BAB II
KAJIAN PUSTAKA

2.1 Teori Dasar

Tujuan dari penjelasan teori dasar ini adalah untuk menjelaskan beberapa teori pendukung dalam proses perancangan dan implementasi dari sistem pakar yang akan dikembangkan. Maka dari itu ada beberapa teori yang dijelaskan adalah sebagai berikut :

2.1.1 Kecerdasan Buatan (Artificial Intelligence)

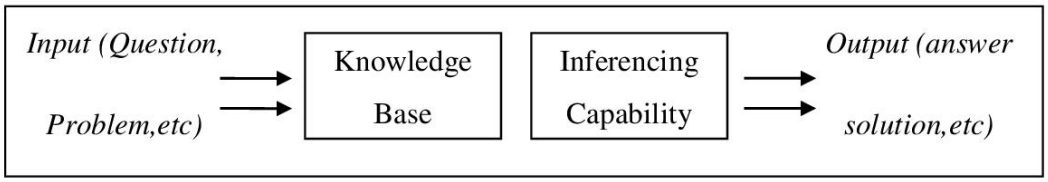
Artificial Intelligence merupakan salah satu bagian ilmu komputer yang membuat agar mesin atau komputer dapat melakukan pekerjaan seperti layaknya dan sebaik yang dilakukan oleh manusia. Teknologi Komputer diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu seperti yang dapat dikerjakan oleh manusia (Martin dan Oxman, 2007).

Untuk membuat aplikasi kecerdasan buatan, ada 2 bagian utama yang sangat dibutuhkan, yaitu :

1. Basis pengetahuan (*Knowledge Base*) berisi fakta-fakta, teori, pemikiran, dan hubungan antara satu dengan yang lainnya.
2. Motor inferensi (*Inference Engine*) yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.

Dengan basis pengetahuan dan kemampuan untuk menarik kesimpulan melalui pengalaman, komputer dapat disejajarkan sebagai alat bantu yang bisa

digunakan secara praktis dalam memecahkan masalah dan pengambilan keputusan yang dapat dilihat pada gambar berikut :



Gambar 2.1 Penerapan konsep *Artificial Intelligence*
Sumber : Suparman & Marlan (2007)

Dengan teknik pelacakan basis pengetahuan untuk mencari fakta dan hubungannya yang *relevan*, komputer bisa mencapai satu atau lebih solusi alternatif pada masalah yang diberikan. Basis pengetahuan komputer dan kemampuan *inference* telah meningkatkan daya guna komputer bagi manusia.

Suyanto (2014:11) mengatakan bahwa saat ini *hardware* dan *software* semakin cepat perkembangannya. Berbagai produk *Artificial Intelligence* (AI) telah berhasil dibangun dan digunakan dalam kehidupan sehari-hari. Dengan teknologi *hardware* yang performansinya semakin tinggi dan berukuran kecil serta didukung teknologi *software* yang semakin beragam dan kuat, produk-produk berbasis *Artificial Intelligence* semakin dekat dengan kehidupan manusia. Pada masa mendatang, *Artificial Intelligence* ditantang untuk membuat suatu kecerdasan yang hampir menyamai kecerdasan manusia.

Menurut Suyanto (2014:4-5) defenisi *Artificial Intelligence* yang paling tepat untuk saat ini adalah *acting rationally* dengan pendekatan *rational agent.* Hal ini berdasarkan pemikiran bahwa komputer bisa melakukan penalaran secara logis dan juga bisa melakukan aksi secara rasional berdasarkan hasil penalaran tersebut.

Artificial Intelligence telah memberikan suatu kemampuan baru kepada komputer untuk memecahkan masalah yang lebih besar dan lebih luas, tidak hanya terbatas kepada soal-soal perhitungan, penyimpanan, dan pengambilan data atau pengendalian yang sederhana.

Kecerdasan buatan berasal dari bahasa inggris "*Artificial intelligence*" atau disingkat AI, yaitu *inteligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia (Sutojo, 2011:1).

2.1.2 Jaringan Syaraf Tiruan (Artificial Neural Networks)

Jaringan Syaraf Tiruan adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplemintasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran (Andrijasa dan Mistianingsih, 2010) dalam penelitian (Eka Pandu Cynthia, 2017).

Pada penelitian ini akan dirancang jaringan syaraf tiruan model lapisan banyak yang mana arsitektur tipe ini memiliki satu atau lebih lapisan yang terletak diantara lapisan masukan dan lapisan keluaran, memiliki juga satu atau lebih lapisan tersembunyi. Umumnya, ada lapisan bobot-bobot yang terletak antara dua lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit juga.

2.1.3 Sistem Pakar (*Expert System*)

Sampai saat ini sudah banyak sistem pakar yang dibuat, seperti *MYCIN* untuk diagnosis penyakit, *DENDRAL* untuk mengidentifikasi struktur molekul campuran yang tak dikenal, *XCON* & *XSEL* ntuk membantu konfigurasi sistem komputer besar, *SOPHIE* untuk analisis sirkuit elektronik, *prospector* digunakan dibidang geologi untuk membantu mencari dan menemukan *deposite*, *FOLIO* digunakan untuk membantu memberikan keputusan bagi seorang manager dalam stok dan investasi, *DELTA* dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya. Istilah sistem pakar berasal dan istilah *knowledge based expert system*. Istilah ini muncul karena untuk memecahkan 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 urituk *knowledge assistant* (Sutojo, 2011:159-160).

Menurut Turban (2011:160) sistem pakar adalah sebuah *system* yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia.

Modul Penyusun Sistem Pakar yang disusun oleh tiga modul utama (Staugaard, 2003) dengan pembagian sebagai berikut :

1. Modul Menerima Informasi atau *Knowledge Acquisition Mode*.

   Sistem yang terdapat pada modul ini, pada saat ia mendapat masukan dari pakar. Pengumpulan informasi yang akan dipergunakan untuk mengembangkan sistem, hal ini memerlukan bantuan dari *knowledge engineer*. Fungsi *knowledge engineer* adalah sebagai penghubung pada suatu sistem pakar dengan pakarnya.

2. Modul konsultasi

   Disaat sistem sudah siap memberikan sebuah jawaban atas beberapa permasalahan yang diberikan dari pengguna sistem pakar berada dalam modul konsultasi. Pada modul ini, *user* berinteraksi dengan sistem dengan menjawab pertanyaan-pertanyaan yang diajukan. Pada kesempatan kali ini penulis membuat modul konsultasi yang berkaitan dengan aspek kepribadian seseorang.

3. Modul penjelasan atau *explanation mode*

   Pada tahap ini, modul memberikan penjelasan dalam proses pengambilan keputusan oleh sistem atau bagaimana caranya suatu keputusan dapat diperoleh yang pada akhirnya memberikan sebuah keluaran yang tepat sesuai dengan permasalahan *user* tersebut.

Sistem pakar merupakan program yang dapat menggantikan keberadaan seorang pakar. Alasan mendasar mengapa sistem pakar dikembangkan untuk menggantikan seorang pakar adalah :

1. Dapat menyediakan kepakaran setiap waktu dan diberbagai lokasi.

2. Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar.
3. Seorang pakar akan pensiun atau pergi.
4. Menggunakan jasa seorang pakar memerlukan biaya yang mahal.
5. Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat (*hostile environtment*).

Untuk memahami perancangan sistem pakar, perlu dipahami mengenai elemen manusia yang berinteraksi dengan sistem antara lain :

1. Pakar (*Expert)* adalah seorang ahli yang dapat menyelesaikan masalah yang sedang diusahakan untuk dipecahkan oleh sistem.
2. Pembangun pengetahuan (*knowledge engineer*) adalah seseorang yang menerjemahkan pengetahuan seorang pakar dalam bentuk deklaratif sehingga dapat digunakan oleh sistem pakar.
3. Pengguna (*User*) adalah seseorang yang berkonsultasi dengan sistem untuk mendapatkan saran yang disediakan oleh pakar.
4. Pembangun sistem (*system engineer*) adalah seseorang yang membuat antarmuka pengguna, merancang bentuk basis pengetahuan secara deklaratif dan mengimplementasikan mesin inferensi.

2.1.3.1 Tujuan dan Manfaat Sistem Pakar

Tujuan pengembangan sistem pakar sebenarnya bukan untuk menggantikan peran manusia sebagai seorang pakar, tetapi untuk mensubsitusikan pengetahuan manusia ke dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak.

Begitu banyak manfaat yang dapat diperoleh dengan mengembangkan sistem pakar, antara lain :

1. Orang non-pakar dapat memanfaatkan keahlian didalam bidang tertentu tanpa kehadiran langsung seorang pakar.
2. Meningkatkan produktivitas kerja, yaitu bertambah efisiensi pekerjaan tertentu serta hasil solusi kerja.
3. Penghematan waktu dan biaya dalam menyelesaikan masalah yang kompleks.
4. Memberikan penyederhanaan solusi untuk kasus-kasus yang kompleks dan berulang-ulang.
5. Memungkinkan untuk menggabungkan pengetahuan dari beberapa orang pakar dan kemudian dikombinasikan.

Selain banyaknya manfaat yang diperoleh, tentu ada kelemahannya, kelemahan pengembangan sistem pakar, yaitu :

1. Daya kerja dan produktifitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh sistem.
2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat konvensional. Hal ini dapat dilihat dari Tabel 2.1 berikut ini :

Tabel 2.1 Perbandingan Perangkat Lunak Konvensional dan Perangkat Lunak Sistem Pakar

Perangkat Lunak Konvensional	**Perangkat Lunak Sistem Pakar**
Fokus pada solusi	Fokus pada permasalahan
Pengembangan dapat dilakukan oleh individu.	Pengembangan dilakukan oleh tim kerja.
Pengembangan secara sekuensial.	Pengembangan secara iteratif.

2.1.3.2 Tahap Pengembangan Sistem Pakar

Pengembangan sistem pakar dapat dibagi menjadi atas beberapa tahap :

1. Identifikasi (Inisialisasi kasus)

   Tahap ini merupakan tahap penentuan hal-hal penting sebagai dasar dari permasalahan yang akan dianalisis. Tahap ini merupakan tahap untuk mengkaji dan membatasi masalah yang akan diimplementasikan dalam sistem.

2. Konseptualisasi (Analisa dan Rancangan Sistem)

   Hasil identifikasi masalah dikonseptualisasikan dalam bentuk relasi antar data, hubungan antar pengetahuan dan konsep-konsep penting yang ideal yang akan diterapkan dalam sistem. Konseptualisasi juga menganalisis data-data penting yang harus dikerjakan bersama dengan pakar dibidang permasalahan tersebut. Hal ini dilakukan untuk memperoleh konfirmasi hasil wawancara dan observasi sehingga hasilnya dapat memberikan jawaban pasti bahwa sasaran permasalahan tepat, benar dan sesuai.

3. Formalisasi (*Prototype* Dasar)

   Pada tahap ini, konsep-konsep yang ada pada tahap konseptualisasi akan diimplementasikan secara formal. Misalnya memberikan kategori sistem yang akan dibangun, mempertimbangkan beberapa faktor pengambilan keputusan seperti keahlian manusia, kesulitan dan tingkat kesulitan yang mungkin terjadi, dokumentasi kerja dan lain-lain.

4. Implementasi Sistem

   Tahap ini dapat dimulai dengan membuat garis besar masalah kemudian memecahkan masalah tersebut ke dalam modul-modul. Untuk memudahkan maka harus diidentifikasikan apa saja yang menjadi inputan, bagaimana proses digambarkan dalam bagan alur dan basis aturannya, apa saja yang menjadi *output* atau hasil dan kesimpulan. Sesudah itu semuanya diubah kedalam bahasa yang mudah dimengerti oleh komputer dengan menggunakan tahapan fase pada pengembangan sistem pakar.

5. Evaluasi

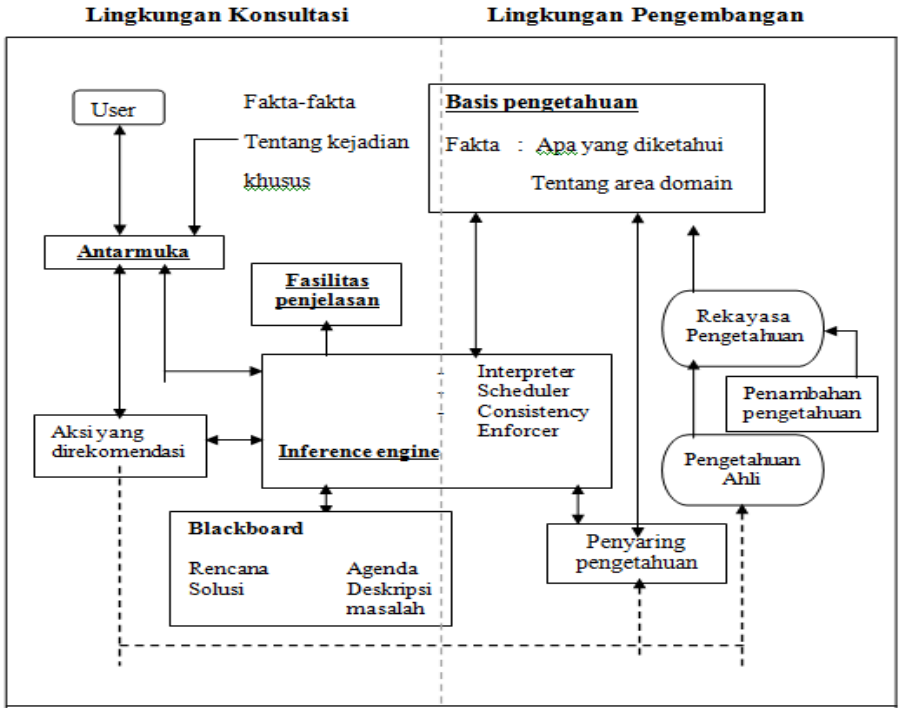
   Sistem pakar yang selesai dibangun perlu untuk dievaluasi untuk menguji dan menemukan kesalahannya. Hal ini merupakan hal yang umum dilakukan, karena suatu sistem belum tentu sempurna setelah selesai pembuatannya, sehingga proses evaluasi diperlukan untuk menyempurnakan.

6. Pengembangan Sistem (Implementasi tahap lanjut)

   Pengembangan sistem diperlukan sehingga sistem yang dibangun investasi sistemnya tidak sia-sia.

2.1.3.3 Struktur Sistem Pakar

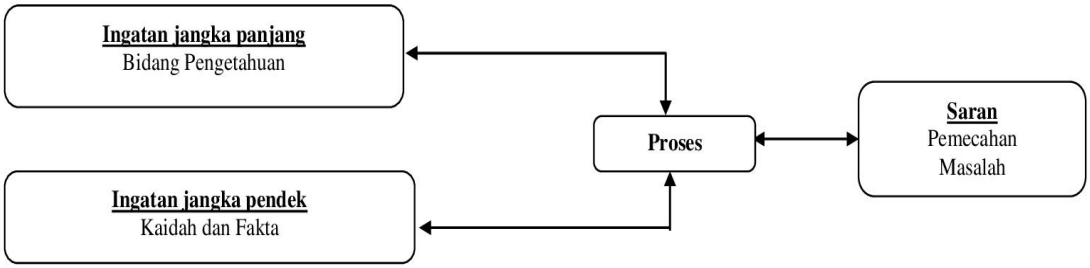
Sistem pakar adalah program komputer yang menirukan penalaran seorang pakar dengan keahlian pada suatu wilayah pengetahuan tertentu (Turban, 2005). sistem pakar dapat ditampilkan dengan dua lingkungan, yaitu :



Gambar 2.2 Struktur Sistem Pakar
Sumber : Sri Hartati & Sari Iswanti (2008)

Seorang pakar mempunyai pengetahuan tentang masalah tertentu yang disebut dengan *Domain Knowledge*. Pakar menyimpan *Domain Knowledge* pada *Long Term Memory* (LTM) atau ingatan jangka panjangnya.

Cara pemecahan masalah dapat dilihat pada gambar berikut :



Gambar 2.3 Pemecahan Masalah Pada Pakar
Sumber : Sri Hartati & Sari Iswanti (2008)

Dalam pembangunanya sistem pakar memiliki beberapa komponen yang saling berhubungan, yaitu : Basis pengetahuan (*Knowledge Base*), Mesin Inferensi (*Inference Engine*) dan Antarmuka Pemakai (*User Interface*).

1. Basis Pengetahuan (*Knowledge Base*)

   Suatu *knowledge base* terdiri dari kumpulan data tertentu untuk permasalahan yang spesifik dan aturan-aturan bagaimana memanipulasi data yang disimpan tersebut. Elemen dasar dari basis pengetahuan terdiri dari fakta yang berupa informasi tentang obyek dan kaidah *(rule)* yang merupakan informasi tentang tatacara bagaimana membangkitkan fakta baru dari fakta yang sudah diketahui atau fakta lama. Pengetahuan diklasifikasikan menjadi :

   a. Pengetahuan procedural (*Procedural knowledge*)

      Pengetahuan ini lebih menekankan kepada bagaimana cara melakukan sesuatu.

b. Pengetahuan deklaratif (*Declarative knowledge*)

Yaitu pengetahuan yang menjawab pertanyaan apakah sesuatu itu bernilai benar atau salah.

c. Pengetahuan tacit (*Tacit knowledge*)

Pengetahuan yang tidak bisa diungkapkan dengan bahasa.

2. *User Interface*

User Interface merupakan bagian *software* yang menyediakan sarana untuk *user* agar bisa berkomunikasi dengan sistem. *Interface* akan mengajukan pertanyaan atau menyajikan menu pilihan untuk memasukkan informasi awal dalam basisdata *User Interface* menyediakan pula sarana komunikasi jawaban atau solusi bila masalahnya sudah ditemukan. Setiap komunikasi antara selama proses pemecahan masalah komunikasi dikendalikan oleh *User Interface*.

3. Representasi pengetahuan (*Knowledge representation*)

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 ciri-ciri dari metode representasi pengetahuan adalah :

a. Harus bisa diprogram dengan bahasa pemrograman dan hasilnya disimpan dalam memori.

b. Dirancang sedemikian baik sehingga isinya dapat digunakan untuk proses penalaran.

c. Model representasi pengetahuan merupakan sebuah struktur data yang dapat dimanipulasi oleh mesin inferensi dan pencairan untuk aktifitas percocokan pola.

Ada banyak cara untuk merepresentasikan pengetahuan, diantaranya adalah logika (*logic*), jaringan semantik (*Semantic Nei*s), objek atribut value (OAV), bingkai (*Frame*) dan kaidah produksi (*Production Rule*) (Kusrini, 2007).

4. Metode *Inferensi*

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 modul yang disebut *Inference Engine* (Mesin Inferensi). Fungsi inferensi adalah sebagai pembuktian hipotesis. Bila hipotesis sudah dimasukkan ke dalam sistem pakar, maka mesin inferensi pertama-tama mengecek apakah hipotesis sudah ada dalam basis data atau belum. Jika sudah, maka hipotesis dianggap sebagai fakta yang sudah dibuktikan, sehingga operasi tidak perlu dilanjutkan (Suparman dan Marlan, 2007).

Dalam melakukan inferensi diperlukan adanya proses pengujian kaidah-kaidah dalam urutan tertentu untuk mencari yang sesuai dengan kondisi awal atau kondisi yang sedang berjalan pada basis data. Perunutan adalah proses pencocokan fakta, pernyataan atau kondisi berjalan yang tersimpan pada basis pengetahuan maupun pada memori kerja dengan kondisi yang

dinyatakan pada premis atau bagian kondisi pada kaidah. Beberapa pendekatannya yaitu :

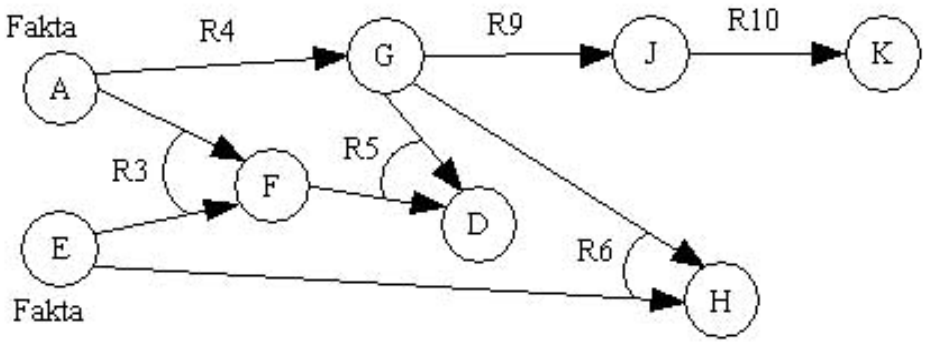
a. Runut Maju (*Forward Chaining*)

Runut maju atau penalaran *forward* (*Forward Reasoning*) merupakan proses perunutan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Jadi dimulai dari premis-premis atau informasi masukan (*if*) dahulu kemudian menuju konklusi atau *driven information (then)* atau dapat dimodelkan seperti berikut :

IF (informasi masukan)

THEN (konklusi)

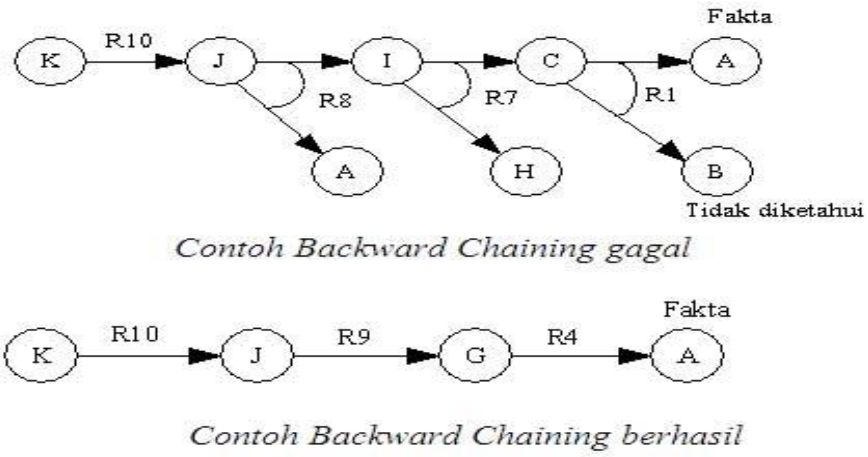
Pada runut maju,system tidak akan melakukan praduga apapun, namun system akan mengecek gejala-gejala tersebut untuk memenuhi konklusi yang sama. Runut maju memulai proses pencarian dengan data sehingga strategi ini disebut juga *data-driven* (Wawan Yunanto, 2007), metode penalaran ini dapat dilihat pada gambar berikut :



Gambar 2.4 *Forward Chaining*

b. Runut Balik (*Backward Chaining*)

Runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam runut balik penalaran dimulai dengan tujuan kemudian merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut (Giarattano dan Riley, 1994). Runut balik disebut juga sebagai *goal-drivenreasoning*, merupakan cara yang efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur, metode penalaran ini dapat dilihat pada gambar berikut :



Gambar 2.5 *Backward Chaining*

2.1.3.4 Ciri - Ciri Sistem Pakar

Ada beberapa cirri-ciri sistem pakar antara lain :

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 rule atau kaidah tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Outputnya bersifat nasihat atau anjuran.
7. Output tergantung dari dialog dengan *user*.

2.1.4 *Fuzzy Logic*

Menurut Sutojo, dkk (2011:211) dalam penelitian (Nasir & Suprianto, 2017) konsep tentang logika *fuzzy* diperkenalkan oleh Prof. Lotfi Astor Zadeh pada Tahun 1962, Logika *fuzzy* adalah metodologi sistem control pemecahan masalah, yang cocok untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana, sistem kecil, *embedded system*, jaringan PC, *multichannel* atau *workstation* berbasis akuisisi data, dan sistem control.

Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai dua kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk” dan lain-lain. Oleh karena itu, sistem ini dapat mempunyai nilai keanggotaan 0 atau 1. Akan tetapi, dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1. Artinya, bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya.

Bila dibandingkan dengan logika konvensional, kelebihan logika *fuzzy* adalah kemampuannya dalam proses penalaran secara bahasa sehingga dalam perancanganya tidak memerlukan persamaan matematik yang rumit. Himpunan *fuzzy* memiliki dua atribut yaitu : (1) *Linguistik*, yaitu nama suatu kelompok yang

mewakili suatu keadaan tertentu dengan menggunakan bahasa alami, misalnya DINGIN, SEJUK, PANAS mewakili variabel temperatur, (2) *Numeris*, yaitu suatu nilai yang menunjukkan ukuran dari suatu variabel, misalnya 10, 35, 40, dan sebagainya.

2.1.5 Metode Mamdani

Menurut Sutojo, dkk (2011:235) Metode Mamdani paling sering digunakan dalam aplikasi-aplikasi karena strukturnya yang sederhana, yaitu menggunakan operasi *MIN-MAX* atau *MAX-PRODUCT*. Untuk mendapatkan *output*, diperlukan 4 tahapan berikut. (1) *Fuzzyfikasi*, (2) Pembentukan basis pengetahuan *Fuzzy* (*rule* dalam bentuk *IF…THEN*), (3) Aplikasi fungsi implikasi menggunakan fungsi *MIN* dan Komposisi antar *rule* menggunakan fungsi *MAX* (menghasilkan himpunan *fuzzy* baru), (4) *Deffuzyfikasi* menggunakan metode *Centroid* (Nasir & Suprianto, 2017).

2.2 Variabel Penelitian

Untuk variabel yang akan digunakan pada penelitian ini dengan menggunakan sistem pakar mendiagnosa kerusakan pada komputer adalah dibagi menjadi 5 variabel yaitu kerusakan pada *Power Supply, RAM (Random Access Memory), Motherboard, Harddisk* dan *Processor*. Variabel kerusakan tersebut sering dialami pada komputer sehingga perlu dirancang dan dibangun sistem untuk mempermudah kerja pengguna sistem dalam mendiagnosa kerusakan pada komputer tersebut.

2.2.1 Komputer

Kata Komputer berasal dari kata bahasa Yunani "*Computare*" yang berarti memperhitungkan atau menggabungkan bersama-sama. Kata *com* berarti menggabungkan dalam pikiran atau secara mental, sedangkan *putare* berarti memikirkan perhitungan atau penggabungan. Dalam bahasa Inggris: "*To Compute*" yang artinya menghitung. Komputer adalah sistem yang terdiri dari beberapa aspek yaitu *Hardware, Software,* dan *Brainware*.

Perangkat Keras (*Hardware*) adalah suatu perangkat berbentuk fisik yang digunakan untuk pemrosesan informasi, menerima informasi, sebagai proses transmisi data yang lebih efektif. *Hardware* berfungsi untuk mengelola informasi sehingga berguna bagi yang menggunakan terutama bagi organisasi. *Hardware* terdiri dari beberapa perangkat seperti *processor, monitor, keyboard,* dan *printer*. *Hardware* dapat digunakan sebagai media penyimpanan. *Hardware* dapat menghubungkan beberapa perangkat dengan menggunakan jaringan sehingga cara kerjanya lebih efektif dan efisien. Maka tak dapat dipungkiri bahwa hardware komputer merupakan salah satu bagian dari komputer yang paling penting (Rainer, 2011:65).

1. *Power Supply*

   Fungsinya untuk menyuplai arus kepada semua komponen yang tersambung pada motherboard sehingga motherboard dapat berfungsi sebagaimana mestinya dan komputer dapat dioperasikan.

Gambar 2.6 *Power Supply*

2. RAM (*Random Access Memory)*

Random Access Memory adalah bagian perangkat keras komputer yang digunakan sebagai penyimpanan data sementara sehingga memudahkan *processor* dalam memproses data. RAM bisa dikatakan *harddisk* dan semacamnya tetapi jauh lebih cepat dalam memproses data, Tetapi apa bila komputer dalam kondisi *off memory* RAM akan kosong artinya tidak bisa menyimpan data dalam jangka panjang, RAM akan terisi memori dalam hal ini intruksi ketika komputer mulai bekerja dan digunakan untuk meringankan kerja *processor* salah satu perangkat keras komputer.

Gambar 2.7 RAM *(Random Access Memory)*

3. *Motherboard*

Motherboard adalah papan sirkuit tempat berbagai komponen elektronik saling terhubung seperti pada *Personal Computer* atau *Macintosh* dan biasa

disingkat dengan kata lain *Mobo*. *Motherboard* yang banyak ditemui dipasaran saat ini adalah *motherboard* milik *personal computer* yang pertama kali dibuat dengan dasar agar dapat sesuai dengan pesifikasi PC IBM. *Motherboard* ini berfungsi untuk menempatkan semua alat-alat *controller*. Termasuk juga *Processor, RAM, Power Supply, I/O Controller, Display Controller* dan semua alat ditempatkan pada *socket* yang telah disediakan oleh *motherboard*.

Gambar 2.8 *Motherboard*

4. *Harddisk*

Harddisk adalah perangkat keras komputer sebagai pusat penyimpanan data secara permanen, *harddisk* bertindak sebagai penyimpanan data utama yang kemudian berfungsi sebagai data *server*. Hasil kerja CPU akan disimpan pada *harddisk* dan sejenisnya. Kelebihan *harddisk* dibanding RAM adalah, data hasil proses bisa disimpan dan kemudian digunakan pada waktu yang berbeda. *Harddisk* (disk) adalah yang paling penting dari berbagai jenis penyimpanan permanen yang digunakan dalam PC (yang lainnya adalah *disket* dan media penyimpanan lainnya seperti *CD ROM, removable drive*, dll) *harddisk* berbeda dari yang lain terutama dalam tiga

cara: ukuran (biasanya lebih besar), kecepatan (biasanya cepat) dan permanen (biasanya tetap di *personal computer* dan tidak dapat dilepas).

Gambar 2.9 *Harddisk*

5. *Processor*

Processor disebut juga otak dari komputer semakin bagus tipe *processor* maka semakin mahal pula komputer, maka *processor* disebut sebagai inti dari komputer. Semakin tinggi kecepatan dan teknologi *processor,* maka semakin baik pula kinerja komputer. Fungsi *processor* adalah untuk memproses semua kegiatan yang dilakukan komputer, yang di*request* pengguna. Saat ini *Intel* telah mengeluarkan *processor Intel Core2Duo, Core2Quad, i3, i5, i7*, yang mempunyai kemampuan jauh diatas *Intel Pentium 4 dan Dual Core*.

Gambar 2.10 *Processor*

2.3 *Software* Pendukung

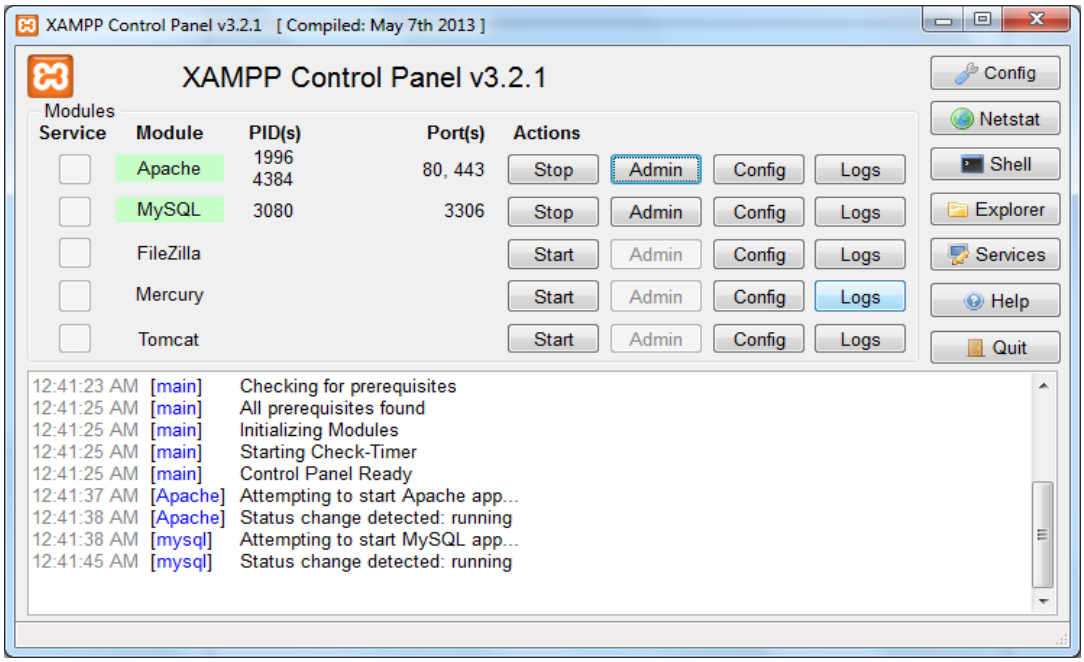
Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999).

Software dalah program komputer yang berfungsi sebagai sarana interaksi antara pengguna (*user*) dan *hardware*. *Software* dapat juga dikatakan sebagai penerjemah perintah yang dijalankan pengguna komputer untuk diteruskan atau diproses oleh *hardware* (Nugroho, 2009).

Rekayasa Perangkat Lunak (*Software enginerring*) adalah ilmu yang mempelajari tentang teknik pembuatan perangkat lunak yang baik dengan pendekatan teknik (*enginerring approach*) (Nugroho, 2009).

2.3.1 *XAMPP*

Web server merupakan perangkat lunak yang dijalankan oleh sistem operasi pada *computer server* maupun *desktop*, yang berfungsi untuk menerima permintaan (*request*) dalam bentuk *protocol*, misalnya HTTP (*Hyper Text Transfer Protocol*) dan HTTPS (*Hyper Text Transfer Protocol Secure*). *Request* tersebut kemudian dibalas (*replay*) dengan cara mengirimkan hasil permintaan tersebut melalui *web browser*. Protokol sendiri merupakan aturan dan standar baku untuk proses komunikasi, hubungan, dan *transfer* data antar *computer* pada jaringan. Aplikasi *web server* yang dapat digunakan antar lain *XAMPP* (Pratama, 2014:439).



Gambar 2.11 Halaman awal *XAMPP*
Sumber : Pengolahan Data Penelitian (2019)

Menurut Pratama (2014:440) *XAMPP* adalah aplikasi *web server* bersifat *instan* (siap saji) yang dapat digunakan baik disistem operasi *Linux* maupun disistem operasi *Windows*.

XAMPP adalah satu paket *software web server* yang terdiri dari *Apache*, MySQL, PHP dan PHPMyAdmin. Mengapa menggunakan *XAMPP*? karena XAMPP sangat mudah penggunaannya, terutama jika seorang pemula. Proses instalasi *XAMPP* sangat mudah, karena tidak perlu melakukan konfigurasi *Apache*, PHP dan MySQL secara Manual, *XAMPP* melakukan instalasi dan konfigurasi secara otomatis (Madcoms 2009:1).

2.3.2 *PHP*

PHP adalah bahasa yang dirancang untuk mudah diletakkan didalam kode *HTML*. Banyak dijumpai kode *PHP* yang menyatu dengan kode *HTML* . kode *PHP* diawali dengan *tag* *<?php* dan diakhiri dengan *tag* ?>. Apabila kita

melakukan konfigurasi terhadap *file php.ini* untuk mengizinkan penggunaan *tag* pendek (*short tag*) dengan mengubah nilai *short_open tag* menjadi *On*, maka *tag* tersebut dapat diganti dengan <? dan ?>. Dalam PHP5, nilai *default* dari *short_open_tag* adalah *off*. Selain itu, PHP kita juga dapat menggunakan *tag* gaya ASP, <% dan %>, dengan mengubah nilai *asp_tags* dalam *file php.ini* menjadi *On* (Raharjo, 2014:48).

Menurut Raharjo (2014:47) PHP adalah salah satu bahasa pemorograman *script* yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan di*parsing* didalam *web server* oleh *interpreter* PHP dan diterjemahkan kedalam dokumen HTML, yang selanjutnya akan ditampilkan kembali pada *web browser*. Karena pemrosesan program PHP dilakukan dilingkungan *web server*, PHP dikatakan sebagai bahasa sisi *server* (*server-side*). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah "*View Source*" pada *web browser* yang mereka gunakan. Selain menggunakan PHP, aplikasi *web* juga dapat dibangun dengan *java* (JSP – *JavaServer Pages* dan *Servlet*), *Perl*, *Python*, *Ruby*, maupun ASP (*Acrive Server Pages*). Meskipun PHP5 dapat digunakan untuk membuat aplikasi CLI (*command Line Interface*) dan juga aplikasi *desktop* (seperti *Perl*, *Python*, dan *Ruby*), namun pada umumnya orang menggunakan PHP untuk tujuan pembuatan aplikasi *web*.

PHP adalah salah satu bahasa pemrograman yang berjalan didalam *server*, dan mampu membuat *web* menjadi interaktif dan dinamis. PHP dapat mengolah

data dari *computer client* dan dari *computer sever* itu sendiri, sehingga mudah disajikan dalam *browser* (Madcoms 2009:133).

2.3.2.1 Kelebihan *PHP* Dari Bahasa Pemrograman Lain

Kelebihan *PHP* dari bahasa pemrograman lain adalah :

1. Bahasa pemrograman *php* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. *Web Server* yang mendukung *php* dapat ditemukan dimana-mana dari mulai IIS sampai dengan *apache*, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya *milis-milis* dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, php adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. *PHP* adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*linux, unix, windows*) dan dapat dijalankan secara *runtime* melalui console serta juga dapat menjalankan perintah-perintah sistem.

2.3.3 SQL (*Structured Query Language*)

SQL (*Structured Query Language*) adalah bahasa yang dipergunakan untuk mengakses data dalam basis data *relation*. Bahasa ini secara *defacto* adalah bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua *server* basis data yang ada mendukung bahasa ini dalam manajemen datanya.

2.3.3.1 Pembagian SQL

SQL dapat dibagi menjadi dua bagian yaitu bahasa manipulasi data (*The Data Manipulation Language***)** dan bahasa definisi data (*the data definition language*). Yang termasuk dalam interface *DML* (*Data Manipulation Language*) adalah : *SELECT , UPDATE, DELETE, INSERT INTO*.

Interface yang temasuk penting dalam pernyataan DDL (*Data Manipulation Language*) adalah *CREATE DATABASE, ALTER DATABASE, CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, DROP INDEX*.

2.3.4 UML (*Unified Modeling Language*)

Menurut Windu Gata, Grace (2013:4) dalam penelitian (Hendini, 2016), *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case Diagram*

   Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

3. Diagram Urutan (*Sequence Diagram*)

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

4. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, attribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau attribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

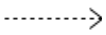
5. *Deployment Diagram*

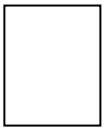
Deployment Diagram digunakan untuk menggambarkan detail bagaimana komponen disusun diinfrastruktur sistem.

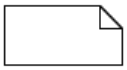
Menurut (A.S & Shalahuddin, 2011) *UML* memiliki diagram grafis untuk membuat suatu model, yaitu :

1. *Use case Diagram*

Pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang dibuat.

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (*ancestor*).

4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan

			prilaku yang lebih besar dari jumlah dan elemen-elemennya
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

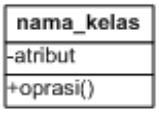
Tabel 2.2 Simbol Diagram *Use Case*
Sumber : (A.S. & M.Shalahuddin, 2014)

2. Diagram Kelas (*Class Diagram)*

Diagram ini menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2.3 Simbol Diagram Kelas

No	Gambar	Nama	Keterangan
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (*ancestor*).

2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang

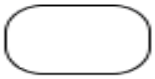
			bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber : (A.S. & M.Shalahuddin, 2014)

3. Diagram Aktifitas (*Activity Diagram*)

Diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Tabel 2.4 Simbol Diagram Aktifitas

No	Gambar	Nama	Keterangan
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan

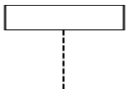
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

Sumber : (A.S. & M.Shalahuddin, 2014)

4. Diagram Sekuen (*Sequence Diagram*)

Diagram ini menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Tabel 2.5 Simbol Diagram Sekuen

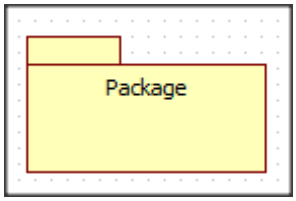
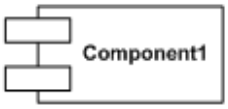
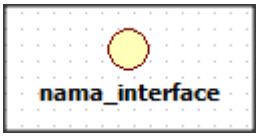
No	Gambar	Nama	Keterangan
1		*Life Line*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : (A.S. & M.Shalahuddin, 2014)

5. Diagram Komponen (*Component Diagram*)

Diagram ini dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram ini berfokus pada komponen sistem yang dibutuhkan dan ada didalam sistem.

Tabel 2.6 Simbol Diagram Komponen

No	Simbol	Nama	Deskripsi
1		*Package*	*Package* merupakan sebuah bungkusan dari satu atau lebih komponen.
2		Komponen	Komponen system
3		Ketergantungan (*Dependency*)	Ketergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4		Antarmuka/ *Interface*	Sama dengan interface pada pemrograman berbasis objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung

			komponen.
5		*Link*	Relasi antar komponen.

Sumber : (A.S. & M.Shalahuddin, 2014)

2.4 Penelitian Terdahulu

(Erdani, 2012): Developing Backward Chaining Algorithm of Inference Engine in Ternary Grid Expert System *"The inference engine is one of main components of expert system that influences the performance of expert system. The task of inference engine is to give answers and reasons to users by inference the knowledge of expert system. Since the idea of ternary grid issued in 2004, there is only several developed method, technique or engine working on ternary grid knowledge model. The in 2010 developed inference engine is less efficient because it works based on iterative process. The in 2011 developed inference engine works statically and quite expensive to compute. In order to improve the previous inference methods, a new inference engine has been developed. It works based on backward chaining process in ternary grid expert system. This paper describes the development of inference engine of expert system that can work in ternary grid knowledge model. The strategy to inference knowledge uses*

backward chaining with recursive process. The design result is implemented in the form of software. The result of experiment shows that the inference process works properly, dynamically and more efficient to compute in comparison to the previous developed methods".

(Komal R.Hole, 2014): ***Expert System for Diagnosis of Memory Low Diseases*** "*The proposed system will initially discuss different approaches in designing of Medical Diagnosis Expert Systems with focus on all the information about the memory loss. The different symptoms and causes of memory loss at different age groups and the precautions for any kind of memory loss. It is an attempt to focus on some of very important diseases related to memory loss like Alzheimer's disease, Parkinson's disease, Huntington's disease, and multi-infarct which are among the most common types of memory loss diseases. The study, conducted on postmortem human brain cells and in mice, revealed that the hippocampus in the brain - a region that plays an important part in memory, lacks a protein called RbAp48 in those who experience age-related memory loss. The finding suggests that a deficiency of this protein is a cause of memory loss, but more importantly, the researchers say this form of memory loss is reversible. This proposed Expert System will help the patients to get the required advice about the different disorders attack to them due to their nervous system disorders. The expert rules were developed on the symptoms of each type of neurological disease, and they were presented using decision tree and inferred using backward-chaining method. The knowledge base consists of information about the memory*

loss and all the diseases related to it which is collected from books and doctors (domain experts) about neurology and its disorders".

(Mahfudin, 2017): ***Web-Based Expert System Application To Recommend Computer Specifications For Gaming Using Backward Chaining Inference Method*** *"This study aims to design and implement an expert system application to provide minimum computerhardware and operating system requirements, which are capable of running games with certain graphical settings. Backward chaining inference method is used to conclude the output, which the requirements are based on user's input. The application is made using PHP general-purpose server-side scripting language and MySQL database. By using this application, a user can consult as well as to an expert to know the computer specifications capable of running a game with preferred graphical settings and to estimate the cost to build that computer. This application can also help sellers of computer parts to set price on the custom-built computers, to create category or sales package, as well as to provide information for potential buyers".*

(Mukhtar, 2014): Sistem Pakar Diagnosa Dampak Penggunaan *Softlens* Menggunakan *Metode Backward Chaining,* "Softlens adalah sejenis lensa yang dibuat dari bahan yang bersifat "lunak", yaitu silicon hydrogen. Penggunaan softlens dalam jangka waktu lama dapat berpotensi menyebabkan iritasi mata, mata merah dan infeksi. Untuk itu diperlukan sebuah sistem pakar untuk membantu mendiagnosa dampak penggunaan softlens. Pembangunan sistem pakar diagnosa dampak penggunaan softlens ini menggunakan metode backward chaining atau runut balik. Metode runut balik bekerja dengan cara

menentukan penyakit yang diderita oleh pengguna softlens kemudian akan dijabarkan sebab-sebab penyakit tersebut. Dari hasil penelitian dapat disimpulkan bahwa sistem pakar ini mempermudah pengguna soflens untuk melakukan diagnosa dampak penggunaan softlens berdasarkan gejala yang dialami, dan mengetahui cara penanggulangannya".

(Herliana, Setiawan, & Prasetio, 2018): Penerapan Inferensi *Backward Chaining* Pada Sistem Pakar Diagnosa Awal Penyakit Tulang, "Tulang merupakan bagian yang sangat penting di dalam bagian *ortopedi* manusia. Tulang bukan hanya kerangka penguat tubuh tetapi juga merupakan bagian dari susunan sendi, sebagai pelindung tubuh, tempat melekatnya bagian ujung otot yang melekat pada tulang. Terbatasnya jumlah pakar Penyakit Tulang serta minimnya pengetahuan masyarakat tentang penyakit tulang menjadi kendala mengapa penyakit ini tidak mudah diatasi. Banyaknya gejala yang mirip untuk menentukan suatu penyakit Tulang. Dari masalah diatas maka dibuatlah aplikasi sistem pakar diagnosa awal penyakit tulang. Dari penelitian yang dilakukan menghasilkan sebuah perangkat lunak Sistem Pendukung Keputusan Klinis berbasis web untuk diagnosa Penyakit Tulang. Informasi yang dihasilkan adalah hasil diagnosa penyakit berdasarkan gejala-gejala yang dipilih oleh user. Hasil uji coba menunjukkan bahwa aplikasi ini layak dan dapat digunakan sebagai alat bantu para medis Penyakit Tulang dalam mendiagnosa awal".

(Wahyudi & Prasetyo, 2018), *Implementing Forward, Backward Chaining and Certainty Factor in Responsive Web-Based Expert System of Cow Disease,* *This research aims to design and build expert systems of cow disease to*

assist farmers in identifying cattle diseases. A large number of cattle in Banyumas is not matched by the number of veterinarians, the Department of Fisheries and Livestock (DINKANAK). Banyumas records 961 cases of sick cows in 2016. This expert system is expected to assist cattle ranchers to identify cow disease and symptom-based remedies illness-symptoms. By using Inference Forward and Backward chaining which is a search method or tracking technique by using information from breeders, and Certainty Factor is used to accommodate the uncertainty of thinking of a data expert that is Doctor the process of extracting knowledge by interview. In this research system development using ESDLC (Expert System Development Life Cycle) with stages of Planning, Knowledge Acquisition, Implementation, and Evaluation. Testing is done with two approaches are Alpha Testing and Beta Testing. Alpha Testing conducted on the developer side to test the functional system using Black Box Testing method result all functional system can function well. Beta Testing is aimed at user acceptance by a Questionnaire method yields an average score of 76% or usability and the quality of system information is easy to.

(Rupnawar, Jagdale, & Navsupe, 2016) Study on Forward Chaining and Reverse Chaining in Expert System, **"***Expert systems are part of a general category of computer applications known as intelligence. Expert system are designed to solve complex problems. Expert Systems is a branch of AI designed to work within a particular domain. To solve expert-level problems, expert systems will need efficient access to a substantial domain knowledge base, and a reasoning mechanism to apply the knowledge to the problems they are given.*

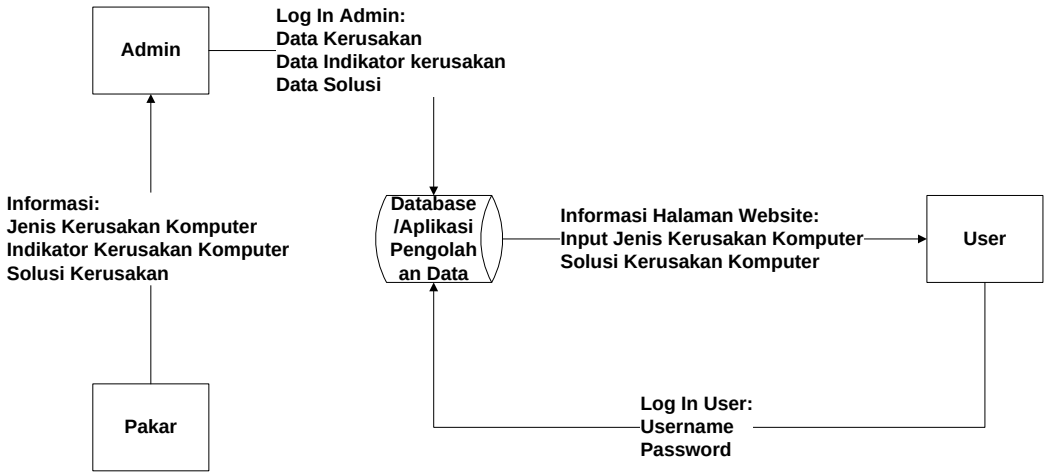
Usually they will also need to be able to explain, to the users who rely on them, how they have reached their decisions. As an expert is a person who can solve a problem with the domain knowledge. This research paper introduces introduction, parts, application of expert system. and difference between forward chaining and Backward chaining and Exactly meaning of Chaining. ETL tools uses functionality to extract, transform and load data from one system into another system, but our expert advises they're not optimal for application-to-application communication. In artificial intelligence, an expert system is a computer system that emulates the decision-making ability of a human expert. The AI technology has become really advanced and its only matter of time when the machines will be able to learn almost anything. The machine learning algorithms are already very smart, however the Processing power has been a challenge in last decade .Now with the big data and distributed computing revolution this problem has become easy to solve. Many programmers and developers can start programming their own robots and other gadgets on their own. Artificial intelligence is a science and technology based on disciplines such as Computer Science, Biology, Psychology, Linguistics, Mathematics, and Engineering. A major thrust of AI is in the development of computer functions associated with human intelligence, such as reasoning, learning, and problem solving.

2.5 Kerangka Pemikiran

Kerangka pemikiran merupakan langkah dan tahapan untuk penerapan sistem pakar mendeteksi kerusakan komputer pada halaman *website* yang dapat diakses oleh pengguna. Dari kerangka tersebut akan menggambarkan secara

umum peran masing-masing aktor yaitu *Admin*, Pakar dan *user*. Secara umum *admin* mempunyai peran untuk mengelola sistem yang dirancang dan dibangun. *Admin* juga mempunyai hak penuh dalam pengelolaan halaman *database* sistem pakar mendeteksi kerusakan pada komputer dengan indikator yang didapatkan dari pakar dibidang komputer.

Berdasarkan informasi yang diberikan oleh pakar maka *admin* dapat merancang sistem dengan menggunakan metode *backward chaining*. Dengan beberapa tahapan yang telah dilakukan oleh *admin* dan pakar maka sebuah halaman *website* baru bisa dikatakan layak dimanfaatkan oleh *user*. Sedangkan *user* merupakan pengguna sistem adalah pengguna yang membutuhkan informasi tentang kerusakan pada komputer dan sistem yang dibangun ini adalah sistem yang akan memberikan solusi berdasarkan kerusakan yang terjadi pada pengguna komputer. *User* wajib melakukan registrasi sebelum menggunakan sistem dengan tujuan untuk memvalidasi data pengguna oleh *admin.*



Gambar 2.12 Kerangka Pemikiran