

**SISTEM PAKAR DIAGNOSA KERUSAKAN  
HARDWARE KOMPUTER MENGGUNAKAN  
METODE PENELUSURAN *BACKWARD CHAINING***

**SKRIPSI**



Oleh:  
**Yosua Hariyanto Sirait**  
130210273

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS PUTERA BATAM  
TAHUN 2020**

**SISTEM PAKAR DIAGNOSA KERUSAKAN  
HARDWARE KOMPUTER MENGGUNAKAN  
METODE PENELUSURAN *BACKWARD CHAINING***

**SKRIPSI**

**Untuk memenuhi salah satu syarat  
Memperoleh gelar Sarjana**



**Oleh:  
Yosua Hariyanto Sirait  
130210273**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN KOMPUTER  
UNIVERSITAS PUTERA BATAM  
TAHUN 2020**

## SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini saya:

Nama : Yosua Hariyanto Sirait

NPM : 130210273

Fakultas : Teknik dan Komputer

Program Studi : Teknik Informatika

Menyatakan bahwa “**Skripsi**” yang saya buat dengan judul:

### **SISTEM PAKAR DIAGNOSA KERUSAKAN *HARDWARE* KOMPUTER MENGUNAKAN METODE PENELUSURAN *BACKWARD CHAINING***

Adalah hasil karya sendiri dan bukan “duplikasi” dari karya orang lain. Sepengetahuan saya, didalam naskah Skripsi ini tidak terdapat karya ilmiah atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip didalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia naskah Skripsi ini digugurkan dan gelar akademik yang saya peroleh dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenarnya tanpa ada paksaan dari siapapun

Batam, 21 Februari 2020

Materai 6000

**Yosua Hariyanto Sirait**  
130210273

**SISTEM PAKAR DIAGNOSA KERUSAKAN  
HARDWARE KOMPUTER MENGGUNAKAN  
METODE PENELUSURAN *BACKWARD CHAINING***

**SKRIPSI**

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

**Oleh:**

**Yosua Hariyanto Sirait  
130210273**

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

**Batam, 21 Februari 2020**

**Very Karnadi, S.Kom., M.Kom.  
Pembimbing**

## ABSTRAK

Peran perangkat komputer dan *internet* sangat berpengaruh dalam pengembangan sumber daya informasi yang *update*. Untuk melakukan perbaikan peralatan elektronik, misalkan pada penelitian ini adalah untuk melakukan perbaikan *hardware* komputer dengan memanfaatkan beberapa teknologi informasi, dengan menggunakan metode sistem pakar. Sistem pakar sangat berperan penting dibidang apapun termasuk perbaikan perangkat keras komputer, karena pengertian dan konsep secara umum bahwa dengan metode sistem pakar berarti kepakaran atau profesionalisme dan kemampuan seseorang yang sudah mengerti tentang permasalahan dan solusi pada sebuah kasus yang dihadapi. *Website* merupakan sebuah media informasi *online* yang mudah dipahami dan dimanfaatkan oleh masyarakat, karena *website* memiliki fitur yang jelas dan kemudian juga tidak terlalu membuat pengguna kesulitan dalam mengoperasikannya, dengan arti lain *website* memiliki fitur dan tampilan dinamis. Penelitian ini bertujuan untuk merancang sistem pakar kerusakan *hardware* komputer, dan di implementasikan terhadap sebuah halaman *website* sederhana. Dalam penelitian ini metode yang digunakan adalah *backward chaining*. Langkah yang dilakukan dimulai dengan pengumpulan data, pengembangan dan hasil. Tahap identifikasi dan analisis kebutuhan sistem dilakukan untuk mengetahui kebutuhan pengguna terhadap sistem yang akan dikembangkan. Berdasarkan dari perhitungan keakuratan data yang diperoleh dari sistem yang dibangun dengan menggunakan sistem pakar dan metode *backward chaining* dalam mengdiagnosa kerusakan komputer adalah sempurna, sehingga sistem tersebut layak digunakan oleh pengguna dengan maksimal sesuai kebutuhan. Setelah diterapkan dengan menggunakan metode *backward chaining*, berdasarkan kepakaran sangat membantu pengguna yang masih awam dengan kerusakan komputer dalam mengatasi permasalahan yang dihadapi pada saat bekerja.

**Kata Kunci:** *Backward Chaining; Hardware; Komputer; Website*

## **ABSTRACT**

*The role of computer and Internet devices is very influential in the development of information resources that update. To perform the repair of electronic equipment, suppose in this research is to perform a repair of computer hardware by utilizing some information technology, using expert system methods. Expert system is very important in any field including repair of computer hardware, because of the understanding and concept in general that with the method of expert system means the expertise or professionalism and the ability of someone who already understands About the problems and solutions in a case encountered. Website is an online information media that is easily understood and utilized by the community, because the website has a clear features and then also does not make the user difficult to operate it, with other means the website Features and dynamic display. This research aims to design a system of expert computer hardware, and implemented against a simple website page. In this study the method used was backward chaining. The steps are initiated with data collection, development and results. The identification and analysis phase of system needs is done to determine the user's need for the system to be developed. Based on the accuracy of the data obtained from the systems built using an expert system and the backward chaining method of diagnoses computer damage is perfect, so the system deserves to be used by Users as needed. After being applied using backward chaining method, based on the expertise very helpful user who still lay with computer damage in solving problems encountered at work.*

**Keyword:** *Backward Chaining; Computer; Hardware; Website*

## **KATA PENGANTAR**

Segala puji dan syukur penulis panjatkan kehadiran Tuhan YME yang telah melimpahkan anugerah-Nya, kemurahan-Nya dan kasih-Nya yang besar, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada program studi Sistem Informasi pada Universitas Putera Batam.

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

1. Rektor Universitas Putera Batam, Ibu Dr. Nur Elfi Husda, S.Kom., M.SI.
2. Ketua Program Studi, Andi Maslan, S.Kom., M.SI.
3. Bapak Very Karnadi, S.Kom., M.Kom. selaku Dosen Pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universits Putera Batam.
5. Kedua Orang Tua tercinta Parman Yanto Sirait dan Sermani Sitorus yang selalu memberikan dukungan Moral dan Moril.
6. Rekan-rekan mahasiswa dan mahasiswi, yang telah banyak memberikan dorongan serta semangat dalam penyelesaian laporan Skripsi ini.

Semoga Tuhan membalas kebaikan dan selalu mencurahkan anugerah-Nya,  
Amin.

Batam, 21 Februari 2020

Yosua Hariyanto Sirait

## DAFTAR ISI

<b>HALAMAN SAMPUL DEPAN.....</b>	<b>i</b>
<b>SURAT PERNYATAAN ORISINALITAS .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iii</b>
<b>ABSTRAK.....</b>	<b>iv</b>
<b>ABSTRACT .....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>vii</b>
<b>DAFTAR GAMBAR.....</b>	<b>viii</b>
<b>DAFTAR TABEL .....</b>	<b>ix</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	3
1.3 Pembatasan Masalah.....	4
1.4 Perumusan Masalah .....	4
1.5 Tujuan Penelitian .....	5
1.6 Manfaat Penelitian .....	5
<b>BAB II KAJIAN PUSTAKA .....</b>	<b>8</b>
2.1 Teori Dasar .....	8
2.2 Variabel Penelitian.....	24
2.3 <i>Software</i> Pendukung .....	29
2.4 Penelitian Terdahulu .....	42
2.5 Kerangka Pemikiran .....	47
<b>BAB III METODE PENELITIAN .....</b>	<b>49</b>
3.1 Desain Penelitian .....	49
3.2 Teknik Pengumpulan Data .....	51
3.3 Operasional Variabel .....	52
3.4 Metode Perancangan Sistem .....	53
3.5 Lokasi dan Jadwal Penelitian .....	71
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>74</b>
4.1 Hasil Penelitian.....	74
4.2 Pembahasan .....	76
4.2.1 Pengujian Validasi .....	76
4.2.2 Pengujian Akurasi.....	80
<b>BAB V SIMPULAN DAN SARAN.....</b>	<b>82</b>
5.1 Simpulan .....	82
5.2 Saran .....	83
<b>DAFTAR PUSTAKA .....</b>	<b>84</b>
<b>LAMPIRAN .....</b>	<b>86</b>



## DAFTAR GAMBAR

<b>Gambar 1.1</b> Survey pengguna TIK .....	2
<b>Gambar 2.1</b> Penerapan konsep Artificial Intelligence.....	9
<b>Gambar 2.2</b> Struktur Sistem Pakar .....	17
<b>Gambar 2.3</b> Pemecahan Masalah Pada Pakar .....	18
<b>Gambar 2.4</b> Forward Chaining.....	21
<b>Gambar 2.5</b> Backward Chaining .....	22
<b>Gambar 2.6</b> Power Supply .....	26
<b>Gambar 2.7</b> RAM (Random Access Memory) .....	26
<b>Gambar 2.8</b> Motherboard.....	27
<b>Gambar 2.9</b> Harddisk.....	28
<b>Gambar 2.10</b> Processor.....	28
<b>Gambar 2.11</b> Halaman awal XAMPP .....	30
<b>Gambar 2.12</b> Kerangka Pemikiran .....	48
<b>Gambar 3.1</b> Desain Penelitian.....	49
<b>Gambar 3.2</b> Pohon Keputusan .....	58
<b>Gambar 3.3</b> Use Case Diagram.....	60
<b>Gambar 3.4</b> Class Diagram.....	61
<b>Gambar 3.5</b> Sequence Diagram Login User.....	63
<b>Gambar 3.6</b> Sequence Diagram Login Pakar .....	64
<b>Gambar 3.7</b> Sequence Diagram Diagnosa.....	65
<b>Gambar 3.8</b> Sequence Diagram Mengelola Gejala.....	66
<b>Gambar 3.9</b> Sequence Diagram Mengelola Solusi .....	67
<b>Gambar 3.10</b> Activity Diagram User Kerusakan Hardware Komputer .....	68
<b>Gambar 3.11</b> Activity Diagram Admin Kerusakan Hardware Komputer .....	69
<b>Gambar 3.12</b> Interface Menu Beranda .....	71
<b>Gambar 4.1</b> Tampilan Halaman Depan .....	74
<b>Gambar 4.2</b> Halaman Konsultasi .....	75
<b>Gambar 4.3</b> Halaman Gejala.....	75
<b>Gambar 4.4</b> Halaman Diagnosa .....	76

## DAFTAR TABEL

<b>Tabel 2.1</b> Perbandingan Perangkat Lunak Konvensional dan Perangkat Lunak Sistem Pakar .....	15
<b>Tabel 2.2</b> Simbol Diagram Use Case .....	35
<b>Tabel 2.3</b> Simbol Diagram Kelas .....	37
<b>Tabel 2.4</b> Simbol Diagram Aktifitas .....	39
<b>Tabel 2.5</b> Simbol Diagram Sekuen .....	40
<b>Tabel 2.6</b> Simbol Diagram Komponen.....	41
<b>Tabel 3.1</b> Operasional Variabel Kerusakan Pada hardware komputer.....	52
<b>Tabel 3.2</b> Kode dan Nama Kerusakan.....	53
<b>Tabel 3.3</b> Kode dan Gejala Kerusakan.....	54
<b>Tabel 3.4</b> Aturan (Rule).....	56
<b>Tabel 3.5</b> Aturan (Rule) Kerusakan Dengan Gejala .....	57
<b>Tabel 3.6</b> Database Gejala .....	70
<b>Tabel 3.7</b> Database Indikator .....	70
<b>Tabel 3.8</b> Database Periksa.....	70
<b>Tabel 3.9</b> Jadwal Penelitian .....	73
<b>Tabel 4.1</b> Tabel Pengujian Validasi Halaman User .....	76
<b>Tabel 4.2</b> Tabel pengujian validasi halaman admin.....	77
<b>Tabel 4.3</b> Hasil Diagnosa .....	80

# **BAB I**

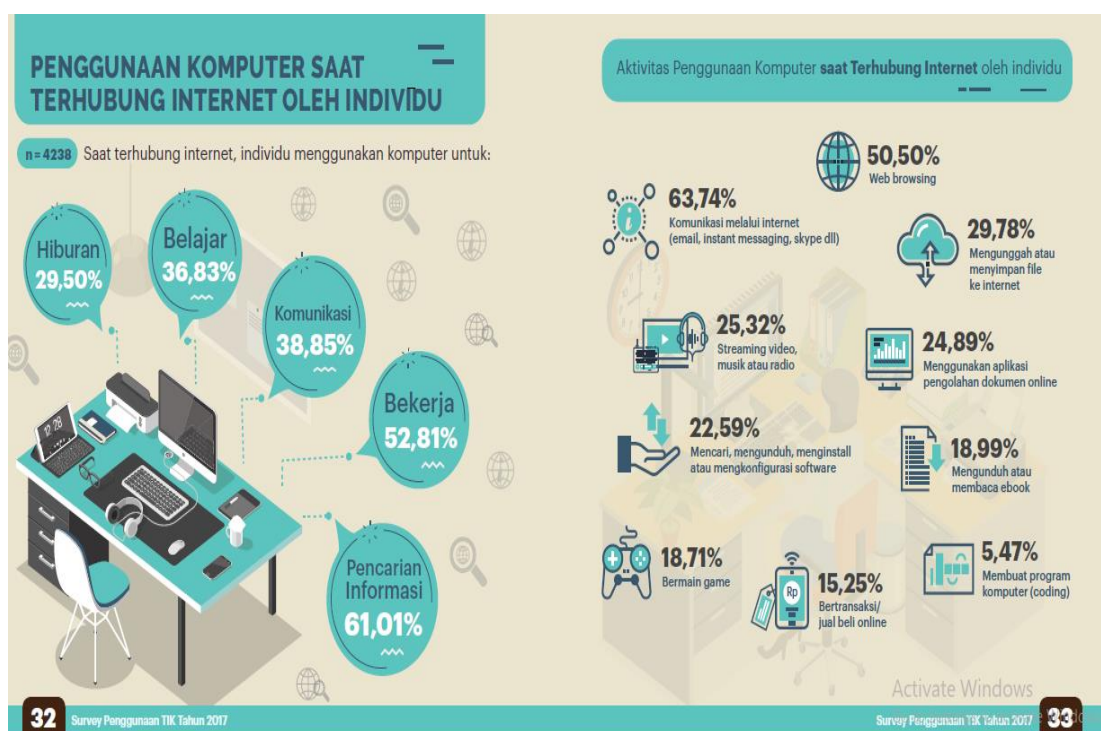
## **PENDAHULUAN**

### **1.1 Latar Belakang**

Istilah Komputer berasal dari kata *Compute*, yang berarti menghitung. Artinya, setiap proses yang dilaksanakan oleh komputer merupakan proses matematika hitungan. Jadi apapun yang dilakukan oleh komputer, baik penampakan pada layar monitor, suara, gambar, dll. Diolah sedemikian rupa dari perhitungan secara elektronik (Sukirman, Munawir & Asri, 2011:30)

Komputer atau laptop saat ini sudah menjadi sebuah kebutuhan bagi pengguna yang beraktifitas dibidang perkantoran maupun untuk bidang pendidikan. Namun tidak semua masyarakat yang mampu memiliki perangkat tersebut sehingga untuk kebutuhan masyarakat menjadi terbatas dikarenakan harga laptop yang lumayan tinggi. Dengan kelebihan laptop dibandingkan perangkat komputer yang mudah dibawa kemana-mana sehingga hanya orang tertentu yang memiliki laptop tersebut.

Sedangkan untuk penggunaan perangkat komputer biasanya didukung dengan pemanfaatan internet sebagai salah satu manfaat untuk sebuah aktifitas baik sebagai pegawai, bisnis, pendidikan maupun games. Sehingga antara perangkat komputer dan internet sudah dijadikan sebuah solusi bagi masyarakat untuk sarana pendukung dalam mendapatkan informasi sesuai dengan kebutuhan pengguna.



**Gambar 1.1** Survey pengguna TIK

Sumber : <https://www.slideshare.net/literasidigital/survey-penggunaan-tik>

Peran perangkat komputer dan *internet* sangat berpengaruh dalam mengembangkan sumber daya informasi yang *update*. Untuk melakukan perbaikan peralatan elektronik, misalkan pada penelitian ini adalah untuk melakukan perbaikan komputer dengan memanfaatkan beberapa teknologi informasi dan metode, misalnya dengan metode sistem pakar.

Sistem pakar sangat berperan penting dibidang apapun, termasuk perbaikan computer, karena pengertian dan konsep secara umum bahwa dengan metode sistem pakar, berarti kepakaran atau *profesionalisme* dan kemampuan seseorang yang sudah mengerti tentang permasalahan dan solusi pada sebuah kasus yang dihadapi. Misalkan kepakaran seorang teknisi dibidang perbaikan komputer,

dimana ilmu dan pengalaman seorang teknisi tersebut yang akan dituangkan kedalam sebuah sistem komputer yang bisa diakses melalui *internet*.

Kemudahan untuk mendapatkan informasi secara *online* sangat berpengaruh besar dalam menunjang hasil kerja seseorang apalagi teknisi yang masih awam dengan kerusakan pada perangkat keras komputer. Dengan cara sistem informasi *online* dan sistem pakar ini teknisi akan lebih mudah untuk menemukan solusi kerusakan pada komputer. Informasi perbaikan komputer akan mudah didapatkan oleh pengguna baik dengan perangkat teknologi atau perangkat digital, karena selain sebagai salah satu alat komunikasi, komputer juga bisa dimanfaatkan sebagai alat pendukung untuk mendapatkan informasi secara *online*.

*Website* merupakan sebuah media informasi *online* yang mudah dipahami dan dimanfaatkan oleh masyarakat karena *website* memiliki fitur yang jelas dan kemudian juga tidak terlalu membuat pengguna susah dalam mengoperasikannya dengan arti lain *website* memiliki fitur dan tampilan dinamis.

Berdasarkan dari kendala-kendala dan pertimbangan yang dipaparkan, maka penulis berusaha membantu dan memudahkan akses informasi dengan mengangkat judul penelitian adalah “Sistem Pakar Diagnosa Kerusakan *Hardware* Komputer Menggunakan Metode Penelusuran *Backward Chaining*”.

## **1.2 Identifikasi Masalah**

Identifikasi masalah adalah untuk mengetahui masalah yang dihadapi dalam merancang sistem yang akan dibangun. Maka berdasarkan uraian diatas masalah yang dihadapi adalah :

1. Susahnya pengguna komputer yang mengalami kerusakan pada *hardware*nya untuk mendeteksi kerusakan dan perbaikan komputer.
2. Dengan keterbatasan informasi untuk mendeteksi kerusakan komputer secara *hardware*, perancang sistem harus membaca dan mengumpulkan beberapa referensi untuk dijadikan sebagai bahan dalam membangun sistem.
3. Perlu sebuah pengetahuan yang dapat memberikan informasi kepada pengguna komputer untuk mengetahui lebih detail komponen yang mengalami permasalahan.

### **1.3 Pembatasan Masalah**

Dengan adanya banyak masalah yang dihadapi, maka perlu rasanya penelitian ini dibatasi yaitu :

1. Hanya meneliti kerusakan pada *Power Supply, Harddisk, Motherboard, Random Access Memory, Processor*.
2. Sistem Pakar yang digunakan adalah dengan metode *Backward Chaining*.
3. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman *PHP* dan *MySQL*.

### **1.4 Perumusan Masalah**

Berdasarkan dari penjelasan latar belakang uraian diatas, maka dapat dirumuskan masalah oleh penulis adalah :

1. Bagaimana merancang Sistem Pakar untuk mendeteksi kerusakan perangkat keras komputer?

2. Bagaimana membangun Sistem Pakar dan diimplementasikan kedalam sebuah halaman *website* sederhana?
3. Bagaimana menerapkan Sistem Pakar dengan menggunakan metode *backward chaining* untuk mendiagnosa kerusakan pada komputer?
4. Bagaimana menerapkan Bahasa Pemrograman *PHP* dan *MySQL* dalam membangun sebuah halaman *website*?

### **1.5 Tujuan Penelitian**

Berdasarkan dari rumusan masalah diatas maka tujuan dari penelitian ini adalah:

1. Merancang Sistem Pakar untuk mendeteksi kerusakan perangkat keras komputer.
2. Membangun Sistem Pakar dan diimplementasikan kedalam sebuah halaman *website* sederhana.
3. Menerapkan Sistem Pakar dengan menggunakan metode *backward chaining* untuk mendiagnosa kerusakan pada komputer.
4. Menerapkan Bahasa Pemrograman *PHP* dan *MySQL* dalam membangun sebuah halaman *website*.

### **1.6 Manfaat Penelitian**

Manfaat penelitian dapat dibagi menjadi dua aspek yaitu aspek teoritis dan aspek praktis :

## 1. Aspek Teoritis

Secara teoritis penelitian ini dapat disimpulkan sebagai berikut :

- a. Penelitian ini diharapkan dapat membantu pengguna sistem dan menambah pengetahuan dalam merancang bangun sistem informasi untuk perbaikan *hardware* pada komputer.
- b. Penelitian ini diharapkan dapat menjadi bahan acuan untuk penelitian selanjutnya, terutama penelitian yang berkaitan dengan perancangan sistem informasi dibidang teknologi informasi.

## 2. Aspek Praktis

Secara praktis penelitian ini dapat disimpulkan sebagai berikut :

### a. Bagi Instansi.

Sebagai suatu sarana penting bagi teknisi untuk berbagi informasi yang bermanfaat untuk pengguna.

### b. Bagi Penulis.

Dapat lebih memahami tentang pembuatan sistem yang bisa diterapkan langsung pada sebuah halaman *website*.

### c. Bagi Peneliti Lain.

Dapat menambah pengetahuan dan wawasan serta dapat menjadi acuan bagi penelitian selanjutnya.

### d. Bagi Akademik

Diharapkan untuk akademik bisa menambah literatur ataupun *referensi* dan bahan bacaan untuk dipertinggal pada perpustakaan



e. Bagi Masyarakat

Dengan menggunakan metode sistem pakar dan diimplementasikan pada sebuah *website* akan membantu masyarakat yang belum paham tentang komputer, sehingga bisa dilakukan tindakan perbaikan.

## **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)**

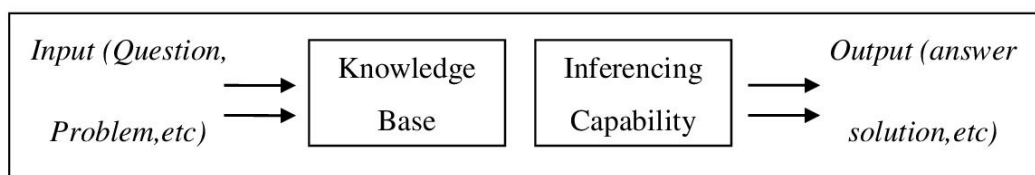
*Artificial Intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin atau komputer dapat melakukan pekerjaan seperti layaknya dan sebaik yang dilakukan oleh manusia. Teknologi Komputer diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu seperti yang dapat dikerjakan 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 *intelligence* 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 diimplementasikan 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* untuk 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 dari 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 untuk *knowledge assistant* (Sutojo, 2011:159-160).

Menurut Turban (2011:160) sistem pakar adalah sebuah *system* yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan ke dalam 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 environment*).

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

<b>Perangkat Lunak Konvensional</b>	<b>Perangkat Lunak Sistem Pakar</b>
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 diidentifikasi 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.

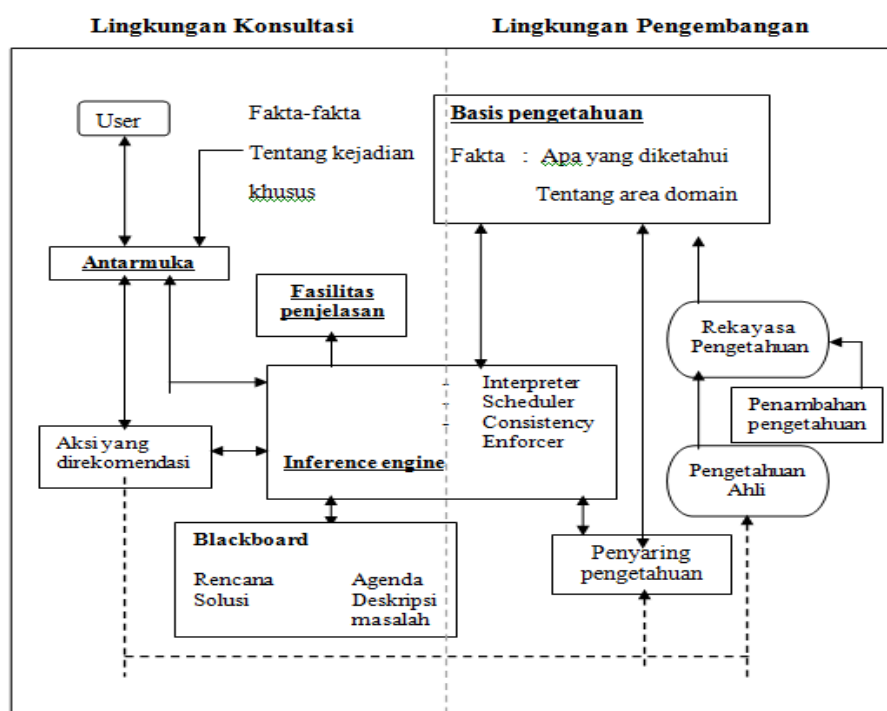
## 6. Pengembangan Sistem (Implementasi tahap lanjut)

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).

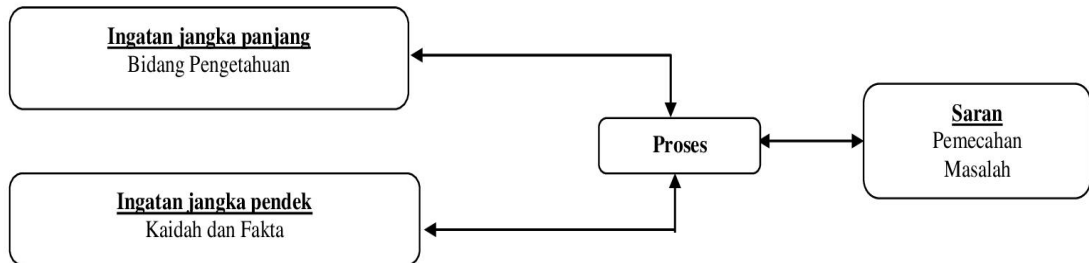
sistem pakar dapat ditampilkan dengan dua lingkungan, yaitu :



**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 pembangunannya 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 pencocokan pola.

Ada banyak cara untuk merepresentasikan pengetahuan, diantaranya adalah logika (*logic*), jaringan semantik (*Semantic Neis*), 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. Peruntutan 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 :

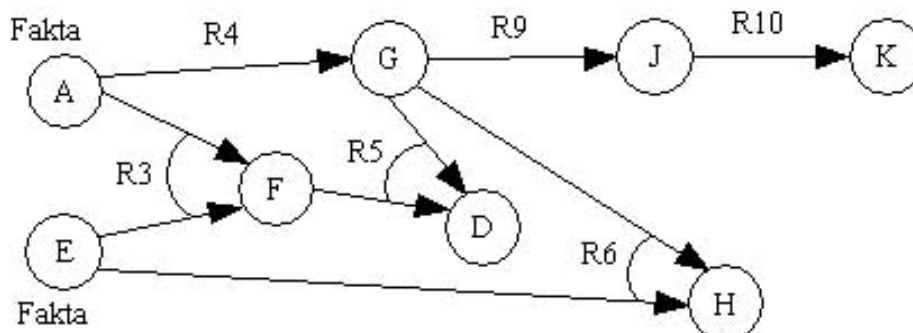
a. Runut Maju (*Forward Chaining*)

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)

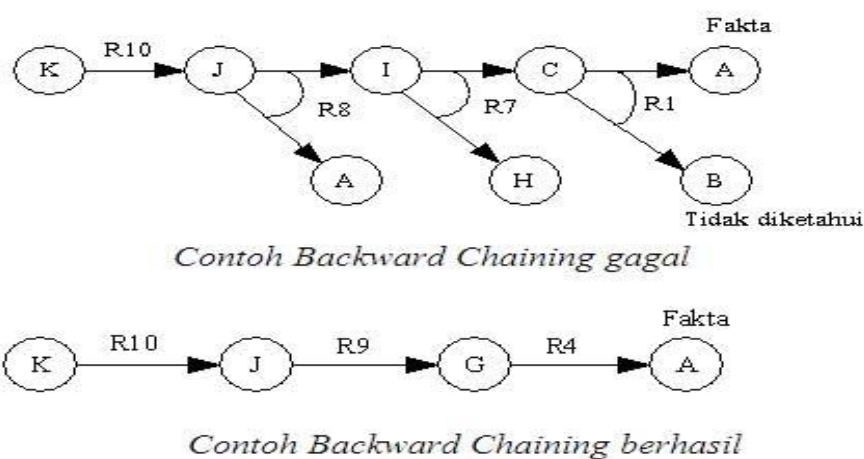
Pada runut maju, system tidak akan melakukan praduga apapun, namun system akan mengecek gejala-gejala tersebut untuk memenuhi konklusi yang sama. Runut maju memulai proses pencarian dengan data sehingga strategi ini disebut juga *data-driven* (Wawan Yunanto, 2007), metode penalaran ini dapat dilihat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

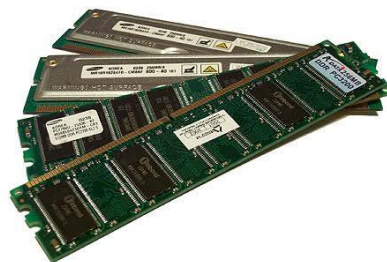
Fungsinya untuk menyuplai arus kepada semua komponen yang tersambung pada motherboard sehingga motherboard dapat berfungsi sebagaimana mestinya dan komputer dapat dioperasikan.



**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 spesifikasi 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 *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

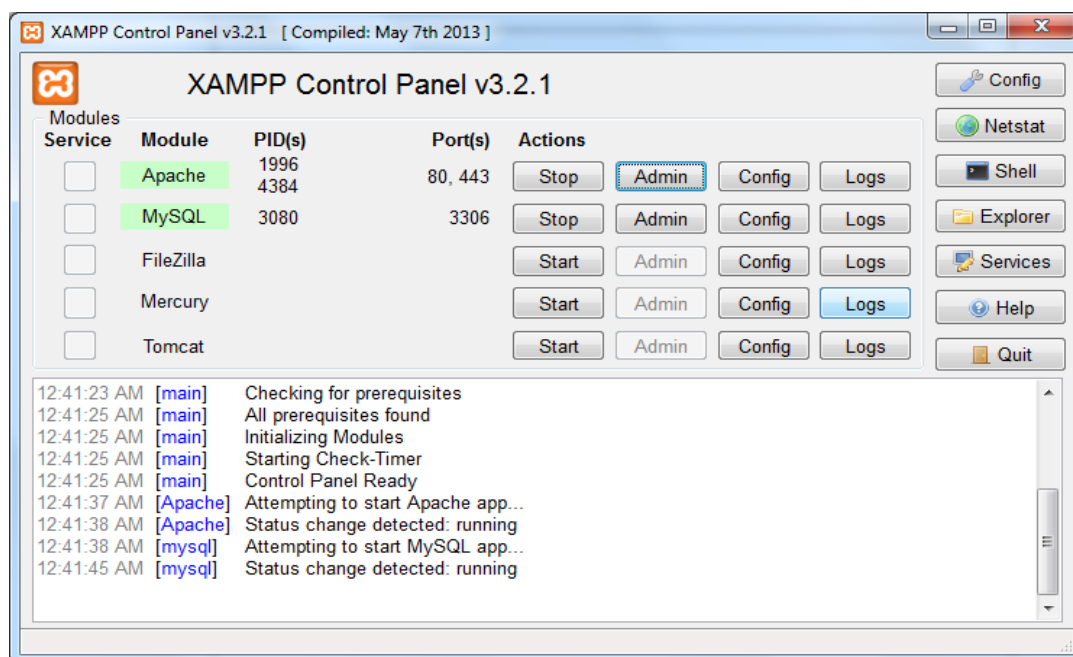
Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999).

*Software* adalah 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 engineering*) adalah ilmu yang mempelajari tentang teknik pembuatan perangkat lunak yang baik dengan pendekatan teknik (*engineering 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 pemrograman *script* yang dirancang untuk membangun aplikasi *web*. Ketika dipanggil dari *web browser*, program yang ditulis dengan PHP akan diparsing 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 (*Active 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 penggunaannya.
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 termasuk 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 berbasis 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 *Associations*, *Generalitiation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

## 5. *Deployment Diagram*


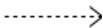

*Deployment Diagram* digunakan untuk menggambarkan detail bagaimana komponen disusun diinfrastruktur sistem.


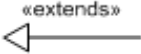




Menurut (A.S & Shalahuddin, 2011) *UML* memiliki diagram grafis untuk membuat suatu model, yaitu :


### 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		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).

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


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


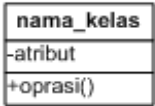



**Tabel 2.2** Simbol Diagram *Use Case*  
**Sumber :** (A.S. & M.Shalahuddin, 2014)

## 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		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).

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







			bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	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		<i>Actifity</i>	Memperlihatkan bagaimana masing-masing kelas antar muka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Actifity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan


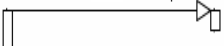
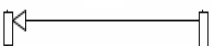
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
---	---	------------------	--

**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

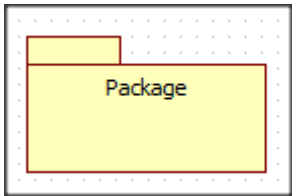
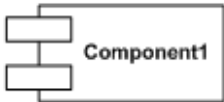

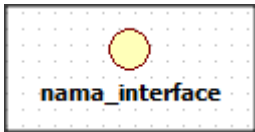
No	Gambar	Nama	Keterangan
1		<i>Life Line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi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

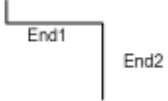
**Sumber :** (A.S. & M.Shalahuddin, 2014)

## 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		<i>Package</i>	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih komponen.
2		Komponen	Komponen system
3		Ketergantungan ( <i>Dependency</i> )	Ketergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4		Antarmuka/ <i>Interface</i>	Sama dengan interface pada pemrograman berbasis objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung

			komponen.
5		<i>Link</i>	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 computer hardware 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 softlens untuk melakukan diagnosa dampak penggunaan softlens berdasarkan gejala yang dialami, dan mengetahui cara penanggulangannya”.

**(Herliana, Setiawan, & Prasetyo, 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.***



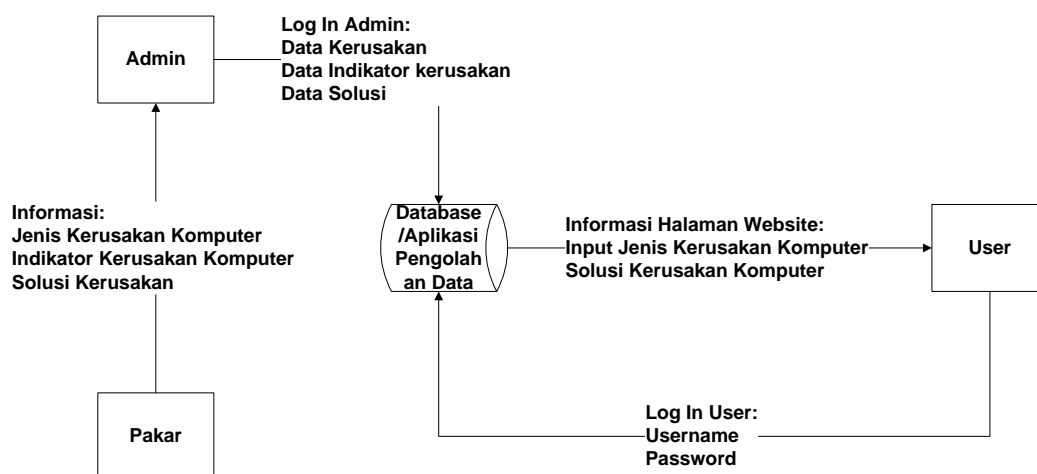
*Usually they will also need to be able to explain, to the users who rely on them, how they have reached their decisions. As an expert is a person who can solve a problem with the domain knowledge. This research paper introduces introduction, parts, application of expert system. and difference between 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*.

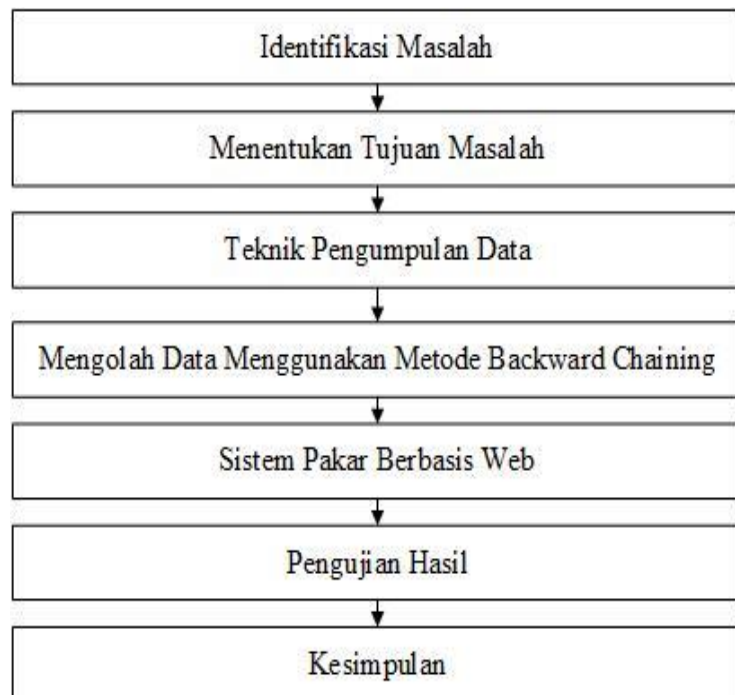


**Gambar 2.12** Kerangka Pemikiran

## BAB III METODE PENELITIAN

### 3.1 Desain Penelitian

Agar penelitian yang dilakukan lebih terarah maka akan digunakan suatu metode. Dalam penelitian ini metode yang digunakan adalah *Backward Chaining*, langkah yang dilakukan dimulai dengan pengumpulan data, pengembangan dan hasil. Tahap identifikasi dan analisis kebutuhan sistem dilakukan untuk mengetahui kebutuhan pengguna terhadap sistem yang akan dikembangkan. Hal ini perlu dilakukan agar sistem yang dikembangkan sesuai dengan kebutuhan pengguna.



**Gambar 3.1** Desain Penelitian  
**Sumber :** Pengolahan data penelitian (2019)

### 1. Identifikasi Masalah

Identifikasi masalah ini dilakukan untuk mengetahui permasalahan ataupun kendala-kendala yang dihadapi oleh masyarakat dalam perbaikan komputer secara umum. Dengan permasalahan yang dihadapi oleh pengguna maka perlu dipersiapkan beberapa masalah dan solusi yang akan diberikan.

### 2. Menentukan tujuan masalah

Dalam penentuan tujuan permasalahan adalah perancang sistem harus mengetahui permasalahan dan keluhan yang dihadapi oleh masyarakat dalam melakukan perbaikan komputer secara umum dan merancang, membangun sistem yang dapat membantu pengguna dalam mengambil tindakan perbaikan komputer secara umum.

### 3. Teknik Pengumpulan Data

Untuk teknik pengumpulan data adalah sebuah teknik yang akan dilakukan dan diberikan untuk mendapatkan informasi ataupun data-data pendukung untuk melakukan perbaikan pada komputer.

### 4. Mengolah data dengan menggunakan Metode *Backward Chaining*

Penentuan metode *Backward Chaining* adalah sebuah metode yang biasa digunakan oleh perancang sistem informasi baik berbasis *web* maupun Android atau dengan aplikasi lainnya. Dengan adanya metode pada sistem pakar, diharapkan dapat membantu perancang dan pengguna dalam menerapkan sistem yang akan dibangun.

## 5. Sistem Pakar Berbasis *Web*

Sistem Pakar berbasis *web* ini bertujuan untuk menerapkan sistem yang akan dirancang, dibangun dengan metode *Backward Chaining* dan diimplementasikan dengan *web*.

## 6. Pengujian Hasil

Tujuan dari pengujian ini adalah untuk melakukan pengujian sistem yang akan dibangun dan akan diuji kelayakannya dengan berbantuan *software XAMPP* dalam pengujian sistem secara *Offline*.

### **3.2 Teknik Pengumpulan Data**

Untuk teknik pengambilan data pada penelitian ini adalah dijabarkan sebagai berikut:

#### 1. Observasi dan wawancara

Proses observasi dan wawancaranya dilakukan untuk memperoleh data dan informasi pakar dibidang perbaikan komputer adalah pada CV Duta Komputer dan untuk pakar dibidang perbaikan adalah Bapak Riced. Wawancara dilakukan dengan mempertanyakan beberapa permasalahan dan kendala yang sering dihadapi oleh masyarakat pada saat menggunakan komputer.

#### 2. Studi Literatur

Studi literature ini dilakukan bertujuan untuk memperoleh perbandingan sumber untuk penguatan baik secara teoritis maupun praktikum. Dengan

proses ini maka peneliti bisa membandingkan beberapa informasi yang didapatkan baik secara *online* maupun berdasarkan buku yang ada.

### 3.3 Operasional Variabel

Definisi operasional merupakan bagian yang mendefinisikan sebuah konsep/variabel agar dapat diukur, dengan cara melihat pada dimensi (indikator) dari suatu konsep/variabel. Dimensi indikator dapat berupa: perilaku, aspek, atau sifat/karakteristik. Dengan demikian, definisi operasional tidak boleh mempunyai makna yang berbeda dengan definisi konseptual. Dengan demikian, definisi operasional bukan berarti definisi/pengertian/makna seperti yang terlihat pada teori dibuku teks, namun lebih menekankan kepada hal-hal yang dapat dijadikan sebagai ukuran/indikator dari suatu variabel dan tidak abstrak (Noor, 2012:97).

**Tabel 3.1** *Operasional Variabel* Kerusakan Pada *hardware* komputer

<b>Variabel</b>	<b>Indikator</b>
Power Supply	Korslet
	Lemah/Overhead
RAM	Overload
	Kotor
Motherboard	Korslet
Hardisk	Bad Sector
	Kabel Lemah
	Piringan/Jarum rusak
Processor	Overhead
	Kotor

**Sumber :** Pengolahan Data Penelitian (2019)

### 3.4 Metode Perancangan Sistem

Perancangan sistem yang dilakukan dalam membangun sistem pakar mendiagnosa kerusakan pada *hardware* komputer menggunakan metode *backward chaining* adalah pengkodean (Nama Kerusakan, Gejala), memberikan aturan (*Rule*), membuat pohon keputusan. Adapun yang menjadi tahapan yang dilakukan sebagai berikut:

#### 3.4.1 Pengkodean

Pada penelitian ini penulis merancang pengkodean untuk nama kerusakan dan gejala kerusakan yang terjadi untuk mempermudah perancangan *database* yang ada pada sistem. Pengkodean tersebut dapat dilihat dalam tabel dibawah ini.

**Tabel 3.2** Kode dan Nama Kerusakan

Variabel	Kode Kerusakan	Indikator
Power Supply	I01	Korslet
	I02	Lemah/Overhead
RAM	I03	Overload
	I04	Kotor
Motherboard	I05	Korslet
Hardisk	I06	Bad Sector
	I07	Kabel Lemah
	I08	Piringan/Jarum rusak
Processor	I09	Overhead
	I10	Kotor

**Sumber :** Pengolahan Data Penelitian (2019)

Pada Tabel 3.2, dapat dilihat bahwa setiap nama kerusakan diwakili dengan kode berdasarkan indikator masing-masing. Kode I01 dan I05 mewakili kerusakan korslet, kode I02 dan I09 mewakili kerusakan *Overhead*, kode I04 dan I10 mewakili *hardware* kotor, kode I06 mewakili kerusakan *Bad Sector*, I07 mewakili kerusakan kabel lemah, I08 mewakili rusaknya piringan atau jarum dan I09 mewakili *Overload*.

Untuk pengkodean gejala kerusakan dapat dilihat pada tabel dibawah ini :

**Tabel 3.3** Kode dan Gejala Kerusakan

Kode Kerusakan	Kode Gejala	Gejala
I01	G01	Bau tidak sedap
	G02	Tidak menyala
I02	G03	Tidak tampil gambar dimonitor
	G04	Menyala sebentar lalu mati
	G05	Cepat panas
	G06	Kipas tidak berputar
I03	G07	Tidak tampil gambar dimonitor
	G08	Bunyi <i>beep</i> beberapa kali
I04	G09	Tidak tampil gambar dimonitor
	G10	Tidak ada bunyi bios indikator



	G11	Lampu indikator menyala tetapi fan prosesor tidak menyala/merespon
I05	G12	Mati total
	G13	Chipset panas
	G14	Tidak tampil gambar dimonitor
	G15	Tidak ada suara <i>alarm</i> saat RAM dilepas
I06	G16	Respon lambat
	G17	Saat booting muncul <i>checkdisk</i>
	G18	Susah untuk diformat (instal ulang OS)
I07	G19	Tidak terdeteksi dibios
	G20	Jika terdeteksi, proses lama
I08	G21	Tidak terdeteksi dibios
	G22	Ada suara pada hardisk
I09	G23	Tidak tampil gambar pada monitor
	G24	Tidak ada suara <i>alarm</i> saat RAM dilepas
I10	G25	Tidak tampil gambar pada monitor

**Sumber** : Pengolahan Data Penelitian (2019)

Pada Tabel 3.3, dapat dilihat bahwa gejala kerusakan diwakili dengan pengkodean G01-G25. Masing-masing kode terdapat gejala yang akan dipergunakan untuk perancangan *database* sehingga akan lebih mudah

diimplementasikan. Setelah pengkodean nama kerusakan dan gejala kerusakan dibuat, maka perlu aturan (*rule*) untuk menggabungkan keduanya. Berikut dapat dilihat aturan (*rule*) dalam tabel dibawah ini.

**Tabel 3.4** Aturan (*Rule*)

<b>Kode Kerusakan</b>	<b>Kode Gejala</b>
I01	G01,G02
I02	G03,G04,G05,G06
I03	G07,G08
I04	G09,G10
I05	G11,G12,G13,G14,G15
I06	G16,G17,G18
I07	G19,G20
I08	G21,G22
I09	G23,G24
I10	G25

**Sumber** : Pengolahan Data Penelitian (2019)

Pada Tabel 3.4 dapat dilihat bahwa kode kerusakan dan kode gejala dipasangkan dengan aturan (*rule*). Untuk mendapatkan hasil atau kode kerusakan yang terjadi, maka kode gejala dipasangkan sesuai dengan gejala kerusakan sebenarnya. Kode kerusakan I01 dipasangkan dengan kode gejala G01 dan G02, kode kerusakan I02 dipasangkan dengan kode gejala G03-G06, kode kerusakan I03 dipasangkan dengan kode gejala G07 dan G08, kode kerusakan I04

dipasangkan dengan kode gejala G09 dan G10, kode kerusakan I05 dipasangkan dengan kode gejala G11-G15, kode kerusakan I06 dipasangkan dengan kode gejala G16-G18, kode kerusakan I07 dipasangkan dengan kode gejala G19 dan G20, kode kerusakan I08 dipasangkan dengan kode gejala G21 dan G22, kode kerusakan I09 dipasangkan dengan kode gejala G23 dan G24 sedangkan kode kerusakan I10 dipasangkan dengan kode gejala G25.

### 3.4.2 Aturan (*Rule*)

Dalam penelitian ini pengetahuan direpresentasikan dengan menggunakan aturan berbentuk: *IF-THEN*. Berikut dapat dilihat *rule* yang berkaitan dengan variabel yang digunakan dalam tabel dibawah ini.

**Tabel 3.5** Aturan (*Rule*) Kerusakan Dengan Gejala

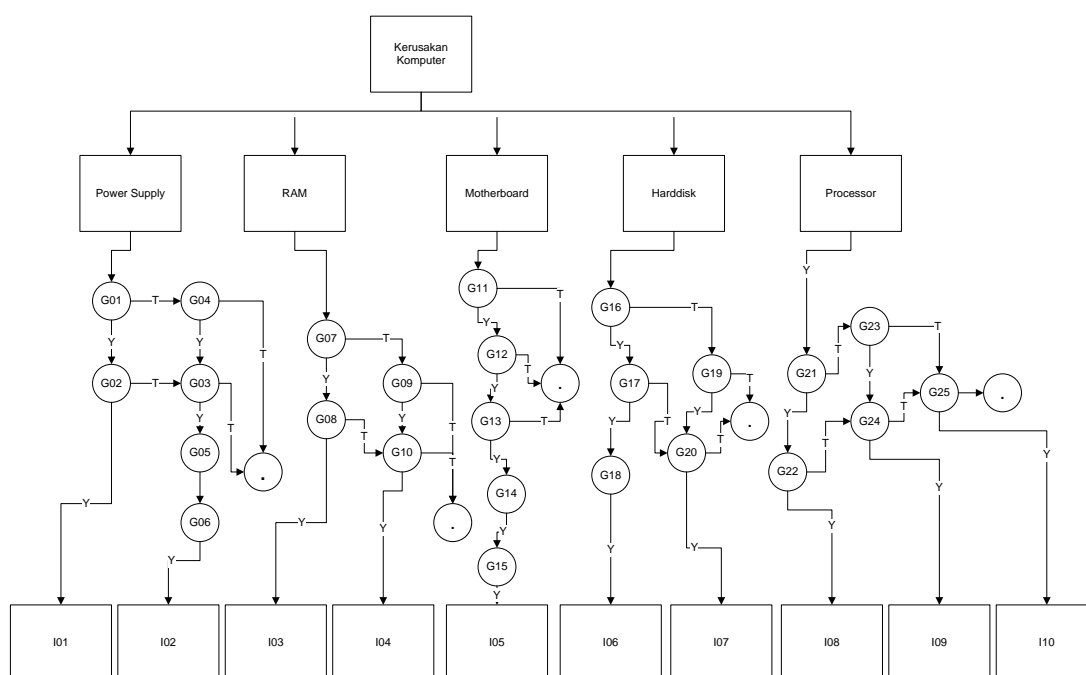
No	Aturan ( <i>Rule</i> )
1	<i>IF G01 AND G02 TRUE THEN I01 ELSE G03</i>
2	<i>IF G03,G04,G05 AND G06 TRUE THEN I02 ELSE G07</i>
3	<i>IF G07 AND G08 TRUE THEN I03 ELSE G9</i>
4	<i>IF G9 AND G10 TRUE THEN I04 ELSE G11</i>
5	<i>IF G11,G12,G13,G14 AND G15 TRUE THEN I05 ELSE G16</i>
6	<i>IF G16,G17 AND G18 TRUE THEN I06 ELSE G19</i>
7	<i>IF G19 AND G20 TRUE THEN I07 ELSE G21</i>
8	<i>IF G21 AND G22 TRUE THEN I08 ELSE G23</i>
9	<i>IF G23 AND G24 TRUE THEN I09 ELSE G25</i>
10	<i>IF G25 TRUE THEN I10</i>

**Sumber** : Pengolahan Data Penelitian (2019)

Pada Tabel 3.4 dapat dilihat bahwa tujuan dari aturan ini sebagai prosedur pemecahan masalah kerusakan. Perlunya mengurutkan langkah-langkah pada setiap permasalahan kerusakan pada tiap *variable hardware* komputer adalah untuk mendapatkan solusi. Jika aturan *IF THEN* bernilai *True*, maka akan menghasilkan solusi dari masing-masing kerusakan sesuai pada tabel yang telah dibuat

### 3.4.3 Pohon Keputusan

Dalam penelitian ini penulis merancang pohon keputusan berdasarkan aturan *rule* dan fakta-fakta yang ada. Adapun pohon keputusan tersebut adalah :



**Gambar 3.2** Pohon Keputusan  
**Sumber :** Pengolahan Data Penelitian (2019)

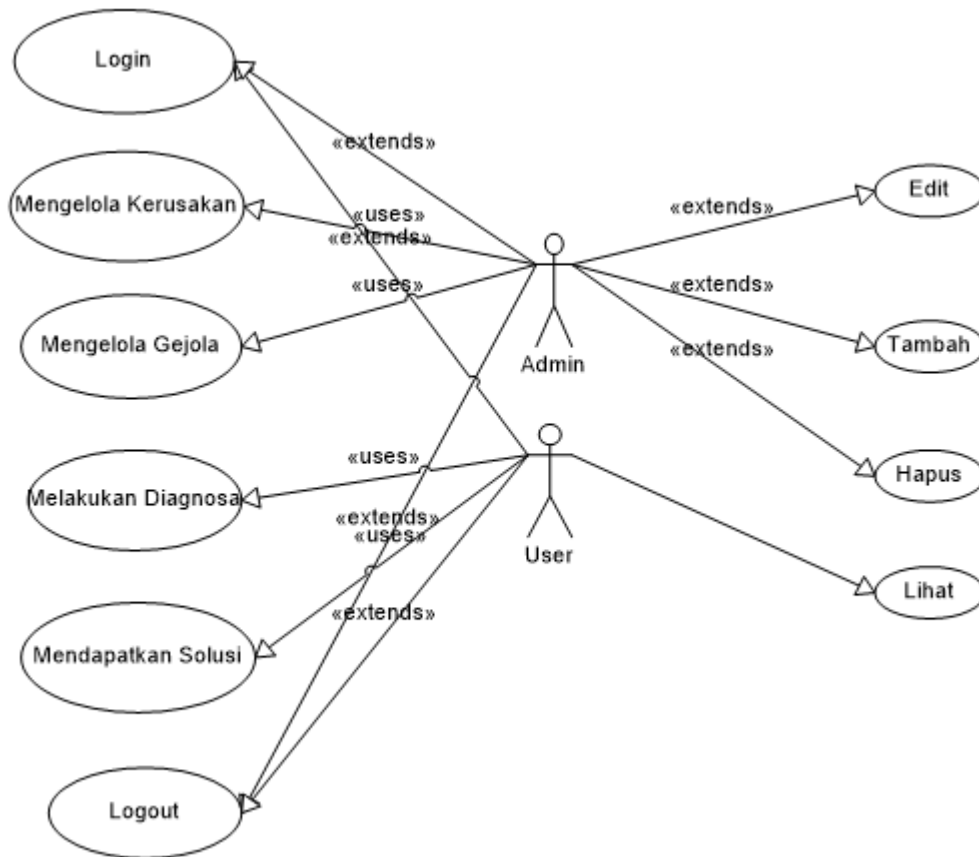
Pada Gambar 3.2 dapat dilihat bahwa pada pohon keputusan menjelaskan alur dari pengetahuan pakar yang akan diimplementasikan ke dalam *database*. Mulai dari G01, G03, G07, G09, G11, G16, G19, G21, G23 dan G 25 menjawab

Tidak maka hasilnya tidak terdiagnosa atau dengan tanda \* dan dilanjutkan pada pertanyaan gejala kerusakan yang lainnya.

Sebagaimana proses pembuatan sebuah perangkat lunak, pembuatan sebuah sistem informasi juga memerlukan beberapa buah tahapan. Tahapan-tahapan tersebut dimulai dari desain, perancangan, hingga implementasi dan pengujian. Pembuatan sebuah sistem informasi diawali dengan kebutuhan pengguna berdasarkan permasalahan yang terjadi. Berdasarkan sudut pandang keilmuan informatika, langkah-langkah tersebut termasuk dalam kajian penelitian. *UML (Unified Modelling Language)* adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemrograman berorientasi objek (Pratama, 2014:27).

#### **3.4.4 Use Case Diagram**

*Use Case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sistem dan merepresentasikan interaksi antara aktor dengan sistem.



**Gambar 3.3** *Use Case Diagram*  
**Sumber :** Pengolahan Data Penelitian (2019)

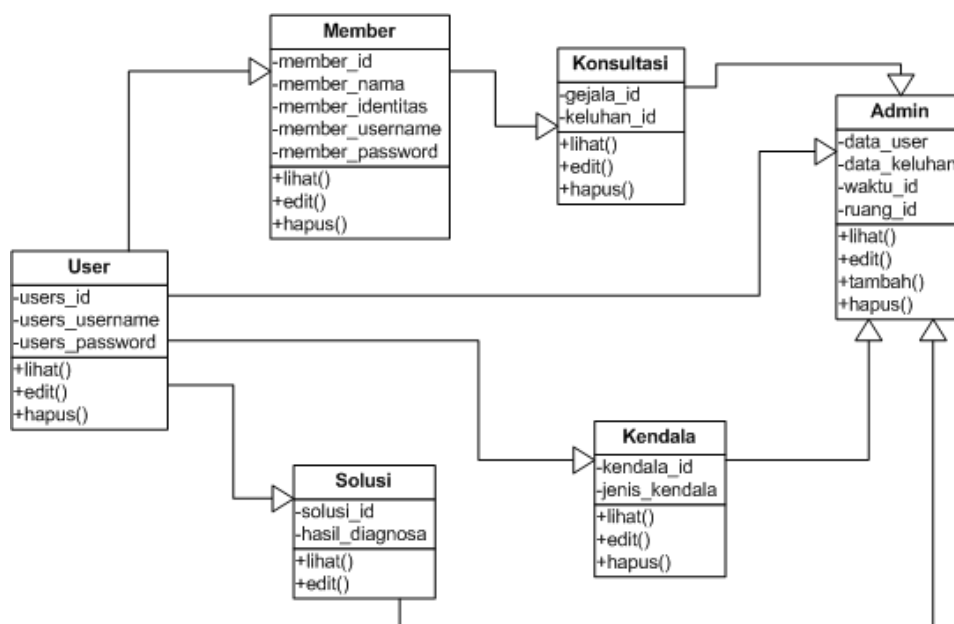
Penjelasan dari Gambar 3.3 *Use Case Diagram*, aplikasi sistem pakar sebagai berikut:

1. Pakar adalah aktor yang menjadi sumber pengetahuan kerusakan mesin *hardware* komputer, dan juga menjadi *administrator* dari sistem aplikasi yang dibangun.
2. *User* adalah aktor yang menggunakan aplikasi sistem pakar.
3. *Login* adalah pintu masuk untuk pakar sebagai *administrator*.
4. Isi data adalah pendaftaran pengguna untuk menjadi *user* dari aplikasi sistem pakar.

5. *Logout* adalah pintu keluar untuk pakar sebagai *administrator*.
6. Mengelola Diagnosa adalah pengetahuan pakar mendiagnosa kerusakan *hardware* komputer.
7. Mengelola Gejala adalah gejala-gejala dari setiap jenis kerusakan *hardware* komputer.
8. Mengelola Solusi adalah solusi atas kerusakan *hardware* komputer.
9. Melakukan Diagnosa adalah pengguna menggunakan fungsi diagnosa kerusakan *hardware* komputer.
10. Mendapatkan Solusi adalah hasil dari diagnosa kerusakan *hardware* komputer.

### 3.4.5 Class Diagram

*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat atribut dan membangun sistem.



**Gambar 3.4** *Class Diagram*  
**Sumber :** Pengolahan Data Penelitian (2019)

Penjelasan dari Gambar 3.4 *Class diagram*, aplikasi sistem pakar sebagai berikut:

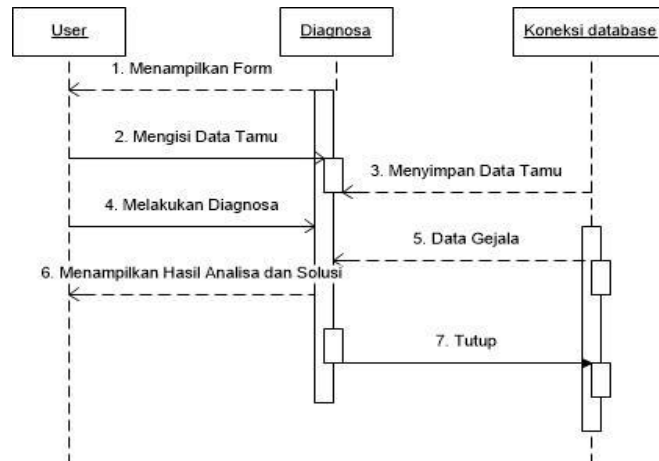
1. *Login* adalah kelas proses yang diambil dari pendefinisian *use case login* untuk pintu masuk pakar *administrator* ke sistem aplikasi pakar.
2. *Tamu* adalah kelas proses yang diambil dari pendefinisian *use case* untuk aktor yang menggunakan aplikasi sistem pakar.
3. *Kelola Diagnosa* adalah kelas proses yang diambil dari pendefinisian *use case* mengelola diagnosa yang didalamnya menangani proses gejala dan proses solusi.
4. *Kelola Gejala* adalah kelas proses yang diambil dari pendefinisian *use case* mengelola gejala yang didalamnya menangani proses pertanyaan gejala untuk setiap jenis kerusakan.
5. *Kelola Solusi* adalah kelas proses yang diambil dari pendefinisian *use case* mengelola solusi yang didalamnya menangani proses kesimpulan atas proses mengelola diagnosa dan proses mengelola gejala.
6. *Pakar* adalah kelas data yang digunakan untuk memproses segala pengaksesan terhadap proses mengelola diagnosa, proses mengelola gejala, dan proses mengelola solusi.
7. *Koneksi database* adalah kelas utilitas untuk koneksi ke *database*.

#### **3.4.6 Sequence Diagram**

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



## 1. *Sequence Diagram User*

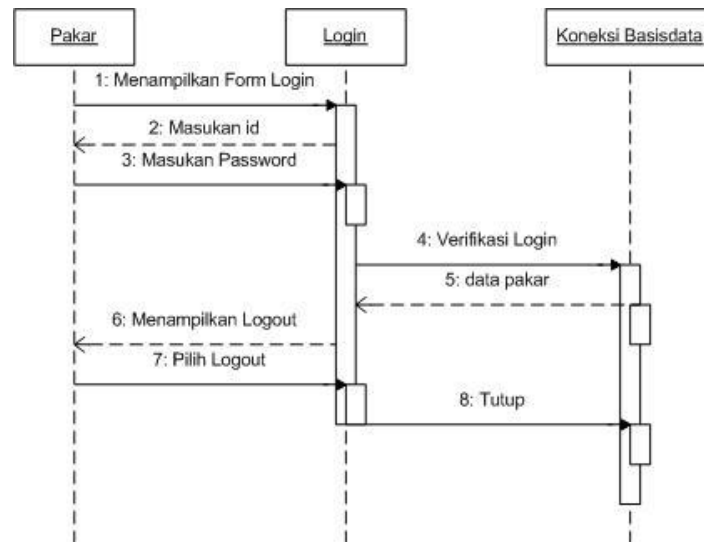


**Gambar 3.5** *Sequence Diagram Login User*  
 Sumber : Pengolahan Data Penelitian (2019)

Penjelasan dari Gambar 3.5 *Sequence Diagram User*, aplikasi sistem pakar sebagai berikut:

1. Untuk melakukan diagnosa *user* akan mengklik menu diagnosa pada sistem aplikasi pakar dan seterusnya akan tampil form untuk diisi.
2. *Diagnosa* menampilkan *form* yang berisi nama, *hardware* komputer dan *type*. *User* akan mengisi kolom hardware komputer dan *type* berupa angka atau huruf serta mengisi nama pengguna pada kolom nama.
3. Setelah tombol mulai diklik, maka data *User* akan disimpan ke dalam *database*.
4. Proses diagnosa akan dijalankan dengan menjawab pertanyaan.
5. Pertanyaan diagnosa akan muncul satu persatu.
6. Menampilkan hasil diagnosa berdasarkan jawaban pertanyaan.
7. Proses *user* selesai.

## 2. *Sequence Diagram Login Pakar*



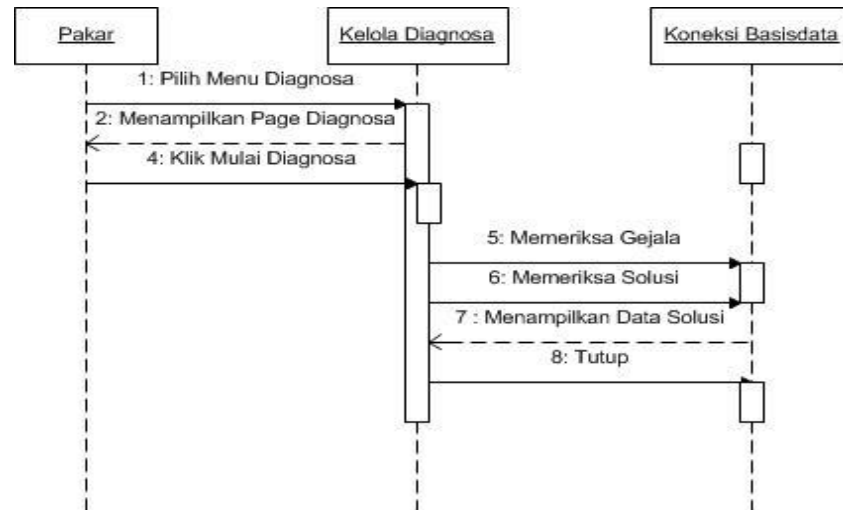
**Gambar 3.6** *Sequence Diagram Login Pakar*  
**Sumber :** Pengolahan Data Penelitian (2019)

Penjelasan dari Gambar 3.6 *Sequence Diagram Login Pakar*, aplikasi sistem pakar sebagai berikut:

1. Pakar memasukan *link* ke *address bar* pada *browser* untuk masuk kedalam *form login* sistem aplikasi pakar.
2. *Login* menampilkan *form* yang berisi *username* dan *password*.
3. Pakar memasukan *username* yang berupa angka atau huruf.
4. Pakar memasukan *password* yang berupa angka atau huruf.
5. Setelah tombol *login* diklik, maka data Pakar diverifikasi kedalam *database*.
6. Jika data *input* benar maka *login* pakar berhasil masuk ke dalam sistem.
7. Tombol *logout* tampil setelah berhasil *login*.
8. Jika ingin keluar dari sistem aplikasi maka silahkan klik tombol *logout*.

9. Proses *login* selesai.

### 3. *Sequence Diagram* Diagnosa

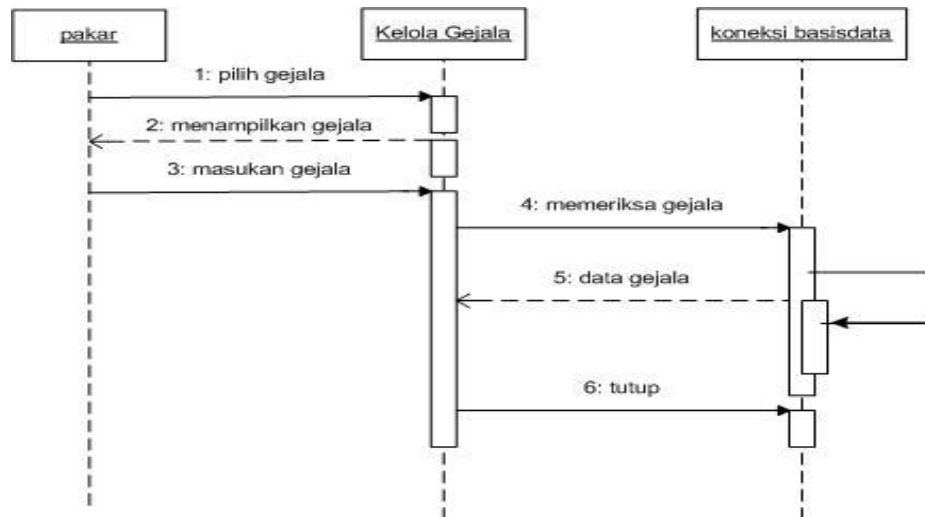


**Gambar 3.7** *Sequence Diagram* Diagnosa  
**Sumber** : Pengolahan Data Penelitian (2019)

Penjelasan dari Gambar 3.7 *Sequence Diagram* Diagnosa, aplikasi sistem pakar sebagai berikut:

1. Pakar memilih *menu* diagnosa untuk mendiagnosa kerusakan.
2. Mengelola Diagnosa terkoneksi ke *database*.
3. Mengelola Diagnosa menampilkan halaman diagnosa.
4. Pakar memulai diagnosa.
5. Mengelola Diagnosa memeriksa gejala dengan mengajukan pertanyaan.
6. Mengelola Diagnosa memeriksa solusi berdasarkan jawaban atas pertanyaan gejala kerusakan.
7. Koneksi basisdata menampilkan solusi kerusakan.
8. Proses Mengelola Diagnosa selesai.

#### 4. *Sequence Diagram* Mengelola Gejala

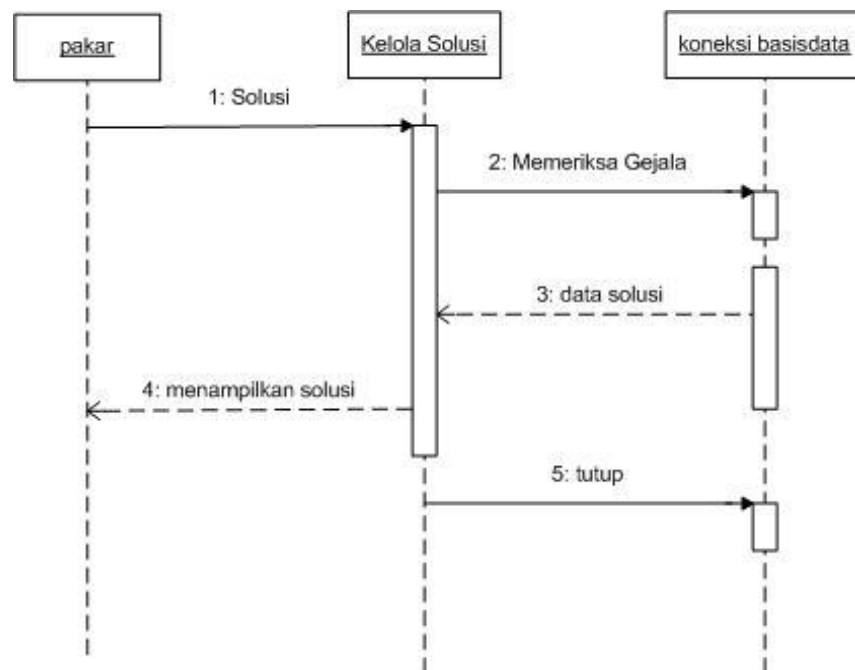


**Gambar 3.8** *Sequence Diagram* Mengelola Gejala  
**Sumber** : Pengolahan Data Penelitian (2019)

Penjelasan dari Gambar 3.8 *Sequence Diagram* Mengelola Gejala, aplikasi sistem pakar sebagai berikut:

1. Pakar menjawab/memilih gejala kerusakan.
2. Mengelola Gejala menampilkan pertanyaan gejala sesuai dengan jenis kerusakan.
3. Pakar memasukan/memilih gejala kerusakan.
4. Mengelola Gejala memeriksa setiap jawaban dari gejala terpilih.
5. Mengelola Gejala memproses dan menampilkan gejala.
6. Proses Mengelola Gejala selesai.

## 5. *Sequence Diagram Mengelola Solusi*



**Gambar 3.9** *Sequence Diagram Mengelola Solusi*  
**Sumber :** Pengolahan Data Penelitian (2019)

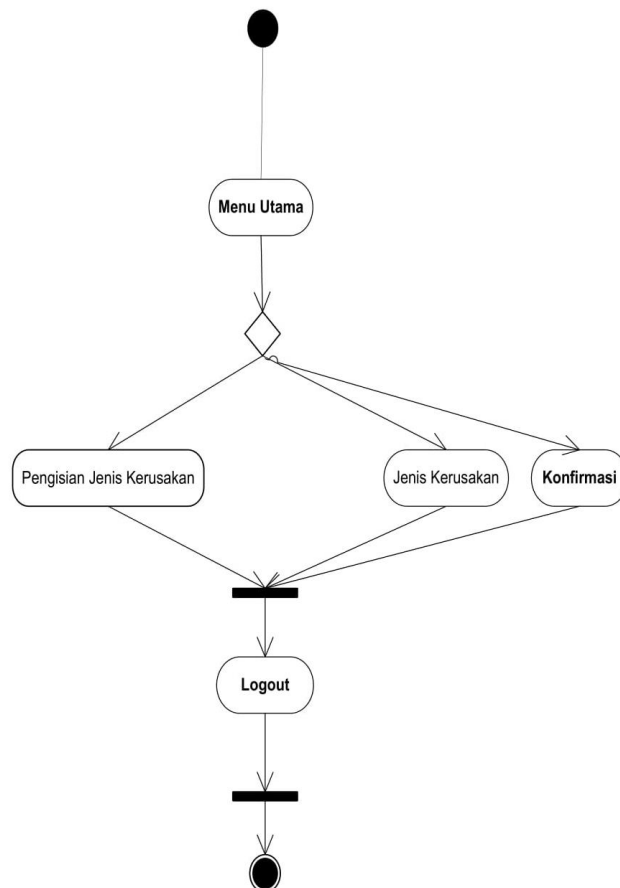
Penjelasan dari Gambar 3.9 *Sequence Diagram Mengelola Solusi*, sebagai berikut:

1. Pakar mengikuti proses untuk mendapatkan solusi.
2. Mengelola Solusi memilih gejala yang sudah dipilih pakar.
3. Mengelola Solusi memeriksa hasil pertanyaan gejala sudah dipilih.
4. Mengelola Solusi memproses analisa dari *database*.
5. Mengelola Solusi menampilkan solusi kerusakan.
6. Proses Mengelola Solusi selesai.

### 3.4.7 Activity Diagram

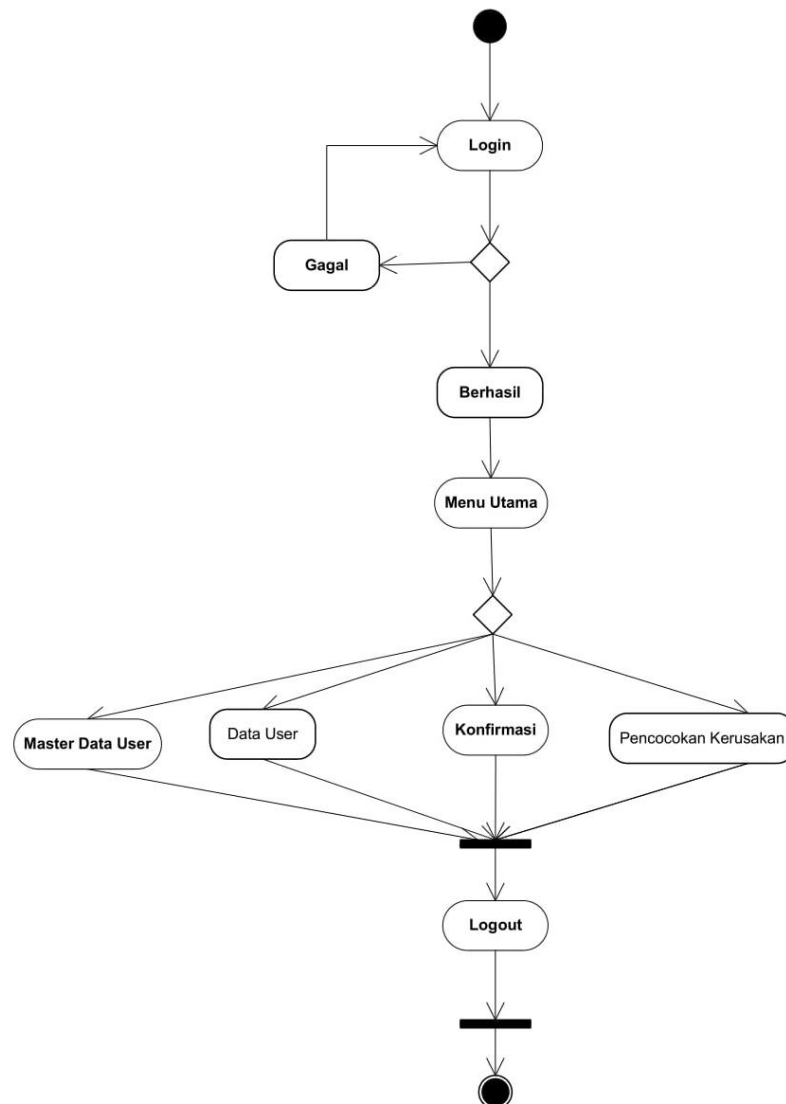
Berikut ini adalah diagram aktifitas yang menunjukkan fungsi *admin* dan *user* pada sistem informasi kerusakan *hardware* pada komputer.

#### 1. Activity Diagram User



**Gambar 3.10** Activity Diagram User Kerusakan Hardware Komputer  
Sumber : Pengolahan Data Penelitian (2019)

## 2. Activity Diagram Admin



**Gambar 3.11** Activity Diagram Admin Kerusakan Hardware Komputer  
**Sumber :** Pengolahan Data Penelitian (2019)

### 3.4.8 Perancangan Database

Rancangan *database* menggambarkan bagaimana rancangan didalam sebuah *database* sistem sehingga sistem tersebut bisa berjalan sesuai dengan rancangan dan implementasi.

### 1. *Database Gejala*

**Tabel 3.6** *Database Gejala*

<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<b>Keterangan</b>
Id_indikator	Int	11	Default Not null
Keterangan_gejala	Varchart	100	Default Not null
Pertanyaan	Varchar	100	Default Not null

### 2. *Database Indikator*

**Tabel 3.7** *Database Indikator*

<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<b>Keterangan</b>
Id_variabel	Int	100	Default Null
Ket_indikator	Varchar	100	Not Null

### 3. *Database Periksa*

**Tabel 3.8** *Database Periksa*

<i>Field Name</i>	<i>Type</i>	<i>Size</i>	<b>Keterangan</b>
Id	Int	11	Not Null
Nama	Varchar	100	Default Null
Gejala	Int	11	Not Null

#### 3.4.9 Perancangan Antar Muka (*Interface*)

Perancangan antarmuka ini bertujuan untuk memberikan gambaran mengenai bentuk antarmuka dari perangkat lunak yang akan digunakan oleh *user* untuk berinteraksi dengan perangkat lunak. Rancangan antar muka ini



mempertimbangkan berbagai kemudahan dan fungsionalitas dari perangkat lunak itu sendiri.

Dlagnosa	Header
Opsi 1	Halaman Konsultasi
Opsi 2	
Opsi 3	

**Gambar 3.12** *Interface* Menu Beranda  
**Sumber** : Pengolahan Data Penelitian (2019)

### 3.5 Lokasi dan Jadwal Penelitian

#### 3.5.1 Lokasi Penelitian

Dalam melakukan penelitian ini penulis meneliti dikota Batam. Penelitian ini dilakukan berdasarkan data-data yang didapatkan dari pihak terkait dengan penelitian ini.

#### 3.5.2 Jadwal Penelitian

Jadwal penelitian untuk memperoleh data dan informasi dilaksanakan pada bulan September 2019 sampai dengan Februari 2020 Sedangkan waktu penelitian ini disesuaikan dengan waktu senggang pembelajaran atau jam tertentu. Berikut jadwal penelitian selengkapnya.

Kegiatan	Jadwal Penelitian																							
	September 2019				Oktober 2019				November 2019				Desember 2019				Januari 2020				Februari 2020			
	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
Pengajuan Judul			■	■																				
Penyusunan Bab I				■	■	■	■																	
Penyusunan Bab II						■	■	■	■	■	■													
Penyusunan Bab III											■	■	■	■	■									
Penyusunan Bab IV														■	■	■	■	■						
Penyusunan Bab V, daftar pustaka, lampiran																		■	■	■	■	■	■	

