

**SISTEM PAKAR DIAGNOSA PENYAKIT PADA IKAN BUDIDAYA AIR
TAWAR DENGAN METODE *FORWARD CHAINING* BERBASIS
ANDROID**

(Skripsi)

Oleh

Ardhika Praseda Ageng Putra



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2016**

ABSTRACT

DISEASE DIAGNOSIS EXPERT SYSTEM ON FRESHWATER AQUACULTURE FISH USING *FORWARD CHAINING* ANDROID BASED

By

ARDHIKA PRASEDA AGENG PUTRA

Fish disease is undesirable for fish farmers, because it can cause low productivity and high mortality aquaculture. Lack of information on disease prevention methods small number of experts causing many farmers suffered losses. Therefore, an expert system is needed to help diagnose the disease in fish. This system aims to help cultivators on resolve fish disease problem. In this research, expert system based on *android* was built using *Java* program language as database, *Forward chaining* was used. This method used to determine which roles will be executed, then that rule is executed, the process was repeated until we got the result. This system can diagnose 35 diseases from 8 species of fish. The data testing result used *Equivalence Partitioning*, showed that management rules system can run based on its function and system can diagnose the disease well. On the other based on questionnaire data, this application is an *user friendly* application (with average value of 4,29 / excellent).

Keywords: *Forward Chaining*, Freshwater Aquaculture Fish Disease, Expert Systems, Android

ABSTRAK

SISTEM PAKAR DIAGNOSA PENYAKIT PADA IKAN BUDIDAYA AIR TAWAR DENGAN METODE *FORWARD CHAINING* BERBASIS ANDROID

Oleh

ARDHIKA PRASEDA AGENG PUTRA

Penyakit ikan merupakan hal yang tidak diinginkan oleh pembudidaya ikan, karena dapat menyebabkan panen tidak maksimal dan kematian masal pada ikan. Kurangnya informasi cara penanggulangan penyakit dan sedikitnya jumlah pakar perikanan menyebabkan pembudidaya mengalami banyak kerugian. Oleh karena itu dibutuhkan sistem pakar yang digunakan untuk mendiagnosa penyakit pada ikan budidaya air tawar. Sistem ini bertujuan untuk membantu pembudidaya dalam mengatasi masalah penyakit ikan. Dalam penelitian ini, sistem pakar yang dibangun berbasis *android* menggunakan bahasa pemrograman *Java* sebagai basis data. Metode penalaran yang digunakan yaitu *forward chaining*. Metode ini digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan, proses diulang hingga ditemukan suatu hasil. Sistem ini dapat mendiagnosa 35 penyakit dari 8 jenis ikan. Hasil data pengujian menggunakan pengujian *Equivalence Partitioning*, menunjukkan bahwa pengelolaan aturan (*rule*) sistem dapat berjalan sesuai fungsinya dan sistem dapat mendiagnosa penyakit dengan baik. Selain itu berdasarkan data angket, aplikasi ini adalah aplikasi yang *user friendly* (dengan nilai rata-rata 4,29 / sangat baik).

Kata kunci : *Forward Chaining*, Penyakit Ikan Budidaya Air Tawar, Sistem Pakar, Android

**SISTEM PAKAR DIAGNOSA PENYAKIT PADA IKAN BUDIDAYA AIR
TAWAR DENGAN METODE *FORWARD CHAINING* BERBASIS
ANDROID**

Oleh

ARDHIKA PRASEDA AGENG PUTRA

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar
SARJANA KOMPUTER

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG**

2016

Judul Skripsi

: **SISTEM PAKAR DIAGNOSA PENYAKIT
PADA IKAN BUDIDAYA AIR TAWAR
DENGAN METODE *FORWARD CHAINING*
BERBASIS ANDROID**

Nama Mahasiswa

: **Ardhika Praseda Ageng Putra**

Nomor Pokok Mahasiswa

: 1117032009

Jurusan

: Ilmu Komputer

Fakultas

: Matematika dan Ilmu Pengetahuan Alam

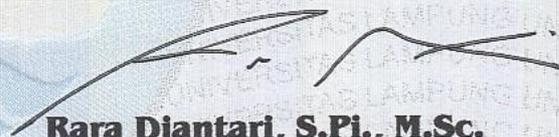
MENYETUJUI

1. Komisi Pembimbing



Aristoteles, S.Si., M.Si.

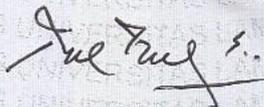
NIP 19810521 200604 1 002



Rara Diantari, S.Pi., M.Sc.

NIP 19790821 200312 2 001

2. Ketua Jurusan Ilmu Komputer



Dr. Ir. Kurnia Muludi, M.Sc.

NIP 19640616 198902 1 001

MENGESAHKAN

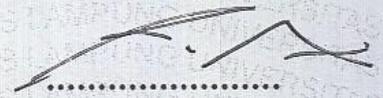
1. Tim Penguji

Ketua : Aristoteles, S.Si., M.Si.



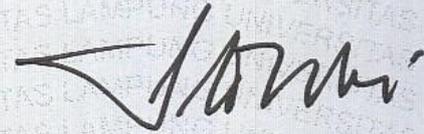
.....

Sekretaris : Rara Diantari, S.Pi., M.Sc.



.....

**Penguji
Bukan Pembimbing : Drs. Rd. Irwan Adi Pribadi, M.Kom.**



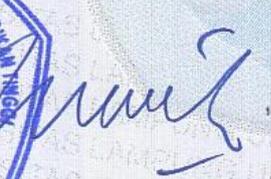
.....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Prof. Warsito, S.Si., D.E.A., Ph.D.

NIP 19710212 199512 1 001



Tanggal Lulus Ujian Skripsi : 28 April 2016

PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul **“Sistem Pakar Diagnosa Penyakit Pada Ikan Budidaya Air Tawar Dengan Metode *Forward Chaining* Berbasis Android”** merupakan karya saya sendiri dan bukan hasil karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 2 Mei 2016



Ardhika Praseda Ageng Putra
NPM. 1117032009

RIWAYAT HIDUP



Penulis dilahirkan di Sukapura Kabupaten Lampung Barat pada tanggal 10 Februari 1994, sebagai anak pertama dari tiga bersaudara dengan Ayah bernama Hadiyat dan Ibu bernama Dasiyam.

Penulis memiliki dua orang adik bernama Derry dan Alfanesya. Penulis menyelesaikan Taman Kanak-Kanak (TK) pada tahun 1999 di TK Banda Mulya Sukapura, Sekolah Dasar (SD) Negeri 2 Sukapura pada tahun 2005, Sekolah Menengah Pertama (SMP) Negeri 1 Sumberjaya pada tahun 2008, dan Sekolah Menengah Kejuruan (SMK) 2 Mei Bandar Lampung pada tahun 2011.

Pada tahun 2011, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur ujian Mandiri. Pada bulan Januari-Maret 2014, penulis melakukan Kuliah Kerja Nyata selama 40 hari di Desa Bandar Dalam Kecamatan Pulau Pisang Kabupaten Pesisir Barat. Pada bulan Januari 2015, penulis melakukan kerja praktik di PT. PLN (Persero) Pembangkitan Sumatera Bagian Selatan, Sektor Pengendalian Pembangkitan Bandar Lampung Unit PL Besai Kecamatan Sumber Jaya, Kabupaten Lampung Barat.

Selama menjadi mahasiswa penulis aktif dalam Organisasi UKM Fotografi ZOOM Universitas Lampung dengan menjabat sebagai Anggota Divisi Pendidikan pada tahun 2013-2014, Kepala Divisi Pendidikan pada tahun 2014-2015 dan Dewan Penasehat Organisasi pada tahun 2015-2016.

PERSEMBAHAN

Puji dan syukur saya panjatkan kepada Allah SWT atas segala berkah-Nya sehingga skripsi ini dapat terselesaikan.

Kupersembahkan karya ini kepada :

Bapak dan Mama yang telah membesarkan, mendidik, memberikan doa, dukungan dan semangat untuk kesuksesan anak-anaknya. Terimakasih atas semua perjuangan, pengorbanan, kesabaran dan kasih sayang telah kalian berikan untukku.

Adik-adikku yang aku sangat sayangi Derry Alfarizi dan Alfanesya Nasita Fadila Putri serta keluarga besar tercinta.

Keluarga Keluarga Ilmu Komputer 2011,

Keluarga besar UKM Fotografi ZOOM UNILA yang telah mengajarkan saya pelajaran hidup yang sangat berharga terimakasih atas pengalamannya.

Serta Almamater Tercinta,

Universitas Lampung.

MOTO

"Kegagalan hanya terjadi bila kita menyerah"

(Ardhika praseda AP)

*"Barang siapa ingin mutiara, harus berani terjun di
lautan yang dalam."*

(BungKarno)

*"Jadikanlah sabar dan sholat sebagai penolongmu,
Sesungguhnya Allah bersama orang-orang yang
sabar".(Q.S Al-Baqarah: 153)*

SANWACANA

Puji syukur penulis panjatkan kehadirat Allah *subhanahu wa ta'ala* atas berkat rahmat, hidayah, dan kesehatan yang diberikan sehingga penulis dapat menyelesaikan penulisan skripsi ini.

Skripsi ini disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Universitas Lampung. Judul dari skripsi ini adalah “Sistem Pakar Diagnosa Penyakit Pada Ikan Budidaya Air Tawar Dengan Metode *Forward Chaining* Berbasis Android”.

Dalam penyusunan skripsi ini, penulis banyak menghadapi kesulitan. Namun, berkat bantuan dan dorongan dari berbagai pihak, akhirnya penulis dapat menyelesaikan skripsi ini. Untuk itu pada kesempatan ini, penulis mengucapkan terimakasih kepada :

1. Bapak dan Mama yang telah memberikan doa, dukungan dan semangat serta memfasilitasi kebutuhan untuk menyelesaikan skripsi ini serta adiku tercinta Alfanesia Nasita Fadila Putri dan Derry Alfarizi.
2. Bapak Aristoteles, S.Si., M.Si., sebagai pembimbing I sekaligus pembimbing akademik penulis yang telah memberikan ide dan masukan dalam pengerjaan skripsi serta memberikan dorongan sehingga skripsi ini dapat diselesaikan.

3. Ibu Rara Diantari, S.Pi., M.Sc., sebagai pembimbing II penulis, yang telah memberikan data, saran, bantuan, dan membimbing penulis dalam pembuatan skripsi ini.
4. Bapak Drs.Rd. Irwan Adi Pribadi, M.Kom., sebagai pembahas yang telah memberikan masukan-masukan dan saran yang bermanfaat dalam skripsi ini.
5. Bapak Dr. Ir. Kurnia Muludi, M.Sc.Selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
6. Bapak Didik Kurniawan, S.Si.,MT Selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Bapak Prof. Warsito, S.Si., D.E.A., Ph.D selaku Dekan FMIPA Universitas Lampung.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu pengetahuan yang bermanfaat bagi penulis.
9. Ibu Anita, A.Md., selaku staf administrasi di Jurusan Ilmu Komputer yang telah membantu segala urusan administrasi selama kuliah.
10. Mahasiswa dan Dosen Jurusan Budidaya Perairan Fakultas Pertanian Universitas Lampung yang telah memberikan masukan-masukan dalam skripsi ini.
11. Keluarga KKN Pulau Pisang Yeni Gusamawati, Yazid, Radit, Iin Winda, Wayan Ferli, Wiwi, Wawan dan Arif.
12. Teman-teman seperjuangan Rifki, Bobby, Okky, Budiman, Rian, Ardye, Fajri, Tryo, Basir, Rudra, Amir, Fathan, Galih, Gamma, Panji, Maya, Aqila, Fitriana serta seluruh teman-teman Ilkom 2011 yang tidak bisa

disebutkan satu persatu. Terimakasih telah memberikan penulis inspirasi dan keceriaan selama perkuliahan.

13. Sahabat penulis terutama angkatan XIV yang keren-keren, tahan banting dan siap naik turun gunung (istilahnya) hehe.... Rifki Wardana (Bang Oma) , Ridwan (si Bunda) , Dona Prayogo (si Belerrr), Tryogo (Moong) dan Ronald Sihombing. Serta untuk Kak Putra, Kak Fachmi, Lian, Rama, Ableh, Ceti, Mbung, Iga, Ebta, Ismia, Adoy dan Okto yang selalu siap jadi tempat sharing dan menggila bersama.... hehe dan adik-adiku Ade , Tata, Irna , Mia, Tiwi, Melia, Mayang, Arif, Dika, Iqbal dan angkatan 17 & 18 dan senior Bangkit, Bang Tegar, bang M, Bang Haris, Bang Arfan dan banyak lagi yang lain yang telah banyak mengajarkan ilmu fotografi dan organisasi selama berada di keluarga besar UKM Fotografi ZOOM tercinta ini. Terimakasih atas pengalaman berharga ini.

Akhir kata, semoga Allah SWT selalu memberikan rahmat dan hidayah-Nya kepada kita semua dan semoga skripsi ini bermanfaat bagi kita semua. Amin.

Bandar Lampung, 2 Mei 2016

Ardhika Praseda Ageng Putra

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN	iv
RIWAYAT HIDUP	v
PERSEMBAHAN.....	vii
MOTTO	viii
SANWACANA	ix
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL	xix
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat	4
BAB II. TINJAUAN PUSTAKA.....	5
2.1 Budidaya Ikan Air Tawar.....	5
2.1.1 Penyakit Ikan Budidaya Air Tawar.....	7
2.2 Sistem Pakar.....	13
2.2.1 Definisi Sistem Pakar.....	13
2.2.2 Ciri-Ciri Sistem Pakar.....	13

2.2.3 Manfaat Sistem Pakar	14
2.2.4 Konsep Dasar Sistem Pakar	15
2.2.4.1 Kepakaran	15
2.2.4.2 Pakar	15
2.2.4.3 Pengalihan Kepakaran	16
2.2.4.4 Inferensi	16
2.2.4.5 Aturan	16
2.2.4.6 Kemampuan Menjelaskan.....	16
2.2.5 Struktur Sistem Pakar	17
2.2.6 Struktur Sistem Pakar	17
2.2.7 Basis Pengetahuan	21
2.2.8 Mekanisme Inferensi.....	21
2.2.8.1 Metode <i>Forward Chaining</i>	24
2.3 Android	27
2.3.1 Pengertian Android	27
2.3.2 Arsitektur Android	29
2.4 Metodologi Pengembangan Sistem.....	32
2.4.1 Metode <i>Waterfall</i>	32
2.4.2 <i>Unified Modelling Language (UML)</i>	34
2.4.3 Keunggulan UML	42
2.5 Teknik Pengujian Perangkat Lunak	43
2.5.1 <i>Equivalence Partitioning</i>	44
2.5.2 Probabilitas Klasik	45
2.5.3 <i>Skala Likert</i>	46
BAB III. METODOLOGI PENELITIAN	47
3.1 Waktu dan Tempat Penelitian	47
3.2 Perangkat Penelitian.....	47
3.3 Tahapan Penelitian.....	48
3.3.1 Studi Literatur	48

3.3.2 Pengumpulan Data	49
3.3.3 Perancangan Sistem	49
3.3.3.1 Perancangan UML (<i>Unified Modelling Language</i>).....	49
3.3.4.2 Perancangan Antarmuka.....	57
3.4 Metode Pengujian Sistem	63
BAB IV. HASIL DAN PEMBAHASAN	69
4.1 Analisa Kebutuhan Data	69
4.2 Representasi Pengetahuan.....	70
4.3 Implementasi Sistem	70
4.4 Tampilan Aplikasi Sistem Pakar Diagnosa Penyakit Ikan.....	71
4.4.1 Tampilan Halaman <i>Splash Screen</i>	71
4.4.2 Tampilan Menu Utama	72
4.4.3 Tampilan Menu Data Penyakit	73
4.4.4 Tampilan Menu Konsultasi.....	73
4.4.4.1 Sub Menu Konsultasi Berdasarkan Penyakit.....	75
4.4.4.2 Sub Menu Konsultasi Berdasarkan Gejala	77
4.4.4.3 Analisa Persentase Penyakit	79
4.4.5 Tampilan Menu Bantuan.....	80
4.4.6 Tampilan Menu Tentang.....	80
4.5 Hasil Pengujian	81
4.5.1 Pengujian Fungsional.....	81
4.5.1.1 Pengujian Versi Android	82
4.5.1.2 Pengujian Resolusi Layar dan Densitas Layar	84
4.5.1.3 Pengujian <i>User Interface</i>	86
4.5.1.4 Pengujian Fungsi dari Menu Aplikasi	89
4.5.2 Pengujian Non Fungsional.....	91
BAB V. SIMPULAN DAN SARAN.....	96
5.1 Kesimpulan	96
5.2 Saran	96

DAFTAR PUSTAKA	98
LAMPIRAN.....	100

DAFTAR GAMBAR

Gambar	Halaman
Gambar 2.1 Ikan Mas	6
Gambar 2.2 Ikan Gurami	6
Gambar 2.3 Ikan Nila.....	7
Gambar 2.4 Ikan Patin.....	7
Gambar 2.5 Ikan Lou Han.....	7
Gambar 2.6 Ikan Arwana	7
Gambar 2.7 <i>Ichthyophthirius multifiliis</i> (Gusrina, 2008).....	8
Gambar 2.8 <i>Saprolegniaceae</i> (Gusrina, 2008).....	9
Gambar 2.9 <i>Aeromonas sp</i> (Gusrina, 2008).....	10
Gambar 2.10 Struktur Sistem Pakar (Kusrini, 2006)	18
Gambar 2.11 Proses <i>Forward Chaining</i> (Arhami, 2005)	22
Gambar 2.12 Proses <i>Backward Chaining</i> (Arhami, 2005)	23
Gambar 2.13 Diagram Alir Teknik Penelusuran <i>Depth-first Search</i>	24
Gambar 2.14 Diagram Alir Teknik Penelusuran <i>Breadth-first Search</i>	24
Gambar 2.15 Iterasi ke-1	25
Gambar 2.16 Iterasi ke-2.....	26
Gambar 2.17 Iterasi ke-3.....	27
Gambar 2.18 Metode Waterfall (Pressman, Roger S. 2001).....	32
Gambar 2.19 Contoh Aktor (Fowler,2004).....	36
Gambar 2.20 Use Case (Fowler, 2004).....	37
Gambar 2.20 Use Case (Fowler, 2004).....	37
Gambar 2.21 Contoh Activity Diagram (uml-diagrams.org, 2014).....	39

Gambar 2.22 Contoh Sequence Diagram (uml-diagrams.org, 2014).....	42
Gambar 3.1 Diagram Alir Penelitian	48
Gambar 3.2 <i>Use Case</i> Diagram.....	50
Gambar 3.3 <i>Activity</i> Diagram Data Penyakit	51
Gambar 3.4 <i>Activity</i> Diagram Konsultasi.....	52
Gambar 3.5 <i>Activity</i> Diagram Bantuan	53
Gambar 3.6 <i>Activity</i> Diagram Tentang	53
Gambar 3.7 <i>Sequence</i> Diagram Data Penyakit	55
Gambar 3.8 <i>Sequence</i> Diagram Konsultasi.....	55
Gambar 3.9 <i>Sequence</i> Diagram Bantuan	56
Gambar 3.10 <i>Sequence</i> Diagram Tentang.....	57
Gambar 3.11 <i>Design Layout Splash Screen</i>	58
Gambar 3.12 <i>Design Layout</i> Menu Utama	58
Gambar 3.13 <i>Design Layout</i> Data Penyakit.....	59
Gambar 3.14 <i>Design Layout</i> Konsultasi	60
Gambar 3.15 <i>Design Layout</i> Sub Menu Konsultasi Gejala	60
Gambar 3.16 <i>Design Layout</i> Sub Menu Hasil Diagnosa	61
Gambar 3.17 <i>Design Layout</i> Hasil Diagnosa Penyakit (Penjelasan Penyakit)	62
Gambar 3.18 <i>Design Layout</i> Menu Bantuan.....	62
Gambar 3.26 <i>Design Layout</i> Menu Tentang.....	63
Gambar 4.1 Tampilan <i>Splash Screen</i>	72
Gambar 4.2 Tampilan Menu Utama.....	72
Gambar 4.3 Tampilan Menu Data Penyakit	74
Gambar 4.4 Tampilan Menu Konsultasi	74
Gambar 4.5 Tampilan Halaman Pilihan Konsultasi.....	75
Gambar 4.6 Tampilan Sub Menu Konsultasi Berdasarkan Penyakit.....	76
Gambar 4.7 Tampilan Hasil Analisa.....	76
Gambar 4.8 Tampilan Sub Menu Konsultasi Berdasarkan Gejala	78
Gambar 4.9 Tampilan Hasil Analisa.....	78

Gambar 4.10 Tampilan Menu Bantuan	80
Gambar 4.11 Tampilan Menu Tentang	81
Gambar 4.12 Grafik Presentasi Rata-rata Jawaban Responden per Kategori	
Penilaian Aplikasi <i>Sistem Pakar</i>	94

DAFTAR TABEL

Gambar	Halaman
Tabel 2.1 Jenis Sistem dan Komoditas akuakultur	6
Tabel 2.2 Notasi <i>Activity Diagram</i> (Meildy, 2014)	38
Tabel 2.3 Notasi <i>Sequence Diagram</i> (Meildy, 2014).....	41
Tabel 2.4 Tabel Kemungkinan Penyakit	45
Tabel 3.1 Tabel Daftar Pengujian	64
Tabel 4.1 Daftar Nama Ikan.....	69
Tabel 4.2 Pengujian Versi Android.....	82
Tabel 4.3 Pengujian Resolusi Layar dan Densitas Layar.....	84
Tabel 4.4 Pengujian <i>User Interface</i>	87
Tabel 4.5 Pengujian fungsi dari menu aplikasi	89
Tabel 4.6 Interval dan Kategori Penilaian.....	92
Tabel 4.7 Hasil Penilaian Variabel <i>Sistem Pakar</i>	93

DAFTAR LAMPIRAN

Lampiran	Halaman
1. Nama Penyakit.....	101
2. Data Penyakit dan Gejala	107
3. Data Penyakit	112
4. Data Gejala	113
5. Tabel Keputusan	115
6. Pohon Keputusan	118
Basis Aturan	119

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ikan merupakan hewan yang hidup di air yang menjadi salah satu dari sekian banyak sumber protein yang dibutuhkan manusia. Ikan sangat bermanfaat bagi manusia sebab mengandung bermacam zat yang dibutuhkan oleh tubuh manusia (Apriadji, 2010). Selain itu apabila dibandingkan dengan sumber penghasil protein lain seperti daging, susu, dan telur harga ikan relatif lebih murah. Mengingat pentingnya ikan bagi manusia, tak heran bila manusia berusaha mendapatkan ikan dalam jumlah yang mencukupi, antara lain dengan mengusahakan melakukan pencarian di sumbernya yakni laut dan ada pula yang memeliharanya dengan sebaik-baiknya yang lazim disebut dengan usaha perikanan. Ikan yang pemeliharaannya di danau biasanya adalah ikan air tawar yang pemeliharaannya secara keseluruhan dilakukan di dalam jaring apung yang telah disediakan oleh para pengusaha perikanan air tawar ini.

Dalam proses budidaya ikan air tawar, pembudidaya ikan mengalami beberapa kendala, salah satu kendala yang dimaksud yaitu terjangkitnya penyakit pada ikan-ikan air tawar yang dibudidayakan. Penyakit ikan dibagi menjadi dua yaitu penyakit infeksi (bakteri, virus, parasit, dan jamur) dan penyakit non- infeksi

(tumor, gangguan gizi, pakan, dan traumatik). Hal ini sangat mempengaruhi kematian pada ikan. Kematian jumlah ikan yang besar tentu akan berdampak kerugian yang sangat besar bagi para pembudidaya.

Penyakit ikan merupakan hal yang tidak diinginkan bagi pembudidaya ikan, karena dapat menyebabkan panen tidak maksimal dan kematian massal pada ikan. Untuk mengatasi kendala tersebut maka pembudidaya membutuhkan suatu pengetahuan tentang informasi penyakit, gejala, dan penanganan untuk penyakit tersebut. Tetapi ketersediaan informasi mengenai penyakit ikan masih sedikit, hal ini menyebabkan kesulitan dalam penanggulangannya maupun cara pengobatannya. Oleh sebab itu dibutuhkan peran seorang pakar dibidang perikanan sebagai tempat konsultasi. Pakar perikanan juga diharapkan dapat memberikan informasi mengenai penyakit, cara penanggulangan, pengobatan, dan solusi mengatasinya. Akan tetapi ketersediaan pakar perikanan saat ini masih kurang dan untuk menghubungi seorang pakar penyakit ikan, pembudidaya membutuhkan biaya, waktu, dan tenaga yang tidak sedikit. Berdasarkan hal tersebut dikembangkan suatu sistem pakar tentang penyakit ikan air tawar, sehingga dapat memberikan solusi untuk menanggulangi penyakit ikan.

Sebelumnya telah dilakukan penelitian oleh (Pratama, 2015) mengenai Sistem Pakar Diagnosa Penyakit Pada Ikan Budidaya Air Tawar Dengan Metode Forward Chaining berbasis *Website*. Pada penelitian ini menghasilkan sebuah media konsultasi bagi para pembudidaya dalam mendiagnosa penyakit ikan dan memberikan solusi terkait penyakit yang diderita layaknya seorang pakar. Akan tetapi dari penelitian tersebut memiliki beberapa kelemahan yaitu sistem pakar diagnosa yang dibangun tersebut kurang praktis dalam penggunaannya, karena

pengguna harus mengakses website yang telah dibuat dan dalam proses penggunaannya kurang efisien. Apabila di bandingkan dengan sistem pakar berbasis mobile tentunya akan lebih jauh efisien dalam proses penggunaannya. Untuk mengatasi masalah ini, maka perlu dikembangkan lebih lanjut tentang sistem pakar yang telah diteliti oleh penulis (Pratama, 2015) dalam bentuk sistem pakar berbasis *Android*.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas dalam penelitian ini adalah bagaimana merancang dan membangun suatu sistem pakar yang dapat digunakan untuk mendiagnosa penyakit pada ikan konsumsi dan hias pada sistem *Android*.

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Jenis ikan yang dapat didiagnosa pada sistem pakar ini yaitu ikan air tawar konsumsi meliputi ikan Mas, ikan Gurami, ikan Lele, ikan Nila, ikan Patin, dan ikan hias air tawar yaitu ikan Koi, ikan Arwana, dan ikan Lou han.
2. Penyakit yang dapat diidentifikasi sebanyak 35 penyakit infeksi dengan 74 gejala.
3. Metode penalaran yang digunakan yaitu *forward chaining*.

1.4 Tujuan Penelitian

1. Tujuan dari penelitian ini adalah merancang dan membangun aplikasi sistem pakar berbasis *Android* menggunakan metode penalaran *forward chaining*.
2. Membantu pengguna untuk konsultasi masalah penyakit ikan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Mempermudah pengguna untuk mengetahui jenis gejala dan penyakit pada ikan.
2. Memberikan informasi kepada pengguna untuk mengetahui bagaimana cara menangani penyakit yang menyerang ikan air tawar konsumsi dan hias.

BAB II

TINJAUAN PUSTAKA

2.1 Budidaya Ikan Air Tawar

Menurut Yani (2007) dalam Pratama (2015) budidaya merupakan kegiatan terencana pemeliharaan sumber daya hayati yang dilakukan pada suatu areal lahan untuk diambil manfaat/hasil panennya. Budidaya perikanan adalah usaha pemeliharaan dan pengembangbiakan ikan atau organisme air lainnya. Perikanan budidaya air tawar ialah perikanan yang terdapat di sawah, sungai, danau, kolam, dan rawa.

Keberhasilan budidaya ikan air tawar sangat ditentukan oleh lingkungan yaitu tanah dan air. Jenis tanah sangat menentukan faktor keberhasilan budidaya ikan air tawar, jenis tanah yang baik untuk budidaya air tawar adalah jenis tanah liat atau lempung. Tanah jenis ini sangat baik untuk pembuatan kolam. Air sebagai media kehidupan ikan, maka keberadaan air sangat mutlak diperlukan. Jumlah dan kualitas air harus selalu menjadi perhatian agar usaha budidaya ikan air tawar bisa menjadi optimal.

Pada sistem budidaya ikan terdapat beberapa komoditas yang sudah lazim dibudidayakan di Indonesia dapat dilihat pada Tabel 2.1

Tabel 2.1. Jenis Sistem dan Komoditas akuakultur yang dibudidayakan di Indonesia.

Sistem	Komoditas
Kolam air tenang	Ikan mas, nila, gurami, udang galah, patin, bawal, tawes, ikan hias, tambakan, sepat, kowan, mola, sidat
Kolam air deras	Ikan mas
Tambak	Udang windu, bandeng, belanak, mujair, nila, kakap putih, kerapu, rumput laut, kepiting bakau, udang galah
Jaring apung	Kerapu, kakap, udang windu, bandeng, samadar, ikan hias laut, ikan mas, nila, mujair, gurami, patin, bawal, sidat, ikan hias air tawar.
Jaring tancap	Kerapu, kakap, udang windu, bandeng, samadar, ikan hias laut, ikan mas, nila, mujair, gurami, patin, bawal, sidat, ikan hias air tawar
Keramba	Ikan mas, nila, mujair, patin, gurami, betutu
Kombongan	Ikan mas, ikan nila
Akuarium/tangki/bak	Ikan hias, benih ikan konsumsi, plankton pakan alami

Beberapa komoditas ikan budidaya air tawar disajikan pada Gambar 2.1 – 2.6.

(Sumber : <http://www.fishbase.org/search.php> .)



Gambar 2.1. Ikan Mas
(*Cyprinus caprio*)



Gambar 2.2. Ikan Gurami
(*Osphronemus gourami*)



Gambar 2.3. Ikan Nila
(*Orheochromis niloticus*)



Gambar 2.4. Ikan Patin
(*Pangasius pangasius*)



Gambar 2.5. Ikan Lou Han
(*Amphilophus citrinellus*)



Gambar 2.6. Ikan Arwana
(*Scleropages formosus*)

2.1.1 Penyakit Ikan Budidaya Air Tawar

Kegiatan budidaya ikan air tawar baik jenis ikan konsumsi dan ikan hias merupakan kegiatan yang mempunyai resiko yang tinggi karena ikan merupakan makhluk bernyawa yang kapan saja mengalami kematian. Salah satu penyebab gagalnya kegiatan budidaya ikan adalah karena faktor penyakit. Munculnya penyakit pada ikan umumnya merupakan hasil interaksi kompleks/ tidak seimbang antara tiga komponen dalam ekosistem perairan yaitu inang (ikan yang lemah), patogen yang ganas, dan kualitas lingkungan yang memburuk (Afrianto).

Secara umum penyakit ikan dapat dikelompokkan menjadi dua, yaitu:

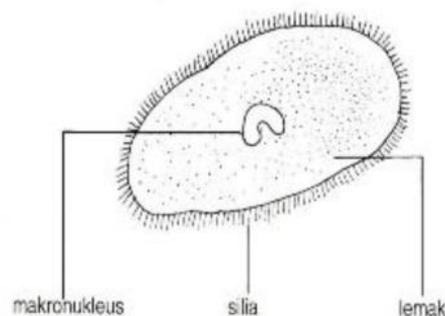
1. Penyakit infeksi

Penyakit infeksi merupakan penyakit yang disebabkan oleh:

a. Parasit

Akibat dari penyakit yang disebabkan oleh parasit secara ekonomis cukup merugikan yaitu dapat menyebabkan kematian, menurunkan bobot, bentuk serta ketahanan tubuh ikan sehingga dapat dimanfaatkan sebagai jalan masuk bagi infeksi sekunder oleh patogen lain seperti jamur, bakteri dan virus. Salah satu jenis *protozoa* yang paling sering menjadi kendala dalam budidaya ikan adalah *Ichthyophthirius multifiliis* atau *ich* (penyakit bintik putih).

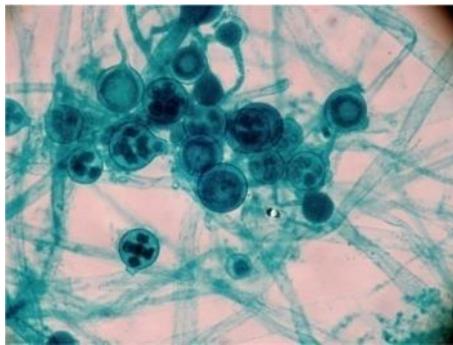
Sifat serangannya sangat sporadis dan kematian yang diakibatkannya dapat mencapai 100% populasi dalam tempo yang relatif singkat. Secara umum gejala ikan yang terserang *protozoa* adalah ikan tampak pucat, nafsu makan berkurang, gerakan lambat (sering menggosok-gosokkan tubuhnya pada dinding kolam), adanya bercak-bercak putih pada permukaan tubuh ikan, dan pada infeksi lanjut ikan megap-megap dan meloncat-loncat ke permukaan air untuk mengambil oksigen.



Gambar 2.7. *Ichthyophthirius multifiliis* (Afrianto, 1992)

b. Jamur

Semua jenis ikan air tawar termasuk telurnya rentan terhadap infeksi jamur. Beberapa faktor yang sering memicu terjadinya infeksi jamur adalah penanganan yang kurang baik (transportasi) sehingga menimbulkan luka pada tubuh ikan, kekurangan gizi, suhu, dan oksigen terlarut yang rendah, bahan organik tinggi, kualitas telur buruk/tidak terbuahi dan padatnya telur pada jaring yang disediakan untuk pembudidayaan telur ikan. Penyakit ini menular terutama melalui spora di air. Gejala-gejalanya dapat dilihat secara klinis adanya benang-benang halus menyerupai kapas yang menempel pada telur. Salah satu jenis jamur yang sering menjadi kendala adalah dari family *saprolegniaceae*.

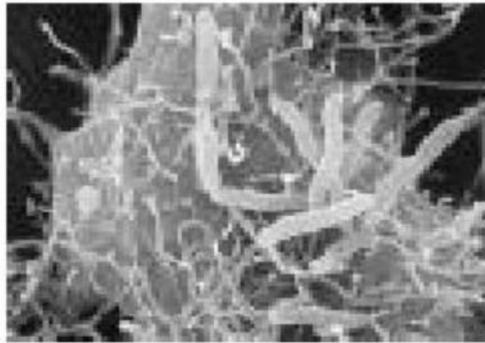


Gambar 2.8. *Saprolegniaceae* (Afrianto, 1992)

c. Bakteri

Penyakit yang disebabkan oleh bakteri adalah penyakit yang paling banyak menyebabkan kegagalan pada budidaya ikan air tawar. Penyakit akibat infeksi bakterial masih sering terjadi dengan intensitas yang variatif. Umumnya pembudidaya masih mengandalkan antibiotik sebagai “*magic bullet*” untuk melawan penyakit bakterial. Jenis penyakit yang disebabkan oleh bakteri antara lain adalah penyakit merah yang disebabkan oleh bakteri gram negatif (*Aeromonas hydrophila*), penyakit columnaris atau luka kulit, sirip dan inang

yang disebabkan oleh infeksi bakteri *Flavobacterium columnare*, penyakit tuberculosis yang tergolong sangat kronis disebabkan oleh bakteri gram positif *Mycobacterium sp*, dan penyakit *Streptocociasis* yang disebabkan oleh bakteri gram positif *Streptococcus sp*. *Aeromonas sp*.



Gambar 2.9. *Aeromonas sp* (Afrianto, 1992)

d. Virus

Patogen virus juga menyebabkan penyakit pada budidaya ikan air tawar. Belum banyak diketahui penyakit yang disebabkan oleh virus di Indonesia kecuali penyakit *Lymphocystis* dan *Koi Herpesvirus* (KHV). Infeksi *Lymphocystis* hanya bersifat kronis dan bila menyerang ikan hias akan mengalami kerugian yang berarti merusak keindahan ikan. Sampai saat ini KHV merupakan penyakit yang paling serius dan sporadis terutama untuk komoditi ikan mas koi.

2. Penyakit non-infeksi

Penyakit non-infeksi merupakan penyakit yang diakibatkan karena faktor:

a. Lingkungan

Pengaruh dari penyakit yang diakibatkan faktor lingkungan sering mengakibatkan kerugian yang serius karena kematian yang berlangsung sangat cepat, tiba-tiba, dan mematikan seluruh populasi ikan. Penyebabnya seperti keracunan akibat

peledakan populasi plankton, keracunan pestisida/limbah industri, bahan kimia, dan lainnya.

Faktor lingkungan yang buruk akan menyebabkan ikan menjadi:

- Tercekik yaitu kekurangan oksigen yang umumnya terjadi menjelang pagi hari pada perairan yang punya populasi *Phytoplankton* tinggi.
- Keracunan nitrit, Nitrit (NO_2^-) adalah ion-ion anorganik alami, yang merupakan bagian dari siklus nitrogen. Aktifitas mikroba di tanah atau air menguraikan sampah yang mengandung nitrogen organik pertama-pertama menjadi ammonia, kemudian dioksidasikan menjadi nitrit. Pengaruh dari nitrit menimbulkan penyakit darah cokelat karena disebabkan oleh konsentrasi nitrit yang tinggi di dalam air yang berasal dari hasil metabolisme ikan.
- Keracunan ammonia, terjadi hampir sama dengan nitrit tetapi pada umumnya karena pengaruh pemberian pakan yang berlebihan atau bahan organik, sedangkan populasi bakteri pengurai tidak mencukupi. Yang sangat beracun adalah dalam bentuk NH_3 .
- Fluktuasi suhu air yang ekstrim, dimana perubahan suhu air yang ekstrim akan merusak keseimbangan hormonal dan fisiologis tubuh ikan dan pada umumnya ikan tidak mampu untuk beradaptasi terhadap perubahan dan mengakibatkan ikan stress bahkan kematian.

b. Malnutrisi

Pemberian pakan yang berlebihan atau kekurangan dan tidak teratur juga dapat menyebabkan penyakit pada ikan. Penyakit karena malnutrisi jarang menunjukkan gejala spesifik sehingga agak sulit didiagnosa penyebab utamanya. Tetapi dalam

diet pakan dapat mengakibatkan kelainan fungsi morfologis dan biologis seperti defisiensi asam *panththenic* penyakit jaring insang ikan yang dapat menyebabkan ikan sulit bernafas yang diikuti dengan kematian, defisiensi vitamin A yang menyebabkan mata menonjol/buta dan terjadi pendarahan pada kulit juga ginjal, defisiensi vitamin B-1 yang menyebabkan kehilangan nafsu makan, pendarahan, dan penyumbatan pembuluh darah, defisiensi asam lemak essensial yang berakibat infiltrasi lemak pada kulit dan minimnya pigmentasi pada tubuh ikan. Yang cukup berbahaya adalah karena defisiensi vitamin C yang merupakan penyakit yang umum terjadi dimana akibat yang paling populer adalah *broken back syndrome* seperti *scoliosis* dan *lordosis*.

c. Genetis

Salah satu penyebab penyakit yang kompleks pada kegiatan budidaya ikan air tawar karena adanya faktor genetik terutama karena adanya perkawinan satu keturunan (*inbreeding*). Pernikahan *inbreeding* yang dilakukan secara terus-menerus akan menurunkan kualitas ikan berupa variasi genetik dalam tubuh ikan. Akibat dari pernikahan *inbreeding* adalah:

- Pertumbuhan ikan lambat (*bantet/kontet*) dan ukuran beragam.
- Lebih sensitif terhadap infeksi patogen.
- Organ tubuh badan yang tidak sempurna serta kelainan lainnya.

2.2 Sistem Pakar

2.2.1 Definisi Sistem Pakar

Sistem pakar sebagai sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Martin dan Oxman, 1988) dalam (Pratama, 2015).

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud di sini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sistem pakar dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusannya maupun hasil keputusan yang diperoleh (Kusrini, 2008).

2.2.2 Ciri-Ciri Sistem Pakar

Ciri-ciri sistem pakar menurut Sutojo (2011) dalam Pratama (2015), adalah:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran-penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Padat penebaran optimal.
4. Dirancang untuk dapat dikembangkan secara bertahap.
5. Pengetahuan dan mekanisme penalaran (inference) jelas terpisah.
6. Keluarannya bersifat anjuran.

7. Sistem dapat mengaktifkan kaidah secara searah yang sesuai dituntun oleh dialog dengan user.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai dituntun oleh dialog dengan *user*.

2.2.3 Manfaat Sistem Pakar

Sistem pakar menjadi populer karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas dengan memberi nasihat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Handal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Meningkatkan kapabilitas sistem komputer. Integrasi sistem pakar dengan sistem komputer lain membuat sistem lebih efektif dan mencakup lebih banyak aplikasi.
9. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
10. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.

11. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

2.2.4 Konsep Dasar Sistem Pakar

Pengetahuan dari suatu sistem pakar mungkin dapat direpresentasikan dalam sejumlah cara. Salah satu metode yang paling umum untuk merepresentasikan pengetahuan adalah dalam bentuk tipe aturan (*rule*) **IF...THEN** (Jika...maka).

Konsep dasar dari suatu sistem pakar mengandung beberapa unsur atau elemen, yaitu (Kusrini, 2006):

2.2.4.1 Kepakaran

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengambil keputusan lebih cepat dan lebih baik daripada seseorang yang bukan pakar.

2.2.4.2 Pakar

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan aturan-aturan serta menentukan relevansi kepakarannya.

2.2.4.3 Pengalihan Kepakaran

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu:

1. Akuisisi pengetahuan (dari pakar atau sumber lain)
2. Representasi pengetahuan (pada komputer)
3. Inferensi pengetahuan
4. Pemindahan pengetahuan ke pengguna

2.2.4.4 Inferensi

Inferensi adalah sebuah prosedur yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

2.2.4.5 Aturan

Kebanyakan aplikasi sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based system*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.

2.2.4.6 Kemampuan Menjelaskan

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem

yang disebut subsistem penjelasan (*explanation*). Bagian dari sistem ini memungkinkan sistem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi- operasinya.

2.2.5 Struktur Sistem Pakar

Sistem pakar dibagi menjadi empat bentuk, yaitu:

1. Berdiri Sendiri

Sistem pakar jenis ini merupakan perangkat lunak yang berdiri sendiri tidak bergabung dengan software lainnya.

2. Tergabung

Sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu algoritma (konvensional), atau merupakan program dimana di dalamnya memanggil algoritma subrutin lain (konvensional).

3. Menghubungkan ke perangkat lunak lain

Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya DBMS.

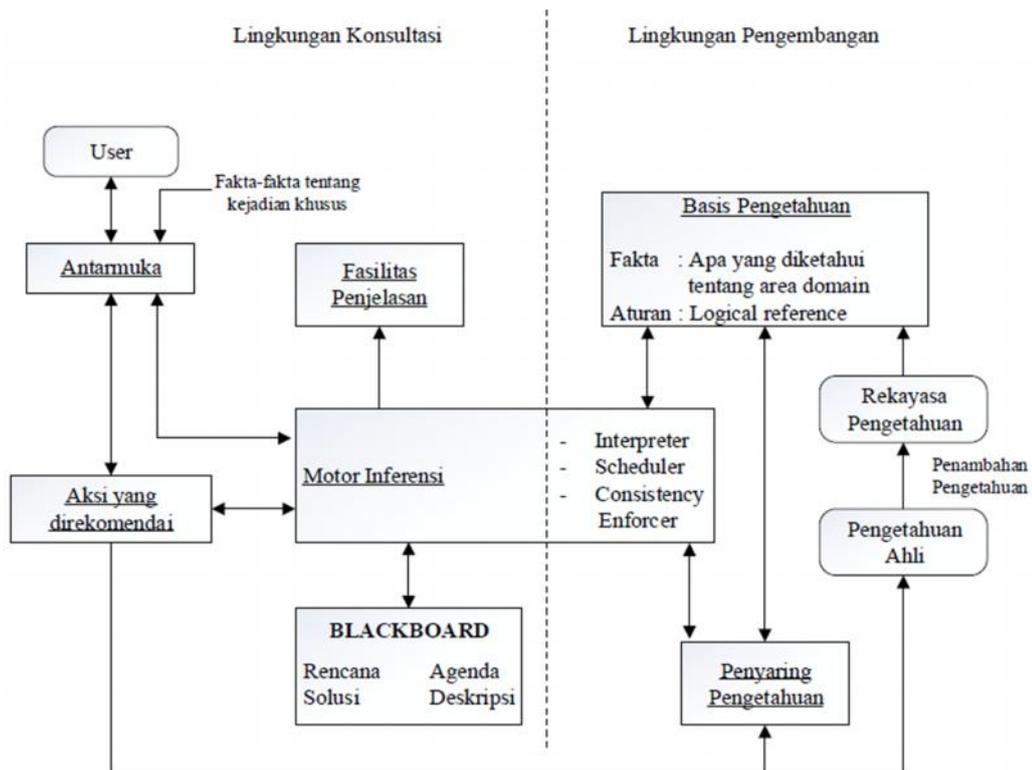
4. Sistem Mengabdi

Sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar.

2.2.6 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*)

(Kusrini,2006). Lingkungan pengembangan sistem pakar digunakan untuk memasukkan pengetahuan pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar. Komponen- komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam Gambar 2.10.



Gambar 2.10. Struktur Sistem Pakar (Kusrini, 2006).

Hasil pemrosesan yang dilakukan oleh mesin inferensi dari sudut pandang pengguna bukan pakar berupa aksi yang direkomendasikan oleh sistem pakar disertai dengan fasilitas-fasilitas penjelasan yang dibutuhkan pengguna. Komponen yang terdapat dalam sistem pakar, yaitu *design interface*, basis pengetahuan, akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, dan perbaikan pengaturan. Komponen-komponen yang terdapat dalam sistem pakar antara lain, sebagai berikut:

1. Antarmuka Pengguna (*User Interface*)

Antarmuka pengguna merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Pada bagian ini terjadi dialog antara program dan pemakai, yang memungkinkan sistem pakar menerima instruksi dan informasi (*input*) dari pemakai, juga memberikan informasi (*output*) kepada pemakai.

2. Basis Pengetahuan

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

3. Mesin Inferensi

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan (Arhami, 2005).

4. Akuisisi Pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian pemecahan masalah dari pakar atau sumber pengetahuan terdokumentasi ke

program komputer, untuk membangun atau memperluas basis pengetahuan. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, laporan penelitian, dan pengalaman pemakai.

5. *Blackboard* (Tempat Kerja)

Blackboard adalah area kerja memori tempat pendeskripsian masalah yang diberikan oleh data input, digunakan juga untuk perekaman hipotesis dan keputusan sementara. Tiga tipe keputusan dapat direkam dalam *blackboard*, yaitu rencana (bagaimana mengatasi persoalan), agenda (tindakan potensial sebelum eksekusi), dan solusi (hipotesis kandidat dan arah tindakan alternatif yang telah dihasilkan sistem sampai dengan saat ini) (Kusrini, 2006).

6. 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.

7. Perbaikan Pengetahuan

Pakar manusia memiliki sistem perbaikan pengetahuan, yakni mereka dapat menganalisis pengetahuannya sendiri dan kegunaannya, belajar darinya, dan meningkatkannya untuk konsultasi mendatang.

2.2.7 Basis Pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan dalam menyelesaikan masalah, tentu di dalam domain tertentu. Ada dua bentuk pendekatan basis pengetahuan yang umum digunakan, yaitu:

1. Penalaran berbasis aturan (*Rule-Based Reasoning*)

Pengetahuan direpresentasikan dengan menggunakan aturan berbentuk **IF-THEN**. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan si pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu juga digunakan apabila dibutuhkan penjelasan tentang langkah-langkah pencapaian solusi.

2. Penalaran berbasis kasus (*Case-Based Reasoning*)

Basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang sekarang. Bentuk ini akan digunakan apabila *user* menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama. Selain itu, bentuk ini juga digunakan apabila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan (Kusumadewi, 2003: 105).

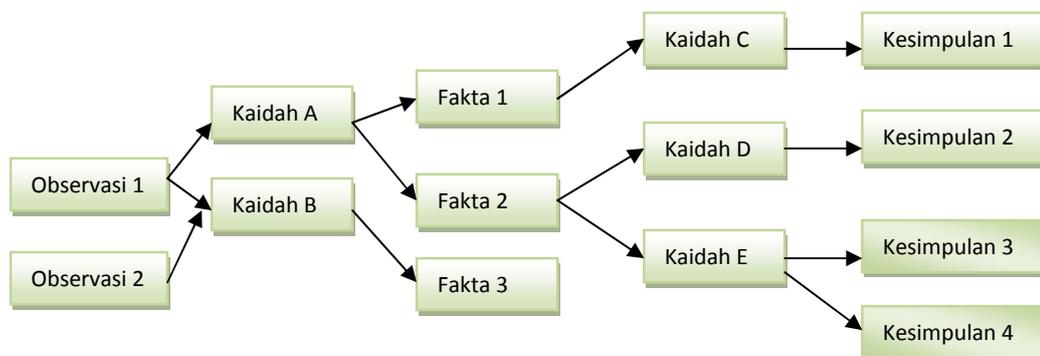
2.2.8 Mekanisme Inferensi

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

merupakan pembuktian hipotesis. Bila hipotesis sudah dimasukkan ke dalam sistem pakar, maka motor inferensi pertama-tama mengecek apakah hipotesis sudah ada dalam basis data atau belum. Jika sudah ada, maka hipotesis dianggap sebagai fakta yang sudah dibuktikan, sehingga operasi tidak perlu dilanjutkan. Ada dua metode inferensi yang paling penting dalam sistem pakar yaitu:

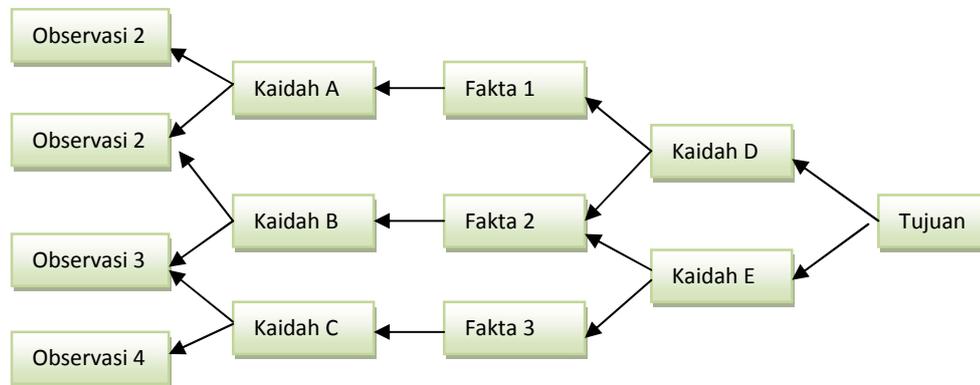
1. Runut Maju (*Forward Chaining*)

Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja, Proses diulang sampai ditemukan suatu hasil (Wilson, 1998) dalam (Pratama, 2015). Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian dan peramalan. Menurut (Arhami, 2005), pelacakan ke depan adalah pendekatan yang dimotori data (*data-driven*). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN. Proses *forward chaining* disajikan pada Gambar 2.11.



Gambar 2.11. Proses *Forward Chaining* (Arhami, 2005)

2. Runut Balik (*Backward Chaining*)



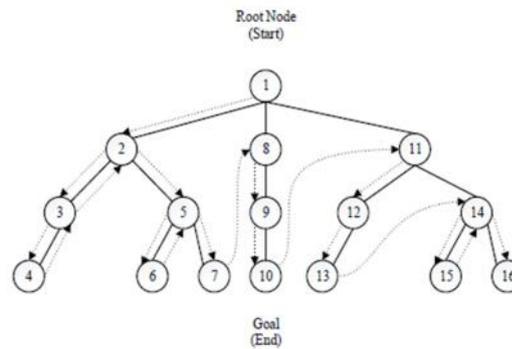
Gambar 2.12. Proses *Backward Chaining* (Arhami, 2005)

Menurut (Giarattano dan Riley, 1994) dalam (Pratama, 2015), runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam runut balik penalaran dimulai dengan tujuan kemudian merunut balik ke jalur yang mengarahkan ke tujuan tersebut. Runut balik disebut juga sebagai *goal-driven reasoning*, merupakan cara yang efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur. Pelacakan ke belakang adalah pendekatan yang dimotori tujuan terlebih dahulu (*goal-driven*). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya.

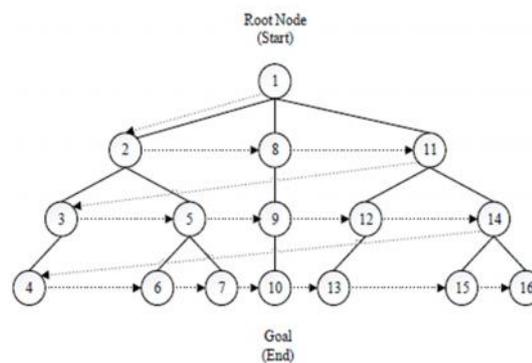
Kedua metode inferensi tersebut dipengaruhi oleh tiga macam penelusuran, yaitu:

1. *Depth-first search*, melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan. Diagram alir *depth-first search* disajikan pada Gambar 2.13.

2. Breadth-first search, bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat selanjutnya. Diagram alir Breadth-first search disajikan pada Gambar 2.14.



Gambar 2.13. Diagram Alir Teknik Penelusuran *Depth-first Search*
(Arhami, 2005)



Gambar 2.14. Diagram Alir Teknik Penelusuran *Breadth-first Search*
(Arhami, 2005)

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

2.2.8.1 Metode *Forward Chaining*

Menurut Sutojo (2011) metode forward chaining merupakan teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari rules IF-THEN. Bila ada fakta yang cocok dengan

bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database. Setiap kali pencocokan berhenti bila tidak ada lagi rule yang bisa dieksekusi.

Contoh:

Misalkan diketahui sistem pakar menggunakan 5 buah rule berikut.

R1 : IF (Y AND D) THEN Z

R2 : IF (X AND B AND E) THEN Y

R3 : IF A THEN X

R4 : IF C THEN L

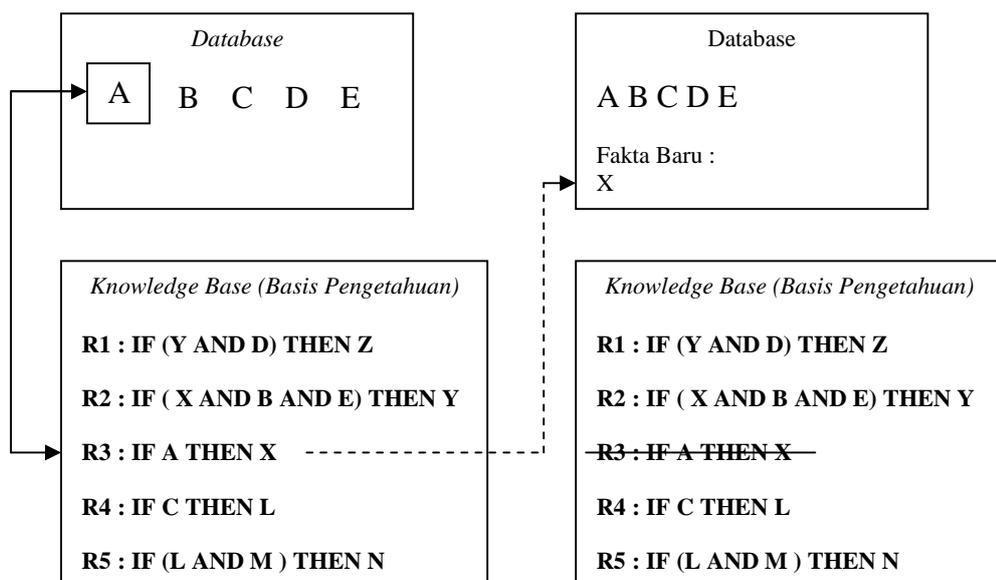
R5 : IF (L AND M) THEN N

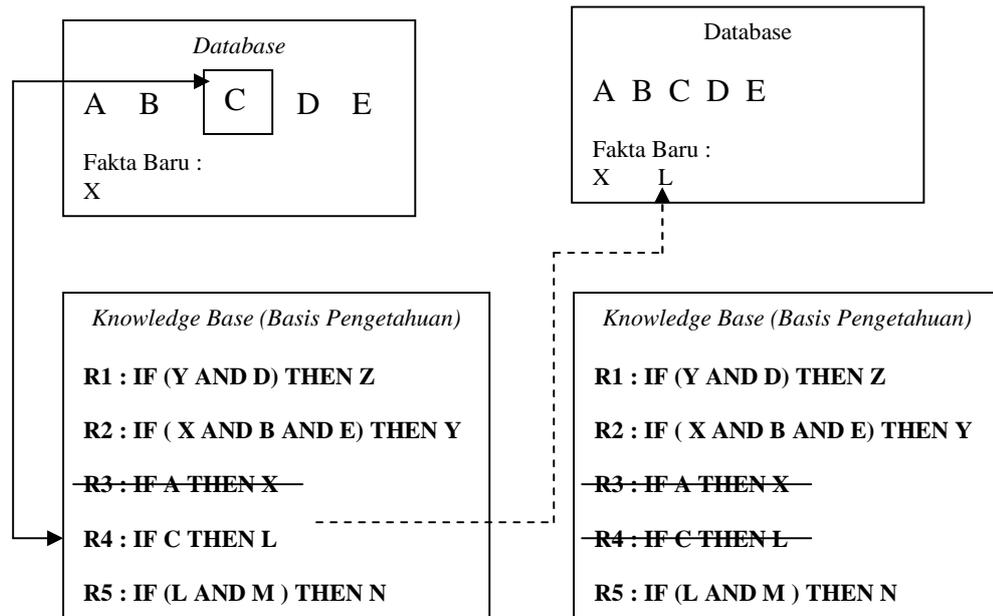
Fakta-fakta : A, B, C, D, dan E bernilai benar

Goal : menentukan apakah Z bernilai benar atau salah

Iterasi ke-1

Iterasi ke-1 dapat dilihat pada Gambar 2.15.

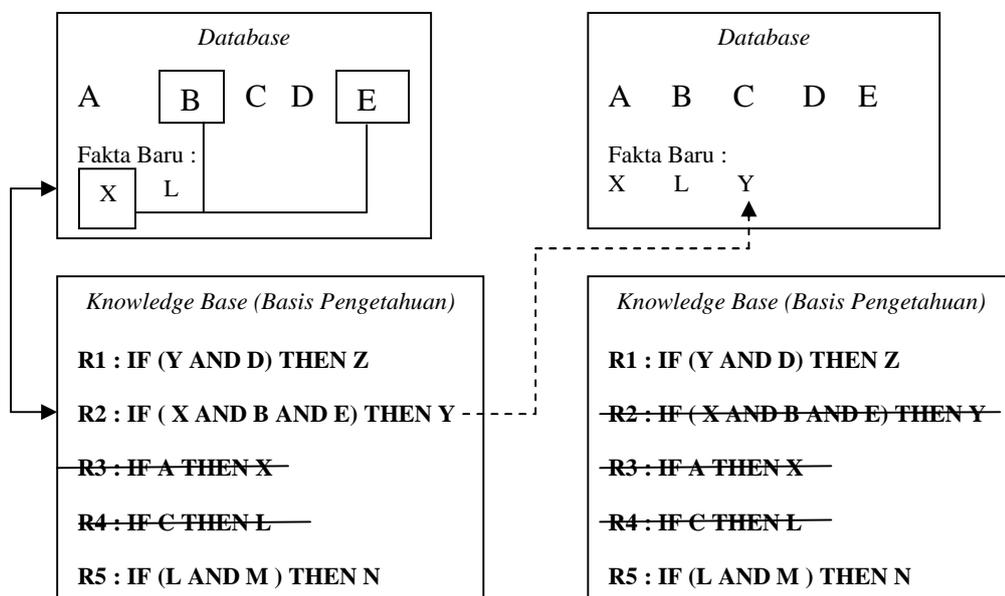




Gambar 2.15. Iterasi ke-1

Iterasi ke-2

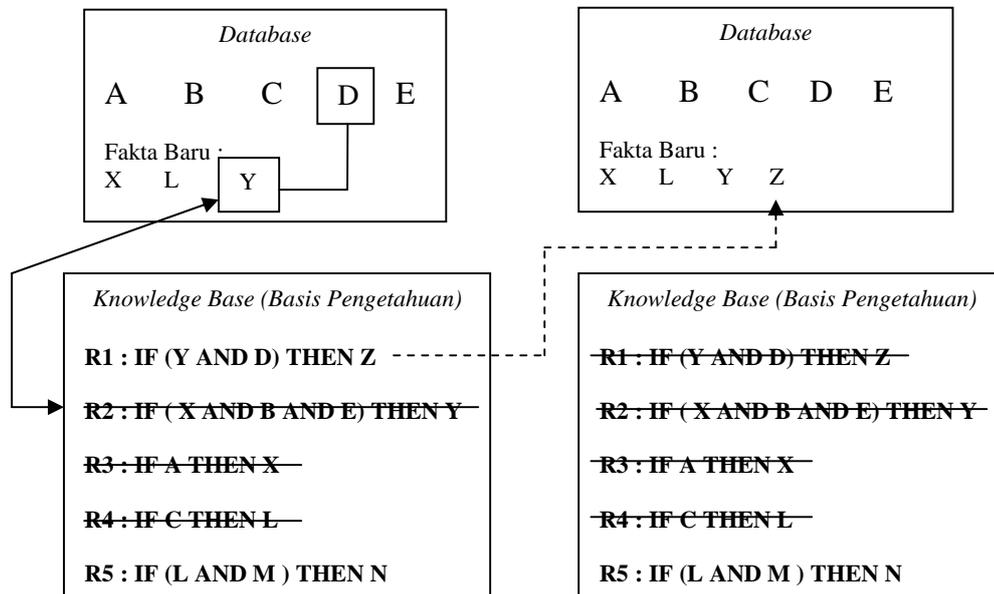
Iterasi ke-2 dapat dilihat pada Gambar 2.16



Gambar 2.16. Iterasi ke-2

Iterasi ke-3

Iterasi ke-3 dapat dilihat pada Gambar 2.17.



Gambar 2.17. Iterasi ke-3

Sampai di sini proses dihentikan karena sudah tidak ada lagi rule yang bisa dieksekusi. Hasil pencarian adalah Z bernilai benar (lihat database di bagian fakta baru).

2.3 Android

2.3.1 Pengertian Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* yang menyertakan *middleware* (*virtual machine*) dan sejumlah aplikasi utama. Android merupakan modifikasi dari kernel Linux (Andry, 2011).

Pada awalnya sistem operasi ini dikembangkan oleh sebuah perusahaan bernama Android, Inc. Dari sinilah awal mula nama Android muncul, Android Inc Adalah sebuah perusahaan *start-up* kecil yang berlokasi di Palo Alto, California, Amerika

Serikat yang didirikan oleh Andy Rubin bersama Rich Miner, Nick Sears, dan Chris White. Pada bulan juli 2005, perusahaan tersebut diakuisisi oleh Google dan para pendirinya bergabung ke Google. Andy Rubin sendiri kemudian diangkat menjadi Wakil Presiden divisi *Mobile* dari Google.

Tujuan pembuatan sistem operasi ini adalah untuk menyediakan *platform* yang terbuka, yang memudahkan orang mengakses internet menggunakan telepon seluler. Android juga dirancang untuk memudahkan pengembang membuat aplikasi dengan batasan yang minim sehingga kreativitas pengembang menjadi lebih berkembang (Andry, 2011).

Sebagai *Open Source* dan bebas dalam memodifikasi, di dalam Android tidak ada ketentuan yang tetap dalam konfigurasi *Software* dan *Hardware*. Fitur- fitur yang didapat dalam Android antara lain (Lee, 2011):

1. *Storage* - Menggunakan SQLite, database yang ringan, untuk sebuah penyimpanan data.
2. *Connectivity* - Mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS.
3. *Bluetooth* (termasuk A2DP dan AVRCP), WiFi, LTE, dan WiMax.
4. *Messaging* –Mendukung SMS dan MMS
5. *Web Browser* – Berbasiskan open-source WebKit, bersama mesin
6. *Chrome's V8 JavaScript*
7. *Media support* – Termasuk mendukung untuk beberapa media berikut :H.263, H.264 (dalam bentuk 3GP or MP4), MPEG-4 SP, AMR, AMRWB (dalam bentuk 3GP), AAC, HE-AAC (dalam bentuk MP4 atau 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, GIF, dan BMP.

8. *Hardware support* – Sensor akselerasi, Kamera, Kompas Digital, Sensor Kedekatan, GPS.
9. *Multi-touch* – Mendukung *multi-touch screens*
10. *Multi-tasking* – Mendukung aplikasi *multi-tasking*
11. *Flash-support* – Android 2.3 mendukung *Flash 10.1*
12. *Tethering* – Mendukung pembagian dari koneksi Internet sebagai *wired/wireless hotspot*
13. *Play store* – Layanan konten digital milik Google yang melingkupi toko daring untuk produk-produk seperti musik/lagu, buku, aplikasi, permainan, ataupun pemutar media.
14. Lingkungan pengembangan yang kaya, termasuk *emulator*, peralatan *debugging*, dan *plugin* untuk Eclipse IDE.

2.3.2 Arsitektur Android

Menurut (Nazrudin, 2012) Arsitektur Android secara garis besar dapat dijelaskan sebagai berikut:

a. Application and Widgets

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

b. *Application Frameworks*

Android adalah “*Open Development Platform*” yaitu Android menawarkan kepada pengembang atau member kemampuan kepada pengembangan untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *servicebackground*, mengatur alarm, dan menambahkan tambahan seperti status *notifications* dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan. Sehingga bisa kita simpulkan *Application Frameworks* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Application Frameworks* adalah sebagai berikut:

1. Views.
2. Content Provider.
3. Resource Manager.
4. Notification Manager.
5. Activity.

c. *Libraries*

Libraries adalah layer dimana fitur-fitur android berada biasanya para pembuat aplikasi kebanyakan mengakses *library* untuk menjalankan aplikasinya berjalan

diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta:

- 1) *Libraries* media untuk pemutar 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 Live Webcore* mencakup modern web *browser* dengan engine embedded web view.
- 7) *Libraries 3D* yang mencakup implementasi OpenGL ES1.0 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 dibangun dalam bahasa Java, sementara *Dalvik* sebagai virtual mesin bukan *Java Virtual Machine*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java/C 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 operating sistem dari Android itu sendiri, berisi file-file sistem yang mengatur sistem *processing memory*, *resources*, *drivers*, dan sistem-sistem operating Android lainnya. Linux Kernel yang digunakan Android adalah Linux Kernel *release 2.6* (Nazruddin, 2012).

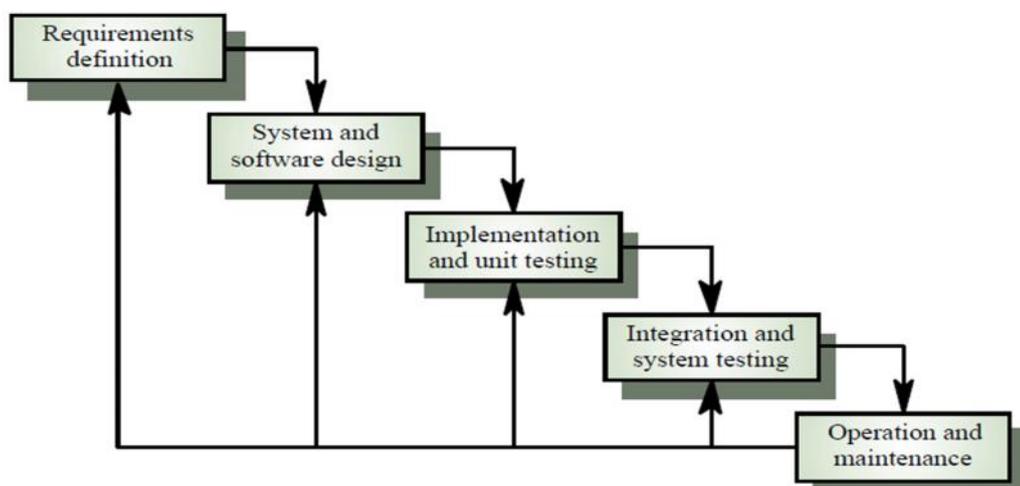
2.4 Metodologi Pengembangan Sistem

Adapun metodologi yang digunakan dalam Penelitian ini yaitu: Metode Waterfall, dan desain menggunakan Unified Modeling Language (UML).

2.4.1 Metode *Waterfall*

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, proses yang berjalan terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian (Pressman, Roger S. 2001).

Tahapan yang dilakukan dalam pengembangan aplikasi ini sesuai dengan Metode Waterfall dapat dilihat pada Gambar 2.18.



Gambar 2.18. Metode Waterfall (Pressman, Roger S. 2001)

Tahapan pengembangan sistem dengan Metode Waterfall dijelaskan sebagai berikut:

1. Analisis Kebutuhan Aplikasi (Requirements Definition)

Analisa kebutuhan sistem atau aplikasi merupakan tahapan pertama yang menjadi dasar proses pembuatan sistem. Pada tahap ini dilakukan pencarian dan pengumpulan semua kebutuhan yang diperlukan untuk menunjang kelengkapan sistem atau aplikasi, kemudian mendefinisikan semua kebutuhan yang dipenuhi dalam perangkat lunak atau aplikasi yang dibuat.

2. Desain Aplikasi (Sistem And Software Design)

Desain aplikasi merupakan tahap perancangan sistem atau aplikasi yang meliputi penyusunan proses, data, aliran proses, dan pemenuhan kebutuhan sesuai dengan hasil analisa kebutuhan. Dokumentasi desain aplikasi yang dihasilkan dari tahapan ini adalah *Use Case Diagram* dan *Activity Diagram*.

3. Penerapan Desain dan Penulisan Kode Program

Penulisan kode program merupakan tahap penerjemahan desain sistem yang telah dibuat ke dalam bentuk perintah-perintah yang dimengerti komputer dengan menggunakan bahasa pemrograman. Penelitian ini menggunakan bahasa pemrograman Java dan Android Studio sebagai software pengembangan aplikasinya. Pada tahap ini, penulis menerjemahkan design kedalam bahasa pemrograman sehingga didapatkan suatu aplikasi yang diinginkan sesuai yang sudah dirancang, sehingga didapatkan suatu file installer dengan ekstensi apk.

4. Pengujian Aplikasi (Integration and System Testing)

Pengujian aplikasi dilakukan untuk memastikan bahwa sistem yang dibuat telah sesuai dengan desain dan semua fungsi dapat dipergunakan dengan baik tanpa ada kesalahan sesuai dengan kebutuhan pengguna. Pengujian aplikasi ini menggunakan metode Blackbox Testing. Pengujian dilakukan secara menyeluruh tanpa melihat struktur internal aplikasi atau komponen yang diuji. Blackbox Testing berfokus pada kebutuhan fungsional aplikasi yang berdasarkan pada spesifikasi kebutuhan aplikasi tersebut.

5. Penerapan Aplikasi dan Perawatan (Operational and Maintenance)

Pada tahapan ini, aplikasi sudah siap untuk diterapkan pada perangkat mobile dan siap digunakan sesuai dengan tujuan dibuatnya aplikasi ini. Perawatan, perbaikan dan pengembangan aplikasi dilakukan untuk menjaga kualitas dan kestabilan aplikasi.

2.4.2 UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software*

Engineering (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan *design* ke dalam empat tahapan iterative, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantic dari hubungan obyek dan kelas tersebut, perincian *interface* dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur pemodelan entity-relationship. Tahapan utama dalam metodologi ini adalah analisis, *design* sistem, *design* obyek, dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OOP. Metode OOSE dari Jacobson lebih memberi penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requirement, analisis, *design*, implementasi, dan model pengujian (*test model*). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Design UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam dari pada metode lainnya (Nugroho, 2010).

UML dideskripsikan oleh beberapa diagram, yaitu:

1. *Use Case* Diagram

Use case Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case* diagram lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur

atau urutan kejadian. Sebuah *use case* diagram merepresentasikan sebuah interaksi antara aktor dengan sistem yang akan dikembangkan (Fowler, 2004).

Komponen-komponen dalam *use case* diagram (Fowler, 2004):

a. Aktor

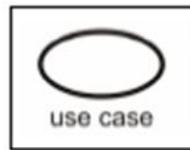
Pada dasarnya aktor bukanlah bagian dari *use case* diagram, namun untuk dapat terciptanya suatu *use case* diagram diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan dengan *stick pan* seperti yang terdapat pada Gambar 2.19.



Gambar 2.19. Contoh Aktor (Fowler, 2004).

b. Use Case

Gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang akan dibangun. Bentuk *use case* dapat terlihat pada Gambar 2.20.



Gambar 2.20. *Use Case* (Fowler, 2004).

Ada beberapa relasi yang terdapat pada *use case* diagram:

1. *Association*, menghubungkan link antar element.
2. *Generalization*, disebut juga pewarisan (*inheritance*), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
3. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi yang mungkin terjadi pada *use case* diagram:

1. *<<include>>*, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. *<<extends>>*, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
3. *<<communicates>>*, merupakan pilihan selama asosiasi hanya tipe relationship yang dibolehkan antara aktor dan *use case*.

2. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat digunakan untuk aktifitas lainnya (Fowler, 2004).

Berikut ini adalah tabel Notasi Activity Diagram yang diilustrasikan pada Tabel 2.2.

Tabel 2.2 Notasi Activity Diagram (Meildy, 2014).

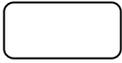
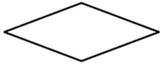
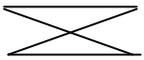
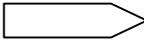
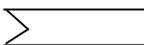
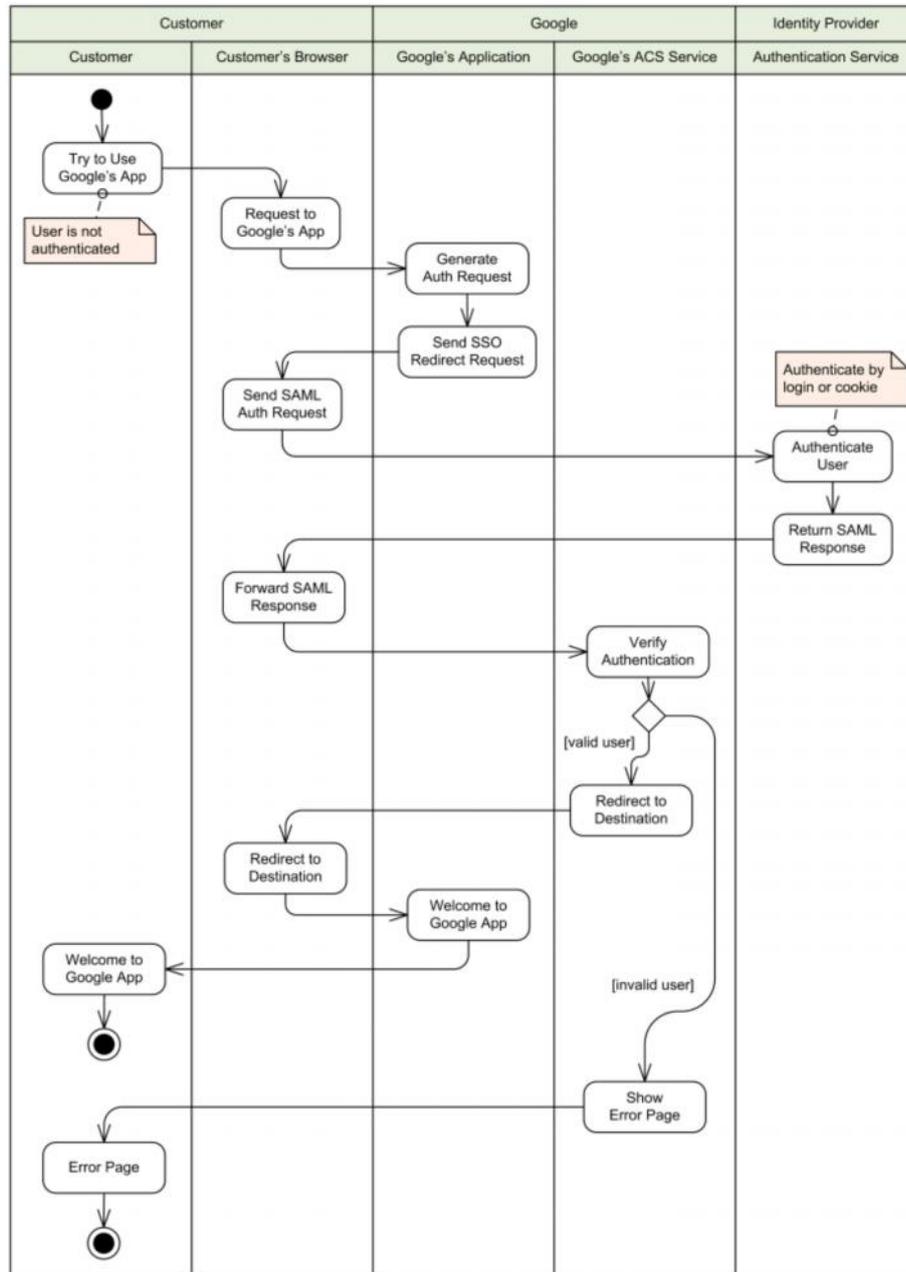
Simbol	Keterangan
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk mengambil keputusan
	<i>Fork</i> ; Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Rake</i> ; Menunjukkan adanya dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir (<i>Flow Final</i>)

Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity*

diagram pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity* diagram juga digunakan untuk menggambarkan interaksi antara beberapa *use case* (Fowler, 2004).

Bentuk dari *activity* diagram dapat terlihat pada Gambar 2.21.



Gambar 2.21. Contoh *Activity* Diagram (uml-diagrams.org, 2014).

3. *Class* Diagram

Class adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class* Diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain (Fowler, 2004).

Class memiliki tiga area pokok:

1. Nama (Class Name)
2. Atribut
3. Metode (Operations)

Pada UML, class digambarkan dengan segi empat yang dibagi beberapa bagian. Bagian atas merupakan nama dari class. Bagian yang tengah merupakan struktur dari class (atribut) dan bagian bawah merupakan sifat dari class (metode/operasi).

Atribut dan metode dapat memiliki salah satu sifat berikut:

1. Private, tidak dapat dipanggil dari luar class yang bersangkutan.
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan class lain yang mewarisinya.
3. Public, dapat dipanggil oleh class lain (Fowler, 2004).

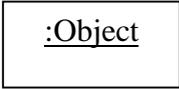
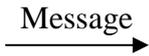
4. *Sequence* Diagram

Sequence diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara

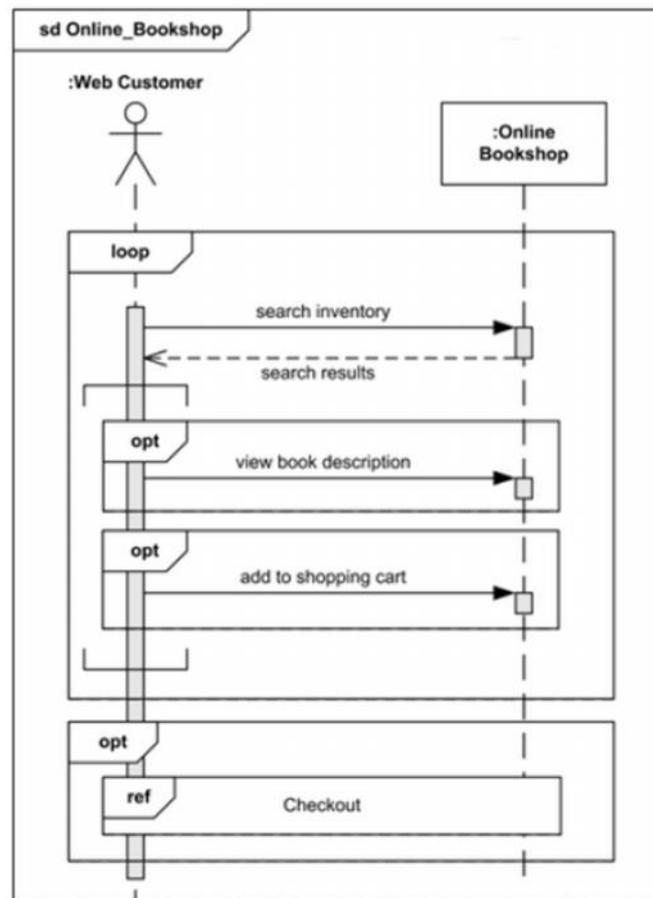
objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem (Fowler, 2014).

Berikut ini adalah Notasi *Sequence Diagram* yang disajikan pada Tabel 2.3.

Tabel 2.3 Notasi *Sequence Diagram* (Meildy, 2014).

Simbol	Nama	Keterangan
	Object	<i>Object</i> merupakan <i>instance</i> dari sebuah class dan dituliskan secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma
	Actor	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>actor</i> sama dengan symbol pada <i>actor Use Case Diagram</i> .
	Lifeline	<i>Lifeline</i> mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertical yang ditarik dari sebuah obyek.
	Activation	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.
	Message	Message, digambarkan dengan anak panah <i>horizontal</i> antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara <i>object-object</i> .

Contoh dari *sequence* diagram dapat dilihat pada Gambar 2.22.



Gambar 2.22. Contoh *Sequence* Diagram (uml-diagrams.org, 2014)

2.4.3 Keunggulan UML

Secara umum UML diterapkan dalam pengembangan sistem/perangkat lunak berorientasi obyek sebab metodologi UML ini umumnya memiliki keunggulan-keunggulan sebagai berikut (Nugroho, 2010):

a. *Uniformity*

Dengan metodologi UML, para pengembang cukup menggunakan satu metodologi dari tahap analisis hingga perancangan. Hal ini tidak bisa dilakukan dalam metodologi pengembangan terstruktur. Dengan perkembangan masa kini ke

arah aplikasi GUI (*Graphical User Interface*), UML juga memungkinkan kita merancang komponen antarmuka pengguna (*user interface*) secara integrasi bersama dengan perancangan perangkat lunak sekaligus dengan perancangan basis data.

b. Understandability

Dengan metodologi ini kode yang dihasilkan dapat diorganisasi ke dalam kelas-kelas yang berhubungan dengan masalah sesungguhnya sehingga lebih mudah dipahami siapapun juga.

c. Stability

Kode program yang dihasilkan relatif stabil sepanjang waktu sebab sangat mendekati permasalahan sesungguhnya di lapangan.

d. Reusability

Dengan metodologi berorientasi obyek, dimungkinkan penggunaan ulang kode, sehingga pada gilirannya akan sangat mempercepat waktu pengembangan perangkat lunak.

2.5 Teknik Pengujian Perangkat Lunak

Ada dua macam pendekatan kasus uji yaitu *white-box* dan *black-box*. Pendekatan *white-box* adalah pengujian untuk memperlihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya (Jiang, 2012). Jalur logika perangkat lunak akan dites dengan menyediakan kasus uji yang akan mengerjakan kumpulan kondisi dan pengulangan secara spesifik. Sehingga melalui penggunaan metode ini akan dapat memperoleh kasus uji yang menjamin bahwa semua jalur independen pada suatu model telah digunakan minimal satu kali, penggunaan

keputusan logis pada sisi benar dan salah, pengekseskusion semua *loop* dalam batasan dan batas operasional perekayasa, serta penggunaan struktur data internal guna menjamin validitasnya (Pressman, 2010).

Pendekatan *black-box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Jiang, 2012). Kasus uji ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik pengujian ini berfokus pada domain informasi dari perangkat lunak, yaitu melakukan kasus uji dengan mempartisi domain *input* dan *output* program. Metode *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, dan inisialisasi dan kesalahan terminal (Pressman, 2010).

2.5.1 Equivalence Partitioning

Equivalence Partitioning (EP) merupakan metode *black box testing* yang membagi domain masukan dari program kedalam kelas-kelas sehingga *test case* dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *Equivalence Partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat

berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi *Boolean* (Pressman, 2001).

2.5.2 Probabilitas Klasik

Menurut Arhami (2005: 135) probabilitas merupakan suatu cara kuantitatif yang berhubungan dengan ketidakpastian yang telah ada. Probabilitas klasik disebut juga *a priori probability* karena berhubungan dengan suatu permainan (*games*) atau sistem. Seperti yang telah disebutkan sebelumnya, istilah *a priori* berarti “sebelum”

(Arhami, 2005:138). Probabilitas ini dianggap sebagai suatu jenis permainan seperti pelemparan dadu, permainan kartu, dan pelemparan koin.

Rumus umum untuk probabilitas klasik di definisikan sebagai peluang $P(A)$ dengan n adalah banyaknya kejadian, $n(A)$ merupakan banyaknya hasil mendapatkan A .

Frekuensi relative terjadinya A adalah $\frac{n(A)}{n}$ maka (Arhami, 2005:138) :

$$P(A) = \frac{n(A)}{n} \dots\dots\dots(Persamaan 1)$$

Probabilitas klasik ini digunakan untuk mendapatkan peluang kemungkinan penyakit, sehingga untuk menghitung persentase penyakit adalah :

$$\text{Persentase (A)} = P(A) \times 100\% \dots\dots\dots(Persamaan 2)$$

Tabel persentase penyakit disajikan pada Tabel 2.4.

Tabel 2.4. Tabel Kemungkinan Penyakit

Kondisi	Persentase
Pasti Tidak	<10%
Tidak Tahu	10-19%
Hampir Mungkin	20 – 39%
Mungkin	40 – 59%
Kemungkinan Besar	60 - 79%
Hampir Pasti	80 – 99%
Pasti	100%

2.5.3 Skala *Likert*

Menurut Likert dalam buku Azwar S (2011, p. 139), sikap dapat diukur dengan metode rating yang dijumlahkan (*Method of Summated Ratings*). Metode ini merupakan metode penskalaan pernyataan sikap yang menggunakan distribusi respons sebagai dasar penentuan nilai skalanya. Nilai skala setiap pernyataan tidak ditentukan oleh derajat *favourable* nya masing-masing akan tetapi ditentukan oleh distribusi *respons* setuju dan tidak setuju dari sekelompok responden yang bertindak sebagai kelompok uji coba (*pilot study*) (Azwar, 2011).

Skala Likert, yaitu skala yang berisi lima tingkat preferensi jawaban dengan pilihan sebagai berikut: 1 = sangat tidak setuju; 2 = tidak setuju; 3 = ragu-ragu atau netral; 4 = setuju; 5 = sangat setuju. Selanjutnya, penentuan kategori interval tinggi, sedang, atau rendah digunakan rumus sebagai berikut :

$$I = \frac{NT - NR}{K}$$

Keterangan :

I = Interval;

NT = Total nilai tertinggi;

NR = Total nilai terendah;

K = Kategori jawaban

Kriteria penilaian = % Total skor tertinggi – Interval (I)

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam dan Jurusan Budidaya Perairan Fakultas Pertanian Universitas Lampung. Waktu penelitian dilakukan pada bulan September 2015 sampai dengan bulan Maret 2016.

3.2 Perangkat Penelitian

Alat pendukung yang digunakan dalam penelitian ini adalah sebagai berikut:

A. Perangkat Keras

- Notebook dengan spesifikasi :

Processor : Intel® Core™ i5-3317 CPU @1.70GHz

RAM : 4096 MB

Harddisk : 500 GB

B. Perangkat Lunak

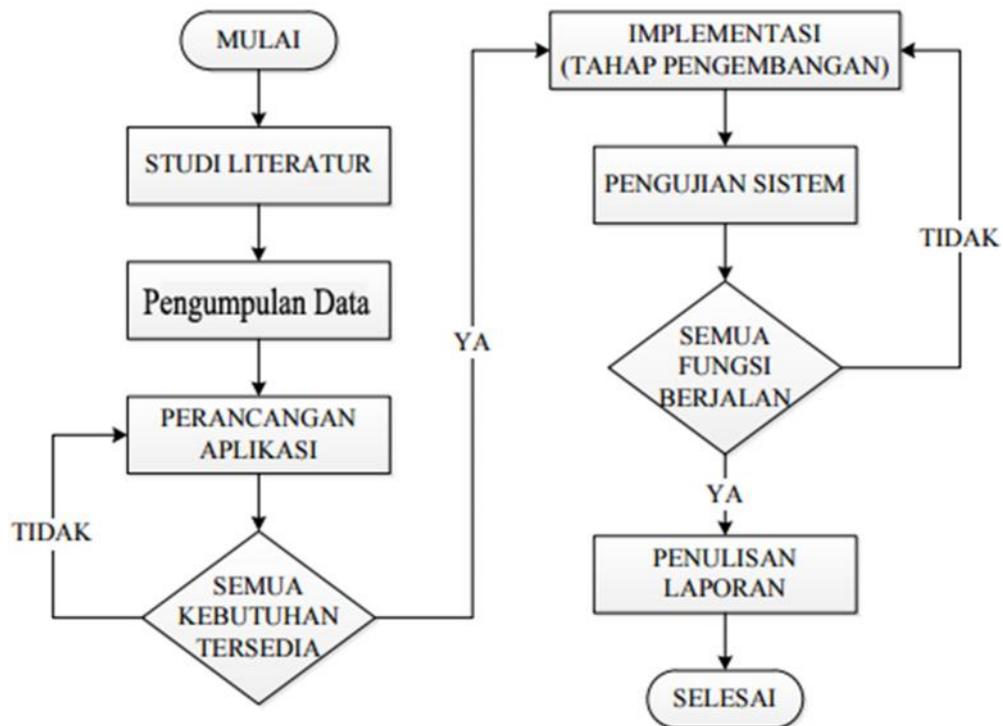
- Sistem Operasi Windows 7 64 Bit
- *Text Editor* Notepad++

- Adobe Photoshop CS 6

- Android Studio

3.3 Tahapan Penelitian

Tahapan penelitian yaitu tahapan yang akan dilakukan peneliti untuk mempermudah dalam melakukan penelitian. Desain penelitian sistem pakar penyakit ikan air tawar dengan metode *forward chaining* digambarkan pada Gambar 3.1.



Gambar 3.1. Diagram Alir Penelitian

3.3.1 Studi Literatur

Studi literatur dilakukan dengan cara mempelajari aspek-aspek yang berkaitan dengan penelitian ini, diantaranya adalah mencari jenis-jenis penyakit ikan air tawar, gejala-gejala dari penyakit tersebut dan solusinya, mempelajari metode

forward chaining. Data-data yang digunakan dalam studi literatur didapat dengan cara mengumpulkan jurnal, penelusuran internet dan buku yang berkaitan dengan topik.

3.3.2 Pengumpulan Data

Tahapan ini mengumpulkan data-data terkait penyakit ikan budidaya air tawar. Data yang dikumpulkan berasal dari hasil konsultasi dengan pakar dan literatur-literatur tentang penyakit ikan budidaya air tawar. Data-data yang dikumpulkan disusun menjadi basis aturan yang akan digunakan dalam sistem pakar.

3.3.3 Perancangan Sistem

Perancangan sistem adalah tahap setelah analisis dari siklus pengembangan sistem. Perancangan sistem di sini berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan. Tahap ini termasuk mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah dilakukan instalasi akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem.

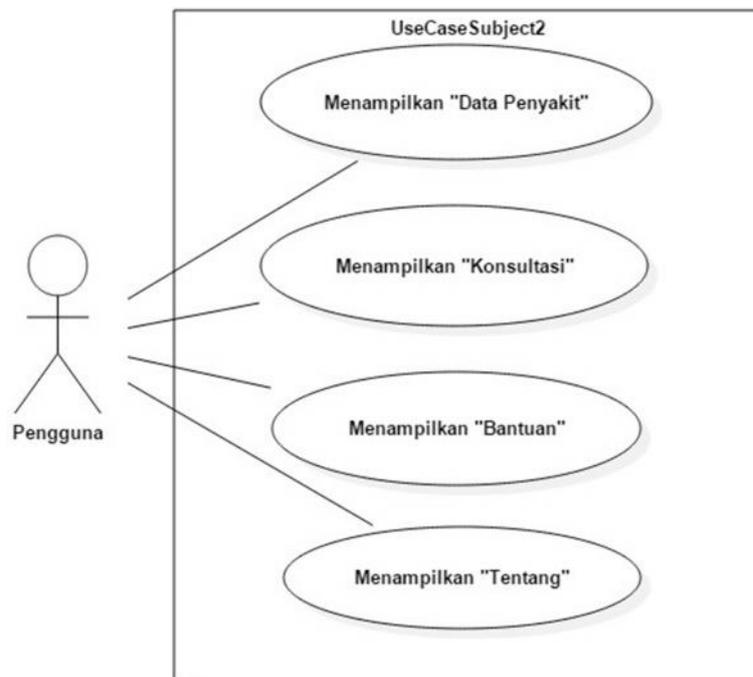
3.3.3.1 Perancangan UML (Unified Modelling Language)

Pemodelan (modeling) adalah tahap merancang perangkat lunak sebelum melakukan tahap pembuatan program (koding). Pada penelitian ini, perancangan

sistem dilakukan dengan memodelkan permasalahan dalam bentuk diagram-diagram UML sebagai berikut.

1. Use Case Diagram

Use case Diagram dibawah ini menggambarkan sistem dari sudut pandang pengguna sistem tersebut (user), sehingga pembuatan *use case* diagram ini lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Pada aplikasi ini pengguna dapat melakukan 4 interaksi antara lain Data Penyakit, Konsultasi, Bantuan, Tentang aplikasi. *Use case* diagram aplikasi Sistem Pakar Diagnosa Penyakit Ikan dapat dilihat pada Gambar 3.2.



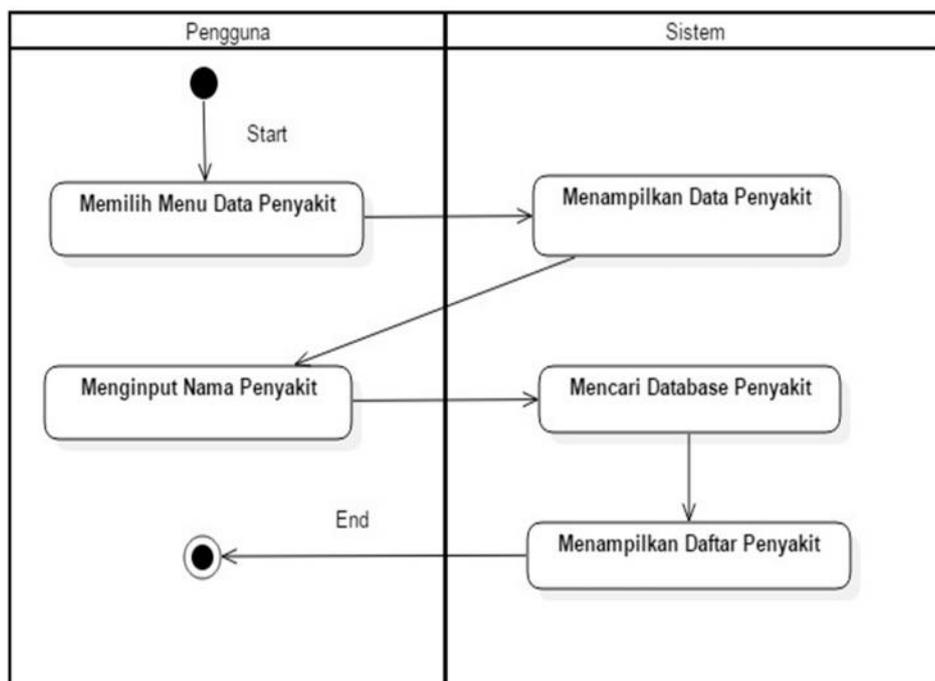
Gambar 3.2. *Use Case Diagram*

2. Activity Diagram

Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam satu operasi sehingga dapat juga untuk aktivitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pada aplikasi Sistem Pakar Diagnosa Penyakit Ikan terdapat 4 (empat) *activity* diagram yaitu sebagai berikut:

a. Activity Diagram Data Penyakit

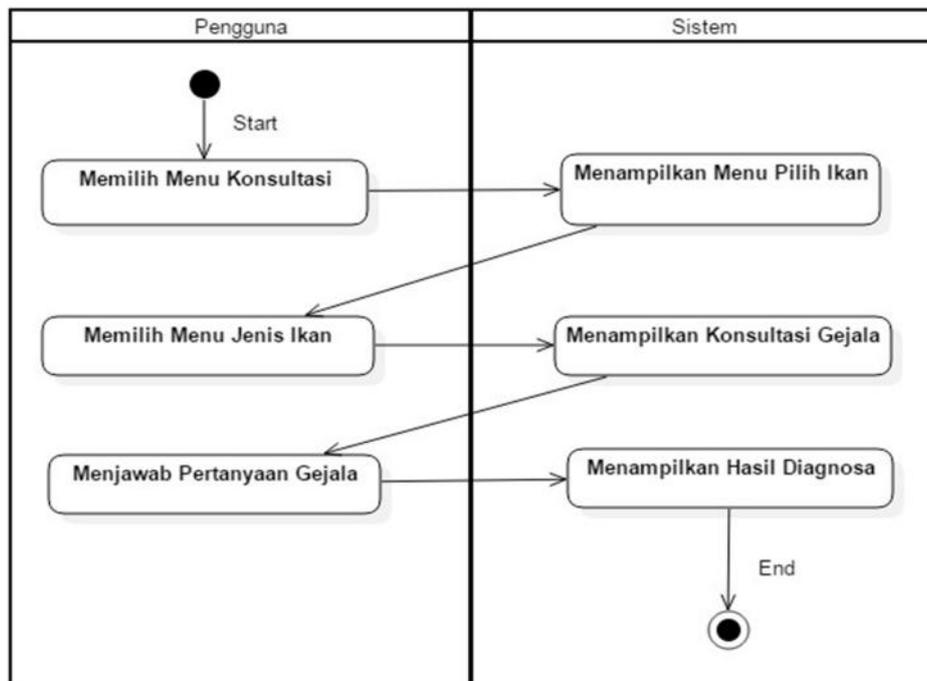
Activity diagram Data Penyakit dimulai dengan pengguna memilih menu “Data Penyakit”, kemudian sistem akan menampilkan daftar penyakit ikan air tawar. Pengguna dapat mengetikkan nama Penyakit yang akan dicari, jika sudah pilih button “cari”. *Activity* diagram Data Penyakit dilihat pada Gambar 3.3.



Gambar 3.3. Activity Diagram Data Penyakit

b. Activity Diagram Konsultasi

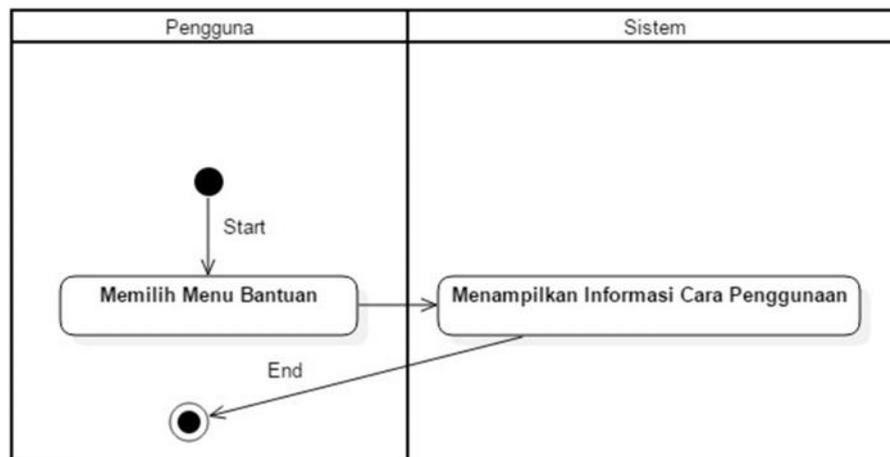
Activity diagram Konsultasi dimulai dengan pengguna memilih menu Konsultasi, selanjutnya terdapat menu pilih ikan. Kemudian pilih menu jenis ikan yang akan di diagnosa, sistem akan menampilkan kolom konsultasi gejala dan pengguna menjawab pertanyaan gejala penyakit ikan. Sistem mencari database penyakit ikan dan menampilkan hasil diagnosa. Activity diagram Konsultasi disajikan pada Gambar 3.4.



Gambar 3.4. Activity Diagram Konsultasi

c. Activity Diagram Bantuan

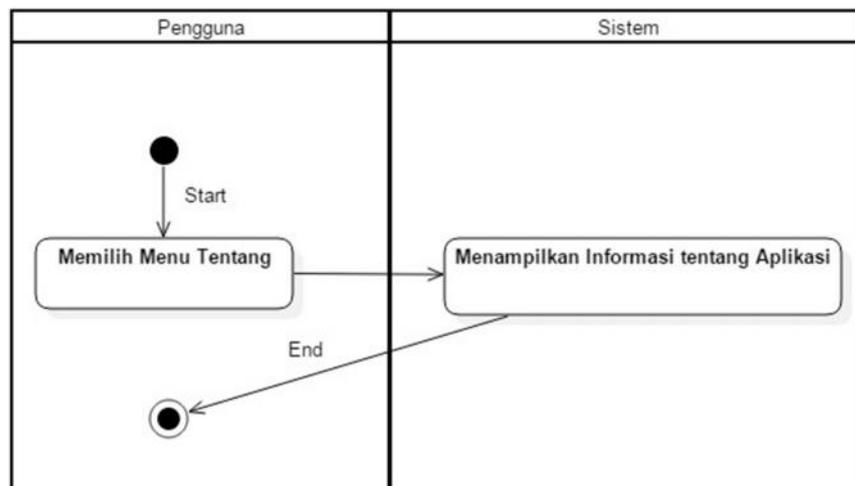
Activity diagram bantuan dimulai dengan pengguna memilih menu “Bantuan”, kemudian sistem akan menampilkan informasi yang berkaitan dengan cara penggunaan aplikasi Sistem Pakar Diagnosa Penyakit Ikan. Activity diagram bantuan disajikan pada Gambar 3.5.



Gambar 3.5. *Activity Diagram Bantuan*

d. *Activity Diagram Tentang*

Activity diagram tentang aplikasi dimulai dengan pengguna memilih menu “Tentang”, kemudian sistem menampilkan informasi mengenai aplikasi Sistem Pakar Diagnosa Penyakit Ikan. *Activity* diagram tentang dapat dilihat pada Gambar 3.6.



Gambar 3.6. *Activity Diagram Tentang*

3. *Sequence* Diagram

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Pada aplikasi Sistem Pakar Diagnosa Penyakit Ikan terdapat 4 (empat) *sequence* diagram yaitu sebagai berikut:

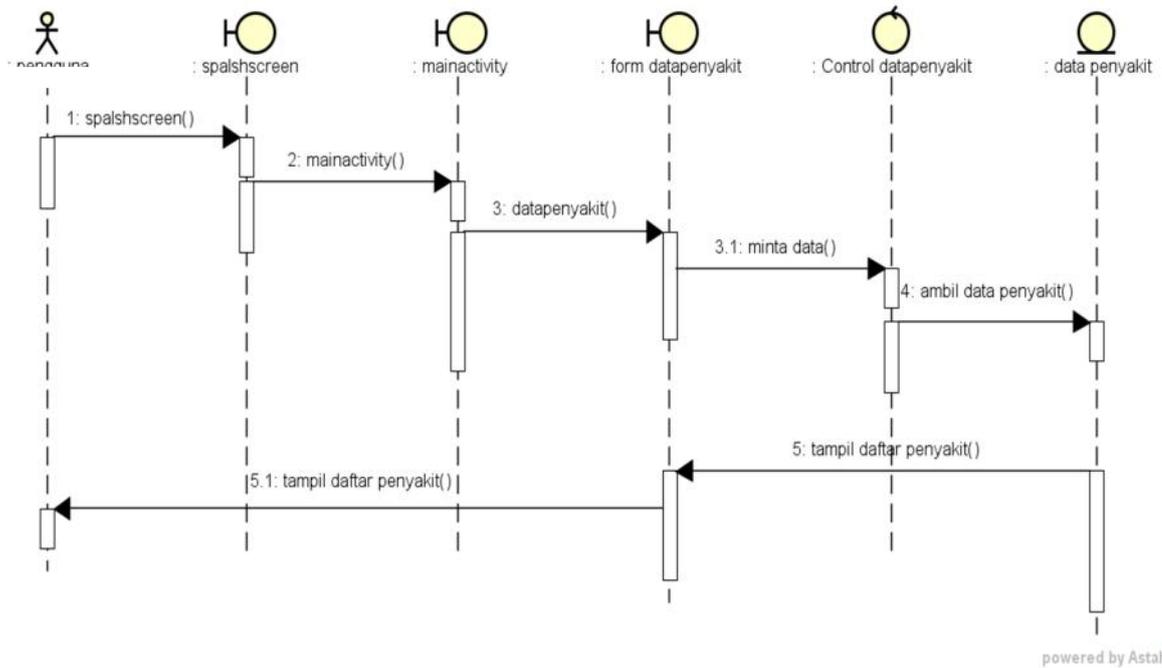
a. *Sequence* Diagram Data Penyakit

Untuk dapat memilih menu “Data Penyakit ” pengguna harus memilih aplikasi Sistem Pakar Diagnosa Penyakit Ikan, kemudian otomatis akan menuju halaman splash screen, selanjutnya akan muncul menu utama aplikasi, dan pengguna dapat memilih menu “Data Penyakit”, maka sistem akan menampilkan daftar penyakit ikan air tawar. Kemudian pengguna dapat mengetikkan nama penyakit yang akan dicari lalu sistem akan otomatis mencari data penyakit yang telah disimpan di database kemudian menampilkan data penyakit ikan sesuai perintah yang diketikkan sebelumnya. Untuk lebih jelasnya *sequence* diagram Data Penyakit dapat dilihat pada Gambar 3.7.

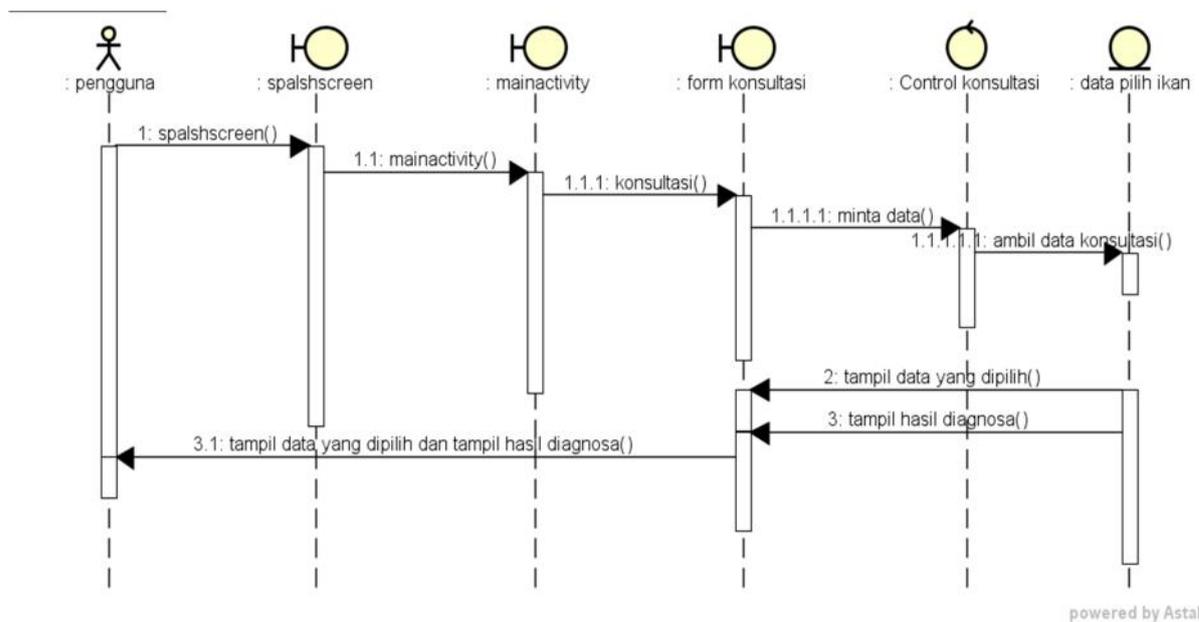
b. *Sequence* Diagram Konsultasi

Untuk dapat memilih menu “Konsultasi” pengguna harus memilih aplikasi Sistem Pakar Diagnosa Penyakit Ikan, kemudian otomatis akan menuju halaman splash screen, selanjutnya akan muncul menu utama aplikasi, dan pengguna dapat memilih menu “Konsultasi”, maka sistem akan menampilkan menu pilih ikan dan memilih menu jenis ikan yang akan diagnosa , lalu menjawab pertanyaan gejala penyakit ikan dan sistem akan memproses database kemudian menampilkan hasil

diagnosa penyakit, gejala dan solusi. *Sequence diagram* Konsultasi dapat dilihat pada Gambar 3.8.



Gambar 3.7. *Sequence Diagram* Data Penyakit



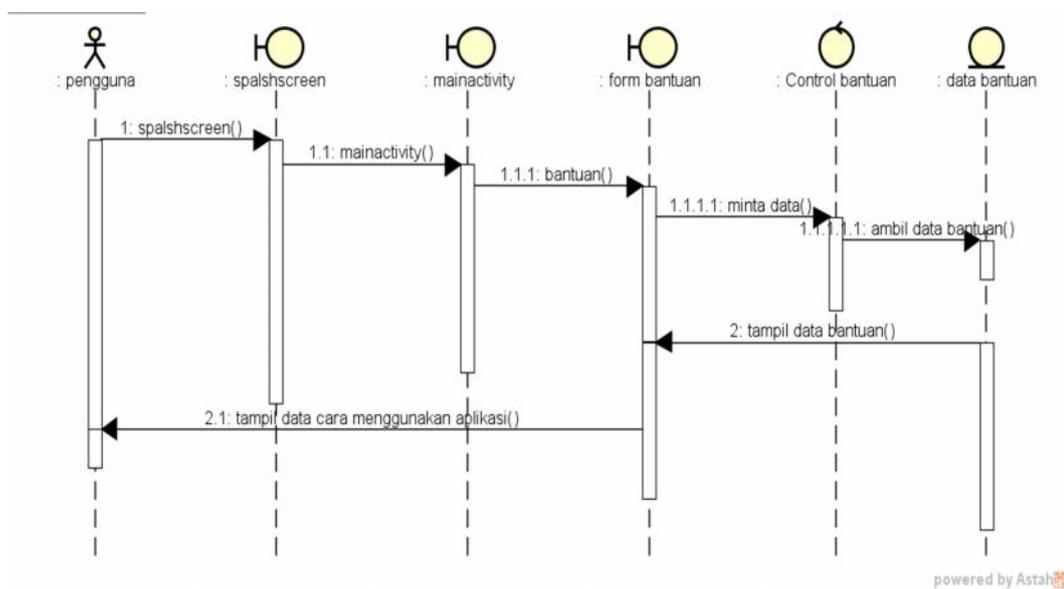
Gambar 3.8. *Sequence Diagram* Konsultasi

c. *Sequence Diagram Bantuan*

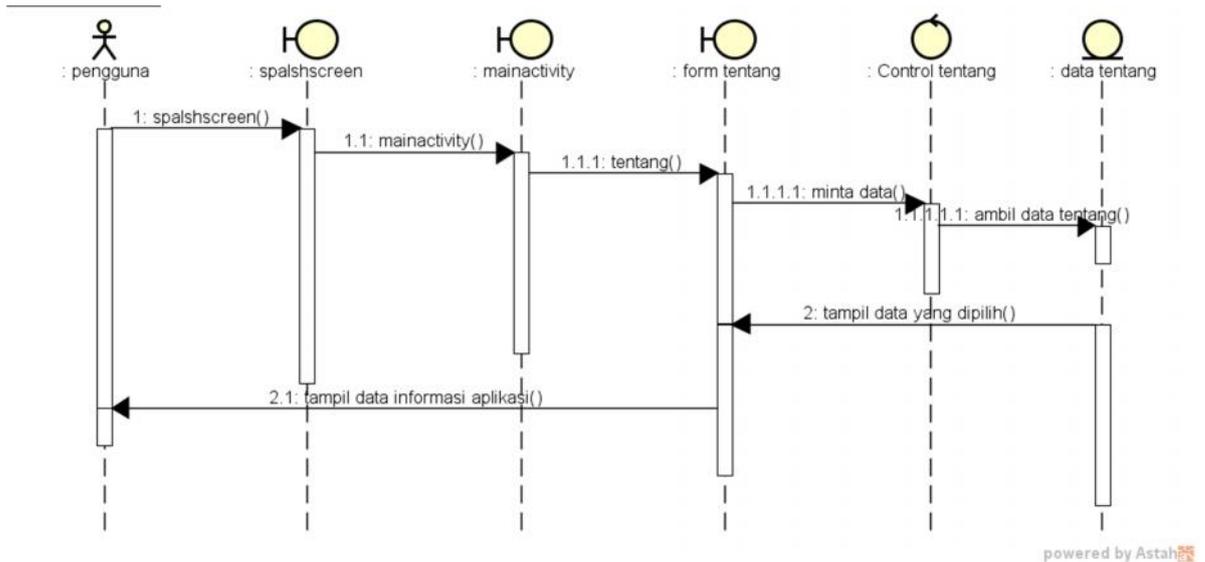
Ketika pengguna sudah berada di menu utama aplikasi, pengguna dapat memilih menu “Bantuan” untuk mengetahui informasi mengenai cara penggunaan aplikasi, maka sistem akan menampilkan informasi tersebut. *Sequence diagram* bantuan disajikan pada Gambar 3.9.

d. *Sequence Diagram Tentang*

Ketika pengguna sudah berada di menu utama aplikasi, pengguna dapat memilih menu “Tentang” untuk mengetahui informasi mengenai informasi aplikasi, maka sistem akan menampilkan informasi tentang aplikasi Sistem Pakar Diagnosa Penyakit Ikan. *Sequence diagram* tentang aplikasi disajikan pada Gambar 3.10.



Gambar 3.9. *Sequence Diagram Bantuan*



Gambar 3.10. *Sequence Diagram* Tentang

3.3.3.2 Perancangan Antarmuka

Perancangan antarmuka merupakan proses penggambaran bagaimana sebuah tampilan (interface) sistem dibentuk. Aplikasi Sistem Pakar Diagnosa Penyakit Ikan dirancang dengan tampilan yang *user friendly*, sehingga diharapkan dapat mempermudah pengguna dalam menggunakan aplikasi ini. Dalam Aplikasi ini terdapat beberapa layout atau form antara lain :

1. Layout *Splash Screen*

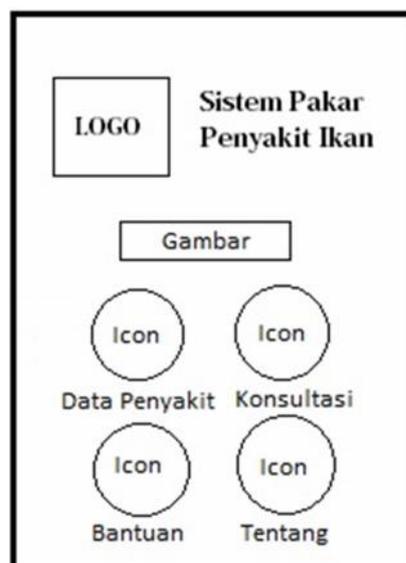
Splash Screen adalah *form* yang ditampilkan diawal ketika aplikasi/program dijalankan. Aplikasi Sistem Pakar Diagnosa Penyakit Ikan menggunakan *splash screen* yang muncul sepersekian detik pada saat pertama membuka aplikasi Sistem Pakar Diagnosa Penyakit Ikan. *Splash screen* di sini dimaksudkan sebagai estetika untuk menunjukan identitas aplikasi saja, tanpa fungsi lainnya. Perancangan layout *splash screen* aplikasi dapat dilihat pada Gambar 3.11.



Gambar 3.11. *Design Layout Splash Screen*

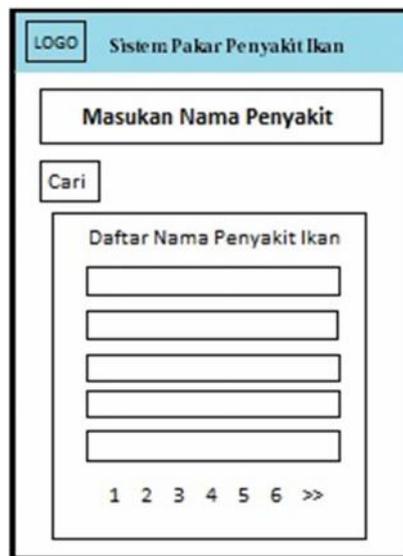
2. Layout Menu Utama

Menu utama berisikan menu-menu pilihan yang dapat digunakan oleh pengguna. Menu yang terdapat pada menu utama antara lain : menu Data Penyakit, menu Konsultasi, menu Bantuan, dan menu Tentang aplikasi. Perancangan layout menu utama aplikasi dapat dilihat pada Gambar 3.12.



Gambar 3.12. *Design Layout Menu Utama*

3. Layout Menu Data Penyakit



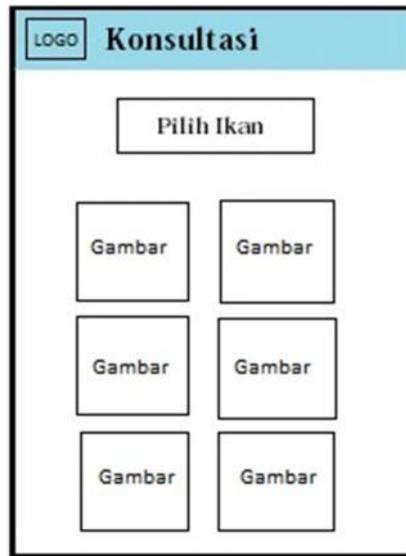
The image shows a software interface for a fish disease expert system. At the top, there is a header bar with a 'LOGO' placeholder and the title 'Sistem Pakar Penyakit Ikan'. Below the header, there is a form titled 'Masukan Nama Penyakit' with a text input field. To the left of this form is a 'Cari' button. Below the input field is a section titled 'Daftar Nama Penyakit Ikan' containing five empty text input fields. At the bottom of this section, there are page navigation controls: the numbers '1 2 3 4 5 6' followed by '>>'.

Gambar 3.13. *Design Layout* Data Penyakit

Pada Gambar 3.13 perancangan layout menu Data Penyakit dapat dilihat ketika pengguna memilih menu ini, pengguna dapat langsung melihat daftar nama penyakit ikan dan mencari jenis penyakit ikan, dengan menekan kolom teks search pengguna harus memasukkan nama penyakit kemudian tekan tombol “cari” untuk melakukan perintah.

4. Layout Menu Konsultasi

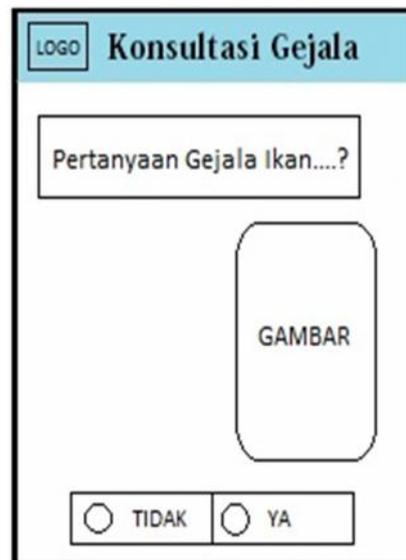
Ketika pengguna memilih menu “Konsultasi”, maka akan muncul sub menu dari Konsultasi yaitu jenis ikan. Pengguna dapat memilih salah satu menu sesuai dengan jenis ikan yang akan di diagnosa. Perancangan layout menu Konsultasi dapat dilihat pada Gambar 3.14.



Gambar 3.14. *Design Layout* Konsultasi

Sub-sub menu pada layout “Konsultasi” memiliki rincian sebagai berikut:

a. Layout Sub Menu Konsultasi Gejala



Gambar 3.15. *Design Layout* Sub Menu Konsultasi Gejala

Pada Gambar 3.15 perancangan *layout* sub menu ini pengguna/pembudidaya menjawab pertanyaan yang disediakan, kemudian jawaban pengguna disimpan untuk di proses. Data digunakan untuk menampilkan pertanyaan yang terkait

dengan jawaban yang diberikan pengguna dan menyimpan aturan analisa kemungkinan penyakit yang diderita ikan pengguna/pembudidaya. Pengguna tinggal menekan button “YA” atau “TIDAK” untuk menjawab beberapa pertanyaan yang diajukan oleh sistem.

b. Layout Sub Menu Hasil Diagnosa Penyakit

Pada sub menu ini menampilkan hasil diagnosa penyakit yang diderita dan memberikan informasi tentang penyakit tersebut, serta memberikan solusi penanganannya. Pengguna dapat melihat apa penyakit yang di derita ikan dan cara penanganannya secara rinci dengan menekan tulisan yang bergaris bawah. Perancangan *layout* Sub Menu Hasil Diagnosa Penyakit dan penjelasan secara rinci tentang solusi penanganannya dapat dilihat pada Gambar 3.16 dan Gambar 3.17.



Gambar 3.16. *Design Layout* Sub Menu Hasil Diagnosa Penyakit



Gambar 3.17. *Design Layout* Hasil Diagnosa Penyakit (Penjelasan Penyakit)

5. Layout Menu Bantuan

Ketika pengguna memilih menu “Bantuan” pengguna akan melihat informasi mengenai cara penggunaan aplikasi Sistem Pakar Diagnosa Penyakit Ikan. Perancangan *layout* menu Bantuan dapat dilihat pada Gambar 3.18.



Gambar 3.18. *Design Layout* Bantuan

6. Layout Menu Tentang

Ketika pengguna memilih menu “Tentang” pengguna akan melihat informasi mengenai aplikasi Sistem Pakar Diagnosa Penyakit Ikan. Pada layout ini terdapat 3 (tiga) Button yang dapat pengguna gunakan untuk memberikan kritik dan saran melalui email developer, google play dan fanpage developer. Perancangan *layout* menu Tentang dapat dilihat pada Gambar 3.19.



Gambar 3.19. *Design Layout* Tentang

3.4 Metode Pengujian Sistem

Pengujian sistem dimaksudkan untuk menguji semua elemen–elemen perangkat lunak yang dibuat apakah sudah sesuai dengan yang diharapkan. Pendekatan kasus uji dalam penelitian ini adalah pengujian *black box* dengan metode *Equivalence Partitioning* (EP). Pengujian ini dilakukan dengan membagi domain masukan dari program ke dalam kelas-kelas sehingga *test case* dapat diperoleh. EP berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. EP didasarkan

pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang akan diperlakukan sama (ekuivalen) oleh komponen tersebut. Pada pengujian ini harus diyakinkan bahwa masukan yang sama akan menghasilkan respon yang sama pula. Alasan menggunakan metode EP pada pengujian aplikasi Sistem Pakar Penyakit Ikan ini adalah karena metode ini dapat digunakan untuk mencari kesalahan pada fungsi yang diberikan ke aplikasi dan dapat mengetahui kesalahan pada *interface* aplikasi sehingga dapat mengurangi masalah terhadap nilai masukan.

Berikut ini merupakan rancangan daftar pengujian yang disajikan pada Tabel 3.1.

Tabel 3.1 Tabel Daftar Pengujian

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Versi Android	Pengujian kompatibilitas versi <i>operatif system</i> android	Pengujian pada android versi 4.0 (<i>Ice Cream Sandwich</i>)	Kompatibel dengan android versi 4.0 (<i>Ice Cream Sandwich</i>)
			Pengujian pada android versi 4.1 (<i>Jelly Bean</i>)	Kompatibel dengan android versi 4.1 (<i>Jelly Bean</i>)
			Pengujian pada android versi 4.4 (<i>Kit-Kat</i>)	Kompatibel dengan android versi android versi 4.4 (<i>Kit-Kat</i>)

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
			Pengujian pada android Versi 5.0 (<i>Lollipop</i>)	Kompatibel dengan android Versi 5.0 (<i>Lollipop</i>)
2	Resolusi Layar dan intensitas Layar	Pengujian Resolusi Layar dan Densitas Layar pada android	Pengujian pada android dengan resolusi 4 inch	Tampilan terlihat baik pada android dengan resolusi 4 inch
			Pengujian pada android dengan resolusi 4,5 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 4,5 inch
			Pengujian pada android dengan resolusi 4,7 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 4,7 inch
			Pengujian pada android dengan resolusi 5 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 5 inch
			Pengujian pada android dengan resolusi 5,5 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 5,5inch

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
3	<i>User Interface</i>	Pengujian pada <i>icon</i> PAKAR IKAN	Klik <i>icon</i> Pakar Ikan pada perangkat android pengguna	Menampilkan <i>layout splash screen</i>
		Pengujian pada menu utama PAKAR IKAN	Klik tombol menu “Data Penyakit”	Menampilkan layout Data Penyakit
		Pengujian pada menu utama PAKAR IKAN	Klik tombol menu “Konsultasi”	Menampilkan <i>layout</i> Konsultasi
		Pengujian pada menu utama PAKAR IKAN	Klik tombol menu “Bantuan”	Menampilkan <i>layout</i> Bantuan
		Pengujian pada menu utama PAKAR IKAN	Klik tombol menu “Tentang”	Menampilkan <i>layout</i> Tentang
4	Fungsi layout Data Penyakit	Pengujian Pencarian database Penyakit Ikan	Klik tombol menu “Cari Penyakit”	Menampilkan kolom teks search dan layout masukan

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
			Klik tombol "Cari"	Menampilkan daftar nama penyakit
5	Fungsi layout	Pengujian pada layout Konsultasi	Klik tombol menu "Konsultasi"	Menampilkan sub menu jenis ikan (Pilih Ikan)
	Konsultasi	Pengujian pada menu Konsultasi	Klik sub menu Konsultasi gejala	Menampilkan <i>Layout</i> Pentanyaan yang disediakan system
			Klik Sub Menu Hasil Diagnosa Penyakit	Menampilkan hasil diagnosa penyakit yang diderita ikan dan memberikan informasi tentang penyakit tersebut, serta memberikan solusi penanganannya.
6	Fungsi pada menu Bantuan	Pengujian pada menu Bantuan	Klik tombol "Bantuan"	Menampilkan Layout mengenai cara menggunakan aplikasi

No.	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
7	Fungsi pada menu Tentang	Pengujian pada menu Tentang	Klik icon "Email"	Menampilkan halaman sosial media pengembang
			Klik icon "Google Play"	Masuk ke akun halaman google play untuk aplikasi Pakar Ikan
			Klik <i>icon</i> "Sosial Media"	Menampilkan halaman sosial media pengembang

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut:

1. Telah berhasil dibangun Aplikasi PAKAR IKAN yang dibuat sebagai sarana penunjang untuk pembudidaya ikan dalam konsultasi masalah penyakit ikan.
2. Sistem pakar yang dibangun dapat memberikan hasil diagnosa berdasarkan fakta-fakta yang telah diberikan.
3. Aplikasi dapat membantu pembudidaya dalam mendiagnosa penyakit ikan dan memberikan solusi terkait penyakit yang diderita layaknya seorang pakar.
4. Berdasarkan dari hasil data pengujian *Equivalence Partitioning*, hasil penilaian pengguna didapat bahwa aplikasi dapat mendiagnosa penyakit dengan sangat baik.

5.2. Saran

Beberapa saran yang diberikan setelah dilakukan penelitian ini untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Penambahan data-data jenis ikan, penyakit, dan gejala serta perbaikan dan penyempurnaan pertanyaan gejala secara lebih detail.

2. Pengembangan selanjutnya dapat membuat aplikasi berjalan di versi Android terbaru dan pengembangan proses diagnosa menggunakan gambar.
3. Aplikasi ini nantinya dapat dikembangkan sehingga kompatibel pada *platform* selain android, seperti iOS, dan Windows Phone.

DAFTAR PUSTAKA

- Andry. 2011. *Android A sampai Z*. PC plus: Jakarta.
- Afrianto Eddy dan Liviawaty Evi. 1992. *Pengendalian Hama dan Penyakit Ikan*. Yogyakarta: Pusat Penerbit Kanisius.
- Apriadi,WH. 2010. *Gizi Keluarga*. Seri Kesejahteraan Keluarga. Jakarta: PT Penebar Swadaya.
- Arhami, M. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi.
- Azwar, S. 2011. *Sikap dan Perilaku. Dalam: Sikap Manusia Teori dan Pengukurannya 2nd ed*. Yogyakarta: Pustaka Pelajar.
- Fishbase. *Freshwater Aquaculture Fish*
<http://www.fishbase.org/search.php>. Diakses pada tanggal 18 April 2016.
- Fowler, Martin. 2004. *UML Distilled Panduan Singkat Bahasa pemodelan Objek Standar, Edisi 3*. Andi Publishing, Yogyakarta.
- Jiang, F., Y. Lu. 2012. Software testing model selection research based on yin-yang testing theory. In: *IEEE Proceeding of International Conference on Computer Science and Information Processing (CISP)*, pp. 590-594.
- Kusrini. 2008. *Aplikasi Sistem Pakar*. Yogyakarta: Andi.
- Kusumadewi, S. 2003. *Artificial Intelegence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.

- Lee, W. M. 2011. *Beginning Android Application Development*. Wiley Publishing, Inc.
- M. Ghufran H. Kordi K. 2010. *Paduan Lengkap Memelihara Ikan Air Tawar Di Kolam Terpal*. Yogyakarta: Lily Publisher.
- Meildy, Bayu. 2014. *Daftar Simbol*. [Online]. Tersedia : <http://elib.unikom.ac.id/download.php?id=83238>. Diakses pada tanggal 11 September 2015.
- Nazruddin, Safaat H. 2012. *(Edisi Revisi) Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika, Bandung.
- Nugroho. 2010. *Rekayasa perangkat lunak menggunakan UML dan Java*. Yogyakarta: Andi.
- Pratama, A.A. 2015. *Sistem pakar diagnosa penyakit pada ikan budidaya air tawar dengan metode forward chaining* (Skripsi). Fakultas Matematika dan Ilmu Pengetahuan Alam. Universitas Lampung. Bandar Lampung.
- Pressman, Roger S. 2001. *Software Engineering A Practitioner's Approach Fifth Edition*. McGraw-Hill Companies, Inc, New York.
- Pressman, R.S. 2010. *Software Engineering: A Practitioner's Approach, 7th Edition*. McGraw-Hill, New York.
- Sutojo, T., Edy, M., dan Vincent, S. 2011. *Kecerdasan Buatan*. Yogyakarta: Andi
- Uml-diagrams.org. 2014. *The Unified Modeling Language*. [Online] Tersedia: <http://www.uml-diagrams.org/>. Diakses pada 10 September 2015.