

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1. Teori Dasar**

Dalam penulisan skripsi ini, diambil judul “Implementasi *Forward Chaining* Dalam Mendiagnosa Kerusakan Pada *Internet Protocol Camera*”. Maka dari itu, diperlukan beberapa teori umum yang berhubungan dengan sistem pakar dan metode *forward chaining* yang bertujuan untuk mendukung proses perancangan sistem tersebut. Teori-teori yang menjadi dasar penulisan skripsi sebagai berikut:

##### **2.1.1. Kecerdasan Buatan (*Artificial Intelligence*)**

Selama bertahun tahun para filsuf mempelajari kecerdasan yang dimiliki manusia, dan dari pemikiran tersebut lahirlah kecerdasan buatan sebagai ilmu yang berusaha mempelajari dan meniru kecerdasan manusia. Seiring dengan berkembangnya zaman, maka peranan komputer semakin mendominasi kehidupan manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, namun komputer diharapkan dapat digunakan untuk mengerjakan segala sesuatu yang bisa dikerjakan manusia.

Kecerdasan buatan berasal dari kata *Artificial Intelligence* yang mengandung arti tiruan atau kecerdasan, secara harfiah kecerdasan buatan adalah salah satu bidang dalam ilmu komputer yang membuat komputer agar

dapat bertindak dan sebaik seperti manusia (menirukan kerja otak manusia) (Octavina & Fadlil, 2014).

Pada 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.

Menurut Verina, kecerdasan buatan atau *Artificial Intellengence* (AI) dapat didefinisikan sebagai sub bidang pengetahuan komputer yang khusus ditujukan untuk membuat software dan hardware yang sepenuhnya biasa menirukan beberapa fungsi otak manusia. Karena itu diharapkan komputer bisa membantu manusia didalam berbagai masalah yang sangat rumit (Verina, 2015).

#### **2.1.1.1. Sistem Pakar**

Saat ini permasalahan-permasalahan yang dihadapi masyarakat semakin banyak. Termasuk di bidang teknologi yang semakin hari semakin berkembang membuat masyarakat sulit memecahkan masalah. Dengan adanya sistem pakar, maka permasalahan yang dihadapi asyarkat akan mudah diselesaikan.

Sistem pakar merupakan cabang dari AI (*Artificial Inteligent*) yang membuat ekstensi untuk spesialisasi pengetahuan guna memecahkan suatu permasalahan pada *Human Expert*. *Human Expert* merupakan seseorang ahli dalam suatu bidang ilmu pengetahuan tertentu, berarti *expert* memiliki suatu

permasalahan yang tidak dapat dipecahkan oleh orang lain secara efisien (Supartini, 2016). Sistem pakar meniru cara berpikir seorang pakar dalam menyelesaikan sebuah permasalahan, membuat keputusan maupun mengambil kesimpulan suatu fakta. Menurut Verna, sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut (Verina, 2015).

Menurut Tamin, Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. (Tamin, 2015). *An expert system is a model and procedures relating, in a particular domain, which can be compared with the level of expertise the expertise of an expert* (Widiyanto, Surarso, & Dwi, 2017).

#### **2.1.1.2. Logika Fuzzy**

Logika *fuzzy* sering dimanfaatkan untuk sistem pendukung keputusan atau sistem rekomendasi. Biasanya logika *fuzzy* akan membuat sebuah keputusan antara benar dan salah.

*Fuzzy Logic is the enhancement of Boolean logic dealing with the concept of partial truth. Where classical logic to claim that everything can be expressed in terms of binary (0 or 1, black or white, yes or no), fuzzy logic replaces Boolean truth with a level of truth* (Gede & Divayana, 2014).

Berikut ini adalah alasan digunakan logika fuzzy antara lain:

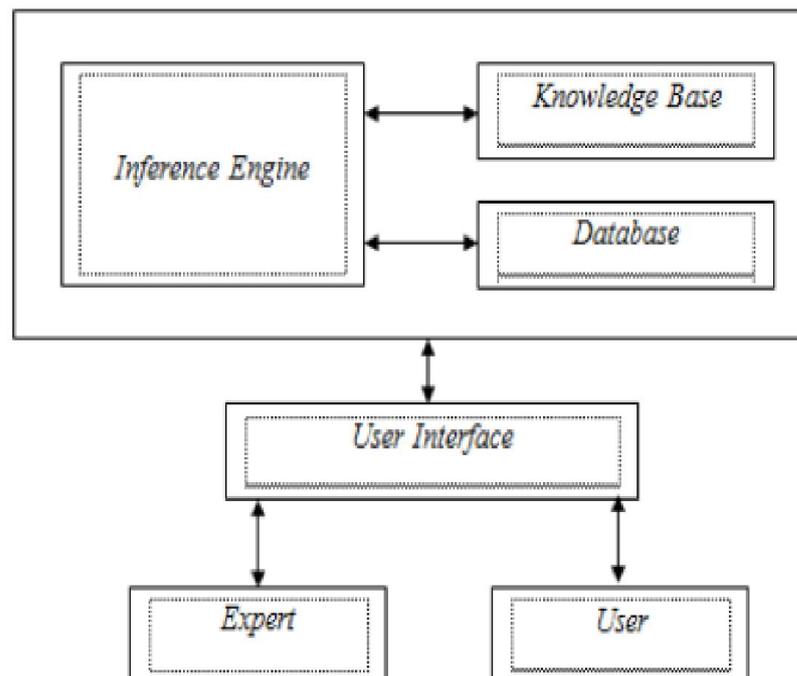
1. Konsep logika *fuzzy* mudah dimengerti, karena logika *fuzzy* menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran *fuzzy* tersebut cukup mudah untuk dimengerti
2. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan dan ketidakpastian yang menyertai permasalahan.
3. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat
4. Logika *fuzzy* didasarkan pada bahasa alami, menggunakan bahasa sehari-hari sehingga mudah dimengerti (Mutammimul Ula, 2014).

#### **2.1.1.3. Jaringan Saraf Tiruan (*Artificial Neural Network*)**

Jaringan saraf tiruan telah banyak dikembangkan oleh para peneliti dewasa ini. Beragam metode pembelajaran untuk jaringan saraf tiruan juga terus dikembangkan, seperti *delta learning rule*, *kohonen self-organizing map*, dan *back-propagation*.

Jaringan Syaraf Tiruan telah dikembangkan sejak tahun 1940. Pada tahun 1943 McCulloch dan W.H.Pitts memperkenalkan pemodelan matematis *neuron*. Tahun 1949, Hebb mencoba mengkaji proses belajar yang dilakukan oleh *neuron*. Teori ini dikenal sebagai Hebbian Law. Tahun 1958, Rosenblatt memperkenalkan konsep *perseptron* suatu jaringan yang terdiri dari beberapa lapisan yang saling berhubungan melalui umpan maju (*feed foward*) (Yanto, Defit, & Nurcahyo, 2015).

Jaringan saraf tiruan merupakan salah satu representasi ‘buatan’ dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Sedangkan saraf itu sendiri merupakan sel-sel yang memproses informasi pada otak dan membentuk sebuah jaringan sebagai suatu kesatuan yang menghubungkan saraf tersebut. Istilah ‘buatan’ yang digunakan pada jaringan saraf tiruan diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Jaringan saraf tiruan memodelkan suatu struktur yang mirip dengan otak manusia (Hendri, 2014).

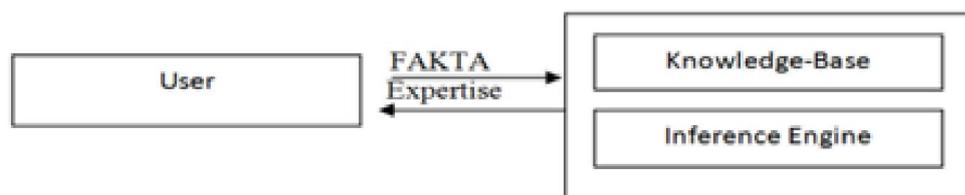


**Gambar 2.1** Struktur Sistem Pakar

**Sumber:** (Yanto et al., 2015)

### 2.1.2. Sistem Pakar

Sistem pakar merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang membuat penggunaan pengetahuan yang dikhususkan secara ekstensif untuk memecahkan masalah pada tingkatan manusia *expert*. Pakar adalah seseorang yang mempunyai keahlian dalam bidang tertentu. Pakar mempunyai pengetahuan atau ketrampilan tertentu yang tidak diketahui atau ada untuk kebanyakan orang. Teknologi sistem pakar ini mungkin mencakup bahasa sistem pakar tertentu/khusus, program, dan *hardware* yang ditentukan untuk tambahan dalam pengembangan dan pembuatan sistem pakar. (Tarigan, 2013)



**Gambar 2.2** Konsep Dasar dari Fungsi Sistem Pakar

Sumber: (Tarigan, 2013)

Gambar 2.2 mengilustrasikan konsep pokok dari sistem pakar yang berdasarkan pada pengetahuan. *User* men-supply fakta atau informasi lainnya ke sistem pakar dan menerima nasihat *expert* atau pakai sebagai jawabannya

Secara *internal*, sistem pakar berisi dua komponen pokok. Yang berdasarkan pengetahuan berisi pengetahuan yang "*inference engine*" menggambarkan kesimpulan. Kesimpulan ini adalah respon sistem pakar atas permintaan pemakai untuk *expertise*. Sistem yang berdasarkan pengetahuan juga dirancang untuk beraksi sebagai asisten *intelligent* pada pakar manusia. *Assistant*

*intelligent* ini dirancang dengan teknologi sistem pakar karena banyak pengetahuan ditambahkan dan akan beraksi lebih seperti pakar.

#### **2.1.2.1. Konsep Dasar Sistem Pakar**

Menurut Jaenal Arifin, Konsep dasar sistem pakar mengandung : keahlian, ahli, pengalihan keahlian, inferensi, aturan, dan kemampuan menjelaskan. Keahlian adalah suatu kelebihan penguasaan pengetahuan di bidang tertentu yang diperoleh dari pelatihan, membaca atau pengalaman. Bentuk pengetahuan yang dapat digolongkan sebagai keahlian adalah :

1. Fakta-fakta pada lingkup permasalahan tertentu.
2. Teori-teori pada lingkup permasalahan tertentu.
3. Prosedur-prosedur dan aturan yang berkaitan dengan permasalahan tertentu.
4. Strategi-strategi tertentu untuk menyelesaikan masalah.

Ahli adalah seseorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan, menyusun kembali pengetahuan jika dipandang perlu, memecahkan aturan-aturan jika dibutuhkan, dan menentukan relevan tidaknya keahlian mereka.

Keahlian adalah pengalihan keahlian dari para pakar ke komputer (sistem) untuk kemudian dialihkan lagi ke orang lain yang membutuhkan baik orang awam maupun untuk para pakar sebagai asistennya.

Salah satu fitur yang harus dimiliki oleh sistem pakar adalah inferensi, yaitu kemampuan sistem pakar untuk menalar, membuat kesimpulan dan membuat rekomendasi. Inferensi dapat dilakukan karena adanya basis

pengetahuan (fakta dan prosedur serta aturan-aturan tertentu). Fitur lainnya dari sistem pakar adalah kemampuan untuk merekomendasi. Kemampuan inilah yang membedakan sistem pakar dengan sistem konvensional (Arifin, 2016).

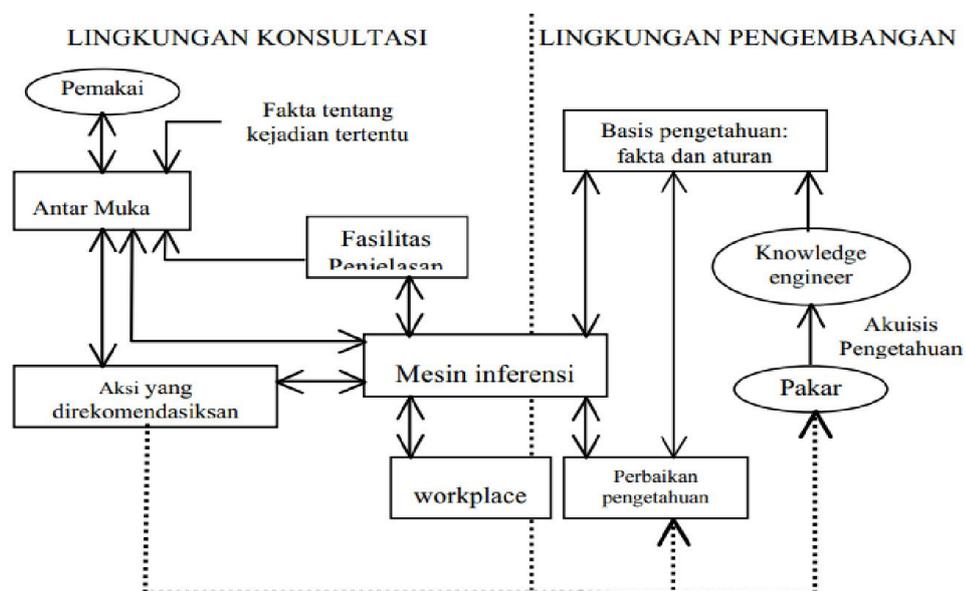
#### **2.1.2.2. Karakteristik Sistem Pakar**

Sistem pakar mempunyai beberapa karakteristik dasar yang membedakan dengan program komputer biasa umumnya :

1. Mempunyai kepakaran, dalam menyelesaikan masalah bukan hanya mendapatkan solusi yang benar saja, namun juga bagaimana mendapatkan pemecahan dengan cepat dan mahir.
2. Domain tertentu, Sistem pakar mengutamakan kedalaman mengenai bidang tertentu.
3. Memiliki kemampuan mengolah data yang mengandung ketidakpastian. Kadang-kadang data yang tersedia tidak lengkap sistem harus dapat memberikan pemecahan sesuai data yang tersedia dengan memberikan pertimbangan, saran atau anjuran sesuai dengan kondisi yang ada.
4. Dirancang untuk dapat dikembangkan secara bertahap program komputer dirancang untuk memberikan jawaban yang tepat setiap waktu. Sedangkan sistem pakar dirancang untuk berlaku sebagai seorang pakar, kadang memberikan jawaban yang benar, dan suatu saat mungkin tidak tepat (*Expert system makes mistake*)(Arifin, 2016).

### 2.1.2.3. Arsitektur Sistem Pakar

Arsitektur sistem pakar dapat dilihat pada gambar 2.3 di bawah ini, terdapat dua bagian utama sistem pakar yaitu lingkungan pengembangan (*development enviroment*) yang digunakan untuk memasukkan pengetahuanpengetahuan pakar kedalam lingkungan sistem pakar dan lingkungan konsultasi (*consultation enviroment*) digunakan oleh pengguna yang bukan pakar untuk memperoleh pengetahuan pakar (Tarigan, 2013).



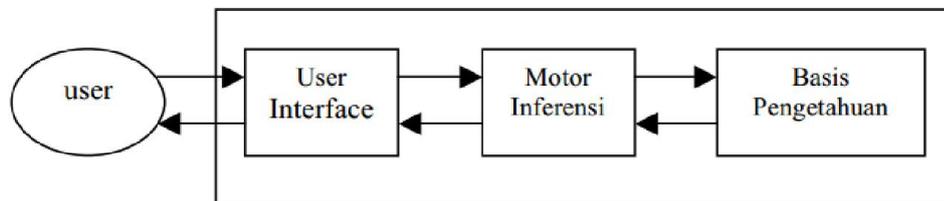
**Gambar 2.3** Struktur Sistem Pakar

Sumber: (Yossi Octavina, 2014)

### 2.1.2.4. Komponen Sistem Pakar

Sistem pakar pada umumnya mempunyai tiga elemen, yaitu : basis pengetahuan (*Knowledge Base*), Mesin Inferensi (*Inference Engine*), dan

Antarmuka Pemakai (*User Interface*). Dengan blok umum hubungan dari ketiga komponen sistem pakar tersebut dapat dilihat pada Gambar 2.4 di bawah ini :



**Gambar 2.4** Diagram Blok umum Sistem Pakar

**Sumber:** (Yossi Octavina, 2014)

Dari gambar diatas, Sistem Pakar terdiri dari 3 komponen utama, yaitu: basis pengetahuan (*knowledge bases*), motor inferensi dan interface.

1. Representasi Pengetahuan (*Knowledge Base*) adalah metode yang digunakan untuk mengkodekan pengetahuan dalam sebuah sistem pakar yang berbasis pengetahuan. Representasi pengetahuan dimaksudkan untuk menangkap sifat-sifat penting problema dan membuat informasi tersebut dapat diakses oleh prosedur pemecahan masalah.

Basis pengetahuan mengandung pengetahuan untuk pemahaman, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui. Ada beberapa metode runtuk mempresentasikan pengetahuan kedalam basis pengetahuan, yaitu :

a. Logika (*Logic*)

Merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah, dan prosedur yang membantu proses penalaran.

b. Jaringan Semantik (*semantic nets*)

Jaringan semantik adalah objek yang paling awal dipakai dalam mempresentasikan pengetahuan. Metode ini didasarkan pada struktur jaringan dan biasa digambarkan dengan grafik hubungan. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tersebut. Objek ini bisa berupa benda atau peristiwa. Antara dua objek dihubungkan oleh *arc* yang menunjukkan hubungan antar objek.

c. Kaidah produksi.

Kaidah produksi biasanya dalam bentuk jika-maka (*IF-THEN*). Kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian, yaitu bagian premis (jika) dan bagian konklusi (maka) dengan format kaidah *IF-THEN*.

d. Pohon Pelacakan

Merupakan struktur penggambaran secara hierarkis. Struktur pohon terdiri atas *node-node* yang menunjukkan objek dan *arc* (busur) yang menunjukkan hubungan antar objek. Untuk menghindari kemungkinan adanya proses pelacakan suatu *node* secara berulang, maka digunakan struktur pohon.

## 2. Antarmuka Pengguna

Antarmuka pengguna adalah perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem. *User Interface* merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu, antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

3. Mesin Inferensi (motor inferensi) adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace* dan untuk memformulasikan kesimpulan (Turban, 1995). Sedangkan untuk pelacakan data terdapat 2 cara yang dapat di gunakan untuk mempersentasikanya yaitu:

### a. *Forward chaining*

Adalah pencocokan data atau pernyataan dimulai dari bagian sebelah kiri (*IF* dulu). Dengan kata lain penalaran lain dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesa.

### b. *Backward chaining*

Adalah pencocokan fakta atau pernyataan yang dimulai dari bagian sebelah kanan (*THEN* dulu). Dengan kata lain penalaran dimulai dari hipotesa terlebih dahulu. Dan untuk menguji hipotesa

tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan.

Komponen lain yang merupakan dari struktur dari sistem pakar antara lain :

1. *Workplace*

*Workplace* merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil dan kesimpulan yang dicapai

2. Fasilitas penjelasan

Fasilitas penjelasan berguna dalam memberikan penjelasan kepada pengguna mengapa komputer meminta suatu informasi tertentu dari pengguna dan dasar apa yang digunakan komputer sehingga dapat menyimpulkan suatu kondisi.

3. Perbaikan pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran terkomputerisasi sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

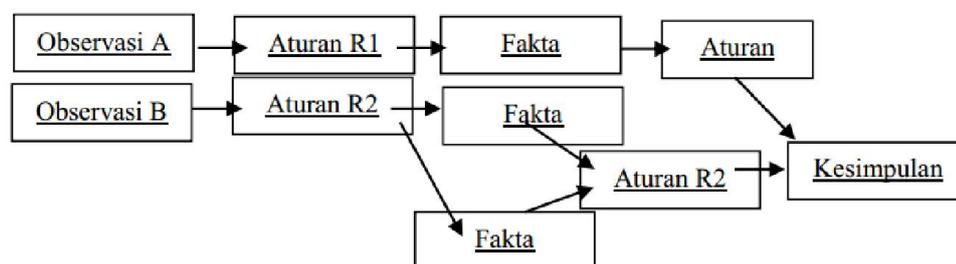
### 2.1.2.5. Rantai Telusur Maju (*Forward Chaining*)

Pada penelitian ini metode yang dipakai adalah metode *forward chaining*. Metode ini di anggap tepat dalam penelusuran terhadap gejala-gejala kerusakan komputer yang akan membantu dalam proses pengambilan keputusan terhadap jenis kerusakan yang terjadi.

*Forward Chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*, maka *rule* tersebut dieksekusi. Bila sebuah *rule* dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambahkan ke dalam *database*. Setiap kali pencocokan, dimulai dari *rule* teratas. Setiap *rule* hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi *rule* yang bisa dieksekusi. Metode pencarian yang digunakan adalah *Depth-First Search* (DFS), *Breadth-First Search* (BFS) atau *Best First Search*. Pendekatan dalam pelacakan dimulai dari informasi masukan dan selanjutnya mencoba menggambarkan kesimpulan, pelacakan kedepan mencari fakta yang sesuai dengan bagian *IF* dari aturan *IF-THEN*. Dengan metode *forward chaining* dari pendekatan dan aturan yang telah dihasilkan dapat ditinjau oleh para ahli untuk diperbaiki atau dimodifikasi untuk memperoleh hasil yang lebih baik (Verina, 2015).

Metode *Forward Chaining* merupakan metode pelacakan yang memulai proses pelacakan dari sekumpulan data atau fakta. Dari data-data tersebut dicari suatu kesimpulan yang menjadi solusi dari masalah yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang

premisnya sesuai dengan data data tersebut. Kemudian dari kaidah-kaidah tersebut diperoleh suatu kesimpulan. Pada metode *Forward Chaining* proses pencarian dengan data sehingga proses ini disebut *data-driven* atau *inferensi* yang dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan (Harison dan Alexyusandera, 2014)

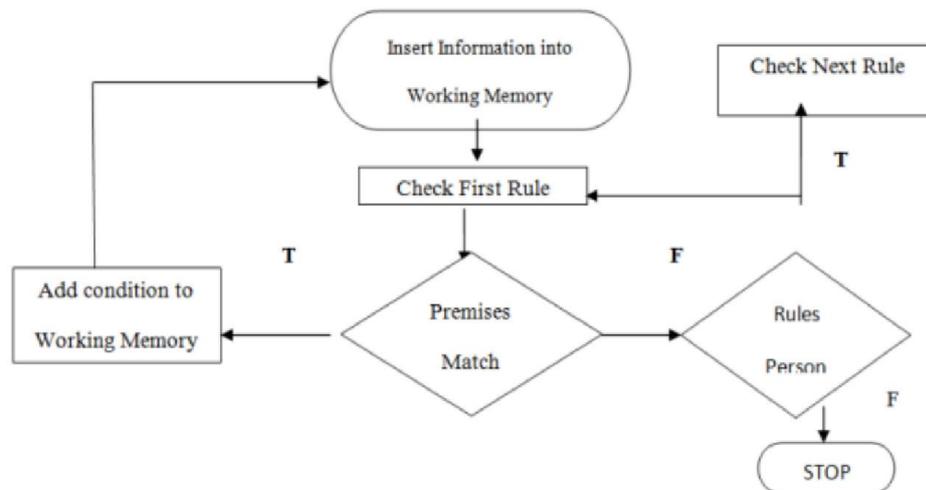


**Gambar 2.5** *Forward Chaining*

**Sumber:** (Verina, 2015)

#### 2.1.2.6. Konsep *Forward Chaining*

Operasi dari sistem *forward chaining* dimulai dengan memasukkan sekumpulan fakta yang diketahui ke dalam memori kerja (*working memory*), kemudian menurunkan fakta baru berdasarkan aturan yang premisnya cocok dengan fakta yang diketahui. Proses ini dilanjutkan sampai dengan mencapai *goal* atau tidak ada lagi aturan yang premisnya cocok dengan fakta yang diketahui. Operasi tersebut digambarkan seperti pada gambar 2.6 (Tarigan, 2013)



**Gambar 2.6** Operasi Sistem Pakar *Forward Chaining*

Sumber: (Tarigan, 2013)

Langkah-langkah yang harus dilakukan dalam membuat sistem *forward chaining* berbasis aturan, yaitu :

1. Pendefinisian Masalah. Tahap ini meliputi pemilihan domain masalah dan akuisisi pengetahuan.
2. Pendefinisian Data Input. Sistem *forward chaining* memerlukan data awal untuk memulai inferensi.
3. Pendefinisian Struktur Pengendalian Data. Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan pengaktifan suatu aturan.
4. Penulisan Kode Awal. Tahap ini berguna untuk menentukan apakah sistem telah menangkap domain pengetahuan secara efektif dalam struktur aturan yang baik.
5. Pengujian Sistem. Pengujian sistem dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan benar.

6. Perancangan Antarmuka. Antarmuka adalah salah satu komponen penting dari suatu sistem. Perancangan antarmuka dibuat bersama-sama dengan pembuatan basis pengetahuan.
7. Pengembangan Sistem. Pengembangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan prototipe sistem.
8. Evaluasi Sistem. Pada tahap ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.

## **2.2. Variabel**

Pada penelitian ini, peneliti mengambil kamera Ip Cam sebagai variabel. Variabel merupakan suatu sifat yang dapat memiliki berbagai macam nilai yang biasanya diekspresikan dalam bentuk simbol/lambang (umunya digunakan simbol  $x$  dan  $y$ ) yang padanya dilekatkan bilangan atau nilai (Ikhsan, 2008, p. 64).

### **2.2.1. Internet Protocol Camera**

Dengan semakin majunya teknologi pada saat ini membuat masyarakat semakin mudah dalam segala bidang. Ip Cam merupakan salah satu contoh teknologi yang selalu berkembang. Di karenakan kriminalitas pada saat ini semakin tinggi maka Ip Cam dijadikan sistem pemantauan utama.

Ip Cam atau juga yang menyebutnya *Netcam (Network Camera)* merupakan perangkat peng-*capture* dan *recording* objek terkini yang memiliki kemampuan

memproses *visual* dan *audio* serta dapat diakses komputer secara langsung, atau melalui *LAN internet*, dan jaringan telepon seluler (Amir, 2009, p. 6).

Instalasinya sangat sederhana. Sebuah Ip Cam ditempatkan di lokasi yang telah di tentukan guna memantau keadaan, kemudian lakukan setting melalui komputer secara langsung atau melalui jaringan. Perangkat ini dapat diakses dari mana saja selama kita terkoneksi dengan *internet*, baik dengan laptop maupun telepon seluler.

### **2.2.2. Pengertian Variabel**

Variabel Penelitian pada dasarnya adalah segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya. Pada penelitian ini peneliti membagi variabel menjadi 2 bagian, yaitu bagian *hardware* dan *software*. Bagian *hardware* di bagi menjadi 5 jenis kerusakan, yaitu pada *Power Supply*, *port RJ45*, kaca Ip Cam, *ASIC*, *Lensa/infrared*, dan bagian *software* hanya meliputi bagian pengaturan Ip Cam. Adapun kerusakan pada bagian *hardware* dan pengaturan sebagai berikut:

#### *1. Hardware*

*Hardware* pada Ip Cam merupakan perangkat keras yang bergantung satu sama lain, jika terjadi satu kerusakan *hardware* maka akan berpengaruh pada kinerja kamera. Di bawah ini terdapat 5 jenis kerusakan *hardware* Ip Cam sebagai berikut:

- a. *Power Supply* pada Ip Cam adalah bagian yang mendistribusikan catu daya dari adaptor ke seluruh bagian agar perangkat dapat beroperasi. Adapun gejala kerusakan pada power supply Ip Cam yaitu mesin cctv hidup tetapi kamera mati total dan lampu pada port hub tidak nyala. *Power supply* ini biasanya bermasalah dikarenakan terkena cairan urin tikus, karat, atau tersambar petir. Maka solusinya Jika *power supply* terkena cairan urin tikus, ambil tisu kering, lap di bagian yang terkena urin tikus lalu keringkan dengan di jemur atau di *blower* dengan hair dryer, Jika *power supply* karat pada bagian listrik, bagian yang karatan bisa di semprot dengan cairan anti karat (WD) dan di sikat dengan sikat gigi, lalu keringkan dengan di jemur, Jika *power supply* tersambar petir, maka harus di tukar dengan *power supply* yang baru.
- b. *Port RG45* adalah lubang untuk mencolokakan kabel data ber-*connector RG45*. Port ini untuk menghubungkan kabel data dari *Hub / mesin cctv* ke kamera. Adapun gejala kerusakan pada *port RG45* yaitu kamera hidup mati dan kamera hidup tetapi tidak tampil. Biasanya permasalahan yang sering terjadi pada *port* ini adalah *port RG45* tersumbat kotoran, kaki tembagaanya bengkok, atau kabel digigit tikus. Maka solusi Jika *port RG45* tersumbat kotoran, ambil sikat gigi kering lalu sikat perlahan pada bagian yang terkena kotoran, Jika kaki tembaga port bengkok, Ambil benda tajam lalu luruskan kaki tembaga perlahan, Jika kabel di gigit tikus, bagian yang di gigit tikus di potong lalu kupas kulit kabelnya dan sambung dengan warna yang sama.

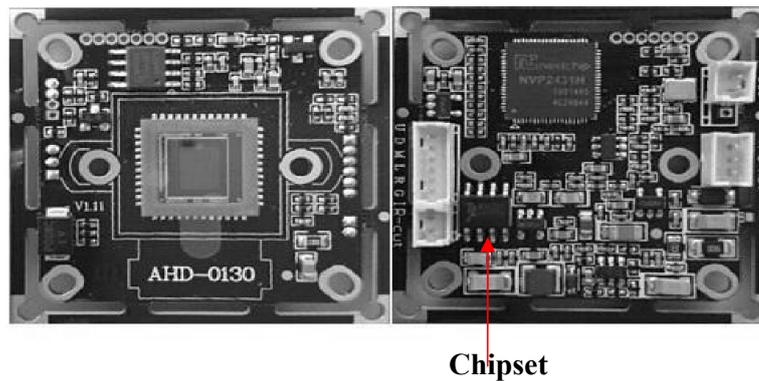
- c. Kaca Ip Cam digunakan untuk melindungi *infrared* dan lensa kamera dari kotoran dan air. Adapun gejala kerusakan pada kaca Ip Cam yaitu tampilan kamera kabur, tampilan gambar berkilau putih dan bulat-bulat kecil, tampilan kamera tertutup sesuatu. Biasanya kaca kamera bermasalah dikarenakan debu yang banyak ataupun uap air. Maka solusinya jika kaca Ip Cam terkena debu, ambil tisu basah, lap perlahan pada bagian yang terkena debu dan keringkan dengan tisu kering, Jika kaca Ip Cam berembun, buka bagian kaca depan. Lap dengan tisu kering lalu pasang kembali bagian kaca depan.
- d. *ASIC* atau biasanya disebut *IC* kamera yang berfungsi sebagai penendali atas seluruh kinerja sebuah Ip Cam. Adapun gejala kerusakan pada *ASIC* yaitu tampilan kamera sering hang atau tidak jalan sama sekali, atau kamera hidup tetapi tidak tampil. Biasanya *ASIC* bermasalah dikarenakan ada kabel yang sot, arus daya tidak stabil, tersambar petir. Solusi untuk pemasalahan ini jika kabel pada motherboard konslate, biasa terjadi karena ada kabel luka yang bersentuhan, cari jalur kabel dan bila ada kabel luka maka di isolasi dengan isolasi listrik, jika arus daya tidak stabil, bisa dikarenakan *power supply* yang lemah, cek dengan *multitester* untuk memastikan arus daya listrik, jika *ASIC* tersambar petir, maka harus digantikan dengan yang baru.
- e. Lensa/*infrared* adalah mata Ip Cam berupa *charge-coupled device* atau CCD yaitu sebuah sensor untuk merekam gambar, terdiri dari sirkuit

terintegrasi berisi kondensator yang berhubungan, atau berpasangan yang mampu meng-*capture* objek pada ruang bercahaya maupun ruang gelap. Adapun gejala kerusakan pada lensa/*infrared* yaitu ketika malam hari tampilan kamera tidak hitam putih. Biasanya kerusakan pada lensa/*infrared* dikarenakan arus daya yang tidak stabil, terkena petir. Solusi untuk permasalahan ini jika arus daya tidak stabil, bisa dikarenakan power supply yang lemah, cek dengan multimeter untuk memastikan arus daya listrik, jika *infrared* tersambar petir, maka harus digantikan dengan yang baru.



**Gambar 2.7** Ip Cam

**Sumber:** (Amir, 2009, p. 7)



**Gambar 2.8** Mainboard Ip Cam

**Sumber:** (Amir, 2009, p. 8)

## 2. Pengaturan

Pada bagian pengaturan Ip Cam adalah pengaturan yang dilakukan setelah selesai instalasi Ip Cam. Biasanya permasalahan pada pengaturan kamera bisa berupa ip *address* pada kamera berubah, pada siang hari kamera hitam putih. Biasanya permasalahan ini terjadi bisa dikarenakan *power supply* mati mendadak atau mesin NVR ter-*update*.

## 2.3. Software Pendukung

*Software* pendukung pada skripsi ini dibutuhkan untuk menjalankan sebuah sistem yang outputnya berupa sebuah solusi kepada pengguna awam. *Software* yang akan digunakan pada penelitian ini adalah *VB.Net* dan *MySQL*.

### 2.3.1. VB.NET

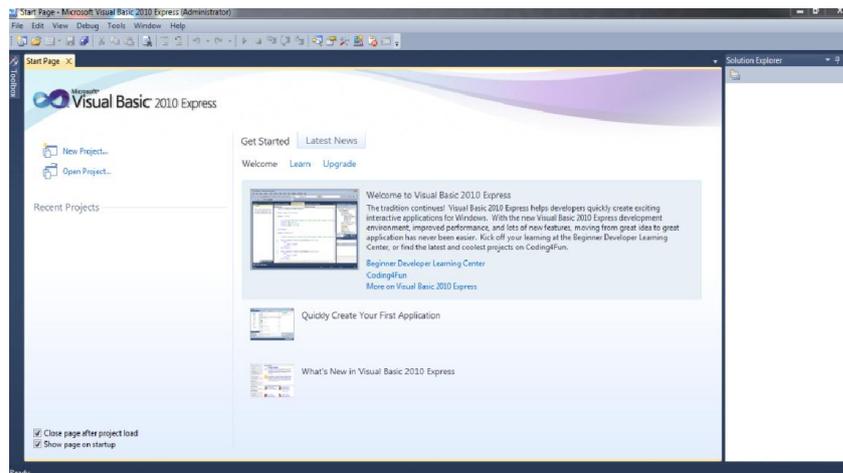
*Visual Basic* 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *Microsoft*, yaitu *Microsoft Visual Studio* 2010.

*Visual Studio* merupakan produk pemrograman andalan dari *Microsoft Corporation*, di mana di dalamnya berisi beberapa jenis *IDE* pemrograman seperti *Visual Basic*, *Visual C ++*, *Visual Web Developer* , *Visual C#*, dan *Visual F#*.

Semua *IDE* pemrograman tersebut sudah mendukung penuh implementasi *.Net Framework* terbaru, yaitu *.Net Framework 4.0* yang merupakan pengembangan dari *.Net Framework 3.5*.

*Visual Basic 2010* merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangan yang terdapat di dalamnya antara lain dukungan terhadap *library* terbaru dari *Microsoft*, yaitu *.Net Framework 4.0*, dukungan terhadap pengembangan aplikasi menggunakan *Microsoft SilverLight*, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap *database-database*, baik *standalone* maupun *database server*.

Bahasa *Visual Basic 2010* sendiri awalnya berasal dari bahasa pemrograman yang sangat populer di kalangan *programmer* komputer, yaitu bahasa *BASIC*, yang oleh *Microsoft* diadaptasi dalam program *Microsoft Quick BASIC*. Seiring dengan berkembangnya teknologi komputasi dan desain, *Microsoft* mengeluarkan produk yang dinamakan *Microsoft Visual Studio* dengan *visual basic* di dalamnya. Saat ini versi *Microsoft Visual Studio* yang beredar adalah versi 10 yang populer dengan nama *Microsoft Visual Visual Studio 2010*, yang di dalamnya termasuk *Microsoft Basic 2010* (Wahana, 2010, pp. 2–3).



**Gambar 2.9** *Visual Basic 2010*

**Sumber:** (Wahana, 2010, pp. 2–3)

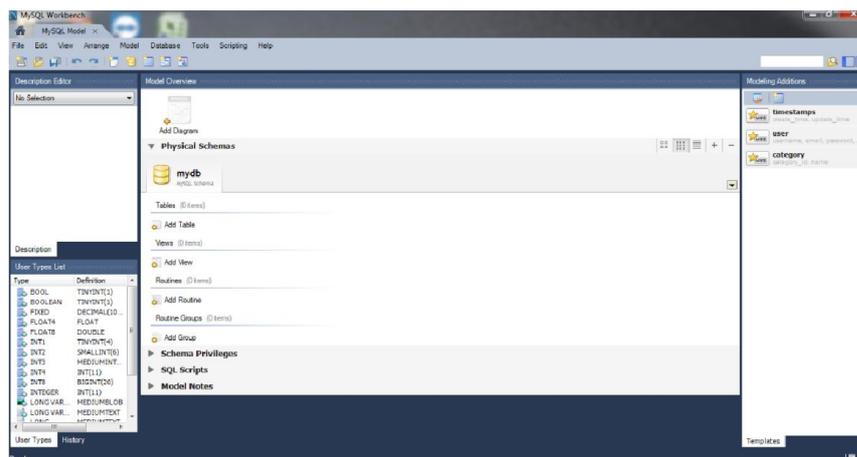
### 2.3.2. *MySQL*

Pada saat ini *MySQL* sudah terkenal dan digunakan oleh para *programer web* pada umumnya, walaupun banyak yang digunakan pula sebagai *database* di *web server* lokal, *mysql* mempunyai fungsi sebagai *SQL (Structured Query Language)* yang sudah dimiliki dan sudah dikembangkan oleh *MySQL*, pada saat ini *MySQL* sudah tidak asing lagi bagi para pemula dalam pembuatan program web, karena digunakan bersamaan dengan PHP untuk membuat aplikasi *server* yang dinamis dan *powerfull*.

*Database MySQL* merupakan aplikasi yang bersifat *daemon* atau menetap dalam memori yang berjalan bersama dengan sistem operasi *microsoft windows*. *Interface* utama *MySQL database server* adalah *command line* atau berbasis DOS sehingga diperlukan pengetahuan khusus mengenai penggunaan perintah atau *command* dalam *command shell MySQL* (Wahana, 2010, p. 4).

*MySQL* adalah salah satu jenis *database server* versi *open source* sehingga dapat digunakan oleh semua orang secara gratis tanpa adanya tuntutan

hak cipta intelektual bagi pemakainya. Kelebihan *database MySQL* adalah dapat dijalankan pada dua *platform Windows* dan *Linux*. *MySQL* mempunyai tiga subbahasa, yaitu *Data Definition Language (DDL)*, *Data Manipulation Language (DML)*, dan *Data Control Language*. DDL berfungsi pada obyek *database*, seperti membuat *table*, mengubah *table* dan menghapus *table*. DML untuk objek *table*, seperti melihat, menambah, menghapus dan mengubah isi *table*. Sedangkan DCL untuk kepentingan keamanan *database*, seperti memberikan hak akses ke *database* dan menghapus hak tersebut dari *database*. Dalam perancangan ini *MySQL* dipergunakan untuk dikoneksikan baik secara *on line* maupun *client server* (Setyowati & Permana, 2013).



**Gambar 2.10** *MySQL*

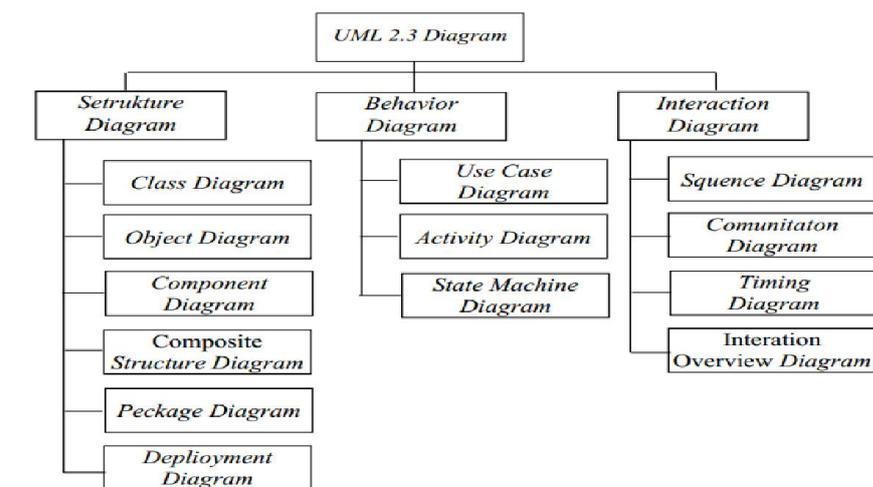
Sumber: (Setyowati & Permana, 2013)

### 2.3.3. Alat Perancangan Sistem UML

Pada saat ini, pemodelan yang paling banyak digunakan adalah pemodelan UML. *Unified Modeling Language (UML)* adalah sebuah bahasa yang diterima dan digunakan oleh *software developer* dan *software analyst* sebagai suatu bahasa

yang cocok untuk merepresentasikan grafik dari suatu relasi antar entitas-entitas *software* (Winata & Setiawan, 2013).

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.11. dibawah ini (A.S. & Shalahuddin, 2013, p. 140).



**Gambar 2.11** Diagram UML

**Sumber:** (A.S. & Shalahuddin, 2013, p. 140)

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

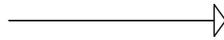
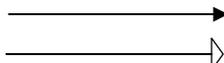
Pada penelitian ini, diagram yang akan digunakan untuk mendesain sistem sebagai berikut:

### 1. Use Case Diagram

*Use case diagram* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat (A.S. & Shalahuddin, 2013: 155-156).

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

**Tabel 2.1** Simbol *Use Case Diagram*

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

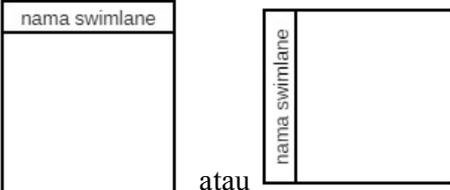
**Sumber:** (A.S. & Shalahuddin, 2013)

## 2. Activity Diagram

Teknik untuk menjelaskan *business process*, *procedural logic*, dan *work flow*. Bisa dipakai untuk menjelaskan teks *use case* dalam notasi grafis dengan menggunakan notasi yang mirip *flow chart*, meskipun terdapat sedikit perbedaan notasi (Winata & Setiawan, 2013).

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

**Tabel 2.2** Simbol *Activity Diagram*

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

**Sumber:** (Winata & Setiawan, 2013)

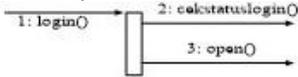
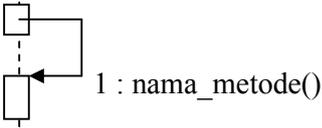
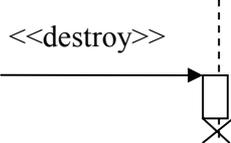
### 3. *Sequence Diagram*

Menjelaskan interaksi obyek-obyek yang saling berkolaborasi (berhubungan), mirip dengan *activity diagram* yaitu menggambarkan alur kejadian sebuah aktivitas tetapi lebih detil dalam menggambarkan aliran data termasuk data atau *behaviour* yang dikirimkan atau diterima namun kurang mampu menjelaskan detil dari sebuah algoritma (Winata & Setiawan, 2013).

Berikut adalah simbol-simbol yang ada pada diagram *sequence*:

**Tabel 2.3** Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p> <p>atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>Nama aktor</u> </div> <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal <i>frase</i> nama aktor</p>
<p>Garis Hidup / lifeline</p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>nama objek : nama kelas</u> </div>	<p>Menyatakan objek yang berinteraksi pesan</p>

<p>Waktu Aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Misalnya</p>  <p>maka <i>cekStatusLogin()</i> dan <i>open()</i> dilakukan di dalam metode <i>login()</i> Aktor tidak memiliki waktu aktif</p>
<p>Pesan Tipe Create</p> <p>&lt;&lt;create&gt;&gt;</p> 	<p>Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat</p>
<p>Pesan Tipe Call</p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode yang di panggil harusvada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan Tipe Send</p> <p>1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dikirim</p>
<p>Pesan Tipe Return</p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p>Pesan Tipe Destroy</p> <p>&lt;&lt;destroy&gt;&gt;</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

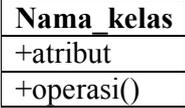
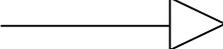
Sumber: (A.S. & Shalahuddin, 2013)

#### 4. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem .

Berikut adalah simbol-simbol yang ada pada *Class Diagram*:

**Tabel 2.4** Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / <i>dependency</i> 	Relasi kelas dengan makna kebergantungan antarkelas
Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

**Sumber:** (A.S. & Shalahuddin, 2013)

#### 2.4. Penelitian Terdahulu

Sistem pakar ini dikembangkan untuk meningkatkan hasil dari penelitian sebelumnya yang sudah pernah dilakukan. Adapun beberapa penelitian sebelumnya yang telah dilakukan sebagai berikut:

Menurut Dewa Gede Hendra Divayana(2014), ( ISSN 2223-4985) dengan judul “*Application of Pineapple Diseases Expert System with FC-FL Method at Badung Regency Agriculture Department*” Data Dinas Pertanian Provinsi Bali menunjukkan bahwa hingga 2011 perkebunan nanas tersebar di beberapa kabupaten di Bali antara lain: Tabanan, Gianyar, Badung, Buleleng, Karangasem, Bangli, Klungkung, dan Jembrana. Namun dua tahun terakhir, terutama di daerah Badung, penyebaran tanaman nanas menurun 16,72%, hal ini disebabkan perkebunan yang semakin sempit, Harga pupuk tidak stabil dan yang paling parah adalah adanya hama dan penyakit. Sedangkan untuk penyakit berbahaya yang Ditemui pada nanas antara lain: *Corynespora*, *Anthracnose*, *Root rot*, *Ring spotting*, and *Bacillus pineapple* (Hayadi, 2017).

Masalah penyakit adalah masalah yang menjadi salah satu faktor utama yang menyebabkan berkurangnya produksi nanas, oleh karena itu dibutuhkan informasi awal yang diperlukan tentang penyakit nanas. Dengan pentingnya informasi awal tentang Penyakit pada nanas, petani perlu dilengkapi dengan pengetahuan atau informasi tentang jenis penyakit di Indonesia tentang tanaman nanas dan cara mengatasinya. Informasi tentang penyakit pada tanaman nanas tidak hanya harus diperoleh melalui penyuluhan yang diberikan Dinas Pertanian,

namun bisa juga dilakukan melalui media seperti buku, majalah, atau media sosial di internet. Namun, dengan hanya menggunakan media seperti informasi di atas tidak cukup. Dengan pertimbangan seperti itu, sangat penting untuk melakukannya mengembangkan dan menerapkan sistem pakar yang dapat memudahkan para petani dalam membuat diagnosis, mendapatkan sendiri informasi langsung berdasarkan fakta yang ada di lapangan tentang penyakit nanas.

Menurut Setiawan Widiyanto(2017), (ISSN: 2349-2163) dengan judul *“Expert System to Determine the Priority Scale of Case in Laboratory of Forensic Using Forward Chaining and Backward Chaining Methods Rule Based”* Setiap negara memiliki lembaga penegakan hukum dengan pengungkap subkelas tugas. Labfor adalah bagian dari Polisi yang berfungsi sebagai pendukung dalam penyelesaian kasus perdata dan pidana. Di bidang hukum, pengadilan perdata dan pengadilan pidana selesai memiliki alur untuk menyelesaikan perselisihan dan batas waktu penahanan. Batas akhir penyelesaian tidak dapat ditentukan karena perpanjangan waktu penahanan; Hasilnya sulit diprediksi kapan rongga akan selesai.

Sistem Pakar yang melibatkan pendapat ahli kriminal diterapkan untuk memudahkan proses peradilan untuk menentukan hukuman, yang menerima tersangka yang melakukan pelanggaran terhadap wanita. Hasil penelitian ini mampu memberikan output dari sistem hukuman berupa denda yang dijatuhkan.

Menurut Wiwi Verina(2015), (ISSN : 2407-4322) dengan judul **“Penerapan Metode Forward Chaining untuk Mendeteksi Penyakit THT”**

Saat ini penyakit Telinga Hidung dan Tenggorokan (THT) telah menjadi suatu penyakit yang cukup banyak diderita oleh masyarakat dunia. Peningkatan penyakit THT yang semakin tinggi, tidak diiringi oleh jumlah tenaga ahli yang bertugas melakukan diagnosa atas seorang pasien yang diperkirakan menderita THT. Pasien disini adalah orang yang menerima perhatian atau perawatan kesehatan. Penyakit THT adalah penyakit yang menyerang sekitar kepala yaitu telinga, hidung dan tenggorokan.

Penyakit telinga terdiri dari 11 jenis penyakit, hidung terdiri dari 8 jenis dan penyakit tenggorokan terdiri 17 jenis penyakit. Karena letak penyakit saling berdekatan maka gejala yang timbul hampir sama tetapi yang membedakannya hanya gejala yang spesifik saja. Oleh sebab itu untuk mendiagnosa penyakit ini harus dilakukan dengan secara cermat dan teliti karena memakai pedoman gejala sebagai aturan. Hal tersebut menambah beban kerja tenaga ahli yang bertugas melakukan diagnosa atas seorang pasien yang diperkirakan menderita THT sehingga dengan permasalahan seperti ini sangat dibutuhkan sebuah sistem pakar yang dapat membantu dalam pemecahan masalah.

Salah satu wujud nyata dari teknologi ini adalah penerapan sistem komputerisasi sebagai contoh kegiatan pengolahan data dengan menggunakan sistem terkomputerisasi adalah sistem pakar. Sistem Pakar adalah sebuah program aplikasi komputer yang berperilaku layaknya seorang ahli. Aplikasi yang

digunakan biasa yaitu membantu mendiagnosa penyakit, kerusakan peralatan dan pengukuran data.

Menurut Nancy Extise Putri(2016), (ISSN : 1693-752X) dengan judul **“Sistem Pakar Kerusakan Hardware Komputer Dengan Metode *Forward Chaining*”** Perkembangan teknologi komputer diikuti pula dengan meningkatnya jumlah pengguna komputer di dunia. Seiring dengan meningkatnya jumlah pengguna komputer, permasalahan kerusakan komputer menjadi masalah yang cukup rumit. Hal ini dapat dimaklumi mengingat banyaknya pengguna komputer yang kurang memiliki pengetahuan tentang komputer, khususnya dalam menangani kerusakan komputer. Permasalahan ini secara umum dialami baik oleh individu, maupun institusi. Banyak sekali dana yang dikeluarkan untuk memperbaiki kerusakan komputer, padahal kerusakan komputer yang terjadi belum tentu rumit dan dapat diperbaiki secara mandiri.

Dengan menggunakan aplikasi sistem pakar kerusakan *hardware* komputer Berbasis *Web*, maka pencarian informasi tentang kerusakan *hardware* komputer yang dilakukan akan lebih mudah. Aplikasi sistem pakar kerusakan hardware komputer Berbasis *Web* dapat diproses secara maksimal dan lebih cepat dimanapun *user* berada dengan menggunakan jaringan internet dapat mengetahui kerusakan maupun gejala kerusakan *hardware* yang diinginkan oleh pengguna komputer.

Menurut Rosmawati Tamin(2015), (ISSN : 2442 - 4512) dengan judul **“Sistem Pakar untuk Diagnosa Kerusakan Pada Printer Menggunakan Metode *Forward Chaining*”** Printer sebagai kebutuhan pokok dalam menunjang aktivitas harian beberapa bidang usaha diantaranya percetakan, desain, kantor dan lain-lain. Oleh karena itu, kerusakan printer yang sering dialami oleh pengguna sangatlah menghambat kelancaran pekerjaan mereka. Analisa kerusakan printer yang dilakukan dengan cara manual dan hanya dikerjakan oleh teknisi terkadang membutuhkan waktu yang tidak sedikit hal ini diperparah dengan jumlah teknisi yang terbatas hal ini tentunya akan berbanding terbalik dengan jumlah pelanggan semakin banyak sebagai akibatnya efektifitas dan efisiensi kerja menjadi menurun. Untuk menangani permasalahan tersebut maka dibutuhkan sebuah sistem yang mampu bekerja otomatis dengan waktu singkat untuk menganalisa, menemukan dan memberikan solusi.

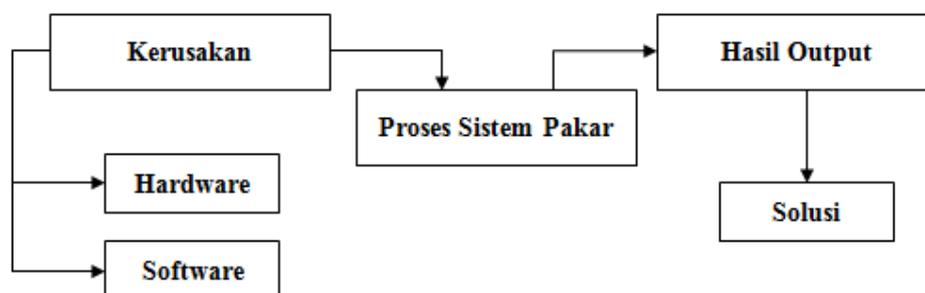
Aplikasi yang dibangun ini memudahkan para pengguna printer jenis canon untuk mengetahui penyebab, akibat dan gejala-gejala yang ditimbulkan dari kerusakan printer, memudahkan para pengguna printer jenis canon untuk mencari solusi kerusakan printer, memudahkan para pengguna printer untuk mendapatkan informasi mengenai cara merawat printer dengan baik melalui penyajian informasi berita yang terdapat dalam website aplikasi mendeteksi kerusakan pada printer dan memudahkan para teknisi untuk memperbaiki printer.

## 2.5. Kerangka Pemikiran

Penelitian ini berawal dari kurangnya pengetahuan user tentang gejala kerusakan Ip Cam. Ketika kerusakan semakin parah maka akan terjadi pengeluaran biaya yang besar. Dengan adanya aplikasi sistem pakar dapat melakukan diagnosa kerusakan Ip Cam secara cepat dan biaya yang perlu dikeluarkan tidak begitu besar.

Penelitian ini dilakukan untuk membuat aplikasi sistem pakar untuk mendiagnosa kerusakan Ip Cam. Sistem ini dibangun menggunakan metode *forward chaining* yaitu metode yang digunakan untuk menguji gejala-gejala yang dimasukkan oleh pengguna yang selanjutnya diambil solusi berdasarkan aturan yang disimpan oleh sistem.

Aplikasi sistem pakar ini dibuat untuk membantu teknisi cctv maupun masyarakat dalam melakukan konsultasi kerusakan Ip Cam secara efektif dan efisien, sehingga kerusakan yang dialami dapat diketahui dan dapat diatasi dengan solusi yang diberikan.



**Gambar 2.12** Kerangka Pemikiran

Sumber: (Data Olahan Peneliti)

Pada bagian kerusakan peneliti akan mengelompokkan menjadi 2 bagian, yaitu bagian *hardware* dan *software*. Pada bagian proses sistem pakar, peneliti akan menggunakan *VB .Net* sebagai bahasa pemrograman dan *MySQL* sebagai database. Setelah proses sistem pakar, maka proses output data akan di olah dan nantinya akan menghasilkan solusi untuk pengguna.

Di bagian kerusakan, pengguna bisa input gejala-gejala pada Ip Cam yang bermasalah. Disana akan keluar pilihan bagian hardware atau software. Setelah di input maka data akan di proses dan menghasilkan beberapa pilihan kemungkinan kerusakan pada bagian mana.