

BAB 2

LANDASAN TEORI

2.1 Kecerdasan Buatan

2.1.1 Pengertian Kecerdasan Buatan

Kecerdasan buatan (*artificial intelligence*) adalah bagian dari ilmu komputer yang mempelajari tentang bagaimana sebuah komputer bisa dibuat dengan sedemikian rupa agar dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Menurut John McCarthy (1956), kecerdasan buatan adalah suatu sistem komputer yang terbentuk untuk mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia.

Manusia bisa dengan pandai menyelesaikan masalah-masalah yang muncul karena manusia memiliki pengetahuan dan pengalaman yang dapat membantu dalam memecahkan masalah. Agar komputer dapat bertindak seperti dan sebaik manusia maka komputer diberikan pengetahuan dan kemampuan untuk menalar agar dapat mendapatkan pengalaman seperti layaknya manusia.

Ada tiga tujuan kecerdasan buatan, yaitu: membuat komputer lebih cerdas, mengerti tentang kecerdasan, dan membuat mesin lebih berguna. Yang dimaksud kecerdasan adalah kemampuan untuk belajar atau mengerti dari pengalaman, memahami pesan yang kontradiktif dan ambigu, menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah serta menyelesaikannya dengan efektif (Winston dan Prendergast, 1994).

Apa itu kecerdasan buatan? Beberapa buku telah mendefinisikan tentang kecerdasan buatan, dan dapat dikelompokkan menjadi empat bagian, yaitu:

- a. Sebuah sistem yang berpikir seperti manusia
- b. Sebuah sistem yang dapat berpikir secara rasional
- c. Sebuah sistem yang berperilaku seperti manusia
- d. Sebuah sistem yang dapat berperilaku secara rasional

Lebih jauh lagi, berikut adalah beberapa definisi mengenai *kecerdasan buatan* yang dapat diketahui, yaitu:

- a. Menurut Rich dan Knight (1991, p3)

Kecerdasan buatan merupakan ilmu yang mempelajari bagaimana membuat sebuah komputer dapat mengerjakan sesuatu yang masih lebih baik dikerjakan manusia.

- b. Menurut Rolston (1988, p15)

Kecerdasan buatan merupakan solusi berbasis komputer terhadap masalah yang ada, yang menggunakan aplikasi yang mirip dengan proses berpikir menurut manusia.

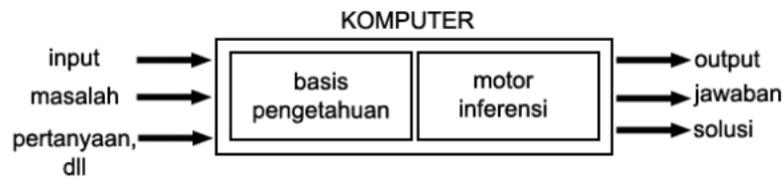
- c. Menurut Setiawan (1993, p1)

Kecerdasan buatan dapat didefinisikan sebagai cabang ilmu komputer yang mempelajari otomisasi tingkah laku cerdas.

Terdapat dua bagian utama yang diperlukan agar dapat melakukan aplikasi kecerdasan buatan, seperti dapat terlihat pada Gambar 2.1, yaitu:

- a. Basis pengetahuan (*knowledge base*), yang berisi fakta, teori, pemikiran, dan hubungan satu dengan yang lainnya,

- b. Motor inferensi (*inference engine*), yang berupa kemampuan mesin untuk menarik kesimpulan berdasarkan pengalaman.



Gambar 2.1 Penerapan Konsep Kecerdasan Buatan di Komputer (Turban dan Frenzel, 1992, p 12)

2.1.1.1 Turing Test

Untuk mengetahui apakah sebuah sistem komputer dapat dikatakan mendekati kemampuan berpikir manusia, terdapat suatu tes yang sering dipakai untuk menentukan apakah komputer tersebut layak untuk dikategorikan sebagai komputer yang ‘cerdas’. Percobaan Turing (*Turing Test*), ditemukan oleh Alan Turing, seorang pioneer AI dan ahli matematika asal Inggris, sekitar tahun 1950-an, melakukan eksperimen dengan menempatkan komputer di satu terminal dan terminal lain dikendalikan oleh manusia. Kedua terminal ini ditempatkan di posisi berjauhan. Operator tersebut berkomunikasi dengan komputer dan menganggap bahwa lawan komunikasinya adalah seorang manusia juga. Menurut pendapat Turing, jika seseorang bisa salah mengira komputer sebagai manusia, maka komputer tersebut bisa dikatakan ‘cerdas’.

Untuk bisa dikatakan lolos *Turing Test*, computer harus memiliki kemampuan untuk memproses beberapa proses berikut:

- a. *Natural language processing*, dimana komputer mampu berkomunikasi dengan lancar menggunakan bahasa Inggris.

- b. *Knowledge representation*, dimana komputer mampu menyimpan informasi-informasi yang diberi tahu.
- c. *Automated reasoning*, dimana komputer menggunakan informasi (pengetahuan) yang telah disimpan untuk menjawab pertanyaan dan membuat kesimpulan baru.
- d. *Machine learning*, dimana komputer mampu beradaptasi dan memperkirakan pola-pola yang akan terjadi/mungkin muncul.

Adapun yang disebut dengan *Total Turing Test*, dimana selain 4 proses diatas, agar komputer bisa dikatakan lolos *Total Turing Test*, komputer juga harus mampu melakukan dua proses berikut:

- a. *Computer vision*, dimana komputer mampu melihat dan menginterpretasikan objek.
- b. *Robotics*, dimana komputer mampu memanipulasi objek dan mampu bergerak.

2.1.2 Kelebihan Kecerdasan Buatan

Kecerdasan buatan memiliki beberapa kelebihan dibandingkan dengan kecerdasan yang dimiliki oleh manusia (Kusumadewi, 2003, p3-4), yaitu:

- a. Bersifat lebih permanen. Selama sistem dan program tidak berubah, maka kecerdasan buatan tersebut tidak akan berubah;
- b. Lebih mudah untuk diduplikasi dan disebar. Akan lebih mudah memindahkan data dari satu komputer ke komputer lain dibandingkan dengan memindahkan pengetahuan dari satu manusia ke manusia yang lain;
- c. Lebih murah dibandingkan dengan mendatangkan seorang ahli;
- d. Konsisten. Kecerdasan buatan merupakan sebuah teknologi komputer sedangkan kecerdasan alami memiliki kecenderungan untuk berubah;

- e. Bisa didokumentasi. Tiap aktifitas yang dilakukan oleh kecerdasan buatan dapat dilacak dengan mudah sedangkan kecerdasan alami termasuk sulit untuk direproduksi.
- f. Mengerjakan pekerjaan dengan waktu lebih cepat dan lebih baik.

2.1.3 Bidang-bidang Terapan pada Kecerdasan Buatan

Kecerdasan buatan adalah sebuah ilmu pengetahuan dan teknologi yang berisikan ide-ide dan konsep yang diperuntukkan untuk sebuah penelitian dan tidak untuk dipasarkan (Turban dan Aronson, 2001, p402-406). Namun kecerdasan buatan menyediakan dasar-dasar ilmu pengetahuan pada beberapa bidang teknologi yang dapat digunakan secara komersial, di antaranya adalah:

- a. Sistem pakar (*expert system*)

Sebuah sistem komputer yang digunakan sebagai sarana untuk menyimpan pengetahuan yang dimiliki oleh seorang atau lebih pakar dengan tujuan agar komputer memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki pakar tersebut.

- b. Pengolahan bahasa alami (*natural language processing*)

Pemrograman sistem komputer dimana pengguna dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.

- c. Pengenalan suara/ucapan (*speech recognition*)

Pengguna dapat berkomunikasi dengan komputer/memberi perintah kepada komputer untuk melakukan suatu pekerjaan.

d. Robotika dan sistem sensor (*robotic and sensory systems*)

Sebuah lengan robotik yang dilengkapi oleh berbagai sensor yang diprogram untuk mampu mendeteksi jenis pekerjaan yang perlu dilakukan oleh lengan tersebut.

e. *Computer vision*

Pemrograman sistem yang bertujuan untuk menginterpretasikan gambar dan objek tampak melalui komputer untuk proses lebih jauh.

f. *Intelligent computer-aided instruction*

Sistem komputer yang digunakan untuk proses pembelajaran/sebagai tutor yang dapat melatih dan mengajar.

g. *Game playing.*

2.2 Sistem Pakar

2.2.1 Pengertian Sistem Pakar

Sistem pakar (*expert system*) merupakan sebuah sistem yang mengadopsi pengetahuan yang dimiliki oleh pakar (manusia) ke dalam komputer, agar komputer dapat menyelesaikan sebuah masalah layaknya seorang ahli menyelesaikan masalah.

Terdapat beberapa definisi tentang sistem pakar, antara lain:

a. Menurut Turban dan Aronson (2001, p402)

Sistem pakar adalah sistem yang menggunakan pengetahuan dari seorang pakar pada bidang tertentu untuk menyelesaikan permasalahan yang pada umumnya membutuhkan keahlian seorang pakar.

- b. Menurut Rich dan Knight (1991, p574)

Sistem pakar adalah sistem yang bertujuan untuk memecahkan masalah yang biasanya dilakukan oleh seorang pakar.

- c. Menurut Setiawan (Setiawan, 1993, p 265).

Sistem pakar adalah suatu sistem berbasis ilmu pengetahuan yang menyediakan solusi-solusi dengan kualitas pakar untuk masalah-masalah dalam suatu bidang yang spesifik.

- d. Menurut Luger dan Stubblefield (1989, p291)

Sistem pakar adalah program berbasis pengetahuan yang memberikan solusi berkualitas pakar untuk permasalahan pada bidang tertentu.

- e. Menurut Jackson (1992, p2)

Sistem pakar adalah sebuah program komputer yang dapat merepresentasikan dan menjelaskan menggunakan pengetahuan yang dimiliki oleh seorang ahli dengan tujuan memecahkan masalah atau memberikan saran solusi pemecahan masalah.

2.2.2 Komponen Dasar Sistem Pakar

Sistem pakar memiliki tiga komponen dasar, yaitu:

1. Basis pengetahuan (*knowledge base*)

Berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan, dan menyelesaikan masalah (fakta-fakta, teori, pemikiran, dan hubungan antara satu dengan yang lainnya). Knowledge base merupakan inti dari sebuah *expert system*.

Terdapat dua bentuk pendekatan basis pengetahuan yang sering digunakan:

a. Penalaran berbasis aturan (*rule-based reasoning*)

Pada bentuk penalaran ini, pengetahuan direpresentasikan dengan menggunakan aturan yang berbentuk IF-THEN, dan digunakan jika memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu. Dan juga, bentuk ini akan digunakan jika dibutuhkan penjelasan tentang langkah-langkah dalam mencapai solusi.

b. Penalaran berbasis kasus (*case-based reasoning*)

Pada bentuk penalaran ini, basis pengetahuan akan berisi solusi-solusi yang dicapai sebelumnya, yang kemudian diturunkanlah suatu solusi yang dicapai dengan keadaan sekarang (fakta yang ada). Bentuk ini digunakan bila *user* menginginkan untuk mengetahui lebih banyak lagi terhadap kasus-kasus yang hampir mirip.

Dalam merepresentasikan pengetahuan, diketahui beberapa bentuk yang sering digunakan, seperti:

a. Logika

Logika merupakan salah satu bentuk representasi pengetahuan yang tertua. Logika adalah proses membentuk kesimpulan berdasarkan fakta yang telah ada, seperti yang terlihat pada Gambar 2.2. Input dari logika berbentuk premis atau fakta yang diakui kebenarannya sehingga dalam melakukan penalaran dengan menggunakan logika akan didapatkan hasil yang benar pula.

Untuk mendapatkan kesimpulan (konklusi), terdapat dua penalaran yang dapat dilakukan, yaitu:

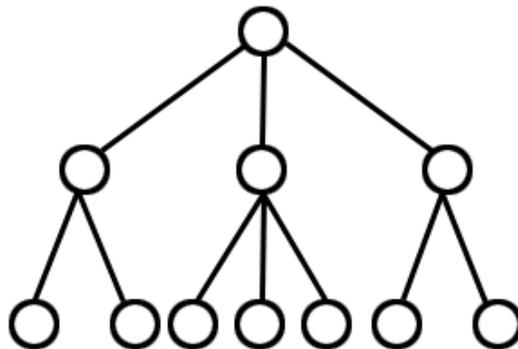
- 1) Penalaran deduktif: penalaran yang dimulai dari prinsip umum untuk mendapatkan konklusi yang khusus.
- 2) Penalaran induktif: penalaran dimulai dari fakta-fakta khusus untuk mendapatkan kesimpulan umum.



Gambar 2.2 Proses Logika

b. Pohon (*tree*)

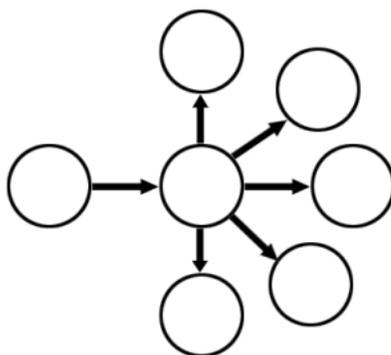
Pohon adalah struktur penggambaran secara hierarkis. Struktur pohon terdiri dari node-node yang menunjukkan objek dan *arc* (busur) yang menggambarkan hubungan antar objek, seperti terlihat pada Gambar 2.3.



Gambar 2.3 Struktur Pohon

c. Jaringan semantik

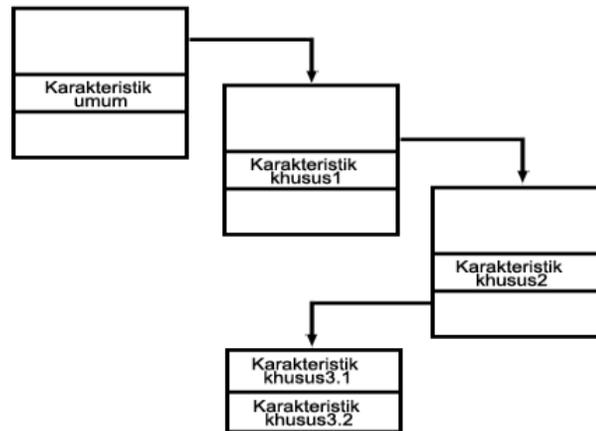
Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek, seperti terlihat pada Gambar 2.4. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tersebut. Jaringan semantik memiliki kelebihan untuk 'bisa mewariskan'.



Gambar 2.4 Perluasan Jaringan Semantik

d. Frame

Frame adalah kumpulan pengetahuan mengenai suatu objek tertentu. Frame memiliki *slot* yang menggambarkan atribut dan karakteristik objek, seperti terlihat pada Gambar 2.5. Frame biasanya digunakan untuk merepresentasikan pengetahuan yang berdasarkan pada karakteristik yang sudah dikenal, yang merupakan pengalaman-pengalaman.



Gambar 2.5 Frame

e. Naskah (*script*)

Naskah adalah skema representasi pengetahuan yang sama dengan frame, yaitu merepresentasikan pengetahuan berdasarkan karakteristik yang telah dikenal sebagai pengalaman-pengalaman. Perbedaannya adalah frame menggambarkan objek sedangkan naskah menggambarkan urutan peristiwa.

f. Sistem produksi

Sistem produksi merupakan salah satu bentuk representasi pengetahuan yang populer dan banyak digunakan. Sistem produksi secara umum terdiri dari komponen berikut, seperti terlihat pada Gambar 2.6:

- 1) Ruang keadaan: berisi keadaan awal, tujuan, dan kumpulan aturan.
- 2) Strategi kontrol: berguna untuk mengarahkan proses pencarian yang akan berlangsung dan mengendalikan arah eksplorasi.



Gambar 2.6 Sistem Produksi

2. Mesin inferensi (*inference engine*)

Bagian yang berisi metodologi yang akan digunakan untuk melakukan proses penalaran terhadap informasi yang ada di dalam basis pengetahuan. Mesin inferensi merupakan otak dari sebuah sistem pakar. Tiga elemen utama dari mesin inferensi, yaitu:

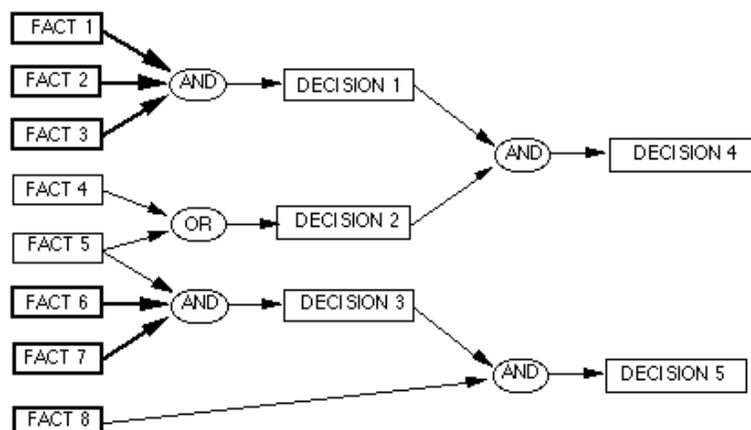
- a. *Interpreter*: mengeksekusi item agenda terpilih dengan menggunakan aturan dalam basis pengetahuan yang sesuai.
- b. *Scheduler*: menjaga kontrol menggunakan agenda.
- c. *Consistency enforcer*: memelihara tingkat konsistensi dalam merepresentasikan solusi yang bersifat darurat.

Teknik inferensi untuk memecahkan masalah dapat dilakukan dengan merangkai kaidah produksi (*chaining*). Jenis arah pencarian menurut Luger dan Stubblefield (1993, p86) yaitu:

a. *Forward chaining*

Penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis. Pemecahan masalah dimulai dari fakta masalah yang diberikan dari kumpulan langkah-langkah sah untuk berpindah *state*. Dengan menggunakan pohon (*tree*), tingkatan pohon ditelusuri untuk mencari semua kemungkinan dengan angka yang cocok. Cara tersebut terus

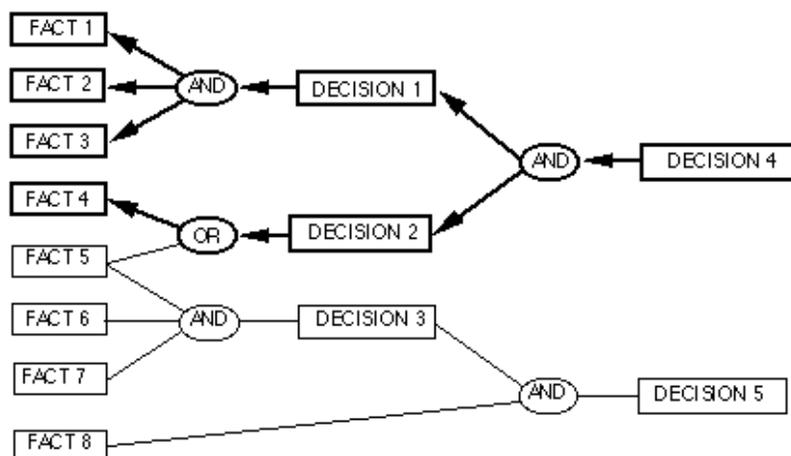
diulang hingga mendapatkan hasil yang sesuai. Lebih jelasnya dapat dilihat pada Gambar 2.7 berikut:



Gambar 2.7 Forward Chaining

b. *Backward chaining*

Pencocokan fakta dimulai dari sebelah kanan (hasil yang terlebih dahulu) dan bergerak menuju kearah fakta-fakta yang mendukung hasil tersebut. Yang pertama kali dilakukan adalah mendefinisikan dahulu keadaan akhir (hasil), kemudian menelusuri keadaan yang mungkin sampai terjadi keadaan awal (fakta). Untuk lebih jelasnya dapat dilihat pada Gambar 2.8 berikut:



Gambar 2.8 Backward Chaining

3. *User interface*

Digunakan sebagai alat untuk berkomunikasi antara program (*expert system*) dengan pengguna (*user*) dan sebaliknya.

Selain tiga komponen utama tersebut, sistem pakar juga memiliki beberapa komponen tambahan, seperti:

1. Subsistem penambah pengetahuan (*knowledge acquisition subsystem*)

Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Proses akuisisi pengetahuan diklasifikasikan dalam lima tahap (Turban, 1992, p126-128):

- a. *Identification* (identifikasi): dilakukan pengidentifikasian masalah atau karakteristiknya.
- b. *Conceptualization* (konseptualisasi): menentukan konsep dan hubungan yang sesuai antara pengetahuan dan permasalahan agar hasil yang muncul relevan.
- c. *Formulation* (perumusan): merepresentasikan pengetahuan ke dalam basis pengetahuan.
- d. *Implementation* (implementasi): penerapan pengetahuan yang dikumpulkan ke dalam program.
- e. *Testing* (pengujian): pengujian sistem dengan satu contoh permasalahan, dan kemudian dikonsultasikan ke pakar.

2. *Blackboard*

Area di dalam memori yang digunakan untuk proses merekam suatu kejadian yang sedang berlangsung, termasuk keputusan sementara yang dibuat sistem. *Blackboard* terdapat hanya dalam beberapa sistem pakar. Penggunaan

blackboard biasanya apabila terdapat beberapa pakar yang bekerjasama dalam memecahkan suatu masalah.

Terdapat tiga tipe keputusan yang dapat direkam, yaitu:

- a. Rencana: yaitu bagaimana menyelesaikan masalah.
- b. Agenda: yaitu aksi-aksi potensial yang sedang menunggu untuk dieksekusi.
- c. Solusi: yaitu hipotesis-hipotesis utama dan aksi alternatif yang dibangun sistem selama program dijalankan.

3. Subsistem penjelasan (*explanation subsystem/justifier*)

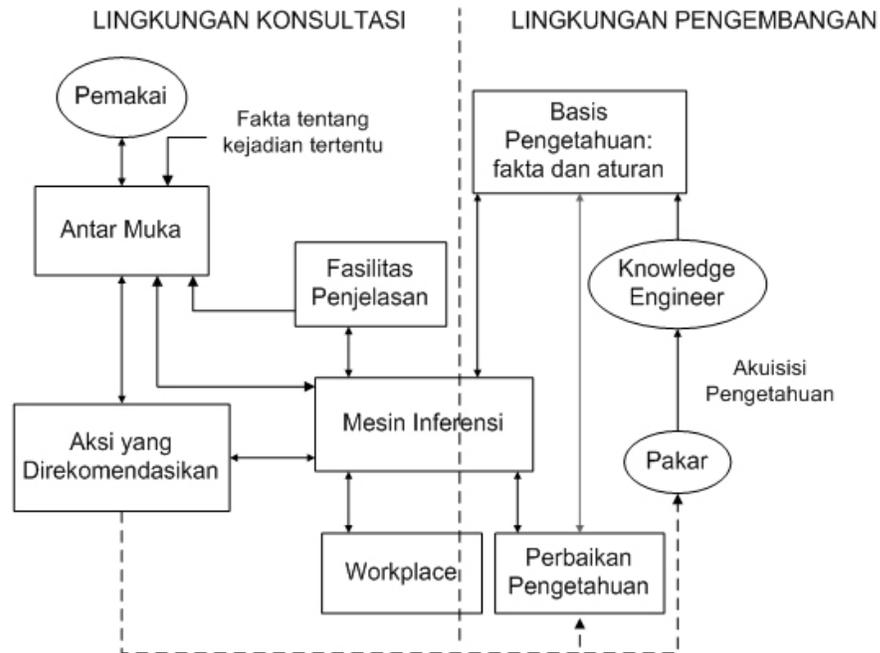
Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif dengan menggunakan pertanyaan-pertanyaan.

4. Sistem penyaring pengetahuan (*knowledge refining system*)

Digunakan untuk mengevaluasi kinerja sistem pakar tersebut dan untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan di masa yang akan datang. Sistem penyaring pengetahuan masih jarang diterapkan pada sistem pakar yang ada sekarang ini.

Untuk lebih jelasnya tentang bentuk struktur sistem pakar, dapat dilihat pada

Gambar 2.9 berikut:



Gambar 2.9 Struktur Sistem Pakar (Turban, 1995, p481)

2.2.3 Karakteristik Sistem Pakar

Berikut adalah beberapa karakteristik dari sistem pakar, dan kelebihanya dibandingkan program konvensional, adalah:

- Dapat melakukan proses simulasi dari penalaran manusia mengenai sebuah domain masalah daripada mensimulasikan domain itu sendiri. Fokusnya adalah untuk mendekati kemampuan penyelesaian masalah dari seorang pakar sehingga dapat mengambil tindakan layaknya seorang pakar, atau bahkan lebih baik.
- Mampu melakukan proses penalaran terhadap perwakilan pengetahuan manusia, selain melakukan proses perhitungan angka/mengambil data. Ada pemisahan antara pengetahuan tersebut dengan kode-kode yang melakukan penalaran.

- c. Memiliki dua metode penyelesaian masalah, yaitu Heuristik dan Aproksimasi (tidak membutuhkan data yang sempurna, dan solusi yang diturunkan sistem dapat diajukan dengan berbagai derajat kepastian yang bervariasi).

2.2.4 Perbedaan Sistem Pakar dengan Sistem Konvensional

Untuk perbedaan antara sistem pakar dan sistem konvensional dapat dilihat lebih lanjut pada Tabel 2.1 berikut:

Tabel 2.1 Perbedaan Sistem Konvensional dan Sistem Pakar

<i>Sistem Konvensional</i>	<i>Sistem Pakar</i>
Informasi dan pemrosesannya biasanya jadi satu dengan program	Basis pengetahuan merupakan bagian dari mekanisme inferensi
Biasanya tidak bisa menjelaskan mengapa suatu input data itu diperlukan, atau bagaimana output diperoleh	Penjelasan adalah bagian terpenting dari sistem pakar
Pengubahan program cukup sulit dan membosankan	Pengubahan aturan dapat dilaksanakan dengan mudah
Sistem hanya akan beroperasi jika sistem tersebut sudah lengkap	Sistem dapat beroperasi hanya dengan beberapa aturan
Eksekusi dilakukan langkah demi langkah	Eksekusi dilakukan pada seluruh basis pengetahuan
Menggunakan data	Menggunakan pengetahuan
Tujuan utamanya adalah efisiensi	Tujuan utamanya adalah efektifitas

2.2.5 Perbedaan Sistem Pakar dengan Kecerdasan Buatan Lainnya

Perbedaan sistem pakar dengan program kecerdasan buatan lainnya, beberapa di antaranya adalah:

- a. Sistem pakar menangani masalah yang bersifat *realistic complexity* (kompleksitas yang realistik) yang umumnya memerlukan banyak pakar/ahli.
- b. Sistem pakar mempertunjukkan kemampuan performa yang tinggi (*high performance*) dalam hal kecepatan dan reliabilitas.
- c. Sistem pakar harus mampu menjelaskan dan memberikan alasan solusi yang dihasilkan (*explaining and justifying solutions*) untuk meyakinkan *user* bahwa cara penalarannya adalah benar.

Selain itu, sistem pakar yang baik haruslah memenuhi beberapa ciri-ciri sebagai berikut:

- a. Memiliki fasilitas informasi yang handal,
- b. Dapat dengan mudah untuk dimodifikasi,
- c. Penggunaannya dapat dilakukan dengan menggunakan berbagai macam komputer,
- d. Mempunyai kemampuan untuk belajar beradaptasi.

2.2.6 Kelebihan dan Kelemahan dari Sistem Pakar

Banyak manfaat yang dapat diambil dengan tersedianya sistem pakar. Secara garis besar, beberapa keuntungan penggunaan sistem pakar adalah sebagai berikut:

- a. Memungkinkan orang awam dapat melakukan pekerjaan yang biasanya dikerjakan oleh seorang ahli.
- b. Dapat melakukan proses secara berulang dengan otomatis.

- c. Menyimpan pengetahuan dan keahlian dari para pakar.
- d. Mampu mengambil dan melestarikan keahlian dari para pakar.
- e. Mampu beroperasi dalam lingkungan yang kurang optimal/berbahaya.
- f. Memiliki reliabilitas.
- g. Memiliki kemampuan untuk bekerja dengan informasi yang kurang lengkap dan memiliki unsur ketidakpastian.

Selain memiliki keuntungan, sistem pakar juga memiliki kekurangan. Beberapa di antaranya adalah:

- a. Biaya yang diperlukan untuk membangun dan memelihara sistem agar selalu berada dalam kondisi yang terbaik sangat mahal.
- b. Sistem bisa menjadi sulit dikembangkan. Hal ini tentunya berkaitan erat dengan ketersediaan pakar di bidangnya.
- c. Sistem pakar tidak 100% bernilai benar.

2.2.7 Pengembangan Sistem Pakar

Secara garis besar, pengembangan sistem pakar dapat dijelaskan sebagai berikut:

- a. Programmer harus dapat mengidentifikasi masalah dan kebutuhan, menganalisa situasi dan memastikan dengan pasti apakah masalah tersebut dapat dikomputasi dan apakah sistem pakar dapat membantu penyelesaian.
- b. Programmer harus mampu menentukan masalah yang cocok, domain masalah tidak bisa terlalu luas/memiliki kompleksitas tingkat menengah dan tersedianya ahli yang cocok.
- c. Programmer juga perlu mempertimbangkan alternatif, yaitu nantinya akan diputuskan akan tetap menggunakan sistem pakar atau komputer tradisional.

- d. Programmer perlu menghitung pengembalian investasi, seperti biaya pembangunan sistem pakar, biaya pemeliharaan, dan lain sebagainya.

2.2.8 Penggunaan Aplikasi Sistem Pakar

Contoh-contoh permasalahan/pemakaian aplikasi yang biasanya menggunakan sistem pakar, di antaranya: (Jackson, 1999)

- a. Interpretasi data; contoh: interpretasi sinyal sonar.
- b. Diagnosa kerusakan; contoh: diagnosa pada kerusakan barang atau penyakit pada manusia.
- c. Analisis pada suatu bidang yang kompleks; contoh: analisa senyawa kimia.
- d. Konfigurasi objek yang kompleks; contoh: konfigurasi pada sistem komputer.
- e. Perencanaan urutan pengambilan tindakan; contoh: penggerakan robot.

2.3 Hypertext Markup Language (HTML)

HTML merupakan sebuah file teks murni yang dapat dibangun dengan menggunakan sembarang editor teks. Dokumen HTML yang ditampilkan pada browser berisi informasi atau *interface* aplikasi dalam internet. Terdapat dua cara untuk membangun sebuah dokumen HTML, yaitu dengan menggunakan editor khusus HTML atau menggunakan editor teks biasa, contohnya: notepad.

Versi terakhir HTML adalah HTML 4.01. Saat ini juga telah berkembang XHTML yang merupakan pengembangan dari HTML.

Sebuah dokumen HTML tersusun dari elemen-elemen. “Elemen” ini adalah istilah dari komponen-komponen dasar yang membentuk sebuah dokumen HTML.

Beberapa contoh elemen adalah: head, body, dan table. Elemen dapat berbentuk teks murni, atau bukan berbentuk teks, atau berbentuk keduanya.

Untuk menandai sebuah elemen pada dokumen HTML, digunakanlah <tag>. Tag umumnya berpasangan, dimana pasangannya akan memiliki tanda garis miring (/) yang menandai akhir dari elemen (<namatag>...</namatag>). Tag tidak bersifat *case sensitive* namun dianjurkan untuk menggunakan huruf kecil (*lowercase*) untuk menulis nama tag.

Beberapa jenis elemen yang terdapat dalam sebuah dokumen HTML:

- a. *Structural*: elemen ini menentukan level atau tingkatan dari sebuah teks. Contohnya, sebuah teks yang memiliki tag <h1>...</h1>, teks tersebut akan ditampilkan sebagai teks yang bersifat tebal dan besar dengan level Heading 1.
- b. *Presentation*: elemen ini akan menentukan tampilan sebuah teks dan tidak mempedulikan level dari teks tersebut. Contohnya, sebuah teks yang memiliki tag ... akan ditampilkan sebagai teks **tebal**. Elemen presentational saat ini sudah tidak disarankan untuk dipakai dan lebih baik mengatur bentuk tampilan teks dengan menggunakan *cascading style sheets* (CSS).
- c. *Hypertext*: elemen ini digunakan untuk membuat sebuah teks menunjuk ke sebuah dokumen tertentu. Contohnya, sebuah teks yang memiliki tag Google berarti teks 'Google' akan mengarah pada website yang memiliki alamat URL yaitu <http://www.google.com>.

2.4 PHP: Hypertext Preprocessor (PHP)

PHP adalah sebuah pemrograman bahasa *script server-side* yang dalam pengembangan web penggunaannya disisipkan ke dalam dokumen HTML. Penggunaan PHP memungkinkan pembuatan sebuah web yang bersifat dinamis sehingga proses *maintenance* situs web tersebut lebih mudah dan efisien.

PHP dapat digunakan oleh semua sistem operasi, seperti Linux, Unix (termasuk variannya), Microsoft Windows, MAC OS, RISC OS. Selain itu PHP mendukung berbagai web server, seperti Apache, Microsoft Internet Information Server (MIIS), Netscape, dan banyak lagi web server yang lainnya. PHP juga tidak hanya terbatas pada hasil keluaran (*output*) HTML. PHP dapat pula menghasilkan teks seperti XHTML dan file XML lainnya.

Beberapa kelebihan yang dimiliki oleh PHP dibandingkan dengan bahasa pemrograman lain adalah sebagai berikut:

- a. Dalam penggunaannya, PHP tidak melakukan proses kompilasi,
- b. Lebih mudah dari sisi pengembangan karena banyaknya milis-milis dan developer yang siap membantu,
- c. Terdapat banyak referensi sehingga mudah untuk dipahami,
- d. Dapat dijalankan secara *runtime* dengan menggunakan *console*, dan juga dapat menjalankan perintah-perintah sistem.

2.5 MySQL

MySQL adalah sebuah DBMS relasional yang bersifat *open-source*, *enterprise-level*. MySQL berbeda dengan SQL, yang merupakan query terstruktur yang dikembangkan oleh IBM. MySQL adalah program yang mengatur database, layaknya program Excel milik Microsoft. SQL adalah bahasa pemrograman yang digunakan oleh MySQL untuk menangani pekerjaan di dalam database dengan menggunakan *spreadsheets* dan *workbooks*. Program lain yang dapat digunakan untuk mengatur database di antaranya adalah *Microsoft SQL Server*, *Sybase Adaptive Server*, dan DB2.

Apa itu database? Database merupakan kumpulan file terstruktur pada sebuah sistem komputer yang terorganisasi secara efisien. File-file dalam database dapat menyimpan informasi yang dapat dimanipulasi dan dipanggil pada saat dibutuhkan (Maslakowski, 2000). Database diatur secara hierarki, dari atas ke bawah. Data disimpan dalam baris dan posisi dimana tiap baris bertemu dengan kolom disebut *field*.

MySQL bukan hanya sekedar database. MySQL merupakan sistem yang mengatur database. MySQL mengontrol akses pengguna dan bagaimana data dapat dimanipulasi.

Keunggulan-keunggulan yang dimiliki oleh MySQL:

- a. Merupakan DBMS relasional yang memiliki fitur-fitur yang lengkap,
- b. Merupakan sebuah *multithreaded-server*,
- c. Memiliki *online help system*,
- d. Bersifat *portable*,
- e. Dapat digunakan pada berbagai macam aplikasi pemrograman, antara lain Perl, TCL, dan Java,
- f. Murah.

2.6 Pemilihan Program Studi

Sebelum memasuki perguruan tinggi, para calon mahasiswa diharapkan sudah mengetahui program studi apa yang ingin dipilih dan dipelajari selama berada di universitas tersebut. Ada calon mahasiswa yang sudah dengan mantap dengan pilihan program studinya, ada pula yang belum yakin atau mereka mengetahui apa yang ingin dilakukan nanti setelah lulus kuliah namun tidak tahu program studi apa yang harus dipilih untuk mendukung rencananya tersebut.

Memilih program studi yang tepat tidak hanya tergantung dengan apa yang ingin dipelajari selama berada di bangku kuliah. Calon mahasiswa perlu memikirkan apa yang ingin dilakukan setelah lulus/ingin bekerja sebagai apa dan bagaimana. Calon mahasiswa juga perlu mengetahui kekuatan dan kekurangan mereka, apa yang disukai dan apa yang tidak, sebelum memutuskan untuk memilih satu program studi tertentu.

Calon mahasiswa memilih program studi yang dipilihnya dikarenakan berbagai alasan, misalnya program studi tersebut merupakan tren (sedang ramai dibicarakan), mengikuti pilihan teman, disuruh oleh orang tua, dan sebagainya.

Salah satu alasan yang ada adalah calon mahasiswa memilih satu program studi karena minat dan bakatnya. Namun, ketika mahasiswa tersebut lulus, menjadi pengangguran karena nyatanya mahasiswa tersebut tidak benar-benar mengetahui minat dan bakatnya dirinya sendiri ketika menempuh kuliah.

Selain itu, ketersediaan lapangan kerja yang ada juga harus diperhatikan sebelum memilih satu program studi. Terlalu banyaknya persaingan dapat menimbulkan pengangguran sehingga apa yang dipelajari oleh mahasiswa tidak dapat diterapkan.

Untuk memutuskan program studi apa yang patut dipilih, terdapat beberapa hal yang perlu diperhatikan oleh mahasiswa. Pertama, mahasiswa perlu mengetahui apa yang menjadi minatnya. Apa tipe pekerjaan yang menarik untuk mahasiswa tersebut.

Kedua, mahasiswa perlu mengetahui kemampuannya sendiri, dimana letak kelebihan dan dimana letak kekurangannya. Keterampilan yang dimiliki oleh mahasiswa tersebut juga bisa dipakai untuk memutuskan program studi apa yang cocok untuk dipilih.

Ketiga, mahasiswa perlu mengetahui apa yang dinilai dalam suatu pekerjaan. Maksudnya adalah sesuatu yang penting buat mahasiswa dalam memilih suatu pekerjaan, contohnya: menginginkan pekerjaan yang mengedepankan hal-hal yang dapat membantuk komunitas, apakah pekerjaan tersebut lebih bersifat individual atau bekerja dalam sebuah grup, tingkatan kestabilan posisi ketika bekerja, dan lain sebagainya. Tiap bidang pekerjaan memiliki tingkatan tekanan atau kestabilan yang berbeda satu dengan yang lainnya.

Keempat, perlu diketahui bagaimana mahasiswa tersebut menginginkan perjalanan karirnya nanti, dalam bidang apa ia ingin bekerja. Terdapat banyak tipe pekerjaan dan semuanya memiliki tingkat persaingan yang berbeda.

Kelima, memastikan keadaan sebenarnya yang ada. Semua pilihan-pilihan yang ada perlu dipertimbangkan. Banyak halangan yang mungkin muncul, seperti ingin menjadi seorang dokter, namun memiliki kemampuan yang minim di bidang ilmu pasti. Atau memiliki minat dalam bidang seni namun ditekan oleh keluarga untuk melanjutkan bisnis keluarga. Banyak cara untuk mengatasi halangan-halangan ini namun akan lebih baik jika dipertimbangkan dengan cermat apakah halangan-halangan tersebut benar-benar bisa dikalahkan.

Terakhir, menyempitkan semua pilihan-pilihan yang ada dan memfokuskan untuk memilih program studi yang cocok. Berdasarkan pemikiran-pemikiran tentang kelebihan dan kekurangan diri serta ketersediaan lapangan kerja dan dukungan yang ada, mahasiswa dapat mengetahui lebih baik apa program studi yang bisa dipilih.