

**PERANCANGAN SISTEM PAKAR DIAGNOSIS ASMA DAN  
TERAPINYA DENGAN METODE *FORWARD CHAINING*  
BERBASIS ANDROID**

**TUGAS AKHIR**

*Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Pendidikan  
Strata I pada Jurusan Teknik Elektronika Program Studi Pendidikan Teknik  
Informatika Fakultas Teknik Universitas Negeri Padang*



**KHILDA RAHMI ZAKI**

**NIM 2011/1102660**

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA  
JURUSAN TEKNIK ELEKTRONIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI PADANG  
2016**

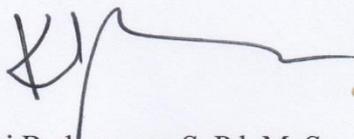
## TUGAS AKHIR

**Judul** : **Perancangan Sistem Pakar Diagnosis Asma dan Terapinya dengan Metode *Forward Chaining* Berbasis Android**  
Nama : Khilda Rahmi Zaki  
NIM/TM : 1102660/2011  
Program Studi : Pendidikan Teknik Informatika  
Jurusan : Teknik Elektronika  
Fakultas : Teknik

Padang, April 2016

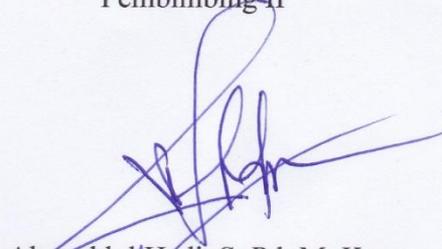
Disetujui oleh:

Pembimbing I



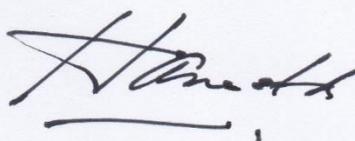
Khairi Budayawan, S. Pd, M. Sc  
NIP. 19760810 200312 1 002

Pembimbing-II



AhmaddulHadi, S. Pd, M. Kom  
NIP. 19761209 200501 1 003

Mengetahui,  
Ketua Jurusan Teknik Elektronika  
Fakultas Teknik UNP



Drs. Hanesman, MM  
NIP. 19610111 198503 1 002

## PENGESAHAN TIM PENGUJI

Nama : Khilda Rahmi Zaki  
NIM : 1102660/2011

Dinyatakan lulus setelah mempertahankan Tugas Akhir di depan Tim Penguji  
Program Studi Pendidikan Teknik Informatika  
Jurusan Teknik Elektronika  
Fakultas Teknik  
Universitas Negeri Padang  
dengan judul

**Perancangan Sistem Pakar Diagnosis Asma dan Terapinya  
dengan Metode *Forward Chaining* Berbasis Android**

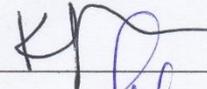
Padang, April 2016

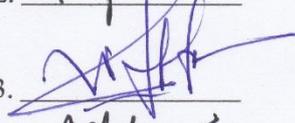
Tim Penguji

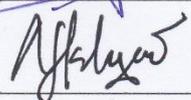
Tanda Tangan

1. Ketua : Drs. Legiman Slamet, MT.
2. Sekretaris : Khairi Budayawan, S. Pd, M. Sc
3. Anggota : Ahmaddul Hadi, S.Pd, M.Kom
4. Anggota : Yeka Hendriyani, S.Kom, M.Kom

1. 

2. 

3. 

4. 

## PERNYATAAN

Dengan ini saya menyatakan bahwa :

1. Karya tulis saya, tugas akhir dengan judul ” Perancangan Sistem Pakar Diagnosis Asma dan Terapinya dengan Metode *Forward Chaining* Berbasis Android”, adalah asli karya saya sendiri;
2. Karya tulis ini murni gagasan, rumusan, dan karya saya sendiri, tanpa bantuan pihak lain, kecuali dari pembimbing;
3. Di dalam karya tulis ini, tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan di dalam naskah dengan menyebutkan pengarang dan dicantumkan pada kepustakaan;
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila terdapat penyimpangan di dalam pernyataan ini, saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai norma dan ketentuan hukum yang berlaku.

Padang, April 2016

Yang menyatakan,



Khilda Rahmi Zaki  
NIM. 2011/1102660

## ABSTRAK

**Khilda Rahmi Zaki. 2016.** “Perancangan Sistem Pakar Diagnosis Asma dan Terapinya dengan Metode *Forward Chaining* Berbasis Android” *Tugas Akhir*. Padang: Program Studi Pendidikan Teknik Informatika, Jurusan Teknik Elektronika, Fakultas Teknik Universitas Negeri Padang.

Asma adalah penyakit kronis yang umum dan berpotensi serius yang memberikan beban besar pada pasien, keluarga dan masyarakat. Seorang pasien yang sudah didiagnosis menderita asma, langkah selanjutnya yang perlu diambil adalah mengetahui tingkat keparahan asma yang dideritanya. Hal ini berguna sebagai panduan terapi untuk pengontrolan. Karena asma tidak bisa disembuhkan, maka yang perlu dilakukan adalah mengontrol serangan asma agar asma tidak mengganggu aktifitas sehari-hari dan pasien dapat hidup seperti manusia normal lainnya. Oleh sebab itu, dibutuhkan suatu alat bantu yang dapat menentukan tingkat keparahan asma berupa sistem pakar, sehingga mempermudah mengetahui tingkat keparahan asma.

Sistem pakar untuk mendiagnosis tingkat keparahan dirancang menggunakan model perancangan *Unified Modelling Language* (UML). Sistem rancangan yang diusulkan digambarkan dalam tiga tahapan diagram yaitu *use case diagram*, *activity diagram* dan *sequence diagram*. Metode yang digunakan adalah metode inferensi runut maju (*Forward Chaining*). Basis pengetahuan disusun sedemikian rupa ke dalam suatu *database* dengan beberapa tabel dan *rule* If-Then sehingga mempermudah kinerja sistem dalam penarikan kesimpulan.

Sistem pakar ini saat digunakan akan menampilkan pertanyaan-pertanyaan sebagai gejala yang dapat dipilih oleh *user* dan setiap pilihan gejala akan membawa *user* kepada pilihan gejala berikutnya sampai mendapatkan hasil akhir. Pada hasil akhir sistem akan memberikan sebuah kesimpulan tingkat keparahan asma *user*. Seorang *user* yang telah mengetahui tingkat keparahan asmanya akan mendapatkan informasi tentang terapi yang sesuai.

## KATA PENGANTAR

Segala Puji hanya milik Allah SWT, hanya untaian kalimat syukur yang patut menghiasi hati dan bibir untuk mengingat nikmat Allah yang tiada terhingga yang diberikanNya kepada kita. Alhamdulillah dengan izin-Nya penulis dapat menyelesaikan Tugas Akhir ini. Tugas Akhir ini berjudul “Perancangan Sistem Pakar Diagnosis Asma dan Terapinya dengan Metode *Forward Chaining* Berbasis Android.” Penulisan laporan Tugas Akhir ini berguna untuk memenuhi salah satu syarat memperoleh gelar Sarjana Pendidikan Strata Satu pada Program Studi Pendidikan Teknik Informatika di Fakultas Teknik Universitas Negeri Padang. Serta tidak lupa shalawat dan salam kepada sosok teladan sepanjang masa Nabi Muhammad SAW.

Dalam menyelesaikan Tugas Akhir ini, penulis dibantu dan dibimbing oleh berbagai pihak, untuk itu penulis mengucapkan terima kasih kepada:

1. Bapak Drs. Syahril, ST, M.Sc.E, Ph.D., selaku Dekan Fakultas Teknik Universitas Negeri Padang.
2. Bapak Drs. Hanesman, MM., selaku Ketua Jurusan Teknik Elektronika Fakultas Teknik Universitas Negeri Padang.
3. Bapak Khairi Budayawan, S.Pd, M.Sc., selaku dosen pembimbing I yang telah membimbing dalam perencanaan, pelaksanaan dan pelaporan Tugas Akhir ini.
4. Bapak Ahmaddul Hadi, S.Pd, M.Kom., selaku pembimbing II yang telah membimbing dalam perencanaan, pelaksanaan dan pelaporan Tugas Akhir ini.

5. Bapak Drs. Legiman Slamet, MT., Ibu Yeka Hendriyani., S.Kom, M.Kom., selaku dosen penguji yang telah banyak meluangkan waktu untuk memberikan arahan dan bimbingan dalam menyelesaikan Tugas Akhir ini.
6. Staf pengajar, Teknisi dan Pegawai Jurusan Teknik Elektronika Universitas Negeri Padang.
7. Orangtua dan keluarga yang telah memberikan semangat, motivasi dan doa selama menyelesaikan Tugas Akhir ini.
8. Teman-teman Pendidikan Teknik Informatika 2011 yang telah memberikan motivasi selama menyelesaikan Tugas Akhir ini.
9. Semua pihak yang telah membantu dalam menyelesaikan Tugas Akhir ini.

Semoga bantuan dan bimbingan yang telah diberikan menjadi amal shaleh bagi Bapak dan Ibu serta mendapat balasan yang berlipat ganda dari Allah SWT.

Penulis menyadari bahwa dalam laporan Tugas Akhir ini masih terdapat kekurangan dan kelemahan. Untuk itu penulis mengharapkan saran untuk menyempurnakan laporan ini. Semoga Tugas Akhir ini bermanfaat bagi pembaca.

Padang, April 2016

Penulis

## DAFTAR ISI

	Halaman
<b>ABSTRAK</b> .....	i
<b>KATA PENGANTAR</b> .....	ii
<b>DAFTAR ISI</b> .....	iv
<b>DAFTAR TABEL</b> .....	vi
<b>DAFTAR GAMBAR</b> .....	vii
<b>BAB I PENDAHULUAN</b>	
A. Latar Belakang .....	1
B. Identifikasi Masalah.....	4
C. Batasan Masalah .....	5
D. Rumusan Masalah .....	5
E. Tujuan .....	6
F. Manfaat .....	6
<b>BAB II LANDASAN TEORI</b>	
A. Kecerdasan Buatan.....	7
B. Sistem Pakar.....	11
1. Pengertian Sistem Pakar .....	11
2. Konsep Dasar Sistem Pakar .....	12
3. Tujuan Sistem Pakar .....	15
4. Ciri-ciri Sistem Pakar.....	15
5. Keuntungan Sistem Pakar .....	15
6. Kelemahan Sistem Pakar .....	16
7. Struktur Sistem Pakar .....	16
C. Metode <i>Forward Chaining</i> .....	20
D. Asma .....	23
1. Pengertian Asma .....	23
2. Faktor Pencetus Asma.....	24
3. Klasifikasi Asma.....	25
E. Android .....	30
1. Definisi Android .....	30
2. Sejarah Perkembangan Android.....	31
3. Fitur Andrid .....	34
4. Arsitektur Android .....	34
F. Desain Antar Muka .....	36
G. Pemodelan Sistem dengan <i>Unified Modelling Language</i> (UML) .....	39
1. <i>Class Diagram</i> .....	40
2. <i>Package Diagram</i> .....	41
3. <i>Use Case Diagram</i> .....	41
4. <i>Sequence Diagram</i> .....	44
5. <i>Collaboration Diagram</i> .....	44
6. <i>Statechart Diagram</i> .....	44

7. <i>Activity Diagram</i> .....	45
8. <i>Component Diagram</i> .....	46
9. <i>Deployment Diagram</i> .....	47

### **BAB III ANALISIS DAN PERANCANGAN SISTEM**

A. Analisis Sistem.....	48
1. Analisis Fisikal.....	48
a. Analisis Permasalahan .....	48
b. Analisis Sistem yang Ada .....	49
c. Analisis User.....	50
2. Kebutuhan Fungsional dan Non Fungsional .....	50
3. Analisis Kebutuhan Perangkat .....	51
a. Analisis Kebutuhan <i>Software</i> .....	51
b. Analisis Kebutuhan <i>Hardware</i> .....	52
4. Analisis Data Aplikasi .....	52
a. Data Tingkat Keparahan Asma.....	52
b. Data Gejala/ <i>Evidence</i> .....	53
B. Perancangan Sistem .....	54
1. Perancangan Basis Pengetahuan ( <i>Knowledge Base</i> ).....	54
a. Perancangan Rule.....	55
b. Perancangan Mesin Inferensi <i>Forward Chaining</i> .....	57
c. <i>Desicion Tree</i> .....	59
2. Model Perancangan.....	61
a. <i>Use Case Diagram</i> .....	62
b. <i>Activity Diagram</i> .....	63
c. <i>Sequence Diagram</i> .....	65
C. Perancangan Disain Sistem.....	66
1. Menu Beranda.....	67
2. Menu Diagnosis .....	68
3. Menu Terapi.....	69
4. Menu Asma.....	69
5. Menu Tentang .....	70
6. Menu Panduan .....	71

### **BAB IV HASIL DAN PEMBAHASAN**

A. Hasil Antar Muka Sistem.....	73
B. Pembahasan Sistem.....	73
C. Pengujian Sistem.....	85
D. Pengujian Tingkat Keparahan Asma.....	90

### **BAB V PENUTUP**

A. Kesimpulan .....	95
B. Saran .....	95

<b>DAFTAR PUSTAKA</b> .....	97
-----------------------------	----

<b>LAMPIRAN</b> .....	98
-----------------------	----

## DAFTAR TABEL

<b>Tabel</b>	<b>Halaman</b>
1. Data Tingkat Keparahan Asma .....	52
2. Gejala/ <i>Evidence</i> .....	53
3. Basis Pengetahuan .....	54
4. Perancangan <i>Rule</i> Tingkat Keparahan Asma .....	55
5. <i>Rule</i> Sistem Pakar Asma .....	57
6. Pengujian <i>Splash Screen</i> .....	86
7. Pengujian Halaman Menu Beranda.....	86
8. Pengujian Halaman Menu Diagnosis .....	87
9. Pengujian Halaman Menu Terapi .....	88
10. Pengujian Halaman Asma .....	88
11. Pengujian Halaman Menu Tentang .....	89
12. Pengujian Halaman Menu Panduan .....	89

## DAFTAR GAMBAR

Gambar	Halaman
1. Penerapan Konsep Kecerdasan Buatan di Komputer .....	9
2. Arsitektur Sistem Pakar .....	17
3. Diagram Pelacakan ke Depan ( <i>Forward Chaining</i> ) .....	19
4. Diagram Pelacakan ke Belakang ( <i>Backward Chaining</i> ) .....	19
5. Cara Kerja Metode Inferensi <i>Forward Chaining</i> .....	21
6. Proses <i>Forward Chaining</i> .....	22
7. Diagram <i>Flowchart</i> Sistem .....	58
8. <i>Desicion Tree</i> Tingkat Keparahan Asma .....	60
9. <i>Use Case</i> Diagram Sistem Pakar.....	63
10. <i>Activity Diagram</i> Sistem .....	64
11. <i>Sequence Diagram</i> Beranda.....	65
12. <i>Sequence Diagram</i> Menu Diagnosis.....	66
13. Disain Tampilan Menu Beranda .....	67
14. Disain Tampilan Menu Diagnosis.....	68
15. Disain Tampilan Menu Terapi .....	69
16. Disain Tampilan Menu Asma .....	70
17. Disain Tampilan Menu Tentang .....	71
18. Disain Tampilan Menu Panduan.....	72
19. Tampilan <i>Splash Screen</i> .....	74
20. Tampilan Halaman Menu Beranda .....	76
21. Tampilan Halaman Pertama Diagnosis.....	78
22. Tampilan Halaman Mulai Diagnosis .....	79
23. Tampilan Halaman Menu Terapi .....	81
24. Tampilan Halaman Terapi Asma .....	82
25. Tampilan Halaman Menu Asma .....	83
26. Tampilan Halaman Menu Tentang .....	84
27. Tampilan Halaman Menu Panduan.....	85
28. Tampilan Hasil Intermiten Ringan.....	91
29. Tampilan Hasil Persisten Ringan .....	92
30. Tampilan Hasil Persisten Sedang.....	93
31. Tampilan Hasil Persisten Berat.....	94

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Perkembangan teknologi saat ini mengalami kemajuan yang pesat, sehingga banyak memunculkan berbagai inovasi. Perkembangan teknologi semakin mempermudah kerja manusia dalam berbagai bidang, seperti ekonomi, sosial, pendidikan, kesehatan dan ilmu pengetahuan. Seiring dengan perkembangan teknologi yang pesat, manusia dituntut untuk dapat memanfaatkan perkembangan teknologi tersebut.

Bidang kesehatan merupakan salah satu bidang kehidupan yang memanfaatkan perkembangan teknologi. Perkembangan teknologi tersebut diantaranya adalah kecerdasan buatan (*artificial intelligence*). Kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

Sistem pakar merupakan salah satu cabang kecerdasan buatan yang mempelajari bagaimana mengadopsi cara seorang pakar berpikir dan bernalar dalam menyelesaikan suatu permasalahan, dan membuat suatu keputusan, maupun mengambil kesimpulan dari sejumlah fakta yang ada. Tujuan utama sistem pakar adalah untuk memasyarakatkan atau memindahkan secara efektif pengetahuan dan pengalaman para pakar kepada mereka yang bukan pakar. Pemecahan masalah diberikan pada pengguna melalui dialog dengan sistem. Sampai saat ini sudah ada beberapa hasil perkembangan sistem pakar dalam berbagai bidang sesuai dengan kepakaran seseorang. Dengan bantuan sistem

pakar pengguna akan lebih menghemat waktu dan biaya. Penerapan sistem pakar salah satunya terdapat dalam ilmu kedokteran. Sistem pakar akan menggantikan peran dokter dalam mendiagnosis penyakit, pemberian saran kesehatan, ataupun solusi atas penyakit yang diderita pengguna.

Beberapa gangguan kesehatan disebabkan oleh polusi udara dan keadaan sekitar yang membuat tubuh manusia rentan terhadap penyakit, terutama penyakit pernafasan, seperti asma. Menurut MIMS Indonesia Petunjuk Konsultasi Edisi 2011/2012, yang merupakan sebuah buku pegangan untuk para profesional tenaga kesehatan yang memberikan informasi mengenai obat dan resep obat secara berkala, asma adalah suatu penyakit pernafasan yang ditandai dengan inflamasi (radang) saluran pernafasan (bronkus) yang menyebabkan aliran udara ke dan dari paru-paru menjadi kurang lancar, sehingga menimbulkan gejala-gejala khas yaitu mengi, batuk, konstruksi dada, dan sesak nafas.

*World Health Organization* (WHO) mencatat pada tahun 2013 terdapat 235 juta pasien asma di seluruh dunia dan sebagian besar kematian terkait asma terjadi di negara-negara berpenghasilan rendah dan menengah ke bawah. Pada tahun 2011, kematian akibat asma di Indonesia mencapai 14.624 jiwa. Angka ini berarti asma menyebabkan sekitar 1% total kematian di Indonesia. Sekitar 1.1% populasi Indonesia menderita asma. Tingginya angka kematian ini banyak disebabkan oleh kontrol asma yang buruk. Berdasarkan hasil penelitian oleh ahli asma di Asia Pasifik dalam studi "*Asthma Insight & Reality in Asia Pasific (AIRIAP 2)*" pada tahun 2007, hanya 2% dari 4805 orang sampel penyandang

asma di Asia Pasifik yang mempelajari bagaimana mengontrol penyakit asma mereka.

Asma adalah penyakit yang tidak bisa disembuhkan dan dapat menyerang siapa pun. *Global Initiative for Asthma (GINA)* : 2014 mengemukakan, “asma adalah penyakit kronis yang umum dan berpotensi serius yang memberikan beban besar pada pasien, keluarga dan masyarakat”. Seorang pasien yang sudah didiagnosis menderita asma, langkah selanjutnya yang perlu diambil adalah mengetahui tingkat keparahan asma yang dideritanya. Hal ini berguna sebagai panduan terapi untuk pengontrolan. Karena asma tidak bisa disembuhkan, maka yang perlu dilakukan adalah mengontrol serangan asma agar asma tidak mengganggu aktifitas sehari-hari dan pasien dapat hidup seperti manusia normal lainnya.

Asma memiliki empat tingkatan keparahan, sesuai dengan berat ringannya penyakit, yaitu asma intermiten ringan, asma persisten ringan, asma persisten sedang, dan asma persisten berat. Berdasarkan tingkat keparahan ini dapat diketahui langkah pengontrolan yang perlu dilakukan pasien. Mengenai obat asma, ada yang bersifat pelega (*reliever*), ada pula yang bersifat pengontrol (*controller*). Pelega akan membuat saluran pernafasan yang menyempit membuka kembali, sementara pengontrol akan membuat saluran pernafasan tidak mudah menyempit bila ada faktor pencetus serangan asma. Dibutuhkan kedisiplinan dan kesadaran dari pasien, keluarga, dan lingkungan yang berada disekitar pasien untuk melakukan pengontrolan asma. Dengan pengontrolan

yang tepat serangan asma dapat ditekan dan tingkat keparahannya pun menjadi lebih ringan.

Kecanggihan teknologi yang merambah ke bidang komunikasi, salah satunya menghasilkan sebuah sistem operasi berbasis Linux untuk telepon seluler seperti *smartphone* dan komputer tablet, yang dikenal dengan android. Android bisa menghasilkan berbagai aplikasi yang dibutuhkan penggunanya, digunakan pada telpon seluler yang dapat diakses dengan lebih mudah, baik aplikasi *online* maupun *offline*. Kelebihan android yang terdapat pada *smartphone* adalah *smartphone* bisa dibawa kemana saja dengan ukuran yang lebih kecil jika dibandingkan dengan komputer, sementara fungsi *smartphone* sama dengan komputer.

Berdasarkan pemaparan di atas, penulis tertarik untuk membuat sebuah aplikasi sistem pakar di bidang kesehatan, dengan fokus pada tingkat keparahan asma dan terapinya menggunakan aplikasi android. Penulis mencoba merealisasikan hal tersebut melalui tugas akhir yang berjudul “*Perancangan Sistem Pakar Diagnosis Asma dan Terapinya dengan Metode Forward Chaining Berbasis Android*”.

## **B. Identifikasi Masalah**

Berdasarkan latar belakang di atas, dapat ditarik poin-poin identifikasi masalah adalah:

1. Perlunya pemahaman bagi masyarakat mengenaitingkatan keparahan pada penyakit asma sebagai langkah pengontrolan asma.
2. Perlunya informasi terapi penanganan asma bagi masyarakat.

3. Asma merupakan penyakit yang memerlukan penanganan yang cepat.
4. Perlunya pemahaman bahwa asma tidak dapat disembuhkan, tapi dapat dikontrol.

### **C. Batasan Masalah**

Agar tugas akhir ini lebih terarah dan sesuai dengan tujuan yang diharapkan maka ditetapkan batasan masalah yaitu:

1. Sistem pakar yang dirancang merupakan aplikasi untuk menentukan tingkat keparahan penyakit asma berdasarkan buku MIMS Indonesia Petunjuk Konsultasi Edisi 11 2011/2012.
2. Metode yang digunakan dalam perancangan sistem pakar ini adalah metode *forward chaining*.
3. Aplikasi dirancang dengan menggunakan bahasa pemrograman Java, Android Studio sebagai IDE, Android SDK sebagai pengembangan Android, serta *emulator* sebagai *emulator* untuk menjalankan sistem android.

### **D. Rumusan Masalah**

Berdasarkan batasan masalah yang telah dipaparkan di atas dapat ditarik rumusan masalah sebagai berikut : “Bagaimana merancang suatu sistem pakar diagnosis asma dan terapinya dengan metode *forward chaining* berbasis android?”

### **E. Tujuan**

Tujuan dari perancangan sistem pakar ini adalah untuk menghasilkan sebuah aplikasi sistem pakar diagnosis asma dan terapinya dengan metode *forward chaining* berbasis android.

### **F. Manfaat**

Manfaat yang didapat dari tugas akhir ini adalah:

1. Sebagai alternatif solusi pemecahan masalah kepada pengguna untuk mengetahui tingkat keparahan asma dan terapi yang sesuai.
2. Memberikan informasi kepada masyarakat mengenai tingkat keparahan asma dan terapinya.

## **BAB II**

### **LANDASAN TEORI**

#### **A. Kecerdasan Buatan**

Menurut Anita Desiani dan Muhammad Arhami (2006:1) “*Artificial Intelligence* (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang konsern dengan pengoutomasi tingkah laku cerdas”.

Kusrini (2006:3) mengemukakan “Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas”.

Anita Desiani dan Muhammad Arhami (2006:6)

Ada dua hal yang sangat mendasar mengenai penelitian-penelitian AI, yaitu *knowledge representation* (representasi pegetahuan) dan *search* (pelacakan). Para peneliti AI terus mengembangkan berbagai jenis teknik baru dalam menangani sejumlah permasalahan yang tergolong ke dalam AI seperti vision dan percakapan, pemrogsesan bahasa alami dan permasalahan khusus seperti diagnosis medis.

Ruang lingkup utama kecerdasan buatan antara lain:

1. Sistem pakar (*expert system*): Komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan masalah dengan meniru keahlian yang dimiliki para pakar.

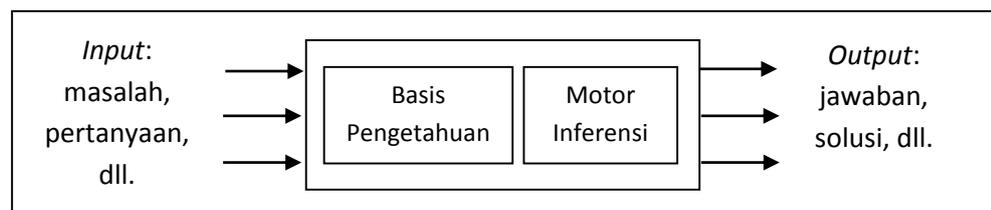
2. Pengolahan bahasa alami (*natural language processing*): dengan pengolahan bahasa alami ini diharapkan *user* mampu berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan ucapan (*speech recognition*): Melalui pengenalan ucapan diharapkan manusia mampu berkomunikasi dengan komputer dengan menggunakan suara.
4. Robotika dan Sistem sensor.
5. *Computer vision*: menginterpretasikan gambar atau objek-objek tampak melalui komputer.
6. *Intelligent Computer-aided Instruction*. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. *Game playing*: suatu metode kecerdasan buatan yang berguna untuk meniru cara berfikir seorang manusia dalam bermain *game*. Contohnya adalah program *Deep Blue* yang mampu berfikir setara dengan seorang *grandmaster* catur.

Kecerdasan buatan mendayagunakan komputer sehingga komputer dapat berperilaku cerdas seperti layaknya manusia. Kecerdasan buatan juga dapat didefinisikan sebagai salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia.

Agar mesin bisa cerdas (bertindak seperti & sebaik manusia) maka harus diberi bekal pengetahuan dan kemampuan untuk menalar. Dua bagian utama yg dibutuhkan untuk aplikasi kecerdasan buatan yaitu:

1. Basis pengetahuan (*knowledge base*): berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor inferensi (*inference engine*): kemampuan menarik kesimpulan berdasarkan pengalaman.

Gambar 1 berikut memperlihatkan penerapan konsep kecerdasan buatan di komputer.



**Gambar 1.** Penerapan Konsep Kecerdasan Buatan di Komputer

Gambar 1 menjelaskan basis pengetahuan berisi *input* berupa masalah atau pertanyaan untuk membuat *output* yang merupakan kesimpulan (*conclusions*) berupa jawaban untuk pertanyaan atau solusi untuk sebuah masalah.

Jika dibanding kecerdasan alami, kecerdasan buatan memiliki keuntungan komersial, antara lain:

1. Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer dan program tidak mengubahnya.

2. Kecerdasan buatan lebih mudah diduplikasi dan disebar. Mentransfer pengetahuan manusia dari satu orang ke orang lain butuh proses dan waktu lama. Disamping itu suatu keahlian tidak akan pernah bisa diduplikasi secara lengkap. Sedangkan jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat ditransfer atau disalin dengan mudah dan cepat dari satu komputer ke komputer lain.
3. Kecerdasan buatan lebih murah dibanding dengan kecerdasan alami. Menyediakan layanan komputer akan lebih mudah dan lebih murah dibanding dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
4. Kecerdasan buatan bersifat konsisten. Hal ini disebabkan karena kecerdasan busatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami senantiasa berubah-ubah.
5. Kecerdasan buatan dapat didokumentasikan. Keputusan yang dibuat komputer dapat didokumentasikan dengan mudah dengan melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
6. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami.
7. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami.

## **B. Sistem Pakar**

Sistem pakar adalah salah satu cabang dari AI yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya. Ketika sistem pakar dikembangkan pertama kali pada tahun 70-an sistem pakar hanya memiliki *knowledge* yang eksklusif. Namun demikian sekarang ini istilah sistem pakar sudah digunakan untuk berbagai macam sistem yang menggunakan teknologi sistem pakar itu. Teknologi sistem pakar ini meliputi bahasa sistem pakar, program dan perangkat keras yang dirancang untuk membantu pengembangan dan pembuatan sistem pakar.

*Knowledge* dalam sistem pakar mungkin saja seorang ahli, atau *knowledge* yang umumnya terdapat dalam buku, majalah dan orang yang mempunyai pengetahuan tentang suatu bidang.

### **1. Pengertian Sistem Pakar**

Professor Edward Feigenbaum dari Universitas Stanford yang merupakan seorang pelopor awal dari teknologi sistem pakar, yang mendefinisikan sistem pakar sebagai “suatu program komputer cerdas yang menggunakan *knowledge* (pengetahuan) dan prosedur inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang yang ahli untuk menyelesaikannya”. Feigenbaum dalam Muhammad Arhami (2005:2) “suatu sistem pakar adalah suatu sistem komputer yang

menyamai (*emulates*) kemampuan pengambilan keputusan dari seorang pakar”.

Menurut Martin dan Oxman dalam Kusrini (2006:11) “Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut”.

Menurut Turban dalam Kusrini (2006:1) “Sistem pakar adalah program komputer yang menirukan penalaran seorang pakar dengan keahlian pada suatu wilayah pengetahuan tertentu.”. Permasalahan yang ditangani oleh seorang pakar bukan hanya permasalahan yang mengandalkan algoritma, namun terkadang juga permasalahan yang sulit dipahami. Permasalahan tersebut dapat diatasi oleh seorang pakar dengan pengetahuan dan pengalamannya. Oleh karena itu, sistem pakar dibangun bukan berdasarkan algoritma tertentu tetapi berdasarkan basis pengetahuan dan aturan.

## **2. Konsep Dasar Sistem Pakar**

Turban dalam Muhammad Arhami (2005:11) “menyatakan bahwa konsep dasar dari suatu sistem pakar mengandung beberapa unsur/elemen, yaitu keahlian, ahli, pengalihan keahlian, inferensi, aturan, dan kemampuan menjelaskan”.

a. Keahlian(*expertise*)

Keahlian (*expertise*) adalah pengetahuan yang mendalam tentang suatu masalah tertentu, dimana keahlian bisa diperoleh dari pelatihan/ pendidikan, membaca dan pengalaman dunia nyata. Ada dua macam pengetahuan yaitu pengetahuan dari sumber yang ahli dan pengetahuan dari sumber yang tidak ahli. Pengetahuan dari sumber yang ahli dapat digunakan untuk mengambil keputusan dengan cepat dan tepat.

b. Ahli (*experts*)

Merupakan seseorang yang memiliki pengetahuan, penilaian, pengalaman, metode tertentu, serta mampu menerapkan keahlian dalam memberikan advise untuk pemecahan persoalan, serta mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan, mengenali & merumuskan permasalahan, menyusun kembali pengetahuan jika dipandang perlu, memecah aturan-aturan jika dibutuhkan, dan mampu menentukan relevan tidaknya keahlian mereka.

c. Pemindahan Keahlian

Merupakan pengalihan keahlian dari para ahli ke komputer untuk kemudian dialihkan lagi ke orang lain yang bukan ahli, dimana pengetahuan yang disimpan di komputer ini disebut dengan nama basis pengetahuan. Ada 2 tipe pengetahuan, yaitu: fakta dan prosedur (biasanya berupa aturan).

d. Inferensi

Kemampuan untuk melakukan penalaran dengan menggunakan pengetahuan yang ada untuk menghasilkan suatu kesimpulan atau hasil akhir. dengan menggunakan *motor interafe* yang merupakan permodelan proses berfikir dan bernalar layaknya manusia.

e. Aturan

Sebagian besar sistem pakar dibuat dalam bentuk *rule-based system*, dimana pengetahuan disimpan dalam bentuk aturan-aturan.

f. Kemampuan Menjelaskan

Karakteristik dari sistem pakar yang memiliki kemampuan menjelaskan atau memberi saran mengapa tindakan tertentu dianjurkan atau tidak dianjurkan.

Menurut Turban dalam Muhammad Arhami (2005:12--13), terdapat tiga orang yang terlibat dalam lingkungan sistem pakar, yaitu:

a. Pakar

Pakar adalah orang yang memiliki pengetahuan khusus, pendapat, pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.

b. *Knowledge Engineer* (Perekayasa Sistem)

*Knowledge Engineer* adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan *counter example* dan menerangkan kesulitan-kesulitan konseptual.

c. Pemakai

Sistem pakar memiliki beberapa pemakai, yaitu: pemakai bukan pakar, pelajar, pembangun sistem pakar yang ingin meningkatkan dan menambah basis pengetahuan, dan pakar.

### 3. Tujuan Sistem Pakar

Tujuan dari sistem pakar adalah mentransfer kepakaran yang dimiliki seorang pakar kedalam komputer, dan kemudian kepada orang lain (*nonexpert*). Aktivitas yang dilakukan untuk memindahkan kepakaran adalah:

- a. Akuisisi Pengetahuan (*KnowledgeAcquisition*).
- b. Representasi Pengetahuan (*KnowledgeRepresentation*).
- c. Inferensi Pengetahuan (*KnowledgeInferencing*).
- d. Pemandahan Pengetahuan ke Pengguna (*KnowledgeTransferring*).

### 4. Ciri Ciri Sistem Pakar

Menurut Kusri (2006 : 14 --15) ciri-ciri sistem pakar antara lain:

- a. Terbatas pada bidang yang spesifik.
- b. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
- c. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
- d. Berdasarkan pada kaidah atau *rule* tertentu.
- e. Dirancang untuk dapat dikembangkan secara bertahap.
- f. *Output* bersifat nasihat atau anjuran.
- g. *Output* tergantung dari dialog dengan *user*.
- h. *Knowledge base* dan *inference engine* terpisah.

### 5. Keuntungan Sistem Pakar

Menurut Muhammad Arhami (2004:10), ada beberapa keuntungan

bila menggunakan sistem pakar, di antara lain adalah :

- a. Menjadikan pengetahuan dan nasihat lebih mudah didapat.
- b. Meningkatkan *output* dan produktivitas.
- c. Menyimpan kemampuan dan keahlian pakar.
- d. Meningkatkan penyelesaian masalah – menerusi paduan pakar, penerangan, sistem pakar khas .
- e. Meningkatkan reliabilitas.

- f. Memberikan respons (jawaban) yang cepat.
- g. Merupakan panduan yang *intelligence* (cerdas).
- h. Dapat bekerja dengan informasi yang kurang lengkap dan mengandung ketidakpastian.
- i. *Intelligence database* (basis data cerdas), bahwa sistem pakar dapat digunakan untuk mengakses basis data dengan cara cerdas.

## 6. Kelemahan Sistem Pakar

Selain keuntungan-keuntungan di atas, sistem pakar seperti halnya sistem lainnya juga memiliki kelemahan, di antaranya adalah :

- a. Masalah didalam mendapatkan pengetahuan di mana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, dan walaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda.
- b. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharaannya.
- c. Boleh jadi sistem tidak dapat membuat keputusan.  
Sistem pakar tidaklah 100% menguntungkan, walaupun seorang tidak tetap tidak sempurna atau tidak selalu benar. Oleh karena itu, perlu diuji ulang secara teliti sebelum digunakan. Dalam hal ini peran manusia tetap merupakan faktor dominan. (Arhami, 2005 : 10--11)

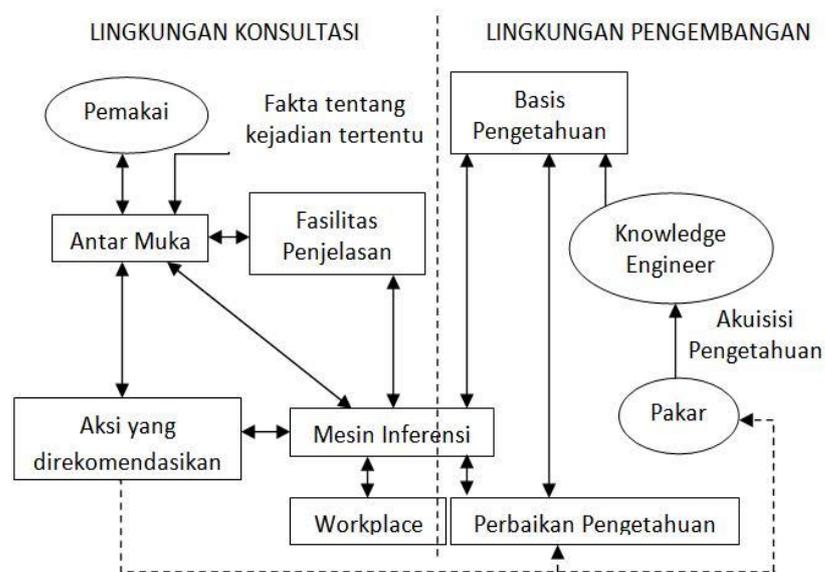
## 7. Struktur Sistem Pakar

Sistem pakar memiliki beberapa komponen utama, yaitu antarmuka pengguna (*user interface*), basis data sistem pakar (*expert system database*), fasilitas akuisisi pengetahuan (*knowledge acquisition facility*), dan mekanisme inferensi (*inference mechanism*). Selain itu ada satu komponen yang hanya ada pada beberapa sistem pakar, yaitu fasilitas penjelasan (*explanation facility*). (Martin dan Oxman dalam Kusri (2006:17))

“Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*).” Turbandalam Muhammad Arhami (2005 : 13).

Lingkungan pengembangan sistem pakar digunakan untuk memasukkan

pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar. Komponen-komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam Gambar 2 berikut ini:



**Gambar 2.** Arsitektur Sistem Pakar

Komponen-komponen yang terdapat dalam sistem pakar adalah seperti yang terdapat pada gambar 2, yaitu:

a. Antarmuka Pengguna (*User Interface*)

Merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Pada bagian ini akan terjadi dialog antara program dengan pemakai. Program akan mengajukan pertanyaan berbentuk “ya / tidak” (*yes or no question*) atau berbentuk menu pilihan. Melalui jawaban yang diberikan oleh pemakai, sistem pakar akan

mengambil kesimpulan yang berupa informasi ataupun anjuran sesuai dengan sifat dari sistem pakar.

b. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

c. Akuisi Pengetahuan (*Knowledge Acquisition*)

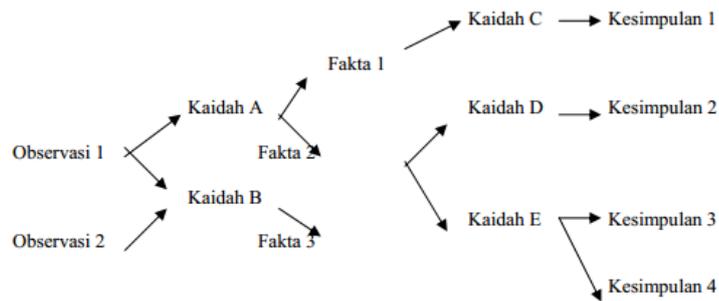
Akuisi Pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer

d. Mesin Inferensi

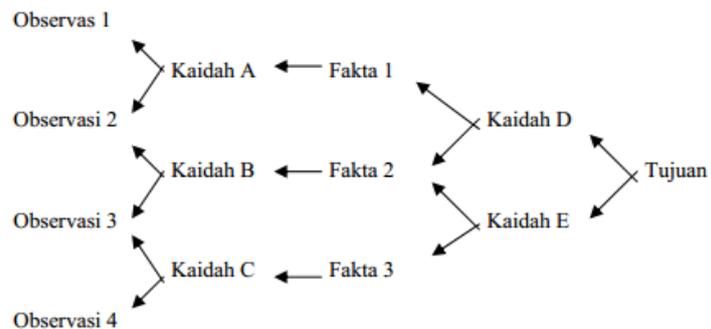
Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berfikir dan pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini akan menganalisa suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik.

Mesin inferensi memulai pelacakannya dengan mencocokkan kaidah dalam basis pengetahuan dengan fakta yang ada dalam basis data. Ada dua teknik inferensi yang ada yaitu pelacakan ke belakang (*backward chaining*) yang memulai penalaran dari kesimpulan hipotesa menuju fakta yang mengandung hipotesa tersebut. Dan yang kedua

yakni pelacakan ke depan (*forward chaining*) yang merupakan kebalikan dari pelacakan ke belakang yaitu memulai dari sekumpulan data menuju kesimpulan.



**Gambar 3.** Diagram Pelacakan ke Depan (*Forward Chaining*)



**Gambar 4.** Diagram Pelacakan ke Belakang (*Backward Chaining*)

Kedua metode inferensi tersebut dipengaruhi oleh tiga macam teknik penelusuran yaitu: *Depth-first search* melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan. *Breadth-first search* bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat

selanjutnya. *Best-first search* bekerja berdasarkan kombinasi kedua metode sebelumnya.

e. *Workplace*

Merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil antara dan kesimpulan yang dicapai.

f. Fasilitas Penjelasan

Merupakan komponen tambahan yang meningkatkan kemampuan sistem pakar.

g. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

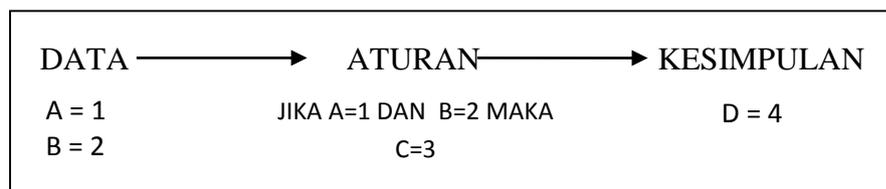
### C. Metode *Forward Chaining*

Rantai (*chain*) merupakan suatu perkalian inferensi yang menghubungkan suatu permasalahan dengan solusinya. Suatu rantai yang dicari atau dilewati/dilintasi dari suatu permasalahan untuk memperoleh solusinya disebut dengan *forward chaining*.

Konsep dari *forward chaining* berangkat dari premis menuju kepada kesimpulan akhir (*data driven*) yaitu pencarian dikendalikan oleh data yang diberikan, artinya suatu proses yang memulai pencarian dari premis atau data

menuju konklusi. Dalam penganalisaan masalah, komputer mencari fakta atau nilai yang sesuai dengan syarat pada posisi IF dari rule IF-THEN.

Menurut Wilson dalam Kusrini (2008 : 8), “*Forward chaining* berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil”. Gambar 5 menggambarkan cara kerja metode inferensi *forward chaining*.



**Gambar 5.** Cara Kerja Metode Inferensi *Forward Chaining*

“Metode inferensi *forward Chaining* cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*)” Girattano dan Riley, (dalam Kusrini 2008). Berikut adalah contoh inferensi dengan menggunakan metode *forward chaining*.

JIKA            penderita terkena penyakit epilepsi idiopatik dengan CF antara 0,4/0,6

MAKA            berikan obat carbamazepine

Secara garis besar proses penalaran dengan *Forward chaining* adalah sebagai berikut :

1. Strategi inferensi dimulai dengan diketahui adanya fakta-fakta.
2. Mendapatkan fakta baru menggunakan aturan-aturan yang premisnya sesuai dengan fakta yang diketahui.

3. Proses tersebut dilanjutkan hingga tujuannya tercapai atau sampai tidak ada lagi aturan yang premisnya sesuai dengan fakta yang ada.

Contoh, jika diketahui kaidah tipe JIKA...MAKA berikut:

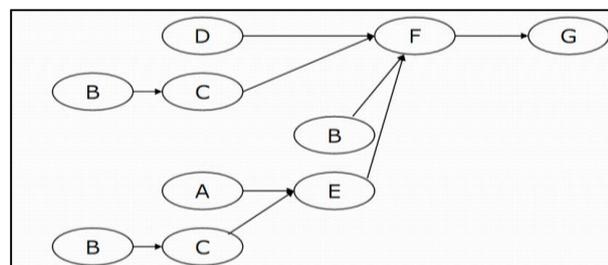
Jika A maka B

Jika B maka C

Jika C maka D

Jika fakta A diketahui dan mesin inferensinya dirancang untuk mencocokkan fakta melawan antecedent maka fakta selanjutnya, B dan C, akan dinyatakan dan konklusinya adalah D.

Berikut merupakan gambaran proses yang akan dilalui oleh metode *forward chaining*.



**Gambar 6.** Proses *Forward Chaining*

Aturan diatas dapat juga dituliskan dengan logika sebagai berikut:

1. Step I : IF B then C A,B,C -> True = R1
2. Step II : IF A and C Then E = R2
3. Step III : If A and C then E A,B,C -> True = R3
4. Step IV : If B ad E then F A,B,C,E,F -> true = R4
5. Step V : if F then G. G->True

## **D. Asma**

### **1. Pengertian Asma**

“Asma adalah suatu penyakit pernafasan yang ditandai dengan inflasi saluran pernafasan (bronkus) yang menyebabkan aliran udara ke dan dari paru menjadi kurang lancar, sehingga menimbulkan gejala-gejala khas yaitu mengi, batuk, konstriksi dada, dan sesak nafas. Selama berlangsungnya serangan asma, otot-otot bronkus mengencang, lapisan mukosa saluran pernafasan membengkak, dan produksi mukus/lendir saluran pernafasan meningkat secara berlebihan sehingga mengakibatkan penyempitan saluran nafas, kondisi ini umumnya dihubungkan dengan adanya faktor alergi, genetik, lingkungan, dan berat badan.”

(MIMS Indonesia Petunjuk Konsultasi, Edisi 11 2011/2012)

Asma didefinisikan menurut ciri-ciri klinis, fisiologis dan patologis.

Ciri-ciri klinis yang dominan adalah riwayat episode sesak, terutama pada malam hari yang sering disertai batuk. Pada pemeriksaan fisik, tanda yang sering ditemukan adalah mengi. Ciri-ciri utama fisiologis adalah episode obstruksi saluran napas, yang ditandai oleh keterbatasan arus udara pada ekspirasi. Sedangkan ciri-ciri patologis yang dominan adalah inflamasi (peradangan) saluran napas yang kadang disertai dengan perubahan struktur saluran napas.

Asma dipengaruhi oleh dua faktor yaitu genetik dan lingkungan, mengingat patogenesisnya tidak jelas, asma didefinisikan secara deskripsi yaitu penyakit inflamasi (peradangan) kronik saluran napas yang menyebabkan hipereaktivitas bronkus terhadap berbagai rangsangan, dengan gejala episodik berulang berupa batuk, sesak napas, mengi dan rasa berat di dada terutama pada malam dan atau dini hari, yang umumnya bersifat reversibel baik dengan atau tanpa pengobatan.

Asma adalah penyakit yang masih menjadi masalah kesehatan masyarakat di hampir semua negara di dunia, diderita oleh anak-anak sampai dewasa dengan derajat penyakit yang ringan sampai berat, bahkan dapat mematikan.

Menurut *The National Asthma Education and Prevention Program*, NAEPP (2007) “asma adalah gangguan inflamasi kronik jalan udara yang melibatkan peran banyak sel dan komponennya. Pada individu yang rentan, inflamasi menyebabkan episode berulang dari bengek, sesak nafas, sempit dada, dan batuk”.

## **2. Faktor Pencetus Asma**

### **a. Pemicu Asma (*Trigger*)**

Pemicu asma dapat mengakibatkan mengencang atau menyempitnya saluran pernapasan (bronkokonstriksi). Gejala-gejala dan bronkokonstriksi yang diakibatkan oleh pemicu cenderung timbul seketika, berlangsung dalam waktu pendek dan relatif mudah diatasi dalam waktu singkat.

Umumnya pemicu mengakibatkan bronkokonstriksi termasuk stimulus sehari-hari, seperti perubahan cuaca dan suhu udara, polusi udara, asap rokok, infeksi saluran pernapasan, gangguan emosi, dan olahraga yang berlebihan.

### **b. Penyebab Asma (*Inducer*)**

Penyebab asma dapat menyebabkan peradangan (inflamasi) dan sekaligus hiperresponsivitas (respon yang berlebihan) dari saluran

pernapasan. Penyebab asma dapat menimbulkan gejala-gejala yang umumnya berlangsung lebih lama (kronis), dan lebih sulit diatasi.

Umumnya penyebab asma adalah *allergen*, yang tampil dalam bentuk ingestan yaitu alergen yang masuk tubuh melalui mulut, inhalan yaitu alergen yang dihirup masuk tubuh melalui hidung atau mulut, dan alergen yang didapat melalui kontak dengan kulit.

### 3. Klasifikasi Asma

Asma dapat diklasifikasikan berdasarkan etiologi, berat penyakit dan pola keterbatasan aliran udara. Klasifikasi asma berdasarkan berat penyakit penting bagi pengobatan dan perencanaan penatalaksanaan jangka panjang, semakin berat asma semakin tinggi tingkat pengobatan. Pengklasifikasian asma dapat dilakukan dengan pengkajian terhadap gejala dan kemampuan fungsi paru. Semakin sering gejala yang dialami, maka semakin parah asma tersebut. Begitu juga dengan kemampuan fungsi paru yang diukur dengan *Peak Flow Meters* untuk mengetahui *Peak Expiratory Flow* (PEF) dan *Spyrometers* untuk mengukur *Force Expiratory Volume* dalam satu detik ( $FEV_1$ ) disertai dengan *Force Vital Capacity* (FVC), semakin rendah kemampuan fungsi paru, maka semakin parah asma tersebut (GINA, 2004).

Asma dapat diklasifikasikan pada saat tanpa serangan dan pada saat serangan. Tidak ada satu pemeriksaan tunggal yang dapat menentukan berat-ringannya suatu penyakit, pemeriksaan gejala-gejala dan uji faal paru berguna untuk mengklasifikasi penyakit menurut berat ringannya.

Klasifikasi itu sangat penting untuk penatalaksanaan asma. Berat ringan asma ditentukan oleh berbagai faktor seperti gambaran klinis sebelum pengobatan (gejala, eksaserbasi, gejala malam hari, pemberian obat inhalasi  $\beta$ -2 agonis, dan uji faal paru) serta obat-obat yang digunakan untuk mengontrol asma (jenis obat, kombinasi obat dan frekuensi pemakaian obat).

“Derajat berat asma dapat dikelompokkan sebagai asma intermiten, asma persisten ringan, asma persisten sedang dan asma persisten berat, tergantung pada frekwensi dan derajat berat gejalanya, termasuk gejala malam, episode serangan dan faal paru” (Sharma dalam Subijanto, 2008).

Tingkat keparahan dan gejala asma:

**a. Asma Intermiten Ringan**

Tingkat keparahan asma intermiten ringan berdasarkan buku MIMS Petunjuk Konsultasi 2011/2012 memiliki gejala: serangan asma kurang dari satu kali seminggu, serangan asma malam 2 kali sebulan atau kurang, serangan berlangsung selama tidak lebih dari beberapa jam, tidak ada gejala (faal paru normal) diantara serangan, dan fungsi paru masih normal.

Disebut asma intermiten bila pasien mengalami episode batuk atau sesak (*eksaserbasi*) kurang dari sekali seminggu dalam jangka waktu sedikitnya 3 bulan, serangan asma malam 2 kali sebulan atau kurang, *eksaserbasi* hanya berlangsung beberapa jam, eksasebasi (bisa berjalan ketika bernapas, bisa mengucapkan kalimat penuh, Respiratory

Rate (RR) meningkat, biasanya tidak ada gejala retraksi iga ketika bernapas), nilai PEF normal dalam kondisi serangan asma, dan fungsi paru masih normal.

Terapi untuk asma intermiten adalah tidak diperlukan pengobatan harian, bila terjadi serangan asma berat, dianjurkan pemberian kortikosteroid sistemik untuk jangka waktu singkat. Bronkodilator kerja singkat: 2-4 semprot  $\beta$ -2 agonis inhalasi kerja singkat, untuk mengatasi gejala. Intensitas terapi tergantung pada berat serangan, jika intensitasnya lebih dari 3 kali pengobatan dalam interval waktu 20 menit atau memerlukan terapi inhalasi, maka dianjurkan pemberian kortikosteroid sistemik. Penggunaan  $\beta$ -2 agonis inhalasi kerja singkat lebih dari 2 kali dalam 1 minggu pada asma intermiten (setiap hari, atau kebutuhan inhaler yang meningkat pada asma persisten) menandakan peningkatan kebutuhan terapi kontrol jangka lama Bronkodilator kerja singkat, terutama  $\beta$ 2 agonis inhalasi direkomendasikan sebagai pengobatan pelega cepat untuk mengobati gejala pada asma intermiten. Aksi utama  $\beta$ 2 agonis adalah untuk merelaksasikan otot polos jalan napas dengan menstimulus  $\beta$ 2 reseptor, sehingga meningkatkan siklik AMP dan menyebabkan bronkodilatasi. Salbutamol adalah  $\beta$ 2 agonis inhalasi yang memiliki profil keamanan baik.

## **b. Asma Persisten Ringan**

Tingkat keparahan asma persisten ringan berdasarkan buku MIMS Petunjuk Konsultasi 2011/2012 memiliki gejala: serangan asma lebih dari 1 x seminggu, tetapi tidak timbul setiap hari, serangan asma malam lebih dari 2 x sebulan, dan serangan asma mengganggu aktifitas.

Disebut asma persisten ringan bila penderita mengalami *eksaserbasi* paling tidak sekali seminggu, tetapi tidak timbul setiap hari, pada saat *eksaserbasi* membaik ketika duduk, bisa mengucapkan kalimat frase, RR meningkat, kadang-kadang menggunakan retraksi iga ketika bernapas dan beberapa *eksaserbasi* mempengaruhi tidur dan aktivitas, dan atau jika pasien memiliki gejala kronis yang memerlukan pengobatan simptomatis hampir setiap hari dan kejadian gejala asma malam lebih dari 2 kali sebulan.

Terapi yang dianjurkan untuk pengobatan kontrol jangka lama pada asma persisten ringan adalah kortikosteroid inhalasi dosis rendah. Kortikosteroid merupakan terapi preventif dan bekerja luas pada proses inflamasi. Efek klinisnya ialah mengurangi gejala beratnya serangan, perbaikan arus puncak ekspirasi dan spirometri, mengurangi hiperresponsif jalan napas, mencegah serangan dan mencegah remodelling dinding jalan napas (NAEPP, 2005). Kortikosteroid mencegah pelepasan sitokin, pengangkutan eosinofil jalan napas dan pelepasan mediator inflamasi (NAEPP, 2003).

### **c. Asma Persisten Sedang**

Tingkat keparahan asma persisten sedang berdasarkan buku MIMS Petunjuk Konsultasi 2011/2012 memiliki gejala: gejala timbul setiap hari, serangan asma malam terjadi lebih dari satu kali dalam seminggu, serangan asma yang lebih berat terjadi sekurang-kurangnya dua kali seminggu dan berlangsung selama sehari-hari, Serangan membutuhkan pengobatan setiap hari, dan serangan asma mengganggu aktifitas sehari-hari.

Asma persisten sedang yaitu khas ditandai gejala harian dalam jangka waktu lama atau serangan asma malam terjadi lebih dari sekali seminggu, serangan asma yang berat terjadi sekurang-kurangnya dua kali seminggu dan berlangsung selama sehari-hari, serangan membutuhkan pengobatan setiap hari, dan serangan asma mengganggu aktifitas sehari-hari. Pada saat eksaserbasi duduk tegak ketika bernapas, hanya dapat mengucapkan kata per kata, RR 30x/menit, biasanya menggunakan retraksi iga ketika bernapas.

Terdapat dua pilihan terapi, kombinasi kortikosteroid inhalasi dosis rendah dan  $\beta$ 2 agonis inhalasi kerja lama atau meningkatkan dosis kortikosteroid inhalasi sampai dosis medium.

### **d. Asma Persisten Berat**

Tingkat keparahan asma persisten berat berdasarkan buku MIMS Petunjuk Konsultasi 2011/2012 memiliki gejala: gejala asma

berlangsung terus menerus dan terjadi setiap hari, serangan berat sering terjadi, serangan asma malam sering terjadi, dan aktifitas fisik terbatas.

Asma persisten berat yaitu penderita mengalami variabilitas yang besar, gejala yang terus menerus dan serangan asma malam sering terjadi, mempunyai aktivitas yang terbatas, dan kadang mengalami *eksaserbasi* berat walaupun sedang dalam pengobatan.

Terapi yang diberikan adalah jika pengobatan asma persisten sedang telah dicapai tetapi masih membutuhkan tambahan terapi, maka dosis kortikosteroid inhalasi harus dinaikkan sampai batas dosis tinggi, serta penambahan terapi budesonid. Jika cara ini gagal dalam mengatasi gejala asma, maka dianjurkan untuk penambahan kortikosteroid sistemik (NAEPP, 2005).

## **E. Android**

### **1. Definisi Android**

Menurut Yosef Murya (2014:1) “Android adalah sistem operasi berbasis Linux yang digunakan untuk telepon seluler (*mobile*) seperti telepon pintar (*smartphone*) dan komputer tablet (PDA).”

Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan oleh bermacam piranti bergerak. Perkembangan Android tidak lepas dari peran sang raksasa Google. Android pada mulanya didirikan oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White pada tahun 2003.

## 2. Sejarah Perkembangan Android

Akhir tahun 2007, sebuah kelompok industri muncul bersamaan dengan *Platform* android untuk membentuk *Open Handset Alliance*. Beberapa diantaranya adalah Sprint Nextel, T-Mobile, Motorola, Samsung, Sony Ericson, Toshiba, Vodafone, Google, Intel, Texas Instrument.

Android sebagai sebuah sistem operasi terbuka yang dikembangkan oleh sebuah perusahaan yang besar, perkembangan android sangat pesat. Android memiliki beberapa versi sebagai pengembangan dari versi sebelumnya, berikut ini beberapa perkembangan android dari tahun ke tahun menurut Wikipedia.

### a. Android versi 1.0

Android versi 1.0 dikenal dengan kode nama *apple pie*, dirilis pada tanggal 23 September 2008 dengan ukuran layar 320 X 480 HVGA.

### b. Android versi 1.1

Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

### c. Android versi 1.5 (*Cupcake*)

Android versi 1.5 dikenal dengan kode nama *cupcake*. Versi ini dirilis pada tanggal 30 April 2009. Pembaruan fitur pada versi ini yaitu mampu merekam dan menonton video, mengunggah video ke youtube dan gambar ke picassa.

**d. Android versi 1.6 (Donut)**

Android versi 1.6 dikenal dengan kode nama *donut*. Versi ini dirilis pada tanggal 15 September 2009. Versi ini merupakan perbaikan dari android *cup cake* dengan menampilkan proses pencarian yang lebih baik dibandingkan sebelumnya, penggunaan baterai indikator dan control applet VPN proses.

**e. Android versi 2.0/2.1 (Eclair)**

Android versi 2.0/2.1 dirilis pada tanggal 3 Desember 2009, dikenal dengan kode nama *eclair*. Versi ini dilengkapi dengan Google Maps 3.2.1, *browser* baru dan dukungan *flash* untuk kamera 3,2 MP, digital zoom, dan Bluetooth 2.1.

**f. Android versi 2.2 (Froyo : Frozen Yoghurt)**

Versi ini dirilis pada 20 Mei 2010, dengan kecepatan kinerja dan aplikasi naik 2 sampai 5 kali lipat lebih cepat.

**g. Android versi 2.3 (Gingerbread)**

Android versi 2.3 dirilis pada 6 Desember 2010, di kenal dengan kode namagingerbread. Pada versi ini terjadi peningkatan kualitas suara, dan format video.

**h. Android versi 3.0/3.1 (Honeycomb)**

Android versi 3.0 dikenal dengan kode nama *honeycomb*, versi ini dirilis pada tahun 2011. Pada versi ini mendukung *multi processor* dan juga peningkatan kemampuan *hardware* untuk grafis.

**i. Android versi 4.0 (*ICS: Ice Cream Sandwich*)**

Android versi 4.0 dirilis pada tanggal 19 Oktober 2011, dikenal dengan kode nama *ice cream sandwich*. Versi ini membawa fitur baru yaitu membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, perangkat tambahan fotografi, dan berbagai fitur lainnya.

**j. Android versi 4.1/4.2/4.3 (*Jelly Bean*)**

Android versi 4.1, 4.2, dan 4.3 dikenal dengan kode nama *jelly bean*. Android versi 4.1 dirilis pada tanggal 9 Juli 2012 dengan kelebihan antarmuka pengguna yang lebih halus, teks dua bahasa dan dukungan bahasa lainnya. Android versi 4.2 dirilis pada tanggal 13 November 2012 dengan kelebihan pada peningkatan kinerja layar kunci. Sedangkan android versi 4.3 dirilis pada tanggal 24 Juli 2013 dan sampai pada saat ini masih membutuhkan pengembangan pada fiturnya.

**k. Android versi 4.4 (*Kitkat*)**

Android versi 4.4 dikenal dengan kode nama *kitkat*. Versi ini dirilis pada tanggal 31 Oktober 2013 dengan kelebihan fitur dari versi sebelumnya yaitu mampu menangkap perintah dalam bentuk suara yang dikenal dengan *voice comment*.

**l. Android versi 5.0 (*Lollipop*)**

Dirilis 3 November 2014.

### m. Android versi 6.0 (*Marshmallow*)

Dirilis 28 Mei 2015.

### 3. Fitur Android

Menurut Yosef Murya (2014:11) :

Fitur yang tersedia di Android adalah:

- a. Kerangka aplikasi: memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- b. Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat telepon seluler.
- c. Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
- d. SQLite: untuk penyimpanan data.
- e. Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- f. GSM, Bluetooth, EDGE, 3G, 4G dan Wifi (tergantung piranti keras)
- g. Kamera, Global Positioning System (GPS), kompas, NFC dan accelerometer (tergantung piranti keras)

### 4. Arsitektur Android

Arsitektur android merupakan sebuah kernel Linux dan sekumpulan pustaka C/C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.

Secara garis besar sistem operasi Android terbagi menjadi lima tingkatan :

#### a. *Linux Kernel*

Linux kernel merupakan kernel dasar pada arsitektur android.

Semua *driver* perangkat tingkat rendah untuk komponen-komponen *hardware* perangkat android berada pada tingkat linux kernel ini.

b. *Libraries*

*Libraries* merupakan tempat penyimpanan kode-kode program tentang layanan-layanan utama pada sistem operasi android. Contohnya *library text-to-speech* dan *speech-to-text* yang memungkinkan perubahan dalam bentuk teks menjadi suara dalam bahasa inggris atau sebaliknya. *Library SQLite*, yang menyediakan dukungan *database* untuk penyimpanan data.

c. *Android Runtime*

Kedudukan android *runtime* setingkat dengan *libraries*, *Android runtime* menyediakan kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi android dengan bahasa pemrograman java. *Dalvik virtual machine* aktif setiap kali aplikasi android berproses. *Dalvik* adalah mesin semu yang dirancang khusus untuk android yang dapat mengoptimalkan daya *battery* perangkat bergerak dengan memori dan CPU terbatas.

d. *Application Framework*

*Application Framework* berupa kumpulan *class built-in* yang tertanam dalam sistem operasi android, hal ini memungkinkan pengembang dapat memanfaatkannya untuk aplikasi yang sedang dibangun.

e. *Applications*

Tingkatan *applications* merupakan tingkatan tempat pengembang bekerja. Semua aplikasi yang dibuat akan terletak pada tingkatan ini.

Pada tingkatan *applications*, ada beberapa contoh aplikasi yaitu *home*, *contact*, *phone*, *browser*, dan lain sebagainya.

## **F. Desain Antarmuka**

Perancangan antarmuka pengguna dapat menciptakan media komunikasi yang efektif diantara manusia dan komputer. Faktor antarmuka ini sering dilupakan oleh para pengembang aplikasi karena terfokus pada fungsi. Kesalahan utama yang sering terjadi adalah anggapan pengembang aplikasi yaitu semakin baik fungsi dari aplikasi yang dibangun, semakin baik juga penerimaan para pengguna, padahal faktor-faktor non-teknis lainnya juga memegang peranan yang penting.

Menurut Shneiderman (2005:75-76) terdapat 8 aturan emas (*8 golden rules*) mengenai desain antarmuka antar manusia dan komputer, yaitu:

### **1. *Strive for consistency***

Rancangan antar muka aplikasi atau sistem harus konsisten. Hal ini meliputi tata letak tombol atau menu, warna yang digunakan, urutan menu, fitur *help*, dan bahasa yang digunakan serta urutan proses sistem. Sehingga *user* tidak mengalami kesulitan atau kebingungan ketika menggunakan aplikasi atau sistem tersebut.

### **2. *Enable frequent users to use shortcut***

Meningkatnya kebutuhan *user* menggunakan aplikasi, menimbulkan peningkatan keinginan *user* terhadap aplikasi yang mudah digunakan dan memiliki interaksi yang sedikit sehingga mempercepat kerja mereka. Oleh

karena itu *design interface* aplikasi perlu menyediakan fitur *shortcut* atau *automation* untuk mengurangi interaksi tersebut.

### **3. Offer informative feedback**

Adanya *feedback* yang informatif pada setiap operasi yang dijalankan. Apabila operasi yang dijalankan bersifat *frequent* seperti *save* atau *delete*, respon yang diberikan dapat secara sederhana. Namun, semakin penting suatu operasi dan bersifat jarang, maka respon yang diberikan harus semakin informatif. Hal yang perlu diperhatikan dalam memberikan respon suatu operasi adalah informasi yang diberikan harus komprehensif dan bermakna, serta memperhatikan *timing* kemunculannya secara tepat. Respon dapat berupa *dialog box*, *alert*, *progress bar* atau yang lainnya.

### **4. Design dialog to yield closure**

Adanya informasi bahwa suatu operasi telah selesai dilakukan. Sehingga pengguna dapat melanjutkan proses selanjutnya atau menggunakan fitur lainnya. Informasi kepada pengguna dapat berupa *dialog box* yang menyatakan suatu operasi telah selesai atau dengan *action grouping* agar pengguna mendapatkan informasi operasi-operasi apa yang sudah dilakukan dan operasi apa selanjutnya.

### **5. Offer error prevention and simple error handling**

Aplikasi dirancang sebaik mungkin agar pengguna terhindar dari kesalahan yang signifikan. Maka diperlukan petunjuk yang jelas serta pemberitahuan penyebab kesalahan kepada pengguna ketika terjadi *error*. Pencegahan *error* ini dapat dilakukan dengan proses validasi input sebelum

mengirimkan (*submit*) atau dengan menampilkan petunjuk (*hints*) saat pengguna berinteraksi dengan aplikasi.

#### **6. *Allow easy reversal of action***

Sediakan fitur untuk mengembalikan suatu operasi seperti tombol *undo* atau tombol *reset*. Hal ini akan mengurangi kecemasan pengguna ketika melakukan kesalahan dan menghindari mencoba fitur-fitur yang tidak dipahami.

#### **7. *Support internal lows of control***

Umumnya pengguna selalu memiliki keinginan untuk memegang kendali atas apa yang dilakukan oleh aplikasi. Oleh karena itu, perlu adanya fitur-fitur yang menunjukkan bahwa penggunalah yang memegang kendali dan aplikasi bekerja sesuai apa yang diperintahkan oleh pengguna. Hal ini dapat dilakukan salah satunya dengan memberikan fitur *cancel* pada operasi yang sedang berjalan atau *form* pengaturan.

#### **8. *Reduce short-term memory load***

Secara alamiah manusia memiliki keterbatasan dalam memproses suatu informasi pada ingatan jangka pendeknya. Oleh karena itu, tampilan dan proses pada suatu aplikasi juga harus memperhatikan hal tersebut. Tampilan harus dirancang sesederhana mungkin, namun berisi informasi yang komprehensif. Sistem kognitif manusia akan lebih mudah untuk mengenali dibandingkan dengan mengingat. Oleh karena itu, penggunaan *iconography*, warna dan visual lainnya akan membantu pengguna aplikasi menemukan fungsi yang dibutuhkan.

## G. Pemodelan Sistem dengan Unified Modeling Language (UML)

Pemodelan sistem ini menggunakan metode *Object Oriented Programming* (OOP) dengan menggunakan *Unified Modelling Language* (UML) untuk menggambarkan pemodelan sistem.

Menurut Fowler (2005:1), “*Unified Modelling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO).”

Beberapa tujuan utama dalam desain UML adalah sebagai berikut:

- a. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspesif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
- b. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.
- c. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
- d. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
- e. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek program (OOP).
- f. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
- g. Memiliki integritas praktik terbaik.

Sejatinya terdapat sembilan jenis diagram yang disediakan oleh UML.

Jenis diagram itu antara lain:

### 1. *Class Diagram*

Diagram kelas ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi dan relasi-relasi. Diagram kelas ini banyak dijumpai pada pemodelan sistem berorientasi objek. Whitten dalam Prabowo P. Widodo (2011:39) mengartikan kelas sebagai satu set objek yang memiliki atribut dan perilaku yang sama. Kelas kadang-kadang disebut sebagai kelas objek. Suatu objek merupakan isi (*instance*) dari kelas.

Menurut Pilone dalam Prabowo P. Widodo (2011:40) kelas menggambarkan suatu group yang memiliki kesamaan keadaan dan perilaku. Kelas merupakan cetak biru suatu objek dalam sistem orientasi objek. Dapat dikatakan kelas adalah sejenis alat pengklasifikasi. Sebagai contoh, *Volkswagen*, *Toyota* dan *Ford* merupakan kumpulan mobil sehingga kita dapat mengelompokkannya dalam kelas yang diberi nama mobil. Suatu kelas bisa menyatakan konsep yang dapat dilihat maupun abstrak.

UML menyarankan cara pemberian nama yang baik untuk membuat diagram kelas Prabowo P. Widodo (2011: 40) yaitu.

- a. Dimulai dengan huruf besar.
- b. Di tengah pada kompartemen bagian atas.
- c. Ditulis dengan cetak tebal.

d. Ditulis dengan cetak *italic* jika kelas tersebut abstrak.

## 2. *Package Diagram*

Diagram ini memperlihatkan kumpulan kelas-kelas yang merupakan bagian dari diagram komponen. Tiap paket memiliki nama yang merupakan ruang lingkungannya. Menurut Pilone dalam Prabowo P. Widodo, (2011:80) “Diagram paket juga dapat memvisualisasikan ketergantungan antar bagian dalam suatu sistem dan bermanfaat dalam mencari letak permasalahan dalam kompilasi.

## 3. *Use Case Diagram*

Diagram use case ini memperlihatkan himpunan use case dan aktor-aktor. Diagram ini sangat penting untuk mengorganisasikan dan memodelkan perilaku suatu sistem yang dibutuhkan pengguna. Menurut Pilone dalam Prabowo P. Widodo (2011:21) “*Use case* menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas”. Komponen pembentuk diagram *use case* adalah:

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use case*, aktivitas atau sarana yang disiapkan oleh sistem.
- c. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

Adapun aturan-aturan penulisan diagram *use case* yang disarankan oleh Universitas Cambridge menurut Prabowo P. Widodo (2011: 32) adalah sebagai berikut.

a. Panduan Pembuatan Aktor

Sebagai objek penting dalam UML, aktor harus dibuat sejelas mungkin untuk mempermudah orang lain memahami diagram *use case* yang dibuat. Tempatkan aktor utama di pojok kiri atas. Karena saat ini sebagian besar rancangan sistem mengutamakan pelanggan maka aktor utamanya pelanggan.

- 1) Gambarkan aktor terpisah dengan *use case*.
- 2) Berilah nama aktor dengan kata benda tunggal.
- 3) Aktor minimal harus terhubung dengan satu *use case*.
- 4) Berilah nama aktor sesuai dengan perannya terhadap model bukan jabatannya.
- 5) Tambahkan <<system>> pada aktor berjenis sistem.
- 6) Jangan menghubungkan langsung antara aktor satu dengan yang lain (kecuali generalisasi). Aktor satu dengan yang lainnya harus terhubung lewat *use case*.
- 7) Tambahkan aktor “waktu (*time*)” untuk sistem yang terjadwal otomatis.

b. Panduan Membuat *Use Case*

Adapun panduan membuat *use case* adalah sebagai berikut.

- 1) Buatlah nama *use case* sejelas mungkin dan orientasinya terhadap *stakeholder/klien* bukan perancang sistem.
- 2) Susunlah *use case* berdasarkan urutannya dari atas ke bawah untuk mempermudah pembacaan.

Selain itu, Chonoles dalam Prabowo P. Widodo (2011:22) menawarkan cara untuk menghasilkan *use case* yang baik, yakni:

- 1) Pilihlah nama yang baik.
- 2) Ilustrasikan perilaku dengan lengkap.
- 3) Identifikasi perilaku dengan lengkap.
- 4) Menyediakan *use case* lawan (*inverse*).
- 5) Batasi *use case* hingga satu perilaku saja.
- 6) Nyatakan *use case* dari sudut pandang aktor.

c. Panduan Pembuatan Relasi

Berikut adalah panduan membuat actor adalah sebagai berikut.

- 1) Hindari penggunaan anak panah antara aktor dan *use case* kecuali salah satunya bersifat pasif.
- 2) Gunakan `<<include>>` jika kita yakin suatu *use case* harus melibatkan *use case* lain.
- 3) Gunakan `<<extend>>` jika suatu *use case* mungkin melibatkan *use case* lain.
- 4) Gunakan `<<extend>>` seperlunya karena kebanyakan `<<extend>>` membuat diagram sulit dibaca.
- 5) Gunakan kata *include* dan *extend* bukannya *includes* dan *extends*.
- 6) Tempatkan *included use case* di sebelah kanan *use case* dasar.
- 7) Tempatkan *extend use case* di bawah *use case* dasar.
- 8) Tempatkan generalisasi *use case* di bawah *use case* induk.

- 9) Tempatkan generalisasi aktor di bawah aktor induk.
- 10) Hindari pembuatan *use case* lebih dari dua tingkat.

#### **4. *Sequence Diagram***

Menurut Prabowo P. Widodo (2011:174), *sequence diagram* menggabungkan bersama diagram-diagram lainnya dan memperlihatkan komunikasi antara diagram-diagram tersebut. Pilone dalam Prabowo P. Widodo (2011:174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan.

Diagram urutan ini merupakan diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu. Anggota interaksi digambarkan dengan persegi panjang yang diberi nama garis hidup (*lifeline*). Anggota interaksi digambarkan dengan garis putus-putus dari persegi panjang turun ke bawah yang memperlihatkan berapa lama objek eksis dalam diagram.

#### **5. *Collaboration Diagram***

*Collaboration Diagram* merupakan perluasan dari *sequence diagram*. Diagram ini menggambarkan asosiasi di antara objek-objek dan urutan pesan-pesan yang terlewatkan. Dalam pengolahan data nilai pesan tersebut berupa perintah-perintah beserta nilai-nilai variabel atau konstanta yang dikirimkan suatu objek lainnya.

#### **6. *Statechart Diagram***

Menurut Pilone dalam Prabowo P. Widodo (2011:126), “Diagram *statechart* dalam UML kadang disebut dengan istilah diagram *state*

*machine*". Diagram ini menggambarkan perilaku sistem perangkat lunak yang kita buat dan perilaku kelas, subsistem dan seluruh aplikasi. Selain itu diagram *state machine* bermanfaat juga untuk menyediakan cara yang baik dalam memodelkan komunikasi yang terjadi dengan entitas luar via protocol atau sistem dasarnya.

Prabowo P. Widodo (2011,126) juga menyebutkan UML memiliki dua *state machine* antara lain:

a. *Behavioral State Machine*

Mesin ini digunakan untuk menunjukkan perilaku elemen yang dimodelkan, misalnya suatu objek. Suatu *behavioral state machine* mempresentasikan implementasi khusus suatu elemen.

b. *Protocol State Machine*

Mesin ini digunakan untuk menunjukkan perilaku protocol. *Protocol state machine* menunjukkan bagaimana suatu elemen memicu perubahan kondisi protocol yang bertimpak pada suatu sistem.

## 7. *Activity Diagram*

Diagram aktivitas ini memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini juga sangat penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Menurut Prabowo P. Widodo (2011:143-144),

“Diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem dari pada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan model bisnis

juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi. Ketika digunakan dalam pemodelan software, diagram aktivitas mempresentasikan pemanggilan suatu fungsi tertentu misalnya *call*. Sedangkan bila digunakan dalam pemodelan bisnis, diagram ini menggambarkan aktivitas yang dipicu oleh kejadian-kejadian di luar seperti pemesanan atau kejadian-kejadian internal misalnya proses penggajian tiap Jumat sore”.

## 8. *Component Diagram*

Diagram komponen ini memperlihatkan organisasi serta ketergantungan sistem pada komponen-komponen yang telah ada sebelumnya. Komponen digambarkan dengan persegi panjang dengan *stereotyped* bertuliskan `<<component>>`. Detail dari komponen sendiri tidak ditulis.

Menurut Chonoles dalam Prabowo P. Widodo (2011:93), manfaat diagram komponen adalah bila ada salah satu komponen yang rusak atau tidak sesuai dengan tujuan sistem, kita tinggal mengganti komponen itu dengan komponen yang lain. Dalam membongkar pasang suatu komponen, yang perlu diperhatikan adalah batasan (*boundary*) komponen itu.

Menurut Prabowo P. Widodo (2011:94), agar komponen dapat dibongkar pasang, harus memenuhi kriteria sebagai berikut:

- a. Memiliki kerja internal yang tersembunyi. Objek yang berada di dalamnya harus bebas akses oleh objek di luar komponen.
- b. Memiliki antar muka (*interface*). *Interface* mendeskripsikan operasi apa yang harus diambil terhadap suatu komponen dan bukan bagaimana operasi itu dikerjakan.

- c. Komponen di dalam harus *independent*. Objek di dalam komponen tidak tahu menahu tentang objek di luar komponen.
- d. Antar muka terhadap komponen lain harus tersedia.

Pender dalam Prabowo P. Widodo (2011:106) menyatakan bahwa komponen bisa terdiri dari beberapa kategori besar:

- a. Komponen *deploy* yang diperlukan untuk menjalankan sistem.
- b. Produk kerja komponen termasuk model-model *source code* dan data file yang digunakan untuk membentuk komponen *deploy*.
- c. Komponen eksekusi, adalah komponen yang dibentuk saat aplikasi berjalan.

## 9. *Deployment Diagram*

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan, oleh karena itu diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin. Diagram *deployment* memodelkan sistem dengan mengidentifikasi perangkat kerasnya. Perangkat keras itu bisa berupa komputer, piranti, *network* atau bahkan operator yang melakukan proses manual.

## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Adapun kesimpulan yang dapat diambil dari Perancangan Sistem Pakar Diagnosis Asma dan Terapinya dengan Metode *Forward Chaining* Berbasis Android adalah sebagai berikut :

1. Dengan pemanfaatan teknologi perangkat bergerak (*mobile device*) dan perkembangan lingkungan pengembang (*development environment*), maka dapat merancang sebuah aplikasi *mobile* berbasis android, seperti “Perancangan Sistem Pakar Diagnosis Asma dan Terapinya dengan Metode *Forward Chaining* Berbasis Android”.
2. Dengan menggunakan metode *Forward Chaining* kita lebih mudah melakukan penarikan kesimpulan berdasarkan pengumpulan fakta-fakta terlebih dahulu.

#### **B. Saran**

Berdasarkan hasil implementasi dan rancangan aplikasi sistem pakar untuk mengetahui tingkat keparahan asma, adapun saran untuk pengembangan berikutnya adalah sbagai berikut:

1. Untuk pengembangan aplikasi ini kedepannya bisa digunakan untuk sistem *database* terpadu dan terbuka untuk umum secara *online* yang bisa dikonsumsi oleh publik.

2. Untuk pengembangan berikutnya agar ada riset lebih lanjut dan mendapatkan pakar yang lebih kompeten untuk menunjang sistem pakar ini. Sehingga perhitungan atau proses penarikan kesimpulan lebih baik dan lebih akurat.

## DAFTAR PUSTAKA

- Anita Desiani dan Muhammad Arhami. 2006. *Konsep Kecerdasan Buatan*. Yogyakarta: Andi.
- Arlina Pramudianto dan Evaria. 2011. *MIMS Indonesia Petunjuk Konsultasi*. Jakarta: CMP Medica.
- Elin Yulinah Sukandar, dkk. 2008. *ISO Farmakoterapi*. Jakarta: ISFI Penerbitan.
- Global Initiative for Asthma*. 2014. *Pocket Guide for Asthma Management and Prevention*. E-Journal GINA. Hlm. 4—7.
- Kusrini. 2006. *Sistem Pakar, Teori dan Aplikasi*. Yogyakarta: Andi.
- Muhammad Arhami. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi.
- National Asthma Education and Prevention Program Expert Panel Report 3*. 2007. *Diagnosis and Management of Asthma*. U. S: National Heart Lung and Blood Institute.
- Prabowo Pudjo Widodo & Herlawati. 2011. *Menggunakan UML (Unified Modeling Language)*. Bandung: Informatika Bandung.
- Shneiderman, Ben. 2005. *Designing the user interface: strategies for effective human-computer interaction*. Amerika: Pearson Education. Ebook.
- Tim EMS. 2013. *Android All In One*. Jakarta: Alex Media Komputindo.
- Universitas Negeri Padang. 2011. *Buku Panduan Penulisan Tugas Akhir/Skripsi Universitas Negeri Padang*. Padang : Universitas Negeri Padang.
- Yekti Mumpuni dan Ari Wulandari. 2013. *Cara Jitu Mengatasi Asma pada Anak dan Dewasa*. Yogyakarta: Andi.
- Yosef Murya. 2014. *Android Black Box*. Jasakom.