

BAB II

KAJIAN PUSTAKA

2.1. Teori Dasar

Teori adalah seperangkat konsep, definisi, dan proposisi yang tersusun secara sistematis sehingga dapat digunakan untuk menjelaskan dan meramalkan fenomena (Sudaryono, 2015: 13)

2.1.1. Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan (*Artificial Intelligence*) dapat diartikan menjadi kecerdasan buatan yang mana prosesnya berarti membuat atau mempersiapkan, mesin seperti komputer agar memiliki sebuah *intelligence* atau kecerdasan berdasarkan perilaku manusia. Kecerdasan buatan pada dasarnya bertujuan untuk membuat komputer melaksanakan suatu perintah, yang dapat dilakukan oleh manusia. Sistem pendukung keputusan dan sistem informasi manajemen adalah hasil kontribusi dari kecerdasan buatan (Sutojo, *et al.*, 2010: 1).

Menurut (Sutojo, *et al.*, 2010: 13) dalam kecerdasan buatan (*Artificial Intelligence*) terdapat 3 bidang ilmu yaitu:

1. Sistem pakar (*expert system*): sebagai sarana untuk menyimpan pengetahuan para pakar sehingga komputer memiliki keahlian menyelesaikan permasalahan dengan meniru keahlian yang dimiliki pakar.
2. *Fuzzy Logic*: merupakan logika yang berhadapan dengan konsep kebenaran sebagian, dimana logika klasik menyatakan bahwa segala hal dapat di ekspresikan dalam *binary* 0 atau 1.
3. Jaringan Syaraf Tiruan: jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. Secara sederhana JST adalah sebuah alat pemodelan data *statistic non-linear*.

2.1.2. Sistem Pakar

Istilah sistem pakar berasal dari istilah *knowledge-based system*. Istilah *knowledge-based expert system* muncul karena untuk memasukkan masalah, sistem pakar menggunakan pengetahuan seorang pakar yang dimasukkan ke dalam komputer. Seseorang yang bukan pakar menggunakan sistem pakar untuk meningkatkan kemampuan pemecahan masalah, sedangkan seorang pakar menggunakan sistem pakar untuk *knowledge assistant* (Sutojo, *et al.*, 2011: 160)

Sistem pakar dibuat pada wilayah pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia di satu bidang. Sistem pakar mencoba mencari solusi yang memuaskan sebagaimana yang dilakukan seorang pakar (Kusrini, 2008: 12). Menurut Merlina, Nita & Hidayat. R (2012: 1), beberapa definisi sistem pakar menurut beberapa ahli yaitu sebagai berikut:

1. Menurut Durkin: Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh seorang pakar.
2. Menurut Ignizio: Sistem pakar adalah suatu model dan prosedur yang berkaitan, dalam suatu domain tertentu, yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar.
3. Menurut Giarratano dan Riley: Sistem pakar adalah suatu sistem komputer yang bisa menyaingi atau meniru kemampuan seorang pakar.

2.1.3. Ciri-ciri Sistem Pakar

Menurut Sutojo, *et al.*, (2010: 162), sistem pakar memiliki beberapa ciri-ciri sebagai berikut:

1. Terbatas pada bidang yang spesifik.
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada *rules* atau aturan tertentu.
5. Dirancang untuk dikembangkan secara bertahap.
6. *Output* bersifat nasihat atau anjuran.
7. *Output* tergantung dari dialog dengan *user*.
8. *Knowledge base* dan *inference engine* terpisah.

2.1.4. Manfaat Sistem Pakar

Menurut Sutojo, *et al.*, (2010: 160-161), sistem pakar menjadi sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari manusia.
2. Membuat seorang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi di lingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.
7. Andal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit.
8. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti.
9. Bisa digunakan sebagai media pelengkap dalam pelatihan.
10. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.
11. Bisa digunakan sebagai media pelengkap dalam pelatihan.

2.1.5. Keuntungan Sistem Pakar

Menurut Merlina, Nita & Hidayat. R (2012: 4), berikut ini adalah manfaat dan kemampuan sistem pakar:

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli

2. Bisa melakukan proses secara berulang secara otomatis
3. Menyimpan pengetahuan dan keahlian para pakar
4. Meningkatkan output dan produktivitas
5. Meningkatkan kualitas
6. Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka)
7. Mampu beroperasi dalam lingkungan yang berbahaya
8. Memiliki kemampuan untuk mengakses pengetahuan
9. Memiliki reabilitas
10. Meningkatkan kapabilitas sistem komputer
11. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian
12. Sebagai media pelengkap dalam pelatihan

2.1.6. Kelemahan Sistem Pakar

Menurut Merlina, Nita & Hidayat. R (2012: 4) sistem pakar juga memiliki kelemahan diantaranya yaitu:

1. Untuk mendapatkan pengetahuan tidaklah selalu mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, kadang-kadang pendekatan yang dimiliki oleh pakar tersebut berbeda-beda.
2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharanya.

3. Boleh jadi sistem tidak dapat membuat keputusan.
4. Sistem pakar perlu diuji ulang secara teliti sebelum digunakan, sehingga dalam hal ini faktor manusia tetaplah dominan.

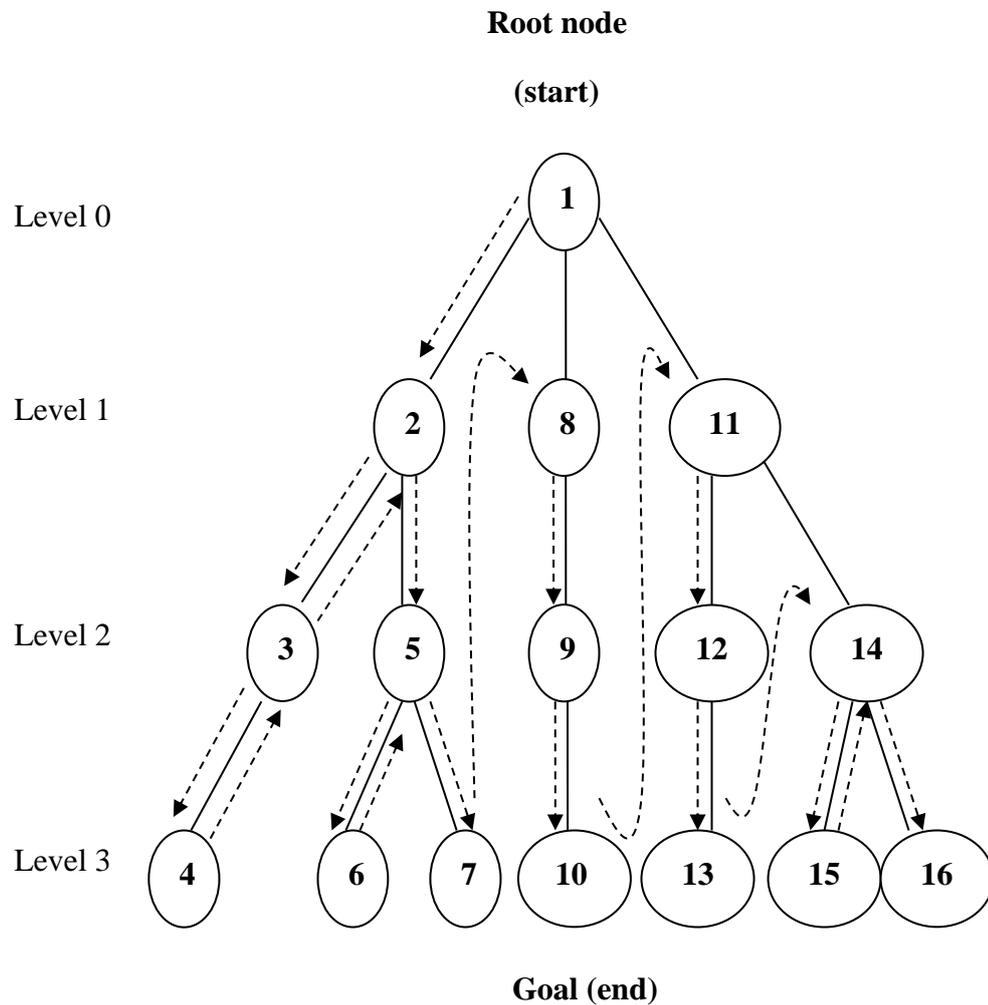
2.1.7. Metode Penelusuran Sistem Pakar

Menurut Suyanto (2014: 17) metode penelusuran sistem pakar sebagai berikut:

1. Depth-First Search

Pencarian dilakukan pada suatu simpul salam setiap level dari yang kiri. Jika pada level yang terdalam dalam solusi belum ditemukan, maka pencarian dilanjutkan pada simpul sebelah kanan dan simpul yang kiri dapat dihapus dari memori.

Kelebihan DFS adalah pemakaian memori yang lebih sedikit. DFS hanya menyimpan sekitar bd simpul, dimana b adalah faktor percabangan dan d adalah kedalaman solusi



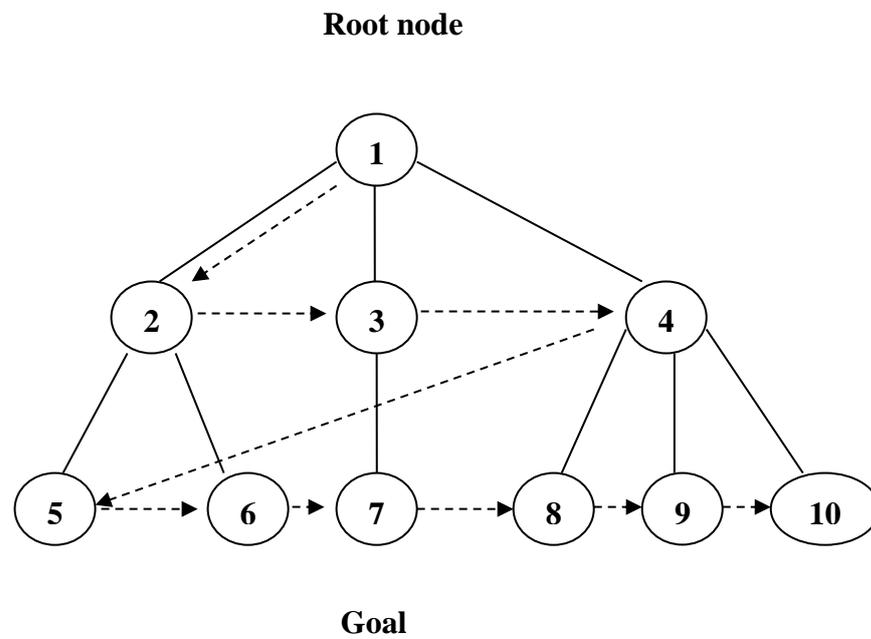
Gambar 2.1 Penelusuran *Depth-First Search*

Sumber: Suyanto (2014: 15)

2. *Breadth-First Search (BFS)*

Pencarian dilakukan pada semua simpul dalam setiap level secara berurutan dari kiri ke kanan. Jika pada satu level belum ditemukan solusi, maka pencarian dilanjutkan pada level berikutnya. Keuntungan dengan pencarian teknik ini adalah sama dengan *depth-first search*, hanya saja penelusuran

dengan teknik ini mempunyai nilai tambah, dimana semua *node* akan di cek secara menyeluruh pada setiap tingkatan *node*.



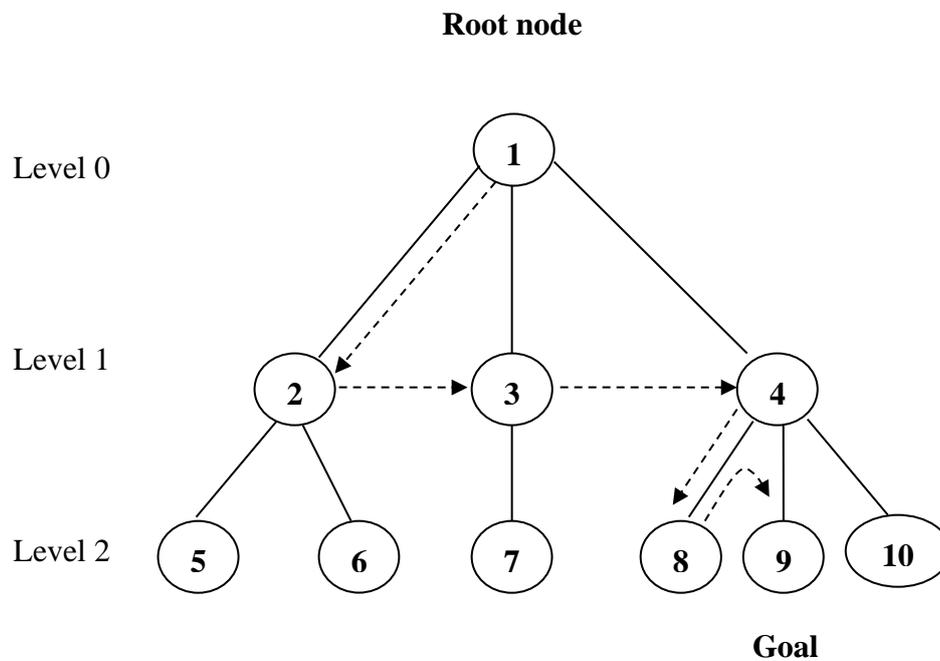
Gambar 2.2 Penelusuran *Bread-first Search*

Sumber: Suyanto (2014: 17)

3. *Best First-Search*

Penelusuran *best-first search* adalah penelusuran yang bekerja berdasarkan kombinasi yang lebih dari metode *depth-first search* dan *breadh-first search*. Pencarian jenis ini dikenal juga sebagai *heuristic*. Pendekatan yang dilakukan adalah mencari solusi yang terbaik berdasarkan pengetahuan yang dimiliki sehingga penelusuran dapat ditentukan harus dimulai dari mana dan bagaimana menggunakan proses terbaik untuk mencari solusi. Keuntungan jenis penelusuran ini adalah mengurangi beban komputasi karena hanya solusi sudah mendekati terbaik. Ini merupakan model yang menyerupai cara manusia

mengambil solusi, hanya saja solusi yang diambil bisa saja salah dan tidak ada jaminan bahwa solusi yang dihasilkan merupakan solusi yang mutlak benar.



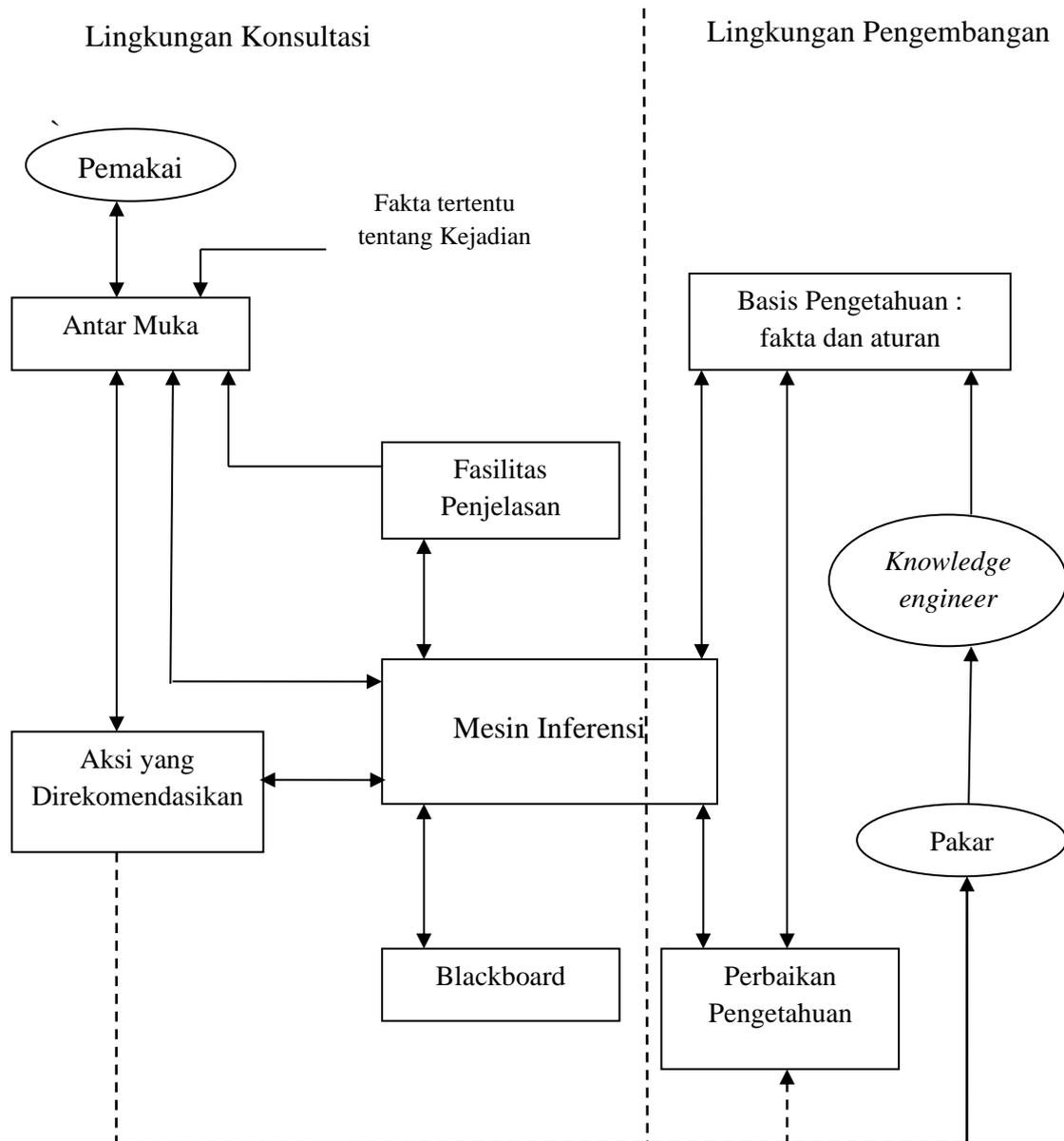
Gambar 2.3 Penelusuran *Best-first Search*

Sumber: Suyanto (2014: 21)

2.1.8. Struktur Sistem Pakar

Menurut (Sutojo, *et al.*, 2011: 166) ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen-komponen dan memasukkan 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 sistem pakar.



Gambar 2.4 Struktur Sistem Pakar

Sumber: Sutojo, et al., (2011:167)

Pada gambar 2.4 dapat dijelaskan sebagai berikut:

1. Basis Pengetahuan

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

a) Penalaran berbasis aturan (*Rule-Based Reasoning*)

Basis pengetahuan dipresentasikan dengan menggunakan aturan berbentuk IF-THEN. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu (Sutojo, *et al.*, 2011: 167).

b) Penalaran berbasis kasus (*Case-Based Reasoning*)

Basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (Sutojo, *et al.*, 2011: 168)

2. Mesin Inferensi (*Inference Engine*)

Mesin inferensi berfungsi untuk melakukan penelusuran pengetahuan yang terdapat dalam basis pengetahuan untuk mencapai kesimpulan tertentu. Mesin Inferensi menyediakan arahan tentang bagaimana menggunakan pengetahuan sistem dalam membangun agenda yang mengorganisasikan dan mengontrol langkah yang diambil untuk memecahkan persoalan saat konsultasi berlangsung (Sutojo, *et al.*, 2011: 168).

Menurut (Sutojo, *et al.*, 2011: 168) ada 3 elemen utama dalam mesin inferensi yaitu:

a. *Interpreter*

Mengeksekusi item-item agenda yang terpilih dengan menggunakan aturan-aturan dalam basis pengetahuan yang sesuai.

b. *Scheduler*

Akan mengontrol agenda.

c. *Consistency enforce*

Bertujuan memelihara konsistensi dalam mempresentasikan solusi yang bersifat darurat.

3. Daerah Kerja *Blackboard*

Blackboard digunakan untuk menggambarkan masalah dan mencatat hasil sementara sebelum mendapatkan solusi terakhir. Tiga tipe keputusan yang dapat disimpan pada *Blackboard* adalah rencana yaitu bagaimana memecahkan persoalan. Agenda yaitu aksi potensial yang menunggu eksekusi. Hipotesa dan aksi yang sudah diproses akan diproses dalam solusi (Sutojo, *et al.*, 2011: 168).

4. Antarmuka Pemakai (*User Interface*)

Antarmuka digunakan mempermudah komunikasi antar pemakai dengan sistem. Komunikasi tersebut berupa permintaan informasi yang diperlukan sistem untuk pencarian solusi, pembagian informasi dari pemakai, pemberian informasi dari pemakai kepada sistem, permintaan informasi penjelasan dari pemakai kepada sistem, permintaan informasi penjelasan oleh pemakai dan pemberian informasi oleh sistem.

5. Fasilitas Penjelasan

Fasilitas penjelasan membantu perekayasa pengetahuan untuk memperbaiki dan meningkatkan pengetahuan, member kejelasan dan keyakinan kepada pemakai tentang proses atau hasil yang diberikan sistem pakar. Fasilitas ini digunakan untuk melacak respon dan memberikan penjelasan tentang sistem pakar secara interaktif melalui pertanyaan:

- a. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar
- b. Bagaimana konklusi dicapai
- c. Mengapa ada alternatif yang dibatalkan
- d. Rencana apa yang akan digunakan untuk mencapai suatu solusi

6. Fasilitas Perbaikan Pengetahuan

Pakar manusia dapat menganalisa performansnya sendiri, belajar darinya dan meningkatkannya untuk konsultasi berikut. Adanya evaluasi dengan sistem pakar ini akan menghasilkan basis pengetahuan yang lebih baik serta penalaran yang lebih efektif.

2.1.9. Metode Inferensi Sistem Pakar

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur-prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar (Sutojo *et al.*, 2010: 164-165).

Menurut (Sutojo, *et al.*, 2010: 171) ada dua teknik penalaran (*inference*) yaitu sebagai berikut:

1. *Forward Chaining*

Forward chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari *rules* IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam *database*. Setiap kali pencocokan dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berisi *rule* yang bisa dieksekusi (Sutojo, *et al.*, 2010: 171).

2. *Backward Chaining*

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* di bagian IF. Jika cocok, *rule* dieksekusi, kemudian hipotesis di bagian *THEN* ditempatkan di basis data

sebagai fakta baru. Jika tidak cocok, simpan *premis* di bagian *IF* ke dalam *stack* sebagai *subGoal*. Proses berakhir jika *Goal* ditemukan atau tidak ada rule yang bisa membuktikan kebenaran dari *subGoal* atau *Goal* (Sutojo *et al*, 2010: 178).

2.2. Variabel

Variabel penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi dan kesimpulannya (Sudaryono, 2015: 16).

2.2.1. Virus Herpes Koi

Menurut (Afrianto Eddy *et al.*, 2015: 56) virus herpes koi (KHV) merupakan agen penyebab yang spesifik menyerang ikan mas (*Cyprinus carpio*) dan ikan koi. Virus tersebut tergolong sangat ganas karena dapat menyebabkan kematian massal hingga mencapai 100% pada populasi ikan mas dan koi di sentra-sentra budidaya perikanan.

Menurut (Afrianto Eddy *et al.*, 2015: 56) herpes virus pada ikan secara umum didefinisikan sebagai penyebab penyakit mulai infeksi sisik hingga infeksi sistemik yang fatal. Pada herpes virus yang menyerang *cyprinid*, sebelumnya sudah dikenal dengan adanya *pox herpes virus* ikan mas (Cyprinid herpes virus 1, CyHV-1) dan *heamatopoitec necrosis* ikan mas koki (Cyprinid herpes virus 2, CyHV-2).

Menurut (Septiama *et al*, 2008: 3) infeksi KHV disebabkan oleh virus herpes dari family *Herpesviridae* (DNA) yang berbeda dari *Cyprinid herpesvirus* (CHV) penyebab *carp pox*. KHV menular secara horizontal diantara ikan mas dan ikan koki dengan masa inkubasi 4-5 hari dan lamanya sakit 24-28 jam (perakut). KHV dapat diidentifikasi melalui kultur sel.

Kasus wabah penyakit KHV tersebut pertama kali dilaporkan pada tahun 1998 di Israel dan Amerika Serikat yang menyebabkan kematian massal ikan mas dan koi. Kerugian ekonomi yang dialami Israel mencapai 4 juta dollar Amerika Serikat. Di Indonesia serangan KHV pertama kali dilaporkan pada awal tahun 2002 dan selama periode tersebut telah menyebar ke seluruh pulau Jawa dan Bali serta Sumatera (Afrianto Eddy *et al.*, 2015: 57).

Wabah KHV telah menyebar diberbagai area termasuk Asia, USA, Eropa. Penyebaran KHV sudah menyebabkan kematian secara besar-besaran pada budidaya ikan termasuk juga ikan hias koki. Dilaporkan bahwa KHV membunuh 4 dari setiap 5 ikan yang terinfeksi di Jawa Timur, Indonesia. Penularan KHV dilaporkan sudah menyebar lebih dari 5000 pembudidaya ikan dan menyebabkan kerugian 0.5 juta dolar dalam 3 bulan (Agus Sunarto 2002 *dalam* Septiama *et al.*, 2008: 3).

2.2.2. Gejala Penyakit

Adapun gejala penyebab terjadinya virus herpes koi pada ikan mas adalah sebagai berikut:

1. Gejala Klinis

Gejala klinis adalah tanda-tanda yang dapat dilihat langsung dengan mata telanjang atau secara kasat mata pada organ luar maupun pada organ dalam tubuh ikan (Septiama *et al.*, 2008: 4).

Tanda- tanda klinis dari KHV seringnya tidak spesifik, serangan kematian dapat terjadi dengan cepat pada populasi, dimulai dengan kematian dalam 24 - 40 jam setelah terdapat tanda-tanda klinis (Septiama *et al.*, 2008: 4).

Serangan infeksi KHV dapat ditunjukkan dengan adanya lesi-lesi dan tingginya angka kematian. Insang yang terserang berwarna merah dan terdapat bercak-bercak berwarna putih, pendarahan pada insang, mata yang cekung dan warna yang tubuh pucat. Biasanya kondisi penginfeksi KHV ini diikuti dengan infeksi sekunder oleh bakterial, kulit melepuh maupun luka borok di permukaan tubuh, kadang-kadang disertai pendarahan pada sirip/badan (Septiama *et al.*, 2008: 4).

2. Gejala internal

Organ dalam seperti hati, limpa dan ginjal mengalami perubahan warna atau rusak (Septiama *et al.*, 2008: 5).

Kematian ikan yang terserang 1-5 hari setelah gejala awal. Kematian mencapai 100% dalam waktu yang singkat. Keganasan KHV ditunjukkan oleh waktu kematian yang berlangsung cepat setelah ikan menunjukkan tanda-tanda awal terinfeksi KHV. Selain itu, waktu penyebaran dan penularan KHV juga relatif sangat cepat (Afrianto, Eddy *et al.*, 2015: 57).



Gambar 2.5 Ikan Mas Terkena KHV

Sumber: Septiama *et al.*, (2008: 5).

3. Pencegahan

Upaya pencegahan dalam menanggulangi virus herpes koi adalah sebagai berikut (Septiama *et al.*, 2008: 14).

- a. Berhati-hati jika akan memasukkan ikan baru ke dalam kolam, pastikan ikan yang masuk sehat, bebas dari KHV dan sudah dikarantina terlebih dahulu.
- b. Pastikan ikan baru berasal dari sumber terpercaya dan bebas dari KHV.
- c. Ada baiknya memasukkan ikan dari satu sumber yang memang terpercaya.
- d. Jangan mencampurkan ikan mas dengan ikan hias lain, KHV bisa menyebar pada ikan hias karena itu jangan mencampurkan ikan mas untuk menghindarkan dari KHV.
- e. Jaga lingkungan kolam dalam kondisi yang baik dan optimal.
- f. Kurangi jumlah ikan dalam kolam dan hindarkan ikan mas dari stress.

4. Pengobatan

Menurut (Septiama *et al.*, 2008: 16) upaya pengobatan yang dapat dilakukan untuk menanggulangi virus herpes koi adalah sebagai berikut:

- a. Pisahkan koi yang sudah terinfeksi dari ikan yang masih sehat masukkan dalam kolam karantina.
- b. Naikkan suhu air kolam karantina secara perlahan-lahan hingga 30 derajat Celsius dan berikan air secukupnya.
- c. Selama masa pengobatan tidak diberi makan dan berikan antibiotik untuk mencegah infeksi sekunder

2.3. Software Pendukung

Didalam penelitian ini penulis menggunakan beberapa *software* yang mendukung dalam penyusunan laporan skripsi ini.

2.3.1. PHP

Menurut Sibero, Alexander F. K (2012: 49) *PHP* adalah pemrograman (*interpreter*) adalah proses penerjemahan baris sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. Dibandingkan dengan produk-produk tersebut, *PHP* memiliki banyak kelebihan, yaitu Abdulloh Rohi (2015: 3):

1. Kinerja

Kinerja *PHP* sangat cepat. Cukup dengan menggunakan server tunggal yang tidak mahal sudah dapat melayani jutaan hits per hari.

2. Skalabilitas

PHP memiliki arsitektur “*shared-nothing*” yang berarti bahwa skala horizontal dengan sejumlah besar komoditas server dapat diterapkan secara efektif dengan biaya yang murah.

3. Integrasi *database*

PHP memiliki koneksi bawaan (asli) yang tersedia untuk beberapa sistem *database* seperti: *MySQL*, *PostgreSQL*, *Oracle*, *dbm*, *FilePro*, *DB2*, *Hyperwave*, *Informix*, *Interbase*, dan *Sybase database*.

4. *Built-in libraries*

PHP dirancang untuk digunakan pada *web* sehingga memiliki banyak fungsi *built-in* untuk dapat melakukan banyak tugas *web* yang berguna seperti menghasilkan gambar dengan cepat, koneksi ke layanan *web* dan layanan jaringan lainnya, *parse XML (eXtended Markup Language)*, mengirim *e-mail*, bekerja dengan *cookie*, dan menghasilkan dokumen *PDF*. Semua itu dilakukan cukup dengan beberapa baris kode.

5. Biaya yang murah

Menggunakan *PHP* tidak membutuhkan biaya alias gratis. *PHP* versi terbaru dapat di-*download* setiap saat dari situs resminya tanpa biaya.

6. Mudah dipelajari dan digunakan

Sintaks *PHP* didasarkan pada bahasa pemrograman lainnya, terutama bahasa *C* dan *Perl*. Jika pengguna sudah mengetahui bahasa *C* atau *Perl*, atau bahasa *C* lainnya seperti *C++* atau *Java*, maka akan segera menjadi produktif dalam menggunakan *PHP*.

7. Dukungan *object-oriented* yang kuat

PHP memiliki rancangan fitur *object-oriented* yang baik seperti pewarisan (*inheritance*), atribut dan metode *private* dan *protected*, kelas dan metode *abstract*, antarmuka (*interfaces*), *constructors*, *destructors*, dan *iterators*.

8. Portabilitas

PHP tersedia untuk banyak sistem operasi yang berbeda. Kode *PHP* dapat ditulis pada sistem operasi *Unix* seperti *Linux*, *FreeBSD*, *Solaris* dan *IRIX*, *Operating System X (OS X)* maupun sistem operasi *Microsoft Windows* yang memiliki versi yang berbeda-beda. Kode yang ditulis biasanya akan bekerja tanpa dimodifikasi pada sistem lain yang menjalankan *PHP*.

9. Fleksibilitas dalam pembangunan

PHP memungkinkan penerapan tugas-tugas sederhana secara ringkas dan sama-sama mudah beradaptasi dalam penerapan aplikasi yang besar menggunakan kerangka kerja berdasarkan pola desain seperti *Model View Control (MVC)*.

10. Tersedianya kode sumber (*source code*)

PHP menyediakan akses ke kode sumber *PHP* (bersifat *open-source*), tidak seperti produk komersial lainnya yang bersifat *closed-source*. Pada *PHP*, dapat dilakukan modifikasi atau penambahan di dalamnya secara bebas tanpa harus menunggu produsen untuk mengeluarkan *patches* (program kecil untuk perbaikan sistem).

11. Tersedianya *support* (dukungan) dan dokumentasi



Gambar 2.6 Logo *PHP*

Sumber: <https://www.php.net/download-logos.php>

2.3.2. HTML

Menurut Sibero, Alexander F.K (2012: 19) HTML (*hypertext Markup Language*) adalah bahasa digunakan dokumen *web* sebagai bahasa untuk pertukaran dokumen *web* untuk dapat menghasilkan halaman *website* sehingga halaman tersebut dapat diakses pada setiap komputer pengakses (*client*). HTML digunakan untuk menampilkan berbagai informasi didalam sebuah penjelajah *web internet* dan *formatting hypertext* yang ditulis ke dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi (Syahputra Bagus, 2012: 1).



Gambar 2.7 Logo *HTML*

Sumber: https://www.w3.org/html/logo/downloads/HTML5_Logo_512.png

2.3.3. Javascript

Berbeda dengan php yang diproses di sisi server, *javascript* diproses pada komputer *client*. Karena pemrosesannya dilakukan di komputer *client*, membuat *javascript* lebih interaktif dibanding *php*. Peran *javascript* dalam membuat website adalah memberikan efek animasi yang menarik, dan interaktifitas dalam penanganan event yang dilakukan oleh pengguna *website* (Abdulloh Rohi, 2015:3).



Gambar 2.8 Logo Javascript

Sumber: http://www.w3devcampus.com/wp-content/uploads/logoAndOther/logo_JavaScript.png

2.3.4. CSS

CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*. Walaupun *HTML* mempunyai kemampuan untuk mengatur desain *website*, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan HTML terlihat lebih rapi dan indah (Abdulloh Rohi 2015: 2).



Gambar 2.9 Logo CSS

Sumber: <http://w3widgets.com/responsive-slider/img/css3.png>

2.3.5. MySQL

Menurut Anhar (2010: 45) *MySql* adalah salah satu *databases management system (DBMS)* dari sekian banyak *DBMS* seperti *Oracle, MS SQL, Postagre SQL* dan lainnya. *MySQL* bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman *PHP* juga mendukung/*support* dengan *database MySQL*.



Gambar 2.10 MySQL

Sumber: <https://en.wikipedia.org/wiki/File:MySQL.svg>

2.3.6. XAMPP

Menurut Wicaksono, Y (2008: 7) *XAMPP* adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis *PHP* dan menggunakan pengolah

data *MySQL* dikomputer *local*. *XAMPP* berperan sebagai *server web* pada komputer anda. *XAMPP* juga dapat disebut sebuah *panel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.



Gambar 2.11 *XAMPP*

Sumber: https://commons.wikimedia.org/wiki/File:Xampp_logo.svg

2.3.7. Notepad++

Notepad++ adalah *source code* editor gratis dan *Notepad++* mendukung beberapa bahasa pemrograman. Berjalan di lingkungan *MS Windows*, penggunaannya diatur oleh *GPL License*. Berdasarkan pada komponen editor yang kuat *Scintilla*, *Notepad++* ditulis dalam *C++* dan murni menggunakan *Win32 API* dan *STL* yang menjamin kecepatan eksekusi lebih tinggi dan ukuran program yang lebih kecil. Dengan mengoptimalkan sebagai rutinitas sebanyak mungkin tanpa kehilangan keramahan pengguna, *Notepad++* berusaha untuk mengurangi emisi karbon dioksida dunia. Bila menggunakan daya CPU yang sedikit, PC dapat bekerja lebih ringan dan mengurangi konsumsi daya, sehingga menghasilkan lingkungan yang lebih hijau. (<http://notepad-plus-plus.org/>)



Gambar 2.12 Logo *Notepad++*

Sumber: <http://www.farescd.com/wp-content/uploads/2015/04/Notepad-.jpg>

2.4. UML (*Unified Modeling Language*)

Menurut A. S., Rosa & Shalahuddin, M (2011: 118) *Unified Modelling Language* (UML) adalah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak (A. S., Rosa & Shalahuddin. M, 2011: 118).

Menurut Nugroho (2010:117) bangunan dasar metodologi UML menggunakan tiga bangunan dasar untuk mendeskripsikan sistem/perangkat lunak yang akan dikembangkan yaitu:

1. Sesuatu (*things*)

Ada empat *things* dalam UML yaitu:

1. *Structural things* merupakan bagian yang relatif statis dalam model *Unified Modeling Language* (UML). Bagian yang relatif statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.

2. *Behavioral things* merupakan bagian yang dinamis pada model *Unified Modeling Language* (UML), biasanya merupakan kata kerja dari model *Unified Modeling Language* (UML), yang mencerminkan perilaku sepanjang ruang dan waktu.
3. *Grouping things* merupakan bagian pengorganisasi dalam *Unified Modeling Language* (UML). Dalam penggambaran model yang rumit kadang diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model dan subsistem-subsistem.
4. *Annotational things* merupakan bagian yang memperjelas model *Unified Modeling Language* (UML) dan dapat berupa komentar-komentar yang menjelaskan fungsi serta ciri-ciri setiap elemen dalam model *Unified Modeling Language* (UML).

2. Relasi (*Relationship*)

Ada empat macam *relationship* dalam *Unified Modeling Language* (UML) yaitu:

1. Ketergantungan merupakan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*dependent*).
2. Asosiasi merupakan apa yang menghubungkan antara objek satu dengan objek lainnya, bagaimana hubungan suatu objek dengan objek lainnya.

Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.

3. Generalisasi merupakan hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). Arah dari atas ke bawah dari objek induk ke objek anak dinamakan spesialisasi, sedangkan arah berlawanan sebaliknya dari arah bawah ke atas dinamakan generalisasi.
4. Realisasi merupakan operasi yang benar-benar dilakukan oleh suatu objek.

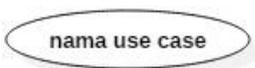
Pada penelitian ini menggunakan empat jenis UML diagram yaitu:

2.4.1. Use Case Diagram

Diagram ini bersifat statis. *Diagram* ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). *Diagram* ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna (A. S., & Shalahuddin. M , 2011: 130).

Dibawah ini akan menjelaskan simbol-simbol dari *Use case*:

Tabel 2.1 Simbol *Use Case*

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/actor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi

 nama aktor	yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>
<i>asosiasi/association</i> <hr style="width: 100%;"/>	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>
<i>Ekstensi/extend</i> <<extend>> <hr style="width: 100%;"/> →	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.
<i>generalisasi/generalization</i> <hr style="width: 100%;"/> →	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)
Menggunakan/ <i>include/uses</i> <<include>> <hr style="width: 100%;"/> → <<uses>> <hr style="width: 100%;"/> →	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

Sumber: A.S., Rosa & Shalahuddin. M (2011: 131-132)

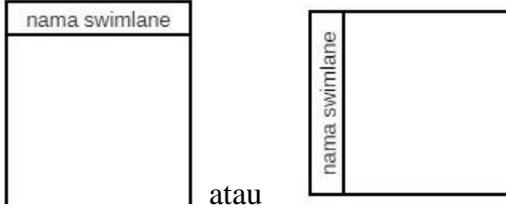
2.4.2. Activity Diagram

Diagram ini bersifat dinamis. *Diagram* ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek (A. S., Rosa & Shalahuddin. M, 2011: 134).

Dibawah ini akan dijelaskan simbol-simbol *Activity* diagram yaitu:

Tabel 2.2 Simbol *Activity* Diagram

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

	
<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi</p>

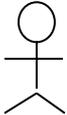
mber: Rosa A.S & Shalahuddin. M (2011: 134-135)

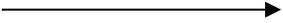
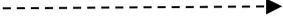
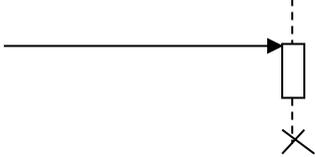
2.4.3. Sequence Diagram

Diagram ini bersifat dinamis. *Diagram sequence* merupakan diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu (A. S., Rosa & Shalahuddin. M, 2011: 137).

Dibawah ini akan menjelaskan simbol-simbol *Sequence* diagram yaitu:

Tabel 2.3 Simbol *Sequence* Diagram

Simbol	Deskripsi
<p>Aktor/<i>actor</i></p>  <p>nama actor</p>	<p>Menggambarkan orang yang sedang berinteraksi dengan sistem.</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p>	<p>Menyatakan objek berinteraksi pesan.</p>

<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <u>nama objek : nama kelas</u> </div>	
<p>Waktu aktif</p> 	<p>Menyatakan suatu objek dalam keadaan aktif dan berinteraksi pesan.</p>
<p>Pesan tipe <i>create</i></p> <p><create></p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> <p>/: nama_metode ()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek-lain atau dirinya sendiri.</p>
<p>Pesan tipe <i>send</i></p> <p>/: masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkandata/masukan/informasi ke objek lainnya.</p>
<p>Pesan tipe <i>return</i></p> <p> : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika <i>create</i> maka ada <i>destroy</i>.</p>

Sumber: A. S., Rosa & Shalahuddin, M (2011: 138-139)

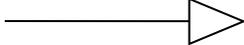
2.4.4. Class Diagram

Diagram class bersifat statis. *Diagram* ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi serta relasi (A. S., Rosa & Shalahuddin. M, 2011: 122).

Dibawah ini akan dijelaskan simbol *class diagram*.

Tabel 2.4 Simbol *Class Diagram*

Simbol	Deskripsi
kelas 	kelas pada struktur sistem
antarmuka/ <i>interface</i>  nama_ <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>multicipy</i>
Generalisasi	relasi antar kelas dengan makna

	generalisasi-spesialisasi (umum khusus)
ketergantungan/ <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., Rosa & Shalahuddin. M (2011: 123)

2.5. Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama seperti judul penelitian penulis. Namun penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya bahan kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa hasil penelitian terdahulu oleh beberapa peneliti yang pernah penulis baca diantaranya:

Penelitian yang dilakukan oleh Aristoteles, Wardiyanto dan Adye Amando Pratama tahun 2015, dengan judul penelitian **Sistem Pakar Diagnosa Penyakit Pada Ikan Budidaya Air Tawar dengan Metode *Forward Chaining***, pada penelitian tersebut dijelaskan bahwa sistem pakar yang dibangun dapat memberikan hasil diagnosa berdasarkan fakta-fakta yang diberikan, aplikasi dapat

membantu pembudidaya dalam mendiagnosa penyakit ikan dan memberikan solusi terkait penyakit yang diderita layaknya seorang pakar.

Penelitian yang dilakukan oleh Elfani dan Ardi Pujiyanta tahun 2013, dengan judul penelitian **Sistem Pakar Mendiagnosa Penyakit Ikan Konsumsi Air Tawar Berbasis Website**, pada penelitian yang telah dilakukan, dihasilkan sebuah perangkat lunak (*software*) baru tentang sistem pakar berbasis *website* untuk mengidentifikasi penyakit pada ikan konsumsi air tawar. Dalam proses penelusuran informasinya di dukung dengan *Theorema Bayes* untuk mendukung kepastiannya.

Penelitian yang dilakukan oleh Murwantoko, Triyanto & Dimas A. Pamungkas tahun 2010, dengan judul **Pengembangan Metode Loop-Mediated Isothermal Amplification Of Dna dan Aplikasinya Untuk Deteksi Koi Herpes Virus Pada Beberapa Jenis Ikan**, pada penelitian tersebut dijelaskan bahwa Ikan karper merupakan inang bagi KHV dan KHV bisa terdeteksi selama pengamatan. Ikan tawes, bawal air tawar, grasscarp dan komet dapat berperan sebagai vector bagi KHV Keberadaan KHV pada tawes dan bawal terdeteksi selama 4 hari, *grasscarp* dan komet selama 5 hari.

Penelitian yang dilakukan oleh Marina F.O Singkoh tahun 2012, dengan judul penelitian **Tingkat Kesukaan Parasit pada Ikan Mas (*Cyprinus carpio*.L) yang Dipelihara dalam Wadah Jaring Apung di Desa Eris, Kabupaten Minahasa, Provinsi Sulawesi Utara (*The parasites preference on carp (Cyprinus carpio.L) cultivated in fish farming cage in the Eris Village, Minahasa District, North Sulawesi Province*)**, pada penelitian tersebut

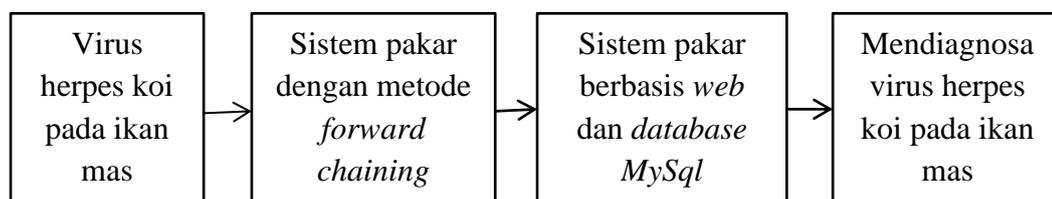
dijelaskan bahwa tingkat kesukaran parasit setelah dianalisis ternyata menunjukkan adanya perbedaan pada bagian organ yang disukai. Organ mulut 0,0014%, organ mata 0,0007%, organ insang 0,8%, organ sirip 0,033% dan organ usus 0,04%. Jenis-jenis parasit yang ditemukan ternyata menyukai bagian eksternal (pemeriksaan ektoparasit) tubuh ikan dibandingkan bagian internal (pemeriksaan endoparasit).

Penelitian yang dilakukan oleh Suwarsito dan Hindayati Mustafidah tahun 2011, dengan judul penelitian **Diagnosa Penyakit Ikan Menggunakan Sistem Pakar (*Diagnosing Fish Disease Using Expert System*)**, pada penelitian tersebut dijelaskan bahwa bisa mendiagnosa sebab-sebab penyakit ikan, bisa memberikan informasi cara penanggulangan penyakit ikan dengan cepat dan efektif.

2.6. Kerangka Pemikiran

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

Pada penulisan sistem pakar ini penulis terlebih dahulu menjabarkan kerangka pemikiran sebagai berikut:



Gambar 2.13 Kerangka Pemikiran

Sumber: Pengolahan Data Penelitian, 2017

Data-data yang dibutuhkan berkaitan dengan penyakit virus herpes koi pada ikan mas dianalisis terlebih dahulu agar lebih sederhana atau mudah dilakukan proses pengolahan datanya. Data-datanya tersebut kemudian diolah menggunakan sistem pakar menggunakan metode *forward chaining*. Sistem pakar yang menggunakan metode *forward chaining* ini menggunakan bahasa pemrograman *PHP* dan *database MySQL* yang dapat digunakan untuk mendiagnosa penyakit virus herpes koi pada ikan mas dan menghasilkan *output* (hasil diagnosa).