BAB II
TINJAUAN PUSTAKA

2.1. Teori Dasar

Deskripsi teori merupakan uraian sistematis tentang teori (bukan sekedar pendapat pakar atau penulis buku) dan hasil penelitian yang *relevan* dengan *variabel* yang diteliti. Bila dalam suatu penelitan terdapat tiga variabel bebas dan satu variabel terikat, kelompok teori yan perlu dideskripsikan ada empat, yaitu kelompok teori yang berkenaan dengan tiga variabel bebas dan satu variabel terikat. Oleh karena itu, semakin banyak variabel yang akan diteliti, semakin banyak pula teori yang perlu dikemukakan. *Deskripsi* teori paling tidak berisi penjelasan tentang variabel-variabel yang diteliti melalui pendefenisian dan urairan yang lengkap serta mendalam dari berbagai *referensi* sehingga ruang lingkup, kedudukan dan prediksi terhadap hubungan antar variabel yang akan diteliti menjadi lebih jelas dan terarah (Sudaryono, 2015:15).

2.1.1 Kecerdasan Buatan *(Artificial Intelligent)*

Menurut Suyanto (2014:11) para ilmuan memiliki dua cara pandang yang berbeda tentang AI. Yang pertama memandang AI sebagai bidang ilmu yang hanya fokus pada proses berpikir. Sedangkan yang ke dua memadang AI sebagai bidang ilmu yang fokus pada tingkah laku. Cara pandang ke dua melihat AI secara lebih luas karena suatu tingkah laku pastilah didahului proses pikir. *Defenisi* yang

paling tepat untuk saat ini adalah *acting rationally* dengan pendekatan *rational agent.* Hal ini berdasarkan pemikiran bahwa komputer bisa melakukan penalaran secara *logis* dan juga bisa melakukan aksi secara *rasional* berdasarkan hasil penalaran tersebut.

Kecerdasan Buatan *(Artificial Intelligence* atau AI) *didefenisikan* sebagai kecerdasan *entitas* alami. Sistem seperti ini pada umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer *(games)* logika *fuzzy*, jaringan syaraf tiruan dan robotika (Husda, 2012:192)

Kecerdasan Buatan *(Artificial Intelligence)* adalah suatu sistem informasi yang berhubungan dengan penangkapan, pemodelan dan penyimpanan kecerdasan manusia dalam sebuah sistem teknologi informasi sehingga sistem tersebut memiliki kecerdasan seperti yang dimiliki manusia. Sistem ini dikembangkan untuk mengembangkan metode dan sistem untuk menyelesaikan masalah, biasanya diselesaikan melalui aktivitas *intelektual* manusia (Husda, 2012:192).

Menurut Husda (2012: 194) komponen kecerdasan buatan adalah:

1. Basis Pengetahuan

   Basis pengetahuan berisi pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan memecahkan masalah. Basis pengetahuan tersusun atas 2 elemen dasar, yaitu:

a. Fakta

   Misalnya: situasi, kondisi, dan kenyataan dari permasalahan yang ada, serta teori dalam bidang itu.

b. Aturan

   Yang mengarahkan pengguna pengetahuan untuk memecahkan masalah yang spesifik dalam bidang yang khusus.

2. Mesin *Inferensi*

   Mesin Inferensi *(Inference Engine),* merupakan otak dari kecerdasan buatan. Juga dikenal sebagai penerjemah aturan *(rule interpreter).* Komponen ini berupa program komputer yang menyediakan suatu *metodologi* untuk memikirkan *(reasoning)* dan memformulasikan kesimpulan. Kenerja mesin inferensi meliputi:

   a. Menentukan aturan mana yang akan dipakai.
   b. Menyajikan pernyataan kepada pemakai ketika diperlukan.
   c. Menambah jawaban ke dalam memori kecerdasan buatan dan sistem pakar.
   d. Menyimpan fakta baru dari sebuah aturan.
   e. Menambah fakta tadi (yang telah diperoleh) ke dalam memori.

3. *Interface*

   Interface kecerdasan buatan dan sistem pakar mengatur komunikasi antara pengguna dan komputer. Komunikasi ini paling baik berupa bahasa alami, biasanya disajikan dalam bentuk tanya-jawab dan kadang ditampilkan dalam

bentuk gambar/grafik. Antarmuka lebih canggih dilengkapi dengan percakapan *(voice communication).*

Sedangkan sistem cerdas yang paling banyak dikembangkan saat ini adalah (Husda, 2012: 196):

1. Sistem Pakar

   Yaitu program konsultasi *(advisory)* yang mencoba menirukan proses penalaran seorang pakar/ahli dalam memecahkan masalah yang rumit.

2. Pemrosesan Bahasa Alami *(Natural Language Processing)*

   Pemrosesan bahas alami *(Natural Language Precessing)* yang memberi kemampuan pengguna komputer untuk berkomunikasi dengan komputer dalam bahasa mereka sendiri (bahasa manusia). Sehingga komunikasi dapat dilakukan dengan cara percakapan alih-alih menggunakan perintah yang biasa digunakan dalam bahasa komputer biasa.

3. Pemahaman Ucapan/suara *(Speech/Voice Understanding).*

   Adalah teknik agar komputer dapat mengenali dan memahami bahasa ucapan. Proses ini mengijinkan seseorang berkomunikasi dengan komputer dengan cara berbicara kepadanya.

4. Sistem Sensor dan Robotika

   Sistem sensor, seperti sistem *visi* dan pencitraan, serta sistem pengolahan sinyal, merupakan bagian dari robotika. Sebuah robot, yaitu perangkat *elektromekanik* yang diprogram untuk melakukan tugas manual, tidak semuanya merupakan bagian dari AI. Robot yang cerdas biasanya

mempunyai perangkat sensor, seperti kamera, yang mengumpulkan informasi mengenai operasi dan lingkunganya.

5. Komputer *Vision*

   Merupakan kombinasi pencitraan, pengolahan citra, pengenalan pola serta proses pengambilan keputusan. Tujuan utama dari komputer *vision* adalah untuk menerjemahkan suatu pemandangan. Komputer *vision* banyak dipakai dalam kendali kualitas produk industri.

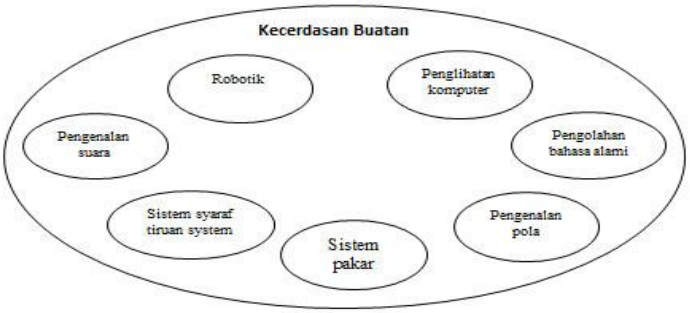
6. *Intelligent Tutoring/Intelligent Computer-Aided Instruction*

   Adalah komputer yang mengajari manusia. Belajar melalui komputer sudah lama digunakan, namun dengan menambahkan aspek kecerdasan di dalamnya, dapat diciptakan komputer "guru" yang dapat mengatur teknik pengajarannya untuk menyesuaikan dengan kebutuhan "murid" secara *individual*. Sistem ini juga mendukung pembelajaran bagi orang yang mempunyai kekurangan fisik atau kelemahan belajar.

7. Mesin Belajar *(Machine Learning)*

   Yang berhubungan dengan sekumpulan metode untuk mencoba mangajari/melatih komputer untuk memecahkan masalah atau mendukung usaha pemecahan masalah dengan menganalisa kasus-kasus yang telah terjadi.

8. Aplikasi lain dari AI misalnya untuk merangkum berita, pemrograman komputer secara otomatis, atau menerjemahkan dari suatu bahasa kebahasa lain, serta aplikasi dalam permainan.



Gambar 2.1. *Area* Dari *Artificial Intelligence*
Sumber : Iswanti dan Hartati (2008:2)

Dapat disimpulkan, sistem pakar merupakan bagian dari AI, dimana selain sistem pakar yang menggunakan AI, ada beberapa yang lain diantaranya *games*, logika *fuzzy*, jaringan syaraf tiruan, dan robotika (Husda, 2012:200)

2.1.1.1 Jaringan Syaraf Tiruan

Menurut Suyanto (2014: 169) **Jaringan Syaraf Tiruan** merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasikan sel-sel penyusunan yang disebut *neuron*, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektivitas yang sangat tinggi.

Menurut Suyanto (2014: 174) Secara umum kita bisa membagi arsitektur jaringan menjadi empat, yaitu:

1. ***Single-layer Feedforward Networks***

Suatu JST berlapis adalah jaringan *neuron* yang diorganisasikan dalam bentuk lapisan-lapisan. Pada bentuk jaringan berlapis yang paling sederhana,

hanya terdapat input layer dengan *nude* sumber yang terproyeksi ke dalam *output layer* dari *neuron (computation nodes).* Tetapi tidak sebaliknya. Dengan kata lain, jaringan ini adalah jaringan jenis *feedforward.*

2. *Multi-layer Feedforward Networks*

Kelas kedua dari *feedforward neural network* adalah jaringan dengan satu atau lebih lapisan tersembunyi *(hidden layer),* dengan *computation nodes* yang berhubungan disebut *hidden neurons* atau *hidden units.*

3. Recurrent Networks

Recurrent neural network adalah jaringan yang mempunyai minimal satu *feedback loop.* Sebagai contoh, suatu *recurrent network* bisa terdiri dari satu lapisan *neuron* tunggal dengan masing-masing *neuron* memberikan kembali *outputnya* sebagai *input* pada semua *neuron* yang lain.

4. Lattice structure

Sebuah *Lattice* (kisi-kisi) terdiri dari satu dimensi, dua dimensi, atau lebih *array neuron* dengan himpunan *node* sumber yang bersesuaian yang memberikan sinyal *input* ke *array,* dimensi *lattice* mengacu pada jumlah dimensi ruang dimana *graph* berada.

Menurut Suyanto (2014: 178) terdapat banyak ide dan defenisi yang berhubungan dengan “belajar”. Mendefenisikan belajar dalam konteks JST sebagai berikut: "belajar adalah suatu proses dimana parameter-parameter bebas JST diadaptasi melalui suatu proses perangsangan berkelanjutan oleh lingkungan dimana jaringan berada. Jenis belajar ditentukan oleh pola dimana pengubah parameter dilakukan”.

1. JST dengan Metode Belajar *Supervised Learning*

Menurut Suyanto (2014:184) banyak model JST yang tergolong dalam *supervised Learning*, tetapi hanya membahas 2 model saja, yaitu JST dengan arsitektur *Multi layer Perceptron* (MLP) dan JST *Probabilistik.*

a. *Multi Layer Perceptron* (MLP)

Merupakan model JST yang paling banyak digunakan dalam bidang pendidikan dan aplikasi. Banyak algoritma yang diusulkan untuk melatih MLP. Salah satu yang populer adalah algoritma pelatihan ***Back Propagasi* atau *Propagasi* Balik**. Sesuai dengan namanya, *algoritma* ini melakukan dua tahap perhitungan, yaitu perhitungan maju untuk menghitung galat antara keluaran dan aktual dan target; dan perhitungan mundur yang mempropagasikan balik galat tersebut untuk memperbaiki bobot-bobot sinaptik pada semua *neuron* yang ada.

b. JST *Probabilistik*

JST *Probabilistik* merupakan model yang dibentuk berdasarkan panaksiran fungsi padat peluang. Model ini memberikan unjuk kerja pengklasifikasian yang sangat baik dan cepat dalam pelatihan karena dilakukan hanya dalam satu tahap pelatihan. **Metode *Bayes*** untuk pengklasifikasian pola menggunakan suatu aturan pengambilan keputusan yang meminimalkan "resiko yang diharapkan" *Arsitektur* JST *Probabilistik* memiliki empat lapisan yang terdiri dari:

1. lapisan dengan *m* unit masukan yang menerima vektor masukan *x.*
2. lapisan unit-unit pola yang terhubung penuh dengan pola masukan.
3. Lapisan unit-unit hasil penjumlahan yang terhubung penuh dengan tiap kelas.
4. Lapisan keputusan untuk memilih nilai yang terbesar.

2. *Unsupervised Learning* (Belajar Tanpa Pengawasan)

Menurut Suyanto (2014:179) *unsupervised atau self-organized learning* tidak membutuhkan guru untuk memantau proses belajar. Dengan kata lain, tidak ada sekumpulan sampel *input-output* atau fungsi tertentu untuk dipelajari oleh jaringan. Salah satu contoh *unsupervised learning* adalah *competitive learning.* Sebagai contoh, kita bisa menggunakan JST yang terdiri dari dua lapisan, satu lapisan masukan dan satu kompetitif. Lapisan masukan menerima data yang disediakan. Lapisan kompetitif terdiri dari *neuron-neuron* yang saling bersaing untuk meraih "kesempatan" memberi respon ke ciri khas yang berisi data masukan. Dalam bentuk paling sederhana, jaringan beroperasi berdasarkan strategi *"winner- takes-all".*

2.1.1.2 Logika fuzzy

Menurut jurnal penelitian Kurnialensya dan Syukur (2013: 64) ***Logika Fuzzy*** merupakan sebuah nilai yang memiliki kesamaran *(fuzziness)* antara benar dan salah. Dalam teori logika *fuzzy* sebuah nilai bisa bernilai benar dan salah secara bersamaan tetapi berapa besar kebenaran dan kesalahan suatu nilai tergantung dari berapa besar bobot keanggotaan yang dimilikinya. Dalam teori logika fuzzy dikenal himpunan *fuzzy (fuzzy Set)* yang merupakan pengelompokan sesuatu berdasarkan variable bahasa *(linguistik variable)* yang dinyatakan dalam fungsi keanggotaan yang bernilai nol sampai dengan satu. *Fuzzy Inferensi System (FIS)* adalag proses merumuskan pemetaan dari *input* yang diberikan ke *output* dengan menggunakan logika *fuzzy*. Pemetaan tersebut akan menjadi dasar dari

keputusan yang akan dibuat. *FIS* berisi metode-metode untuk melakukan *inferensi fuzzy*, antara lain *Metode Tsukamoto, Metode Mamdani, Metode Sugeno*. Perbedaan antara ketiga sistem *inferensi* samar terdapat pada konsekuen dari aturan samar, *aggregasi* dan prosedur *defuzzyfikasi*.

2.1.1.3 Sistem Pakar

Menurut Kusrini (2008:3) **Sistem Pakar** adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Ada 2 metode *inferensi* yang penting dalam sistem pakar yaitu: runut maju *(forward chaining)* dan runut balik *(backward chaining).*

Menurut Hayadi (2016: 1) sistem pakar atau *expert system* biasa disebut juga dengan *Knowledge Base System* yaitu suatu aplikasi *computer* yang ditujukan untuk membantu pengambilan keputusan atau memecahkan persoalan dalam bidang yang spesifik. Sistem ini bekerja dengan menggunakan pengetahuan dan metode analisis yang telah didefenisikan terlebih dahulu oleh pakar yang sesuai dengan bidang keahliannya. Sistem ini disebut sistem pakar karena fungsi dan peranannya sama seperti orang seorang ahli yang harus memiliki pengetahuan, pengalaman dalam memecahkan masalah suatu persoalan.

1. Pengertian Sistem Pakar Menurut Beberapa Ahli

Arhami (2004) dalam Hayadi (2016: 2) Sistem Pakar adalah cabang dari AI yang membuat penggunaan secara luas *Knowledge* yang khusus untuk menyelesaikan masalah tingkat manusia yang pakar.

Siswanto (2010) dalam Hayadi (2016: 2) Sistem pakar adalah suatu cabang dari *Artificial Intelligence (AI)* yang cukup tua karena sistem ini mulai dikembangkan 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 *Interensi/Inferensi Engine* yang melakukan penalaran atau pelacakan terhadap sesuatu atau fakta-fakta dan aturan kaidah yang ada dibasis pengetahuan setelah dilakukan pencarian sehingga dicapai kesimpulan.

Menurut Rosnelly (2012: 10) Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu. Sorang pakar memiliki kemampuan kapakaran, yaitu:

1. Dapat mengenali dan merumuskan suatu masalah
2. Menyelesaikan masalah dengan cepat dan tepat.
3. Menjelaskan solusi dari suatu masalah.
4. *Restrukturisasi* pengetahuan.
5. Belajar dari pengalaman
6. Memahami batas kemampuan.

2. Manfaat Dan Kekurangan Sistem Pakar

T.Sutejo, e.t. (2010) dalam Hayadi (2016: 2) sistem pakar menjadi sangat popular 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 pengatahuan dan kepakaran seseorang.
5. Memudahkan akses pengetahuan seorang pakar.
6. Bisa digunakan sebagai pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.
7. Meningkatkan pengetahuan untuk menyelesaikan masalah karena sistem pakar mengambil sumber pengetahuan dari banyak pakar.

Ada juga 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.

3. Ciri-Ciri Sistem Pakar

Menurut Hayadi (2016: 3) Ciri-ciri sistem pakar adalah sebagai berikut:

1. Terbatas pada *domain* keahlian tertentu.
2. Dapat memberikan penalaran untuk data yang tidak pasti.
3. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
4. Berdasarkan pada kaidah *rule* tertentu.

5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Pengetahuan dan mekanisme *inferensi* jelas terpisah.
7. Keluarannya bersifat anjuran.
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai yang dituntun oleh dialog dengan pemakai.

4. Klasifikasi Sistem Pakar

Siswanto (2004) dalam Hayadi (2016:4) *Klasifikasi* Sistem Pakar berdasarkan kegunaanya, yaitu:

1. Diagnosis

   a. Digunakan untuk merekomendasikan: obat untuk orang sakit, kerusakan mesin, kerusakan rangkaian elektronik.
   b. Menemukan apa masalah/kerusakan yang terjadi.
   c. Menggunakan pohon keputusan *(decision tree)* sebagai *representasi* pengetahuan.

2. Pengajaran.

   a. Digunakan untuk pengajaran, mulai dari SD sampai dengan PT.
   b. Membuat dignosa apa penyebab kekurangannya dari siswa, kemudian memberi cara untuk memperbaiki.

3. Interprestasi.

   Untuk menganalisa data yang tidak lengkap, tidak teratur, dan data yang *kontradiktif.* Misalnya untuk *interprestasi* citra.

4. Prediksi.
    a. Contoh: bagaimana seorang pakar *meteorologi* memprediksi cuaca besok berdasarkan data-data sebelumnya.
    b. Untuk peramalan cuaca.
    c. Penentuan masa tanam.
5. Perencanaan
    a. Mulai dari perencanaan mesin-mesin sampai dengan manajemen bisnis.
    b. Untuk menghemat biaya, waktu dan material, sebab pembuatan model.
    c. Sudah tidak diperlukan.
    d. Contoh: Sistem *kontingurasi* komputer.
6. *Control.*
    a. Digunakan untuk mengontrol kegiatan yang membutuhkan presisi waktu tinggi.
    b. Misal: pengontrolan pada industri-industri berteknologi tinggi.

5. Konsep Dasar Sistem Pakar

Menurut Hayadi (2016:5) Sistem Pakar terdiri dari beberapa konsep yang harus dimilikinya. Konsep dasar dari suatu sistem pakar sebagai berikut:

1. Keahlian

    Adalah suatu pengetahuan khusus yang diperoleh dari latihan, belajar dan pengetahuan. Pengetahuan dapat berupa fakta, teori, aturan, strategi *global* untuk memecahkan masalah.

2. Ahli *(Expert)*

   Melibatkan kegiatan mengenali dan *memformulasikan* permasalahan memecahkan masalah secara cepat dan tepat, menerangkan pemecahannya, belajar dari pengalaman, *merestrukturisasikan* pengetahuan, memecahkan aturan serta menentukan *relevansi.*

3. Mentransfer Keahlian *(Transfering Expertise)*

   Adalah proses *pentrasferan* keahlian dari seorang pakar kedalam komputer agar dapat digunakan oleh orang lain yang bukan pakar. Pengetahuan tersebut ditempatkan kedalam sebuah komponen yang dinamakan basis pengetahuan.

4. Menyimpulkan Aturan *(Inferencing Rule)*

   Merupakan kemampuan komputer yang telah diprogram. Penyimpulan ini dilakuan oleh mesin *inferensi* yang meliputi prosedur tentang penyelesaian masalah.

5. Peraturan *(Rule).*

   Diperlukan karena mayoritas dari sistem pakar bersifat *rule base sistem,* yang berarti pengetahuan disimpan dalam bentuk peraturan.

6. Kemampuan menjelaskan *(Explanation Capability)*

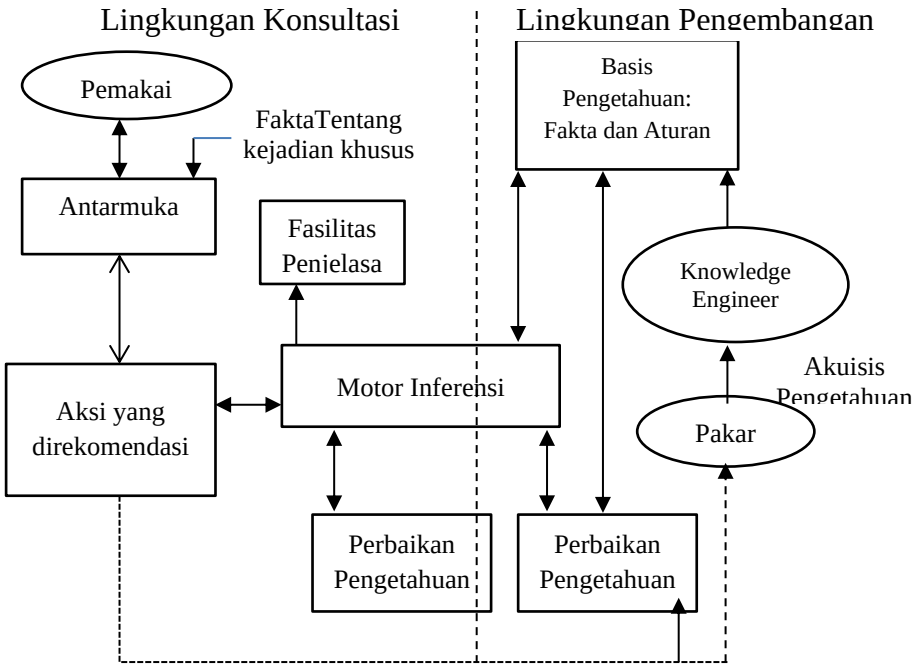
   Adalah karakteristik dari sistem pakar yang memiliki kemampuan menjelaskan atau memberi saran mengapa tindakan tertentu dianjurkan atau tidak dianjurkan.

6. Struktur Sistem Pakar

Menurut Husda (2012:182) sistem pakar disusun oleh 2 bagian utama, yaitu:

1. Lingkungan Pengembangan *(Development Environtment)*

2. Lingkungan Konsultasi *(Consultation Environtment)*



Gambar 2.2. *Arsitektur* Sistem Pakar
Sumber: Nur Elfi Husda (2012: 183)

Lingkungan pengembang sistem pakar digunakan untuk memasukkan pengetahuan pakar kedalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan untuk pengguna yang bukan pakar guna memperoleh pengetahuan pakar. Komponen- komponen sistem pakar dalam kedua bagian tersebut dapat dilihat dalam Gambar 2.2.

Komponen yang terdapat dalam arsitektur/struktur sistem pakar:

1. Antarmuka Pengguna *(User Interface)*

   Merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya kedalam bentuk yang dapat diterima oleh sistem.

2. Basis Pengetahuan

Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan menyelesaikan masalah. komponen sistem pakar ini disusun atas dua elemen dasar, yaitu:

1. Fakta: informasi tentang obyek dalam area permasalahan tertentu.
2. Aturan: informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

3. Akuisisi Pengetahuan *(Knowledge Acquisition)*

Akuisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan kedalam program komputer. Dalam tahap ini *knowledge engineering* berusaha menyerap pengetahuan untuk selanjutnya ditransfer kedalam *basis* pengetahuan. Pengetahuan diperoleh dari pakar, dilengkapi dengan buku, basis data, laporan penelitian, dan pengalaman pemakai.

Metode akuisisi pengetahuan:

a. Wawancara

Metode paling banyak digunakan, yang melibatkan pembicara dengan pakar secara langsung dalam suatu wawancara.

b. Analisis Protokol

Dalam metode ini pakar diminta untuk melakukan suatu pekerjaan dan mengungkapkan proses pemikirannya dengan menggunakan kata-kata. Pekerjaan tersebut direkam, ditulis dan dianalisis.

c. Observasi Pada Pekerjaan Pakar

Pekerjaan dalam bidang tertentu yang dilakukan pakar direkam dan diobservasi.

d. *Induksi* Aturan

Induksi adalah suatu proses penalaran dari khusus keumum. Salah satu sistem *induksi* aturan diberi contoh-contoh dari suatu masalah yang hasilnya telah diketahui. Setelah diberi beberapa contoh, sistem *induksi* aturan tersebut dapat membuat aturan yang benar untuk kasus-kasus contoh. Selanjutnya aturan dapat digunakan untuk menilai kasus lain yang hasilnya tidak diketahui.

4. Mesin/Motor Inferensi *(inference engine)*

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin *inferensi* adalah program komputer yang memberi *metodologi* untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace,* dan untuk memformulasikan kesimpulan.

5. Workplace / *Blackboard*

Workplace merupakan *area* dari sekumpulan memori kerja *(working memory)* yang digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara. Ada tiga keputusan yang dapat direkam:

1. Rencana: bagaimana menghadapi masalah.
2. Agenda: aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi.
3. Solusi : calon aksi yang akan dibangkitkan.

6. Fasilitas Penjelas

   Adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Digunakan untuk melacak respon dan memberi penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan:

   a. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
   b. Bagaimana *konklusi* dicapai?
   c. Mengapa ada alternatif yang dibatalkan?
   d. Rencana apa yang digunakan untuk mendapatkan solusi?

7. Perbaikan Pengetahuan

   Pakar memiliki kemampuan untuk menganalisa dan meningkatkan kinerjanya serta kemampuan untuk belajar dari kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran *terkomputerisasi*, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya dan juga mengevaluasi apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa sekarang.

7. Mesin Inferensi

Menurut Kusrini (2008: 8) *inferensi* merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. *Inferensi* adalah konklusi logis *(logical conclusion)* atau *implikasi* berdasarkan informasi yang tersedia. Dalam sistem pakar proses *inferensi* dilakukan dalam suatu modul yang disebut *Inferensi Engine* (mesin inferensi). Ketika *Representasi* Pengatahuan (RP) pada bagian *knowledge base* telah lengkap, atau paling tidak telah berada pada level yang cukup akurat, maka RP tersebut telah siap digunakan. *Inferensi Engine*

merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning.* Ada dua metode yang paling penting digunakan dalam *system* pakar yaitu: runut manju *(forward chaining)* dan runut balik *(backward chaining).*

T.Sutojo, e.t. (2010) dalam Hayadi (2016:9) Setiap *rule* terdiri dari dua bagian, yaitu bagian *IF* disebut *evidende* (fakta-fakta) dan bagian *THEN* disebut *Hipotesis* atau kesimpulan. *Syntax Rule* adalah:

> *IF* E *THEN* H
>
> E : *EVIDENCE* (fakta-fakta) yang ada
>
> H : *Hipotesis* atau kesimpulan yang dihasilkan

Secara umum, *rule* mempunyai *evidence* labih dari satu yang berhubungan oleh kata penghubung *AND* atau *OR*, atau kombinasi keduanya. Tetapi sebaiknya biasakan menghindari penggunaan *AND* dan *OR* secara sekaligus dalam satu *rule*.

> *IF* (E1 AND E2 AND E3……..*AND* En) *THEN* H
>
> *IF* (E1 OR E2 E3…….…….*OR* En) *THEN* H

Satu *evidence* bisa juga mempunyai hipotesis lebih dari satu.

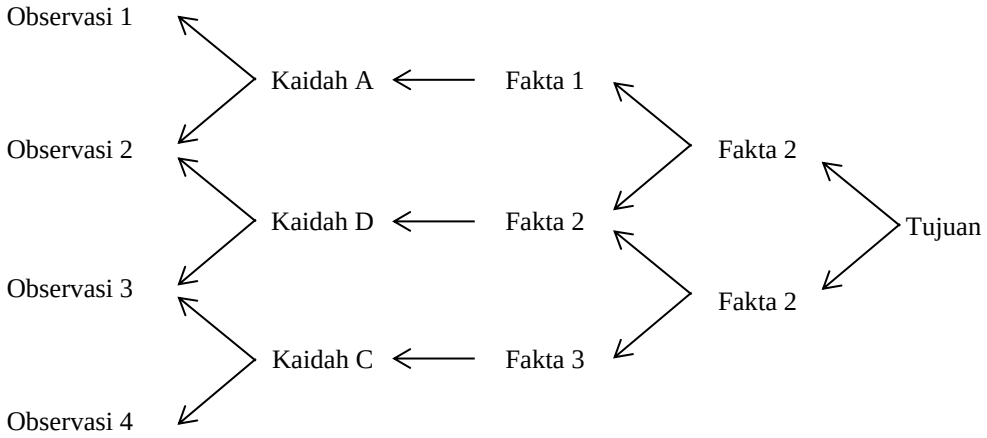
> *IF* E *THEN* (H1 *AND* H2 *AND* H3……..…..*AND* Hn)

a. Runut Balik *(Forward Chaining)*

Giarattano dan Riley (1994) dalam Kusrini (2008: 11) Runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam runut balik penalaran dimulai dengan tujuan kemudian merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut. Runut balik disebut juga sebagai *Goal-driven*

reasoning, merupakan cara yang efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur.

Schnupp (1989) dalam Kusrini (2008: 11) tujuan *inferensi* adalah mengambil pilihan terbaik dari banyak kemungkinan. Metode inferensi runut balik ini cocok digunakan untuk memecahkan masalah *diagnosis*.



Gambar 2.3. Diagram Pelacakan Ke Belakang (Backward Chaining)
Sumber: B.Herawan Hayadi (2016: 8)

b. Runut Maju *(Forward Chaining)*

Wilson (1998) dalam Kusrini (2008: 8) Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses ulang sampai ditemukan suatu hasil. Metode runut maju cocok digunakan untuk menangani masalah pengendalian *(controlling)* dan peramalan *(prognosis).* Ingin diperoleh konklusi dari daftar konklusi yang ada berdasarkan *premis-premis* dalam aturan dan fakta yang diberikan oleh *user*. Berikut adalah daftar aturannya:

Aturan 9 :

Jika Premis 1

Dan Premis 2

Dan Premis 3

Maka Konklusi 1

Aturan 10:

Jika Premis 1

Dan Premis 3

Dan Premis 4

Maka Konklusi 2

Aturan 11:

Jika Premis 2

Dan Premis 3

Dan Premis 5

Maka Konklusi 3

Aturan 12:

Jika Premis 1

Dan Premis 4

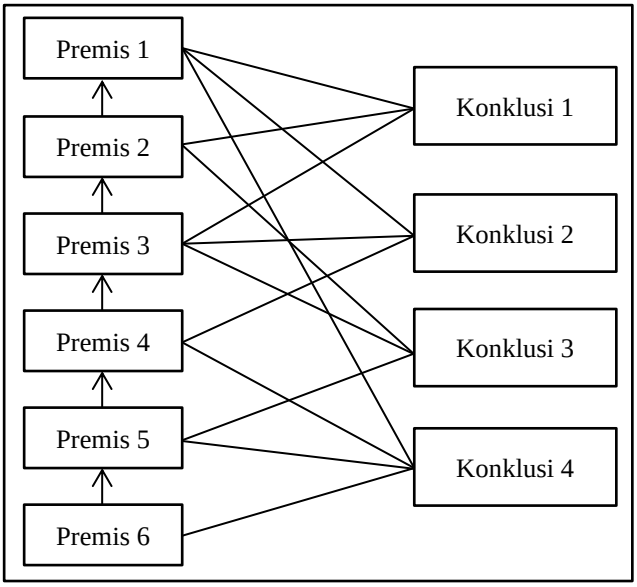
Dan Premis 5

Dan Premis 6

Maka Konklusi 4

Jika aturan ini kita gambarkan sebagai *graph* yang meletakan antara *premis-premis* dan *konklusi-konklusi* akan tampak seperti gambar 2.3.

Penelusuran maju pada kasus ini adalah untuk mengetahui apakah suatu fakta yang dialami oleh pengguna itu termasuk *konklusi 1, konklusi 2, konklusi3,* atau *konklusi 4* atau bahkan bukan salah satu dari *konklusi* tersebut, yang artinya sistem belum mampu mengambil kesimpulan karena keterbatasan aturan.



Gambar 2.4. *Graph* Pengetahuan
Sumber: Kusrini (2008: 10)

Dalam penalaran ini, *user* diminta memasukkan *premis-premis* yang dialami. Untuk memudahkan pengguna, sistem dapat memunculkan daftar *premis* yang mungkin sehingga *user* dapat memberikan umpan balik *premis* mana yang dialami dengan memilih salah satu beberapa dari daftar *premis* yang tersedia. Berarti daftar *premis* adalah:

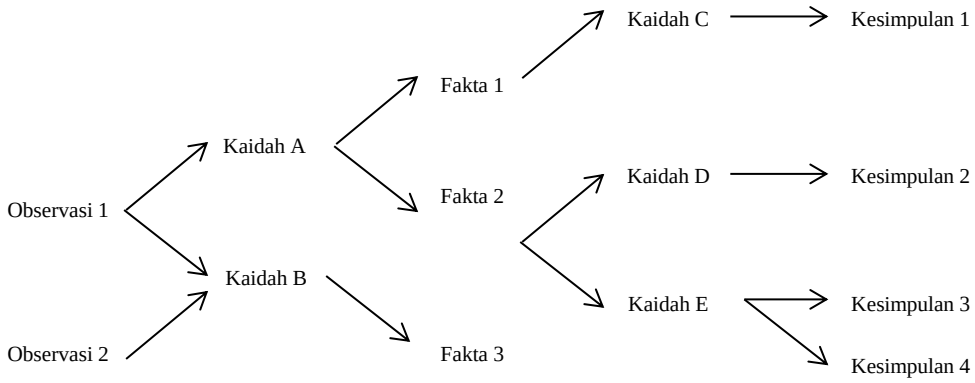
Premis 1, premis 2, premis 3,Premis 4, Premis 5 dan Premis 6

Berdasarkan *premis-premis* yang dipilih, maka sistem akan mencari aturan yang sesuai, sehingga akan diperoleh *konklusinya*.

Seandainya *user* memilih *Premis 1, Premis 2* Dan *Premis 3* maka aturan yang terpilih adalah aturan 1 dengan *konklusinya* adalah *konklusi 1*. Seandainya

user memilih *premis* 1 dan *premis* 6, maka sistem akan mengarahkan pada aturan 4 dengan *konklusinya* adalah adalah *konklusi* 4, tetapi karena aturan tersebut *premisnya* adalah *premis* 1, *premsi* 4, *premis* 5 dan *premis* 6, maka *premis-premis* yang dipilih oleh *user* tidak cukup untuk mengambil kesimpulan *konklusi* 4 sebagai *konklusi* terpilih.

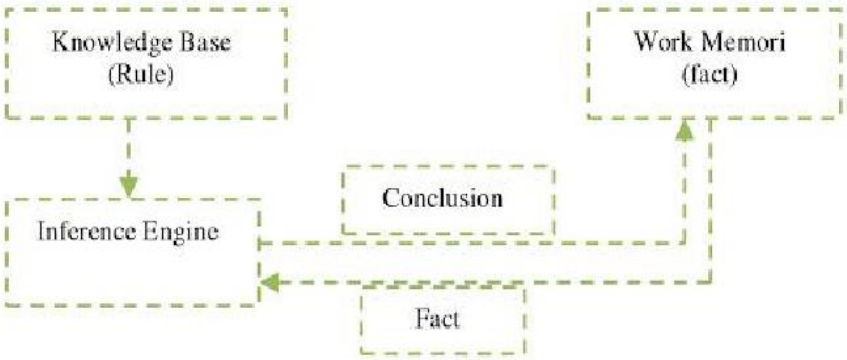
Menurut Russel S,Norvig P (2003) dalam Hayadi (2016: 9) Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan kedepan yang dimulai dengan informasi yang ada dan penggabungan *rule* untuk menghasilkan suatu kesimpulan atau tujuan.



Gambar 2.5 *Diagram* Pelacakan Kedepan (*Forward Chaining*)
Sumber: Hayadi, B. H. (2016: 3)

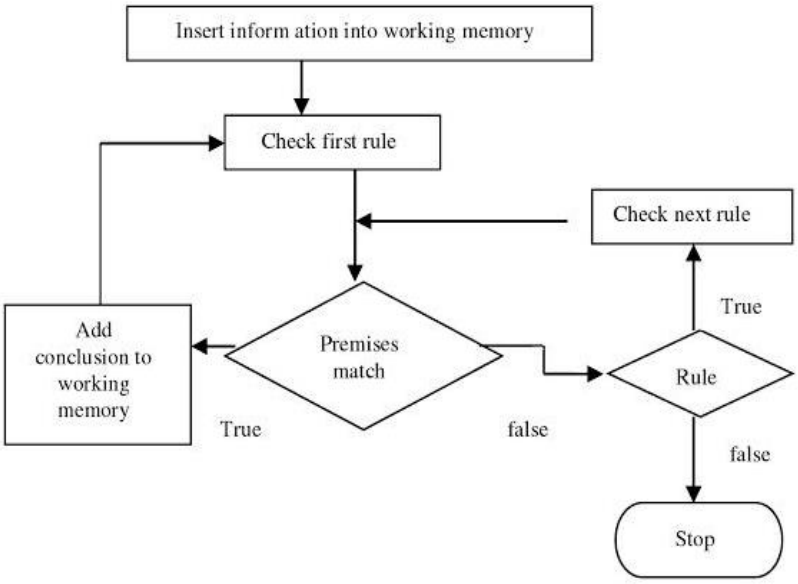
T.Sutojo.e.t.(2010) dalam Hayadi (2016: 10) *Forward Chaining* teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokan fakta-fakta tersebut dengan bagian *IF* dari *rules IF-THEN*. Bila ada fakta yang cocok dengan bagian *IF*. Maka *rule* tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian *THEN*) ditambah kedalam *database.* Setiap *rule* hanya boleh dieksekusi sekali saja.

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



Gambar 2.6. Model Basis Aturan
Sumber: B.Herawan Hayadi (2016: 10)

Operasi dari *forward chaining* dimulai dengan memasukkan sekumpulan fakta yang diketahui kedalam memori kerja *(working memory)*, kemudian menurunkan fakta baru berdasarkan aturan *premisnya* cocok dengan fakta yang diketahui. Operasi tersebut dapat digambarkan seperti gambar:



Gambar 2.7. Operasi Sistem *Forward Chaining*
Sumber: Hayadi, B. H. (2016: 10)

Menurut Hayadi (2016:11) langkah-langkah yang harus dilakukan dalam membuat sistem *forward chaining* berbasis aturan, yaitu:

1. Pendefenisian Masalah.

   Tahap ini meliputi pemilihan *domain* masalah dan *akuisis*i pengetahuan

2. Pendefenisian Data *Input*.

   Sistem *forward chaining* memerlukan data awal untuk memulai *inferensi*.

3. Pendefenisian Struktur Pengendalian Data.

   Aplikasi yang kompleks memerlukan premis tambahan untuk membantu mengendalikan pengaktifan suatu aturan.

4. Penulisan Kode Awal

   Tahap ini berguna untuk menentukan apakah sistem telah menangkap domain pengetahuan secara efektif dalam struktur aturan yang baik.

5. Pengujian Sistem.

   Pengujian sistem dilakukan dengan beberapa aturan untuk menguji sejauh mana sistem berjalan dengan baik.

6. Perancangan Antarmuka

   Antarmuka adalah suatu komponen penting dari suatu sistem. Perancangan antarmuka dibuat bersama-sama dengan pembuatan basis pengetahuan.

7. Pengembangan Sistem

   Pengambangan sistem meliputi penambahan antarmuka dan pengetahuan sesuai dengan prototype sistem.

8. Evaluasi Sistem

   Tahapan ini dilakukan pengujian sistem dengan masalah yang sebenarnya. Jika sistem belum berjalan dengan baik maka akan dilakukan pengembangan kembali.

2.1.2 Tabel Keputusan dan Pohon Keputusan

Menurut Hartati dan Iswanti (2008:26) tabel keputusan merupakan suatu cara untuk mendokumentasi pengetahuan. Tabel keputusan merupakan metrik kondisi yang dipertimbangkan dalam pendeskripsian kidah.

Kaidah yang disajikan dalam bentuk kaidah produksi disusun dari tabel keputusan (dibentuk dari pengubahan tabel keputusan). Meskipun kaidah secara langsung dapat dihasilkan dari tabel keputusan tetapi untuk menghasilkan kaidah yang efisien terdapat suatu langkah yang harus ditempuh yaitu membuat pohon keputusan telebih dahulu. Dari pohon keputusan dapat diketahui *atribut* (kondisi) yang dapat di *reproduksi* sehingga menghasilkan kaidah yang efisien dan normal.

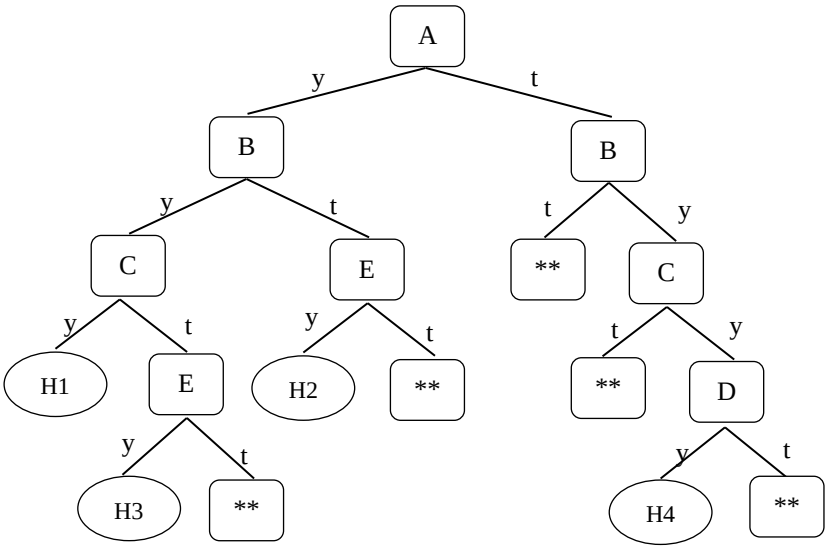
Berikut akan diberikan variasi contoh yang terkait dengan penyajian dalam bentuk tabel keputusan dan pohon keputusan.

Table 2.1 Tabel keputusan

Hipotesa / Evidence	Hipotesa 1	Hipotesa 1	Hipotesa 1	Hipotesa 1
Evicence A	Ya	Ya	Ya	Tidak
Evicence B	Ya	Tidak	Ya	Ya
Evicence C	Ya	Tidak	Tidak	Ya
Evicence D	Tidak	Tidak	Tidak	Ya
Evicence E	Tidak	Ya	Ya	Tidak

Sumber: Hartati dan Iswanti (2008:32)

Menurut Tabel 2.1 Tabel Keputusan dapat dibuat pohon keputusan sebagai berikut:



Keterangan:
A = evidence A, H1 = evidence 1, y = ya
B = evidence B, H2 = evidence 2, t = tidak
C = evidence C, H3 = evidence 3, ** = tidak menghasilkan hipotesa tertentu
D = evidence D, H4 = evidence 4

Gambar 2.8. Pohon Keputusan
Sumber: Hartati dan Iswanti (2008:33)

Dengan melihat pohon keputusan yang yang dihasilkan, dapat diketahui 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. demikian juga 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.

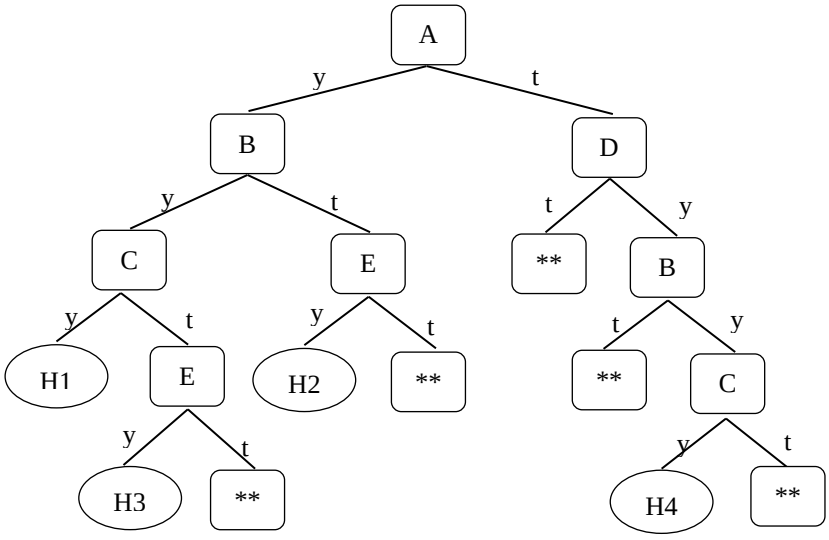
Dalam implementasi sistem pakar terutama dalam sesi konsultasi, *node-node* yang mewakili *evidence* biasanya akan menjadi pertanyaan yang akan

diajukan sistem. Mengacu pada Gambar 2.9. permasalahan bisa muncul awal sesi konsultasi pada saat sistem menanyakan " apakah memiliki *evidence* A? " terlihat dari pohon keputusan apapun jawaban pengguna "ya" atau "tidak maka sistem akan menanyakan *evidence* B. Hal ini dapat diatasi dengan mengubah urutan pada tabel keputusan.

Table 2.2 Tabel Keputusan Alternatif

Hipotesa / Evidence	Hipotesa 1	Hipotesa 1	Hipotesa 1	Hipotesa 1
Evicence A	Ya	Ya	Ya	Tidak
Evicence D	Tidak	Tidak	Tidak	Ya
Evicence B	Ya	Tidak	Ya	Ya
Evicence C	Ya	Tidak	Tidak	Ya
Evicence E	Tidak	Ya	Ya	Tidak

Sumber: Hartati dan Iswanti (2008:34)



Keterangan:
A = evidence A, H1 = evidence 1, y = ya
B = evidence B, H2 = evidence 2, t = tidak
C = evidence C, H3 = evidence 3, ** = tidak menghasilkan hipotesa tertentu
D = evidence D, H4 = evidence 4

Gambar 2.10. Pohon Keputusan Alternatif
Sumber: Hartati dan Iswanti (2008:35)

Terlihat pada pada Gambar 2.10. Pohon Keputusan Alternatif, masing-masing node yang mewakili *evidence* tertentu untuk kondisi “y” atau “t” sudah tidak mengarah *evidence* yang sama. Dalam sesi konsultasi hal yang mengandung arti jawaban pengguna yang berbeda, akan mengarah pada pertanyaan yang berbeda pula.

2.1.3 Defenisi *Web*

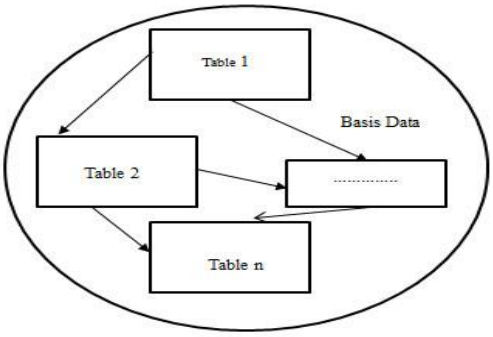
Menurut e-Media Solusindo (2008: 1) Dunia *Web* disusun oleh berbagai elemen, elemen paling kecil dari dunia *web* adalah halaman *web*. Halaman *web* adalah sebuah halaman yang dapat dibuka di *browser* dan berguna menampilkan informasi. Sebuah halaman *web* biasanya ditulis dalam format *HTML* atau *XHTML*. Yang membuat halaman *web* menarik adalah karena adanya *link* yang membuat seorang peselancar halaman *web* berkunjung. Sebuah halaman *web* diletakkan di sebuah lokasi yang disebut *webserver*. Lokasi *web server* bisa ada dikomputer yang sama dengan pengakses (sering juga disebut sebagai komputer *local*) atau komputer lain, *web server* bisa ada di jaringan *local* (intranet) ataupun di jaringan *internet global*.

Menurut e-Media Solusindo (2008:4) dengan adanya *web*, manusia di zaman sekarang dapat *mempublish* informasi yang dapat dinikmati oleh banyak orang dengan mudah, praktis, tidak ribet, skala internasional, dan murah. Tak heran, *website* kini sudah menjadi daya tarik tersendiri bagi *internet*, selain layanan *internet* lainnya, seperti *chat*, komunikasi data, *VoIP*, dan berbagai teknologi baru yang terus bermunculan.

Menurut Husda (2012:163) *Worl Wide Web (www)* atau dikenal dengan *web* atau situs adalah sistem dimana informasi dalam bentuk teks, suara dan gambar dan lain-lain yang disimpan di *server-server* yang terdapat diseluruh dunia. *Documen web* dibuat dengan menggunakan format *HTML (Hypertext Mark-up Language). Web Browser* atau Situs *Web* adalah setiap komputer atau tempat *(Space)* dalam sebuah komputer yang terhubung dengan *internet* dan menjalankan fungsi dan proses sebagai *server web* yang berisi dokumen-dokumen dalam format *HTML*. Sebuah *website* memiliki *URL* (alamat *web*) atau *domain name* (nama domain) yang biasanya berakhiran .com .net .org dan lain-lain; contoh bismillah.co.nr.

2.1.4 Defenisi Database

Menurut A.S dan Shalahuddin (2011: 44) basis data adalah sistem *terkomputerisasi* yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat *diakses* dengan mudah dan cepat.



Gambar 2.11. Ilustrasi Basis Data
Sumber: Rosa A.S dan Shalahuddin (2011: 44)

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa file teks ataupun *Database Management System (DBMS).*

Kebutuhan basis data dalam sistem informasi meliputi:

1. Memasukkan, menyimpan, dan mengambil data.
2. Membuat laporan berdasarkan data yang telah disimpan.

Menurut Husda (2012:151) istilah basis data *(database)* mengacu pada koleksi dari data-data yang saling berhubugan, sebagai contoh sehari-hari buku alamat dan nomor telepon bisa disebut basis data, dimana terdapat informasi mengenai nama, alamat, serta nomor telpon seseorang. Sebuah data pada basis data mempunyai beberapa tingkatan diantaranya:

1. *Characters* (Karakter)

   Karakter merupakan bagian data yang paling kecil, dapat berupa angka, huruf atau karakter khusus yang membentuk suatu item data atau *field.* Contohnya: 1,2,3,A,B,c,d,=,>,# dan sebagainya.

2. *Field* (Data Item)

   Field merupakan *representasi* suatu *atribut* dari *record* yang sejenis yang menunjukan suatu item dari data. Contohnya: Nama, Alamat, Tanggal dan sebagainya.

3. *Record*

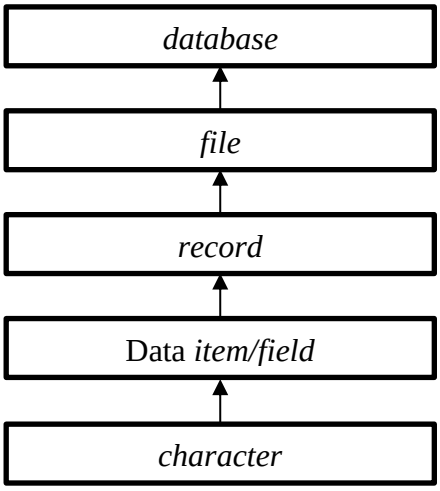
   Record adalah sekumpulan *elemen* yang saling berkaitan yang menginformasikan tentang suatu *entitas* secara lengkap. Satu *record* mewakili satu data atau informasi. Contohnya: NPM, Nama, Alamat, Tanggal_Lahir.

4. *File*

   File merupakan kumpulan dari *record-record* dalam basis data yang menggambarkan satu kesatuan data yang sejenis. Contohnya: *field* mata_kuliah yang berisi data tentang semua mata kuliah yang ada.

5. *Database* (Basis Data)

   Merupakan kumpulan dari beberapa file atau tabel yang saling berhubungan sehingga membentuk satu basis data. Cotohnya: Basis Data Perpustakaan, dan sebagainya.



Gambar 2.12. Tingkatan Basis Data
Sumber: Nur Elfi Husda (2012: 153)

2.1.5 Pengujian Perangkat Lunak

Menurut A.S dan Shalahuddin (2011: 209) sebuah perangkat lunak perlu dijaga kualitasnya di mana kualitasnya bergantung pada kepuasan pelanggan *(customer),* kualitas perangkat lunak perlu dijaga untuk keperluan sebagai berikut:

1. Agar dapat *"survive"* bertahan hidup di dunia bisnis perangkat lunak.
2. Dapat bersaing dengan perangkat lunak yang lain.

3. Penting untuk pemasaran global *(global marketing)*
4. Mengefektifkan biaya agar tidak banyak membuang perangkat lunak karena kegagalam pemasaran atau kegagalan produksi.
5. Mempertahankan pelanggan *(customer)* dan meningkatkan keuntungan.

Sering perangkat lunak mengandung kesalahan *(error)* pada proses-proses tertentu pada saat perangkat lunak sudah berada ditangan *user*. Kesalahan-kesalahan *(error)* pada perangkat lunak sering disebut dengan *"bug"*. Untuk menghindari banyaknya *bug* maka diperlukan adanya pengujian perangkat lunak sebelum perangkat lunak diberikan kepelanggan atau selama perangkat lunak masih terus dikembangkan. Adanya *bug* adalah suatu yang biasa, bahkan disebuah perangkat lunak yang sudah besar dan terkenal pun biasanya masih ada *bug*, sehingga tidak perlu merasa tersinggung atau bersedih jika ditemukan *bug* pada perangkat lunak yang dikembangkan. Yang bisa dilakukan pengembang perangkat lunak adalah *meminimalisasi bug* dengan melakukan pengujian. Kelakukan perangkat lunak yang tidak sesuai dengan spesifikasi yang dibutuhkan bisa dianggap *bug* (Rosa dan Shalahuddin, 2011: 209)

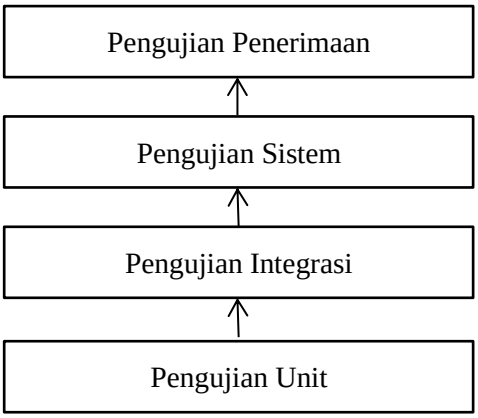
Menurut Rosa dan Shalahuddin (2011: 210) Pengujian adalah satu *set* aktifitas yang direncakan dan sistematis untuk menguji atau mengeveluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian. Secara umum pola pengujian pada perangkat lunak adalah sebagai berikut:

1. Pengujian dimulai dari *level* komponen hingga *integrasi* antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula tergantung pada pengujian pada bagaimana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak *(independent test group (ITG)).*
4. Pengujian dan penirkutan *(debugging)* merupakan aktifitas yang berbeda, tapi penirkutan *(debugging)* harus diakomodasi pada berbagai strategi pengujian. Pengujian lebih fokus pada mencari kesalahan *(error)* baik dari sudut pandang orang secara umum atau sudut pandang tanpa harus menemukan lokasi kesalahan pada kode program.

Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikatakan dengan verifikasi *(verification)* dan validasi *(validation)* (V&V). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak *mengimplementasikan* dengan benar sebuah fungsi yang spesifik. *Validasi* mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan *(customer).* Dapat juga dikatakan sebagai berikut:

> ***Verifikasi:*** "Apakah produk dibangun dengan benar?" (lebih kearah apakah proses pengembangan produk sudah benar dan telah berhasil mengimplementasikan fungsi yang benar).

Validasi: "Apakah sudah membangun produk yang benar?" (lebih kearah hasil produk apakah sudah sesuai dengan yang diinginkan).



Gambar 2.13. Pengujian Perangkat Lunak
Sumber: Rosa A.S dan Shalahuddin (2011: 212)

Pengujian diawali dari pengujian unit. Unit disini bisa berupa kumpulan fungsi atau prosedur yang memiliki keterkaitan pada pemrograman terstruktur (misalnya saja unit untuk menuliskan atau membaca data di basis data). Unit juga dapat berupa modul atau dikenal juga sebagai *package.* Seteleah unit selesai diuji maka dilakukan pengujian *integrasi.*

Pengujian *integrasi* sebaiknya dilakukan secara bertahap, tidak dilakukan secara satu tahap langsung diakhiri untuk menghindari kesulitan penelusuran jika terjadi kesalahan *(error).* Pengujian *integrasi* lebih pada pengujian penggabungan dari dua atau lebih unit pada perangkat lunak. Setelah pengujian *integrasi* maka dilakukan pengujian sistem dimana unit-unit proses yang sudah di *integrasi* diuji dengan antarmuka yang sudah dibuat sehingga pengujian ini dimaksudkan untuk menguji sistem perangkat lunak secara keseluruhan dan diuji sacara satu sistem (tidak terpisah-pisah lagi).

Setelah pengujian sistem dilakuan maka dapat dilakukan pengujian penerimaan perangkat lunak oleh pelanggan *(customer)* atau *user* (pemakai perangkat lunak). Pengujian penerimaan dilakukan untuk mengetahui kepuasan pelanggan atau *user* terhadap perangkat lunak yang sudah dibuat.

Pengujian untuk *validasi* memiliki beberapa pendekatan sebagai berikut:

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

   Yaitu pengujian perangkat lunak dari segi spesifikasi *fungsional* tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

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

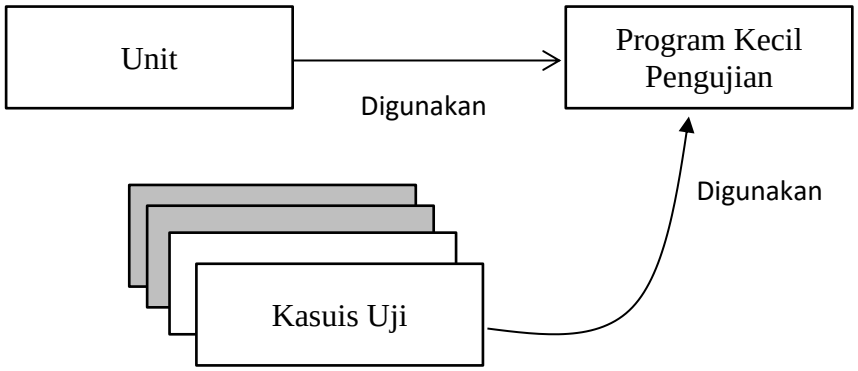
   Yaitu menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih dilakukan dengan memeriksa lojik dari kode program. Contoh pengujian kotak putih misalnya menguji alur (dengan menelusuri) pengulangan *(looping)* pada logika pemrograman.

2.1.5.1 Pengujian *Unit*

Menurut Rosa dan Shalahuddin (2011: 215) Pengujian unit fokus pada usaha verifikasi pada unit yang terkecil pada desain perangkat lunak (komponen atau model perangkat lunak). Setiap unit perangkat lunak di uji agar dapat diperiksa apakah aliran masukan *(input)* dan keluaran *(output)* dari unit sudah

sesuai dengan yang diinginkan. Pengujian unit biasanya dilakukan saat kode program dibuat. Karena dalam sebuah perangkat lunak banyak memiliki unit-unit kecil maka untuk menguji unit-unit kecil ini biasanya dibuat program kecil *(main program)* untuk menguji unit-unit perangkat lunak.

Unit disini secara fisik dapat berupa prosedur atau fungsi, sekumpulan prosedur atau fungsi yang ada dalam satu berkas *(file)* jika dalam program terstruktur, atau kelas, bisa juga kumpulan kelas dalam satu *package* dalam program berorientasi objek.



Gambar 2.14. Ilustrasi Pengujian Unit
Sumbar: Rosa A.S dan Shalahuddin (2011: 216)

2.1.5.2 Pengujian *Integrasi*

Menurut Rosa dan Shalahuddin (2011: 216) Pengujian *integrasi* adalah sebuah teknik yang sistematis untuk mengkonstruksikan struktur program seiring dengan menggabungkan fungsi program dengan antarmukanya. Pengujian *integrasi* bertujuan untuk mempergunakan komponen unit program yang sudah diuji dan membangun struktur seperti yang telah didesain sebelumnya.

Ada sebuah kecendrungan pada pengujian *integrasi* dimana integrasi tidak dilakukan secara bertahap, tetap langsung dilakukan pada akhir pengabungan perangkat lunak *("big bang").* Dengan menggunakan pendekatan *"big bang"* sebuah sistem diuji secara kesatuan hingga sering ketika kesalahan *(error)* akan menemui kesulitan untuk menemukan letak kesalahan *(error)* yang terjadi.

Pengujian *integrasi* memiliki beberapa tipe strategi pengujian seperti:

1. Pengujian *integrasi* dari atas ke bawah *(top-down integratiom)*
2. Pengujian *integrasi* dari bawah ke atas *(botton-up integration)*
3. Pengujian *integrasi* regresi *(regression integration)*
4. Pengujian *integrasi* "asap" *(smoke integration)*
5. Pengujian *integrasi* "roti isi" *(sandwich integration)* yaitu mengkombinasikan lebih dari satu strategi pengujian integrasi (misalnya pengujian dari atas ke bawah dan pengujian dari bawah ke atas.

2.2. Anak Usia Dini

Menurut Habibi (2015: 13) anak usia dini adalah anak dengan dengan usia 0-6 tahun. Beberapa orang menyebut *fase* atau masa ini sebagai *golden age*, karena masa ini sangat menentukan seperti apa mereka kelak jika dewasa, baik dari segi fisik, mental maupun kecerdasan. Tentu saja ada banyak faktor yang sangat mempengaruhi dalam perjalanan mereka menuju dewasa, tetapi apa yang mereka dapat dan apa yang diajarkan pada mereka pada usia dini akan tetap membekas dan bahkan memiliki pengaruh yang dominan dalam menentukan setiap pilihan dan langkah hidup mereka.

2.2.1 Pengertian Pendidikan Anak Usia Dini

Undang-undang RI No, 20 Tahun 2003 tentang sistem Pendidikan Nasional Bab 1, Pasal 1, Butir 14 menyatakan bahwa "Pendidikan Anak Usia Dini adalah suatu upaya pembinaan yang ditujukan kepada anak sejak lahir sampai dengan usia 6 tahun yang dilakukan dengan memberikan rangsangan pendidikan untuk membantu pertumbuhan dan perkembangan jasmani dan rohani agar anak memiliki kesiapan dalam memasuki pendidikan lanjut".

Menurut pasal 28 undang-Undang Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional, bentuk satuan pendidikan anak usia dini dikelompokkan menjadi 3, yaitu:

1. Jalur pendidikan *formal*

   Terdiri atas Taman Kanak-Kanak dan Raudhatul Atfal. Taman Kanak-Kanak dan Raudhatul Atfal dapat diikuti anak usia lima tahun keatas.

2. Jalur Pendidikan *Non Formal*

   Terdiri atas penitipan anak, kelompok bermain dan satuan PAUD sejenis. Kelompok bermain dapat diikuti usia dua tahun ke atas, sedangkan penitipan anak dan satuan PAUD sejenis dapat diikuti sejak lahir, atau usia tiga bulan.

3. Jalur Pendidikan *Informal*

   Terdiri atas pendidikan yang diselenggarakan di keluarga dan di lingkungan. Ini menunjukan bahwa pemerintah melindungi hak anak untuk mendapatkan layanan pendidikan, meskipun tidak masuk lembaga pendidikan anak usia dini, baik formal maupun nonformal.

Menurut Tim Pengembangan Ilmu Pendidikan FIP-UPI (2007:95) pendidikan dapat dipandang sebagai suatu proses perberdayaan dan pembudayaan individu agar dia mampu memenuhi kebutuhan perkembangannya dan sekaligus memenuhi tuntutan sosial, kultural, dan religius dalam lingkungan kehidupannya. Pengertian pendidikan ini mengimplikasikan bahwa upaya apa pun yang dilakukan dalam konteks pendidikan seyogyanya terfokus pada fasilitasi proses perkembangan individu sesuai dengan nilai agama dan kehidupan yang dianut.

Sejalan dengan pandangan diatas, Pendidikan Anak Usia Dini (PAUD) diartikan sebagai segenap upaya pendidikan (orang tua, guru, dan orang dewasa lainnya) dalam memfasilitasi perkembangan dan belajar anak sejak lahir sampai dengan usia 6 tahun malalui penyediaan berbagai pengalaman dan rangsangan yang bersifat mengambangkan, terpadu, dan menyeluruh sehingga anak dapat bertumbuh-kembang secara sehat dan optimal sesuai dengan nilai dan norma kehidupan yang dianut.

Dalam pengertian PAUD tersebut terdapat beberapa gagasan pokok yang perlu dijelaskan lebih lanjut. *Pertama,* aktivitas pendidikan tidak dibatasi secara sempit pada kegiatan belajar-mengajar di kelas, melainkan mencakup segenap aktivitas yang diarahkan untuk mendukung proses perkembangan dan belajar belajar anak secara menyeluruh. *Kedua,* yang berperan sebagai pendidik tidak terbatas pada orang tua ata guru, melainkan bisa pula melibatkan orang dewasa lainnya yang ikut terlibat dalam proses pendidikan anak. *Ketiga,* sesuai dengan istilah yang digunakan usia dini, masa pendidikan dibatasi pada jenjang usia sejak lahir sampai dengan usia 6 tahun. *Terakhir*, sasaran akhir dari PAUD adalah

tercapainya perkembangan anak yang optimal sesuai dengan nilai dan norma yang dianut melalui penyediaan berbagai rangsangan serta lingkungan dan pengalaman belajar yang relevan dan berpengaruh positif terhadap pertumbuhan dan perkembangan anak.

2.2.2 Pengertian Pengembangan Potensi

Menurut Wibhowo dan Sanjaya (2011: 95) Tuhan memberikan potensi kepada tiap manusia, sebanyak sepertiga yang tidak dapat diutak-atik oleh manusia. Kita berjumpa dengan anak yang sejak lahir memang menunjukkan kepandaiannya bernyanyi, baru 2 tahun sudah bisa menirukan nada dan cepat hafal syair lagu yang belum lama ia dengar, itu potensi yang sebesar sepertiga tadi. Tetapi coba saja anak itu dibiarkan berkembang sendiri tanpa stimulasi apapun. Hasilnya kemampuan menyanyi hanya sepertiga itu saja, karena stimulasi dari orang lain maupun lingkungan, ada sebesar dua pertiga.

Solehudin (2000) dalam Tim Pengembangan Ilmu Pendidikan FIP-UPI (2007: 96) dalam konteks perkembangan anak, PAUD memiliki lima fungsi dasar, yakni: (1) Pengembangan potensi (2) Penanaman dasar-dasar aqidah keimanan (3) Pembentukan dan pembiasaan perilaku yang diharapkan (4) Pengembangan pengetahuan dan keterampilan dasar yang diperlukan (5) Pengembangan motivasi dan sikap belajar yang positif.

Pada dasarnya setiap bayi yang dilahirkan kedunia dilengkapi sejumlah potensi yang diperlukan untuk kehidupannya. Ia memiliki potensi untuk beragama, berpikir, berkreasi, merasa, berkomunikasi dengan orang lain, dan

potensi-potensi lainnya. Upaya pengembangan potensi anak perlu dilakukan sejak usia dini sebab pada masa itulah terjadinya masa-masa emas dari perkembangan berbagai potensi tersebut.

2.2.3 Kecerdasan Pada Anak Usia Dini

Menurut Gordon dan Cooper (2013: 6) awalnya, manusia diyakini memiliki satu macam kecerdasan yang merupakan bawaan sejak lahir dan tidak akan berubah seumur hidup, yaitu *Intelligence Quotiente* atau IQ. Pandangan tersebut bertahan hingga tahun 1983. Howard Gardner, profesor pendidikan dan Universitas Havard, menerbitkan buku fenomenal *Frames of Mind: The Teory of Multiple Intelligence.* Melalui bukunya, Gardner memperkenalkan defenisi baru tentang kecerdasan, mendobrak paradigma lama yang mengerucutkan kepintaran dalam sebuah skor IQ. Dalam penelitiannya, Gardner menyatakan bahwa manusia bukan memiliki satu kecerdasan, melainkan tujuh jenis kecerdasan yang semuanya dapat berubah dan dikembangkan. Gardner juga menemukan kecerdasan baru diluar yang ia temukan sebelumnya, misalnya *natural intelligence* atau kecerdasan natural.

Teori *Multiple Intelligence* adalah sebuah model kecerdasan yang positif dan *inklusif,* mencakup semua bakat yang ada dalam diri kita, bukan hanya bakat "akademis" semata. Kebenaran teori ini juga semakin diperkuat dengan banyaknya orang sukses yang memiliki kecerdasan di luar "defenisi tradisional". Kecerdasan dapat ditingkatkan dengan pola pembelajaran dan pengembangan sesuai kelebihan setiap individu. Setiap jenis kecerdasan saling berinteraksi

dengan cara yang kompleks, tidak ada kecerdasan yang dapat berdiri sendiri. Seorang pemain sepak bola profesional, misalnya, memerlukan kecerdasan *spasial* dan kecerdasan *kinetik* untuk mengoordinasikan dan berinteraksi dengan pemain lain ketika akan mengoper bola. Dengan motivasi dan pembelajaran yang memadai, anak bisa memiliki kopetensi yang cukup untuk tiap kecerdasan. Teori *multiple intelligence* meyakini tidak ada seorang anak pun yang terjebak dalam kecerdasan yang ia bawa sejak lahir. Ketika anda sebagai orang tua, memahami keunikan kecerdasan anak, anda dapat mengarahkan untuk menggunakan kecerdasan secara maksimal. Dengan mengenali kelemahannya, anda pun membantu anak mengembangkan diri.

Menurut Gordon dan Cooper (2013:6) ada 2 tipe yang umum digunakan untuk menilai skor IQ:

1. Logika Matematika, kemapuan menggunakan angka dengan efektif, termasuk menyimpulkan dan berpikir menggunakan logika. Kemampuan tersebut dibagi menjadi Kecerdasan Angka dan Kecerdasan Berpikir
2. Linguistik atau kecerdasan kata, kemampuan menggunaka kata-kata dengan efektif saat berbicara atau menulis.

Berikutnya ada 4 kecerdasan spesial:

1. Musikal atau Kecerdasan Musik, kemampuan untuk mencerap, mengubah, dan mengekspresikan bentuk-bentuk musikal.
2. *Spasial* atau kecerdasan gambar, kemampuan untuk mencerap dan memanipulasi objek atau gambar.

3. *Kinestetik* atau kecerdasan tubuh, keahlian fisik seseorang dalam menggunakan tubuhnya untuk mengekspresikan atau memproduksi sesuatu.
4. *Naturalis* atau kecerdasan Natural, kemampuan untuk mengenali dan mengklasifikasikan tanaman, binatang, dan fenomena alam.

Ada pula 2 kecerdasan tambahan yang belum lazim dikenal dalam *multiple intelligence*:

1. *Intrapersonal* atau Kecerdasan personal, kemampuan untuk memahami dan mengelola perasaan sendiri
2. *Interpersonal* atau kecerdasan Sosial. Kemampuan untuk mengerti dan mengelola perasaan orang lain.

Menurut Wibhowo dan Sanjaya (2011: 95) seorang Professor Pendidikan bernama Howard Gardner melakukan penelitian tentang perkembangan kapasitas *kognitif* manusia. Dia mengejutkan dunia pendidikan saat itu yang masih menganut paham bahwa kecerdasan manusia itu dapat diukur dan bersifat tunggal. Gardner mengatakan bahwa ada sedikitnya 8 kecerdasan manusia. Ia juga tidak memandang kecerdasan sebagai skor semata (IQ) tetapi ia menjelaskan bahwa:

1. Kecerdasan adalah kemampuan untu menyelesaikan masalah yang terjadi dalam kehidupan sehari-hari.
2. Kecerdasan menghasilkan persoalan-persoalan baru untuk diselesaikan.
3. Kecerdasan juga menciptakan sesuatu dan menawarkan jasa yang akan menimbulkan penghargaan dalam budaya seseorang.

Telah disebutkan bahwa minimal manusia memiliki delapan kecerdasan. Dikatakan ‘minimal’, karena bisa saja suatu saat akan muncul tokoh-tokoh pendidikan yang bisa menemukan kecerdasan dalam hal lain.

1. Kecerdasan *logis matematis*

Kecerdasan logis matematis adalah kemampuan untuk menguasai pola-pola katagori dan dapat memanipulasi objek atau symbol dengan cara yang sistematis dan teratur. Ia akan terlihat suka pada hal-hal yang terkait dengan angka.

2. Kecerdasan *Verbal Linguistik*

Senang menulis dan berdebat. Suka sekali menulis tentang apa pun yang dialami sepanjang hari. Senang mendengarkan cerita, sangat hafal pada nama tempat yang ia pernah kunjungi.

3. Kecerdasan *Intrapersonal*

Kecerdasan *intrapersonal* memang tidak ada kaitannya langsung dengan presatasi akademis disekolah, namun ini merupakan modal untuk hidup. Anak yang memiliki kecerdasan ini akan tidak mudah stres karena mampu mengkomunikasikan perasaanya. Senang menjadi diri sendiri, mampu membuat rencana untuk mencapai cita-citanya dan percaya diri. Intinya, anak ini adalah “tahu apa yang dia mau”.

4. Kecerdasan *interpersonal*

Anak dengan kecerdasan interpersonal yang baik akan nampak sangat menyenangkan. Karena ia bisa menunjukkan rasa empati kepada orang lain. Ia bisa berhubungan baik dengan teman-teman sebayanya dan orang dewasa disekitarnya. Karean ia mampu mengkomunikasikan perasaan/pikiran kepada

orang lain, maka ia memiliki kemampuan menjadi seorang pemimpin. Tidak hanya itu, ia bisa mengatur, membujuk bahkan memanupulasi orang lain.

5. Kecerdasan *Motorik* Kasar dan Halus

Anak dengan kecerdasan ini, sangat menikmati jika diajak bermain "menjawab pertanyaan dengan bahasa tubuh" atau didorong untuk mengikuti klub tari/olah raga.

6. Kecerdasan *Visual-Spasial*

Dengan kecerdasan visual-spasial, seseorang mampu memanipulasi sebuah bentuk atau objek dan membuatnya tampil menarik.

7. Kecerdasan *Musikal*

Dikatakan memiliki kecerdasan musikal, asal mereka suka mendengarkan lagu, menyanyi dan sangat sensitif pada pola-pola suara.

8. Kecerdasan *Naturalis*

Anak dengan kecerdasan naturalis sangat antusias jika diajak ketempat-tempat seperti *Tree-Top,* taman safari atau panti. Mencintai binatang atau tanaman.

2.3. Software Pendukung

Software pendukung merupakan beberapa perangkat lunak *(software)* yang penulis gunakan untuk mendukung pembuatan sistem pakar pada Tugas akhir/skripsi ini. Adapun perangkat lunak yang penulis gunakan antara lain: *StarUML, XAMPP, phpAdmin, PHP, HTML, CSS, MySQL, Notepad++.*

2.3.1 StarUML

Menurut Triandini dan Suardika (2012: 1) StartUML adalah *platform* pemodelan yang mendukung UML (*Unified Modeling Language*). StarUML™ yang berbasiskan UML versi 1.4, menyediakan sebelas jenis diagram 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 kustomisasi lingkungan kerja pengguan dan memiliki *ekstensibilitas* tinggi pada fungsionalitasnya. StarUML™ mengklaim dirinya sebagai salah satu alat pemodelan perangkat lunak terkemuka yang menjamin dapat memaksimalkan produktivitas dan kualitas proyek perangkat lunak.

Gambar 2.15. Logo StarUML
Sumber: https://upload.wikimedia.org/wikipedia/en/7/71/StarUML_logo.png

Fitur-fitur utama dalam StarUML versi 5.0 adalah:

1. Dukungan terhadap UML 2.0:
    a. *Use Case diagram.*
    b. *Class Diagram.*
    c. *Sequence Diagram.*
    d. *Collaboration Diagram*

e. Statechart Diagram

f. Activity Diagram.

g. Component Diagram,

h. Deployment diagram.

i. Composite Structure Diagram (UML2.0)

2. Dukungan terhadap beberapa bahasa pemrograman:
   a. Java profile, Code Generator, dan Reverse Engineering,
   b. C++ profile, Code Generator, dan Reverse Engineering,
   c. C# profile, Code Generator, dan Reverse Engineering,
   d. Microsoft Office Document Generator.
   e. Microsoft Word document template and generation.
   f. Microsoft excel document template and generation.
   g. Microsoft powerpoint document template and generation.
   h. Customizable code Generation.
   i. Mendukung teknologi MDA (UML profile dan Diagram yang dapat dikustomisasi.
   j. Diagram yang dapat diperluas (tentukan sendiri jenis diagram anda diluar diagram UML)
   k. Extensibility.
   l. Kompatible yang tinggi.
   m. Editing.
   n. User interface

2.3.2 *Unified Modeling Language* (UML)

Menurut AS dan Shalahuddin (2011: 113) *Unified Modeling Language* adalah salah satu standard bahasa yang banyak digunakan di dunia industri untuk mendefenisikan *requiretment*, membuat analisa dan desain sistem, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Pada diagram UML2.3 terdiri 13 macam diagram yang dikelompokkan dalam 3 katagori, yaitu:

1. *Struktur Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan, yaitu:
   a. *Class Diagram*
   b. *Objek Diagram*
   c. *Componen Diagram*
   d. *Componen structure diagram*
   e. *Composite structure diagram*
   f. *Package diagram*
   g. *Deploymen diagram*
2. *Behavior Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem, yaitu:
   a. *Usecase Diagram*
   b. *Activity Diagram*
   c. *State Machine Diagram*

3. *Instraction Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem. Yaitu:
   a. *Sequence Diagram*
   b. *Communication Diagram*
   c. *Timming Diagram*
   d. *Interacton Overview Diagram*

2.3.2.1 *Class Diagram*

Menurut AS dan Shalahuddin (2011: 122) Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefenisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut *atribut* dan *metode* atau *operasi*. *Atribut* merupakan variable-variabel yang dimiliki oleh suatu kelas dan *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 ke pemakai
3. Kelas yang diambil dari pendefinisian *use case*

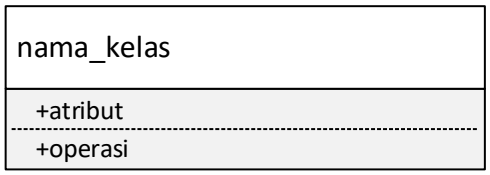
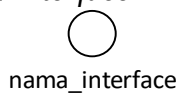
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4. Kelas yang diambil dari pendefinisian data

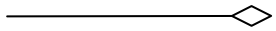
   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas

Tabel 2.3 Simbol-Simbol Pada *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / *interface* 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi/*association* 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah / *direct association* 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, *asosiasi* biasanya juga disertai dengan *multiplicity*.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan kelas.

Tabel 2.2 Lanjutan

Agresi / *aggregation* 	Relasi antar kelas dengan makna semua-bagian (*whole-part*).

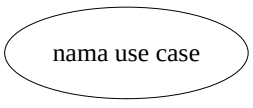
Sumber: Rosa dan Shalahudin (2011: 123)

2.3.2.2 *Use Case Diagram*

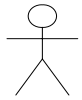
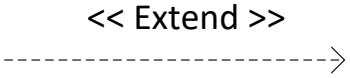
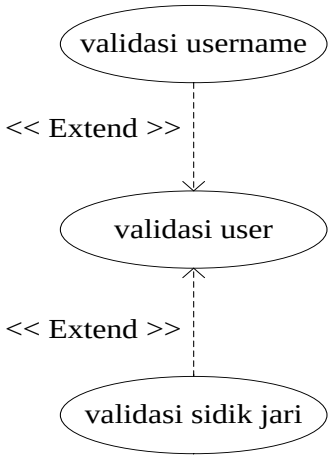
Menurut Rosa dan Shalahuddin (2011: 130) *Use case* atau *diagram use case* merupakan pemodelan untuk kelakukan *(behavior)* sistem informasi yang akan diperbuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada sebuah sistem informasi dan siapa saja yang berhak menggunakn fungsi-fungsi itu. Ada 2 hal utama pada *use case* yaitu:

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tepi aktor belum tentu merupakan orang.
2. *Use Case* merupakan fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

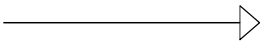
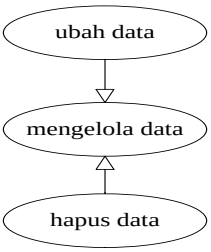
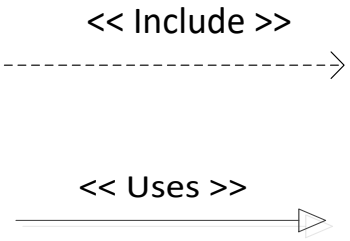
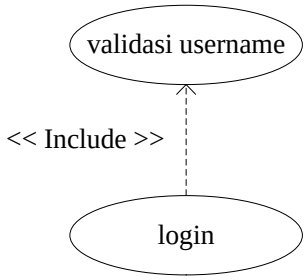
Tabel 2.4 Simbol-Simbol Pada *Use Case Diagram*

Simbol	Deskripsi
Use case 	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 *use case*.

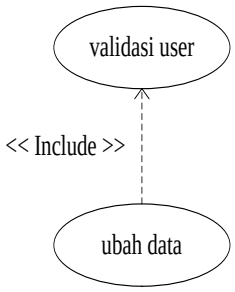
Tabel 2.4 Lanjutan

Aktor / a*ctor*  Nama aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal *frase* nama aktor.
Asosiasi / *association* _____	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor
Ekstensi / *extend* 	Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu, biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misalnya:  arah panah mengarah pada *use case* yang ditambahkan, biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.

Tabel 2.4 Lanjutan

Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum - khusus) antara 2 buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:  arah panah mengarah pada *use case* yang menjadi generalisasainya (umum).
Menggunakan / *include* / *uses* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat diljalankan *use case* ini. Ada dua sudut pandang yang cukup besar mengenai *include* di *use case*: 1. *Include* berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut:  2. *Include* berarti *use case* yang ditambahkan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan, missal pada kasus berikut:

Tabel 2.4 Lanjutan

	 Kedua sudut pandang diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan kebutuhan

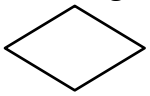
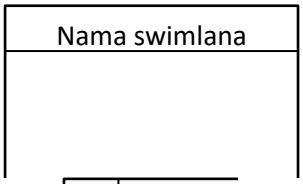
Sumber: Rosa dan Shalahudin (2011: 131)

2.3.2.3 *Activity Diagram*

Menurut AS dan Shalahuddin (2011: 134) Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram activitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan sistem. Diagram aktivitas juga banyak digunakan untuk mendefenisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang akan digambarkan merupakan proses bisnis sistem yang didefenisikan.
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 didefenisikan kasus ujinya.

Tabel 2.5 Simbol-Simbol Pada *Activity Diagram*

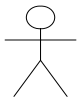
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan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

Sumber: AS dan Shalahudin (2011: 134)

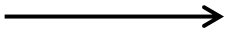
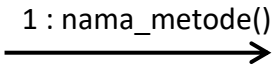
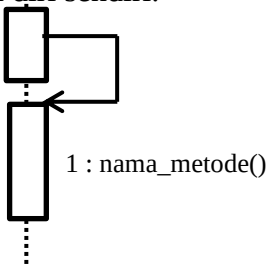
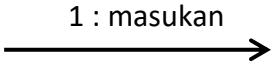
2.3.2.4 *Sequence Diagram*

Menurut AS dan Shalahuddin (2011: 134) Diagram sekuen menggambarkan kelakuan abjek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek. Oleh karena itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang *diinstansiasi* menjadi objek itu.

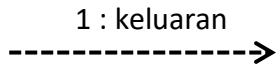
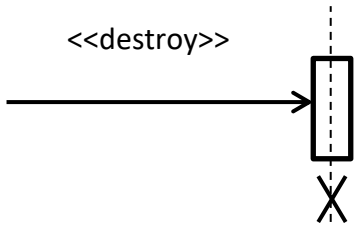
Tabel 2.6 Simbol-Simbol Pada *Activity Diagram*

Simbol	Deskripsi
aktor  Nama aktor Atau Nama aktor Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat duluar sistem informasi yang akan dibuat itu sendiri, jadi walupun symbol dari aktor adalah gambar orang, tetap belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Garis hidup/ *lifeline* 	Menyatakan kehidupan suat objek
objek Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan

Tabel 2.6 lanjutan

Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan. 6
Pesan tipe *create* <<create>> 	Menyatakan suatu objek membuat Objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau diri sendiri.  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe send 	Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

Tabel 2.6 lanjutan

Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian keobjek tertentu, arah penah mengarah pada objek yang menerima kembalian.
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destrot.

Sumber: AS dan Shalahudin (2011: 138)

2.3.3 *Notepad++*

Gambar 2.16. Logo *Notepad++*
Sumber: https://en.wikimedia.org/wiki/File:Notepad%2B%2B_Logo.png

Menurut Kurniawan (2010: 108) *Notepad++* adalah sebuah program *Freeware* yang berfungsi sebagai *editor* pengganti *Notepad* bawaan *windows. Notepad ++* ditulis dalam *C++* yang menjamin kecepatan eksekusi lebih tinggi

dan ukuran program yang lebi kecil. *Editor* ini biasa digunakan untuk *mengedit* halaman *web* berformat html standar mengganti *dreamweaver*.

Berikut ini beberapa keunggulan *Notepad++* dibandingkan dengan *notepad* standar:

1. Tampilan lebih menarik dan menyegarkan
2. Lebih *user friendly* dan mudah penggunaanya
3. Mendukung multi tab.
4. Mendukung banyak bahasa pemrograman
5. Dan masih banyak keunggulan lainnya, jadi silahkan dicoba sendiri.

2.3.4 Bahasa Pemrograman HTML

Gambar 2.17. Logo HTML
Sumber: https://en.wikimedia.org/wiki/File:HTML5_logo_and_wordmark.svg

Menurut Saputra (2012: 1) HTML merupakan singkatan dari *Hypertext Marup Language.* HTML bisa disebut bahasa laping dasar dan paling penting yang digunakan untuk menampilkan dan mengelola tampilan pada halaman *web*. HTML digunakan untuk menampilkan berbagai informasi dalam sebuah penjelajah *web internet* dan *formatting hypertext* sederhana yang ditulis dalam

berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Beberapa elemen wajib yang ada pada file HTML apabila kita ingin membangun suatu pondasi kerangka *website,* elemen tersebut diantaranya:

1. *Elemen HTML*

   Elemen html merupakan *tag* dasar apabila kita ingin memulai suatu dokumen html. Contohnya: <html> dan diskhiri dengan </html>.

2. *Elemen Head*

   Head merupakan *tag* berikutnya setelah elemen html, yang berfungsi untuk menuliskan keterangan tentang dokumen web yang akan ditampilkan. Jika secara runtun dapat dituliskan:

   ```
   <html>
   <head>
   </head>
   </html>
   ```

3. *Elemen Title*

   Elemen title merupakan suatu elemen yang harus dituliskan dalam elemen *head* yang digunakan untuk memberikan judul/informasi pada *caption web browser* tentang *topic*/tema dari suatu dokumen web.

   ```
   <html>
   <head>
         <title>Tulis Judul Disini</title>
   </head>
   </html>
   ```

4. *Elemen Body*

   Elemen body merupakan suatu bagian utama dalam dokumen *web*. Jika kita ingin menampilkan suatu teks atau informasi yang dikenal dengan sebutan konten, maka harus meletakkan teks tersebut pada *elemen body*.

```
<html>
<head>
        <title>Tulis Judul Disini</title>
</head>
        <body>
        Tulis Konten disini
        </body>
</html>
```

2.3.5 Bahasa Pemrograman PHP

Gambar 2.18. Logo *PHP*
Sumber: https://en.wikipedia.org/wiki/File:PHP-logo.svg

Menurut Zaki (2008:2) PHP adalah sebuah bahasa pemrograman *scripting* untuk membuat halaman *web* yang dinamis. Walaupun dikenal sebagai bahasa untuk membuat halaman *web*, tapi PHP sebenarnya juga dapat digunakan untuk membuat aplikasi *command line* dan juga GUI. Cara kerja PHP adalah dengan menyelipkannya di antara kode HTML *(Hypertext Markup Language)*. *Website* yang dibuat menggunakan PHP memerlukan *software* bernama *webserver* tempat pemrosesan kode PHP dilakukan. *Server web* yang memiliki *software* PHP Parser akan memproses input berupa kode PHP dan menghasilkan *output* berupa halaman *web*. PHP bersifat terbuka dan *multiplatform*, karenanya dapat dijalankan dibanyak merek *web server* (seperti apache dan IIS).

2.3.6 Bahasa Pemrograman CSS

Gambar 2.19. Logo *CSS*
Sumber: http://www.logotypes101.com/logo/css3

Menurut Saputra (2012: 27) CSS yang merupakan singkatan dari *Cascading Style Sheet*, merupakan bahasa pemrograman *web* yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* lebih rapi, terstruktur, dan seragam. Tujuan utama dari CSS adalah memisahkan konten utama dengan tampilan dokumen lainnya (html dan sejenisnya). Dengan adanya pemisahan konten ini, akses konten pada *web* akan meningkat.

CSS sampai saat ini terdapat 3 versi yang tiap versinya pasti ada peningkatan yang dilakukan, yaitu:

1. *CSS-1*, masih kuno, *CSS* hanya dikembangkan dan digunakan untuk *formatting document* html.
2. *CSS-2*, sudah menggunakan *font*, *table-layout*, dan berbagai media untuk printer.

3. *CSS-3,* mengembangkan versi sebelumnya. Peningkatan paling mencolok pada versi 3 ini adalah peningkatan fitur yang mengarah pada efek animasi.

2.3.7 MySQL *Database*

Gambar 2.20. Logo MySQL
Sumber: https://upload.wikimedia.org/wikipedia/en/6/62/MySQL.svg

Menurut Saputra (2012: 77) MySQL merupakan salah satu *database* kelas dunia yang sangat cocok bila dipadukan dengan bahasa pemrograman PHP. MySQL bekerja dengan menggunakan bahasa SQL *(Structure Query Language)* yang merupakan bahasa *standard* yang digunakan untuk memanipulasi *database*. Pada umumnya, perintah yang sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, ataupun index untuk menambah atau menghapus data.

Ada beberapa alasan yang menjadikan *database* MySQL sangat diminati oleh para programmer, diataranya:

1. Bersifat *open source*

2. Menggunakan bahasa SQL *(Structure Query Language)*, yang merupakan standar bahasa dalam pengolahan data.
3. Performance dan *reliable*, pemrosesan *database*-nya sangat cepat dan stabil.
4. Sangat mudah dipelajari *(easy of use)*
5. Memiliki dukungan *(group)* pengguna MySQL.
6. Lintas *platform*, dapat digunakan pada berbagai sistem operasi berbeda.
7. *Multiuser*, dimana MySQL dapat digunakan oleh banyak *user* dalam waktu yang bersamaan tanpa mengalami konflik.

2.3.8 *XAMPP*

Gambar 2.21. Logo *XAMPP*
Sumber: https://upload.wikimedia.org/wikipedia/commons/0/03/Xampp_logo.svg

Menurut e-Media Solusindo (2008:58) adanya *XAMPP* benar-benar memperingankan pekerjaan karena kesulitan yang sering dialami dalam implementasi *web server open source* tidak lagi dihadapi. Anda tinggal menginstal satu *installer* XAMPP, mengeksekusinya, dan server siap digunakan dalam waktu yang singkat. Walaupun demikian, *instalasi* menggunakan *XAMPP* memiliki kekurangan. Kekurangan yang paling patut diwaspadai adalah masalah

keamanan. Umumnya versi-versi *server* yang dimasukkan di *server* adalah yang tidak mutakhir, demi alasan stabilitas.

2.3.9 *PHP MyAdmin*

Gambar 2.22. Logo phpMyAdmin
Sumber: https://en.wikipedia.org/wiki/File:PhpMyAdmin_logo.svg

Menurut e-Media Solusindo (2008: 57) salah satu alat paling ciamik untuk mengelola *database* MySQL adalah *PHPMyAdmin*. Memanfaatkan *PHP MyAdmin*, proses pengaturan *previlege* dan manipulasi *database* MySQL lebih mudah dilakukan menggunakan antarmuka *web* oleh orang awam sekalipun.

2.4. Penelitian Terdahulu

Sebagai bahan pertimbangan dalam penelitian ini, maka penulis mencantumkan beberapa penelitian yang diambil dari beberapa jurnal ilmiah, yaitu:

1. Nama Penulis : M. Rosidi Zamroni, Choirul Anggun Cahyani dan Ahmadin Jalaluddin

Judul Jurnal : Sistem Pakar Pengembangan Anak Usia 0 - 12 Bulan Berbasis Web Dengan Metode Forward Chaining

Volume / ISSN : Vol.5 No.2 September 2013 / No.2085-0859

Kesimpulan : Sistem Pakar adalah salah satu cabang kecerdasan buatan yang mempelajari mengadopsi cara seorang pakar dan bernalar dalam menyelesaikan suatu permasalahan, dan membuat suatu keputusan kesimpulan dari perhitungan fakta yang ada. Perkembangan anak adalah dimana bertambahnya kemapuan dalam struktur dan fungsi yang lebih dalam pola yang teratur. Dari hasil penelitian proses perkembangan anak usia 0-12 bulan masih dilakukan secara manual. Sistem Pakar Perkembangan Anak Usia 0-12 Bulan yang berbasis web adalah penyelesaian dari masalah yang terjadi dalam proses perkembangan Anak usia 0-12 bulan. Maka dari itu sistem pakar yang dibuat ini diharapkan dapat mempermudah proses dalam menentukan perkembangan anak usia 0-12 bulan. Maka dari itu sistem pakar yang dibuat ini diharapkan dapat mempermudah proses dalam menentukan perkembangan anak usia 0-12 bulan.

2. Nama Penulis : Ahan Pramusti dan Krisnawati

Judul Jurnal : Membangun Aplikasi Sistem Pakar Psikologi klinis Pada Remaja Berbasis Android (Studi Kasus : Puskesmas Seyegan

Volume / ISSN : Vol.14 No.04 Desember 2013 / No.1411-3201

Kesimpulan : Remaja merupakan fase yang paling rentan dan sangat perlu diperhatikan satu demi satu tahapan perkembangan. Perubahan mood atau suasana hati pada remaja relatif masih labil. Dimana sang anak masih belum bisa menguasai dan mengelola emosi dalam dirinya sendiri. Apabila sang anak selalu bersikap seperti itu maka dampaknya akan sangat buruk bagi perkembangan sosial anak tersebut. Beberapa orang tua yang memiliki anak dengan gangguan mood maka yang dilakukan adalah membawanya ke psikolog. Dimana hal tersebut juga akan membutuhkan tidak hanya biaya tetapi juga waktu dan tenaga. Salah satu implementasi yang diterapkan sistem pakar dalam bidang psikologi, yaitu untuk menentukan jenis gangguan mood atau suasana hati pada remaja. dengan menggunakan metode Forward Channing. Dimana aplikasi ini dapat diakses melalui gadget yang berbasis android. Dengan begitu setiap orang dapat mengunduh aplikasi ini nanti di appstore dan dapat menginstalnya secara gratis.

3. Nama Penulis : Faresi Daeli

Judul Jurnal : Sistem Pakar Dalam Menentukan Tingkat Iq Anak Yang Mengalami Reterdasi Mental Dengan *Metode Certainty Factor* (Studi Kasus: Pendidikan Slb/B Karya Murni)”

Volume / ISSN : Vol IV No.3 Agustus 2013 / 2301-9425

Kesimpulan : Perkembangan teknologi yang sangat pesat seiring dengan kebutuhan manusia yang semakin banyak dapat memungkinkan untuk digunakan sacara luas di berbagai bidang seperti dalam dunia bisnis, kesahatan, pendidikan dan perkantoran. System pakar (expert system) adalah program berbasis pengetahuan yang menyediakan solusi-solusi dengan kualitas pakar untuk masalah-masalah dalam suatu domain yang spesifik. Implementasi sistem pakar banyak digunakan dalam bidang kesehatan karena sistem pakar dipandang sebagai cara penyimpanan pengetahuan pakar pada bidang tertentu sehingga dalam program computer keputusan dapat diberikan dalam melakukan penalaran secara cerdas. Pada proposal skripsi ini yang dibahas adalah penerapan metode certainty factor untuk menentukan anak yang mengalami reterdasi mental, menggunakan bantuan bahasa pemrograman Microsoft Visual Studio dot Net 2008 dengan database Microsoft Access, diharapkan dapat memberikan hasil dalam menentukan anak yang mengalami reterdasi mental dengan metode certainty factor.

4. Nama Penulis : Dewi Miranti Wijaya dan Wakyu Kusuma Raharja

Judul Jurnal : Implementasi Metode Forward Chaining Pada Sistem Pakar Penentuan Karakter Diri Berbasis Website menggunakan Framework Codeigniter

Volume / ISSN : Vol.6 Oktober 2015 / 1858-2559

Kesimpulan : Sistem pakar pada aplikasi ini digunakan untuk mengenali karakter diri. Terdapat empat karakter yang terdapat pada diri individu, yaitu dominan, intim, stabil, dan cermat. Karakter diri terbentuk dari dua faktor, yaitu faktor keturunan dan faktor lingkungan. Faktor keturunan akan membentuk karakter yang disebut dengan karakter natural yang dipengaruhi oleh orang tua biologis individu Sedangkan faktor lingkungan dipengaruhi oleh lingkungan seseorang tumbuh. Kedua hal tersebut menimbulkan dua karakter pada diri individu, sehingga seseorang dapat memiliki karakter yang berbeda ketika berada di rumah dan di lingkungan. Mengenal karakter diri dapat dimanfaatkan untuk memilih bidang pekerjaan yang cocok dengan kepribadian seseorang, dengan mengetahui kelebihan yang ada pada diri dan memperbaiki kekurangan. Dengan adanya aplikasi ini diharapkan seseorang dapat mengenali potensi dirinya dengan mudah dan menjadi alternatif bagi para psikolog. Aplikasi ini dibuat dengan metode runut maju

menggunakan Bahasa Pemrograman *Hypertext Preprocessor* (PHP) dan *framework CodeIgniter*.

5. Nama Penulis : Febi Nur Salisah, Leony Lidya dan Sarjon Defit

Judul Jurnal : Sistem Pakar Penentuan Bakat Anak Dengan Menggunakan Metode Forward Chaining

Volume / ISSN : Vol.1, No.1, Februari 2015 / 24608181

Kesimpulan : Saat ini masih banyak orang tua belum mengetahui bakat anak mereka. Sedikitnya jumlah pakar untuk berkonsultasi merupakan salah satu penyebab hal ini. Penelitian ini menggunakan sistem pakar sistem pakar untuk mengatasi permasalahan tersebut. Sistem pakar akan memindahkan kemampuan pakar tersebut ke dalam komputer. Bakat-bakat yang digunakan dalam oenelitian ini adalah bakat anak menurut standar USEO Amerika. Untuk mesin inferensi penelitian ini menggunakan forward chaining. Anak-anak yang diidentifikasi bakatnya adalah anak TK usia 4-6 tahun. Hasil analis menunjukan bahwa sistem pakar ini membutuhkan 27 indikator, 83 variabel dan 33 rule. Berdasarkan hasil percobaan, sistem pakar ini berhasil mengidentifikasi bakat anak.

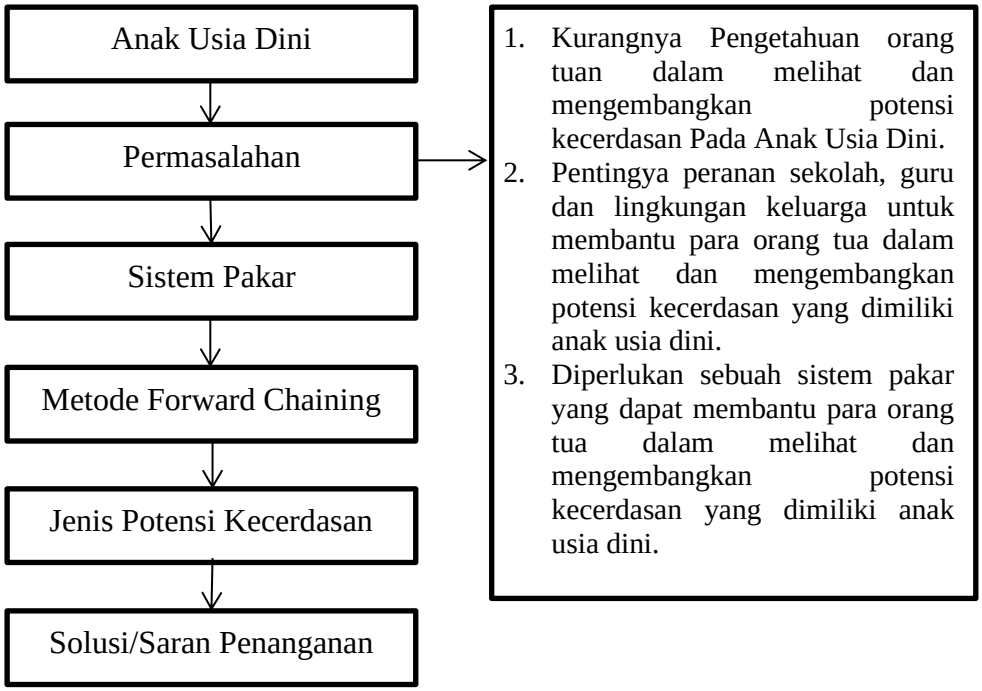
2.5. Kerangka Pemikiran

Menurut Hamdi dan Bahrudin (2014:32) melalui uraian dalam kerangka berpikir, peneliti dapat menjelaskan secara *konprehensif* variabel-variabel apa saja yang dan diteliti dan dari teori apa variabel-variabel itu diturunkan, serta mengapa variabel-variabel itu saja yang diteliti. Uraian dalam kerangka berpikir harus mampu menjelaskan dan menegaskan secara *komprehensif* asal-usul variabel yang diteliti, yang sinyalemennya telah dikemukakan dalam rumusan masalah dan identifikasi masalah semakin jelas asal-usulnya.

Dengan demikian, uraian atau paparan yang harus dilakukan dalam kerangka berpikir adalah paduan antara asumsi-asumsi teoritis dan asumsi-asumsi logika dalam menjelaskan atau memunculkan *variabel-variabel* yang diteliti serta bagaimana kaitan diantara variabel-variabel tersebut, ketika dihadapkan pada kepentingan untuk mengungkapkan fenomena atau masalah yang diteliti. Didalam menulis kerangka berpikir, ada 3 kerangka yang perlu dijelaskan, yakni: kerangka teoritis, kerangka konseptual, dan kerangka operasional.

1. Kerangka *Teoritis* adalah uraian yang menegaskan tentang teori apa saja yang dijadikan landasan serta asumsi-asumsi teoritis yang mana dari teori tesebut yang akan digunakan untuk menjelaskan fenomena yang diteliti.
2. Kerangka *Konseptual* adalah uraian yang menjelaskan konsep-konsep apa saja yang terkandung dalam asumsi-asumsi teoritis yang akan digunakan untuk *mengabstraksikan* (mengistilahkan) unsur-unsur yang terkandung didalam fenomena yang akan diteliti dan bagaimana hubungan di antara konsep-konsep tersebut

3. Kerangka *Operasional* adalah penjelasan tentang variabel-variabel apa saja yang diturunkan dari konsep-konsep terpilih tadi dan bagaimana hubungan di antara variabel-variabel tersebut, serta hal-hal apa saja yang dijadikan indikator untuk mengukur variabel-variabel tersebut.



Gambar 2.23. Kerangka Pemikiran
Sumber: Data Penelitian, 2017

Dari **Gambar 2.22**. Kerangka Pemikiran, penelitian dimulai dari Anak Usia Dini dimana terdapat beberapa permasalahan yang telah dapat di identifikasiakan yaitu:

1. Kurangnya Pengetahuan orang tua dalam melihat dan mengembangkan potensi kecerdasan Pada Anak Usia Dini.

2. Pentingya peranan sekolah, guru dan lingkungan keluarga untuk membantu para orang tua dalam melihat dan mengembangkan potensi kecerdasan yang dimiliki anak usia dini.
3. Diperlukan sebuah sistem pakar yang dapat membantu para orang tua dalam melihat dan mengembangkan potensi kecerdasan yang dimiliki anak usia dini.

Dari identifikasi masalah yang telah didapatkan, dibuatlah Sistem pakar dengan menggunakan Metode *Forward Chaining*. Dimana Metode *Forward Chaining* merupakan penalaran dimulai dari fakta-fakta terlebih dahulu untuk menguji kebenaran hipotesis. Dari fakta-fakta yang telah ditentukan didapatkan jenis petensi kecerdasan yang terdapat pada Anak Usia Dini, kemudian dari jenis kecerdasan tersebut diberikan solusi/saran agar potensi kecerdasan tersebut dapat berkembang.