

**SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT
KULIT WAJAH BERBASIS *MOBILE***

TUGAS AKHIR



Oleh:

Yusuf Sofyan
0911510923

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR
JAKARTA
2013

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT KULIT WAJAH BERBASIS *MOBILE*

TUGAS AKHIR

**Diajukan untuk memenuhi salah satu persyaratan
Memperoleh gelar Sarjana Komputer (S.Kom)**



Oleh:

Yusuf Sofyan
0911510923

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

JAKARTA
2013



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

LEMBAR PENGESAHAN

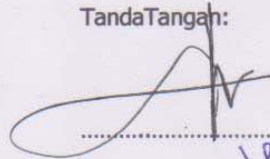
Nama : Yusuf Sofyan
Nomor Induk Mahasiswa : 0911510923
Program Studi : Teknik Informatika
Jenjang Studi : Strata 1
Judul : SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT KULIT
WAJAH BERBASIS *MOBILE*

Jakarta, 9 Juli 2013

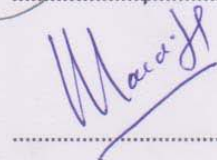
Tim Penguji:

TandaTangan:

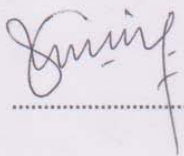
Ketua,
(Krisna Adiyarta Musodo, S.Kom, M.Sc)




Anggota
(Mardi Hardjianto, M.Kom)



Pembimbing,
(Sri Mulyati M.Kom)



Ketua Program Studi



.....
Muhammad Ainur Rony, S.Kom, M.T.I

ABSTRAKSI

Nim **0911510923**
Nama **YUSUF SOFYAN**
Judul **SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT KULIT WAJAH**
 BERBASIS *MOBILE*

Wajah merupakan salah satu bagian yang paling penting dari penampilan. Tidak harus memiliki wajah cantik dan berkulit putih untuk bisa tampil menarik, tetapi dengan memiliki kulit wajah yang sehat dan bersih pun kita dapat tampil dengan menarik dan percaya diri. Sistem pakar untuk mendiagnosa penyakit kulit wajah merupakan suatu sistem pakar yang dirancang sebagai alat bantu dalam mendiagnosa penyakit kulit wajah bagi para penderita dengan tujuan dalam pengambilan keputusan. Sehingga diharapkan sistem pakar ini dapat membantu para ahli dalam mendiagnosa penyakit kulit wajah. Tujuan dari sistem pakar untuk mendiagnosa penyakit kulit wajah berbasis *mobile* ini agar dapat digunakan oleh *user* untuk dapat mengetahui penyakit kulit wajah yang diderita. Dengan menggunakan sistem pakar ini diharapkan dapat mengurangi kesalahan manusia dalam mendiagnosa penyakit kulit wajah karena sistem ini sangat membantu masyarakat hanya cukup memegang *handphone* saja dapat mengetahui penyakit kulit wajah yang dialami serta solusi atau cara pengobatannya. Aplikasi yang digunakan menggunakan bahasa pemrograman Java dan basis data Mysql.

Kata Kunci : Sistem Pakar, *Mobile*, Penyakit Kulit Wajah

xii+82 halaman; 57 gambar; 6 tabel; 2 lampiran

SURAT PERNYATAAN TIDAK PLAGIAT DAN PERSETUJUAN PUBLIKASI

Saya yang bertanda tangan di bawah ini :

Nama : YUSUF SOFYAN
NIM : 0911510923
Program Studi : TEKNIK INFORMATIKA
Fakultas : TEKNOLOGI INFORMASI

Menyatakan bahwa TUGAS AKHIR yang berjudul :

SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT KULIT WAJAH
BERBASIS MOBILE

1. Merupakan hasil karya tulis ilmiah sendiri dan bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain.
2. Saya ijin untuk dikelola oleh Universitas Budi Luhur sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila dikemudian hari pernyataan ini tidak benar.

Jakarta, 24 Juni 2013



(YUSUF SOFYAN)

KATA PENGANTAR

Puji Syukur penulis panjatkan atas kehadiran Allah S.W.T Yang Maha Pemurah lagi maha penyayang, yang telah memberikan anugrah, rahmat serta hidayah-Nya kepada penulis, sehingga skripsi ini dapat terselesaikan dengan baik.

Skripsi ini disusun sebagai salah satu syarat untuk memenuhi kurikulum program sarjana strata satu (S1) pada program studi Teknik Informatika Fakultas Teknologi Informasi Universitas Budi Luhur Jakarta.

Pada kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu, baik secara langsung maupun tidak langsung dalam pembuatan skripsi ini hingga selesai. Ucapan terima kasih penulis sampaikan kepada :

1. Allah SWT atas rahmat dan karunia-Nya yang diberikan untuk menyelesaikan skripsi ini.
2. Ayah, Bunda, Adik yang selama ini memberi semangat, perhatian, dan doa hingga skripsi ini terselesaikan.
3. Bapak Prof. Ir. Suryo Hapsoro Tri Utomo, Ph,.D selaku Rektor Universitas Budi Luhur.
4. Bapak Goenawan Brotosaputro, S.Kom, M.Sc selaku Dekan Fakultas Teknologi Informasi.
5. Bapak Muhammad Ainur Ronny, S.Kom, M.T.I sebagai Ketua Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur Jakarta.
6. Ibu Sri Mulyati, M.Kom selaku dosen pembimbing yang telah dengan baik membimbing penulis untuk menyelesaikan skripsi ini tepat pada waktunya.
7. Semua pihak dan teman-teman lainnya yang tidak dapat saya sebutkan satu persatu.

Penulis juga menyadari sepenuhnya bahwa masih banyak terdapat kekurangan, untuk itu diharapkan adanya kritik dan saran yang bersifat membangun demi meningkatkan wawasan serta besar harapan semoga skripsi ini dapat bermanfaat bagi semua pihak.

Jakarta, Juni 2013

Penulis

DAFTAR TABEL

	Halaman
Tabel 3.1 : Tabel pertanyaan	17
Tabel 3.2 : Tabel Kesimpulan	18
Tabel 3.3 : Tabel Admin	18
Tabel 3.4 : Tabel Kamus Bahasa	18
Tabel 3.5 : Tabel Arahkan	19
Tabel 3.6 : Tabel Jurusan	19


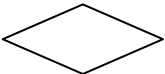

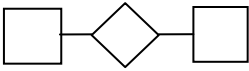
DAFTAR GAMBAR

	Halaman
Gambar 2.1 : Teknik <i>Depth-First Search</i>	6
Gambar 2.2 : Teknik <i>Breadth-First Search</i>	7
Gambar 2.3 : Teknik <i>Best-First Search</i>	7
Gambar 2.4 : Tahapan Pembuatan Sistem Pakar	10
Gambar 2.5 : Hubungan Java ME dan Java SE	12
Gambar 3.1 : ERD (<i>Entity Relationship Diagram</i>).....	16
Gambar 3.2 : Transformasi ERD ke <i>Logical Record Structure</i>	16
Gambar 3.3 : Diagram <i>Logical Record Structure</i>	17
Gambar 3.4 : Rancangan layar <i>Splashscreen</i>	19
Gambar 3.5 : Rancangan Layar Menu Utama	20
Gambar 3.6 : Rancangan Layar Pertanyaan	20
Gambar 3.7 : Rancangan Layar Kesimpulan	21
Gambar 3.8 : Rancangan Layar pada Arahan	21
Gambar 3.9 : Rancangan Layar Kamus Bahasa	22
Gambar 3.10 : Rancangan Layar <i>Login</i> Admin.....	22
Gambar 3.11 : Rancangan Layar <i>About Me</i>	23
Gambar 3.12 : Rancangan Layar Menu Utama Admin	23
Gambar 3.13 : Rancangan Layar Admin Pertanyaan	24
Gambar 3.14 : Rancangan Layar Admin Kesimpulan	24
Gambar 3.15 : Rancangan Layar Admin Arahan	25
Gambar 3.16 : Rancangan Layar Admin Jurusan	25
Gambar 3.17 : Rancangan Layar Admin Kamus Bahasa	26
Gambar 3.18 : Rancangan Layar <i>Update</i> Admin	26
Gambar 3.19 : <i>Flowchart</i> Layar Menu Utama	27
Gambar 3.20 : <i>Flowchart</i> Layar Identifikasi	28
Gambar 3.21 : <i>Flowchart</i> Layar Kamus Bahasa.....	29
Gambar 3.22 : <i>Flowchart</i> Layar <i>Login</i> Admin.....	30
Gambar 3.23 : <i>Flowchart</i> Layar Menu Utama Admin	31
Gambar 3.24 : <i>Flowchart</i> Layar Admin Pertanyaan	32
Gambar 3.25 : <i>Flowchart</i> Layar Admin Kesimpulan	33
Gambar 3.26 : <i>Flowchart</i> Layar Admin Arahan	34
Gambar 3.27 : <i>Flowchart</i> Layar Admin Jurusan	35
Gambar 3.28 : <i>Flowchart</i> Layar Admin Kamus Bahasa	36
Gambar 3.29 : <i>Flowchart</i> Layar <i>Update</i> Admin	37
Gambar 4.1 : Tampilan Layar Awal.....	47
Gambar 4.2 : Tampilan layar <i>form</i> Utama	48
Gambar 4.3 : Tampilan Layar Identifikasi.....	49
Gambar 4.4 : Tampilan Layar <i>Form</i> Kesimpulan	49
Gambar 4.5 : Tampilan Layar <i>form</i> Kamus	50
Gambar 4.6 : Tampilan Layar <i>form</i> Profil	50
Gambar 4.7 : Tampilan Layar <i>Login</i>	51
Gambar 4.8 : Tampilan Layar <i>Form</i> Admin.....	51
Gambar 4.9 : Tampilan Layar Ubah Pertanyaan	52
Gambar 4.10 : Tampilan Layar Ubah Kesimpulan	52
Gambar 4.11 : Tampilan Layar Ubah Arahan.....	53
Gambar 4.12 : Tampilan Layar Ubah Jurusan.....	53


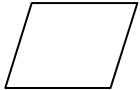

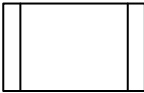
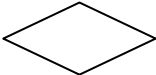



Gambar 4.13 : Tampilan Layar Ubah Kamus	54
Gambar 4.14 : Tampilan Layar Ubah Data Admin	54
Gambar 4.15 : Tampilan Layar <i>form</i> Utama Sebelum Identifikasi Dimulai.....	55
Gambar 4.16 : Tampilan layar untuk memulai identifikasi	56
Gambar 4.17 : Tampilan layar pertanyaan pertama (level 1)	56
Gambar 4.18 : Tampilan layar pertanyaan kedua(level 2)	57
Gambar 4.19 : Tampilan layar pertanyaan ketiga(level 3)	57
Gambar 4.20 : Tampilan layar pertanyaan keempat(level 4)	58
Gambar 4.21 : Tampilan layar pertanyaan kelima(level 5).....	58
Gambar 4.22 : Tampilan layar Identifikasi menemukan kesimpulan	59
Gambar 4.23 : Tampilan layar Identifikasi tidak menemukan data yang cocok	59

DAFTAR SIMBOL

1. ERD (*Entity Relationship Diagram*)

Simbol	Keterangan
	<i>Entity</i> (Entitas) Menggambarkan suatu objek, orang, tempat yang <i>field-field</i> nya dipergunakan dalam aplikasi program.
	<i>Relationship</i> (hubungan) Menunjukkan adanya hubungan antara satu entitas dengan entitas lainnya.
	Garis Menunjukkan hubungan antara relasi dengan entitas.
	<i>Cardinality</i> (kardinalitas) Menggambarkan tingkat hubungan yang terjadi pada dua buah entitas.

2. *Flowchart* Diagram

Simbol	Keterangan
	Terminal Points Menggambarkan kegiatan awal atau akhir suatu proses.
	<i>Input / Output</i> Menggambarkan <i>Input</i> data atau <i>Output</i> data yang di proses atau informasi.
	Proses Menggambarkan suatu operasi atau proses.
	<i>Predefine Proses</i> Menggambarkan rincian proses berada di tempat lain atau masih bisa dijabarkan dalam algoritma.
	Keputusan Menggambarkan suatu keputusan dalam program.
	Penghubung Menghubungkan keluar atau masuk suatu simbol dengan simbol lain pada halaman yang sama.
	<i>Off-page Reference</i> Menghubungkan suatu simbol dengan simbol yang lain pada halaman yang berbeda
	<i>Line Connector</i> Menghubungkan suatu simbol dengan simbol yang lain pada modul yang sama.

DAFTAR ISI

	Halaman
COVER	i
LEMBAR PERSETUJUAN	ii
ABSTRAKSI	iii
SURAT TIDAK PLAGIAT	iv
KATA PENGANTAR	v
DAFTAR TABEL	vi
DAFTAR GAMBAR	vii
DAFTAR SIMBOL	ix
DAFTAR ISI	x
BAB I PENDAHULUAN	1
1. Latar Belakang	1
2. Masalah	1
3. Tujuan Penulisan	1
4. Batasan Permasalahan	1
5. Metode Pembangunan	1
a. Permasalahan Identifikasi Masalah	2
b. Analisa Masalah	2
c. Membentuk Basis Pengetahuan	2
d. Memilih Teknik Inferensi Pengetahuan	2
e. Pengujian	2
f. Pemeliharaan	2
6. Sisitematika Penulisan	2
BAB II LANDASAN TEORI	4
1. Konsep Dasar Kecerdasan Buatan	4
2. Sistem Pakar	4
a. Definisi Sistem Pakar	4
b. Pembentukan Umum <i>Expert System</i>	4
c. Ciri-ciri Sistem pakar	5
d. Komponen Sistem Pakar	6
e. Tipe-tipe Sistem Pakar	8
f. Klasifikasi Sistem Pakar	8
g. Kemampuan Tambahan yang Diperlukan Sistem Pakar	9
h. Keuntungan Sistem Pakar	9
i. Kerugian Sistem Pakar	9
j. Struktur Sistem Pakar	10
k. Tahapan Pembuatan Sistem Pakar	10
3. Teori Pemrograman	11
a. Java	11
b. <i>Personal Home Page</i>	13
c. <i>Database</i>	14
4. Penyakit Kulit	14

BAB III ANALISA MASALAH DAN RANCANGAN SISTEM.....	15
1. Analisa Masalah	15
2. Penyelesaian Masalah.....	15
3. Rancangan Sistem Pakar	15
a. Mesin Inferensi	15
b. Pohon Keputusan	15
4. Struktur Basis Data	16
a. ERD (<i>Entity Relationship Diagram</i>)	16
b. Transformasi ERD ke <i>Logical Record Structure</i>	16
c. <i>Logical Record Structure</i>	17
5. Rancangan <i>Database</i>	17
a. Tabel Pertanyaan	17
b. Tabel Kesimpulan	18
c. Tabel Admin	18
d. Tabel Kamus Bahasa	18
e. Tabel Arahkan	18
f. Tabel Jurusan	19
6. Rancangan Layar	19
a. Rancangan Layar <i>Splash Screen</i>	19
b. Rancangan Layar Menu Utama <i>Mobile</i>	20
c. Rancangan Layar Pertanyaan	20
d. Rancangan Layar Kesimpulan.....	21
e. Rancangan Layar Arahkan	21
f. Rancangan Layar Kamus Bahasa.....	22
g. Rancangan Layar <i>Login</i> Admin	22
h. Rancangan Layar <i>About Me</i>	23
i. Rancangan Layar Menu Utama Admin	23
j. Rancangan Layar Admin Pertanyaan	24
k. Rancangan Layar Admin Kesimpulan	24
l. Rancangan Layar Admin Arahkan.....	25
m. Rancangan Layar Admin Jurusan.....	25
n. Rancangan Layar Admin Kamus Bahasa.....	26
o. Rancangan Layar <i>Update</i> Admin.....	26
7. <i>Flowchart</i>	27
a. <i>Flowchart</i> Layar Menu Utama	27
b. <i>Flowchart</i> Layar Identifikasi	28
c. <i>Flowchart</i> Layar Kamus Bahasa	29
d. <i>Flowchart</i> Layar <i>Login</i> Admin	30
e. <i>Flowchart</i> Layar Menu Utama Admin.....	30
f. <i>Flowchart</i> Layar Admin Pertanyaan	32
g. <i>Flowchart</i> Layar Admin Kesimpulan	33
h. <i>Flowchart</i> Layar Admin Arahkan.....	34
i. <i>Flowchart</i> Layar Admin Jurusan.....	35
j. <i>Flowchart</i> Layar Admin Kamus Bahasa.....	36

k. <i>Flowchart</i> Layar <i>Update</i> Admin	37
8. Algoritma	38
a. Algoritma Menu Utama	38
b. Algoritma Layar Identifikasi.....	38
c. Algoritma Kamus Bahasa.....	39
d. Algoritma <i>Login</i> Admin	40
e. Algoritma Menu Utama Admin.....	40
f. Algoritma Admin Pertanyaan.....	40
g. Algoritma Admin Kesimpulan.....	41
h. Algoritma Admin Arahkan	41
i. Algoritma Admin Jurusan	43
j. Algoritma <i>Update</i> Admin Kamus Bahasa	44
k. Algoritma Layar <i>Update</i> Admin	45
BAB IV IMPLEMENTASI DAN ANALISA PROGRAM.....	47
1. Implementasi	47
a. Tampilan Layar Awal	47
b. Tampilan Layar <i>Form</i> Utama	48
c. Tampilan Layar <i>Form</i> Identifikasi.....	49
d. Tampilan Layar <i>Form</i> Kesimpulan.....	49
e. Tampilan Layar Kamus	50
f. Tampilan Layar Profil	50
g. Tampilan Layar <i>Login</i>	51
h. Tampilan Layar <i>Form</i> Admin	51
i. Tampilan Layar Ubah Pertanyaan	52
j. Tampilan Layar Ubah Kesimpulan	52
k. Tampilan Layar Ubah Arahkan	53
l. Tampilan Layar Ubah Jurusan	53
m. Tampilan Layar Ubah Kamus.....	54
n. Tampilan Layar Ubah Data Admin	54
2. Uji Kasus	55
a. Spesifikasi <i>Hardware & Software</i>	55
b. Cara Menjalankan Program Penelusuran Atau Sistem Pakar	55
3. Kelebihan dan Kekurangan Program	60
a. Kelebihan Program	60
b. Kekurangan Program.....	60
BAB V PENUTUP.....	61
1. Kesimpulan	61
2. Saran	61
DAFTAR PUSTAKA.....	62
Lampiran Rule.....	63
Lampiran Pohon Keputusan	75

BAB I PENDAHULUAN

1. Latar Belakang

Kemajuan ilmu pengetahuan dan teknologi khususnya dibidang teknologi informasi, dalam sisi kehidupan tidak terlepas dari komputerisasi termasuk dunia kesehatan. Langkanya ketersediaan seorang pakar untuk menangani suatu bidang keahlian tertentu dapat menghambat kelancaran penyelesaian pekerjaan dalam bidang tersebut. Sebagai contoh yaitu dibidang penyakit kulit wajah. Hal ini tentu saja akan menghambat aktivitas pihak yang meneliti tentang penyakit kulit wajah, karena akan memakan waktu yang lama untuk menentukan letak penyakit yang diteliti oleh setiap orangnya. Analisa dan penentuannya dapat saja dilakukan oleh seorang yang telah mengerti tentang teknik penyakit, tetapi apabila orang tersebut belum memiliki pengetahuan yang cukup, akan mengakibatkan hasil penentuan-penentuan yang akan berakibat fatal.

Oleh karena itu dalam penulisan tugas akhir ini akan dibuat sistem pakar mendiagnosa penyakit kulit wajah berbasis *mobile*. Sistem pakar ini nantinya akan mempermudah diagnosa secara cepat mengenai penyakit kulit wajah, serta menemukan solusi dari penyakit tersebut. Sistem pakar ini akan lebih terasa efektif serta efisien, apabila pengguna dapat dengan mudah dan cepat mudah mendapatkan informasi dimana pun dan kapan pun. Hal tersebut dapat dilakukan dengan menggunakan aplikasi sistem pakar berbasis *mobile* yang bersifat dinamis.

2. Masalah

Kulit wajah merupakan salah satu kesehatan kulit yang dapat mempengaruhi penampilan seseorang. Terbukti dengan banyak masyarakat terutama remaja dan ibu-ibu yang sering memiliki penyakit kulit wajah dan menyebabkan seseorang menjadi tidak percaya diri. Terbatasnya tim dokter yang dapat mendiagnosa penyakit kulit wajah, sehingga membuat penulis tertarik untuk mengimplementasikan pengetahuan sang pakar kedalam sebuah sistem pakar.

3. Tujuan Penulisan

Tujuan penulisan sistem pakar ini adalah membuat aplikasi yang dapat mendiagnosa panyakit kulit wajah, yaitu aplikasi sistem pakar (*expert system*). Manfaat yang didapat dari sistem pakar ini adalah memberikan pengetahuan bagi pihak - pihak yang membutuhkan informasi mengenai penyakit kulit wajah. Selain itu sistem pakar ini juga dapat digunakan oleh calon pakar sebagai dokumentasi untuk menuangkan pengetahuannya kedalam *database*.

4. Batasan Permasalahan

Pada rancangan sistem pakar ini, permasalahan yang diangkat hanya terbatas tentang mendiagnosa penyakit kulit wajah, dengan menjelaskan informasi mengenai keluhan-keluhan yang dialami oleh pasien dengan berdasarkan pada sumber pengetahuan yang didapat dari buku penyakit kulit dan hasil riset langsung di Maharis Clinic Jl. Malabar No. 19 Guntur-Setia Budi, Jakarta Selatan.

5. Metode Pembangunan

Agar dapat memperoleh data yang akurat, dapat diuji kebenarannya dan lengkap, maka dalam proses pengembangan skripsi ini diantaranya sebagai berikut :

a. Identifikasi Masalah

Merupakan tahapan untuk mengidentifikasi masalah yang akan dikaji. Dilakukan studi pustaka yaitu mengumpulkan bahan-bahan referensi baik dari buku, artikel, paper, jurnal, makalah, maupun dari situs internet.

b. Analisa Masalah

Setelah memahami bidang yang akan dibuat sistem pakarnya, selanjutnya pada tahap ini akan dilakukan analisa mengenai jenis-jenis penyakit kulit wajah pada manusia.

c. Membentuk Basis Pengetahuan

Setelah mengetahui struktur pengetahuan, selanjutnya menentukan metode representasi pengetahuan yang digunakan dan merepresentasikan pengetahuan yang didapat.

d. Memilih Teknik Inferensi Pengetahuan

Setelah mengetahui dan memahami masalah serta konsep pengetahuan yang akan dikembangkan dalam sistem pakar, maka tahapan selanjutnya adalah menentukan teknik inferensinya. Teknik inferensi yang akan dipakai dalam pembuatan aplikasi sistem pakar ini adalah teknik pencarian *heuristic* dengan menggunakan topologi penelusuran menggunakan *Depth First Search* dan *backtracking*.

e. Pengujian

Pengujian ini dilakukan untuk mengetahui apakah *software* yang dibuat telah sesuai dengan desainnya dan masih dapat kesalahan atau tidak. Pengujian biasanya dilakukan dalam beberapa tahap yang saling independen.

f. Pemeliharaan

Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

6. Sistematika Penulisan

Sistematika penulisan tugas akhir ini diatur dan disusun dalam 5 bab yang bertujuan memberikan gambaran tentang masalah yang dibahas. Pembahasan masing - masing bab mempunyai keterkaitan antara satu dengan yang lainnya. Penyusunan ini secara ringkas diuraikan dari bab pertama sampai bab akhir yaitu sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan beberapa masalah yang meliputi : latar belakang masalah, masalah yang diangkat, tujuan dan manfaat penulisan, ruang lingkup masalah, metode pengembangan yang diajukan, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini memberikan uraian tentang teori dan konsep yang berkaitan dengan penyusunan tugas akhir ini, serta menjelaskan sistem - sistem yang digunakan dalam perancangan tugas akhir ini, baik yang berkaitan dengan teori sistem pakar, kecerdasan tiruan, komponen - komponennya serta tahapannya. Dan juga menjelaskan perangkat lunak yang digunakan dalam sistem pakar.

BAB III ANALISA MASALAH DAN PERANCANGAN SISTEM PAKAR

Bab ini akan dijelaskan tentang analisa dan perancangan sistem yang akan dibangun, dan pengklasifikasian penyakit kulit wajah berdasarkan gejala - gejala yang dialami oleh pasien. Pada sistem pakar dikembangkan secara bertahap mulai dari permasalahan, strategi pemecahan masalah, aplikasi yang diusulkan serta cara kerjanya, struktur basis data yang digunakan, rancangan layar, *flowchart*, algoritma, spesifikasi *software* dan *hardware* yang digunakan untuk pengembangan program.

BAB IV IMPLEMENTASI DAN EVALUASI PROGRAM

Bab ini menjelaskan tentang cara implementasi dan analisa program serta cara pengoperasian sistem pakar dengan menampilkan tampilan layar serta tampilan program.

BAB V PENUTUP

Bab ini berisi kesimpulan dari penulisan tugas akhir dan harapan yang berupa saran - saran membangun yang diperlukan untuk memperbaiki perancangan sistem dan implementasinya.

BAB II

LANDASAN TEORI

1. Konsep Dasar Kecerdasan Tiruan

Artificial Intelligence merupakan bagian tengah atau inti dari ilmu komputer dan merupakan salah satu ilmu dasar dari ilmu komputer yang harus dipahami, bila kita ingin membuat perangkat lunak dengan penerapan konsep *Artificial Intelligence* dalam memecahkan persoalan yang ada di dunia ini, sehingga keberadaannya mutlak harus ada. Karena kecerdasan tiruan adalah ilmu yang berdasarkan proses manusia berpikir, maka penelitian bagaimana proses manusia berpikir adalah hal yang pokok.

Semua proses berpikir menolong manusia untuk menyelesaikan sesuatu masalah. Pada saat otak manusia mendapat informasi dari luar, maka suatu proses berpikir memberikan petunjuk tindakan atau respon apa yang dilakukan. Hal ini merupakan suatu reaksi otomatis dan respon yang spesifik dicari untuk menyelesaikan masalah tertentu. Hasil akhir dari semua proses berpikir tersebut disebut tujuan (*goal*). Berdasarkan fakta-fakta yang ada keterangan diatas, maka komputer dapat digunakan sebagai alat bantu dalam mengambil keputusan.

2. Sistem Pakar

a. Definisi Sistem Pakar

Menurut Siswanto (2005), sistem pakar atau sistem berbasis pengetahuan adalah yang paling banyak aplikasinya dalam membantu menyelesaikan masalah-masalah dalam dunia nyata. Perangkat lunak ini dapat sekali dijalankan oleh perangkat komputer pribadi, sehingga untuk aplikasi kecerdasan tiruan ini dapat dilakukan dengan mudah dan dengan biaya yang relatif lebih murah. Sistem pakar adalah sebuah program yang dapat :

- 1) Menangani masalah dunia nyata, masalah yang kompleks yang sangat membutuhkan interpretasi pakar.
- 2) Menyelesaikan masalah dengan menggunakan komputer dengan model penalaran manusia dan mencapai kesimpulan yang sama dengan yang dicapai oleh seorang pakar jika berhadapan langsung dengan sebuah masalah yang sedang di hadapinya.

Komputer yang berbasis sistem pakar adalah program komputer yang mempunyai pengetahuan yang berasal dari manusia yang berpengetahuan luas dalam suatu domain tertentu, dimana pengetahuan di sini adalah pengetahuan manusia yang sangat minim penyebarannya, mahal dan serta susah untuk di dapatkannya.

Kondisi-kondisi dimana sistem pakar dapat membantu manusia dalam menyelesaikan masalahnya, antara lain:

- 1) Kebutuhan tenaga ahli yang banyak, tetapi pakar yang tersedia jumlahnya hanya sedikit dan sangat terbatas.
- 2) Pemakaian pakar yang berlebihan dalam membuat keputusan, walaupun dalam suatu tugas yang rutin.
- 3) Pertimbangan kritis harus dilakukan dalam waktu yang singkat untuk menghindari hal-hal yang tidak diinginkan.
- 4) Hasil yang optimal, seperti dalam hal perencanaan atau konfigurasi.
- 5) Sejumlah besar data yang harus diteliti oleh pakar secara berkelanjutan.

b. Pembentukan Umum *Expert System*

Dengan metoda *Forward-Channing*, *data-driven* karena *inference engine* menggunakan informasi yang ditentukan oleh *user* untuk memindahkan keseluruhan jaringan dari logika 'AND' dan 'OR' sampai sebuah terminal di tentukan sebagai obyek.

Bila *inference engine* tidak dapat menentukan obyek maka akan meminta informasi lain. Aturan (*Rule*) di mana menentukan obyek, membentuk lintasan (*Path*) yang mengarah ke obyek. Oleh karna itu, hanya satu cara untuk mencapai obyek adalah memenuhi semua aturan.

c. Ciri-ciri Sistem Pakar

Ciri-ciri dari sistem pakar adalah:

- 1) Terbatasnya pada domain keahlian tertentu.
- 2) Dapat memberikan penalaran data yang tidak pasti.
- 3) Dapat mengemukakan rangkaian alasan-alasan yang tidak diberikannya dengan cara yang tidak dapat di pahami.
- 4) Berdasarkan kaidah-kaidah/ketentuan/rule yang berlaku.
- 5) Dirancang untuk dapat dikembangkan secara bertahap.
- 6) Pengetahuan dan mekanisme penalaran jelas terpisah.
- 7) Keluaran bersifat anjuran.
- 8) Sistem dapat mengaktifkan kaidah secara searah yang sesuai di tuntun oleh dialog dengan *user*.

d. Komponen Sistem Pakar

Menurut Siswanto (2005), komponen sistem pakar terdiri dari 4 komponen yaitu:

- 1) Basis pengetahuan (*Knowledge Base*)
 - a) Inti program sistem pakar
 - b) Merupakan representasi pengetahuan dari seorang pakar.
 - c) Tersusun atas fakta-fakta yang berupa objek dan kaidah/ketentuan yang merupakan informasi tentang cara bagaimana membangkitkan fakta baru dari fakta yang telah diketahui.
- 2) Mesin Inferensi (*Inference Engine*)
 - a) Bagian-bagian yang mengandung mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan seorang pakar.
 - b) Mekanisme ini akan menganalisa masalah tertentu dan selanjutnya mencari sebuah jawaban atau kesimpulan yang yang terbaik.
 - c) Memilih pengetahuan yang relevan dalam rangka mencapai suatu kesimpulan.
 - d) Memulai pelacakannya dengan mencocokkan kaidahnya (*rule*) dalam basis pengetahuan dengan fakta-fakta yang ada dalam daftar fakta-fakta (*fact list*) yang di simpan dalam basis pengetahuan di *hardisk*.
- 3) *User Interface*
 - a) Bagian penghubung antara sistem pakar dengan pemakai.
 - b) Akan terjadi dialog antara program dan pemakai.
 - c) Program akan mengajukan pertanyaan-pertanyaan dengan berbentuk panduan menu (*menu driven*), pernyataan-pernyataan bahasa alami (*natural language*), dan *graphics interface style*. Program sistem pakar akan mengambil kesimpulan berdasarkan jawaban-jawaban dari si pemakai tadi.

4) *Development Engine*

Bagian dari sistem pakar sebagai fasilitas untuk mengembangkan mesin inferensi dan penambahan basis pengetahuan yang akan dilakukan *knowledge engineer* atau harus punya keahlian dalam mengerti bagaimana pakar menerapkan pengetahuan mereka dalam memecahkan suatu masalah, dan juga mampu mengekstrasikan penjelasan (*knowledge acquisition*) mengenai pengetahuan dari pakar, bila si pakar menemukan pengetahuan dan aturan-aturan yang baru dari pengalaman ia bekerja.

5) Teknik Penalaran (*interface*)

Ada dua teknik dalam melakukan penalaran (*interface*):

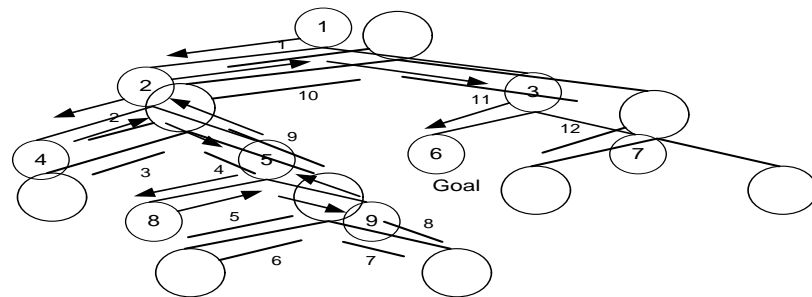
- Pelacakan ke belakang (*backward chaining*) yang memulai penalarannya dari kesimpulan (*goal*), dengan mencari sekumpulan hipotesa-hipotesa yang mendukung menuju fakta-fakta yang mendukung sekumpulan hipotesa-hipotesa tersebut.
- Pelacakan ke depan (*forward chaining*) memulai dari sekumpulan fakta-fakta dengan mencari kaidah yang cocok dengan dugaan/hipotesa yang ada menuju kesimpulan.

6) Teknik Penelusuran (*Searching*)

Kedua teknik penalaran dipengaruhi oleh tiga macam teknik penelusuran (*searching*):

a) Teknik *Depth-First Search*

Adalah teknik penelusuran data pada *node-node* secara *vertical* dan sudah terdefinisi, misalnya kiri ke kanan, keuntungan pencarian dengan teknik ini adalah bahwa penelusuran masalah dapat di gali secara mendalam sampai di temukannya kapasitas suatu solusi yang optimal. Kekurangan teknik penyelesaian ini adalah membutuhkan waktu yang sangat lama untuk ruang lingkup masalah yang besar.

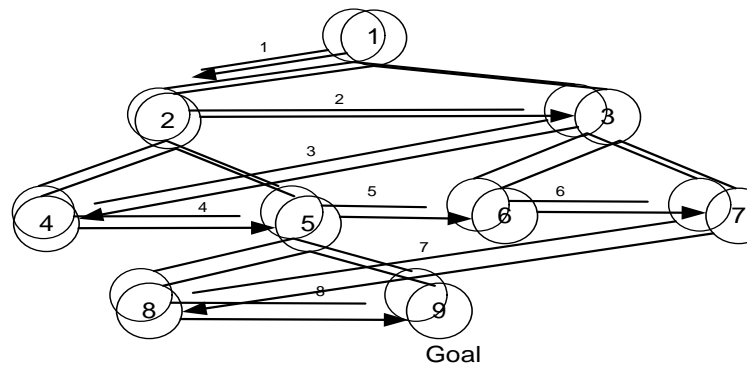


Gambar 2.1 : Teknik *Depth-First Search*

b) Teknik *Breadth-First Search*

Adalah teknik penelusuran data pada semua *node* dalam satu level atau salah satu tingkatan sebelum ke *level* atau tingkatan di bawahnya. Keuntungan pencarian dengan teknik ini adalah sama dengan *depth first search*, hanya saja penelusuran dengan teknik ini mempunyai nilai tambah, dimana semua *node* akan di cek secara menyeluruh pada setiap tingkatan *node*. Kekurangan teknik penelusuran ini terletak pada waktu yang dibutuhkan yang sangat lama apabila solusi berada dalam posisi *node* terakhir sehingga menjadi tidak efisien.

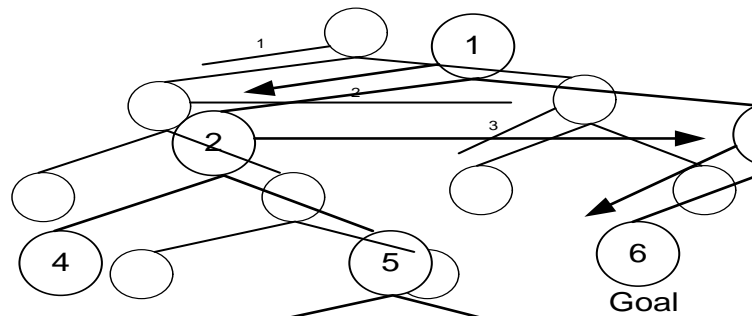
Kekurangan dalam implementasi juga perlu di pertimbangkan, misalnya teknik penelusuran menjadi tidak interaktif antara suatu topik dengan topik yang lain atau harus melompat dari satu topik ke topik yang lain sebelum topik tersebut selesai di telusuri.



Gambar 2.2 : Teknik *Breadth-First Search*

c) Teknik *Best-First Search*

Adalah teknik penelusuran yang menggunakan pengetahuan akan suatu masalah untuk melakukan panduan pencarian ke arah node tempat dimana solusi berada. Pencarian jenis ini dikenal juga sebagai *heuristic*. Pendekatan yang dilakukan adalah mencari solusi yang terbaik berdasarkan pengetahuan yang dimiliki sehingga penelusuran dapat ditentukan harus di mulai dari mana dan bagaimana menggunakan proses terbaik untuk mencari solusi. Keuntungan jenis pencarian ini adalah mengurangi beban komputasi karena hanya solusi yang memberikan harapan saja yang diuji dan akan berhenti apabila solusi sudah mendekati yang terbaik. Ini merupakan model yang menyerupai cara manusia mengambil solusi yang dihasilkan merupakan solusi yang mutlak benar.



Gambar 2.3 : Teknik *Best-First Search*

e. Tipe-tipe Sistem Pakar

Menurut Siswanto (2010), tipe-tipe sistem pakar berdasarkan struktur program, ada 3 tipe yaitu:

- 1) Program mandiri
Sistem pakar yang murni dan berdiri sendiri, artinya program utamanya tanpa mengandung *subroutine* yang memakai algoritma konvensional.
- 2) Program terkait
Sistem pakar yang dikelilingi program lainnya, artinya sebuah *subroutine* yang akan dipanggil oleh program utama.
- 3) Program terhubung
Sistem pakar yang merupakan program yang dapat berhubungan dengan paket program lainnya.

f. Klasifikasi Sistem Pakar

- 1) Sistem Pakar Diagnosis
Digunakan untuk melakukan pencarian penyelesaian masalah atau kerusakan suatu alat atau mesin tertentu. Prinsipnya adalah menemukan masalah atau kerusakan apa yang terjadi. Sistem ini merupakan jenis sistem pakar yang paling populer saat ini, melakukan diagnosis, menggunakan deskripsi keadaan karakteristik tingkah laku, atau pengetahuan tentang pembuatan komponen sehingga dapat menentukan kemungkinan kerusakan pada sistem.
Contoh: Diagnosis medis kedokteran, diagnosis kerusakan alat elektronik dan perancangan sistem komunikasi dan radio.
- 2) Sistem Pakar Pengajaran
Digunakan untuk mengajar, mulai dari murid Sekolah Dasar sampai mahasiswa Perguruan Tinggi. Kelebihan dari sistem pakar yang digunakan untuk mengajar adalah membuat diagnosa penyebab kekurangan dari seorang siswa, kemudian memberi cara untuk memecahkannya.
Contoh: Contoh sistem pakar untuk pengajaran bahasa.
- 3) Sistem Pakar Interpretasi
Digunakan untuk menganalisa data yang tidak lengkap, tidak teratur dan data yang kontradiktif, misalnya untuk interpretasi pengawasan, pengertian bahasa, analisis citra, dan lain-lain.
Contoh: sistem pakar interpretasi lingkungan pengendapan delta.
- 4) Sistem Pakar Prediksi
Digunakan untuk melakukan peramalan suatu keadaan dengan ditunjang oleh data yang diperoleh sebelumnya, sistem ini dapat memberikan kemungkinan penyelesaian tertentu.
Contoh: Prediksi Penyebab dan Solusi Ketidaknyamanan Kerja dengan Aplikasi Sistem Pakar.
- 5) Sistem Pakar Perencanaan
Merupakan suatu sistem yang sangat luas mulai dari perencanaan mesin-mesin sampai manajemen bisnis. Penggunaan sistem pakar jenis ini menghemat biaya, waktu, dan material, sebab pembuatan model sudah tidak diperlukan lagi.
Contoh: Perencanaan Sistem Distribusi Menggunakan Sistem Pakar.
- 6) Sistem Pakar Kontrol
Digunakan untuk mengendalikan kegiatan yang membutuhkan presisi waktu yang tinggi. Sistem ini memperhatikan tingkah laku sistem yang dapat disebut normal atau tidak normal. Sistem ini bergantung pada waktu untuk menginterpretasikan tingkah laku yang diamati. Misalnya pada pengendali manajemen bisnis dan pengendali pada industri berteknologi tinggi.

Contoh: Aplikasi Sistem Pakar Pada Sistem Kontrol Temperatur Ruang.

g. Kemampuan Tambahan Yang Diperlukan Sistem Pakar

Untuk lebih meningkatkan kemampuan *expert system*, diperlukan komponen-komponen tambahan:

- 1) Fasilitas penjelasan
 - a) Untuk menjelaskan bagaimana prosesnya sampai kesimpulan-kesimpulan tersebut di peroleh.
 - b) Dengan cara memperlihatkan *rule-rule* yang digunakan.
- 2) Kemudahan memodifikasi
 - a) Di karenakan ilmu pengetahuan berkembang.
 - b) kemampuan seorang pakar bertambah terus.
- 3) Kompabilitas
Dapat dijalankan pada berbagai jenis komputer.
- 4) Kemampuan belajar
Kemampuan selama *expert system* untuk menambah sendiri pengolahannya, selama interaksi dengan pemakainya.

h. Keuntungan Sistem Pakar

Secara garis besar, banyak manfaat yang dapat di ambil dengan adanya sistem pakar, antara lain:

- 1) Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
- 2) Bisa melakukan proses secara berulang-ulang secara otomatis.
- 3) Menyiapkan pengetahuan dan keahlian para pakar.
- 4) Meningkatkan *output* dan produktifitas.
- 5) Meningkatkan kualitas.
- 6) Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
- 7) Mampu beroperasi dalam lingkungan yang berbahaya.
- 8) Memiliki kemampuan untuk mengakses pengetahuan
- 9) Memiliki reliabilitas.
- 10) Meningkatkan kapabilitas sistem komputer.
- 11) Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dengan mengandung ketidakpastian.
- 12) Sebagai media pelengkap dalam pelatihan.
- 13) Meningkatkan kapabilitas dalam penyelesaian masalah.
- 14) Menghemat waktu dalam pengambilan suatu keputusan.

i. Kerugian sistem pakar

- 1) Penelitian yang dilakukan untuk aplikasi program sistem pakar membutuhkan biaya yang tinggi.
- 2) Dalam pembuatan dan pengembangan program sistem pakar memakan waktu yang lama.
- 3) Program sistem pakar yang dibuat tidak selamanya benar dan tetap sempurna mungkin saja ada kesalahan pada mesin inferensi. (Suparman,1991)

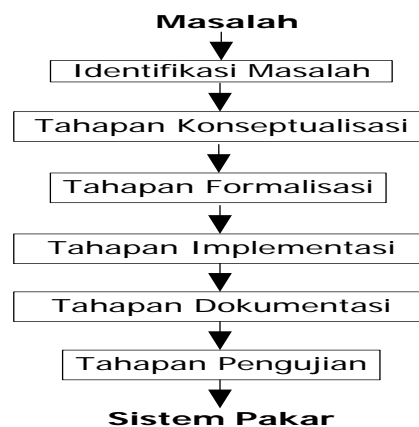
j. Struktur Sistem Pakar

Sistem pakar terdiri dari 2 bagian kelompok, yaitu: lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation*)

environment). Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi.

k. Tahapan Pembuatan Sistem Pakar

Proses dalam pembuatan sebuah program sistem pakar melibatkan beberapa unsur, unsur yang paling berinteraksi yaitu perekayasa pengetahuan (*Knowledge engineer*), pakar pada bidang keahlian (*domain expert*), dan pemakai akhir atau pemakai sistem pakar yang diinginkan untuk dibuat (*end user*). Tentunya dengan melalui proses dan langkah tahapan dari sistem pakar itu sendiri.



Gambar 2.4 : Tahapan Pembuatan Sistem Pakar

1) Identifikasi Masalah

Identifikasi masalah merupakan tahapan mengidentifikasi permasalahan yang akan dikaji, dalam hal ini adalah batasan masalah yang akan dikaji, menentukan pakar yang terlibat, dan sumberdaya yang diperlukan (seperti waktu, biaya dan fasilitas komputerisasi) dan tujuan yang akan dicapai. Sebaiknya permasalahan yang akan dikaji bersifat sempit tapi kompleks (mendalam). Pada tahapan ini pengembangan pengetahuan dapat menentukan masalah secara umum dan memilih pakar terlebih dahulu, kemudian mendiskusikannya dengan pakar tersebut untuk menentukan batasan masalah yang akan dikaji.

2) Tahapan Konseptualisasi

Konseptualisasi merupakan tahapan dimana pakar menentukan konsep yang akan dikembangkan menjadi sistem pakar dengan bantuan pengetahuan yang sudah ada. Dari konsep-konsep tersebut akan dirinci seluruh kegiatan-kegiatan apa saja yang akan dilakukan dan mengkaji hal-hal yang berhubungan dalam pembuatan sistem pakar itu sendiri, serta menyusun mekanisme pengendalian yang diperlukan untuk mencapai suatu solusi.

3) Tahapan Formalisasi

Pada tahapan formalisasi dilakukan pemilihan teknik yang tepat dan benar untuk digunakan oleh perancang sistem pakar di dalam pengembangan sistem pakar tersebut di antaranya adalah struktur data, cara atau teknik inferensi dan

mempresentasikan pengetahuan, strategi, kontrol serta peralatan pengembangan yang digunakan.

4) Tahapan Implementasi

Pada tahapan ini dilakukan penterjemahan hasil formalisasi kedalam bentuk format program, yang disesuaikan dengan peralatan pengembangan sistem pakar.

5) Tahapan Dokumentasi

Tahapan dokumentasi adalah tahapan akhir dimana program sistem pakar yang telah dibangun harus didokumentasikan, karena dengan didokumentasikan sebuah program sistem pakar tersebut menjadi lebih baik, maka apabila suatu saat di perlukan atau terjadi permasalahan pada programnya akan mudah untuk mencarinya.

6) Tahapan Pengujian

Pada tahapan ini sistem pakar yang telah dibuat kemudian diuji coba dengan para pakarnya untuk mencari kemungkinan kesalahan yang bisa terjadi dan melakukan perbaikan apabila diperlukan. Sehingga hasil dari sistem pakar tersebut benar-benar sesuai dengan keinginan, yaitu sebuah sistem pakar yang lengkap dan akurat serta dapat memberikan solusi terhadap masalah yang dihadapi sistem pakar tersebut.

3. Teori Pemrograman

a. Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java2 adalah generasi kedua dari Java *platform* (generasi awalnya adalah *Java Development Kit*). Java berdiri atas sebuah *interpreter* yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam *file* .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa Java disebut sebagai bahasa pemrograman yang *portable* karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Platform Java memiliki tiga buah edisi yang berbeda, yaitu :

1) *Java Platform, Enterprise Edition* (Java EE)

Java EE yaitu kelompok dari beberapa API dari Java. Java EE juga dibuat untuk membuat aplikasi yang kompleks / teknologi yang berjalan di *server*, tapi faktanya Java EE juga mencakup teknologi yang dapat digunakan disemua lapisan dari sebuah sistem informasi. Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada lingkungan *enterprise*, dengan menambahkan fungsi-fungsi Java seperti EJB (*Enterprise Java Bean*), Java COBRA, *Servlet* dan JSP, serta Java XML (*Extensible Markup Language*).

2) *Java Platform, Standard Edition* (Java SE)

Java SE Adalah inti dari bahasa pemrograman Java. JDK (*Java Development Kit*) adalah salah satu tools dari Java SE mengompilasi dan menjalankan program Java. Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada level PC (*Personal Computer*).

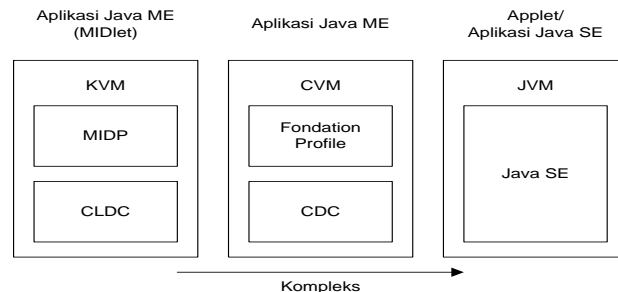
3) *Java Platform, Micro Edition (Java ME)*

Java ME adalah lingkungan pengembangan yang didisain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Kategori ini digunakan untuk menjalankan dan mengembangkan aplikasi-aplikasi Java pada *handheld device* atau perangkat-perangkat elektronik beserta pendukungnya, seperti *handphone*, *pager*, PDA, dan pocket PC.

Namun dengan beberapa contoh perangkat yang disebutkan, bukan berarti sebuah program yang dikembangkan untuk perangkat tertentu dapat dipakai oleh perangkat lainnya dengan baik. Hal ini memang menjadi sebuah masalah mengenai standar kinerja dan portabilitas Java ME. Sehingga para pengembang Java ME menspesifikasikan beberapa arsitektur yang terbagi atas konfigurasi, profil dan paket opsi yang bertujuan untuk mencapai kinerja dengan memanfaatkan kelebihan piranti yang digunakan.

Konfigurasi merupakan *Java library* minimum dan kapabilitas yang dimiliki oleh para pengembang Java ME, yang maksudnya sebuah *mobile device* dengan kemampuan Java akan dioptimalkan untuk menjadi sesuai. Sedangkan profil membahas sesuatu yang spesifik untuk sebuah perangkat.

Pada Java ME terdapat dua buah profil yaitu MIDP dan *Foundation Profile*. Keterhubungan antara konfigurasi dan profil yang ada pada Java ME beserta jenis mesin virtualnya dapat dilihat pada gambar berikut :



Gambar 2.5 : Hubungan Java ME dan Java SE

Dari gambar diatas, dapat diketahui bahwa Java ME memiliki serangkaian profil yang masing-masing memiliki konfigurasi yang berbeda pula. Di bawah ini akan dijelaskan mengenai masing-masing konfigurasi tersebut.

1) *Connected Limited Device Configuration (CLDC)*

CLDC atau Connected Limited Device Configuration adalah perangkat dasar dari J2ME, spesifikasi dasar yang berupa library atau API yang diimplementasikan pada J2ME, seperti yang digunakan pada telepon selular, pager, dan PDA. Perangkat tsb dibatasi dengan keterbatasan memory, sumber daya, dan kemampuan memproses. Spesifikasi CLDC pada J2ME adalah spesifikasi minimal pada package, class, dan sebagian fungsi Java Virtual Machine yang dikurangi agar dapat diimplementasikan dengan keterbatasan sumber daya pada alat tersebut, JVM yang digunakan adalah KVM (*Kilobyte Virtual Machine*).

2) *Mobile Information Device Profile (MIDP)*

MIDP atau *Mobile Information Device Profile* adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan diatas CLDC, API tambahan untuk daur hidup aplikasi, antar muka, jaringan, dan penyimpanan persisten. Pada saat ini terdapat MIDP 1.0 dan MIDP 2.0. Fitur tambahan pada MIDP 2.0 dibandingkan MIDP 1.0 adalah API untuk multimedia. Pada MIDP 2.0 terdapat dukungan memainkan tone, tone sequence, dan file WAV walaupun tanpa adanya *Mobile Media API* (MMAPI).

3) MIDlet

MIDlet adalah aplikasi yang ditulis untuk MIDP. Aplikasi MIDlet adalah bagian dari class `javax.microedition.midlet.MIDlet` yang didefinisikan pada MIDP. MIDlet berupa sebuah class abstrak yang merupakan subclass dari bentuk dasar aplikasi sehingga antarmuka antara aplikasi J2ME dan aplikasi manajemen pada perangkat bisa terbentuk.

4) Daur hidup MIDlet

MIDlet terdiri dari beberapa method yang harus ada, yaitu `constructor()`, `protected void startApp() throws MIDletStateChangeException`, `protected void pauseApp()`, `protected void destroyApp (Boolean unconditional) throws MIDletStateChangeException`

Ketika MIDlet dijalankan maka akan diinisialisasi dengan kondisi pause dan dijalankan `pauseApp()`, kondisi berikutnya adalah fungsi MIDlet dijalankan, yaitu pada `startApp()`. Metode yang ada tersebut diimplementasikan sebagai `protected`, hal ini dimaksudkan agar MIDlet lain tidak dapat memanggil method tersebut. Pada saat pemakai keluar dari MIDlet, maka method `destroyApp()` akan dijalankan sebelum MIDlet benar-benar tidak berjalan lagi. Method `notifyDestroyed()` akan dipanggil sebelum MIDlet benar-benar tidak berjalan lagi, `DestroyApp()` akan memanggil `notifyDestroyed()`, dan `notifyDestroyed()` akan memberi tahu platform untuk menterminasi MIDlet dan membersihkan semua sumber daya yang mengacu pada MIDlet.

b. *Personal Home Page (PHP)*

Menurut Bunafit Nugroho dalam *PHP dan MYSQL dengan Editor Dreamweaver MX*, Andi Offset Yogyakarta, 2004, hal. 141, menjelaskan bahwa PHP bermula saat Rasmus Lerdof membuat sejumlah skrip *Perl* untuk membuat *resumenya*, yang dapat mengamati siapa saja yang melihat daftar riwayat hidupnya, yakni pada tahun 1994, skrip-skrip ini selanjutnya dikemas menjadi *tool* yang disebut dengan "*Personal Home Page*", paket inilah yang menjadi cikal bakal dari PHP. Pada tahun 1995, Rasmus menciptakan *tool* yang lain untuk mengambil suatu inputan dari *HTML form: FI. Form Interpreter* juga dikenal sebagai PPHP/FI atau PHP versi 2, pada versi inilah pemrogram dapat menempelkan kode-kode terstruktur pada *Tag HTML* dan PHP juga biasa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks sambil berjalan. Konsep kerja PHP diawali dengan permintaan suatu halaman oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan halaman internet, *browser* mendapatkan alamat dari web server, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode html) ke web server, selanjutnya *web server* menyampaikannya ke klien atau si pengguna.

c. Database

Dalam penulisan skripsi ini *database* yang digunakan adalah *MYSQL*, karena *php* dan *mysql* merupakan kombinasi piranti (*tools*) yang banyak dan umum digunakan dalam pembangunan dan pengembangan situs web berbasis *database* dalam sajian informasinya. Berikut ini merupakan fungsi-fungsi *mysql*:

- 1) Fungsi *mysql_connect*: digunakan untuk membuat hubungan ke *database mysql* yang terdapat pada suatu *host*. Bentuk pemanggilannya:
mysql_connect (host, nama_pemakai, password)
- 2) Fungsi *mysql_select_db* : digunakan untuk memilih *database*.
format pemanggilannya:
mysql_select_db (database, pengenal hubungan)
- 3) Fungsi *mysql_fetch_query()*: digunakan untuk mengeksekusi permintaan terhadap sebuah table atau sejumlah tabel. Bentuk pemanggilannya:
mysql_query (permintaan, pengenal hubungan)
- 4) Fungsi *mysql_fetch_array()*: digunakan untuk mengambil hasil *query* baris perbaris ke *database* dalam bentuk *array assosiatif*. Bentuk pemanggilannya:
mysql_fetch_array (pengenal_hasil)
- 5) Fungsi *mysql_fetch_row()*: digunakan untuk mengambil hasil *query* baris perbaris ke *database* dalam bentuk *array*. Bentuk pemanggilannya:
mysql_fetch_row (pengenal_hasil)

Di bawah ini merupakan *skrip* yang dipakai atau yang digunakan untuk pengoperasian isi *database*:

- 1) Untuk mengatur tampilan atau keluaran sesuai yang kita inginkan, yaitu:
*SELECT (Field | *) FROM nama_tabel [WHERE kondisi]*
- 2) Untuk menambah data ke *database*, bentuk *skripnya*:
INSERT INTO nama_tabel (field1, field2,...) VALUES ('data1', 'data2',,...)
- 3) Untuk mengedit data digunakan *skrip*:
UPDATE nama_tabel SET field1 = nilai_baru, field2 = nilai_baru,... WHERE Syarat1, syarat2,...
- 4) Untuk menghapus data digunakan *skrip*:
DELETE FROM nama_tabel WHERE syarat1, syarat2,...

4. Penyakit Kulit Wajah

Penyakit kulit wajah merupakan suatu penyakit yang menyerang pada permukaan kulit wajah dan disebabkan oleh berbagai macam penyebab. Penyakit kulit wajah mempunyai macam-macam yang akan menunjukkan varians dalam gejala dan keparahan dan mungkin menampilkan beberapa karakteristik yang unik. Jenis penyakit kulit wajah dapat berkisar dari hampir tidak terlihat hingga mengancam kehidupan. Namun jenis penyakit kulit wajah akan membantu menentukan kemungkinan penyebab dan perawatan terbaik (Keen Achroni : 2012).

Beberapa makhluk hidup dapat menyebabkan penyakit kulit wajah. Penyakit kulit wajah dapat diakibatkan oleh makhluk hidup seperti bakteri, virus maupun jamur. Bakteri, virus dan jamur menginfeksi kulit wajah dan dapat merusak kulit wajah tetapi tidak pernah sampai mematikan.

BAB III

ANALISA MASALAH DAN RANCANGAN SISTEM

1. Analisa Masalah

Permasalahan yang akan dibahas dalam tugas akhir ini adalah membuat suatu sistem yang dapat memiliki kepastian berdasarkan data yang didapat dari beberapa buku. Untuk mendiagnosa penyakit kulit wajah perlu melakukan konsultasi ke dokter kesehatan kulit wajah atau klinik perawatan kulit wajah, namun dengan cara ini tidak semua orang memiliki banyak waktu dan biaya yang cukup. Oleh karena itu perlu dibuatkan suatu program komputer berupa sistem pakar untuk membantu masyarakat agar dapat dengan mudah mengetahui penyakit kulit wajah yang diderita. Penerapan sistem pakar dalam penyakit kulit wajah meliputi pengumpulan data gejala, penyakit, dan pengobatan dalam permasalahannya. Untuk penyakit kulit wajah ini dibutuhkan sistem pakar yang dapat digunakan tidak hanya terpaku pada suatu tempat seperti komputer, tetapi lebih dibutuhkan perangkat yang sifatnya *mobile* agar bisa digunakan dimana saja.

2. Penyelesaian Masalah

Dari permasalahan yang telah diuraikan diatas, maka penulis mencari teknologi pemrograman yang aplikasinya memiliki kemampuan untuk dapat diakses dimana saja dan dapat dijalankan ditelepon selular atau *handphone*. Aplikasi Java ME dirasa tepat untuk memecahkan masalah tersebut.

3. Rancangan Sistem Pakar

Sistem pakar untuk mendiagnosa penyakit kulit wajah yang diusulkan ialah berbasis *mobile* agar dapat memudahkan penerimaan informasi secara cepat dan dapat segera mengambil tindakan terhadap penyakit yang muncul pada kulit wajah.

Sistem ini memiliki beberapa proses sebelum mencapai sebuah keputusan yaitu proses pertanyaan dan proses memilih jawaban yang disediakan:

a. Mesin Inferensi

Pada komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menentukan juga menyelesaikan suatu penyakit. Mesin inferensi merupakan bagian dari sistem pakar yang melakukan penalaran mengenai informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk merumuskan kesimpulan.

Dalam pembuatan sistem ini teknik inferensi yang digunakan adalah pelacakan dan pencarian. Teknik yang digunakan adalah teknik pelacakan ke depan atau *forward chaining*, sedangkan untuk pencarian keputusan dari permasalahan yang ada adalah dengan menggunakan teknik *best-first search*. *Best-first search* adalah teknik penggabungan metode *depth-first search* dan *breadth-first search*.

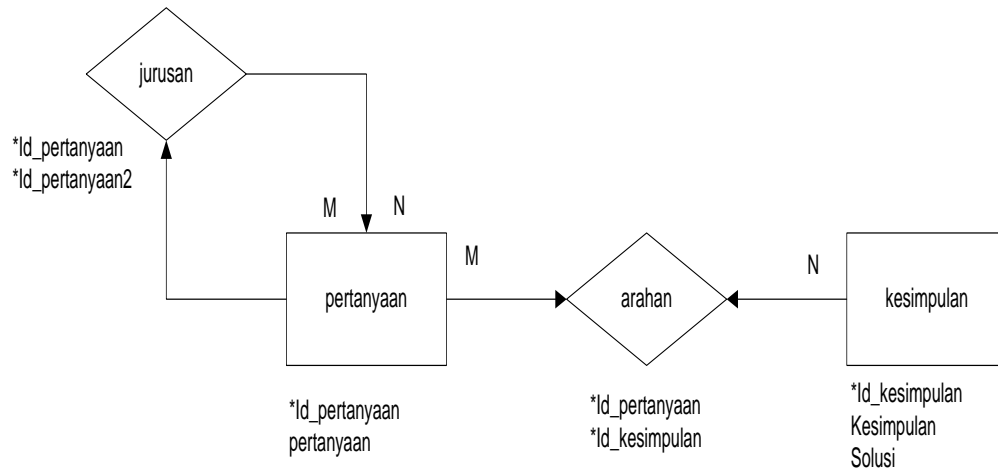
b. Pohon Keputusan

Pengambilan keputusan dari sistem pakar akan lebih mudah apabila dibentuknya sebuah pohon keputusan, dimana pohon keputusan merupakan gambaran sederhana mengenai permasalahan dan kesimpulan. Pohon keputusan dan *rule*-nya dapat dilihat pada lampiran 1 dan 2.

4. Struktur Basis Data

Seluruh tabel yang dibutuhkan disimpan didalam *database* yang diberi nama *db_wajah*. *Database* ini berfungsi untuk menampung pertanyaan yang dibutuhkan. Proses identifikasi ini pada sistem pakar dapat dimodifikasi seperti penambahan, pengurangan ataupun pengubahan sesuai dengan kebutuhan.

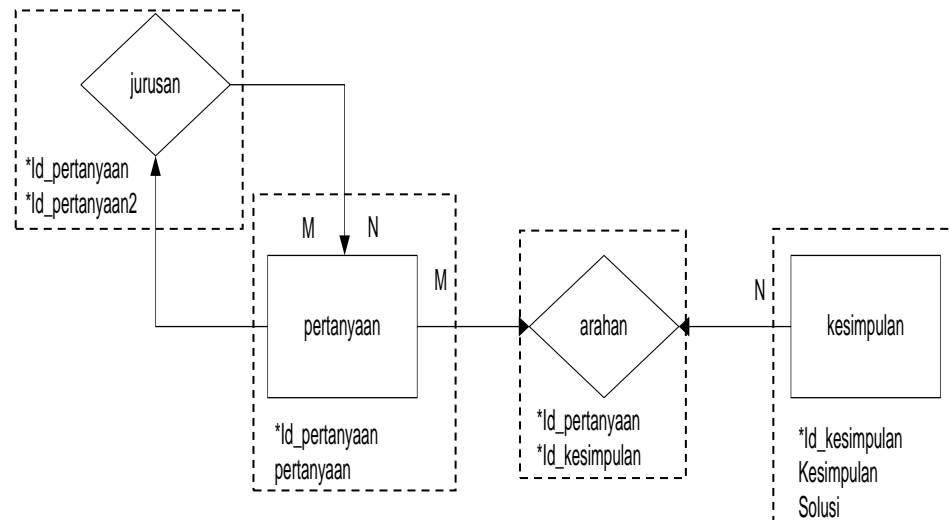
a. ERD (*Entity Relationship Diagram*)



Gambar 3.1 : ERD (*Entity Relationship Diagram*)

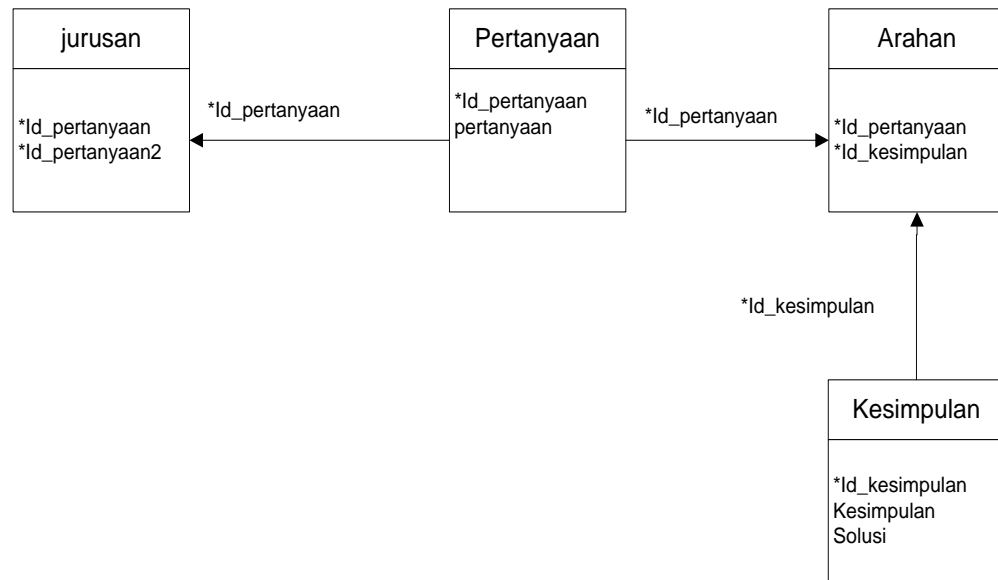
b. Transformasi ERD ke *Logical Record Structure*

Bentuk ERD yang ditransformasikan menjadi LRS (*Logical Record Structure*) adalah sebagai berikut :



Gambar 3.2 : Transformasi ERD ke *Logical Record Structure*

c. *Logical Record Structure*



Gambar 3.3 : Diagram *Logical Record Structure*

5. **Rancangan Database**

Pada program sistem pakar ini, membutuhkan *database* yang terdiri dari 6 buah tabel yaitu tabel pertanyaan, tabel kesimpulan, tabel admin, tabel kamus bahasa, tabel arahan dan tabel jurusan. *Database* yang digunakan bersifat dinamis, yaitu terdiri dari penambahan data, pengubahan data serta penghapusan data tanpa mengubah program.

a. **Tabel Pertanyaan**

Tabel pertanyaan menyimpan data tentang semua pertanyaan yang berkaitan dengan gejala penyakit kulit wajah. Tabel pertanyaan ini terdiri dari 2 (dua) *field*, yaitu Id_pertanyaan dan pertanyaan. Tabel pertanyaan akan diperlihatkan pada tabel 3.1.

Tabel 3.1 : Tabel Pertanyaan

Field	Type	Length	Keterangan
*Id_pertanyaan	Varchar	4	Id Pertanyaan
pertanyaan	Varchar	255	Isi Pertanyaan

b. Tabel Kesimpulan

Tabel kesimpulan berisi data kesimpulan yang terdiri dari 3 (tiga) *field*, yaitu *Id_kesimpulan*, *kesimpulan* dan *solusi*. Tabel kesimpulan akan diperlihatkan pada tabel 3.2.

Tabel 3.2 : Tabel Kesimpulan

Field	Type	Length	Keterangan
*Id_kesimpulan	Varchar	4	Id kesimpulan
kesimpulan	Varchar	50	Isi kesimpulan
Solusi	Varchar	255	Solusi

c. Tabel Admin

Tabel admin berisi data *user* admin yang terdiri dari 2 (dua) *field*, yaitu *username* dan *password*. Tabel admin akan diperlihatkan pada tabel 3.3.

Tabel 3.3 : Tabel Admin

Field	Type	Length	Keterangan
Username	Varchar	20	Nama admin
password	Varchar	20	Password

d. Tabel Kamus Bahasa

Tabel Kamus Bahasa berisi data nama penyakit yang masih terdengar asing yaitu terdiri dari 2 (dua) *field*, yaitu *kata* dan *keterangan*. Tabel Kamus Bahasa ada pada tabel 3.4.

Tabel 3.4 Tabel Kamus Bahasa

Field	Type	Length	Keterangan
kata	Varchar	50	Kata
keterangan	Varchar	255	Keterangan Kata

e. Tabel Arahan

Tabel arahan berisi data arahan yang terdiri dari 2 (dua) *field*, yaitu *Id_pertanyaan* dan *Id_kesimpulan*. Tabel arahan akan diperlihatkan pada tabel 3.5.

Tabel 3.5 Tabel Arah

Field	Type	Length	Keterangan
*Id_pertanyaan	Varchar	4	Id Pertanyaan
*Id_kesimpulan	Varchar	4	Id Kesimpulan

f. Tabel Jurusan

Tabel jurusan berisi data jurusan setiap *rule* yang terdiri dari 2 (dua) *field*, yaitu Id_pertanyaan dan Id_pertanyaan2. Tabel jurusan akan diperlihatkan pada tabel 3.6.

Tabel 3.6 Tabel jurusan

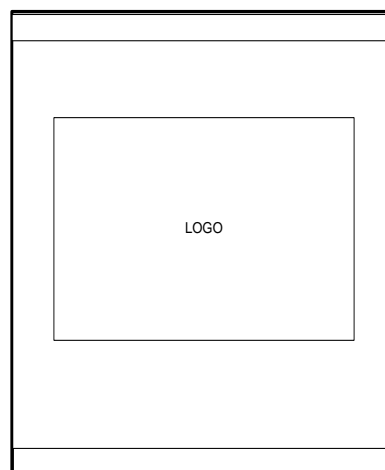
Field	Type	Length	Keterangan
*Id_pertanyaan	Varchar	4	Id pertanyaan
*Id_pertanyaan2	Varchar	4	Id pertanyaan2

6. Rancangan Layar

Agar suatu sistem mudah digunakan, maka diperlukasn *user interface* yang dapat dengan mudah dimengerti oleh *user*. Untuk menghasilkan *user interface* yang mudah dimengerti dan dipahami oleh *user* maka diperlukan rancangan layar sebelum diimplementasikan dalam bentuk program.

a. Rancangan Layar *Splashscreen*

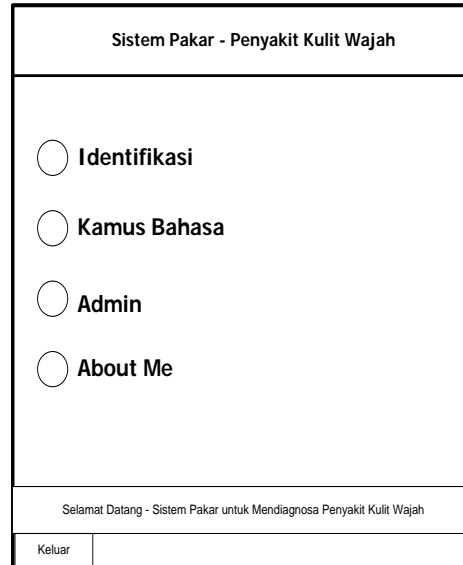
Gambar 3.4 berikut ini merupakan rancangan layar *Splashscreen*. *Splashscreen* ini adalah tampilan pertama kali aplikasi ini dijalankan. Tampilan ini berisi logo atau gambar.



Gambar 3.4 : Rancangan layar *Splashscreen*

b. Rancangan Layar Menu Utama *Mobile*

Gambar 3.5 berikut ini merupakan rancangan layar Menu Utama *Mobile*. Pada layar ini terdapat beberapa pilihan *radio button*, yaitu : Identifikasi, Kamus Bahasa, Admin dan *About Me*. *Button* Identifikasi untuk menuju halaman Identifikasi, *button* Kamus Bahasa untuk menuju halaman Kamus Bahasa, *button* Admin untuk menuju halaman admin, dan *button* About Me untuk menuju halaman tentang pembuat program.

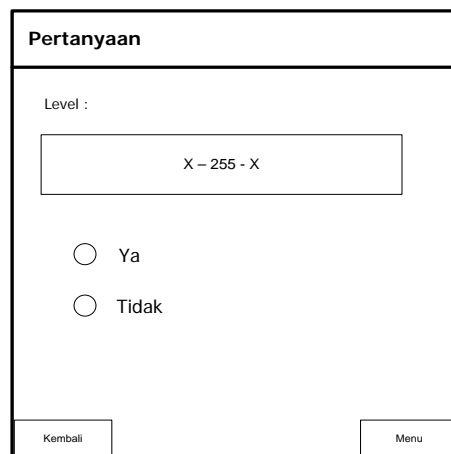


The image shows a mobile application menu screen. At the top, there is a header bar with the text "Sistem Pakar - Penyakit Kulit Wajah". Below the header, there are four radio button options listed vertically: "Identifikasi", "Kamus Bahasa", "Admin", and "About Me". At the bottom of the screen, there is a footer bar containing the text "Selamat Datang - Sistem Pakar untuk Mendiagnosa Penyakit Kulit Wajah" and a "Keluar" button.

Gambar 3.5 : Rancangan Layar Menu Utama

c. Rancangan Layar Pertanyaan

Gambar 3.6 berikut ini merupakan rancangan layar Pertanyaan. Pada layar ini menampilkan pertanyaan dalam proses penentuan. *User* dapat memilih pilihan "Ya" atau "Tidak" untuk menjawab pertanyaan tersebut.



The image shows a mobile application question screen. At the top, there is a header bar with the text "Pertanyaan". Below the header, there is a label "Level :" followed by a text box containing the expression $X - 255 - X$. Below the text box, there are two radio button options: "Ya" and "Tidak". At the bottom of the screen, there are two buttons: "Kembali" and "Menu".

Gambar 3.6 : Rancangan Layar Pertanyaan

d. Rancangan Layar Kesimpulan

Gambar 3.7 berikut ini adalah Rancangan Layar Kesimpulan. Pada layar ini menampilkan kesimpulan yaitu hasil identifikasi yang merupakan akhir dari proses Penentuan setelah *user* memilih beberapa pertanyaan.

Hasil Identifikasi

X - 50 - X

Solusi

X - 255 - X

Kembali Menu

Gambar 3.7 : Rancangan Layar Kesimpulan

e. Rancangan Layar Arahan

Gambar 3.8 berikut ini adalah Rancangan Layar Arahan. Pada layar ini menampilkan arahan untuk pertanyaan berupa kemungkinan kesimpulan karena *user* tidak menjawab sampai pertanyaan terakhir.

Arahan

X - 255 - X

X - 255 - X

X - 255 - X

X - 255 - X

Kembali Menu

Gambar 3.8 : Rancangan Layar pada Arahan

f. Rancangan Layar Kamus Bahasa

Gambar 3.9 berikut ini adalah Rancangan layar kamus bahasa. Pada layar ini menampilkan kamus bahasa kata seputar nama penyakit kulit wajah yang terdapat dalam *database*.

Kamus Bahasa

X - 255 - X

X - 255 - X

X - 255 - X

Halaman : 999/999

Kembali Menu

Gambar 3.9 : Rancangan Layar Kamus Bahasa

g. Rancangan Layar *Login* Admin

Gambar 3.10 berikut ini adalah rancangan layar *Login* Admin. Pada layar ini menampilkan halaman *login* dan terdapat *button login* untuk *login* ke *form* halaman utama admin.

Log In Admin

Username : X - 20 - X

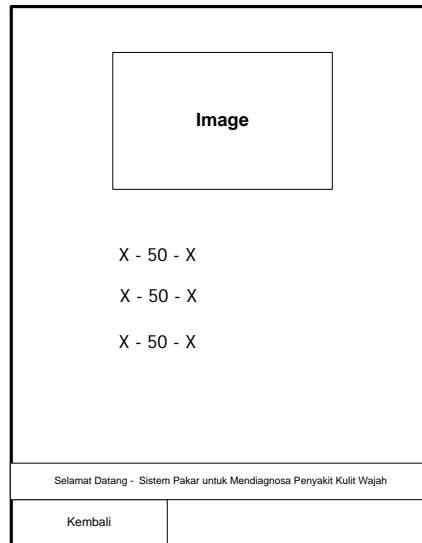
Password : X - 32 - X

Kembali Menu

Gambar 3.10 : Rancangan Layar *Login* Admin

h. Rancangan Layar *About Me*

Gambar 3.11 berikut ini adalah rancangan layar *About Me*. Pada layar ini menampilkan penjelasan singkat pembuat program.

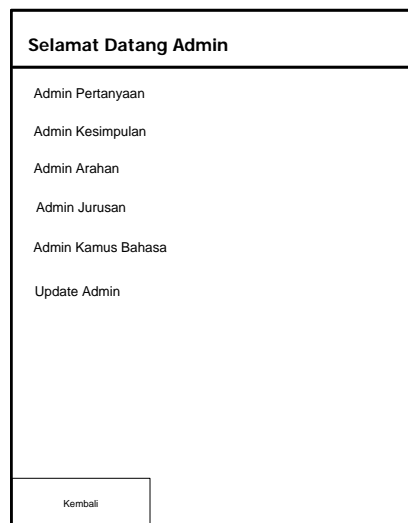


The image shows a wireframe for an 'About Me' page. It features a large rectangular box at the top labeled 'Image'. Below this box, there are three lines of placeholder text, each consisting of 'X' characters separated by hyphens: 'X - 50 - X', 'X - 50 - X', and 'X - 50 - X'. At the bottom of the page, there is a horizontal bar containing the text 'Selamat Datang - Sistem Pakar untuk Mendiagnosa Penyakit Kulit Wajah'. Below this bar is a table with two columns; the left column contains the text 'Kembali'.

Gambar 3.11 : Rancangan Layar *About Me*

i. Rancangan Layar Menu Utama Admin

Gambar 3.12 berikut ini adalah rancangan layar menu utama admin. Pada layar ini menampilkan *form* menu utama admin dan terdapat *link Edit* Pertanyaan untuk masuk ke *form* Admin Pertanyaan, *Edit* Kesimpulan untuk masuk ke *form* Admin Kesimpulan, *Edit* Arahan untuk masuk ke *form* Admin Arahan, *Edit* Jurusan untuk masuk ke *form* Admin Jurusan, *Edit* Kamus Bahasa untuk masuk ke *form* Admin Kamus Bahasa, *Edit* Admin untuk masuk ke *form* *Edit* Admin dan *keluar* untuk keluar dari *form* Menu Utama Admin.



The image shows a wireframe for an 'Admin Main Menu' page. It has a header bar with the text 'Selamat Datang Admin'. Below the header, there is a list of menu items: 'Admin Pertanyaan', 'Admin Kesimpulan', 'Admin Arahan', 'Admin Jurusan', 'Admin Kamus Bahasa', and 'Update Admin'. At the bottom left of the page, there is a button labeled 'Kembali'.

Gambar 3.12 : Rancangan Layar Menu Utama Admin

j. Rancangan Layar Admin Pertanyaan

Gambar 3.13 berikut ini adalah rancangan layar Admin pertanyaan. Pada layar ini menampilkan data pertanyaan dan terdapat *form* untuk menambah, menghapus dan mengubah pertanyaan.

Pertanyaan	
<X - 4 - X>	
<X - 4 - X>	
<X - 4 - X>	
Halaman	999/99
<div> <div>Kembali</div> <div> <div>Menu</div> <div> Tambah Ubah Hapus Berikutnya Sebelumnya </div> </div> </div>	

Gambar 3.13 : Rancangan Layar Admin Pertanyaan

k. Rancangan Layar Admin Kesimpulan

Gambar 3.14 berikut ini adalah rancangan layar admin kesimpulan. Pada layar ini menampilkan data kesimpulan dan terdapat *form* untuk menambah, menghapus dan mengubah kesimpulan itu sendiri.

Kesimpulan	
<X - 4 - X>	
<X - 4 - X>	
<X - 4 - X>	
Halaman	999/99
<div> <div>Kembali</div> <div> <div>Menu</div> <div> Tambah Ubah Hapus Berikutnya Sebelumnya </div> </div> </div>	

Gambar 3.14 : Rancangan Layar Admin Kesimpulan

I. Rancangan Layar Admin Arahkan

Gambar 3.15 berikut ini adalah rancangan layar admin arahan. Pada layar ini menampilkan data arahan dan terdapat *form* untuk menambah, menghapus dan mengubah arahan.

Arahan	
<X - 4 - X>	<X - 4 - X>
<X - 4 - X>	<X - 4 - X>
<X - 4 - X>	<X - 4 - X>
Halaman 999/99	
<div> <div>Menu</div> <div> Tambah Ubah Hapus Berikutnya Sebelumnya </div> </div>	
Kembali	Menu

Gambar 3.15 : Rancangan Layar Admin Arahan

m. Rancangan Layar Admin Jurusan

Gambar 3.16 berikut ini adalah rancangan layar admin jurusan. Pada layar ini menampilkan data jurusan dan terdapat *form* untuk menambah, menghapus dan mengubah jurusan.

Jurusan	
<X - 4 - X>	- <X - 4 - X>
<X - 4 - X>	- <X - 4 - X>
<X - 4 - X>	- <X - 4 - X>
Halaman 999/99	
<div> <div>Menu</div> <div> Tambah Ubah Hapus Berikutnya Sebelumnya </div> </div>	
Kembali	Menu

Gambar 3.16 : Rancangan Layar Admin Jurusan

n. Rancangan Layar Admin Kamus Bahasa

Gambar 3.17 berikut ini adalah rancangan layar admin kamus bahasa. Pada layar ini menampilkan data kamus bahasa dan terdapat *form* untuk menambah, menghapus dan mengubah kamus bahasa.

Admin Kamus Bahasa			
<X – 30 – X> <X – 30 – X> <X – 30 – X> Halaman 999/99			
<table border="1"> <thead> <tr> <th>Menu</th> </tr> </thead> <tbody> <tr> <td> Tambah Ubah Hapus Berikutnya Sebelumnya </td> </tr> </tbody> </table>		Menu	Tambah Ubah Hapus Berikutnya Sebelumnya
Menu			
Tambah Ubah Hapus Berikutnya Sebelumnya			
Kembali	Menu		

Gambar 3.17 : Rancangan Layar Admin Kamus Bahasa

o. Rancangan Layar *Update Admin*

Gambar 3.18 berikut ini adalah rancangan layar *Update Admin*. Pada layar ini menampilkan data admin dan terdapat *form* untuk menambah, menghapus dan mengubah admin.

Update Admin			
x – 30 – x x – 30 – x x – 30 – x Halaman 99/99			
<table border="1"> <thead> <tr> <th>Menu</th> </tr> </thead> <tbody> <tr> <td> Tambah Ubah Hapus Berikutnya Sebelumnya </td> </tr> </tbody> </table>		Menu	Tambah Ubah Hapus Berikutnya Sebelumnya
Menu			
Tambah Ubah Hapus Berikutnya Sebelumnya			
Kembali	Menu		

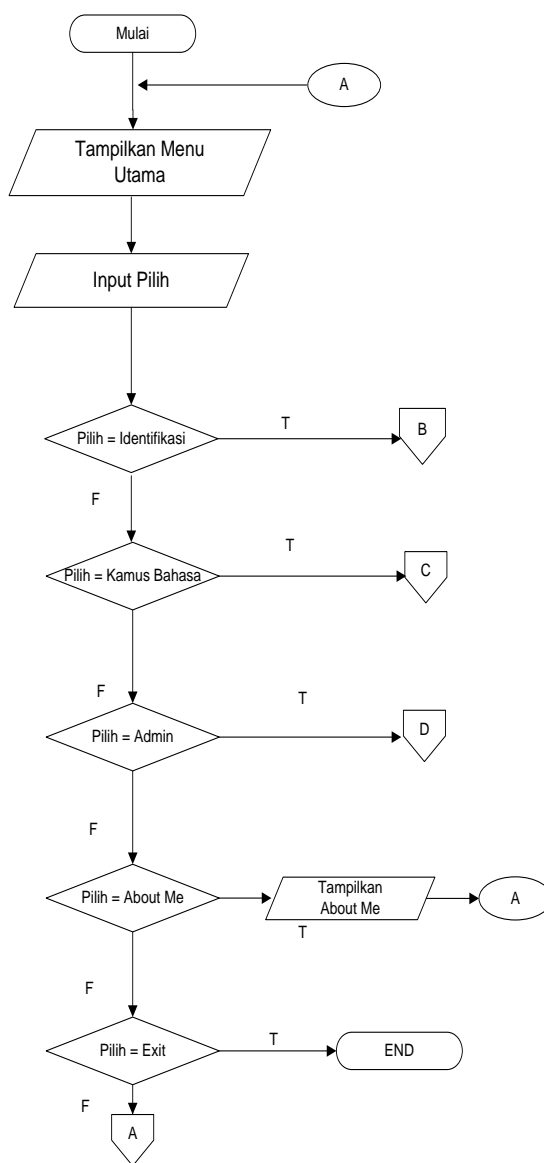
Gambar 3.18 : Rancangan Layar *Update Admin*

7. Flowchart

Dalam pengembangan program sistem pakar ini menggunakan *flowchart* untuk membantu dalam pembuatan program. Berikut ini beberapa *flowchart* untuk masing-masing proses.

a. Flowchart Layar Menu Utama

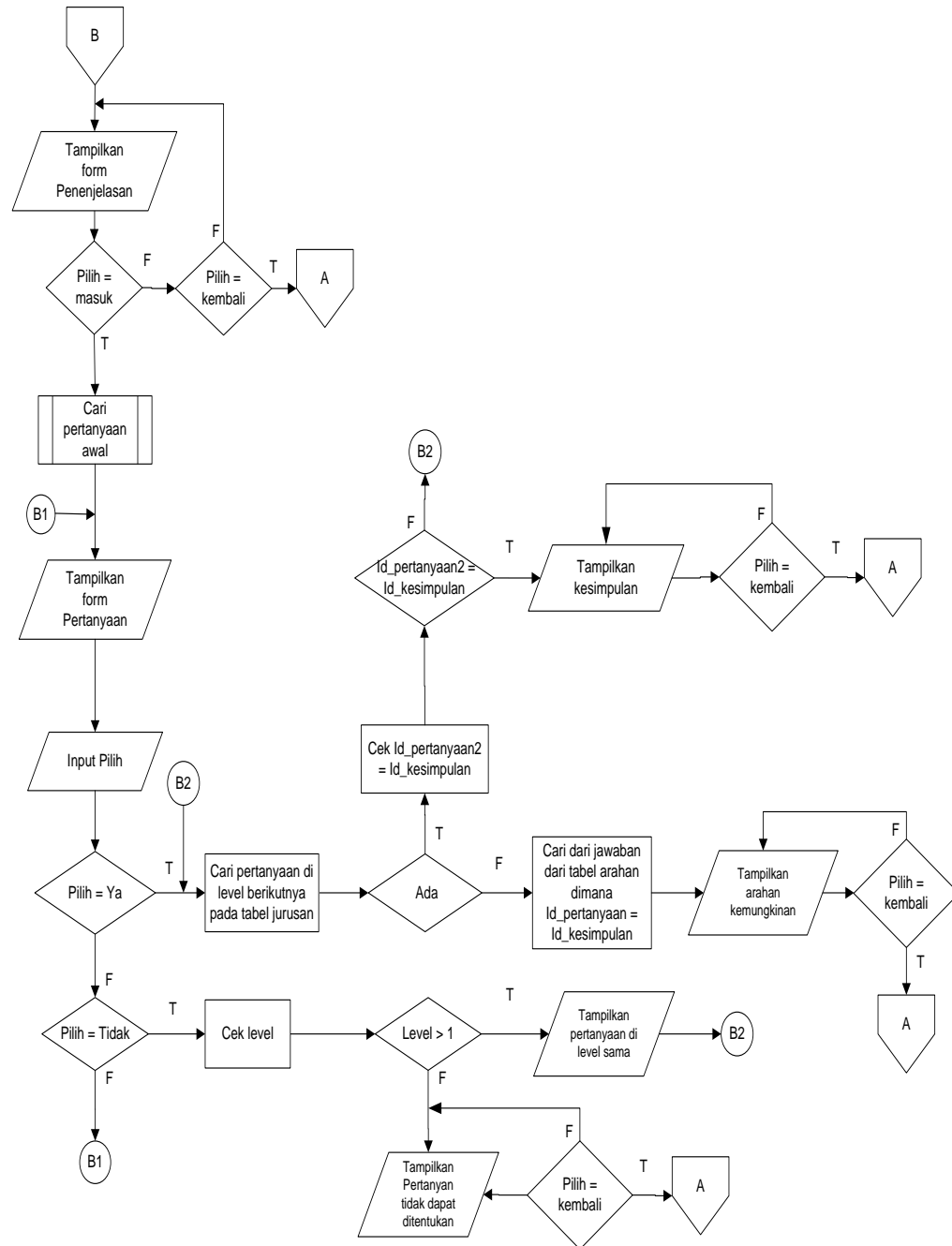
Gambar 3.19 berikut ini adalah *flowchart* layar menu utama. *Flowchart* ini menggambarkan alur proses layar yang menampilkan tampilan program halaman utama dari sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.19 : *Flowchart* Layar Menu Utama

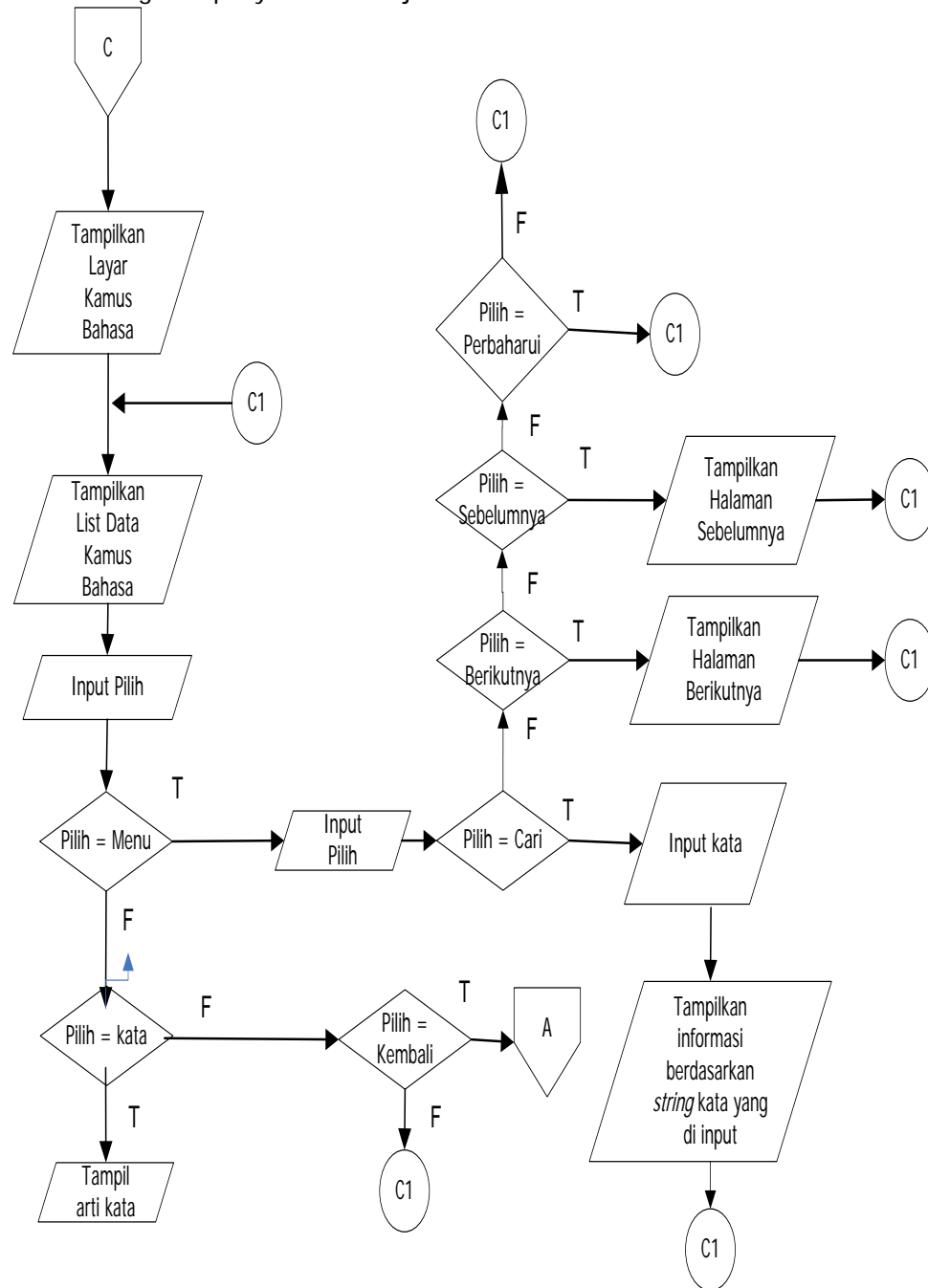
b. Flowchart Layar Identifikasi

Gambar 3.20 berikut ini adalah *flowchart* layar identifikasi. *Flowchart* ini menggambarkan alur program sistem pakar untuk mendiagnosa penyakit kulit wajah berdasarkan gejala yang dialami.



c. **Flowchart Layar Kamus Bahasa**

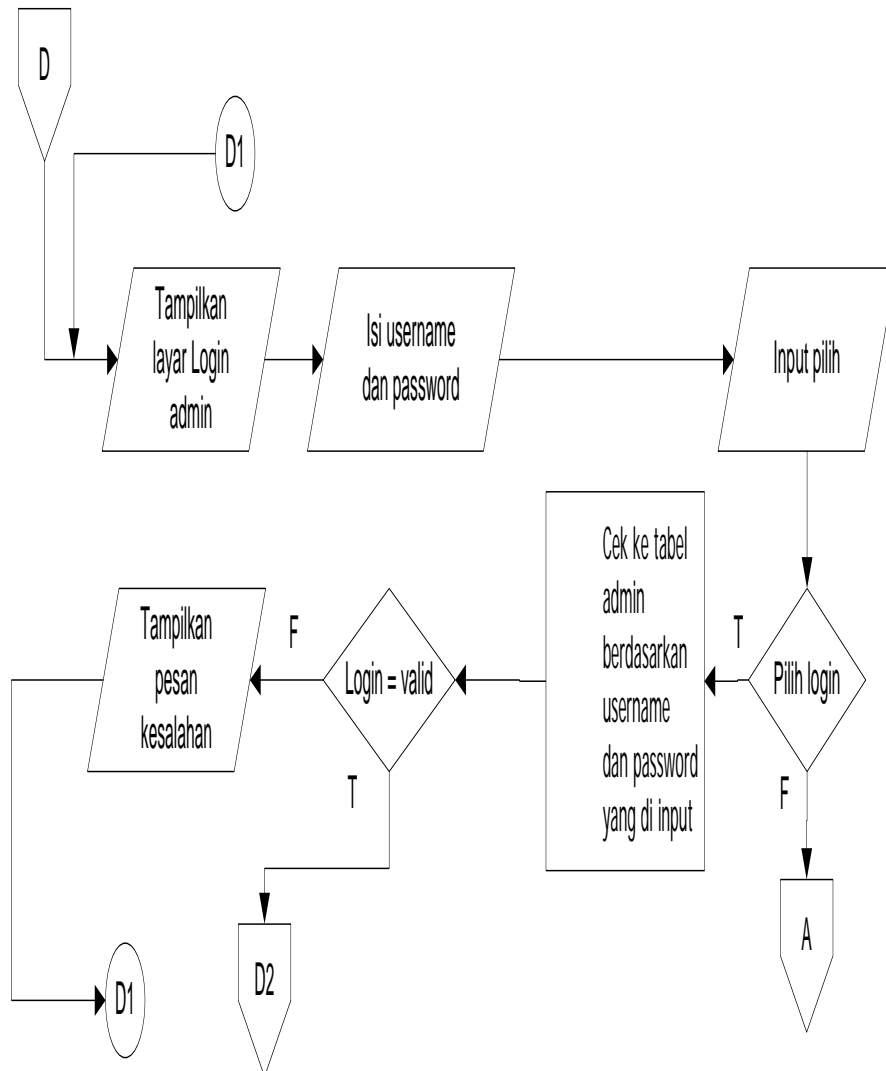
Gambar 3.21 berikut ini adalah *flowchart* layar kamus bahasa. *Flowchart* ini menggambarkan alur program kamus bahasa pada sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.21 : *Flowchart* Layar Kamus Bahasa

d. **Flowchart Layar Login Admin**

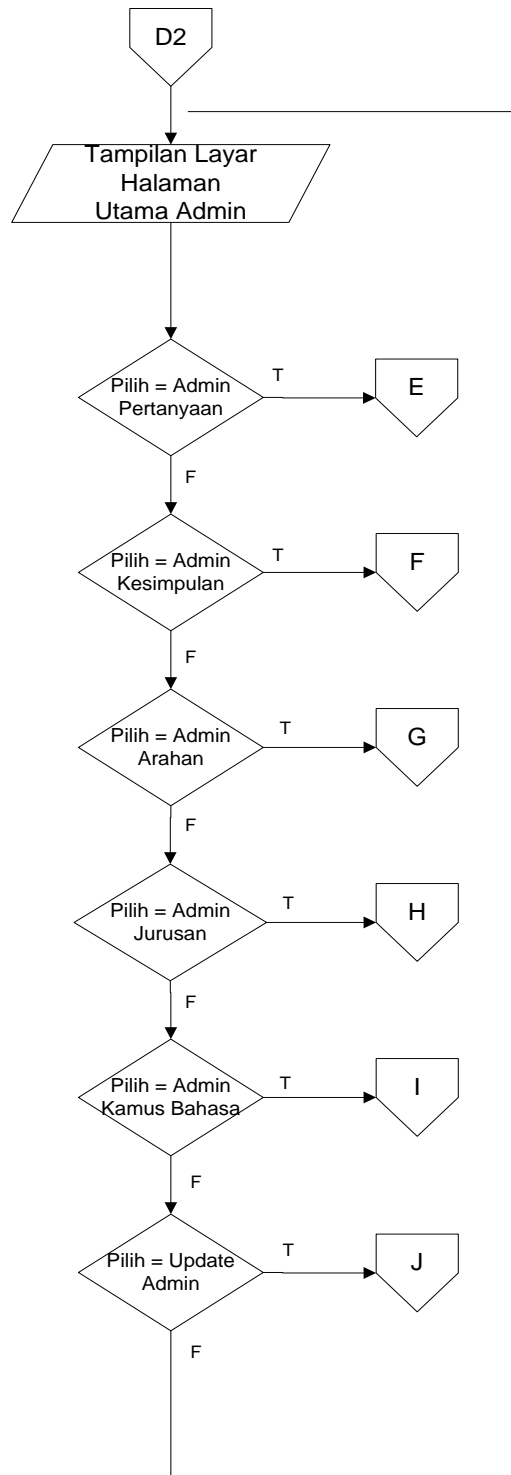
Gambar 3.22 berikut ini adalah *flowchart* layar *login* admin. *Flowchart* ini menggambarkan alur program *login* admin sistem pakar.



Gambar 3.22 : *Flowchart* Layar *Login* Admin

e. **Flowchart Layar Menu Utama Admin**

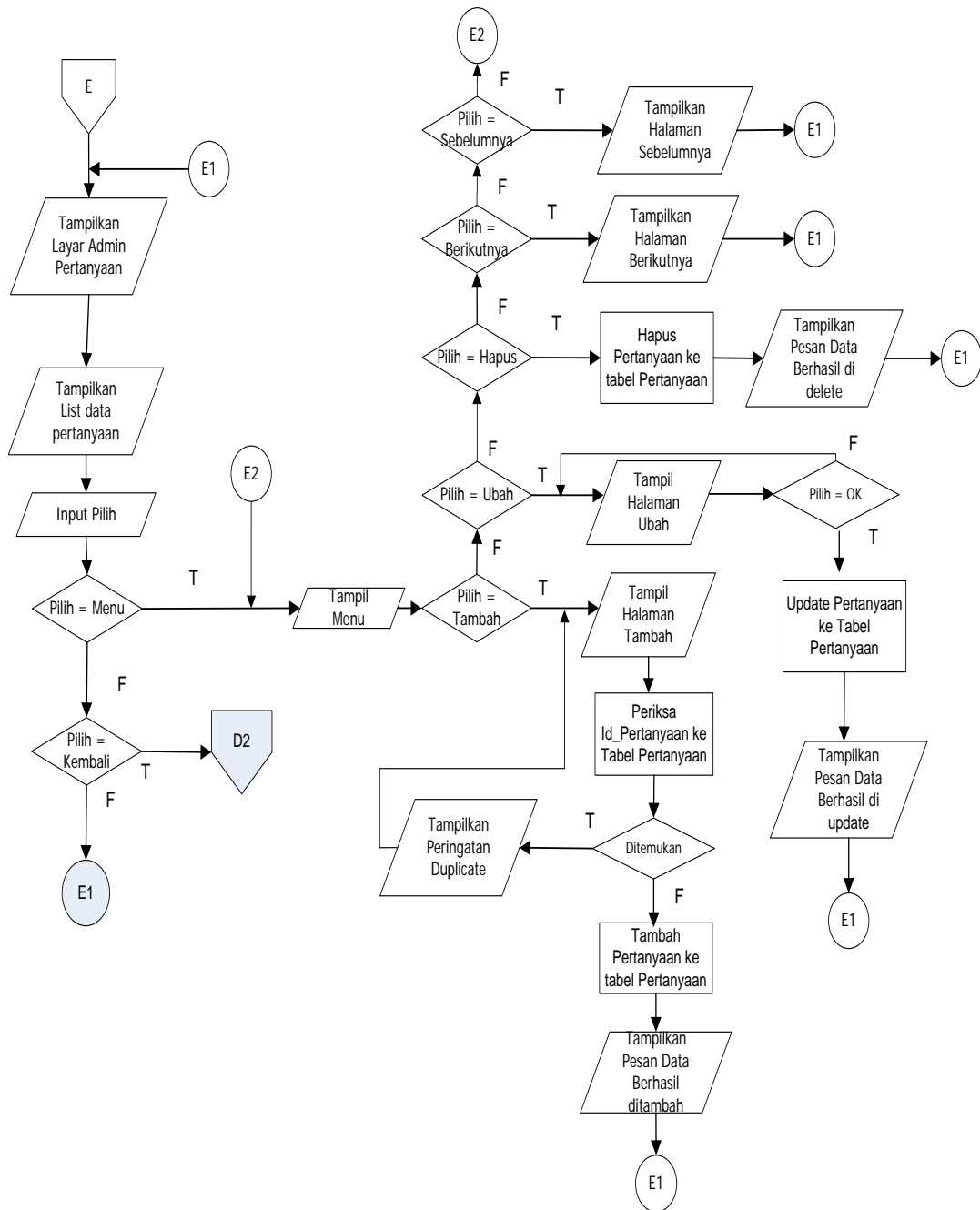
Gambar 3.23 berikut ini adalah *flowchart* layar menu utama admin. *Flowchart* ini menggambarkan alur proses layar yang menampilkan tampilan awal program menu utama admin dari sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.23 : *Flowchart* Layar Menu Utama Admin

f. **Flowchart Layar Admin Pertanyaan**

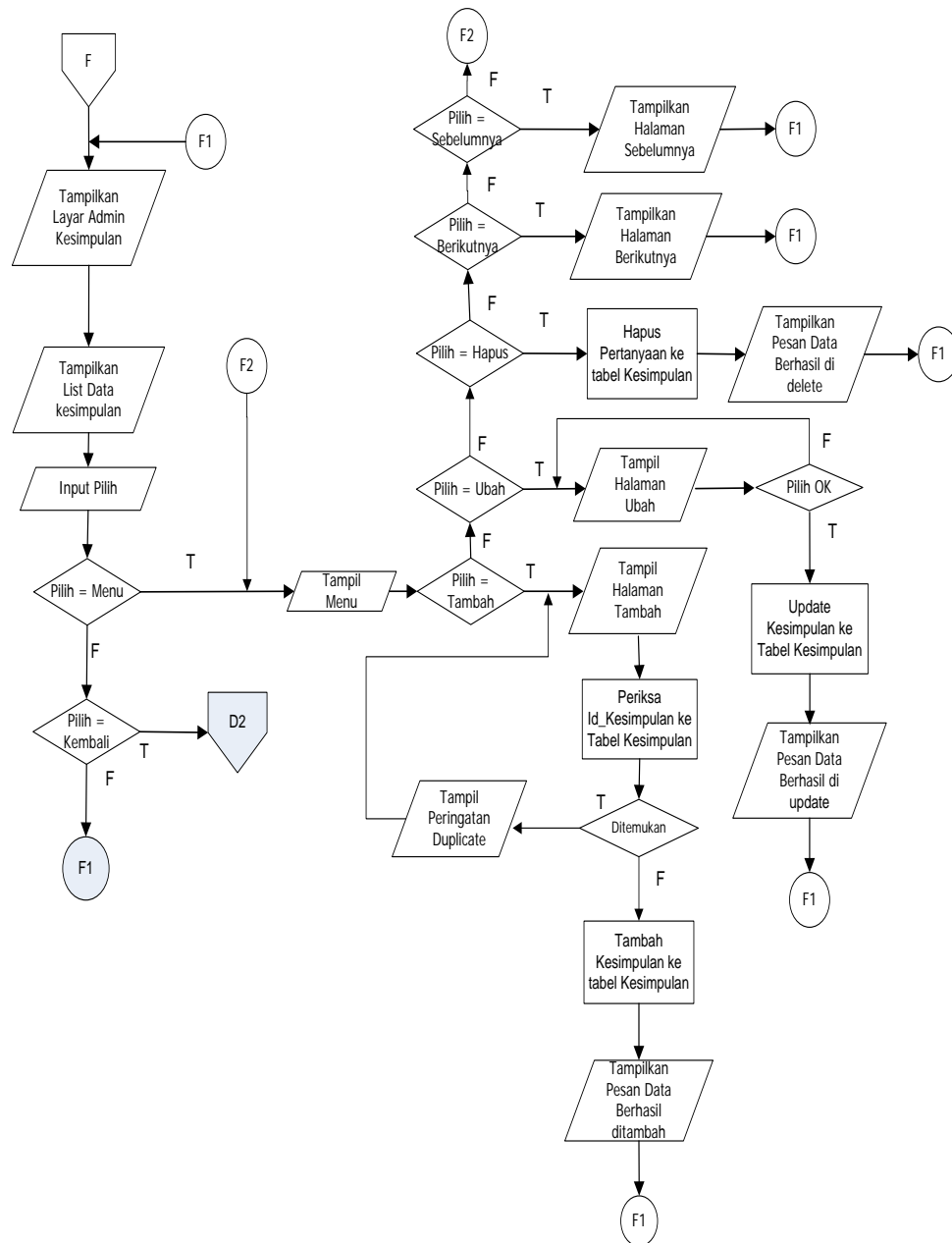
Gambar 3.24 berikut ini adalah *flowchart* layar admin pertanyaan. *Flowchart* ini menggambarkan alur program admin pertanyaan pada sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.24 : *Flowchart* Layar Admin Pertanyaan

g. Flowchart Layar Admin Kesimpulan

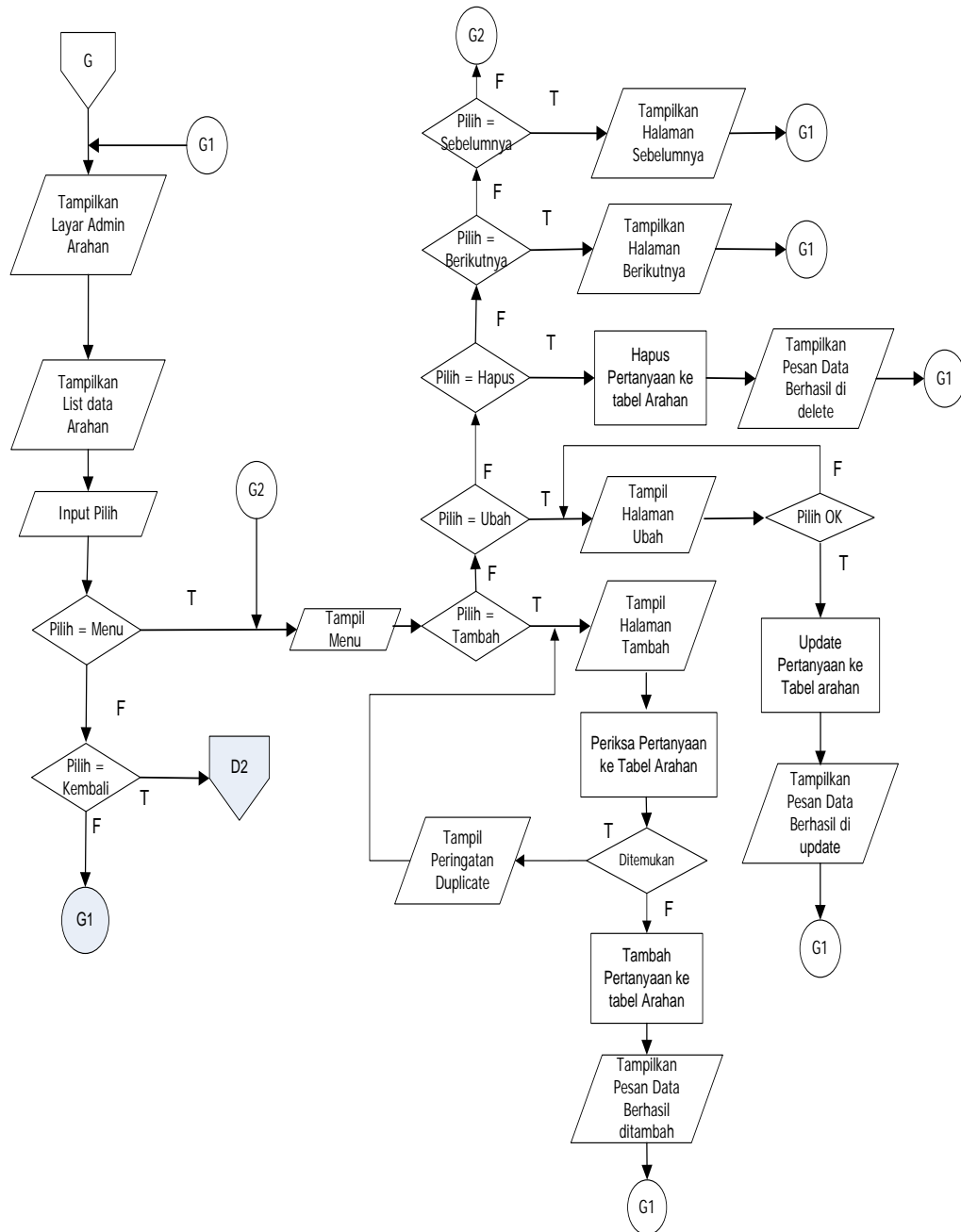
Gambar 3.25 berikut ini adalah *flowchart* layar admin kesimpulan. *Flowchart* ini menggambarkan alur program admin kesimpulan pada sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.25 : *Flowchart* Layar Admin Kesimpulan

h. *Flowchart* Layar Admin Arah

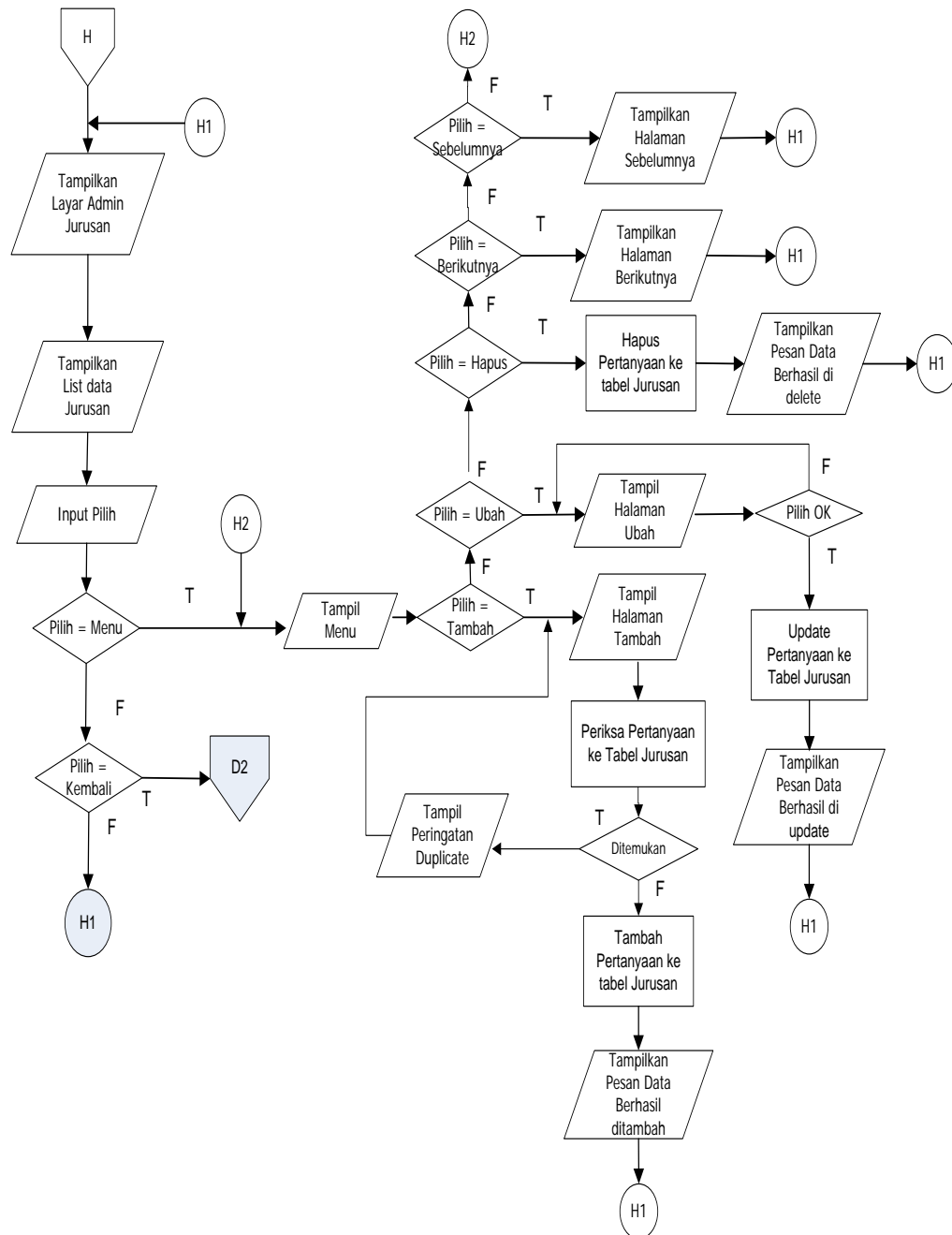
Gambar 3.26 berikut ini adalah *flowchart* layar admin arahan. *Flowchart* ini menggambarkan alur program admin arahan pada sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.26 : *Flowchart* Layar Admin Arah

i. **Flowchart Layar Admin Jurusan**

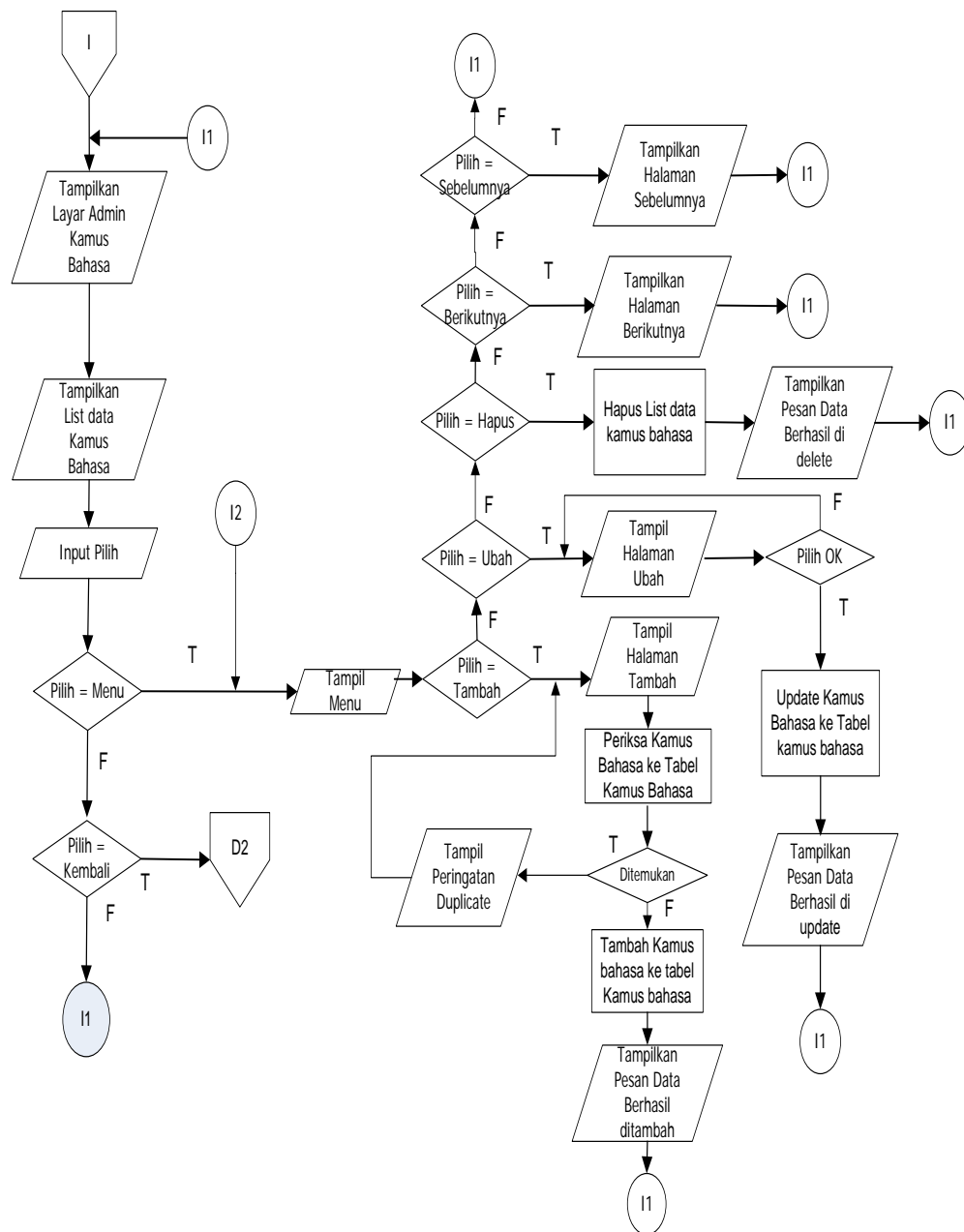
Gamabr 3.27 berikut ini adalah *flowchart* layar admin jurusan. *Flowchart* ini menggambarkan alur program admin jurusan pada sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.27 : *Flowchart* Layar Admin Jurusan

j. **Flowchart Layar Admin Kamus Bahasa**

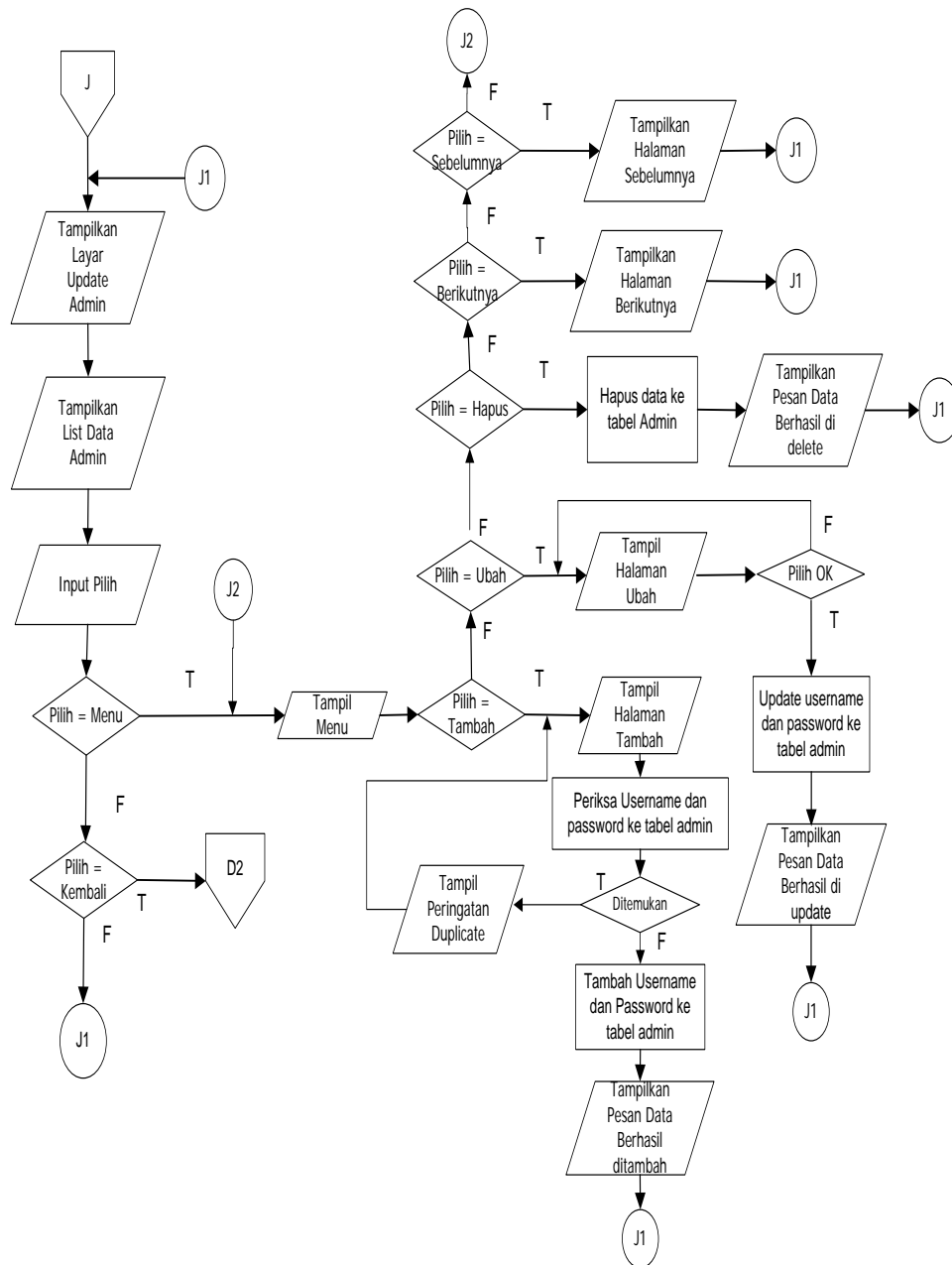
Gambar 3.28 berikut ini adalah *flowchart* layar admin Kamus Bahasa. *Flowchart* ini menggambarkan alur program admin kamus bahasa pada sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.28 : *Flowchart* Layar Admin Kamus Bahasa

k. **Flowchart Layar Update Admin**

Gambar 3.29 berikut ini adalah *flowchart* layar *Update* admin. *Flowchart* ini menggambarkan alur program admin *Update* admin pada sistem pakar untuk mendiagnosa penyakit kulit wajah.



Gambar 3.29: *Flowchart* Layar *Update* Admin

8. Algoritma

Sebelum membuat program terlebih dahulu dibuat alur algoritma, hal ini bertujuan untuk mempermudah dalam melacak dan menelusuri kesalahan-kesalahan pada program. Berikut ini adalah algoritma proses aplikasi sistem pakar yang dikelompokkan dalam beberapa proses dari fungsinya masing-masing.

a. Algoritma Menu Utama

```

1. Tampilan Layar menu utama
2. Input pilih
3. If pilih = "Identifikasi" Then
4.     Tampilkan layar Identifikasi
5. Else if pilih = "Kamus Bahasa" Then
6.     Tampilkan layar Kamus Bahasa
7. Else if pilih = "Admin" Then
8.     Tampilkan layar Admin
9. Else if pilih = "About Me" Then
10.    Tampilkan halaman About Me
11. Else if pilih = exit Then
12.    End
13. Else
        Kembali ke Baris 1

```

b. Algoritma Layar Identifikasi

```

1. Menu Identifikasi
2. Tampilkan Form Penjelasan
3. if pilih = "masuk" Then
4.     Cari Pertanyaan Awal
5.     Tampilkan form pertanyaan
6.     Input pilih
7.     If pilih = "Ya"
8.         Cari pertanyaan di level berikutnya pada table jurusan
9.         if pilih = "ada" Then
10.            Cek Id_pertanyaan2=Kesimpulan
11.            If pilih = "id_pertanyaan2=id_kesimpulan" Then
12.                Tampilkan Kesimpulan
13.                if pilih = "kembali" then
14.                    Kembali Ke menu utama
15.                Else Tampilkan Kesimpulan
16.                Else Kembali pada baris 8
17.            Else Cari dari jawaban dari table arahan dimana id_pertanyaan
18.                = id_kesimpulan
19.            Tampilkan arahan kemungkinan
20.            If pilih = "kembali" then
21.                Kembali ke menu Utama
22.            Else tampil arahan kemungkinan
23.        if pilih = "tidak" then
24.            cek level

```

```

25.      if pilih = "level >1" then
26.          tampilkan pertanyaan di level sama
27.          kembali ke baris 8
28.      Else
29.          Tampilkan pertanyaan tidak dapat ditentukan
30.          If pilih = "kembali" then
31.              Kembali ke menu utama
32.          else
33.              tampilkan pertanyaan tidak dapat ditentukan
34.          else if pilih = "kembali" then
35.              kembali ke menu utama
36.          else
37.              kembali ke baris 1
38.          end if

```

c. Algoritma Kamus Bahasa

```

1.  Tampilkan Layar kamus bahasa
2.  Tampilkan list kamus bahasa
3.  Input Pilih
4.  If Pilih = "Menu" then
5.      Input pilih
6.      if pilih = " cari" then
7.          Input data
8.          Tampilkan Informasi berdasarkan String Kata yang diinput
9.          Kembali ke tampilan list data kamus bahasa
10. Else if pilih berikutnya then
11.     Tampilkan halaman berikutnya
12.     Kembali tampilkan list data kamus bahasa
13. Else if pilih = "sebelumnya" then
14.     Tampilkan halaman sebelumnya
15.     Kembali tampilkan list data kamus bahasa
16. Else if pilih = "perbaharui" then
17.     Kembali tampilkan list data kamus bahasa
18. else kembali ke tampilan list data kamus bahasa
19. End if
20. Else if input pilih kata then
21.     Tampil arti kata
22. Else if pilih = "kembali" then
23.     Kembali ke menu utama
24. Else
25.     Kembali kelist data kamus bahasa
26. end if

```

d. Algoritma *Login* Admin

1. Tampilan layar login admin
2. Isi username dan password
3. Input pilih
4. *If* pilih = "login" *then*
5. Cek ke table admin berdasarkan username dan password yang diinput
6. *if* login = valid *then*
7. Menuju halaman layar utama admin
8. *Else*
9. Tampilkan pesan kesalahan
10. Tampilka layar login admin
11. *else* kembali ke menu utama
12. *End if*

e. Algoritma Layar Menu Utama Admin

1. *Tampilan menu utama Admin*
2. *If* pilih = "admin pertanyaan *then*
3. Menuju menu layar admin pertanyaan
4. *Else if* pilih = "admin kesimpulan" *then*
5. Menuju menu layar admin kesimpulan
6. *Else if* pilih = "admin arahan" *then*
7. Menuju menu admin arahan
8. *Else if* pilih = admin jurusan *then*
9. Menuju menu admin jurusan
10. *Else if* pilih = admin kamus bahasa *then*
11. Menuju menu admin kamus bahasa
12. *Else if* pilih = "update admin" *then*
13. Menuju menu update admin
14. *Else*
15. Kembali ke baris 1
16. *End if*

f. Algoritma Admin Pertanyaan

1. Buka Tampilan layar admin pertanyaan
2. Tampilkan list data pertanyaan
3. Input pilih
4. *If* pilih = "menu" *then*
5. Tampil menu
6. *If* pilih = "tambah" *then*
7. Tampil Halaman Tambah
8. Periksa id_pertanyaan ke table pertanyaan
9. *If* "ditemukan" *then*
10. Tampilkan peringatan duplicate
11. Kembali ke tampil halaman ubah
12. *Else*

```

13.   Tambah pertanyaan ke table pertanyaan
14.   Tampilkan pesan data berhasil ditambah
15.   Kembali ke baris 1
16.   else if pilih="ubah" then
17.   tampil halaman ubah
18.   else if pilih = "OK" then
19.   update pertanyaan ke table pertanyaan
20.   tampilkan pesan data berhasil diupdate
21.   kembali ke baris 1
22.   else
23.   kembali tampil halaman ubah
24.   else if pilih = "hapus" then
25.   hapus pertanyaan ke table pertanyaan
26.   tampilkan pesan data berhasil di delete
27.   kembali ke baris 1
28.   else if pilih = "berikutnya" then
29.   tampilkan halaman berikutnya
30.   kembali ke baris satu
31.   else if pilih = "sebelumnya" then
32.   tampilkan halaman sebelumnya
33.   kembali ke baris Satu
34.   else
35.   kembali ke tampil menu
36.   else if pilih = "kembali" then
37.   kembali ke layar utama admin
38.   else
39.   kembali kebaris 1
40.   end if

```

g. Algoritma Admin Kesimpulan

```

1.   Buka Tampilan layar admin Kesimpulan
2.   Tampilkan list data kesimpulan
3.   Input pilih
4.   If pilih ="menu" then
5.   Tampil menu
6.   If pilih ="tambah" then
7.   Tampil Halaman Tambah
8.   Periksa id_Kesimpulan ke table kesimpulan
9.   If "ditemukan" then
10.  Tampilkan peringatan duplicate
11.  Kembali ke tampil halaman ubah
12.  Else
13.  Tambah Kesimpulan ke table Kesimpulan
14.  Tampilkan pesan data berhasil ditambah
15.  Kembali ke baris 1
16.  else if pilih="ubah" then
17.  tampil halaman ubah

```

```

18.     else if pilih = "OK" then
19.         update Kesimpulan ke table Kesimpulan
20.         tampilkan pesan data berhasil diupdate
21.         kembali ke baris 1
22.     else
23.         kembali tampil halaman ubah
24.     else if pilih = "hapus" then
25.         hapus Kesimpulan ke table Kesimpulan
26.         tampilkan pesan data berhasil di delete
27.         kembali ke baris 1
28.     else if pilih = "berikutnya" then
29.         tampilkan halaman berikutnya
30.         kembali ke baris satu
31.     else if pilih = "sebelumnya" then
32.         tampilkan halaman sebelumnya
33.         kembali ke baris Satu
34.     else
35.         kembali ke tampil menu
36.     else if pilih = "kembali" then
37.         kembali ke layar utama admin
38.     else
39.         kembali kebaris 1
40.     end if

```

h. Algoritma Admin Arahkan

```

1.     Buka Tampilan layar admin arahan
2.     Tampilkan list data arahan
3.     Input pilih
4.     If pilih = "menu" then
5.         Tampil menu
6.     If pilih = "tambah" then
7.         Tampil Halaman Tambah
8.         Periksa id_Pertanyaan ke table arahan
9.         If "ditemukan" then
10.            Tampilkan peringatan duplicate
11.            Kembali ke tampil halaman ubah
12.        Else
13.            Tambah pertanyaan ke table Arahan
14.            Tampilkan pesan data berhasil ditambah
15.            Kembali ke baris 1
16.        else if pilih="ubah" then
17.            tampil halaman ubah
18.            else if pilih = "OK" then
19.                update Pertanyaan ke table Arahan
20.                tampilkan pesan data berhasil diupdate
21.                kembali ke baris 1
22.            else
23.                kembali tampil halaman ubah

```

```

24.     else if pilih = "hapus" then
25.         hapus Pertanyaan ke table Arahkan
26.         tampilkan pesan data berhasil di delete
27.         kembali ke baris 1
28.     else if pilih = "berikutnya" then
29.         tampilkan halaman berikutnya
30.         kembali ke baris satu
31.     else if pilih = "sebelumnya" then
32.         tampilkan halaman sebelumnya
33.         kembali ke baris Satu
34.     else
35.         kembali ke tampil menu
36.     else if pilih = "kembali" then
37.         kembali ke layar utama admin
38.     else
39.         kembali kebaris 1
40.     end if

```

i. Algoritma Admin Jurusan

```

1.  Buka Tampilan layar admin Jurusan
2.  Tampilkan list data Jurusan
3.  Input pilih
4.  If pilih = "menu" then
5.      Tampil menu
6.  If pilih = "tambah" then
7.      Tampil Halaman Tambah
8.      Periksa id_Pertanyaan ke table jurusan
9.      If "ditemukan" then
10.         Tampilkan peringatan duplicate
11.         Kembali ke tampil halaman ubah
12.     Else
13.         Tambah pertanyaan ke table Jurusan
14.         Tampilkan pesan data berhasil ditambah
15.         Kembali ke baris 1
16.     else if pilih="ubah" then
17.         tampil halaman ubah
18.         else if pilih = "OK" then
19.             update Pertanyaan ke table Jurusan
20.             tampilkan pesan data berhasil diupdate
21.             kembali ke baris 1
22.         else
23.             kembali tampil halaman ubah
24.     else if pilih = "hapus" then

```



```

25.      hapus Pertanyaan ke table Jurusan
26.      tampilkan pesan data berhasil di delete
27.      kembali ke baris 1
28.      else if pilih ="berikutnya" then
29.          tampilkan halaman berikutnya
30.          kembali ke baris satu
31.      else if pilih ="sebelumnya" then
32.          tampilkan halaman sebelumnya
33.          kembali ke baris Satu
34.      else
35.          kembali ke tampil menu
36.      else if pilih = "kembali" then
37.          kembali ke layar utama admin
38.      else
39.          kembali kebaris 1
40.      end if

```

j. Algoritma *Update Admin Kamus Bahasa*

```

1.      Buka Tampilan layar admin Kamus Bahasa
2.      Tampilkan list data Kamus Bahasa
3.      Input pilih
4.      If pilih ="menu" then
5.          Tampil menu
6.      If pilih ="tambah" then
7.          Tampil Halaman Tambah
8.          Periksa kamus bahasa ke table kamus bahasa
9.          If "ditemukan" then
10.             Tampilkan peringatan duplicate
11.             Kembali ke tampil halaman ubah
12.          Else
13.             Tambah kamus bahasa ke table Kamus bahasa
14.             Tampilkan pesan data berhasil ditambah
15.             Kembali ke baris 1
16.      else if pilih ="ubah" then
17.          tampil halaman ubah
18.          else if pilih = "OK" then
19.             update Kamus bahasa ke table Kamus bahasa
20.             tampilkan pesan data berhasil diupdate
21.             kembali ke baris 1
22.          else
23.             kembali tampil halaman ubah
24.      else if pilih ="hapus" then
25.          hapus Kamus bahasa ke table Kamus bahasa
26.          tampilkan pesan data berhasil di delete
27.          kembali ke baris 1
28.      else if pilih ="berikutnya" then
29.          tampilkan halaman berikutnya

```

```

30.     kembali ke baris satu
31.     else if pilih = "sebelumnya" then
32.         tampilkan halaman sebelumnya
33.         kembali ke baris Satu
34.     else
35.         kembali ke tampil menu
36.     else if pilih = "kembali" then
37.         kembali ke layar utama admin
38.     else
39.         kembali kebaris 1
40.     end if

```

k. Algoritma Layar Update Admin

```

1.     Buka Tampilan layar update admin
2.     Tampilkan list data admin
3.     Input pilih
4.     If pilih = "menu" then
5.         Tampil menu
6.     If pilih = "tambah" then
7.         Tampil Halaman Tambah
8.         Periksa username dan password ke table admin
9.         If "ditemukan" then
10.            Tampilkan peringatan duplicate
11.            Kembali ke tampil halaman ubah
12.        Else
13.            Tambah username dan password ke table Admin
14.            Tampilkan pesan data berhasil ditambah
15.            Kembali ke baris 1
16.    else if pilih = "ubah" then
17.        tampil halaman ubah
18.        else if pilih = "OK" then
19.            update Username dan password ke table Admin
20.            tampilkan pesan data berhasil diupdate
21.            kembali ke baris 1
22.        else
23.            kembali tampil halaman ubah
24.    else if pilih = "hapus" then
25.        hapus Data ke table Admin
26.        tampilkan pesan data berhasil di delete
27.        kembali ke baris 1
28.    else if pilih = "berikutnya" then
29.        tampilkan halaman berikutnya
30.        kembali ke baris satu
31.    else if pilih = "sebelumnya" then
32.        tampilkan halaman sebelumnya
33.        kembali ke baris Satu
34.    else

```

```
35.      kembali ke tampil menu
36.  else if pilih = "kembali" then
37.      kembali ke layar utama admin
38.  else
39.      kembali kebaris 1
40.  end if
```

BAB IV IMPLEMENTASI DAN ANALISA PROGRAM

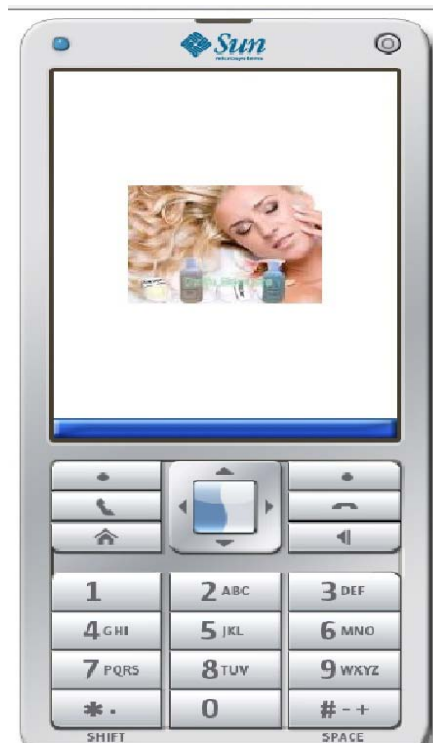
Pada pengimplementasian dan analisa program ini akan dilakukan pengujian serta analisa dari program Sistem Pakar yang telah dirancang. Tujuannya adalah untuk mengukur sejauh mana Sistem Pakar ini dapat menyelesaikan masalah. Dengan adanya pengimplementasian serta uji coba tersebut dapat mempermudah *knowledge engineer* atau Sistem Pakar untuk melihat adanya kekurangan pada sistem yang telah dirancang. Dengan demikian pada masa yang akan datang dapat dilakukan pengembangan sistem kearah yang lebih baik lagi.

1. Implementasi

Pengoperasian sistem pakar ini dapat dilakukan dengan cara membuka *browser* setelah XAMPP terpasang tetapi sebelumnya, masukkan *file* Sistem Pakar yang akan digunakan pada direktori *D:\htdocs* terlebih dahulu.

a. Tampilan Layar Awal

Gambar 4.1 adalah tampilan Awal yang akan muncul apabila program pertama kali dijalankan. Tampilan ini akan otomatis hilang dan diganti apabila waktu telah mencapai 2 detik.



Gambar 4.1 : Tampilan Layar Awal

b. Tampilan Layar *form* Utama

Gambar 4.2 adalah tampilan menu utama yang akan muncul apabila *splashscreen* selesai ditampilkan oleh aplikasi. Menu utama ini berfungsi untuk menampilkan *list menu* yang ada didalam aplikasi sistem pakar *mobile* ini.



Gambar 4.2 : Tampilan layar *form* Utama

c. Tampilan Layar *Form* Identifikasi

Gambar 4.3 adalah tampilan menu identifikasi. Menu identifikasi ini berfungsi untuk menampilkan pertanyaan dan kemudian diolah oleh aplikasi untuk mengetahui penyakit yang terjadi pada kulit wajah. Menu identifikasi akan memberikan pilihan jawaban **Ya** atau **Tidak** yang akan diolah oleh sistem secara *online*.



Gambar 4.3 : Tampilan Layar Identifikasi

d. Tampilan Layar *Form* Kesimpulan

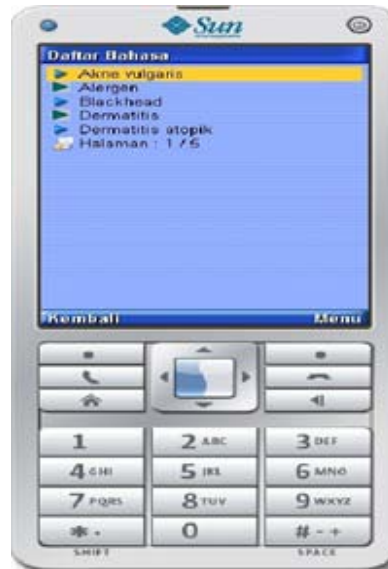
Gambar 4.4 adalah tampilan *form* Kesimpulan. *Form* Kesimpulan ini berfungsi untuk menampilkan masalah dan solusi dari penyakit kulit wajah.



Gambar 4.4 : Tampilan Layar *Form* Kesimpulan

e. **Tampilan Layar Kamus**

Gambar 4.5 adalah tampilan *form* kamus. Tampilan menu kamus yang berfungsi untuk mencari arti kamus kata.



Gambar 4.5 : Tampilan Layar *form* Kamus

f. **Tampilan Layar *About Me***

Gambar 4.6 adalah tampilan layar *about me*. Pada tampilan layar *form about me*, *user* dapat melihat biodata dari pembuat program sistem pakar seperti: "Nama", "Nim", "Email", "Twitter" dan "Fakultas".



Gambar 4.6 : Tampilan Layar *form About Me*

g. Tampilan Layar *Login*

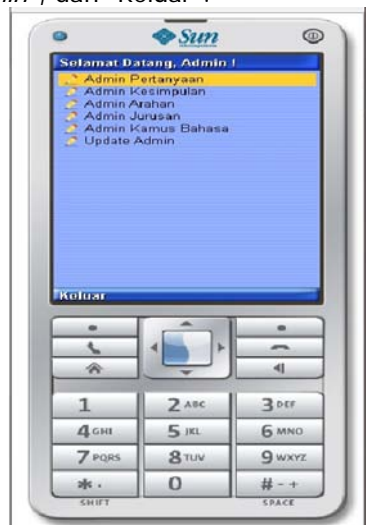
Gambar 4.7 adalah tampilan layar *form Login*. Pada tampilan layar *login* user dapat melakukan *login* untuk masuk ke *form* admin dengan memasukkan *username* dan *password*.



Gambar 4.7 : Tampilan Layar *Login*

h. Tampilan Layar *Form Admin*

Gambar 4.8 adalah tampilan layar menu *Form Admin*. Pada tampilan layar *Form Admin*, user harus *login* kemudian muncul *Form Admin* yang terdiri dari "Admin Pertanyaan", "Admin Kesimpulan", "Admin Arahan", "Admin Jurusan", "Admin Kamus Bahasa", "Update Admin", dan "Keluar".



Gambar 4.8 : Tampilan Layar *Form Admin*

i. Tampilan Layar Admin Pertanyaan

Gambar 4.9 adalah Tampilan Layar admin pertanyaan. Pada tampilan layar *Form*. Tambah Pertanyaan, *user* bisa menambah, mengubah dan menghapus pertanyaan dengan mengisi *field* "id_pertanyaan", "pertanyaan", kemudian pilih menu "Tambah" untuk menambah data, pilih menu "Ubah" untuk mengubah data, pilih menu "Hapus" untuk menghapus data, dan menu "Back" untuk kembali ke menu *Admin*.



Gambar 4.9 : Tampilan Layar Admin Pertanyaan

j. Tampilan Layar Admin Kesimpulan

Gambar 4.10 adalah Tampilan Layar Admin Kesimpulan. Pada tampilan layar *Form* Admin Kesimpulan, *user* bisa menambah, mengubah dan menghapus kesimpulan dengan mengisi *field* "id_kesimpulan", "kesimpulan", "solusi", kemudian pilih menu "Tambah" untuk menambah data, pilih menu "Ubah" untuk mengubah data, pilih menu "Hapus" untuk menghapus data, dan menu "Back" untuk kembali ke menu *Admin*.



Gambar 4.10 : Tampilan Layar Admin Kesimpulan

k. Tampilan Layar Admin Arahkan

Gambar 4.11 adalah tampilan layar Admin Arahkan. Pada tampilan layar *Form* Admin Arahkan, *user* bisa menambah, mengubah dan menghapus arahan dengan mengisi *field* "id_pertanyaan", "id_kesimpulan", kemudian pilih menu "Tambah" untuk menambah data, pilih menu "Ubah" untuk mengubah data, pilih menu "Hapus" untuk menghapus data, dan menu "Back" untuk kembali ke menu *Admin*.



Gambar 4.11 : Tampilan Layar Admin Arahkan

l. Tampilan Layar Admin Jurusan

Gambar 4.12 adalah Tampilan layar Admin Jurusan. Pada tampilan layar *Form* Admin Jurusan, *user* bisa menambah, mengubah dan menghapus Jurusan dengan mengisi *field* "id_pertanyaan", "id_pertanyaan2", kemudian pilih menu "Tambah" untuk menambah data, pilih menu "Ubah" untuk mengubah data, pilih menu "Hapus" untuk menghapus data, dan menu "Back" untuk kembali ke menu *Admin*.



Gambar 4.12 : Tampilan Layar Admin Jurusan

m. Tampilan Layar Admin Kamus

Gambar 4.13 adalah Tampilan Layar Admin Kamus. Pada tampilan layar *Form* Admin Kamus, *user* bisa menambah, mengubah dan menghapus kamus dengan mengisi *field* "kata", "Keterangan", kemudian pilih menu "Tambah" untuk menambah data, pilih menu "Ubah" untuk mengubah data, pilih menu "Hapus" untuk menghapus data, dan menu "Back" untuk kembali ke menu *Admin*.



Gambar 4.13 : Tampilan Layar Admin Kamus

n. Tampilan Layar *Update Admin*

Gambar 4.14 adalah Tampilan Layar *Update Admin*. Pada tampilan layar *Form* Update Admin, *user* bisa mengubah *password* dengan mengisi *field* "password", kemudian pilih menu "Ok" untuk mengubah data dan menu "Kembali" untuk kembali ke menu *Admin*.



Gambar 4.14 : Tampilan Layar Update Admin

2. Uji kasus

a. Spesifikasi *Hardware* Dan *Software*

Program Sistem Pakar untuk mendiagnosa penyakit kulit wajah ini telah diujicoba pada komputer dengan spesifikasi *hardware* sebagai berikut :

Program ini telah diuji coba pada komputer dengan spesifikasi *hardware* sebagai berikut :

- 1) *Motherboard* : Intel
- 2) *Procesor* : Intel Core 2 Duo
- 3) *Memory* : 3 GB
- 4) *DVD* : Super multi DR

Software yang digunakan dalam uji coba pada *hardware* diatas memiliki spesifikasi yang sama, yaitu :

- 1) *Operating System* : Microsoft Windows XP SP2
- 2) *Web Server* : Apache
- 3) Bahasa Pemrograman : PHP dan JAVA

b. Cara Menjalankan Program Penelusuran Atau Sistem Pakar

Pengujian program dilakukan untuk mengetahui apakah sudah sesuai dengan tujuan pembuatan sistem pakar ini. Berikut ini contoh pengoperasian program mendiagnosa penyakit kulit wajah untuk menuntun *user* mengoperasikan program.

1) Langkah Pertama

Pada gambar 4.15 ini langkah pertama yaitu menjalankan aplikasi. Pertama silahkan pilih terlebih dahulu menu identifikasi yang ada dimenu utama.



Gambar 4.15: Tampilan Layar *form* Utama Sebelum Identifikasi Dimulai

2) Langkah Kedua

Pada gambar 4.16 ini langkah kedua. Setelah menu identifikasi dipilih, kemudian tekan tombol masuk untuk memanggil pertanyaan pertama dan kemudian ditampilkan dilayar *handphone*.



Gambar4.16 :Tampilan layar untuk memulai identifikasi

3) Langkah Ketiga

Pada gambar 4.17 ini langkah ketiga. Muncul pertanyaan pertama, *user* akan diharuskan untuk menekan tombol pilih untuk memilih jawaban yang diperlukan sistem untuk melakukan pengolahan data kembali berdasarkan jawaban yang diinput oleh *user*. Fungsi *button* Ya dan Tidak untuk ke pertanyaan selanjutnya dan juga tombol kembali yang berfungsi untuk kembali ke menu utama.



Gambar4.17 :Tampilan layar pertanyaan pertama (level 1)

4) Langkah Keempat

Pada gambar 4.18 ini langkah keempat. Di sini pertanyaan kedua muncul setelah *user* menginput atau menekan tombol YA. Di tahap ini, pertanyaan di level 2 muncul menggantikan pertanyaan pertama. Disini *user* juga bisa melanjutkan dengan memilih menu yang disediakan oleh aplikasi seperti halnya dengan langkah ketiga.



Gambar4.18 :Tampilan layar pertanyaan kedua (level 2)

5) Langkah Kelima

Pada gambar 4.19 ini langkah kelima. Disini pertanyaan ketiga muncul setelah *user* menginput atau menekan tombol YA. Ditahap ini, pertanyaan di level 3 muncul menggantikan pertanyaan kedua. Disini *user* juga bisa melanjutkan dengan memilih menu yang disediakan oleh aplikasi seperti halnya dengan langkah ketiga.



Gambar4.19 :Tampilan layar pertanyaan ketiga (level 3)

6) Langkah Keenam

Pada gambar 4.20 ini langkah keenam. Di sini pertanyaan keempat muncul setelah *user* menginput atau menekan tombol YA. Di tahap ini, pertanyaan di level 4 muncul menggantikan pertanyaan ketiga. Disini *user* juga bisa melanjutkan dengan memilih menu yang disediakan oleh aplikasi seperti halnya dengan langkah ketiga.



Gambar 4.20 :Tampilan layar pertanyaan keempat (level 4)

7) Langkah Ketujuh

Padagambar 4.21 ini langkah ketujuh. Disini pertanyaan kelima muncul setelah *user* menginput atau menekan tombol YA. Di tahap ini, pertanyaan di level 5 muncul menggantikan pertanyaan keempat. Disini *user* juga bisa melanjutkan dengan memilih menu yang disediakan oleh aplikasi seperti halnya dengan langkah ketiga.



Gambar 4.21 : Tampilan layar pertanyaan kelima (level 5)

8) Langkah Kedelapan / kesimpulan

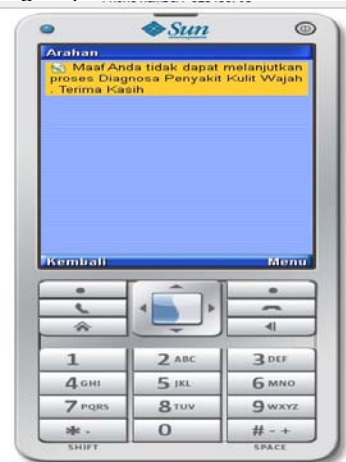
Pada gambar 4.22 ini langkah kedelapan. Disini setelah pertanyaan kelima dijawab *user* dengan menginput atau menekan tombol YA, maka kesimpulan akan menggantikan pertanyaan kelima karena ditahap ini sistem telah menemukan kesimpulan berdasarkan data ditabel jurusan. Apabila kesimpulan telah ditemukan, maka kesimpulan akan muncul dengan *mode alert* atau pesan.



Gambar 4.22 : Tampilan layar Identifikasi menemukan kesimpulan

9) Langkah Kesepuluh

Pada gambar 4.23. Ditahap ini *user* akan ditampilkan peringatan atau pesan dari sistem karena sistem tidak menemukan data yang cocok di tabel jurusan atau pun ditabel arahan. Apabila sistem tidak menemukan kecocokan data, maka otomatis sistem akan menghentikan identifikasi dan memunculkan *alert* yang berisi dengan pesan *error* tidak menemukan kesimpulan penyakit.



Gambar 4.23 : Tampilan layar Identifikasi Tidak menemukan data yang cocok

3. Kelebihan dan Kekurangan Program

a. Kelebihan Program

Kelebihan yang dimiliki program sistem pakar ini, antara lain :

- 1) Program sistem pakar ini memiliki antar muka (*interface*) yang berbasis *mobile* sehingga lebih mudah dalam penggunaan.
- 2) Sistem pakar ini bisa menjadi dokumentasi dan *database* bagi seorang pakar untuk menuangkan pengetahuan dan pengalamannya.
- 3) Biaya yang dikeluarkan dalam penggunaan program aplikasi sistem pakar ini relatif murah karena menggunakan jaringan GPRS.
- 4) Kemudahan dalam penginstalan aplikasinya.

b. Kekurangan Program

Kekurangan yang dimiliki program sistem pakar ini, antara lain :

- 1) Karena keterbatasan waktu, biaya dan kemampuan, sistem pakar ini belum mampu menampilkan data yang spesifik.
- 2) Konsultasi yang disajikan hanya sebatas yang umum saja.
- 3) Masalah pada koneksi yang kurang cepat dan terkadang tidak stabil karena menggunakan jaringan GPRS.

BAB V PENUTUP

Berdasarkan analisa yang telah dilakukan terhadap sistem pakar yang telah dibuat, maka dapat diambil kesimpulan dan saran yang mungkin diperlukan untuk membuat sistem yang lebih baik lagi.

1. Kesimpulan

Dari hasil perancangan dan pembuatan sistem pakar untuk mendiagnosa penyakit kulit wajah berbasis *mobile* dapat diambil kesimpulan sebagai berikut :

- a. Aplikasi sistem pakar ini dapat mendiagnosa penyakit kulit wajah dan solusinya.
- b. Dengan adanya program sistem pakar ini diharapkan dapat membantu masyarakat pada umumnya dan staf Maharis Clinic pada khususnya untuk dapat menyelesaikan masalah mendiagnosa penyakit kulit wajah.
- c. Penarikan solusi untuk setiap permasalahan didasarkan pada hubungan antara pertanyaan dan solusi yang disimpan dalam basis pengetahuan.

2. Saran

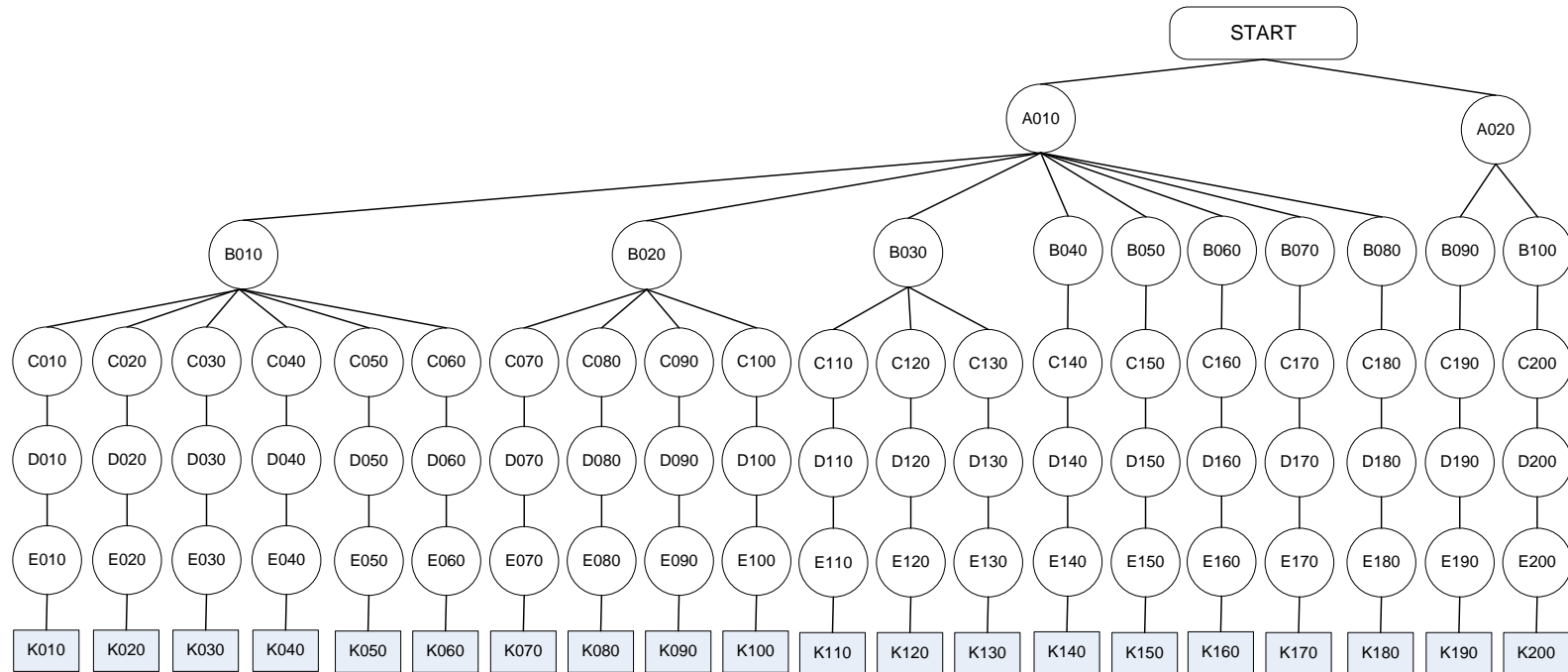
Sistem pakar yang dibuat ini masih memiliki banyak kekurangan, untuk pengembangan sistem pakar berikutnya ada beberapa hal yang perlu diperhatikan agar pengembangan sistem pakar berikutnya menjadi lebih baik lagi. Beberapa hal yang dapat dikembangkan antara lain:

- a. sistem ini masih dapat dikembangkan dengan menggunakan metode lain untuk mendiagnosa penyakit kulit wajah pada manusia.
- b. Sistem ini dapat dikembangkan lebih lanjut dalam pemanfaatan media internet atau sistem yang berbasis web.

DAFTAR PUSTAKA

- Achroni, Keen, 2012. *Semua Rahasia Kulit Cantik & Sehat*. Jakarta : PT. BUKU KITA
- Arhami, Muhammad, 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi.
- Budi Santoso, Teguh, 2009. *Data Penyakit* : medlineplus dan mayoclinic.
www.health.detik.com
- Hardini Meita Sari, Nurita, 2010, *Gejala Pencegahan dan Penyembuhan Rosacea*.
www.spesialiskulit.com
- Maspri, 2012, *Penyebab dan Cara Mengatasi Jerawat Rosacea*. www.jerawat hilang.com
- Medicastore, 2009. *Tahi Lalat, permasalahan, dan solusinya*. www.medicastore.com
- Septiana, Renny, 2010. *Meloma Magligna*, www.rennyseptiana.wordpress.com
- Shalahuddin, M., Rosa. A.S., 2012. *Pemrograman J2ME (Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile)*, Bandung : Informatika.
- Siswanto, 2005. *Kecerdasan Tiruan (Edisi 2)*. Yogyakarta : Graha Ilmu.
- Sjamsoe Daili, Emmy S, Linuwih Menaldi, Sri, Wisnu, I Made, 2005. *Penyakit Kulit Umum yang Ada Di Indonesia*. Jakarta : PT. Medical Multimedia Indonesia
- Sjukani, Moh., 2007. *Struktur Data (Algoritma & Struktur Data 2)*. Jakarta : Mitra Wacana Media.
- Solihin, Achmad., 2010, *MySQL 5 Dari Pemula Hingga Mahir*. Jakarta : Achmatim.net.
- Suparman, 1991. *Mengenal Artificial Intelligence*, Andi Offset, Yogyakarta

Lampiran 1
Pohon Keputusan
Pohon Keputusan Mendiagnosa Penyakit Kulit Wajah



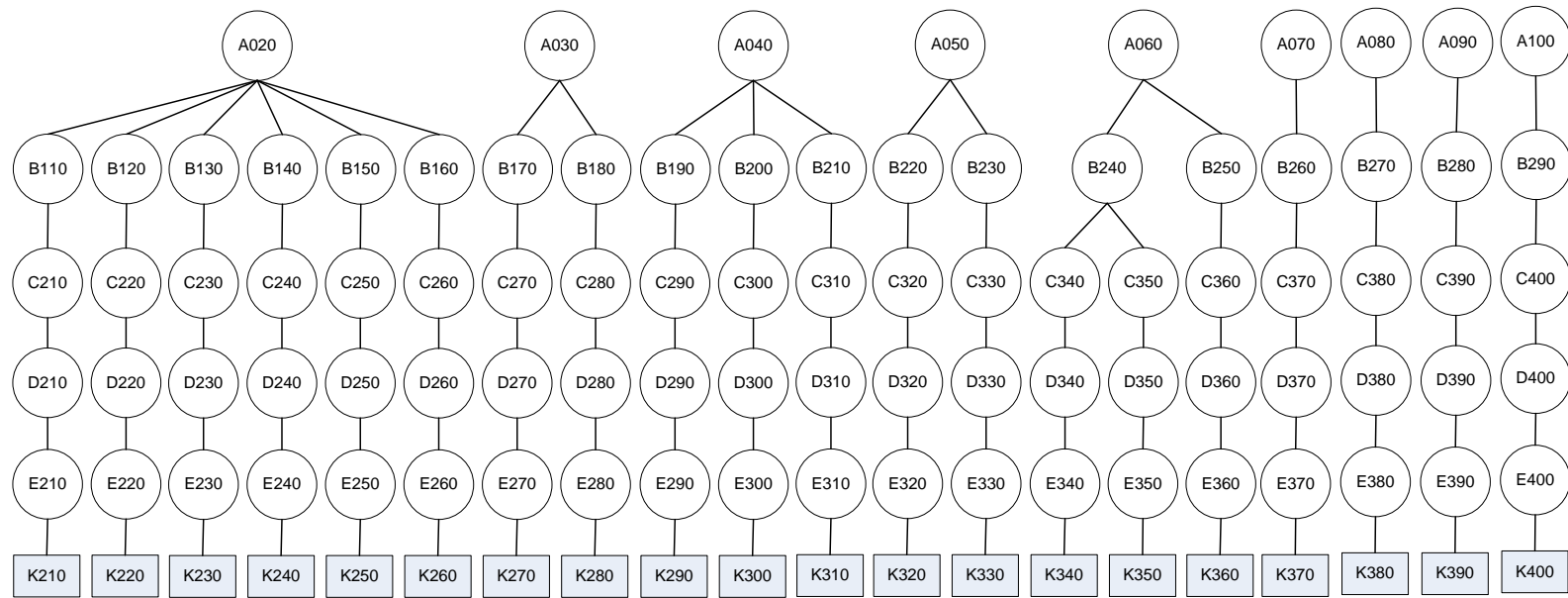
A010 : Ada Faktor Genetik
 A020 : Terjadi Gatal - Gatal pada Kulit
 B010 : Sering terkena sinar matahari
 B020 : Produksi Minyak berlebih
 B030 : sering kurang tidur
 B040 : terjadi karena disfungsi Endokrin
 B050 : berwarna coklat, biru abu-abu, coklat abu-abu
 B060 : infeksi jamur

B070 : sering dijumpai pada anak
 B080 : timbul asma
 B090 : terdapat Benjolan Kenyal Seperti Karet, licin, nyeri saat disentuh
 B100 : gatal terutama terasa pada malam hari
 C010 : terjadi kemerahan pada kulit muka
 C020 : memiliki Kebiasaan Merokok
 C030 : pemakaian Kosmetik Mengandung Parfum

C040	: kemerahan pada wajah yang tidak segera menghilang	D130	: ligamen yang mengikat lemak di bawah mata melemah
C050	: pemaparan sinar X yang berlebihan	D140	: disfungsi hati
C060	: benjolan dan jerawat baik kecil padat (papula) ataupun penuh nanah (pustule)	D150	: terjadi pada saat Kehamilan
C070	: benjolan berwarna merah muda atau kemerahan	D160	: Aktivitas kelenjar sebacea yang berlebih pada wajah
C080	: sedang mengalami stress atau menstruasi	D170	: sering berada dilingkungan yang banyak debu atau asap kendaraan
C090	: benjolan berwarna kemerahan	D180	: Alergen makanan seperti susu sapi, telur, ikan laut
C100	: pertumbuhan sel kulit tidak normal	D190	: bentuk benjolan berbatas jelas
C110	: memiliki kebiasaan merokok	D200	: Bagian Kulit Menjadi Kering
C120	: Kurang Menjalani Olahraga	E010	: edema atau pembengkakan pada wajah
C130	: wajah tampak tidak segar, dan terlihat lelah	E020	: Sering Mengalami Dehidrasi
C140	: defisiensi nutrisi	E030	: Menggunakan obat tertentu yang dapat merangsang hiperpigmentasi kulit
C150	: pemakaian kosmetik, obat-obatan	E040	: iritasi sekitar pipi
C160	: penebalan dan pengerasan kulit	E050	: muncul pada usia di atas 40 tahun
C170	: Kering pada pipi	E060	: penebalan kulit dengan pori-pori yang besar (rhinophyma);
C180	: demam hay atau demam jerami	E070	: penggunaan obat-obatan kortikosteroid, seperti prednisone, dexametason, hydrocortisone
C190	: lama kelamaan benjolan menjadi keras	E080	: banyak mengkonsumsi makanan berminyak, berlemak, gorengan, keju, susu full cream
C200	: Kemerahan muncul pada wajah	E090	: Benjolan Menyebar Disekujur Wajah
D010	: pembesaran dan pelebaran pembuluh darah.	E100	: infeksi bakteri yang cukup serius
D020	: memiliki kebiasaan minuman beralkohol	E110	: Kebiasaan Tidur Miring atau tengkurap
D030	: Ada Bekas Jerawat	E120	: Terlalu Banyak Konsumsi Makanan Asin
D040	: timbul bintil-bintil merah yang menandakan adanya pembuluh darah yang pecah	E130	: elastisitas kulit yang sangat berkurang
D050	: tumor berawal sebagai benjolan licin yang sangat kecil	E140	: berwarna coklat, biru abu-abu, coklat abu-abu
D060	: kekeringan kulit wajah secara berlebihan	E150	: memakai Kontrasepsi Hormonal
D070	: pemakaian kosmetik yang mengandung minyak	E160	: kelainan imunokompromais dan kelainan neurologis
D080	: sering berada dilingkungan yang banyak debu atau asap kendaraan	E170	: gatal-gatal pada Kulit
D090	: Memiliki Benjolan Besar – Besar	E180	: Terjadi Peradangan kulit
D100	: Disertai Peradangan Hebat	E190	: bentuk benjolan tidak beraturan
D110	: Sering Mengalami Dehidrasi	E200	: timbul keropeng dan bersisik
D120	: Kurang Minum Air Putih		

K010 : masalah kulit wajah adalah rosacea
K020 : masalah kulit wajah adalah kerutan di wajah
K030 : masalah kulit wajah adalah Flek hitam
K040 : masalah kulit wajah adalah rosacea
K050 : masalah kulit wajah adalah Karsinoma sel Basal
K060 : masalah kulit wajah adalah rosacea
K070 : masalah kulit wajah adalah Jerawat biasa
K080 : masalah kulit wajah adalah Jerawat biasa
K090 : masalah kulit wajah adalah Jerawat Batu
K100 : masalah kulit wajah adalah Jerawat Batu

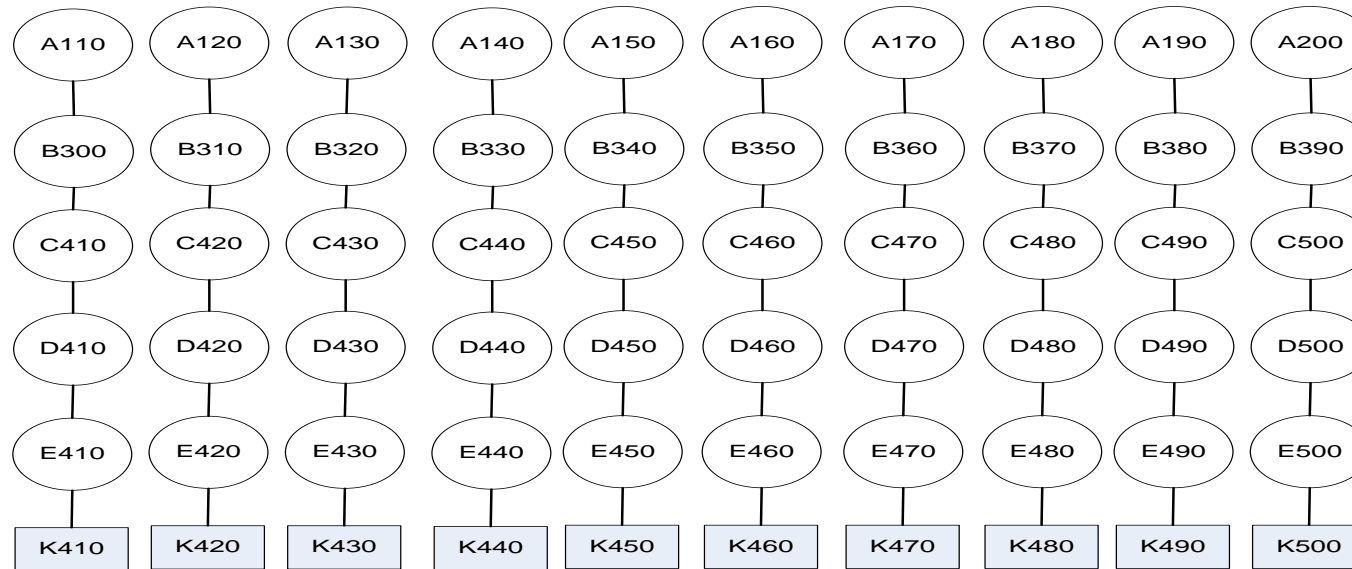
K110 : masalah kulit wajah adalah kerutan di wajah
K120 : masalah kulit wajah adalah Kantung Mata
K130 : masalah kulit wajah adalah Kantung Mata
K140 : masalah kulit wajah adalah Melasma
K150 : masalah kulit wajah adalah Melasma
K160 : masalah kulit wajah adalah Dermatitis seboroik
K170 : masalah kulit wajah adalah Dermatitis atopik
K180 : masalah kulit wajah adalah Dermatitis atopik
K190 : masalah kulit wajah adalah keloid
K200 : masalah kulit wajah adalah eksim



A030	:	produksi minyak berlebih	C240	:	Bercak Berwarna Putih atau coklat kehitaman atau merah
A040	:	terjadi edema	C250	:	konsumsi obat-obatan antibiotik
A050	:	Penonjolan padat (papul)	C260	:	Warnanya bertambah gelap
A060	:	Munculnya bercak Kecoklatan	C270	:	Pori - pori Berwarna Hitam
A070	:	memiliki Berbentuk Memanjang	C280	:	Memiliki Tonjolan Kecil – Kecil
A080	:	terjadi bercak putih	C290	:	Berbatas Tegas, Pinggirannya Mengering
A090	:	tidak Enak Badan Menggigil, Lemah, lesu, demam, cepat lelah, sulit berkemih, gatal singkat	C300	:	mengenai epidermis dan dermis
A100	:	nyeri saraf	C310	:	Perubahan cuaca dan temperature secara tiba-tiba.
B110	:	permukaan Kulit Melepuh	C320	:	Koreng berwarna kuning madu
B120	:	dan adanya bercak pada kulit	C330	:	bentuknya bundar atau tidak beraturan
B130	:	berwarna kemerahan	C340	:	Perlahan akan membesar
B140	:	kulit berkeriat	C350	:	Wajah Berubah pucat
B150	:	berukuran kurang dari 1 milimeter sampai dengan lebih dari 1 sentimeter	C360	:	Berukuran lebih dari setengah inci
B160	:	Ukurannya bertambah besar	C370	:	Dan Bintik – Bintik Halus
B170	:	Pori - Pori Wajah Membesar	C380	:	seukuran koin sampai plak
B180	:	Ada Lapisan Kulit Yang Tumbuh Diatas Penyumbatan	C390	:	Timbul Bintik - Bintik kecil Berwarna Merah
B190	:	Eritematosa Berwarna Cerah	C400	:	kulit sekitarnya menjadi merah meradang
B200	:	Disertai radang akut	D210	:	kulit mengering dan timbul keropeng
B210	:	rasa terbakar dan gatal pada mata	D220	:	menimbulkan warna kecoklatan
B220	:	Kemerahan Pada kulit (makula)	D230	:	setiap lesi bertahan beberapa jam
B230	:	Permukaan yang kasar	D240	:	Terdapat Bercak Bercak Pada kulit
B240	:	Ukurannya bervariasi	D250	:	kelembaban yang tinggi
B250	:	Berbentuk oval	D260	:	Peradangan
B260	:	Bintik berwarna Kuning Kecoklatan	D270	:	Terdapat Penumpukan Sel Kulit Mati
B270	:	Bersisik halus	D280	:	Tonjolan Berwarna Putih
B280	:	Kulit Terasa Nyeri dan Kesemutan	D290	:	Disertai Radang Akut
B290	:	infeksi kulit	D300	:	mengenai subkutis
C210	:	keluar cairan kadang bernanah	D310	:	rendahnya kadar asam dalam lambung
C220	:	Dan bercak mengalami likenifikasi	D320	:	disekitar luka berwarna kemerahan dan basah
C230	:	reaksi simpang obat (RSO)	D330	:	berwarna keabuan, kuning atau coklat
			D340	:	Posisinya terangkat dari Permukaan kulit
			D350	:	Terlihat berlapis seperti lilin
			D360	:	Bergerombol

D370 : terjadi peningkatan pertumbuhan jaringan kulit lokal
 D380 : berbentuk bulat
 D390 : Bintik-Bintik Berubah Menjadi Gelembung transparan Berisi Air
 D400 : peradangan sebagian atau seluruh bagian mata
 E210 : Terbentuk Semacam Kantong Tipis Lunak
 E220 : Dan Kecemasan, depresi ataupun gangguan psikis lainnya
 E230 : umumnya tidak terjadi melebihi 24 jam
 E240 : Gajela subyektif yang dirasakan hampir tidak ada
 E250 : suhu udara yang panas
 E260 : Perubahan warna dalam bentuk bintik-bintik
 E270 : Berhubungan langsung dengan udara di luar sehingga terjadi oksidasi dan pigmentasi
 E280 : Tidak berhubungan dengan udara di luar
 E290 : Demam, mual dan muntah
 E300 : kelainan kulit berupa infiltrat difus di subkutan
 E310 : intoleransi terhadap jenis makanan tertentu
 E320 : Muncul pada musim panas
 E330 : memiliki garis tengah kurang dari 1 cm
 E340 : terpapar sinar matahari
 E350 : Bersisik
 E360 : Tampak menonjol
 E370 : tidak menimbulkan rasa sakit dan nyeri
 E380 : muncul pada usia antara 3 dan 16 tahun
 E390 : terasa sakit seperti terbakar dikulit
 E400 : kelemahan otot

K210 : masalah kulit wajah adalah Impetigo
 K220 : masalah kulit wajah adalah Likens Simpleks
 K230 : masalah kulit wajah adalah Urtikaria
 K240 : masalah kulit wajah adalah Pitriyasis Versikolor
 K250 : masalah kulit wajah adalah Pitriyasis Versikolor
 K260 : masalah kulit wajah adalah nevi
 K270 : masalah kulit wajah adalah komedo terbuka
 K280 : masalah kulit wajah adalah komedo tertutup
 K290 : masalah kulit wajah adalah Erisipelas
 K300 : masalah kulit wajah adalah selulitis
 K310 : masalah kulit wajah adalah rosacea
 K320 : masalah kulit wajah adalah Impetigo Krustosa
 K330 : masalah kulit wajah adalah Veruka Vulgaris
 K340 : masalah kulit wajah adalah Melanoma Maligna Lentigo
 K350 : masalah kulit wajah adalah Seborrheic keratosis
 K360 : masalah kulit wajah adalah Age Spots
 K370 : masalah kulit wajah adalah Veruka Filiformis
 K380 : masalah kulit wajah adalah Pityriasis Alba
 K390 : masalah kulit wajah adalah Herpes Zoster
 K400 : masalah kulit wajah adalah Komplikasi Herpes Zoster



A110 : tumbuh dalam suatu kelompok bintik-bintik halus
 A120 : berbentuk bulat dan datar
 A130 : ditemukan papul-papul berbentuk kubah
 A140 : terbentuknya makula, papul
 A150 : ditandai dengan lesi target yang khas
 A160 : papul berkelompok
 A170 : Makula kecoklatan
 A180 : Jika Sedang Menjalani Diet Ketat
 A190 : Jika terjadi gatal, nyeri, estetis
 A200 : Jika terjadi sensitisasi zat kimia dalam kosmetik
 B300 : berwarna kuning-coklat di wajah

B310 : umumnya berwarna kekuningan, kecoklatan atau warna kuning
 B320 : berwarna putih seperti lilin
 B330 : plak eritematosa berbentuk koin/logam
 B340 : disertai berbagai lesi seperti makula, papul, vesikel dan bula
 B350 : skuama juga hiperpigmentasi
 B360 : sering terkena matahari
 B370 : Usia Mulai Memasuki 35 tahun
 B380 : terdapat pada wajah, leher
 B390 : pigmentasi berwarna coklat keabuan pada wajah
 C410 : tidak menimbulkan rasa sakit dan nyeri

C420 : papul kecil berukuran 1-3 mm
 C430 : lekukan pada permukaannya (delle)
 C440 : berbatas tegas dan tertutup skuama melekat meluas
 C450 : bersifat ringan
 C460 : reaksi simpang obat (RSO)
 C470 : berukuran kurang 0.5 cm
 C480 : Sering Stres
 C490 : komedo tertutup
 C500 : pigmentasi berwarna coklat keabuan pada pelipis, dahi
 D410 : bersifat swasirna
 D420 : biasanya tidak menimbulkan rasa nyeri
 D430 : berisi massa yang mengandung badan moluskum
 D440 : pada tepi lesi tampak eritema dan hiperpigmentasi
 D450 : swasirna
 D460 : sensitisasi berlangsung secara topikal atau tanpa sensitisasi topikal sebelumnya
 D470 : warna bertambah jelas bila terpajan sinar matahari
 D480 : Kelelahan atau Kurang Tidur
 D490 : komedo terbuka
 D500 : penggunaan pemutih
 E410 : Ditemukan pada anak-anak dan dewasa
 E420 : terjadi pada anak-anak dan remaja
 E430 : Ditemukan pada anak-anak dan dewasa
 E440 : terjadi pada usia 20 -40 tahun
 E450 : tidak melibatkan lebih dari satu mukosa
 E460 : mirip dengan dermatitis akut
 E470 : muncul pada masa kanak-kanak dan semakin bertambah pada musim panas
 E480 : terjadi radikal bebas
 E490 : peradangan kulit
 E500 : penggunaan formaldehida, anilinde dyes
 K410 : masalah kulit wajah adalah Kutil Datar

K420 : masalah kulit wajah adalah Veruka plana
 K430 : masalah kulit wajah adalah moluskum contagiosum
 K440 : masalah kulit wajah adalah Lupus Eritematosus Diskoid
 K450 : masalah kulit wajah adalah Eritema Multiforme
 K460 : masalah kulit wajah adalah Dermatitis Medikamentosa
 K470 : masalah kulit wajah adalah Efelid
 K480 : masalah kulit wajah adalah Kulit Wajah Mengendur
 K490 : masalah kulit wajah adalah Akne Vulgaris
 K500 : masalah kulit wajah adalah Melanosis Riehl

Lampiran 2

Rule Sistem Pakar Mendiagnosa Penyakit Kulit Wajah

Rule 1

Jika ada faktor genetik

Dan sering terkena sinar matahari

Dan terjadi kemerahan pada kulit muka

Dan pembesaran dan pelebaran pembuluh darah.

Dan edema atau pembengkakan pada wajah

Maka masalah kulit wajah adalah Rosacea

Solusi :

Bersihkan dengan lembut beberapa kali sehari. Gunakan pembersih yang lembut dan menghindari pembersih muka yang kasar sehingga dapat menyebabkan iritasi. Pakailah tabir surya yang lembut.

Rule 2

Jika Ada Faktor Genetik

Dan Sering Terkena Sinar Matahari

Dan Memiliki Kebiasaan Merokok

Dan memiliki kebiasaan minuman beralkohol

Dan Sering Mengalami Dehidrasi

Maka masalah kulit wajah adalah Kerutan di wajah

Solusi :

Menghindari kulit dari terpapar sinar matahari secara berlebihan, menghindari rokok, tidur 7 hingga 8 jam setiap hari.

Rule 3

Jika Ada Faktor Genetik

Dan sering terkena Sinar Matahari

Dan pemakaian kosmetik Mengandung Parfum

Dan ada Bekas Jerawat

Dan menggunakan obat tertentu yang dapat merangsang hiperpigmentasi kulit

Maka masalah kulit wajah adalah flek hitam

Solusi :

Siapkan 1 buah jeruk nipis belah menjadi 2 bagian. Gosokkan dengan lembut ke wajah.

Rule 4

Jika Ada Faktor Genetik

Dan sering terkena sinar matahari

Dan kemerahan pada wajah yang tidak segera menghilang

Dan timbul bintil-bintil merah yang menandakan adanya pembuluh darah yang pecah

Dan iritasi sekitar pipi

Maka masalah kulit wajah adalah Rosacea

Solusi :

Bersihkan dengan lembut beberapa kali sehari. Gunakan pembersih yang lembut dan menghindari pembersih muka yang kasar sehingga dapat menyebabkan iritasi. Pakailah tabir surya yang lembut.

Rule 5

Jika Ada Faktor genetik

Dan sering terkena sinar matahari

Dan pemaparan sinar X yang berlebihan

Dan tumor berawal sebagai benjolan licin yang sangat kecil

Dan muncul pada usia di atas 40 tahun

Maka masalah kulit wajah adalah Karsinoma sel Basal

Solusi :

Dapat dilakukan radiasi tumor, bedah beku dengan N2 cair.

Rule 6

Jika Ada Faktor Genetik

Dan Sering terkena sinar matahari

Dan benjolan dan jerawat baik kecil padat (papula) ataupun penuh nanah (pustule)

Dan kekeringan kulit wajah secara berlebihan

Dan penebalan kulit dengan pori-pori yang besar (rhinophyma);

Maka masalah kulit wajah adalah rosacea

Solusi :

Bersihkan dengan lembut beberapa kali sehari. Gunakan pembersih yang lembut dan hindari pembersih muka yang kasar sehingga dapat menyebabkan iritasi. Pakailah tabir surya yang lembut.

Rule 7

Jika ada faktor genetik

Dan produksi Minyak berlebih

Dan benjolan Berwarna Merah muda atau kemerahan

Dan pemakaian kosmetik yang mengandung minyak

Dan penggunaan obat-obatan kortikosteroid, seperti prednisone, dexametason, hydrocortisone

Maka masalah kulit wajah adalah jerawat biasa

Solusi :

Jauhi semua factor yang dapat memperparah kondisi jerawat seperti mengkonsumsi makanan berminyak, berlemak dan pedas, jagalah kebersihan wajah, gunakan krim wajah, dan rajin berolahraga

Rule 8

Jika ada faktor genetik

Dan produksi Minyak berlebih

Dan sedang mengalami stress atau menstruasi

Dan sering berada dilingkungan yang banyak debu atau asap kendaraan

Dan banyak mengkonsumsi makanan berminyak, berlemak, gorengan, keju, susu full cream

Maka masalah kulit wajah adalah jerawat biasa

Solusi :

Jauhi semua factor yang dapat memperparah kondisi jerawat seperti mengkonsumsi makanan berminyak, berlemak dan pedas, jagalah kebersihan wajah, gunakan krim wajah, dan rajin berolahraga.

Rule 9

Jika ada Faktor Genetik

Dan produksi Minyak berlebih

Dan benjolan Berwarna kemerahan

Dan memiliki Benjolan Besar – Besar

Dan benjolan Menyebar Disekujur Wajah

Maka masalah kulit wajah adalah jerawat batu

Solusi :

Menjauhi semua makanan pemicu jerawat, hati – hati menggunakan kosmetik, menjaga kebersihan wajah, dan rajin berolahraga.

Rule 10

Jika Ada Faktor Genetik

Dan produksi Minyak berlebih

Dan pertumbuhan sel kulit tidak normal

Dan disertai Peradangan Hebat

Dan infeksi bakteri yang cukup serius

Maka masalah kulit wajah adalah jerawat batu

Solusi :

Menjauhi semua makanan pemicu jerawat, hati – hati menggunakan kosmetik, menjaga kebersihan wajah, dan rajin berolahraga

Rule 11

Jika Ada Faktor Genetik

Dan sering kurang tidur

Dan Memiliki Kebiasaan Merokok

Dan Sering Mengalami Dehidrasi

Dan kebiasaan tidur Miring atau tengkurap

Maka masalah kulit wajah adalah kerutan di wajah

Solusi :

Menghindari kulit dari terpapar sinar matahari secara berlebihan, menghindari rokok, tidur 7 hingga 8 jam setiap hari.

Rule 12

Jika Ada Faktor Genetik

Dan sering kurang tidur

Dan kurang Menjalani Olahraga

Dan kurang Minum Air Putih

Dan Terlalu Banyak Konsumsi Makanan Asin

Maka masalah kulit wajah adalah kantung mata

Solusi :

Tidur 7 hingga 8 jam setiap hari, menggunakan krim mata yang berfungsi untuk mengurangi kantung mata. Terapkan pola makan sehat.

Rule 13

Jika Ada Faktor Genetik

Dan sering kurang tidur

Dan wajah tampak tidak segar, dan terlihat lelah

Dan ligamen yang mengikat lemak di bawah mata melemah

Dan elastisitas kulit yang sangat berkurang

Maka masalah kulit wajah adalah kantung mata

Solusi :

Tidur 7 hingga 8 jam setiap hari, menggunakan krim mata yang berfungsi kantung mata.
 Terapkan pola makan sehat.

Rule 14

Jika ada faktor genetik

Dan terjadi karena disfungsi Endokrin

Dan defisiensi nutrisi

Dan disfungsi hati

Dan berwarna coklat, biru abu-abu, coklat abu-abu

Maka masalah kulit wajah adalah melasma

Solusi :

Hindari sinar matahari langsung (07.00 s/d 16.00), gunakan tabir surya

Rule 15

Jika ada faktor genetik

Dan berwarna coklat, biru abu-abu, coklat abu-abu

Dan pemakaian kosmetik, obat-obatan

Dan terjadi pada saat Kehamilan

Dan memakai Kontrasepsi Hormonal

Maka masalah kulit wajah adalah melasma

Solusi :

Hindari sinar matahari langsung (07.00 s/d 16.00), gunakan tabir surya

Rule 16

Jika ada Faktor Genetik

Dan infeksi jamur

Dan penebalan dan pengerasan kulit

Dan aktifitas kelenjar sebacea yang berlebih pada wajah

Dan kelainan imunokompromais dan kelainan neurologis

Maka masalah kulit wajah adalah Dermatitis seboroik

Solusi :

Pengobatan topikal dapat diberikan kortikosteroid seperti hidrokortison krim 2 kali sehari atau pemberian sulfur presipitatum 4-20%, resorsin 1-3%, likuor karbonas detergens 2-5%, serta golongan azol.

Rule 17

Jika ada faktor genetik

Dan sering dijumpai pada anak

Dan Kering pada pipi

Dan sering berada dilingkungan yang banyak debu atau asap kendaraan

Dan gatal-gatal pada Kulit

Maka masalah kulit wajah adalah Dermatitis atopik

Solusi :

Menjaga kelembapan kulit dengan menggunakan losion atau krim pelembab, kompres dingin bagian kulit yang terkena, terutama pada bagian kulit yang terasa sangat kering dan gatal.

Rule 18

Jika ada faktor genetik

Dan timbul asma

Dan demam hay atau demam jerami

Dan alergen makanan seperti susu sapi, telur, ikan laut

Dan Terjadi Peradangan kulit

Maka masalah kulit wajah adalah Dermatitis atopik

Solusi :

Menjaga kelembapan kulit dengan menggunakan losion atau krim pelembab, kompres dingin bagian kulit yang terkena, terutama pada bagian kulit yang terasa sangat kering dan gatal.

Rule 19

Jika Terjadi Gatal - Gatal pada Kulit

Dan Terdapat Benjolan Kenyal Seperti Karet,licin,nyeri saat disentuh

Dan lama kelamaan benjolan menjadi keras

Dan bentuk benjolan berbatas jelas

Dan bentuk benjolan tidak beraturan

Maka masalah kulit wajah adalah keloid

Solusi :

penyuntikan langsung pada permukaan keloid, bedah beku menggunakan nitrogen cair, dengan sinar laser, metode-metode ini dilakukan oleh dokter ahli

Rule 20

Jika Terjadi Gatal - Gatal pada Kulit

Dan gatal terutama terasa pada malam hari

Dan kemerahan muncul pada wajah

Dan bagian Kulit Menjadi Kering

Dan timbul keropeng dan bersisik

Maka masalah kulit wajah adalah eksim

Solusi :

Menjaga kelembapan kulit dengan menggunakan losion atau krim pelembab, kompres dingin bagian kulit yang terkena, terutama pada bagian kulit yang terasa sangat kering dan gatal.

Rule 21

Jika Terjadi Gatal - Gatal pada Kulit

Dan permukaan Kulit Melepuh

Dan keluar cairan kadang bernanah

Dan kulit mengering dan timbul keropeng

Dan terbentuk Semacam Kantong Tipis Lunak

Maka masalah kulit wajah adalah Impetigo

Solusi :

Menggunakan antibiotik, baik dalam beluk salep yang dioleskan maupun antibiotik yang diminum. Pengobatan ini harus dilakukan dengan petunjuk dan resep dokter.

Rule 22

Jika Terjadi Gatal - Gatal pada Kulit

Dan adanya bercak pada kulit

Dan bercak mengalami likenifikasi

Dan menimbulkan warna kecoklatan

Dan Kecemasan, depresi ataupun gangguan psikis lainnya

Maka masalah kulit wajah adalah Likens Simpleks

Solusi :

Oleskan salap coal tar atau pasta zinci yang mengandung coal tar pada malam hari untuk mengurangi rasa gatal.

Rule 23

Jika Terjadi Gatal - Gatal pada Kulit

Dan berwarna kemerahan

Dan reaksi simpang obat (RSO)

Dan setiap lesi bertahan beberapa jam

Dan umumnya tidak terjadi melebihi 24 jam

Maka masalah kulit wajah adalah Urtikaria

Solusi :

Berikan antihistamin dan antipruritus topikal.

Rule 24

Jika terjadi gatal - gatal pada kulit

Dan kulit berkerengat

Dan bercak Berwarna Putih atau coklat kehitaman atau merah

Dan terdapat Bercak Bercak Pada kulit

Dan gejala subyektif yang dirasakan hampir tidak ada

Maka masalah kulit wajah adalah Pitriyasis Versikolor

Solusi :

Gunakan obat oles yang mengandung ketokonazol, klotrimazol, atau terbinafin. Penggunaan obat oles ini sebaiknya sekitar dua minggu.

Rule 25

Jika terjadi gatal - gatal pada kulit

Dan berukuran kurang dari 1 milimeter sampai dengan lebih dari 1 sentimeter

Dan konsumsi obat-obatan antibiotik

Dan kelembaban yang tinggi

Dan suhu udara yang panas

Maka masalah kulit wajah adalah Pitriyasis Versikolor

Solusi :

Gunakan obat oles yang mengandung ketokonazol, klotrimazol, atau terbinafin. Penggunaan obat oles ini sebaiknya sekitar dua minggu.

Rule 26

Jika terjadi gatal - gatal pada kulit

Dan ukurannya bertambah besar

Dan warnanya bertambah gelap

Dan Peradangan

Dan Perubahan warna dalam bentuk bintik-bintik

Maka masalah kulit wajah adalah nevi

Solusi :

operasi eksisi. Nevi yang menyeramkan diangkat lebih awal untuk mengurangi ketakutan fisik maupun emosional.

Rule 27

Jika Produksi Minyak berlebih

Dan pori - pori wajah membesar

Dan pori - pori berwarna hitam

Dan terdapat penumpukan sel kulit mati

Dan berhubungan langsung dengan udara di luar sehingga terjadi oksidasi dan pigmentasi

Maka masalah kulit wajah adalah komedo terbuka

Solusi :

Bersihkan wajah secara teratur menggunakan pembersih atau sabun muka yang sesuai dengan jenis kulit.

Rule 28

Jika produksi minyak berlebih

Dan Ada Lapisan Kulit Yang Tumbuh Diatas Penyumbatan

Dan Memiliki Tonjolan Kecil – Kecil

Dan Tonjolan Berwarna Putih

Dan tidak berhubungan dengan udara di luar

Maka masalah kulit wajah adalah komedo tertutup

Solusi :

Bersihkan wajah secara teratur menggunakan pembersih atau sabun muka yang sesuai dengan jenis kulit.

Rule 29

Jika Terjadi Edema

Dan Eritematosa Berwarna Cerah

Dan Berbatas Tegas, Pinggirannya Mengering

Dan Disertai Radang Akut

Dan demam, mual dan muntah

Maka masalah kulit wajah adalah Erisipelas

Solusi :

Banyak istirahat, juga dapat diberikan kompres terbuka dengan larutan antiseptik.

Rule 30

Jika terjadi edema

Dan disertai radang akut

Dan mengenai epidermis dan dermis

Dan mengenai subkutis

Dan kelainan kulit berupa infiltrat difus di subkutan

Maka masalah kulit wajah adalah selulitis

Solusi :

Banyak minum air putih, olahraga, membatasi konsumsi makanan tinggi kalori, menggunakan krim.

Rule 31

Jika terjadi edema

Dan rasa terbakar dan gatal pada mata

Dan perubahan cuaca dan temperature secara tiba-tiba.

Dan rendahnya kadar asam dalam lambung

Dan intoleransi terhadap jenis makanan tertentu

Maka masalah kulit wajah adalah rosacea

Solusi :

Bersihkan dengan lembut beberapa kali sehari. Gunakan pembersih yang lembut dan menghindari pembersih muka yang kasar sehingga dapat menyebabkan iritasi. Pakailah tabir surya yang lembut.

Rule 32

Jika ada penonjolan padat (papul)

Dan kemerahan Pada kulit (makula)

Dan Koreng berwarna kuning madu

Dan ada disekitar luka berwarna kemerahan dan basah

Dan Muncul pada musim panas

Maka masalah kulit wajah adalah Impetigo Krustosa

Solusi :

Membersihkan luka yang lecet atau mengalami pengausan secara perlahan-lahan. Tidak boleh melakukan gosokan-gosokan pada luka terlalu dalam.

Rule 33

Jika ada Penonjolan padat (papul)

Dan Permukaan yang kasar

Dan bentuknya bundar atau tidak beraturan

Dan berwarna keabuan, kuning atau coklat

Dan memiliki garis tengah kurang dari 1 cm

Maka masalah kulit wajah adalah Verruca Vulgaris

Solusi :

Oleskan larutan untuk mempercepat hilangnya atau plester yang mengandung asam salisilat dan asam laktat.

Rule 34

Jika Munculnya bercak Kecoklatan

Dan Ukurannya bervariasi

Dan Perlahan akan membesar

Dan Posisinya terangkat dari Permukaan kulit

Dan terpapar sinar matahari

Maka masalah kulit wajah adalah Melanoma Maligna Lentigo

Solusi :

Harus menjalani bedah eksisi.

Rule 35

Jika Munculnya bercak Kecoklatan

Dan Ukurannya bervariasi

Dan Wajah Berubah pucat

Dan Terlihat berlapis seperti lilin

Dan Bersisik

Maka masalah kulit wajah adalah Seborrheic keratosis

Solusi :

Perawatan Seborrheic keratosis biasanya tidak diperlukan. Bisa saja dihilangkan jika penderita menginginkannya.

Rule 36

Jika Munculnya bercak Kecoklatan

Dan Berbentuk oval

Dan berukuran lebih dari setengah inci

Dan Bergerombol

Dan Tampak menonjol

Maka masalah kulit wajah adalah Age Spots

Solusi :

Krim pemutihan Resep atau dengan retinoid dan steroid ringan secara bertahap dapat memudahkan tempat munculnya age spots selama beberapa bulan.

Rule 37

Jika memiliki Berbentuk Memanjang

Dan Bintik berwarna Kuning Kecoklatan

Dan Bintik – Bintik Halus

Dan terjadi peningkatan pertumbuhan jaringan kulit lokal

Dan tidak menimbulkan rasa sakit dan nyeri

Maka masalah kulit wajah adalah Verruca Filiformis

Solusi :

Membekukan dengan menggunakan cairan nitrogen yang dilakukan berulang-ulang (bedah beku), meneteskan bahan kimia yang bersifat kaustik (membakar), pengobatan dengan arus listrik, dan bedah laser.

Rule 38

Jika terjadi bercak putih

Dan Bersisik halus

Dan seukuran koin sampai plak

Dan berbentuk bulat

Dan muncul pada usia antara 3 dan 16 tahun

Maka masalah kulit wajah adalah Pityriasis Alba

Solusi :

Dapat diberikan tabir surya, pengobatan simptomatik dengan oleum bergamot 15% yang dioleskan pada lesi di pagi hari, atau kortikosteroid potensi ringan.

Rule 39

Jika tidak enak badan menggigil, lemah, lesu, demam, cepat lelah, sulit berkemih, gatal singkat

Dan Kulit Terasa Nyeri dan Kesemutan

Dan Timbul Bintik - Bintik kecil Berwarna Merah

Dan Bintik - Bintik Berubah Menjadi Gelembung transparan Berisi Air

Dan terasa sakit seperti terbakar di kulit

Maka masalah kulit wajah adalah Herpes Zoster

Solusi :

Pemberian antinyeri, penurun panas, atau obat yang mengurangi rasa gatal pada periode masa penyembuhan.

Rule 40

Jika nyeri saraf

Dan infeksi kulit

Dan kulit sekitarnya menjadi merah meradang

Dan peradangan sebagian atau seluruh bagian mata

Dan terjadi kelemahan otot

Maka masalah kulit wajah adalah Komplikasi Herpes Zoster

Solusi :

Pemberian antinyeri, penurun panas, atau obat yang mengurangi rasa gatal pada periode masa penyembuhan. Mengonsumsi buah-buahan yang banyak mengandung vitamin C.

Rule 41

Jika tumbuh dalam suatu kelompok bintik-bintik halus

Dan berwarna kuning-coklat di wajah

Dan tidak menimbulkan rasa sakit dan nyeri

Dan bersifat swasirna

Dan Ditemukan pada anak-anak dan dewasa

Maka masalah kulit wajah adalah Kutil Datar

Solusi :

Diobati dengan pemberian zat pengelupas, misalnya asam retinoat atau asam salisilat.

Rule 42

Jika berbentuk bulat dan datar

Dan umumnya berwarna kekuningan, kecoklatan atau warna kuning

Dan papul kecil berukuran 1-3 mm

Dan Biasanya tidak menimbulkan rasa nyeri

Dan terjadi pada anak-anak dan remaja

Maka masalah kulit wajah adalah Verruca plana

Solusi :

Dihaluskan dengan larutan atau plester yang mengandung asam salisilat yang kuat.

Rule 43

Jika ditemukan papul-papul berbentuk kubah

Dan berwarna putih seperti lilin

Dan lekukan pada permukaannya (delle)

Dan berisi massa yang mengandung badan moluskum

Dan Ditemukan pada anak-anak dan dewasa

Maka masalah kulit wajah adalah moluskum contagiosum

Solusi :

Eukleasi menggunakan jarum dan pinset. Solusio kantaridin. Elektrokauterisasi. Bedah buku dengan CO₂, N₂O.

Rule 44

Jika terbentuknya makula, papul

Dan plak eritematosa berbentuk koin/logam

Dan berbatas tegas dan tertutup skuama melekat meluas

Dan pada tepi lesi tampak eritema dan hiperpigmentasi

Dan terjadi pada usia 20 -40 tahun

Maka masalah kulit wajah adalah Lupus Eritematosus Diskoid

Solusi :

menghindari paparan sinar ultraviolet dari matahari atau sumber artifisial lain dengan cara menggunakan pakaian dari bahan tenunan padat, serta tabir surya spektrum luas dengan SPF minimal 15.

Rule 45

Jika ditandai dengan lesi target yang khas

Dan disertai berbagai lesi seperti makula, papul, vesikel dan bula

Dan bersifat ringan

Dan swarsina

Dan tidak melibatkan lebih dari satu mukosa

Maka masalah kulit wajah adalah Eritema Multiforme

Solusi :

Terapi simptomatik, namun dapat juga diberikan kortikosteroid sistemik jangka pendek misalnya prednison 30mg/hari selama 5-10 hari.

Rule 46

Jika papul berkelompok

Dan skuama juga hiperpigmentasi

Dan reaksi simpan obat (RSO)

Dan sensitisasi berlangsung secara topikal atau tanpa sensitisasi topikal sebelumnya

Dan mirip dengan dermatitis akut

Maka masalah kulit wajah adalah Dermatitis Medikamentosa

Solusi :

Pengobatan topikal dapat diberikan kortikosteroid seperti hidrokortison krim 2 kali sehari atau pemberian sulfur presipitatum 4-20%, resorsin 1-3%, likuor karbonas detergens 2-5%, serta golongan azol.

Rule 47

Jika Makula kecoklatan

Dan sering terkena matahari

Dan berukuran kurang 0.5 cm

Dan warna bertambah jelas bila terpajan sinar matahari

Dan muncul pada masa kanak-kanak dan semakin bertambah pada musim panas

Maka masalah kulit wajah adalah Efelid

Solusi :

Hindari sinar matahari langsung, gunakan tabir surya, hilangkan/kurangi faktor etiologi.

Rule 48

Jika Sedang Menjalani Diet Ketat

Dan Usia Mulai Memasuki 35 tahun

Dan Sering Stres

Dan Kelelahan atau Kurang Tidur

Dan terjadi radikal bebas

Maka masalah kulit wajah adalah Kulit Wajah Mengendur

Solusi :

Menggunakan pelembap dengan kualitas tinggi secara teratur setiap pagi dan malam hari. Menggunakan krim khusus untuk perawatan daerah sekitar mata.

Rule 49

Jika terjadi gatal, nyeri, estetis

Dan terdapat pada wajah, leher

Dan komedo tertutup

Dan Komedo terbuka

Dan peradangan kulit

Maka masalah kulit wajah adalah Akne Vulgaris

Solusi:

Pengobatan topikal, kombinasi antara Benzoil Peroksida (BP) dengan antibiotik topikal seperti erithromisin atau klindamisin.

Rule 50

Jika terjadi sensitisasi zat kimia dalam kosmetik

Dan pigmentasi berwarna coklat keabuan pada wajah

Dan pigmentasi berwarna coklat keabuan pada pelipis, dahi

Dan penggunaan pemutih

Dan penggunaan formaldehida, aniline dyes

Maka masalah kulit wajah adalah Melanosis Riehl

Solusi :

Hindari sinar matahari langsung, gunakan tabir surya.

Surat Keterangan
017/IT/MC/06-13

Yang bertanda tangan dibawah ini :

Nama : Yunita Sartika Sari
Jabatan : Bagian Umum



Menerangkan bahwa :

Nama : Yusuf Sofyan
Nim : 0911510923

Telah melaksanakan riset pada Maharisclinic sejak tanggal 15 April 2013 sampai dengan 17 Juni 2013 dengan baik.

Demikian surat keterangan ini dibuat untuk dapat dipergunakan sebagaimana mestinya

Jakarta, 17 Juni 2013


The logo for maharisClinic, featuring the word "maharis" in a blue sans-serif font and "Clinic" in a smaller, lighter blue sans-serif font, with a small blue square icon to the left of "maharis".

Yunita Sartika Sari

1. Tampilan Menu Utama

```
package com.client;

import com.client.MenuLogin;
import com.model.tes;
import com.utils.Consts;
import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;
import javax.microedition.lcdui.Choice;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.List;
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Ticker;
import javax.microedition.lcdui.StringItem;

/**
 *
 */
public class MenuUtama extends List implements CommandListener, Runnable {

    private Display display;
    private MIDlet midlet;
    private Ticker ticker;
    // private Command cChoice = new Command("Pilih", Command.SCREEN, 0);
    private Command cExit = new Command("Keluar", Command.EXIT, 0);
    Timer timer;
    TimerTask timerTask;
    // private StringItem txt ;
    private static MenuUtama mainMenu;

    private MenuUtama(MIDlet midlet, Display display) {
        super("Sistem Pakar - Penyakit wajah", Choice.IMPLICIT);
        ticker = new Ticker("Selamat Datang - Sistem Pakar Untuk Mendiagnosa Penyakit Kulit Wajah");
        this.display = display;
        this.midlet = midlet;
        // txt = new StringItem("tes", null);

        append("Identifikasi", Consts.IDENTIFICATION_IMG);
    }
}
```

```

// append("Buku Tamu", Consts.GBOOK_IMG);
append("Kamus Bahasa", Consts.DICTIONARY_IMG );
append("Admin", Consts.LOGIN_IMG);
append("About Me", Consts.ABOUT_IMG);

setTicker(ticker);

//    addCommand(cChoice);
addCommand(cExit);
setCommandListener(this);

}

public static MenuUtama getInstance(MIDlet midlet, Display display) {
    if (mainMenu == null) {
        mainMenu = new MenuUtama(midlet, display);
    }
    return mainMenu;
}

public static MenuUtama getInstance() {
    if (mainMenu == null) {
        System.out.println("Main menu not created !!!");
    }
    return mainMenu;
}

MenuLogin loginMenu = null;
sebelumidentifikasi identificationMenu = null;
// BukuTamu guideBookMenu = null;
kamusbahasa dictionaryMenu = null;
AboutMe aboutMenu = null;

public void run() {
    switch (getSelectedIndex()) {
        case 2:
            loginMenu = new MenuLogin(midlet, display);
            display.setCurrent(loginMenu);
            break;
        case 0:
            identificationMenu = new sebelumidentifikasi(midlet, display);
            display.setCurrent(identificationMenu);
            break;
        /* case 1:
            if (guideBookMenu == null) {
                guideBookMenu = new BukuTamu(midlet, display);
            }
            display.setCurrent(guideBookMenu);
            break;
        */
    }
}

```

```

        }
        display.setCurrent(guideBookMenu);
        break; */
    case 1:

        dictionaryMenu = new kamusbahasa(midlet, display);

        display.setCurrent(dictionaryMenu);
        break;
    case 3:
        if (aboutMenu == null) {
            aboutMenu = new AboutMe(midlet, display);
        }
        display.setCurrent(aboutMenu);
        break;
    default:
    }
//    if (isSelected(getSelectedIndex())) {
//        if (getSelectedIndex() == 0) {
//            if (loginMenu == null) {
//                loginMenu = new MenuLogin(startMIDlet, dMainMenu);
//            }
//            dMainMenu.setCurrent(loginMenu);
//        }
//    }
//    }
}

public void commandAction(Command c, Displayable d) {

    if (c == this.SELECT_COMMAND) {
        new Thread(this).start();
    }
    if (c == cExit) {
        midlet.notifyDestroyed();
    }
}
}

```

2. Identifikasi

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package com.client;

import com.http.connection;
import com.model.Identification;
import com.model.Message;

```

```

import com.model.Parameter;
import com.utils.Consts;
import com.utils.GlobalUtils;
import java.util.Vector;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.ChoiceGroup;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.ImageItem;
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.List;
import javax.microedition.lcdui.StringItem;
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;

public class identifikasi extends Form implements CommandListener {

    private MIDlet midlet;
    private Display display;
    private StringItem level = new StringItem("Level : ", null);
    private StringItem text = new StringItem(" ", null);
    private StringItem progress = new StringItem(null, null);
    private Command cBack = new Command("Kembali", Command.SCREEN, 0);
    private Command cNext = new Command("Berikutnya", Command.OK, 1);
    private Command cPrev = new Command("Sebelumnya", Command.OK, 2);
    // private Command backDirection = new Command("Back", Command.BACK, 0);
    // private ImageItem level = new ImageItem("test", Consts.ALL_IMG,
    ImageItem.LAYOUT_LEFT, "level", ImageItem.BUTTON);
    private Command cNextPage = new Command("Halaman Selanjutnya", Command.OK,
    1);
    private Command cPrevPage = new Command("Halaman Sebelumnya",
    Command.OK, 1);
    private final ChoiceGroup choiceGroup = new ChoiceGroup(null,
    ChoiceGroup.EXCLUSIVE, new String[]{"Ya", "Tidak"}, null);
    private Vector vector;
    private Vector tempVec;
    private Vector memory = new Vector();
    private Vector lastIndex=new Vector();
    private int memoryAddress = -1;
    private Identification idx;
    private int index = 0;
    private List listIns = new List(null, List.IMPLICIT);
    private int page = 0;
    private int no_page = 1;

```

```

private MenuUtama mainMenu;
private Alert altest;

public identifikasi() {
    super("Menu Identifikasi");
}

private void deepcopy(Vector oldV, Vector newV) {

    int size = newV.size();
    for (int i = 0; i < size; i++) {
        Object object = newV.elementAt(i);
        oldV.addElement(object);
    }
}

public identifikasi(MIDlet midlet, Display display) {
    super("Menu Identifikasi");
    this.display = display;
    this.midlet = midlet;
    initComponents();
    initIns();
}

private class InsCommandListeners implements CommandListener {

    public void commandAction(Command c, Displayable d) {
        if (c == cNextPage) {
            if (no_page < page) {
                no_page++;
                Thread thread = new Thread(nextIns);
                thread.start();
            }
            return;
        }
        if (c == cPrevPage) {
            if (no_page > 1) {
                no_page--;
                Thread thread = new Thread(nextIns);
                thread.start();
            }
            return;
        }
        if (c == cBack) {
            mainMenu = MenuUtama.getInstance();

```

```

        display.setCurrent(mainMenu);
        return;
    }

    if (c == List.SELECT_COMMAND) {
        System.out.println(" ");
        int i = listIns.getSelectedIndex();

        if (listIns.getString(1).startsWith(""))
        {
            return;
        }

        if (i < vector.size()) {
            idx = (Identification) vector.elementAt(listIns.getSelectedIndex());

            if (listIns.size() > vector.size() + 1) {
                listIns.delete(listIns.size() - 1);
            }
            StringBuffer sb = new StringBuffer();
            sb.append("\n").append("Keterangan :").append('\n');
            sb.append(idx.getCode()).append("
-
").append(idx.getDesc()).append("\n").append("
-
Solusinya
Adalah
").append(idx.getSolusi()).append("\n");

//          this.append(sb.toString(), null);

            Alert alert = new Alert("Keterangan", sb.toString(), null, AlertType.INFO);
            alert.setTimeout(Alert.FOREVER);
            display.setCurrent(alert);
        }
    }
    return;
}

}

private void initIns() {
    page = 0;
    no_page = 1;
    listIns.setTitle("Arahan");
    listIns.addCommand(cBack);
    listIns.addCommand(cNextPage);
    listIns.addCommand(cPrevPage);

    listIns.setCommandListener(new InsCommandListeners());

```

```

    }

    private void initComponents() {
        this.setTitle("Pertanyaan");
        this.addCommand(cBack);
        this.addCommand(cNext);
        this.addCommand(cPrev);
        this.setCommandListener(this);
        Thread thread = new Thread(startIdentification);
        thread.start();
        this.append(level);
        this.append(text);
        this.append(choiceGroup);
        this.append(progress);
        GlobalUtils.resetCodeInstruction();
    }

    private String getLevel(String s) {
        char c = Character.toUpperCase(s.charAt(0));
        return String.valueOf((int) c - 64);
    }

    //halaman awal
    private Runnable startIdentification = new Runnable() {

        public void run() {
            try {
                progress.setText(Consts.CONNECT);
                connection hc = new connection();
                hc.setUrl(Consts.IDX_URL);
                //as
                Parameter parameter = new Parameter();
                parameter.add(Consts.PARAM, Consts.START_IDX);

                hc.setParameter(parameter.toString());
                progress.setText(Consts.READ);

                System.out.println(hc.getUrl() + "" + hc.getParameter());
                if (hc.doGet()) {
                    vector = GlobalUtils.parseToIdx(hc.getResponse());
                    // tempVec = new Vector();
                    // deepcopy(tempVec, vector);
                    //
                    // memory.insertElementAt(tempVec, memoryAddress);

                    index = 0;
                    idx = (Identification) vector.elementAt(index);
                    text.setText(" " + idx.getDesc() + " ");
                    String lvl = getLevel(idx.getCode()) + "." + (index + 1);

```

```

        level.setText(lvl);
    } else {

        Alert alert = new Alert(Consts.SERVER_ERROR, hc.getRespon(), null,
AlertType.ERROR);
        display.setCurrent(alert);
    }
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
        progress.setText(null);
    }
}
};

private void initSummary(String desc, String solution) {
    listIns.setTitle("Hasil Identifikasi");
    listIns.deleteAll();

    listIns.append(desc, Consts.TES_IMG);
    listIns.append("Solusi :", null);
    listIns.append(solution, Consts.TES_IMG);

    display.setCurrent(listIns);
}
//identifikasi berikutnya
private Runnable nextIdentification = new Runnable() {

    public void run() {
        try {
            progress.setText(Consts.CONNECT);
            connection hc = new connection();
            hc.setUrl(Consts.IDX_URL);
            idx = (Identification) vector.elementAt(index);
            GlobalUtils.setCodeInstruction(idx.getCode());
            //as1
            Parameter parameter = new Parameter();
            parameter.add(Consts.ID, idx.getCode());

            hc.setParameter(parameter.toString());
            progress.setText(Consts.READ);
            if (hc.doGet()) {
                vector = GlobalUtils.parseToIdx(hc.getRespon());
                // tempVec=new Vector();
                // deepcopy(tempVec, vector);

                index = 0;
                idx = (Identification) vector.elementAt(index);
                if (!GlobalUtils.getSummary()) {

```



```

        text.setText(" " + idx.getDesc() + " ");
        String lvl = getLevel(idx.getCode()) + "." + (index + 1);
        level.setText(lvl);
    } else {
        String desc = idx.getDesc();
        initSummary(desc, idx.getSolusi());
    }
} else {
    System.out.println(Consts.SERVER_ERROR);
}
} catch (Exception ex) {
    ex.printStackTrace();
} finally {
    progress.setText(null);
}
}
};
//ganti halaman arahan
private Runnable nextIns = new Runnable() {

    public void run() {
        try {
            progress.setText(Consts.CONNECT);
            connection hc = new connection();
            hc.setUrl(Consts.IDX_URL);
            //as3
            Parameter parameter = new Parameter();
            parameter.add(Consts.INX, GlobalUtils.getCodeInstruction());
            parameter.add(Consts.NO_PAGE, new Integer(no_page));

            hc.setParameter(parameter.toString());
            progress.setText(Consts.READ);
            // System.out.println(hc.getUrl()+"?" +hc.getParameter());
            if (hc.doGet()) {
                vector = GlobalUtils.parseToIdx(hc.getResponse());
                listIns.deleteAll();

                for (int i = 0; i < vector.size(); i++) {
                    idx = (Identification) vector.elementAt(i);
                    // String solusi = idx.getSolusi();

                    listIns.append(idx.getDesc(), Consts.TES_IMG);
                    // listIns.append(idx.getSolusi(),Consts.TES_IMG);

                }

                StringBuffer sb = new StringBuffer();

```

```

        sb.append("==== Thank u ====\\n");
        sb.append("Page : ").append(no_page).append(" / ").append(page);
        listIns.append(sb.toString(), Consts.FOLDER_IMG);
        display.setCurrent(listIns);
    } else {
        System.out.println(Consts.SERVER_ERROR);
    }
} catch (Exception ex) {
    ex.printStackTrace();
} finally {
    progress.setText(null);
}
}
};
// awal arahan
private Runnable startIns = new Runnable() {

    public void run() {
        try {
            progress.setText(Consts.CONNECT);
            connection hc = new connection();
            hc.setUrl(Consts.IDX_URL);
            //as2
            Parameter parameter = new Parameter();
            parameter.add(Consts.INX, GlobalUtils.getCodeInstruction());
            parameter.add(Consts.INIT_PAGE, new String());

            hc.setParameter(parameter.toString());
            System.out.println(hc.getUrl() + " " + hc.getParameter());
            progress.setText(Consts.READ);
            if (hc.doGet()) {
                page = Integer.parseInt(hc.getRespon());
                if (page == 0) {
                    listIns.append("Maaf Anda tidak dapat melanjutkan proses Diagnosa  
Penyakit Kulit Wajah . Terima Kasih", Consts.TES_IMG);
                    display.setCurrent(listIns);
                } else {
                    Thread thread = new Thread(nextIns);
                    thread.start();
                }
            } else {
                System.out.println(Consts.SERVER_ERROR);
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        } finally {
            progress.setText(null);
        }
    }
}
}

```

```

};

private void toIns() {
    Thread page = new Thread(startIns);
    page.start();
}
//tombol berkiutnya

private void next() {
    int selected = choiceGroup.getSelectedIndex();
    if (Consts.NO == selected) {

        //Image im = Image.createImage("");
        //    altest = new Alert("New alert", "Nooooo!", null, AlertType.INFO);
        //    altest.setTimeout(Alert.FOREVER);
        //    display.setCurrent(altest);
        index++;
        if (vector.size() > index) {
            idx = (Identification) vector.elementAt(index);
            text.setText("" + idx.getDesc() + "");
            String lvl = getLevel(idx.getCode()) + "." + (index + 1);
            level.setText(lvl);
        } else {
            toIns();
        }
    } else {

        if (GlobalUtils.getSummary()) {
        } else {

            tempVec = new Vector();
            deepcopy(tempVec, vector);

            memoryAddress++;
            System.out.println(memoryAddress);
            memory.insertElementAt(tempVec, memoryAddress);
            lastIndex.insertElementAt(new Integer(index), memoryAddress);

            //
            //    for (int i = 0; i < memory.size(); i++) {
            //        Object object = memory.elementAt(i);
            //        System.out.println(i+" : "+((Vector)object).elementAt(0));
            //    }

            Thread thread = new Thread(nextIdentification);
            thread.start();

```

```

//      altest = new Alert("New alert", "Yess!", null, AlertType.INFO);
//      altest.setTimeout(Alert.FOREVER);
//      display.setCurrent(altest);
    }
}

private void prev() {
    System.out.println(memoryAddress);

    if (memoryAddress > -1 && index == 0) {

        vector = new Vector();

        deepcopy(vector, (Vector) memory.elementAt(memoryAddress));

        index = (Integer) lastIndex.elementAt(memoryAddress).intValue();
        idx = (Identification) vector.elementAt(index);
        System.out.println(idx);

        text.setText("" + idx.getDesc() + "");
        String lvl = getLevel(idx.getCode()) + "." + (index + 1);
        level.setText(lvl);
        memoryAddress--;
        return;
    }

    if (index > 0) {
        index--;
        idx = (Identification) vector.elementAt(index);
        System.out.println(idx);

        text.setText(" " + idx.getDesc() + " ");
        String lvl = getLevel(idx.getCode()) + "." + (index + 1);
        level.setText(lvl);
    }
}

else
    mainMenu = MenuUtama.getInstance();
    display.setCurrent(mainMenu);
    return;
}

public void commandAction(Command c, Displayable d) {
    if (c == cBack) {
        mainMenu = MenuUtama.getInstance();
        display.setCurrent(mainMenu);
        return;
    }
}

```

```

    }
    if (c == cNext) {
        next();
        return;
    }
    if (c == cPrev) {
        prev();
        System.out.println("Prev");
        return;
    }
}
}
}

```

3. Menu Login

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package com.client;

import com.admin.AdminMenu;
import com.http.connection;
import com.model.Parameter;
import com.utils.Consts;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.StringItem;
import javax.microedition.lcdui.TextField;
import javax.microedition.midlet.MIDlet;

/**
 *
 *
 */
public class MenuLogin extends Form implements CommandListener, Runnable {

    private MIDlet midlet;
    private Display display;
    private Command cBack = new Command("Kembali", Command.BACK, 0);
    private Command cLogin = new Command("Login", Command.SCREEN, 0);
    private TextField tfUser = new TextField("Username : ", null, 10, TextField.ANY);

```

```

private TextField tfPass = new TextField("Password : ", null, 10, TextField.PASSWORD);
private StringItem siMsg = new StringItem(null, null);

public MenuLogin(MIDlet midlet, Display display) {
    super("Silakan Log In, Admin !");
    this.midlet = midlet;
    this.display = display;
    append(tfUser);
    append(tfPass);
    append(siMsg);

    addCommand(cLogin);
    addCommand(cBack);
    setCommandListener(this);
}

public void commandAction(Command c, Displayable d) {

    if (c == cBack) {
//        MenuUtama mainMenu = new MenuUtama(midlet, display);
        MenuUtama mainMenu = MenuUtama.getInstance();
        display.setCurrent(mainMenu);
    } else if (c == cLogin) {
        if (isValid()) {
            Thread t = new Thread(this);
            t.start();
        } else {
            siMsg.setText("Pengisian Form Belum Lengkap !!!");
        }
    }
}

private boolean isValid() {
    if ("".equals(tfPass.getString()) || "".equals(tfUser.getString())) {
        return false;
    }
    return true;
}

public void run() {
    try {
        siMsg.setText(Constants.LOAD);
        connection hc = new connection();
        hc.setUrl(Constants.LOGIN_URL);

        Parameter parameter = new Parameter();
        parameter.add("username", tfUser.getString());
        parameter.add("password", tfPass.getString());
        hc.setParameter(parameter.toString());
    }
}

```

```

        if (hc.doPost()) {
            if (hc.getResponse().startsWith(Consts.TRUE)) {
//                siMsg.setText("Login Success !!!");
                AdminMenu adminMenu = AdminMenu.getInstance(midlet, display);
                display.setCurrent(adminMenu);
            } else {
                siMsg.setText("Login Gagal !!!");
            }
        } else {
            Alert alert = new Alert(Consts.SERVER_ERROR, hc.getResponse(), null,
AlertType.ERROR);
            display.setCurrent(alert);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
//        siMsg.setText(null);
    }
}
}

```

4. Kamus Bahasa

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package com.client;

import com.http.connection;
import com.model.Identification;
import com.model.Parameter;
import com.utils.Consts;
import com.utils.GlobalUtils;
import java.util.Vector;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Choice;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.List;
import javax.microedition.lcdui.TextField;
import javax.microedition.midlet.MIDlet;

```

```

/**
 *
 */
public class kamusbahasa extends List implements CommandListener {

    private MIDlet midlet;
    private Display display;
    private Command cBack = new Command("Kembali", Command.BACK, 0);
    private Command cRefresh = new Command("Perbaharui", Command.SCREEN, 1);
    private Command cSearch = new Command("Cari", Command.SCREEN, 2);
    private Command cNextPage = new Command("Berikutnya", Command.SCREEN, 3);
    private Command cPrevPage = new Command("Sebelumnya", Command.SCREEN, 4);
    private Form form = new Form("Cari Istilah");
    private TextField tfWord = new TextField(null, null, 50, TextField.ANY);
    private Command cBack2List = new Command("Kembali", Command.BACK, 0);
    private Command cSubmit = new Command("Tampilkan", Command.SCREEN, 0);
    private int page;
    private int no_page;
    private Vector vector;
    private Identification idx;
    private String word = "";

    public kamusbahasa(MIDlet midlet, Display display) {
        super("Daftar Bahasa", Choice.IMPLICIT);
        this.midlet = midlet;
        this.display = display;
        initComponents();
        Thread t = new Thread(initList);
        t.start();
    }

    public void commandAction(Command c, Displayable d) {

        if (c == cRefresh) {
            Thread t = new Thread(initList);
            t.start();
            return;
        }
        if (c == cBack) {
            MenuUtama mainMenu = MenuUtama.getInstance();
            display.setCurrent(mainMenu);
            return;
        }
        if (c == cSearch) {
            display.setCurrent(form);
            return;
        }
    }

```



```

        if (c == cBack2List) {
            display.setCurrent(this);
            return;
        }
        if (c == cSubmit) {
            word = tfWord.getString();
            Thread t = new Thread(initList);
            t.start();
            display.setCurrent(this);
            return;
        }
        if (c == cNextPage) {
            if (no_page < page) {
                no_page++;
                Thread thread = new Thread(viewList);
                thread.start();
            }
            return;
        }
        if (c == cPrevPage) {
            if (no_page > 1) {
                no_page--;
                Thread thread = new Thread(viewList);
                thread.start();
            }
        }
        if (c == List.SELECT_COMMAND) {
            int i = this.getSelectedIndex();
            if (i < vector.size()) {
                idx = (Identification) vector.elementAt(this.getSelectedIndex());
                if (this.size() > vector.size() + 1) {
                    this.delete(this.size() - 1);
                }
                StringBuffer sb = new StringBuffer();
                sb.append("Keterangan : ").append(idx.getDesc());
                // this.append(sb.toString(), null);

                Alert alert = new Alert("Penjelasan", sb.toString(), null, AlertType.INFO);
                alert.setTimeout(Alert.FOREVER);
                display.setCurrent(alert);
            }
            return;
        }
    }

    private void initComponents() {
        page = 0;
        no_page = 1;
    }
}

```

```

        this.addCommand(cBack);

        this.addCommand(cRefresh);

        this.addCommand(cSearch);
        this.addCommand(cNextPage);
        this.addCommand(cPrevPage);
        this.setCommandListener(this);

        form.append(tfWord);
        form.addCommand(cBack2List);
        form.addCommand(cSubmit);
        form.setCommandListener(this);
    }
    private Runnable initList = new Runnable() {

        public void run() {
            try {

                connection hc = new connection();
                hc.setUrl(Consts.DICTIONARY_URL);
                //ds
                Parameter parameter = new Parameter();
                parameter.add(Consts.ACT, Consts.VIEW);
                parameter.add(Consts.WORD, word);
                parameter.add(Consts.INIT_PAGE, null);

                hc.setParameter(parameter.toString());
                if (hc.doGet()) {
                    page = Integer.parseInt(hc.getRespon());
                    Thread thread = new Thread(viewList);
                    thread.start();
                } else {
                    Alert alert = new Alert(Consts.SERVER_ERROR, hc.getRespon(), null,
AlertType.ERROR);
                    display.setCurrent(alert);
                }
            } catch (Exception ex) {
                Alert alert = new Alert(Consts.SERVER_ERROR, ex.getMessage(), null,
AlertType.ERROR);
                display.setCurrent(alert);
                ex.printStackTrace();
            } finally {
            }
        }
    };
    private Runnable viewList = new Runnable() {

```

```

public void run() {
    try {

        connection hc = new connection();
        hc.setUrl(Consts.DICTIONARY_URL);
        //ds1
        Parameter parameter = new Parameter();
        parameter.add(Consts.ACT, Consts.VIEW);
        parameter.add(Consts.WORD, word);
        parameter.add(Consts.NO_PAGE, new Integer(no_page));

        hc.setParameter(parameter.toString());
        kamusbahasa.this.deleteAll();

        System.out.println(hc.getUrl() + hc.getParameter().toString());

        if (hc.doGet()) {
            vector = GlobalUtils.parseToIdx(hc.getResponse());
            for (int i = 0; i < vector.size(); i++) {
                idx = (Identification) vector.elementAt(i);
                kamusbahasa.this.append(idx.getCode(), i % 2 == 0 ?
Consts.PANAH_BIRU : Consts.PANAH_KUNING);

            }
            StringBuffer sb = new StringBuffer();
            sb.append("Halaman : ").append(no_page).append(" / ").append(page);
            kamusbahasa.this.append(sb.toString(), Consts.FOLDER_IMG);

        } else {
            Alert alert = new Alert(Consts.SERVER_ERROR, hc.getResponse(), null,
AlertType.ERROR);
            display.setCurrent(alert);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    } finally {
    }
}
};
}

```

5. About Me

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

```

```

package com.client;

```

```

import com.utils.Consts;
import java.util.TimerTask;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.Image;
import javax.microedition.lcdui.StringItem;
import javax.microedition.lcdui.Ticker;
import javax.microedition.midlet.MIDlet;

/**
 *
 */
public class AboutMe extends Form implements CommandListener {

    private MIDlet midlet;
    private Display display;
    private Command cmdback;
    private StringItem txt, txt2, txt3, txt4, txt5, txt6;
    private Command obj;
    // private Ticker ticker;
    int h, hImg;
    private Image img;
    TimerTask tmTask;

    public AboutMe(MIDlet midlet, Display display) {
        super("About Me");

        h = getHeight();
        try {
            img = Consts.USER_IMG;
        } catch (Exception e) {
            e.printStackTrace();
        }
        // hImg = img.getHeight();

        this.midlet = midlet;
        this.display = display;

        cmdback = new Command("Kembali", Command.BACK, 0);

        txt = new StringItem("Nama : Yusuf","\n");
        txt2 = new StringItem("Nim : 0911510923","\n");
        txt3 = new StringItem("Email: yusuf.sofyan9@gmail.com","\n");
        txt4 = new StringItem("Twitter : @I","\n");

```

```
txt5 = new StringItem("Fakultas Teknologi Informasi", "\n");
txt6 = new StringItem("Universitas Budi Luhur", "\n");

append(img);
append(txt);
append(txt2);
append(txt3);
append(txt4);
append(txt5);
append(txt6);
// setTicker(ticker);
addCommand(cmdback);
setCommandListener(this);
}

public void commandAction(Command c, Displayable d) {
    obj = c;
    if (c == cmdback) {
        MenuUtama mainMenu = MenuUtama.getInstance();
        display.setCurrent(mainMenu);
    }
}
}
```