

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Landasan teori perlu ditegakkan agar penelitian mempunyai dasar yang kokoh, dan bukan sekedar perbuatan coba-coba (*trial and error*). Adanya landasan teori merupakan ciri bahwa penelitian merupakan cara ilmiah untuk mendapatkan data. Kerlinger mengatakan bahwa teori adalah seperangkat konstruk (konsep), definisi, dan proposisi yang berfungsi untuk melihat fenomena secara sistematis melalui spesifikasi hubungan antarvariabel sehingga dapat berguna untuk menjelaskan dan meramalkan fenomena. Dengan kata lain, teori adalah generalisasi atau kumpulan generalisasi yang dapat digunakan untuk menjelaskan berbagai fenomena secara sistematis (Sudaryono, 2015:13).

Pada bab ini akan dijelaskan tentang beberapa teori dasar antara lain kecerdasan buatan atau *artificial intelligence* serta beberapa subdisiplin ilmunya seperti logika fuzzy (*fuzzy logic*), jaringan syaraf tiruan dan sistem pakar (*expert system*) serta teori dasar tentang *software* pendukung yang digunakan dalam penelitian ini, dari *undefined modelling language* (uml) sebagai bahasa standar untuk merancang sistem, *hypertext preprocessor* (php) sebagai bahasa pemrograman, xampp sebagai *web server*, dan *MySQL* sebagai *database*.

2.1.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan adalah salah satu bagian ilmu komputer yang mempelajari cara membuat mesin (komputer) melakukan sesuatu seperti yang dilakukan oleh manusia (Sri Kusumadewi, 2007:1).

“Definisi lain oleh Sri Hartati dan Sari Ismawanti (2008:1) mengatakan kecerdasan buatan (*Artificial Intelligence*) sebagai sebuah studi tentang bagaimana membuat komputer dapat berperilaku cerdas seperti melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia (dalam Akim dan Budi, 2013:36).

Kecerdasan Buatan atau *Artificial Intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia.

Tujuan dari kecerdasan buatan menurut Winston dan Prendergast (1984) adalah:

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

Dengan kecerdasan buatan, diharapkan tidak menutup kemungkinan hanya dengan data pengetahuan yang terbatas, sebuah komputer dapat berpikir seperti manusia dalam menghadapi masalah dan memberikan sebuah solusi dari masalah yang dihadapi.

Macam-macam kecerdasan buatan, Sutojo *et al* (2011:13-21) antara lain:

1. Sistem pakar (*Expert System*), adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar memiliki beberapa metode yaitu metode *forward chaining*, *backward chaining*, *dempster-shafer*, *certainly factor* dan probabilitas *bayesian*.
2. Logika fuzzy (*Fuzzy Logic*), adalah metode yang dipakai untuk mengatasi hal yang tidak pasti pada masalah-masalah yang mempunyai banyak jawaban. Pada dasarnya *fuzzy logic* merupakan logika bernilai banyak yang mampu mendefinisikan nilai diantara keadaan yang konvensional seperti benar atau salah, ya atau tidak, putih atau hitam dan lain lain. Logika *fuzzy* memiliki beberapa metode yaitu metode tsukamoto, metode mamdani (metode *Max-Min*), dan metode sugeno.
3. Jaringan syaraf tiruan, merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut.
4. Pengolahan bahasa alami adalah pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antarmanusia.

5. Pengenalan ucapan (*Speech Recognition*), atau yang sering disebut dengan *Automatic Speech Recognition* (ASR), adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan.
6. *Computer vision*, adalah salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati atau diobservasi.

2.1.2 Sistem Pakar

Sistem pakar merupakan salah satu alternatif terbaik untuk menyelesaikan berbagai persoalan dengan menggunakan komputer yang didukung oleh teknik-teknik kecerdasan buatan

Ada beberapa definisi tentang sistem pakar menurut para ahli, Sutojo (2011:160) antara lain:

1. Menurut *Turban*, Sistem pakar adalah sebuah sistem yang menggunakan pengetahuan manusia dimana pengetahuan tersebut dimasukkan kedalam sebuah komputer dan kemudian digunakan untuk menyelesaikan masalah-masalah yang biasanya membutuhkan kepakaran atau keahlian manusia.
2. Menurut *Jackson*, Sistem pakar adalah program komputer yang merepresentasikan dan melakukan penalaran dengan pengetahuan beberapa pakar untuk memecahkan masalah atau memberikan saran.

3. Menurut Luger dan Stubbelfield, Sistem pakar adalah program yang berbasiskan pengetahuan yang menyediakan solusi berupa kualitas seorang pakar kepada masalah-masalah dalam bidang (domain) yang spesifik.

Jadi, dari beberapa pengertian diatas maka pengertian sistem pakar sendiri adalah suatu sistem yang mempunyai kemampuan seperti seorang pakar atau mempunyai kemampuan seperti seseorang yang ahli dalam bidang tertentu, dimana dapat menjawab masalah dan memberikan keputusan berupa solusi terhadap permasalahan yang dihadapi seperti layaknya seorang pakar.

Manfaat sistem pakar antara lain yaitu (Sutojo *et al.*,2011:160-161):

1. Dapat meningkatkan output dan produktivitas, karena Sistem Pakar dapat bekerja lebih cepat dari manusia.
2. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
3. Dapat beroperasi di lingkungan yang berbahaya.
4. Mampu menangkap kepakaran yang sangat terbatas.
5. Memudahkan akses ke pengetahuan.
6. Mampu menyediakan pelatihan. Pengguna pemula yang bekerja dengan Sistem Pakar akan menjadi lebih berpengalaman. Fasilitas penjelas dapat berfungsi sebagai guru.
7. Meningkatkan kemampuan problem solving, karena mengambil sumber pengetahuan dari banyak pakar.

Disamping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan, antara lain (Sutojo *et al.*, 2011:161):

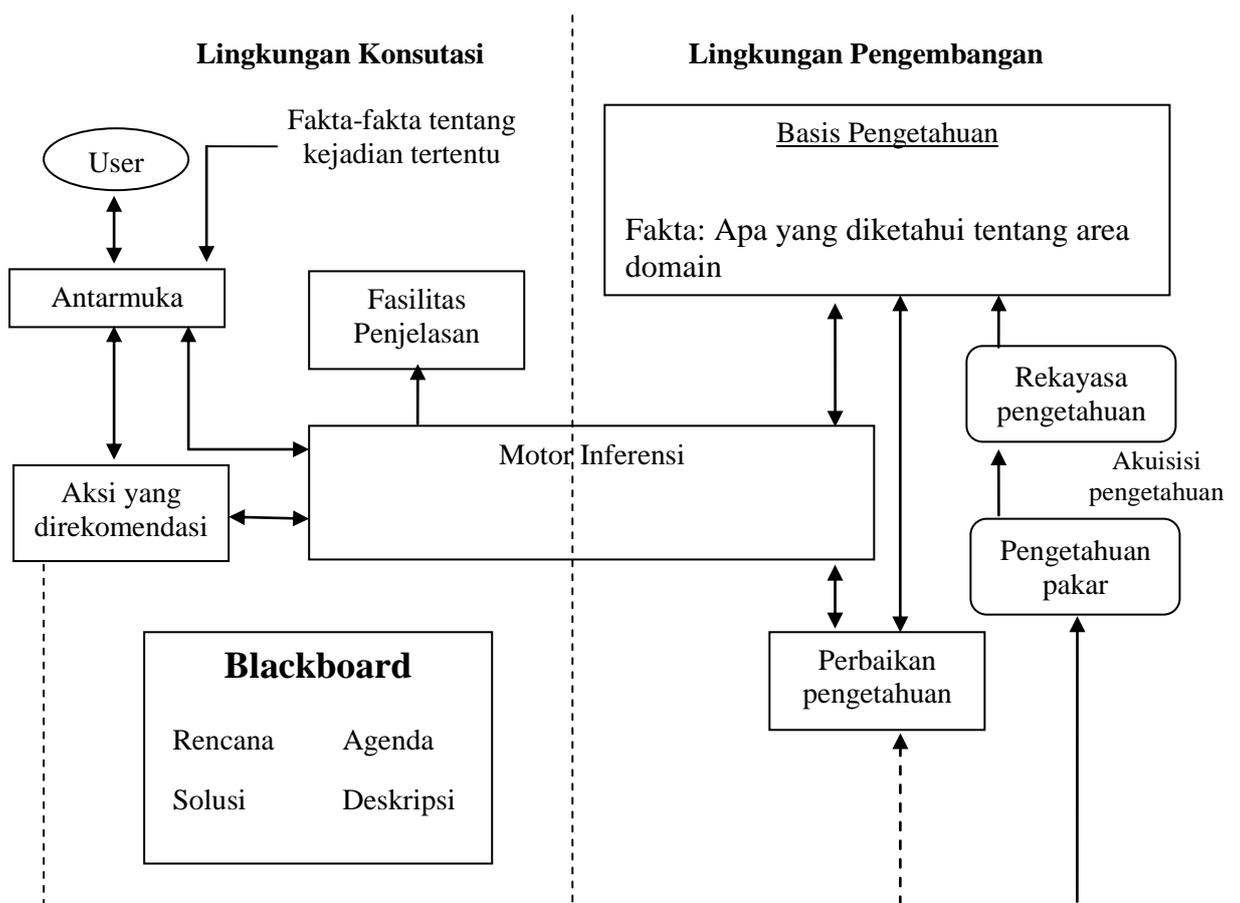
1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
3. Sistem pakar tidak 100% bernilai benar

Suatu sistem dapat dikatakan sistem pakar jika memiliki ciri-ciri sebagai berikut (Sutojo *et al.*, 2011:162):

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

2.1.2.1 Struktur Sistem Pakar

Ada dua bagian penting dari sistem pakar, yaitu Lingkungan pengembangan yang digunakan oleh pembuat sistem pakar untuk membangun komponen-komponennya dan memperkenalkan pengetahuan kedalam basis pengetahuan (*knowledge base*) dan lingkungan konsultasi yang digunakan oleh pengguna berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan seorang pakar. Berikut komponen-komponen yang penting dalam sebuah sistem pakar (Sutojo *et al.*, 2011:167).



Gambar 2.1. Komponen-komponen penting dalam sistem pakar
Sumber: Sutojo *et al* (2011:167)

Berikut penjelasan dari subsistem yang ada pada struktur sistem pakar pada Gambar 2.1:

1. Akuisisi Pengetahuan

Subsistem ini digunakan untuk memasukkan pengetahuan dari seorang pakar dengan cara merekayasa pengetahuan agar bisa diproses oleh komputer dan menaruhnya kedalam basis pengetahuan dengan format tertentu.

2. Basis pengetahuan

Basis pengetahuan mengandung pengetahuan yang diperlukan untuk memahami, memformulasikan, dan menyelesaikan masalah. Basis pengetahuan terdiri dari dua elemen dasar yaitu fakta dan *rule*.

3. Mesin Inferensi

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

4. Daerah kerja (*Blackboard*)

Blackboard merupakan area pada memori yang berfungsi sebagai basis data. Tiga tipe keputusan yang dapat direkam pada *blackboard* yaitu rencana, agenda dan solusi.

5. Antarmuka pengguna (*User Interface*)

Digunakan sebagai media komunikasi antara pengguna dengan sistem pakar.

6. Sistem perbaikan pengetahuan

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

7. Pengguna (*user*)

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

2.1.2.2 Mesin Inferensi

Mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, biasa dikatakan sebagai mesin pemikir (Sri Hartati dan Sari Iswanti, 2008:5). Pada prinsipnya mesin inferensi inilah yang akan mencari solusi dari suatu permasalahan. Konsep yang biasanya digunakan untuk mesin inferensi adalah runut balik (*top-down*), yaitu proses penalaran dari tujuan yang kita inginkan, menelusuri fakta-fakta yang mendukung untuk mencapai tujuan. Selain itu juga dapat menggunakan runut maju (*bottom-up*), yaitu proses penalaran yang bermula dari kondisi yang diketahui menuju tujuan yang diinginkan.

Mesin inferensi sesungguhnya adalah program komputer yang menyediakan metodologi untuk melakukan penalaran tentang informasi pada basis pengetahuan dan memori kerja, serta untuk merumuskan kesimpulan kesimpulan. Komponen ini menyajikan arahan tentang bagaimana menggunakan pengetahuan dari sistem dengan membangun agenda yang mengelola dan mengontrol langkah langkah yang diambil untuk menyelesaikan masalah ketika dilakukan konsultasi.

2.1.2.3 Metode Forward Chaining

Metode *forward chaining* adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi-informasi yang ada berupa data *real* lalu bergerak maju melalui premis-premis dan penggabungan rule untuk menghasilkan suatu kesimpulan (*botton up reasoning*) atau tujuan (Tutik dkk, 2009).

Metode *forward chaining* ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai suatu penyelesaian dalam bentuk keputusan, karena dalam *forward chaining* seluruh proses dikerjakan secara berurutan maju.

Langkah penyelesaian menggunakan metode *forward chaining* dalam sistem pakar ini dimulai dengan fakta yang diketahui berupa masalah yang ada pada *printer*, 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 kondisi yang ditambahkan ke sistem akan di proses, jika ditemui suatu kondisi baru dari konklusi yang diminta, sistem akan kembali ke langkah dua dan mencari *rule-rule* dalam *knowledge base* kembali. Jika tidak ada konklusi baru, maka sesi ini akan berakhir.

Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju. Untuk memahami cara kerja metode forward chaining, berikut contoh langkah penyelesaian dengan menggunakan metode forward chaining:

Misalkan diketahui sebuah sistem pakar menggunakan 5 buah rule berikut:

R1 : IF (Y AND D) THEN Z

R2 : IF (X AND B AND E) THEN Y

R3 : IF A THEN X

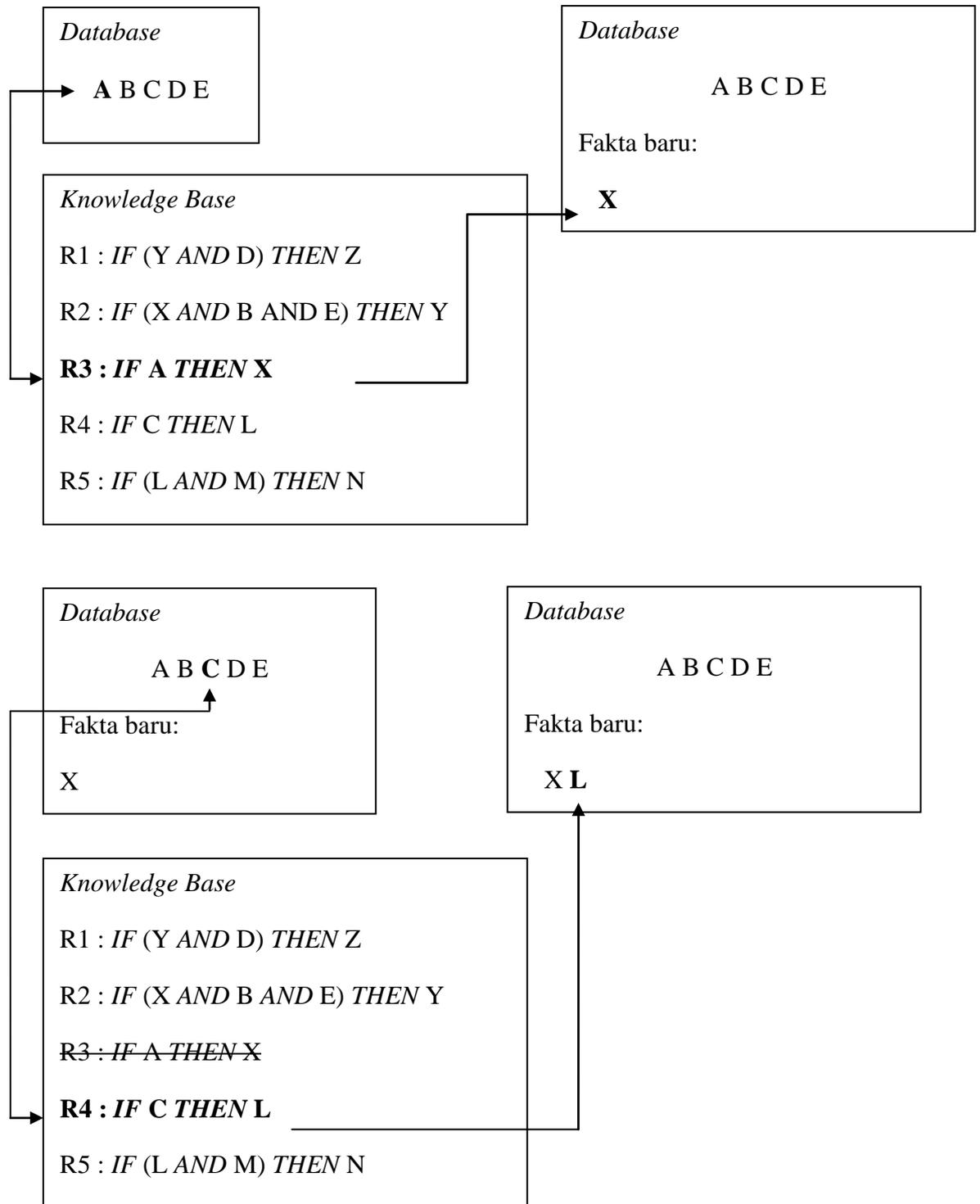
R4 : IF C THEN L

R5 : IF (L AND M) THEN N

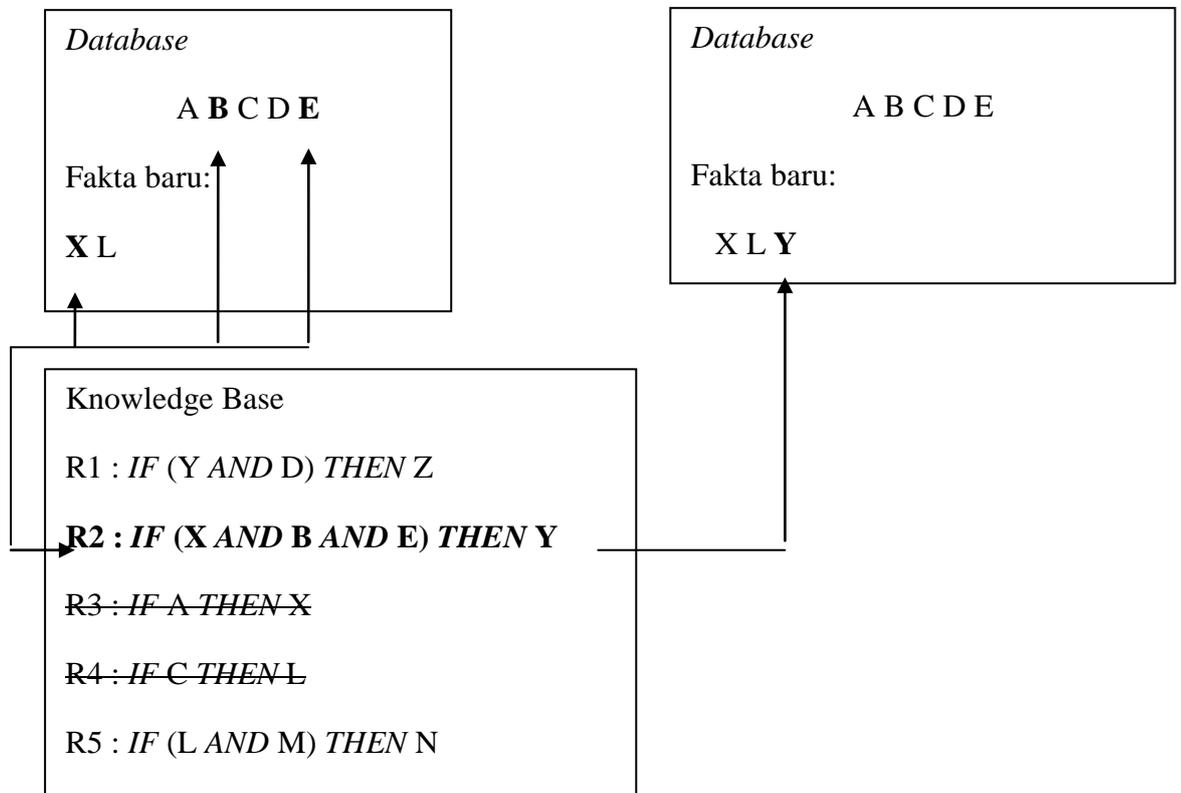
Fakta-fakta : A,B,C,D, dan E bernilai benar

Goal : menentukan apakah Z bernilai benar atau salah

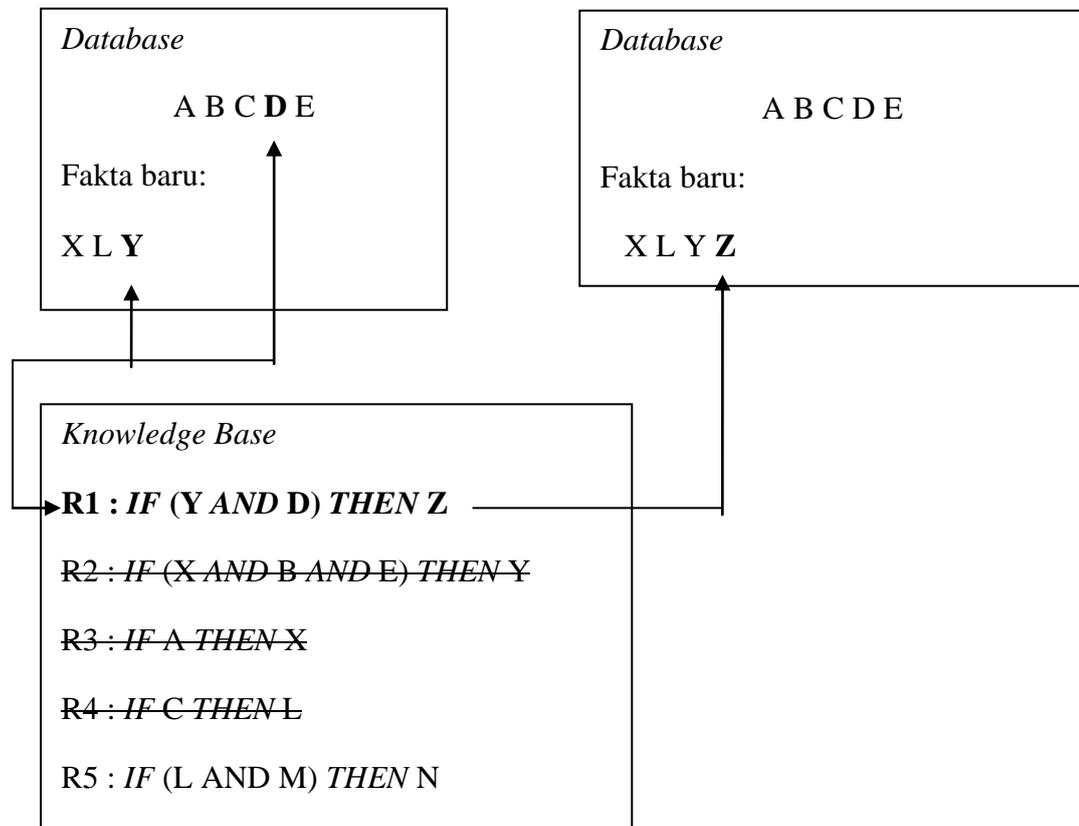
Literasi 1:



Gambar 2.2. Literasi ke-1 Forward Chaining
Sumber: Sutojo,dkk (2011:173).

Literasi ke-2:

Gambar 2.3. Literasi ke-2 Forward Chaining
Sumber: Sutojo,dkk (2011:173).

Literasi ke-3:

Gambar 2.4. Literasi ke-3 Forward Chaining
Sumber: Sutojo,dkk (2011:173).

Sampai disini proses dihentikan karena sudah tidak ada lagi rule yang bisa dieksekusi. Hasil penilaian adalah Z bernilai benar.

2.1.2.4 Tabel Keputusan dan Pohon Keputusan

Tabel keputusan merupakan suatu cara untuk mendokumentasikan pengetahuan. Tabel keputusan merupakan matrik kondisi yang dipertimbangkan dalam pendeskripsian kaidah (Sri Hartati dan Sari Iswanti, 2008:26).

Kaidah produksi dituliskan dalam bentuk jika-maka (*if-then*) menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Berbagai struktur kaidah *if-then* yang menghubungkan obyek atau atribut sebagai berikut:

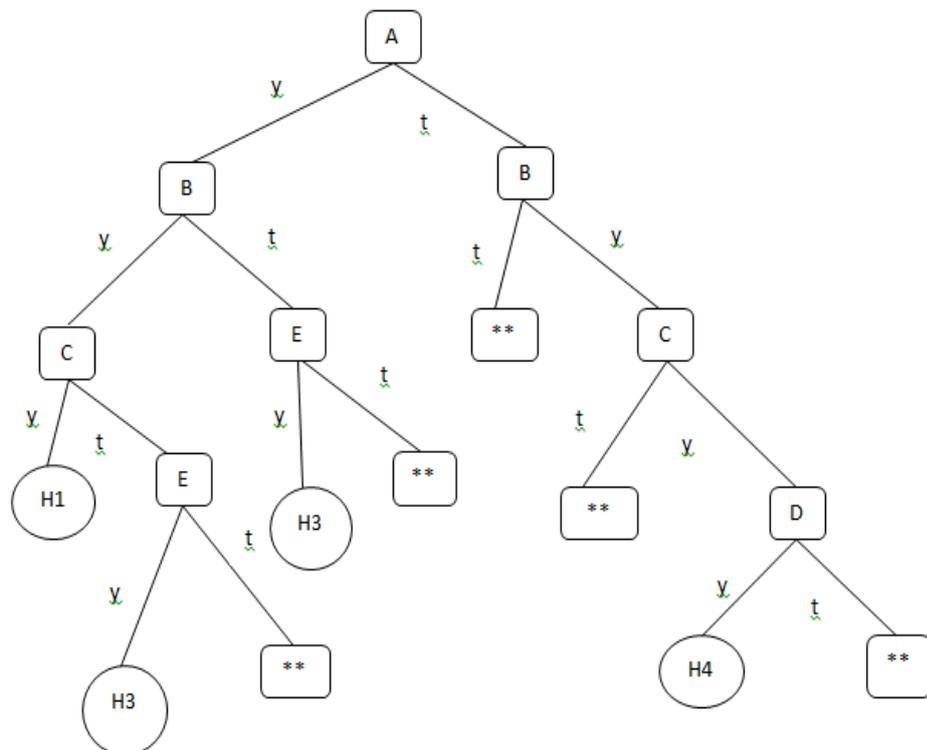
1. *IF* premis *THEN* konklusi
2. *IF* masukan *THEN* keluaran
3. *IF* kondisi *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan
7. *IF* aksi *THEN* reaksi
8. *IF* sebab *THEN* akibat
9. *IF* gejala *THEN* diagnosa

Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk tabel keputusan kemudian dibuat pohon keputusannya. Dari pohon keputusan dapat diketahui atribut atau kondisi yang dapat direduksi sehingga menghasilkan kaidah yang efisien dan optimal. Berikut ini adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan:

Tabel 2.1. Tabel Keputusan

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

Sumber: Hartati, dan Sari Iswanti (2008:32)



Gambar 2.5. Pohon Keputusan
(Sumber: Hartati, dan Sari Iswanti,2008:33)

Keterangan:

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

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

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

D = *evidence* D, H4 = hipotesa 4,

Dilihat dari pohon keputusan gambar 2.2, masing-masing *node* yang mewakili *evidence* tertentu untuk kondisi “y” dan ”t” sudah tidak mengarah pada *evidence* yang sama. Dalam ssi konsultasi hal ini mengandung arti jawaban pengguna yang berbeda, akan mengarah pada pertanyaan yang berbeda pula. Kaidah yang dapat dihasilkan dengan mengacu pohon keputusan gambar 2.2 adalah sebagai berikut:

1. Kaidah 1 : *IF A and B and C THEN H1*
2. Kaidah 2 : *IF A and B and E THEN H3*
3. Kaidah 3 : *IF A and E then H2*
4. Kaidah 4 : *IF D and B and C then H4*

Model representasi pengetahuan kaidah produksi banyak digunakan pada aplikasi sistem pakar karena model representasi ini mudah dipahami dan bersifat deklaratif sesuai dengan jalan pikiran manusia dalam menyelesaikan suatu masalah, dan mudah diinterpretasikan.

2.2 Printer

Printer adalah salah satu *hardware* (perangkat keras) yang terhubung ke komputer dan dialiri listrik. Fungsinya adalah untuk mencetak tulisan, gambar maupun tampilan lain dari komputer ke suatu media kertas atau yang lainnya (Milawati Hartono, Eko Nur Muhammad Irsyad:2016).

Menurut Lim Rusyamsi (2009:1) “Printer merupakan sebuah perangkat keras yang dihubungkan pada komputer yang berfungsi untuk menghasilkan cetakan baik berupa tulisan ataupun gambar dari komputer pada media kertas ataupun yang sejenisnya”.

2.2.1 Jenis Printer

Jenis printer ada tiga macam, yaitu jenis printer *dot matrix*, printer *ink jet*, dan printer *laser jet*.

1. Printer *Dot Matix*

Printer *Dot-Matrix* adalah pencetak yang resolusi cetaknya masih sangat rendah. Selain itu ketika sedang mencetak, printer jenis ini suaranya cenderung keras serta kualitas untuk mencetak gambar kurang baik karena gambar yang tercetak akan terlihat seperti titik-titik yang saling berhubungan. Umumnya, printer jenis *dot-matrix* juga hanya mempunyai satu warna, yaitu warna hitam. Kelebihan lainnya, pita printer *dot-matrix* jauh lebih murah dibandingkan dengan toner (tinta) untuk printer jenis inkjet dan *laserjet*.

2. Printer *Ink Jet*

Printer *ink jet* adalah alat cetak yang sudah menggunakan tinta untuk mencetak dan kualitas untuk mencetak gambar berwarna cukup bagus. Kecepatan mencetak jumlah halaman pada printer *Inkjet* tidak sama, tergantung pada jenis merk printer tersebut. Tetapi pada *inkjet* printer, hasil cetakan lebih lama keringnya jika dibandingkan dengan laser printer. Pada Printer *Inkjet* Merk Canon, Epson, dan HP inkjet semua tipe, jumlah pencetakannya akan selalu diakumulasikan hingga batas tertentu. dan bila sudah mencapai batas yang ditetapkan, maka printer akan *overload* dengan ditandai lampu *LED* menyala secara bergantian (hijau dan *orange* untuk printer canon) atau disebut dengan *blinking* dan bahkan untuk printer merk dan jenis tertentu akan mati total.

3. Printer *Laser Jet*

Printer *laserjet* cara kerjanya hampir sama dengan mesin fotokopi hanya saja *input* data printer laser berasal dari komputer sedangkan *fotocopy* berasal dari *scanning* yang diteruskan ke *DC board*. Daya cetaknya juga cukup cepat bisa mencapai lebih dari 10 lembar per menit bahkan lebih tergantung dari spesifikasi printernya. Kualitas hasil cetak laser printer pun sangat bagus dan tidak mudah hilang walaupun kena air sekalipun karena printer laser menggunakan serbuk *toner* seperti *photocopy*, sehingga mirip sekali dengan aslinya. Selain itu hasil cetakan cepat kering.

2.2.2 Rangkaian Dasar Printer

Rangkaian dasar pada sebuah printer antara lain:

1. *Paper feed* (pengumpan kertas)

Merupakan metode transportasi kertas mulai dari kertas masuk, dibawa ke depan *head* dan dicetak hingga dicetak keluar dengan membawa hasil sesuai dengan *output* yang ada pada layar komputer.

2. *Head Printer*

Adalah alat yang memberikan cetakan permanen pada permukaan halaman.

3. *Carriage* (pembawa) *head printer*

Adalah komponen mekanisme yang membawa pergerakan *head* maju mundur melintasi permukaan halaman kertas untuk mendapatkan hasil cetakan sesuai dengan output yang diinginkan.

4. *Power supply*

Komponen ini berfungsi dan bertanggung jawab dalam mengubah tegangan saluran AC menjadi satu atau lebih tegangan DC yang diperlukan oleh komponen elektronik dan listrik pada printer.

5. *Electronic control package*

Sebuah paket pengontrol elektronik yang merupakan satu kombinasi komponen dan sirkuit elektronik yang menjalankan operasi printer.

2.2.3 Komponen Utama pada Printer Canon ip2770

Komponen-komponen utama pada printer canon ip2770 adalah:

1. Cartridge

Cartridge merupakan komponen printer yang berfungsi menampung tinta dan membawa tinta ke *head* printer untuk membantu proses percetakan.



Gambar 2.6. Cartridge Printer Canon
Sumber : Data Penelitian (2017)

2. Sensor *encoder*

Sensor *encoder* pada printer canon ip2770 terbagi menjadi 2, yaitu *encoder* panjang dan *encoder* bulat. *Encoder* panjang berbentuk pita tipis transparan yang berfungsi untuk proses percetakan pada garis, contoh nya garis pada tabel.



Gambar 2.7. *Encoder Panjang Printer Canon ip2770*
Sumber: Data Penelitian (2017)

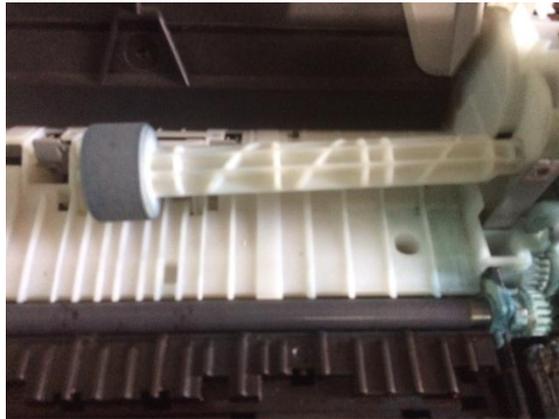
Sedangkan *encoder* bulat berbentuk lingkaran dengan gerigi di bagian dalam, berfungsi sebagai *timing* untuk menarik kertas.



Gambar 2.8. *Encoder Bulat Printer Canon ip2770*
Sumber: Data penelitian (2017)

3. *Roller*

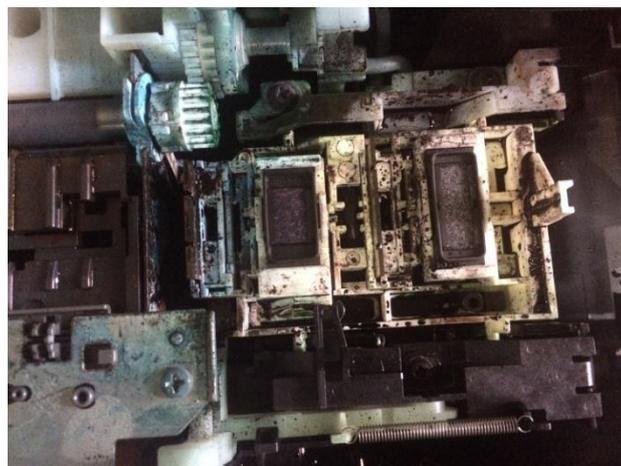
Roller merupakan komponen printer yang berfungsi untuk menarik kertas, jika *roller* mengalami kerusakan maka kertas yang akan dicetak tidak akan masuk kedalam printer atau kertas ditarik lebih dari satu.



Gambar 2.9. *Roller* Printer Canon ip2770
Sumber: Data Penelitian (2017)

4. *Purging*

Purging merupakan komponen printer yang berfungsi untuk memompa tinta dari cartridge printer.



Gambar 2.10. *Purge Unit* Printer Canon ip2770
Sumber: Data penelitian (2017)

5. *Motherboard*

Motherboard berbentuk papan sirkuit yang berfungsi mengendalikan seluruh kerja pada printer, mengendalikan yang dimaksud adalah sebagai alat yang menjembatani antara *driver (software)* dengan komponen yang ada didalam printer.



Gambar 2.11. *Motherboard* Printer Canon ip2770
Sumber: Data Penelitian (2017)

2.2.4 Masalah Umum Pada Printer

Berikut contoh permasalahan umum yang biasa dialami oleh printer, antara lain:

1. Printer tidak bisa hidup

Hal ini disebabkan oleh beberapa hal antara lain:

- a. Kabel power listrik belum terhubung dengan printer atau tidak terpasang dengan baik
- b. Kabel power menuju listrik putus
- c. Printer memang sudah rusak secara fisik

2. Printer hidup tetapi tidak bisa mencetak

Hal-hal yang menyebabkan printer tidak dapat mencetak meskipun printer sudah menyala adalah:

- a. Cartridge atau pita belum terpasang dengan baik
- b. Kertas belum terpasang
- c. Tinta untuk mencetak telah habis

3. Printer mencetak tetapi tidak sampai selesai

Hal-hal yang menyebabkan printer tidak dapat mencetak sampai habis adalah:

- a. Pada saat mencetak, tinta printer habis
- b. Pada saat mencetak tiba-tiba listrik mati
- c. Pada saat mencetak, tiba-tiba program aplikasi tersebut ditutup

4. Hasil cetakan printer cacat

Penyebab hasil cetakan printer menjadi cacat antara lain:

- a. Tidak lengkapnya jaarum pengetuk pada printer, atau memang ada kerusakan pada *device driver* nya.
- b. Gerakan *head* printer tidak lancer akibat kotor atau tersumbat sesuatu

5. Printer tidak terdeteksi oleh sistem

Permasalahan ini dapat diatasi dengan instalasi ulang pada *driver* printer.

6. Lampu indikator pada printer berkedip terus

Permasalahan ini dapat diatasi dengan melakukan riset tinta printer.

2.2.5 Langkah Perbaikan pada Kerusakan Printer

Secara umum langkah-langkah awal yang dapat dilakukan untuk perbaikan printer antara lain:

1. Lepas *cartridge* dengan hati-hati untuk mengecek apakah tinta sudah habis atau belum. Setelah itu lakukan pembersihan pada *head* nya.
2. Membatasi tebal tumpukan kertas sesuai dengan kapasitas yang didukung oleh printer.
3. Menggoyang *toner* printer agar lebih merata dan dapat dipakai kembali, namun tetap bersiap untuk mengisinya dengan yang baru.
4. Usaplah *drum* dengan kain halus untuk membersihkan benda asing yang menempel atau dengan mengganti *drum* jika terdapat lubang kecil pada permukaan *drum*.
5. Pastikan posisi kertas terpasang dengan baik, apabila sudah dilakukan tetapilampu masih menyala kemungkinan sensor kertas printer rusak. Disarankan untuk mengganti sensor printer yang baru.
6. Coba lakukan pengecekan di kabel *cutternya*, apakah ada yang putus atau tidak, dan cek juga apakah sensor gerak masih berfungsi dengan baik atau tidak dan lakukanlah pembersihan sensor dengan menggunakan cairan pembersih printer

2.3 Software Pendukung

Software pendukung merupakan beberapa perangkat lunak yang digunakan untuk mendukung pembuatan sistem pakar dalam penelitian ini. Perangkat lunak tersebut antara lain *undefined modelling language* (uml) sebagai bahasa standar untuk merancang sistem, *hypertext preprocessor* (php), *Hypertext Markup Language* (HTML) sebagai bahasa pemrograman, *xampp* sebagai *web server*, *Adobe Dreamweaver*, dan *MySQL* sebagai *database*.

2.3.1 Unified Modeling Language (UML)



Gambar 2.12. Logo Uml (*Unified Modeling Language*)
(Sumber:<http://staruml.sourceforge.net/image/staruml-logo.jpg>)

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

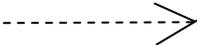
2.3.1.1 Simbol-simbol pada *Unified Modeling Language* (UML)

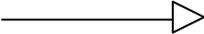
Pada perancangan sistem pakar ini, penulis menggunakan empat macam diagram yang ada pada UML yaitu use case diagram, activity diagram, sequence diagram, dan class diagram. Berikut ini dijelaskan tentang masing-masing diagram yang akan digunakan serta deskripsi dari simbol-simbol yang ada pada masing-masing diagram:

1. *Use Case* diagram

Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi fungsi tersebut. Berikut adalah simbol simbol yang ada pada diagram *use case*:

Tabel 2.2. Simbol pada diagram *use case*

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor
Aktor / actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Asosiasi / <i>association</i> 	Komunikasi antara aktor dengan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri

	sendiri walau tanpa <i>use case</i> tambahan itu
Generalisasi / generalization 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya.

Sumber : Rosa A.S dan M. Shalahuddin (2011:130)

2. Activity diagram

Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3. Simbol pada *activity diagram*

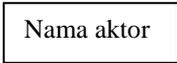
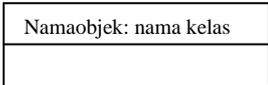
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan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

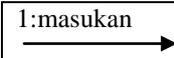
Sumber : Rosa A.S, *et al.* (2011:134).

3. Sequence diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Berikut adalah simbol-simbol yang ada pada diagram sekuen:

Tabel 2.4. Simbol pada sequence diagram

Simbol	Deskripsi
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem yang akan dibuat itu sendiri.
Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i> 1: nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan

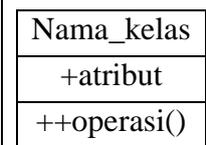
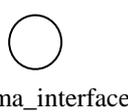
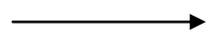
	data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu

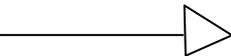
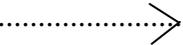
Sumber : Rosa A.S, *et al.* (2011:138).

4. Class diagram

Kelas diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.5. Simbol pada class diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>interface</i> 	Sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi 	Relasi antar kelas dengan makna umum.
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh

	kelas yang lain.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi / <i>agregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber : Rosa A.S, *et al.* (2011:123).

2.3.2 Hypertext Preprocessor (PHP)

PHP adalah kepanjangan dari Hypertext Preprocessor, merupakan sebuah bahasa pemrograman yang berupa kode atau *script* yang bisa ditambahkan kedalam bahasa pemrograman HTML (Milawati Hartono,Eko Nur M Irsyad, 2016:51). PHP sendiri sering digunakan untuk hal merancang, membuat dan memprogram sebuah website.

2.3.2.1 Sejarah PHP

Pada awalnya PHP merupakan kependekan dari *personal homepage* (Situs personal). Php pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama form interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis 2.0 ini, *interpreter* PHP sudah diimplementasikan dalam program C. Di dalam rilis ini juga ikut disertakan modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Kemudian pada Juni 1998, sebuah perusahaan bernama Zend merilis *interpreter* terbaru untuk PHP dan meresmikan perilsan tersebut sebagai php 3.0 dan singkatan PHP diubah menjadi akronim berulang *Hypertext Preprocessing*.



Gambar 2.13. Logo PHP

Sumber: <https://www.quora.com/What-font-is-the-php-logo>

Pada pertengahan tahun 1999, Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0. Versi ini banyak dipakai karena kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0, dalam versi ini, inti dari *interpreter* PHP mengalami perubahan yang besar, versi ini juga memasukkan model

pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman kearah pemrograman berorientasi objek.

2.3.2.2 Keunggulan PHP

PHP berbasis *server side scripting*. PHP sendiri dapat melakukan tugas-tugas yang dilakukan dengan mekanisme CGI seperti mengambil, mengumpulkan data dari *database*, meng-*generate* halaman dinamis, atau bahkan menerima dan mengirim *cookie*. Untuk lebih jelasnya berikut adalah keunggulan dari PHP (Lukmanul, 2014:3):

1. PHP berbasis *server side scripting*

Server side scripting pada PHP dapat bekerja jika ada komponen berikut: PHP parser (CGI atau *server* modul), *web server* contohnya XAMPP, dan *web browser*.

2. *Cross platform*

Artinya dapat digunakan di berbagai sistem operasi, mulai dari linux, windows, mac os dan sistem operasi yang lain.

3. Mendukung banyak *database*

Salah satu fitur yang paling signifikan dalam PHP adalah dukungan penggunaan untuk berbagai database.

4. *On the fly*

PHP sudah mendukung *on the fly*, artinya dengan menggunakan PHP, anda dapat membuat dokumen text, menciptakan image dan flash, juga dapat menciptakan file-file seperti zip, xml, dan banyak lagi.

2.3.3 Hypertext Markup Language (HTML)



Gambar 2.14. Logo *Hypertext Markup Language* (HTML)

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

Hypertext Markup Language adalah bahasa standar yang digunakan untuk menampilkan halaman *web* (Priyanto dan Jauhari., 2014:13). Yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dari halaman web dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mempublikasikan halaman web secara *online*.
4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
5. Menampilkan area gambar (*canvas*) di *browser*.

Penanda perintah dalam HTML biasanya disebut TAG. TAG digunakan untuk menentukan tampilan dari dokumen HTML. Namun perlu diperhatikan bahwa tag `<p>` lebih baik diakhiri dengan tag `</p>`, karena untuk kedepannya, HTML mengharuskan penggunaan *end tag*.

2.3.4 Adobe Dreamweaver



Gambar 2.15. Logo *Adobe Dreamweaver*

Sumber: http://logos.wikia.com/wiki/Adobe_Dreamweaver

Adobe dreamweaver adalah aplikasi desain pengembangan *web* yang menyediakan halaman visual (bahasa sehari-hari yang disebut sebagai *design view*), maksudnya adalah tampilan hasil akhir *web* kita nanti akan sama pada saat tampilan proses perancangan halaman *web*. (Milawati Hartono *et al*, 2016:50)

Aplikasi ini menggunakan kode editor dengan fitur standar seperti *syntax highlighting*, *code completion*, dan *code collapsing* serta fitur lebih canggih seperti *real-time syntax*, *checking* dan *code introspection* untuk menghasilkan petunjuk kode untuk membantu pengguna dalam menulis kode.

Dreamweaver memiliki fitur *browser* yang terintegrasi untuk melihat halaman web yang dikembangkan di jendela pratinjau program sendiri agar konten memungkinkan untuk terbuka di *web browser* yang sedang telah terinstall. Aplikasi ini menyediakan transfer dan fitur sinkronisasi, kemampuan untuk mencari dan mengganti baris teks atau kode untuk mencari kata dan kalimat biasa di seluruh situs.

2.3.5 XAMPP



Gambar 2.16. Logo XAMPP

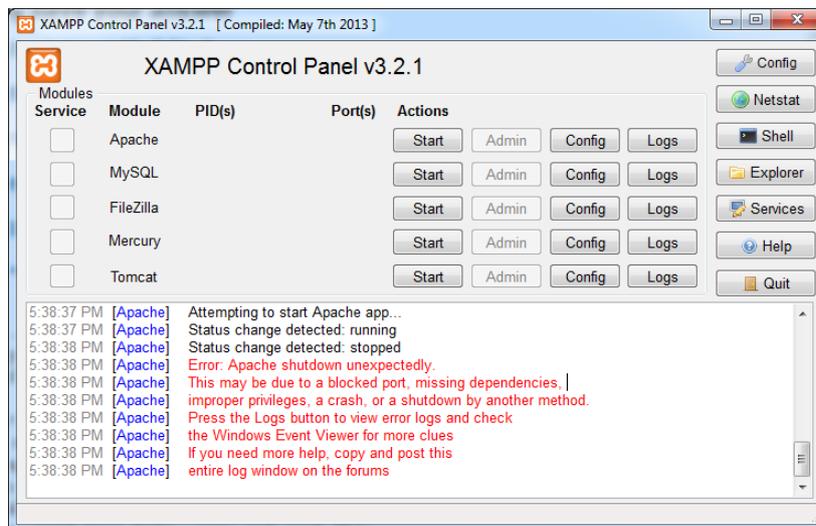
Sumber : <http://cortexcity.com/blog/2015/05/01/aspiring-developers-start-here/>

Xampp adalah software web server dari apache yang didalamnya tertanam *server MySQL* yang didukung oleh bahasa pemrograman PHP. *Web server* ini fungsinya untuk menyimpan aplikasi *web* yang sudah dibuat, agar kemudian dapat diakses melalui internet sehingga setiap perubahan, kecil ataupun besar yang di upload ke *web server* dapat diperiksa apakah sesuai dengan yang sudah diinginkan atau belum (Priyanto dan Jauhari, 2015:125).

Kata XAMPP sendiri berasal dari:

1. X yang berarti *cross platform* karena XAMPP bisa dijalankan di *windows*, Linux, Mac dan sebagainya.
2. A yang berarti Apache sebagai *web server*-nya
3. M yang berarti *MySQL* sebagai *database management system* (DBMS)-nya
4. PP yang berarti PHP dan Perl sebagai bahasa yang didukungnya.

Berikut merupakan tampilan control panel dari xampp:



Gambar 2.17. Tampilan control panel xampp

Sumber: <https://netbeans.org/kb/docs/php/configure-php-environment-windows.html>

2.3.6 PhpMyAdmin

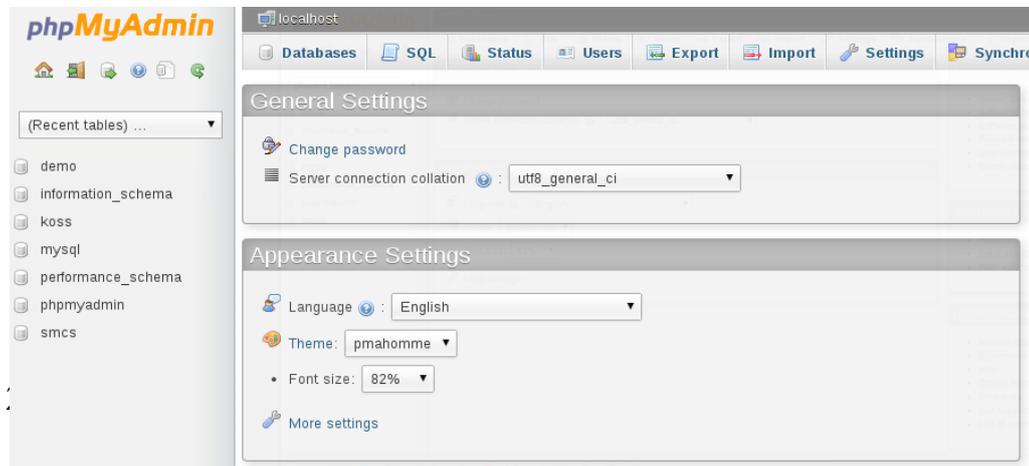


Gambar 2.18. Logo phpMyAdmin

Sumber: MadComs (12:2016)

PhpMyAdmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan *phpmyadmin*, anda dapat membuat *database*, membuat tabel, meng-*insert*, menghapus dan meng-*update* data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan

perintah SQL secara manual. Berikut adalah tampilan halaman utama pada *phpmyadmin*:



Gambar 2.19. Tampilan Halaman utama *phpMyAdmin*
Sumber: MadComs (12:2016)

2.3.7 MySQL



Gambar 2.20. Logo *MySQL*
Sumber : <https://en.wikipedia.org/wiki/File:MySQL.svg>

Database Management System (DBMS) adalah aplikasi yang dipakai untuk mengelola basis data. DBMS biasanya menawarkan beberapa kemampuan yang terintegrasi seperti (Priyanto dan Jauhari, 2015:179):

1. Membuat, menghapus, menambah dan memodifikasi basis data.
2. Pada beberapa DBMS pengelolaannya berbasis *windows* sehingga lebih mudah digunakan.
3. Kemampuan berkomunikasi dengan program aplikasi yang lain. Misalnya dimungkinkan untuk mengakses basis data MySQL menggunakan aplikasi yang dibuat menggunakan PHP.
4. Kemampuan pengaksesan melalui komunikasi antar komputer (*client server*)
5. Tidak semua orang bisa mengakses basis data yang ada sehingga memberikan keamanan bagi data.

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Contoh DBMS lainnya adalah: PostgreSQL (freeware), SQL server, MS Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro dan sebagainya.

Kelebihan dari MySQL adalah gratis, handal, selalu di-update dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL juga menjadi DBMS yang sering di bundling dengan web server sehingga proses instalasi nya jadi lebih mudah.

Tipe data dalam MySQL terdiri dari tipe data numerik, tipe data tanggal dan waktu, dan tipe data string, berikut adalah penjabaran tipe data yang ada yang ada pada MySQL dan deskripsi dari setiap tipe data:

1. Tipe data numerik

MySQL menggunakan seluruh tipe data numerik standar *American National Standards Institute* (ANSI). Berikut ini adalah tipe data numerik yang biasanya digunakan beserta penjelasannya.

Tabel 2.6. Tipe Data Numerik

Tipe Data	Deskripsi
INT	Nilai <i>integer</i> yang bisa bertanda atau tidak. Jika bertanda maka nilai yang diperbolehkan adalah -2147483648 sampai 2147483647, sedangkan jika tidak bertanda maka rentangnya dari 0 sampai 4294967295.
TINYINT	Nilai <i>integer</i> yang sangat kecil. Rentangnya -128-127 untuk yang bertanda dan 0-255 untuk yang tidak bertanda
SMALLINT	Nilai <i>integer</i> yang sangat kecil dengan rentang 31768-32767 untuk yang bertanda dan 0-65535 untuk yang tidak bertanda.
MEDIUMINT	<i>Integer</i> dengan ukuran sedang dengan rentang -8388608 sampai 8388607 atau 0-16777215.
BIGINT	<i>Integer</i> dengan ukuran besar dengan rentang -9223372036854775808 sampai 9223372036854775807 atau 0 sampai 18446744073709551615.
FLOAT (M,D)	Bilangan pecahan dengan panjang (termasuk jumlah desimal) M dan jumlah desimal D. Presisi desimal nya bisa sampai 24 digit.
DOUBLE (M,D)	Bilangan pecahan dengan presisi dua kali lipat. Presisi desimal nya bisa sampai 53 digit.
DECIMAL (M,D)	Bilangan pecahan dan harus didefinisikan M dan D-nya. Setiap desimal membutuhkan tempat 1 <i>byte</i> .

Sumber: Priyanto Hidayatullah dan Jauhari K.K. (2015:181).

2. Tipe data tanggal dan waktu

Berikut adalah tipe data tanggal dan waktu didalam *MySQL*:

Tabel 2.7. Tipe Data Tanggal dan Waktu

Tipe Data	Deskripsi
DATE	Tipe data tanggal dengan format YYYY-MM-DD, antara 1000-01-01 sampai 9999-12-31.
DATETIME	Kombinasi tanggal dan waktu dengan format YYYY-MM-DD HH:MM:SS dan rentang data antara 1000-01-01 00:00:00 sampai dengan 9999-12-31 23:59:59.
TIMESTAMP	Sebuah penanda waktu antara 1 Januari 1970 tengah malam sampai dengan tahun 2007.
TIME	Menyimpan waktu dengan format HH:MM:SS.
YEAR (M)	Menyimpan data tahun dalam format 2 atau 4 digit.

Sumber: Priyanto Hidayatullah, *et al* (2015:182).

3. Tipe Data String

Berikut ini adalah tipe data string yang paling umum didalam *MySQL*:

Tabel 2.8. Tipe Data String

Tipe Data	Deskripsi
CHAR (M)	String dengan ukuran tetap. Ukurannya antara 1-255 karakter.
VARCHAR (M)	String dengan ukuran yang bervariasi antara 1-255 karakter.
TEXT	String dengan ukuran maksimal 65535 karakter.
BLOB	Binary large objects (BLOB) adalah tipe data untuk menyimpan data biner dalam jumlah besar.
ENUM	Enumerasi atau sebuah list.

Sumber: Priyanto Hidayatullah, *et al* (2015:183)

2.3.8 Browser Web



Gambar 2.21. Logo Browser Web (Mozilla Firefox)

Sumber: <https://www.mozilla.org/enUS/styleguide/identity/firefox/branding/>

Browser web adalah software yang digunakan untuk menampilkan informasi dari server web. Software ini kini telah dikembangkan dengan menggunakan grafis user interface (GUI), sehingga pemakai dapat dengan melakukan *point and click* untuk pindah antar dokumen (Betha Sidik dan Husni Iskandar Pohan, 2009:5).

Lynx adalah browser web yang masih menggunakan mode teks, yang akibatnya adalah tidak ada gambar yang dapat ditampilkan. Lynx ini ada pada keluarga DOS dan *nix (keluarga sistem operasi Unix). Akan tetapi perkembangan dari mode teks ini tidaklah secepat browser web dengan GUI.

Dapat dikatakan saat ini banyak browser web yang populer diantaranya adalah Mozilla Firefox dan Google Chrome. Kedua web browser ini bersaing untuk merebut pemakainya, dengan berusaha untuk mendekati standar spesifikasi dokumen HTML yang direkomendasikan oleh W3C.

2.4 Penelitian Terdahulu

Pada penelitian-penelitian sebelumnya yang sudah dilakukan yang digunakan sebagai dasar dan pertimbangan dalam pembuatan penelitian ini antara lain:

Milawati Hartono, Eko Nur Muhammad Irsyad, 2016, SISTEM PAKAR MENDETEKSI KERUSAKAN PRINTER BERBASIS WEB MENGGUNAKAN ALGORITMA FORWARD CHAINING, Membuat sistem pakar mendeteksi kerusakan printer berbasis web dengan menggunakan desain *Adobe Dreamweaver*, bahasa pemrograman PHP dan *database* MySQL. Sistem pakar ini dapat mengidentifikasi berbagai kerusakan printer dan gejala-gejala yang dialami serta mengetahui dan mengerti cara memperbaiki kerusakan tersebut.

Akim Manaor Hara Pardede, Budi Serasi Ginting, 2013, PERANCANGAN SISTEM PAKAR DETEKSI KERUSAKAN PRINTER CANON BERBASIS WEB, Membuat sistem pakar untuk menentukan jenis kerusakan mesin printer *canon* jenis *ink jet* dan intensitas yang tepat bagi penggunaannya, serta menghasilkan pengeluaran berupa saran jenis kerusakan dan intensitas solusi yang tepat bagi penggunaannya berdasarkan gejala yang diinputkan oleh *user*. Sistem pakar ini di implementasikan dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

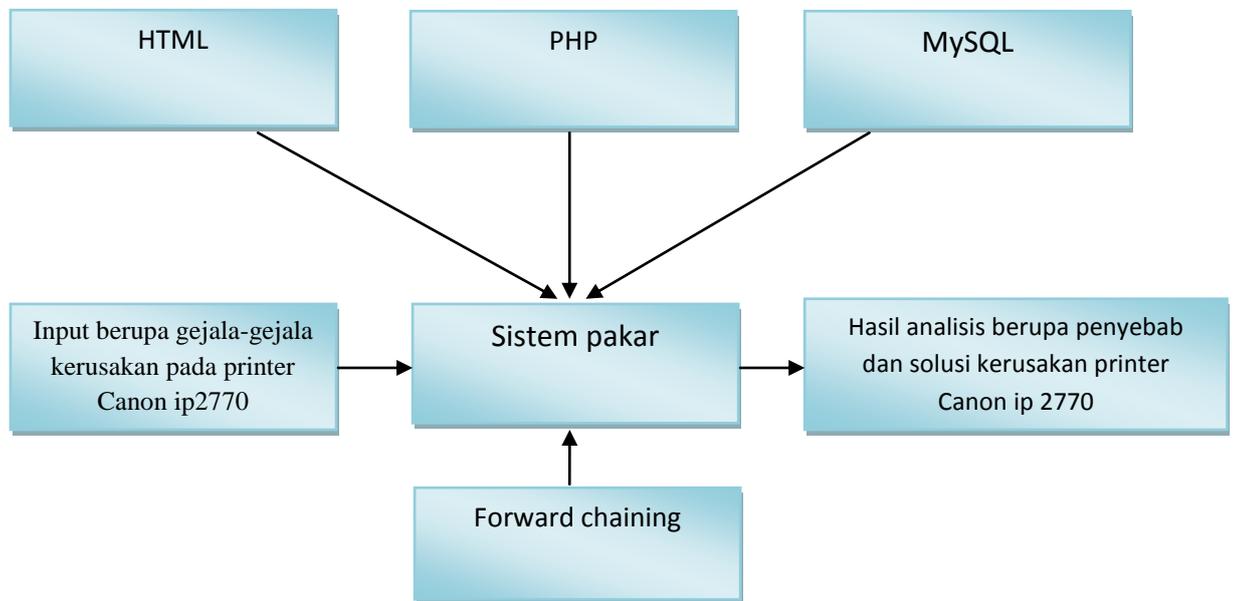
Rosmawati Tamin, 2015, SISTEM PAKAR UNTUK DIAGNOSA KERUSAKAN PADA PRINTER MENGGUNAKAN METODE FORWARD CHAINING, Merancang sebuah sistem yang dapat digunakan untuk menangani kerusakan printer , pengguna aplikasi ini dibuat seolah-olah berhadapan dengan pakar di bidang hardware khususnya printer. Perencanaan sistem ini dibuat dengan menggunakan metode *forward chaining* dan bahasa pemrograman *visual basic*. Hasil penelitian ini mengungkapkan jenis-jenis kerusakan yang terjadi pada sebuah printer serta penanganan dari kerusakan tersebut.

Imam Gunawan, 2013, PERANCANGAN SISTEM PAKAR UNTUK DIAGNOSIS KERUSAKAN *HARDWARE* LAPTOP, Merancang sebuah sistem pakar yang dapat memberikan solusi atas kerusakan hardware yang terjadi berdasarkan data-data yang dipilih pada proses penelusuran, yang terdiri dari pemilihan data jenis komponen hardware yang mengalami kerusakan, dilanjutkan dengan pemilihan data gejala kerusakan yang dialami dan pemilihan data kerusakan yang di dapat.

Bagus Widi Priyono, Muhammad Hasbi, Sri Tomo, SISTEM PAKAR DETEKSI KERUSAKAN KULKAS, Mengembangkan sebuah sistem untuk membantu mendeteksi kerusakan kulkas, pembuatan sistem ini dimaksud sebagai referensi kerusakan kulkas agar bisa diketahui lebih dini tanpa perlu melibatkan seorang pakar teknisi mesin kulkas. Metode yang digunakan untuk sistem pakar ini adalah metode *forward chaining*. Sistem pakar ini akan mendiagnosa gejala kerusakan kulkas kemudian mencari kemungkinan terjadinya kerusakan dan memberikan solusi perbaikan dan penanganannya sehingga dapat membantu dalam perbaikan kerusakan kulkas.

Suriyanti, 2013, APLIKASI SISTEM PAKAR PENDETEKSI KERUSAKAN PRINTER DENGAN CASE BASED REASONING, Merancang sebuah sistem pakar dengan penalaran *case based reasoning* yang menggunakan bahasa pemrograman *Visual Basic.Net* 2008. Sumber pengetahuan sistem diperoleh dengan mengumpulkan penanganan kasus-kasus oleh seorang ahli/pakar. Sistem pakar ini mampu mengidentifikasi kerusakan-kerusakan printer dengan gejala-gejala yang dialami serta mengerti cara memperbaiki kerusakan tersebut.

2.5 Kerangka Pemikiran



Gambar 2.22. Kerangka Pemikiran
Sumber: Data Olahan Penulis (2017)

Dengan gambar kerangka pemikiran diatas menjelaskan tentang prosesnya sistem pakar mendeteksi kerusakan pada printer diawali dengan input gejala-gejala kerusakan pada printer, lalu sistem pakar akan memprosesnya dengan menggunakan metode *forward chaining* berbasis *web* yang sudah dirancang dengan menggunakan bahasa pemrograman php dan html, dan *database* yang digunakan untuk pengolahan data di dalam sistem pakar berupa *create*, *read*, *update*, dan *delete* adalah dengan menggunakan database MySQL. Setelah sistem pakar sudah bekerja dengan baik setelah dimasukkan gejala-gejala kerusakan dan sudah diproses sesuai dengan metode yang ditentukan, maka sistem akan mengeluarkan *output* berupa hasil analisis dan solusi untuk perbaikan printer kepada *user*.