

## **BAB II**

### **KAJIAN PUSTAKA**

#### **2.1 Teori Dasar**

Menurut Kerlinger (Indrianto dan supomo, 1999,57; Rachmat, 2005,6) teori merupakan suatu kumpulan konstruk, konsep, definisi, dan proposisi yang menggambarkan fenomena secara sistematis melalui penentuan hubungan antara variabel dengan tujuan untuk menjelaskan fenomena.(Almanda, 2016: 55)

Pada bab ini akan dijelaskan tentang beberapa teori dasar antara lain kecerdasan buatan atau *Artificial Intelligence (AI)* dan beberapa subdisiplin ilmunya seperti logika *fuzzy (fuzzy logic)*, jaringan saraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*); *web*, basis data.

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

Menurut Jones (2008: 3) tahun 1950 merupakan saat-saat awal dari *AI* yaitu saat awal sistem komputer dibangun dan ide-ide pembangunan mesin cerdas mulai terbentuk. Pada tahun 1950, Alan Turing menyimpan pertanyaan dalam pikirannya “apakah sebuah mesin mampu untuk berpikir”. Alan Turing melakukan percobaan yang cukup sederhana untuk menentukan apakah suatu mesin bisa dikatakan cerdas. Hasil percobaannya ini disebut dengan *Turing Test*. Dalam *Turing Test*, jika sebuah mesin mampu mengelabui seseorang yang

menganggap mesin itu adalah manusia, maka mesin itu dianggap telah lulus dari tes kecerdasan (*intelligence test*).

Menurut Rosnelly (2012:1) Definisi yang populer dari AI lainnya adalah bahwa AI menjadikan komputer berakting dan bergaya seperti halnya pada artis seperti di bioskop. Untuk saat ini banyak permasalahan dunia nyata yang diselesaikan menggunakan AI dan banyak juga aplikasinya yang dikomersialkan.

Menurut Jones (2008: 5) pada tahun 1956, *Dartmouth AI Conference* membawa para peneliti yang terlibat dalam penelitian AI seperti John McCarthy bersama peneliti-peneliti lainnya untuk sesi diskusi dan penelitian AI di *Dartmouth College*. Sejak saat itu, banyak konferensi AI telah diselenggarakan di seluruh dunia, dan berbagai disiplin ilmu belajar di bawah nama AI. Pada awal 80-an penelitian tentang AI sukses di bidang komersial dari *software* jenis *Expert System*. Pada era 90-an dan awal abad 21, AI mencapai kesuksesan terbesarnya setelah diadopsi secara luas oleh industri teknologi, memberikan sumbangan besar pada logistik, data mining, diagnosis medis dan berbagai bidang lainnya. Kombinasi antara AI dengan bidang ilmu yang lainnya melahirkan subdisiplin ilmu dalam AI. Beberapa diantaranya adalah logika *fuzzy* (*fuzzy logic*), jaringan syaraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*) (Al Fatta, 2009: 4).

Kecerdasan dapat diartikan sebagai kemampuan untuk memperoleh pengetahuan dan menggunakannya atau kecerdasan adalah apa yang di ukur oleh sebuah "test kecerdasan". Dan ini merupakan bagian dari ilmu pengetahuan komputer yang ditujukan dalam perancangan otomatisasi tingkah laku cerdas

dalam sistem kecerdasan komputer. Sistem memperlihatkan sifat-sifat khas yang dihubungkan dengan kecerdasan dalam kelakuan yang sepenuhnya bisa menirukan beberapa fungsi otak manusia, seperti pengertian bahasa, pengetahuan. *Artificial Intelligent* atau Kecerdasan buatan merupakan bagian dari kecerdasan buatan (*artificial intelligence*). Kecerdasan ini merupakan inovasi baru di bidang ilmu pengetahuan. Mulai ada sejak muncul komputer modern, yakni pada 1940 dan 1950. Ini kemampuan mesin elektronika baru yang menyimpan sejumlah besar info, juga memproses dengan kecepatan sangat tinggi menandingi kemampuan manusia (Azmi, dkk., 2013: 11)

### **2.1.2 Logika *Fuzzy* (*fuzzy logic*)**

Menurut Lotfi Zadeh (1960) mengemukakan bahwa *fuzzy* merupakan suatu aspek ketidaktentuan yang berbeda dengan keacakan. Kemudian Zadeh mengusulkan bentuk matematika untuk melihat bagaimana ketidakjelasan dapat dinyatakan dalam bahasa manusia yang pendekatannya disebut “logika fuzzy”. Dalam *fuzzy* dikenal derajat keanggotaan yang memiliki interval  $[0,1]$ . Dalam teori logika fuzzy suatu nilai bisa bernilai benar atau salah secara bersama, namun berapa besar keberadaan dan kesalahan tergantung pada bobot keanggotaan yang dimilikinya (Navianti, dkk., 2012: 2).

Logika adalah salah satu cabang ilmu matematika ilmu yang mempelajari secara sistematis kaidah-kaidah penalaran yang abash (valid). Secara umum, terdapat 2 konsep logika didalam ilmu matematika, yaitu logika tegas dan logika fuzzy. Logika tegas hanya mengenal dua keadaan yaitu: ya atau tidak, *on* atau *off*,

*high* atau *low*, 1 atau 0. Logika semacam ini disebut dengan logika himpunan tegas. Sedangkan logika fuzzy adalah logika yang menggunakan konsep sifat kesamaran. Dengan demikian logika fuzzy adalah logika dengan tak hingga banyak nilai kebenaran yang dinyatakan dalam bilangan real dalam selang  $[0,1]$  Logika fuzzy pertama kali diperkenalkan oleh Lofti A. Zadeh dari universitas Berkley California pada tahun 1965. Zadeh memodifikasi teori himpunan yang setiap anggotanya memiliki derajat keanggotaan yang bernilai, kontinu antara 0 sampai 1 yang digunakan untuk menangani kekaburan. Himpunan ini disebut dengan himpunan kabur. Logika fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang input ke dalam suatu ruang output. Logika fuzzy sudah banyak diterapkan di berbagai bidang, baik di dunia industri maupun bisnis. Berbagai teori didalam perkembangan logika fuzzy dapat digunakan memodelkan berbagai sistem. Bahkan sekarang ini aplikasi logika fuzzy semakin menjamur seiring dengan pesatnya perkembangan teknologi komputasi. Penelitian aplikasi logika fuzzy telah banyak dilakukan dalam kehidupan masyarakat. Hal tersebut dikarenakan logika fuzzy merupakan salah satu cabang matematika yang dapat di aplikasikan dengan mudah ke semua bidang yang ada didalam kehidupan *Fuzzy Inference System* (FIS) merupakan salah satu bab yang dibahas didalam logika fuzzy. FIS merupakan sebuah sistem yang menggunakan teori himpunan fuzzy untuk memetakan variabel-variabel input ke variabel output melalui serangkaian rule. Secara garis besar, *input crisp* dimasukkan ke FIS. Input ini kemudian dikirim ke basis pengetahuan yang berisi aturan fuzzy dalam bentuk *if-then*. Derajat kebenaran (*fire strength*) akan dicari pada setiap aturan. Jika jumlah

aturan lebih dari satu maka dilakukan inferensi dari semua aturan. Untuk mendapatkan nilai *crisp* sebagai *output* sistem dilakukan defuzzyfikasi dari hasil inferensi. *Fuzzy inference system* (FIS) dapat dilakukan dengan tiga metode, yaitu dengan metode Mamdani, metode Sugeno dan metode Tsukamoto. Implementasi *Fuzzy inference system* (FIS) sudah banyak seiring dengan berkembang pesatnya ilmu pengetahuan dan teknologi. Salah satunya yaitu sistem pendukung dalam mengambil keputusan. (Susilo, 2006 dalam Muhammad, 2015 : 68 ).

Kelebihan logika *fuzzy* ada pada kemampuan penalaran secara bahasa, sehingga, dalam perancangannya tidak memerlukan persamaan matematis yang kompleks dari objek yang akan dikendalikan.

Beberapa alasan digunakannya logika *fuzzy*, antara lain:

1. Konsep logika *fuzzy* mudah dimengerti, karena di dalam logika *fuzzy* terdapat konsep matematis sederhana dan mudah dimengerti yang mendasari penalaran *fuzzy* .
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi–fungsi nonlinier yang sangat kompleks.
5. Logika *fuzzy* dapat bekerjasama dengan teknik– teknik kendali secara konvensional.
6. Logika *fuzzy* didasarkan pada bahasa alami.

7. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan. (Kusuma dewi, Sri (2003) *dalam* Widodo, 2014: 26 )

### **2.1.3 Jaringan Saraf Tiruan (*Artificial Neural Network*)**

(Azmi, dkk., 2013: 11) Jaringan Saraf Tiruan (JST) merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. Jaringan Saraf Tiruan sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa:

1. Pemroses informasi terjadi banyak elemen sederhana (neuron)
2. Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung
3. Penghubung antara neuron memiliki bobot yang akan memperkuat atau mepeperlemah sinyal.
4. Untuk menentukan input, setiap neuron menggunakan fungsi aktivasi yang dikenakan pada jumlah input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

(Suyanto, 2014: 168) JST merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melakukan tugas tertentu.

(Tanjung, 2009: 29) Jaringan Saraf Tiruan (*artificial neural network*) atau di singkat JST adalah sistem komputasi dengan arsitektur dan operasinya diilhami dari pengetahuan tentang sel saraf biologi di dalam otak. JST dapat digambarkan

sebagai model matematis dan komputasi untuk fungsi *aproksimasi nonlinear*, klasifikasi data, *cluster* dan *regresi non parametric* atau sebagai sebuah simulasi dari koleksi model saraf biologi.

#### 1. *Fungsi Aktivasi Sigmoid Biner*

Fungsi ini digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan saraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan saraf yang nilai outputnya 0 atau 1

#### 2. *Metode Backpropagation*

*Backpropagation* merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang berhubungan dengan *neuron-neuron* yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu.

### **2.1.4 Sistem Pakar (*Expert System*)**

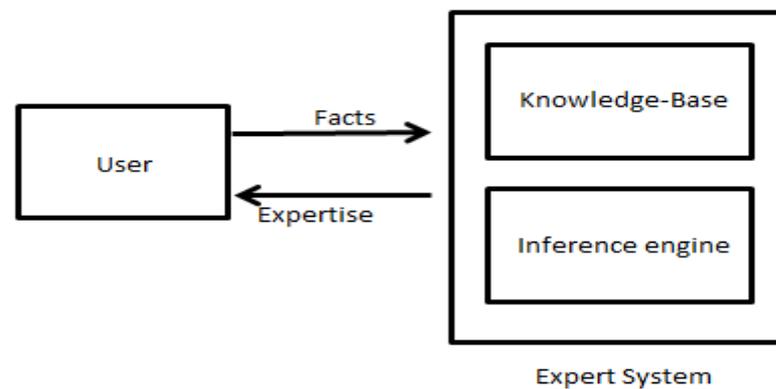
Sistem pakar adalah salah satu cabang dari AI yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang

tertentu,yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya. Ketika sistem pakar dikembangkan pertama kali sekitar tahun 70-an sistem pakar hanya berisi *knowledge*. Namun demikian sekarang ini istilah sistem pakar sudah diuntuk berbagai macam sistem yang menggunakan teknologi sistem pakar itu.(Listiono,2008 : 1)

(Rosnelly,2012:2) Bidang sistem pakar merupakan pengelesaian pendekatan yang sangat berhasil dan bagus untuk permasalahan AI klasik dari pemrograman *intelligent*(cerdas). Sistem pakar merupakan solusi AI bagi masalah pemrograman pintar.Profesor Edward Feigenbaum dari Stanford University yang merupakan pioner dalam teknologi sistem pakar yang memanfaatkan pengetahuan dan prosedur inferensi untuk memecahkan masalah yang cukup sulit sehingga membutuhkan keahlian khusus dari manusia.Dengan kata lain sistem pakar adalah system komputer yang ditunjuk untuk meniru semua aspek kemampuan pengambilan keputusan seorang pakar.

Pakar atau ahli didefinisikan sebagai seseorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Seorang pakar dapat memecahkan masalah yang tidak mampu di pecahkan kebanyakan orang. Dengan kata lain dapat memecahkan masalah dengan lebih efisien namun bukan berarti lebih murah. Pengetahuan yang dimuat ke dalam sistem pakar dapat berasal dari seorang pakar atau pun pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang di publikasikan lainnya, serta orang yang memiliki pengetahuan meskipun bukan ahli.

Bagian dalam sistem pakar terdiri dari 2 komponen utama, yaitu basis pengetahuan (*knowledge-base*) yang berisi pengetahuan dan mesin inferensi yang menggunakan kesimpulan. Kesimpulan tersebut merupakan respon dari sistem pakar atas permintaan pengguna.



**Gambar 2.1** Konsep dasar fungsi sistem pakar berbasis pengetahuan  
(Sumber: Rosnelly, 2012: 4)

Penggunaan sistem basis-pengetahuan (*knowledge-base*) juga dirancang untuk aksi pemandu cerdas seorang ahli pemandu cerdas dirancang dengan teknologi sistem pakar karena memberikan banyak keuntungan terhadap pengembangnya. Semakin banyak *knowledge* yang ditambahkan untuk pemandu cerdas maka sistem tersebut akan semakin baik dalam bertindak sehingga semakin menyerupai pakar sebenarnya (Listiyono,2008 :2)

(Hayadi,2016 :1) Sistem pakar suatu cabang dari *Artificial Intelligent (AI)* yang cukup tua karena system ini mulai di kembangkan pada tahun 1960.Sistem pakar adalah program AI dengan basis pengetahuan (*Knowledge Base*) yang diperoleh dari pengalaman atau pengetahuan pakar atau ahli dalam memecahkan persoalan pada bidang tertentu dan didukung mesin *inferensi/inference engine*

yang melakukan penalaran atau pelacakan terhadap sesuatu atau fakta –fakta dan aturan kaidah yang bisa di basis pengetahuan setelah dilakukan pencarian, sehingga dicapai kesimpulan.

#### **2.1.4.1 Ciri-Ciri sistem pakar**

Suatu sistem dikatakan sebagai sistem pakar jika memiliki ciri-ciri sebagai berikut (Hayadi., 2016: 3):

1. Terbatas pada *domain* 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 tertentu
5. Mudah dimodifikasi
6. Basis pengetahuan dan mekanisme *inferensi* diletakkan terpisah
7. Keluarannya (*output*) bersifat anjuran
8. Sistem dapat mengaktifkan kaidah secara terpisah secara searah, sesuai dengan dialog dengan pengguna

#### **2.1.4.2 Manfaat dan kekurang sistem pakar**

(Hayadi, 2016 :2-3) Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya T. Sutejo, e.t.2010, Diantaranya :

1. Meningkatkan produktivitas, karena system pakar dapat bekerja lebih cepat daripada manusia.
2. Membuat seseorang 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. Memudahkan akses pengetahuan seorang pakar.
6. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelasan yang berfungsi sebagai guru.
7. Meningkatkan kemampuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

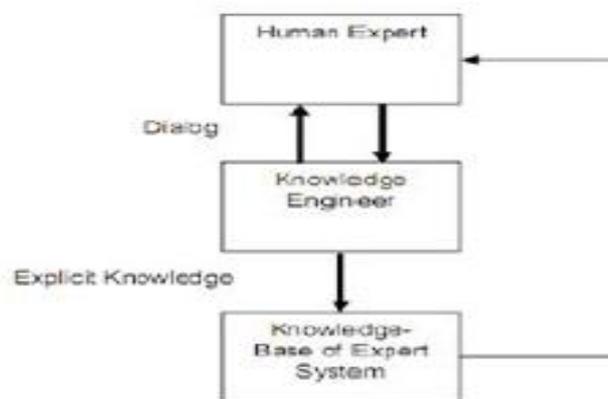
(Hayadi, 2016 :3) Selain manfaat ada beberapa kekurangan yang ada pada sistem pakar, diantaranya :

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar

### **2.1.4.3 Konsep umum sistem pakar**

(Ronelly,2012:6-7)Pengetahuan yang dimiliki sistem pakar direpresentasikan dalam beberapa cara, salah satunya yang paling umum digunakan adalah tipe rules menggunakan format IF THEN. Banyak sistem pakar

yang dibangun dengan mengekspresikan pengetahuan (*knowledge-based approach*) untuk membangun sistem pakar telah mematahkan pendekatan awal yang digunakan pada sekitar tahun 1950-an dan 1960-an yang menggunakan teknik penalaran (*reasoning*) yang tidak mengandalkan pengetahuan. Pengetahuan tidak tertulis yang dimiliki oleh seorang pakar harus diekstraksi melalui wawancara secara ekstensif oleh *knowledge engineering*. Proses pengembangan sistem pakar yang berhubungan dengan perolehan pengetahuan dari pakar maupun sumber lain dan kodingnya disebut sebagai *knowledge engineering* yang dilaksanakan oleh *knowledge engineer*. Tahapan pengembangan sistem pakar secara umum tergambar pada Gambar 1.4 dibawah ini.



**Gambar 2.2** Pengembangan Sistem Pakar  
(Sumber: Rosnelly, 2012: 6-7)

Tahap awal *Knowledge engineer* melakukan diskusi dengan pakar untuk mengumpulkan pengetahuan yang dimiliki pakar yang bersangkutan. Tahap ini serupa dengan proses diskusi persyaratan atau kebutuhan yang dilakukan *system engineer* pada sistem konvensional dengan kliennya. Setelah itu *Knowledge engineer* melakukan koding pengetahuan secara eksplisit ke dalam *knowledge*

*base*, pakar kemudian mengevaluasi sistem pakar dan memberikan kritik, proses ini berlangsung secara iterative hingga dinilai sesuai oleh pakar.

Menurut Hartati dan Iswanti (2008: 22) representasi pengetahuan dimaksudkan untuk mengorganisasikan pengetahuan dalam bentuk dan format tertentu agar dapat dimengerti oleh komputer. Pemilihan representasi pengetahuan yang tepat akan menghasilkan sebuah sistem pakar yang efektif. Salah satu model representasi pengetahuan yang penting yaitu kaidah produksi (*production rule*).

Sistem pakar pada penelitian ini menggunakan model representasi pengetahuan berbasis kaidah produksi. Menurut Firebaugh (1988) dalam Hartati dan Iswanti (2008: 10) struktur sistem pakar yang berbasis kaidah produksi terdiri dari 4 komponen, yaitu:

1. Antarmuka pemakai

Menurut Kusri (2006: 17) antarmuka merupakan penghubung antara pemakai dengan sistem pakar. Komponen ini berfungsi sebagai alat komunikasi antara sistem dan pengguna (*user*) yang penting sekali bagi pengguna. Komponen ini harus didesain sedemikian rupa sehingga efektif dan mudah digunakan terutama bagi pengguna yang tidak ahli dalam bidang yang diterapkan pada sistem pakar (Hartati dan Iswanti, 2008: 4-5).

2. Basis pengetahuan

Basis pengetahuan adalah komponen yang berisi sekumpulan kaidah yang berasal dari pengetahuan dalam domain tertentu dan secara umum disajikan dalam bentuk kaidah produksi (*IF...THEN...*). Pengetahuan pakar yang disajikan dalam format tertentu didapat dari sekumpulan pengetahuan pakar dan sumber-sumber

pengetahuan lainnya seperti buku-buku, jurnal ilmiah, majalah, maupun dokumentasi tercetak lainnya. Basis pengetahuan diletakkan terpisah dari mesin inferensi agar pengembangan pengetahuan sistem pakar dapat dilakukan secara leluasa tanpa mengganggu mesin inferensi (Hartati dan Iswanti, 2008: 5).

### 3. Struktur kontrol (Mesin Inferensi)

Struktur kontrol merupakan *interpreter* kaidah atau mesin inferensi yang menggunakan pengetahuan-pengetahuan yang tersimpan dalam basis pengetahuan untuk memecahkan atau menyelesaikan permasalahan yang ada. Menurut Kusriani (2006: 35) inferensi merupakan proses untuk menghasilkan informasi berupa konklusi logis berdasarkan informasi yang tersedia atau fakta yang diketahui.

Dalam melakukan proses inferensi, sistem pakar memerlukan adanya proses pengujian kaidah-kaidah dalam urutan tertentu untuk mencari suatu kondisi yang sesuai dengan kondisi awal atau untuk memastikan kondisi yang sedang berjalan sudah dimasukkan ke dalam *database*. Proses pengujian itu disebut dengan perunutan atau penalaran, yaitu proses pencocokan fakta atau kondisi tertentu yang tersimpan dalam basis pengetahuan maupun pada memori kerja dengan kondisi yang dinyatakan dalam premis atau bagian kondisi pada suatu kaidah atau aturan (Hartati dan Iswanti, 2008: 45).

Ada beberapa konsep penalaran yang dapat digunakan oleh mesin inferensi yaitu:

#### a. Penalaran maju (*forward chaining*)

Konsep ini dapat juga disebut sebagai pencarian yang dimotori data (*data driven search*). Runut maju melakukan proses perunutan (penalaran) dimulai dari

premis-premis atau informasi masukan (*IF*) terlebih dahulu kemudian menuju konklusi atau *derived information* (*THEN*). Konsep ini dapat dimodelkan sebagai berikut:

*IF* (informasi masukan)

*THEN* (konklusi)

Informasi masukan dapat berupa suatu pengamatan sedangkan konklusi dapat berupa diagnosa sehingga dapat dikatakan jalannya penalaran runut maju dimulai dari pengamatan menuju diagnosa. Pada metode ini, sistem tidak melakukan praduga apapun terhadap konklusi, namun sistem akan menerima semua gejala yang diberikan pengguna lalu sistem akan memeriksa gejala-gejala tersebut dan selanjutnya mencocokkan dengan konklusi yang sesuai (Hartati dan Iswanti, 2008: 45-47).

b. Penalaran mundur (*backward chaining*)

Secara umum, konsep ini diaplikasikan ketika tujuan ditentukan sebagai kondisi atau keadaan awal. Konsep ini disebut juga *goal-driven search*. Arah penalaran atau peruntutan dalam konsep ini berlawanan dengan *forward chaining*. Konsep ini dapat dimodelkan sebagai berikut:

Tujuan,

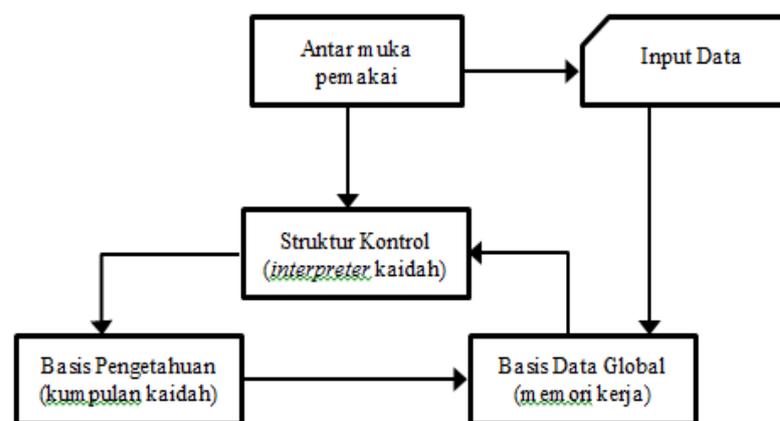
*IF* (kondisi)

Proses penalaran pada *backward chaining* dimulai dari tujuan kemudian merunut balik ke jalur yang mengarah ke tujuan tersebut, untuk membuktikan bahwa bagian kondisi pada kaidah atau aturan benar-benar terpenuhi. Proses *internal* selalu memeriksa konklusi (tujuan) terlebih dahulu sebagai praduga awal,

kemudian memeriksa dan memastikan gejala-gejala (kondisi) telah terpenuhi dan selanjutnya mengeluarkan konklusi sebagai *output*. Jika sistem menemukan ada bagian kondisi yang tidak terpenuhi maka sistem akan memeriksa konklusi (tujuan) pada aturan atau kaidah berikutnya (Hartati dan Iswanti, 46-47).

#### 4. *Working memory* (memori kerja) atau basis data global

Berfungsi untuk mencatat status masalah yang terjadi dan *history* solusi. Memori kerja merupakan bagian yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi saat proses konsultasi terjadi (Kusrini, 2008: 19).



**Gambar 2.3** Struktur Sistem Pakar Kaidah Produksi  
(Sumber: Firebaugh, 1988 dalam Hartati dan Iswanti, 2008: 10)

Kusrini (2008: 33) menjelaskan bahwa kaidah menyediakan cara formal yang dituliskan dalam bentuk jika-maka (*IF-THEN*) untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah *IF-THEN* menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Berikut ini adalah contoh struktur kaidah *IF-THEN* yang menghubungkan obyek (Adedeji, 1992 dalam Hartati dan Iswanti, 2008: 25):

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* gejala *THEN* diagnosa

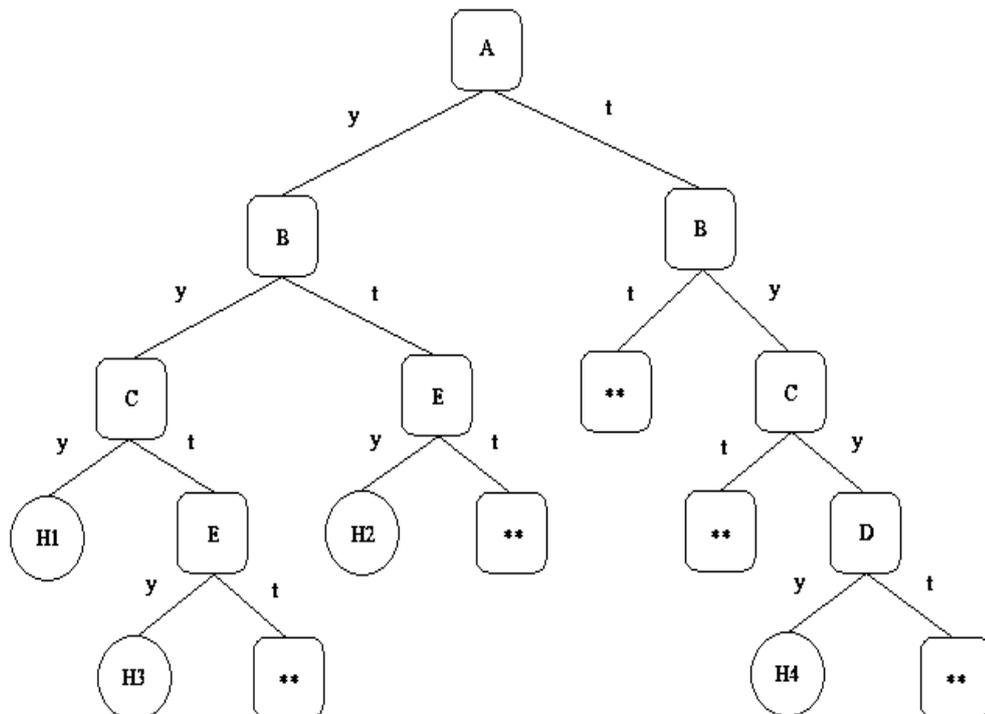
Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Aksi mengacu pada kegiatan yang menyebabkan munculnya efek dari tindakan tersebut. Gejala mengacu pada keadaan yang menyebabkan adanya kerusakan atau keadaan tertentu yang mendorong adanya pemeriksaan (diagnosa) (Hartati dan Iswanti, 2008: 25-26).

Sebelum sampai pada bentuk kaidah produksi, pengetahuan yang berhasil didapatkan dari domain tertentu disajikan dalam bentuk tabel keputusan kemudian dibuat pohon keputusannya. Berikut ini adalah contoh penyajian dalam bentuk tabel keputusan dan pohon keputusan (Hartati dan Iswanti, 2008: 26-39).

**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 Iswanti (2008: 32)



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

**Gambar 2.4** Pohon Keputusan  
 (Sumber: Hartati dan Iswanti, 2008: 33)

Dari gambar diatas dapat diketahui bahwa hipotesa H1 terpenuhi jika memenuhi *evidence* A, B, dan C. Hipotesa H2 terpenuhi jika memiliki *evidence* A dan *evidence* E. Hipotesa H3 akan terpenuhi jika memiliki *evidence* A, B, dan E. Hipotesa H4 akan dihasilkan jika memenuhi *evidence* B, C, dan D. Notasi “y” mengandung arti memenuhi *node (evidence)* di atasnya, notasi “t” artinya tidak memenuhi.

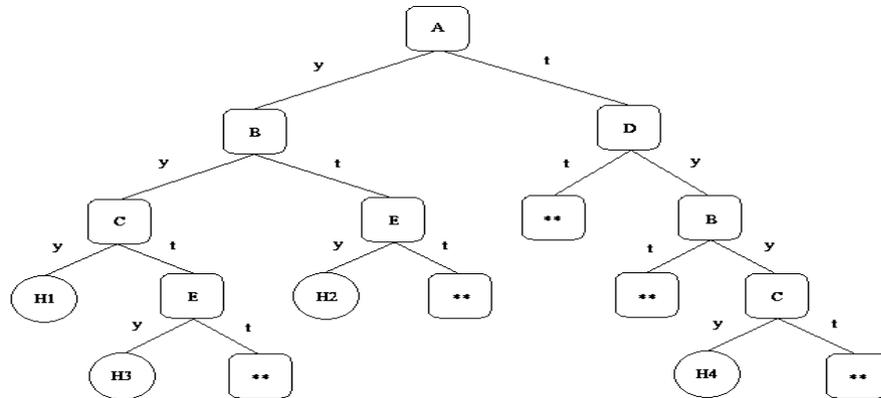
Dalam sesi konsultasi pada sistem pakar, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang diajukan oleh sistem. Dengan melihat pohon keputusan pada gambar 2.2 permasalahan dapat saja terjadi pada awal sesi konsultasi yaitu pada saat sistem pakar menanyakan “apakah memiliki *evidence* A?”. Permasalahannya adalah apapun jawaban pengguna baik “ya” atau “tidak” maka sistem akan menanyakan *evidence* B. Ini berarti jawaban pengguna tidak akan mempengaruhi sistem. Salah satu cara untuk mengatasi hal ini adalah dengan mengubah urutan pada tabel keputusan seperti terlihat pada tabel 2.2.

**Tabel 2.2** Alternatif Tabel Keputusan

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

Sumber: Hartati dan Iswanti (2008: 34)

Berdasarkan tabel 2.2 dapat dihasilkan pohon keputusan sebagai berikut:



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

**Gambar 2.5** Alternatif Pohon Keputusan  
 (Sumber: Hartati dan Iswanti, 2008: 35)

Dilihat dari gambar 2.5 masing-masing *node* yang mewakili *evidence* tertentu untuk kondisi “y” dan “t” sudah tidak mengarah pada *evidence* yang sama. Hal ini berarti jawaban pengguna yang berbeda akan mengarah pada pertanyaan yang berbeda pula.

Kaidah yang dapat dihasilkan berdasarkan pohon keputusan pada gambar 2.5 adalah sebagai berikut:

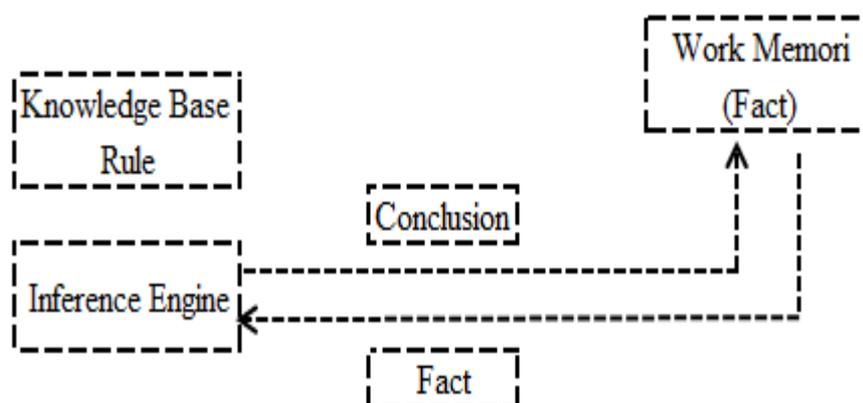
1. Kaidah 1: *IF A AND B AND C THEN H1*
2. Kaidah 2: *IF A AND B AND E THEN H3*
3. Kaidah 3: *IF A AND E THEN H2*
4. Kaidah 4: *IF D AND B AND C THEN H4*

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.

### 2.1.5 *Forward Chaining*

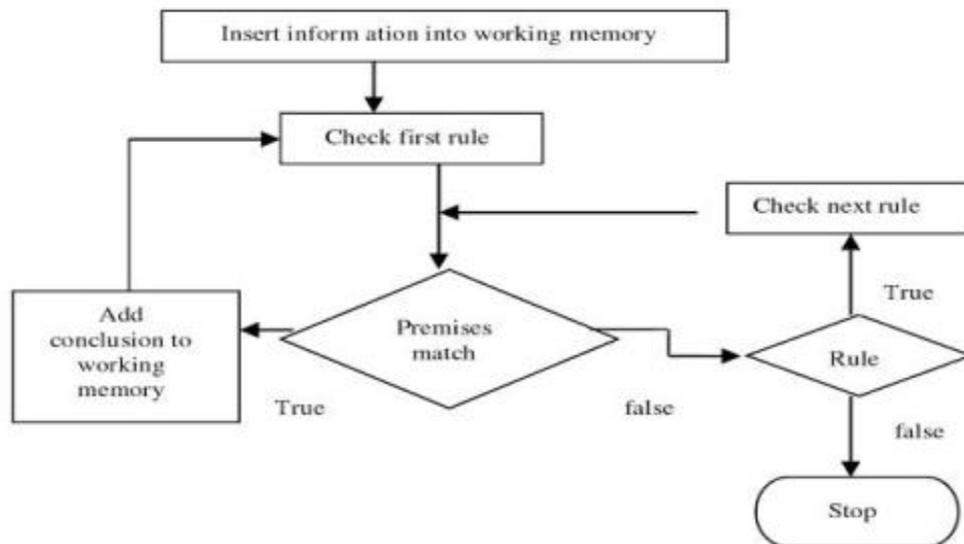
(Hayadi,2016:10-11) *Forward Chaining* adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari *rules* IF\_THEN. Bila ada fakta yang cocok dengan bagian IF maka *rule* tersebut di eksekusi. Bila sebuah *rule* di eksekusi maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database.

Sistem pakar *forward chaining* berbasis aturan dapat dimodelkan seperti gambar berikut ini.



**Gambar 2.6** Model Berbasis Aturan  
(Sumber: Hayadi,2016:11)

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. Operasi tersebut dapat di gambarkan seperti gambar.



**Gambar 2.7** Operasi Sistem Forward Chaining  
(Sumber: Hayadi,2016:11)

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

1. Pendefinisian masalah

Tahapan 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 pengendalian pengaktifan suatu aturan.

#### 4. Penulisan kode awal

Tahapan 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 satu komponen penting dari suatu sistem. Perancangan antarmukan dibuat bersama-sama dengan pembuatan basis pengetahuan.

#### 7. Pengembangan sistem

Pengembangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan *prototype 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.

### **2.1.6 Website**

(Nuruddin 2013 *dalam* Ishak,2016: 23) WWW (*World Wide Web*) atau juga dikenal dengan *WEB* adalah salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet. *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu baik

yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan satu sama lain. Sedangkan menurut A.Taufiq Hidayatullah, *web* adalah bagian paling terlihat sebagai jaringan terbesar dunia, yakni internet.

### **2.1.7 Database (basis data)**

Menurut (A.S. dan Shalahuddin, 2013: 45-46 ) sistem *database* adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data atau informasi yang sudah diolah dan membuat informasi tersedia saat dibutuhkan. *Database* adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data meliputi memasukkan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan. Salah satu bentuk basis data yang dibutuhkan dalam sebuah sistem yaitu *Database Management System (DBMS)*. *DBMS* adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Syarat minimal dari *DBMS* antara lain :

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan secara bersamaan
4. Mampu menangani *backup* data

Ada beberapa *DBMS* yang paling banyak digunakan saat ini antara lain:

1. *DBMS* versi komersial, yaitu *Oracle*, *Microsoft SQL Server*, *IBM DB2*, dan *Microsoft Access*

2. *DBMS* versi *open source*, yaitu *MySQL*, *PostgreSQL*, *Firebird*, dan *SQLite*

### 2.1.8 Validasi Sistem

Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa sistem atau perangkat lunak yang dibangun telah sesuai dengan yang diharapkan. Beberapa pendekatan dalam melakukan pengujian untuk validasi sistem antara lain (A.S. dan Shalahuddin, 2013: 275-276):

1. *Black-Box Testing* (pengujian kotak hitam)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Tujuannya untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari sistem atau perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem atau perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan *black-box testing* harus dibuat dengan kasus benar dan kasus salah.

2. *White-Box Testing* (pengujian kotak putih)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi yang dibutuhkan. *White-box testing* dilakukan dengan memeriksa logika dari kode program. Pembuatan kasus uji dapat mengikuti standar pengujian dari standar pemrograman yang ada.

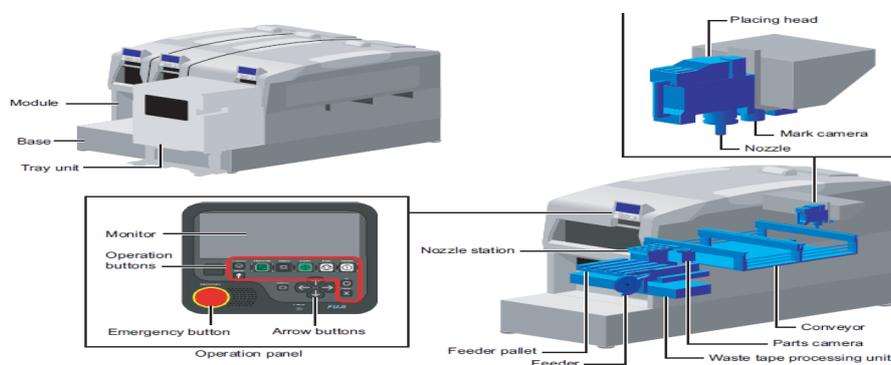
## 2.2 Variabel Penelitian

Obyek yang digunakan dalam penelitian ini adalah Mesin *NXT* dan variabel penelitian yang ditetapkan yaitu kerusakan umum (*Feeder*) Mesin *NXT*.

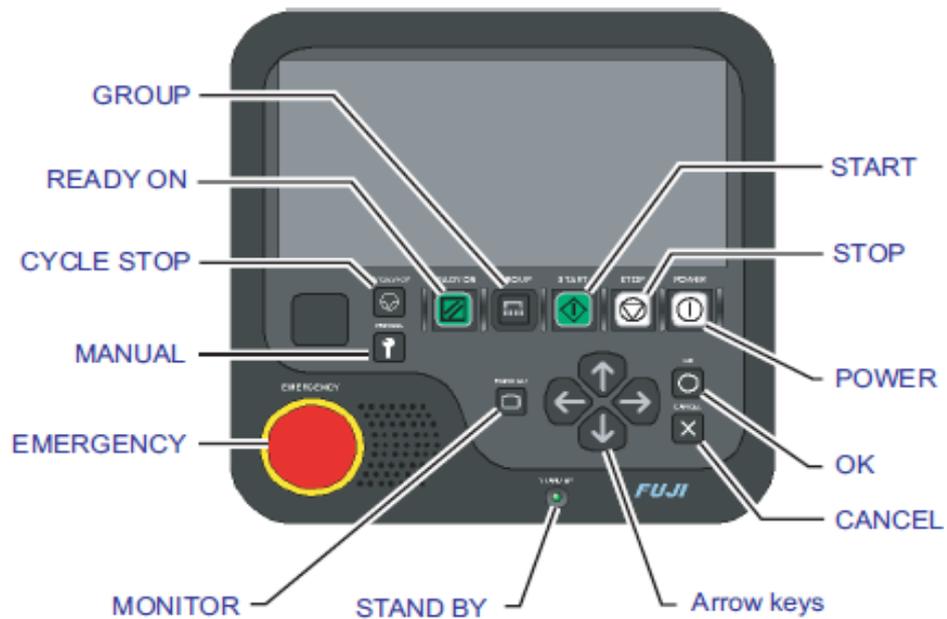
### 2.2.1 Mesin *NXT*

Mesin *NXT* bukan hanya mesin untuk menempatkan (*placement*) tetapi merupakan hasil dari usaha fuji untuk memberikan revolusi sistem konsep *smt line* baru, memberikan solusi total untuk kebutuhan dari generasi berikutnya. Dengan menggabungkan teknologi yang unik untuk memungkinkan mesin secara otomatis beradaptasi dengan *highmix*, volume produksi, dan menggunakan sensor untuk penempatan komponen. Mesin baru ini merupakan inovasi tingkat tinggi, menggabungkan *flexibiliti* dua jenis (*M3 dan M6*) dalam konfigurasi modul, mesin ini sangat efisien di bandingkan dengan kenerasi sebelumnya.

#### 2.2.1.1 Komponen mesin *NXT*



**Gambar 2.8** Komponen Mesin *NXT*  
(Sumber: Fuji Machine Manufacturing, 2006)



**Gambar 2.9** Monitor Mesin *NXT*  
(Sumber: Fuji Machine Manufacturing, 2006)

1. *Start* ( Memulai produksi )
2. *Stop* ( Berhenti Produksi )
3. *Power* ( Untuk Menghidupkan Modul )
4. *OK* ( Menegaskan Pilihan Menu dan Memulai perintah operasi )
5. *Cancel* ( membatalkan pilihan menu dan mengembalikan panel operasi ke *screen*. Tombol ini juga dapat digunakan untuk menghentikan suara *alarm* ketika terjadi suatu kesalahan,
6. *Ready on* ( tekan tombol ini setelah kesalahan telah cleared. Jika tombol ini ditekan di layar maka status kesalahan akan dihapus.

7. *Cycle stop* ( mengirim perintah untuk pengoperasian *motor axis* )
8. *Manual* ( pertukaran antara modul dan pengoperasian layar manual )
9. *Emergency* ( memotong power 200V untuk modul dan berhenti beroperasi)
10. *Arrow keys* ( memindahkan kursor di layar menu )
11. *Stand by* ( mengindikasikan bahwa power yang disediakan untuk modul tersebut sudah siap )

### 2.2.1.2 Feeder

*Feeder* adalah bagian dari mesin *NXT* yang berfungsi untuk *mensupply* komponen yang akan di pasang di *pcb*. Ada beberapa tipe *feeder* yang digunakan pada mesin *nxt*, dan untuk mengetahui tipe *feeder* tersebut dapat melihat table berikut.

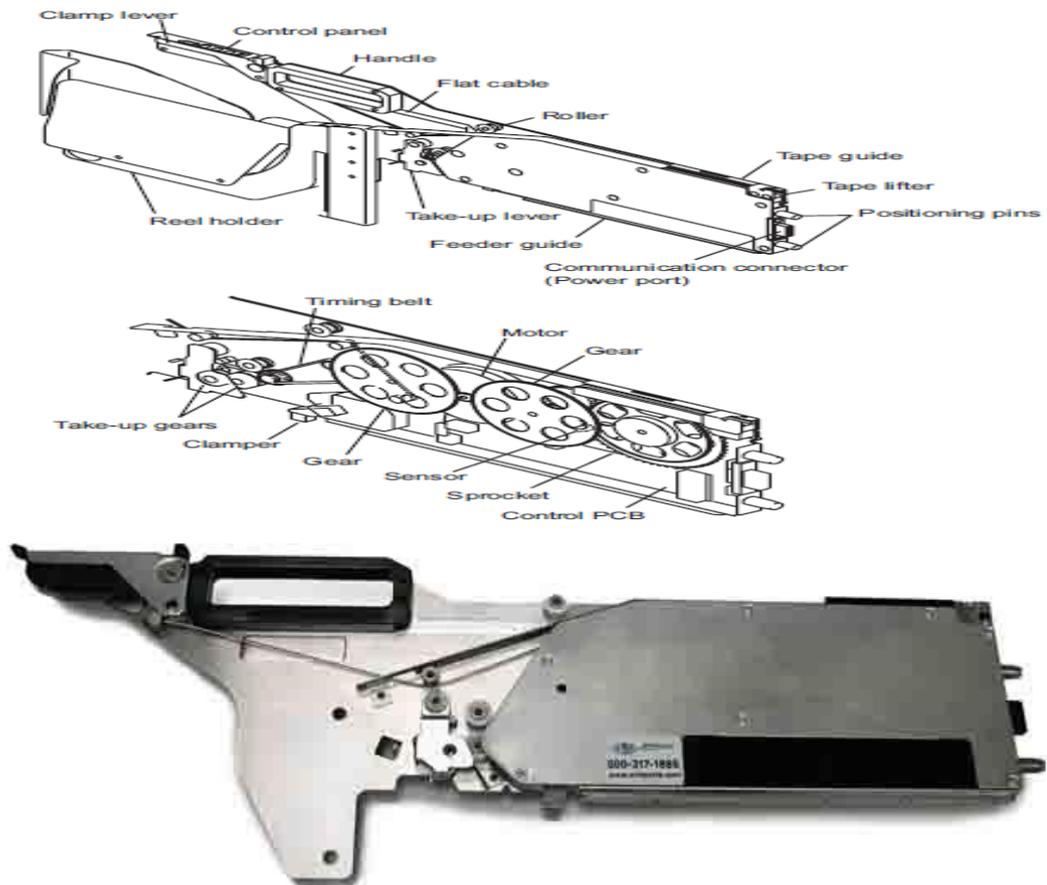
**Tabel 2.3** Tabel Tipe *Feeder*

<b>Feeder types</b>	<b>Tape types</b>	<b>Feeder types</b>	<b>Tape types</b>
W8 mm feeder	For 8 mm carrier tape	W44 mm feeder	For 44 mm carrier tape
W12 mm feeder	For 12 mm carrier tape	W56 mm feeder	For 56 mm carrier tape
W16 mm feeder	For 16 mm carrier tape	W72 mm feeder	For 72 mm carrier tape
W24 mm feeder	For 24 mm carrier tape	W88 mm feeder	For 88 mm carrier tape
W32 mm feeder	For 32 mm carrier tape		

**Sumber:** Fuji Machine Manufacturing,2006

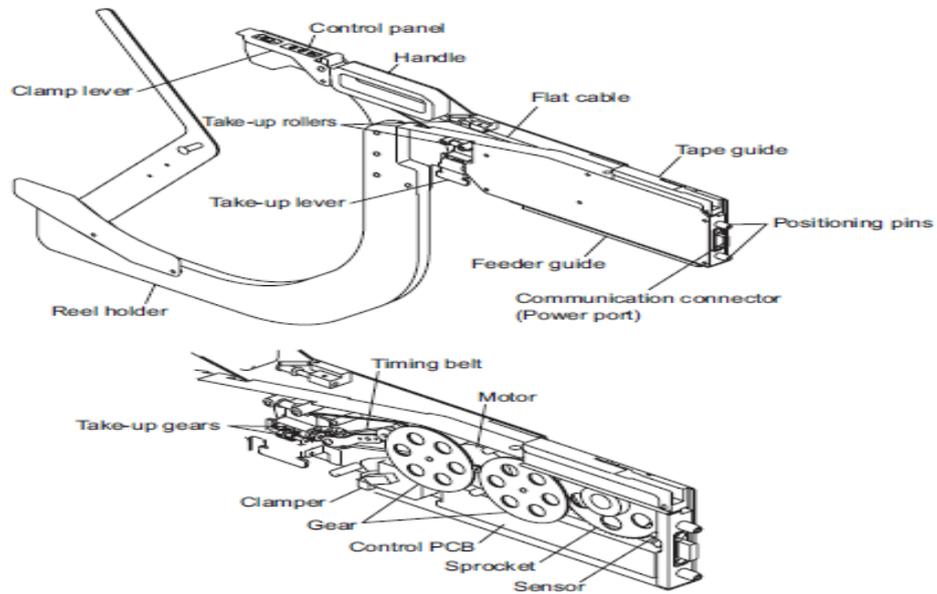
### 2.2.1.2.1 Komponen Feeder

#### 1. Feeder W8



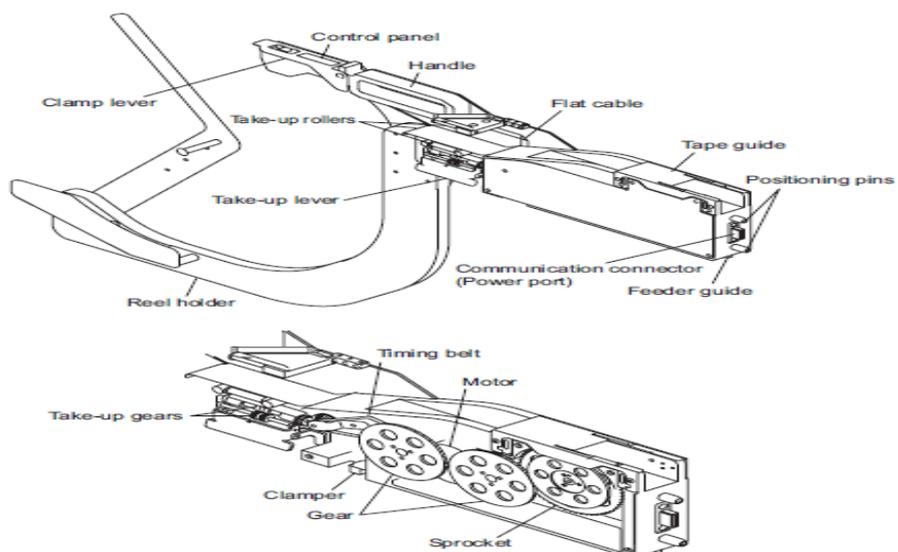
**Gambar 2.10 Feeder W8**  
(Sumber: Fuji Machine Manufacturing, 2006)

2. Feeder W12, W16 dan W24



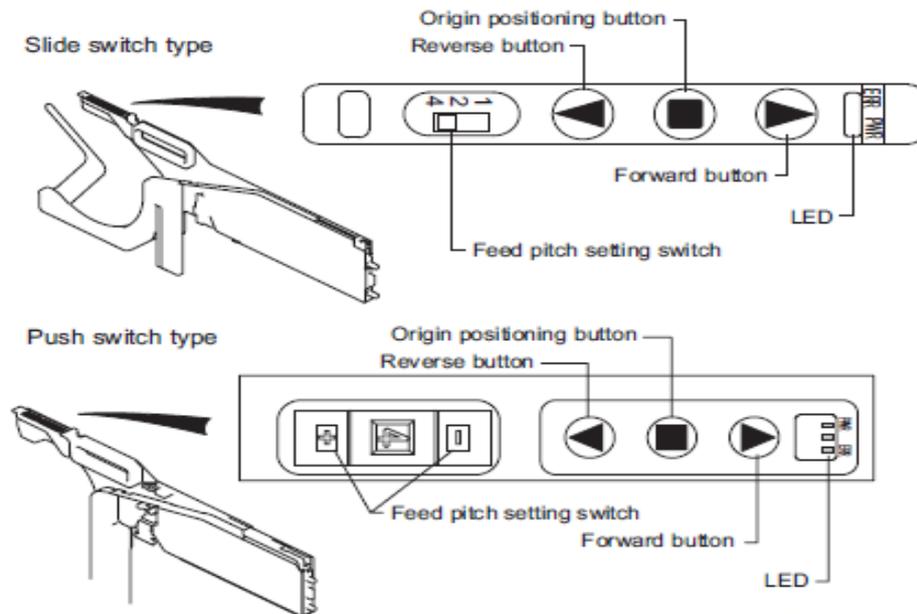
**Gambar 2.11** *Feeder W12, W16, W24*  
(Sumber: Fuji Machine Manufacturing, 2006)

3. Feeder W32 – W88



**Gambar 2.12** *Feeder W32 – W88*  
(Sumber: Fuji Machine Manufacturing, 2006)

#### 4. Panel Kontrol



**Gambar 2.13** *Panel Kontrol*  
(Sumber: Fuji Machine Manufacturing, 2006)

##### 1. *Forward Button*

Tekan tombol ini untuk memajukan komponen yang sudah terpasang di *feeder*. Tekan terus tombol ini untuk memajukan komponen terus menerus dengan kecepatan tinggi.

##### 2. *Origin positioning button*

Tekan tombol ini untuk memajukan bagian pada posisi pickup. Tahan selama kurang lebih 0,5 detik untuk memajukan komponen pada kecepatan rendah.

##### 3. *Reverse Button*

Tekan tombol ini untuk membalikkan arah komponen saat terlalu maju jauh, atau posisi *pickup* terlalu jauh kedepan.

#### 4. *LED*

PWR ( Hijau ) : LED ini menyala ketika *power* untuk feeder dalam keadaan ON

#### 5. *Feed Pitch Setting Switch*

gunakan untuk *tape feed pitch* untuk *tape loaded* di *feeder*. geser *switch* dan mendorong jenis saklar yang tersedia.

## 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: *XAMPP*, *phpMyAdmin*, *PHP*, *HTML*, *CSS*, *jQuery*, *MySQL*, *Notepad++*, dan *StarUML*.

### 2.3.1 *XAMPP (X Apache MySQL PHP Perl)*



**Gambar 2.14** Logo XAMPP

(Sumber: <https://wiki.bitnami.com/@api/deki/files/527/=xampp-logo.jpg>)

XAMPP adalah sebuah software *web server* apache yang didalamnya sudah tersedia database *server* MySQL dan dapat mendukung pemrograman PHP.

XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support* (PHP 4 dan PHP 5) dan beberapa *module* lainnya (Februariyanti dan Zuliarso, 2012: 129).

### 2.3.2 *phpMyAdmin*



**Gambar 2.15** Logo *phpMyAdmin*

(Sumber: <https://www.phpmyadmin.net/static/images/logo-og.png>)

*phpMyAdmin* adalah perangkat lunak gratis yang ditulis dalam bahasa pemrograman *PHP* bertujuan untuk menangani administrasi *MySQL* melalui *web*. *phpMyAdmin* mendukung berbagai operasi pada *MySQL* dan *MariaDB*. Operasi operasi yang sering digunakan seperti mengelola *database*, tabel, kolom, relasi, indeks, *users*, *permissions*, dan lain-lain, dapat dilakukan melalui antarmuka pengguna dengan tetap dapat mengeksekusi pernyataan *SQL* secara langsung. ([www.phpmyadmin.net/](http://www.phpmyadmin.net/))

### 2.3.3 *PHP: Hypertext Preprocessor (PHP)*



**Gambar 2.16** Logo *PHP*  
(Sumber: <https://www.php.net/download-logos.php>)

PHP (Hypertext Preprocessor), merupakan bahasa pemrograman pada sisi server yang memperbolehkan programmer menyisipkan perintah – perintah perangkat lunak web server (Apache, IIS, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke browser yang me-*request*-nya, contohnya adalah bagaimana memungkinkannya memasukkan tanggal sekarang pada sebuah halaman web setiap kali tampilan tanggal dibutuhkan. Sesuai dengan fungsinya yang berjalan di sisi server maka PHP adalah bahasa pemrograman yang digunakan untuk membangun teknologi *web application*. PHP telah menjadi bahasa *scripting* untuk keperluan umum yang pada awalnya hanya digunakan untuk pembangunan web yang menghasilkan halaman web dinamis. Untuk tujuan ini, kode PHP tertanam ke dalam dokumen sumber *HTML* dan diinterpretasikan oleh server web dengan modul PHP prosesor, yang menghasilkan dokumen halaman web. Sebagai bahasa pemrograman untuk tujuan umum, kode PHP diproses oleh aplikasi penerjemah dalam modus baris - baris perintah modus dan

melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program di channel output standar. Hal ini juga dapat berfungsi sebagai aplikasi grafis. PHP tersedia sebagai prosesor untuk server web yang paling modern dan sebagai penerjemah mandiri pada sebagian besar system operasi dan komputer *platform*. (Februariyanti dan Zuliarso, 2012: 128).

#### 2.3.4 HTML (*Hyper Text Markup Language*)



**Gambar 2.17** Logo *HTML*

(**Sumber:** [https://www.w3.org/html/logo/downloads/HTML5\\_Logo\\_512.png](https://www.w3.org/html/logo/downloads/HTML5_Logo_512.png))

HTML yang dikenal sebagai Hypertext Mark-up Language adalah bahasa untuk menjelaskan halaman web. Dalam hal struktur itu sendiri dari *head* dan *body*. *Head* mengandung informasi deskripsi *browser*. *Body* mengandung isi konteks, gambar dan sesuatu yang perlu ditampilkan. Sebuah file HTML terdiri dari elemen dan tag. Tag digunakan untuk mengatur elemen property (Khansa,dkk, 2013: 1)

HTML (*Hypertext Markup Language*) adalah sebuah bahasa pemrograman yang berbentuk skrip-skrip yang berguna untuk membuat sebuah halaman web.

HTML dapat dibaca oleh berbagai platform seperti : Windows, Linux, Macintosh. Kata "Markup Language" pada HTML menunjukkan fasilitas yang berupa tanda tertentu dalam skrip HTML dimana kita bisa mengatur judul, garis, tabel, gambar, dan lain-lain dengan perintah yang telah ditentukan pada elemen HTML. HTML sendiri dikeluarkan oleh W3C (*Word Wide Web Consortin*), setiap terjadi perkembangan level HTML harus dievaluasi ketat dan disetujui oleh W3C.(Hariadi, 2013: 49)

### 2.3.5 CSS (*Cascading Style Sheet*)



**Gambar 2.18** Logo CSS

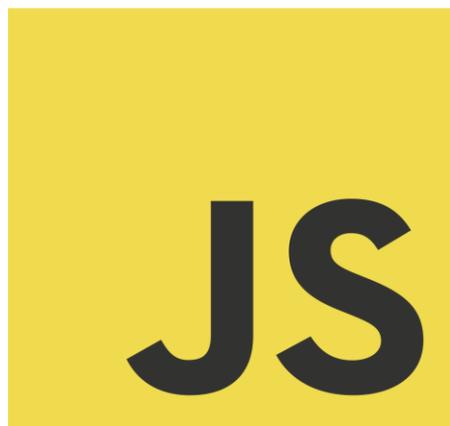
(Sumber: <http://w3widgets.com/responsive-slider/img/css3.png>)

CSS adalah singkatan dari *Cascading Style-Sheet*, yaitu sebuah pengembangan atas kode HTML yang sudah ada sebelumnya. Dengan CSS, bisa menentukan sebuah struktur dasar halaman web secara lebih mudah dan cepat, serta irit size (Hariadi, 2013: 50).

CSS atau *Cascading Style Sheets* adalah sebuah dokumen yang berisi aturan yang digunakan untuk memisahkan isi dengan *layout* dalam halaman halaman web yang dibuat. CSS memperkenalkan "*template*" yang berupa *style*

untuk dibuat dalam mengizinkan penulisan kode yang lebih mudah dari halaman halaman web yang dirancang. Sedangkan dalam buku CSS untuk orang awam yang menjelaskan tentang kegunaan CSS adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris, teks, warna, dan format border bahkan penampilan file gambar (Ishak dan simin, 2016: 24).

### **2.3.6 JavaScript dan jQuery**



**Gambar 2.19** Logo *JavaScript*  
(Sumber: [http://www.w3devcampus.com/wp-content/uploads/logoAndOther/logo\\_JavaScript.png](http://www.w3devcampus.com/wp-content/uploads/logoAndOther/logo_JavaScript.png))

Java Script merupakan modifikasi dari bahasa c++ dengan pola penulisan yang lebih sederhana. Interpreter bahasa ini sudah disediakan ASP ataupun internet explorer. Kelebihan Java Script adalah berinteraksi dengan HTML, ini membolehkan pembuat web untuk memasukkan web mereka dengan kandungan-kandungan yang dinamik, menukar warna background, menukar banner, efek mouse, menu interaktif dan sebagainya (Hariadi, 2013: 50).



**Gambar 2.20** Logo *jQuery*

(Sumber: <http://precision-software.com/wp-content/uploads/2014/04/jQurery.gif>)

jQuery adalah pustaka JavaScript yang cepat, ringan dan kaya fitur. Dengan adanya jQuery halaman HTML dapat dimanipulasi, diberikan event, animasi, dan AJAX yang jauh lebih sederhana dengan API yang mudah digunakan yang bekerja di banyak *browser*. Dengan kombinasinya yang fleksibel, jQuery telah mengubah cara bahwa jutaan orang menulis JavaScript. Sudah banyak *plugin-plugin* yang dibuat dari jQuery untuk mempermudah pembuat aplikasi. Salah satunya adalah *Jquery Page-Flip Plugin*, yaitu *plugin* jQuery yang dapat memanipulasi dalam sehingga terlihat seperti buku yang dapat dibalik.

(Khansa,dkk, 2013: 2)

### **2.3.7 MySQL**

MySQL (*My Structured Query Language*) atau yang biasa dibaca mai-sekuel adalah sebuah program pembuat dan pengelola database atau yang sering disebut dengan DBMS (*DataBase Management System*), sifat dari DBMS ini adalah Open Source. MySQL sebenarnya produk yang berjalan pada platform

Linux, dengan adanya perkembangan dan banyaknya pengguna, serta lisensi dari database ini adalah Open Source, maka para pengembang kemudian merilis versi Windows. Selain itu MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi Multi User (*Banyak Pengguna*). Kelebihan lain dari MySQL adalah menggunakan bahasa query (*permintaan*) standard SQL (*Structured Query Language*). Sebagai sebuah program penghasil database, MySQL tidak mungkin berjalan sendiri tanpa adanya sebuah aplikasi pengguna (hariadi,2013: 50)



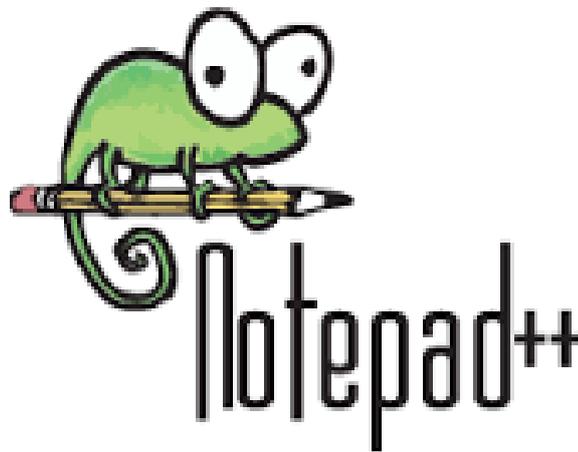
**Gambar 2.21** Logo *MySQL*

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

secara umum, database berfungsi sebagai tempat atau wadah untuk menyimpan, mengklasifikasikan data secara professional. MySQL bekerja menggunakan SQL Language (*Structure Query Language*) itu dapat diartikan bahwa MySQL merupakan standar penggunaan database didunia untuk pengolahan data. Sedangkan Raharjo (2011:21) MySQL merupakan RDBMS atau (*server database*) yang mengelolah database dengan cepat menampung dalam jumlah sangat besar dan dapat diakses oleh banyak user. MySQL adalah sebuah

manajemen system database server yang mampu menangani beberapa user, yaitu mampu menangani beberapa instruksi sekaligus dari beberapa user dalam satu waktu (Ishak, 2016: 24)

### 2.3.8 Notepad++



**Gambar 2.22** Logo *Notepad++*

(Sumber: <https://www.gavick.com/blog/wpcontent/uploads/2010/07/notepadd.jpg>)

*Notepad++* merupakan editor teks *open source* yang matang dan diakui sebagai pengganti *Notepad*. *Notepad++* tersedia untuk platform *Windows* yang dapat digunakan untuk menulis kode dengan beberapa pilihan bahasa (pemrograman). *Notepad++* menawarkan beragam kenyamanan fitur yang diharapkan dari setiap kemampuan *IDE (Integrated Development Environment)*, termasuk kemampuan untuk menunjukkan baris tertentu dari suatu dokumen sebagai referensi yang mudah; sintaks, tanda kurung, *indentation highlighting*, fasilitas pencarian yang tangguh, *macro recording* untuk tugas-tugas seperti memasukkan *template* komentar, dan sebagainya. Salah satu kelebihan *Notepad++* adalah dukungan dasar untuk *auto-completion* dari nama fungsi yang

ditawarkan sehingga akan mengurangi beberapa proses pengetikan kode. Gilmore (2010: 36)

### 2.3.9 *BootStrap*



**Gambar 2.23** Logo *Bootstrap*  
(Sumber: <http://getbootstrap/image/logo.png>)

*Bootstrap* merupakan sebuah *framework* css yang memudahkan pengembang untuk membangun website yang menarik dan responsif. Tidak konsistensinya terhadap aplikasi individual membuat sulitnya untuk mengembangkan dan pemeliharannya. *Bootstrap* adalah css tetapi dibentuk dengan *LESS*, sebuah *pre-processor* yang memberi fleksibilitas dari css biasa. *Bootstrap* memberikan solusi rapi dan seragam terhadap solusi yang umum, tugas *interface* yang setiap pengembang hadapi. *Bootstrap* dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan *design* yang butuhkan

Keunggulan dalam menggunakan *bootstrap* adalah semua bagian untuk antarmuka pengguna menggunakan *style* css yang dapat diintegrasikan dengan *JavaScript* untuk menjadikan lebih menarik dengan efekefek yang dapat

diberikan. Kelemahan dalam menggunakan *bootstrap* adalah tidak adanya keunikan didalam website karena akan samanya tampilan yang diberikan (Mustakin,dkk,2016:4)

### 2.3.10 *StarUML*



**Gambar 2.24** Logo *StarUML*  
(Sumber: <http://staruml.sourceforge.net/image/staruml-logo.jpg>)

*StarUML* merupakan platform pemodelan perangkat lunak yang mendukung *UML (Unified Modeling Language)*. *StarUML* yang berbasis pada UML versi 1.4 menyediakan sebelah jenis *Digram* yang berbeda, dan mendukung notasi *UML 2.0*. *StarUML* juga secara aktif mendukung pendekatan *MDA (Model Driven Architecture)* dengan mendukung konsep *UML profile*. *StarUML* unggul dalam hal kustomasi lingkungan kerja pengguna, dan memiliki ekstensibilitas tinggi pada fungsionalitasnya. *StarUML* pada mulanya dikenal dengan nama “*Plastic*” atau *Agora Plastic*”.

Terdapat 13 macam diagram dalam *UML 2.3* yang dibagi menjadi 3 kategori yaitu (A.S. dan Shalahuddin, 2011: 120-121):

1. *Structure diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. *Behaviour diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

3. *Interaction diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

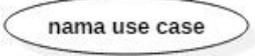
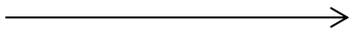
Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem yaitu:

1. *Use case diagram*

*Use case diagram* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada 2 hal utama yang terdapat pada *use case* yaitu aktor dan *use case*. Berikut

ini adalah simbol-simbol yang digunakan dalam *use case diagram* (A.S. dan Shalahuddin, 2011: 130).

**Tabel 2.4** Simbol *Use Case Diagram*

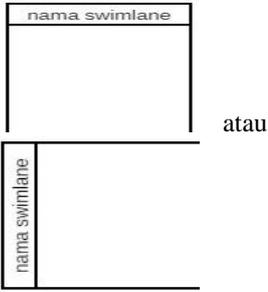
Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain 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 frase nama aktor</p>
<p>asosiasi/<i>association</i></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>Ekstensi/<i>extend</i></p> <p>&lt;&lt;extend&gt;&gt;</p> 	<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>
<p>generalisasi/<i>generalization</i></p> 	<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>
<p>Menggunakan/<i>include/uses</i></p> <p>&lt;&lt;include&gt;&gt;</p>  <p>&lt;&lt;uses&gt;&gt;</p> 	<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</p>

Sumber: A.S. dan Shalahuddin (2011: 131)

## 2. Activity diagram

*Activity diagram* merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Jadi dapat dikatakan bahwa *activity diagram* menggambarkan aktifitas sistem, bukan apa yang dilakukan oleh aktor. Simbol-simbol yang digunakan dalam *activity diagram* ditampilkan dalam tabel berikut (A.S. dan Shalahuddin: 2011: 134-135).

**Tabel 2.5** Simbol *Activity Diagram*

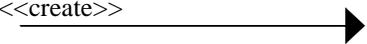
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktifitas 	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu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

Sumber: A.S. dan Shalahuddin (2011: 134-135)

### 3. *Sequence diagram*

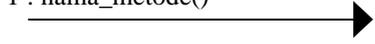
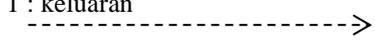
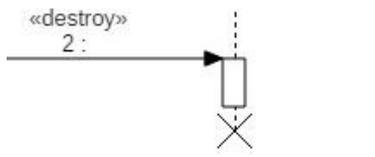
*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup (*life cycle*) objek dan *message* (pesan) yang dikirimkan dan diterima antar objek. Jumlah *sequence diagram* yang harus digambar minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri. Semakin banyak *use case* yang didefinisikan semakin banyak pula *sequence diagram* yang harus dibuat. Simbol-simbol yang digunakan pada *sequence diagram* ditampilkan dalam tabel berikut (A.S. dan Shalahuddin, 2011: 137-138).

**Tabel 2.6** Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor/ <i>actor</i>  nama aktor	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 frase nama aktor
Garis hidup/ <i>lifeline</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat

Sumber: A.S. dan Shalahuddin (2011: 138- 139)

**Tabel 2.6** Lanjutan

<p>pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.</p>
<p>Pesan tipe <i>send</i></p> <p>1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju</p>
<p>pesan tipe <i>return</i></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 penerima</p>
<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri</p>

Sumber: A.S. dan Shalahuddin (2011: 138- 139)

## 2.4 Penelitian Terdahulu

Untuk mendukung teori yang berkaitan dengan penelitian, peneliti mencantumkan beberapa penelitian terdahulu di bidang sistem pakar dalam kategori diagnosis.

**Hersatoto Listiyono** ( 2008 ), Merancang dan Membuat Sistem Pakar. Suatu sistem pakar adalah suatu sistem komputer yang menyamai kemampuan pengambilan keputusan dari seorang pakar. Suatu emulsi jauh lebih kuat daripada suatu simulasi yang hanya membutuhkan sesuatu yang bersifat nyata dalam beberapa bidang atau hal. *Knowledge* dalam sistem pakar mungkin saja seorang ahli, atau *knowledge* yang umumnya terdapat dalam buku, majalah dan orang yang mempunyai *knowledge* tentang suatu bidang. Istilah sistem pakar, sistem

basis-pengetahuan (*knowledgebase*), atau sistem pakar basis-pengetahuan (*knowledge-base*), sering digunakan dalam arti yang sama.

**Abid Yanuar Badharudin, dan Dwi Aryanto** ( 2011 ), Aplikasi Sistem Pakar untuk Mendiagnosa Kerusakan Monitor CRT (*Cathode Ray Tube*) (*Application of Expert System to Diagnose the Damages in The CRT (Cathode Ray Tube ) Monitor*), Layar monitor merupakan komponen utama pendukung keutuhan suatu perangkat komputer. Perkembangan layar monitor dimulai saat IBM meluncurkan monitor dengan resolusi 640x340 pixel dengan tulisan *monochrome* dengan warna hijau, hingga model CRT maupun LCD. Kompleksnya infrastruktur pada perangkat layar monitor menyulitkan pengguna komputer, apa yang harus dilakukan jika mengalami kendala kerusakan pada layar monitor komputernya. Komputer telah berkembang sejak penemuan mesin hitung yang diprakarsai oleh Turing, meskipun Turing adalah merupakan pengembangan dari mesin hitung konvensional, yang bertujuan sebagai alat bantu bagi manusia. Bahkan komputer pada saat ini juga turut berperan dalam pemberian keputusan. Komponen penelusuran atau mesin inferensi dalam sistem pakar mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan masalah. Mesin inferensi merupakan mesin program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan. Penelusuran dalam sistem pakar memiliki dua pendekatan kontrol, yaitu runut mundur atau pelacakan ke belakang (*backward chaining*) dan runut maju atau pelacakan ke depan (*forward chaining*). Dalam runut mundur dimulai

dari tujuan (*goal-driven*) selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya.

**Dharmawan Hadi Santoso dan Harjono** ( 2012 ), Mendiagnosa Kerusakan Handphone Menggunakan Aplikasi Sistem Pakar (*Identity the Mobile Phone's Damage Type Using Expert System*). *Handphone* atau biasa disebut dengan telepon genggam yang dahulu termasuk barang mewah, namun dengan seiring berkembangnya zaman dan semakin bertambahnya perusahaan pembuat *handphone* membuat harga *handphone* semakin murah sehingga hamper semua orang bisa memilikinya. Seiring berkembangnya teknologi saat ini *handphone* tidak hanya memiliki fungsi untuk menelepon dan mengirim pesan, pada beberapa *handphone* kelas atas (*smartphone*) bahkan hampir memiliki fungsi seperti komputer. Dengan adanya *handphone* sangat membantu kelancaran kegiatan manusia, karena *handphone* bersifat mobile sehingga tak jarang *handphone* rusak karena terjatuh, terkena air atau karena salah dalam perawatan dan penggunaan.

**Imam Gunawan** (2013), Perancangan sistem pakar untuk diagnose kerusakan hardware laptop, Kasus kerusakan hardware laptop merupakan kasus yang memerlukan bantuan seorang pakar (teknisi) dalam menyelesaikan masalah dengan mengandalkan pengetahuan yang dimilikinya, tetapi kendala yang sering dialami apabila mengandalkan jasa seorang pakar adalah biaya yang harus dikeluarkan relatif tidak sedikit dan juga seorang pakar tidak akan selalu ada di tempat untuk melayani client setiap saat, apalagi jarak yang harus di tempuh ke tempat servis mungkin saja sangat jauh. Pada Penelitian ini dibuat perangkat lunak untuk mengatasi kasus tersebut, dimana perangkat lunak yang dimaksud

adalah sistem pakar untuk diagnosis kerusakan hardware laptop yang di desain untuk memodelkan/mengemulasi kemampuan seorang pakar dalam memecahkan suatu masalah yang berbasiskan pada pengetahuan pakar itu sendiri. Pengetahuan yang di dapat dari proses akuisisi pengetahuan direpresentasikan dengan menggunakan model kaidah produksi, sementara itu dalam mengembangkan mesin inferensi teknik yang digunakan yaitu forward chaining. Dalam mendeskripsikan alur program digunakan dua pendekatan, yaitu flowmap untuk mendeskripsikan mekanisme kerja aplikasi sistem pakar yang dirancang dan diagram aliran data (*DFD*) untuk mendeskripsikan proses aliran data yang ada dalam aplikasi sistem pakar yang dirancang. Untuk menggambarkan pemodelan data atau desain basis data digunakan model entity relationship diagram yang selanjutnya ditransformasikan ke dalam bentuk basis data fisik, yaitu dalam bentuk tabel-tabel yang diuji tingkat akurasi datanya menggunakan teknik normalisasi dan diteruskan dengan penjelasan objek data pada setiap tabel dengan menggunakan kamus data. Dengan fasilitas yang diberikan untuk pemakai dan pakar, memungkinkan baik pemakai maupun pakar untuk menggunakan sistem ini sesuai kebutuhannya masing-masing. Pemakai diberi kemudahan untuk mendapatkan solusi atas kerusakan hardware yang terjadi berdasarkan data-data yang telah dipilih pada proses penelusuran, yang terdiri dari pemilihan data macam komponen hardware yang mengalami kerusakan, diteruskan dengan pemilihan data jenis komponen hardware yang mengalami kerusakan, dilanjutkan dengan pemilihan data gejala kerusakan yang dialami dan pemilihan data ciri

kerusakan yang di dapat. Sedangkan pakar diberi kemudahan dalam manajemen sistem, baik proses tambah, hapus maupun update data terkait.

**Harison dan Alexyusandera** (2014), Sistem pakar Perawatan dan perbaikan mobil bensin menggunakan video tutorial berbasis web. Teknologi informasi bisa memberikan peranan yang sangat berarti dalam membantu masyarakat dalam menyelesaikan permasalahan yang mereka hadapi tanpa harus mengeluarkan dana yang besar, karena bagi mereka yang mau menerapkan aplikasi atau teknologi informasi yang akan dirancang ini maka mereka bisa mengerjakannya dengan sendiri. Tujuan dari penelitian ini adalah membantu para pengguna atau pemilik kendaraan roda empat atau mobil dalam perawatan dan perbaikan ringan kendaraan mereka tanpa harus membawa kebengkel, jika dibengkel mereka akan mengeluarkan biaya pelayan dan beli barang atau komponen yang harus ditukar, tetapi jika mereka mau menerapkan sistem pakar yang akan dirancang ini maka mereka hanya mengeluarkan dana untuk pembelian komponen saja dan mereka akan bisa lebih efisien dalam penukaran komponen mobil jika mereka mengikuti cara-cara perawatan mobil mereka. Metode yang akan dipergunakan dalam merancang sistem pakar ini adalah metode Forward chining 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 yang disusun tersebut diperoleh suatu kesimpulan dari permasalahan tersebut, kesimpulan ini merupakan data yang diambil dari

pakar yang berupa tulisan, lisan dan videoperbaikan ringan yang nantinya bisa diakses oleh pengguna.

**Ahmad Kariadi, Budi Rahman, Syahid Natarsyah ( 2014 ),** Sistem Pakar Deteksi Kerusakan *PLC (Power Line Carrier)* dengan *Metode Forward chaining*. kehidupan dunia diwarnai dengan penerapan teknologi. Terutama pada PT PLN (Persero) AP2B Sistem Kalselteng yang terdapat peralatan *PLC (Power Line Carrier)*, peralatan ini sangat sensitif jika terjadi kerusakan. Karena dapat mengakibatkan gagalnya komunikasi antar Gardu Induk PLN yang berakibat fatal pada sistem penyaluran tenaga listrik serta penurunan penilaian kinerja peralatan. 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 pada *PLC (Power Line Carrier)* dicoba untuk dirangkum dan dianalisis, sehingga hasil diagnosa terhadap kerusakan dapat diberikan kesimpulan dan solusinya. Hasil dari penelitian ini diterapkan pada PT PLN (Persero) AP2B Sistem Kalselteng.

**Rosmawati Tamin ( 2015 ).** Sistem Pakar untuk Diagnosa Kerusakan Pada Printer Menggunakan Metode *Forward Chaining*. Mensubtitusikan pengetahuan manusia ke dalam bentuk sistem dengan menampung kemampuan/keahlian seorang pakar untuk melakukan proses analisa suatu masalah sehingga sistem dapat sistem bekerja menyelesaikan masalah sebagaimana manusia mengerjakannya dan menyelesaikan masalah tersebut. Kerusakan printer terkadang menjadi masalah besar ketika seorang yang awam tidak mengetahui

letak kesalahan printer maka dibutuhkan sistem yang mampu bekerja otomatis untuk memberikan solusi kerusakan printer. Penelitian ini bertujuan untuk merancang sebuah sistem yang dapat digunakan untuk menangani kerusakan printer. Pengguna aplikasi ini seolah-olah berhadapan langsung dengan pakar dibidang hardware khususnya printer. Perencanaan sistem dilakukan dengan membuat *knowledge base* menggunakan *decision tree* dan Aturan *if-then* sebagai representasi pengetahuan. Sistem dibuat dengan menggunakan metode *forward chaining* dan bahasa pemrograman Visual Basic. Hasil penelitian ini mengungkapka jenis-jenis kerusakan yang terjadi pada sebuah printer serta penanganan dari kerusakan tersebut. Pengujian aplikasi juga dilakukan untuk mengetahui akurasi dan variasi serta *user frindly* dan fleksibilitas sistem. Hasil dari keseluruhan pengujian ini dapat disimpulkan bahwa program sudah cukup baik walaupun jenis kerusakan yang dihasilkan belum lengkap karena pada sistem ini hanya mendeteksi 15 jenis kerusakan mesin secara umum.

**Ahmad Hilmi , Dini Destiani** ( 2015 ), Pengembangan sistem pakar diagnosa kerusakan sepeda motor automatic non injeksi berbasis android. Sepeda motor merupakan alat transportasi yang banyak di sukai oleh masyarakat Indonesia, karena itu pengetahuan mengenai sepeda motor, khususnya jika ada kerusakan perlu dikuasai oleh penggunanya. sistem yang di kembangkan ini untuk mendiagnosis kerusakan sepeda motor disebut dengan sistem pakar diagnosis kerusakan sepeda motor automatic non injeksi. Tujuan penelitian ini adalah mengembangkan sistem pakar diagnosis kerusakan sepeda motor automatic non injeksi dimana pengembangannya menggunakan metode *ESDLC (Expert System*

*Development Life Cycle*) dari Durkin (1994) dimana tahapan yang dilakukan merupakan langkah untuk membangun basis pengetahuan dalam bentuk aturan yang digunakan dalam mendiagnosis kerusakan sepeda motor automatic non injeksi. Tahapan metode dimulai dari assessment, akuisisi pengetahuan, desain, pengujian, dokumentasi dan pemeliharaan. Berdasarkan tahapan yang telah dilakukan maka diperoleh suatu prototype sistem pakar diagnosis kerusakan sepeda motor automatic non injeksi dengan menggunakan bahasa pemrograman android. Sistem pakar ini menyediakan fasilitas berupa halaman yang berisi tentang sistem pakar diagnosis kerusakan sepeda motor automatic non injeksi, kemudian halaman daftar kerusakan, kemudian pengguna bisa melakukan konsultasi mengenai kerusakan sepeda motor automatic non injeksi sesuai dengan gejalanya, sehingga sistem akan menampilkan hasil mendiagnosis kerusakan sepeda motor automatic non injeksi berupa nama kerusakan kendaraan beserta solusinya.

**Andi Tia Ari Wibowo, Tedi Lesmana Marselino** (2016), Aplikasi Bantu Diagnosa masalah sepeda motor Honda beat tipe matic berbasis mobile android, Sepeda motor “Honda Beat” tipe matic adalah salah satu kendaraan favorit di Indonesia, Namun masih banyak pengguna yang tidak memiliki pengetahuan tentang pernaikan sepeda motor. Berdasarkan latar belakang tersebut peneliti bermaksud membuat aplikasi sistem pakar, agar pengguna sepeda motor dapat mengetahui masalah pada tahap awal dan solusi yang tepat. Untuk mendapatkan diagnosis masalah, peneliti menggunakan teknik forward chaining. Penelitian ini menggunakan bahasa pemrograman Java dengan *IDE Eclipse* dan *XAMPP*

*MySQL database* yang meliputi tahap identifikasi, formalisasi, implementasi, dan pengujian. Teknik pengumpulan data pada penelitian ini dilakukan dengan cara obserfasi. Hasil dari penelitian ini berupa aplikasi bantu dognosis masalah sepeda motor “Honda Beat” tipe matic berbasis piranti bergerak android.

## 2.5 Kerangka Pemikiran

Kerangka pemikiran memuat pemikiran terhadap alur yang dipahami sebagai acuan dalam pemecahan masalah yang diteliti secara logis dan sistematis. Kerangka berfikir yang baik akan menjelaskan secara teoritis pertautan antar variabel yang diteliti (Sugiyono, 2012: 60). Berikut ini adalah kerangka pemikiran yang mendasari penelitian ini.



**Gambar 2.25** Kerangka Pemikiran  
(Sumber: Data Penelitian: 2016)

Data-data yang dibutuhkan berkaitan dengan permasalahan mesin *NXT* dianalisis terlebih dahulu agar lebih sederhana dan mudah dilakukan proses pengolahan datanya. Data-data tersebut kemudian diolah menggunakan metode sistem pakar *forward chaining* untuk membuat aturan (*rule*) yang akan digunakan. Sistem pakar dengan metode *forward chaining* dibuat menggunakan bahasa pemrograman *PHP* dan *database MySQL* sehingga menghasilkan sebuah sistem

pakar untuk mendeteksi kerusakan mesin *NXT* menggunakan metode *forward chaining* berbasis *web*.