

**METODE *FORWARD CHAINING* PADA SISTEM PAKAR  
UNTUK DIAGNOSA CACAT LAHIR AKIBAT FAKTOR  
TERATOGEN BERBASIS ANDROID**

**SKRIPSI**



**OLEH**

**DIAN PURNAMA SARI**

**G1A 009 009**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS BENGKULU  
2014**

**METODE *FORWARD CHAINING* PADA SISTEM PAKAR  
UNTUK DIAGNOSA CACAT LAHIR AKIBAT FAKTOR  
TERATOGEN BERBASIS ANDROID**

**SKRIPSI**

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Pada  
Fakultas Teknik Program Studi Teknik Informatika  
Universitas Bengkulu**



**OLEH  
DIAN PURNAMA SARI  
G1A 009 009**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS BENGKULU  
2014**

## MOTTO

- ❖ *Aku tidak pernah sendiri, karena Allah tidak pernah meninggalkanku.*
- ❖ *Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya (QS. Al Baqarah:286).*
- ❖ *Kepunyaan Allah-lah segala yang ada di langit dan di bumi, dan kepada Allah-lah dikembalikan segala urusan (QS. Ali Imran:109).*
- ❖ *Bersyukur adalah cara paling sederhana untuk menyenangkan diri, yang tidak dapat dibeli dengan apapun namun sulit untuk dilakukan.*
- ❖ *Pengorbanan, perjuangan, air mata, kesabaran, dan keikhlasanlah yang menjadi teman perjalanan menuju kesuksesan.*
- ❖ *Cukup menjadi diri sendiri.*

## PERSEMBAHAN

*Skripsi ini kupersembahkan untuk :*

- ❖ *Allah S.W.T*
- ❖ *Nabi Muhammad S.A.W.*
- ❖ *Bapak dan Ibu tercinta (Abdul Aziz dan Nurlela) yang selalu mendoakan, memberikan kasih sayang secara utuh dengan tulus dan ikhlas, serta selalu menjadi alasan utamaku untuk tetap semangat dalam proses penyelesaian Tugas Akhir ini.*
- ❖ *Kakak-kakakku tersayang (Dang Desi dan Jnga Dwi).*
- ❖ *Datuk Abd.Sani, Nenek Aspah, dan Alm.Non Khatidjah.*
- ❖ *Sahabat-sahabat terbaik Teknik dan Teknik Informatika 2009.*
- ❖ *Almamaterku.*

## KATA PENGANTAR



السَّلَامُ عَلَيْكُمْ وَرَحْمَةُ اللَّهِ وَبَرَكَاتُهُ

*Alhamdulillahirobbilalamin*, penulis ucapkan kepada Allah SWT atas rahmat dan nikmat yang telah diberikan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Metode *Forward Chaining* Pada Sistem Pakar Untuk Diagnosa Cacat Lahir Akibat Faktor Teratogen Berbasis Android”**. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Program Studi Teknik Informatika Fakultas Teknik Universitas Bengkulu.

Selesainya penulisan dan penyusunan skripsi ini tidak terlepas dari bimbingan, arahan, masukan, dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Ibu dan Bapakku tercinta Nurlela dan Abdul Aziz yang telah memberi semangat, mendoakan, selalu mengerti keadaan penulis, dan membantu secara moril dan materil kepada penulis.
2. Kakak-kakakku tersayang Dang Desi dan Inga Dwi, kakak ipar Kak Charles dan Kak Heri, serta keponakanku tersayang Babay dan Abydh yang selalu memberikan keceriaan, motivasi, dan menjadi hiburan penulis dalam menjalani hari-hari perkuliahan dan penyelesaian skripsi ini.
3. Bapak Khairul Amri, S.T., M.T sebagai Dekan Fakultas Teknik Universitas Bengkulu.

4. Ibu Desi Andreswari, S.T., M.Cs sebagai Ketua Program Studi Teknik Informatika dan penguji pendamping yang telah memberikan dukungan dan bersedia meluangkan waktunya untuk menguji penulis sehingga dapat menyelesaikan skripsi ini dengan baik dan lancar.
5. Bapak Rusdi Efendi, S.T., M.Kom selaku dosen pembimbing utama yang penuh kecermatan, kesabaran dalam memberikan bimbingan dan dukungan serta motivasi bagi penulis dalam penyelesaian skripsi ini.
6. Bapak Dr. Aceng Ruyani, M.S yang telah berbesar hati meluangkan waktunya untuk membimbing, memberikan motivasi, dukungan, dan mengarahkan penulis dalam menyelesaikan skripsi ini.
7. Ibu DR. Diyah Puspitaningrum, S.T., M.Kom selaku dosen penguji utama yang telah memberikan bimbingan dan meluangkan waktunya untuk menguji penulis.
8. Bapak Drs. Boko Susilo, M.Kom selaku dosen Pembimbing Akademik yang telah memberikan bimbingan kepada penulis sejak dimulainya perkuliahan hingga penulis menyelesaikan perkuliahan.
9. Segenap Bapak dan Ibu Dosen Pengajar di Fakultas Teknik terutama Program Studi Teknik Informatika beserta staf–staf di Fakultas Teknik Universitas Bengkulu.
10. Keluarga besar Datuk Abd.Sani, Nenek Aspah , Alm. Non Khatijah, Kakak Saparudin, Omdan, Omjon, Omber, Dodo Era, Wancek Uji, Papi Johan, Abah Jon, serta sepupu-sepupu dan keponakanku.
11. Sahabat-sahabatku di Teknik Informatika 2009 (Ryza, Iin, Handri, Abdur, Feri, Disa, Odie, Sostri, Roro, Gita, Yody, Regia, Oni, Ariansyah, Julia,

Dita, Wisnu, Randy, Eko, Indri, Leni, Yessica, Mei, Apni, Lian, Yoggy, Azhar, Irawan, Radias, Chibi, Zulmi, Tri, Rinof, Rewa, Egi, Munadi, Daus, Fuad, Eleo, Aji, dan Robby,) yang melengkapi cerita dan langkah hidup penulis baik suka maupun duka.

12. Kakak senior dan adik-adik tingkat Teknik Informatika Universitas Bengkulu yang masih berjuang mencapai kesuksesan.
13. Sahabat KKN Desa Marga Bakti Kecamatan Ketahun Kabupaten Bengkulu Utara (Arif, Ryry, Citra, Anja, Fibrisio, Harum, Nova, Bang Cik, Fadli, dan Bang Yusril).
14. Adelia Puspa, S.Pd, Orlando Riki Akbar, Amd.Kep , Retno Tri Lestari, SST , dan teman-teman SDN 67, SMPN07, dan SMAN 04 Kota Bengkulu.
15. Semua pihak yang tidak bisa disebutkan satu persatu yang telah membantu penulis dalam menyelesaikan skripsi ini.

Akhirnya penulis berharap semoga semua perhatian, motivasi, bimbingan dan dukungan yang telah diberikan menjadi amal dan mendapat balasan yang setimpal dari Allah SWT. Semoga skripsi ini dapat bermanfaat bagi semua pihak di masa yang akan datang.

وَسَّلَامٌ عَلَيْكُمْ وَرَحْمَةُ اللَّهِ وَبَرَكَاتُهُ

Bengkulu, 18 Juni 2014

**Dian Purnama Sari**  
**G1A009009**

# **METODE *FORWARD CHAINING* PADA SISTEM PAKAR UNTUK DIAGNOSA CACAT LAHIR AKIBAT FAKTOR TERATOGEN BERBASIS ANDROID**

Oleh  
DIAN PURNAMA SARI  
NPM G1A009009  
E-mail : purnamasaridian18@yahoo.com

## **ABSTRAK**

Metode *forward chaining* adalah salah satu teknik mekanisme inferensi untuk pengujian aturan. *Forward chaining* atau pelacakan maju yaitu dimana aturan-aturan diuji satu demi satu dalam urutan tertentu yang berarti menggunakan aturan kondisi-aksi atau fakta-kesimpulan. Tujuan penelitian dalam tugas akhir ini adalah membangun aplikasi sistem pakar untuk mendiagnosa cacat lahir akibat faktor teratogen dengan menggunakan metode *forward chaining*. Aplikasi dibuat dengan bahasa pemrograman Java dengan bantuan *tool* IDE Eclipse Kepler, metode *Linear Model of Expert System Development*, dan analisis berorientasi objek *Unified Modeling Language* (UML). Aplikasi diuji dengan pengujian *white box*, *black box*, dan uji kelayakan sistem dengan teknik *linkert scale*. Dari hasil pengujian, aplikasi termasuk kedalam kategori baik dengan persentase variabel tampilan 65%, variabel kemudahan penggunaan 63%, variabel kinerja sistem 66%, dan variabel isi 69%. Aplikasi ini dapat berjalan pada sistem operasi android dengan *version android 2.3.3 – 4.2.2*. Berdasarkan hasil tersebut, aplikasi sistem pakar untuk mendiagnosa cacat lahir yang disebabkan oleh faktor teratogen berbasis android telah layak untuk digunakan.

**Kata kunci:** Sistem Pakar, *Forward Chaining*, Cacat Lahir, Teratogen, Android.

**ANDROID-BASED FORWARD CHAINING EXPERT SYSTEM  
METHOD FOR DIAGNOSING BIRTH DEFECTS  
CAUSED BY TERATOGEN FACTOR**

By

DIAN PURNAMA SARI

NPM G1A009009

E-mail : [purnamasaridian18@yahoo.com](mailto:purnamasaridian18@yahoo.com)

**ABSTRACT**

*Forward chaining method is one of mechanism inference techniques for testing rules. Forward chaining or advanced tracking is where the rules are tested one by one in a certain order, which means using rules of condition-action or fact-conclusion. The aim of this research was to build an expert system application for diagnosing birth defects caused by teratogen factors by using forward chaining method. This application was built by using the Java programming language with the help of Kepler Eclipse IDE tool, Linear Model of Expert System Development method, and object-oriented analysis by using the Unified Modeling Language (UML). The application was tested with white box testing, black box testing, and test the feasibility of the system with linkert scale technique. From the tests results, it can be seen that the application was included in good category, with the display variable percentage of 65%, the practical-use variable percentage of 63%, the performance variable percentage of 66%, and content variable percentage of 69%. This application can run on android system with android version of 2.3.3 – 4.2.2. Based on those results, the android-based forward chaining expert system method for diagnosing birth defects caused by teratogen factor has been feasible to use.*

**Key words:** *Expert System, Forward Chaining, Birth Defects, Teratogen, Android.*

## DAFTAR ISI

<b>HALAMAN JUDUL .....</b>	<b>i</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>ii</b>
<b>MOTTO DAN PERSEMBAHAN.....</b>	<b>iv</b>
<b>KATA PENGANTAR.....</b>	<b>v</b>
<b>ABSTRAK .....</b>	<b>viii</b>
<b>ABSTRACT .....</b>	<b>ix</b>
<b>DAFTAR ISI.....</b>	<b>x</b>
<b>DAFTAR GAMBAR.....</b>	<b>xiii</b>
<b>DAFTAR TABEL .....</b>	<b>xvi</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Batasan Masalah .....	5
1.4 Tujuan Penelitian .....	6
1.5 Manfaat Penelitian .....	6
<b>BAB II LANDASAN TEORI .....</b>	<b>7</b>
2.1 Sistem Pakar.....	7
2.1.1 Pengertian Sistem Pakar .....	7
2.1.2 Ciri-Ciri dan Karakteristik Sistem Pakar .....	8
2.1.3 Bidang-Bidang Pengembangan Sistem Pakar.....	9
2.1.4 Struktur Sistem Pakar .....	10
2.1.5 Kelebihan dan Kekurangan Sistem Pakar.....	20
2.1.6 Metode Pengembangan Sistem Pakar.....	21
2.2 Cacat Lahir .....	24
2.2.1 Pengertian Cacat Lahir.....	24
2.2.2 Penyebab Cacat Lahir .....	25
2.3 Ilmu Teratologi .....	26
2.3.1 Pengertian Ilmu Teratologi .....	26
2.3.2 Teratogen .....	27
2.4 Android .....	28
2.4.1 Sejarah Android .....	28
2.4.2 Fitur Android .....	29
2.4.3 Arsitektur Android.....	30
2.4.4 Komponen Android .....	33
2.4.5 Versi Android .....	35
2.4.6 Kelebihan dan Kekurangan Android .....	37
2.5 Eclipse.....	38

2.5.1	Pengertian Eclipse .....	38
2.5.2	Arsitektur Eclipse .....	39
2.6	<i>Unified Modelling Language (UML)</i> .....	40
2.6.1	Pengertian <i>Unified Modelling Language</i> .....	40
2.6.2	Diagram-Diagram UML .....	41
2.7	<i>Flowchart (Diagram Alir)</i> .....	47
<b>BAB III METODE PENELITIAN .....</b>		<b>48</b>
3.1	Jenis Penelitian.....	48
3.2	Sarana Pendukung Pembuatan Sistem .....	48
3.3	Jenis Data .....	49
3.4	Teknik Pengumpulan Data.....	49
3.5	Metode Pengembangan Sistem Pakar .....	50
3.6	Metode Uji Kelayakan Sistem .....	53
3.7	Diagram Alir Penelitian .....	54
3.8	Jadwal Penelitian.....	56
<b>BAB IV ANALISA DAN PERANCANGAN SISTEM.....</b>		<b>57</b>
4.1	Perencanaan.....	57
4.1.1	Penilaian Kelayakan .....	57
4.1.2	Spesifikasi Kebutuhan .....	59
4.1.3	Penggambaran Fungsi Awal Sistem.....	59
4.2	Definisi Pengetahuan .....	61
4.2.1	Proses Identifikasi dan Seleksi Sumber Pengetahuan.....	61
4.2.2	Proses Akuisisi, Analisa, dan Ekstraksi Pengetahuan .....	61
4.3	Desain Pengetahuan .....	64
4.3.1	Definisi Desain .....	64
4.3.2	Desain Secara Detail.....	64
4.4	Pengkodean .....	84
<b>BAB V HASIL DAN PEMBAHASAN .....</b>		<b>85</b>
5.1	Implementasi Sistem .....	85
5.1.1	Halaman Utama .....	86
5.1.2	Menu Konsultasi .....	87
5.1.3	Menu Daftar Cacat Lahir .....	90
5.1.4	Menu Daftar Agen Teratogen .....	93
5.1.5	Menu Periode Sensitif Kehamilan .....	95
5.1.6	Menu Bantuan.....	97
5.1.7	Menu Tentang .....	99
5.2	Implementasi Pada <i>Smartphone</i> Android .....	100
5.3	Verifikasi Pengetahuan .....	109
5.4	Uji Kelayakan Sistem.....	112

<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>119</b>
6.1 Kesimpulan .....	119
6.2 Saran.....	120
<b>DAFTAR PUSTAKA .....</b>	<b>121</b>
<b>LAMPIRAN.....</b>	<b>123</b>

## DAFTAR GAMBAR

Gambar 2.1	Struktur Bagan Sistem Pakar. ....	11
Gambar 2.2	<i>Graph</i> Pengetahuan .....	14
Gambar 2.3	<i>Depth First Search</i> .....	16
Gambar 2.4	<i>Breadth First Search</i> .....	18
Gambar 2.5	<i>Linear Model of Expert System Development</i> .....	22
Gambar 2.6	Arsitektur Android .....	31
Gambar 2.7	Arsitektur Eclipse .....	39
Gambar 2.8	Contoh <i>Use Case Diagram</i> .....	42
Gambar 2.9	Contoh <i>Class Diagram</i> .....	43
Gambar 2.10	Contoh <i>Statechart Diagram</i> .....	43
Gambar 2. 11	Contoh <i>Sequence Diagram</i> .....	44
Gambar 2. 12	Contoh <i>Component Diagram</i> .....	44
Gambar 2. 13	Contoh <i>Collaboration Diagram</i> .....	45
Gambar 2. 14	Contoh <i>Deployment Diagram</i> .....	45
Gambar 2. 15	Contoh <i>Activity Diagram</i> .....	46
Gambar 3. 1	Diagram Alir Penelitian .....	54
Gambar 4. 1	<i>Flowchart</i> Sistem Pakar .....	60
Gambar 4. 2	<i>Use Case Diagram</i> .....	65
Gambar 4. 3	<i>Class Diagram</i> .....	66
Gambar 4. 4	<i>Statechart Diagram</i> .....	67
Gambar 4. 5	<i>Sequence Diagram</i> Konsultasi .....	68
Gambar 4. 6	<i>Sequence Diagram</i> Daftar Cacat Lahir .....	69
Gambar 4. 7	<i>Sequence Diagram</i> Daftar Agen Teratogen .....	70
Gambar 4. 8	<i>Sequence Diagram</i> Periode Sensitif Kehamilan .....	71
Gambar 4. 9	<i>Sequence Diagram</i> Bantuan .....	71
Gambar 4. 10	<i>Sequence Diagram</i> Tentang .....	72
Gambar 4. 11	<i>Activity Diagram</i> Konsultasi .....	72
Gambar 4. 12	<i>Activity Diagram</i> Daftar Cacat Lahir .....	73
Gambar 4. 13	<i>Activity Diagram</i> Daftar Agen Teratogen .....	74
Gambar 4. 14	<i>Activity Diagram</i> Periode Sensitif Kehamilan .....	75
Gambar 4. 15	<i>Activity Diagram</i> Bantuan.....	75
Gambar 4. 16	<i>Activity Diagram</i> Tentang .....	76
Gambar 4. 17	<i>Communication Diagram</i> .....	76

Gambar 4. 18 <i>Component Diagram</i> .....	77
Gambar 4. 19 <i>Deployment Diagram</i> .....	77
Gambar 4. 20 Struktur Antarmuka Sistem .....	78
Gambar 4. 21 <i>Form</i> Halaman Utama .....	79
Gambar 4. 22 <i>Form</i> Menu Konsultasi .....	79
Gambar 4. 23 <i>Form</i> Menu Hasil Diagnosa .....	80
Gambar 4. 24 <i>Form</i> Menu Daftar Cacat Lahir .....	81
Gambar 4. 25 <i>Form</i> Menu Daftar Agen Teratogen.....	81
Gambar 4. 26 <i>Form</i> Menu Periode Sensitif Kehamilan.....	82
Gambar 4. 27 <i>Form</i> Menu Bantuan .....	82
Gambar 4. 28 <i>Form</i> Menu Tentang.....	83
Gambar 4. 29 Perancangan Strategi Implementasi .....	84
Gambar 5. 1 Tampilan Halaman Utama .....	87
Gambar 5. 2 Tampilan Menu Konsultasi .....	89
Gambar 5. 3 Tampilan Hasil Diagnosa .....	89
Gambar 5. 4 Tampilan Hasil Diagnosa Tidak Ditemukan.....	90
Gambar 5. 5 Tampilan Menu Daftar Cacat Lahir .....	92
Gambar 5. 6 Tampilan Detail Cacat Lahir .....	92
Gambar 5. 7 Tampilan Menu Daftar Agen Teratogen .....	94
Gambar 5. 8 Tampilan Detail Agen Teratogen .....	95
Gambar 5. 9 Tampilan Menu Periode Sensitif Kehamilan .....	96
Gambar 5. 10 Tampilan Menu Bantuan.....	98
Gambar 5. 11 Tampilan Menu Tentang .....	99
Gambar 5. 12 Tampilan Halaman Utama Pada <i>Smartphone</i> Android.....	101
Gambar 5. 13 Tampilan Menu Konsultasi Pada <i>Smartphone</i> Android.....	102
Gambar 5. 14 Tampilan Hasil Diagnosa Pada <i>Smartphone</i> Android.....	102
Gambar 5. 15 Tampilan Hasil Pemberitahuan Jika Kesimpulan Tidak Ditemukan Pada <i>Smartphone</i> Android.....	103
Gambar 5. 16 Tampilan Daftar Cacat Lahir Pada <i>Smartphone</i> Android .....	104
Gambar 5. 17 Tampilan Detail Cacat Lahir Pada <i>Smartphone</i> Android.....	104
Gambar 5. 18 Tampilan Menu Daftar Agen Teratogen Pada <i>Smartphone</i> Android.....	105
Gambar 5. 19 Tampilan Detail Agen Teratogen Pada <i>Smartphone</i> Android .....	106
Gambar 5. 20 Tampilan Menu Peroide Sensitif Kehamilan Pada <i>Smartphone</i> Android.....	107
Gambar 5. 21 Tampilan Menu Bantuan Pada <i>Smartphone</i> Android .....	107

Gambar 5. 22 Tampilan Menu Tentang Pada <i>Smartphone</i> Android .....	108
Gambar 5. 23 Grafik Persentase Hasil Angket Variabel Tampilan .....	115
Gambar 5. 24 Grafik Persentase Hasil Angket Variabel Kemudahan Penggunaan.....	116
Gambar 5. 25 Grafik Persentase Hasil Angket Variabel Kinerja Sistem.....	117
Gambar 5. 26 Grafik Persentase Hasil Angket Variabel Isi ( <i>Content</i> ) .....	118

## DAFTAR TABEL

Tabel 2.1	Simbol-simbol <i>Flowchart</i> .....	47
Tabel 3.1	Jadwal Penelitian.....	56
Tabel 4.1	Cacat Lahir .....	62
Tabel 4.2	Agen Teratogen Penyebab Cacat Lahir .....	62
Tabel 4.3	Hubungan Cacat Lahir dan Agen Teratogen.....	63
Tabel 4.4	Representasi Pengetahuan.....	64
Tabel 5.1	Pengujian <i>Black Box</i> Halaman Utama .....	87
Tabel 5.2	Pengujian <i>Black Box</i> Menu Konsultasi.....	90
Tabel 5.3	Pengujian <i>Black Box</i> Menu Daftar Cacat Lahir .....	93
Tabel 5.4	Pengujian <i>Black Box</i> Menu Daftar Agen Teratogen.....	95
Tabel 5.5	Pengujian <i>Black Box</i> Menu Periode Sensitif Kehamilan .....	97
Tabel 5.6	Pengujian <i>Black Box</i> Menu Bantuan.....	98
Tabel 5.7	Pengujian <i>Black Box</i> Menu Tentang.....	100
Tabel 5.8	Pengujian <i>Black Box</i> Halaman Utama Pada <i>Smartphone</i> Android..	101
Tabel 5.9	Pengujian <i>Black Box</i> Menu Konsultasi Pada <i>Smartphone</i> Android	103
Tabel 5.10	Pengujian <i>Black Box</i> Menu Daftar Cacat Lahir Pada <i>Smartphone</i> Android .....	105
Tabel 5.11	Pengujian <i>Black Box</i> Menu Daftar Agen Teratogen Pada <i>Smartphone</i> Android .....	106
Tabel 5.12	Pengujian <i>Black Box</i> Menu Periode Sensitif Kehamilan Pada <i>Smartphone</i> Android .....	107
Tabel 5.13	Pengujian <i>Black Box</i> Menu Bantuan Pada <i>Smartphone</i> Android ....	108
Tabel 5.14	Pengujian <i>Black Box</i> Menu Tentang Pada <i>Smartphone</i> Android ....	108
Tabel 5.15	Uji Keakuratan Sistem Pakar .....	115
Tabel 5.16	Kategori Penilaian.....	109
Tabel 5.17	Hasil Penialain Variabel Tampilan .....	114
Tabel 5.18	Hasil Penilaian Variabel Kemudahan Penggunaan.....	115
Tabel 5.19	Hasil Penilaian Variabel Kinerja Sistem.....	117
Tabel 5.20	Hasil Penialain Variabel Isi ( <i>Content</i> ) .....	118

## DAFTAR LAMPIRAN

Lampiran A-1	Hasil Pengujian Kelas dan <i>Layout</i> Sistem (Uji <i>White Box</i> ) .....	A-1
Lampiran B-1	Hasil Pengujian Fungsional Sistem (Uji <i>Black Box</i> ).....	B-1
Lampiran C-1	Wawancara ( <i>Interview</i> ) Pakar .....	C-1
Lampiran D-1	Angket Uji Kelayakan .....	D-1
Lampiran E-1	Tabel Penilaian Responden.....	E-1
Lampiran E-2	Tabel Persentase Variabel Tampilan .....	E-6
Lampiran E-3	Tabel Persentase Variabel Kemudahan Penggunaan.....	E-6
Lampiran E-4	Tabel Persentase Variabel Kinerja Sistem.....	E-7
Lampiran E-5	Tabel Persentase Variabel Isi ( <i>Content</i> ) .....	E-7
Lampiran F-1	Surat Keterangan Telah Melaksanakan Wawancara.....	F-1
Lampiran F-2	Surat Keterangan Sebagai Pakar .....	F-2
Lampiran G-1	Hasil Uji Coba Aplikasi Pada <i>Smartphone</i> Android.....	G-1

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pernikahan merupakan dambaan bagi setiap manusia untuk membina sebuah keluarga. Dengan adanya pernikahan diharapkan dapat membentuk sebuah keluarga yang harmonis dan bahagia serta memperoleh keturunan yang baik sehingga dapat menjadi kebanggaan orang tuanya. Pernikahan pada usia muda saat ini sering terjadi, baik itu disebabkan keinginan untuk menikah ataupun akibat pergaulan bebas yang mengharuskan seseorang untuk menikah.

Anak yang terlahir normal dan tumbuh dengan sehat akan menjadi kebanggaan tersendiri bagi orang tuanya. Namun, pengetahuan mengenai kesehatan kehamilan agar bayi yang dikandung dapat lahir dengan sehat dan sempurna tampaknya belum terlalu diketahui oleh masyarakat. Sebagai contoh bayi cacat yang lahir dari pasangan Riko Yan Saputra dan Welda Sulita, warga Jalan Bumi Ayu 5 No. 9 R.5 Rw.1, Kelurahan Kandang, Kecamatan Selebar, Kota Bengkulu. Bayi yang lahir pada Rabu, 11 Desember 2013 dalam keadaan normal lengkap dengan anggota tubuh lainnya, namun memiliki dua kepala yang terletak dibagian atas badan dan dibagian bawah diantara selangkang (Firmansyah, 2013).

Menurut Nadya (2013) bahwa 4.64% bayi baru lahir menderita cacat bawaan, 0.78% bayi baru lahir meninggal akibat cacat bawaan, dan bayi yang membawa cacat bawaan hidup sebanyak 4.15% tersebar di masyarakat Bengkulu. Persentase bayi yang membawa cacat bawaan hidup lebih tinggi dibandingkan pada tahun sebelumnya yang hanya sebanyak 1.38%.

Meningkatnya angka kasus bayi cacat lahir ini diduga bisa disebabkan oleh beberapa faktor seperti kurangnya tenaga kesehatan di daerah yang jauh dari perkotaan, kurangnya pengetahuan bagaimana menjaga kehamilan, faktor genetik, dan faktor lingkungan. Faktor-faktor yang mempengaruhi cacat lahir yang ditemukan di Rumah Sakit M. Yunus Bengkulu yaitu 60% disebabkan oleh faktor genetik dan lingkungan (multifaktorial), 20% disebabkan oleh faktor genetik, dan 20% disebabkan oleh faktor lingkungan (Nadya, 2013).

Faktor genetika yaitu faktor yang diturunkan oleh orang tua melalui ovum atau spermatozoa. Hal ini dapat terjadi karena adanya kerusakan gen tunggal dan kerusakan kromosom, contohnya akondroplasia (tubuh kerdil) dan sindrom marfan (penyakit jaringan ikat). Faktor lingkungan adalah faktor yang diterima selama proses kehamilan baik pada tahap embriogenesis ataupun fetogenesis, contohnya sindrom alkohol (*fetal alcohol syndrome*) yang disebabkan oleh konsumsi minuman alkohol pada saat hamil. Faktor genetik tidak dapat dicegah karena merupakan faktor bawaan, sedangkan faktor lingkungan masih dapat dicegah agar kehamilan tidak terganggu pada bayi yang dikandung.

Salah satu cara untuk mengatasi persoalan tersebut salah satunya dengan ilmu teratologi. Ilmu teratologi adalah ilmu pengetahuan yang mengkaji cacat lahir akibat substansi lingkungan. Substansi lingkungan adalah agen-agen lingkungan penyebab munculnya cacat lahir yang disebut dengan teratogen. Cacat lahir yang dimaksud adalah bayi yang mengalami cacat pada saat dilahirkan dan teratogen yang dimaksud adalah agen-agen yang diterima oleh janin yang berasal dari ibu selama proses kehamilan berlangsung.

Disamping itu, dengan perkembangan teknologi informasi yang ada dapat menjadi salah satu sarana untuk menyelesaikan permasalahan diatas. Salah satunya adalah perkembangan teknologi informasi sistem pakar. Sistem pakar merupakan program komputer yang dapat meniru proses pemikiran dan pengetahuan pakar untuk menyelesaikan suatu masalah yang spesifik. Implementasi sistem pakar banyak digunakan untuk kepentingan komersial karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar dalam bidang tertentu ke dalam suatu program, sehingga dapat memberikan keputusan dan melakukan penalaran secara cerdas. Diagnosa penyakit dengan menggunakan sistem pakar akan mencatat gejala-gejala dan akan mendiagnosa jenis penyakitnya yang berbasis pada pengetahuan yang didapat dari seorang pakar.

Sistem pakar sudah banyak dikembangkan oleh para peneliti. Sistem pakar dikembangkan mulai dari yang berbasis *stand alone*, berbasis WEB, dan berbasis telepon seluler atau *handphone* dengan teknologi J2ME. Baru-baru ini mulai berkembang ditengah masyarakat sistem operasi android yang merupakan sistem operasi berbasis linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi sendiri. Dengan adanya aplikasi android, diharapkan dapat memberikan kemudahan bagi pengguna android dalam mencari informasi. Selain itu, *smartphone* android masih terus mendominasi pasaran dan pada kuartal ketiga di tahun 2013, sebanyak 81.3 persen *smartphone* android berhasil mengalahkan *platform mobile* lainnya seperti BlackBerry ataupun Windows Phone. Peningkatan pengguna android ini menggambarkan bahwa teknologi dengan *platform* android dapat menarik minat pengguna *mobile phone* untuk beralih pada *smartphone* android (Fingas, 2013).

Adapun penelitian sistem pakar yang berhubungan dengan kehamilan yaitu “Program Bantu Diagnosa Gangguan Kesehatan Kehamilan Dengan Metode *Forward Chaining*” (Whenty H, 2010). Aplikasi yang dibangun berupa sistem pakar yang berfungsi untuk mendiagnosa gangguan kesehatan kehamilan. Dalam penelitian tersebut, gangguan kesehatan kehamilan yang dapat didiagnosa oleh sistem berdasarkan usia kandungan yaitu usia kandungan kurang dari dua puluh delapan minggu dan usia kandungan setelah dua puluh delapan minggu. *User* melakukan konsultasi layaknya berkonsultasi pada seorang pakar, kemudian menjawab gejala-gejala yang dirasakan dan sistem akan memberikan jenis gangguan apa yang terjadi dan memberikan solusi gangguan. *Output* yang dihasilkan pada sistem pakar tidak menyertakan tingkat persentase kebenaran dari proses pengolahan data. Dalam aplikasi sistem pakar tersebut terdapat lima belas jenis gangguan kehamilan beserta solusinya.

Sistem pakar memiliki penalaran yang cerdas yang diperoleh dari pengetahuan seorang pakar. Pada sistem pakar terdapat dua mekanisme inferensi yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu yaitu *forward chaining* dan *backward chaining*. Mekanisme inferensi dengan metode *forward chaining* atau penalaran maju merupakan metode yang menguji setiap aturan yang ada untuk menghasilkan suatu kesimpulan. Sebaliknya, jika ingin mendapatkan suatu kesimpulan tanpa harus menguji setiap aturan yang ada atau hanya tertarik pada satu kesimpulan tanpa ingin membuktikan fakta-fakta yang ada maka tidak cocok jika menggunakan metode *forward chaining*. Dengan metode *forward chaining*, akan menghasilkan kesimpulan dengan pasti dimana setiap fakta yang ada telah di uji.

Berdasarkan uraian yang telah dijelaskan, maka penulis akan melakukan penelitian dan membangun suatu aplikasi sistem pakar untuk mendiagnosa cacat lahir yang disebabkan oleh faktor teratogen (*eksternal*) dengan menggunakan metode *forward chaining* berbasis android. Aplikasi ini nantinya memberikan informasi mengenai cacat lahir dan penyebab cacat lahir sehingga dapat digunakan oleh pasangan subur yang akan merencanakan kehamilan agar bayi yang dikandungnya dapat terhindar dari bahan-bahan yang dapat menyebabkan bayi cacat lahir.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang ada adapun yang menjadi rumusan masalah adalah bagaimana merancang dan membangun aplikasi sistem pakar dengan menggunakan metode *forward chaining* untuk mendiagnosa cacat lahir yang disebabkan oleh faktor teratogen berbasis android?

## **1.3 Batasan Masalah**

Adapun batasan masalah pada penelitian ini adalah :

- a. Menggunakan metode *forward chaining* untuk penarikan kesimpulan.
- b. Cacat lahir dalam penelitian ini adalah cacat lahir yang tampak dengan kasat mata yang disebabkan oleh faktor teratogen (*eksternal*).
- c. Pengetahuan yang digunakan pada sistem ini berdasarkan pengetahuan pakar dan analisis pustaka.
- d. Sistem yang dibangun merupakan sistem berbasis *stand alone* pada sistem operasi android, sehingga tidak diperlukan akses internet dalam penggunaan sistem ini.

- e. Sistem pakar dikembangkan dengan menggunakan bahasa pemrograman Java pada editor IDE Eclipse Kepler.
- f. Menggunakan *Unified Modelling Language* (UML) sebagai model perancangan sistem yang berfungsi untuk memvisualisasi fungsi dan kinerja sistem yang dibangun.

#### **1.4 Tujuan Penelitian**

Tujuan yang ingin dicapai dari penelitian ini yaitu sebagai berikut :

- a. Terwujudnya rancangan aplikasi sistem pakar berbasis android untuk mendiagnosa cacat lahir yang disebabkan oleh faktor teratogen (*eksternal*).
- b. Menerapkan metode *forward chaining* pada sistem pakar untuk pengambilan keputusan.
- c. Memberikan informasi mengenai cacat lahir morfologis (cacat lahir bentuk) dan agen-agen teratogen yang dapat menyebabkan cacat lahir.

#### **1.5 Manfaat Penelitian**

Manfaat penelitian yang dapat diambil adalah :

- a. Menambah wawasan dan pengetahuan tentang cacat lahir dan agen-agen teratogen khususnya cacat lahir morfologis (bentuk).
- b. Dengan adanya aplikasi yang dibangun, dapat dijadikan sebagai media informasi bagi wanita, laki-laki ataupun pasangan subur yang akan merencanakan kehamilan.

## **BAB II**

### **PENDAHULUAN**

Pada bab ini membahas beberapa teori yang akan digunakan dalam penelitian yaitu sistem pakar, ilmu teratologi, cacat lahir, android, eclipse, *Unified Modeling Language* (UML), dan *Flowchart* (Diagram Alir).

#### **2.1 Sistem Pakar**

##### **2.1.1 Pengertian Sistem Pakar**

Sistem pakar merupakan pengembangan kecerdasan buatan atau *Artificial Intelligence* (AI) yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia. Tujuan pengembangan sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mensubstitusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak (Tim Penerbit ANDI, 2009).

Sistem pakar berusaha untuk menirukan metodologi dan kinerja dari seorang pakar. Seorang pakar adalah seseorang yang ahli dalam suatu bidang pengetahuan tertentu yang berarti bahwa seorang pakar memiliki pengetahuan atau kemampuan khusus yang tidak dimiliki oleh orang lain. Kekuatan sistem pakar terletak pada kemampuannya untuk memecahkan masalah-masalah praktis pada saat sang pakar berhalangan. Orientasi sistem pakar adalah konsultasi. Karena sistem pakar berfungsi sebagai konsultan, sehingga pemakai berkonsultasi dengan sistem pakar untuk mendapatkan nasihat. Bagi pemakai yang tidak berpengalaman sekalipun bisa memecahkan suatu masalah yang bagaimanapun rumitnya dan bisa mengambil keputusannya yang tepat dan akurat.

### 2.1.2 Ciri dan Karakteristik Sistem Pakar

Ciri dan karakteristik yang membedakan sistem pakar dengan sistem lainnya adalah sebagai berikut (Tim Penerbit ANDI, 2009):

- a. Pengetahuan sistem pakar merupakan suatu konsep, bukan numeris. Hal ini dikarenakan komputer melakukan proses pengolahan data secara numeris sedangkan keahlian dari seorang pakar adalah fakta dan aturan-aturan, bukan numerik.
- b. Informasi dalam sistem pakar tidak selalu lengkap, subyektif, dan tidak konsisten, subyek terus berubah tergantung pada kondisi lingkungan sehingga keputusan yang diambil bersifat tidak pasti dan tidak mutlak “ya” atau “tidak” akan tetapi menurut ukuran kebenaran tertentu.
- c. Kemungkinan solusi sistem pakar terhadap suatu permasalahan adalah bervariasi dan mempunyai banyak pilihan jawaban yang dapat diterima, semua faktor yang ditelusuri memiliki ruang masalah yang luas dan tidak pasti. Oleh karena itu, diperlukan fleksibilitas sistem dalam menangani kemungkinan solusi dari berbagai permasalahan.
- d. Perubahan atau pengembangan pengetahuan dalam sistem pakar dapat terjadi setiap saat bahkan sepanjang waktu sehingga diperlukan kemudahan dalam modifikasi sistem untuk menampung jumlah pengetahuan yang semakin besar dan bervariasi.
- e. Pandangan dan pendapat sistem pakar tidaklah selalu sama, oleh karena itu tidak ada jaminan bahwa solusi sistem pakar merupakan jawaban pasti benar. Setiap pakar akan memberikan pertimbangan berdasarkan faktor subyektif.

- f. Keputusan merupakan bagian terpenting dari sistem pakar. Sistem pakar harus memberikan solusi yang akurat berdasarkan masukan pengetahuan meskipun solusinya sulit, sehingga fasilitas informasi sistem harus selalu diperlukan.

### **2.1.3 Bidang-Bidang Pengembangan Sistem Pakar**

Ada banyak area atau wilayah yang menjadi daerah kerja *artificial intelligence* yaitu jaringan syaraf tiruan, sistem persepsi, robotika, bahasa ilmiah, sistem pendukung keputusan, sistem informasi berbasis manajemen dan sistem pakar. Ada beberapa kategori pengembangan sistem pakar yaitu (Tim Penerbit ANDI, 2009) :

- a. Kontrol.

Contoh pengembangan banyak ditemukan dalam kasus pasien rumah sakit, dimana dengan kemampuan sistem pakar dapat dilakukan kontrol terhadap pengobatan dan perawatan melalui sensor data atau kode alarm dan memberikan solusi terapi pengobatan yang tepat bagi si pasien yang sakit.

- b. Desain

Contoh sistem pakar dibidang ini adalah PEACE yang dibuat oleh Dincbas pada tahun 1980 untuk membantu desain pengembangan sirkuit elektronik.

- c. Diagnosis

Sistem pakar yang pernah diciptakan dibidang diagnosis, seperti diagnosis penyakit, diagnosis kerusakan kendaraan, diagnosis kerusakan kendaraan bermotor, dan lain-lain.

d. Prediksi

Sistem pakar ini mampu memprediksi kejadian masa mendatang berdasarkan informasi dan model permasalahan yang dihadapi. Program yang pernah dibuat adalah PLANT oleh Boulanger tahun 1983.

e. Monitor

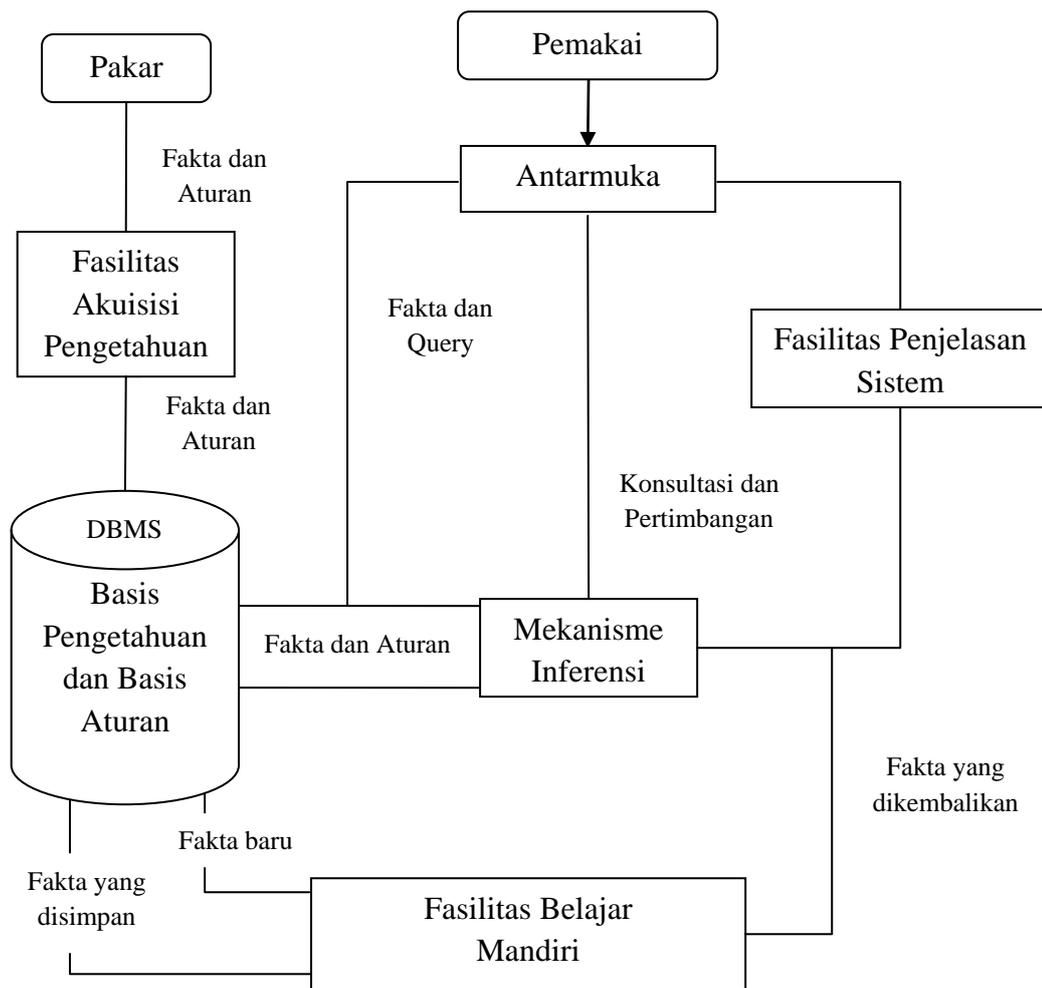
Sistem pakar dibidang ini banyak digunakan militer, yaitu menggunakan sensor radar kemudian menganalisisnya dan menentukan posisi objek berdasarkan radar tersebut.

f. Simulasi

Sistem ini memproses operasi dari beberapa variasi kondisi yang ada dan menampilkannya dalam bentuk simulasi. Contohnya adalah PLANT yang menggabungkan prediksi dan simulasi, dimana program tersebut mampu menganalisis hama dengan berbagai suhu dan cuaca.

#### **2.1.4 Struktur Sistem Pakar**

Komponen penting dalam sistem pakar adalah akuisisi pengetahuan, basis pengetahuan dan basis aturan, mesin inferensi, fasilitas penjelasan program, dan antarmuka pemakai yang merupakan satu kesatuan yang tidak dapat dipisahkan. Sedangkan fasilitas belajar mandiri (konsultasi) merupakan komponen yang mendukung sistem pakar sebagai suatu kecerdasan buatan tingkat lanjut (Tim Penerbit ANDI, 2009). Struktur sistem pakar dapat digambarkan pada Gambar 2.1.



Gambar 2.1 Struktur Bagan Sistem Pakar (Tim Penerbit ANDI, 2009)

Dari Gambar 2.1 ada lima unsur penting dari pengembangan sistem pakar yaitu fasilitas akuisisi pengetahuan, basis pengetahuan dan basis aturan, mekanisme inferensi, fasilitas penjelasan sistem, dan antarmuka pengguna yang merupakan satu kesatuan yang tidak dapat dipisahkan. Sedangkan fasilitas belajar mandiri merupakan komponen yang mendukung sistem pakar sebagai suatu kecerdasan buatan tingkat lanjut. Sistem pakar akan merepresentasikan kemampuan pakar yang ada ke dalam sistem kemudian sistem tersebut dapat digunakan pengguna berdasarkan basis pengetahuan yang ada. Berikut penjelasan dari komponen-komponen dari sistem pakar (Tim Penerbit ANDI, 2009).

### **a. Fasilitas Akuisisi Pengetahuan**

Fasilitas ini merupakan suatu proses untuk mengumpulkan data-data pengetahuan akan suatu masalah dari pakar. Bahan pengetahuan dapat diperoleh dengan beberapa cara, misalnya mendapatkan pengetahuan dari buku, jurnal ilmiah, para pakar dibidangnya, laporan, dan literatur. Sumber pengetahuan tersebut dijadikan dokumentasi untuk dipelajari, diolah, dan diorganisasikan secara terstruktur menjadi basis pengetahuan.

### **b. Basis Pengetahuan dan Basis Aturan**

Dari pengetahuan yang diperoleh, maka pengetahuan tersebut harus direpresentasikan menjadi basis pengetahuan dan basis aturan yang selanjutnya dikumpulkan, dikodekan, diorganisasikan dan digambarkan dalam bentuk rancangan lain menjadi bentuk yang sistematis. Beberapa cara merepresentasi data yaitu logika, pohon keputusan, jaringan semantik, *frame*, dan aturan-aturan.

Dalam pemrograman non-visual, basis aturan sering diimplementasikan dalam teknik IF THEN atau kondisi-aksi. IF kondisi (fakta) terjadi, THEN beberapa tindakan (hasil atau kesimpulan) akan terjadi. Pada umumnya, setiap aturan akan dikelompokkan dan diberi nama dengan "R". Untuk aturan pertama akan diberi nama R1, aturan kedua diberi nama R2, dan begitu seterusnya. Berikut ini adalah contoh dari satu kelompok aturan :

R1 = IF badan nyeri AND berat badan turun AND nyeri di perut bagian atas AND kulit berwarna kekuningan AND putih mata berwarna kekuningan THEN Adenokarsinoma Pankreas (Dewi, 2010) .

### c. Mekanisme Inferensi

Mekanisme inferensi adalah bagian dari sistem pakar yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi antar sistem dan pemakai, mekanisme inferensi menguji aturan satu demi satu sampai kondisi aturan itu benar.

Secara umum, ada dua teknik yang digunakan dalam mekanisme inferensi untuk pengujian aturan, yaitu pelacakan maju (*forward chaining*) dan pelacakan mundur (*backward chaining*) (Tim Penerbit ANDI, 2009).

#### 1) Metode Pelacakan Maju (*Forward Chaining*)

*Forward Chaining* atau pelacakan maju yaitu dimana aturan-aturan diuji satu demi satu dalam urutan tertentu yang berarti menggunakan aturan kondisi-aksi atau fakta-kesimpulan. Dalam aturan fakta-kesimpulan ini, fakta-fakta yang ada diuji kebenarannya untuk digunakan dalam menentukan kesimpulan apa yang ada. Saat tiap aturan di uji, sistem pakar akan mengevaluasi apakah kondisinya benar atau salah. Jika kondisi benar, maka aturan itu disimpan dan jika kondisinya salah, maka aturan itu tidak disimpan dan aturan berikutnya diuji. Berikut contoh metode *forward chaining* (Dewi, 2010):

Aturan 1:

IF Gejala A

AND Gejala B

AND Gejala C

THEN Penyakit 1

Aturan 3:

IF Gejala B

AND Gejala C

AND Gejala E

THEN Penyakit 3

Aturan 2:

IF Gejala A

AND Gejala C

AND Gejala D

THEN Penyakit 2

Aturan 4:

IF Gejala A

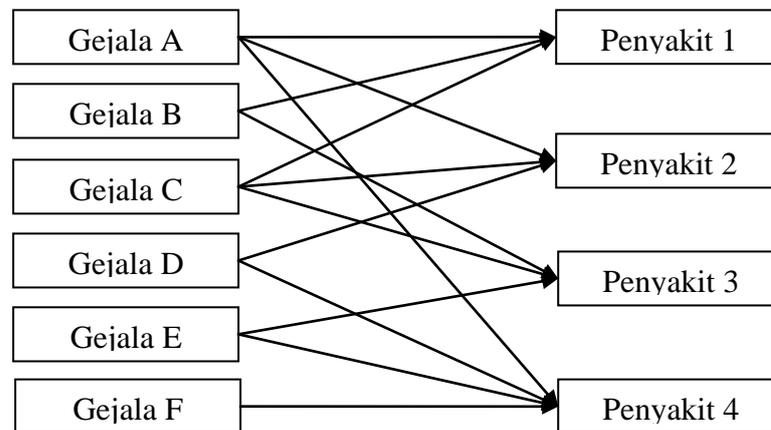
AND Gejala D

AND Gejala E

AND Gejala F

THEN Penyakit 4

Jika aturan ini digambarkan sebagai *graph* yang memetakan antara gejala dan penyakit akan tampak seperti pada Gambar 2.2 :



Gambar 2.2 *Grap* Pengetahuan (Dewi, 2010)

Metode *forward chaining* pada kasus ini adalah untuk mengetahui apakah suatu gejala yang dialami oleh *user* termasuk penyakit 1, penyakit 2, penyakit 3, penyakit 4 atau bahkan bukan salah satu dari kesimpulan tersebut, yang artinya sistem tersebut belum mampu mengambil kesimpulan karena keterbatasan aturan. Dalam penalaran ini, *user* diminta memasukkan gejala-gejala yang dialami. Berdasarkan gejala yang telah dimasukkan atau dipilih, maka sistem akan mencari aturan yang sesuai, sehingga akan diperoleh kesimpulannya.

Seandainya *user* memilih gejala A, gejala B dan gejala C maka aturan yang dipilih adalah aturan 1 dengan kesimpulannya adalah penyakit 1. Jika *user* memilih gejala A dan gejala F, maka sistem akan mengarah pada aturan 4 dengan kesimpulannya adalah penyakit 4, tetapi karena aturan tersebut gejalanya adalah gejala A, gejala D, gejala E dan gejala F, maka gejala-gejala yang dipilih oleh *user* tidak cukup untuk mengambil penyakit 4 sebagai kesimpulan terpilih.

## 2) Metode Pelacakan Mundur (*Backward Chaining*)

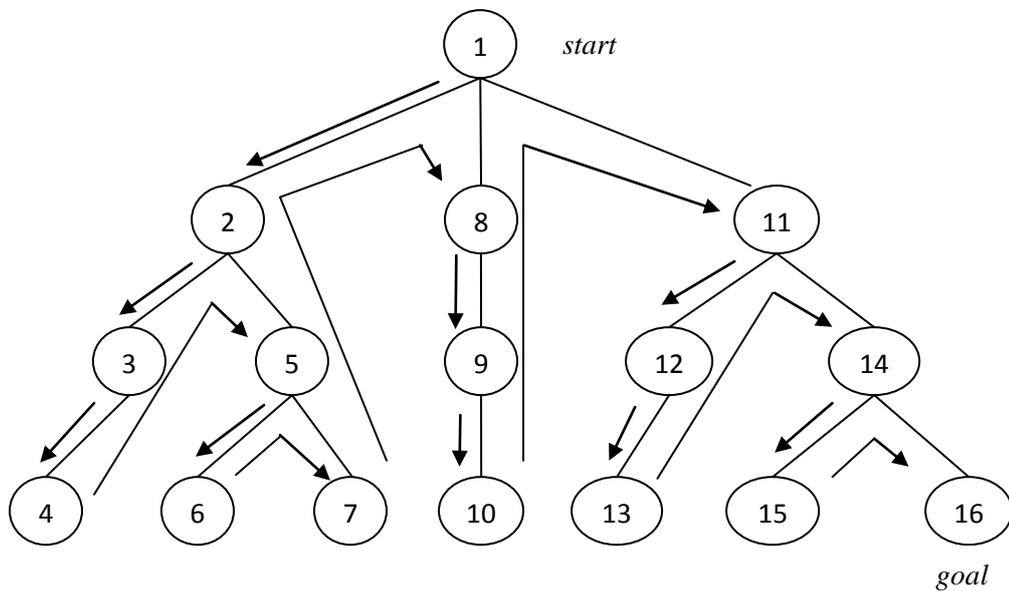
Metode *backward chaining* atau pelacakan mundur merupakan metode kebalikan dari *forward chaining* atau pelacakan maju. Dalam *backward chaining*, penalaran dimulai dari hipotesis terlebih dahulu, untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan.

Sebagai contoh dari kasus penyakit diatas, pengecekan penyakit tidak dimulai dari pengecekan gejala-gejala penyakit tetapi di mulai dengan hipotesis akhir. Dimana bahwa yang terjadi ialah penyakit 1 dan ingin dibuktikan bahwa penyakit 1 tersebut merupakan gejala dari gejala A, gejala B, dan gejala C. Oleh sebab itu, pelacakan akan dimulai dari hipotesis terlebih dahulu hingga kemudian sampai pada pembuktiannya.

Selain teknik yang digunakan dalam mekanisme inferensi untuk pengujian aturan, digunakan juga teknik penelusuran data dalam bentuk *tree* atau pohon. Ada tiga teknik yang digunakan dalam proses penelusuran data yaitu *depth first search*, *breadth first search*, dan *best first search* (Tim Penerbit ANDI, 2009).

### 1) *Depth-first search*

*Depth-first search* adalah teknik penelusuran data pada node-node secara vertikal dan sudah terdefiniskan. Pencarian dilakukan pada suatu simpul dalam setiap level dari yang paling kiri. Jika pada level terdalam belum ditemukan, maka pencarian dilanjutkan pada simpul sebelah kanan dan simpul yang kiri dapat dihapus dari memori. Pencarian metode *depth first search* digambarkan pada Gambar 2.3.



Gambar 2.3 *Depth First Search* (Desiani, 2006)

Dari Gambar 2.3 diperoleh pencarian DFS berikut :

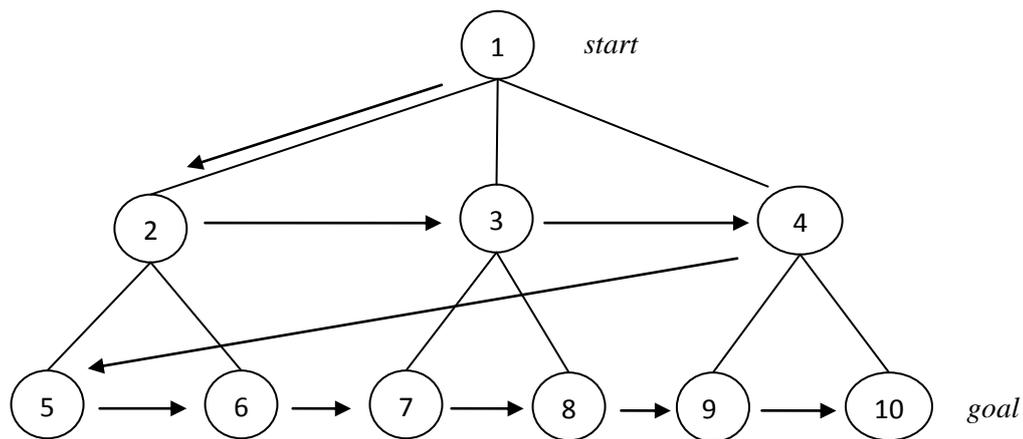
1. open = [1]; closed = []
2. open = [2,8,11]; closed = [1]
3. open = [3,5,8,11]; closed = [2,1]
4. open = [4,5,8,11]; closed = [3,2,1]
5. open = [5,8,11]; closed = [4,3,2,1]
6. open = [6,7,8,11]; closed = [5,4,3,2,1]
7. open = [7,8,11]; closed = [6,5,4,3,2,1]

8. open = [8,11]; closed = [7,6,5,4,3,2,1]
9. open = [9,11]; closed = [8,7,6,5,4,3,2,1]
10. open = [10,11]; closed = [9,8,7,6,5,4,3,2,1]
11. open [11]; closed = [10,9,8,7,6,5,4,3,2,1]
12. open = [12,14]; closed = [11,10,9,8,7,6,5,4,3,2,1]
13. open = [13,14]; closed = [12,11,10,9,8,7,6,5,4,3,2,1]
14. open = [14]; closed = [13,12,11,10,9,8,7,6,5,4,3,2,1]
15. open = [15,16]; closed = [14,13,12,11,10,9,8,7,6,5,4,3,2,1]
16. open = [16]; closed = [15,14,13,12,11,10,9,8,7,6,5,4,3,2,1]
17. open = []; closed = [16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1]

Kelebihan *depth first search* adalah pemakaian memori yang sedikit dan penelusuran masalah dapat digali secara mendalam sampai ditemukannya kepastian suatu solusi yang optimal. Selain itu, jika solusi yang dicari berada pada level yang dalam dan paling kiri, maka *depth first search* akan menemukannya dengan cepat. Sedangkan kelemahannya adalah memungkinkan tidak ditemukannya tujuan yang diharapkan dan hanya akan mendapatkan satu solusi pada setiap pencarian. Selain itu, penelusuran *depth first search* akan membutuhkan waktu yang sangat lama untuk ruang lingkup masalah yang besar (Suyanto, 2011).

## 2) *Breadth-First Search*

*Breadth-first search* adalah teknik penelusuran data pada semua node dalam satu level atau satu tingkatan sebelum ke level atau tingkatan dibawahnya (Suyanto, 2011). Pencarian *breadth-first search* diilustrasikan pada Gambar 2.4.



Gambar 2.4 *Breadth First Search* (Desiani, 2006)

Pencarian *breadth-first search* dilakukan pada semua simpul dalam setiap level secara berurutan dari kiri ke kanan.

Dari Gambar 2.4 diperoleh pencarian seperti berikut :

1. open = [1]; closed = [];
2. open = [2,3,4]; closed = [1]
3. open = [3,4,5,6]; closed = [2,1]
4. open = [4,5,6,7,8]; closed = [3,2,1]
5. open = [5,6,7,8,9,10]; closed = [4,3,2,1]
6. open = [6,7,8,9,10]; closed = [5,4,3,2,1]
7. open = [7,8,9,10]; closed = [6,5,4,3,2,1]
8. open = [8,9,10]; closed = [7,6,5,4,3,2,1]
9. open = [9,10]; closed = [8,7,6,5,4,3,2,1]
10. open = [10]; closed = [9,8,7,6,5,4,3,2,1]
11. open = []; closed = [10,9,8,7,6,5,4,3,2,1]

Kelebihan dari *breadth first search* diantaranya tidak akan menemui jalan buntu, karena dimana semua node akan dicek secara menyeluruh pada

setiap tingkatan dan jika ada satu solusi maka *breadth first search* akan menemukannya, dan jika ada lebih dari satu solusi maka solusi minimum akan ditemukan. Sedangkan kelemahan dari *breadth first search* adalah dibutuhkan waktu yang sangat lama apabila solusi berada pada dalam posisi node terakhir, sehingga menjadi tidak efisien. Kekurangan lainnya yaitu teknik penelusuran menjadi tidak efektif antara pemakai dan sistem karena menyebabkan tidak adanya relasi antara suatu topik dengan topik lainnya (Tim Penerbit ANDI, 2009).

### 3) *Best-First Search*

Penelusuran *best-first search* adalah penelusuran yang menggunakan pengetahuan akan suatu masalah untuk melakukan panduan pencarian ke arah node tempat dimana solusi berada. Pendekatan yang dilakukan adalah mencari solusi yang terbaik berdasarkan pengetahuan yang dimiliki sehingga penelusuran dapat ditentukan harus mulai dari mana dan bagaimana menggunakan proses terbaik untuk mencari solusi. Ini merupakan model yang menyerupai cara manusia mengambil solusi, hanya saja solusi yang diambil bisa saja salah dan bukan solusi yang mutlak benar.

#### **d. Fasilitas Penjelasan Sistem**

Fasilitas penjelasan sistem merupakan bagian dari sistem pakar yang memberikan penjelasan tentang bagaimana program dijalankan, apa yang harus dijelaskan kepada pemakai tentang suatu masalah, memberikan rekomendasi kepada pemakai, mengakomodasi kesalahan pemakai dan menjelaskan bagaimana suatu masalah terjadi.

Dalam sistem pakar, fasilitas penjelasan sistem sebaiknya diintegrasikan kedalam tabel basis pengetahuan dan basis aturan karena hal ini lebih memudahkan dalam perancangan sistem.

#### **e. Antarmuka Pemakai**

Antarmuka pemakai memberikan fasilitas komunikasi antara pemakai dan sistem, memberikan berbagai fasilitas informasi dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi. Pada umumnya, antarmuka pemakai juga berfungsi untuk menginputkan pengetahuan baru ke dalam basis pengetahuan sistem pakar, menampilkan fasilitas penjelasan sistem dan memberikan tuntunan pengguna sistem secara menyeluruh langkah demi langkah sehingga pemakai mengerti apa yang harus dilakukan terhadap sistem.

Syarat utama membangun antarmuka pemakai adalah kemudahan dalam menjalankan sistem. Semua kesulitan dalam membangun suatu program harus disembunyikan, yang ditampilkan hanyalah tampilan yang interaktif, komunikatif, dan kemudahan pakai.

### **2.1.5 Kelebihan dan Kekurangan Sistem Pakar**

Kekuatan sistem pakar terletak pada kemampuannya memecahkan masalah-masalah praktis pada saat sang pakar berhalangan (Tim Penerbit ANDI, 2009).

Beberapa kelebihan dari sistem pakar, antara lain:

- a. Masyarakat awam non-pakar dapat memanfaatkan keahlian di dalam bidang tertentu tanpa kehadiran langsung seorang pakar.

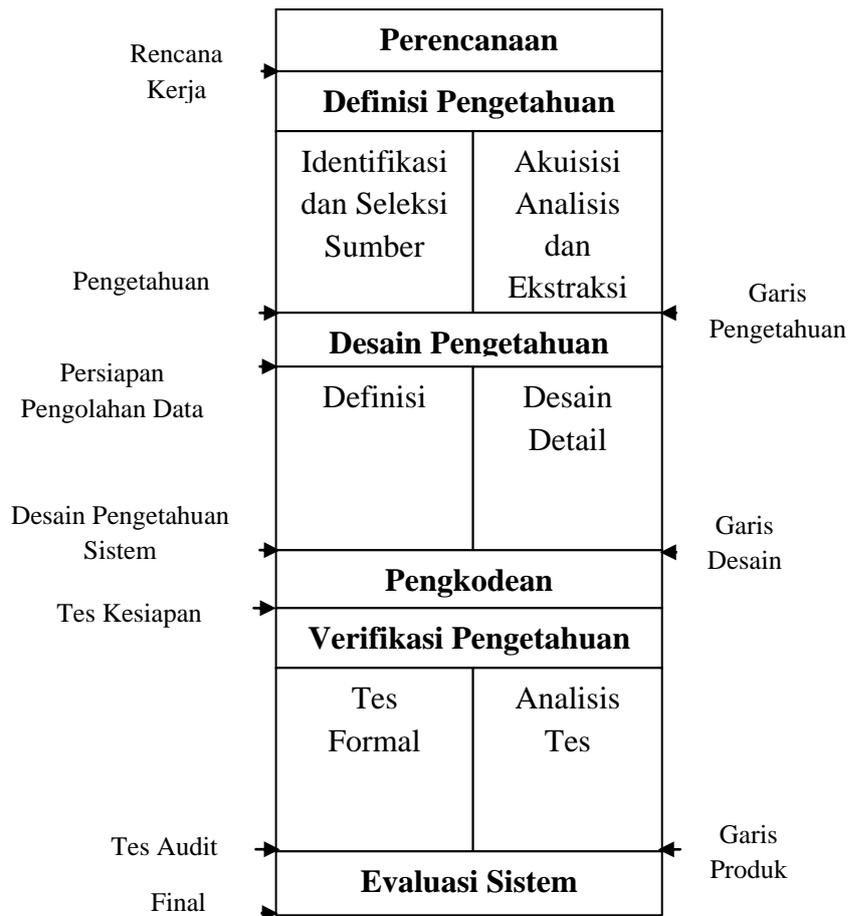
- b. Sistem pakar dapat digunakan dimanapun dan kapanpun.
- c. Kecepatan dalam memberikan solusi dan pengetahuan yang konsisten.
- d. Sistem dapat digunakan berulang-ulang oleh siapa saja, sehingga dapat meminimalisir biaya yang dikeluarkan.
- e. Pengetahuan dari seorang pakar dapat didokumentasikan tanpa ada batas waktu.

Selain memiliki kelebihan, sistem pakar juga memiliki kekurangan yaitu:

- a. Dengan adanya sistem pakar, daya kerja dan produktivitas manusia menjadi berkurang karena dilakukan oleh sistem.
- b. Solusi yang diberikan oleh sistem pakar tidak bernilai 100%, karena ilmu pengetahuan yang selalu mengalami perkembangan dan perubahan.

#### **2.1.6 Metode Pengembangan Sistem Pakar**

Metode pengembangan *Linear Model of Expert System Development* merupakan model pengembangan yang dikhususkan untuk pengembangan sistem pakar. Model ini memberikan arahan yang sangat akurat dalam pengembangan sistem pakar (Nurhayati, 2006). Secara garis besar, model ini membagi proses pengembangan sistem pakar menjadi enam tahap, yaitu perencanaan, definisi pengetahuan, desain pengetahuan atau representasi pengetahuan, pengkodean, verifikasi pengetahuan, dan evaluasi sistem. Keenam tahap pengembangan sistem pakar digambarkan pada Gambar 2.5 .



Gambar 2.5 *Linear Model of Expert System Development*  
(Nurhayati, 2006)

Berikut penjelasan metode pengembangan sistem berdasarkan Gambar 2.5.

**a. Perencanaan**

Tahap ini merupakan tahap awal yang menjadi dasar pengembangan sistem pakar. Dalam tahap ini untuk penentuan hal-hal penting sebagai dasar dari permasalahan yang akan dianalisis. Tahap ini untuk mengkaji dan membatasi masalah yang akan diimplementasikan dalam sistem. Adapun proses dalam tahap ini yaitu proses penilaian kelayakan, spesifikasi kebutuhan, dan penggambaran fungsi awal sistem.

## **b. Definisi Pengetahuan**

Tahap definisi pengetahuan terdiri dua kegiatan pokok, yaitu :

### 1) Proses identifikasi dan seleksi sumber pengetahuan

Diawali dengan identifikasi sumber, yaitu menentukan siapa dan apa sumber pengetahuan yang diperlukan dalam pembangunan sistem pakar, prioritas sumber yaitu menentukan sumber pengetahuan dalam urutan prioritas, ketersediaan sumber, yaitu membuat daftar sumber pengetahuan diurutkan berdasarkan ketersediaannya, dan pemilihan sumbernya.

### 2) Proses akuisisi, analisis dan ekstraksi pengetahuan.

Proses ini dilakukan setelah diperoleh fokus pengetahuan yang akan dikembangkan menjadi sistem pakar. Proses ini meliputi beberapa tahapan, yaitu strategi perolehan untuk menentukan bagaimana pengetahuan akan diperoleh, identifikasi pengetahuan dasar, yaitu memperoleh pengetahuan yang spesifik dari sumber yang akan digunakan, dan sistem klasifikasi pengetahuan, yaitu mengklasifikasikan pengetahuan yang akan membantu verifikasi dan pemahaman dalam mengembangkan sistem sesuai dengan kebutuhan sistem.

## **c. Desain Pengetahuan**

Tahap desain pengetahuan terbagi menjadi dua proses, yaitu proses definisi dan desain detail. Proses definisi meliputi representasi pengetahuan dan perancangan aturan. Sedangkan proses desain meliputi rancangan struktur kendali sistem yaitu dan perancangan *user interface*.

#### **d. Pengkodean**

Tahap pengkodean meliputi proses pembuatan baris program (*coding*) dan melakukan berbagai prosedur didalamnya.

#### **e. Verifikasi Pengetahuan**

Dalam tahap implementasi ini, desain pengetahuan dan pengkodean sudah diformulasikan secara lengkap, proses inputan, instalasi, demonstrasi penerapan sistem, integrasi dan pengujian kasus. Akan lebih menjelaskan lagi sebatas mana kemampuan sistem pakar yang dibangun dan siap untuk diaudit jika diperlukan.

#### **f. Evaluasi Sistem**

Tahap evaluasi sistem merupakan tahapan pengujian akhir dan validasi sebelum sistem benar-benar digunakan. Pada tahap ini juga dimungkinkan adanya rekomendasi, sehingga terjadi perubahan pada sistem apabila ada perubahan dan penambahan data yang digunakan.

## **2.2 Cacat Lahir**

### **2.2.1 Pengertian Cacat Lahir**

Cacat lahir adalah abnormalitas struktur, fungsi atau metabolisme (kimiawi tubuh), gangguan mental, atau kematian. Cacat lahir merupakan suatu keadaan cacat pada bayi yang baru lahir dan penyebab utama kematian pada tahun pertama kehidupan bayi. Cacat lahir ada dua macam yaitu anomali kongenital dan malformasi kongenital.

Anomali kongenital adalah cacat fisik maupun non fisik. Sedangkan malformasi kongenital adalah cacat fisik saja. Malformasi kongenital dimana kelainan dalam pertumbuhan janin yang terjadi sejak konsepsi dan selama dalam kandungan. Diperkirakan 10-20% dari kematian janin dalam kandungan dan kematian neonatal disebabkan oleh kelainan kongenital. Malformasi kongenital merupakan sebab penting terjadinya abortus, lahir mati atau kematian segera setelah lahir. Kematian bayi dalam bulan-bulan pertama kehidupannya sering diakibatkan oleh kelainan kongenital yang cukup berat, hal ini seakan-akan merupakan suatu seleksi alam terhadap kelangsungan hidup bayi yang dilahirkan (Mochtar, 1998).

### **2.2.2 Penyebab Cacat Lahir**

Adapun penyebab cacat lahir terdiri dari dua faktor yaitu faktor genetika dan faktor lingkungan. Berikut penjelasan faktor penyebab cacat lahir:

#### **a. Faktor Genetik**

Faktor informasi genetik, yaitu faktor yang diturunkan oleh orang tua melalui ovum atau sperma. Kelainan faktor genetik dikelompokkan ke dalam kelainan/kerusakan gen, kerusakan kromosom, dan pewarisan multifaktor. Contoh kelainan kongenital yang dikarenakan faktor genetik adalah sindrome down adalah cacat lahir pada bayi yang disebabkan oleh kelebihan kromosom 21 pada sel-selnya, penderita memiliki 47 kromosom, mereka mengalami keterbelakangan mental dan mempunyai tubuh yang pendek dan puntung, lengan atau kaki kadang-kadang bengkok, kepala lebar, wajah membulat, mulut selalu terbuka, ujung lidah besar dan datar, kedua lubang hidung terpisah lebar (Nadya, 2013).

## **b. Faktor Non-Genetik**

Faktor non-genetik yaitu gangguan metabolisme dan faktor lingkungan. Gangguan metabolisme seperti ibu hamil yang mengalami kekurangan vitamin B<sub>6</sub> (*pyridoxin*) dapat menimbulkan bibir sumbing pada bayinya. Faktor lingkungan yaitu faktor yang berhubungan dengan lingkungan yang menyebabkan cacat bawaan seperti obat-obatan, bahan kimia yang bersifat mutagenik atau teratogenik, malnutrisi dan trauma mekanik (Nadya, 2013).

## **2.3 Ilmu Teratologi**

### **2.3.1 Pengertian Ilmu Teratologi**

Teratologi adalah ilmu pengetahuan yang mengkaji tentang perkembangan anomali janin dan cacat lahir akibat substansi lingkungan. Teratologi merupakan cabang dari ilmu embriologi yang khusus mempelajari tentang akibat, mekanisme dan manifestasi embrionik yang cacat (abnormal).

Adapun prinsip-prinsip teratologi adalah sebagai berikut (Wilson & Philadelphia, 2007):

- a. Kerentanan terhadap teratogenesis tergantung pada genotip konseptus dan cara komposisi genetik berinteraksi dengan lingkungan.
- b. Kerentanan terhadap teratogen berbeda-beda menurut stadium perkembangannya saat paparan. Masa yang paling sensitif untuk timbulnya cacat lahir adalah masa kehamilan minggu ketiga hingga kedelapan, dimana masa tersebut adalah masa embriogenesis. Masing-masing sistem organ tubuh mungkin mempunyai satu atau beberapa stadium kerentanan. Selanjutnya, meskipun kebanyakan kelainan terjadi

selama embriogenesis, cacat bisa juga terjadi sebelum atau sesudah masa ini, sehingga tidak ada satu masa yang benar-benar aman.

- c. Teratogen bekerja dengan cara (mekanisme) yang spesifik pada sel-sel dan jaringan yang sedang berkembang. Setiap obat atau bahan kimia yang diberikan kepada ibu akan menembus plasenta sampai batas tertentu kecuali dihancurkan atau diubah selama pembentukan plasenta. Proses pembentukan plasenta dimulai pada minggu kelima atau embrio ketujuh minggu kehamilan. Untuk obat-obatan atau bahan kimia dengan berat molekul rendah, transmisi dari plasenta ke janin didasarkan pada gradien konsentrasi.
- d. Manifestasi perkembangan abnormal tergantung pada dosis atau lamanya paparan terhadap suatu teratogen. Manifestasi perkembangan abnormal adalah kematian, malformasi, keterlambatan pertumbuhan, dan gangguan fungsi.

### **2.3.2 Teratogen**

Teratogen adalah sesuatu atau agensia lingkungan penyebab munculnya cacat lahir yang dapat mempengaruhi perkembangan normal embriologi atau janin. Potensi teratogenik pada beberapa teratogen tergantung pada dosis dan tahap perkembangan janin pada saat terjadi paparan. Faktor-faktor ini mempengaruhi perkembangan janin (embrio) yang terjadi pada tahap embriogenesis. Hal ini terjadi saat proses organogenesis melalui mekanisme mutasi pada sel-sel somatik jaringan embrio, sehingga dapat menyebabkan kelainan kongenital.

Adapun teratogen penyebab kelainan kongenital adalah sebagai berikut:

- a. Agen infeksi, yaitu agen teratogen yang disebabkan oleh virus seperti rubella, cytomegalovirus, virus herpes simpleks, HIV, dan sifilis.
- b. Agen fisik, yaitu agen teratogen yang disebabkan oleh benda-benda fisik ataupun sentuhan fisik seperti sinar X, radiasi nuklir, dan hipertermia.
- c. Agen kimia yaitu agen teratogen yang disebabkan oleh obat dan kimia seperti phenytoin, thalidomide, alkohol, merkuri, kokain, dan lainnya. Setiap obat yang dikonsumsi oleh ibu hamil, bergantung pada dosis dan periode kehamilan.
- d. Penyakit ibu yaitu agen teratogen yang disebabkan oleh penyakit yang diidap oleh ibu hamil. Adapun penyakit ibu yang menjadi teratogen yaitu penyakit diabetes melitus.

## **2.4 Android**

### **2.4.1 Sejarah Android**

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi sendiri. Awalnya, Google Inc membeli Android Inc yang merupakan pendatang baru yang membuat *software* untuk ponsel. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance, konsorsium dari tiga puluh empat perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Safaat, 2011).

Pada saat perilisan perdana 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. September 2007, Google mengenalkan Nexus One sebagai salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon seluler ini di produksi oleh *HTC Corporation* pada 5 Januari 2008. Pada masa saat ini, *smartphone* berbasis android sudah banyak di gunakan oleh vendor-vendor *smartphone* antara lain HTC, Motorola, Samsung, LG, Huawei, Sony Ericsson, Acer, Philips, Nexian, IMO, dan Lenovo. Hal ini dikarenakan android merupakan sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun (Safaat, 2011).

#### **2.4.2 Fitur Android**

Sebagai *platform* aplikasi netral, Android memberikan kesempatan untuk membuat aplikasi baru atau yang kita butuhkan yang bukan merupakan aplikasi bawaan *handphone/smarthphone*.

Beberapa fitur-fitur android yang paling penting adalah (Safaat, 2011):

a. *Framework*

Fitur aplikasi yang mendukung penggantian komponen dan *reusable*.

b. *Virtual Dalvik*

Mesin yang dioptimalkan untuk perangkat *mobile*.

b. *Integrated browser*

*Browser web* berdasarkan *engine open source WebKit*.

c. *Handset Layout*

Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan OpenGL ES 1.0 .

d. SQLite

Untuk penyimpanan data (*database*).

e. *Media Support*

Yang mendukung audio, video, dan gambar (MPEG<sub>4</sub>, H.264 (MP4 dalam 3GP), MP<sub>3</sub>, AAC, JPG, PNG, GIF), GSM Telephony (tergantung hardware).

f. *Additional hardware support* (Dukungan hardware tambahan)

Android menyediakan/dapat menggunakan kamera, GPS, kompas, dan *accelerometer* (tergantung kamera).

g. *Connectivity*

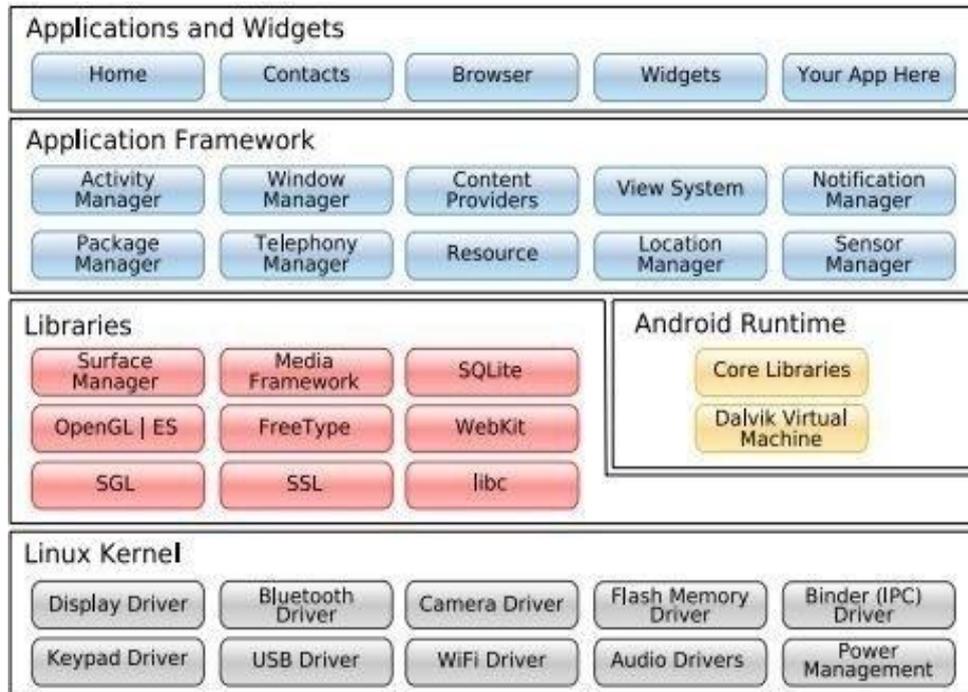
Android mendukung teknologi konektivitas termasuk GSM / EDGE, CDMA, EV-DO, UMTS, *Bluetooth*, dan Wi-Fi.

h. *Development environment*

Android memiliki *Development environment* yang lengkap dan kaya termasuk perangkat emulator, *tools* untuk *debugging*, kinerja memori, dan *plugin* untuk Eclipse IDE.

### 2.4.3 Arsitektur Android

Arsitektur android yaitu bagian-bagian yang terdapat pada android yang terdiri dari lima bagian. Secara garis besar arsitektur android pada gambar 2.6 (Safaat, 2011).



Gambar 2.6. Arsitektur Android (Safaat, 2011)

Berikut penjelasan arsitektur android :

a. *Applications dan Widgets*

*Application* dan *Widgets* ini adalah *layer* dimana berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut, di *layer* inilah terdapat aplikasi inti termasuk klien *email*, program SMS, kalender, peta, *browser*, kontak dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b. *Applications Frameworks*

*Applications frameworks* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem android, karena pada *layer* inilah aplikasi dapat dirancang, seperti *content-providers* yang berupa SMS dan lain sebagainya.

c. *Libraries*

*Libraries* ini adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi kebanyakan mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *layer* ini meliputi berbagai *library* C/C++ inti seperti Libc dan SSL, serta:

- 1) *Libraries* media untuk pemutaran media audio dan video.
- 2) *Libraries* untuk manajemen tampilan.
- 3) *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
- 4) *Libraries SQLite* untuk dukungan *database*.
- 5) *Libraries SSL* dan WebKit terintegrasi dengan *web browser* dan *security*.
- 6) *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embedded web view*.
- 7) *Libraries 3D* yang mencakup implementasi OpenGL ES 1.0 API's

d. *Android Run Time*

*Layer* yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Didalam *Android Run Time* dibagi menjadi dua bagian yaitu:

- 1) *Core Libraries*. Aplikasi Android yang dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan virtual mesin Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java yang dihandle oleh *core libraries*.

2) *Dalvik Virtual Machine*. Virtual mesin yang berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat linux kernel untuk threading dan manajemen tingkat rendah.

e. Linux Kernel

Linux Kernel adalah *layer* dimana inti dari sistem operasi Android itu sendiri, berisi file-file sistem yang mengatur *system processing, memory, resource, drivers*, dan sistem-sistem operasi Android lainnya. Linux Kernel yang digunakan Android adalah Kernel Linux *realase* 2.6.

#### 2.4.4 Komponen Android

Ada empat jenis komponen penting yang harus ada pada aplikasi android, yaitu (Safaat, 2011) :

a. *Activities*

Suatu *activity* akan menyajikan *user interface* (UI). *User interface* (UI) ini nantinya berfungsi untuk pengguna sehingga 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. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* yang lain kita dapat melakukan dengan satu even misalnya klik tombol, memilih opsi atau menggunakan *triggers* tertentu.

*b. Service*

*Service* tidak memiliki *visual user interface* (UI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* haruslah berada dalam kelas induknya. Misalnya *media player* sedang memutar lagu dari *list* yang ada, aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu atau menulis SMS sambil *player* sedang jalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service* untuk membuat aplikasi tetap berjalan. *Service* dijalankan pada *thread* utama dari proses aplikasi.

*c. Broadcast Receiver*

*Broadcast Receiver* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. *Broadcast Receiver* tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya. Contoh *broadcast* seperti notifikasi zona waktu berubah dan baterai *low*.

*d. Content Provider*

*Content provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file sistem* seperti *database SQLite*. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*.

#### 2.4.5 Versi Android

Versi-versi android yang telah dirilis adalah sebagai berikut (Safaat, 2011):

a. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search*, pengiriman pesan dengan *Gmail*, dan pemberitahuan *email*.

b. Android versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, , dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

c. Android versi 1.6 (*Donut*)

*Donut* (versi 1.6) dirilis pada September 2009 dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan galeri yang dintegrasikan, CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*, kemampuan dial kontak, teknologi *text to change speech* (tidak tersedia pada semua *handphone*) dan pengadaan resolusi VWGA.

d. Android versi 2.0/2.1 (*Éclair*)

Pada 3 Desember 2009 kembali diluncurkan *smartphone* Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan *Bluetooth* 2.1.

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

Pada bulan Mei 2010 di luncurkan Android versi 2.2 (*Froyo: Frozen Yoghurt*) dengan 2 kali revisi, revisi pertama bulan Mei 2010 dan revisi kedua bulan Juli 2010. Android 2.2 adalah rilis minor *platform* termasuk fitur pengguna, fitur pengembang, perubahan API, dan perbaikan *bug*.

f. Android versi 2.3 (*Gingerbread*)

Android versi 2.3 diluncurkan pada desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan sebagai berikut:

1) *Near Field Communication* (NFC)

2) *Multiple cameras support*

3) *Mixable Audio effect*

4) *Download Manager*

g. Android versi 3.0-3.26 (*Honeycomb*)

Android *Honeycomb* diluncurkan pada bulan Maret 2011. Tidak seperti versi android lainnya, android *honeycomb* ini hanya tersedia pada perangkat tablet pc saja. Alasan tidak dapat diterapkan pada *smartphone* karena android *honeycomb* memerlukan spesifikasi yang lebih tinggi dari kemampuan *smartphone*, seperti kebutuhan *processor* dan GPU-nya.

h. Android versi 4.0-4.0.4 (*Ice Cream Sandwice*)

Android *Ice Cream Sandwice* dirilis pada 19 Oktober 2011. Terdapat peningkatan fitur seperti, pesan suara visual, penambahan jenis huruf roboto, mendukung *flash player*, dan meningkatkan stabilitas video dan resolusi QVGA.

i. Android versi 4.1-4.3 (*Jelly Bean*)

Versi android *Jelly Bean* pertama kali diluncurkan pada Juli 2012. *Jelly Bean* adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna (UI).

j. Android versi 4.4 (*Kitkat*)

Versi terbaru dari android adalah *Kitkat* yang diluncurkan pada 31 Oktober 2013. Pembaruan antarmuka dengan bar status dan navigasi transparan pada layer depan. Adanya fitur inframerah dan dukungan *Bluetooth Message Access Profile* (MAP).

#### **2.4.6 Kelebihan dan Kekurangan Android**

Beberapa kelebihan android antara lain :

1. *Open source*, yaitu *platform* terbuka dimana pengembang dapat menciptakan aplikasi sendiri, sehingga aplikasi tidak terbatas.
2. Android tersedia di ponsel dari berbagai produsen seperti dari Sony Ericsson, Motorola, HTC dan Samsung. Setiap pabrikan ponsel pun menghadirkan ponsel android dengan gaya masing-masing, sehingga konsumen dapat memilih ponsel android sesuai dengan merek yang favorit dan dengan harga yang terjangkau.

Adapun kelemahan android yaitu adanya iklan atau *notification* iklan. Beberapa aplikasi di android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap aplikasi tersebut akan selalu ada iklan yang terpampang. Selain itu konsumen akan sulit membedakan produk dari berbagai vendor, karena perangkat android yang telah diumumkan terlihat sangat mirip.

## **2.5 Eclipse**

### **2.5.1 Pengertian Eclipse**

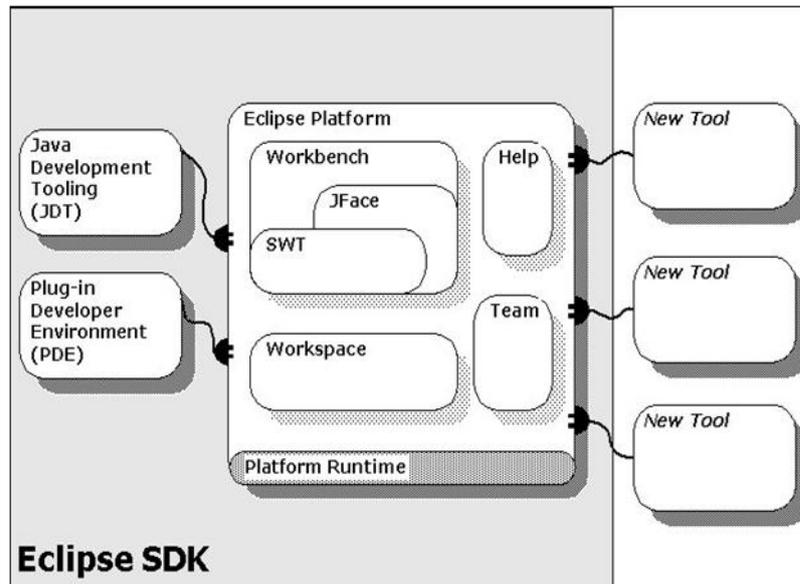
Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*). Berikut ini adalah ciri-ciri dari Eclipse :

1. *Multi-platform* artinya target sistem operasi Eclipse yaitu Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. *Mult-language* artinya Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi eclipse mendukung pengembangan aplikasi bahasa pemrograman lainnya seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role* artinya selain sebagai IDE untuk pengembangan aplikasi, eclipse bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti tes perangkat lunak, dan pengembangan web.

Eclipse pada saat ini merupakan salah satu IDE yang banyak diminati dikarenakan gratis dan *open source* yang berarti setiap orang bisa mengembangkan perangkat lunaknya sendiri dan boleh melihat kode pemrograman perangkat lunak. Dengan demikian aplikasi-aplikasi berbasis android tidak terbatas (bertambah) dan selalu mengalami perubahan.

## 2.5.2 Arsitektur Eclipse

Pada tingkat yang paling dasar dalam arsitektur eclipse adalah eclipse *platform*. Fungsi utama dari eclipse *platform* adalah untuk menyediakan layanan yang diperlukan untuk mengintegrasikan kakas-kakas pengembangan dalam hal ini sebagai editor pengembangan. Arsitektur eclipse terdapat pada Gambar 2.7.



Gambar 2.7 Arsitektur Eclipse (Nugroho, 2007 dalam Yutama, 2013)

Berikut penjelasan arsitektur/komponen penting dari eclipse :

### a. *Workspace*

*Workspace* adalah bagian dari eclipse yang bertanggung jawab dalam pengelolaan sumber daya yang dimiliki pengguna, terorganisasi ke dalam satu atau lebih proyek pada peringkat paling atas.

### b. *Workbench*

*Workbench* merupakan antarmuka pengguna berbasis grafis. Selain bermanfaat untuk menampilkan menu dan *toolbar* yang *familiar*, *workbench* juga mengorganisasikan aneka perspektif yang mengandung tampilan (*view*) dan layar penyuntingan (*editor*) termasuk SWT dan JFace.

c. *Team Support*

*Plug-in* yang ada dalam Eclipse pada dasarnya member dukungan terhadap kerja dan kinerja dukungan tim (*team support*) dan memfasilitasi pengelolaan konfigurasi sistem untuk mengelola berbagai sumber daya dalam pengguna dan mendefinisikan aliran kerja (*workflow*) untuk menyimpan serta memanggil berkas dari tempat penyimpanan terpusat.

d. *Help*

Seperti *eclipse platform* itu sendiri, komponen bantuan terhadap pemrogram merupakan dokumen yang dapat diperluas. Para penyedia kakas yang akan dimanfaatkan oleh para pengguna Eclipse dapat menambahkan dokumen bantuan dalam format HTML.

## **2.6 *Unified Modeling Language (UML)***

### **2.6.1 Pengertian UML**

*Unified Modeling Language* atau singkatan dari UML yang berarti bahasa pemodelan standard. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. UML telah diaplikasi dalam bidang investasi perbankan, lembaga kesehatan, departemen pertahanan, dan *supplier*.

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk (Widodo & Herlawati, 2011):

- a. Merancang Perangkat Lunak
- b. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
- c. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
- d. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangun utama UML adalah diagram. Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun, dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota tim untuk bekerja sama dengan bahasa model yang sama dalam mengaplikasikan beragam sistem. Intinya UML merupakan alat komunikasi yang konsisten dan *support* para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti akan menjumpai UML, baik kita sendiri yang membuat atau sekedar membaca diagram UML buatan orang lain (Pilone, 2005 dalam Widodo, 2011).

### **2.6.2 Diagram-Diagram UML**

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan diagram karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Berikut penjelasan diagram-diagram UML (Widodo & Herlawati, 2011):

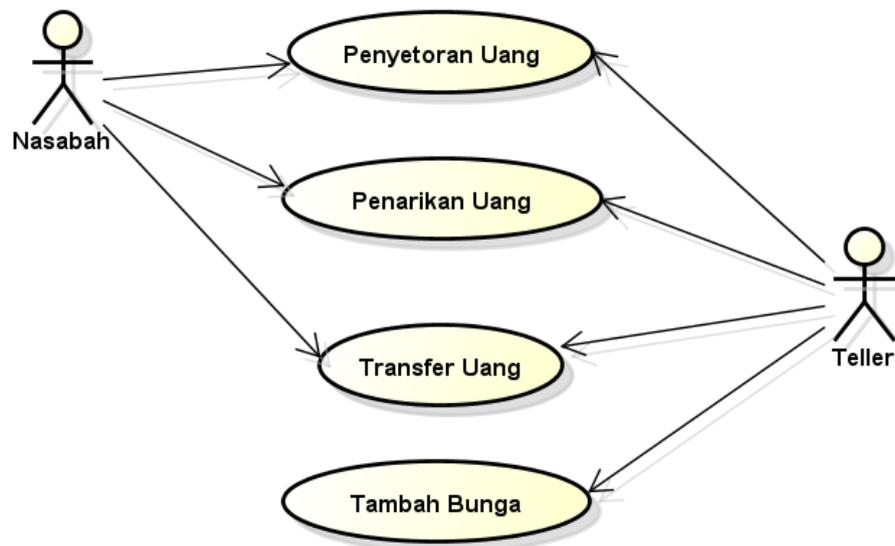
#### ***a. Use Case Diagram***

Diagram *use case* bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor. *Use case* mengorganisasikan, memodelkan perilaku suatu sistem, dan menggambarkan *external view* dari sistem yang akan dibuat modelnya. Model *use case* dapat dijabarkan dalam diagram *use case*, tetapi yang perlu diingat diagram tidak identik dengan model karena model lebih luas dari diagram (Pooley, 2003 dalam Widodo & Herlawati, 2011).

Komponen pembentuk diagram *use case* adalah:

- 1) Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- 2) Use case, aktivitas/sarana yang disiapkan oleh bisnis/sistem.
- 3) Hubungan (*link*), aktor mana saja yang terlibat dalam use case ini.

Contoh *use case diagram* pada Gambar 2.8 .

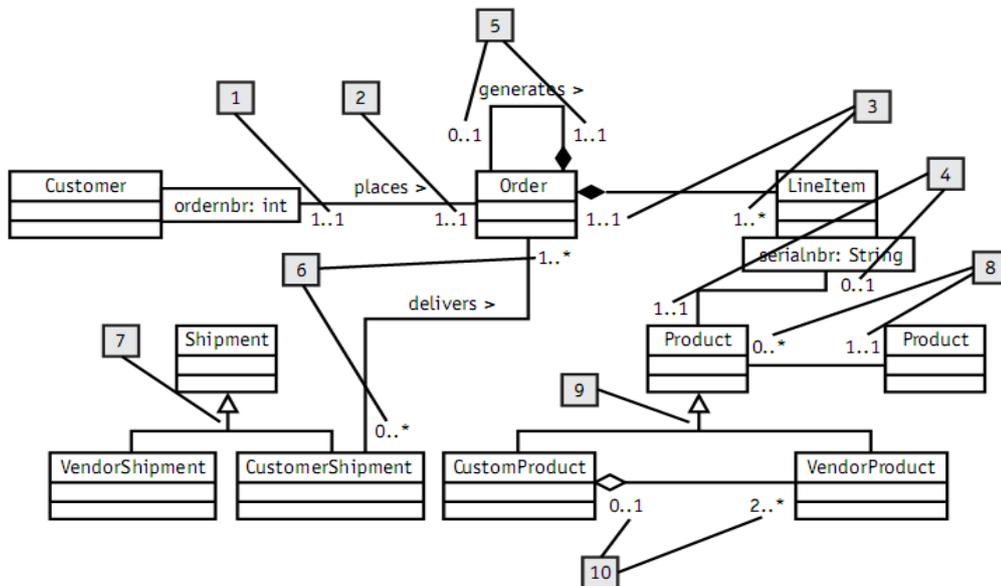


Gambar 2.8 Contoh *Use Case Diagram* (Widodo & Herlawati, 2011)

### **b. Class Diagram**

*Class diagram* memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi yang ada pada sistem. *Class* memiliki tiga area pokok yaitu nama (*stereotype*), atribut, dan metoda. Contoh *class diagram* terdapat pada Gambar 2.9.

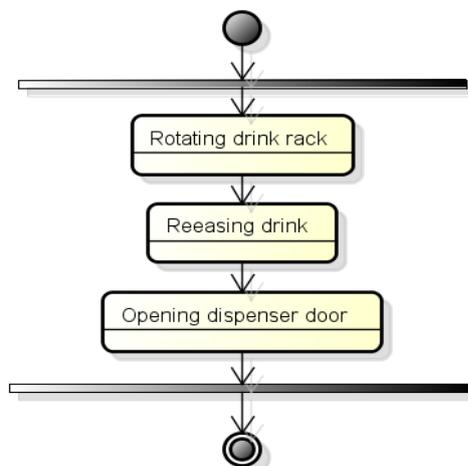
- a. Sudah teruji dengan baik pada sistem operasi Windows serta Linux.
- b. Fasilitas *commit* dan *rollback* perlindungan integritas data yang baik.



Gambar 2.9 Contoh Class Diagram (Pender, 2002)

c. **Statechart Diagram**

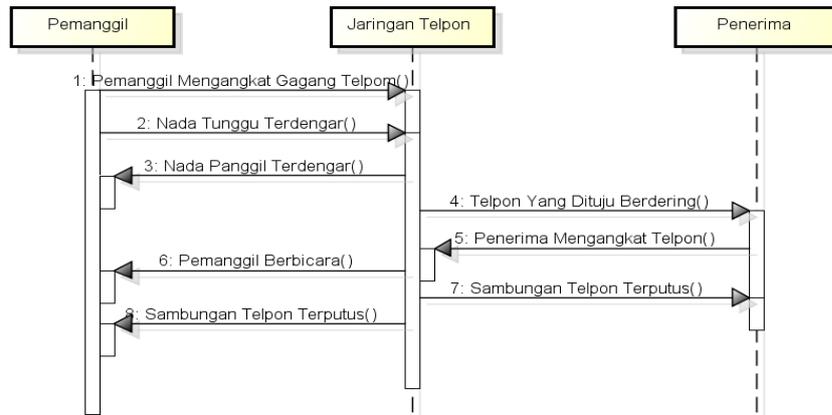
*Statechart diagram* memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian, dan aktifitas. Diagram ini penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif. *Statechart diagram* bermanfaat juga untuk menyediakan cara yang baik dalam memodelkan komunikasi yang terjadi dengan entitas luar via protokol atau sistem dasarnya. Contoh *statechart diagram* pada Gambar 2.10.



Gambar 2.10 Contoh Statechart Diagram (Widodo & Herlawati, 2011)

d. **Sequence Diagram**

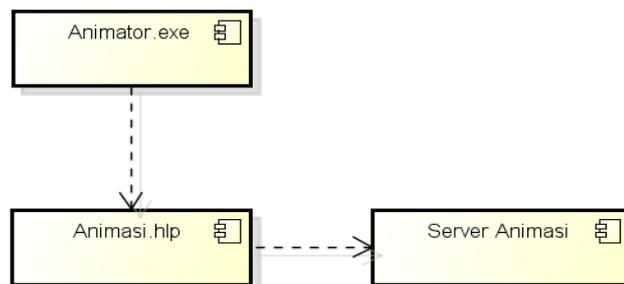
*Sequence diagram* adalah diagram interaksi yang menekankan pada urutan pengiriman pesan (*message-message*) dalam suatu waktu tertentu. *Sequence diagram* menggambarkan setiap objek sebagai garis tegak (vertikal) dan setiap event dituliskan pada arah mendatar (horizontal). Contoh *sequence diagram* pada Gambar 2.11.



Gambar 2.11 Contoh *Sequence Diagram* (Nugroho, 2005)

e. **Component Diagram**

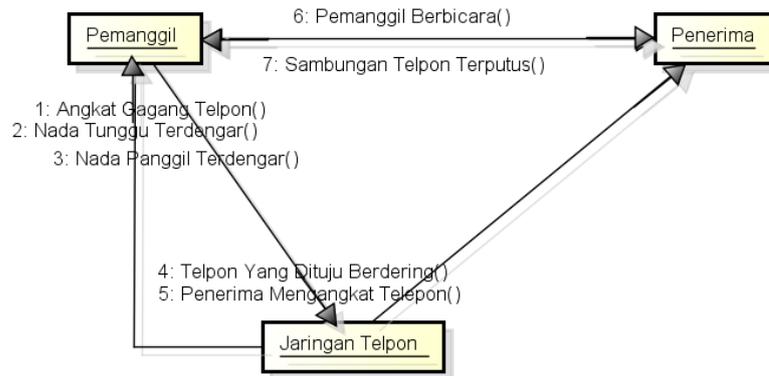
*Component diagram* memperlihatkan organisasi serta ketergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya, dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi. Contoh *component diagram* pada Gambar 2.12.



Gambar 2.12 Contoh *Component Diagram* (Nugroho, 2005)

f. **Collaboration Diagram**

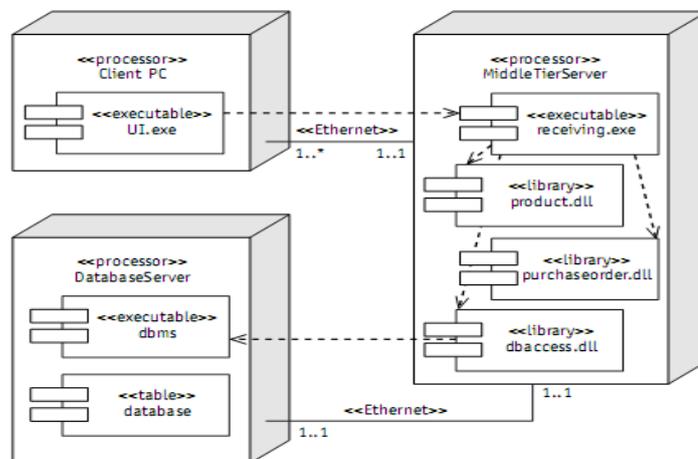
*Collaboration diagram* menggambarkan organisasi setiap objek-objek yang berpartisipasi pada interaksi. *Collaboration diagram* memberikan pemahaman yang jelas pada pembaca tentang aliran kendali dalam konteks organisasi struktural objek-objek yang berkolaborasi. Contoh *collaboration diagram* terdapat pada Gambar 2.13.



Gambar 2.13 Contoh *Collaboration Diagram* (Nugroho, 2005)

g. **Deployment Diagram**

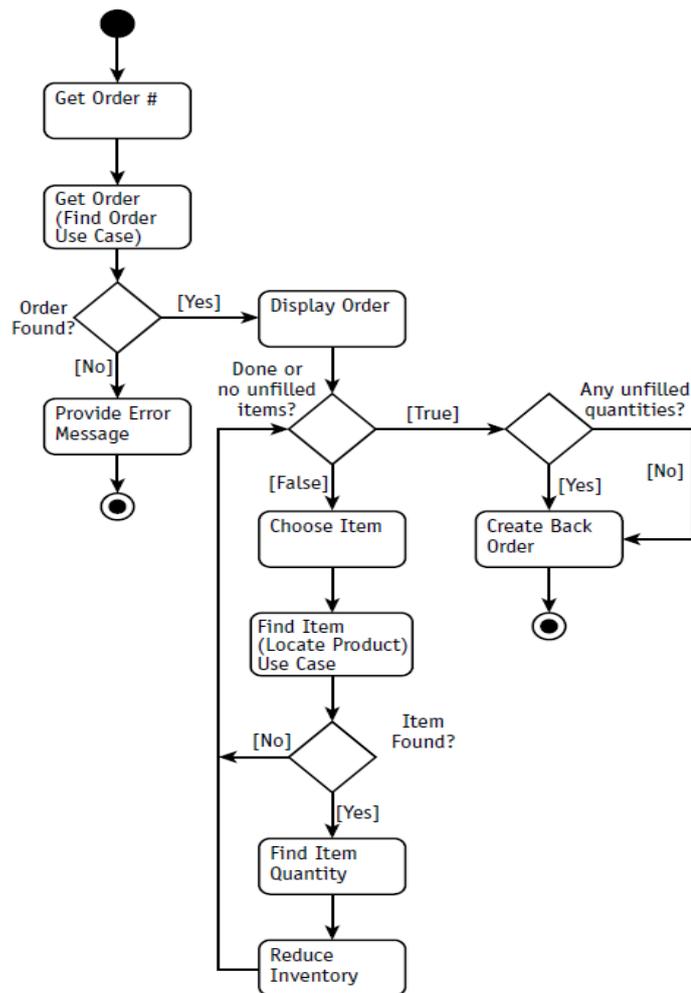
*Deployment diagram* memperlihatkan konfigurasi saat aplikasi dijalankan yang berhubungan erat dengan diagram komponen yang memuat satu atau lebih komponen-komponen. Contoh *deployment diagram* ditunjukkan pada Gambar 2.14.



Gambar 2.14 Contoh *Deployment Diagram* (Pender, 2002)

#### h. *Activity Diagram*

*Activity diagram* adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek. *Activity diagram* mungkin berdiri sendiri untuk memvisualisasikan, menspesifikasikan, mengkonstruksi, serta mendokumentasikan sifat dinamis dari sekumpulan objek atau dapat digunakan untuk memodelkan aliran kendali dari suatu operasi. Contoh *activity diagram* pada Gambar 2.15.

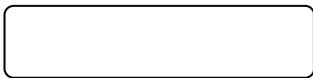
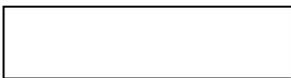
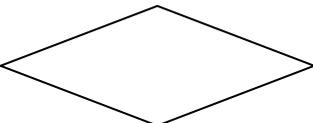
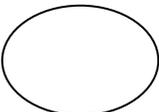
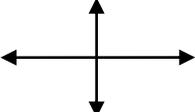


Gambar 2.15 Contoh *Activity Diagram* (Pender, 2002)

## 2.7 Flowchart (Diagram Alir)

Bagian alir program atau *flowchart* adalah bagian yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir. Gambar dan fungsi dari simbol-simbol yang digunakan pada *flowchart* terdapat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Flowchart* (Nugroho, 2005)

No	Simbol	Fungsi
1		Awal-akhir, untuk memulai dan mengakhiri suatu proses/ kegiatan.
2		Proses, suatu yang menunjukkan setiap pengolahan yang dilakukan oleh komputer.
3		Masukan-keluaran, memasukan hasil dari suatu proses.
4		Pengujian, suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan
5		<i>Connector</i> , suatu prosedur akan masuk atau keluar dalam lembar yang sama
6		<i>Off Page Connector</i> , merupakan simbol masuk atau keluarnya suatu prosedur pada kertas lembar lain
7		Arus <i>Flow</i> , simbol ini digunakan untuk menggambarkan arus proses.
8		<i>Predefined Process</i> , untuk menyatakan sekumpulan langkah proses
9		<i>Printer</i> , simbol ini digunakan untuk menggambarkan kegiatan mencetak suatu informasi dengan mesin printer

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Jenis Penelitian**

Dalam penelitian, peneliti menggunakan jenis penelitian terapan (*applied / practical research*). Jenis penelitian terapan adalah satu jenis penelitian yang hasilnya dapat secara langsung diterapkan untuk memecahkan permasalahan yang dihadapi. Penelitian terapan ini bertujuan membangun aplikasi berupa sistem pakar dengan menggunakan metode *forward chaining* untuk diagnosa cacat lahir yang diakibatkan oleh faktor teratogen berbasis android.

#### **3.2 Sarana Pendukung Pembuatan Sistem**

Dalam merancang dan membangun aplikasi ini, dibutuhkan perangkat keras dan perangkat lunak. Perangkat keras yang akan digunakan dalam penelitian ini yaitu laptop Axioo dengan spesifikasi *Processor Intel (R) Dual-Core CPU T4400 @2.20GHz (2 CPUs) Harddisk 250 GB, free space harddisk 25 GB* , Memori 896 MB RAM, Monitor LCD 14” dengan resolusi 1280 x 800 *pixels*, Keyboard (papan kunci), *mouse, Printer Canon PIXMA MP237*.

Adapun perangkat lunak yang digunakan dalam penelitian ini, antara lain Sistem Operasi Windows XP Professional, Bahasa Pemrograman Java, *Java Development Kit (JDK) 1.6, Java Runtime Environment (JRE)*, IDE Eclipse, *Microsoft Office Visio 2003* untuk perancangan *form* aplikasi, dan *Astah Community* untuk perancangan diagram UML.

### **3.3 Jenis Data**

Jenis data yang digunakan dalam penelitian ini yaitu:

#### **a. Data Primer**

Data primer adalah data yang diperoleh langsung dari responden melalui wawancara atau langsung dari sumber datanya. Data primer pada penelitian ini terdiri dari :

- 1) Jenis data masukan dan keluaran yang akan ditampilkan di sistem, seperti cacat lahir dan agen-agen teratogen.
- 2) Spesifikasi fungsi, kemampuan serta fasilitas dari sistem yang akan dibangun sehingga dapat digunakan oleh pengguna.

#### **b. Data Sekunder**

Data sekunder adalah data yang sudah jadi atau dipublikasikan untuk umum oleh instansi atau lembaga yang mengumpulkan, mengolah dan menyajikan. Data sekunder berupa data-data deskripsi maupun abstrak hasil penelitian, buku ilmiah, laporan penelitian, karangan ilmiah, skripsi, dan sumber-sumber tertulis lainnya baik tercetak maupun elektronik lain.

### **3.4 Teknik Pengumpulan Data**

Teknik pengumpulan data yang digunakan dalam penelitian ini yaitu:

#### **a. Wawancara**

Wawancara dilakukan dengan Dosen Fakultas Keguruan Ilmu dan Pendidikan Universitas Bengkulu yaitu Dr. Aceng Ruyani, M.S dalam pengumpulan data mengenai ilmu teratologi dan jenis-jenis data cacat lahir yang akan ditampilkan pada sistem, desain antarmuka sistem, spesifikasi fungsi, kemampuan serta fasilitas dari sistem pakar yang akan dibangun.

b. Studi Dokumentasi

Studi dokumentasi dalam pengumpulan data penelitian ini dimaksudkan sebagai cara mengumpulkan data-data dengan mempelajari dan mencatat bagian-bagian penting dari berbagai risalah resmi, baik langsung dari instansi/lembaga tertentu, buku-buku, laporan kegiatan, jurnal ilmiah yang relevan dengan penelitian.

### 3.5 Metode Pengembangan Sistem Pakar

Metode yang akan digunakan dalam pengembangan sistem pakar ini adalah *Linear Model of Expert System Development* yang merupakan model pengembangan khusus untuk pengembangan sistem pakar yang terdiri dari enam tahap yaitu (Tim Penerbit ANDI, 2009) :

a. Perencanaan

Tahap perencanaan merupakan pedoman awal dalam membangun sistem pakar ini. Pada tahap perencanaan ini terdapat beberapa proses, yaitu:

- 1) Penilaian kelayakan yaitu menentukan apakah perlu dibangun sistem pakar, jika perlu apakah sistem pakar yang direncanakan tersebut benar-benar mengaplikasikan sistem pakar yang sebenarnya. Kelayakan yang akan dinilai meliputi kelayakan teknik yaitu ketersediaan teknologi dan ketersediaan ahli, dan kelayakan operasi yaitu kemampuan dari pengguna sistem untuk menghasilkan informasi, kemampuan pengendalian sistem, serta efisiensi dan efektifitas sistem.
- 2) Spesifikasi kebutuhan, yaitu mendefinisikan apa saja yang akan dilakukan oleh sistem.

- 3) Penggambaran fungsi awal sistem, yaitu mendefinisikan sistem yang akan dibuat dengan menentukan fungsi dan tujuan sistem.

#### **b. Definisi Pengetahuan**

Tahap definisi pengetahuan yaitu:

- 1) Proses identifikasi dan seleksi sumber pengetahuan.

Identifikasi sumber yaitu menentukan siapa dan apa sumber pengetahuan yang diperlukan dalam pembangunan sistem pakar.

Prioritas sumber yaitu menentukan sumber pengetahuan dalam urutan prioritas. Ketersediaan sumber yaitu membuat daftar sumber pengetahuan diurutkan berdasarkan ketersediaannya.

- 2) Proses akuisisi, analisis dan ekstraksi pengetahuan.

Pada tahap ini terdapat tiga proses yaitu strategi perolehan dengan tujuan menentukan bagaimana pengetahuan akan diperoleh, proses identifikasi pengetahuan dasar yaitu memperoleh pengetahuan yang spesifik dari sumber yang akan digunakan dan proses sistem klasifikasi pengetahuan yaitu mengklasifikasikan pengetahuan yang akan membantu verifikasi dan pemahaman dalam mengembangkan sistem.

#### **c. Desain Pengetahuan**

Tahap desain pengetahuan terbagi menjadi dua proses, yaitu proses definisi dan desain detail.

- 1) Proses Definisi

Pada proses definisi, dilakukan beberapa tahap yaitu representasi pengetahuan dan perancangan aturan. Representasi pengetahuan

menspesifikasikan bagaimana pengetahuan disajikan. Perancangan aturan yaitu merancang aturan-aturan yang nantinya akan dijadikan sebagai acuan dalam pembuatan sistem pakar. Dari proses definisi ini diperoleh data dan siap untuk diolah sesuai dengan kebutuhan.

## 2) Proses Detail

Terdapat tiga tahap yaitu perancangan struktur kendali yang merancang tahap-tahap yang akan dilalui sistem pakar, perancangan *user interface*, yang merancang bentuk atau tampilan dari sistem pakar yang akan dibangun dan perancangan strategi implementasi yang merancang alur pertanyaan yang menggunakan metode inferensi *forward chaining*.

### **d. Pengkodean**

Tahap pengkodean meliputi proses pembuatan baris program (*coding*) dengan menggunakan bahasa pemrograman Java pada editor eclipse. Dari tahap pengkodean ini, maka diperoleh sistem pakar yang siap untuk ke tahap pengujian (tes).

### **e. Verifikasi Pengetahuan**

Dalam tahap implementasi ini, desain pengetahuan dan pengkodean sudah diformulasikan secara lengkap, proses inputan, instalasi, demonstrasi penerapan sistem, integrasi dan pengujian kasus. Akan lebih menjelaskan lagi sebatas mana kemampuan sistem pakar yang dibangun dan siap untuk diaudit jika diperlukan. Dalam tahap ini akan dilakukan pengujian prosedur-prosedur yang ada di dalam sistem dan pengujian proses analisis hasil pengujian dilakukan untuk rekomendasi perbaikan jika diperlukan.

## **f. Evaluasi Sistem**

Tahap evaluasi sistem merupakan tahapan pengujian akhir dan validasi sebelum sistem benar-benar digunakan. Evaluasi sistem mencakup semua uji coba dari fungsi sistem pada aplikasi. Hal ini dilakukan untuk mengetahui kesesuaian analisis yang telah dilakukan dengan sistem yang telah dibuat, dan dimungkinkan adanya perubahan pada sistem.

### **3.6 Metode Uji Kelayakan Sistem**

Uji kelayakan dilakukan untuk mendapatkan penilaian langsung terhadap sistem yang dihasilkan. Adapun tahapan dari uji kelayakan ini adalah :

#### **a. Kuesioner**

Kuesioner merupakan daftar pertanyaan yang diajukan pada responden untuk mencari jawaban dari permasalahan yang diteliti (Hasibuan, 2007). Teknik pengambilan sampel yang digunakan adalah teknik *purposive sampling*. *Purposive sampling* (sampel bertujuan) ialah teknik pengambilan sampel yang berdasarkan penilaian peneliti bahwa sampel tersebut merupakan pihak yang paling baik untuk dijadikan sampel penelitian.

#### **b. Tabulasi Data**

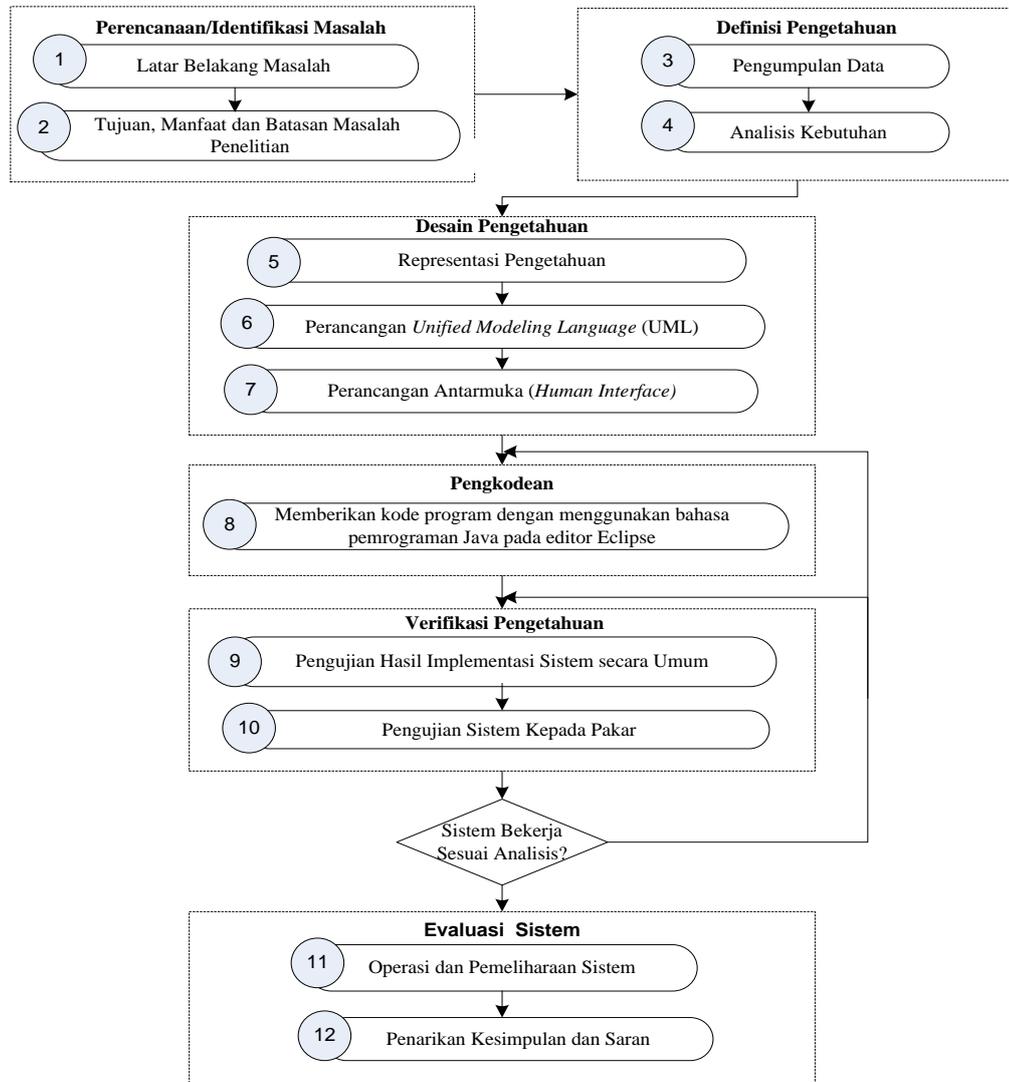
Proses perhitungan data kuesioner menggunakan skala *likert*. Skala *likert* adalah perhitungan skor pada tiap-tiap interval dari pernyataan yang diberikan ke responden. Sebelum melakukan perhitungan dengan menggunakan skala *likert*, maka terlebih dahulu dicari interval kelas dengan persamaan (3.1). Kemudian hasil dari proses perhitungan yang telah dilakukan, akan disajikan dalam bentuk tabel.

$$i = \frac{m-n}{k} \dots\dots\dots(3.1)$$

- Keterangan:
- i = Interval kelas
  - m = Angka tertinggi skor
  - n = Angka terendah skor
  - k = Banyak Kelas

### 3.7 Diagram Alir Penelitian

Penelitian dilakukan berdasarkan metode pengembangan sistem pakar yang digunakan. Alur kerja penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

Penjelasan:

- 1) Pada tahap pertama adalah penentuan basis awal dari sebuah penelitian yaitu latar belakang penelitian.
- 2) Tahap kedua yaitu menentukan tujuan, manfaat dan ruang lingkup penelitian. Ruang lingkup penelitian untuk menentukan batasan masalah penelitian agar tidak terlalu luas.
- 3) Pada tahap ketiga dilakukan pengumpulan data-data yang diperlukan dalam penelitian, meliputi jenis data masukan dan keluaran, spesifikasi fungsi, kemampuan dan fasilitas dari aplikasi.
- 4) Tahap keempat adalah proses analisis kebutuhan yang terdiri dari analisis kebutuhan sistem, kebutuhan proses, kebutuhan *input*, kebutuhan *output*, dan kebutuhan perangkat lunak dan perangkat keras.
- 5) Tahap kelima yaitu representasi pengetahuan berdasarkan pengetahuan yang telah diperoleh dari pakar.
- 6) Tahap keenam yaitu perancangan *Unified Modeling Language* (UML) yang memvisualisasikan fungsi dan kinerja sistem.
- 7) Tahap ketujuh yaitu antarmuka (*human interface*). Tahap perancangan antarmuka akan dibuat dalam beberapa rancangan tata letak aplikasi sesuai dengan analisis kebutuhan dari aplikasi.
- 8) Tahap kedelapan yaitu pengkodean. Pemberian kode program dengan menggunakan bahasa pemrograman Java pada *editor* Eclipse Kepler.
- 9) Tahap kesembilan yaitu dilakukan pengujian hasil implementasi sistem secara umum. Pengujian meliputi pengujian *input*, proses dan *output* aplikasi apakah telah sesuai dengan analisa sebelumnya.

- 10) Tahap kesepuluh yaitu pengujian kepada pakar. Hal ini bertujuan agar pengetahuan yang telah direpresentasikan ke dalam sistem sesuai dengan pengetahuan pakar.
- 11) Tahap kesebelas adalah operasi dan pemeliharaan sistem. Tahapan ini dilakukan setelah proses pengujian sistem telah berhasil dilakukan.
- 12) Tahapan terakhir adalah penarikan kesimpulan dan saran. Pada tahap ini dilakukan penarikan kesimpulan untuk mengetahui hasil yang telah diperoleh selama melakukan penelitian.

### 3.8 Jadwal Penelitian

Adapun jadwal penelitian terdapat pada Tabel 3.1.

Tabel 3.1. Jadwal Penelitian

No	Kegiatan	Bulan – Tahun					
		Jan '14	Feb' 14	Mar '14	Apr '14	Mei '14	Jun '14
1	Perencanaan						
	a. Latar belakang penelitian						
	b. Tujuan, manfaat, batasan penelitian						
2	Desain Pengetahuan						
	a. Mengumpulkan data						
	b. Analisis kebutuhan						
3	Definisi Pengetahuan						
	a. Perancangan UML						
	b. Perancangan antarmuka sistem						
4	Pengkodean						
5	Verifikasi Pengetahuan						
	a. Pengujian sistem secara umum						
	b. Pengujian kepada pakar						
6	Evaluasi Sistem						
	a. Operasi dan Pemeliharaan						
	b. Kesimpulan dan Saran						