

BAB II

TINJAUAN PUSTAKA

2.1 Teori Dasar

Deskripsi teori paling tidak berisi tentang penjelasan terhadap variabel-variabel yang diteliti melalui pendefinisian, dan uraian yang lengkap dan mendalam dari berbagai referensi, sehingga ruang lingkup, kedudukan dan prediksi terhadap hubungan antara variabel yang akan diteliti menjadi lebih jelas dan terarah (Sugiyono, 2012:58)

2.1.1 Pengertian *Artificial Intelligence*

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat AI, yaitu *Intelligence* adalah kata sifat yang berarti cerdas, sedangkan *Artificial* artinya buatan. Kecerdasan yang dimaksud disini merujuk kepada mesin yang mampu berfikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

Sebagian kalangan menerjemahkan *Artificial Intelligence* sebagai kecerdasan buatan, kecerdasan artifisial, inteligensia artifisial, atau intelegensi buatan. Pada buku ini, istilah *Artificial Intelligence* sengaja tidak diterjemahkan ke bahasa Indonesia karena istilah tersebut sudah sangat akrab bagi orang Indonesia. Begitu

juga dengan singkatan istilah tersebut, yaitu AI, sudah sangat melekat di berbagai media ilmiah maupun non-ilmiah. (Suyanto, 2014:3)

2.1.1.1 Logika Fuzzy

Dua buah *logic* yang kita bahas di atas adalah untuk masalah-masalah yang pasti. Bagaimana merepresentasikan masalah yang mengandung ketidakpastian ke dalam suatu bahasa formal yang dipahami komputer. Untuk masalah seperti ini, kita bisa menggunakan *fuzzy logic*. Beberapa kalangan menerjemahkan istilah ini sebagai logika samar. Teori tentang *fuzzy logic set* atau himpunan samar pertama kali dikemukakan oleh Lotfi Zadeh sekitar tahun 1965 pada sebuah makalah yang berjudul *fuzzy set* setelah itu, sejak pertengahan 1970-an, para peneliti Jepang berhasil mengaplikasikan teori ini ke dalam berbagai permasalahan praktis. (Suyanto, 2014:99)

1. Metode *Tsukamoto*

Pada metode *Tsukamoto*, setiap konsekuensi pada aturan yang berbentuk *IF-Then* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton, sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan α -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

2. Metode *Mamdani*

Metode ini sering juga dikenal dengan nama metode *Max-Min*. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Untuk mendapatkan *output*, diperlukan 4 tahapan :

1. Pembentukan himpunan *fuzzy*
2. Aplikasi fungsi implikasi (aturan)
3. Komposisi aturan
4. Penegasan (*defuzzy*)

3. Metode *Sugeno*

Penalaran dengan metode *sugeno* hampir sama dengan penalaran *mamdani*, hanya saja *output* (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau permasalahan *linier*. Metode ini diperkenalkan oleh *Takagi-sugeno kang* pada tahun 1985.

2.1.1.2 Jaringan saraf tiruan (*Artificial neural network*)

Jaringan Saraf Tiruan (JST) adalah merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf tiruan ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Jst adalah proses tersebar paralel (*paralel distributed processor*) yang sangat besar yang memiliki kecenderungan untuk menyimpan pengetahuan

yang bersifat pengalaman dan membuatnya siap digunakan. Jst menyerupai otak manusia dalam dua hal, yaitu: pengetahuan diperoleh jaringan melalui proses belajar; kekuatan hubungan antar sel syaraf (*neuron*) yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan.

JST mempunyai struktur tersebar parallel yang sangat besar dan mempunyai kemampuan belajar, sehingga bisa melakukan *generalization* atau diterjemahkan sebagai generalisasi, yaitu bisa menghasilkan *output* yang benar untuk *input* yang belum pernah dilatihkan. Dengan kemampuan kedua pemrosesan informasi ini, Jst mampu menyelesaikan masalah-masalah yang sangat kompleks. (Suyanto, 2014:170)

2.1.2 Sistem pakar

Sri Hartati dan Sari Ismawanti (2008:1) Sistem pakar (*Expert System*) adalah sistem yang berusaha mengadopsi pengetahuan manusia kekomputer, agar komputer dapat menyelesaikan masalah yang seperti dilakukan manusia.

Sri Kusumadewi (2007:109) Sistem pakar adalah suatu sistem yang bisa menyalin atau meniru kemampuan seorang pakar.

Pengertian lain menurut Sri Kusumadewi (2007:109) Sistem pakar adalah modal prosedur yang berkaitan dalam suatu domain tertentu, yang mana tingkat keahlian dapat dibandingkan dengan keahlian seorang pakar. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para ahli. Dengan sistem pakar, orang awampun dapat menyelesaikan masalah yang cukup rumit, yang sebenarnya hanya dapat

diselesaikan dengan buatan oleh para ahli, sistem pakar ini juga akan membantu aktifitasnya sebagai asisten yang sangat berpengalaman.

Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Sistem pakar akan memberikan pemecahan suatu masalah yang didapat dari dialog dengan pengguna. Dengan bantuan sistem pakar seseorang yang bukan pakar/ahli dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar. (Sutojo.T, dkk, 2011:13)

2.1.3 Bentuk Sistem Pakar

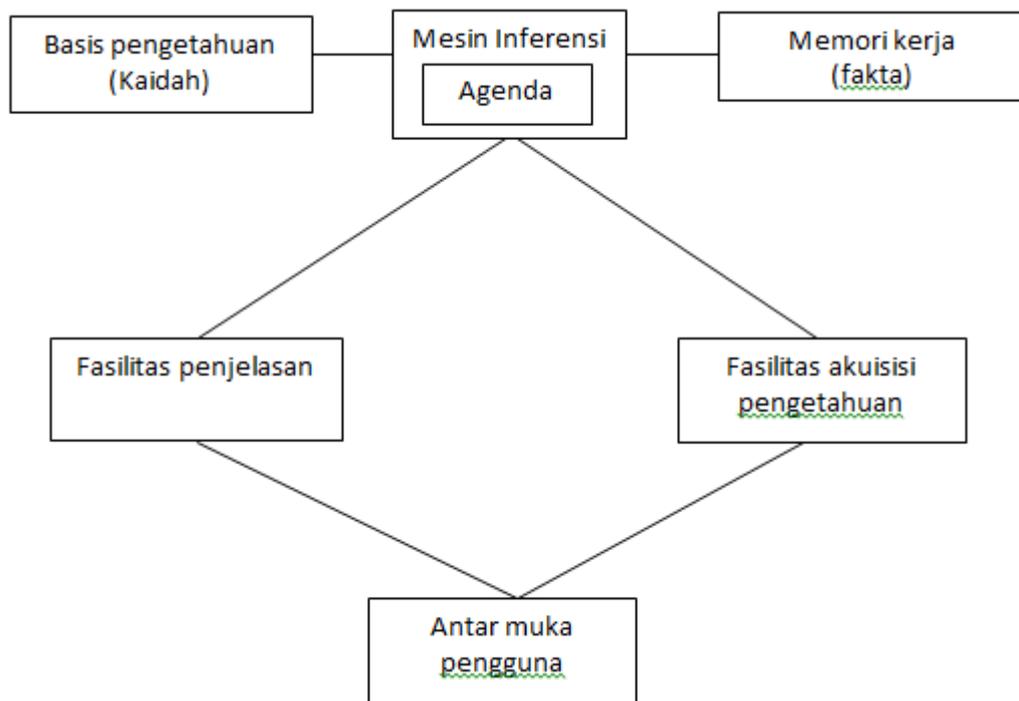
Menurut ada empat bentuk sistem pakar yaitu sebagai berikut:

1. Berdiri sendiri. Sistem pakar jenis ini merupakan *software* yang lainnya.
2. Tergabung. Sistem pakar jenis ini merupakan bagian program yang terkandung di dalam suatu algoritma (konvensional), atau merupakan program dimana didalamnya memanggil algoritma subrutin lain (konvensional)
3. Menghubungkan ke *software* lain. Bentuk ini biasanya merupakan sistem pakar yang menghubungkan ke suatu paket program tertentu, misalnya dengan DBMS
4. Sistem mengabdikan. Sistem pakar merupakan bagian dari komputer khusus yang dihubungkan dengan suatu fungsi tertentu. Misalnya sistem pakar yang digunakan untuk membantu menganalisis data radar.

2.1.4 Struktur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu:

- a. Lingkungan pengembangan (*Development Environment*), digunakan untuk memasukkan pengetahuan pakar kedalam lingkungan sistem pakar.
- b. Lingkungan konsultasi (*Consultation Environment*), digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar.



Gambar 2.1 Struktur sistem pakar
(Sumber: Hartiti, Sari Iswanti 2008:4)

2.1.5 Manfaat Sistem Pakar

Menurut Sutojo T, dkk(2011:160) Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, diantaranya:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia.
2. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seseorang.
5. Dapat beroperasi dilingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.

2.1.6 Komponen Sistem Pakar

Hartono Milawati, Eko Nur Irsyad (2016:101) Menyatakan tentang komponen sistem pakar adalah:

1. Antar muka pengguna (*User Interface*)

Antar muka pengguna merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi.

2. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan penyelesaian masalah.

3. Akuisisi Pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan kedalam program komputer.

4. Mesin Inferensi

Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace* untuk memformulasikan kesimpulan. Dalam perancangan sistem pakar ini penulis menggunakan metode runut maju (*forward chaining*). Merupakan proses peruntutan yang dimulai dengan menampilkan kumpulan data atau fakta yang menyakinkan menuju kesimpulan akhir.

5. Memori Kerja

Merupakan bagian dari sistem pakar yang menyimpan fakta-fakta yang diperoleh saat dilakukan proses konsultasi.

6. Fasilitas Penjelasan

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar.

7. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis, meningkatkan kinerja dan kemampuan yang semua tersebut adalah penting dalam pembelajaran terkomputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang akan dialaminya.

2.1.7 Runut Maju (*Forward Chaining*)

Tutik, dkk (2009) Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada berupa data *real* lalu bergerak maju melalui premis-premis dan penggabungan *rule* untuk menghasilkan suatu kesimpulan (*botton up reasoning*) atau tujuan. *Forward chaining* bekerja dari suatu masalah menu kepada solusinya (dalam Rosmawati Tamin, 2015:41).

Forward chaining merupakan metode inferensi yang melakukan penalaran dari suatu masalah kesolusinya. Jika klausa premis sesuai dengan situasi (bernilai *TRUE*), maka proses menyatakan konklusi. *Forward chaining* adalah *data-driven* karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan *tree* yang lebar dan tidak dalam, maka gunakan *forward chaining*.

Sistem yang dapat dicari dengan metode *forward chaining*

1. Sistem yang direpresentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari *rule-rule* yang berkorespondensi dengan kondisi bagian *IF*
3. Setiap *rule* dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian *THEN*. Kondisi ini ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang ditambah ke sistem akan diproses. Jika ditemui suatu kondisi dari konklusi yang diminta, sistem akan kembali ke langkah 2 dan

mencari *rule-rule* dalam *knowledge base* kembali, jika tidak ada konklusi baru, sesi ini berakhir.

Runut maju merupakan proses peruntutan yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju konklusi akhir. Runut maju bisa juga disebut sebagai penalaran *forward* (*forward reasoning*) atau pencarian yang dimotori data (*data driven search*). Jadi dimulai dari premis-premis atau informasi masukan (*if*) dahulu kemudian menuju konklusi atau *drived information* (*then*) atau dapat dimodelkan sebagai berikut:

IF (Informasi masukan)

THEN (Konklusi)

Informasi masukan dapat berupa data, bukti, temuan, atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan, atau diagnosis. Sehingga jalannya penalaran runut maju dapat dimulai dari data menuju tujuan, dari bukti menuju hipotesa, dari temuan menuju penjelasan, atau pengamatan menuju diagnosa

Hartati, dan Sari Iswanti, (2008:25) Menjelaskan bahwa kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. 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

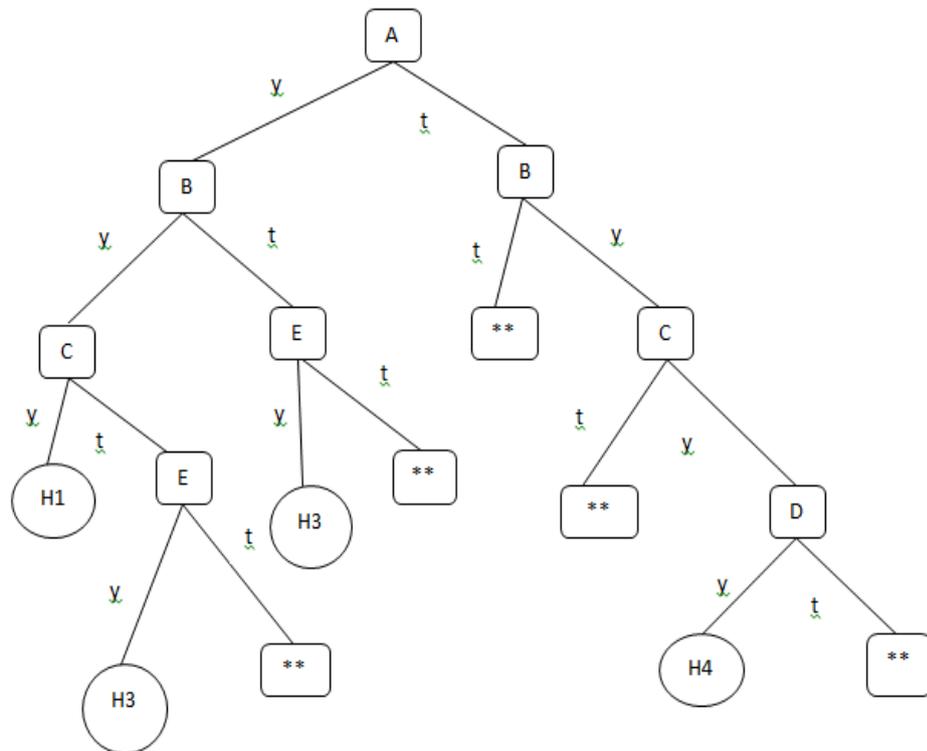
3. *IF* kondisi *THEN* tindakan
4. *IF* antesenden *THEN* konsekuen
5. *IF* data *THEN* hasil
6. *IF* tindakan *THEN* tujuan
7. *IF* aksi *THEN* reaksi
8. *IF* sebab *THEN* akibat
9. *IF* gejala *THEN* diagnosa

Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk table keputusan kemudian dibuat pohon keputusannya. Berikut ini adalah contoh penyajian dalam bentuk table keputusan dan pohon keputusan (Hartati, dan Sari Iswanti, 2008:26)

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)



Gambar2.2Pohon Keputusan
(Sumber: Hartati, dan Sri 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 sesi 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 Variabel Penelitian

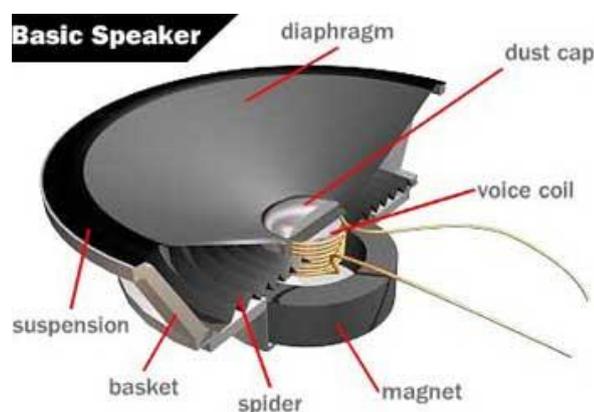
Variabel penelitian merupakan segala sesuatu yang berbentuk apa saja yang ditetapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulan. (Sugiyono, 2012:38). Objek yang digunakan pada penelitian ini adalah Speaker dan variabel yang ditetapkan yaitu kerusakan Speaker.

2.2.1 Speaker

PT FOTER, (2016: 1) Speaker adalah Suatu alat atau *transduser* yang digunakan untuk merubah gelombang listrik menjadi gelombang getaran *electro magnetic* yang akan menghasilkan gelombang suara yang bisa di dengar oleh gendang telinga.

1. Komponen utama

Komponen utama merupakan komponen-komponen penting yang harus ada dalam sebuah perangkat speaker. Tanpa komponen ini. Perangkat speaker tidak dapat bekerja menjalankan fungsinya. Adapun beberapa komponen dari speaker dan fungsinya antara lain adalah:



Gambar 2.3 Speaker
(Sumber: PT FOSTER, 2016:1)

1. *Basket* atau Frame adalah kompoenen speaker yang berfungsi untuk melindungi seluruh bagian dalam speaker. Bahan yang digunakan biasanya terbuat dari Besi logam, plastik atau bahan campuran lainnya.



Gambar 2.4Frame
(Sumber: PT FOSTER, 2016:1)

2. *Spider* adalah komponen yang berfungsi untuk menghasilkan gelombang tekanan yang diakibatkan oleh gerakan udara di sekitarnya dari pergerakan *voice coil*.



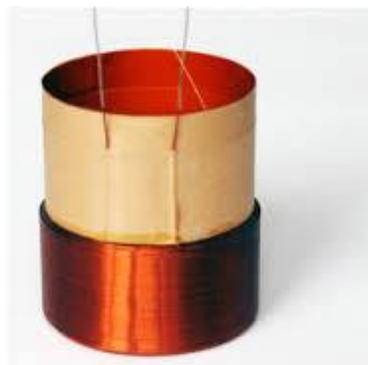
Gambar 2.5Spider
(Sumber: PT FOSTER, 2016:1)

3. Magnet adalah komponen yang berfungsi melakukan induksi sehingga menghasilkan medan magnet.



Gambar 2.6Magnet
(Sumber: PT FOSTER, 2016:2)

4. *Voice Coil* adalah komponen yang berfungsi mengubah dan mengalirkan energi listrik menjadi energi gerak menuju ke cone paper dan spider, perubahan yang terjadi dalam medan magnet speaker menyebabkan gerakanya kumparan yang diakibatkan oleh interaksi antara kumparan dengan medan magnet.



Gambar 2.7Voice Coil
(Sumber: PT FOSTER,2016:2)

5. *Dust cap* atau Center Cap adalah komponen yang digunakan untuk menutup lobang yang ada pada *voice coil* sehingga benda – benda tidak dapat masuk kedalam rongga *voice coil*.



Gambar 2.8Dust cap
(Sumber: PT FOSTER,2016:2)

6. *Diaphragm* atau Cone paper adalah komponen yang berfungsi untuk menerima proses induksi dari magnet yang kemudian menghasilkan bunyi yang diakibatkan oleh getarannya atau induksi.



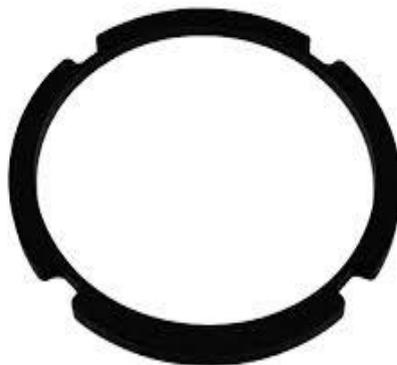
Gambar 2.9Diaphragm
(Sumber: PT FOSTER,2016:2)

7. Terminal adalah komponen yang digunakan untuk penghubung kabel listrik dan biasanya terminal terdiri dari posisi terminal positif dan terminal negatif.



Gambar 2.10Terminal
(Sumber: PT FOSTER,2016:2)

8. *Suspension* atau gasket adalah penahan dari cone paper agar tidak bergeser selama terjadi getaran.



Gambar 2.11Suspension
(Sumber: PT FOSTER, 2016:2)

9. *Yoke* berguna untuk tempat berkumpulnya medan magnet dan membuat Voice coil bergerak.



Gambar 2.12Yoke
(Sumber: PT FOSTER, 2016:3)

10. *Plate* untuk merekatkan antara Magnet dengan *Frame*



Gambar 2.13Plate
(Sumber: PT FOSTER, 2016:3)

2.2.2 Kerusakan Speaker

PT FOSTER, (2016:9) Speaker dapat menjalankan fungsinya sebagai alat keluaran suara karena di dalamnya terdapat banyak komponen yang membutuhkan perawatan dan pemeliharaan yang baik.

1. *No sound*

Lilitan *voice coil* ada yang putus, proses solder yang kurang bagus, terminal dengan solder ada yang terputus, magnet yang tidak diberikan medan magnet.

Solusi, ganti *voice coil*

2. *Sound Precision NG* (Tekanan suara di beberapa frequency problem)

Pada permasalahan ini, proses pemberian medan magnet pada magnet tidak optimal, posisi *voice coil* agak miring (kurang center), bahan perekat atau lem yang digunakan kurang cepat kering. Salah menggunakan komponen.

Solusi, penambahan lapisan *flugs* pada lapisan terminal

3. *Abnormal Sound* (perbedaan suara)

Permasalahan ini dapat disebabkan oleh adanya magnet yang pecah dan masuk ke dalam rongga *voice coil*, *voice coil* tidak bulat, posisi *yoke* bergeser, spider bersentuhan dengan *frame* dan posisi *cone paper* miring.

Solusi, *recharger magnet*.

4. *Appearance Defect* (Problem secara visual)

Permasalahn ini dapat saja terjadi secara visual, *frame* pecah, *cone paper* atau *center cap* berjamur, magnet pecah, perketat menempel pada bagian speaker yang menyebabkan kotor, bekas solder menempel dan *frame* bengkok

Solusi, perbaiki metode penyolderan.(PT FOSTER, 2016)

2.3 Software Pendukung

Software pendukung merupakan beberapa perangkat lunak yang digunakan untuk mendukung pembuatan sistem pakar dalam peelitian ini. Perangkat lunak tersebut antara lain: *UML(Unifield Modelling Language)*, *PHP*, *Adobe Dreamweaver*, *XAMPP*, *Domain name Url*, *MySQL*

2.3.1 Unified Modeling Language (UML)



Gambar 2.14 Uml (*Unified Modeling Language*)
(Sumber:<http://staruml.sourceforge.net/image/staruml-logo.jpg>)

Unified Modeling Language (UML) adalah salah standar bahasa yang banyak digunakan didunia industri untuk mendefenisikan *requirement*, membuat analisis dan desain (Rosa A.S, M. Shalahuddin, 2011:117)

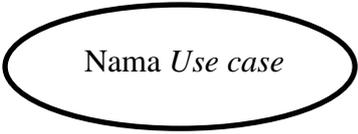
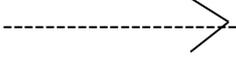
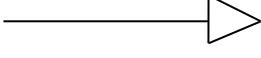
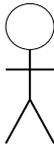
Paling tidak ada tiga karekter penting yang melekat pada UML, yaitu sketsa, cetak biru dan bahasa pemrograman. Sebagai sebuah sketsa, UML bisa berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dari sistem. UML bisa juga berfungsi sebagai cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bisa diketahui informasi penting tentang *coding* program (*forward engeneering*) atau bahkan membaca program dan menginterpretasikannya kembali ke dalam diagram (*reverse engeneering*), sebagai bahasa pemrograman. UML dapat menterjemahkan diagram yang ada di UML menjadi code program yang siap untuk dijalankan.

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

1. Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. (Rosa A.S, M. Shalahuddin, 2011:130)

Tabel 2.2 Simbol pada diagram *use case*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau actor</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antara actor dengan <i>use case</i> yang berpartisipasi pada <i>use case</i></p>
<p>Ekstensi / <i>Extend</i></p> 	<p>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</p>
<p>Generalisasi / <i>Generalization</i></p> 	<p>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</p>
<p>Aktor / <i>Actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat, jadi walaupun simbol dari actor adalah gambar orang. Biasanya dinyatakan menggunakan kata benda dinyatakan menggunakan kata benda di awal frase nama actor</p>

Sumber: Rosa A. S, M. Shlahudin (2011:131)

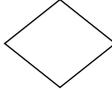
2. Activity Diagram

Diagram aktivitas atau *activity* diagram 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 actor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujiannya. (Rosa A.S, M. Shalahuddin, 2011:134)

Tabel 2.3 Simbol pada *activity* diagram

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

Sumber: Rosa A. S, M. Shlahudin (2011:134)

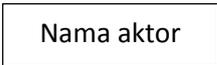
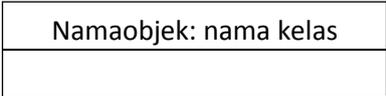
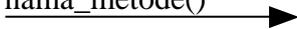
3. *Sequence* Diagram

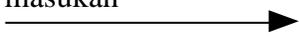
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyak diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use*

case yang telah didefinisikan interaksinya jalannya pesan sudah dicakup pada diagram sukuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. (Rosa A.S, M. Shalahuddin, 2011137)

Tabel 2.4 Simbol pada *sequence* diagram

Simbol	Deskripsi
Garis hidup / <i>Lifeline</i> 	Menyatakan kehidupan suatu objek
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem yang akan dibuat itu sendiri
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 lin, arah panah mengarah pada objek yang dibuat
Pesan tipe <i>call</i> 1: <u>nama_metode()</u> 	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 lainya, arah panah mengarah pada objek yang dikirim
Pesan tipe <i>return</i>	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

Tabel 2.4Lanjutan

Sumber: Rosa A. S, M. Shlahudin (2011:138)

4. *Class* diagram

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

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan

2. Kelas yang menangani tampilan sistem

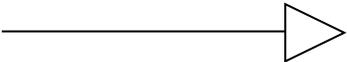
Kelas yang mendefinisikan dan mengatur tampilan kepemakai

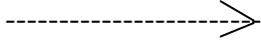
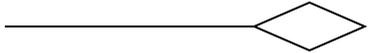
3. Kelas yang diambil dari pendefenisian *use case*

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*

4. Kelas yang diambil dari pendefinisian data
5. Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kestuan yang diambil maupun akan disimpan kebasis data. (Rosa A.S, M. Shalahuddin, 2011:122)

Tabel 2.5 Simbol pada *class diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem
<p>Antarmuka / <i>Interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
<p>Asosiasi berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain
<p>Asosiasi</p> 	Relasi antar kelas dengan makna umum

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>)

Tabel 2.5 Lanjutan

Sumber: Rosa A. S, M. Shlahudin (2011:123)

2.3.3 Hypertext Preprocessor (PHP)



Gambar 2.15PHP

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

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

2.3.3.1 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*(F1), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/F1. Dengan perilisan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk mengembangkan PHP.

Pada November 1997, dirilis PHP/F1 2.0. Pada rilis 2.0 ini, *interpreter* PHP sudah diimplementasikan dalam program C. Didalam rilis ini juga ikut disertakan modul-modul ekstensi yang meningkatkan kemampuan PHP/F1 secara signifikan. Kemudian pada juni 1998, sebuah perusahaan bernama Zend merilis *interpreter* terbaru untuk PHP dan meresmikan perilisan tersebut sebagai PHP 3.0 dan singkatan PHP menjadi akronim berulang *Hypertext Preprocessing*.

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 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 kedalam PHP untuk menawab perkembangan bahasa pemrograman kearah pemrograman berorientasi objek.

2.3.3.2 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 PHP:

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. *Command Line Scripting Pada PHP*

Menggunakan PHP dengan dan tanpa *web server* atau *browser*. Dengan pemanfaatan PHP pada *command line*.

3. PHP dapat membuat aplikasi dekstop

PHP mungkin bukan baasah yang ukup baik untuk suatu aplikasi di desktop karena untuk bagian tampilan (*user interface*) agak sedikit sulit diimplementasikan dengan PHP saja, sedangkan *userinterface* merupakan salah satu point utama ketika kita membuat suatu aplikasi desktop.

4. Digunakan untuk berbagai macam *platform OS*

Linux, Unix (including HP-UX, Solaris *and* Open BSD), Microsoft Windows, Mac OS X, RISC OS, dan lainnya. (Hidayatullah Priyanto, Jauhari Khairul Kawistara, 2015:234)

2.3.4 Adobe Dreamweaver



Gambar 2.16 Adobe Dreamweaver

(Sumber: <http://www.adobe.com/products/dreamweaver.html>)

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*.

Aplikasi ini menggunakan kode editor dengan fitur standar seperti *syntaxhighlighting*, *code completion*, dan *codecollapsing* serta fitur canggih seperti

real-time syntax, checking dan *code introspection* untuk menghasilkan petunjuk kode untuk membantu pengguna dalam menulis kode.

Dreamweaver memiliki fitur dari *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, kemampun untuk mencari dan mengganti baris teks atau kode untuk mencari kata dan kalimat biasa diseluruh situs.

Adobe dreamweaver merupakan program penyunting halaman *web* *adobe system*, yang dulu dikenal sebagai *Macromedia Dreamweaver* keluaran *macromedia*. Program ini banyak digunakan oleh pengembang *web* karena fiturnya yang menarik dan kemudahan penggunaannya. (Maudi Meiska Firstiar, dkk, 2014:102)

2.3.5 XAMPP



Gambar 2.17Xampp

(Sumber:<https://brandeps.com/logo/X/Xampp-10>)

XAMPP adalah sebuah *software web server apache* yang didalamnya sudah tersedia *database Mysql* dan dapat mendukung pemrograman *Php*. Xampp merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *linux* dan *windows*. Keuntungan lainnya adalah cuman menginstall satu kali sudah tersedia *Apache web server, Mysql database server, php support* (PHP 4 dan PHP 5) dan beberapa modul lainnya. (Februarianti Herny, Eri Zuliarso, 2012:129)

XAMPP adalah salah satu paket instalasi *apache*, *PHP*, dan *Mysql* secara instant yang dapat masuk kedalam halaman administrator. *User* (pengguna) hanya memiliki hak akses, seperti melihat koleksi buku-buku digital yang ada, mendownload buku digital, mendaftar sebagai member baru, dan menuliskan pesan kepada administrator melalui menu “*Contact Us*” (Wahana,2009:30).

2.3.6 Domain name/Url-Uniform Resource Lacator



Gambar 2.18 Domain name/URL
(Sumber: <https://www.hostinger.co.id/>)

Pengertian nama domain atau biasa disebut *Domain Name* atau *Url* adalah alamat unik didunia internet yang digunakan untuk mengidentifikasi sebuah *website*,

atau dengan kata lain *domain name* adalah alamat yang digunakan untuk menemukan sebuah *website* pada dunia internet. Contoh <http://www.detik.com>, www.unsrat.ac.id. Nama domain diperjual belikan secara bebas di internet dengan status sewa tahunan.(Pangerapan Octavia Ritha, 2013:765)

2.3.7MySQL



Gambar 2.19MySQL

(Sumber:<https://mysql.com/about/legal/logos.html>)

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License).Setiap pengguna dapat secara bebas menggunakan Mysql, Namun perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.Mysql sebnarnya turunan salah satu konsep utama dalam basisdata yang ada sebelumnya. (Februarianti Herny, Eri Zuliarso, 2012:128)

Database Management System (DBMS) adalah aplikasi yang dipakai untuk mengelola basis data. DBMS biasanya menawarkan beberapa kemampuan yang terintegrasi seperti:

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

MySQL adalah sebuah perangkat lunak manajemen berbasis data *Sql Database Management System* atau DBMS dari sekian banyak DBMS seperti Oracle, MS SQL, PostgreSQL dan lainnya.

2.4 Penelitian Terdahulu

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

RIVO YOUFLY RURUT, 2016, RANCANG BANGUN ALAT PENGONTROL POLARIS LOUD SPEAKER BERBASIS ATMEGA8535. Pada perancangan sistem tester Polaris audio komponen-komponen dirangkai pada satu papan PCB agar lebih efisien dan praktis. Komponen utama dari rangkaian yang

pertama *Microcontroller* ATmega 8535. Dengan alat ini dapat memeriksa *polaritis* dari speaker, apakah itu pertama *feedback* atau non *feedback* (seiring keliru disebut *fase*, yang juga waktu dan frekuensi yang terkait).

AKIM MANAOR HARA PARDEDE, 2013, PERANCANGAN SISTEM PAKAR DETEKSI KERUSAKAN PRINTER CANON BERBASIS WEB. Sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya dapat dipecahkan oleh orang pakar dalam hal bidang tertentu. Aplikasi sistem pakar ini dibuat untuk menentukan jenis kerusakan mesin printer canon jenis ink jet dan intensitas yang tepat bagi penggunaannya menghasilkan keluaran berupa saran jenis kerusakan dan intensitas solusi yang tepat bagi penggunaannya berdasarkan gejala yang diinput oleh user. Data dianalisa dengan menggunakan metode *backward chaining*, dengan bahasa pemrograman PHP dan database MySQL.

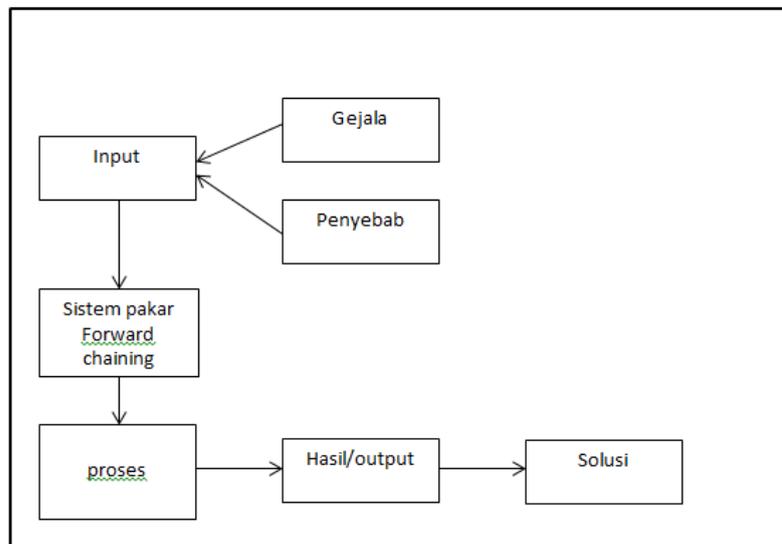
ROSMAWATI TAMIN, 2015, SISTEM PAKAR UNTUK DIAGNOSA KERUSAKAN PADA PRINTER MENGGUNAKAN METODE FORWARD CHAINING. Perancangan sistem dengan membuat *knowledge base* menggunakan *decision tree* dan aturan *if-then* sebagai representasi pengetahuan. Sistem dibuat dengan metode *forward chaining* dan bahasa pemrograman visual basic. Pengujian aplikasi juga dilakukan untuk mengetahui kurasi dan variasi serta *user friendly* dan fleksibilitas sistem. Dari keseluruhan pengujian ini dapat disimpulkan bahwa program sudah cukup baik walaupun jenis kerusakannya yang dihasilkan belum lengkap karena pada sistem ini hanya mendeteksi 15 jenis kerusakan secara umum.

SARTINI, 2015, SISTEM PAKAR IDENTIFIKSI KERUSAKAN HARDWARE HANDPHONE DENGAN MENGGUNAKAN METODE FORWARD CHAINING. Basis pengetahuan untuk membuat sistem pakar identifikasi gejala mulai dari data penginputan data pertanyaan, data jawaban, pendaftaran pengguna, hingga proses konsultasi. Metodologi yang digunakan adalah membuat aturan (*rule*) dengan teknik pelacakan kedepan (*forward chaining*) dan menggunakan topology pencarian pertama yang terbaik. Dengan *website* sistem pakar ini diharapkan dapat menyediakan informasi tentang gejala kerusakan *handphone*.

ACHMAD KARIDI, BUDI RAHMAN, SYAHIB NATARSYAH, 2014, SISTEM PAKAR DETEKSI KERUSAKAN PLC (POWER LINE CARRIER) DENGAN METODE FORWARD CHAINING. Sistem pakar yang dibangun adalah untuk mendeteksi kerusakan PLC dengan menerapkan metode *forward chaining*. Sistem ini menggunakan bahasa pemrograman *Delphi* yang *databasenya* dengan *Microsoft acces*. Pada penelitian yang telah dilakukan, gejala kerusakan PLC (Power Line Carrier) dicoba untuk dirangkum dan dianalisis.

2.5 Kerangka Pemikiran

Menurut Sugiono, (2012:63) Setelah sintesa atau kesimpulan sementara dapat dirumuskan maka selanjutnya disusun kerangka berfikir. Kerangka berfikir yang dihasilkan dapat berupa kerangka berfikir yang *assosiatif* atau hubungan maupun *komparatif* atau perbandingan.



Gambar 2.20 Kerangka pemikiran
(Sumber: Data Penelitian, 2017)

Dengan gambar ini diatas menjelaskan tentang prosesnya sistem pakar mendeteksi kerusakan pada speaker. Kemudian sistem pakar akan memprosesnya menggunakan metode *forward chaining* berbasis *web*. Dengan menggunakan aplikasi berbasis *web* maka akan mempermudah masyarakat umum untuk mencari informasi yang diperlukan hanya dengan mengakses melalui situs *web* setelah proses berjalan dan selesai maka aplikasi *web* akan menghasilkan analisis dan solusi.