

**SISTEM PAKAR UNTUK MENDIAGNOSA
PENYAKIT IKAN LELE MENGGUNAKAN METODE
FORWARD CHAINING BERBASIS *ANDROID***

SKRIPSI



**Oleh:
Kirisman Sinaga
130210338**

**FAKULTAS TEKNIK DAN KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA – S1
UNIVERSITAS PUTERA BATAM
TAHUN 2018**

**SISTEM PAKAR UNTUK MENDIAGNOSA
PENYAKIT IKAN LELE MENGGUNAKAN METODE
FORWARD CHAINING BERBASIS *ANDROID***

SKRIPSI

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



**Oleh:
KirismanSinaga
130210338**

**FAKULTAS TEKNIK DAN KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA – S1
UNIVERSITAS PUTERA BATAM
TAHUN 2018**

**SISTEM PAKAR UNTUK MENDIAGNOSA
PENYAKIT IKAN LELE MENGGUNAKAN METODE
FORWARD CHAINING BERBASIS *ANDROID***

**Oleh:
Kirisman Sinaga
130210338**

SKRIPSI

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

**Telah disetujui oleh Pembimbing pada tanggal
seperti tertera di bawah ini**

Batam, 09 Februari 2018

**Nia Ekawati, S.Kom, M.SI
Pembimbing**

PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, dan/atau magister), baik di Universitas Putera Batam maupun di perguruan tinggi lain.
2. Skripsi ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam, 09 Februari 2018
Yang membuat pernyataan,

Kirisman Sinaga
130210338

ABSTRAK

Kebutuhan akan informasi tentang penyakit lele saat ini sangatlah dibutuhkan, hal ini biasanya terjadi pada daerah yang jauh dari pemukiman salah satunya para petani ikan yang berlokasi di pedesaan yang jauh dari perkembangan teknologi, kurangnya jumlah para ahli atau pakar yang ada di lingkungan sehingga menyebabkan gagal panen yang disebabkan oleh penyakit dan virus. Ikan Lele merupakan salah satu jenis ikan yang hidupnya di air tawar, yang memiliki ciri-ciri tubuh berbentuk sedikit bulat memanjang, yang makin ke arah ekor makin mengecil dan memipih, kepala besar berbentuk segitiga dan pipih. Ikan lele dapat diserang oleh beberapa jenis-jenis penyakit, penyakit tersebut dapat diketahui dari gejala-gejala yang alami. Oleh karena itu dibutuhkan sistem pakar yang digunakan untuk mendiagnosa penyakit pada ikan lele. Penelitian ini bertujuan mengimplementasikan sistem pakar untuk mendiagnosa penyakit ikan lele dengan menggunakan metode *Forward Chaining* berbasis *android*. 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 merupakan penalaran yang dimulai dari gejala-gejala terlebih dahulu menuju kesimpulan menderita penyakit tersebut atau tidak. Hasil penelitian berupa aplikasi sistem pakar untuk mendiagnosa penyakit ikan lele berbasis *android* yang membantu pembudidaya dan masyarakat umum.

Kata kunci: Sistem Pakar, Ikan Lele, *Forward Chaining*, *Android*

ABSTRACT

The need for information about catfish diseases today is very necessary, this usually occurs in areas far from the settlement of one of the fish farmers located in the countryside away from technological developments, the lack of experts or experts in the environment causing crop failure caused by illness and virus. Catfish is one type of fish that live in fresh water, which has the characteristics of the body shaped slightly rounded elongated, which increasingly toward the tail of smaller and flattened, large head triangular and flat. Catfish can be attacked by several types of diseases, the disease can be known from the natural symptoms. Therefore it takes an expert system used to diagnose disease in catfish. This study aims to implement expert system to diagnose catfish disease by using Forward Chaining method based on android. In this study, expert systems built on android using Java programming language as a database. The reasoning method used is forward chaining. This method is a reasoning that starts from the symptoms beforehand to the conclusion of suffering the disease or not. The results of the research is the application of expert systems to diagnose android catfish-based diseases that help farmers and the general public.

Keywords: Expert System, Catfish, Forward Chaining, Android

KATA PENGANTAR

Segala puji atas ke hadirat Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan karuniaNya, sehingga peneliti dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam.

Peneliti menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa peneliti terima dengan senang hati. Dengan segala keterbatasan, peneliti menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, peneliti menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam.
2. Ketua Program Studi Teknik Informatika Universitas Putera Batam.
3. Ibu Nia Ekawati, S.Kom., M.SI., selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam.
5. Bapak Pramudya Dwiwahyu Irawanto (dokter hewan di Stasiun Karantina Ikan Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. Nahar No.1 Batam Center dan juga narasumber yang telah rela meluangkan banyak waktunya untuk mendukung penelitian ini.
6. Keluarga yang selalu memberikan doa dan motivasi yang baik kepada peneliti.
7. Rekan-rekan mahasiswa/i Universitas Putera Batam yang turut memberikan doa dan dukungannya.

8. Serta pihak-pihak lain yang tidak dapat disebutkan satu per satu.

Semoga Tuhan Maha Esa membalas kebaikan dan selalu mencurahkan hidayah serta taufik-Nya, Amin.

Batam, 09 Februari 2018

Kirisman Sinaga

DAFTAR ISI

HALAMAN	JUDUL
PERNYATAAN.....	v
HALAMAN PENGESAHAN.....	vi
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Penelitian.....	1
1.2 Identifikasi Masalah.....	10
1.3 Pembatasan Masalah.....	11
1.4 Perumusan Masalah.....	11
1.5 Tujuan Penelitian.....	12
1.6 Manfaat Penelitian.....	12
BAB II TINJAUAN PUSTAKA.....	14
2.1. Teori Dasar	14
2.1.1 Kecerdasan Buatan atau <i>Artificial Intelligence (AI)</i>	14
2.1.2 Sistem Pakar (<i>Expert System</i>).....	16
2.1.2.1 Manfaat Dan Kekurangan Sistem Pakar.....	17
2.1.2.2 Ciri-Ciri Sistem Pakar	18
2.1.2.3 Konsep-Konsep Dasar Sistem Pakar	19
2.1.2.4 Struktur Sistem Pakar	21
2.1.2.5 Representasi Pengetahuan	24
2.1.2.6 Metode Inferensi.....	33
2.2.2.6.1 <i>Forward Chaining</i>	33
2.2.2.6.2 <i>Backward Chaining</i>	36
2.1.3 Logika <i>Fuzzy (Fuzzy Logic)</i>	37
2.1.4 Jaringan Saraf Tiruan (<i>Artificial Neural Network</i>).....	40

2.1.4.1 Kelebihan Jaringan Saraf Tiruan	41
2.1.4.2 Kelemahan Jaringan Saraf Tiruan	41
2.1.4.3 Arsitektur Jaringan Saraf Tiruan	42
2.1.5 Algoritma Pembelajaran Dengan Supervisi	43
2.1.6 Pelatihan Tanpa Supervisi (Jaringan Kohonen)	45
2.1.7 <i>Android</i>	45
2.1.8 Pengetahuan (<i>Knowledge</i>).....	46
2.1.9 Validasi Sistem.....	47
2.2 Variabel	48
2.2.1 <i>Trichodina sp.</i>	49
2.2.2 <i>Dactylogyrus sp</i> dan <i>Gyrodactylus sp</i>	50
2.2.3 <i>Ichthyophthirius Multifiliis</i>	51
2.3 Software Pendukung.....	52
2.3.1 <i>Eclipse</i>	52
2.3.2 <i>Java</i>	53
2.3.3 <i>UML (Unified Modeling Language)</i>	54
2.3.3.1 Pengenalan <i>UML (Unified Modeling Language)</i>	54
2.3.3.2 <i>Use Case Diagram</i>	55
2.3.3.3 <i>Activity Diagram</i>	56
2.3.3.4 <i>Sequence Diagram</i>	57
2.3.3.5 <i>Class Diagram</i>	59
2.4 Penelitian Terdahulu.....	61
2.5 Kerangka Pemikiran	69
BAB III METODE PENELITIAN	72
3.1 Desain Penelitian	72
3.2 Teknik Pengumpulan Data	77
3.3 Operasional Variabel	78
3.4 Perancangan Sistem.....	82
3.4.1 Perancangan Basis Pengetahuan	82
3.4.2 Struktur kontrol (mesin inferensi)	90
3.4.3 Desain <i>UML (Unified Modeling Language)</i>	91
3.4.4 Desain antarmuka	103
3.5 Lokasi dan Jadwal Penelitian	107
3.5.1 Lokasi	107
3.5.2 Jadwal Penelitian	107
BAB IV HASIL DAN PEMBAHASAN.....	109

4.1 Hasil Penelitian.....	109
4.2 Pembahasan	114
4.2.1 Pengujian validasi sistem	114
BAB V SIMPULAN DAN SARAN.....	116
5.1 Simpulan.....	116
5.2 Saran.....	117
DAFTAR PUSTAKA	
RIWAYAT HIDUP	
SURAT KETERANGAN PENELITIAN	
LAMPIRAN	

DAFTAR TABEL

	Halaman
Tabel 2. 1 Tabel Keputusan.....	28
Tabel 2. 2 Alternatif Tabel Keputusan	30
Tabel 2. 3 <i>Use CaseDiagram</i>	55
Tabel 2. 4 <i>ActivityDiagram</i>	57
Tabel 2. 5 <i>SequenceDiagram</i>	58
Tabel 2. 6 <i>Class Diagram</i>	61
Tabel 3. 1 Tabel Variabel dan indikator	82
Tabel 3. 2 Tabel data indikator.....	83
Tabel 3. 3 Tabel indikator penyakit ikan lele	83
Tabel 3. 4 Tabel gejala penyakit ikan lele	84
Tabel 3. 5 Tabel aturan.....	85
Tabel 3. 6 Tabel keputusan.....	86
Tabel 3. 7 Jadwal Penelitian.....	108
Tabel 4. 1 Tabel pengujian validasi sistem.....	115

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Komponen-komponen dalam sebuah sistem pakar	22
Gambar 2. 2 Model Representasi Pengetahuan	25
Gambar 2. 3 Pohon Keputusan	29
Gambar 2. 4 Pohon Keputusan	31
Gambar 2. 5 Graph Pengetahuan	35
Gambar 2. 6 Cabang-Cabang <i>Episemology</i>	46
Gambar 2. 7 Kerangka Pemikiran	70
Gambar 3. 1 Desain Penelitian	73
Gambar 3. 2 Pohon Keputusan	88
Gambar 3. 3 Desain <i>Knowledge Base</i>	89
Gambar 3. 4 <i>Use case diagram</i>	91
Gambar 3. 5 <i>Activity diagram</i> mengelola <i>data base</i>	93
Gambar 3. 6 <i>Activity diagram</i> <i>execute file apk</i>	94
Gambar 3. 7 <i>Activity diagram</i> memperbarui versi	95
Gambar 3. 8 <i>Activity diagram</i> <i>sharing file apk</i>	95
Gambar 3. 9 <i>Activity diagram</i> mengunduh <i>file apk</i>	96
Gambar 3. 10 <i>Activity Diagram</i> melakukan instalasi	97
Gambar 3. 11 <i>Activity diagram</i> melakukan diagnosa.....	98
Gambar 3. 12 <i>Sequence diagram</i> mengelola <i>knowledge base</i>	99
Gambar 3. 13 <i>Sequence diagram</i> <i>execute file Apk.exe</i>	100
Gambar 3. 14 <i>Sequence diagram</i> memperbarui versi.....	100
Gambar 3. 15 <i>Sequence diagram</i> <i>sharing file apk</i>	101
Gambar 3. 16 <i>Sequence diagram</i> mengunduh <i>file Apk</i>	101
Gambar 3. 17 <i>Sequence diagram</i> melakukan instalasi	102
Gambar 3. 18 <i>Sequence diagram</i> melakukan diagnosa.....	102
Gambar 3. 19 Rancangan <i>Form</i> Beranda	103
Gambar 3. 20 Rancangan <i>Form</i> Hasil Diagnosa	104
Gambar 3. 21 Rancangan <i>Form</i> Hasil Diagnosa	105
Gambar 3. 22 Rancangan <i>Form</i> Tentang.....	105
Gambar 3. 23 Rancangan <i>Form</i> Artikel	106
Gambar 3. 24 Rancangan <i>Form</i> <i>Profil</i>	107
Gambar 4. 1 Beranda	109
Gambar 4. 2 Tampilan <i>form</i> diagnosa	110
Gambar 4. 3 Tampilan hasil diagnosa	111
Gambar 4. 4 Tampilan menu artikel	112
Gambar 4. 5 Tampilan <i>Form</i> Tentang	113
Gambar 4. 6 Tampilan menu profil	114

DAFTAR LAMPIRAN

LAMPIRAN I WAWANCARA

LAMPIRAN II FOTO WAWANCARA

LAMPIRAN III DATA KASUS PENYAKIT

LAMPIRAN IV KODING PROGRAM

BAB I

PENDAHULUAN

1.1 Latar Belakang Penelitian

Kota Batam berada di Provinsi Kepulauan Riau yang dikelilingi laut. Kota yang bagian dari Provinsi Kepulauan Riau ini, memiliki luas wilayah daratan seluas 715 km², sedangkan luas wilayah keseluruhan mencapai 1.575 km². Kota Batam beriklim tropis dengan suhu rata-rata 26 sampai 34 derajat celsius. Kota ini memiliki dataran yang berbukit dan berlembah. Tanahnya berupa tanah merah yang kurang subur dan cuaca yang sering berubah sehingga untuk dijadikan lahan pertanian hanya tanaman yang dapat tumbuh tanpa mengikuti musim. Kota Batam sebuah [kota](#) terbesar di Provinsi [Kepulauan Riau, Indonesia](#). Wilayah Kota Batam terdiri dari [Pulau Batam](#), [Pulau Rempang](#) dan [Pulau Galang](#) dan pulau-pulau kecil lainnya di kawasan [Selat Singapura](#) dan [Selat Malaka](#). Pulau Batam, Rempang, dan Galang terkoneksi oleh [Jembatan Bareleng](#). Menurut Dinas Kependudukan dan Catatan Sipil Kota Batam per 2015, jumlah penduduk Batam mencapai 1.037.187 jiwa. Batam salah satu kota dengan letak yang sangat strategis. Selain berada di jalur pelayaran internasional, kota ini memiliki jarak yang sangat dekat dan berbatasan langsung dengan [Singapura](#) dan [Malaysia](#). Sebagai [kota terencana](#), Batam salah satu kota dengan pertumbuhan terpesat di Indonesia. Selain letaknya yang strategis pulau Batam menyediakan tempat wisata menarik serta keanekaragaman hayati lainnya seperti halnya dalam bidang pertanian dan

perikanan, pada saat ini industri perikanan mengalami kemajuan yang sangat pesat yang ditandai dengan mulai banyaknya minat masyarakat yang mulai mengembangkan budidaya ikan, baik ikan tawar maupun ikan air asin. Salah satunya Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center.

Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center. suatu unit organisasi di bawah Kementerian Kelautan dan Perikanan yang mempunyai tugas menyelenggarakan perkarantinaan ikan, pengendalian mutu dan keamanan hasil perikanan, serta keamanan hayati ikan yang terletak di Jl. M. Nahar. Batam Centre Kota Batam. Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center. bertugas sebagai pelayanan jasa karantina ikan dalam rangka pencegahan masuk dan tersebarnya hama penyakit ikan karantina dan pengujian mutu hasil perikanan dalam rangka pengendalian mutu dan keamanan hasil perikanan.

Berdasarkan hasil wawancara di Stasiun Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center. dengan drh. Pramudya Dwiwahyu Irawanto. Penyakit ikan disebabkan oleh bakteri, parasit, serta jamur yang melekat di kulit dan insang, seperti *protozoa*, *copepod*, *pseudomonas*, dan *aeromonas*. Dampaknya bisa berupa infeksi yang menyebabkan fisik terus melemah, bila tidak diatasi dapat mengakibatkan ikan mati. Penyakit sangat ditakuti pembudidaya karena menjadi faktor utama penyebab kematian pada ikan. Munculnya penyakit ikan disebabkan

oleh faktor, diantaranya karena air yang kotor atau jarang diganti, pemberian pakan yang berlebihan, fasilitas budidaya yang tercemar penyakit, suhu yang ekstrim dan kesalahan penanganan. Penanganan yang dilakukan budidaya harus diperhatikan dengan serius, agar tidak menimbulkan bahaya dan dapat menyebabkan kematian serta menghindari kesalahan yang terulang. Untuk mengatasinya, harus dilakukan pencegahan ataupun pengobatan yang sesuai dengan jenis penyakit agar penularan penyakit tidak lebih parah dan meluas, seperti pembersihan dan pengeringan dasar kolam setiap selesai panen, penggunaan bibit ikan yang sehat dan bebas penyakit, menghindari penebaran bibit ikan terlalu padat (melebihi kapasitas kolam), pakan diberikan dalam jumlah yang cukup dan tidak berlebihan, sisa-sisa pakan akan mengendap didasar kolam.

Berdasarkan penelitian (Hernawati, 2015) diperoleh fakta: Penyakit merupakan kendala utama dalam budidaya ikan hias, salah satunya adalah ikan botia. Banyaknya jenis penyakit yang menyerang ikan botia dapat menimbulkan kerugian besar. Upaya yang dapat dilakukan untuk meminimalisir terjadinya kerugian tersebut adalah dengan mencegah tersebarnya penyakit. Tujuan penelitian ini adalah untuk mengetahui jenis *patogen* yang menyerang ikan botia (*Chromobotia macracanthus Bleeker*) yang dilalulintaskan melalui Stasiun Karantina Ikan Kelas I Supadio, Pontianak. Pengamatan yang dilakukan adalah identifikasi parasit, bakteri dan jamur. Hasil pengamatan menunjukkan bahwa agen penyakit yang menyerang ikan botia yaitu *Gyrodactylus sp.*, *Myxobolus sp.*, *Argulus sp.*, *Vorticella sp.* dari agen parasit, *Neisseria sp.*, *Aeromonas hydrophila*, *Chromobacterium violaceum*, *Pasteurella haemolytica*

dari agen bakteri dan *Saprolegnia sp.* dari agen jamur. Seluruh agen penyakit yang teramati bukan termasuk golongan Hama Penyakit Ikan Karantina.

Berdasarkan hasil wawancara di Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center. dengan drh. Pramudya Dwiwahyu Irawanto. Ikan Lele salah satu jenis ikan yang hidupnya di air tawar, yang memiliki ciri-ciri tubuh berbentuk sedikit bulat memanjang, yang makin ke arah ekor makin mengecil dan memipih, kepala besar berbentuk segitiga dan pipih. Ikan lele komoditas perikanan yang sangat penting, dan tidak dapat dipisahkan dari kebutuhan pangan (lauk) masyarakat sehari-hari khususnya Kota Batam. Oleh karena itu, lele juga telah lama populer di tengah-tengah masyarakat Batam. Sebagai bahan pangan, ikan lele memiliki manfaat bagi masyarakat sebagai kandungan protein tinggi, dan juga mengandung zat-zat nutrisi lainnya seperti kalori, lemak, karbohidrat, kalsium, fosfor, besi, dan vitamin. Manfaat bagi pembudidaya peningkatan produksi perikanan untuk meningkatkan ketersediaan protein hewani dan tingkat konsumsi ikan bagi masyarakat Batam. Ikan lele mudah didapatkan dengan harga yang relatif murah jika dibandingkan dengan jenis ikan air tawar lainnya, seperti gurame, nila, mas, patin. Oleh karena itu, ikan lele sangat tinggi dan cepat laku. Hal ini tentu saja suatu prospek yang menjanjikan bagi masyarakat dan pembudidaya ikan yang mau berbisnis. Ikan lele memiliki jenis-jenis penyakit seperti bakteri *Pseudomonas*, *Aeromonas hydrophyla*, *Aeromonas punctata*, *Columnaris*, *Penduncle*, *Edwardsiella*, *Tuberculosis*. Penyakit serangan jamur seperti *Saprolegnia* (jamur putih seperti kapas). Penyakit serangan parasit seperti

Ichthyophthirius Multifilis (bercak putih dan gatal), *Trichodina sp*, *Gyrodactylus sp* & *Dactylogyrus sp*, *Lernae sp*. *Trichodina sp* sejenis penyakit parasit yang dapat menyebabkan gatal (*Trichodiniasis*), gejala ikan ini biasanya nafsu makan berkurang, bintik-bintik putih bagian kepala serta punggung, ikan menjadi sangat lemah dan kurus, pada warna tubuh menjadi kusam, sering terlihat ikan menggosokkan tubuhnya pada dasar atau dinding kolam/tambak serta benda-benda keras di sekitarnya. Tindakan pengobatan yang dapat dilakukan adalah pergantian air kolam dengan air yang terbaru untuk menjaga kualitas air kolam. Sementara itu, pengobatan rendam dalam larutan formalin 150-200 cc/m³. Rendam dalam larutan *malachyte green oxalate* 0,1 g/m³ selama 12-24 jam. *Dactylogyrus sp* dan *gyrodactylus sp* jenis penyakit parasit, dan biasanya sering disebut cacing, bentuknya pipih. Gejala-gejala dari penyakit ini yaitu nafsu makan berkurang, ikan kelihatan lemah dan kurus, kulit ikan kusam, kulit berlendir, menggosokkan badan pada dinding kolam atau benda-benda keras, sirip rontok. Tindakan yang dilakukan mengganti dan mengurangi kepadatan ikan, penyakit ini dapat diobati dengan merendam dalam larutan *formalin* 250 cc/m³ air selama 15 menit lalu rendam dalam larutan *methyline blue* 3 g/m³ air selama 24 jam. *Ichthyophthirius Multifiliis* jenis penyakit parasit, memiliki gejala tubuh berubah menjadi pucat dan mengelupas, perubahan fisik seperti timbul bintik-bintik putih pada tubuh, sirip tampak robek-robek dan ingang memucat, terjadinya kerusakan pada kulit dan insang memicu ikan mengalami stress, ikan sering mengosokkan tubuh pada dasar dan dinding kolam. Tindakan pengobatan siapkan wadah berupa bak untuk mengobati ikan yang sakit, buat larutan baku dengan mencampurkan 1

gram *methylene blue* ke dalam 100 ml air bersih, teteskan larutan baku tadi ke dalam bak sebanyak 2-4 ml untuk setiap 4 liter air. Rendam ikan yang sakit selama 24 jam dan ulangi pengobatan sebanyak 3-5 kali dalam selang waktu satu hari.

Berdasarkan penelitian (Suprpto & Iswanto, 2015) diperoleh fakta: Ikan lele Mutiara merupakan *strain* baru ikan lele Afrika (*Clarias gariepinus*) hasil pemuliaan yang memiliki keunggulan karakteristik budidaya, terutama pertumbuhan. Selain karakteristik budidayanya, karakteristik morfologis ikan lele Mutiara juga perlu dieksplorasi. Salah satu aspek morfologi yang perlu dieksplorasi tersebut adalah abnormalitas morfologis benihnya. Penelitian ini bertujuan untuk mengetahui abnormalitas morfologis benih ikan lele Mutiara dibandingkan dengan benih *strain-strain* ikan lele Afrika lain yang digunakan dalam kegiatan budidaya di Indonesia, yakni ikan lele Sangkuriang, Dumbo, Sukhoi, Burma, Paiton, Phyton dan Masamo. Karakteristik yang diamati adalah abnormalitas bentuk morfologis (deformitas) dan fluktuasi asimetri sirip dada dan sirip perut. Hasil penelitian ini menunjukkan bahwa tingkat deformitas (4,00%) dan fluktuasi asimetri (sirip dada 0,14 dan sirip perut 0,02) benih ikan lele Mutiara lebih rendah daripada benih-benih ikan lele Sangkuriang, Dumbo, Sukhoi, Burma, Paiton, Phyton dan Masamo (deformitas berkisar 6,00-42,00%, fluktuasi asimetri sirip dada berkisar 0,30-0,68 dan sirip perut berkisar 0,12-0,62). Hasil tersebut menunjukkan bahwa bentuk morfologis benih ikan lele Mutiara lebih normal daripada benih-benih ikan lele Sangkuriang, Dumbo, Sukhoi, Burma, Paiton, Phyton dan Masamo. Hal tersebut mengindikasikan bahwa mutu dan keragaman genetik ikan lele Mutiara lebih tinggi

dari pada *strain-strain* ikan lele Afrika lain yang digunakan dalam kegiatan budidaya di Indonesia tersebut.

Sistem pakar berguna untuk menyimpan pengetahuan seorang pakar agar pengetahuan tersebut tetap ada, walaupun pakar tersebut sudah tidak ada lagi. Sistem pakar terdiri dari basis pengetahuan yang berisi pengetahuan-pengetahuan dalam menyelesaikan masalah, dan mesin inferensi sebagai otak dari sistem pakar. Dimana sistem pakar yang dirancang ini dapat menggantikan seorang pakar atau dokter yang bisa mendiagnosa penyakit ikan lele tersebut dengan hanya menggunakan sistem pakar tersebut di komputer. Dan setiap orang pun bisa menggunakannya tanpa harus konsultasi kedokter atau pun ke seorang pakar.

Berdasarkan penelitian (Tamin, 2015) diperoleh fakta: Mensubtitusikan pengetahuan manusia ke dalam bentuk sistem dengan menampung kemampuan/keahlian seorang pakar untuk melakukan proses analisa suatu masalah sehingga sistem dapat sistem bekerja menyelesaikan masalah sebagaimana manusia mengerjakannya dan menyelesaikan masalah tersebut. Kerusakan *printer* terkadang menjadi masalah besar ketika seorang yang awam tidak mengetahui letak kesalahan *printer* maka dibutuhkan sistem yang mampu bekerja otomatis untuk memberikan solusi kerusakan *printer*. Penelitian ini bertujuan untuk merancang sebuah sistem yang dapat digunakan untuk menangani kerusakan *printer*. Pengguna aplikasi ini seolah-olah berhadapan langsung dengan pakar dibidang *hardware* khususnya *printer*. Perencanaan sistem dilakukan dengan membuat *knowledge base* menggunakan *decision tree* dan Aturan *if-then* sebagai representasi pengetahuan. Sistem dibuat dengan

meenggunakan metode *forward chaining* dan bahasa pemrograman *Visual Basic*. Hasil penelitian ini mengungkapka jenis-jenis kerusakan yang terjadi pada sebuah *printer* serta penanganan dari kerusakan tersebut. Pengujian aplikasi juga dilakukan untuk mengetahui akurasi dan variasi serta *user friendly* dan fleksibilitas sistem. Hasil dari keseluruhan pengujian ini dapat disimpulkan bahwa program sudah cukup baik walaupun jenis kerusakan yang dihasilkan belum lengkap karena pada sistem ini hanya mendeteksi 15 jenis kerusakan mesin secara umum.

Metode *Forward Chaining* atau pelacakan maju dapat diketahui sebagai teknik penalaran dalam pengambilan keputusan yang dimulai dari fakta-fakta terlebih dahulu untuk menentukan kesimpulan. Berdasarkan metode *Forward Chaining*, fakta-fakta akan diuji satu persatu dalam urutan tertentu untuk menghasilkan suatu kesimpulan dari permasalahan yang dihadapi. Metode *Forward Chaining* dimulai dari informasi masukan (*if*) dahulu kemudian menuju konklusi (*then*). Informasi masukan dapat berupa data, bukti, temuan atau pengamatan, sedangkan konklusi dapat berupa tujuan, penjelasan, atau diagnosis.

Berdasarkan penelitian (Pasaribu, 2015) diperoleh fakta: Mata merupakan suatu panca indra yang sangat penting dalam kehidupan manusia untuk melihat. Jika mata mengalami gangguan atau penyakit mata, maka akan berakibat sangat fatal bagi kehidupan manusia. Jadi sudah mestinya mata merupakan anggota tubuh yang perlu dijaga dalam kesehatan sehari-hari. Sistem pakar merupakan suatu bagian metode ilmu-ilmu *artificial intelligence* untuk dibuat suatu program aplikasi diagnosa penyakit mata pada manusia yang terkomputerisasi serta

berusaha menggantikan dan menirukan proses penalaran dari seorang ahlinya atau pakar dalam memecahkan masalah spesifikasi yang dapat dikatakan duplikat dari seorang pakar karena pengetahuan ilmu tersebut tersimpan di dalam suatu sistem *database*. Sistem Pakar Untuk Diagnosa Penyakit Mata Pada Manusia menggunakan metode *forward chaining* bertujuan menelusuri gejala yang ditampilkan dalam bentuk pertanyaan-pertanyaan agar dapat mendiagnosa jenis penyakit dengan perangkat lunak berbasis *desktop management system*. Perangkat lunak sistem pakar dapat mengenali jenis penyakit mata setelah melakukan konsultasi dengan menjawab beberapa pertanyaan-pertanyaan yang ditampilkan oleh aplikasi sistem pakar serta dapat menyimpulkan beberapa jenis penyakit mata yang di derita oleh pasien. Data penyakit yang dikenali menyesuaikan *rules* (aturan) yang dibuat untuk dapat mencocokkan gejala-gejala penyakit mata dan memberi nilai persentase agar mengetahui nilai pendekatan jenis penyakit pasien. *Android* sistem operasi yang sekarang sedang terkenal di pasaran *smartphone* saat ini. *Android* sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi.

Berdasarkan penelitian (Ichwan & Hakiky, 2011) diperoleh fakta: Metode pencarian adalah cara menemukan data pada kumpulan data untuk menghasilkan informasi yang penting, saat ini metode pencarian yang digunakan sangat banyak salah satunya *Goodreads*. *Goodreads* adalah situs jejaring sosial yang mengkhususkan pada katalog buku yang menyediakan *Application Programming Interface* (API) yang dijalankan pada Teknologi *Web 2.0*. Metode *Goodreads*

digunakan karena menyediakan *Application Programming Interface* (API) memungkinkan pengembang perangkat lunak mengakses kedalam *database*. Metode tersebut diimplementasikan pada *smartphone* berbasis *Android*. *Android* sebagai sistem operasi yang dapat ditanamkan pada perangkat *smartphone* yang memiliki kemampuan diinstal aplikasi yang diperlukan oleh pengguna. Aplikasi tersebut untuk mengukur kinerja *Goodreads application programming interface* pada pencarian data buku. Berdasarkan pengujian dan penelitian yang dilakukan, melakukan pengukuran kinerja pada suatu aplikasi sangat dibutuhkan untuk mengetahui performa Aplikasi. Aplikasi “*Books on Goodreads*” yang dibangun dapat dimanfaatkan untuk melakukan pencarian data buku menggunakan metode *Goodreads* dengan memanfaatkan API, hasil pencarian dapat disimpan di *database*, untuk mengambilnya kembali tanpa harus melakukan pencarian dan tidak memerlukan koneksi *internet*.

Berdasarkan uraian diatas, maka melakukan penelitian dengan judul “**SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT IKAN LELE MENGGUNAKAN METODE *FORWARD CHAINING* BERBASIS *ANDROID***”

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, maka identifikasi masalah pada penelitian ini adalah:

1. Penyakit ikan disebabkan oleh bakteri, parasit, serta jamur yang melekat di kulit dan insang, seperti *protozoa*, *copepod*, *pseudomonas*, dan *aeromonas*.

2. Dampaknya bisa berupa infeksi yang menyebabkan fisik terus melemah, bila tidak diatasi dapat mengakibatkan ikan mati.
3. Penyakit sangat ditakuti pembudidaya karena menjadi faktor utama penyebab kematian pada ikan.
4. Munculnya penyakit ikan disebabkan oleh faktor, diantaranya karena air yang kotor atau jarang diganti, pemberian pakan yang berlebihan, fasilitas budidaya yang tercemar penyakit, suhu yang ekstrim dan kesalahan penanganan.

1.3 Pembatasan Masalah

Berdasarkan identifikasi masalah, maka memberikan batasan permasalahan sebagai berikut:

1. Penelitian ini membahas penyakit ikan lele yaitu: parasit *Trichodina sp*, *Dactylogyrus sp* dan *gyrodactylus sp*, dan *Ichthyophthirius Multifiliis*.
2. Penelitian dilakukan di Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center.dengan drh. Pramudya Dwiwahyu Irawanto.
3. Sistem pakar yang dibuat hanya untuk mendiagnosa penyakit ikan lele dengan menggunakan metode *Forward Chaining*.
4. Aplikasi *android* menggunakan *Android* versi 4.2 (*Jelly Bean*) dengan bahasa pemrograman *Java* dengan menggunakan aplikasi *editor eclipse*.

1.4 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka rumusan masalah penelitian ini adalah bagaimana implementasi sistem pakar untuk mendiagnosa penyakit ikan lele dengan menggunakan metode *forward chaining* berbasis *android*?

1.5 Tujuan Penelitian

Sesuai dengan konsep yang ada dan upaya untuk menyelesaikan hasil penelitian maka tujuan dari penelitian ini adalah mengimplementasikan sistem pakar untuk mendiagnosa penyakit ikan lele dengan menggunakan metode *Forward Chaining* berbasis *android*

1.6 Manfaat Penelitian

Manfaat penelitian dapat dikategorikan menjadi dua, yaitu manfaat teoritis dan manfaat praktis. Berikut manfaat yang diharapkan dalam penelitian ini:

a) Aspek Teoritis

1. Menambah pengetahuan dan pengalaman bagi peneliti dalam bidang pembuatan perancangan sistem pakar untuk mendiagnosa penyakit ikan lele berbasis *android*.
2. Sebagai tambahan referensi untuk mahasiswa yang membutuhkan informasi mengenai metode *Forward Chaining*.

b) Aspek Praktis

1. Untuk tenaga budidaya ikan lele, aplikasi ini dapat digunakan sebagai tambahan acuan untuk mendiagnosa penyakit ikan lele berbasis *android*.
2. Diharapkan aplikasi ini dapat digunakan sebagai alat bantu untuk mendiagnosa penyakit ikan lele bagi masyarakat umum.

BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

Menurut (Sugiyono, 2012: 52) Setelah masalah penelitian dirumuskan, maka langkah kedua dalam proses penelitian (kuantitatif) adalah mencari teori-teori, konsep-konsep, generalisasi-generelisasi hasil penelitian yang dapat dijadikan sebagai landasan teoritis untuk pelaksanaan penelitian.

Pada penelitian ini akan dikemukakan landasan teori yang terkait dengan permasalahan untuk mendukung perancangan sistem. Adapun landasan teori yang digunakan sebagai berikut:

2.1.1 Kecerdasan Buatan atau *Artificial Intelligence (AI)*

Kecerdasan buatan (*artificial intelligence*) salah satu bagian dari ilmu komputer yang mempelajari tentang bagaimana sebuah komputer bisa dibuat dengan sedemikian rupa agar dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia.

Menurut (Sutojo, Mulyanto, & Suhartono, 2011: 1-3) Kecerdasan Buatan (*Artificiasl Intelligence*) dimaksud merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia. Berikut adalah beberapa definisi kecerdasan buatan yang telah didefinsikan oleh beberapa ahli.

1. Alan Turing menetapkan definisi *artificial intelligence*. Jika komputer tidak dapat dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan.
2. Herber Alexander Simon (2001) Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.
3. Rich and knight (1991) Kecerdasan buatan (AI) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

Menurut Ahli Winston dan Prendergast (1984), tujuan dari kecerdasan buatan adalah:

1. Membuat mesin menjadi lebih pintar (tujuan utama)
2. Memahami apa itu kecerdasan (tujuan ilmiah)
3. Membuat mesin lebih bermanfaat (tujuan *entrepreneurial*).

Cerdas adalah memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Jadi, agar mesin bisa jadi cerdas (bertindak seperti manusia) maka harus dibeli bekal pengetahuan dan diberi kemampuan untuk menalar.

Menurut (Budihartono & Suhartono, 2014: 2-3) Kecerdasan buatan atau *Artificial Intelligence* (AI) merupakan bidang ilmu komputer yang mempunyai peran penting di era kini dan masa akan datang. Bidang ini telah berkembang sangat pesat di 20 tahun terakhir seiring dengan pertumbuhan kebutuhan akan

perangkat cerdas pada industri dan rumah tangga. Selama lebih dari ribuan tahun, cara manusia berpikir terus diteliti. Proses tersebut mencakup cara manusia mengetahui, memahami, memprediksi, dan melakukan manipulasi terhadap hal-hal yang lebih besar dan lebih rumit dari yang pernah ada. Bidang keilmuan kecerdasan buatan sampai saat ini terus mencoba untuk melakukan pekerjaan tersebut. Tidak hanya untuk memecahkan berbagai masalah, tetapi juga untuk membangun sebuah sistem atau alat yang memiliki kecerdasan. Kata *intelligence* berasal dari bahasa Latin *intelligo* yang berarti 'saya paham'. Jadi dasar dari *intelligence* adalah kemampuan memahami dan melakukan aksi. Sebenarnya area kecerdasan buatan (*Artificial Intelligence*) atau disingkat dengan AI, bermula dari kemunculan komputer sekitar tahun 1940-an, meskipun sejarah perkembangan dapat dilacak hingga zaman Mesir kuno. Pada masa sekarang, perhatian difokuskan pada kemampuan komputer untuk mengerjakan sesuatu yang dapat dilakukan oleh manusia. Dalam hal ini, komputer tersebut dapat meniru kemampuan kecerdasan dan perilaku manusia.

2.1.2 Sistem Pakar (*Expert System*)

Menurut (Budihartono & Suhartono, 2014: 132) Sistem pakar adalah program komputer yang menyimulasi penilaian dan perilaku manusia atau organisasi yang memiliki pengetahuan dan pengalaman ahli dalam bidang tertentu. Sistem seperti ini berisi basis pengetahuan yang berisi akumulasi pengalaman dan satu set aturan untuk menerapkan pengetahuan dasar untuk setiap situasi tertentu. Sistem pakar yang canggih dapat ditingkatkan dengan

penambahan basis pengetahuan atau set aturan. Diantara banyak sistem yang ada, yang terkenal adalah aplikasi bermain catur dan sistem diagnosis medis.

Menurut (Sutojo et al., 2011: 159) Sistem pakar merupakan cabang dari *Artificial Intelligence (AI)* yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose problem solver (GPS)* yang dikembangkan oleh *Newel* dan *Simon*. Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti *MYCIN* untuk diagnosis penyakit, *DENDRAL* untuk mengidentifikasi struktur molekul campuran yang tak dikenal, *XCON* dan *XSEL* untuk membantu konfigurasi sistem komputer besar, *SOPHIE* untuk analisis sirkuit elektronika, *Prospector* digunakan dibidang geologi untuk membantu mencari dan menemukan deposit, *FOLIO* digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, *DELTA* dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya.

2.1.2.1 Manfaat Dan Kekurangan Sistem Pakar

Menurut (Budihartono & Suhartono, 2014: 134) Sistem pakar banyak digunakan pada aplikasi terkini dan kompleks karena:

1. Sistem pakar dapat bertindak sebagai konsultan, instruktur, atau pasangan/rekan.
2. Meningkatkan *availability* atau kepastian tersedia pada semua perangkat komputer.
3. Mengurangi bahaya.

4. Permanen
5. Pengetahuan dapat tidak lengkap, namun keahlian dapat diperluas sesuai kebutuhan. Program konvensional harus “lengkap” sebelum mereka dapat digunakan.
6. *Database* yang cerdas, sistem pakar digunakan untuk mengakses *database* secara cerdas, misalnya data *mining*.

Menurut (Sutojo et al., 2011: 161) Sistem pakar juga memiliki beberapa kekurangan, yaitu:

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

2.1.2.2 Ciri-Ciri Sistem Pakar

Menurut (Sutojo et al., 2011: 162) Sistem pakar memiliki ciri-ciri sebagai berikut:

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami.
4. Bekerja berdasarkan kaidah/*rule* tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarannya bersifat anjuran.

Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pengguna.

2.1.2.3 Konsep-Konsep Dasar Sistem Pakar

Menurut (Sutojo et al., 2011: 163-165) Konsep-konsep dasar dari sistem pakar, di antaranya:

1. Kepakaran (*Expertise*)

Kepakaran (*Expertise*), merupakan suatu pengetahuan yang diperoleh dari pelatihan, membaca, dan pengalaman. Memungkinkan para ahli dapat mengambil keputusan lebih cepat, dan lebih daripada seorang yang bukan pakar. Kepakaran itu sendiri meliputi pengetahuan tentang :

- a. Fakta-fakta tentang bidang permasalahan tertentu.
- b. Teori-teori tentang bidang permasalahan tertentu.
- c. Aturan-aturan prosedur-prosedur menurut bidang permasalahan umumnya.
- d. Aturan *heuristic* yang harus dikerjakan dalam situasi tertentu.
- e. Strategi global untuk memecahkan permasalahan.
- f. Pengetahuan tentang pengetahuan (*meta knowledge*).

2. Pakar (*Expert*)

Pakar (*Expert*) adalah seorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkan untuk memecahkan masalah atau memberi nasihat. Jadi seorang pakar harus mampu melakukan kegiatan-kegiatan berikut:

- a. Mengenal dan memformulasikan permasalahan.
 - b. Memecahkan permasalahan secara cepat dan tepat.
 - c. Menerangkan pemecahannya.
 - d. Belajar dari pengalaman.
 - e. Merestrukturisasi pengetahuan.
 - f. Memecahkan aturan-aturan.
 - g. Menentukan relevansi.
3. Pemindahan Kepakaran (*Transferring Expertise*)

Pemindahan Kepakaran (*Transferring Expertise*) adalah memindahkan kepakaran dari seseorang ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar.

- a. Akuisisi pengetahuan (dari pakar atau sumber lain)
 - b. Epresentasi pengetahuan (pada komputer)
 - c. Inferensi pengetahuan
 - d. Pemindahan pengetahuan ke pengguna.
4. Inferensi (*Inferencing*)

Inferensi (*Inferencing*) adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Tujuan mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimiliki.

5. Aturan-aturan (*rule*)

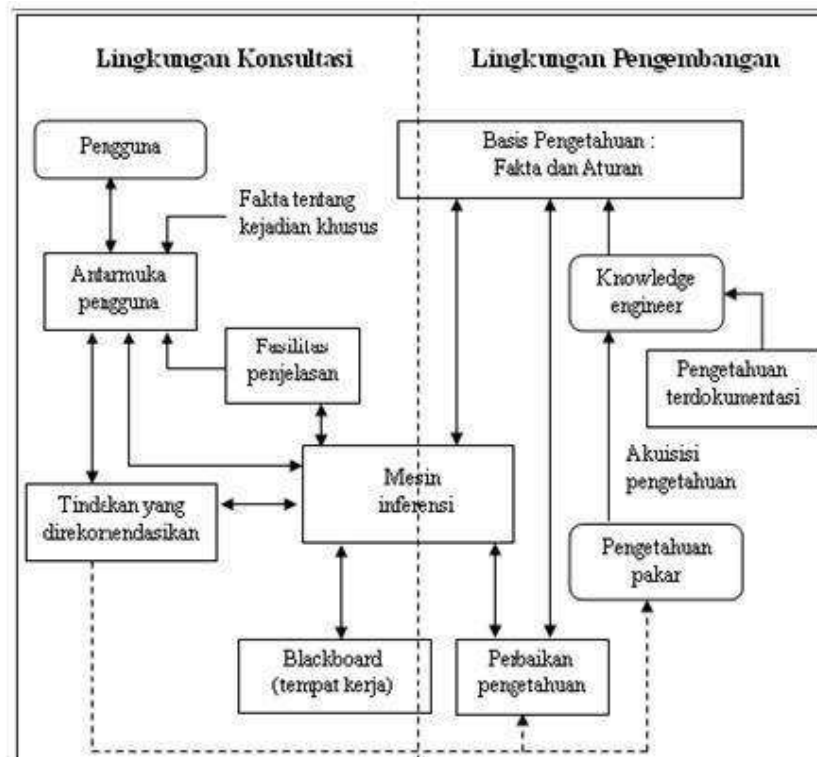
Aturan-aturan (*rule*) kebanyakan *software* sistem pakar komersial adalah sistem yang berbasis *rule* (*rule-based system*), yaitu pengetahuan disimpan terutama dalam bentuk *rule*, sebagai prosedur-prosedur pemecahan masalah.

6. Kemampuan menjelaskan (*Explanation capability*)

Kemampuan menjelaskan (*Explanation capability*) adalah kemampuan untuk menjelaskan saran atau rekomendasi yang diberikan. Sistem ini memungkinkan untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi-operasinya.

2.1.2.4 Struktur Sistem Pakar

Menurut (Sutojo et al., 2011: 166-169) Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembang digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan ke dalam *knowledge base* (basis pengetahuan). Lingkungan konsultasi digunakan oleh pengguna untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar.



Gambar 2. 1 Komponen-komponen dalam sebuah sistem pakar
(Sumber : Sutojo, 2011)

Keterangan:

1. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan meletakkannya ke dalam basis pengetahuan dengan format tertentu (dalam bentuk representasi pengetahuan). Sumber-sumber pengetahuan bisa diperoleh dari pakar, buku, dokumen, *multimedia*, basis data, laporan riset khusus, dan informasi yang terdapat di *web*.

2. Basis Pengetahuan (*Knowledge base*)

Basis pengetahuan berisi pengetahuan yang diperlukan untuk memahami, memformulasikan dan menyelesaikan masalah. Basis pengetahuan terdiri dari

dua elemen dasar yaitu fakta dan *rule* atau aturan. Basis pengetahuan terdiri dari dua elemen dasar, yaitu:

- a. Fakta, misalnya situasi, kondisi, atau permasalahan yang ada.
- b. *Rule* (Aturan), untuk mengarahkan penggunaan dalam memecahkan masalah.

3. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah sebuah program yang berfungsi untuk memandu proses penalaran terhadap suatu kondisi berdasarkan pada basis pengetahuan yang ada, memanipulasi dan mengarahkan kaidah, model, dan fakta yang disimpan dalam basis pengetahuan untuk mencapai solusi atau kesimpulan.

4. Daerah Kerja (*Blackboard*)

Untuk merekam hasil sementara yang akan dijadikan sebagai keputusan dan untuk menjelaskan sebuah masalah yang sedang terjadi, Sistem pakar membutuhkan *Blackboard*, yaitu area pada memori yang berfungsi sebagai basis data. Tiga keputusan yang dapat direkam pada *blackboard*, yaitu:

- a. Rencana : bagaimana menghadapi masalah.
- b. Agenda, aksi-aksi potensial yang sedang menunggu untuk eksekusi.
- c. Solusi : calon aksi yang akan dibangkitkan.

5. Antarmuka Pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dan sistem pakar. Komunikasi ini paling bagus bila disajikan dalam bahasa alami (*natural language*) dan dilengkapi dengan grafik, menu, dan formulir elektronik.

6. Subsistem penjelasan (*Explanation Subsystem*)

Berfungsi memberi penjelasan kepada *user*, bagaimana suatu kesimpulan dapat diambil.

7. Sistem Perbaikan Pengetahuan (*Knowledge Refining Sytem*)

Kemampuan memperbaiki pengetahuan dari seorang pakar diperlukan untuk menganalisis pengetahuan, belajar dari kesalahan masa lalu, kemudian memperbaiki pengetahuannya sehingga dapat dipakai pada masa mendatang.

8. Pengguna (*User*)

Pengguna sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran, atau pelatihan (*training*) dari berbagai permasalahan yang ada.

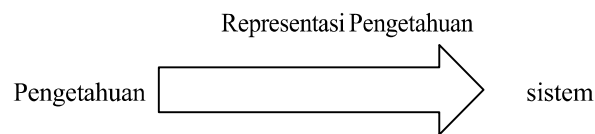
2.1.2.5 Representasi Pengetahuan

Menurut (Kusrini, 2008: 6) Representasi pengetahuan merupakan metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar. Representasi dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi itu dapat diakses oleh prosedur pemecahan masalah. Adapun karakteristik representasi pengetahuan adalah:

1. Harus bisa diprogramkan dengan bahasa pemograman atau dengan *shells* dan hasilnya disimpan dalam memori.
2. Dirancang sedemikian sehingga isinya dapat digunakan untuk proses penalaran.

3. Model representasi pengetahuan merupakan sebuah struktur data yang dapat dimanipulasi oleh mesin inferensi dan pencarian untuk aktivasi pencocokkan pola.

Menurut (Hartati & Iswanti, 2008: 22) Sistem pakar merupakan sistem yang berbasis pengetahuan, mengerjakan tugas yang biasanya dilakukan oleh seorang pakar. Representasi pengetahuan dimaksudkan untuk mengorganisasikan pengetahuan dalam bentuk dan format tertentu untuk bisa dimengerti oleh komputer, dapat digambarkan sebagai berikut:



Gambar 2. 2 Model Representasi Pengetahuan
(Sumber : Hartati, 2008)

Untuk membuat sistem pakar yang efektif harus dipilih representasi pengetahuan yang tepat. Pemilihan representasi pengetahuan yang tepat akan membuat sistem pakar dapat mengakses basis pengetahuan tersebut untuk keperluan pembuatan keputusan. Beberapa model representasi pengetahuan yang penting.

1. Jaringan Semantik

Menurut (Hartati & Iswanti, 2008: 22) Jaringan semantik adalah teknik representasi pengetahuan yang digunakan untuk informasi proposional, sedangkan yang dimaksud dengan informasi proposional adalah pernyataan yang mempunyai nilai benar atau salah. Sebagai contoh: sebuah bujur sangkar

mempunyai empat sisi. Informasi proporsional merupakan bahasa deklaratif karena menyatakan fakta. Representasi jaringan semantik merupakan penggambaran grafis dari pengetahuan yang memperlihatkan hubungan hirarkis dari obyek-obyek. Komponen dasar untuk merepresentasikan pengetahuan dalam bentuk jaringan semantik adalah simpul (*node*) dan penghubung (*link*). Obyek direpresentasikan oleh simpul. Hubungan antar obyek-obyek dinyatakan oleh penghubung yang diberi label untuk menyatakan hubungan yang direpresentasikan.

2. Bingkai (*Frame*)

Menurut (Hartati & Iswanti, 2008: 22-23) Bingkai berupa kumpulan *slot-slot* yang berisi atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam *slot* dapat berupa kejadian, lokasi, situasi ataupun *elemen-elemen* lainnya. Bingkai digunakan untuk representasi pengetahuan deklaratif. Bingkai memuat deskripsi sebuah objek dengan menggunakan tabulasi informasi yang berhubungan dengan objek. Representasikan pengetahuan menggunakan bingkai sesuai untuk jenis pengetahuan yang memiliki subjek sempit, lebih bersifat pasti dan jarang berubah-ubah isinya kecuali terdapat kondisi khusus.

3. Kaidah Produksi

Menurut (Hartati & Iswanti, 2008: 25-26) Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah reproduksi dituliskan dalam bentuk jika-maka (*if-then*). Kaidah *if-then* menghubungkan atesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Berbagai

struktur kaidah *if-then* yang menghubungkan obyek atau atribut sebagai berikut (Adedeji,1992):

IF premis *THEN* konklusi

IF masukan *THEN* keluaran

IF kondisi *THEN* tindakan

IF antesenden *THEN* konsekuen

IF data *THEN* hasil

IF tindakan *THEN* tujuan

IF aksi *THEN* reaksi

IF sebab *THEN* akibat

IF gejala *THEN* diagnosa

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. *Masukan* mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. *Kondisi* mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. *Antesenden* mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. *Data* mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. *Tindakan* mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. *Aksi* mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. *Gejala* mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan.

Menurut (Sutojo et al., 2011: 170) Setiap *rule* terdiri dari dua bagian, yaitu *IF* disebut *evidence* (fakta-fakta) dan bagian *THEN* disebut Hipotesis atau kesimpulan.

Syntax Rule adalah:

IF E THEN H

E: *evidence* (fakta-fakta)

H: Hipotesis atau kesimpulan yang dihasilkan.

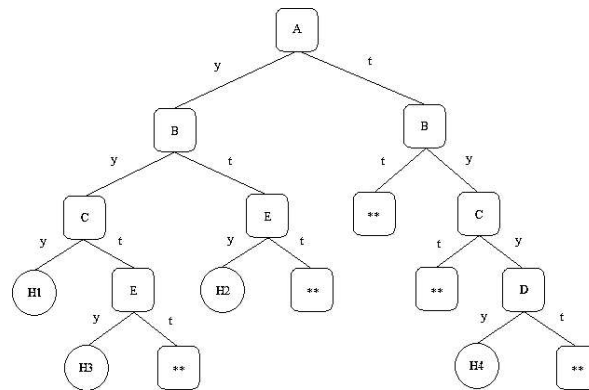
Menurut (Hartati & Iswanti, 2008: 26) Tabel keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Tabel keputusan merupakan matriks kondisi yang dipertimbangkan dalam pendeskripsian kaidah. Menurut (Hartati & Iswanti, 2008: 32) Berikut penyajian dalam bentuk tabel keputusan dan pohon keputusan.

Tabel 2. 1 Tabel Keputusan

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

(Sumber: Hartati, 2008).

Mengacu tabel 2.1, dapat dibuat pohon keputusan sebagai berikut:



Gambar 2. 3 Pohon Keputusan
(Sumber : Hartati, 2008)

Keterangan:

A = *evidence* A, H1 = hipotesa 1, y = ya

B = *evidence* B, H2 = hipotesa 2, t = tidak

C = *evidence* C, H3 = hipotesa 3, ** = tidak menghasilkan hipotesa tertentu

D = *evidence* D, H4 = hipotesa 4

Menurut (Hartati & Iswanti, 2008: 33-34) Dengan melihat pohon keputusan yang dihasilkan, dapat diketahui hipotesa H1 terpenuhi jika memenuhi *evidence* A, B, dan C. Hipotesa H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. Hipotesa H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. Hipotesa H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi *node (evidence)* di atasnya, notasi “t” artinya tidak memenuhi. Dapat dicontohkan untuk menghasilkan hipotesis H2 kalau diurutkan dari *node* A adalah memiliki *evidence* A, tidak memiliki *evidence* B, dan memiliki *evidence* E. Dalam hal ini jika tidak memenuhi *evidence* tertentu, maka diabaikan, sehingga hail

yang didapatkan adalah hiptesis H2 terpenuhi jika memiliki *evidence* A dan E saja.

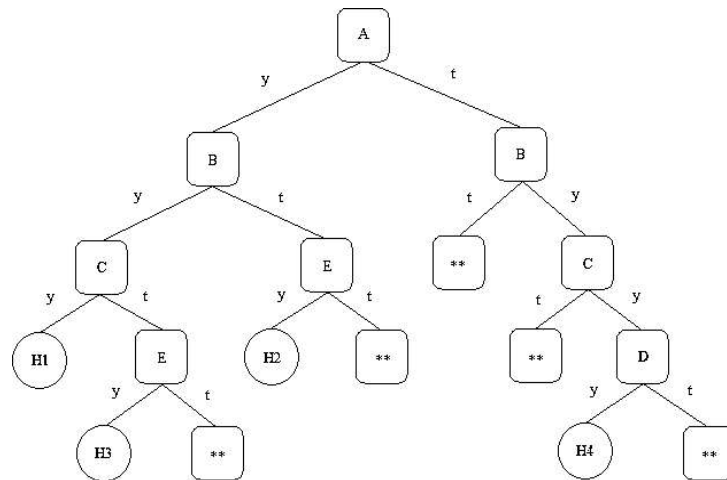
Dalam implementasi sistem pakar terutama dalam sesi konsultasi, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.4 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak” maka sistem akan menanyakan *evidence* B. Dengan kata lain apapun jawaban pengguna maka tidak akan mempengaruhi sistem. Hal ini dapat diatasi, salah satu caranya adalah dengan mengubah urutan pada tabel keputusan. Alternatif keputusan yang lain seperti terlihat pada tabel 2.2

Tabel 2. 2 Alternatif Tabel Keputusan

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

(Sumber: Hartati, 2008)

Mengacu tabel keputusan pada tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



Gambar 2. 4 Pohon Keputusan
(Sumber : Hartati, 2008)

Keterangan:

A = *evidence* A, H1 = hipotesa 1, y = ya

B = *evidence* B, H2 = hipotesa 2, t = tidak

C = *evidence* C, H3 = hipotesa 3, ** = tidak menghasilkan hipotesa tertentu

D = *evidence* D, H4 = hipotesa 4

Menurut (Hartati & Iswanti, 2008: 35-36) Telihat dari pohon keputusan gambar, 2.4 masing-masing node yang mewakili *evidence* tertentu kondisi “y” dan “t” sudah tidak mengarah pada *evidence* yang sama. Dalam sesi konsultasi hal yang mengandung arti jawaban pengguna yang berbeda, akan mengarah pada pertanyaan yang berbeda pula.

Proses akusisi pengetahuan dalam sistem pakar bukanlah hal yang mudah dan sederhana. Pengetahuan yang berhasil didapatkan dan disajikan dalam salah satu bentuk representasi pengetahuan misalnya disajikan dalam bentuk tabel keputusan dan dapat diubah susunannya seperti contoh kasus diatas. Secara teori sudah benar tetapi pada pengetahuan domain pengetahuan tertentu, perubahan

yang seperti itu terkadang tidak diperkenalkan. Dengan kata lain, pengetahuan yang akan disajikan dalam bentuk representasi tertentu harus mengikuti kaidah-kaidah kepakaran dalam domainnya dan tidak boleh bertentangan.

Menurut (Hartati & Iswanti, 2008: 38) Kaidah yang dapat dihasilkan dengan mengacu pohon keputusan gambar 2.5 adalah sebagai berikut:

Kaidah 1: *IF A and B and C THEN H1*

Kaidah 2: *IF A and B and C THEN H3*

Kaidah 3: *IF A and E then H2*

Kaidah 4: *IF D and B and C then H4*

1. Logika Predikat

Menurut (Hartati & Iswanti, 2008: 39-40) Logika yang pertama kali dikembangkan oleh Aristoteles merupakan logika formal. Logika Formal adalah logika yang berhubungan dengan bentuk (*syntax*) pernyataan bukan pada arti (*semantic*) dari pernyataannya. Logika formal berkonsentrasi pada penalarannya itu sendiri tanpa terganggu oleh obyek yang sedang dinalarkan.

Logika predikat berdasarkan pada kebenaran dan kaidah inferensi untuk merepresentasikan simbol-simbol dan hubungan satu dengan yang lainnya. Logika predikat selain digunakan untuk menentukan kebenaran (*truthfulness*) atau kesalahan (*falsity*) sebuah pernyataan, juga dapat digunakan untuk merepresentasikan pernyataan tentang objek tertentu.

2.1.2.6 Metode Inferensi

Menurut (Sutojo et al., 2011: 171) Pada sistem pakar berbasis *rule*, domain pengetahuan direpresentasikan dalam sebuah kumpulan *rule* berbentuk *IF-THEN*, sedangkan data direpresentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian saat ini. Mesin inferensi membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam *database*. Jika dibagian *IF* (kondisi) dari *rule* cocok dengan fakta, maka *rule* dieksekusi dan bagian *THEN* (aksi) diletakkan dalam *database* sebagai fakta baru yang ditambahkan

Menurut (Kusrini, 2008: 8) Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar proses inferensi dilakukan dalam suatu model yang disebut mesin inferensi. Ada dua metode Inferensi yang penting dalam sistem pakar yaitu:

2.2.2.6.1 Forward Chaining

Menurut (Sutojo et al., 2011: 171) *Forward chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan *IF* dari *rule 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 kedalam *database*. Setiap kali pencocokkan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi

sekali saja. Proses pencocokkan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. Metode pencarian yang digunakan adalah *Depth-First Search (DFS)*, *Breath-First Search (BFS)* atau *Best First Search*.

Menurut (Kusrini, 2008: 9-10) Ingin diperoleh konklusi dari daftar konklusi yang ada berdasarkan *premis-premis* dalam aturan dan fakta yang diberikan oleh *user*. Berikut ini daftar aturannya :

Aturan 9:

Jika Premis 1
Dan Premis 2
Dan Premis 3
Maka Konklusi 1

Aturan 10:

Jika Premis 1
Dan Premis 2
Dan Premis 4
Maka Konklusi 2

Aturan 11:

Jika Premis 1
Dan Premis 2
Dan Premis 5
Maka Konklusi 3

Aturan 12:

Jika Premis 1

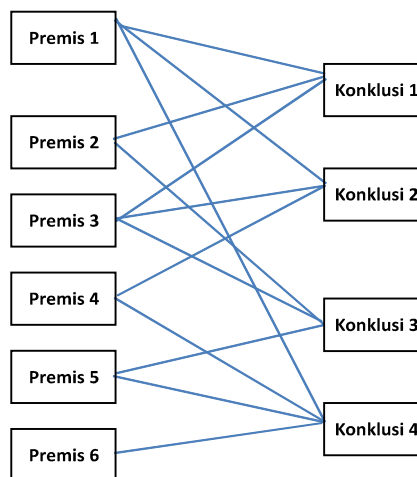
Dan Premis 4

Dan Premis 5

Dan Premis 6

Maka Konklusi 4

Penelusuran maju pada kasus ini adalah untuk mengetahui apakah suatu fakta yang dialami oleh pengguna itu termasuk konklusi 1, konklusi 2, konklusi 3, atau konklusi 4 atau bahkan bukan salah satu dari konklusi tersebut, yang artinya sistem belum mampu mengambil kesimpulan karena keterbatasan aturan.



Gambar 2. 5 Graph Pengetahuan
(Sumber : Kusriani, 2008)

Menurut (Kusriani, 2008: 10-11) Dalam penalaran ini, pengguna diminta memasukkan premis-premis yang alami. Untuk memudahkan pengguna, sistem dapat memunculkan daftar premis yang mungkin sehingga pengguna dapat memberikan umpan balik premis mana yang dialami dengan memilih satu atau beberapa dari daftar premis yang tersedia. Berarti premisnya adalah:

Premis 1, Premis 2, Premis 3, Premis 4, Premis 5 dan Premis 6

Berdasarkan *premis-premis* yang dipilih, maka sistem akan mencari aturan yang sesuai, sehingga akan diperoleh konklusinya. Seandainya *user* memilih *premis 1, premis 2, premis 3*, maka aturan yang terpilih adalah aturan 1 dengan konklusinya adalah Konklusi 1. Seandainya *user* memilih *Premis 1 dan Premis 6*, maka sistem akan mengarahkan pada aturan 4 dengan konklusinya adalah konklusi 4, tetapi aturan tersebut premisnya adalah *premis 1, premis 4, premis 5 dan premis 6*, maka *premis-premis* yang dipilih oleh *user* tidak cukup untuk mengambil kesimpulan konklusi 4 sebagai konklusi 4 sebagai konklusi terpilih.

2.2.2.6.2 Backward Chaining

Menurut (Sutojo et al., 2011: 178) *Backward chaining* adalah metode inferensi yang bekerja mundur ke arah kondisi awal. Proses diawali dari *Goal* (yang berada dibagian *THEN* dari *rule IF-THEN*), kemudian pencarian mulai dijalankan untuk mencocokkan apakah fakta-fakta yang ada cocok dengan premis-premis dibagian *IF*. Jika cocok, *rule* dieksekusi, kemudian hipotesis dibagian *THEN* ditempatkan dibasis data sebagai fakta baru. Jika tidak cocok, simpan *premis* dibagian *IF* ke dalam *stack* sebagai *subGoal*. Proses berakhir jika *Goal* ditemukan atau tidak ada *rule* yang bisa membuktikan kebenaran dari *subGoal* atau *Goal*.

2.1.3 Logika *Fuzzy* (*Fuzzy Logic*)

Menurut (Sutojo et al., 2011: 211-212) Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang sesuai untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan komputer, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1, artinya suatu keadaan memungkinkan mempunyai dua nilai “Ya” dan “Tidak” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan di berbagai bidang seperti pada sistem diagnosis penyakit (dalam bidang kedokteran); pemodelan sistem pemasaran, sistem operasi (dalam bidang ekonomi); kendali kualitas air, prediksi adanya gempa bumi, klasifikasi dan pencocokan pola (dalam bidang teknik).

Menurut (Budihartono & Suhartono, 2014: 151) Logika *Fuzzy* digunakan untuk menerjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*). Misalnya, besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. *Fuzzy Logic* dapat mengolah data yang tidak pasti berupa batasan, seperti “sangat”, dan “kurang lebih”. Manusia dapat dengan mudah mengartikan kalimat “Saya pergi sebentar saja”, mungkin sebentar bisa selama 4 atau 5 menit. Komputer tidak mengerti nilai asli dari kata “sebentar”. Dengan *fuzzy logic*, komputer dapat mengolah untuk memutuskan sesuatu yang membutuhkan kepintaran manusia dalam penalaran.

Menurut (Budihartono & Suhartono, 2014: 152) *Fuzzy logic* memiliki banyak kelebihan, yaitu dapat mengontrol sistem yang kompleks, *non-linier*, dan sistem yang sulit direpresentasikan secara sistematis. Berikut beberapa alasan menggunakan *fuzzy logic*:

1. Konsep *fuzzy logic* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. *Fuzzy logic* sangat fleksibel.
3. *Fuzzy logic* memiliki toleransi terhadap data-data yang tidak tepat.
4. *Fuzzy logic* mampu memodelkan fungsi-fungsi *non linier* yang sangat kompleks.
5. *Fuzzy logic* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui pelatihan.
6. *Fuzzy logic* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. *Fuzzy logic* didasarkan pada bahasa alami.

Menurut (Sutojo et al., 2011: 232) Struktur elemen dasar inferensi *fuzzy* berikut.

1. Basis pengetahuan *fuzzy*, yaitu kumpulan aturan (*rule*) *fuzzy* dalam bentuk pernyataan *IF...THEN*.
2. Fuzzifikasi, yaitu proses untuk mengubah *input* sistem yang mempunyai nilai tegas menjadi variabel linguistik menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.

3. Mesin inferensi, yaitu proses untuk mengubah *input fuzzy* menjadi *output fuzzy* dengan cara mengikuti aturan-aturan yang telah ditetapkan pada basis pengetahuan *fuzzy*.
4. Defuzzifikasi, yaitu mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzifikasi.

Menurut (Sutojo et al., 2011: 233-237) Metode yang digunakan dalam sistem inferensi *fuzzy* adalah :

1. Metode Tsukamoto

Dalam inferensinya, metode Tsukamoto menggunakan tahapan sebagai berikut.

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

2. Metode Mamdani

Metode digunakan dalam aplikasi-aplikasi karena strukturnya yang sederhana, yaitu menggunakan operasi MIN-MAX atau MAX-PRODUCT . Untuk mendapatkan *output*, diperlukan 4 tahapan sebagai berikut:

- a. Fuzzifikasi
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)

- c. Aplikasi fungsi implikasi menggunakan fungsi *MIN* (*Minimum*) dan komposisi antar-*rule* menggunakan fungsi *MAX* (*Maximum*) dengan menghasilkan himpunan *fuzzy* baru
 - d. Defuzzifikasi menggunakan metode *Centroid* (Titik Tengah)
3. Metode Sugeno

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

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

2.1.4 Jaringan Saraf Tiruan (*Artificial Neural Network*)

Menurut (Sutojo et al., 2011: 283) Jaringan saraf tiruan adalah paradigma pengolahan informasi yang terinspirasi oleh sistem saraf secara biologi, seperti proses informasi pada otak manusia. Elemen kunci paradigma adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), berkerja serentak untuk menyelesaikan masalah tertentu. Cara kerja JST seperti manusia, yaitu belajar melalui contoh.

2.1.4.1 Kelebihan Jaringan Saraf Tiruan

Menurut (Sutojo et al., 2011: 284) Kelebihan-kelebihan yang diberikan jaringan saraf tiruan antara lain:

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

2.1.4.2 Kelemahan Jaringan Saraf Tiruan

Menurut (Sutojo et al., 2011: 284-285) Selain mempunyai beberapa kelebihan, jaringan saraf tiruan juga mempunyai kelemahan-kelemahan berikut:

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

2.1.4.3 Arsitektur Jaringan Saraf Tiruan

Menurut (Sutojo et al., 2011: 292) Seperti yang telah disebutkan sebelumnya, baik tidaknya suatu model JST salah satunya ditentukan oleh hubungan antar *neuron*. *Neuron-neuron* tersebut terkmpul dalam lapisan-lapisan yang disebut *neuron layer*. Ada 3 penyusun lapisan-lapisan JST, yaitu:

1. Lapisan *Input (Input Layer)*

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

2. Lapisan Tersembunyi (*Hidden Layer*)

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

3. Lapisan *Output (Output Layer)*

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

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

1. Jaringan Lapisan Tunggal

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

2. Jaringan Lapisan Banyak

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

3. Jaringan dengan Lapisan Kompetitif

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

2.1.5 Algoritma Pembelajaran Dengan Supervisi

Menurut (Sutojo et al., 2011: 307-387) Algoritma pembelajaran dengan supervisi antara lain:

a. *Hebb-Rule*

Pada tahun 1949, D.O. *Hebb* memperkenalkan cara menghitung bobot w dan bias secara iteratif dengan memanfaatkan model pembelajaran dengan supervisi sehingga bobot w dan bias dapat dihitung secara otomatis tanpa harus melakukan coba-coba. Model *Hebb* merupakan model jaringan tertua yang menggunakan pembelajaran dengan dengan supervisi. Arsitektur jaringan *Hebb* sama seperti arsitektur jaringan *McCulloch-Pitts*, yaitu beberapa unit input

dihubungkan langsung dengan sebuah unit *output*, ditambah dengan sebuah bias.

b. Perceptron

Model jaringan *perceptron* merupakan model yang paling baik pada saat itu. Ditemukan oleh Rosenblatt (1962) dan Minsky-Papert (1969).

c. Delta-Rule

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

d. Backpropagation

Backpropagation adalah metode penurunan gradien untuk meminimalkan kuadrat *error* keluaran. Ada tiga tahap yang harus dilakukan dalam pelatihan jaringan, yaitu tahap perambatan maju (*forward propagation*), tahap perambatan balik, dan tahap perubahan bobot dan bias. Arsitektur jaringan ini terdiri dari *input layer*, *hidden layer*, dan *output layer*.

e. Heteroassociative Memory

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

f. Bidirectional Associative Memory (BAM)

Bidirectional Associative Memory (BAM) adalah model jaringan saraf yang memiliki 2 lapisan, yaitu lapisan *input* dan lapisan *output* yang mempunyai

hubungan timbal balik antara keduanya. Arsitektur jaringan ini terdiri dari 3 *neuron* pada lapisan *input* dan 2 *neuron* pada lapisan *output*.

g. *Learning Vector Quantization (LVQ)*

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

2.1.6 Pelatihan Tanpa Supervisi (Jaringan Kohonen)

Menurut (Sutojo et al., 2011: 392) Prof. Teuvo Kohonen adalah orang yang pertama memperkenalkan jaringan kohonen pada 1982. Pada jaringan ini, *neuron-neuron* pada suatu lapisan data akan menyusun dirinya sendiri berdasarkan *input* nilai tertentu dalam suatu *cluster*. *Cluster* yang dipilih sebagai pemenang adalah *cluster* yang mempunyai vektor bobot paling cocok dengan pola *input* (memiliki jarak yang paling dekat).

2.1.7 Android

Menurut (Supriadi, 2017: 1-2) *Android* merupakan sebuah sistem operasi perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Beberapa pengertian lain *android*, yaitu:

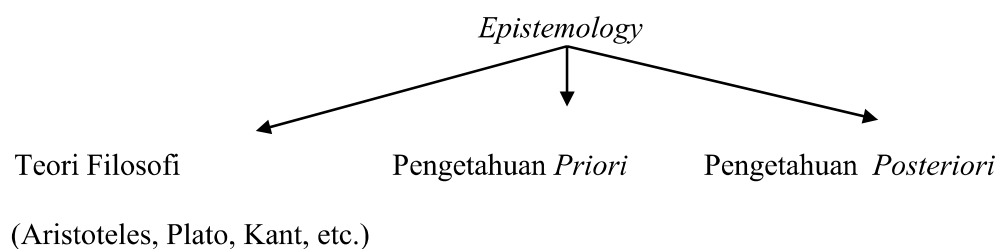
1. Merupakan *platform* terbuka (*open source*) bagi para pengembang (*programmer*) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli *Google Inc.* dari *Android Inc.*

3. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *run time environment* yang disebut DVM (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil.

Untuk mengembangkan *android*, dibentuk OHA (*Open Handset Alliance*), yaitu konsorsium dari 34 perusahaan peranti keras (*hardware*), peranti lunak (*software*), dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Nvidia*.

2.1.8 Pengetahuan (*Knowledge*)

Menurut (Sutojo et al., 2011: 123-125) Pengetahuan (*knowledge*) adalah suatu yang hadir dan terwujud dalam jiwa dan pikiran seseorang karena adanya reaksi, sentuhan, dan hubungan lingkungan dan alam sekitarnya. Definisi lain dari pengetahuan adalah fakta atau keadaan yang timbul karena suatu pengalaman. Contoh: pengetahuan tentang binatang, sifat-sifat dan peilakunya. Pengetahuan tentang penyakit, gejala-gejala, dan pengobatannya. Pengetahuan tentang tanaman, jenis-jenisnya dan cara hidupnya, dan lain-lain. Cabang ilmu yang mempelajari pengetahuan disebut "*epistemology*". Berikut adalah cabang-cabang dari *epistemology*.



Gambar 2. 6 Cabang-Cabang *Episemology*
(Sumber : Sutojo, 2011)

1. Pengetahuan *Priori*

Pengetahuan *priori* adalah pengetahuan yang datang sebelumnya dan bebas dari arti. Pengetahuan ini mempunyai kebenaran yang *universal* dan tidak dapat disangkal tanpa kontradiksi. Contoh: gula rasanya manis, matahari terbit di sebelah timur, jumlah seluruh sudut segitiga adalah 180^0 , dan lain-lain.

2. Pengetahuan *Posteriori*

Pengetahuan *posteriori* adalah pengetahuan yang diturunkan dari akal pikiran yang sehat. Pengetahuan ini mempunyai kebenaran atau kesalahan yang dapat dibuktikan dengan menggunakan pengalaman akal sehat. Contoh: air yang telah diberi gula akan terasa manis, tetapi bisa juga terasa asin bila air tadi diberi garam yang berlebihan.

2.1.9 Validasi Sistem

Menurut (A.S & Shalahuddin, 2011: 213-214) Pengujian integrasi lebih pada pengujian penggabungan dari dua atau lebih unit pada perangkat lunak. Setelah integrasi maka dilakukan pengujian sistem dimana unit-unit proses yang sudah diintegrasikan diuji dengan antarmuka yang sudah dibuat sehingga pengujian ini dimaksudkan untuk menguji sistem perangkat lunak secara keseluruhan dan diuji secara satu sistem (tidak terpisah-pisah lagi). Pengujian untuk validasi memiliki beberapa pendekatan sebagai berikut:

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

Menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem atau perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan *black-box testing* harus dibuat dengan kasus benar dan kasus salah.

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

Menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih dilakukan dengan memeriksa logik dan kode program. Pembuatan kasus uji bisa mengikuti standar pengujian dari standar yang seharusnya.

2.2 Variabel

Menurut (Sugiyono, 2012: 38) Variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulan.

2.2.1 *Trichodina sp*

Menurut (Samadi, 2016: 96-97) *Trichodina sp* dan *trichodinella sp* merupakan parasit. Ia tergolong hewan bersel tunggal (*protozoa*) berbulu getar, kedua parasit tersebut berukuran mikroskopis dan hanya dapat dilihat dengan menggunakan alat bantu mikroskop. Parasit ini berbentuk lingkaran transparan dengan sejumlah bulu-bulu getar (*silia*) yang menempel di sekeliling lingkaran itu. Pada tubuh bagian bawah terdapat tubuh bagian perekat yang berfungsi untuk melekatkan dirinya ke tubuh mangsanya. Parasit ini menginfeksi kulit, sirip, dan insang. Ia menyerang ikan lele segala ukuran, baik yang masih kecil maupun yang sudah besar. Infeksi parasit ini mengakibatkan kerusakan pada insang dan tubuh ikan, dan penyakitnya dikenal sebutan penyakit "*Trichodiniasis*". Gejala ikan berbintik-bintik dibagian kepala dan punggung, kurus dan lemah, nafsu makan berkurang, tubuh kusam. Ikan yang terinfeksi ini juga tampak sering menggosok-gosokan tubuhnya ke dinding kolam, dasar kolam, atau benda keras. Pengobatan penyakit *trichodiniasis* diobati dengan menggunakan formalin dengan dosis 40 ppm, yang kemudian ikan-ikan yang sakit direndam dalam larutan tersebut selama satu hari satu malam (24 jam). Perlakuan ini dilakukan selama 4-7 hari.

Menurut (Irmawan, 2016: 57) Penyakit ini disebabkan oleh *protozoa*, di mana mereka menyerang bagian insang dari ikan lele. Ikan yang terserang oleh penyakit ini akan berputar-putar dan muncul diatas permukaan air. Tindakan pengobatan yang dapat dilakukan adalah merendam ikan lele yang sakit ke dalam air berformalin berkonsentrasi 15-20 ppm.

2.2.2 *Dactylogyrus sp* dan *Gyrodactylus sp*

Menurut (Samadi, 2016: 94-95) *Dactylogyrus sp* dan *gyrodactylus sp* merupakan parasit. Ia adalah cacing yang tergolong kelas *trematoda*, termasuk cacing *monogenea*. Tubuhnya berbentuk pipih dan pada bagian ujung tubuhnya terdapat alat yang berfungsi sebagai alat penghisap darah mangsanya. Gejala penyakit ini ditandai dengan kondisi ikan kurus, kulit berlendir dan tubuh ikan tampak tidak cerah, sirip rontok, tutup insang agak terbuka, lemah, nafsu makan menurun, megap-megap seperti kekurangan oksigen. Ikan yang terserang penyakit ini juga sering menggosok-gosokkan tubuhnya pada dinding kolam atau pada benda-benda keras lainnya. Pengobatan gangguan penyakit *dactylogyriasis* dan *gyrodactyliasis* dapat diobati dengan *Methylene blue* dengan dosis 1 g/m³ air, lalu ikan yang sakit direndam dalam larutan tersebut selama 30 menit. Selain itu, dapat juga menggunakan garam dapur dengan dosis 13 g/m³ air, *formalin* dengan dosis 40 ppm. Ikan sakit direndam dalam larutan tersebut selama satu hari satu malam (24 jam).

Menurut (Irmawan, 2016: 56) Jenis cacing yang dapat menyerang adalah *dactylogyrus* dan *gyrodactylus*, di mana mereka sering dijumpai di kolam yang berkepadatan tebarnya terlalu tinggi serta baru saja mengalami perubahan lingkungan hidup yang drastis dan mendadak. Mereka sering menyerang bagian insang dan ikan lele (akan menyebabkan kesulitan dalam hal bernapas) serta kulitnya (menjadi berlendir). Gejala yang ditunjukkan ikan lele yang terserang oleh penyakit ini adalah menurun nafsu makan serta ikan lele sering berenang ke atas permukaan air. Beberapa tindakan pengobatan yang dapat dilakukan adalah

mengganti air dalam jumlah yang besar, air kolam ditaburi dengan garam dapur sejumlah 40 grm/m², merendam ikan lele yang sakit ke dalam air PK berkonsentrasi 0,01% selama 30 menit.

2.2.3 *Ichthyophthirius Multifiliis*

Menurut (Samadi, 2016, pp. 91–92) *Ichthyophthirius Multifiliis* merupakan jenis parasit. Ia tergolong hewan bersel tunggal (*protozoa*). Infeksi parasit tersebut menimbulkan penyakit bercak putih. Nama lain penyakit bercak putih yaitu “*Ich*” atau “*Ichthyophthiriasis*” atau lebih populer dengan sebutan *white spot*. Gejala ikan tampak malas berenang, pucat, sering menggosok-gosokan badannya pada dinding kolam atau dasar kolam, ada bercak-bercak putih pada tubuh, megap-megap, dan ikan juga sering berada di dekat pintu pemasukan air, serta tampak mengapung di permukaan air. Bagian tubuh terinfeksi menjadi pucat dan bengkak. Pengobatan pengobatan penyakit yang disebabkan oleh parasit. *Ichthyophthirius Multifiliis* dapat menggunakan garam (NaCl) dengan dosis 1-3 g/liter, lalu ikan yang sakit direndam dalam larutan garam dapur tersebut selama 5-10 menit. Selain itu, dapat juga menggunakan *Methilene blue* dengan dosis 1 g/liter air. Lalu ikan yang sakit direndam ke dalam lautan *Methilene blue* dari larutan baku tersebut dengan dosis 0,5-1 cc/liter air selama 24 jam. Pengobatan diulang hingga 3 kali atau lebih dengan selang waktu satu hari.

Menurut (Irmawan, 2016: 54) Penyebab dari munculnya penyakit ini adalah *ichthyophthirius multifiliis* dimana mereka akan menyerang ikan lele yang dipelihara di dalam kolam yang airnya menyerang. Gejala yang ditunjukkan oleh

penyakit ini bahwa pada permukaan kulit dan juga insang. Ikan lele banyak dijumpai bintik-bintik berwarna putih yang apabila dibiarkan terlalu lama, kulit dan insang ini akan rusak sebelum pada akhirnya nanti ikan lele akan mati dalam hitungan jam. Beberapa tindakan pengobatan yang dapat dilakukan adalah dengan memperbaiki sistem sanitasi. Air kolam ditaburi dengan garam dapur sejumlah 30gram/liter air, sebanyak 2-3 kali berturut-turut, penggunaan *malachyte green* berdosisi 0,1 gram/m² sebanyak 2 hari sekali hingga ikan lele sembuh.

2.3 Software Pendukung

2.3.1 Eclipse

Menurut penelitian (Pambudi, 2013: 111) dalam Pisa (2009), *eclipse* adalah sebuah pengembangan lingkungan yang terintegrasi yang membantu anda untuk membangun aplikasi-aplikasi dalam banyak bahasa. *Eclipse* adalah *software opensource* yang didukung oleh *Eclipse foundation*, sebuah asosiasi *nonprofit* yang serupa dalam ruang lingkup dasar *Apache*. *Eclipse* dapat berjalan di beberapa sistem operasi dengan mudah. Inti dari *Eclipse* adalah *plug-in manager* yang dapat memuat, menginstal, dan menjalankan berbagai *plug-in*, masing-masing seperti komponen yang menyediakan berbagai jenis layanan. Terdapat *plug-in* yang mengetes, *debug*, menjalankan kode, dan membantu menulis kode dalam banyak bahasa yang berbeda, seperti AS, *javascript*, CF, PHP dan lain-lain. Bahasa umumnya adalah selalu *Java*.

Menurut penelitian (Anggraini, Ardianty, & Widiyanto, 2014: 241) *Eclipse* adalah IDE (*Integrated Development Environment*) yang paling populer untuk pengembangan *Android*, karena memiliki *plug-in* yang tersedia untuk memfasilitasi pengembangan *Android*. *Eclipse* juga mendapat dukungan langsung dari *Google* hal ini terbukti dengan adanya penambahan *plug-in* untuk membuat *project Android* dimana *source software* langsung dari situs resminya *Google*.

2.3.2 Java

Menurut penelitian (Pambudi, 2013: 111) dalam Jeff Friesen (2010) *java* adalah *platform* untuk mengeksekusi program dan terdiri dari *library* untuk menjalankan program dan berinteraksi dengan sistem operasi yang mendasarinya. *Java* adalah bahasa pemrograman yang sederhana, berorientasi objek, terdistribusi, diinterpretasikan, kuat, aman, arsitektur netral, *portable*, kinerja tinggi, *multithreaded*, dan dinamis. *Java* dapat dijalankan diberbagai jenis sistem operasi dan arsitektur komputer. Menurut penelitian (Lengkong, Sinsuw, & Lumenta, 2015: 21) *Java* adalah bahasa berorientasi objek yang dapat digunakan untuk pengembangan aplikasi mandiri, aplikasi berbasis *internet*, serta aplikasi untuk perangkat perangkat cerdas yang dapat berkomunikasi lewat *internet* atau jaringan komunikasi. Dalam *java* ada 2 (program berbeda, yaitu aplikasi dan adalah program yang biasanya disimpan dari komputer lokal sedangkan yang biasanya disimpan pada komputer yang jauh, yang dikoneksikan pemakai lewat *java* bukan turunan langsung dari bahasa manapun. OOP (*object oriented*

programming) cara yang ampuh dalam pengorganisasian dan pengembangan perangkat lunak.

2.3.3 UML (*Unified Modeling Language*)

Menurut (A.S & Shalahuddin, 2011: 113) *UML (Unified Modeling Language)* adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

2.3.3.1 Pengenalan UML (*Unified Modeling Language*)

Menurut (A.S & Shalahuddin, 2011: 117) Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

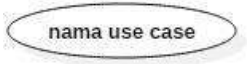
Menurut (A.S & Shalahuddin, 2011: 120-121) Pada *UML 2.3* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut:

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



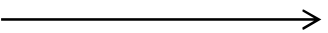
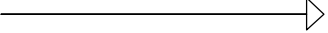
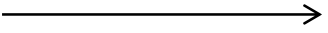
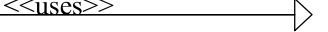
2.3.3.2 Use Case Diagram

Menurut (A.S & Shalahuddin, 2011: 130) *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2. 3 Use CaseDiagram

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

Tabel 2.3 Lanjutan

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

Sumber: (A.S & Shalahuddin, 2011)

2.3.3.3 Activity Diagram






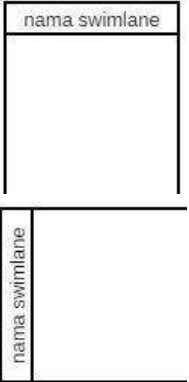
Menurut (A.S & Shalahuddin, 2011: 134-135) Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokan tampilan dari sistem atau *user interface* di mana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2. 4 *ActivityDiagram*



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

Sumber: (A.S & Shalahuddin, 2011)


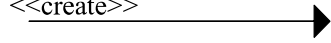
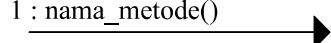
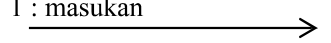
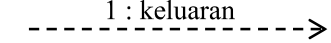
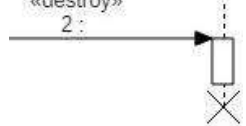
2.3.3.4 *Sequence Diagram*

Menurut (A.S & Shalahuddin, 2011: 137-139) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyak diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol yang ada pada diagram sekuen:

Tabel 2.5 *SequenceDiagram*

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

Tabel 2.5 Lanjutan

 <p>Waktu aktif</p>	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
<p>Pesan tipe <i>create</i> <code><<create>></code></p> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat
<p>pesan tipe <i>call</i> 1 : nama_metode()</p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.
<p>Pesan tipe <i>send</i> 1 : masukan</p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju
<p>pesan tipe <i>return</i> 1 : keluaran</p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu. Arah panah mengarah pada objek penerima
<p>Pesan tipe <i>destroy</i> <code><<destroy>></code> 2 :</p> 	Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri

Sumber: (A.S & Shalahuddin, 2011)

2.3.3.5 Class Diagram

Menurut (A.S & Shalahuddin, 2011: 122) Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case*

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

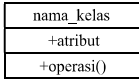
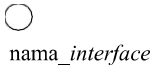





4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Menurut (A.S & Shalahuddin, 2011: 122) Jenis-jenis kelas di atas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

Menurut (A.S & Shalahuddin, 2011: 123-124) Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan apa yang disebut dengan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam sebuah kelas. Sebagai aturan secara umum maka sebuah metode yang dibuat harus memiliki kadar *cohesion* yang kuat dan kadar *coupling* yang lemah. Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2. 6 Class Diagram

Simbol	Deskripsi
kelas 	kelas pada struktur sistem
antarmuka / <i>interface</i> 	sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
asosiasi / <i>association</i> 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
asosiasi berarah / <i>directed association</i> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
kebergantungan / <i>dependency</i> 	relasi antar kelas dengan makna kebergantungan antar kelas
agregasi / <i>aggregation</i> 	relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber: (A.S & Shalahuddin, 2011)

2.4 Penelitian Terdahulu

Penelitian terdahulu merupakan penelitian-penelitian yang dilakukan oleh para ahli terdahulu sebelum penelitian ini. Hasil penelitian-penelitian tersebut dijadikan referensi dalam penelitian ini, baik variabel-variabel terkait dan asumsi-asumsi yang relevan dari hasil penelitian tersebut. Sebagai bahan pertimbangan dalam penelitian ini, maka peneliti mencantumkan beberapa penelitian yang diambil dari beberapa jurnal ilmiah, yaitu:

1. (Hernawati, 2015) “**Inventarisasi Patogen pada Ikan Botia (*Chromobotia macracanthus Bleeker*) di Stasiun Karantina Ikan Kelas I Supadio,**

Pontianak”, diperoleh fakta: Penyakit merupakan kendala utama dalam budidaya ikan hias, salah satunya adalah ikan botia. Banyaknya jenis penyakit yang menyerang ikan botia dapat menimbulkan kerugian besar. Upaya yang dapat dilakukan untuk meminimalisir terjadinya kerugian tersebut adalah dengan mencegah tersebarnya penyakit. Tujuan penelitian ini adalah untuk mengetahui jenis *patogen* yang menyerang ikan botia (*Chromobotia macracanthus Bleeker*) yang dilalulintaskan melalui Stasiun Karantina Ikan Kelas I Supadio, Pontianak. Pengamatan yang dilakukan adalah identifikasi parasit, bakteri dan jamur. Hasil pengamatan menunjukkan bahwa agen penyakit yang menyerang ikan botia yaitu *Gyrodactylus sp.*, *Myxobolus sp.*, *Argulus sp.*, *Vorticella sp.* dari agen parasit, *Neisseria sp.*, *Aeromonas hydrophila*, *Chromobacterium violaceum*, *Pasteurella haemolytica* dari agen bakteri dan *Saprolegnia sp.* dari agen jamur . Seluruh agen penyakit yang teramati bukan termasuk golongan Hama Penyakit Ikan Karantina.

2. (Tamin, 2015) “**Sistem Pakar untuk Diagnosa Kerusakan Pada Printer Menggunakan Metode Forward Chaining**”, diperoleh fakta: Mensubstitusikan pengetahuan manusia ke dalam bentuk sistem dengan menampung kemampuan/keahlian seorang pakar untuk melakukan proses analisa suatu masalah sehingga sistem dapat sistem bekerja menyelesaikan masalah sebagaimana manusia mengerjakannya dan menyelesaikan masalah tersebut. Kerusakan *printer* terkadang menjadi masalah besar ketika seorang yang awam tidak mengetahui letak kesalahan *printer* maka dibutuhkan sistem yang mampu bekerja otomatis untuk memberikan solusi kerusakan *printer*. Penelitian ini

bertujuan untuk merancang sebuah sistem yang dapat digunakan untuk menangani kerusakan *printer*. Pengguna aplikasi ini seolah-olah berhadapan langsung dengan pakar dibidang *hardware* khususnya *printer*. Perencanaan sistem dilakukan dengan membuat *knowledge base* menggunakan *decision tree* dan Aturan *if-then* sebagai representasi pengetahuan. Sistem dibuat dengan menggunakan metode *forward chaining* dan bahasa pemrograman *Visual Basic*. Hasil penelitian ini mengungkapkan jenis-jenis kerusakan yang terjadi pada sebuah *printer* serta penanganan dari kerusakan tersebut. Pengujian aplikasi juga dilakukan untuk mengetahui akurasi dan variasi serta *user freindly* dan fleksibilitas sistem. Hasil dari keseluruhan pengujian ini dapat disimpulkan bahwa program sudah cukup baik walaupun jenis kerusakan yang dihasilkan belum lengkap karena pada sistem ini hanya mendeteksi 15 jenis kerusakan mesin secara umum.

3. (Hamdani, 2010) “**Sistem Pakar Untuk Diagnosa Penyakit Mata Pada Manusia**”, diperoleh fakta: Mata merupakan suatu panca indra yang sangat penting dalam kehidupan manusia untuk melihat. Jika mata mengalami gangguan atau penyakit mata, maka akan berakibat sangat fatal bagi kehidupan manusia. Jadi sudah mestinya mata merupakan anggota tubuh yang perlu dijaga dalam kesehatan sehari-hari. Sistem pakar merupakan suatu bagian metode ilmu-ilmu *artificial intelligence* untuk dibuat suatu program aplikasi diagnosa penyakit mata pada manusia yang terkomputerisasi serta berusaha menggantikan dan menirukan proses penalaran dari seorang ahlinya atau pakar dalam memecahkan masalah spesifikasi yang dapat dikatakan duplikat dari

seorang pakar karena pengetahuan ilmu tersebut tersimpan di dalam suatu sistem *database*. Sistem Pakar Untuk Diagnosa Penyakit Mata Pada Manusia menggunakan metode *forward chaining* bertujuan menelusuri gejala yang ditampilkan dalam bentuk pertanyaan-pertanyaan agar dapat mendiagnosa jenis penyakit dengan perangkat lunak berbasis *desktop management system*. Perangkat lunak sistem pakar dapat mengenali jenis penyakit mata setelah melakukan konsultasi dengan menjawab beberapa pertanyaan-pertanyaan yang ditampilkan oleh aplikasi sistem pakar serta dapat menyimpulkan beberapa jenis penyakit mata yang di derita oleh pasien. Data penyakit yang dikenali menyesuaikan *rules* (aturan) yang dibuat untuk dapat mencocokkan gejala-gejala penyakit mata dan memberi nilai persentase agar mengetahui nilai pendekatan jenis penyakit pasien.

4. (Ichwan & Hakiky, 2011) **“Pengukuran Kinerja *Goodreads Application Programming Interface (API)* pada Aplikasi *Mobile Android (Studi Kasus untuk Pencarian Data Buku)*”**, diperoleh fakta: Metode pencarian adalah cara menemukan data pada kumpulan data untuk menghasilkan informasi yang penting, saat ini metode pencarian yang digunakan sangat banyak salah satunya *Goodreads*. *Goodreads* adalah situs jejaring sosial yang mengkhususkan pada katalog buku yang menyediakan *Application Programming Interface (API)* yang dijalankan pada Teknologi *Web 2.0*. Metode *Goodreads* digunakan karena menyediakan *Application Programming Interface (API)* memungkinkan pengembang perangkat lunak mengakses kedalam *database*. Metode tersebut diimplementasikan pada *smartphone* berbasis *Android*. *Android* sebagai sistem

operasi yang dapat ditanamkan pada perangkat *smartphone* yang memiliki kemampuan diinstal aplikasi yang diperlukan oleh pengguna. Aplikasi tersebut untuk mengukur kinerja *Goodreads application programming interface* pada pencarian data buku. Berdasarkan pengujian dan penelitian yang dilakukan, melakukan pengukuran kinerja pada suatu aplikasi sangat dibutuhkan untuk mengetahui performa Aplikasi. Aplikasi “*Books on Goodreads*” yang dibangun dapat dimanfaatkan untuk melakukan pencarian data buku menggunakan metode *Goodreads* dengan memanfaatkan API, hasil pencarian dapat disimpan di *database*, untuk mengambilnya kembali tanpa harus melakukan pencarian dan tidak memerlukan koneksi *internet*.

5. (Baldissera et al., 2017) “*Melaleuca alternifolia essential oil preventoxidative stress and ameliorates the antioxidant system in the liver of silver catfish (Rhamdia quelen) naturally infected with Ichthyophthirius multifiliis*”, diperoleh fakta:

Oxidative stress has been linked to a pathological mechanism that contributes to the initiation and progression of hepatic injury during Ichthyophthirius multifiliis infection. Thus, natural compounds with antioxidant and free radical scavenger properties, such as the Melaleuca alternifolia essential oil (TTO), might help to prevent or reduce hepatic damage. Thus, the aim of this study was to investigate whether TTO is capable of preventing or reducing hepatic oxidative damage in silver catfish naturally infected with I. multifiliis. After a TTO bath treatment of 1 h/day for 4 days, the number of trophonts on the skin of catfish was reduced by 94.87% compared to the control group. Hepatic samples from fish infected by I. multifiliis showed increased thiobarbituric acid reactive substances (TBARS) and protein carbonylation levels, while catalase (CAT) activity decreased compared to uninfected and untreated animals. Moreover, the histopathology analyses demonstrated the presence of necrosis, nuclear lateralization and cariorexsis, as well as swelling and vacuolization of cytoplasm. Treatment with TTO reduced hepatic

*TBARS and protein carbonylation levels and ameliorated the antioxidant system of infected fish, as well as preserve the hepatocytes morphology, but not prevented the occurrence of necrosis. Based on our findings, we propose that TTO is an alternative therapeutic option for silver catfish infected with *I. multifiliis*. Moreover, TTO treatment is a useful approach for avoiding or minimizing the hepatic oxidative stress caused by *I. multifiliis*, thus improving the hepatic antioxidant system.*

(Baldissera et al., 2017) **“Minyak esensial *Melaleuca alternifolia* mencegah stres oksidatif dan memperbaiki sistem antioksidan di hati ikan lele perak (*Rhamdia quelen*) yang secara alami terinfeksi *Ichthyophthirius multifiliis*”**, diperoleh fakta: Stres oksidatif telah dikaitkan dengan mekanisme patologis yang berkontribusi terhadap inisiasi dan perkembangan cedera hati selama infeksi *Myophyrius multifiliis*. Dengan demikian, senyawa alami dengan antioksidan dan sifat pemulung radikal bebas, seperti minyak *atsiri melaleuca* (TTO), dapat membantu mencegah atau mengurangi kerusakan hati. Dengan demikian, tujuan dari penelitian ini adalah untuk mengetahui apakah TTO mampu mencegah atau mengurangi kerusakan oksidatif hati pada ikan lele perak yang secara alami terinfeksi dengan *I. multifiliis*. Setelah mandi TTO 1 hari / hari selama 4 hari, jumlah trofont pada kulit ikan lele berkurang sebesar 94,87% dibandingkan kelompok kontrol. Sampel hepatic dari ikan yang terinfeksi oleh *I. multifiliis* menunjukkan peningkatan zat reaktif asam *thiobarbiturat* (TBARS) dan kadar karbonilasi protein, sedangkan aktivitas katalase (CAT) menurun dibandingkan dengan hewan yang tidak terinfeksi dan tidak diobati. Selain itu, analisis histopatologi menunjukkan adanya *nekrosis*, *lateralisasi nuklir* dan *kariorraxis*, serta pembengkakan dan vakuolisasi

sitoplasma. Pengobatan dengan TTO mengurangi TBARS hati dan kadar karbonilasi protein dan memperbaiki sistem antioksidan ikan yang terinfeksi, serta mempertahankan morfologi hepatosit, namun tidak mencegah terjadinya nekrosis. Berdasarkan temuan kami, kami mengusulkan bahwa TTO adalah pilihan terapeutik alternatif untuk ikan lele perak yang terinfeksi. *multifiliis*. Selain itu, pengobatan TTO adalah pendekatan yang berguna untuk menghindari atau meminimalkan stres oksidatif hati yang disebabkan oleh *I. multifiliis*, sehingga memperbaiki sistem antioksidan hati.

6. (Bebak & Wagner, 2012) ***“Use of vaccination against enteric septicemia of catfish and columnaris disease by the U.S. catfish industry”***, diperoleh fakta:

Vaccination is an effective strategy used for the protection of food animals against infectious diseases. A 2010 U.S. Department of Agriculture questionnaire examined U.S. catfish industry use (in 2009) of two commercial vaccines that provide protection against enteric septicemia of catfish (ESC) and columnaris disease, catfish producers' opinions regarding the percentage of vaccinated fish they expect to be protected, and producers' general expectations regarding survival of vaccinated fish compared with unvaccinated fish. During 2009, 9.7 % of the total fingerling operations used one or both vaccines; 12.3% of the total industry fry production was vaccinated against ESC, and 17.0% was vaccinated against columnaris disease. Of the producers who grew food-sized catfish to harvest, 6.7% used vaccinated catfish. The farms that did not use vaccinated fish for grow out had a mean size of 63.4 water surface hectares (156.6 water surface acres). The operations that used vaccinated fish were larger (mean size= 206.6 water surface hectares, or 510.6 water surface acres). The producers that stocked ESC-vaccinated fish for grow out represented 19.0% of the total water surface area of food fish production; producers that stocked columnaris-vaccinated fish represented 16.6% of the total area. Of the producers that stocked ESC-vaccinated catfish, 41.9% thought that survival was better in vaccinated fish than unvaccinated fish; of the producers that stocked columnaris-vaccinated catfish, 46.2% thought that vaccinated fish displayed better survival. However, 37.5% of producers that used the ESC vaccine and 39.7% of producers that used the columnaris vaccine did not know

whether vaccination improve d survival rates. When all producers were asked about their expectations regarding the percentage of vaccinated fish that would be protected from disease, 52.4% responded that they expected 100% of their fish to be protected. More producer information about reason able expectations regarding vaccine efficacy, the condition sunder which immunosuppression and vaccine failure can occur, and assessment of vaccine performance may result in increased use of vaccination as a tool for the catfish industry.

(Bebak & Wagner, 2012) **“Penggunaan vaksinasi terhadap *septicemia enterik* penyakit ikan patin dan penyakit koloni oleh industri ikan lele A.S”**, diperoleh fakta: Vaksinasi adalah strategi efektif yang digunakan untuk perlindungan hewan makanan terhadap penyakit menular. Kuesioner Departemen Pertanian AS tahun 2010 meneliti penggunaan industri ikan patin AS (pada 2009) dari dua vaksin komersial yang Memberikan perlindungan terhadap *septicemia enterik* penyakit ikan lele (ESC) dan penyakit kolumnaris, pendapat produsen lele Mengenai *persentase* ikan yang divaksinasi yang mereka harapkan dapat dilindungi, dan harapan umum produsen mengenai Kelangsungan hidup ikan yang divaksinasi dibandingkan dengan ikan yang tidak divaksinasi. Selama tahun 2009, 9,7% dari total operasi fingerling Digunakan satu atau kedua vaksin; 12,3% dari produksi total industri kentang telah divaksinasi terhadap ESC, dan 17,0% divaksinasi terhadap penyakit kolumaris. Dari produsen yang menanam ikan patin berukuran besar sampai panen, 6,7% menggunakan ikan patin yang divaksinasi. Peternakan yang tidak menggunakan ikan yang divaksinasi untuk tumbuh memiliki ukuran rata-rata 63,4 hektar permukaan air (156,6 hektar permukaan air). Operasi yang menggunakan ikan divaksinasi lebih besar (rata-rata ukuran = 206,6 hektar

permukaan air, atau 510,6 permukaan air). Produsen yang menebar ikan yang divaksinasi ESC untuk tumbuh di luar mewakili 19,0% dari total luas permukaan air produksi ikan; produsen yang memiliki ikan kolonisasi kolomaris mewakili 16,6% dari total luas lahan. Dari produsen yang mengisi ikan lele yang divaksinasi ESC, 41,9% menganggap bahwa kelangsungan hidup lebih baik pada ikan yang divaksinasi daripada ikan yang tidak divaksinasi; dari produsen yang menebar ikan lele koloraris, 46,2% berpikir bahwa ikan yang divaksinasi menunjukkan tingkat kelangsungan hidup yang lebih baik. Namun, 37,5% produsen yang menggunakan vaksin ESC dan 39,7% produsen yang menggunakan vaksin kolumnaris tidak tahu apakah vaksinasi memperbaiki tingkat ketahanan hidup. Ketika semua produsen ditanya tentang harapan mereka mengenai persentase ikan yang divaksinasi yang akan terlindungi dari penyakit, 52,4% menjawab bahwa mereka memperkirakan 100% ikan mereka harus dilindungi. Lebih banyak informasi produsen tentang harapan yang dapat diharapkan mengenai keefektifan vaksin, kondisi sunder yang dapat menyebabkan imunosupresi dan kegagalan vaksin, dan penilaian kinerja vaksin dapat menyebabkan peningkatan penggunaan vaksinasi sebagai alat untuk industri lele.

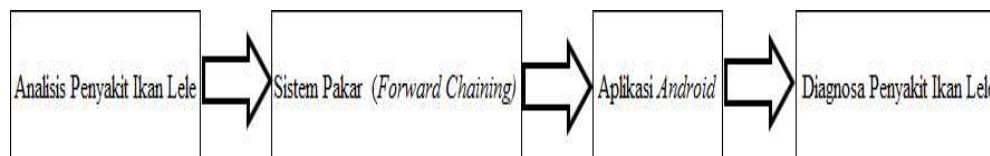
2.5 Kerangka Pemikiran

Berdasarkan kerangka pemikiran diketahui sebagai suatu gambaran menjelaskan garis besar dari kerangka pemikiran.

Menurut (Sugiyono, 2012: 60) Kerangka berfikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting.

Identifikasi masalah yaitu penyakit ikan disebabkan oleh bakteri, parasit, serta jamur yang melekat di kulit dan insang, seperti *protozoa*, *copepod*, *pseudomonas*, dan *aeromonas*. Dampaknya bisa berupa infeksi yang menyebabkan fisik terus melemah, bila tidak diatasi dapat mengakibatkan ikan mati. Penyakit sangat ditakuti pembudidaya karena menjadi faktor utama penyebab kematian pada ikan. Munculnya penyakit ikan disebabkan oleh faktor, diantaranya karena air yang kotor atau jarang diganti, pemberian pakan yang berlebihan, fasilitas budidaya yang tercemar penyakit, suhu yang ekstrim dan kesalahan penanganan.

Berdasarkan identifikasi diatas, maka kerangka pemikiran penelitian ini sebagai berikut:



Gambar 2. 7 Kerangka Pemikiran
(Sumber : Data Penelitian, 2017)

Pada gambar 2.7 diatas, penelitian meneliti mengenai penyakit ikan lele yang kemudian di implementasikan ke dalam sistem pakar dengan metode *Forward Chaining*. Hasil penelitian berupa aplikasi sistem pakar berbasis *android*

BAB III

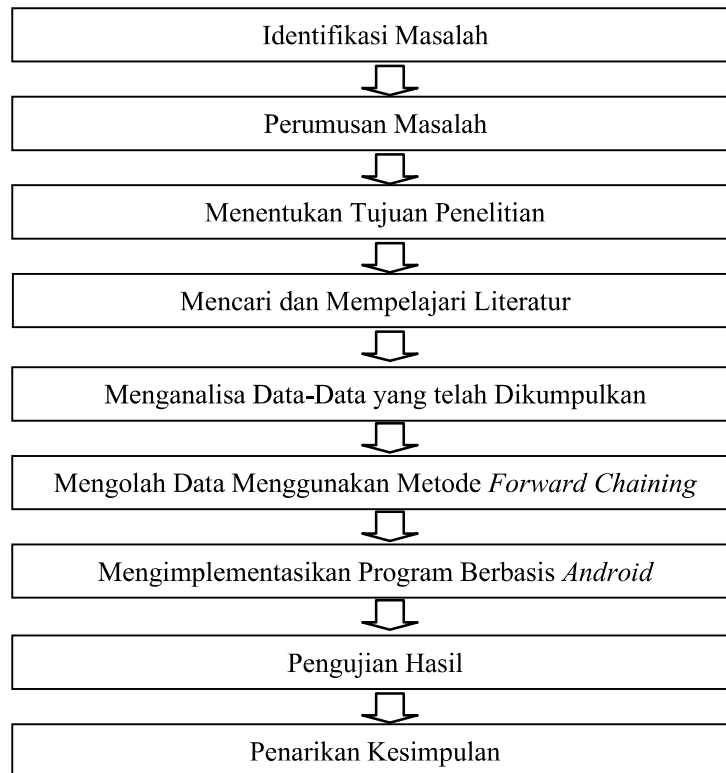
METODE PENELITIAN

Menurut (Sugiyono, 2014: 3) Metode penelitian merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Suatu penelitian mempunyai desain penelitian (*research design*) tertentu. Desain ini menggambarkan prosedur atau langkah-langkah yang harus ditempuh, waktu penelitian, sumber data dan kondisi arti apa data dikumpulkan, dan dengan cara bagaimana data tersebut dihimpun dan diolah. Tanpa desain penelitian yang benar, peneliti tidak mempunyai pedoman arah penelitian yang jelas sehingga penelitian tidak dapat dilakukan dengan baik.

3.1 Desain Penelitian

Sebelum menjalankan penelitian diperlukan suatu desain penelitian agar dalam proses pelaksanaannya dapat berjalan baik dan sistematis. Desain penelitian ini digunakan sebagai acuan dalam setiap tahapan pada proses penelitian.

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



Gambar 3. 1 Desain Penelitian
(Sumber: Data Penelitian, 2017)

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

1 Identifikasi Masalah

Penelitian diawali dengan melakukan studi pendahuluan untuk mengidentifikasi permasalahan yang berkaitan dengan topik penelitian agar peneliti mendapatkan apa yang sesungguhnya menjadi masalah untuk dipecahkan. Adapun identifikasi masalah penelitian adalah penyakit ikan disebabkan oleh bakteri, parasit, serta jamur yang melekat di kulit dan insang, seperti *protozoa*, *copepod*, *pseudomonas*, dan *aeromonas*. Dampaknya bisa berupa infeksi yang menyebabkan fisik terus melemah, bila tidak diatasi dapat

mengakibatkan ikan mati. Penyakit sangat ditakuti pembudidaya karena menjadi faktor utama penyebab kematian pada ikan. Munculnya penyakit ikan disebabkan oleh faktor, diantaranya karena air yang kotor atau jarang diganti, pemberian pakan yang berlebihan, fasilitas budidaya yang tercemar penyakit, suhu yang ekstrim dan kesalahan penanganan.

2 Perumusan Masalah

Pada tahap ini, peneliti merumuskan masalah yang telah didapatkan secara lebih spesifik agar masalah tersebut dapat dijawab dengan baik melalui penelitian. Adapun rumusan masalah yang diteliti oleh peneliti adalah bagaimana implementasi sistem pakar untuk mendiagnosa penyakit ikan lele dengan menggunakan metode *Forward Chaining* berbasis *android*.

3 Menentukan Tujuan Penelitian

Peneliti menentukan tujuan penelitian yaitu untuk mengimplementasikan sistem pakar untuk mendiagnosa penyakit ikan lele menggunakan metode *Forward Chaining* berbasis *android*.

4 Mencari dan Mempelajari Literatur

Untuk mendukung jalannya penelitian, peneliti mencari dan mempelajari sumber-sumber pengetahuan berupa buku membongkar rahasia sukses budidaya ikan lele, nila & gurami pengarang Andi Irmawan. Buku pembesaran ikan lele pengarang Budi Samadi. Buku kecerdasan buatan pengarang T. Sutojo, S. Si., M. Kom, Edy Mulyanto, S. Si., M. Kom. Buku koleksi pemograman tugas akhir dan skripsi dengan *android* pengarang Ir. Yuniar Supardi. Buku metode penelitian kuantitatif, kualitatif dan R&D pengarang Prof. Dr Sugiyono. Buku

rekayasa perangkat lunak pengarang Rosa A.S – M. Shalahuddin. Buku aplikasi sistem pakar pengarang Kusriani. Buku metodologi riset di bidang TI pengarang Dr.Sudaryono, serta melakukan wawancara langsung dengan drh. Pramudya Dwiwahyu Irawanto di Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center.

5 Menganalisa dan Mempelajari Literatur

Setelah data-data yang berkaitan dengan permasalahan penyakit ikan lele didapatkan baik melalui studi literatur maupun wawancara dengan drh. Pramudya Dwiwahyu Irawanto di Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center. Sebagai tempat pengambilan data, peneliti melakukan analisis data dengan menentukan operasional variabel dan indikator yang digunakan agar lebih mudah dilakukan proses pengolahan datanya. Peneliti membuat tabel penyakit ikan lele, tabel gejala, tabel solusi, tabel keputusan, pohon keputusan, dan tabel aturan.

6 Mengolah data sistem pakar metode *Forward Chaining*

Sistem pakar pada penelitian ini menggunakan model representasi pengetahuan berbasis kaidah produksi. Data-data yang telah dianalisis kemudian diolah menggunakan metode *forward chaining* agar dapat dibuat aturan (*rule*) yang akan digunakan saat sistem pakar melakukan penelusuran untuk menidagnosa penyakit ikan lele berdasarkan indikator yang ada.

7 Mengimplementasikan dalam bentuk program berbasis *android*

Pada tahap rancangan sistem, peneliti melakukan kegiatan perancangan mulai dari desain basis pengetahuan, desain *UML*, desain aplikasi *android*, dan desain antarmuka. Hal ini dilakukan untuk menggambarkan bagaimana struktur sistem yang akan dibuat.

8 Pengujian Hasil

Sebelum digunakan oleh pengguna akhir (*end-user*), peneliti melakukan pengujian sistem terlebih dahulu untuk memastikan bahwa sistem pakar benar-benar dapat menjalankan fungsinya sesuai dengan yang diharapkan. Pengujian dilakukan pada sistem dari segi logika dan fungsional serta memastikan bahwa semua bagian telah diuji. Proses ini bertujuan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Sistem pakar dalam penelitian ini diuji dengan menggunakan metode *black-box testing* untuk memvalidasi setiap fungsi yang terdapat dalam sistem sehingga sesuai dengan kebutuhan.

9 Menarik kesimpulan

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

3.2 Teknik Pengumpulan Data

Berdasarkan penelitian ini, peneliti menggunakan teknik pengumpulan data, yaitu:

1. Wawancara (*Interview*)

Menurut (Sugiyono, 2014: 188) Wawancara digunakan sebagai teknik pengumpulan data apabila peneliti ingin melakukan studi pendahuluan untuk menemukan permasalahan yang harus diteliti, dan juga apabila peneliti ingin mengetahui hal-hal dari responden yang lebih mendalam dan jumlah respondennya sedikit atau kecil.

Menurut (Sudaryono, 2015: 88) Wawancara atau *interview* adalah suatu cara pengumpulan data yang digunakan untuk memperoleh informasi langsung dari sumbernya. Wawancara digunakan bila kita ingin mengetahui hal-hal dari responden yang jumlahnya sedikit, secara lebih mendalam.

Adapun pada penelitian ini peneliti melakukan wawancara di Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center dengan drh. Pramudya Dwiwahyu Irawanto sebagai pakar. Wawancara dilakukan secara terbuka, beliau menjelaskan apa saja yang menyebabkan terjadinya penyakit ikan lele dan solusi untuk mengobati penyakit ikan lele tersebut.

2. Dokumentasi

Menurut (Sudaryono, 2015: 92) Dokumentasi ditujukan untuk memperoleh data langsung dari tempat penelitian, meliputi buku peraturan, laporan kegiatan, foto, film dokumenter, dan data yang relevan dengan

penelitian. Dokumen merupakan catatan peristiwa yang sudah berlalu. Dokumen bisa berbentuk tulisan, gambar, atau monumental seseorang. Dokumen yang berbentuk tulisan, misalnya adalah catatan harian, sejarah kehidupan, cerita, biografi, peraturan, dan kebijakan. Dokumen yang berbentuk gambar misalnya foto, gambar hidup, sketsa, dan sebagainya. Dokumen yang berbentuk karya misalnya gambar, patung, film, dan sebagainya.

Peneliti melakukan dokumentasi dengan mengumpulkan, membaca, dan memahami referensi teoritis yang berasal dari buku-buku teori, buku elektronik (*e-book*), jurnal-jurnal penelitian, dan sumber pustaka otentik lainnya yang berkaitan dengan penelitian.

Buku yang dimaksud adalah buku mengenai membongkar rahasia sukses budidaya ikan lele nila & gurami, buku pembesaran ikan lele, buku kecerdasan buatan, buku sistem pakar, buku koleksi pemograman tugas akhir dan skripsi dengan *android*, buku metode penelitian kuantitatif, kualitatif dan R&D, buku rekayasa perangkat lunak, buku aplikasi sistem pakar, buku metodologi riset di bidang TI.

3.3 Operasional Variabel

Sebelum mengadakan penilaian dalam penelitian, peneliti harus menentukan operasional variabel, hal ini dimaksudkan agar dapat mempermudah dalam melakukan penelitian.

Menurut (Sugiyono, 2014: 63) Variabel penelitian adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya.

Adapun manfaat operasionalisasi variabel antara lain untuk mengidentifikasi kriteria yang dapat diobservasi dan yang sedang didefinisikan, menunjukkan bahwa suatu konsep atau objek mungkin mempunyai lebih dari satu definisi operasional, dan untuk mengetahui bahwa definisi operasional bersifat unik dalam situasi dimana definisi tersebut harus digunakan. Variabel yang digunakan dalam penelitian ini adalah penyakit ikan lele.

Berdasarkan hasil wawancara di Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center dengan drh. Pramudya Dwiwahyu Irawanto sebagai pakar, diperoleh gejala-gejala penyakit ikan lele *Trichodina sp* sejenis penyakit parasit yang dapat menyebabkan gatal (*Trichodiniasis*), gejala ikan ini biasanya nafsu makan berkurang, bintik-bintik putih bagian kepala serta punggung, ikan menjadi sangat lemah dan kurus, pada warna tubuh menjadi kusam, sering terlihat ikan menggosokkan tubuhnya pada dasar atau dinding kolam/tambak serta benda-benda keras di sekitarnya. *Dactylogyrus sp* dan *gyrodactylus sp* jenis penyakit parasit. Gejala-gejala dari penyakit ini yaitu nafsu makan berkurang, ikan kelihatan lemah dan kurus, kulit ikan kusam, kulit berlendir, menggosokkan badan pada dinding kolam atau benda-benda keras, sirip rontok. *Ichthyophthirius Multifiliis* jenis penyakit parasit, memiliki gejala tubuh berubah menjadi pucat dan mengelupas, perubahan fisik seperti timbul bintik-bintik putih pada tubuh, sirip tampak robek-robek dan insang

memucat, terjadinya kerusakan pada kulit dan insang memicu ikan mengalami stress, ikan sering mengosokkan tubuh pada dasar dan dinding kolam.

Sedangkan dari studi pustaka Menurut (Samadi, 2016: 94-95) *Dactylogyrus sp* dan *gyrodactylus sp* merupakan parasit. Ia adalah cacing yang tergolong kelas *trematoda*, termasuk cacing monogenea. Tubuhnya berbentuk pipih dan pada bagian ujung tubuhnya terdapat alat yang berfungsi sebagai alat penghisap darah mangsanya. Gejala penyakit ini ditandai dengan kondisi ikan kurus, kulit berlendir dan tubuh ikan tampak tidak cerah, sirip rontok, tutup insang agak terbuka, lemah, nafsu makan menurun, megap-megap seperti kekurangan oksigen. Ikan yang terserang penyakit ini juga sering menggosok-gosokkan tubuhnya pada dinding kolam atau pada benda-benda keras lainnya. Pengobatan gangguan penyakit *dactylogyriasis* dan *gyrodactyliasis* dapat diobati dengan *Methylene blue* dengan dosis 1 g/m³ air, lalu ikan yang sakit direndam dalam larutan tersebut selama 30 menit. Selain itu, dapat juga menggunakan garam dapur dengan dosis 13 g/m³ air, *formalin* dengan dosis 40 ppm. Ikan sakit direndam dalam larutan tersebut selama satu hari satu malam (24 jam). Menurut (Samadi, 2016: 91-92) *Ichthyophthirius Multifiliis* merupakan jenis parasit. Ia tergolong hewan bersel tunggal (*protozoa*). Infeksi parasit tersebut menimbulkan penyakit bercak putih. Nama lain penyakit bercak putih yaitu “*Ich*” atau “*Ichthyophthiriasis*” atau lebih populer dengan sebutan *white spot*. Gejala ikan tampak malas berenang, pucat, sering menggosok-gosokkan badannya pada dinding kolam atau dasar kolam, ada bercak-bercak putih pada tubuh, megap-megap, dan ikan juga sering berada di dekat pintu pemasukan air, serta tampak mengapung di permukaan air. Bagian tubuh terinfeksi menjadi

pucat dan bengkak. Pengobatan pengobatan penyakit yang disebabkan oleh parasit. *Ichthyophthirius Multifiliis* dapat menggunakan garam (NaCl) dengan dosis 1-3 g/liter, lalu ikan yang sakit direndam dalam larutan garam dapur tersebut selama 5-10 menit. Selain itu, dapat juga menggunakan *Methilene blue* dengan dosis 1 g/liter air. Lalu ikan yaang sakit direndam ke dalam lautan *Methilene blue* dari larutan baku tersebut dengan dosis 0,5-1 cc/liter air selama 24 jam. Pengobatan diulang hingga 3 kali atau lebih dengan selang waktu satu hari. Menurut (Samadi, 2016:96-97) *Trichodina sp* dan *trichodinella sp* merupakan parasit. Ia tergolong hewan bersel tunggal (*protozoa*) berbulu getar, kedua parasit tersebut berukuran mikroskopis dan hanya dapat dilihat dengan menggunakan alat bantu mikroskop. Parasit ini berbentuk lingkaran transparan dengan sejumlah bulu-bulu getar (*silia*) yang menempel di sekeiling lingkaran itu. Pada tubuh bagian bawah terdapat tubuh bagian perekat yang berfungsi untuk melekatkan dirinya ke tubuh mangsanya. Parasit ini menginfeksi kulit, sirip, dan insang. Ia menyerang ikan lele segala ukuran, baik yang masih kecil maupun yang sudah besar. Infeksi parasit ini mengakibatkan kerusakan pada insang dan tubuh ikan, dan penyakitnya dikenal sebutan penyakit “*Trichodiniasis*”. Gejala ikan berbintik-bintik dibagian kepala dan punggung, kurus dan lemah, nafsu makan berkurang, tubuh kusam. Ikan yang terinfeksi ini juga tampak sering menggosok-gosokan tubuhnya ke dinding kolam, dasar kolam, atau benda keras. Pengobatan penyakit *trichodiniasis* diobati dengan menggunakan formalin dengan dosis 40 ppm, yang kemudian ikan-ikan yang sakit direndam dalam larutan tersebut selama satu hari satu malam (24 jam). Perlakuan ini dilakukan selama 4-7 hari.

Tabel 3. 1 Tabel Variabel dan indikator

Variabel	Indikator
Penyakit Ikan Lele	<i>Trichodina sp</i>
	<i>Dactylogyrus sp</i> dan <i>Gyrodactylus sp</i>
	<i>Ichthyophthirius Multifiliis</i>

(Sumber: Data Penelitian, 2017)

Indikator dalam penelitian ini ada 3 indikator yaitu penyakit *Trichodina sp*, *Dactylogyrus sp* dan *Gyrodactylus sp* dan *Ichthyophthirius Multifiliis*, seperti yang terlihat pada tabel 3.1 Tabel Indikator.

3.4 Perancangan Sistem

Menurut (Sudaryono, 2015: 230) Perancangan sistem bisa digambarkan dalam suatu bagian aliran yang menjelaskan keseluruhan proses yang kita lakukan.

3.4.1 Perancangan Basis Pengetahuan

Sebelum melakukan desain basis pengetahuan, peneliti telah melakukan proses akuisisi pengetahuan dengan mengumpulkan pengetahuan dan fakta dari sumber-sumber yang tersedia. Sumber pengetahuan dan fakta diperoleh melalui wawancara di Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center dengan drh. Pramudya Dwiwahyu Irawanto sebagai pakar dan dokumentasi tentang materi yang berkaitan dengan hal tersebut.

Sumber pengetahuan dan fakta yang didapat berupa data-data yang berhubungan dengan penyakit ikan lele, indikator, dan solusi. Pengetahuan dan fakta tersebut ditampilkan dalam tabel data indikator (Tabel 3.2), tabel indikator penyakit ikan lele (Tabel 3.3), tabel gejala penyakit ikan lele (Tabel 3.4) dan tabel aturan (Tabel 3.5).

Tabel 3. 2 Tabel data indikator

Kode	Nama Indikator
P01	<i>Trichodina sp</i>
P02	<i>Dactylogyrus sp</i> dan <i>Gyrodactylus sp</i>
P03	<i>Ichthyophthirius Multifiliis</i>

(Sumber: Data Penelitian, 2017)

Tabel 3. 3 Tabel indikator penyakit ikan lele

Nama Penyakit	Keterangan	Solusi
<i>Trichodina sp</i>	Sejenis penyakit parasit yang dapat menyebabkan gatal (<i>Trichodiniasis</i>), gejala ikan ini biasanya nafsu makan berkurang, bintik-bintik putih bagian kepala serta punggung, ikan menjadi sangat lemah dan kurus, pada warna tubuh menjadi kusam, sering terlihat ikan menggosokkan tubuhnya pada dasar atau dinding kolam/tambak serta benda-benda keras di sekitarnya.	Rendam dalam larutan formalin 150-200 cc/m ³ . Rendam dalam larutan <i>malachyte green oxalate</i> 0,1 g/ m ³ selama 12-24 jam.

<i>Dactylogyrus sp</i> dan <i>Gyrodactylus sp</i>	<i>Dactylogyrus sp</i> dan <i>gyrodactylus sp</i> jenis penyakit parasit, dan biasanya sering disebut cacing, bentuknya pipih. Gejala-gejala dari penyakit ini yaitu nafsu makan berkurang, ikan kelihatan lemah dan kurus, kulit ikan kusam, kulit berlendir, menggosokkan badan pada dinding kolam atau benda-benda keras, sirip rontok	Merendam dalam larutan formalin 250 cc/m ³ air selama 15 menit. Lalu rendam dalam larutan <i>methyline blue</i> 3 g/ m ³ air selama 24 jam.
<i>Ichthyophthirius Multifiliis</i>	<i>Ichthyophthirius Multifiliis</i> jenis penyakit parasit, memiliki gejala tubuh berubah menjadi pucat dan mengelupas, perubahan fisik seperti timbul bintik-bintik putih pada tubuh, sirip tampak robek-robek dan insang memucat, terjadinya kerusakan pada kulit dan insang memicu ikan mengalami stress, ikan sering mengosokkan tubuh pada dasar dan dinding kolam.	Siapkan wadah berupa bak untuk mengobati ikan yang sakit. Buat larutan baku 1 gram <i>methylene blue</i> ke dalam 100 ml air bersih. Tetaskan larutan ke dalam bak sebanyak 2-4 ml untuk setiap 4 liter air. Rendam ikan yang sakit selama 24 jam dan ulangi pengobatan sebanyak 3-5 kali dalam selang waktu satu hari.

(Sumber: Data Penelitian, 2017)

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

Tabel 3. 4 Tabel gejala penyakit ikan lele

Kode Gejala	Nama Gejala
G01	Ikan berbintik-bintik dibagian kepala dan punggung
G02	Kurus dan lemah
G03	Nafsu makan berkurang
G04	Tubuh kusam
G05	Sering menggosok-gosokkan tubuhnya ke dinding kolam, dasar kolam, atau benda-benda keras.
G06	Ikan terlihat kurus
G07	Kulit berlendir dan tubuh ikan tidak cerah
G08	Sirip rontok
G09	Tutup insang agak terbuka
G10	Ikan tampak lemah

Tabel 3. 84

G11	Megap-megap seperti kekurangan oksigen
G12	Ikan tampak malas berenang
G13	Ikan tampak pucat
G14	Ada bercak-bercak putih pada tubuh
G15	Ikan sering berada di dekat pintu pemasukan air
G16	Ikan mengapung pada permukaan air

(Sumber : Data Penelitian, 2017)

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

Tabel 3. 5 Tabel aturan

Kode Gejala	Kode Indikator
G01, G02, G03, G04, G05	P01
G06, G07, G08, G09, G10, G11	P02
G12, G13, G14, G15, G16	P03

(Sumber : Data Penelitian, 2017)

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

1. Kaidah 1: *IF G01 AND G02 AND G03 AND G04 AND G05 THEN P01*

2. Kaidah 2: *IF G06 AND G07 AND G08 AND G09 AND G10 AND G011 THEN P02*

3. Kaidah : *IF G12 AND G13 AND G14 AND G15 AND G16 THEN P03*

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

1. Jika Ikan berbintik-bintik dibagian kepala dan punggung, kurus dan lemah, nafasu makan berkurang, tubuh kusam dan sering menggosok-gosokan tubuhnya ke dinding kolam, dasar kolam, atau benda-benda keras maka jenis penyakit ikan lele adalah *Trichodina sp.*
2. Jika ikan lele terlihat kurus, kulit berlendir dan tubuh ikan tidak cerah, sirip rontok, tutup insang agak terbuka, ikan tampak lemah dan megap-megap seperti kekurangan oksigen maka jenis penyakit ikan lele adalah *Dactylogyrus sp* dan *Gyrodactylus sp.*
3. Jika ikan tampak malas berenang, ikan tampak pucat, ada bercak-bercak putih pada tubuh, ikan sering berada di dekat pintu pemasukan air dan ikan mengapung di atas permukaan air maka jenis penyakit pada ikan lele adalah *Ichthyophthirius Multifiliis.*

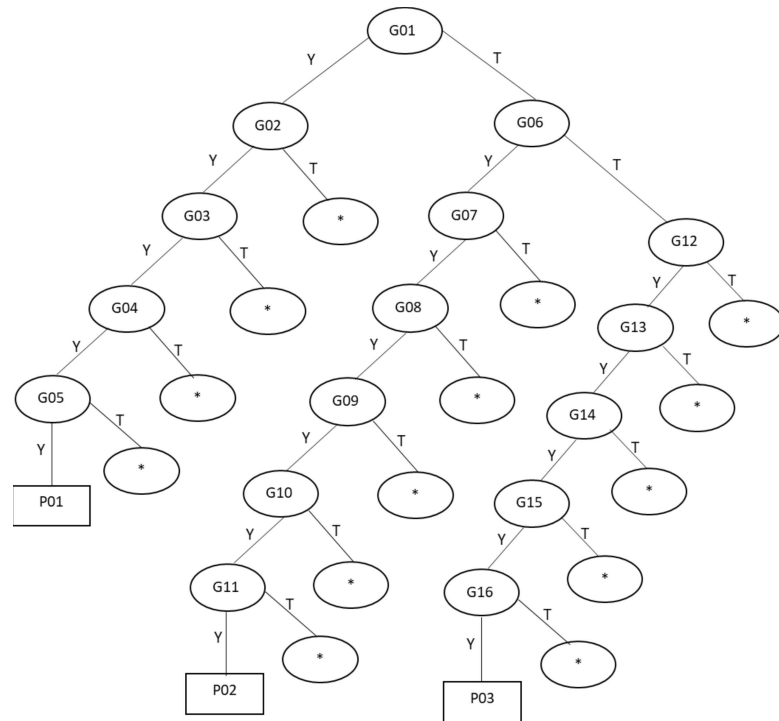
Berdasarkan kaidah yang telah dibuat tersebut maka tabel keputusannya adalah sebagai berikut:

Tabel 3. 6 Tabel keputusan

Gejala \ Penyakit	P01	P02	P03
G01	√		
G02	√		
G03	√		
G04	√		
G05	√		
G06		√	
G07		√	
G08		√	
G09		√	
G10		√	
G11		√	
G12			√
G13			√
G14			√
G15			√
G16			√

(Sumber: Data Penelitian (2017))

Berdasarkan tabel keputusan tersebut maka pohon keputusannya adalah sebagai berikut:

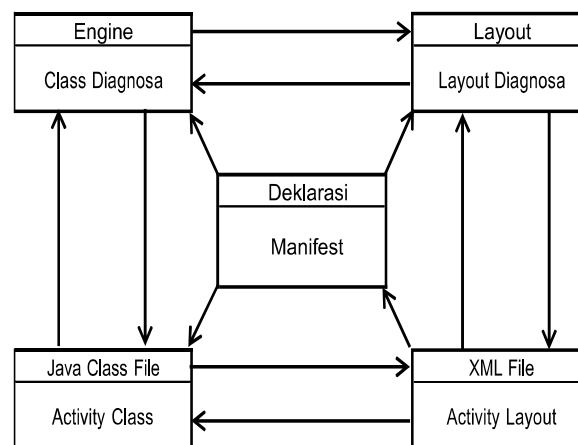


Gambar 3. 2 Pohon Keputusan
(Sumber : Data Penelitian, 2017)

Data gejala ditentukan sebagai keadaan awal dalam sistem saat melakukan penelusuran sebelum diperoleh sebuah kesimpulan. Pohon keputusan pada gambar 3.2 digunakan untuk memperlihatkan hubungan terkait antar gejala yang ada. Arah penelusuran pada pohon keputusan tersebut dimulai dari simpul akar (yang paling atas) ke bawah. Alur keputusan sistem pakar dimulai dari G01 yaitu ikan tampak sering menggosok-gosokkan tubuhnya ke dinding kolam, dasar kolam dan benda-benda keras. Gejala ini dipilih sebagai keadaan awal dalam penelusuran karena gejala ini adalah gejala yang paling mudah terlihat dan diketahui.

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

Dalam penelitian ini, peneliti membuat desain *knowledge base* seperti berikut:



Gambar 3. 3 Desain *Knowledge Base*
(Sumber : Data Penelitian, 2017)

Tabel yang digunakan dalam sistem pakar ini terdiri dari 5 tabel, yaitu tabel *Class Diagnosa* sebagai *Engine*, *Layout Diagnosa* sebagai tampilan, *Manifest*

sebagai Deklarasi, *Activity Class* sebagai *Java Class File* dan *Activity Layout* sebagai *XML File* sebagai *Layout* dihubungkan ke *Main Activity*. Lalu *Main Activity* dihubungkan kembali ke *Java Class* lainnya disertai dengan *XML File*. Semua tabel saling berhubungan satu dengan yang lainnya untuk menghasilkan aplikasi sistem pakar diagnosa penyakit ikan lele.

3.4.2 Struktur kontrol (mesin inferensi)

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

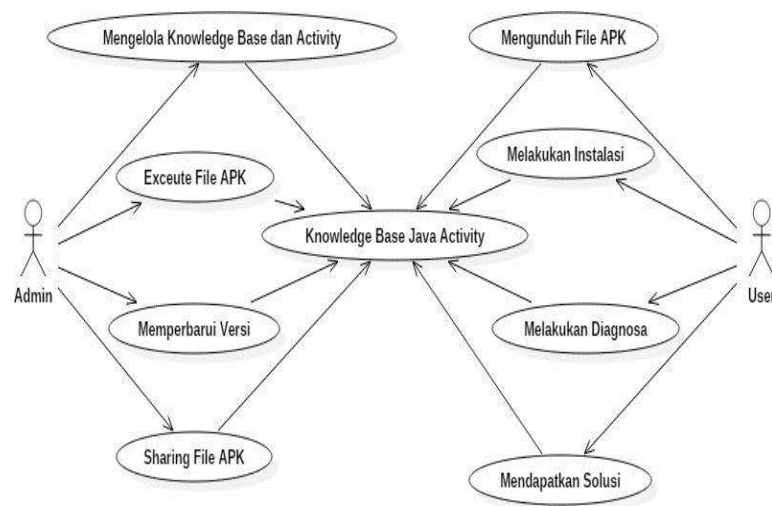
1. Mengajukan pertanyaan tentang gejala penyakit kepada pengguna.
2. Jika jawaban pengguna “Ya” maka sistem akan melakukan langkah 3. Jika jawaban pengguna “Tidak” maka sistem akan melakukan langkah 4.
3. Menyimpan gejala dalam memori kerja lalu memeriksa relasi gejala dengan penyakit yang telah dibuat.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

3.4.3 Desain UML (*Unifed Modeling Language*)

Desain sistem pada penelitian ini menggunakan bahasa pemodelan *Unifed Modelling Language (UML)* yang digambarkan dengan bantuan aplikasi *StarUML*. Diagram *UML* dalam penelitian ini, yaitu:

1. *Use case diagram*

Aktor yang digunakan dalam sistem pakar ini terdiri dari 2 orang yaitu administrator dan pengguna (*user*). *Use case* yang terdapat dalam sistem antara lain *Log In*, mengelola daftar pengguna, mengelola daftar administrator, mengelola data gejala, mengelola data aturan dan diagnosa. *Use case diagram* yang dirancang untuk sistem pakar dalam penelitian ini dapat dilihat pada gambar di bawah ini.



Gambar 3. 4 *Use case diagram*
(Sumber : Data Penelitian, 2017)

Pada Gambar 3.4 diatas, terdapat 2 aktor yaitu *admin* dan *user*. *Admin* melakukan interaksi dengan sistem berupa mengelola *knowledge base* dan

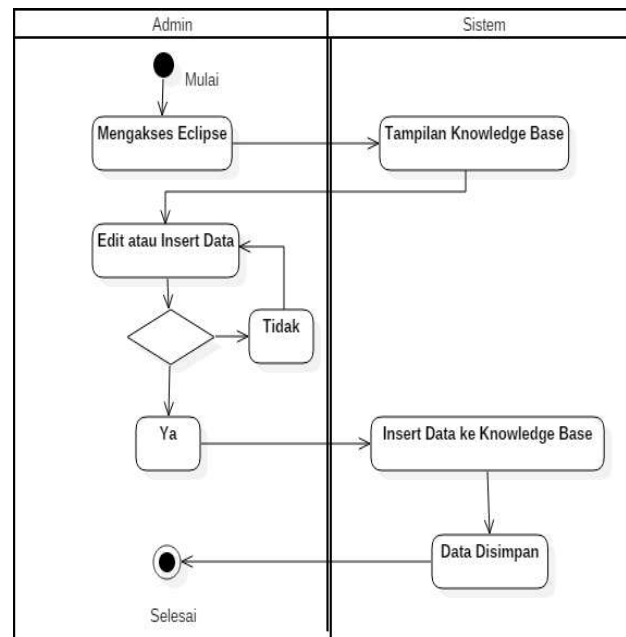
activity, *execute file APK*, *memperbaruiversi* dan *sharing file APK*. Sedangkan *user* berinteraksi dengan sistem yaitu mengunduh file *APK*, melakukan *instalasi*, melakukan diagnosa dan mendapatkan solusi. Diagnosa dilakukan dengan cara menjawab pertanyaan yang diajukan oleh sistem, setelah semua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

2. Activity Diagram

Menurut (A.S & Shalahuddin, 2011: 134) Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan sistem. Adapun *Activity diagram* dalam penelitian ini adalah:

a. Admin

1. *Activity Diagram Mengelola Knowledge Base* merupakan *UML* yang menggambarkan kegiatan admin dalam mengelola *data base*. Berikut ini gambar *activity diagram* mengelola *data base*:

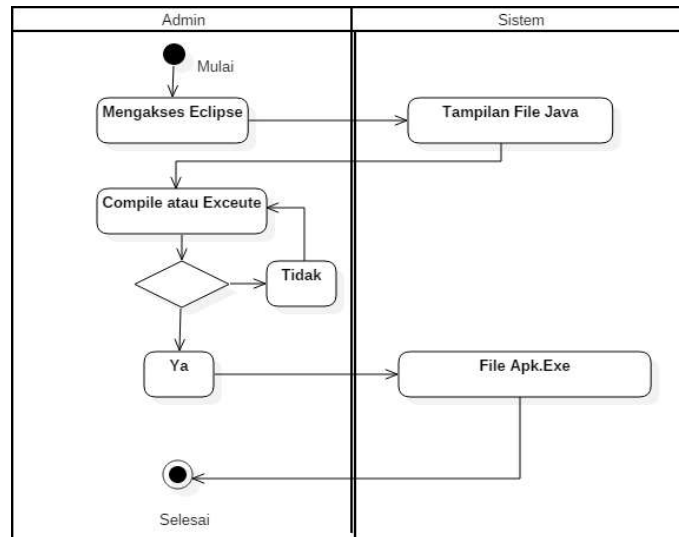


Gambar 3. 5 *Activity diagram* mengelola *data base*
(Sumber : Data Penelitian, 2017)

Pada Gambar 3.5 di atas, admin akan mulai mengakses *eclipse* dan sistem akan menampilkan *data base*. Admin akan melakukan *edit* atau *insert* data, jika ya maka sistem akan menampilkan *insert data ke data base* lalu data disimpan dan admin selesai mengelola *data base*.

2. *Activity diagram* execute file *Apk*

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

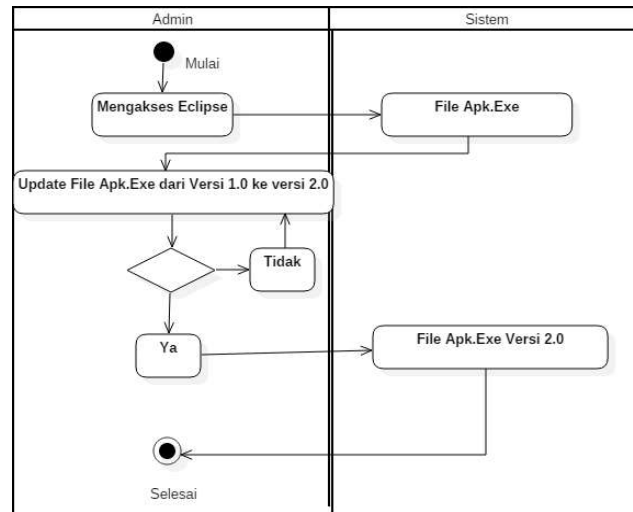


Gambar 3. 6 *Activity diagram execute file apk*
(Sumber : Data Penelitian, 2017)

Pada Gambar 3.6 di atas admin mulai mengakses *eclipse* dan sistem menampilkan *file java*. Admin akan melakukan *compile* atau *execute* dan jika ya maka sistem akan membuat file *Apk.exe*. Admin selesai melakukan *execute file apk*.

3. *Activity diagram* memperbarui versi

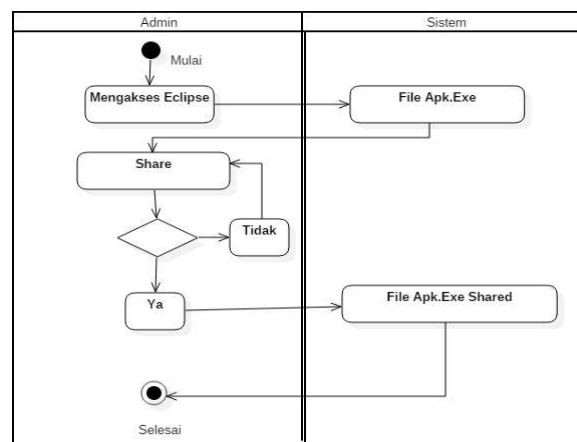
Activity diagram memperbarui versi merupakan UML yang menggambarkan kegiatan admin dalam memperbarui versi. Berikut ini gambar *activity diagram* memperbarui versi.



Gambar 3.7 Activity diagram memperbarui versi
(Sumber : Data Penelitian, 2017)

4. Activity diagram sharing file apk

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



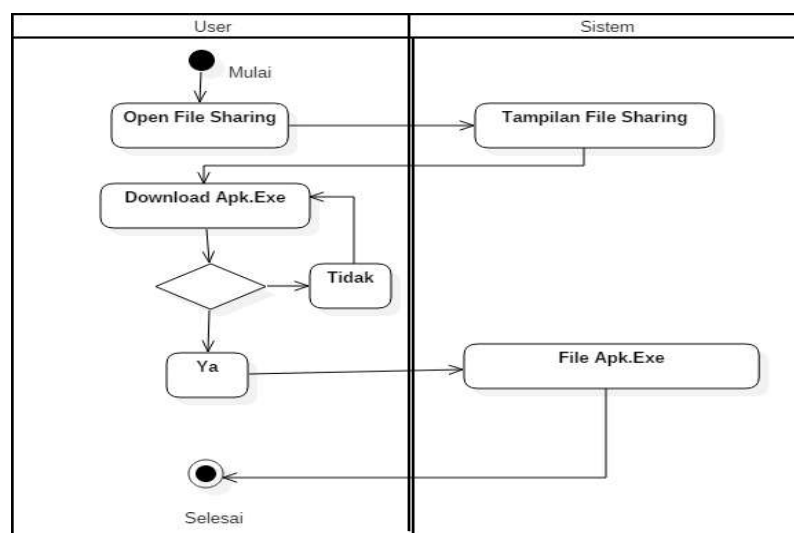
Gambar 3.8 Activity diagram sharing file apk
(Sumber : Data Penelitian, 2017)

Pada Gambar 3.8 di atas, admin akan mulai mengakses *eclipse* dan sistem akan menampilkan *file apk.exe*. Admin akan share, jika ya maka sistem akan menampilkan *file spk.exe. shared*. Admin selesai mengelola *data base*.

b. *User*

1. *Activity diagram* mengunduh *file apk*

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

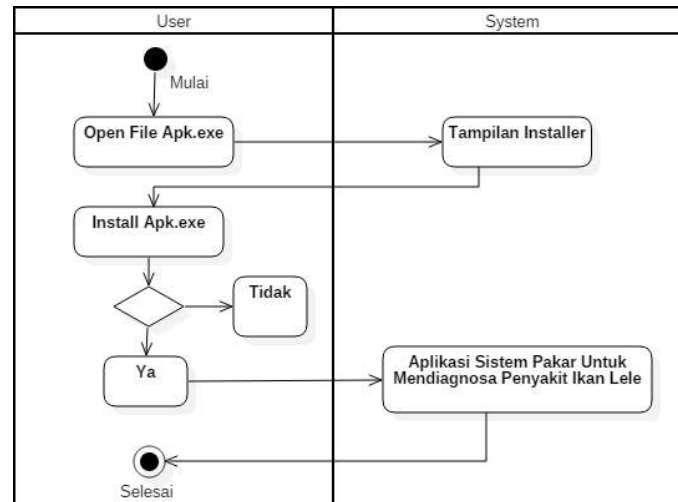


Gambar 3.9 *Activity diagram* mengunduh *file apk*
(Sumber : Data Penelitian, 2017)

Pada Gambar 3.9 di atas, *user* akan mulai *open file* dan sistem akan menampilkan *file sharing*. *User* akan melakukan *download Apk.exe*, jika ya maka sistem akan menampilkan *file Apk.exe* dan *user* selesai mengunduh *file apk*.

2. *Activity diagram* melakukan instalasi

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

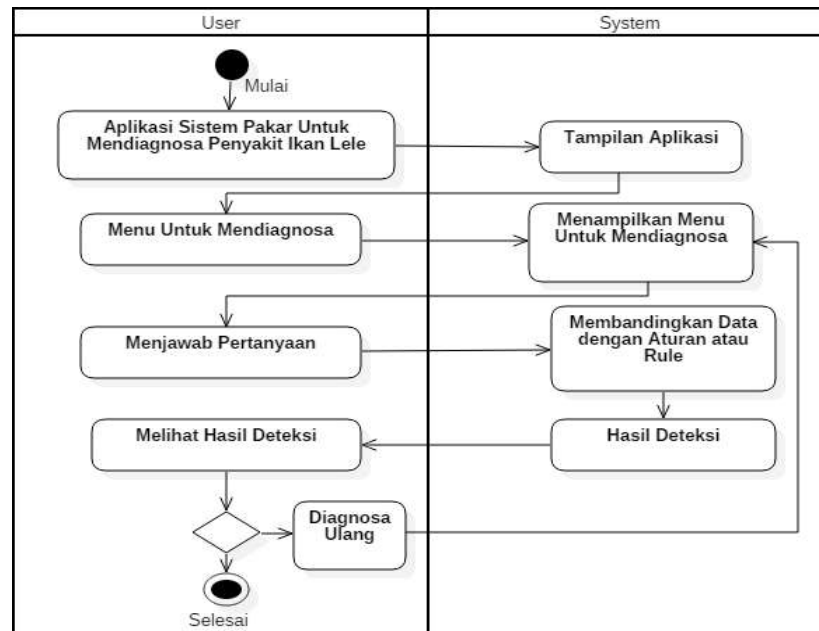


Gambar 3. 10 *Activity Diagram* melakukan instalasi
(Sumber : Data Penelitian, 2017)

Pada Gambar 3.10 di atas, *user* akan mulai menginstall aplikasi dan sistem akan menampilkan tampilan *installer*. *User* akan memilih untuk melanjutkan menginstall atau tidak, jika ya maka sistem akan terinstall dan *user* selesai melakukan instalasi.

3. *Activity diagram* melakukan diagnosa

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



Gambar 3. 11 *Activity diagram* melakukan diagnosa
(Sumber : Data Penelitian, 2017)

Pada Gambar 3.11 di atas, *user* akan mulai membuka aplikasi dan sistem akan menampilkan tampilan aplikasi. Setelah itu *User* akan membuka menu deteksi dan sistem akan menampilkan menu deteksi. Pada menu deteksi, *User* akan menjawab pertanyaan yang berupa gejala-gejala penyakit ikan lele yang telah di *input* ke dalam *knowledge base*, jika *user* menjawab “ya” maka sistem akan membandingkan data dari *user* dengan *rule* atau aturan yang telah disimpan di dalam *knowledge base*. *User* selesai menjawab pertanyaan, sistem menampilkan hasil diagnosa dan *user* selesai melakukan deteksi/ diagnosa.

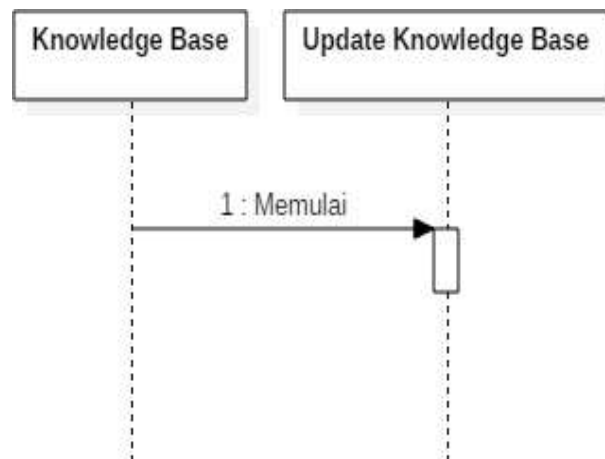
3. *Sequence diagram*

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

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

a. Admin

1. *Sequence diagram* mengelola *knowledge base*

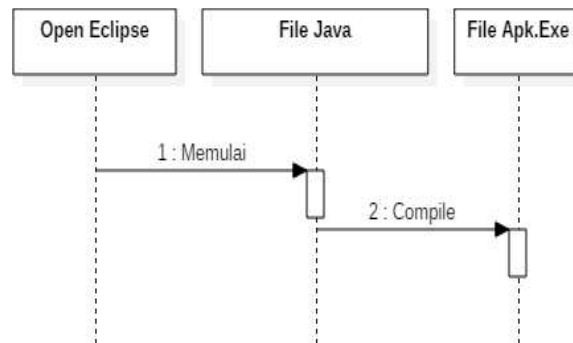


Gambar 3. 12 *Sequence diagram* mengelola *knowledge base*
(Sumber : Data Penelitian, 2017)

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

2. *Sequence diagram* execute file *Apk*

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

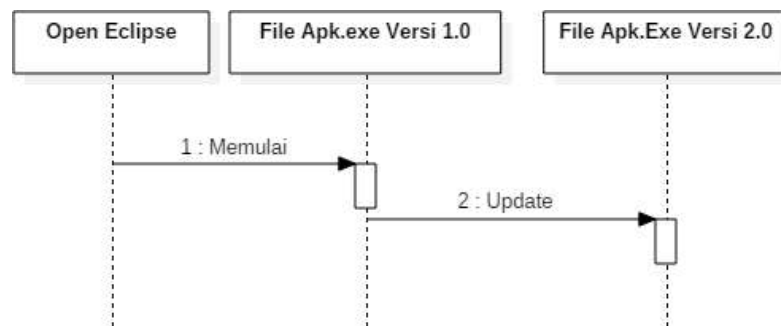


Gambar 3. 13 *Sequence diagram execute file Apk.exe*
(Sumber : Data Penelitian, 2017)

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

3. *Sequence diagram* memperbarui versi

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

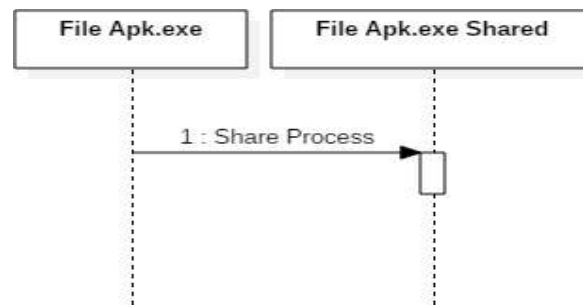


Gambar 3. 14 *Sequence diagram* memperbarui versi
(Sumber : Data Penelitian, 2017)

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

4. Sequence diagram sharing file Apk

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



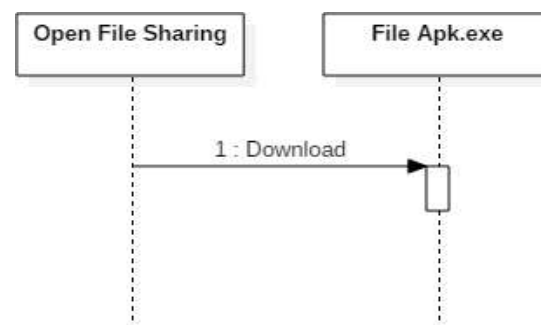
Gambar 3. 15 Sequence diagram sharing file apk
(Sumber : Data Penelitian, 2017)

Admin akan membuka *file Apk.exe* kemudian admin akan *sharefile Apk.exe shared*.

b. User

1. Sequence diagram mengunduh

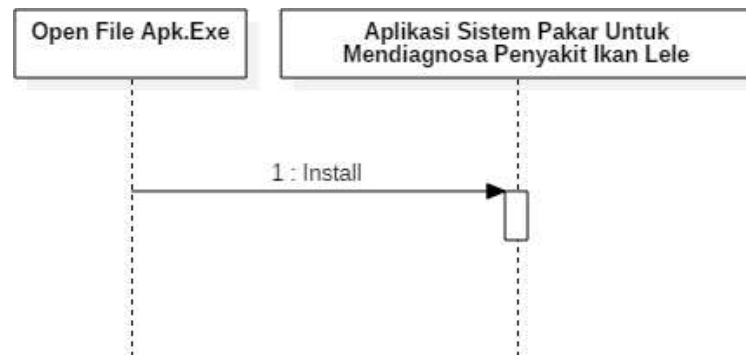
file apk merupakan urutan kegiatan *user* saat mengunduh *file Apk*. Berikut ini gambar sequence diagram mengunduh *file Apk*:



Gambar 3. 16 Sequence diagram mengunduh file Apk
(Sumber : Data Penelitian, 2017)

2. *Sequence diagram* melakukan instalasi

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

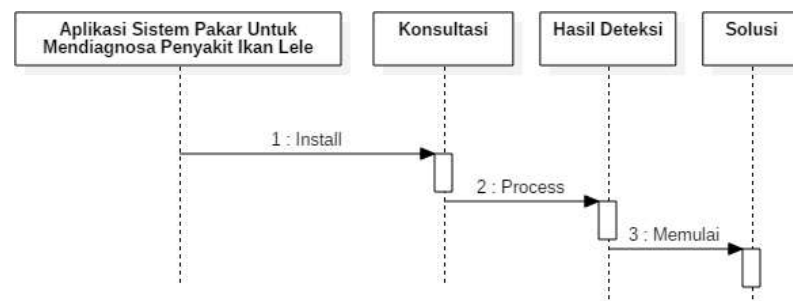


Gambar 3. 17 *Sequence diagram* melakukan instalasi
(Sumber : Data Penelitian, 2017)

User akan membuka *file Apk.exe* kemudian *user* menginstal aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Ikan Lele.

3. *Sequence diagram* melakukan diagnosa

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



Gambar 3. 18 *Sequence diagram* melakukan diagnosa
(Sumber : Data Penelitian, 2017)

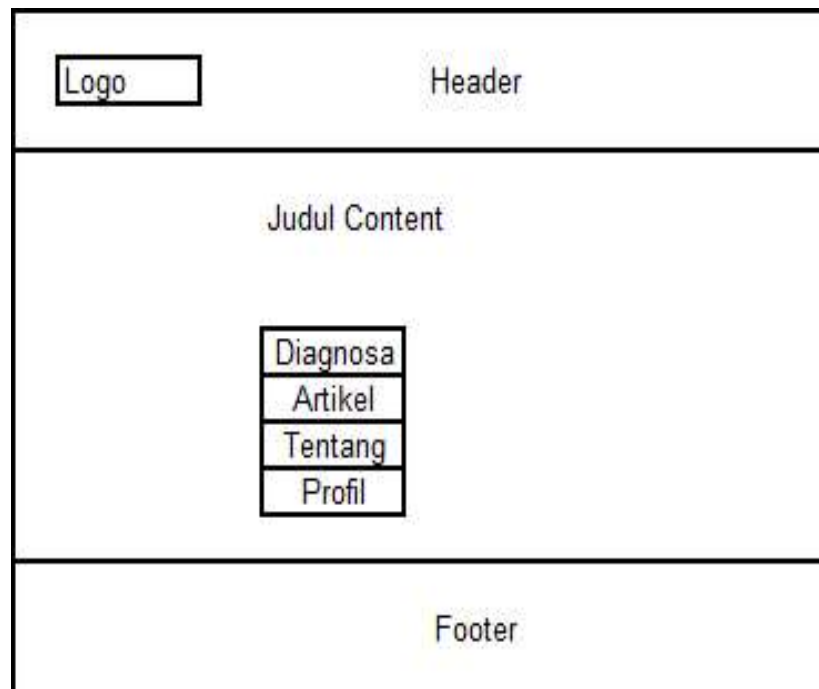
User akan membuka Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Pada Ikan Lele dan memulai konsultasi, setelah menjawab pertanyaan yang ada akan diproses dan sistem akan menampilkan hasil diagnosa dan solusi.

3.4.4 Desain antarmuka

Adapun desain tampilan sistem pakar mendiagnosa penyakit ikan lele adalah sebagai berikut:

1. Rancangan *Form* Beranda

Form beranda memiliki beberapa tampilan yaitu: *header*, *logo*, judul, *contenc*, menu utama dan *text area*.



Gambar 3. 19 Rancangan *Form* Beranda
(Sumber : Data Penelitian, 2017)

2. Rancangan *Form* Diagnosa

Form deteksi digunakan oleh pengguna untuk berkonsultasi dengan sistem pakar. Sistem akan mengajukan beberapa pertanyaan tentang gejala-gejala penyakit ikan lele yang tampak.

The diagram shows a rectangular form divided into three horizontal sections. The top section is labeled 'Header' and contains a 'Logo' box on the left. The middle section is titled 'Pertanyaan mengenai Gejala Penyakit Ikan Lele' and contains two buttons labeled 'Ya' and 'Tidak'. The bottom section is labeled 'Footer'.

Gambar 3. 20 Rancangan *Form* Hasil Diagnosa
(Sumber : Data Penelitian, 2017)

3. Rancangan *form* Hasil Diagnosa

Form hasil deteksi digunakan untuk menampilkan hasil diagnosa yang berisi jenis penyakit ikan lele dan solusi yang diberikan oleh sistem pakar.

Logo	Header
Jenis Penyakit Ikan Lele Gambar Penyakit Ikan Lele Solusi: <input type="button" value="Diagnosa Ulang"/> <input type="button" value="Menu Utama"/>	
Footer	

Gambar 3. 21 Rancangan *Form* Hasil Diagnosa
 (Sumber : Data Penelitian, 2017)

4. Rancangan *Form* Tentang

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

Logo	Header
Versi Sistem Pakar Keterangan:	
Footer	

Gambar 3. 22 Rancangan *Form* Tentang
 (Sumber : Data Penelitian, 2017)

5. Rancangan *Form* Artikel

Form artikel berisi tentang jenis-jenis penyakit ikan lele, penyebab penyakit ikan lele dan solusi penyakit ikan lele.

The diagram illustrates the layout of an article form, organized into three horizontal sections:

- Header:** Contains a "Logo" box on the left and the text "Header" on the right.
- Content Area:** Contains the text "Judul Artikel:" followed by a large empty space for the title, and "Content Artikel:" followed by a large empty space for the article content. A "Menu Utama" box is positioned in the lower right of this section.
- Footer:** Contains the text "Footer" centered at the bottom.

Gambar 3. 23 Rancangan *Form* Artikel
(Sumber : Data Penelitian, 2017)

6. Rancangan *Form* Profil

Form profil merupakan *form* yang berisi data mengenai pembuat dari sistem pakar untuk mendeteksi penyakit ikan lele ini.

The diagram illustrates a profile form layout. It is a rectangular box divided into three horizontal sections. The top section is labeled 'Header' and contains a 'Logo' box on the left. The middle section contains three labels on the left: 'Nama:', 'NPM:', and 'Prodi:'. To the right of these labels is a 'Photo' box. The bottom section is labeled 'Footer'.

Gambar 3. 24 Rancangan *Form Profil*
(Sumber : Data Penelitian, 2017)

3.5 Lokasi dan Jadwal Penelitian

3.5.1 Lokasi

Penelitian ini dilakukan melalui wawancara dengan drh. Pramudya Dwiwahyu Irawanto yang bertugas di Stasiun Karantina Ikan, Pengendalian Mutu dan Keamanan Hasil Perikanan Kelas I Batam, Jl. M. Nahar Batam Center .Alasan memilih Beliau sebagai narasumber ahli karena beliau sebagai pakar yang mengetahui dalam permasalahan mengenai penyakit ikan lele.

3.5.2 Jadwal Penelitian

Tabel 3. 7 Jadwal Penelitian

No	Kegiatan	Waktu Penelitian (2017-2018)																
		Sep-17				Okt-17				Nop-17				Des-17				Jan-18
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
1	Pengajuan judul	■																
2	Pengajuan surat penelitian		■	■														
3	Pengambilan surat penelitian				■													
4	Wawancara dengan narasumber				■	■												
5	Penyusunan bab 1 "Pendahuluan"																	
6	Penyusunan bab 2 "Landasan teori"																	
7	Penyusunan bab 3 "Metode pene																	
8	Penyusunan bab 4																	
9	Pe																	

(Sumber: Data Penelitian (2017))