BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

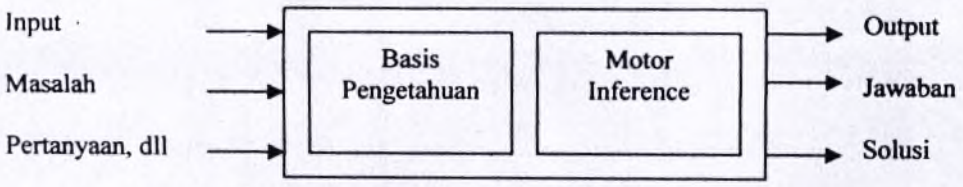
2.1.1 *Artificial Intellegent [Sri Kusumadewi:2003]*

Kecerdasan buatan atau *Artificial Intelegence (AI)* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yag dilakukan oleh manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia. Bentuk dari kecerdasan buatan diterapkan dalam suatu piranti lunak. Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan, dan mempunyai kemampuan untuk menalar.

Untuk melakukan aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan yaitu:

a. Basis Pengetahuan *(Knowledge Base)*, berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.

b. Motor Inferensi *(Inference Engine)*, yaitu kemampuan menaarik kesimpulan berdasarkan pengalaman.



Gambar 2.1 Penerapan Konsep Kecerdasan Buatan di Komputer

2.1.1.1 Sejarah Kecerdasan Buatan

Kecerdasan Buatan dimunculkan oleh seorang profesor dari Massachusetts Institute of Technology yang bernama John McCarthy pada tahun 1956 pada Dartmouth Conference yang dihadiri oleh para peneliti *AI*. Pada konfrensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu mengetahui dan memodelkan proses-proses berfikir manusia dan mendesain mesin agar dapat menirukan kelakuan manusia tersebut.

Pada tahun yang sama komputer berbasis *AI* pertama kali dikembangkan dengan nama *Logic Theorist* yang melakukan penalaran terbatas untuk teorema kalkulus. Perkembangan ini mendorong para peneliti untuk mengembangkan program lain yang disebut sebagai *General Problem Solver* (GPS). Program ini bertujuan untuk memecahkan berbagai jenis masalah dan ternyata menjadi tugas yang sangat besar dan sangat berat untuk dikembangkan.

Beberapa program *AI* yang mulai dibuat pada tahun 1956-1966, antara lain:

1. Sad Sam, diprogram oleh Robert K. Lindsay (1960). Program ini dapat mengetahui kalimat-kalimat sederhana yang ditulis dalam bahasa Inggris dan mampu memberikan jawaban dari fakta-fakta yang didengar dalam sebuah percakapan.
2. ELIZA, diprogram oleh Joseph Weizenbaum (1967). Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan.

2.1.1.2 Lingkup Kecerdasan Buatan Pada Aplikasi Komersial

Karakteristik 'cerdas' sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. Kecerdasan buatan tidak hanya domain di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang lain. Lingkup utama dalam kecerdasan buatan adalah:

- Sistem Pakar *(Expert System)*, komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalah dengan meniru keahlian yang dimiliki seorang pakar.
- Pengolahan bahasa alami *(Natural Language Processing)*, dengan pengolahan bahasa alami ini diharapkan *user* dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
- Pengenalan Ucapan *(Speech Recognition)*, melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer dengan menggunakan suara.

- Robotika dan Sistem Sensor.
- *Computer Vision*, mencoba untuk dapat menginterpretasikan gambar atau obyek-obyek tampak melalui komputer.
- *Intelligent Computer-aided Instruction*, komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
- *Game Playing*.

2.1.2 Sistem Pakar

Menurut *Setiawan Sandy (1994:265)*, Sistem Pakar adalah suatu program berbasis pengetahuan yang menyediakan solusi-solusi dengan kualitas pakar untuk memecahkan masalah-masalah dalam suatu *domain* (wilayah) yang spesifik. Pada umumnya, pengetahuan diambil dari seorang manusia yang pakar dalam *domain* tersebut dan sistem pakar berusaha menirukan metodologi dan kinerjanya. Sistem Pakar adalah salah satu cabang *AI* yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak tidak mengetahui atau mampu dalam bidang yang dimilikinya.

Knowledge dalam sistem pakar mungkin saja seorang ahli, atau *knowledge* yang umumnya terdapat dalam buku, majalah dan orang yang mempunyai pengetahuan tentang suatu bidang. Sistem pakar dapat

mengumpulkan dan menyimpan pengetahuan seorang pakar atau beberapa pakar ke dalam program komputer.

Tujuan dari sebuah sistem pakar adalah untuk men-*transfer* kepakaran yang dimiliki seorang pakar ke dalam komputer, dan kemudian kepada orang lain. Ada beberapa alasan mendasar mengapa sistem pakar dikembangkan untuk menggantikan seorang pakar, di antaranya:

- Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
- Secara otomatis mengerjakan tugas-tugas rutin yang membutuhkan seorang pakar
- Seorang pakar adalah mahal.
- Seorang pakar akan pensiun atau pergi.
- Kepakaran dibutuhkan juga pada lingkungan yang tidak bersahabat *(hostile environment).*

Beberapa contoh aplikasi sistem:

- MYCIN, untuk mendiagnosa penyakit.
- DENDRAL, mengidentifikasi struktur molekular campuran yang tidak dikenal.
- XCON dan XSEL, membantu konfigurasi sistem komputer besar.
- SOPHIE, analisis sirkit elektronik.

2.1.2.1 Konsep Sistem Pakar

Menurut Turban, konsep sistem pakar adalah:

a. **Keahlian *(Expertise)***

Keahlian merupakan suatu pengetahuan khusus yang dimiliki oleh seorang yang diperoleh melalui latihan, belajar dan pengalaman dalam bidang tertentu dalam jangka waktu yang cukup lama. Pengetahuan tersebut dapat berupa fakta-fakta, teori, aturan serta strategi untuk menyelesaikan suatu masalah. Dengan pengetahuan tersebut seseorang dapat memberikan keputusan yang lebih baik dan cepat dalam menyelesaikan suatu permasalahan.

b. **Ahli Pakar (*Expert)***

Seorang pakar adalah seorang individu yang memiliki kemampuan pemahaman yang superior dari suatu masalah. Kemampuan kepakaran antara lain: dapat mengenali dan merumuskan masalah, menyelesaikan masalah dengan cepat dan tepat, menjelaskan solusi, belajar dari pengalaman, restrukturisasi pengetahuan, menentukan relevansi/hubungan, memahami batas kemampuan. Seluruh karakteristik diatas harus dimiliki oleh seorang pakar dan untuk meniru kepakaran seorang perlu dibangun suatu sistem komputer yang menunjukan karakteristik di atas.

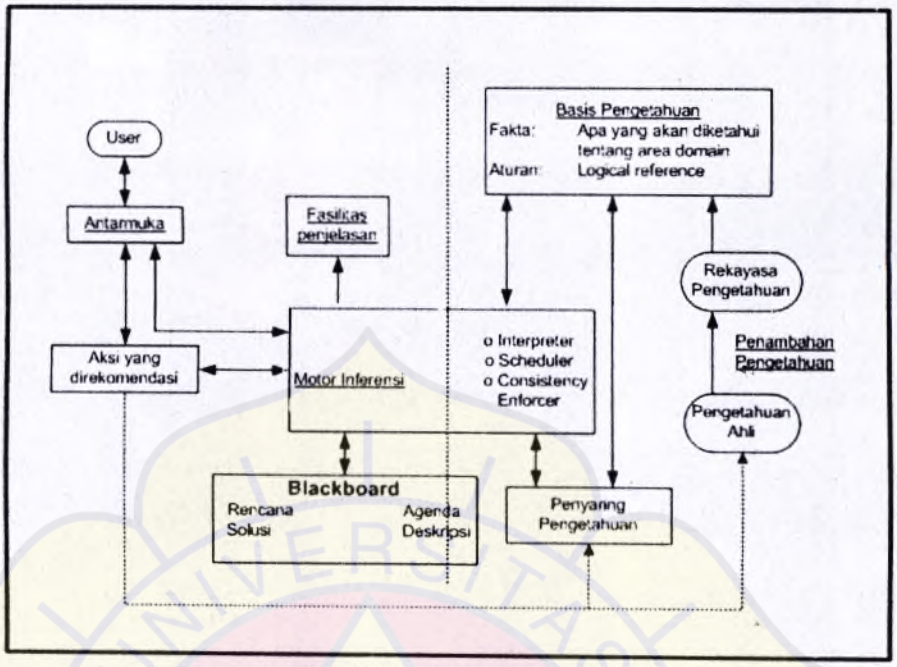
c. **Mentransfer Keahlian *(Transfering Expertise)***

Aktifitas yang dilakukan untuk memindahkan kepakaran:

- *Knowledge Acquisition*, meliputi proses pengumpulan, pemindahan, dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi (buku, dan lain-lain) ke dalam komputer.
- *Knowledge Representation*, melakukan representasi pengetahuan ke dalam komputer.
- *Knowledge Inferencing*, mencocokan fakta-fakta dengan domain pengetahuan, untuk menarik kesimpulan dari masalah yang dihadapi.
- *Knowledge Transfering*, memindahkan pengetahuan dari komputer kepada orang lain *(non-expert)*.

2.1.2.2 Struktur Sistem Pakar

Sistem pakar terdiri dari dari 2 bagian pokok, yaitu: lingkungan pengembangan *(development environment)* dan lingkungan konsultasi *(consultation environment)*. Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangunan komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seseorang yang bukan ahli untuk berkonsultasi.



Gambar 2.2 Struktur sistem pakar

Keterangan gambar :

- Subsistem penambahan pengetahuan, untuk memasukan pengetahuan, mengkontruksi atau memperluas pengetahuan dalam basis pengetahuan.
- Basis pengetahuan, berisi pengetahuan-pengetahuan yang dibutuhkan untuk memaahami, memformulasikan dan menyelesaikan masalah.
- Motor inferensi *(inference engine)*, berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan dan *blackboard.*
  - *Interpreter*: mengeksekusi item-item agenda yang terpilih dengan menggunakan aturan-aturan dalam basis pengetahuan yang sesuai.
  - *Scheduler*: mengontrol agenda.

- *Consistency enfoercer*: berusaha memelihara kekonsistenan dalam merepresentasikan solusi yang bersifat darurat.

- *Blackboard*, merekam kejadian yang sedang berlangsung termasuk keputusan sementara.
  - Rencana: bagaimana menghadapi masalah.
  - Agenda: aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi.
  - Solusi: calon aksi yang akan dibangkitkan.
- Antarmuka, media komunikasi antara *user* dan program.
- Subsistem penjelasan, untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif.
- Sistem penyaring pengetahuan, untuk mengevaluasi kinerja sistem pakar itu sendiri untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok ùntuk digunakan di masa mendatang.

2.1.2.3 Komponen Utama Sistem Pakar

Sebuah program sistem pakar terdiri dari komponen-komponen utama, sebagai berikut:

a. Basis Pengetahuan (*Knowledge Base)*

Basis pengetahuan merupakan inti dari progam sistem pakar berisikan semua fakta yang berisi informasi tentang objek, kaidah *(rule)* dan merupakan informasi tentang bagaimana membangkitkan fakta baru dari fakta yang sudah diketahui

prosedur yang terorganisasi dengan teranalisa agar lebih dimengerti dan diterapkan pada pemecahan masalah dan pengambilan keputusan.

b. *Working memory*

Working Memory adalah bagian dari sistem pakar yang berisi fakta-fakta masalah ditemukan dalam satu sesi. Berisi fakta-fakta tentang suatu masalah yang ditemukan dalam proses konsultasi.

c. Mesin Inferensi *(Inference Engine)*

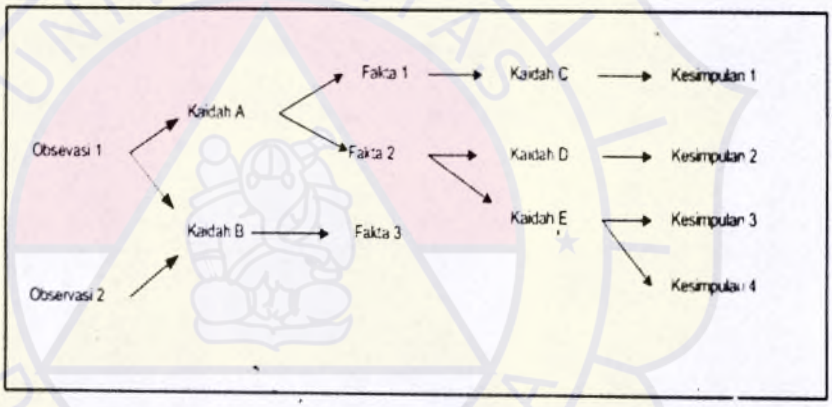
Mesin inferensi bertugas untuk menganalisa pengetahuan dan menarik kesimpulan berdasarkan basis pengetahuan dengan cara yang sistematis pada saat menjawab memecahkan persoalan atau membuat keputusan dalam suatu domain yang telah ditentukan. Cara komputer berfikir tentang suatu subjek domain, mesin inferensi melakukan pelacakan melalui basis pengetahuan untuk mencari jawaban dan solusinya.

Ada dua cara yang dapat dikerjakan dalam melakukan inferensi, yaitu *[Sri Kusumadewi:2003]:*

- ***Forward Chaining***

  Pelacakan kedepan *(Forward Chaining)*, pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Setiap aturan dapat kembali membuat kondisi baru dari kesimpulan yang diambil dari bagian maka kondisi yang baru ini ditambahkan pada kondisi yang telah ada. Semua kondisi

yang telah ditambahkan kedalam proses, jika ada kondisi seperti itu, sistem kembali kelangkah kedua dan mencari aturan dalam pengetahuan lagi. Jika tidak ada kondisi baru, tahap ini terakhir. Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis. Kesimpulan dari suatu kasus yang dibanguń berdasarkan fakta-fakta yang telah diketahui, proses pelacakan dimulai dari keadaan awal menuju kepada tujuan.



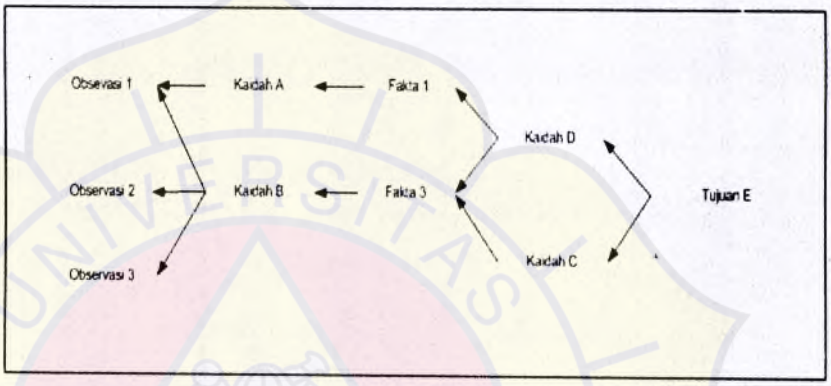
Gambar 2.3 Pelacakan Forward Chaining

Dalam metode *forward chaining* ini pelacakan dimulai dari informasi awal dan bergerak maju sambil mencocokan informasi fakta yang ada dengan kaidah yang ada untuk mendapatkan kesimpulan.

- ***Backward Chaining***

  Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN dulu). *Backward chaining* digunakan jika hasil (kesimpulan) telah dimiliki dan selanjutnya ingin ditentukan

kondisi yang menyebabkan hal tersebut. Dengan kata lain, penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut dicari harus dicari fakta-fakta yang ada dalam basis pengetahuan.



Gambar 2.4 Pelacakan Backward Chaining

Dalam metode *backward chaining* ini pelacakan dimulai dengan mengidentifikasikan sasaran akhir (kesimpulan sementara atau akhir) yaitu Tujuan E, kemudian dilacak ke belakang melalui kaidah-kaidah dan fakta-fakta yang ada untuk dapat menentukan kondisi yang menyebabkan hal tersebut.

d. *User Interface*

Antarmuka pemakai *(user interface)* adalah bagian penghubung antara program sistem pakar dengan pemakai. Pada bagian ini akan terjadi dialog antara program dengan pemakai. Program akan

memberikan pertanyaan yang bersifat "ya" atau "tidak" atau berbentuk menu pilihan. Program sistem pakar akan mengambil kesimpulan berdasarkan jawaban-jawaban yang diberikan *user*.

2.1.2.3 Ciri-Ciri Sistem Pakar

Sistem pakar yang baik harus memenuhi ciri-ciri sebagai berikut:

- Terbatas pada domain keahlian tertentu.
- Berdasarkan pada kaidah tertentu.
- Dapat menghasilkan rangkaian alasan-alasan yang diberikan dengan cara yang dapat dipahami.
- Memiliki fasilitas informasi yang handal.
- Dapat digunakan dalam berbagai jenis komputer.
- Memiliki kemampuan untuk belajar beradaptasi.

2.1.2.4 Keuntungan dan Kelemahan Sistem Pakar

Keuntungan dari sistem pakar, antara lain:

- Menyimpan pengetahuan dan keahlian seorang pakar.
- Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka).
- Mampu beroperasi dalam lingkungan yang berbahaya.
- Memiliki kemampuan untuk mengakses pengetahuan.
- Memiliki realibilitas dan meningkatkan kapabilitas sistem komputer.

- Menghemat waktu dalam pengambilan keputusan.

Kelemahan dari sistem pakar antara lain:

- Masalah dalam mendapatkan pengetahuan di mana pengetahuan tidak selalu bisa didapatkan dengan mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, dan kalaupun ada kadang-kadang pendekatan yang dimiliki oleh pakar berbeda-beda.
- Untuk membuat suatu sistem pakar yang benar-benar berkualitas sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dari pemeliharaannya.
- Boleh jadi sistem tidak dapat membuat keputusan.
- Sistem pakar tidaklah 100% benar.

2.1.2.4 Kategori Problema Sistem Pakar

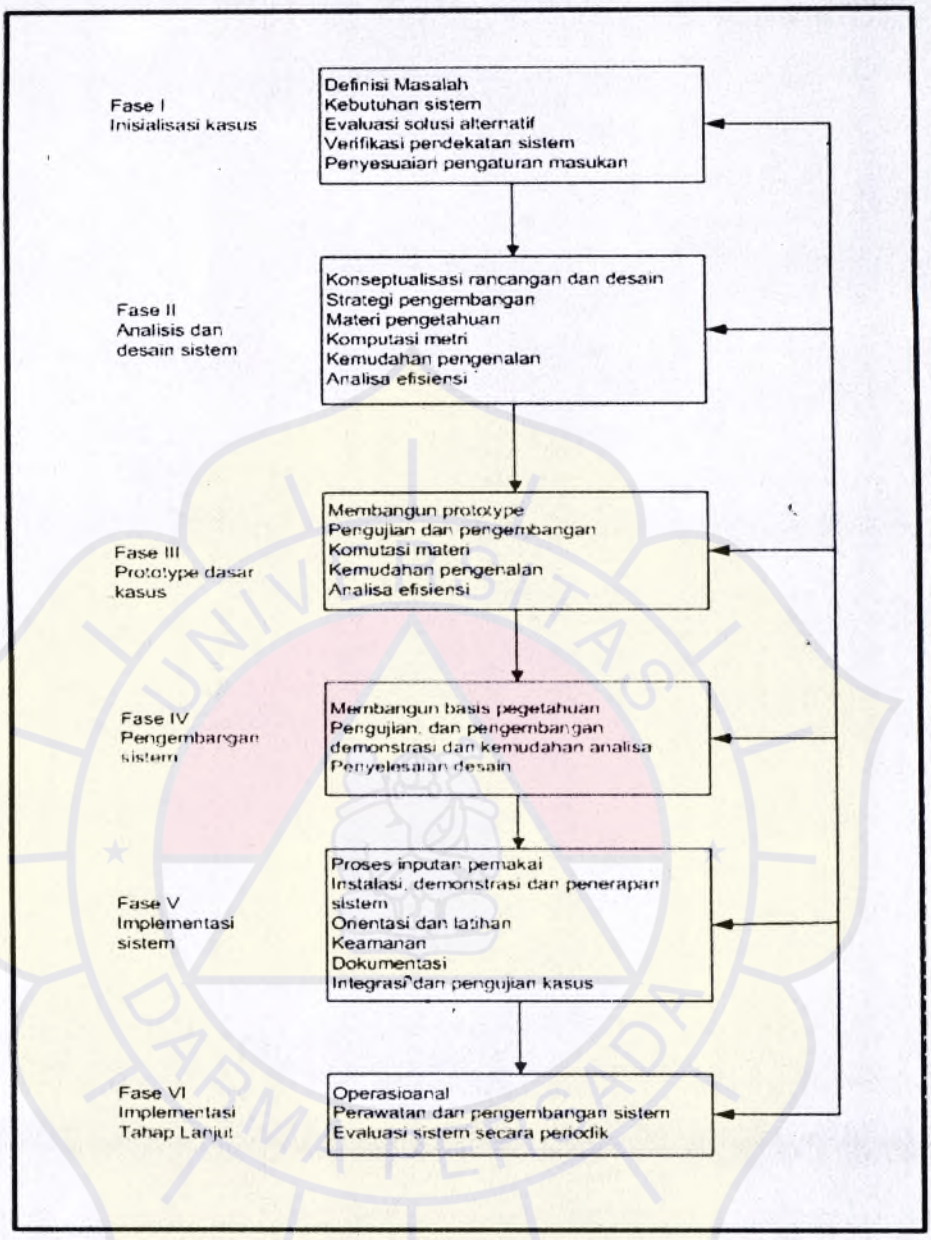
Ada beberapa masalah yang menjadi area luas aplikasi sistem pakar, antara lain:

- Interpretasi, membuat kesimpulan atau deskripsi dari sekumpulan data mentah.
- Prediksi, memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu.
- Diagnosis, menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati.

- Perancangan, menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala tertentu.
- Perencanaan, merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu.
- *Monitoring*, membandingkan hasil pengamatan dengan kondisi yang diharapkan.
- *Debbuging* dan *Repair*, menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.
- Instruksi, mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subyek.
- Kontrol, mengatur tingkah laku suatu *environment* yang kompleks.

2.1.2.5 Pengembangan Sistem Pakar

Pengembangan sistem yang paling berguna adalah proses dokumentasi sistem di mana di dalamnya tersimpan semua hal penting yang dapat menjadi tolak ukur pengembangan sistem di masa mendatang termasuk di dalamnya adalah kamus pengetahuan yang diselesaikan



Gambar 2.5 Tahap-tahap pengembangan sistem pakar

Secara garis besar pengembangan sistem pakar pada gambar adalah sebagai berikut:

1. Mengidentifikasi masalah dan kebutuhan.
2. Menentukan masalah yang cocok. Ada beberapa syarat yang harus dipenuhi agar sistem pakar dapat bekerja dengan baik, yaitu:

- Domain masalah tidak terlalu luas
- Kompleksitasnya menengah, artinya jika masalah terlalu mudah atau masalah yang sangat kompleks seperti peramalan inflasi tidak perlu menggunakan sistem pakar
- Tersedianya ahli atau pakar
- Tidak melibatkan hal-hal yang bersifat *common sense*, yaitu penalaran yang diperoleh dari pengalaman.

3. Mempertimbangkan alternatif (menggunakan sistem pakar atau komputer tradisional).
4. Menghitung pengembalian investasi, termasuk diantaranya biaya pembuatan sistem pakar, biaya pemeliharaan, dan biaya training.
5. Memilih alat pengembangan.
6. Rekayasa pengetahuan. Perlu dilakukan penyempurnan terhadap aturan-aturan yang sesuai.
7. Merancang sistem. Bagian ini termasuk pembuatan *prototype*, serta menterjemahkan pengetahuan menjadi aturan-aturan.
8. Melengkapi pengembangan. Termasuk pengembangan *prototype* apabila sistem yang telah ada sudah sesuai dengan keinginan.
9. Menguji dan mencari kesalahan sistem
10. Memelihara sistem (memperbaharui pengetahuan, mengganti pengetahuan yang sudah ketinggalan, dan meluweskan sistem agar bisa lebih baik lagi dalam menyelesaikan masalah).

2.1.3 Web Based Application

2.1.3.1 Pengertian *Website*

Sebuah **situs web** (sering pula disingkat menjadi **situs** saja; *web site*, *site*) adalah sebutan bagi sekelompok halaman web (*web page*), yang umumnya merupakan bagian dari suatu nama *domain* (*domain name*) atau *subdomain* di *World Wide Web* (WWW) di Internet. WWW terdiri dari seluruh situs web yang tersedia kepada publik. Pada awalnya aplikasi web dibangun dengan menggunakan bahasa yang disebut HTML *(Hyper Text Markup Language)* dan protokol yang digunakan dinamakan HTTP *(Hyper Text Transfer Protocol)*. Pada perkembangan berikutnya, sejumlah *script* dan objek dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak *script* seperti itu, antara lain yaitu PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah applet (Java)

Aplikasi Web sendiri dapat dibagi menjadi *[Abdul Kadir:2003]:*

- Web statis, dan
- Web dinamis.

Web statis dibentuk hanya menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perubahan yang terjadi.

Dengan memperluas kemampuan HTML, yaitu dengan menggunakan perangkat lunak tambahan seperti PHP atau ASP serta DBMS, perubahan informasi dalam halaman-halaman Web dapat ditangani melalui perubahan data, bukan melalui perubahan program.

Sebagai implementasinya, aplikasi web dapat dikoneksikan ke *database*, halaman dilengkapi dengan animasi gambar, ini yang dimaksud dengan web dinamis.

Pengertian dasar dari komponen yang terdapat pada *website* antara lain:

- **HTML** ***(Hyper Text Markup Language)***

  Format dokumen yang digunakan dalam *World Wide Web* (WWW). HTML merujuk pada tampilan halaman, jenis, huruf, elemen grafis, juga *link hypertext* ke dokumen lain di Internet.

- **HTTP** ***(Hyper Text Transfer Protocol)***

  Sebuah metode atau protokol untuk men*download* file ke komputer. Protokol ini berbasis *hypertext*, sebuah format yang umum digunakan di Internet.

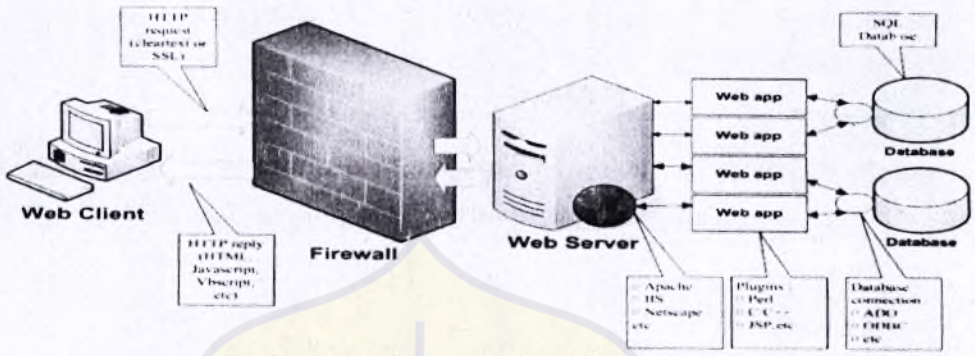
- **WWW** *[wikipedia.com: 2006]*

  World Wide Web ("**WWW**", atau singkatnya "**Web**") adalah suatu ruang informasi di mana sumber-sumber daya yang berguna diidentifikasi oleh pengenal global yang disebut *Uniform Resource Identifier* (URI).

- **ISP** ***(Internet Service Provider)***

  Perusahaan yang menawarkan dan menyediakan layanan akses Internet ke kalangan umum dengan mengenakan biaya. Contoh ISP yang terdapat di Indonesia antara lain Telkomnet, Indosatnet, Centrin, Cbn, Wasantara.

2.1.3.2 Arsitektur Website



Gambar 2.6 Komponen Aplikasi Website

Komponen-komponen yang terdapat pada arsitektur *website* terdiri dari:

- ***Web Sever***

  Server web adalah sebuah perangkat lunak *server* yang berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. *Web Server* yang terkenal diantaranya adalah *Apache* dan *Microsoft Internet Information Service* (IIS). Apache merupakan *web server* antar-*platform*, sedangkan IIS hanya dapat beroperasi di sistem operasi Windows.

- **DBMS**

  DBMS *(DataBase Management System)* adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien. DBMS dapat digunakan untuk mengakomodasikan

berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda. Umumnya DBMS menyediakan fitur-fitur seperti independensi data-program, keamanan, integritas, konkurensi, *recovery*, katalog sistem, perangkat produktivitas. Perangkat lunak yang termasuk DBMS seperti dBase III+, dBase IV, FoxBase, Rbase, MS-Access dan Borland-Paradox (untuk kelas sederhana) atau Borland-Interbase, MS_SQL Server, CA-Open Ingres, Oracle, Informix dan Sybase (untuk kelas kompleks). MySQL, PostgreSQL *(open source).*

- ***Middleware** [Abdul Kadir:2003]*

  Middleware adalah perangkat lunak yang bekerja sama dengan *Web Server* dan berfungsi menerjemahkan kode-kode tertentu, menjalankan kode-kode tersebut, dan memungkinkan berinteraksi dengan basis data. Ada banyak bahasa web yang populer di dunia Internet seperti HTML, DHTML, XML, Perl, PHP, ASP, Cold Fusion, dan lain-lain. HTML merupakan *framework* Internet, hampir semua situs web yang ada menggunakan HTML untuk menampilkan teks, grafik, suara, dan animasi.

2.1.3.3 Teknologi Website

Teknologi yang digunakan dalam membuat *website* antara lain :

- **Apache**

Apache adalah *Web Server* yang melayani permintaan klien terhadap halaman Web. Apache telah menjadi *Web Server* terpopuler saat ini karena telah bekerja pada semua *platform* seperti NetBSD, UNIX, AIX, OS/2, Windows, HPUX, Novell Netware, Macintosh, BeOS, FreeBSD, IRIX, dan Solaris. Fitur-fitur yang ditawarkan seperti *host virtual, server-side includes*, halaman Web dinamis dari *CGI*, pesan kesalahan yang dapat dikonfigurasi, dan pemetaan *URL* ke *system file*. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (*GUI*) yang memungkinkan penanganan *server* menjadi mudah.

- **PHP *(Personal Home Page)***

  PHP merupakan skrip yang berjalan di *server* dan sangat populer di lingkungan Linux. Saat ini PHP dapat berjalan pada berbagai *platform*, dari UNIX hingga Windows. PHP adalah salah satu *middleware* yang berkerja sama dengan *Web Server* dan berfungsi menerjemahkan kode-kode tertentu, menjalankan kode-kode tersebut, dan memungkinkan berinteraksi dengan *database*. PHP awalnya digunakan untuk mencatat pengunjung yang membuka halaman tertentu. PHP merupakan bahasa *server-side* yang paling banyak digunakan untuk membuat aplikasi *standalone* yang tidak terkait dengan Web.

- **JavaScript** *[Abdul Kadir:2003]*

JavaScript adalah bahasa skrip yang ditempelkan pada kode HTML dan di proses pada sisi klien. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi luas. Sebagai contoh, dengan menggunakan JavaScript dimungkinkan untuk memvalidasi masukan-masukan pada formulir sebelum formulir dikirimkan ke *server*.

- **PostgreSQL** *[Andhie Lala Adam:2004]*

  PostgreSQL merupakan hubungan DBMS yang membantu sebuah model data yang terdiri dari hubungan nama dan berisikan atribut dari sebuah tipe spesifik. PostgreSQL memiliki keluwesan dan kinerja yang tinggi. Dengan arsitektur yang luwes, maka sebuah *user* PostgreSQL mampu mendefinisikan sendiri SQL-nya. PostgreSQL merupakan *software* yang bersifat gratis.

2.1.4 Penyakit THT

Penyakit THT dibagi menjadi beberapa bagian, antara lain *[Prof, dr, H. Nurbaiti Iskandar, Sp, THT: 2006]* :

2.1.4.1 Telinga

Telinga terdiri atas telinga luar, telinga tengah dan telinga dalam. Kelainan telinga dibagi menjadi:

- Kelainan telinga luar.
- Penyakit telinga tengah.
- Kelainan telinga dalam.

2.1.4.2 Hidung

Fungsi hidung antara lain untuk mencium bau, untuk bernapas, sebagai resonansi suara, memanaskan dan melembabkan udara yang diinspirasi melalui hidung. Beberapa contoh kelainan dan penyakit hidung yaitu *Atresia* hidung, *Espitaksis*, Trauma hidung, *Rinitis*, Kelainan *Septum*, Kelainan *Nasofaring*, *Sinus Paranasal*.

2.1.4.3 Tenggorok

Tenggorok (*faring)* dimulai sejak dasar tengkorak sampai setinggi *vertebra servikal* ke 6, yang kemudian dilanjutkan dengan *esofagus*, sebagai alat cerna. *Faring* dibagi atas *nasofaring, orofaring,* dan *hipofaring.* Kelainan pada tenggorok antara lain Kelainan *Faring* dan Kelainan *Laring.*

2.2 Penelitian yang Relevan

Pada penulisan tugas akhir ini, penulis juga mempunyai referensi yang didapat dari hasil penelitian (skripsi) mahasiswa YAI tentang sistem pakar diagnosa penyakit mulut, digunakan sebagai informasi dan bahan acuan untuk membuat skripsi sistem pakar diagnosa penyakit THT ini.

2.3. Kerangka Pikir

Dalam sistem pakar dianosa penyakit THT ini, terdapat keterkaitan antara komponen-komponen di dalam sistem. Sistem pakar

dibuat karena adanya kebutuhan *user* untuk menggunakan suatu produk tentang *knowledge* dari suatu *domain* masalah yang dapat memberikan kesimpulan/solusi tanpa mereka harus langsung bertanya kepada pakarnya. Dengan adanya sistem pakar, dapat memberikan solusi tentang terbatasnya jumlah pakar dan biaya pakar yang mahal. *Knowledge Base* didapat dari pengetahuan seorang pakar, dokumentasi, buku, dan sumber lainnya yang mendukung pengetahuan tersebut. *User* dapat berinteraksi dengan sistem pakar dari suatu input yang diberikan dengan menjawab pertanyaan-pertanyaan yang diberikan komputer, kemudian komputer memberikan suatu kesimpulan. Sistem pakar harus bersifat *User Friendly*, agar mudah digunakan oleh *user* yang awam sekalipun. Teknologi *website* dapat dijadikan sebagai media sistem pakar, sehingga dapat diakses dimana saja.