bentuk *Entity Relationship Diagram* (ERD). ERD diilustrasikan pada Gambar 5.8. Terdapat 7 (tujuh) entitas yaitu admin, pengguna, artikel, produk, review, komentar, suka_artikel.



Gambar 5.8 Entity Relationship Diagram (ERD)

5.1.5 Perancangan Komponen

Perancangan akan dibuat dalam bentuk algoritme. Algoritme yang akan dibuat adalah create() dari klas C_Register, tambah_produk() dari klas C_Admin, dan simpanProfile dari klas C_Pengguna.

5.1.5.1 Perancangan Algoritme Method create

Tabel 5.1 merupakan algoritme *method* create() dari klas C_Register.

Tabel 5.1 Algoritme Method create

No	Uraian
1	Mulai method create
2	Inisialisasi rules untuk validasi email
3	Inisialisasi rules untuk validasi username
4	IF form validation adalah false
5	Mengalihkan ke halaman register
6	ELSE
7	Inisialisasi variabel data dalam bentuk array
8	Memanggil fungsi insert di model m_pengguna
9	Menampilkan message success
10	Mengalihkan ke halaman login
11	ENDIF
12	Selesai



5.1.5.2 Algoritme *Method* tambah_produk

Tabel 5.2 adalah algoritme *method* tambah_produk() dari kelas C_Admin.

Tabel 5.2 Algoritme *Method* tambah_produk

No	Uraian
1	Mulai method tambah_produk
2	Inisialisasi rules untuk validasi brand
3	Inisialisasi rules untuk validasi nama produk
4	Inisialisasi rules untuk validasi deskripsi
5	Inisialisasi rules untuk validasi harga
6	Inisialisasi rules untuk validasi kualitas
7	Inisialisasi rules untuk validasi bahan
8	Inisialisasi rules untuk validasi tempat
9	IF form validation adaTah false
10	Menampilkan halaman add_dataAlter
11	Else
12	Membuat konfigurasi untuk upload gambar
13	Inisialisasi variabel data dalam bentuk array
14	Memanggil fungsi create_product di model m_admin
15	Mengahlika ke halaman data_produk
16	ENDIF
17	Selesai

5.1.5.3 Algoritme *Method* simpanProfile

Tabel 5.3 adalah algoritme *method* simpanProfile() dari kelas C_Pengguna.

Tabel 5.3 Algoritme *Method* simpanProfile

No	Uraian
1	Mulai method simpanProfile sesuai dengan id_pengguna
2	Mengambil data dari form edit
3	Memanggil library upload
4	Membuat konfigurasi untuk upload gambar
5	Menginisialisasi variabel config
6	If upload gambar baru
7	Inisialisasi gambar dengan variabel baru
8	Else
9	Inisialisasi gambar dengan variabel lama
10	ENDIF
11	Inisialisasi variabel data dalam bentuk array
12	Memanggil fungsi updateProfile di model m_pengguna berdasarkan variabel id_pengguna dan data
13	Selesai

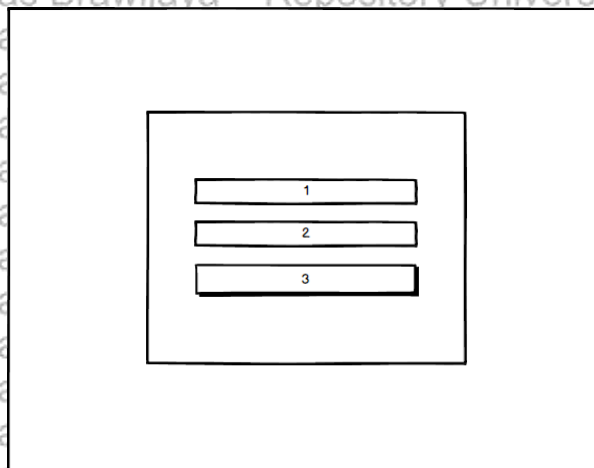
5.1.6 Perancangan Antarmuka

Perancangan antarmuka merupakan perancangan tampilan dari sistem yang akan dibuat. Dalam perancangan antarmuka ini akan ditampilkan 5 (lima) rancangan. Di antaranya adalah antarmuka halaman *login*, antarmuka halaman utama sistem, antarmuka halaman *find* recommendation, antarmuka halaman utama admin, dan antarmuka halaman tambah produk.



5.1.6.1 Antarmuka Halaman *Login*

Perancangan antarmuka halaman *login* akan dijelaskan pada Gambar 5.9 dan penjelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.4.



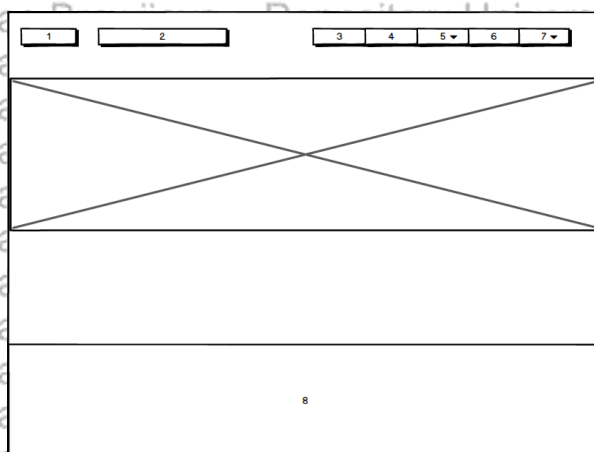
Gambar 5.9 Antarmuka Halaman *login*

Tabel 5.4 Penjelasan Antarmuka Halaman *login*

No.	Nama Objek	Tipe	Keterangan
1	<i>Username</i>	<i>Text Field</i>	Berfungsi untuk memasukkan <i>username member</i>
2	<i>Password</i>	<i>Text Field</i>	Berfungsi untuk memasukkan <i>password member</i>
3	Tombol <i>login</i>	<i>Button</i>	Tombol untuk masuk kedalam halaman utama sistem.

5.1.6.2 Antarmuka Halaman Utama Sistem

Perancangan antarmuka halaman utama sistem akan dijelaskan pada Gambar 5.10 dan penjelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.5.



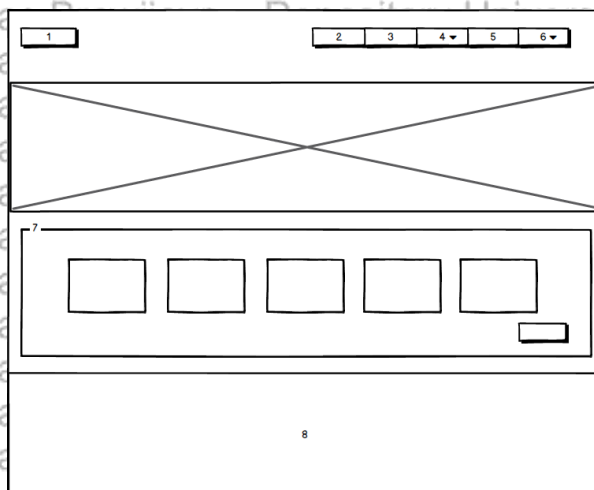
Gambar 5.10 Antarmuka Halaman Utama Sistem

Tabel 5.5 Penjelasan Antarmuka Halaman Utama Sistem

No.	Nama Objek	Tipe	Keterangan
1	Nama Aplikasi	Text Button	Berfungsi untuk menampilkan nama aplikasi
2	Search	Text Field	Berfungsi untuk memasukkan kata kunci pencarian
3	Menu <i>About Us</i>	Menu <i>Tab</i>	Berfungsi untuk menuju halaman tentang aplikasi
4	Menu <i>Find Recommendation</i>	Menu <i>Tab</i>	Berfungsi untuk menuju halaman rekomendasi
5	Menu <i>Products</i>	Menu <i>dropdown</i>	Berfungsi untuk menampilkan <i>sub-menu</i> dari menu <i>products</i>
6	Menu <i>Article</i>	Menu <i>Tab</i>	Berfungsi untuk menuju halaman daftar artikel
7	Menu <i>Account</i>	Menu <i>dropdown</i>	Berfungsi untuk menampilkan <i>sub-menu</i> dari menu <i>account</i>
8	Footer	Footer	Berfungsi untuk menampilkan beberapa informasi mengenai aplikasi seperti nama aplikasi dan deskripsi singkat aplikasi

5.1.6.3 Antarmuka Halaman *Find Recommendation*

Perancangan antarmuka halaman *find recommendation* akan dijelaskan pada Gambar 5.11 dan penjelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.6.



Gambar 5.11 Antarmuka Halaman *Find Recommendation*

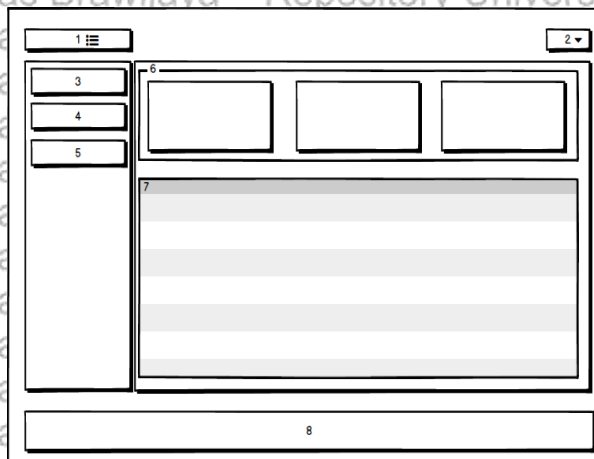
Tabel 5.6 Penjelasan Antarmuka Halaman *Find Recommendation*

No.	Nama Objek	Tipe	Keterangan
1	Nama Aplikasi	<i>Text Button</i>	Berfungsi untuk menampilkan nama aplikasi
2	<i>Search</i>	<i>Text Field</i>	Berfungsi untuk memasukkan kata kunci pencarian
3	Menu <i>About Us</i>	<i>Menu Tab</i>	Berfungsi untuk menuju halaman tentang aplikasi
4	Menu <i>Find Recommendation</i>	<i>Menu Tab</i>	Berfungsi untuk menuju halaman rekomendasi
5	Menu <i>Products</i>	<i>Menu dropdown</i>	Berfungsi untuk menampilkan <i>sub-menu</i> dari menu <i>products</i>
6	Menu <i>Article</i>	<i>Menu Tab</i>	Berfungsi untuk menuju halaman daftar artikel
7	Tipe Kulit, Permasalahan Kulit, dan Kategori Produk	<i>Multi-Page</i>	Berfungsi untuk menampilkan multi halaman yang terdiri dari halaman tipe kulit, halaman permasalahan kulit, dan halaman kategori produk
8	<i>Footer</i>	<i>Footer</i>	Berfungsi untuk menampilkan beberapa informasi mengenai aplikasi seperti nama aplikasi dan deskripsi singkat aplikasi



5.1.6.4 Antarmuka Halaman Utama Admin

Perancangan antarmuka halaman utama admin akan dijelaskan pada Gambar 5.12 dan penjelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.7.



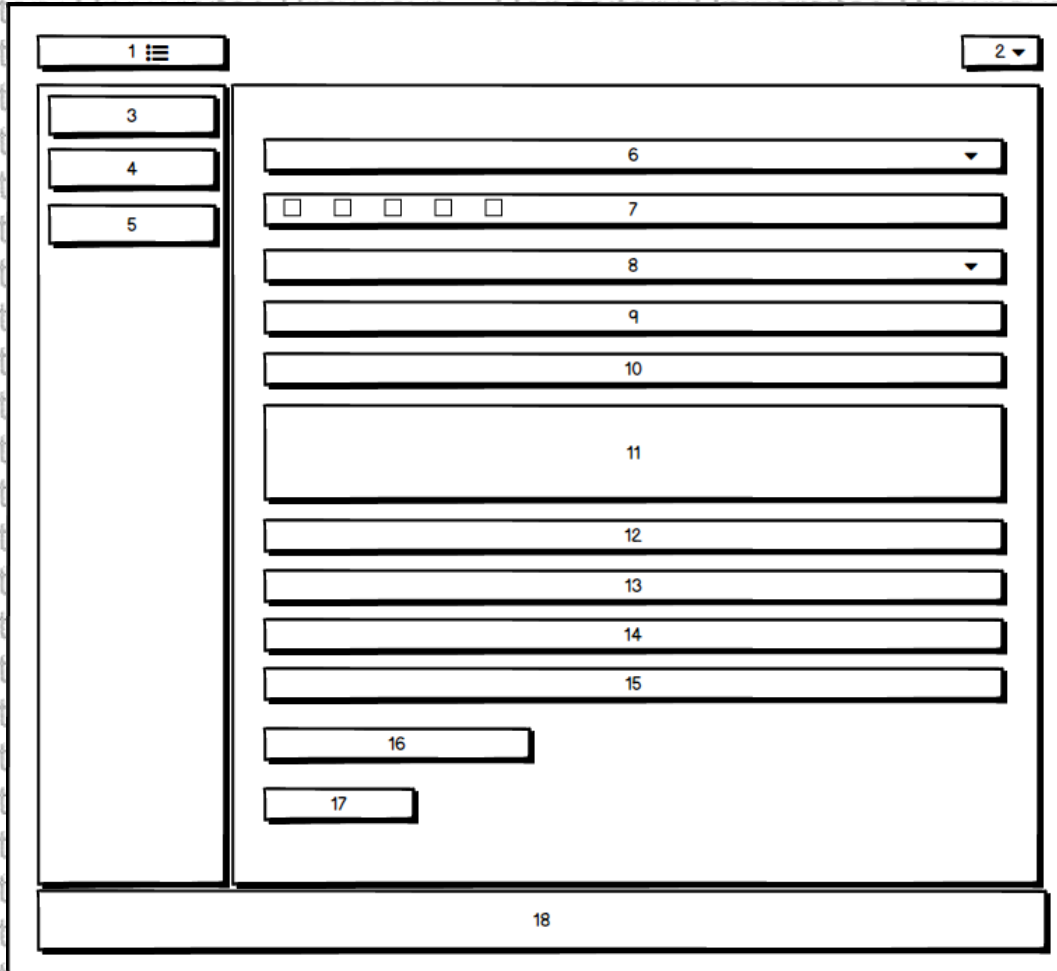
Gambar 5.12 Antarmuka Halaman Utama Admin

Tabel 5.7 Penjelasan Antarmuka Halaman Utama Admin

No.	Nama Objek	Type	Keterangan
1	<i>Navigation Bar</i>	<i>Button</i>	Berfungsi untuk menampilkan dan menyembunyikan menu <i>sidebar</i>
2	Menu <i>Account</i>	Menu <i>dropdown</i>	Berfungsi untuk menampilkan <i>sub-menu</i> dari menu <i>account</i>
3	Menu <i>Daftar Pengguna</i>	Menu <i>sidebar</i>	Berfungsi untuk menuju halaman daftar pengguna
4	Menu <i>Daftar Produk</i>	Menu <i>sidebar</i>	Berfungsi untuk menuju halaman daftar produk
5	Menu <i>Daftar Artikel</i>	Menu <i>sidebar</i>	Berfungsi untuk menuju halaman daftar artikel
6	Menu <i>Overview</i>	Menu <i>bar</i>	Berfungsi untuk menampilkan <i>shortcut</i> dari menu yang ada di <i>sidebar</i>
7	<i>Table Data</i>	<i>Table</i>	Berfungsi untuk menampilkan <i>table data</i>
8	<i>Footer</i>	<i>Footer</i>	Berfungsi untuk menampilkan <i>footer</i>

5.1.6.5 Antarmuka Halaman Tambah Produk

Perancangan antarmuka halaman tambah produk akan dijelaskan pada Gambar 5.13 dan penjelasan detail dari perancangan antarmuka tersebut akan dijelaskan pada Tabel 5.8.



Gambar 5.13 Antarmuka Halaman Tambah Produk

Tabel 5.8 Penjelasan Antarmuka Tambah Produk

No.	Nama Objek	Tipe	Keterangan
1	Navigation Bar	Button	Berfungsi untuk menampilkan dan menyembunyikan menu sidebar
2	Menu Account	Menu dropdown	Berfungsi untuk menampilkan sub-menu dari menu account
3	Menu Daftar Pengguna	Menu sidebar	Berfungsi untuk menuju halaman daftar pengguna
4	Menu Daftar Produk	Menu sidebar	Berfungsi untuk menuju halaman daftar produk
5	Menu Daftar Artikel	Menu sidebar	Berfungsi untuk menuju halaman daftar artikel
6	Permasalahan Kulit	Menu dropdown	Berfungsi untuk memilih permasalahan kulit yang ada



Tabel 5.8 Penjelasan Antarmuka Tambah Produk (Lanjutan)

No.	Nama Objek	Tipe	Keterangan
7	Tipe Kulit	<i>Checkbox</i>	Berfungsi untuk memilih tipe kulit yang ada
8	Kategori Produk	Menu <i>dropdown</i>	Berfungsi untuk memilih kategori produk yang ada
9	Nama <i>Brand</i>	<i>Text Field</i>	Berfungsi untuk memasukkan nama <i>brand</i> produk
10	Nama Produk	<i>Text Field</i>	Berfungsi untuk memasukkan nama produk
11	Deskripsi	<i>Text Field</i>	Berfungsi untuk memasukkan deskripsi produk
12	Harga	<i>Text Field</i>	Berfungsi untuk memasukkan harga produk
13	Kualitas	<i>Text Field</i>	Berfungsi untuk memasukkan nilai kualitas
14	Bahan	<i>Text Field</i>	Berfungsi untuk memasukkan nilai bahan
15	Tempat Pembelian	<i>Text Field</i>	Berfungsi untuk memasukkan tempat pembelian produk
16	Gambar Produk	<i>Button</i>	Berfungsi untuk berfungsi untuk mengunggah gambar produk
17	Tombol <i>save</i>	<i>Button</i>	Berfungsi untuk menyimpan data produk
18	<i>Footer</i>	<i>Footer</i>	Berfungsi untuk menampilkan <i>footer</i>

5.2 Implementasi

Tahap selanjutnya adalah tahap implementasi. Tujuan dari tahap ini adalah mengimplementasikan hasil perancangan sistem yang telah dibuat.

5.2.1 Spesifikasi Sistem

5.2.1.1 Spesifikasi Perangkat Keras

Dalam membangun sistem ini, berikut diuraikan dalam Tabel 5.9 daftar *hardware* yang digunakan.

Tabel 5.9 Spesifikasi Perangkat Keras

Komponen	Spesifikasi
<i>System Model</i>	Aspire E-112M



Tabel 5.9 Spesifikasi Perangkat Keras (Lanjutan)

Komponen	Spesifikasi
Processor	Intel® Celeron® CPU @ 2.16GHz (2 CPUs), ~2.2GHz
Memory	2.00 GB RAM
Display	Intel® HD Graphics

5.2.1.2 Spesifikasi Perangkat Lunak

Dalam membangun sistem ini, berikut diuraikan dalam Tabel 5.10 *software* yang digunakan.

Tabel 5.10 Spesifikasi Perangkat Lunak

Komponen	Spesifikasi
Operating System	Windows 8.1 Pro 64-bit
Browser	Firefox, Google Chrome
Web Server	Apache 2.0
Framework	CodeIgniter
Markup language	HTML, CSS, JavaScript
Programming	PHP
Database	MySQL
Editor Script	Sublime Text 3, Notepad
Editor Design	Visual Paradigm 15.1, Visio 2013, Balsamiq Mockups 3

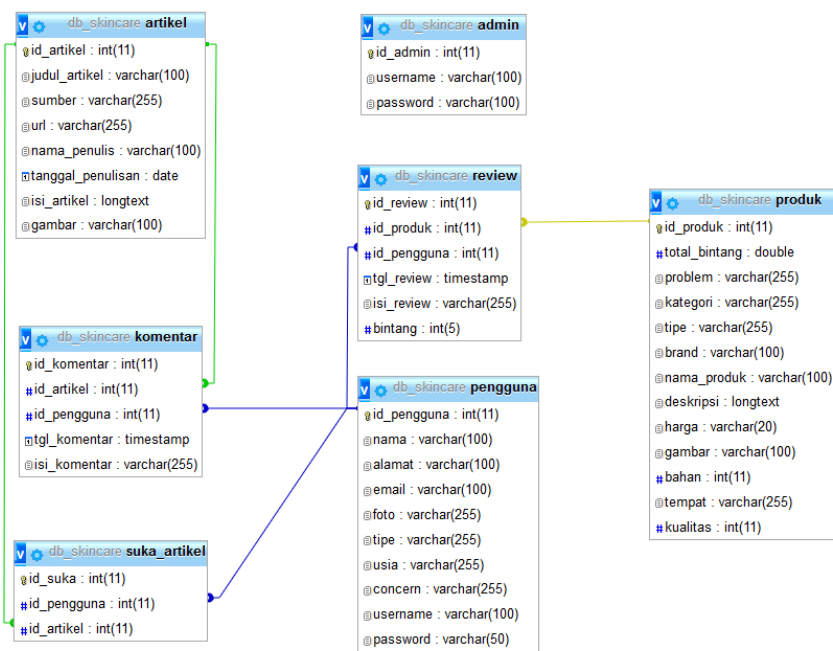
5.2.2 Batasan Implementasi

Batasan pada implementasi adalah:

1. Bahasa pemrograman yang digunakan dalam membangun sistem adalah bahasa pemrograman PHP yang diimplementasikan pada *framework* CodeIgniter
2. Basis data yang digunakan adalah MySQL.
3. Antarmuka sistem dibuat menggunakan Bootstrap, CSS, dan JavaScript.

5.2.3 Implementasi Data

Implementasi basis data dapat dilihat pada Gambar 5.14. Tabel-Tabel tersebut didapat dari hasil perancangan data. Terdapat Tabel admin, Tabel pengguna, Tabel artikel, Tabel komentar, Tabel suka_artikel, Tabel produk, Tabel review.



Gambar 5.14 Implementasi Basis Data

5.2.3.1 Tabel Admin

Penjelasan pada Tabel 5.11.

Tabel 5.11 Implementasi Basis Data Tabel Admin

No.	Nama Atribut	Tipe	Panjang	Keterangan
1	id_admin	Int	11	Primary key dari tabel admin
2	username	Varchar	100	Username yang digunakan untuk login
3	password	Varchar	100	Password yang digunakan untuk login

5.2.3.2 Tabel Pengguna

Penjelasan pada Tabel 5.12.

Tabel 5.12 Implementasi Basis Data Tabel Pengguna

No.	Nama Atribut	Tipe	Panjang	Keterangan
1	id_pengguna	Int	11	Primary key dari tabel pengguna
2	nama	Varchar	100	Nama pengguna
3	alamat	Varchar	100	Alamat pengguna



Tabel 5.12 Implementasi Basis Data Tabel Pengguna (Lanjutan)

No.	Nama Atribut	Tipe	Panjang	Keterangan
4	foto	Varchar	100	Foto Profil pengguna
5	tipe	Varchar	255	Tipe kulit pengguna
6	usia	Varchar	255	Usia pengguna
7	concern	Varchar	255	Masalah kulit yang dialami pengguna
8	username	Varchar	100	Username yang digunakan untuk login
9	password	Varchar	50	Password yang digunakan untuk login

5.2.3.3 Tabel Artikel

Penjelasan pada Tabel 5.13.

Tabel 5.13 Implementasi Basis Data Tabel Artikel

No.	Nama Atribut	Tipe	Panjang	Keterangan
1	id_artikel	Int	11	Primary key dari tabel artikel
2	judul_artikel	Varchar	100	Judul artikel pada artikel
3	sumber	Varchar	255	Sumber artikel didapat
4	url	Varchar	255	URL artikel
5	nama_penulis	Varchar	100	Nama penulis artikel
6	tanggal_penulisan	Date		Tanggal penulisan artikel
7	isi_artikel	longtext		Isi artikel
8	gambar	Varchar	100	Gambar artikel yang berkaitan

5.2.3.4 Tabel Komentar

Penjelasan pada Tabel 5.14.

Tabel 5.14 Implementasi Basis Data Tabel Komentar

No.	Nama Atribut	Tipe	Panjang	Keterangan
1	id_komentar	Int	11	Primary key dari tabel komentar



Tabel 5.14 Implementasi Basis Data Tabel Komentar (lanjutan)

No.	Nama Atribut	Tipe	Panjang	Keterangan
2	id_artikel	Int	11	<i>Foreign key</i> dari tabel komentar
3	id_pengguna	Int	11	<i>Foreign key</i> dari tabel komentar
4	tgl_komentar	Timestamp	-	Berisi keterangan waktu saat pengguna membuat ulasan
5	isi_komentar	Varchar	255	Berisi ulasan artikel

5.2.3.5 Tabel Suka Artikel

Penjelasan pada Tabel 5.15.

Tabel 5.15 Implementasi Basis Data Tabel Suka Artikel

No.	Nama Atribut	Tipe	Panjang	Keterangan
1	id_suka	Int	11	<i>Primary key</i> dari tabel suka_artikel
2	id_artikel	Int	11	<i>Foreign key</i> dari tabel suka_artikel
3	id_pengguna	Int	11	<i>Foreign key</i> dari tabel suka_artikel

5.2.3.6 Tabel Produk

Penjelasan pada Tabel 5.16.

Tabel 5.16 Implementasi Basis Data Tabel Produk

No.	Nama Atribut	Tipe	Panjang	Keterangan
1	id_produk	Int	11	<i>Primary key</i> dari tabel ahp_dataalter
2	problem	Varchar	100	Berisi data masalah kulit
3	kategori	Varchar	255	Berisi data kategori produk
4	tipe	Varchar	255	Berisi data tipe kulit
5	brand	Varchar	100	Nama <i>brand</i> produk
6	nama_produk	Varchar	100	Nama produk



Tabel 5.16 Implementasi Basis Data Tabel Produk (lanjutan)

No.	Nama Atribut	Tipe	Panjang	Keterangan
7	deskripsi	longtext	-	Deskripsi produk
8	harga	Varchar	20	Harga produk
9	gambar	Varchar	100	Gambar produk yang berkaitan
10	bahan	Int	11	Berisi nilai untuk kriteria bahan
11	kualitas	Int	11	Berisi nilai untuk kriteria bahan
12	tempat	Varchar	255	Tempat produk bisa dibeli
13	total_bintang	Double	-	Total <i>rating</i> produk yang diberi oleh pengguna

5.2.3.7 Tabel Review

Penjelasan pada Tabel 5.17.

Tabel 5.17 Implementasi Basis Data Tabel Komentar

No.	Nama Atribut	Tipe	Panjang	Keterangan
1	id_review	Int	11	<i>Primary key</i> dari tabel <i>review</i>
2	id_produk	Int	11	<i>Foreign key</i> dari tabel <i>review</i>
3	id_pengguna	Int	11	<i>Foreign key</i> dari tabel <i>review</i>
4	tgl_review	Timestamp	-	Berisi keterangan waktu saat pengguna membuat ulasan
5	isi_review	Varchar	255	Berisi ulasan produk
6	bintang	Int	5	Berisi <i>rating</i> yang diberi pengguna

5.2.4 Implementasi Kode Program

Implementasi kode program merupakan hasil dari perancangan komponen yang telah dibuat sebelumnya dalam bentuk algoritme. Algoritme yang akan



ditulis implementasinya adalah membuat akun baru, menambah data produk, dan mengubah data diri.

5.2.4.1 Kode Program *Method* create() Klas C_Register

Tabel 5.18 adalah kode program *method* create() dari kelas C_Register.

Tabel 5.18 Kode Program *Method* create() Klas C_Register

No	Uraian
1	public function create() {
2	\$this->form_validation->set_rules('email',
3	'Email',
4	'trim required valid_email is_unique[pengguna.email]');
5	\$this->form_validation->set_rules('username',
6	'Username', 'trim required is_unique[pengguna.username]');
7	
8	if(\$this->form_validation->run() == false) {
9	\$this->load->view('register');
10	}
11	else {
12	\$username = \$this->input->post('username');
13	\$password = md5(\$this->input-
14	->post('password');
15	\$email = \$this->input->post('email');
16	\$name = \$this->input->post('nama');
17	\$alamat = \$this->input->post('alamat');
18	\$foto = \$this->input->post('foto');
19	\$data = array(
20	'username' => \$username,
21	'password' => \$password,
22	'email' => \$email,
23	'nama' => \$name,
24	'foto' => \$foto,
25	'alamat' => \$alamat
26);
27	
28	\$this->m_pengguna->insert(\$data);
29	\$this->session-
30	->set_flashdata('message_success', 'Register berhasil
31	Silahkan Login!');
32	redirect('C_login/login');
33	}
34	
35	}

5.2.4.2 Kode Program *Method* tambah_produk() Klas C_Admin

Tabel 5.19 adalah kode program *method* tambah_produk() dari kelas C_Admin.

Tabel 5.19 Kode Program *Method* tambah_produk() Klas C_Admin

No	Uraian
1	public function tambah_produk() {
2	\$this->form_validation->set_rules('brand', 'nama
3	brand', 'required trim');
4	\$this->form_validation->set_rules('nama_produk',

Tabel 5.19 Kode Program *Method* tambah_produk() Kelas C_Admin (Lanjutan)

No	Uraian
5	'nama_produk', 'required trim');
6	\$this->form_validation->set_rules('deskripsi',
7	'deskripsi', 'required trim');
8	\$this->form_validation->set_rules('harga',
9	'harga', 'required trim');
10	\$this->form_validation->set_rules('kualitas',
11	'kualitas', 'required trim');
12	\$this->form_validation->set_rules('bahan',
13	'bahan', 'required trim');
14	if(\$this->form_validation->run() == false) {
15	redirect('C_Admin/add_dataAlter');
16	} else {
17	\$config['upload_path'] =
18	'./assets/Halaman_Admin/img/Produk/';
19	\$config['allowed_types'] = 'jpg png';
20	\$this->upload->initialize(\$config);
21	\$this->upload->do_upload('gambar');
22	\$gambar_produk = \$this->upload->file_name;
23	\$checkboxes = \$this->input->post('type');
24	
25	\$data = array (
26	'problem' => \$this->input->post('problem'),
27	'kategori' => \$this->input-
28	->post('kategori'),
29	'tipe' => implode(", ", \$checkboxes),
30	'brand' => \$this->input->post('brand'),
31	'nama_produk' => \$this->input-
32	->post('nama_produk'),
33	'deskripsi' => \$this->input-
34	->post('deskripsi'),
35	'harga' => \$this->input->post('harga'),
36	'kualitas' => \$this->input-
37	->post('kualitas'),
38	'bahan' => \$this->input->post('bahan'),
39	'gambar' => \$gambar_produk
40);
41	\$this->m_admin->create_product(\$data);
42	redirect('C_Admin/data_produk');

5.2.4.3 Kode Program *Method* simpanProfile() Kelas C_Pengguna

Tabel 5.20 adalah kode program *method* simpanProfile() dari kelas C_Pengguna.

Tabel 5.20 Kode Program *Method* simpanProfile() Kelas C_Pengguna

No	Uraian
1	public function simpanProfile(\$id_pengguna)
2	{
3	\$this->load->view('default/header');
4	\$nama = \$this->input->post('nama');
5	\$salamat = \$this->input->post('alamat');
6	\$email = \$this->input->post('email');
7	\$tipe = \$this->input->post('tipe');
8	\$usia = \$this->input->post('usia');

Tabel 5.20 Kode Program *Method* *simpanProfile()* Kelas *C_Pengguna* (Lanjutan)

No	Uraian
9	<code>\$checkboxes = \$this->input->post('concern');</code>
10	<code>\$concern = implode(" ", \$checkboxes);</code>
11	
12	<code>//upload foto</code>
13	<code>\$this->load->library('upload');</code>
14	<code>\$path</code>
15	<code>='./assets/Halaman_Pengguna/images/Profile';</code>
16	<code>\$config['upload_path'] = \$path;</code>
17	<code>\$config['allowed_types'] = 'jpg jpeg png';</code>
18	
19	<code>\$this->upload->initialize(\$config);</code>
20	<code>if(\$FILES['foto']['name']){</code>
21	<code> \$this->upload->do_upload('foto');</code>
22	<code> \$result1 = \$this->upload->data();</code>
23	<code> \$result = array('gambar1'=>\$result1);</code>
24	
25	<code> \$namaImage = \$result['gambar1']['file_name'];</code>
26	<code> \$foto</code>
27	<code> 'assets/Halaman_Pengguna/images/Profile/'.\$namaImage;</code>
28	<code>}else {</code>
29	<code> \$foto = \$this->input->post('fotolama');</code>
30	<code>}</code>
31	<code> \$data = array(</code>
32	<code> 'nama' => \$nama,</code>
33	<code> 'alamat' => \$alamat,</code>
34	<code> 'email' => \$email,</code>
35	<code> 'foto' => \$foto,</code>
36	<code> 'tipe' => \$tipe,</code>
37	<code> 'usia' => \$usia,</code>
38	<code> 'concern' => \$concern</code>
39	<code>);</code>
40	<code> \$this->m_pengguna->updateProfile(\$id_pengguna,</code>
41	<code> \$data);</code>
42	<code> redirect('C_Pengguna/profile');</code>
43	<code>}</code>

5.2.5 Implementasi Antarmuka

Jumlah halaman antarmuka yang ditampilkan berjumlah 5 (lima) sesuai jumlah di tahap perancangan. Di antaranya adalah halaman *login*, halaman utama *member* dan halaman utama admin.

5.2.5.1 Antarmuka Halaman *Login*

Gambar 5.15 adalah halaman *login*. Terdapat *field username* dan *password* serta tombol *login*.



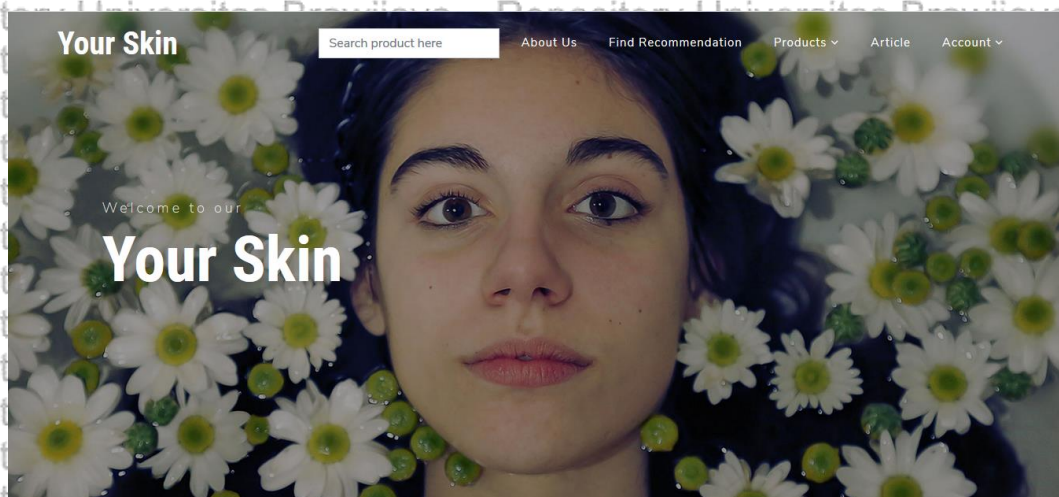
Login Form

© 2019 | 155150201111169

Gambar 5.15 Antarmuka Halaman *login*

5.2.5.2 Antarmuka Halaman Utama Sistem

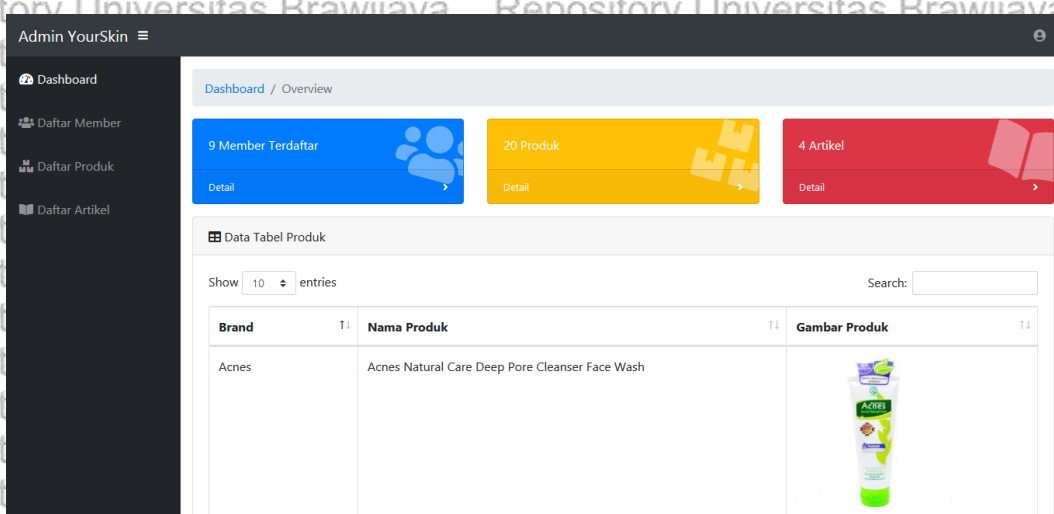
Gambar 5.16 adalah halaman utama sistem. Terdapat nama aplikasi, kolom *search*, menu *about us*, menu *find recommendation*, menu *products*, menu *article*, menu *account* pada bagian *header*.



Gambar 5.16 Antarmuka Halaman Utama Sistem

5.2.5.3 Antarmuka Halaman Utama Admin

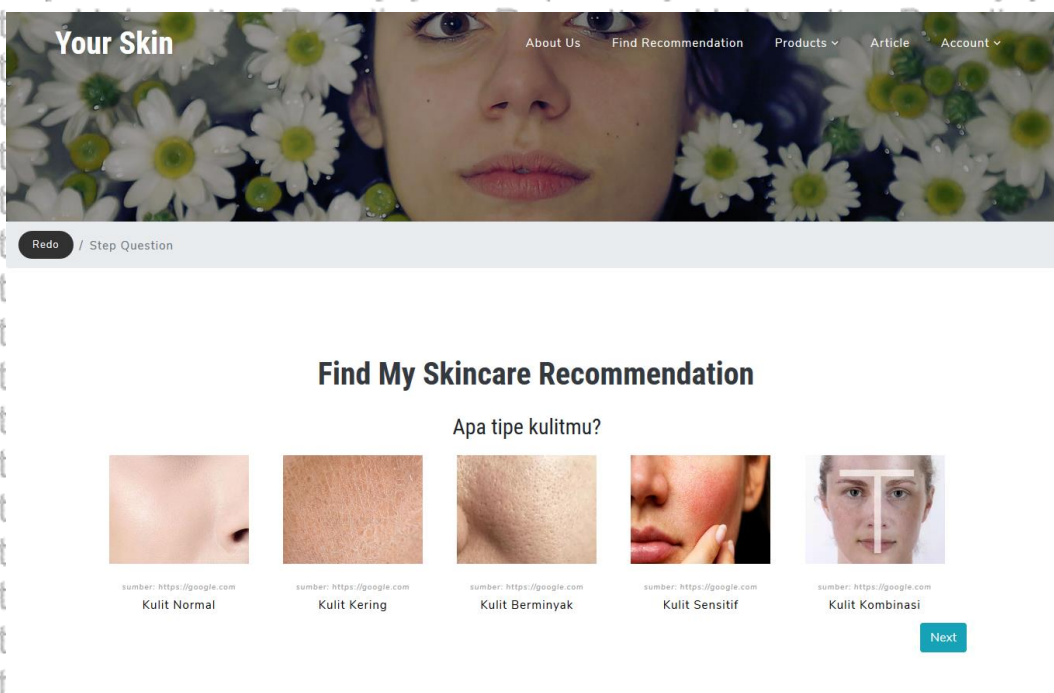
Gambar 5.17 adalah halaman utama admin. Terdapat menu daftar pengguna, daftar produk, dan daftar artikel pada bagian *sidebar*.



Gambar 5.17 Antarmuka Halaman Utama Admin

5.2.5.4 Antarmuka Halaman Find Recommendation

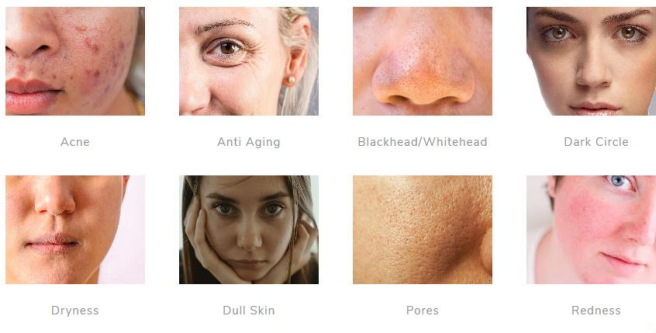
Gambar 5.18 adalah halaman *find recommendation*. Terdapat *multi-page* yang terdiri dari halaman tipe kulit, permasalahan kulit, kategori produk serta tombol *next*, *previous*, dan *submit*.





Find My Skincare Recommendation

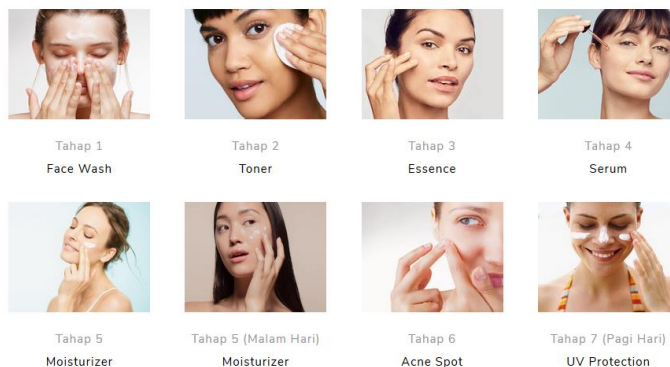
Apa permasalahan kulitmu?



Previous Next

Produk apa yang kamu cari?

(Perhatikan tahapan pemakaian. Pemakaian dimulai dari produk yang bertekstur ringan dan dilanjutkan dengan produk bertekstur berat.)



Previous Submit

Gambar 5.18 Antarmuka Halaman Find Recommendation

5.2.5.5 Antarmuka Halaman Tambah Produk

Gambar 5.19 adalah halaman tambah produk. Terdapat *field* permasalahan kulit, tipe kulit, kategori produk, nama *brand*, nama produk, deskripsi, harga, bahan, kualitas, bahan, tempat pembelian, dan gambar serta tombol *save*.



Admin YourSkin

Dashboard / Data Produk Alternative / Add

[← Back](#)

Permasalahan Kulit

Pilih

Tipe Kulit

Normal Oily Dry Sensitive Combination

Kategori Produk

Pilih

Nama Brand

Nama Brand

Nama Produk

Nama Produk

Deskripsi

Deskripsi Produk...

Harga

Harga

Kualitas

Kualitas

Bahan

Bahan

Tempat Pembelian

Tempat Pembelian

Gambar

Telusuri... Tidak ada berkas dipilih.

Save

Copyright © Your Website 2019

Gambar 5.19 Antarmuka Halaman Tambah Produk

BAB 6 PENGUJIAN

Pengujian dalam penelitian ini dilakukan dalam penelitian ini adalah pengujian unit, pengujian integrasi, pengujian validasi, pengujian *compatibility*, dan pengujian akurasi.

6.1 Pengujian Unit (Fungsional)

Pengujian ini dilakukan dengan menggunakan metode *basis path*. Klas *test* akan digunakan dalam pengujian ini. Klas ini tempat menjalankan *method* yang akan diuji. *Method* yang diuji adalah *method create()*, *method tambah_produk()*, dan *method simpanProfile()*.

6.1.1 Method create() Klas C_Register

1. Algoritme

Tabel 6.1 merupakan algoritme dari *method create()*.

Tabel 6.1 Algoritme Method create()

Node	Algoritme
1	Mulai method create()
2	Inisialisasi rules untuk validasi email
3	Inisialisasi rules untuk validasi username
4	IF form_validation adalah false Mengalihkan ke halaman register
5	Else
6	Inisialisasi variabel data dalam bentuk array Memanggil fungsi insert di model m_pengguna Menampilkan message_success Mengalihkan ke halaman login
7	ENDIF
8	Selesai

2. Basis Path Testing

2.1. Flow Graph

Gambar 6.1 adalah *flow graph* dari *method create()*.

2.2. Cyclomatic Complexity (V(G))

$$V(G) = \text{jumlah region} = 2$$

$$V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (8-8) + 2 = 2$$

$$V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$$

2.3. Independent path

Jalur 1 : 1-2-3-4-7-8

Jalur 2 : 1-2-5-6-7-8

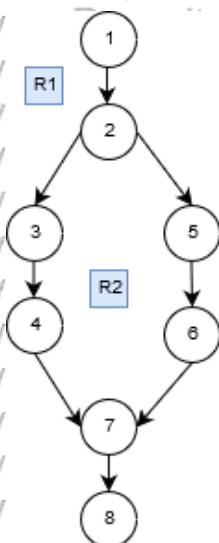


2.4. Kasus Uji

Tabel 6.2 adalah kasus uji *method create()* dari kelas *C_Register*.

Tabel 6.2 Kasus Uji *Method create()*

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas test memanggil <i>method create()</i> dari kelas <i>C_Register</i> dengan nilai <i>input</i> dari <i>submit = false</i>	<i>echo "Halaman register"</i>	<i>echo "Halaman register"</i>	<i>pass</i>
2	Klas test memanggil <i>method create()</i> dari kelas <i>C_Register</i> dengan nilai <i>input</i> dari <i>submit = true</i>	<i>echo "Halaman login"</i>	<i>echo "Halaman login"</i>	<i>pass</i>



Gambar 6.1 Flow Graph *Method create()*

6.1.2 *Method tambah_produk()* Kelas *C_Admin*

1. Algoritme

Tabel 6.3 adalah algoritme dari *method tambah_produk()*.

Tabel 6.3 Algoritme *Method tambah_produk()*

Node	Algoritme
1	Mulai <i>method tambah_produk</i>
2	Inisialisasi <i>rules</i> untuk validasi <i>brand</i> Inisialisasi <i>rules</i> untuk validasi <i>nama produk</i>

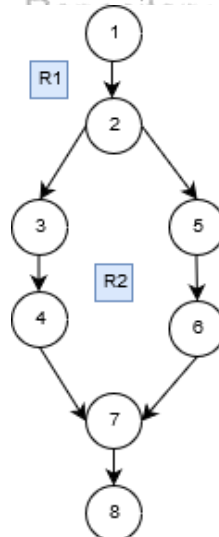


	Inisialisasi rules untuk validasi deskripsi
	Inisialisasi rules untuk validasi harga
	Inisialisasi rules untuk validasi kualitas
	Inisialisasi rules untuk validasi bahan
3	IF form validation adalah false
4	Menampilkan halaman add_dataAlter
5	Else
6	Membuat konfigurasi untuk upload gambar
	Inisialisasi variabel data dalam bentuk array
	Memanggil fungsi create_product di model m_admin
	Mengalihkan ke halaman data_produk
7	ENDIF
8	Selesai

2. Basis Path Testing

2.1. Flow Graph

Gambar 6.2 adalah *flow graph* dari *method* tambah_produk()



Gambar 6.2 *Flow Graph Method* tambah_produk()

2.2. Cyclomatic Complexity (V(G))

$$V(G) = \text{jumlah region} = 2$$

$$V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (8 - 8) + 2 = 2$$

$$V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$$

2.3. Independent path

Jalur 1: 1-2-3-4-7-8

Jalur 2: 1-2-5-6-7-8

2.4. Kasus Uji

Tabel 6.4 adalah kasus uji *method* tambah_produk() dari kelas C_Admin.

Tabel 6.4 Kasus Uji *Method* tambah_produk()

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas <i>test method</i> memanggil tambah_produk() dari klas C_Admin dengan nilai <i>input</i> dari <i>submit</i> = <i>false</i>	echo "Halaman add_dataAlter"	echo "Halaman add_dataAlter"	pass
2	Klas <i>test method</i> memanggil tambah_produk() dari klas C_Admin dengan nilai <i>input</i> dari <i>submit</i> = <i>true</i>	echo "Halaman data produk"	echo "Halaman data produk"	pass

6.1.3 *Method* simpanProfile() Klas C_Pengguna

1. Algoritme

Tabel 6.5 merupakan algoritme dari *method* simpanProfile().

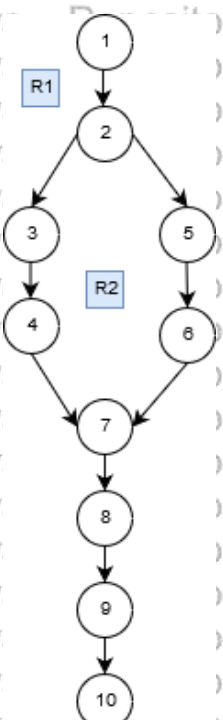
Tabel 6.5 Algoritme *Method* simpanProfile()

simpanProfile()	
Node	Algoritme
1	Mulai <i>method</i> simpanProfile sesuai dengan id pengguna
2	Mengambil data dari form edit
	Memanggil library upload
	Membuat konfigurasi untuk upload gambar
	Menginisialisasi variabel config
3	IF upload gambar baru
4	Inisialisasi gambar dengan variabel baru
5	Else
6	Inisialisasi gambar dengan variabel lama
7	ENDIF
8	Inisialisasi variabel data dalam bentuk array
9	Memanggil fungsi updateProfile di model m_pengguna berdasarkan variabel id_pengguna dan data
10	Selesai

2. Basis Path Testing

2.1. Flow Graph

Gambar 6.3 adalah *flow graph* dari *method* simpanProfile().



Gambar 6.3 Flow Graph Method simpanProfile()

2.2. Cyclomatic Complexcity (V(G))

- $V(G) = \text{jumlah region} = 2$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (10-10) + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

2.3. Independent path

- Jalur 1 : 1-2-3-4-7-8-9-10
- Jalur 2 : 1-2-5-6-7-8-9-10

2.4. Desain Kasus Uji

Tabel 6.6 merupakan kasus uji *method simpanProfile()* dari kelas *C_Pengguna*.

Tabel 6.6 Kasus Uji *Method simpanProfile()*

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas <i>test</i> memanggil <i>method simpanProfile()</i> berdasarkan	<i>echo "Halaman profile"</i>	<i>echo "Halaman profile"</i>	<i>pass</i>

Tabel 6.6 Kasus Uji *Method* `simpanProfile()` (lanjutan)

No. Jalur	Prosedur Uji	Expected Result	Result	Status
	id_pengguna dari kelas C_Pengguna dengan nilai input dari foto = foto			
2	Kelas <code>test</code> memanggil <code>method</code> <code>simpanProfile()</code> berdasarkan id_pengguna dari kelas C_Pengguna dengan nilai input dari foto = fotolama	<code>echo</code> "Halaman <code>profile</code> "	<code>echo</code> "Halaman <code>profile</code> "	<code>pass</code>

6.2 Pengujian Integrasi (Fungsional)

Pengujian ini dilakukan dengan menggunakan metode *basis path*. Pengujian ini dilakukan untuk mengetahui interaksi yang terjadi antar kelas. Tahap ini akan dilakukan pada `method` `tambah_produk()` kelas C_Admin. Pengujian ini menggunakan kelas test untuk menjalankan `method` yang akan diuji.

6.2.1 *Method* `tambah_produk()` Kelas C_Admin

6.2.1.1 *Method* `tambah_produk()`

1. Algoritme

Tabel 6.7 merupakan algoritme `method` `tambah_produk()`.

Tabel 6.7 Algoritme *Method* `tambah_produk()`

Node	Algoritme
1	Mulai <code>method</code> <code>tambah_produk()</code>
2	Inisialisasi rules untuk validasi brand
	Inisialisasi rules untuk validasi nama produk
	Inisialisasi rules untuk validasi deskripsi
	Inisialisasi rules untuk validasi harga
	Inisialisasi rules untuk validasi kualitas
	Inisialisasi rules untuk validasi bahan
3	IF form_validation adalah false
4	Menampilkan halaman <code>add_dataAlter</code>
5	Else
6	Membuat konfigurasi untuk upload gambar
	Inisialisasi variable data dalam bentuk array
	Memanggil <code>method</code> <code>create_product</code> di model <code>m_admin</code>
	Menghentikan ke halaman <code>data_produk</code>
7	ENDIF

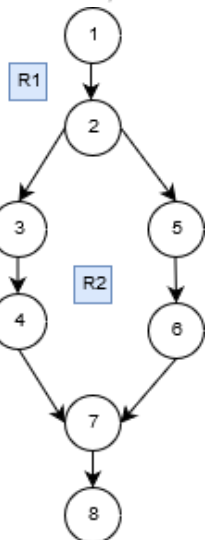


8	Selesai
---	---------

2. Basis Path Testing

2.1. Flow Graph

Gambar 6.4 adalah flow graph dari method tambah_produk()



Gambar 6.4 Flow Graph Method tambah_produk

2.2. Cyclomatic Complexity (V(G))

- $V(G) = \text{jumlah region} = 2$
- $V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (8-8) + 2 = 2$
- $V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$

2.3. Independent path

- Jalur 1 : 1-2-3-4-7-8
- Jalur 2 : 1-2-5-6-7-8

6.2.1.2 Method create_product()

1. Algoritme

Tabel 6.8 merupakan algoritme method create_product()

Tabel 6.8 Algoritme Method create_product()

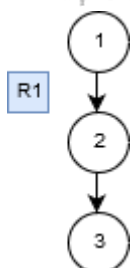
create_product ()	
Node	Algoritme
1	Mulai method create_product()
2	Inisialisasi variabel insert dengan parameter nama Tabel di database dan data
3	Selesai

2. Basis Path Testing



2.1. Flow Graph

Gambar 6.5 adalah *flow graph* dari *method create_product()*.



Gambar 6.5 Flow Graph Method create_product()

2.2. Cyclomatic Complexity (V(G))

$$V(G) = \text{jumlah region} = 1$$

$$V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (2-3) + 2 = 1$$

$$V(G) = \text{jumlah predicate node} + 1 = 0 + 1 = 1$$

2.3. Independent path

Jalur 1: 1-2-3

6.2.1.3 Gabungan Method tambah_produk() dan create_product()

1. Algoritme

Tabel 6.9 merupakan algoritme gabungan *method tambah_produk()* dan *create_product()*.

Tabel 6.9 Algoritme Gabungan Method tambah_produk dan create_product()

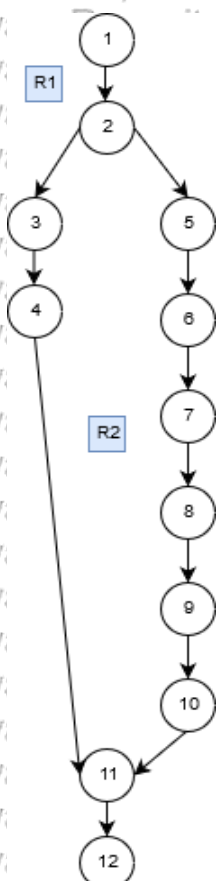
Node	Algoritme
1	Mulai method tambah_produk()
2	Inisialisasi rules untuk validasi brand Inisialisasi rules untuk validasi nama produk Inisialisasi rules untuk validasi deskripsi Inisialisasi rules untuk validasi harga Inisialisasi rules untuk validasi kualitas Inisialisasi rules untuk validasi bahan
3	IF form_validation adalah false
4	Menampilkan halaman add_dataAlter
5	Else
6	Membuat konfigurasi untuk upload gambar Inisialisasi variabel data dalam bentuk array
7	Memanggil method create_product di model m_admin
8	Inisialisasi variabel insert dengan parameter nama Tabel di database dan data
9	Tutup method create_product()
10	Mengalihkan ke halaman data_produk
11	ENDIF
12	Selesai



2. Basis Path Testing

2.1. Flow Graph

Gambar 6.6 merupakan *flow graph* dari gabungan *method* *tambah_produk()* dan *create_product()*.



Gambar 6.6 Flow Graph Gabungan Method *tambah_produk()* dan *create_product()*

2.2. Cyclomatic Complexity (V(G))

$$V(G) = \text{jumlah region} = 2$$

$$V(G) = (\text{jumlah edge} - \text{jumlah node}) + 2 = (12 - 12) + 2 = 2$$

$$V(G) = \text{jumlah predicate node} + 1 = 1 + 1 = 2$$

2.3. Independent path

Jalur 1: 1-2-3-4-11-12

Jalur 2: 1-2-5-6-7-8-9-10-11-12

2.4. Kasus Uji

Tabel 6.10 merupakan kasus uji *method* *tambah_produk()* dari kelas *C_Admin*.

Tabel 6.10 Kasus Uji *Method* tambah_produk()

No. Jalur	Prosedur Uji	Expected Result	Result	Status
1	Klas <i>test</i> memanggil <i>method</i> tambah_produk() dari kelas C_Admin dengan nilai <i>input</i> dari <i>submit</i> = <i>false</i>	<i>echo</i> "Halaman add_dataAlter"	<i>echo</i> "Halaman add_dataAlter"	<i>pass</i>
2	Klas <i>test</i> memanggil <i>method</i> tambah_produk() dari kelas C_Admin dengan nilai <i>input</i> dari <i>submit</i> = <i>true</i>	<i>echo</i> "Halaman data produk"	<i>echo</i> "Halaman data produk"	<i>pass</i>

6.3 Pengujian Validasi (Fungsional)

Pengujian validasi dilakukan untuk mengetahui apakah sistem yang telah dibangun sesuai dengan kebutuhan atau belum. Pengujian validasi dilakukan dengan memeriksa setiap kebutuhan fungsional yang ada.

6.3.1 Validasi Login

1. Validasi Login

Tabel 6.11 Validasi *login* yang berhasil.

Tabel 6.11 Validasi Login

Nama Kasus Uji	Login
Prosedur	<ol style="list-style-type: none"> Mengakses halaman <i>login</i> <i>Form</i> diisi dengan memasukkan: <ul style="list-style-type: none"> <i>Username</i>: red <i>Password</i>: red Menekan tombol "LOGIN"
Hasil Akhir Yang Diharapkan	Halaman utama sistem akan tampil
Hasil	Halaman utama sistem tampil
Status	Valid



2. Validasi *login alternative* satu

Tabel 6.12 adalah validasi *login* yang gagal karena ada *field* kosong

Tabel 6.12 Validasi Login Alternative Satu

Nama Kasus Uji	<i>Login alternatifve</i> satu
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman <i>login</i> 2. Tombol "LOGIN" ditekan
Hasil Akhir Yang Diharapkan	Pesan "Isi isian ini" akan tampil
Hasil	Pesan "Isi isian ini" tampil
Status	Valid

3. Validasi *login alternative* dua

Tabel 6.13 adalah validasi untuk *login* yang gagal karena salah memasukkan *username* atau *password*.

Tabel 6.13 Validasi Login Alternative Dua

Nama Kasus Uji	<i>Login alternatifve</i> dua
Prosedur	<ol style="list-style-type: none"> 1. Halaman <i>login</i> diakses 2. <i>Form</i> diisi dengan <i>username</i> dan <i>password</i> salah 3. Tombol "LOGIN" ditekan
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan "Username Atau Password Salah"
Hasil	Sistem menampilkan pesan "Username Atau Password Salah"
Status	Valid

6.3.2 Validasi Registrasi Akun

1. Validasi registrasi akun

Tabel 6.14 adalah validasi untuk registrasi akun yang berhasil.

Tabel 6.14 Validasi Registrasi Akun

Nama Kasus Uji	Registrasi akun
Prosedur	<ol style="list-style-type: none"> 1. Halaman <i>register</i> diakses 2. <i>Form register</i> diisi dengan memasukkan: <ul style="list-style-type: none"> • Name: Jojor Yeanesy Sinaga



Tabel 6.14 Validasi Registrasi Akun (Lanjutan)

Nama Kasus Uji	Registrasi akun <ul style="list-style-type: none"> • Address: Jln. Watugong 15a • Email: jojoryeanesysinaga@gmail.com • Username: red • Password: red
Hasil Akhir Yang Diharapkan	3. Tombol "REGISTER" ditekan Akan menampilkan pesan "Register berhasil. Silahkan login!" di halaman login
Hasil	Menampilkan pesan "Register berhasil. Silahkan login!" di halaman login
Status	Valid

2. Validasi registrasi akun *alternative* satu

Tabel 6.11 adalah validasi untuk registrasi akun gagal karena ada *field* yang tidak diisi.

Tabel 6.15 Validasi Registrasi Akun *Alternative* Satu

Nama Kasus Uji	Registrasi akun <i>alternative</i> satu
Prosedur	1. Halaman <i>register</i> diakses 2. Menekan tombol "REGISTER" tanpa mengisi <i>field</i>
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan "Isi isian ini" pada <i>field</i>
Hasil	Sistem menampilkan pesan "Isi isian ini" pada <i>field</i>
Status	Valid

3. Validasi registrasi akun *alternative* dua

Tabel 6.16 adalah validasi untuk registrasi akun gagal karena terdapat *username* atau *email* yang sama.

Tabel 6.16 Validasi Registrasi Akun *Alternative* Dua

Nama Kasus Uji	Registrasi akun <i>alternative</i> satu
Prosedur	1. Halaman <i>register</i> diakses 2. Mengisi <i>form register</i> dengan memasukkan:

Tabel 6.16 Validasi Registrasi Akun *Alternative* Dua (Lanjutan)

Nama Kasus Uji	Registrasi akun <i>alternative</i> satu <ul style="list-style-type: none"> • <i>Name</i>: Jojor Sinaga • <i>Address</i>: Jln. Watugong 15a • <i>Email</i>: jojoryeanesysinaga@gmail.com • <i>Username</i>: red • <i>Password</i>: red <p>3. Tombol “REGISTER” ditekan</p>
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan “ <i>Username dan Email harus unique</i> ”
Hasil	Sistem menampilkan pesan “ <i>Username dan Email harus unique</i> ”
Status	Valid

4. Validasi registrasi akun *alternative* tiga

Tabel 6.17 adalah validasi untuk registrasi akun gagal karena terdapat *username* atau *password* yang kurang dari jumlah minimal karakter.

Tabel 6.17 Validasi Registrasi Akun *Alternative* Tiga

Nama Kasus Uji	Registrasi akun <i>alternative</i> tiga
Prosedur	<ol style="list-style-type: none"> 1. Halaman <i>register</i> diakses 2. Mengisi <i>form register</i> dengan memasukkan: <ul style="list-style-type: none"> • <i>Name</i>: Jojor Sinaga • <i>Address</i>: Jln. Watugong 15a • <i>Email</i>: jojoryeanesysinaga@gmail.com • <i>Username</i>: re • <i>Password</i>: re 3. Tombol “REGISTER” ditekan
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan “Mohon gunakan panjang karakter yang telah ditentukan.”
Hasil	Sistem menampilkan pesan “Mohon gunakan panjang karakter yang telah ditentukan.”
Status	Valid



6.3.3 Validasi Logout

1. Validasi logout

Tabel 6.18 adalah pengujian validasi untuk *logout* yang berhasil.

Tabel 6.18 Validasi Logout

Nama Kasus Uji	<i>logout</i>
Prosedur	<ol style="list-style-type: none"> Memilih menu <i>logout</i> pada menu <i>dropdown account</i> Pilihan "Yes" ditekan
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman utama sistem dengan pengguna sebagai <i>guest</i>
Hasil	Sistem menampilkan halaman utama sistem dengan pengguna sebagai <i>guest</i>
Status	Valid

6.3.4 Pengujian Validasi Melihat Data Diri

1. Validasi melihat data diri

Tabel 6.19 adalah validasi untuk melihat data diri yang berhasil.

Tabel 6.19 Validasi Melihat Data Diri

Nama Kasus Uji	Melihat data diri
Prosedur	<ol style="list-style-type: none"> Memilih menu <i>profile</i> pada menu <i>dropdown account</i>
Hasil Akhir Yang Diharapkan	Halaman <i>profile</i> akan tampil
Hasil	Halaman <i>profile</i> tampil
Status	Valid

6.3.5 Validasi Mengubah Data Diri

1. Validasi mengubah data diri

Tabel 6.20 adalah validasi untuk mengubah data diri yang berhasil.

Tabel 6.20 Validasi Mengubah Data Diri

Nama Kasus Uji	Mengubah data diri
Prosedur	<ol style="list-style-type: none"> Tombol "Setting Profile" ditekan halaman <i>profile</i> Mengubah data diri pada halaman <i>edit profile</i> dengan isian berikut:



Tabel 6.20 Validasi Mengubah Data Diri (Lanjutan)

Nama Kasus Uji	Mengubah data diri <ul style="list-style-type: none"> • Nama: Jojor Sinaga • Alamat: Jln. Kerto Asri <p>3. Tombol “Simpan” ditekan</p>
Hasil Akhir Yang Diharapkan	Sistem akan melakukan perubahan dan menyimpan data yang telah diubah
Hasil	Sistem melakukan perubahan dan menyimpan data yang telah diubah
Status	Valid

6.3.6 Validasi Melihat Halaman Rekomendasi

1. Validasi melihat halaman rekomendasi

Tabel 6.21 adalah validasi untuk melihat halaman rekomendasi yang berhasil.

Tabel 6.21 Validasi Melihat Halaman Rekomendasi

Nama Kasus Uji	Melihat halaman rekomendasi
Prosedur	<ul style="list-style-type: none"> • Menu <i>find recommendation</i> diakses
Hasil Akhir Yang Diharapkan	Halaman rekomendasi akan tampil
Hasil	Halaman rekomendasi tampil
Status	Valid

6.3.7 Validasi Mengisi Data Rekomendasi

1. Validasi mengisi data rekomendasi

Tabel 6.22 adalah validasi untuk mengisi data rekomendasi yang berhasil.

Tabel 6.22 Validasi Mengisi Data Rekomendasi

Nama Kasus Uji	Mengisi data rekomendasi
Prosedur	<ol style="list-style-type: none"> 1. Menu <i>find recommendation</i> diakses 2. Memilih tipe kulit: normal 3. Memilih permasalahan kulit: <i>dark circle</i> 4. Memilih kategori produk: <i>eye cream</i> 5. Tombol “<i>Submit</i>” ditekan



Tabel 6.22 Validasi Mengisi Data Rekomendasi (Lanjutan)

Nama Kasus Uji	Mengisi data rekomendasi
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman kriteria
Hasil	Sistem menampilkan halaman kriteria
Status	Valid

2. Validasi mengisi data rekomendasi *alternative* dua

Tabel 6.23 adalah validasi untuk mengisi data rekomendasi yang menampilkan *error* jika pada *form* tipe kulit menekan tombol *next* tanpa memilih pilihan.

Tabel 6.23 Validasi Mengisi Data Rekomendasi *Alternative* Dua

Nama Kasus Uji	Mengisi data rekomendasi <i>alternative</i> dua
Prosedur	1. Halaman <i>find recommendation</i> diakses 2. Tombol " <i>next</i> " pada halaman tipe kulit
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan peringatan untuk memilih pilihan dari pilihan yang tersedia
Hasil	Sistem menampilkan pesan peringatan untuk memilih pilihan dari pilihan yang tersedia
Status	Valid

3. Validasi mengisi data rekomendasi *alternative* tiga

Tabel 6.24 adalah validasi untuk mengisi data rekomendasi yang menampilkan *error* jika pada *form concern* menekan tombol *next* tanpa memilih pilihan.

Tabel 6.24 Validasi Mengisi Data Rekomendasi *Alternative* Tiga

Nama Kasus Uji	Mengisi data rekomendasi <i>alternative</i> tiga
Prosedur	1. Mengakses halaman <i>find recommendation</i> 2. Memilih tipe kulit: normal 3. Tombol " <i>next</i> " ditekan 4. Halaman permasalahan kulit tampil 5. Tombol " <i>next</i> " ditekan pada halaman permasalahan kulit

Tabel 6.24 Validasi Mengisi Data Rekomendasi *Alternative* Tiga (Lanjutan)

Nama Kasus Uji	Mengisi data rekomendasi <i>alternative</i> tiga
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan peringatan untuk memilih pilihan dari pilihan yang tersedia
Hasil	Sistem menampilkan pesan peringatan untuk memilih pilihan dari pilihan yang tersedia
Status	Valid

4. Validasi mengisi data rekomendasi *alternative* empat

Tabel 6.25 adalah validasi untuk mengisi data rekomendasi yang menampilkan *error* jika pada *form* kategori produk menekan tombol *next* tanpa memilih pilihan.

Tabel 6.25 Validasi Mengisi Data Rekomendasi *Alternative* Empat

Nama Kasus Uji	Mengisi data rekomendasi <i>alternative</i> empat
Prosedur	<ol style="list-style-type: none"> 1. Halaman <i>find recommendation</i> 2. Memilih tipe kulit: normal 3. Tombol "<i>next</i>" ditekan 4. Halaman permasalahan kulit tampil 5. Memilih permasalahan kulit: <i>dark circle</i> 6. Tombol "<i>next</i>" ditekan 7. Sistem menampilkan halaman kategori produk 8. Tombol "<i>next</i>" pada halaman kategori produk ditekan
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan peringatan untuk memilih pilihan dari pilihan yang tersedia
Hasil	Sistem menampilkan pesan peringatan untuk memilih pilihan dari pilihan yang tersedia
Status	Valid

6.3.8 Validasi Mengisi Nilai Perbandingan

1. Validasi mengisi nilai perbandingan

Tabel 6.26 adalah validasi untuk mengisi nilai perbandingan yang berhasil.



Tabel 6.26 Validasi Mengisi Nilai Perbandingan

Nama Kasus Uji	Mengisi nilai perbandingan
Prosedur	<ol style="list-style-type: none"> 1. Mengakses menu <i>find recommendation</i> 2. Memilih tipe kulit: normal 3. Memilih permasalahan kulit: <i>dark circle</i> 4. Memilih kategori produk: <i>eye cream</i> 5. Menekan tombol "<i>Submit</i>" 6. Halaman kriteria tampil 7. Memilih nilai: <ul style="list-style-type: none"> • Harga terhadap bahan: 5 • Bahan terhadap kualitas: 7 • Kualitas terhadap harga: 9
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman kriteria dengan nilai yang dipilih
Hasil	Sistem menampilkan halaman kriteria dengan nilai yang dipilih
Status	Valid

6.3.9 Validasi Melihat Daftar Rekomendasi

1. Validasi melihat daftar rekomendasi

Tabel 6.27 adalah validasi untuk melihat daftar rekomendasi yang berhasil.

Tabel 6.27 Validasi Melihat Daftar Rekomendasi

Nama Kasus Uji	Melihat daftar rekomendasi
Prosedur	<ol style="list-style-type: none"> 1. Menu <i>find recommendation</i> diakses 2. Memilih tipe kulit: normal 3. Memilih permasalahan kulit: <i>dark circle</i> 4. Memilih kategori produk: <i>eye cream</i> 5. Tombol "<i>Submit</i>" ditekan 6. Halaman kriteria tampil 7. Memilih nilai: <ul style="list-style-type: none"> • Harga terhadap bahan: 5



Tabel 6.27 Validasi Melihat Daftar Rekomendasi (Lanjutan)

Nama Kasus Uji	Melihat daftar rekomendasi <ul style="list-style-type: none"> • Bahan terhadap kualitas: 7 • Kualitas terhadap harga: 9 8. Tombol “Submit” ditekan
Hasil Akhir Yang Diharapkan	Halaman daftar rekomendasi akan tampil
Hasil	Halaman daftar rekomendasi tampil
Status	Valid

2. Validasi melihat halaman rekomendasi *alternative* satu

Tabel 6.28 adalah validasi untuk melihat halaman rekomendasi yang gagal karena nilai tidak dimasukkan.

Tabel 6.28 Pengujian Validasi Melihat Halaman Rekomendasi *Alternative* Satu

Nama Kasus Uji	Melihat halaman rekomendasi <i>alternatif</i> satu
Prosedur	<ol style="list-style-type: none"> 1. Mengakses menu <i>find recommendation</i> 2. Memilih tipe kulit: normal 3. Memilih permasalahan kulit: <i>dark circle</i> 4. Memilih kategori: produk <i>eye cream</i> 5. Tombol “Submit” ditekan 6. Halaman kriteria tampil 7. Tombol “Submit” ditekan
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan peringatan “Anda belum memilih. Silahkan pilih terlebih dahulu!!”
Hasil	Sistem menampilkan pesan peringatan “Anda belum memilih. Silahkan pilih terlebih dahulu!!”
Status	Valid

6.3.10 Validasi Menulis Ulasan Produk

1. Validasi menulis ulasan produk

Tabel 6.29 adalah validasi untuk menulis ulasan produk yang berhasil.



Tabel 6.29 Validasi Menulis Ulasan Produk

Nama Kasus Uji	Menulis ulasan produk
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman detail produk 2. Membuat ulasan dan memberi <i>rating</i> di kolom <i>review</i> 3. Tombol “<i>Submit</i>” ditekan
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman detail produk beserta ulasan
Hasil	Sistem menampilkan halaman detail produk beserta ulasan
Status	Valid

2. Validasi menulis ulasan produk *alternative* satu

Tabel 6.30 adalah validasi untuk menulis ulasan produk yang gagal karena kolom *review* tidak diisi.

Tabel 6.30 Validasi Menulis Ulasan Produk *Alternative* Satu

Nama Kasus Uji	Menulis ulasan produk <i>alternative</i> satu
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman detail produk 2. Menekan tombol “<i>Submit</i>” di kolom <i>review</i> dengan keadaan tidak terisi.
Hasil Akhir Yang Diharapkan	Pesan “Isi isian ini” akan tampil
Hasil	Pesan “Isi isian ini” tampil
Status	Valid

6.3.11 Validasi Menghapus Ulasan Produk

1. Validasi menghapus ulasan produk

Tabel 6.31 adalah validasi untuk menghapus ulasan produk yang berhasil.

Tabel 6.31 Validasi Menghapus Ulasan Produk

Nama Kasus Uji	Menghapus ulasan produk
Prosedur	<ol style="list-style-type: none"> 1. Halaman detail produk diakses 2. Memilih ulasan yang ingin di hapus 3. Menekan tombol “<i>Delete this review?</i>”



Tabel 6.31 Validasi Menghapus Ulasan Produk (Lanjutan)

Nama Kasus Uji	Menghapus ulasan produk 4. Menekan pilihan "Oke"
Hasil Akhir Yang Diharapkan	Sistem akan menghapus ulasan
Hasil	Sistem menghapus ulasan
Status	Valid

6.3.12 Validasi Menulis Ulasan Artikel

1. Validasi menulis ulasan artikel

Tabel 6.32 adalah validasi untuk menulis ulasan artikel yang berhasil.

Tabel 6.32 Validasi Menulis Ulasan Artikel

Nama Kasus Uji	Menulis ulasan artikel
Prosedur	1. Mengakses halaman detail artikel 2. Membuat ulasan di kolom <i>comment</i> 3. Tombol "Submit" ditekan
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman detail artikel beserta ulasan
Hasil	Sistem menampilkan halaman detail artikel beserta ulasan
Status	Valid

2. Validasi menulis ulasan artikel *alternative* satu

Tabel 6.33 adalah validasi untuk menulis ulasan artikel yang gagal karena kolom *comment* tidak diisi.

Tabel 6.33 Validasi Menulis Ulasan Artikel *Alternative* Satu

Nama Kasus Uji	Menulis ulasan artikel <i>alternative</i> satu
Prosedur	1. Mengakses halaman detail artikel 2. Menekan tombol "Submit" dengan keadaan kosong pada kolom <i>comment</i>
Hasil Akhir Yang Diharapkan	Pesan "Isi isian ini" akan tampil
Hasil	Pesan "Isi isian ini" tampil
Status	Valid



6.3.13 Validasi Menghapus Ulasan Artikel

1. Validasi menghapus ulasan artikel

Tabel 6.34 adalah validasi untuk menghapus ulasan artikel yang berhasil.

Tabel 6.34 Validasi Menghapus Ulasan Artikel

Nama Kasus Uji	Menghapus ulasan artikel
Prosedur	<ol style="list-style-type: none"> 1. Halaman detail artikel diakses 2. Memilih ulasan yang ingin di hapus 3. Menekan tombol "<i>Delete this comment?</i>" pada ulasan yang dipilih 4. Menekan pilihan "Oke"
Hasil Akhir Yang Diharapkan	Sistem akan menghapus ulasan
Hasil	Sistem menghapus ulasan
Status	Valid

6.3.14 Validasi Menyukai Artikel

1. Validasi menyukai artikel

Tabel 6.35 adalah validasi untuk menyukai artikel yang berhasil.

Tabel 6.35 Validasi Menyukai Artikel

Nama Kasus Uji	Menyukai artikel
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman detail artikel 2. Tombol "<i>like</i>" berbentuk <i>love</i> ditekan pada bagian akhir artikel
Hasil Akhir Yang Diharapkan	Sistem akan menambah jumlah <i>like</i>
Hasil	Sistem menambah jumlah <i>like</i>
Status	Valid

6.3.15 Validasi Membatalkan Menyukai Artikel

1. Validasi membatalkan menyukai artikel

Tabel 6.36 adalah validasi untuk membatalkan menyukai artikel yang berhasil.



Tabel 6.36 Validasi Membatalkan Menyukai Artikel

Nama Kasus Uji	Membatalkan menyukai artikel
Prosedur	<ol style="list-style-type: none"> 1. Halaman detail artikel diakses 2. Tombol “like” berbentuk <i>love</i> ditekan pada bagian akhir artikel
Hasil Akhir Yang Diharapkan	Sistem akan mengurangi jumlah <i>like</i>
Hasil	Sistem mengurangi jumlah <i>like</i>
Status	Valid

6.3.16 Validasi Melihat Daftar Pengguna

1. Validasi melihat daftar pengguna

Tabel 6.37 adalah validasi untuk melihat daftar pengguna yang berhasil.

Tabel 6.37 Validasi Melihat Daftar Pengguna

Nama Kasus Uji	Melihat daftar pengguna
Prosedur	1. Menu “Daftar Pengguna” dipilih pada <i>sidebar menu</i>
Hasil Akhir Yang Diharapkan	Halaman daftar pengguna akan tampil
Hasil	Halaman daftar pengguna tampil
Status	Valid

6.3.17 Validasi Menambah Data Produk

1. Validasi menambah data produk

Tabel 6.38 adalah validasi untuk menambah data produk yang berhasil.

Tabel 6.38 Validasi Menambah Data Produk

Nama Kasus Uji	Menambah data produk
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman tambah data produk 2. <i>Form</i> diisi dengan data: <ul style="list-style-type: none"> • Problem: Acne • Kategori: Face Wash • Tipe Kulit: Normal • Nama Brand: Pond's



Tabel 6.34 Validasi Menambah Data Produk (Lanjutan)

Nama Kasus Uji	Menambah data produk <ul style="list-style-type: none"> • Nama Produk: Acne Solution Facial Cleanser Foam • Deskripsi: POND'S Acne Solution AntiAcne Facial Foam with Active Thymo-T Essence. Mengunci bakteri dengan lock & Clear Technology untuk membantu menghilangkan jerawat dari akarnya. Efektif membantu mengatasi 10 penyebab kulit berminyak dan jerawat. Terbukti klinis mengurangi jerawat. • Harga: RP 25000 • Kualitas: 1 • Bahan: 4 • Tempat pembelian: Guardian Indonesia • Gambar: ponds.jpg
	3. Menekan tombol "Save"
Hasil Akhir Yang Diharapkan	Data disimpan dan menampilkan halaman daftar produk
Hasil	Data disimpan dan menampilkan halaman daftar produk
Status	Valid

2. Validasi menambah data produk *alternative* satu

Tabel 6.39 adalah validasi untuk menambah data produk yang gagal karena ada *field* yang kosong.

Tabel 6.39 Validasi Menambah Data Produk *Alternative* Satu

Nama Kasus Uji	Menambah data produk <i>alternative</i> satu
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman tambah data produk 2. Menekan tombol "Save" tanpa mengisi <i>field</i>
Hasil Akhir Yang Diharapkan	Pesan "Isi isian ini" akan tampil pada <i>field</i>
Hasil	Pesan "Isi isian ini" tampil pada <i>field</i>

Tabel 6.39 Validasi Menambah Data Produk *Alternative* Satu (Lanjutan)

Nama Kasus Uji	Menambah data produk <i>alternative</i> satu
Status	Valid

6.3.18 Validasi Melihat Daftar Produk

1. Validasi melihat daftar produk

Tabel 6.40 adalah validasi untuk melihat daftar produk yang berhasil.

Tabel 6.40 Validasi Melihat Daftar Produk

Nama Kasus Uji	Melihat daftar produk
Prosedur	1. Menu “Data Produk” dipilih pada <i>sidebar menu</i>
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman daftar produk yang telah tersimpan di sistem
Hasil	Sistem menampilkan halaman daftar produk yang telah tersimpan di sistem
Status	Valid

6.3.19 Validasi Mengubah Data Produk

1. Validasi mengubah data produk

Tabel 6.41 adalah validasi untuk mengubah data produk yang berhasil.

Tabel 6.41 Validasi Mengubah Data Produk

Nama Kasus Uji	Mengubah data produk
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman daftar produk 2. Memilih data produk dengan tombol “<i>Edit</i>” ditekan 3. Ubah data dengan: <ul style="list-style-type: none"> • Problem: Acne • Kategori: Face Wash • Tipe Kulit: Kering • Nama Brand: Pond’s • Nama Produk: Acne Solution Facial Cleanser Foam • Deskripsi: POND’S Acne Solution AntiAcne Facial Foam with Active Thymo-T Essence. Mengunci bakteri dengan lock & Clear Technology untuk



Tabel 6.41 Validasi Mengubah Data Produk (Lanjutan)

Nama Kasus Uji	Mengubah data produk
	<p>membantu menghilangkan jerawat dari akarnya. Efektif membantu mengatasi 10 penyebab kulit berminyak dan jerawat. Terbukti klinis mengurangi jerawat.</p> <ul style="list-style-type: none"> • Harga: RP 25000 • Kualitas: 1 • Bahan: 4 • Tempat pembelian: Guardian Indonesia • Gambar: ponds.jpg <p>4. Menekan tombol "Save"</p>
Hasil Akhir Yang Diharapkan	Sistem akan melakukan perubahan dan menyimpan data
Hasil	Sistem melakukan perubahan dan menyimpan data serta
Status	Valid

6.3.20 Validasi Menghapus Data Produk

1. Validasi menghapus data produk

Tabel 6.42 adalah validasi untuk menghapus data produk yang berhasil.

Tabel 6.42 Validasi Menghapus Data Produk

Nama Kasus Uji	Menghapus data produk
Prosedur	<ol style="list-style-type: none"> 1. Mengakses halaman daftar produk 2. Memilih data produk dengan memilih tombol "Hapus" 3. Memilih "Yes"
Hasil Akhir Yang Diharapkan	Sistem akan menghapus data produk yang terpilih
Hasil	Sistem menghapus data produk yang terpilih
Status	Valid

6.3.21 Validasi Menambah Data Artikel

1. Validasi menambah data artikel



Tabel 6.43 adalah validasi untuk menambah data artikel yang berhasil.

Tabel 6.43 Validasi Menambah Data Artikel

Nama Kasus Uji	Menambah data artikel
Prosedur	<ol style="list-style-type: none"> Mengakses halaman tambah data artikel Form diisi dengan data: <ul style="list-style-type: none"> Judul artikel: Apa yang dimaksud dengan SPF? Sumber: THREAD by ZAIORA URL: https://thread.zalora.co.id Nama penulis: Ajeng Quamila Tanggal penulisan: 16/09/2019 Isi Artikel: POND'S Acne Solution AntiAcne Facial Foam with Active Thymo-T Essence. Mengunci bakteri dengan lock & Clear Technology untuk membantu menghilangkan jerawat dari akarnya. Efektif membantu mengatasi 10 penyebab kulit berminyak dan jerawat. Terbukti klinis mengurangi jerawat. Gambar: aaaaa.jpeg Menekan tombol "submit"
Hasil Akhir Yang Diharapkan	Data artikel akan disimpan
Hasil	Data artikel disimpan
Status	Valid

2. Validasi menambah data artikel *alternative* satu

Tabel 6.44 adalah validasi untuk menambah data artikel yang gagal karena ada *field* yang kosong

Tabel 6.44 Validasi Menambah Data Artikel *Alternative* Satu

Nama Kasus Uji	Menambah data artikel <i>alternative</i> satu
Prosedur	<ol style="list-style-type: none"> Mengakses halaman tambah data artikel Menekan tombol "Submit" tanpa mengisi <i>field</i>

Tabel 6.44 Validasi Menambah Data Artikel *Alternative* Satu (Lanjutan)

Nama Kasus Uji	Menambah data artikel <i>alternative</i> satu
Hasil Akhir Yang Diharapkan	Pesan “Isi isian ini” akan tampil pada <i>field</i>
Hasil	Pesan “Isi isian ini” tampil pada <i>field</i>
Status	Valid

6.3.22 Validasi Melihat Daftar Artikel

1. Validasi melihat daftar artikel

Tabel 6.45 adalah validasi untuk melihat daftar artikel yang berhasil.

Tabel 6.45 Validasi Melihat Daftar Artikel

Nama Kasus Uji	Melihat daftar artikel
Prosedur	1. Mengakses halaman daftar artikel dengan memilih menu “Data Artikel” pada <i>sidebar menu</i>
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman daftar artikel yang tersimpan
Hasil	Sistem menampilkan halaman daftar artikel yang tersimpan
Status	Valid

6.3.23 Validasi Mengubah Data Artikel

1. Validasi mengubah data artikel

Tabel 6.42 adalah validasi untuk mengubah data artikel yang berhasil.

Tabel 6.46 Validasi Mengubah Data Artikel

Nama Kasus Uji	Mengubah data artikel
Prosedur	<ol style="list-style-type: none"> 1. Halaman daftar artikel diakses 2. Memilih data artikel dengan memilih tombol “<i>Edit</i>” 3. Mengubah data dengan: <ul style="list-style-type: none"> • Judul artikel: Apa yang dimaksud dengan SPF? • Sumber: THREAD by ZALORA • URL: https://thread.zalora.co.id



Tabel 6.42 Validasi Mengubah Data Artikel (Lanjutan)

Nama Kasus Uji	Mengubah data artikel <ul style="list-style-type: none"> • Nama penulis: Ajeng Quamilaaa • Tanggal penulisan: 16/09/2019 • Isi Artikel: POND'S Acne Solution AntiAcne Facial Foam with Active Thymo-T Essence. Mengunci bakteri dengan lock & Clear Technology untuk membantu menghilangkan jerawat dari akarnya. Efektif membantu mengatasi 10 penyebab kulit berminyak dan jerawat. Terbukti klinis mengurangi jerawat. • Gambar: aaaaa.jpeg
	4. Menekan tombol "Save"
Hasil Akhir Yang Diharapkan	Sistem akan melakukan perubahan dan menyimpan data
Hasil	Sistem melakukan perubahan dan menyimpan data
Status	Valid

6.3.24 Validasi Menghapus Data Artikel

1. Validasi menghapus data artikel

Tabel 6.47 adalah validasi untuk menghapus data artikel yang berhasil.

Tabel 6.47 Validasi Menghapus Data Artikel

Nama Kasus Uji	Menghapus artikel
Prosedur	<ol style="list-style-type: none"> 1. Halaman daftar artikel diakses 2. Menekan tombol "Hapus" pada artikel yang dipilih
Hasil Akhir Yang Diharapkan	Data artikel akan dihapus
Hasil	Data artikel berhasil dihapus
Status	Valid

6.3.25 Validasi Melihat Produk

1. Validasi melihat halaman produk

Tabel 6.48 adalah validasi untuk melihat detail produk yang berhasil.



Tabel 6.48 Validasi Melihat Produk

Nama Kasus Uji	Melihat halaman detail produk
Prosedur	1. Memilih produk di menu <i>dropdown products</i> 2. Menekan tombol "View Detail"
Hasil Akhir Yang Diharapkan	Halaman detail produk akan ditampilkan
Hasil	Halaman detail produk ditampilkan
Status	Valid

6.3.26 Validasi Melihat Artikel

1. Validasi melihat halaman artikel

Tabel 6.49 adalah validasi untuk melihat detail artikel yang berhasil.

Tabel 6.49 Validasi Melihat Artikel

Nama Kasus Uji	Melihat halaman detail artikel
Prosedur	1. Mengakses halaman artikel 2. Memilih artikel yang ingin dilihat detailnya dengan menekan tombol "Read More"
Hasil Akhir Yang Diharapkan	Halaman detail artikel akan ditampilkan
Hasil	Halaman detail artikel ditampilkan
Status	Valid

6.3.27 Validasi Mencari Produk

1. Validasi mencari produk

Tabel 6.50 adalah validasi untuk mencari produk yang berhasil.

Tabel 6.50 Validasi Mencari Produk

Nama Kasus Uji	Mencari produk
Prosedur	1. Memasukkan <i>keyword</i> "wardah" pada <i>field</i> pencarian yang berada di bagian <i>header</i>
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman hasil pencarian produk sesuai <i>keyword</i>
Hasil	Sistem menampilkan halaman hasil pencarian produk sesuai <i>keyword</i>
Status	Valid



2. Validasi mencari produk *alternative* satu

Tabel 6.51 adalah validasi mencari produk yang gagal karena *field* pencarian tidak terisi.

Tabel 6.51 Validasi Mencari Produk *Alternative* Satu

Nama Kasus Uji	Mencari produk <i>alternative</i> satu
Prosedur	1. Menekan <i>enter</i> pada <i>keyboard</i> tanpa mengisi <i>field</i>
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan pesan “Isi isian ini” pada <i>field</i>
Hasil	Sistem menampilkan pesan “Isi isian ini” pada <i>field</i>
Status	Valid

3. Validasi mencari produk *alternative* dua

Tabel 6.52 adalah validasi untuk mencari produk yang gagal karena *keyword* tidak ditemukan.

Tabel 6.52 Validasi Mencari Artikel *Alternative* Dua

Nama Kasus Uji	Mencari produk <i>alternative</i> dua
Prosedur	1. Memasukkan <i>keyword</i> “abcdef” di <i>field</i> pencarian
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman hasil pencarian dengan pesan “Data Tidak Ditemukan”
Hasil	Sistem menampilkan halaman hasil pencarian dengan pesan “Data Tidak Ditemukan”
Status	Valid

6.3.28 Validasi Mencari Artikel

1. Validasi mencari artikel

Tabel 6.53 adalah validasi untuk mencari artikel yang berhasil.

Tabel 6.53 Validasi Mencari Artikel

Nama Kasus Uji	Mencari artikel
Prosedur	1. Halaman artikel diakses 2. <i>Keyword</i> “toner” dimasukkan di <i>field</i> pencarian
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman hasil pencarian artikel sesuai <i>keyword</i>



Tabel 6.53 Validasi Mencari Artikel (Lanjutan)

Nama Kasus Uji	Mencari artikel
Hasil	Sistem menampilkan halaman hasil pencarian artikel sesuai <i>keyword</i>
Status	Valid

2. Validasi mencari artikel *alternative* satu

Tabel 6.54 adalah validasi untuk mencari artikel yang gagal karena *field* pencarian yang tidak diisi.

Tabel 6.54 Validasi Mencari Artikel *Alternative* Satu

Nama Kasus Uji	Mencari artikel dengan <i>alternative</i> satu
Prosedur	1. Mengakses halaman artikel 2. Menekan tombol "Search" tanpa mengisi <i>field</i>
Hasil Akhir Yang Diharapkan	Pesan peringatan "Isi isian ini" akan ditampilkan pada <i>field</i>
Hasil	Sistem menampilkan pesan "Isi isian ini" pada <i>field</i>
Status	Valid

3. validasi mencari artikel *alternative* dua

Tabel 6.55 adalah validasi untuk mencari artikel yang gagal karena *keyword* tidak ditemukan.

Tabel 6.55 Validasi Mencari Artikel *Alternative* Dua

Nama Kasus Uji	Mencari artikel <i>alternative</i> dua
Prosedur	1. Mengakses halaman artikel 2. Memasukkan <i>keyword</i> "abcdef" pada <i>field</i> pencarian
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman hasil pencarian dengan pesan "Data Tidak Ditemukan"
Hasil	Sistem menampilkan halaman hasil pencarian dengan pesan "Data Tidak Ditemukan"
Status	Valid

6.3.29 Validasi Melihat Halaman *About Us*1. Validasi melihat halaman *about us*



Tabel 6.56 adalah validasi untuk melihat halaman *about us* yang berhasil.

Tabel 6.56 Validasi Melihat Halaman *About Us*

Nama Kasus Uji	Melihat halaman <i>about us</i>
Prosedur	1. Memilih menu <i>about us</i> yang terdapat di <i>header</i> atau <i>footer</i>
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman <i>about us</i>
Hasil	Halaman <i>about us</i> ditampilkan
Status	Valid

6.3.30 Validasi Melihat Halaman FAQ

1. Validasi melihat halaman FAQ

Tabel 6.57 adalah validasi untuk melihat halaman FAQ yang berhasil.

Tabel 6.57 Validasi Melihat Halaman FAQ

Nama Kasus Uji	Melihat halaman FAQ
Prosedur	1. Memilih menu FAQ yang terdapat di <i>footer</i>
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman FAQ
Hasil	Halaman FAQ ditampilkan
Status	Valid

6.3.31 Validasi Melihat Data Diri *Member* lain

1. Validasi melihat data diri *member* lain

Tabel 6.58 adalah validasi untuk melihat data diri *member* lain yang berhasil.

Tabel 6.58 Validasi Melihat Data Diri *Member* lain

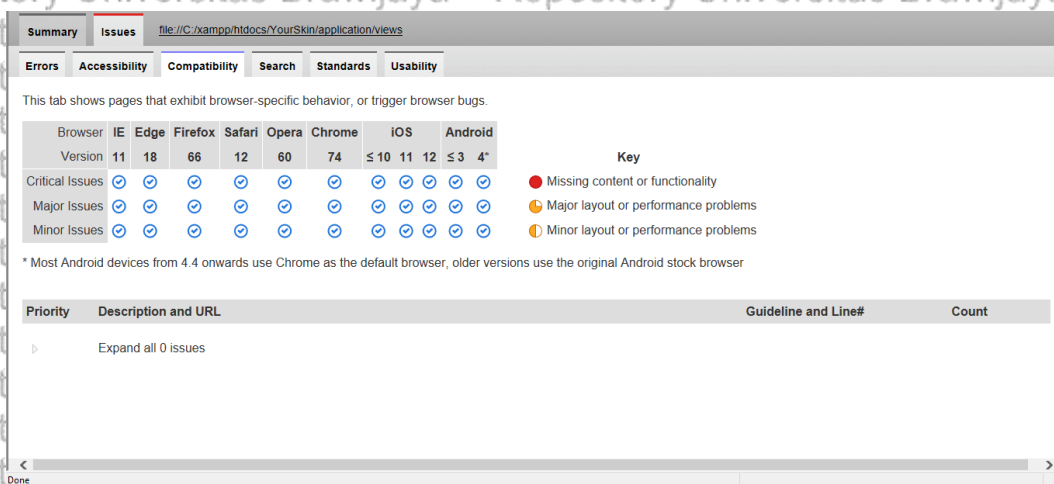
Nama Kasus Uji	Melihat data diri <i>member</i> lain
Prosedur	1. Mengakses halaman detail artikel atau detail produk 2. Melihat bagian kolom komentar atau <i>review</i> 3. Menekan nama <i>member</i> yang dipilih
Hasil Akhir Yang Diharapkan	Sistem akan menampilkan halaman <i>profile</i> dari <i>member</i>

Tabel 6.58 Validasi Melihat Data Diri *Member* Lain (Lanjutan)

Hasil	Halaman <i>profile</i> dari <i>member</i> ditampilkan
Status	Valid

6.4 Pengujian *Compatibility* (Non-Fungsional)

Untuk menguji non-fungsional sistem, digunakan pengujian *compatibility*. Tujuannya untuk melihat apakah sistem terhadap *browser* atau sistem operasi yang berbeda yang akan digunakan oleh pengguna bisa berjalan dengan baik atau tidak. Pengujian ini menggunakan aplikasi *sortsite*. Hasil pengujian *compatibility* dapat dilihat pada Gambar 6.4.

Gambar 6.7 Hasil Pengujian *Compatibility* Menggunakan Aplikasi *Sortsite*

Dapat dilihat dari hasil diatas bahwa sistem bisa berjalan/digunakan pada bermacam-macam web *browser* sesuai dengan hasil dari pengujian yang tidak mengandung indicator masalah. Sistem dapat berjalan pada web *browser* IE versi 11 atau dibawahnya, Edge versi 18 atau dibawahnya, Firefox versi 66 atau dibawahnya, Safari versi 11 atau dibawahnya, Opera versi 60 atau dibawahnya, Google Chrome versi 74 atau dibawahnya. Hasil diatas juga menunjukkan bahwa tidak ada masalah jika sistem dijalankan pada *browser* versi baru maupun versi lama.

6.5 Pengujian Akurasi

Akurasi adalah pengujian yang dibuat untuk Pengujian akurasi dilakukan untuk mengetahui nilai akurasi yang dihasilkan dari sistem. Pengujian akurasi yang dilakukan adalah dengan membandingkan hasil keluaran sistem dengan hasil pakar. Pengujian dilakukan pada tipe kulit normal, permasalahan kulit *dark circle*, dan produk yang dicari adalah krim mata.



Data uji 1 yaitu, pakar memilih bahan jelas lebih penting dari harga, bahan lebih penting dari kualitas, dan kualitas lebih penting dari harga. Hasil perbandingan akurasi dapat dilihat pada Tabel 6.59.

Tabel 6.59 Perbandingan Hasil Pakar dan Sistem

Alternatif	Peringkat Pakar	Peringkat Sistem
Viva revitalizing eye cream	3	2
Wardah White Secret Brightening Eye Cream	1	1
L'Oreal Paris White Perfect Magic White Eye Cream	6	6
Nurish Organic Brightening Eye Serum	4	4
Clinelle Eye Bright	5	5
Olay White Radiance Brightening Eye Serum	2	3

$$\text{Nilai akurasi 1} = \frac{\text{Jumlah data akurat}}{\text{Jumlah seluruh data}} \times 100\% = \frac{4}{6} \times 100\% = 66,67\%$$

Maka, pada data uji 1 didapatkan nilai akurasi 66,67%.

Data uji 2 yaitu, pakar memilih bahan lebih penting dari harga, kualitas jelas lebih penting dari bahan, dan kualitas jelas lebih penting dari harga. Hasil perbandingan akurasi dapat dilihat pada Tabel 6.60.

Tabel 6.60 Perbandingan Hasil Pakar dan Sistem

Alternatif	Peringkat Pakar	Peringkat Sistem
Viva revitalizing eye cream	4	3
Wardah White Secret Brightening Eye Cream	1	1
L'Oreal Paris White Perfect Magic White Eye Cream	6	6
Nurish Organic Brightening Eye Serum	3	5



Clinelle Eye Bright	4	4
Olay White Radiance Brightening Eye Serum	2	2

$$\text{Nilai akurasi 2} = \frac{\text{Jumlah data akurat}}{\text{Jumlah seluruh data}} \times 100\% = \frac{4}{6} \times 100\% = 66,67\%$$

Maka, pada data uji 1 didapatkan nilai akurasi 66,67%.

Berdasarkan hasil perhitungan diatas, didapatlah nilai rata-rata yang dihasilkan dari perhitungan:

$$\text{Nilai rata-rata akurasi} = \frac{\text{nilai akurasi 1} + \text{nilai akurasi 2}}{\text{Jumlah data}} = \frac{66,67\% + 66,67\%}{2} = 66,67\%$$

Dilihat dari hasil tersebut, sistem menghasilkan nilai akurasi 66,67% yang diambil dari rata-rata nilai akurasi. Jadi, dapat disimpulkan bahwa sistem ini dapat berfungsi dengan baik sesuai dengan metode *Analitycal Hierarchy Process (AHP)*.



BAB 7 PENUTUP

Bagian penutup mencakup penarikan kesimpulan dan pemberian saran dari proses analisis, proses perancangan, proses implementasi, maupun proses pengujian.

7.1 Kesimpulan

Diperoleh kesimpulan dari penelitian ini berdasarkan proses analisis, proses perancangan, pengimplementasian, maupun proses pengujian:

1. Pada proses analisis, diperoleh 31 kebutuhan fungsional dan 3 aktor. Kebutuhan diperoleh dari tahap elisitasi dengan menggunakan teknik wawancara, kuisisioner dan observasi. Wawancara dilakukan dengan pakar di bidang kesehatan. Kuisisioner dilakukan dengan memberi pertanyaan kepada responden. Observasi dilakukan dengan mengunjungi tempat yang menjual produk perawatan kulit seperti Guardian Indonesia, Watsons Indonesia, dan Century Healthcare.
2. Pada proses perancangan, diperoleh perancangan arsitektur, perancangan *sequence diagram*, perancangan *class diagram*, perancangan data, perancangan komponen, dan perancangan antarmuka. Perancangan data direpresentasikan dengan ERD (*Entity Relationship Diagram*) yang memiliki 7 (tujuh) entitas.
3. Pada proses implementasi, diperoleh implementasi basis data, implementasi kode program, dan implementasi antarmuka. Implementasi kode program dilakukan menggunakan *framework* CodeIgniter dan menggunakan bahasa pemrograman PHP dengan menerapkan metode AHP pada rekomendasi produk.
4. Pada proses pengujian, dilakukan dengan 5 (lima) pengujian. Pengujian unit dan integrasi dilakukan dengan teknik *white-box testing*. Pengujian ini menghasilkan nilai 100% *valid* pada seluruh kasus yang diuji. Pengujian validasi dilakukan dengan teknik *black-box testing*. Pengujian ini menghasilkan nilai 100% *valid* pada 31 kebutuhan fungsional. Pengujian *compatibility* dilakukan untuk menguji non-fungsional sistem. Hasil dari pengujian *compatibility* menunjukkan bahwa sistem dapat diakses dari berbagai *browser* dan sistem operasi yang berbeda. Pengujian metode rekomendasi dilakukan dengan pengujian akurasi. Diperoleh nilai akurasi 66,67% yang diambil dari rata-rata nilai akurasi. Nilai ini menunjukkan sistem dapat berfungsi dengan baik sesuai dengan metode *Analitycal Hierarcy Process* (AHP).



7.2 Saran

Saran yang dapat diberikan berdasarkan proses analisis, proses perancangan, proses implementasi, maupun proses pengujian untuk penelitian selanjutnya dengan topik yang berkaitan adalah:

1. Terdapat batasan pada kriteria dalam penelitian ini, dengan harapan pada penelitian selanjutnya dapat ditambahkan kriteria lainnya.
2. Melakukan pengujian menggunakan pengujian *usability*. Pengujian ini dilakukan untuk menguji kualitas dari sistem yang dibuat.

DAFTAR REFERENSI

- Agustini, F., 2018. Penerapan Metode AHP Pada Pemilihan Kosmetik Yang Tepat Untuk Siswi SMA. *SWABUMI*, 6(2), pp. 165-173.
- Andini, W. C., 2019. *hellosehat.com*. [Online] Available at: <https://hellosehat.com/hidup-sehat/kecantikan/urutan-pemakaian-skin-care/> [Accessed 4 March 2019].
- Benrnadifta, E. Y., Eliantara, F., K. F. D. & Sulistiowati, F., 2016. *www.academia.edu*. [Online] Available at: https://www.academia.edu/11717820/SISTEM_PENDUKUNG_KEPUTUSAN_PENERIMAAN_PEGAWAI_MENGGUNAKAN_METODE_ANALYTICAL_HIERARCHY_PROCESS_AHP_DAN_SIMPLE_ADDITIVE_WEIGHTING_SAW [Accessed 4 March 2019].
- Bertolino, 2001. Chapter 5: Software Testing. *IEEE*, 0.95(1), pp. 5-18
- Connolly, T., Begg, C., 2010. Database Systems: a practical approach to design, implementation, and management. 5th. America: Pearson Education.
- Dennis, A., Wixom, B. H. & Tegarden, D., 2012. *Systems analysis design, UML version 2.0 : an object oriented approach*. 4 ed. New York: John Wiley & Sons, Inc.
- Djuanda, A., 2007. *Ilmu Penyakit Kulit Dan Kelamin*. 5 penyunt. Jakarta: Balai Penerbit FKUI.
- Hakim, L., 2010. Membangun Web Berbasis PHP dengan Framework Codeigniter.
- Irawati, L., 2013. PENGARUH KOMPOSISI MASKER KULIT BUAH MANGGIS (GARCINIA MANGOSTANA L) DAN PATI BENGKUANG TERHADAP HASIL PENYEMBUHAN JERAWAT PADA KULIT BERMINYAK. *e-Journal*, 02(02), pp. 40-48.
- Koesheryatin & Suryana, T., 2014. *Aplikasi Internet Menggunakan HTML, CSS, & JavaScript*. Indonesia: Elex Media Komputindo.
- Kurniawan, A., 2016. Sistem Rekomendasi Produk Sepatu Dengan Menggunakan Metode Collaborative Filtering. *Seminar Nasional Teknologi Informasi dan Komunikasi*, 2016(Sentika), pp.2089–9815.
- Marthasari, G.I., Azhar, Y. and Puspitaningrum, D.K., 2015. Perlengkapan Pesta Menggunakan Collaborative Filtering Dan Penggalan. 5(1), pp.1–8.
- Miciula, I. & Nowakowska-Grunt, J., 2019. Using the AHP method to select an energy supplier for household in Polan.
- Mulyono, S., 2004. *Riset Operasi*. Jakarta: Fakultas Ekonomi Universitas Indonesia.
- Munir, 2011. *Algoritma dan pemrograman : Dalam bahasa pascal dan C*. Bandung: Informatika.



Nisswara, B. A. T., 2018. Rekomendasi Pemilihan Paket Personal Computer Menggunakan Metode AHP-TOPSIS. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(5), pp. 1998-2007.

Parrish, J. A., 1975. <https://archive.org>. [Online] Available at: <https://archive.org/stream/dermatologyskin00parr#page/n9/mode/2up/search/skin+care> [Accessed 3 Oktober 2019].

Penzer, R. & Ersser, S., 2010. *Principles of Skin Care: A Guide for Nurses and Health Care Practitioners*. 1nd ed. United Kingdom: John Wiley & Sons.

Perdhana, S. A., Laksito, W. & Siswanti, S., 2013. Sistem Pendukung Keputusan Pemilihan Jenis Laptop Dengan Menggunakan Metode Analytical Hierarchy Process.

Pressman, R. S., 2010. *Software Engineering a Practitioner's Approach*. 7 ed. New York: Mc-Graw-Hill Higher Education.

Romney, M. B. & Steinbart., P. J., 2015. *Sistem Informasi Akuntansi*. 13 ed. Jakarta: Salemba Empat.

Saaty, T. L., 2000. *Fundamentals of Decision Making and Priority Theory With the Analytic Hierarchy Process*. Pittsburg: RWS Publications.

Sommerville, I., 2011. *Software Engineering (Rekayasa Perangkat Lunak)*. 6th ed. Jakarta: Erlangga

Sutanta, E., 2009. *Sistem Informasi Manajemen*. Yogyakarta: GRAHA ILMU

System, S., 2016. *UML Models*. 1 ed. England: Enterprise Architect.

Syukron, A., 2014. *Pengantar Manajemen Industri*. Jakarta: Graha Ilmu.

Tresna, P., 2010. *MODUL PERAWATAN KULIT WAJAH (FACIAL)*. Bandung: Universitas Pendidikan Indonesia.

Upton, D. and Blanco, J.A., 2009. *Codeigniter*. Birmingham: Packt Publishing.

Yuliani, I. D. A. E., 2016. Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Dengan Metode Analytical Hierarchy Process. *JURNAL SISFOTENIKA*, III(2)



LAMPIRAN A

A.1 Wawancara

A.1.1 Latar Belakang

Salah satu cara pengumpulan data adalah melalui kegiatan wawancara. Dalam kegiatan wawancara, didapatkan sejumlah informasi dari narasumber. Wawancara ini ditujukan kepada satu narasumber yaitu dr. Grace. Wawancara ini dilakukan untuk mendapatkan informasi bagaimana memilih produk perawatan kulit yang tepat. Kulit sangat penting untuk dijaga karena kulit memiliki banyak fungsi. Untuk menjalankan fungsi-fungsi tersebut, kulit harus dijaga dan dirawat dengan benar. Terutama pada kulit wajah. Selain dapat menarik perhatian, kulit wajah yang indah dan sehat juga dapat meningkatkan kepercayaan diri. Kulit bisa menjadi iritasi dan dehidrasi jika salah memilih dan memakai produk. Maka dari itu, dikumpulkanlah informasi mengenai bagaimana memilih produk yang tepat melalui kegiatan wawancara ini.

A.1.2 Maksud dan Tujuan

1. Mendapatkan informasi tentang pentingnya merawat kulit.
2. Mendapatkan informasi tentang bagaimana memilih produk perawatan kulit wajah yang tepat
3. Mendapatkan informasi bahan yang disarankan ada pada produk untuk mengatasi permasalahan-permasalahan kulit.

A.1.3 Topik Wawancara

Topik kegiatan wawancara ini adalah pemilihan produk perawatan kulit wajah yang tepat.

A.1.4 Waktu Pelaksanaan

Wawancara dilakukan pada:

- Hari/Tanggal : 18 Maret 2019
- Pukul : 14:00 – 16:00

A.1.5 Narasumber dan Pewawancara

- Narasumber : dr. Grace Roseliny Pangaribuan
- Pewawancara : Jojo Yeanesy S



A.1.6 Hasil Wawancara

Pertanyaan	Jawaban
<p>Penting kah melakukan perawatan pada kulit wajah?</p>	<p>Iya penting. Karena kulit banyak memiliki fungsi. Fungsi kulit dapat berperan dalam sistem kekebalan tubuh yaitu pertahanan pertama tubuh, fungsi estetika, fungsi pengaturan suhu tubuh, sebagai indra peraba, melindungi jaringan dibawahnya dari kerusakan akibat sinar matahari, tempat perubahan pro vitamin D menjadi D, dan lainnya. Kulit wajah adalah cermin pertama seseorang yang langsung berhadapan dengan mata pertama kali. Maka dari itu menjaga kesehatan kulit wajah sangat penting.</p>
<p>Cara apa saja yang dilakukan untuk menjaga Kesehatan kulit wajah?</p>	<p>Cara nya ada bermacam macam. Apalagi dengan perkembangan teknologi sekarang. Perawatan bisa dilakukan di klinik kecantikan dengan alat-alat yang canggih. Tapi seperti yang kita tahu, bahwa melakukan perawatan di klinik membutuhkan biaya yang tidak sedikit. Ada juga perawatan dengan menggunakan bahan-bahan alami. Tapi hasil yang kita dapat tidak secepat dengan yang dilakukan di klinik. Ada juga yang dengan memakai produk perawatan kulit. Cara ini paling umum dilakukan karena selain produk dapat ditemukan dengan mudah, biaya yang kita keluarkan tidak banyak.</p>
<p>Bagaimana memilih produk perawatan kulit?</p>	<p>Untuk memilih produk perawatan kulit disarankan agar memilih produk yang sesuai dengan tipe kulit dan permasalahan kulit. Selain itu diperhatikan juga bahan yang terkandung dalam produk. Hindari produk yang mengandung bahan berbahaya seperti paraben, sulfat, merkuri. Lihat juga masa berlaku produk. Pilihlah produk yang mengandung bahan yang dapat mengatasi permasalahan kulit. Kualitas dari produk juga harus diperhatikan. Kualitas bisa dilihat pada adanya izin edar BPOM dan label halal. Halal disini untuk memastikan produk diproduksi dengan halall dan menggunakan bahan halal.</p>



<p>Apa yang terjadi jika produk yang dipilih tidak tepat?</p>	<p>Kulit kita bisa mengalami iritasi atau dehidrasi. Bukannya memperbaiki masalah yang ada, sebaliknya membuat masalah menjadi lebih parah.</p>
<p>Jika terjadi iritasi pada kulit, apa yang harus dilakukan?</p>	<p>Jika terjadi iritasi pada kulit, hentikan dahulu pemakaian produk. Apabila iritasi berlanjut, silahkan datang ke dokter untuk menanyakan lebih lanjut penyebabnya. Mungkin saja itu karena ada alergi pada bahan yang terkandung pada produk.</p>
<p>Tipe kulit itu sendiri ada apa saja?</p>	<p>Berdasarkan kelembapannya, tipe kulit dibagi atas kulit kering dan kulit berminyak. Berdasarkan kadar melaninnya, tipe kulit dibagi atas kulit berpigmen dan kulit tidak berpigmen. Berdasarkan kecenderungan untuk bereaksi, tipe kulit dibagi atas kulit sensitif dan kulit resisten (tidak sensitif).</p>
<p>Permasalahan kulit itu sendiri apa saja?</p>	<p>Untuk permasalahan kulit ada bermacam-macam. Ada kulit berjerawat, keriput, komedo, lingkaran hitam di area sekitar mata, kulit kusam, kulit kering, pori-pori besar, milia (bintik putih) dan kemerahan/sensitif di wajah.</p>
<p>Bahan-bahan apa saja yang harus ada dalam produk berdasarkan permasalahan yang diatasi?</p>	<p>Untuk masalah jerawat, pilihlah produk yang mengandung asam salisilat, <i>tea tree oil</i>, atau <i>sulfur</i>. Untuk mengatasi tanda-tanda penuaan, pilihlah produk yang mengandung antioksidan. Untuk masalah komedo pilihlah produk yang mengandung <i>hyaluronic acid</i>, AHA, atau BHA. Untuk lingkaran hitam di area mata pilihlah produk yang terdapat kandungan bahan untuk mencerahkan seperti retinol, niacinamide, atau vitamin C dan kandungan bahan untuk melembapkan seperti <i>hyaluronic acid</i> atau glycerin atau ceramide. Untuk mengatasi kulit kering atau kulit dehidrasi, pilihlah produk yang terdapat kandungan <i>hyaluronic acid</i>. Untuk mengatasi kulit kusam, pilihlah produk yang terdapat kandungan niacinamide. Untuk mengatasi pori-pori besar, pilihlah produk yang terdapat kandungan bahan AHA atau BHA. Untuk mengatasi kemerahan di wajah, pilihlah</p>



	produk yang terdapat kandungan bahan untuk menyejukkan seperti oatmeal.
Berapa waktu yang diperlukan untuk melihat hasil dari penggunaan produk perawatan kulit?	Untuk melihat hasilnya tidak bisa instan. Diperlukan waktu beberapa minggu bahkan dalam hitungan bulan. Oleh karena itu, saya sarankan produk digunakan secara rutin agar hasilnya dapat terlihat.
Apakah ada pengaruh urutan pemakaian produk pada hasil penggunaan?	Ada. Gunakanlah produk yang memiliki tekstur ringan (mudah diserap oleh kulit) dan dilanjutkan dengan produk yang memiliki tekstur berat (berminyak).
Apa saja urutan pemakaian produk perawatan kulit?	<p>Urutan pemakaian dimulai dari:</p> <ol style="list-style-type: none"> 1. Pemakaian produk pembersih 2. Pemakaian produk toner (penyegar) 3. Pemakaian produk essence/serum <p>Untuk urutan ini disesuaikan saja mana yang dipakai terlebih dahulu. Karena terkadang ada essence yang memiliki tekstur cair daripada serum. Begitupun sebaliknya.</p> <ol style="list-style-type: none"> 4. Pemakaian produk pelembap 5. Pemakaian sunscreen (tabir surya) pada siang hari

Narasumber

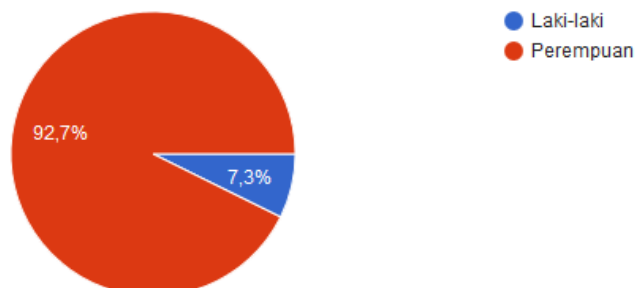
dr. Grace Roseliny P.



A.2 Hasil Kuesioner

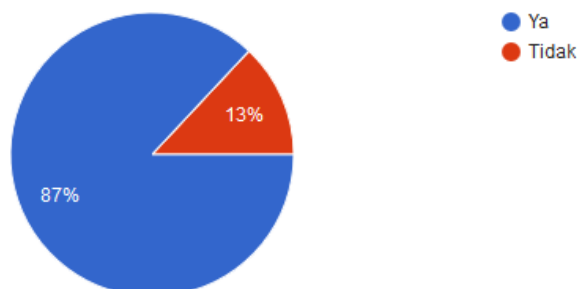
Jenis Kelamin

123 tanggapan



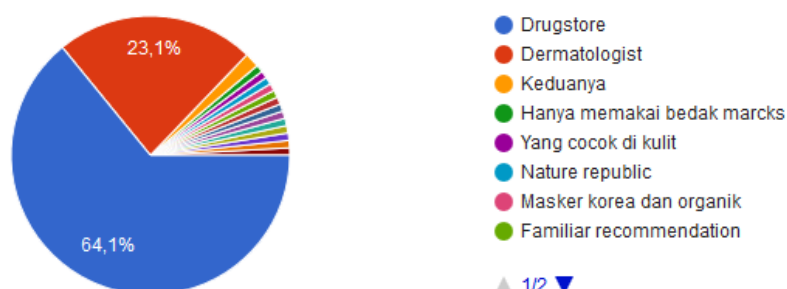
Apakah anda memakai produk skincare?

123 tanggapan



Jika anda menggunakan produk skincare, produk apa yang anda gunakan?

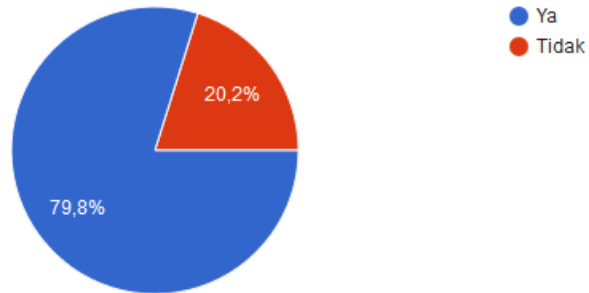
117 tanggapan





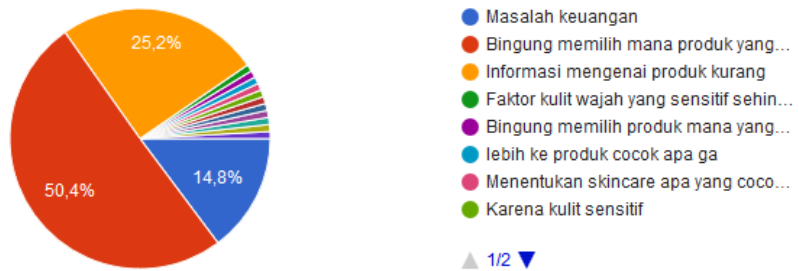
Apakah anda pernah kesulitan dalam memilih produk skincare yang dibutuhkan?

119 tanggapan



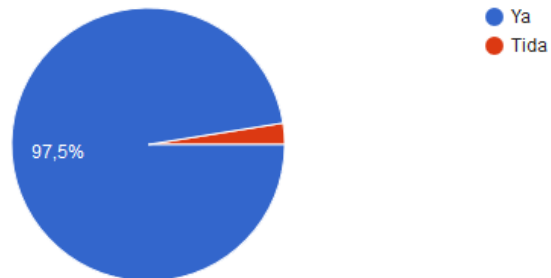
Jika anda pernah mengalami kesulitan dalam memilih produk skincare, apa saja kesulitannya?

115 tanggapan



Apakah anda setuju jika ada sebuah sistem yang dapat membantu anda dalam mengatasi kesulitan-kesulitan yang dialami dalam memilih produk skincare?

119 tanggapan





LAMPIRAN B

B.1 Contoh Perhitungan Manual AHP

1. Input skala perbandingan saaty dan jumlahkan nilai kriteria setiap kolom.

TABEL PERBANDINGAN			
Kriteria	Harga	Bahan	Kualitas
Harga	1	5	3
Bahan	1/5	1	1/3
Kualitas	1/3	3	1
Jumlah	1.533333333	9	4.333333333

2. Normalisasi Matriks Perbandingan

Hasil normalisasi didapatkan dari pembagian antara tiap *cell* dengan jumlah baris tiap kriteria.

Contoh:

$$1/1.533333333 = 0.652173913$$

Matriks Normalisasi			
Kriteria	Harga	Bahan	Kualitas
Harga	0.652173913	0.555555556	0.692307692
Bahan	0.130434783	0.111111111	0.076923077
Kualitas	0.217391304	0.333333333	0.230769231
Jumlah	1	1	1

3. Menghitung Bobot Prioritas

Hasil bobot prioritas didapatkan dari menjumlahkan nilai kolom tiap baris kemudian membagi nilai tersebut dengan banyak kriteria yang ada.

Contoh:

$$\text{Baris Harga} = \frac{0.652173913 + 0.555555556 + 0.692307692}{3} = 0.63334572$$

Kriteria	Bobot Prioritas
Harga	0.63334572
Bahan	0.106156324
Kualitas	0.260497956



4. Menghitung Eigen Maksimum

Eigen maksimum dihitung dimulai dengan cara menjumlahkan baris hasil perkalian nilai tiap *cell* dengan nilai bobot kriteria, kemudian hasilnya dibagi dengan nilai bobot per kriteria untuk masing masing baris. Hasil perhitungan lamda tiap kriteria dapat dilihat sebagai berikut:

Contoh:

$$\text{Baris Harga} = \frac{(1 \times 0.63334572) + (5 \times 0.106156324) + (3 \times 0.260497956)}{0.63334572} = 3.071973401$$

Kriteria	Harga	Bahan	Kualitas	Rata-rata
Harga	0.63334572	0.530781618	0.781493868	3.071973401
Bahan	0.126669144	0.106156324	0.086832652	3.011201867
Kualitas	0.21111524	0.318468971	0.260497956	3.032968775

Eigen maksimum dihitung dengan cara menjumlahkan setiap lamda kriteria dibagi dengan banyak kriteria. Proses perhitungannya sebagai berikut:

$$\lambda_{\max} = \frac{3.071973401 + 3.011201867 + 3.032968775}{3}$$

$$\lambda_{\max} = 3.038714681$$

5. Menghitung Indeks Konsisten dan Rasio Kosisten

Proses perhitungan nilai indeks konsisten.

$$CI = \frac{\lambda_{\max} - n}{n - 1}$$

Penjelasan:

λ_{\max} = Nilai eigen maksimum

n = banyaknya kriteria

$$CI = \frac{3.038714681 - 3}{3 - 1}$$

$$= 0.01935734$$

$$CR = \frac{CI}{RI}$$

Untuk indeks rasio (RI) menggunakan nilai 0.58 karena menggunakan matriks berordo 3.



Proses perhitungannya sebagai berikut.

$$CR = \frac{0.01935734}{0.58}$$

$$CR = 0.033374725$$

Karena nilai CR yang didapat ≤ 0.1 , maka bobot prioritas yang didapatkan sudah dapat dikatakan layak dan konsisten.