

Merancang dan Membuat Sistem Pakar

Hersatoto Listiyono

Fakultas Teknologi Informasi, Universitas Stikubank Semarang

email : herlis@unisbank.ac.id

Abstrak : Suatu sistem pakar adalah suatu sistem komputer yang menyamai kemampuan pengambilan keputusan dari seorang pakar. Suatu emulsi jauh lebih kuat daripada suatu simulasi yang hanya membutuhkan sesuatu yang bersifat nyata dalam beberapa bidang atau hal. *Knowledge* dalam sistem pakar mungkin saja seorang ahli, atau *knowledge* yang umumnya terdapat dalam buku, majalah dan orang yang mempunyai *knowledge* tentang suatu bidang. Istilah sistem pakar, sistem basis-pengetahuan (*knowledge-base*), atau sistem pakar basis-pengetahuan (*knowledge-base*), sering digunakan dalam arti yang sama.

Kata Kunci : knowledge, User Interface Knowledge Acquisition, pelacakan

PENDAHULUAN

Pada saat ini terjadi perkembangan yang pesat dalam bidang ilmu pengetahuan dan teknologi, khususnya teknologi komputer dan komunikasi atau sering disebut dengan *era information and Communication Technology (ICT)*. Jika pada mulanya komputer digunakan hanya sekedar alat penghitung, maka saat ini komputer telah mampu menggantikan peran atau tugas-tugas rumit yang dilakukan oleh manusia, bahkan sanggup menirukan proses biologis manusia dalam pengambilan keputusan yang disebut kecerdasan buatan.

Berbeda dengan program komputer biasa, sistem kecerdasan buatan dapat digunakan untuk memecahkan masalah yang tidak terstruktur dan dimana tidak ada suatu prosedur tertentu untuk memecahkan masalah tersebut. Sedangkan definisi pengetahuan (*knowledge*) menurut *Webster's New World Dictionary of the American Language*: persepsi tentang sesuatu yang jelas dan tentu, semua yang telah dirasakan dan diterima oleh otak, serta merupakan informasi terorganisasi yang dapat diterapkan untuk penyelesaian masalah.

Dari kecerdasan buatan yang akhir-akhir ini mengalami perkembangan yang sangat pesat. Sistem ini dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan menyelesaikan suatu permasalahan baik di bidang kesehatan, bisnis, ekonomi, keuangan dan sebagainya

Sistem pakar merupakan program komputer yang mampu menyimpan pengetahuan

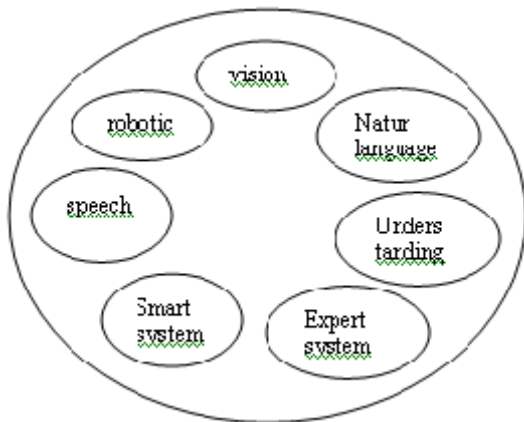
dan kaidah dari domain pakar yang khusus. Dengan bantuan sistem pakar seorang yang awam atau tidak ahli dalam suatu bidang tertentu akan dapat menjawab pertanyaan, menyelesaikan masalah dan mengambil keputusan yang biasanya dilakukan oleh seorang pakar. Sesuai dengan namanya, suatu "Sistem Pakar" akan sangat tergantung pada pengetahuan (*knowledge*) yang didapat dari pakar yang menyumbangkan keahlian dan pengalamannya.

SISTEM PAKAR SEBAGAI BAGIAN ARTIFICIAL INTELLIGENCE (AI)

Definisi yang populer dari AI adalah bahwa AI menjadikan komputer berakting dan bergaya seperti halnya para artis berakting di bioskop. Dan untuk saat ini banyak permasalahan dunia nyata yang diselesaikan dengan menggunakan AI dan banyak juga aplikasinya yang dikomersilkan (Muhammad Ahrami : 2005 : 1).

Sistem pakar adalah salah satu cabang dari AI yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah. Seorang pakar adalah orang yang mempunyai keahlian dalam bidang tertentu, yaitu pakar yang mempunyai *knowledge* atau kemampuan khusus yang orang lain tidak mengetahui atau mampu dalam bidang yang dimilikinya. Ketika sistem pakar dikembangkan pertama kali sekitar tahun 70-an sistem pakar hanya berisi *knowledge*. Namun demikian sekarang ini istilah sistem pakar sudah digunakan untuk berbagai macam sistem yang menggunakan teknologi sistem pakar itu.

Teknologi sistem pakar ini meliputi bahasa sistem pakar, program dan perangkat keras yang dirancang untuk membantu pengembangan dan pembuatan sistem pakar.



Gambar 1. Ruang lingkup kecerdasan buatan (Artificial Intelligence)

SISTEM PAKAR

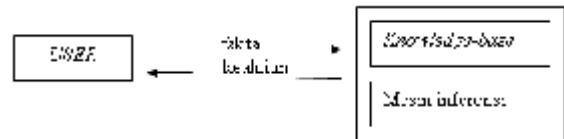
Profesor Edward Fieganbaum(1982 : 1) dari Universitas Stanford sebagai seorang pelopor awal dari teknologi sistem pakar, mendefinisikan sistem pakar sebagai "... suatu program komputer cerdas yang menggunakan *knowledge* dan prosedur inferensi untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang yang ahli untuk menyelesaikannya.

Suatu sistem pakar adalah suatu sistem komputer yang menyamai kemampuan pengambilan keputusan dari seorang pakar. Suatu emulsi jauh lebih kuat daripada suatu simulasi yang hanya membutuhkan sesuatu yang bersifat nyata dalam beberapa bidang atau hal (Muhammad Ahrami : 2005 : 1).

Knowledge dalam sistem pakar mungkin saja seorang ahli, atau *knowledge* yang umumnya terdapat dalam buku, majalah dan orang yang mempunyai *knowledge* tentang suatu bidang. Istilah sistem pakar, sistem basis-pengetahuan (*knowledge-base*), atau sistem pakar basis-pengetahuan (*knowledge-base*) , sering digunakan dalam arti yang sama.

Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Bagian dalam sistem pakar terdiri dari 2

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

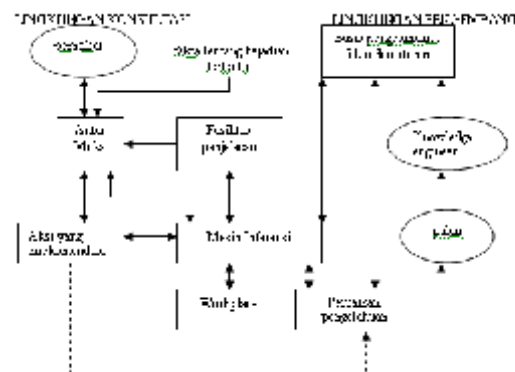


Gambar 2. konsep dasar fungsi sistem pakar

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

STRUKTUR SISTEM PAKAR

Sistem pakar disusun oleh dua bagian yaitu lingkungan pengembangan dan lingkungan konsultasi. Lingkungan pengembangan sistem pakar digunakan untuk memasukan *knowledge* pakar ke dalam lingkungan sistem pakar, sedangkan lingkungan konsultasi digunakan oleh pengguna yang bukan pakar guna memperoleh *knowledge* pakar.



Gambar 3. Arsitektur sistem pakar (sumber : turban : 1995 : 1)

Komponen-komponen yang terdapat dalam sistem pakar yaitu antarmuka pengguna,

basis pengetahuan (*knowledge-base*), akuisisi pengetahuan, mesin inferensi, *workplace*, fasilitas penjelasan, perbaikan pengetahuan.

ANTARMUKA PENGGUNA (*USER INTERFACE*)

User interface merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Sistem pakar menampilkan pertanyaan-pertanyaan yang hanya perlu dijawab oleh pengguna. Pertanyaan-pertanyaan itu harus dijawab dengan benar dan sesuai dengan masalah yang dihadapi pengguna. Antarmuka menerima jawaban dari pengguna dan selanjutnya sistem pakar mencari dan mencocokkan ke dalam aturan sehingga diperoleh suatu kesimpulan. Jadi antarmuka menerima input berupa jawaban dari pemakai dan mengubahnya kedalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menyajikan informasi dalam bentuk yang dapat dimengerti oleh pemakai. Menurut Mcloed (1995:1),

Pada bagian ini terjadi dialog antar program dan pemakai, yang memungkinkan sistem pakar menerima instruksi dan input dari pemakai, juga memberikan informasi (*output*) kepada pemakai.

BASIS PENGETAHUAN (*KNOWLEDGE-BASE*)

Basis pengetahuan tidak dapat dipisahkan dari mesin inferensi. Basis pengetahuan (*knowledge-base*) mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah. Komponen sistem pakar ini disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.

Dalam studi kasus pada sistem berbasis pengetahuan terdapat beberapa karakteristik yang dibangun untuk membantu kita dalam membentuk serangkaian prinsip-prinsip arsitekturnya. Prinsip tersebut meliputi :

a. Pengetahuan merupakan kunci kekuatan sistem pakar.

- b. Pengetahuan sering tidak pasti dan tidak lengkap.
- c. Pengetahuan sering miskin spesifikasi.
- d. Amatir menjadi ahli secara bertahap.
- e. Sistem pakar harus fleksibel.
- f. Sistem pakar harus transparan.

Sejarah peneliti dibidang AI telah menunjukan berulang kali bahwa pengetahuan adalah kunci untuk setiap sistem cerdas (*intelligence system*).

AKUISISI PENGETAHUAN (*KNOWLEDGE ACQUISITION*)

Akuisisi *knowledge* adalah komulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan kedalam program komputer. Dalam tahap ini *knowledge enginer* berusaha menyerap *knowledge* untuk selanjutnya ditransfer ke dalam basis pengetahuan (*knowledge-base*). *Knowledge* diperoleh dari pakar dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai. Menurut Turban (1998 : 1), terdapat tiga metode utama dalam akuisisi pengetahuan yaitu :

a. Wawancara

Wawancara adalah metode akuisisi yang paling banyak digunakan. Dalam metode ini melibatkan pembicaraan dengan pakar secara langsung dalam suatu wawancara. terdapat bentuk wawancara yang dapat digunakan, masing-masing bentuk wawancara tersebut mempunyai tujuan yang berbeda.

- 1) Contoh Masalah (Kasus), bentuk wawancara ini, pakar dihadapkan dengan suatu masalah nyata.
- 2) Wawancara Klasifikasi, maksud dari wawancara ini adalah untuk memperoleh wawasan pakar untuk domain permasalahan tertentu.
- 3) Wawancara Terarah, metode ini biasanya merupakan pelengkap bagi metode wawancara dengan menggunakan contoh masalah dan wawancara klasifikasi. Dalam bentuk wawancara ini pakar dan *knowledge*

engineer mendiskusikan domain dan cara menyelesaikan masalah dalam tingkat yang lebih umum dari dua metode sebelumnya.

b. Analisis Protokol

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

c. Observasi Pada Pekerjaan Pakar

Dalam metode ini, pekerjaan dalam bidang tertentu yang dilakukan pakar direkam dan diobservasi.

d. Induksi Aturan Dari Contoh

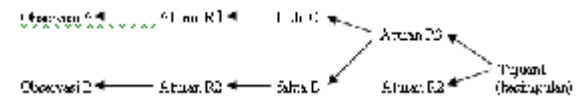
Metode ini dibatasi untuk sistem berbasis aturan. Induksi adalah suatu proses penalaran dari kasus ke umum. Suatu sistem induksi aturan diberi contoh-contoh dari suatu masalah yang hasilnya telah diketahui. Setelah diberikan beberapa contoh, sistem induksi aturan tersebut dapat membuat aturan yang benar untuk kasus-kasus contoh. selanjutnya aturan dapat digunakan untuk menilai kasus lain yang hasilnya tidak diketahui.

MESIN INFERENSI

Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan masalah. Mesin inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan (*knowledge-base*) dan dalam *workplace*, dan untuk memformulasikan kesimpulan (turban : 1995 :1). Kebanyakan sistem pakar berbasis aturan menggunakan strategi inferensi yang dinamakan modus ponens. Jika terdapat aturan “ IF A THEN B “, dan jika diketahui bahwa A benar maka dapat di simpulkan bahwa B juga benar.

Terdapat dua pendekatan untuk mengontrol inferensi dalam sistem pakar berbasis aturan, yaitu pelacakan kebelakang (*backward chaining*) dan pelacakan kedepan (*forward chaining*). Pelacakan kebelakang (*backward chaining*) adalah pendekatan yang

dimotori tujuan (*goal driven*). dalam pendekatan ini pelacakan mulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Selanjutnya proses pelacakan menggunakan premis aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya.



Gambar 4. Proses *Backward Chaining*

Sebagai contoh dari pendekatan dari metode pelacakan kebelakang (*backward chaining*) yaitu terdapat penderita yang mempunyai penyakit flu mempunyai gejala-gejala penyakit seperti pusing, bersin, dan demam yang di sebabkan oleh virus flu maka dari itu obat yang di gunakan untuk menyembuhkannya yaitu mixagrif, ultraflu.

Pelacakan ke depan (*forward chaining*) adalah pendekatan yang dimotori data (*data-driven*). dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan kedepan mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN.



Gambar 5. Proses *Forward Chaining*

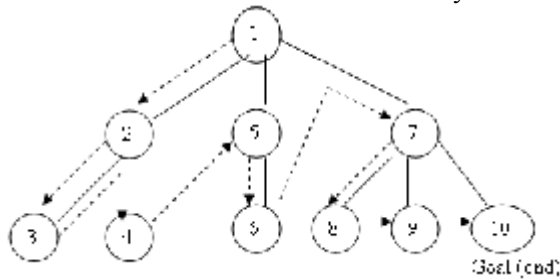
Sedangkan contoh dari pendekatan pelacakan ke depan yaitu apabila seseorang mempunyai gejala penyakit pusing, bersin dan demam yang di sebabkan oleh virus flu dan membutuhkan obat mixagrif, ultraflu maka dapat di pastikan bahwa di pastikan seseorang terkena penyakit flu.

Kedua metode inferensi tersebut dipengaruhi oleh tiga macam penelusuran, yaitu *Depth-first search*, *Breadth-first search*, dan *best-first search*.

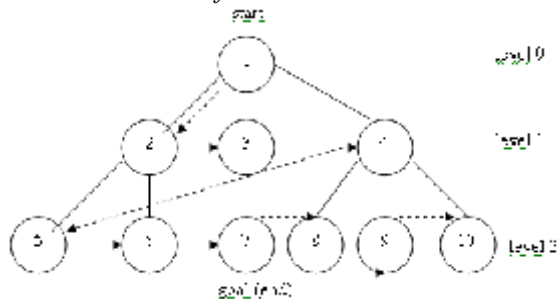
- Depth-first search*, melakukan penelusuran kaidah secara mendalam dari simpul akar

bergerak menurun ketingkat dalam yang berurutan.

- Breadth-first search*, bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ketingkat selanjutnya.
- Best-first search*, bekerja berdasarkan kombinasi kedua metode sebelumnya.



Gambar 6. Diagram teknik penelusuran dept-first-search



Gambar 7. Diagram teknik penelusuran Breadth-first search

Untuk sebuah sistem pakar yang besar, dengan jumlah rute yang relatif banyak, metode pelacakan kedepan (*forward chaining*) akan dirasakan sangat lamban dalam pengambilan kesimpulan, sehingga untuk sistem-sistem yang besar digunakan metode pelacakan kebelakang (*backward chaining*).

WORKPLACE

Workplace merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil-hasil antara dan kesimpulan yang dicapai. Ada 3 tipe keputusan yang dapat direkam, yaitu :

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

FASILITAS PENJELAS

Fasilitas penjelas adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan penalaran sistem kepada pemakai. Fasilitas penjelas dapat menjelaskan perilaku sistem pakar dengan menjawab pertanyaan-pertanyaan sebagai berikut (Turban : 1995 :1).

- Mengapa pertanyaan tertentu ditanyakan oleh sistem pakar?
- Bagaimana kesimpulan tertentu diperoleh?
- Mengapa alternatif tertentu ditolak?
- Apa rencana untuk memperoleh penyelesaian?

PERBAIKAN PENGETAHUAN

Pakar memiliki kemampuan untuk menganalisa dan meningkatkan kinerja serta kemampuan untuk belajar dan kinerjanya. Kemampuan tersebut adalah penting dalam pembelajaran komputerisasi, sehingga program akan mampu menganalisis penyebab kesuksesan dan kegagalan yang dialaminya.

REPRESENTASI PENGETAHUAN

Knowledge sering disinonimkan dengan data, fakta dan informasi. Pelajaran dari *knowledge* merupakan suatu *epistemologi* yang merupakan bagian dari ilmu filsafat yang membahas tentang asal. Hal ini berkenaan dengan sifat, struktur dan keaslian dari *knowledge*. Disamping jenis filosofi yang dikemukakan oleh Aristoteles, Plato, Descartes, Hume, Khan dan lainnya, ada dua tipe khusus yang dinamakan priori dan posteriori. Istilah priori berasal dari bahasa latin yang berarti “yang mendahului”.

Priori *knowledge* dianggap menjadi kebenaran yang universal dan tidak dapat disangkal tanpa kontradiksi. Pernyataan logika, hukum matematika dan *knowledge* yang dipengaruhi oleh anak belasan tahun merupakan contoh dari priori *knowledge*. Kebalikan dari *knowledge* yang diturunkan dari akal pikiran yang sehat, yaitu posteriori *Knowledge*. Kebenaran atau kesalahan posteriori *knowledge* dapat dibuktikan dengan menggunakan pengalaman akal sehat, seperti pernyataan “

lampu berwarna biru “. Namun demikian karena berhubungan dengan pengalaman maka boleh jadi tidak selalu bias di percaya dan posteriori *knowledge* dapat di sangkal berdasarkan *knowledge* baru tanpa memerlukan kontradiksi. *Knowledge* dapat diklasifikasikan ke dalam tiga katagori yaitu prosedural *declarative knowledge*, dan *tacit Knowledge*.

Procedural knowledge berkenaan untuk mengetahui bagaimana melakukan sesuatu. Sebagai contoh, *knowledge* tentang bagaimana mendidihkan air dalam mangkok. *Deklarative knowledge* berkenaan untuk mengetahui sesuatu itu benar atau salah. Hal ini berkenaan dengan *knowledge* yang menunjukan bentuk pernyataan deklarasi seperti “ jangan celupkan tangan anda ke dalam mangkok air yang mendidih”. *Tacit knowledge* kadang disebut juga dengan *unconscious knowledge*, karena tidak dapat di ungkapkan dengan bahasa. Sebagai contoh adalah mengetahui bagaimana memindahkan tangan anda dari dalam air yang panas. Pada suhu tinggi anda harus mengatakan bahwa menarik tangan anda cepat-cepat atau santai.

Dalam sistem komputer ANS (*Artificial Neural System*) dikaitkan dengan *tacit knowledge* karena secara normal jaringan saraf tidak secara langsung menjelaskan *knowledge*nya akan tetapi mungkin mampu jika disediakan program yang sesuai. *Knowledge* merupakan kunci utama dari sistem pakar. Analoginya dengan ekspresi klasik dari wirth adalah :

Algoritma + Struktur Data = Program.

Dan untuk sistem pakar.

Knowledge + Inferensi = Sistem Pakar.

Gambar : Tingkatan *knowledge*

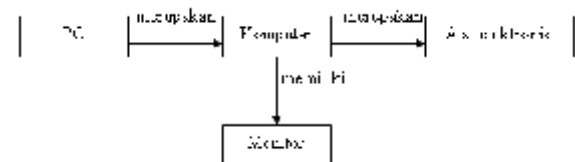
Yang ada pada level paling bawah adalah gangguan (*noise*). Gangguan (*noise*) merupakan data yang masih kabur. Level berikutnya adalah data yang merupakan hal potensial yang paling menarik. Data yang sudah diproses adalah informasi yang penting. Berikutnya adalah *knowledge* yang menggambarkan informasi yang sangat khusus. Level paling atas adalah *meta knowledge* yang merupakan *knowledge* dan keahlian. Suatu sistem pakar dapat dirancang dengan *knowledge* akan memerlukan *knowledge*

dari beberapa domain yang berbeda dan *meta knowledge* akan menentukan dasar *knowledge* yang dapat digunakan.

Selanjutnya istilah *knowledge* disebut sebagai *knowledge*. Dalam pengembangan sistem berbasis pengetahuan (*knowledge-base*), *knowledge* yang telah diekstrak dipresentasikan ke dalam bentuk yang dapat di proses oleh komputer. Beberapa model representasi pengetahuan yang penting (kusrini : 2006 : 1) adalah :

a. Jaringan Semantik (*Semantic Net*)

Konsep jaringan semantik diperkenalkan pada tahun 1968 oleh Ross Quillian. Jaringan semantik merupakan teknik representasi kecerdasan buatan klasik yang digunakan untuk informasi proposional (Giarranto dan Riley : 1994 :1). Yang dimaksud dengan informasi proposional adalah pernyataan yang mempunyai nilai benar atau salah. Informasi proposional merupakan bahasa deklaratif karena menyatakan fakta. Komponen utama yang mempresentasikan *knowledge* dalam bentuk jaringan semantik adalah simpul (*node*) dan penghubung (*link*). Simpul mempresentasikan obyek, konsep atau situasi. Simpul digambarkan dengan kotak atau lingkaran. Penghubung menghubungkan antar simpul. Penghubung dinyatakan dengan panah berarah dan diberi label untuk bmenyatakan hubungan yang dipresentasikan.



Gambar 8. Representasi jaringan semantik

Jaringan semantik pada gambar 2.9 mempresentasikan pernyataan bahwa semua komputer merupakan alat elektronik, semua PC merupakan komputer, dan semua komputer memiliki monitor. Dari pernyataan tersebut dapat diketahui bahwa semua PC memiliki monitor dan hanya sebagai alat elektronik yang memiliki monitor.

b. Object-Attribute-Value (OAV)

Objek dapat berupa bentuk fisik atau konsep. *Attribute* adalah karakteristik atau sifat dari objek tersebut. Nilai besaran / nilai / takaran spesifik dari *attribute* tersebut pada situasi tertentu, dapat berupa numerik, string, atau boolean.

Sebuah objek bias memiliki beberapa *attribute*, bias disebut OAV *multi-attribute*.

Contoh representasi pengetahuan dengan OAV ditunjukkan pada tabel 4.3

Tabel 1. Representasi Pengetahuan dengan OAV

<i>Object</i>	<i>Attribute</i>	<i>Value</i>
Mangga	Warna	Hijau, Orange
Mangga	Berbiji	Tunggal
Mangga	Rasa	Asam, Manis
Mangga	Bentuk	Oval
Pisang	Warna	Hijau, Kuning
Pisang	Bentuk	Lonjong

c. Bingkai (Frame)

Bingkai berupa ruang-ruang (*slot*) yang berisi atribut untuk mendeskripsikan *knowledge*. *Knowledge* yang termuat dalam *slot* dapat berupa kejadian, lokasi, situasi, ataupun elemen-elemen lainnya. Bingkai digunakan untuk mempresentasikan *knowledge* deklaratif (Giarranto dan Riley : 1994 :1).

Bingkai memuat deskripsi sebuah obyek dengan menggunakan tabulasi informasi yang berhubungan dengan obyek. Dengan demikian bingkai membantu menirukan cara seseorang mengorganisasikan informasi tentang sebuah obyek yang menjadi kumpulan data.

Bingkai merupakan cara yang lebih kompleks untuk menyimpan obyek dan nilai atributnya bila dibandingkan dengan jaringan semantik. Bingkai menambahkan

kecerdasan pada representasi data dan mengizinkan obyek untuk menurunkan nilai dari obyek yang lain.

Tabel 2. Bingkai Penyakit untuk Sistem Pakar penyakit

Ruang (<i>slot</i>)	Isi (<i>filler</i>)
Nama	Batuk
Gejala	a. Tenggorokan gatal b. pusing c. demam
Obat	a. Mixadin b. Laserin c. Bisolvon d. Komix

Seperti pada jaringan semantik, tidak ada standar untuk mendefinisikan sistem berberbasikan bingkai. Bingkai dapat dipandang sebagai suatu sebagai suatu struktur *record* pada bahasa tingkat tinggi atau sebuah atom dengan daftar propertinya.

PENALARAN DENGAN KETIDAKPASTIAN (*UNCERTAINTY*)

Dalam sistem pakar penalaran merupakan teknik umum dalam menyelesaikan masalah, digunakan bila suatu algoritma tidak cukup atau tidak ada algoritma yang ada solusinya dan hanya penalaran yang memungkinkan untuk solusi tersebut.

Ketidakpastian dapat dianggap sebagai suatu kekurangan informasi yang memadai untuk membuat suatu keputusan. Ketidakpastian merupakan suatu permasalahan karena mungkin menghalangi kita dalam membuat suatu keputusan yang terbaik bahkan mungkin dapat menghasilkan suatu keputusan yang buruk.

Dalam dunia medis, ketidakpastian mungkin menghalangi pemeriksaan yang terbaik untuk para pasien dan berperan dalam suatu terapi yang keliru.

Sejumlah teori yang berhubungan dengan ketidakpastian telah ditemukan, diantaranya

probabilitas Klasik, probabilitas Bayes, teori Hartley yang berdasarkan pada himpunan klasik, teori Shanon yang didasarkan pada peluang, teori Dempster-Shafer dan teori Fuzzy Zadeh.

PROBABILITAS DAN TEOREMA BAYES

a. Probabilitas

Suatu yang sudah lama sekali tetapi masih tetap sangat penting sebagai alat dalam menyelesaikan masalah AI adalah probabilitas (Farley : 1993 :1). Probabilitas merupakan suatu cara kuantitatif yang berhubungan dengan ketidakpastian yang telah ada sejak abad ke-17, ketika penjudi-penjudi Francis meminta bantuan dari para ahli matematika yang terkemuka seperti Pascal, Fermat dan lainnya.

Perjudian telah jadi sangat populer karena melibatkan uang yang sangat besar didalamnya. Pejudi menginginkan metode untuk membantu mereka dalam menghitung semua rintangan untuk menuju kemenangan.

Teori probabilitas klasik pertama kali diperkenalkan oleh Pascal dan Fermat pada tahun 1654 (parrat : 1961 :). Kemudian banyak cara yang telah dilakukan untuk mengerjakan probabilitas dan ada beberapa cabang baru dari probabilitas yang dikembangkan. Banyak aplikasi probabilitas yang telah ditunjukkan dalam ilmu pengetahuan, teknik, bisnis, ekonomi, dan bidang-bidang lainnya.

b. Teorema Bayes

Probabilitas Bayes merupakan salah satu cara untuk mengatasi ketidakpastian data dengan menggunakan formula Bayes yang dinyatakan :

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

Dimana :

$P(H|E)$: probabilitas hipotesis H jika diberikan *evidence* E.

$P(E|H)$: probabilitas munculnya *evidence* E jika diketahui hipotesis H.

$P(H)$: probabilitas hipotesis H tanpa memandang *evidence* apapun.

$P(E)$: probabilitas *evidence* E.

Dalam bidang kedokteran teorema Bayes sudah dikenal tetapi teori ini lebih

banyak diterapkan dalam logika kedokteran moderen (Cutler : 1991 :1). Teorema ini lebih banyak diterapkan pada hal-hal yang berkenaan dengan diagnosis secara statistik yang berhubungan dengan probabilitas serta kemungkinan dari penyakit dan gejala-gejala yang berkaitan.

Teorema Bayes dapat dikembangkan jika setelah dilakukan pengujian terhadap hipotesis kemudian muncul lebih dari sebuah *evidence*. Dalam hal ini maka persamaannya akan menjadi :

$$P(H|E,e) = \frac{P(H|E) \cdot P(e|E,H)}{P(e|E)}$$

Dimana :

e : *Evidence* lama.

E : *Evidence* baru.

$P(H|H,e)$: Probabilitas hipotesis H benar jika muncul *evidence* baru E dari *evidence* lama e.

$P(H|E)$: Probabilitas hipotesis H benar jika diberikan *evidence* E.

$P(e|E,H)$: Kaitan antar E jika hipotesis H benar.

$P(e|E)$: Kaitan antara e dan E tanpa memandang hipotesis apapun.

Teori Dempster-Shafer

Teori Dempster-Shafer pertama kali diperkenalkan oleh Dempster, yang melakukan percobaan ketidakpastian dengan *range probabilities* daripada sebagai probabilitas tunggal. Kemudian pada tahun 1976 Shafer mempublikasikan teori Dempster pada buku yang berjudul *Mathematical Theory of Evident* (Giarratano dan Riley, 2004)

Suatu interval (*Belief, Plausibility*) dalam teori Dempster-Shafer adalah total ukuran kekuatan *evidence* dalam mendukung suatu himpunan bagian. Jika bernilai 0 maka mengidentifikasikan bahwa tidak ada *evidence*, dan bernilai 1 menunjukkan adanya *evidence*. Menurut Giarratano dan Riley, *Belief* dapat diformulasikan sebagai berikut :

$$Bel(X) = \sum_{m \subseteq X} m(Y)$$

TAHAPAN-TAHAPAN UNTUK MEMBUAT SISTEM PAKAR

Memang diakui bahwasannya sistem pakar merupakan program komputer yang sangat rumit dibanding dengan program komputer konvensional lainnya. Meskipun demikian, tidaklah perlu kecil hati atau menyerah sebelum mulai. Sudah banyak program-program sistem pakar yang sudah berhasil dengan baik, artinya bagaimanapun sulitnya toh masih bisa juga dibuat dan dikembangkan.

Dalam pembuatan program sistem pakar ada beberapa langkah atau tahapan-tahapan yang perlu diperhatikan :

a. Identifikasi masalah dan kebutuhan.

Seperti banyak program komputer lainnya, pada dasarnya sistem pakar pun sama yaitu suatu solusi yang menjawab masalah. Agar pembuatan sistem pakar dapat dibenarkan, maka harus ada satu masalah yang harus dipecahkan atau harus dicocokkan. Untuk ini maka langkah pertama yang harus dilakukan mengkaji situasi dan memutuskan dengan pasti tentang masalah yang akan dikomputerisasi dan apakah dengan sistem pakar bisa lebih membantu atau tidak.

Dalam usaha untuk memperoleh suatu hasil yang memuaskan, sering dihadapkan kepada problema, yaitu problema waktu, produktivitas dan problema orang. Problem yang diidentifikasi harus benar-benar cocok untuk solusi sistem pakar. Didalam identifikasi dibahas tentang prosedur-prosedur seperti memahami keadaan gangguan, pengandaian blok-blok yang rusak serta membagi sebuah blok yang rusak dan menemukan bagian-bagian yang rusak serta pencarian gangguan kerusakan/kesalahan utama dan pakar yang terlibat.

b. Menentukan kesesuaian masalah.

Jika masalahnya sudah diidentifikasi dengan jelas, kemudian dilakukan pengkajian lebih mendalam untuk mengetahui apakah tepat menggunakan sistem pakar atau tidak. Hal penting yang harus diingat adalah hanya masalah tertentu yang bisa dipecahkan secara baik dengan

menggunakan sistem pakar. Misalnya diagnosis kerusakan TV, diagnosis penyakit batuk, diagnosis penyakit mata dan lain-lain.

c. Mempertimbangkan alternatif.

Apabila sudah bisa mendapatkan masalah yang dianggap cocok untuk diterapkan dalam sistem pakar, perlu adanya pengkajian terlebih dahulu tentang alternatif-alternatif lain yang lebih mudah, cepat dan sesuai dengan masalah yang ingin diselesaikan.

d. Menghitung pengembalian investasi.

Langkah berikutnya adalah menentukan apakah sistem pakar lebih menguntungkan atau tidak. Perhitungan kembali tidaknya investasi dengan jalan menganalisis biaya dan kemungkinan keuntungan. Hal ini akan membantu dalam investasi pembuatan sistem pakar dan menentukan apakah biaya yang dikeluarkan itu akan sesuai dengan hasil yang akan dicapai.

e. Menyeleksi alat pembuatan.

Alat pengembangan sistem pakar adalah paket software dan hardware yang memungkinkan dan cocok untuk memasukkan pengetahuan pakar ke dalam komputer. Yakni melalui suatu proses analisis dan desain yang kemudian dilanjutkan dengan pembuatan suatu prototipe. Hampir semua alat pengembang sistem pakar menggunakan pangkalan kaidah, beberapa diantaranya menggunakan jaringan semantik atau matrik, tetapi bisa juga lebih mahal dan hanya bisa dioperasikan dalam komputer besar.

Dalam pembuatan sistem pakar sangat disarankan untuk menggunakan software khusus untuk sistem pakar (Prolog atau Visual Prolog, Lisp, Corvid). Hal ini karena software-software tersebut mempunyai kekuatan untuk mengambil kesimpulan (jawaban) dari data-data yang ada. Seperti contoh dalam bahasa prolog tidak memerlukan prosedur (algoritma), maka sangat ideal untuk memecahkan masalah yang tidak terstruktur dan yang prosedurnya tidak diketahui. Khususnya untuk memecahkan masalah non numerik.

Selain itu pula tidak memerlukan spesifikasi dan kapasitas data yang sangat tinggi. Sebagai contoh: Sistem Pakar yang pernah dibuat oleh penulis yaitu Sistem pakar untuk diagnosis kerusakan TV, dengan jumlah 52 rule ternyata hanya membutuhkan kapasitas sebesar 14 KB untuk file data dan 3 KB untuk file mesin inferensi serta 77 KB untuk file aplikasinya. Sehingga dari kenyataan ini menunjukkan betapa sangat kecilnya kapasitas yang dibutuhkannya dan cukup dimuat dalam disket ukuran kecil. Jadi lebih praktis dan ekonomis serta dapat dengan mudah untuk diedarkan dan bermanfaat.

f. Melaksanakan rekayasa pengetahuan.

Pengembangan sistem pakar dimulai dengan merekayasa pengetahuan, yaitu bagaimana caranya memperoleh pengetahuan. Seperti kita ketahui, pengetahuan dapat diperoleh dengan berbagai cara, yaitu melalui buku-buku, artikel-artikel ilmiah atau acuan lainnya yang bisa diperoleh dengan mudah dan cepat.

Pengetahuan aktual dapat diperoleh dari individu atau seseorang yang memang ahli di bidangnya. Walaupun bisa memperoleh pengetahuan dari buku-buku, tapi toh masih tetap membutuhkan satu atau dua orang ahli yang khusus menekuni pekerjaan tersebut. Adapun cara atau teknik-teknik untuk memperoleh pengetahuan dari pakar misalnya observasi, diskusi masalah, diskripsi masalah, analisis masalah dan tata cara perbaikan.

Format atau bentuk pengetahuan akan menuntun dan mengarahkan dalam memilih skema penampilan pengetahuan yang diperlukan. Jika itu merupakan pengetahuan yang luar biasa, maka dapat digunakan untuk merepresentasikan pengetahuan dalam bentuk kaidah produksi. Untuk itu selama tahap rekayasa pengetahuan, hendaknya harus berusaha menyempurnakan banyak kaidah yang paling sesuai. Dengan demikian akan mempunyai banyak pilihan alat pengembangan yang paling tepat.

g. Merancang sistem.

Dengan menggunakan pengetahuan yang sudah didapatkan beserta alatnya yaitu software dan hardware, maka sekarang dapat dilakukan tahap merancang sistem pakar. Pertama yaitu memilih alat representasi pengetahuan (misalnya matriks), kemudian mengembangkan matriks tersebut dengan membuat diagram pohon klasifikasi yang nantinya akan membantu dalam mengorganisasi dan memahami pengetahuan itu.

Dengan menggunakan bantuan ini mulailah mengkonversi pengetahuan dalam bentuk kaidah produksi. Sebaiknya mengikuti prosedur tertentu dengan yang disarankan oleh software yang dipilih. Bila sudah selesai barulah mulai menerjemahkan kaidah ke dalam basis data dan menguji bagian yang sudah dibuat. Hal ini dimaksudkan untuk menguji konsep sebelum melanjutkan pembuatan seluruh program. Sebagai langkah terakhir adalah membuat antarmuka pemakai (user interface).

KELEBIHAN DAN KEKURANGAN EXPERT SYSTEM

Sistem pakar sekarang banyak digunakan baik pada aplikasi bisnis maupun aplikasi lainnya. Akan tetapi perlu juga diketahui bahwa seperti halnya sistem yang lainnya, selain memberikan banyak kelebihan, sistem pakar juga mempunyai beberapa kelemahan.

Kelebihan-kelebihan dari sistem pakar secara umum adalah sebagai berikut:

1. Memberikan pengambilan keputusan yang lebih baik. Karena sistem pakar memberikan jawaban yang konsisten dan logis dari waktu ke waktu. Jawaban yang diberikan logis karena alasan logikanya dapat diberikan oleh sistem pakar dalam proses konsultasi.
2. Memberikan solusi tepat waktu. Kadang kala seorang manajer membutuhkan jawaban dari pakar, tetapi pakar yang dibutuhkan tidak berada ditempat, sehingga keputusan menjadi terlambat. Dengan sistem pakar, jawaban yang dibutuhkan oleh

pengambil keputusan selalu tersedia setiap saat dibutuhkan.

3. Menyimpan pengetahuan di organisasi. Pengetahuan pakar merupakan hal yang penting dan kadang kala pengetahuan ini akan hilang jika pakar keluar atau telah pensiun dari perusahaan. Dengan sistem pakar, pengetahuan dari pakar dapat disimpan di sistem pakar dan tersedia terus selama dibutuhkan.

Kekurangan-kekurangan dari sistem pakar adalah sebagai berikut:

1. Sistem pakar hanya dapat menangani pengetahuan yang konsisten. Sistem pakar dirancang dengan aturan-aturan yang hasilnya sudah pasti dan konsisten sesuai dengan alur di diagram pohonnya. Untuk pengetahuan yang cepat berubah-ubah dari waktu ke waktu, maka knowledge base di sistem pakar harus selalu diubah (*perbarui-red*), yang tentu cukup merepotkan.
2. Sistem pakar tidak dapat menangani hal yang bersifat *judgement*. Sistem pakar memberikan hasil yang pasti, sehingga keputusan akhir pengambilan keputusan jika melibatkan kebijaksanaan dan institusi masih tetap di tangan manajemen.
3. Format *knowledge base* sistem pakar terbatas. *Knowledge base* pada sistem pakar berisi aturan-aturan (*rules*) yang ditulis dalam bentuk statemen *if-then*.

DAFTAR PUSTAKA

1. Durkin, John, *Expert Systems Design and Development*. Prentice Hall International Inc., 1994.
2. <http://rndy.wordpress.com/2007/11/15/sistem-pakar/>
3. Sadly M., <http://www.beritaipstek.com/zberita-beritaipstek-2007-09-10-Kajian-Pemanfaatan-Teknologi-Knowledge-based-Expert-System-di-dalam-Pengelolaan-Sumberdaya-Alam.shtml>)