

**SISTEM PAKAR DIAGNOSA PENYAKIT KULIT
PADA MANUSIA MENGGUNAKAN METODE
FORWARD CHAINING BERBASIS *ANDROID***

SKRIPSI



**Oleh:
Lambas Romauli Manalu
130210101**

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2017**

**SISTEM PAKAR DIAGNOSA PENYAKIT KULIT
PADA MANUSIA MENGGUNAKAN METODE
FORWARD CHAINING BERBASIS *ANDROID***

SKRIPSI

**Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana**



**Oleh:
Lambas Romauli Manalu
130210101**

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2017**

PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, dan/atau magister), baik di Universitas Putera Batam maupun di perguruan tinggi lain.
2. Skripsi ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam, Maret 2017

Yang membuat pernyataan,

Lambas Romauli Manalu
130210101

**SISTEM PAKAR DIAGNOSA PENYAKIT KULIT
PADA MANUSIA MENGGUNAKAN METODE
FORWARD CHAINING BERBASIS *ANDROID***

Oleh
Lambas Romauli Manalu
130210101

SKRIPSI
Untuk memenuhi salah satu syarat guna
memperoleh gelar Sarjana

Telah disetujui oleh Pembimbing pada tanggal
seperti tertera dibawah ini

Batam, 17 Februari 2017

Rahadian Aulia Firda, S.Kom., M.Kom.
Pembimbing

ABSTRAK

Kesehatan kulit merupakan salah satu hal yang harus diperhatikan. Karena kulit merupakan hal yang sensitif. Penelitian ini dilakukan berdasarkan kebutuhan akan adanya alat bantu bagi penderita penyakit kulit dalam mendiagnosis penyakit kulit yang diderita. Sistem pakar ini dalam konsultasinya dapat dijalankan dengan menjawab pertanyaan yang diajukan sistem dengan menjawab YA atau TIDAK, semua jawaban disesuaikan dengan keluhan yang dirasakan oleh penderita. Metode inferensi yang digunakan adalah *Forward Chaining*, yaitu penelusuran yang dilakukan berdasarkan fakta-fakta yang ada menuju konklusi. Metodologi pengembangan menggunakan java ADT (*Android Developer Tool*) dan Android SDK. Keluaran dari sistem ini berupa nama dan gambar penyakit kulit serta saran sesuai dengan jenis penyakit kulit yang diderita.

Kata kunci: sistem pakar, diagnosa penyakit kulit, *forward chaining*, *android*.

ABSTRACT

Healthy skin is one thing that must be considered. Because the skin is sensitive. This study was conducted based on the need for tools for people with skin diseases in diagnosing skin diseases suffered. This expert system can be run in consultation with the system to answer questions with answer YES or NO, all the answers tailored to the grievances felt by the patient. Inference method used is the Forward Chaining, ie searches performed based on the facts that exist to conclusion. Methodology development using Java ADT (Android Developer Tool) and the Android SDK. The output of this system in the form of names and pictures of skin diseases as well as advice in accordance with the type of skin disease suffered.

Keywords: expert system, diagnosis of skin diseases, forward chaining, android.

KATA PENGANTAR

Segala puji dan syukur bagi Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan karunia-NYA, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam.
2. Ketua Program Studi Teknik Informatika Universitas Putera Batam.
3. Bapak Rahadian Aulia Firda, S.Kom., M.Kom., selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam.
5. Terimakasih untuk kedua orang tua saya yang selalu mendukung dan memberikan semangat khususnya untuk Ibunda tercinta Kartina Siburian.
6. Terimakasih untuk Debora Pestaria Dongoran, Ricky Sadewa, Bagus Remulya Fajri, Zefly Haposan Gultom yang telah memberikan hati, waktu dan fikiran untuk membantu dalam penyelesaian Skripsi ini.

7. Terimakasih untuk Ai Nurhandayani, Asri Ayuningtias, Yuli Krisiana Susanti dan teman-teman seperjuangan lainnya.
8. Terimakasih kepada Ibu Sestri Novia Rizki selaku Dosen pembimbing Akademik Saya.
9. Terimakasih untuk Bapak Nanang Hartoyo selaku Manager saya yang selalu memberikan dukungan dalam penyelesaian skripsi ini.
10. Terimakasih untuk keluarga besar yang tidak bosan-bosannya untuk memberikan semangat, terkhusus adik saya Roma Mastaida Manalu.
11. Terimakasih kepada keluarga besar Nesha yang sangat membantu selama proses penyelesaian penelitian ini.
12. Rekan-rekan mahasiswa/i Universitas Putera Batam yang turut memberikan doa dan dukungannya
13. Terimakasih kepada Dr. M. Indrawan Fauzy selaku Pakar dalam penelitian ini.
14. Mitra kerja yang selalu memberikan masukan yang berguna untuk penelitian ini terkhusus kepada Devi Aknes Widyawati.
15. Serta pihak-pihak lain yang tidak dapat disebutkan satu per satu.

Semoga Tuhan membalas kebaikan dan selalu mencurahkan Kasih dan rahmat-Nya, Amin.

Batam, Februari 2017

Penulis

DAFTAR ISI

HALAMAN PERNYATAAN	i
HALAMAN PENGESAHAN	ii
ABSTRAK.....	iii
<i>ABSTRACT</i>	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Penelitian	2
1.2 Identifikasi Masalah.....	3
1.3 Pembatasan Masalah.....	4
1.4 Perumusan Masalah	4
1.5 Tujuan Penelitian	5
1.6 Manfaat Penelitian	5
BAB II TINJAUAN PUSTAKA	6
2.1 Teori Dasar.....	6
2.1.1 Kecerdasan buatan atau <i>Artificial Intelligence (AI)</i>	6
2.1.1.1 Logika <i>fuzzy (fuzzy logic)</i>	8
2.1.1.2 Jaringan saraf tiruan (<i>artificial neural network</i>).....	11
2.1.1.3 Sistem pakar (<i>expert system</i>).....	16
2.1.2 <i>Android</i>	29
2.1.3 <i>Database</i> (basis data).....	30
2.1.4 Validasi Sistem	31
2.2 Variabel Penelitian.....	32
2.2.1 Penyakit Kulit Eksim	32
2.2.2 Penyakit Kulit Kurap	33
2.2.2 Penyakit Kulit Panu	34
2.2.2 Penyakit Kulit Bisul	34
2.2.2 Penyakit Kulit Herpes	35
2.3 <i>Software</i> Pendukung	35
2.3.1 <i>Star UML</i>	35
2.3.2 <i>Android</i>	43
2.3.3 Bahasa Pemrograman <i>Java</i>	46
2.3.2 <i>Eclipse</i>	50
2.4 Penelitian Terdahulu	50
2.5 Kerangka Pemikiran.....	54
BAB III METODE PENELITIAN	55

3.1	Desain Penelitian	55
3.1.1	Teknik pengumpulan data.....	59
3.2	Operasional Variabel.....	60
3.3	Perancangan Sistem	61
3.3.1	Desain basis pengetahuan	61
3.3.2	Struktur kontrol mesin inferensi)	70
3.3.3	Desain <i>UML (Unified Modeling Language)</i>	70
3.3.5	Desain antarmuka	83
3.4	Lokasi dan Jadwal Penelitian.....	86
3.4.1	Lokasi.....	86
3.4.2	Jadwal Penelitian	86
BAB IV HASIL DAN PEMBAHASAN		88
4.1	Hasil Penelitian	88
4.2	Pembahasan.....	95
4.2.1	Pengujian validasi sistem	95
BAB V SIMPULAN DAN SARAN.....		97
5.1	Simpulan	97
5.2	Saran	98

DAFTAR PUSTAKA
 DAFTAR RIWAYAT HIDUP
 SURAT KETERANGAN PENELITIAN
 LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Forward Chaining	21
Gambar 2.2 Backward Chaining	21
Gambar 2.3 Struktur Sistem Pakar Kaidah Produksi	22
Gambar 2.4 Pohon Keputusan	25
Gambar 2.5 Alternatif Pohon Keputusan	26
Gambar 2.6 Penyakit Eksim	33
Gambar 2.7 Penyakit Kurap	33
Gambar 2.8 Penyakit Panu	34
Gambar 2.9 Penyakit Bisul	35
Gambar 2.10 Penyakit Herpes	36
Gambar 2.11 Logo <i>StarUML</i>	36
Gambar 2.12 Logo Android	43
Gambar 2.13 Tampilan <i>Android Development Tools</i>	45
Gambar 2.14 Logo Bahasa Pemrograman Java	46
Gambar 2.15 Logo Eclipse	50
Gambar 2.16 Kerangka Pemikiran	54
Gambar 3.1 Desain Penelitian	56
Gambar 3.2 Pohon Keputusan	67
Gambar 3.3 Desain <i>Data Base</i>	69
Gambar 3.4 <i>Use Case Diagram</i>	71
Gambar 3.5 <i>Activity diagram</i> mengelola <i>knowlwdge base</i>	72
Gambar 3.6 <i>Activity diagram</i> execute file <i>apk</i>	73
Gambar 3.7 <i>Activity diagram</i> update versi	74
Gambar 3.8 <i>Activity diagram</i> <i>sharing file Apk</i>	75
Gambar 3.9 <i>Activity diagram</i> mengunduh file <i>Apk</i>	77
Gambar 3.10 <i>Activity diagram</i> melakukan instalasi	78
Gambar 3.11 <i>Activity diagram</i> melakukan diagnosa	79
Gambar 3.12 <i>Sequence diagram</i> mengelola <i>knowledge base</i>	79
Gambar 3.13 <i>Sequence diagram</i> execute file <i>Apk</i>	80
Gambar 3.14 <i>Sequence diagram</i> memperbaharui versi	80
Gambar 3.15 <i>Sequence diagram</i> <i>Sharing file Apk</i>	81
Gambar 3.16 <i>Sequence diagram</i> mengunduh file <i>Apk</i>	81
Gambar 3.17 <i>Sequence diagram</i> melakukan instalasi	82
Gambar 3.18 <i>Sequence diagram</i> melakukan diagnosa	82
Gambar 3.19 Rancangan <i>Form Beranda</i>	83
Gambar 3.20 Rancangan <i>Form Diagnosa</i>	84
Gambar 3.21 Rancangan <i>Form Hasil Diagnosa</i>	84
Gambar 3.22 Rancangan <i>Form About</i>	85
Gambar 3.23 Rancangan <i>Form Artikel</i>	85

Gambar 3.24 Rancangan <i>Form</i> Profil	86
Gambar 4.1 Menu Utama	88
Gambar 4.2 Tampilan form diagnosa	89
Gambar 4.3 Tampilan hasil diagnosa	90
Gambar 4.4 Tampilan menu article	91
Gambar 4.5 Tampilan jenis penyakit	92
Gambar 4.6 Tampilan form about	93
Gambar 4.7 Tampilan profile	94

DAFTAR TABEL

Tabel 2.1 Tabel Keputusan.....	24
Tabel 2.2 Alternatif Tabel Keputusan	26
Tabel 2.3 Simbol <i>Use Case Diagram</i>	38
Tabel 2.4 Simbol <i>Activity Diagram</i>	40
Tabel 2.5 Simbol <i>Sequence Diagram</i>	41
Tabel 3.1 Indikator	60
Tabel 3.2 Tabel Variabel	61
Tabel 3.3 Tabel Gejala	61
Tabel 3.4 Tabel Aturan.....	63
Tabel 3.5 Tabel Keputusan.....	66
Tabel 3.6 Jadwal Penelitian.....	87
Tabel 4.1 Tabel Pengujian Validasi Sistem	95

DAFTAR LAMPIRAN

LAMPIRAN I SURAT PENELITIAN
LAMPIRAN II FOTO WAWANCARA
LAMPIRAN III FORM WAWANCARA
LAMPIRAN IV KODING PROGRAM

BAB I

PENDAHULUAN

1.1 Latar Belakang Penelitian

Kulit merupakan organ terbesar pada tubuh manusia, luasnya sekitar 2 m² Maharani (2015: 1). Kulit sebagai organ tubuh terluar yang melapisi seluruh organ tubuh manusia memiliki karakteristik yang sensitif terhadap lingkungan luar. Kulit sangat berperan penting untuk melindungi jaringan sel yang terdapat pada organ tubuh lainnya. Dalam penelitian Fitri Nuraeni et al. (2016), berbagai penyakit dan gangguan pada kulit dapat disebabkan oleh beberapa faktor seperti perubahan iklim, lingkungan dan kesehatan diri yang buruk, virus, bakteri, daya tahan tubuh, reaksi alergi dan lain-lain. Penentuan penyakit kulit tidak boleh dilakukan sembarangan, karena penyakit kulit bisa sangat berbahaya bila terjadi kesalahan dalam perawatan dan penanganannya.

Eksim adalah istilah kedokteran untuk kelainan kulit yang mana kulit tampak meradang dan iritasi. Kurap adalah salah satu penyakit kulit menular yang disebabkan oleh fungi. Panu adalah penyakit kulit yang menyerang manusia yang disebabkan oleh jamur. Bisul merupakan sekumpulan nanah yang telah terakumulasi di rongga jaringan setelah terinfeksi sesuatu atau barang asing. Herpes adalah jenis penyakit kulit yang disebabkan oleh virus (Maharani, 2015: 82).

Penyakit kulit Eksim, Kurap, Panu, Bisul dan Herpes diawali dengan gejala gatal-gatal dan peradangan. Beberapa dari jenis penyakit kulit ini memiliki gejala

yang hampir sama dan bahkan sama. Hal ini membuat masyarakat awam sulit untuk mengidentifikasi jenis penyakit kulit yang sedang diderita. Sementara, untuk dapat konsultasi dengan dokter ahli membutuhkan biaya konsultasi yang cukup mahal. Selain itu, kesibukan aktivitas kerja membuat masyarakat awam cenderung mengabaikan jenis penyakit kulit ini.

Perkembangan teknologi di Indonesia sudah sangat pesat sekali. Semua lapisan masyarakat turut merasakan dari kemajuan teknologi sekarang ini. Dengan adanya teknologi yang semakin maju, membuat proses dalam membantu mendiagnosa penyakit kulit eksim, kurap, panu, bisul dan herpes kini dapat dipermudah. Kemampuan teknologi untuk mengolah informasi sudah tidak diragukan lagi, hal ini terlihat dengan munculnya program kecerdasan buatan yang merupakan salah satu bentuk dari perkembangan teknologi.

Salah satu bentuk dari kecerdasan buatan yang banyak digunakan pada saat ini antara lain adalah sistem pakar. Sistem pakar merupakan sistem yang berbasis pengetahuan, mengerjakan tugas yang biasanya dilakukan oleh seorang pakar (Sri Hartati & Sari Iswanti, 2008: 22). *Forward Chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari *rules* IF-THEN (Sutojo, et al., 2011: 171). Dalam hal ini diagnosa penyakit kulit eksim, kurap, panu, bisul dan herpes akan dilakukan berdasarkan gejala-gejala yang dirasakan oleh user atau penderita penyakit kulit tersebut. Metodologi pengembangan menggunakan ADT (*Android Java Tool*) dan *Android SDK* sehingga keluaran dari sistem ini dapat memberikan kemudahan

untuk mendiagnosis penyakit pada kulit, yaitu dengan cara menginstal aplikasi pada *smartphone*.

Aplikasi sistem pakar ini diharapkan mampu untuk memberikan jawaban atau solusi sementara kepada penderita penyakit kulit, sebelum melanjutkan konsultasi kepada dokter ahli. Berdasarkan latar belakang tersebut maka dapat diangkat judul penelitian, yaitu **“Sistem Pakar untuk Mendiagnosis Penyakit Kulit pada Manusia dengan Menggunakan Metode *Forward Chaining* Berbasis *Android*”**.

1.2 Identifikasi Masalah

Berdasarkan latar belakang permasalahan yang ada, maka identifikasi masalah dari permasalahan yang ada adalah sebagai berikut:

1. Minimnya pengetahuan masyarakat awam mengenai gejala dan jenis penyakit kulit.
2. Biaya konsultasi cukup mahal.
3. Kesibukan aktivitas kerja masyarakat awam, sehingga membutuhkan aplikasi yang mudah digunakan.
4. Membutuhkan sebuah aplikasi yang dapat menyerupai kemampuan seorang pakar.

1.3 Pembatasan Masalah

Untuk menghindari pembahasan yang terlalu luas dalam penelitian ini, maka batasan-batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian dilakukan dengan menggunakan Sistem Pakar dengan menggunakan metode *Forward Chaining* berbasis *Android*.
2. Jenis penyakit kulit yang diteliti adalah penyakit kulit yang memiliki gejala gatal-gatal dan peradangan yaitu penyakit kulit eksim, kurap, panu, bisul dan herpes.
3. Diagnosis penyakit kulit ini hanya untuk usia 20 tahun keatas, perempuan dan laki-laki.
4. Pakar atau ahli dalam penelitian ini adalah Dokter M. Indrawan Fauzy seorang Dokter Umum yang bertugas di Rumah Sakit Charis Medica, Batam.
5. Bahasa pemrograman yang digunakan bahasa pemrograman *Java* dengan sistem operasi *Android* dan *Eclipse* sebagai emulator.

1.4 Perumusan Masalah

Berdasarkan pembatasan masalah tersebut, maka rumusan masalah dalam penelitian ini adalah bagaimana sistem pakar untuk mendiagnosa penyakit kulit pada manusia dengan menggunakan metode *forward chaining* berbasis *android* ?

1.5 Tujuan Penelitian

Secara umum, tujuan penelitian ada tiga macam yaitu yang bersifat penemuan, pembuktian, dan pengembangan (Sugiyono, 2014: 3). Adapun tujuan dilakukannya penelitian ini adalah merancang sistem pakar untuk mendiagnosa penyakit kulit pada manusia menggunakan metode *Forward Chaining* berbasis *Android*.

1.6 Manfaat Penelitian

Secara spesifik, penelitian ini diharapkan mampu memberikan manfaat baik dari aspek teoritis (keilmuan) maupun aspek praktis (guna laksana). Manfaat yang akan didapatkan dari penelitian ini antara lain:

(a) Aspek teoritis (keilmuan)

Mengembangkan ilmu pengetahuan tentang konsep sistem pakar agar dapat diterapkan dalam bidang kesehatan sehingga menambah wawasan bagi penderita penyakit kulit.

(b) Aspek praktis (guna laksana)

Secara khusus, penelitian ini menghasilkan sistem pakar yang bermanfaat untuk membantu pengguna untuk mengetahui jenis penyakit kulit yang sedang di derita

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau sering disingkat dengan AI yaitu *Intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud di sini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

Para ahli mendefinisikan AI secara berbeda-beda tergantung pada sudut pandang mereka masing-masing. Ada yang fokus pada logika berpikir manusia saja, tetapi ada juga yang mendefinisikan AI secara lebih luas pada tingkah laku manusia. Berikut beberapa pendapat para ahli mengenai kecerdasan buatan:

2.1.1 Kecerdasan buatan atau *Artificial Intelligence (AI)*

Menurut Hartati dan Iswanti (2008: 1) kecerdasan buatan adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Ilmu komputer tersebut mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia seperti penalaran, pembelajaran, pemecahan masalah, dan sebagainya. Cerdas berarti memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Untuk membuat sebuah mesin

menjadi cerdas (dapat bertindak seperti manusia) maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar. Kecerdasan buatan memungkinkan komputer untuk berpikir atau menalar dan menirukan proses belajar manusia sehingga informasi baru dapat diserap sebagai pengetahuan, pengalaman, dan proses pembelajaran serta dapat digunakan sebagai acuan di masa-masa yang akan datang (Sutojo, Mulyanto, dan Suhartono, 2011: 3).

Menurut Jones (2008: 3) tahun 1950 merupakan saat-saat awal dari *AI* yaitu saat awal sistem komputer dibangun dan ide-ide pembangunan mesin cerdas mulai terbentuk. Pada tahun 1950, Alan Turing menyimpan pertanyaan dalam pikirannya “apakah sebuah mesin mampu untuk berpikir”. Alan Turing melakukan percobaan yang cukup sederhana untuk menentukan apakah suatu mesin bisa dikatakan cerdas. Hasil percobaannya ini disebut dengan *Turing Test*. Dalam *Turing Test*, jika sebuah mesin mampu mengelabui seseorang yang menganggap mesin itu adalah manusia, maka mesin itu dianggap telah lulus dari tes kecerdasan (*intelligence test*).

Seseorang yang menjadi subjek percobaan diminta untuk menentukan terminal mana yang terkoneksi dengan komputer. Subjek boleh mengajukan pertanyaan, membuat pernyataan, menanyakan perasaan dan motivasi selama diperlukan. Jika subjek ternyata gagal menentukan terminal mana yang terkoneksi dengan komputer, maka komputer dinyatakan lulus tes dan dikatakan memiliki *consciousness* (kesadaran)(Al Fatta, 2009: 2).

Menurut Jones (2008: 5) pada tahun 1956, *Darmouth AI Conference* membawa para peneliti yang terlibat dalam penelitian *AI* seperti John McCarthy bersama peneliti-peneliti lainnya untuk sesi diskusi dan penelitian *AI* di *Dartmouth*

College. Sejak saat itu, banyak konferensi *AI* telah diselenggarakan di seluruh dunia, dan berbagai disiplin ilmu belajar di bawah nama *AI*. Pada awal 80-an penelitian tentang *AI* sukses di bidang komersial dari *software* jenis *Expert System*. Pada era 90-an dan awal abad 21, *AI* mencapai kesuksesan terbesarnya setelah diadopsi secara luas oleh industri teknologi, memberikan sumbangan besar pada logistik, data mining, diagnosis medis dan berbagai bidang lainnya (Al Fatta, 2009: 4).

Kombinasi antara *AI* dengan bidang ilmu yang lainnya melahirkan subdisiplin ilmu dalam *AI*. Beberapa diantaranya adalah logika *fuzzy* (*fuzzy logic*), jaringan syaraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*) (Sutojo, dkk., 2011: 12-25).

2.1.1.1 Logika *Fuzzy* (*Fuzzy Logic*)

Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang sesuai untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan komputer, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol. Dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1, artinya suatu keadaan memungkinkan mempunyai dua nilai “Ya” dan “Tidak” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan di berbagai bidang seperti pada sistem diagnosis penyakit (dalam bidang kedokteran); pemodelan sistem pemasaran, sistem operasi (dalam bidang

ekonomi); kendali kualitas air, prediksi adanya gempa bumi, klasifikasi dan pencocokan pola (dalam bidang teknik) (Sutojo, dkk., 2011: 211-212).

Ada beberapa keuntungan yang dapat diambil ketika menggunakan logika *fuzzy* untuk memecahkan suatu masalah, yaitu (Sutojo, dkk., 2011: 212):

1. Perancangannya tidak memerlukan persamaan matematik yang rumit
2. Mudah dimengerti
3. Memiliki toleransi terhadap data-data yang tidak tepat
4. Mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan
6. Dapat bekerja sama dengan teknik-teknik kendali secara konvensional
7. Logika *fuzzy* didasarkan pada bahasa alami

Sistem inferensi *fuzzy* adalah cara memetakan ruang *input* menuju ruang *output* menggunakan logika *fuzzy*. Empat elemen dasar sistem inferensi *fuzzy* antara lain (Sutojo, dkk., 2011: 232):

1. Basis pengetahuan *fuzzy*, yaitu kumpulan aturan (*rule*) *fuzzy* dalam bentuk pernyataan *IF...THEN*.
2. Fuzzifikasi, yaitu proses untuk mengubah *input* sistem yang mempunyai nilai tegas menjadi variabel linguistik menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.
3. Mesin inferensi, yaitu proses untuk mengubah *input fuzzy* menjadi *output fuzzy* dengan cara mengikuti aturan-aturan yang telah ditetapkan pada basis pengetahuan *fuzzy*.

4. Defuzzifikasi, yaitu mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzifikasi.

Beberapa metode yang digunakan dalam sistem inferensi *fuzzy* adalah (Sutojo, dkk., 2011: 233-237):

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (*Minimum*)
- d. Defuzzifikasi menggunakan metode Rata-rata (*Average*)

2. Metode Mamdani

Metode ini sering digunakan karena strukturnya yang sederhana. Pada metode ini, untuk mendapatkan *output* diperlukan 4 tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Aplikasi fungsi implikasi menggunakan fungsi *MIN* (*Minimum*) dan komposisi antar-*rule* menggunakan fungsi *MAX* (*Maximum*) dengan menghasilkan himpunan *fuzzy* baru
- d. Defuzzifikasi menggunakan metode *Centroid* (Titik Tengah)

3. Metode Sugeno

Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Dalam metode ini, *output* sistem berupa konstanta atau persamaan linier. Dalam inferensinya, metode Sugeno menggunakan tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (*Minimum*)
- d. Defuzzifikasi menggunakan metode Rata-rata (*Average*)

2.1.1.2 Jaringan saraf tiruan (*artificial neural network*)

Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologis, seperti proses informasi pada otak manusia. Elemen utamanya adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu. Cara kerja jaringan saraf tiruan sama seperti cara kerja manusia, yaitu belajar melalui contoh. Beberapa contoh aplikasi jaringan saraf tiruan adalah implementasi di bidang kedokteran, yaitu pemodelan dan diagnosis sistem kardiovaskular, hidung elektronik, dan dokter instan; dan implementasi di bidang bisnis, yaitu jaringan saraf tiruan yang diintegrasikan dengan merek dagang *The Airline Marketing Tactician* (*AMT*) menggunakan *back-propagation* untuk membantu kontrol pemasaran dari alokasi kursi penerbangan (Sutojo, dkk., 2011: 283-288).

Beberapa kelebihan yang dimiliki jaringan saraf tiruan antara lain (Sutojo, dkk., 2011: 284):

1. Belajar adaptif, yaitu kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. *Self-Organization*, yaitu kemampuan membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real Time Operation*, yaitu perhitungan jaringan saraf tiruan yang dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Selain mempunyai beberapa kelebihan, jaringan saraf tiruan juga mempunyai kelemahan-kelemahan, yaitu (Sutojo, dkk., 2011: 284-285):

1. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatika, operasi logika, dan simbolis.
3. Membutuhkan pelatihan untuk dapat beroperasi sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.

Salah satu elemen yang menentukan baik tidaknya suatu mode jaringan saraf tiruan adalah hubungan antar-*neuron* atau arsitektur jaringan. *Neuron-neuron* tersebut terkumpul dalam lapisan-lapisan yang disebut *neuron layers*. Terdapat 3 bagian lapisan penyusun jaringan saraf tiruan, yaitu (Sutojo, dkk., 2011: 292):

1. Lapisan *Input (Input Layer)*

Unit-unit dalam lapisan ini disebut unit-unit *input* yang bertugas menerima pola *input*-an dari luar yang menggambarkan suatu permasalahan.

2. Lapisan Tersembunyi (*Hidden Layer*)

Unit-unit dalam lapisan ini disebut unit-unit tersembunyi, yang mana nilai *output*-nya tidak dapat diamati secara langsung.

3. Lapisan *Output (Output Layer)*

Unit-unit dalam lapisan ini disebut unit-unit *output*, yang merupakan solusi jaringan saraf tiruan terhadap suatu permasalahan.

Beberapa arsitektur jaringan yang sering digunakan dalam jaringan saraf tiruan antara lain (Sutojo, dkk., 2011: 292-295):

1. Jaringan Lapisan Tunggal

Jaringan ini terdiri dari 1 lapisan *input* dan 1 lapisan *output*, yang mana setiap unit dalam lapisan *input* selalu terhubung dengan setiap unit yang terdapat pada lapisan *output*. Jaringan ini menerima *input* kemudian mengolahnya menjadi *output* tanpa melewati lapisan tersembunyi. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *ADALINE*, *Hopfield*, dan *Perceptron*.

2. Jaringan Lapisan Banyak

Jaringan ini mempunyai 3 jenis lapisan, yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Jaringan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan jaringan lapisan tunggal. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *MADALINE*, *backpropagation*, dan *Neocognitron*.

3. Jaringan dengan Lapisan Kompetitif

Jaringan ini memiliki bobot yang telah ditentukan dan tidak memiliki proses pelatihan. Jaringan ini digunakan untuk mengetahui *neuron* pemenang dari sejumlah *neuron* yang ada sehingga sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Contoh jaringan saraf tiruan yang menggunakan jaringan ini adalah *Learning Vector Quantization (LVQ)*.

Berdasarkan cara memodifikasi bobotnya, pelatihan jaringan saraf tiruan dibagi menjadi dua, yaitu (Sutojo, dkk., 2011: 301- 392):

1. Pelatihan dengan Supervisi (pembimbing)

Dalam pelatihan ini, jaringan dipandu oleh sejumlah pasangan data (masukan dan target) yang berfungsi sebagai pembimbing untuk melatih jaringan hingga diperoleh bobot yang terbaik. Algoritma yang termasuk dalam pelatihan dengan supervisi antara lain:

a. *Hebb-Rule*

Model ini diperkenalkan oleh D.O. Hebb yang menggunakan cara menghitung bobot dan bias secara iteratif dengan memanfaatkan model pembelajaran dengan supervisi sehingga bobot dan bias dapat dihitung secara otomatis tanpa harus melakukan cara coba-coba. Arsitektur jaringan ini terdiri dari beberapa unit *input* dihubungkan langsung dengan sebuah unit *output*, ditambah dengan sebuah bias.

b. *Perceptron*

Model ini ditemukan oleh Rosenblatt (1962) dan Minsky – Papert (1969). Model jaringan ini merupakan model yang terbaik pada saat itu. Algoritma

pelatihan *perceptron* digunakan baik untuk *input* biner maupun bipolar, dengan θ tertentu.

c. *Delta-Rule*

Selama pelatihan pola, *Delta-Rule* akan mengubah bobot dengan cara meminimalkan *error* antara *output* jaringan dengan target.

d. *Backpropagation*

Backpropagation adalah metode penurunan gradien untuk meminimalkan kuadrat *error* keluaran. Pelatihan jaringan ini terdiri dari 3 tahap, yaitu tahap perambatan maju (*forward propagation*), tahap perambatan balik, dan tahap perubahan bobot dan bias. Arsitektur jaringan ini terdiri dari *input layer*, *hidden layer*, dan *output layer*.

e. *Heteroassociative Memory*

Jaringan saraf *heteroassociative memory* adalah jaringan yang dapat menyimpan kumpulan pengelompokan pola dengan cara menentukan bobot-bobotnya sedemikian rupa. Algoritma pelatihan yang biasa digunakan adalah *Hebb-Rule*.

f. *Bidirectional Associative Memory (BAM)*

Bidirectional Associative Memory (BAM) adalah model jaringan saraf yang memiliki 2 lapisan, yaitu lapisan *input* dan lapisan *output* yang mempunyai hubungan timbal balik antara keduanya (bersifat *bidirectional*). Arsitektur jaringan ini terdiri dari 3 *neuron* pada lapisan *input* dan 2 *neuron* pada lapisan *output*. Model jaringan ini terbagi menjadi 2 jenis yaitu *BAM* Diskrit dan *BAM* Kontinu.

g. *Learning Vector Quantization (LVQ)*

Learning Vector Quantization (LVQ) adalah suatu model pelatihan pada lapisan kompetitif terawasi yang akan belajar secara otomatis untuk mengklasifikasikan vektor-vektor *input* ke dalam kelas-kelas tertentu.

2. Pelatihan tanpa Supervisi

Dalam pelatihan ini, tidak ada pembimbing yang digunakan untuk memandu proses pelatihan. Jaringan hanya diberi *input* tetapi tidak mendapatkan target yang diinginkan sehingga modifikasi bobot pada jaringan dilakukan menurut parameter tertentu. Model jaringan yang termasuk dalam pelatihan tanpa supervisi adalah jaringan kohonen yang diperkenalkan oleh Prof. Teuvo Kohonen pada tahun 1982.

Pada jaringan kohonen, *neuron-neuron* pada suatu lapisan data akan menyusun dirinya sendiri berdasarkan *input* nilai tertentu dalam suatu *cluster*. *Cluster* yang dipilih sebagai pemenang adalah *cluster* yang mempunyai vektor bobot paling cocok dengan pola *input*, yaitu *cluster* yang memiliki jarak yang paling dekat.

2.1.1.3 Sistem Pakar (*Expert System*)

Sistem pakar mulai dikembangkan pada pertengahan 1960, ditandai dengan lahirnya sistem pakar pertama bernama *General-purpose Problem Solver (GPS)* yang dikembangkan oleh Newel dan Simon. Kemudian bermunculan sistem pakar lain di berbagai bidang seperti MYCIN untuk diagnosis penyakit, DENDRAL untuk mengidentifikasi struktur molekul campuran yang tak dikenal, XCON & XSEL untuk membantu konfigurasi sistem komputer besar, SOPHIE untuk analisis

sirkuit elektronik, Prospector digunakan di bidang geologi untuk membantu mencari dan menemukan deposit, FOLIO digunakan untuk membantu memberikan keputusan bagi seorang manajer dalam masalah stok dan investasi, DELTA dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya (Sutojo, dkk., 2011: 159-160).

Menurut Hartati dan Iswanti (2008: 2) sistem pakar merupakan salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia. Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Dengan bantuan sistem pakar, seseorang yang bukan pakar dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar (Sutojo, dkk., 2011: 13).

Pakar adalah seseorang yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan masalah dalam bidang tertentu. Seorang pakar memiliki kemampuan kepakaran seperti mengenali dan merumuskan suatu masalah, menyelesaikan masalah dengan cepat dan tepat, menjelaskan solusi dari suatu masalah, restrukturisasi pengetahuan, belajar dari pengalaman, memahami batas kemampuan, kemampuan untuk mengaplikasikan pengetahuannya dan memberi saran serta pemecahan masalah pada bidang tertentu (Hartati dan Iswanti, 2008: 11).

Menurut Kusri (2006: 14-15) sistem pakar dapat digunakan oleh orang awam yang bukan pakar untuk meningkatkan kemampuan dalam memecahkan masalah. Sistem pakar juga dapat digunakan oleh pakar sebagai *assistant* yang

berpengetahuan, serta digunakan untuk memperbanyak atau menyebarkan sumber pengetahuan yang semakin langka.

Suatu sistem dikatakan sebagai sistem pakar jika memiliki ciri-ciri sebagai berikut (Sutojo, dkk., 2011: 162):

1. Terbatas pada domain keahlian tertentu
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami
4. Bekerja berdasarkan kaidah tertentu
5. Mudah dimodifikasi
6. Basis pengetahuan dan mekanisme inferensi diletakkan terpisah
7. Keluarannya (*output*) bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara terpisah secara searah, sesuai dengan dialog dengan pengguna

Representasi pengetahuan merupakan metode yang digunakan untuk mengodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Hal ini dimaksudkan untuk menangkap sifat-sifat penting dari suatu masalah sehingga informasi itu dapat diakses oleh prosedur pemecahan masalah. Bahasa representasi harus dirancang agar fakta-fakta dan pengetahuan lain yang terkandung di dalamnya dapat digunakan untuk penalaran (Kusrini, 2006: 24).

Menurut Hartati dan Iswanti (2008: 22) representasi pengetahuan dimaksudkan untuk mengorganisasikan pengetahuan dalam bentuk dan format tertentu agar dapat dimengerti oleh komputer. Pemilihan representasi pengetahuan yang tepat akan

menghasilkan sebuah sistem pakar yang efektif. Salah satu model representasi pengetahuan yang penting yaitu kaidah produksi (*production rule*).

Sistem pakar pada penelitian ini menggunakan model representasi pengetahuan berbasis kaidah produksi. Menurut Firebaugh (1988) dalam Hartati dan Iswanti (2008: 10) struktur sistem pakar yang berbasis kaidah produksi terdiri dari 4 komponen, yaitu:

1. Antarmuka pemakai

Menurut Kusri (2006: 17) antarmuka merupakan penghubung antara pemakai dengan sistem pakar. Komponen ini berfungsi sebagai alat komunikasi antara sistem dan pengguna (*user*) yang penting sekali bagi pengguna. Komponen ini harus didesain sedemikian rupa sehingga efektif dan mudah digunakan terutama bagi pengguna yang tidak ahli dalam bidang yang diterapkan pada sistem pakar (Hartati dan Iswanti, 2008: 4-5).

2. Basis pengetahuan

Basis pengetahuan adalah komponen yang berisi sekumpulan kaidah yang berasal dari pengetahuan dalam domain tertentu dan secara umum disajikan dalam bentuk kaidah produksi (*IF...THEN...*). Pengetahuan pakar yang disajikan dalam format tertentu didapat dari sekumpulan pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti buku-buku, jurnal ilmiah, majalah, maupun dokumentasi tercetak lainnya. Basis pengetahuan diletakkan terpisah dari mesin inferensi agar pengembangan pengetahuan sistem pakar dapat dilakukan secara leluasa tanpa mengganggu mesin inferensi (Hartati dan Iswanti, 2008: 5).

3. Struktur kontrol (Mesin Inferensi)

Struktur kontrol merupakan *interpreter* kaidah atau mesin inferensi yang menggunakan pengetahuan-pengetahuan yang tersimpan dalam basis pengetahuan untuk memecahkan atau menyelesaikan permasalahan yang ada. Menurut Kusri (2006: 35) inferensi merupakan proses untuk menghasilkan informasi berupa konklusi logis berdasarkan informasi yang tersedia atau fakta yang diketahui.

Dalam melakukan proses inferensi, sistem pakar memerlukan adanya proses pengujian kaidah-kaidah dalam urutan tertentu untuk mencari suatu kondisi yang sesuai dengan kondisi awal atau untuk memastikan kondisi yang sedang berjalan sudah dimasukkan ke dalam *database*. Proses pengujian itu disebut dengan peruntukan atau penalaran, yaitu proses pencocokan fakta atau kondisi tertentu yang tersimpan dalam basis pengetahuan maupun pada memori kerja dengan kondisi yang dinyatakan dalam premis atau bagian kondisi pada suatu kaidah atau aturan (Hartati dan Iswanti, 2008: 45).

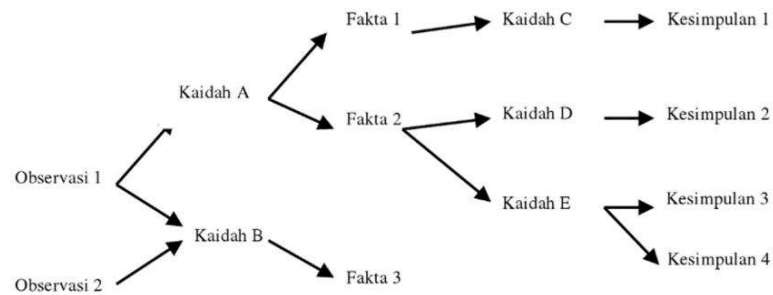
Ada beberapa konsep penalaran yang dapat digunakan oleh mesin inferensi yaitu:

a. Penalaran maju (*forward chaining*)

Konsep ini dapat juga disebut sebagai pencarian yang dimotori data (*data driven search*). Runut maju melakukan proses peruntukan (penalaran) dimulai dari premis-premis atau informasi masukan (*IF*) terlebih dahulu kemudian menuju konklusi atau *derived information* (*THEN*). Konsep ini dapat dimodelkan sebagai berikut:

IF (informasi masukan)

THEN (konklusi)



Gambar 2.1. Metode *Forward Chaining*
(Sumber: Hayadi, 2016: 7)

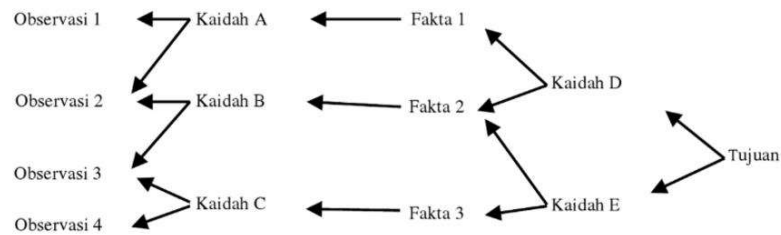
Informasi masukan dapat berupa suatu pengamatan sedangkan konklusi dapat berupa diagnosa sehingga dapat dikatakan jalannya penalaran runut maju dimulai dari pengamatan menuju diagnosa. Pada metode ini, sistem tidak melakukan praduga apapun terhadap konklusi, namun sistem akan menerima semua gejala yang diberikan pengguna lalu sistem akan memeriksa gejala-gejala tersebut dan selanjutnya mencocokkan dengan konklusi yang sesuai (Hartati dan Iswanti, 2008: 45-47).

b. Penalaran mundur (*backward chaining*)

Secara umum, konsep ini diaplikasikan ketika tujuan ditentukan sebagai kondisi atau keadaan awal. Konsep ini disebut juga *goal-driven search*. Arah penalaran atau perunutan dalam konsep ini berlawanan dengan *forward chaining*. Konsep ini dapat dimodelkan sebagai berikut:

Tujuan,

IF (kondisi)

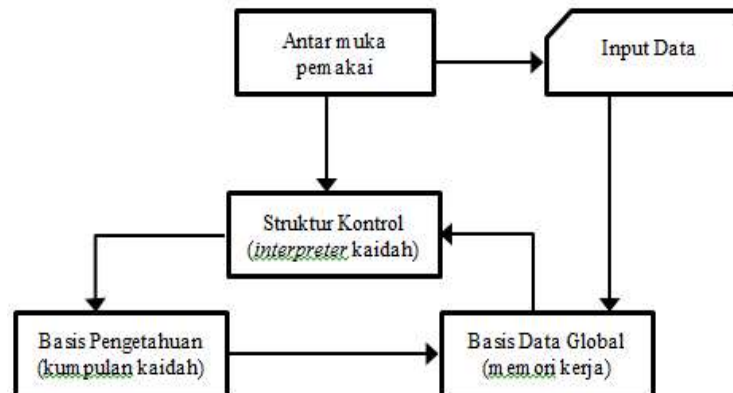


Gambar 2.2 Metode *Backward Chaining*
(Sumber: Hayadi, 2016: 8)

Proses penalaran pada *backward chaining* dimulai dari tujuan kemudian merunut balik ke jalur yang mengarah ke tujuan tersebut, untuk membuktikan bahwa bagian kondisi pada kaidah atau aturan benar-benar terpenuhi. Proses *internal* selalu memeriksa konklusi (tujuan) terlebih dahulu sebagai praduga awal, kemudian memeriksa dan memastikan gejala-gejala (kondisi) telah terpenuhi dan selanjutnya mengeluarkan konklusi sebagai *output*. Jika sistem menemukan ada bagian kondisi yang tidak terpenuhi maka sistem akan memeriksa konklusi (tujuan) pada aturan atau kaidah berikutnya (Hartati dan Iswanti, 46-47).

4. *Working memory* (memori kerja) atau basis data global

Berfungsi untuk mencatat status masalah yang terjadi dan *history* solusi. Memori kerja merupakan bagian yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi saat proses konsultasi terjadi (Kusrini, 2006: 19).



Gambar 2.3 Struktur Sistem Pakar Kaidah Produksi
(Sumber: Firebaugh, 1988 dalam Hartati dan Iswanti, 2008: 10)

Kusrini (2008: 33) menjelaskan bahwa kaidah menyediakan cara formal yang dituliskan dalam bentuk jika-maka (*IF-THEN*) untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah *IF-THEN* menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Berikut ini adalah contoh struktur kaidah *IF-THEN* yang menghubungkan obyek (Adedeji, 1992 dalam Hartati dan Iswanti, 2008: 25):

1. *IF* premis *THEN* konklusi
2. *IF* masukan *THEN* keluaran
3. *IF* kondisi *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan
7. *IF* aksi *THEN* reaksi
8. *IF* gejala *THEN* diagnosa

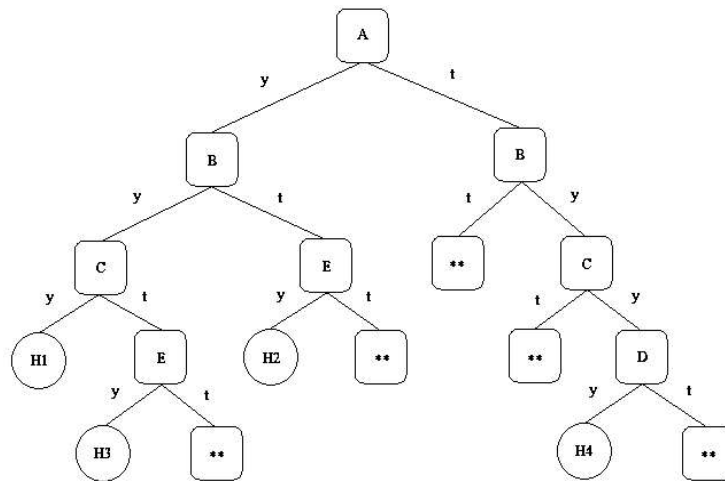
Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan (diagnosa) (Hartati dan Iswanti, 2008: 25-26).

Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk tabel keputusan kemudian dibuat pohon keputusannya. Berikut ini adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan (Hartati dan Iswanti, 2008: 26-39).

Tabel 2.1 Tabel Keputusan

Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>	ya	ya	ya	Tidak
<i>Evidence B</i>	ya	tidak	ya	Ya
<i>Evidence C</i>	ya	tidak	tidak	Ya
<i>Evidence D</i>	tidak	tidak	tidak	Ya
<i>Evidence E</i>	tidak	Ya	ya	Tidak

Sumber: Hartati dan Iswanti (2008: 32)



Keterangan:

A = *evidence* A, H1 = hipotesa 1, y = ya
 B = *evidence* B, H2 = hipotesa 2, t = tidak
 C = *evidence* C, H3 = hipotesa 3, ** = tidak menghasilkan hipotesa tertentu
 D = *evidence* D, H4 = hipotesa 4

Gambar 2.4 Pohon Keputusan
 (Sumber: Hartati dan Iswanti, 2008: 33)

Dari gambar 2.4 dapat diketahui bahwa hipotesa H1 terpenuhi jika memenuhi *evidence* A, B, dan C. Hipotesa H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. Hipotesa H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. Hipotesa H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi *node* (*evidence*) di atasnya, notasi “t” artinya tidak memenuhi.

Dalam sesi konsultasi pada sistem pakar, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.4 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak”

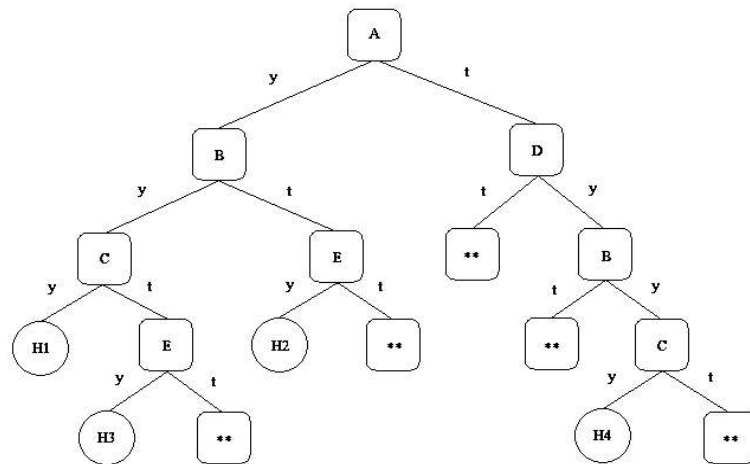
maka sistem akan menanyakan *evidence* B. Ini berarti jawaban pengguna tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.2.

Tabel 2.2 Alternatif Tabel Keputusan

Hipotesa <i>Evidence</i>	Hipotesa 1	Hipotesa 2	Hipotesa 3	Hipotesa 4
<i>Evidence A</i>	Ya	ya	ya	Tidak
<i>Evidence D</i>	Tidak	tidak	tidak	Ya
<i>Evidence B</i>	Ya	tidak	ya	ya
<i>Evidence C</i>	Ya	tidak	tidak	Ya
<i>Evidence E</i>	Tidak	ya	ya	Tidak

Sumber: Hartati dan Iswanti (2008: 34)

Berdasarkan tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



Keterangan:

A = *evidence A*, H1 = hipotesa 1, y = ya
 B = *evidence B*, H2 = hipotesa 2, t = tidak
 C = *evidence C*, H3 = hipotesa 3, ** = tidak menghasilkan hipotesa tertentu
 D = *evidence D*, H4 = hipotesa 4

Gambar 2.5 Alternatif Pohon Keputusan
 (Sumber: Hartati dan Iswanti, 2008: 35)

Dilihat dari gambar 2.5, masing-masing *node* yang mewakili *evidence* tertentu untuk kondisi “y” dan “t” sudah tidak mengarah pada *evidence* yang sama. Hal ini berarti jawaban pengguna yang berbeda akan mengarah pada pertanyaan yang berbeda pula.

Kaidah yang dapat dihasilkan berdasarkan pohon keputusan pada gambar 2.5 adalah sebagai berikut:

1. Kaidah 1: *IF A AND B AND C THEN H1*
2. Kaidah 2: *IF A AND B AND E THEN H3*
3. Kaidah 3: *IF A AND E THEN H2*
4. Kaidah 4: *IF D AND B AND C THEN H4*

Model representasi pengetahuan kaidah produksi banyak digunakan pada aplikasi sistem pakar karena model representasi ini mudah dipahami dan bersifat deklaratif sesuai dengan jalan pikiran manusia dalam menyelesaikan suatu masalah, dan mudah diinterpretasikan.

Menurut Kusri (2006: 14) terdapat beberapa alasan mengapa sistem pakar dikembangkan untuk menggantikan seorang pakar:

1. Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Menghadirkan/menggunakan jasa seorang pakar memerlukan biaya yang mahal.

5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat (*hostile environment*).

Adapun kelebihan yang dimiliki sistem pakar antara lain (Kusrini, 2006: 15):

1. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar.
2. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti.
3. Meningkatkan *output* dan produktifitas, bekerja lebih cepat dari manusia sehingga mengurangi jumlah pekerja yang dibutuhkan dan akan mereduksi biaya.
4. Meningkatkan kualitas.
5. Sistem pakar menyediakan nasihat yang konsisten dan dapat mengurangi tingkat kesalahan.
6. Membuat peralatan yang kompleks lebih mudah dioperasikan karena sistem pakar dapat melatih pekerja yang tidak berpengalaman.
7. Handal (*reability*).
8. Sistem pakar tidak dapat lelah atau bosan serta konsisten dalam memberikan jawaban dan selalu memberikan perhatian penuh.
9. Memiliki kemampuan untuk memecahkan masalah yang kompleks.
10. Memungkinkan pemindahan pengetahuan ke lokasi yang jauh serta memperluas jangkauan seorang pakar, dapat diperoleh dan dipakai dimana saja. Sistem pakar merupakan arsip yang terpercaya dari sebuah keahlian sehingga *user* seolah-olah berkonsultasi langsung dengan sang pakar meskipun sang pakar sudah pensiun.

Selain memiliki beberapa kelebihan yang dapat dimanfaatkan, sistem pakar juga memiliki beberapa kekurangan, yaitu (Sutojo, dkk., 2011: 161):

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar.

2.1.2 *Android*

Menurut *Nasruddin Safaat*, Android merupakan suatu sistem operasi dihandphone yang bersifat terbuka dan berbasis pada sistem operasi Linux. Android bersifat multiplatform yaitu bias digunakan oleh setiap orang yang ingin menggunakannya. Android menyediakan suatu platform terbuka untuk para pengembang dalam menciptakan suatu aplikasi sendiri untuk bermacam peranti bergerak. Android memiliki beberapa kelebihan diantaranya:

1. Multitasking

Yaitu android bisa membuka beberapa aplikasi sekaligus tanpa harus menutup salah satunya.

2. Kemudahan dalam Notifikasi

Adanya pemberitahuan seperti yang ada di SMS, Email, Facebook dan lain-lain, sehingga tidak akan melewatkan pemberitahuan tersebut

3. Mudah terhadap macam-macam Aplikasi Android lewat Google Android App Market

Bisa didownload gratis dengan beraneka aplikasies yang siap di download di ponsel anda.

4. Bisa menginstal ROM yang dimodifikasi
5. Widget

Bisa mengakses berbagai setting atau pengaturan dengan cepat dan mudah.

2.1.3 *Database (basis data)*

Menurut A.S. dan Shalahuddin (2013: 43-44) sistem *database* adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data atau informasi yang sudah diolah dan membuat informasi tersedia saat dibutuhkan. *Database* adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data meliputi memasukkan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan. Salah satu bentuk basis data yang dibutuhkan dalam sebuah sistem yaitu *Database Management System (DBMS)*. *DBMS* adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Syarat minimal dari *DBMS* antara lain (A.S. dan Shalahuddin, 2013: 44-45):

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara bersamaan
4. Mampu menangani *backup* data

Ada beberapa *DBMS* yang paling banyak digunakan saat ini antara lain:

1. *DBMS* versi komersial, yaitu *Oracle*, *Microsoft SQL Server*, *IBM DB2*, dan *Microsoft Access*
2. *DBMS* versi *open source*, yaitu *MySQL*, *PostgreSQL*, *Firebird*, dan *SQLite*

Dalam alur hidup basis data (*Database Life Cycle*), terdapat tahapan yang dinamakan *physical database design*. Biasanya pada tahap ini dibuat rancangan fisik *database* yaitu *Physical Data Model (PDM)*. *PDM* adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antar data. Setiap tabel mempunyai sejumlah kolom yang mempunyai nama unik beserta tipe data yang digunakan. *PDM* merupakan konsep yang digunakan untuk menerangkan secara detail bagaimana data disimpan dalam *database*. *PDM* sudah dalam bentuk fisik perancangan *database* yang siap diimplementasikan ke dalam *DBMS* sehingga nama tabel pada *PDM* merupakan nama asli tabel yang akan diimplementasikan ke dalam *DBMS* (A.S. dan Shalahuddin, 2013: 63).

2.1.4 Validasi Sistem

Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa sistem atau perangkat lunak yang dibangun telah sesuai dengan yang diharapkan. Beberapa pendekatan dalam melakukan pengujian untuk validasi sistem antara lain (A.S. dan Shalahuddin, 2013: 275-276):

1. *Black-Box Testing* (pengujian kotak hitam)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Tujuannya untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari sistem atau perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem atau perangkat lunak apakah sesuai dengan

spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan *black-box testing* harus dibuat dengan kasus benar dan kasus sala

2. *White-Box Testing* (pengujian kotak putih)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. *White-box testing* dilakukan dengan memeriksa logika dari kode program. Pembuatan kasus uji dapat mengikuti standar pengujian dari standar pemrograman yang ada.

2.2 Variabel Penelitian

Kulit adalah bagian paling luar dari jaringan tubuh kita. Kulit membungkus otot tubuh kita. Pada saat kulit terkelupas, rasa perih menyengat. Hal itu menunjukkan bahwa betapa kulit sangat memiliki fungsi yang sangat penting. Selain membungkus otot tubuh, juga memberikan perlindungan bagi jaringan-jaringan tubuh lain yang ada dibawahnya. Kulit meliputi seluruh jaringan kulit secara umum, termasuk kulit wajah.

2.2.1 Penyakit Kulit Eksim

Maharani (2015:55) penyakit kulit eksim termasuk jenis penyakit kulit akibat gangguan inflamasi kebanyakan kondisinya jangka panjang, menyebabkan kemerahan, pembengkakan, lesi dan plak pada kulit, kulit meradang, melepuh dan berisi cairan. Biasanya menimbulkan rasa gatal.



Gambar 2.6 Penyakit eksim pada kulit
(Sumber: Maharani, 2015: 55)

Eksim adalah istilah kedokteran untuk kelainan kulit yang mana kulit tampak meradang dan iritasi. Peradangan ini bisa terjadi dimana saja namun yang paling sering terkena adalah tangan dan kaki.

2.2.2 Penyakit Kulit Kurap

Maharani (2015: 104) penyakit kurap adalah salah satu penyakit kulit yang menular yang disebabkan oleh fungi. Masa infeksi kurap hingga terkena penyakit adalah beberapa hari.



Gambar 2.7 Penyakit kurap pada kulit
(Sumber: Maharani, 2015: 104)

2.2.3 Penyakit Kulit Panu

Maharani (2015: 102) panu adalah penyakit kulit yang menyerang manusia yang disebabkan oleh jamur. Penyakit panu ditandai dengan bercak yang terdapat pada kulit disertai rasa gatal pada saat berkeringat. Bercak-bercak ini bisa berwarna putih, coklat atau merah tergantung warna kulit si penderita. Penyakit ini biasanya menyerang pada semua bagian kulit. Umumnya menular, biasanya ditemukan pada kulit kepala, lipatan lengan, leher, wajah dan kaki.



Gambar 2.8 Penyakit panu pada kulit
(Sumber: Maharani, 2015: 102)

2.2.4 Penyakit Kulit Bisul

Maharani (2015: 94) bisul merupakan sekumpulan nanah (neutrofil mati) yang telah terakumulasi di rongga jaringan setelah terinfeksi sesuatu (umumnya karena bakteri dan parasit) atau barang asing (seperti luka tembakan/tikaman).

Bisul merupakan reaksi ketahanan jaringan untuk menghindari menyebarnya barang asing di dalam tubuh. Organisme atau barang asing membunuh sel disekitarnya, mengakibatkan keluarnya toksin. Toksin tersebut

menyebabkan radang, sel darah putih mengalir menuju tempat tersebut dan kemudian meningkatkan aliran darah di tempat tersebut.

Struktur terakhir bisul adalah dinding bisul yang terbentuk oleh sel sehat untuk mencegah barang asing tersebut masuk ke dalam tubuh dan mencegah terkenanya sel lain.



Gambar 2.9 Penyakit bisul pada kulit
(Sumber: Maharani, 2015: 94)

2.2.5 Penyakit Kulit Herpes

Maharani (2015: 83) penyakit herpes adalah jenis penyakit kulit yang disebabkan oleh virus. Bintik-bintik kecil yang tumbuh ini berubah menjadi gelembung-gelembung transparan berisi cairan, persis seperti pada cacar air namun hanya bergerombol di sepanjang kulit yang dilalui oleh saraf yang terkena. Bintik-bintik baru dapat terus bermunculan dan membesar sampai seminggu kemudian.



Gambar 2.10 Penyakit herpes pada kulit
(Sumber: Maharani, 2015: 83)

2.3 Software Pendukung

Software pendukung merupakan beberapa perangkat lunak yang digunakan untuk mendukung pembuatan sistem pakar dalam penelitian ini. Perangkat lunak tersebut antara lain: SDK *Android*, Bahasa Pemrograman *Java*, *Eclipse* dan *StarUML*.

2.3.1 *StarUML*



Gambar 2.11 Logo *StarUML*
(Sumber: A.S. dan Shalahuddin, 2011: 1)

StarUML merupakan salah satu *CASE (Computer-Aided Software Engineering) tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upper CASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak (A.S. dan Shalahuddin, 2013: 122-123).

Terdapat 13 macam diagram dalam *UML 2.3* yang dibagi menjadi 3 kategori yaitu (A.S. dan Shalahuddin, 2013: 140-141):

1. *Structure diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Diagram UML yang termasuk dalam kategori ini antara lain *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.

2. *Behaviour diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Diagram UML yang termasuk dalam kategori ini antara lain *use case diagram*, *activity diagram*, dan *state machine diagram*.

3. *Interaction diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Diagram UML yang termasuk dalam kategori ini

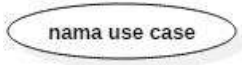
antara lain *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction overview diagram*.

Menurut A.S. dan Shalahudin (2013: 18) *use case* dan *sequence diagram* merupakan bagian dari desain sistem. Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem yaitu:




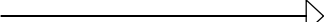
1. *Use case diagram*

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada 2 hal utama yang terdapat pada *use case* yaitu aktor dan *use case*. Berikut ini adalah simbol-simbol yang digunakan dalam *use case diagram* (A.S. dan Shalahuddin, 2013: 155).

Tabel 2.3 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>

Tabel 2.3 Tabel lanjutan

<p>Aktor/<i>actor</i></p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi/<i>extend</i></p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu. Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>






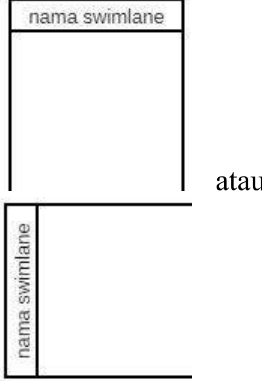
Sumber: A.S. dan Shalahuddin (2013: 162)

2. Activity diagram

Activity diagram merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Jadi dapat dikatakan bahwa *activity diagram* menggambarkan aktifitas sistem, bukan apa yang dilakukan oleh aktor. Simbol-simbol yang

digunakan dalam *activity diagram* ditampilkan dalam tabel berikut (A.S. dan Shalahuddin: 2013: 161-162).

Tabel 2.4 Simbol *Activity Diagram*



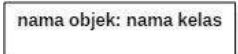

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktifitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi

Sumber: A.S. dan Shalahuddin (2013: 162-163)

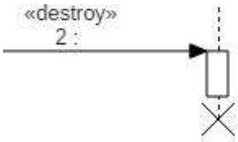
3. *Sequence diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup (*life cycle*) objek dan *message* (pesan) yang dikirimkan dan diterima antar objek. Jumlah *sequence diagram* yang harus digambar minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. Semakin banyak *use case* yang didefinisikan semakin banyak pula *sequence diagram* yang harus dibuat. Simbol-simbol yang digunakan pada *sequence diagram* ditampilkan dalam tabel berikut (A.S. dan Shalahuddin, 2013: 165-166).

Tabel 2.5 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor/ <i>actor</i>  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Garis hidup/ <i>lifeline</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Objek  nama objek: nama kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif

Tabel 2.5 Lanjutan

<p>Pesan tipe <i>create</i> <code><<create>></code> →</p>	<p>Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat</p>
<p>pesan tipe <i>call</i> 1 : <code>nama_metode()</code> →</p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.</p>
<p>Pesan tipe <i>send</i> 1 : <code>masukan</code> →</p>	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju</p>
<p>pesan tipe <i>return</i> 1 : <code>keluaran</code> - - - - -></p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu. Arah panah mengarah pada objek penerima</p>
<p>Pesan tipe <i>destroy</i> <code><<destroy>></code> 2 : → </p>	<p>Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri</p>

Sumber: A.S. dan Shalahuddin (2013: 166- 167)

2.3.2 *Android*



Gambar 2.12 Logo *Android*
(Sumber: Tim EMS, 2014:2)

Android merupakan sebuah sistem operasi untuk perangkat lunak *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas dari distribusinya tanpa dukungan langsung dari Google atau dikenal sebagai *Open Handset Distribution* (OHD). Android dipuji sebagai “*platform mobile* pertama yang Lengkap, Terbuka, dan Bebas”.

1. Lengkap (*Complete Platform*): Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan platform Android. Android merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun software yang memungkinkan untuk peluang pengembangan aplikasi.
2. Terbuka (*Open Source Platform*): Platform Android disediakan melalui lisensi open source. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.

3. *Free (Free Platform)*: Android adalah *Platform/aplikasi* yang bebas untuk *develop*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan yang diperlukan. Tidak ada biaya yang diperlukan untuk pengujian. Tidak ada kontrak yang diperlukan.

Aplikasi Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun. Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, dimana prosesnya dipackage oleh *tools* yang dinamakan “*apt tools*” ke dalam paket Android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang kita sebut aplikasi, dan nantinya dapat diinstall di perangkat mobile. Ada empat jenis komponen pada aplikasi Android yaitu:

1. *Activities*: Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut.
2. *Service*: *Service* tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya.
3. *Broadcast Receiver*: *Broadcast receiver* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh camera, atau perubahan

referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi broadcast misalnya memberikan informasi pada aplikasi lain.

4. *Content Provider*: *Content Provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem sistem seperti database SQLite. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta (Map), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi content navigasi.



Gambar 2.13 Tampilan *Android Developer Tools*
(Sumber Data: Nazruddin, 2015: 6)

2.3.3 Bahasa Pemrograman *Java*



Gambar 2.14 Logo Bahasa Pemrograman Java
(Sumber: Bambang Haryanto, 2014: 3)

Bahasa pemrograman Java memberi harapan menjadi perekat universal yang mengkoneksi pemakai dengan informasi dari *web server*, basis data, penyedia informasi dan sumber-sumber lain. Bahasa Java memiliki fitur keamanan *built-in*. Bahasa Java juga mempermudah pemrograman aplikasi *multithreading* (Bambang Haryanto, 2014:1). Bahasa Java merupakan karya **Sun Microsystems Inc.** Rilis resmi level beta dilakukan pada November 1995. Pada tahun 1996, **Sun** mengeluarkan JSDK (*Java Software Development Kit*). Java telah berjalan pada segala perangkat dari laptop sampai pusat data, konsol game, sampai super komputer ilmiah.

Java telah berkembang dari semula ditujukan untuk pemrograman *applet* di *web browser* menjadi bahasa pemrograman pengembangan aneka ragam aplikasi, mulai dari yang berjalan di *handheld devices* seperti *handphone*, PDA (*Personal Digital Assistant*) sampai aplikasi tersebar skala *enterprise* di beragam komputer *server*. Java merupakan bahasa orientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis *internet*, aplikasi untuk perangkat cerdas yang dapat

berkomunikasi lewat *internet*/jaringan komunikasi. Melalui teknologi Java, dimungkinkan perangkat *audio stereo* di rumah terhubung jaringan komputer. Java tidak lagi hanya bahasa untuk membuat **applet** yang memperindah halaman *web*, tapi Java telah menjadi bahasa untuk pengembangan aplikasi skala *enterprise* berbasis jaringan besar.

Sebagian besar bahasa pemrograman *modern* berdiri di atas pustaka-pustaka kelas yang telah ada untuk mendukung fungsionalitas. Pada bahasa Java, kelompok-kelompok kelas yang berkaitan erat dimasukkan di suatu paket, bervariasi sesuai edisi java. Masing-masing paket untuk maksud tertentu: *applet*, aplikasi standar, skala *enterprise*, dan produk konsumen. Java adalah bahasa yang dapat dijalankan di sembarang *platform*, di beragam lingkungan: *internet*, *consumer electronic products*, dan *computer applications*. *The java 2 Platform* tersedia dalam tiga edisi untuk keperluan berbeda berikut:

1. Java 2 *Standard Edition* (J2SE)
2. Java 2 *Enterprise Edition* (J2EE)
3. Java 2 *Micro Edition* (J2ME)

Pada pengembangan *enterprise applications*, kita menggunakan sejumlah besar paket. Pada *consumer electronic product*, hanya sejumlah kecil bagian bahasa yang digunakan. Masing-masing edisi berisi *Java 2 Software Development Kit* (SDK) untuk mengembangkan aplikasi dan *Java 2 Runtime Environment* (JRE) untuk menjalankan aplikasi. Fitur penting bahasa Java adalah bahasa ditujukan untuk membuat beragam jenis aplikasi secara seragam, yaitu:

1. Program di lingkungan *web browser*

Applet, program ini di eksekusi di *web browser* dari halaman web yang memuat *Java applet*. *Web Browser* kemudian menugaskan *Java Interpreter* (JRE – *Java Runtime Environment*) untuk mengeksekusi *Java Applet* yang diterima. *Java applet* membuat langkah besar, yang memungkinkan *web* menjadi sarana/media interaktif di mana halaman *web* menjadi dapat bereaksi terhadap masukan atau tanggapan pemakai.

2. Program di lingkungan *web server*

Java Server Pages, sebagai *web scripting* serupa dengan ASP, PHP dan sebagainya. Program ditempelkan di halaman html. Html ini tidak langsung dikirim ke *web browser* tapi diolah dulu oleh *web server* ke *web browser*. *Java Servlet*, komponen ini adalah semacam modul di *web server*. JSP akan diterjemahkan menjadi *servlet* agar mempercepat proses eksekusi.

3. Program mandiri (atau biasa disebut *stand-alone application*).

Java merupakan pilihan bagus untuk membuat *applet*. Sekaligus sebagai bahasa bermaksud umum (*general-purpose language*) untuk mengembangkan semua jenis program yang dapat dijalankan di komputer, sistem operasi apa pun asalkan terdapat *Java Interpreter* di *platform* itu. Sebuah program Java didaftarkan dengan *file* teks ekstensi *.java*. Program ini dikompilasi menghasilkan satu *file bytecode* berekstensi **.class** atau lebih. *Bytecode* adalah serangkaian instruksi serupa kode

mesin. Perbedaannya adalah kode mesin harus dijalankan sistem komputer tertentu, sementara *bytecode* berjalan di *Java interpreter* yang tersedia di semua *platform*.

4. Program mandiri sebagai pustaka komponen

Untuk pengembangan aplikasi, java menyediakan Bean untuk mendukung RAD (rapid application development) yang berbasis visual seperti Visual Basic. Bahasa untuk pengembangan aplikasi objek-objek tersebar skala *enterprise*. Terdapat teknologi Java untuk mengembangkan komponen yaitu EJB (**Enterprises JavaBeans**) yang berjalan di *application server*. EJB mendukung *Component-based Software Engineering*. *Application Server* adalah *middleware* yang bertugas menjadi intermediasi beragam *server* di aplikasi tersebar skala *enterprise*. Dukungan Java terhadap sistem tersebar berupa spesifikasi J2EE lengkap meliputi JDBC (*Java Database Connectivity*), RMI (*Remote Method Invocation*), EJB (*Enterprise Java Beans*), JMS (*Java Messaging System*)

5. *JavaScript* bukan merupakan program Java

Namun sintaks dan semantiknya seperti bahasa Java. *JavaScript* tidak termasuk teknologi Java. Java adalah pemrograman orientasi objek yang berukuran kecil, sederhana, aman, diinterpretasi atau dioptimasi secara dinamis, ber-*bytecode*, netral arsitektur, mempunyai *garbage-collector*, *multithreading*, mempunyai mekanisme penanganan pengecualian (*exception handling*), berbasis tipe untuk penulisan program mudah diperluas secara dinamis serta diperuntukkan untuk sistem tersebar (Bill Joy *dalam* Bambang Hariyanto, 2014: 10).

2.3.4 Eclipse



Gambar 2.15 Eclipse
(Sumber: Tim EMS, 2015: 37)

Untuk IDE (*Integrated Development Environment*) yang dipakai dalam pemrograman Android, baik berbasis *Java* atau pemaketan dengan HTML dengan *Cordova* dapat menggunakan *Eclipse*. Sebenarnya tools yang dapat dipakai tidak hanya *Eclipse*, tetapi ada juga tool lainnya seperti *Android Studio* (Tim EMS, 2015: 35)

Pemrograman *Android* secara garis besar terdiri atas 2 metode, di mana keduanya menggunakan *eclipse*. Yang pertama menggunakan *Java* dan yang kedua menggunakan *html* (Tim EMS, 2015: 53).

2.4 Penelitian Terdahulu

Untuk mendukung teori yang berkaitan dengan penelitian, peneliti mencantumkan beberapa penelitian terdahulu di bidang sistem pakar dalam kategori diagnosis.

Paryati (2011), Sistem Pakar Berbasis Web untuk Mendiagnosa Penyakit Kulit. Sistem pakar yang dibangun untuk mendiagnosa penyakit dan cara

penyembuhannya. Sistem ini bertujuan membantu user dapat mengetahui jenis penyakit yang di derita dan penyembuhannya melalui air beherang dan ramuan tradisional atau obat tradisional, serta informasi ramuan obat guna membantu proses penyembuhannya. Data rekomendasi yang dihasilkan dalam sistem ini dilengkapi dengan jenis pennisakit, gejala penyakit dan cara penyembuhannya sehingga user dapat mengetahui jenis penyakit yang diderita dan cara pengobatannya. Sistem ini akan menganalisis jawaban dari setiap pertanyaan yang diberikan agar dapat memperoleh jawaban berdasarkan basis pengetahuan yang terdapat dalam sistem pakar ini. Sebelum menganalisis jawaban, sistem terlebih dahulu memberikan sejumlah pertanyaan kepada *user* melalui *interface* tentang gejala penyakit yang diderita. Sistem akan menganalisis jawaban dari setiap *user* dengan melakukan proses pelacakan pada basis pengetahuan.

Joko S Dwi Raharjo, Damdam Damiyana, Supardi (2016), Sistem Pakar Diagnosa Penyakit Kulit Dengan Menggunakan Metode *Forward Chaininng* Berbasis Android. Di era digital saat ini komputer digunakan untuk mengolah pengetahuan sehingga proses pengambilan keputusan dapat lebih cepat, efisien dan akurat. Sebuah teknik untuk membuat komputer dapat mengolah pengetahuan telah dikenal sebagai teknik kecerdasan buatan (*artificial intelligence technique*). Dengan teknologi ini maka komputer dapat melakukan hal-hal yang sebelumnya hanya dapat dilakukan oleh manusia. Sistem pakar ini menggunakan *Forward Chaining*. *Forward Chaining* merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Dalam melakukan proses *Forward Chaining*, perlu suatu kumpulan aturan atau (*rules*), aturan yang ada ditelusuri satu

persatu hingga penelusuran dihentikan karena kondisi terakhir telah terpenuhi. Dalam melakukan perancangan aplikasi ini menggunakan metode *Rational Unified Model (RUP)*. Metode *RUP* merupakan metode rekayasa perangkat lunak yang berfokus mengembangkan dengan model *Unified Model Language (UML)* dan menggunakan konsep Object Oriented. Hasil rancangan sistem ini dalam bentuk aplikasi *mobile android* yang dapat digunakan oleh semua orang yang ingin mengetahui tentang gejala penyakit kulit yang disebabkan oleh virus. Perancangan sistem pakar inimenggunakan bahasa pemrograman *Android Studio*.

I putu Bayu Krisnawan, I Ketut Gede Darma Putra, I Putu Agung Bayupati (2014), Sistem Pakar Diagosa Penyakit Kulit dan Kelamin Dengan Metode *Certainty Factor* dan *Fuzzy Logic*, penyakit kulit dan kelamin menjadi sangat marak di kalangan masyarakat di berbagai tingkatan di negara Indonesia. Puskesmas atau rumah sakit negri jarang menyediakan fasilitas pengobatan kulit dan kelamin, mengakibatkan orang miskin harus berobat ke dokter spesialis yang memakan banyak sekali biaya. Sistem Pakar Penyakit Kulit dan Kelamin dapat memberikan hasil diagnosa penyakit kulit berupa nama penyakit yang di derita oleh pasien beserta keterangan informasi penyakit dan langkah-langkah yang harus di lakukan dalam menyembuhkan penyakit yang di derita. Sistem Pakar Penyakit Kulit dan Kelamin dapat memberikan informasi presentase kemungkinan pasien mengidap suatu penyakit dengan tingkat rata-rata akurasi diagnosa menurut pakar 73%.

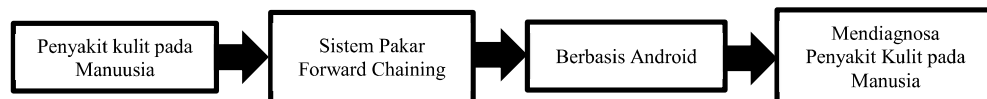
Fitri Nuraeni (2016), Aplikasi Pakar Untuk Diagnosa Penyakit Kulit Menggunakan Metode Forward Chaining di Al Arif Skin *Care* Kabupaten Ciamis.

Berbagai penyakit dan gangguan pada kulit dapat disebabkan oleh beberapa faktor seperti perubahan iklim, lingkungan dan kesehatan yang buruk, virus bakteri, daya tahan tubuh, reaksi alergi dan lain-lain. Penentuan penyakit kulit tidak boleh dilakukan secara sembarangan, karena penyakit kulit bisa sangat berbahaya bila terjadi kesalahan dalam perawatan dan penanganannya. Oleh sebab itu, konsultasi mengenai penyakit kulit harus dilakukan dengan dokter ahli atau pakar. Permasalahan yang sering muncul adalah ketersediaan dokter ahli atau pakar yang memiliki pengetahuan di bidang tertentu cukup terbatas sementara banyak pasien yang harus segera diketahui penyakitnya dan segera ditangani. Metode penelitian yang digunakan dalam penelitian ini adalah metode *Forward Chaining*, yang dapat memberikan informasi berupa gambar dan kata-kata. Sehingga aplikasi sistem pakar ini menjadi media konsultasi bagi pasien penyakit kulit.

Linda Marlinda (2015), Sistem Pakar Diagnosa Penyakit Kulit Pada Manusia Menggunakan Apotek Hidup Menggunakan Simple Additive Weighting. Pengobatan dengan menggunakan apotek hidup sekarang ini banyak diminati oleh masyarakat, karena harganya terjangkau dan mudah didapat. Namun banyak dari masyarakat yang meracik obat-obat tradisional hanya berdasarkan perkataan orang lain atau pengalaman sendiri, sehingga menyebabkan salah penggunaan racikan dan dosis yang kurang tepat dalam pengolahannya khususnya untuk mendeteksi penyakit kulit yang sering terjadi dalam kehidupan sehari-hari. Metode yang digunakan dalam penelitian ini menggunakan metode simple additive weighting yang merupakan salah satu metode dari *attribute decision making* atau banyaknya pemilihan kriteria dari rating kinerja pada setiap alternatif.

2.5 Kerangka Pemikiran

Kerangka pemikiran memuat pemikiran terhadap alur yang dipahami sebagai acuan dalam pemecahan masalah yang diteliti secara logis dan sistematis. Kerangka berfikir yang baik akan menjelaskan secara teoritis pertautan antar variabel yang diteliti (Sugiyono, 2014: 60). Berikut ini adalah kerangka pemikiran yang mendasari penelitian ini.



Gambar 2.16 Kerangka Pemikiran
(Sumber: Data Penelitian: 2016)

Data-data yang dibutuhkan berkaitan dengan penyakit kulit yang dianalisis terlebih dahulu agar lebih sederhana dan mudah dilakukan proses pengolahan datanya. Data-data tersebut kemudian diolah menggunakan metode sistem pakar *forward chaining* untuk membuat aturan (*rule*) yang akan digunakan. Sistem pakar dengan metode *forward chaining* dibuat menggunakan bahasa pemrograman *java* yang ditampilkan dalam bentuk layout dengan format xml. Sistem pakar ini dibuat berdasarkan basis pengetahuan yang ditanamkan pada *java class*, sehingga aplikasi ini merupakan aplikasi *stand alone* yang tidak memerlukan aplikasi pihak kedua seperti SQLite untuk membangun database.

BAB III

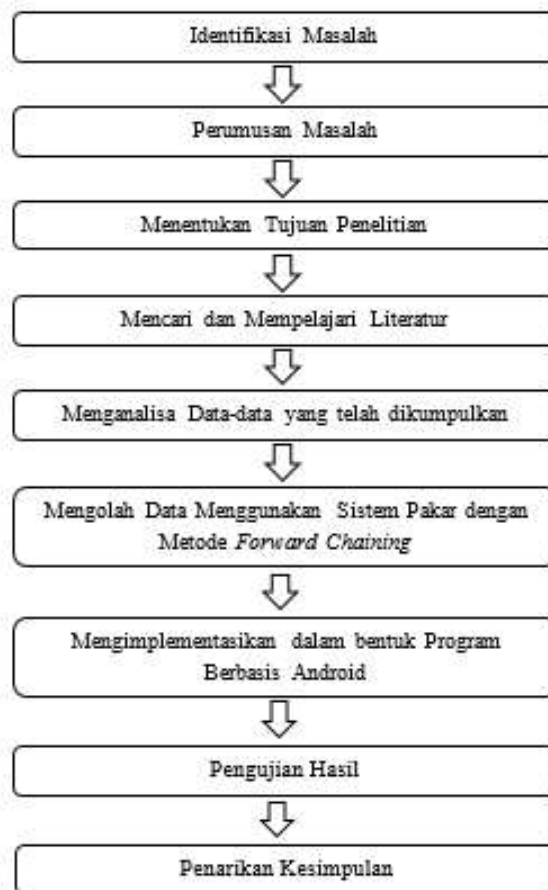
METODE PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Data yang telah diperoleh dari penelitian dapat digunakan untuk memahami, memecahkan dan mengantisipasi suatu masalah (Sugiyono, 2014: 2-3).

3.1 Desain Penelitian

Menurut Sarwono (2006: 27) desain penelitian merupakan alat yang akan menentukan berhasil atau tidaknya suatu penelitian yang sedang dilakukan. Desain penelitian yang baik akan mendukung jalannya penelitian dengan baik pula. Desain penelitian berfungsi sebagai penuntun bagi peneliti yang akan menentukan arah berlangsungnya proses penelitian secara benar dan tepat sesuai dengan tujuan yang ditetapkan. Desain penelitian yang benar adalah desain yang terhindar dari sumber potensial kesalahan dalam proses penelitian secara keseluruhan seperti kesalahan dalam perencanaan, pengumpulan data, melakukan analisis data, dan kesalahan dalam pelaporan hasil penelitian. Tanpa desain penelitian yang benar, peneliti tidak mempunyai pedoman arah penelitian yang jelas sehingga penelitian tidak dapat dilakukan dengan baik (Sarwono, 2006: 79).

Penelitian ini menggunakan desain penelitian dengan beberapa tahap proses penelitian seperti yang terlihat pada gambar di bawah ini.



Gambar 3.1 Desain Penelitian
(Sumber: Data Penelitian, 2016)

Berikut ini adalah penjelasan dari desain penelitian yang ada pada gambar di atas:

1. Identifikasi permasalahan

Penelitian diawali dengan melakukan studi pendahuluan untuk mengidentifikasi permasalahan yang berkaitan dengan topik penelitian agar peneliti mendapatkan apa yang sesungguhnya menjadi masalah untuk dipecahkan.

2. Perumusan masalah

Pada tahap ini, peneliti merumuskan masalah yang telah didapatkan secara lebih spesifik agar masalah tersebut dapat dijawab dengan baik melalui penelitian.

3. Menentukan tujuan penelitian

Peneliti menentukan tujuan penelitian yaitu mengetahui bagaimana sistem pakar untuk mendiagnosa penyakit kulit pada manusia menggunakan metode *forward chaining* berbasis *android*.

4. Mencari dan mempelajari literatur

Untuk mendukung jalannya penelitian, peneliti mencari dan mempelajari sumber-sumber pengetahuan berupa buku-buku teori, jurnal-jurnal penelitian, dan sumber pustaka otentik lainnya yang berkaitan dengan penelitian, diantaranya yaitu kecedasan buatan, sistem pakar, jenis-jenis penyakit kulit, *android* dan *UML*.

5. Menganalisa data-data yang telah didapatkan

Setelah data-data yang berkaitan dengan penyakit kulit didapatkan baik melalui studi literatur maupun wawancara dengan dokter sebagai pakarnya, peneliti menganalisa data-data yang dibutuhkan dalam sistem pakar kemudian data-data tersebut disederhanakan dan dikelompokkan agar lebih mudah dilakukan proses pengolahan datanya.

6. Mengolah data menggunakan sistem pakar dengan metode *forward chaining*

Sistem pakar pada penelitian ini menggunakan model representasi pengetahuan berbasis kaidah produksi. Sistem pakar dapat menghasilkan suatu kesimpulan berdasarkan aturan atau kaidah yang ada. Oleh karena itu, data-data yang telah dianalisa kemudian diolah menggunakan metode *forward chaining*

untuk membuat kaidah (*rule*) yang akan digunakan saat sistem pakar melakukan penelusuran sebelum menyimpulkan hasil.

7. Mengimplementasikan dalam bentuk program berbasis *android*

Pada tahap ini, peneliti melakukan kegiatan perancangan mulai dari desain basis pengetahuan, desain *UML* dan desain antarmuka. Setelah itu dilakukan pengodean untuk mentranslasikan desain yang telah dibuat ke dalam program perangkat lunak sehingga menghasilkan sebuah program aplikasi. Pengodean dilakukan menggunakan bahasa pemrograman *Java*.

8. Pengujian hasil

Proses ini bertujuan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Pengujian dilakukan dengan menggunakan salah satu pendekatan pengujian untuk validasi yaitu *black-box testing*. Sistem juga diuji dengan membandingkan hasil diagnosa pakar dengan hasil diagnosa sistem untuk melihat apakah sistem telah berjalan dengan baik.

9. Menarik kesimpulan

Tahapan terakhir dalam penelitian ini yaitu menyimpulkan hasil penelitian yang berisi jawaban singkat terhadap rumusan masalah berdasarkan data-data yang ada. Dalam tahap ini, peneliti juga memberikan saran yang penting untuk membantu dalam memecahkan permasalahan yang ada.

3.1.1 Teknik pengumpulan data

Teknik pengumpulan data merupakan cara-cara yang digunakan peneliti dalam mendapatkan data-data yang berkaitan dengan pokok bahasan dalam rangka untuk mendukung penelitian yang sedang dilakukan. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah:

1. Wawancara

Untuk mendapatkan data-data yang berkaitan dengan penelitian, peneliti melakukan wawancara langsung dengan Dokter Indrawan yang bertugas di Rumah Sakit Charis Medica. Dalam metode wawancara, alat bantu yang digunakan peneliti berupa alat tulis untuk menulis informasi dari pembicaraan selama proses wawancara dilakukan. Pedoman wawancara yang digunakan berupa garis-garis besar permasalahan yang akan ditanyakan yaitu hal-hal yang berkaitan dengan penyakit kulit dan pencegahannya.

2. Studi literatur

Peneliti melakukan studi literatur dengan mengumpulkan, membaca, dan memahami referensi teoritis yang berasal dari buku-buku teori, buku elektronik (*e-book*), jurnal-jurnal penelitian, dan sumber pustaka otentik lainnya yang berkaitan dengan penelitian. Studi literatur bertujuan untuk menemukan variabel yang akan diteliti, membedakan hal-hal yang sudah dilakukan dan menentukan hal yang perlu dilakukan, melakukan sintesa dan memperoleh perspektif baru, dan menentukan makna dan hubungan antar variabel (Sarwono, 2006: 47).

3.2 Operasional Variabel

Variabel harus didefinisikan secara operasional agar lebih mudah dicari hubungannya antara satu variabel dengan lainnya dan pengukurannya. Adapun manfaat operasionalisasi variabel antara lain: untuk mengidentifikasi kriteria yang dapat diobservasi yang sedang didefinisikan, menunjukkan bahwa suatu konsep atau objek mungkin mempunyai lebih dari satu definisi operasional, dan untuk mengetahui bahwa definisi operasional bersifat unik dalam situasi dimana definisi tersebut harus digunakan. Definisi operasional adalah suatu definisi yang didasarkan pada karakteristik yang dapat diobservasi dari apa yang sedang didefinisikan (Sarwono, 2006: 67). Variabel yang digunakan dalam penelitian ini adalah penyakit kulit.

Tabel 3.1 Tabel Indikator

Variabel	Indikator
Penyakit Kulit	Eksim
	Kurap
	Panu
	Bisul
	Herpes

Sumber: Data Penelitian (2017)

Indikator dalam penelitian ini 5 indikator yaitu penyakit kulit eksim, kurap, panu, bisul, dan herpes, seperti yang dijelaskan pada tabel 3.1 Tabel indikator.

3.3 Perancangan Sistem

Perancangan sistem merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performa maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat (A.S. dan Shalahuddin, 2013: 23).

3.3.1 Desain basis pengetahuan

Sebelum melakukan desain basis pengetahuan, proses akuisisi pengetahuan dengan mengumpulkan pengetahuan dan fakta dari sumber-sumber yang tersedia telah dilakukan. Sumber pengetahuan dan fakta diperoleh melalui wawancara dengan dokter dan studi literatur tentang materi yang berkaitan dengan penyakit kulit pada manusia. Sumber pengetahuan dan fakta yang didapat berupa data-data yang berhubungan dengan jenis penyakit kulit, gejala penyakit kulit, dan solusi mengatasinya. Pengetahuan dan fakta tersebut ditampilkan dalam tabel variabel (Tabel 3.2), gejala penyakit kulit (Tabel 3.3), dan tabel aturan penyakit kulit (Tabel 3.4).

Tabel 3.2 Tabel Variabel

Kode Penyakit	Nama Penyakit	Solusi
P01	Eksim	1. Gunakan salep yang cocok. 2. Jangan menggaruk kulit 3. Hindari penggunaan kosmetik dan obat-obatan.

Tabel 3.2 Lanjutan

P02	Kurap	<ol style="list-style-type: none"> 1. Cuci tangan dengan bersih 2. Menjaga kebersihan tubuh 3. Menghindari kontak langsung dengan penderita.
P03	Panu	<ol style="list-style-type: none"> 1. Jaga kebersihan anggota tubuh 2. Selalu mengonsumsi vitamin C. 3. Biasakan tidak memakai peralatan mandi secara bergantian.
P04	Bisul	<ol style="list-style-type: none"> 1. Jaga kebersihan tubuh agar tetap bersih. 2. Jangan menggaruk kulit.
P05	Herpes	<ol style="list-style-type: none"> 1. Pemberian Vaksinasi 2. Mengonsumsi buah-buahan yang mengandung vitamin C.

Sumber: Data Penelitian (2017)

Berikut ini akan disajikan tabel gejala untuk setiap penyakit. Tabel gejala ini disusun berdasarkan data penelitian seperti pada tabel 3.3 berikut:

Tabel 3.3 Tabel Gejala

Kode Gejala	Nama Gejala
G01	Kulit terasa gatal
G02	Kulit kering
G03	Kulit tampak merah
G04	Kulit meradang
G05	Terdapat lepuhan-lepuhan kecil
G06	Kulit bersisik dan menebal
G07	Terjadi pembengkakan
G08	Tampak bintik-bintik merah
G09	Nyeri pada sendi
G10	Kulit bernanah
G11	Terasa gatal pada saat berkeringat

Tabel 3.3 Lanjutan

G12	Terasa perih jika terkena air
G13	Muncul ruam pada kulit
G14	Kulit bersisik halus

Sumber: Data Penelitian (2017)

Data aturan merupakan data yang berisi relasi antara data-data bagian kerusakan, penyebab kerusakan dan gejala kerusakan yang telah diberi kode sebelumnya. Relasi antar data tersebut disusun berdasarkan sumber pengetahuan dan fakta yang telah didapatkan. Data aturan ini disusun untuk memudahkan peneliti dalam menyusun kaidah yang akan digunakan sebagai basis pengetahuan dalam sistem pakar pada penelitian ini. Susunan data aturan yang digunakan dapat dilihat pada tabel dibawah ini.

Tabel 3.4 Tabel Aturan

Kode Gejala	Kode Penyakit
G01, G04, G05, G06	P01
G01, G02, G11, G12	P02
G01, G13, G14	P03
G03, G08, G10	P04
G08, G09	P05

Sumber: Data Penelitian (2017)

Berdasarkan tabel keputusan tersebut maka pohon keputusannya adalah sebagai berikut: Berdasarkan data aturan yang telah disusun, maka kaidah (*rule*) yang akan digunakan dalam sistem pakar adalah sebagai berikut:

1. Kaidah 1: *IF G01 AND G04 AND G05 AND G06 THEN P01*
2. Kaidah 2: *IF G01 AND G02 AND G11 AND G12 THEN P02*
3. Kaidah 3: *IF G01 AND G13 AND G14 THEN P03*
4. Kaidah 4: *IF G03 AND G07 AND G10 THEN P04*
5. Kaidah 4: *IF G08 AND G09 THEN P04*

Berdasarkan kaidah (*rule*) yang telah dibuat maka dapat dijelaskan bahwa:

1. Jika gejala penyakit yang dirasakan adalah rasa gatal, kulit meradang, terdapat lepuhan-lepuhan kecil dan bersisik tebal maka jenis penyakit yang diderita adalah Eksim.
2. Jika gejala penyakit yang dirasakan adalah rasa gatal, kulit kering, gatal pada saat berkeringat dan terasa perih saat terkena air maka jenis penyakit yang diderita adalah penyakit kurap.
3. Jika gejala penyakit yang dirasakan adalah terasa gatal, muncul ruam pada kulit dan bersisik halus maka jenis penyakit yang diderita adalah jenis penyakit kulit panu.
4. Jika gejala penyakit yang dirasakan adalah timbul kemerahan pada kulit, terjadi pembengkakan pada kulit dan bernanah maka jenis penyakit yang diderita adalah jenis penyakit kulit bisul.

5. Jika gejala penyakit yang dirasakan adalah tampak bintik-bintik merah dan merasakan nyeri pada sendi maka jenis penyakit kulit yang diderita adalah penyakit kulit herpes.

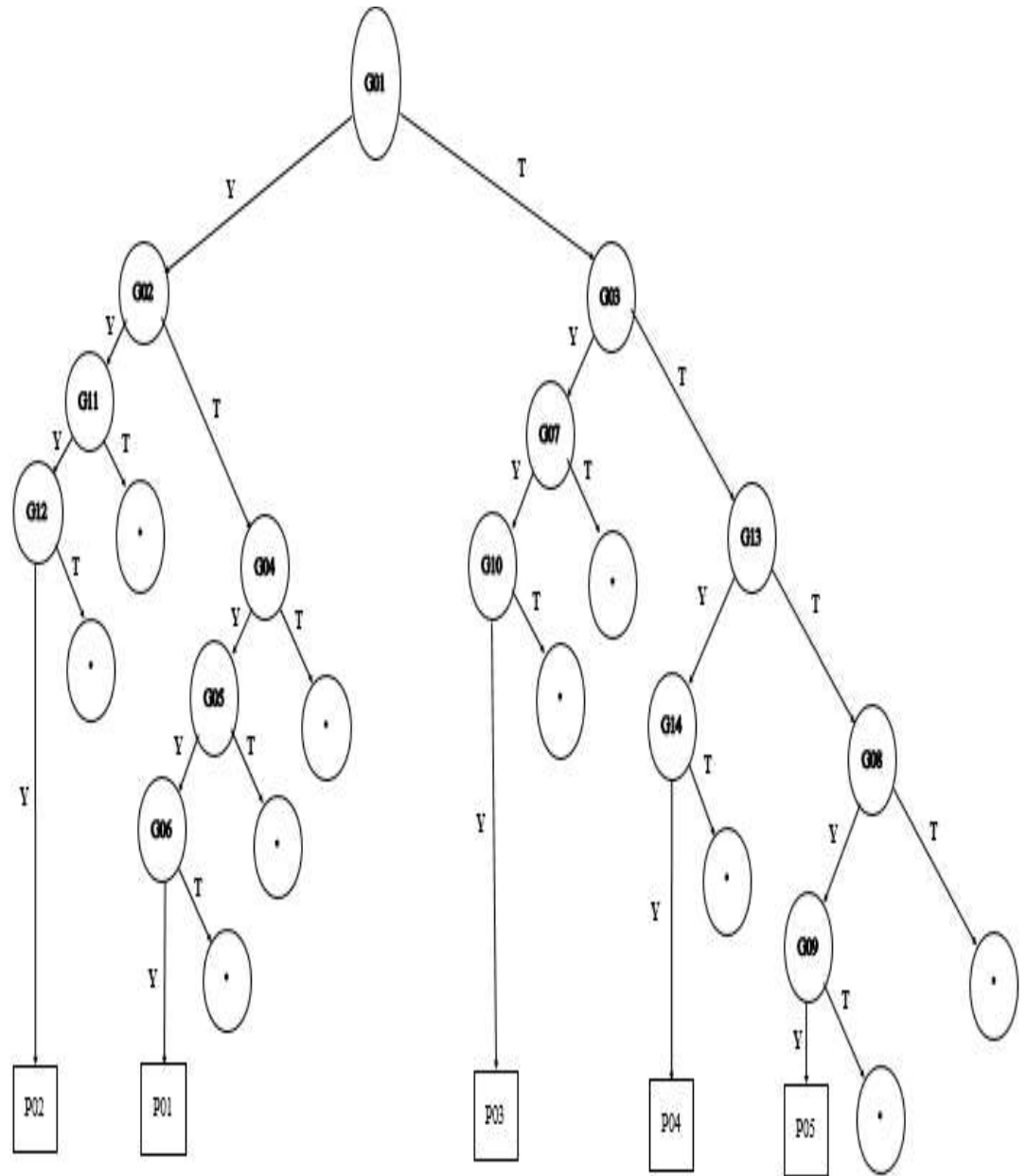
Berdasarkan kaidah yang telah dibuat tersebut maka tabel keputusannya adalah sebagai berikut:

Tabel 3.5 Tabel keputusan

Gejala \ Penyakit	P01	P02	P03	P04	P05
G01	✓	✓	✓		
G02		✓			
G03				✓	
G04	✓				
G05	✓				
G06	✓				
G07				✓	
G08					✓
G09					✓
G10				✓	
G11		✓			
G12		✓			
G13			✓		
G14			✓		

Sumber: Data Penelitian (2017)

Berdasarkan tabel keputusan tersebut maka pohon keputusannya adalah sebagai berikut:

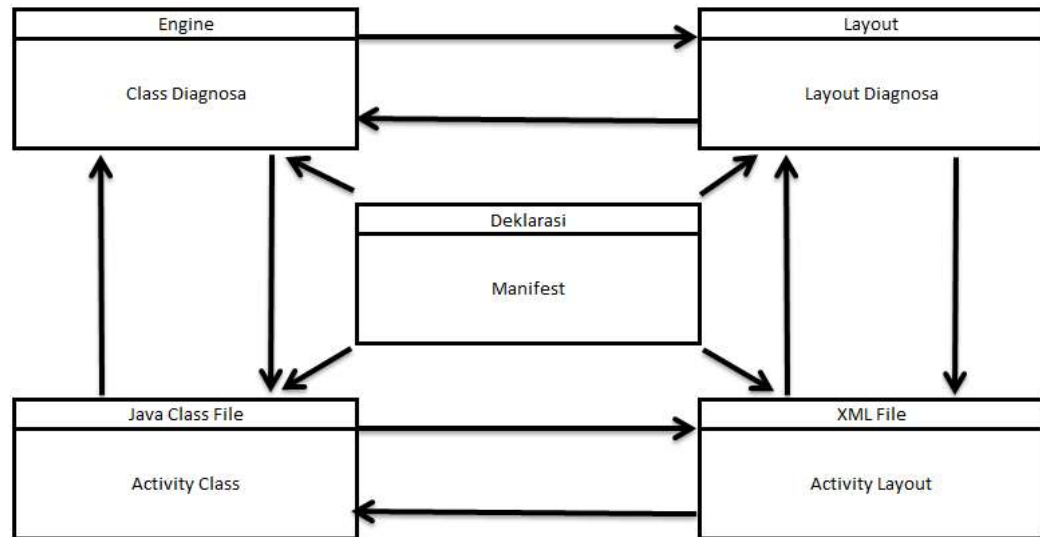


Gambar 3.2 Pohon Keputusan
(Sumber: Data Penelitian 2017)

Data gejala ditentukan sebagai keadaan awal dalam sistem saat melakukan penelusuran sebelum diperoleh sebuah kesimpulan. Pohon keputusan pada gambar 3.2 digunakan untuk memperlihatkan hubungan terkait antar gejala yang ada. Arah penelusuran pada pohon keputusan tersebut dimulai dari simpul akar (yang paling atas) ke bawah. Alur penelusuran sistem pakar dimulai dari G01 yaitu rasa gatal. Gejala ini dipilih sebagai keadaan awal dalam penelusuran karena gejala ini adalah gejala yang paling mudah diperiksa dan diketahui.

Proses penelusuran selanjutnya tergantung bagaimana jawaban yang diberikan pengguna. Jika pengguna memberikan jawaban “ya”, maka penelusuran menuju simpul kiri pada level berikutnya (G02) dan jika pengguna memberikan jawaban “tidak”, maka penelusuran menuju simpul kanan pada level berikutnya (G03). Begitu seterusnya sampai penelusuran menemukan simpul G atau simpul *. Simpul G berasosiasi dengan simpul P yang berarti bahwa simpul G tersebut merupakan bagian dari P. Misalnya G01, yaitu suhu badan panas dan dingin yang berlebihan maka penyakit yang dimaksud berada pada bagian P01, yaitu Eksim. Simpul * berarti tidak menghasilkan kesimpulan tertentu. Pada sistem pakar ini, jika penelusuran menemukan simpul * maka sistem akan kembali melakukan penelusuran mulai dari keadaan awal (simpul G01).

Dalam penelitian ini, peneliti membuat desain data base seperti berikut:



Gambar 3.3 Desain Data Base
(Sumber: Data Penelitian, 2017)

Tabel yang digunakan dalam sistem pakar ini terdiri dari 5 tabel, yaitu tabel *Class Diagnosa* sebagai *Engine*, *Layout Diagnosa* sebagai tampilan, *Manifest* sebagai Deklarasi, *Activity Class* sebagai *Java Class File* dan *Activity Layout* sebagai *XML File*. Awalnya *Manifest* akan memperkenalkan atau mendeklarasikan *Class Java* kemudian *XML File* sebagai *Layout* dihubungkan ke *Main Activity*. Lalu *Main Activity* dihubungkan kembali ke *Class Java* lainnya disertai dengan *XML File*. Semua tabel saling berhubungan satu dengan yang lainnya untuk menghasilkan aplikasi sistem pakar diagnosa penyakit kulit.

3.3.2 Struktur kontrol (mesin inferensi)

Mesin inferensi dalam sistem pakar ini menggunakan metode penelusuran *forward chaining*. Langkah-langkah yang digunakan dalam proses penelusurannya adalah sebagai berikut:

1. Mengajukan pertanyaan tentang gejala kerusakan kepada pengguna.
2. Jika jawaban pengguna “Ya” maka sistem akan melakukan langkah 3. Jika jawaban pengguna “Tidak” maka sistem akan melakukan langkah 4.
3. Menyimpan gejala dalam memori kerja lalu memeriksa kombinasi gejala dengan aturan yang telah dibuat. Jika ada aturan yang cocok maka sistem akan melakukan langkah 5. Jika tidak ada aturan yang cocok maka sistem akan melakukan langkah 4.
4. Memeriksa apakah masih ada gejala lain yang belum ditanyakan. Jika masih ada, maka sistem akan mengajukan pertanyaan tentang gejala penyakit, selanjutnya kepada pengguna dan ulangi langkah 2 sampai dengan 4. Jika tidak ada, maka sistem akan melakukan langkah 5.
5. Menampilkan hasil diagnosa.

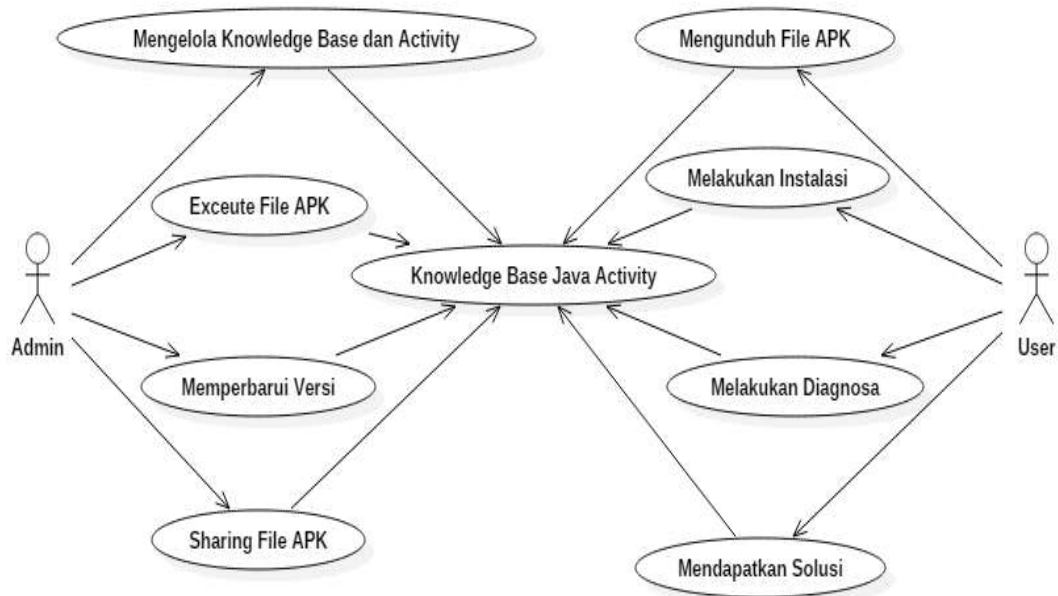
3.3.3 Desain UML (*Unified Modeling Language*)

Desain sistem pada penelitian ini menggunakan bahasa pemodelan *Unified Modelling Language (UML)* yang digambarkan dengan bantuan aplikasi *StarUML*.

Diagram *UML* dalam penelitian ini yaitu:

1. Use case diagram

Menurut AS aktor yang digunakan dalam sistem pakar ini terdiri dari 2 orang yaitu administrator dan pengguna. *Use case* yang terdapat dalam sistem antara lain *Log In*, mengelola daftar pengguna, mengelola daftar administrator, mengelola data penyebab, mengelola data gejala, mengelola data aturan, pendaftaran dan diagnosa. *Use case diagram* yang dirancang untuk sistem pakar dalam penelitian ini dapat dilihat pada gambar dibawah ini.



Gambar 3.3 Use case diagram
(Sumber: Data Penelitian, 2017)

Pada Gambar 3.3 diatas, terdapat 2 aktor yaitu *admin* dan *user*. *Admin* melakukan interaksi dengan sistem berupa mengelola *knowledge base* dan *activity*, *execute file APK*, *memperbarui versi* dan *sharing file APK*. Sedangkan *user*

berinteraksi dengan sistem yaitu mengunduh *file APK*, melakukan *instalasi*, melakukan diagnosa dan mendapatkan solusi. Diagnosa dilakukan dengan cara menjawab pertanyaan yang diajukan oleh sistem, setelah semua jawaban sesuai *rule*, maka sistem akan menampilkan solusi permasalahan. Semua interaksi yang dilakukan oleh *admin* dan *user* terhubung ke *knowledge base java activity*.

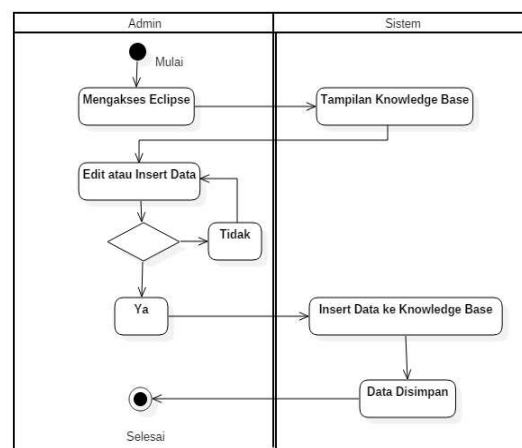
2. Activity Diagram

Activity diagram menggambarkan *workflow* dari system. *Activity diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor (A.S. dan Shalahuddin, 2011: 134). Adapun *Activity diagram* dalam penelitian ini adalah :

2.1 Admin

a. Activity diagram mengelola data base

Activity diagram mengelola *knowledge base* merupakan *UML* yang menggambarkan kegiatan Admin dalam mengelola *data base*. Berikut ini gambar *activity diagram* mengelola *data base*:

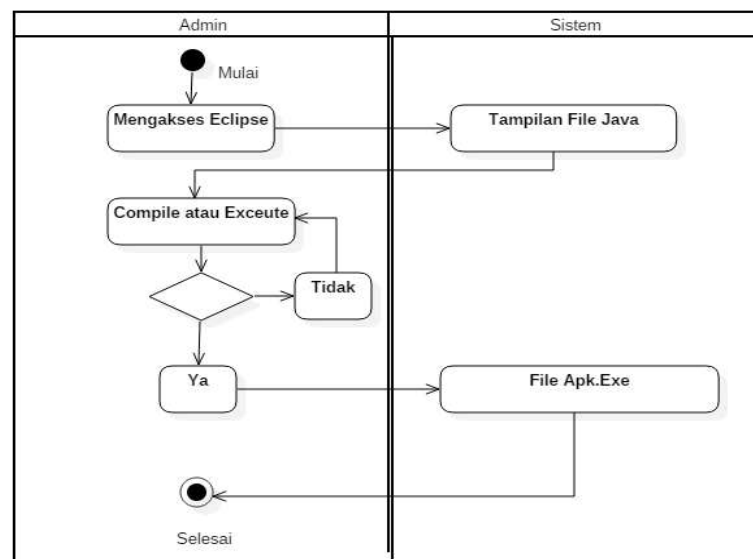


Gambar 3.4 Activity diagram mengelola data base
(Sumber: Data Penelitian, 2017)

Pada **Gambar 3.4** di atas, admin akan mulai mengakses eclipse dan sistem akan menampilkan *data base*. Admin akan melakukan *edit* atau *insert* data, jika ya maka sistem akan menampilkan *insert* data ke *data base* lalu data disimpan dan admin selesai mengelola *data base*.

b. *Activity diagram execute file Apk*

Activity diagram execute file Apk merupakan *UML* yang menggambarkan kegiatan Admin dalam *execute file apk*. Berikut ini gambar *Activity diagram execute*:

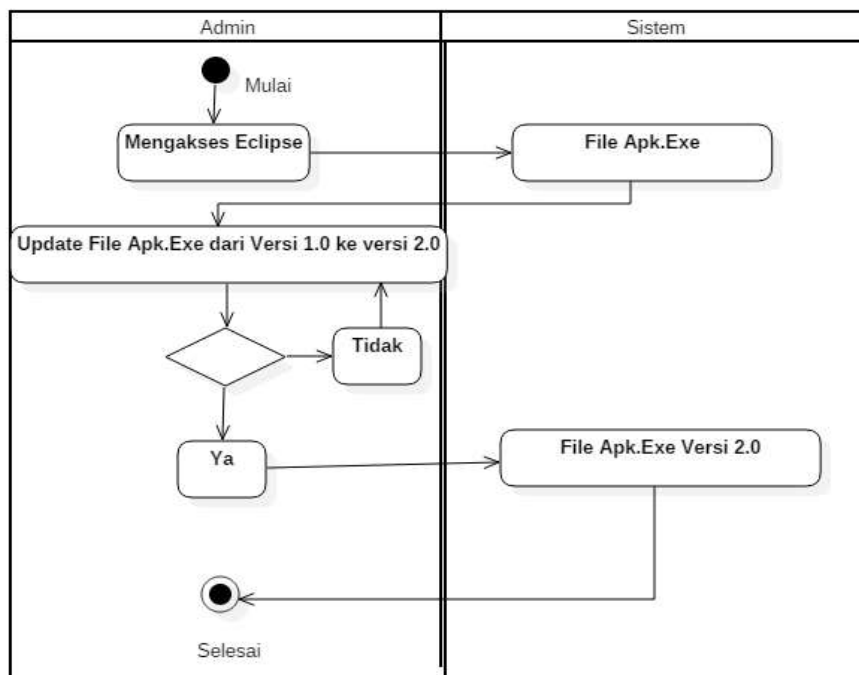


Gambar 3.5 *Activity diagram execute file Apk*
(Sumber: Data Penelitian, 2017)

Pada **Gambar 3.5** di atas admin mulai mengakses eclipse dan sistem menampilkan *file java*. Admin akan *mengcompile* atau *execute* dan jika ya maka sistem akan membuat *file apk.exe*. Admin selesai melakukan *execute file apk*.

c. *Activity diagram* memperbarui versi

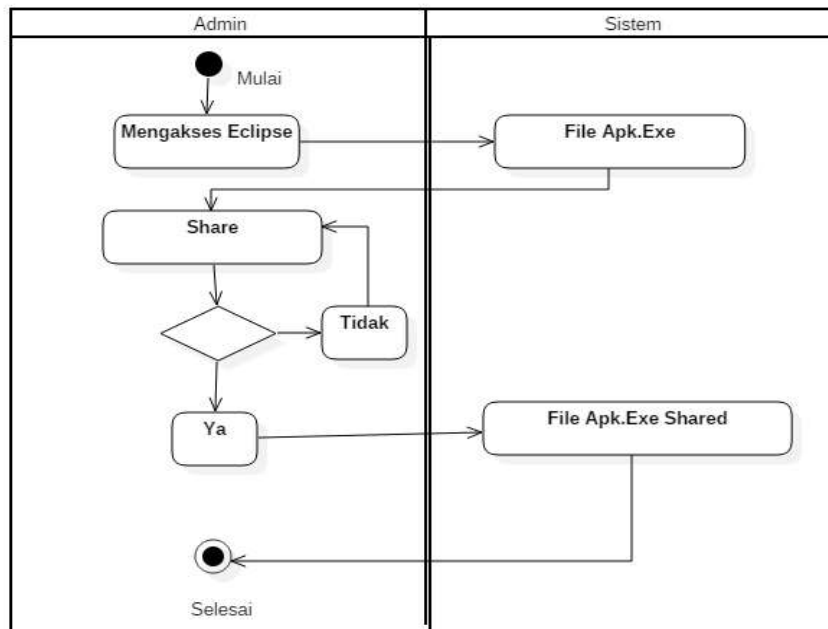
Activity diagram memperbarui versi merupakan *UML* yang menggambarkan kegiatan Admin dalam memperbarui versi. Berikut ini gambar *activity diagram* memperbarui versi:



Gambar 3.6 *Activity diagram* memperbarui versi
(Sumber: Data Penelitian, 2017)

d. *Activity diagram* sharing file apk

Activity diagram sharing file apk memperbarui versi merupakan *UML* yang menggambarkan kegiatan Admin dalam memperbarui versi. Berikut ini gambar *activity diagram* memperbarui versi:



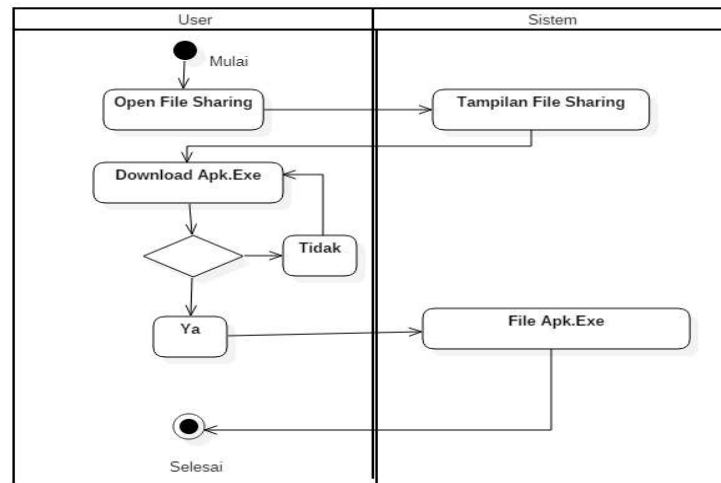
Gambar 3.7 Activity diagram sharing file apk
(Sumber: Data Penelitian, 2017)

Pada **Gambar 3.7** di atas, admin akan mulai mengakses eclipse dan sistem akan menampilkan file apk.exe. Admin akan share, jika ya maka sistem akan menampilkan *file apk.exe shared*. Admin selesai mengelola *data base*.

2.2 User

a. Activity diagram mengunduh file apk

Activity diagram mengunduh *file apk* merupakan *UML* yang menggambarkan kegiatan *User* dalam mengunduh *file apk*. Berikut ini gambar *activity diagram* mengunduh *file apk*:

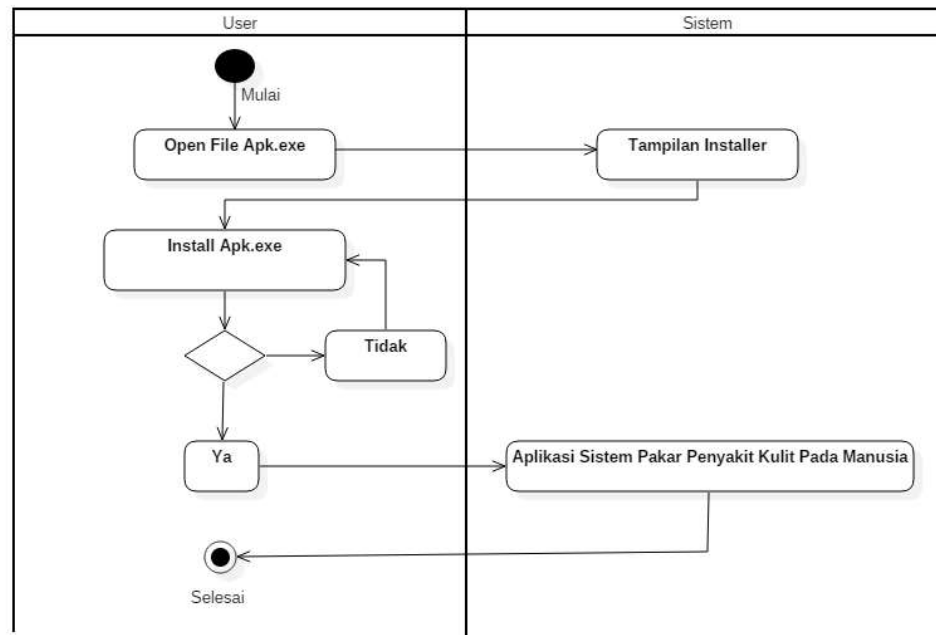


Gambar 3.8 Activity diagram mengunduh file apk
(Sumber: Data Penelitian, 2017)

Pada **Gambar 3.8** di atas, *user* akan mulai *open file sharing* dan sistem akan menampilkan *file sharing*. *User* akan *mendownload Apk.Exe*, jika ya maka sistem akan menampilkan *file Apk.Exe* dan *user* selesai mengunduh *file apk*.

b. Activity diagram melakukan instalasi

Activity diagram melakukan instalasi merupakan *UML* yang menggambarkan kegiatan *User* dalam melakukan instalasi. Berikut ini gambar *activity diagram* melakukan instalasi:

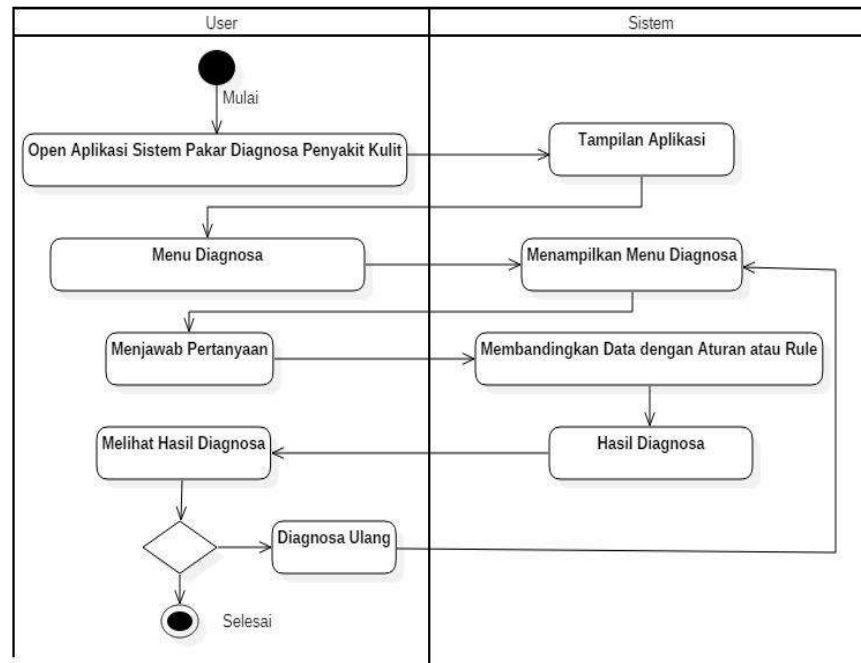


Gambar 3.9 *Activity diagram* melakukan instalasi
(Sumber: Data Penelitian, 2017)

Pada **Gambar 3.9** di atas, *user* akan mulai menginstall aplikasi dan sistem akan menampilkan tampilan installer. *User* akan memilih untuk melanjutkan menginstall atau tidak, jika ya maka sistem akan terinstall dan *user* selesai melakukan instalasi.

c. *Activity diagram* melakukan diagnosa

Activity diagram melakukan diagnosa merupakan *UML* yang menggambarkan kegiatan *User* dalam melakukan diagnosa. Berikut ini gambar *activity diagram* melakukan diagnosa:



Gambar 3.10 *Activity diagram* melakukan diagnosa
(Sumber: Data Penelitian, 2017)

Pada Gambar 3.10 di atas, *user* akan mulai membuka aplikasi dan sistem akan menampilkan tampilan aplikasi. Setelah itu *User* akan membuka menu diagnosa dan sistem akan menampilkan menu diagnosa. Pada menu diagnosa, user akan menjawab pertanyaan yang berupa gejala-gejala penyakit kulit yang telah diinput ke dalam *knowledge base*, jika user menjawab “ya” maka sistem akan membandingkan data dari user dengan *rule* atau aturan yang telah disimpan di dalam *knowledge base*. User selesai menjawab pertanyaan, sistem menampilkan hasil diagnosa dan user selesai melakukan diagnosa.

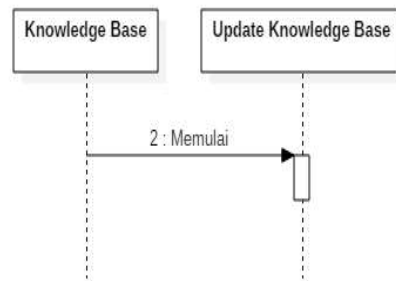
3. *Sequence diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar

objek (A.S. dan Shalahuddin, 2011: 137). Adapun *sequence diagram* dalam penelitian ini adalah :

1. Admin

1.1 *Sequence diagram* mengelola *knowledge base*

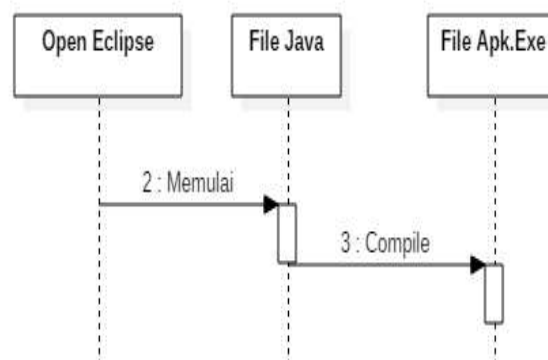


Gambar 3.11 *Sequence diagram* mengelola *knowledge base*
(Sumber: Data Penelitian, 2017)

Admin mengelola *knowledge base* untuk melakukan update data pada *knowledge base*.

1.2 *Sequence diagram* execute file *Apk*

Sequence diagram execute file *Apk* merupakan urutan kegiatan admin saat mengexecute file *Apk*. Berikut ini gambar *sequence diagram* execute file *Apk*:

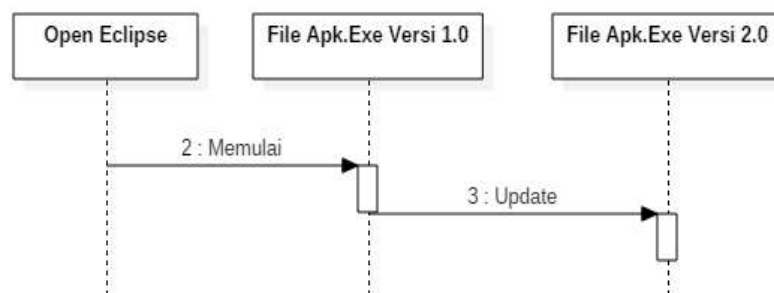


Gambar 3.12 *Sequence diagram* execute file *Apk.exe*
(Sumber: Data Penelitian, 2017)

Admin akan membuka eclipse dan memulai mengolah file *Java* kemudian di compile jadi file *Apk.exe*.

1.3 *Sequence diagram* memperbarui versi

Sequence diagram memperbarui versi merupakan urutan kegiatan admin saat memperbarui versi. Berikut ini gambar *sequence diagram* memperbarui versi:

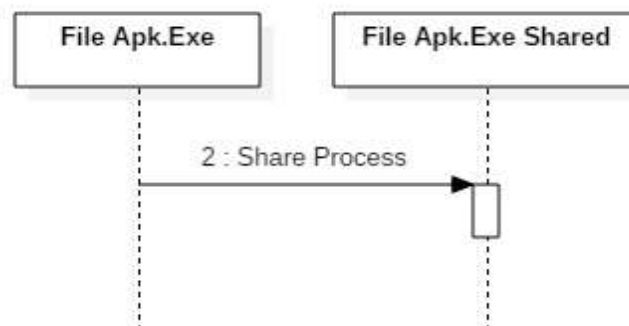


Gambar 3.13 *Sequence diagram* memperbarui versi
(Sumber: Data Penelitian, 2017)

Admin akan membuka eclipse dan memulai file *Apk.Exe* versi 1.0 kemudian admin akan update file *Apk.exe* versi 2.0.

1.4 *Sequence diagram* sharing file apk

Sequence diagram sharing file apk merupakan urutan kegiatan admin saat sharing file apk. Berikut ini gambar *sequence diagram* sharing file apk:



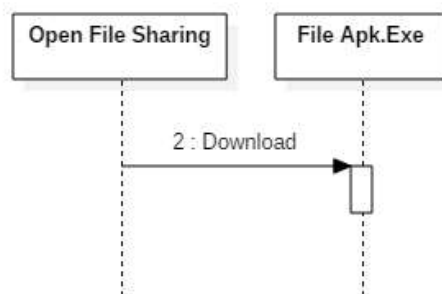
Gambar 3.14 *Sequence diagram* sharing file apk
(Sumber: Data Penelitian, 2017)

Admin akan membuka file Apk.Exe kemudian admin akan *share* file *Apk.Exe Shared*.

2. User

2.1 *Sequence diagram* mengunduh *file Apk*

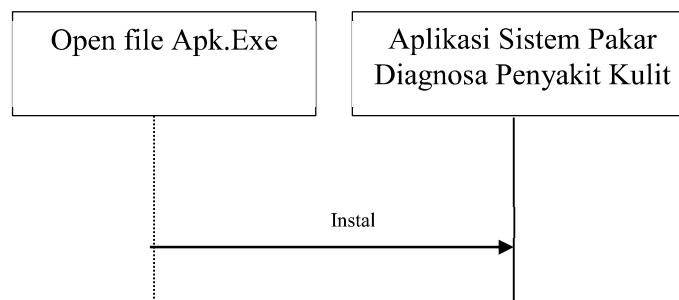
Sequence diagram mengunduh *file Apk* merupakan urutan kegiatan *user* saat mengunduh *file Apk*. Berikut ini gambar *sequence diagram* mengunduh *file Apk*:



Gambar 3.15 *Sequence diagram* mengunduh *file Apk*
(Sumber: Data Penelitian, 2017)

2.2 *Sequence diagram* melakukan instalasi

Sequence diagram melakukan instalasi merupakan urutan kegiatan *user* saat melakukan instalasi. Berikut ini gambar *sequence diagram* melakukan instalasi:

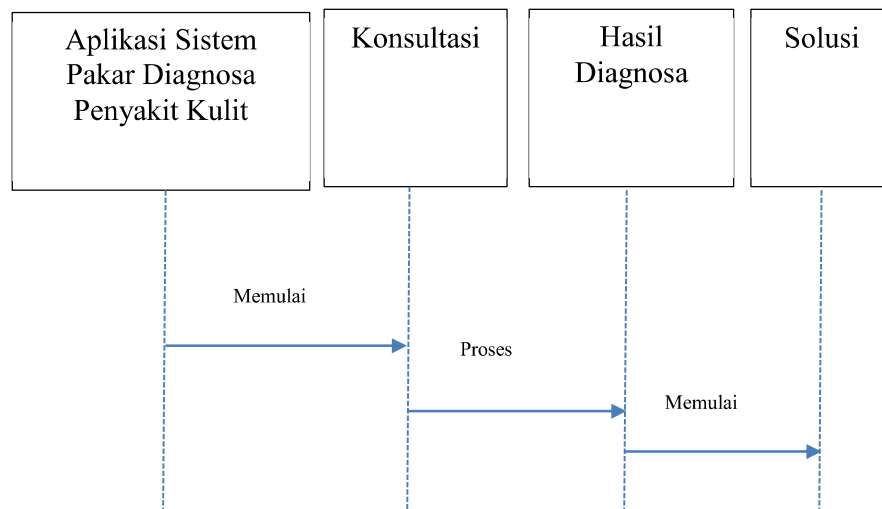


Gambar 3.16 *Sequence diagram* melakukan instalasi
(Sumber: Data Penelitian, 2017)

User akan membuka file *Apk.Exe* kemudian *user* menginstal Aplikasi Sistem Pakar Diagnosa Penyakit Kulit.

2.3 *Sequence diagram* melakukan diagnosa

Sequence diagram melakukan diagnosa merupakan urutan kegiatan *user* saat melakukan diagnosa. Berikut ini gambar *sequence diagram* melakukan diagnosa:



Gambar 3.17 *Sequence diagram* melakukan diagnosa
(Sumber: Data Penelitian, 2017)

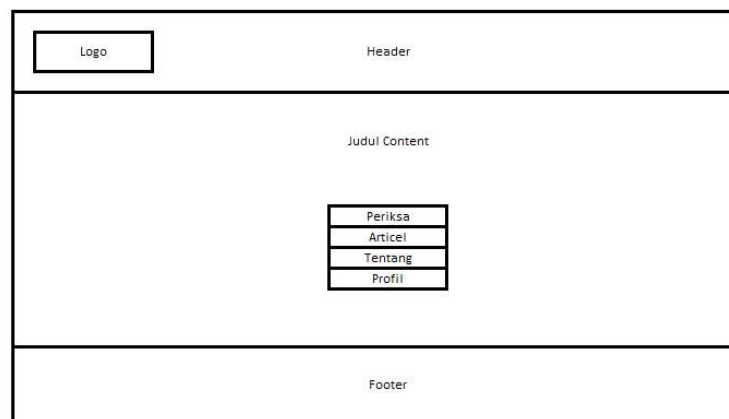
User akan membuka Aplikasi Sistem Pakar Diagnosa Penyakit Kulit dan memulai konsultasi, setelah menjawab pertanyaan yang ada akan diproses dan sistem akan menampilkan hasil diagnosa dan solusi.

3.3.4 Desain antarmuka

Adapun desain tampilan sistem pakar mendiagnosa penyakit kulit adalah sebagai berikut:

1. Rancangan *Form* Beranda

Form Beranda memiliki beberapa tampilan, yaitu, *header*, logo, judul *content*, menu utama dan *text area*.



Gambar 3.17 Rancangan *Form* Beranda
(Sumber: Data Penelitian, 2017)

2. Rancangan *Form* Diagnosa

Form diagnosa digunakan oleh pengguna untuk berkonsultasi dengan sistem pakar. Sistem akan mengajukan beberapa pertanyaan tentang gejala-gejala penyakit kulit yang dialami oleh pengguna.

The diagram shows a form layout for a diagnosis system. It is divided into three horizontal sections:

- Header:** Contains a 'Logo' box on the left and the word 'Header' on the right.
- Main Content:** Starts with the text 'Pertanyaan mengenai gejala penyakit:'. Below this are two buttons: 'Ya' on the left and 'Tidak' on the right.
- Footer:** Contains the word 'Footer' centered at the bottom.

Gambar 3.18 Rancangan *Form* Diagnosa
(Sumber: Data Penelitian, 2017)

3. Rancangan *Form* Hasil Diagnosa

Form Hasil Diagnosa digunakan untuk menampilkan hasil diagnosa yang berisi jenis penyakit kulit dan solusi yang diberikan oleh sistem pakar.

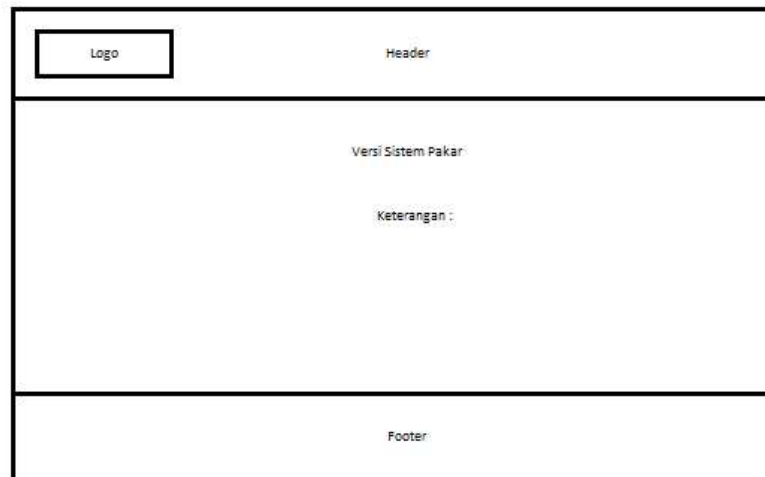
The diagram shows a form layout for displaying diagnosis results. It is divided into three horizontal sections:

- Header:** Contains a 'Logo' box on the left and the word 'Header' on the right.
- Main Content:** Starts with the text 'Jenis Penyakit Kulit' and 'Gambar Penyakit'. Below this is the text 'Solusi Pencegahan :'. At the bottom of this section are two buttons: 'Diagnosa Ulang' on the left and 'Menu Utama' on the right.
- Footer:** Contains the word 'Footer' centered at the bottom.

Gambar 3.19 Rancangan *Form* Hasil Diagnosa
(Sumber: Data Penelitian, 2017)

4. Rancangan *Form* About

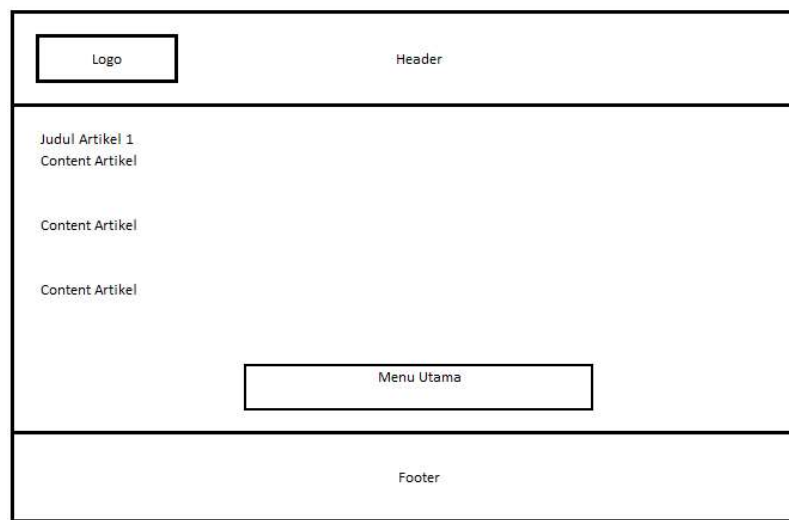
Form tentang berisi versi dari sistem pakar diagnosa penyakit kulit pada manusia dan terdapat keterangan bahwa setiap *knowledge base* diperbarui maka versi akan berubah seperti dari Versi 1.0 ke Versi 2.0 dan seterusnya.



Gambar 3.20 Rancangan *Form About*
(Sumber: Data Penelitian, 2017)

5. Rancangan *Form Artikel*

Form artikel berisi tentang jenis-jenis penyakit kulit, penyebab penyakit kulit, dan pencegahan penyakit kulit.



Gambar 3.21 Rancangan *Form Artikel*
(Sumber: Data Penelitian, 2017)

6. Rancangan *Form* Profil

Form profil merupakan *form* yang berisi data mengenai pembuat dari sistem pakar diagnosa penyakit kulit pada manusia.

Logo	Header
Nama : NPM : Prodi :	Foto
Footer	

Gambar 3.22 Rancangan *Form Profil*
(Sumber: Data Penelitian, 2017)

3.4 Lokasi dan Jadwal Penelitian

3.4.1 Lokasi

Penelitian ini dilakukan melalui wawancara dengan Dokter Indrawan, yang bertugas di Rumah Sakit Charis Medica, Batam. Alasan memilih Dokter Indrawan sebagai narasumber ahli yaitu karena kemampuan Dokter Indrawan dalam menyelesaikan permasalahan penyakit kulit, seperti penyakit kulit eksim, kurap, panu, bisul dan herpes.

3.4.2 Jadwal Penelitian

Setiap rancangan penelitian perlu dilengkapi dengan jadwal kegiatan yang akan dilaksanakan yang berisi jadwal kegiatan apa saja yang akan dilakukan selama

penelitian (Sugiyono, 2014: 286). Berikut ini adalah tabel jadwal kegiatan yang dilakukan selama penelitian berlangsung.

Tabel 3.7 Jadwal Penelitian

Nno	Kegiatan	Tahun 2016/2017																		
		Okt '16				Nov '16				Des '16				Jan '17				Feb '17		
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1		
1	Pengajuan Judul	■	■	■	■															
2	Penyusunan Bab I			■	■	■	■													
3	Penyusunan Bab II			■	■	■	■	■												
4	Penyusunan Bab III									■	■	■	■	■	■					
5	Penyusunan Bab IV															■	■			
6	Penyusunan Bab V, Daftar Pustaka, Lampiran																■	■		

Sumber: Data Penelitian (2017)