

BAB II

LANDASAN TEORI

2.1 Kecerdasan Buatan (*Artificial Intelligence*)

Intelligence atau *intelegensia* adalah seorang yang pandai melaksanakan pengetahuan yang dimilikinya. Dengan pengertian tersebut diatas maka dapat ditarik suatu kesimpulan bahwa walaupun seorang banyak memiliki pengetahuai, tetapi bila ia tidak bisa melaksanakannya dalam praktek, maka ia tidak dapat digolongkan ke dalam *intelegensia*. Dengan kata lain, *intelegensia* adalah kemampuan manusia untuk memperoleh pengetahuai dan pandai melaksanakannya dalam praktek. Hal ini berarti kemampuan dalam berfikir dan menalar. Pada batas-batas tertentu *Artificial intelligence* (AI) memungkinkan komputer bisa menerima pengetahuan melalui input manusia dan menggunakan pengetahuannya itu melalui simulasi proses penalaran dan berfikir manusia untuk memecahkan berbagai masalah.

Bagian utama aplikasi *Artificial Intelligence* adalah pengetahuan (*knowledge*), suatu pengertian tentang beberapa wilayah subjek yang diperoleh melalui pendidikan dan pengalaman. Walaupun komputer tidak mungkin mendapat pengalaman atau belajar dan meneliti seperti manusia, tetapi ia dapat memperoleh pengetahuan yang dibutuhkannya melalui upaya yang diberikan oleh seorang pakar manusia.

Hampir semua pangkalan pengetahuan (*knowledge base*) sangat terbatas dalam arti terfokuskan kepada suatu masalah khusus. Pada saat pangkalan pengetahuan itu sudah terbentuk, maka teknik *Artificial Intelligence* bisa digunakan untuk memberi kemampuan baru kepada komputer agar bisa berfikir, menalar, dan membuat *inferensi* (mengambil keputusan berdasarkan pengalaman) dan membuat pertimbangan-pertimbangan yang didasarkan oleh fakta-fakta atau aturan-aturan dan hubungan-hubungannya yang terkandung dalam pangkalan pengetahuan tersebut.

Dengan pangkalan pengetahuan dan kemampuan untuk menarik kesimpulan melalui pengalaman (*inferensi*), komputer dapat disejajarkan sebagai

alat bantu yang bisa digunakan secara praktis dalam memecahkan masalah dan pengambilan keputusan. (Gambar II.1 menggambarkan konsep komputer yang menggunakan teknik *Artificial Intelligence* dalam suatu aplikasi).



Gambar II. 1 Penerapan Konsep AI Dalam Komputer

Dengan teknik pelacakan pangkalan pengetahuan untuk mencari fakta dan hubungannya yang relevan, komputer bisa mencapai satu atau lebih solusi alternatif pada masalah yang diberikan. Pangkalan pengetahuan komputer dan kemampuan inferensi telah meningkatkan daya guna komputer bagi manusia.

Artificial intelligence telah memberikan suatu kemampuan baru kepada komputer untuk memecahkan masalah yang lebih besar dan lebih luas, tidak hanya terbatas pada soal-soal perhitungan, penyimpanan dan pengambilan data atau pengendalian yang sederhana saja. Berdasarkan Gambar II.1 diatas dapat disimpulkan bahwa terdapat beberapa kemampuan komputer tradisional dalam mengolah data.

Tujuan kedua *Artificial Intelligence* adalah untuk bisa memahami *intelegensia* manusia. Dengan menerapkan pola berfikir manusia pada komputer, maka kita dilatih untuk belajar bagaimana cara menyimpan pengetahuan dalam otak kita dan bagaimana cara mengaplikasikannya. Terlebih dahulu kita harus mengerti betul tentang pola berfikir kita sendiri, bagaimana teknik penalarannya, dan bagaimana teknik pendekatannya dalam memecahkan suatu masalah. Dengan kata lain kita harus belajar dari diri sendiri, bagaimana cara kita belajar, dan sampai sejauh mana kelemahan dan kekuatan kita. Singkatnya kita harus mengetahui bagaimana cara kita belajar. Sehingga dengan demikian, kita bisa memperoleh pengertian yang lebih baik atas pikiran dan akan bisa mengarahkan kepada metode belajar yang lebih baik dan mampu menerapkan intelegensia kita

Inputs
(Questions, Problems, etc)
Knowledge Base

ke dalam masalah dunia nyata. Bidang AI yang khusus ditujukan kepada penelitian seperti tersebut diatas disebut sebagai *sain kognitif*.

2.2 Sistem Pakar

Sistem pakar adalah sebuah perangkat lunak komputer yang berbasis pengetahuan pakar yang menyediakan solusi-solusi dengan kualitas pakar untuk masalah-masalah tertentu yang spesifik.

Sistem pakar adalah sebuah teknik inovasi baru dalam menangkap dan memadukan pengetahuan. Kekuatannya terletak pada kemampuan memecahkan masalah-masalah praktis pada saat sang pakar berhalangan. Kemampuan sistem pakar ini karena didalamnya terdapat basis pengetahuan non formal yang sebagian besar berasal dari pengalaman, buku dari "text book". Pengetahuan ini diperoleh seorang pakar dari pengalaman bekerja selama bertahun-tahun pada bidang keahlian tertentu.

Kenyataan bahwa pengetahuan adalah sebuah kekuatan yang tidak dapat kita tolak, tetapi pengetahuan yang tidak dapat diterapkan untuk memecahkan masalah yang kita temui sehari-hari adalah percuma, yang penting adalah pemecahan dari masalah yang ditemui sehari-hari, dan sistem pakar adalah salah satu jalan untuk memecahkan masalah, secara cepat dan mudah. Saat ini beberapa paket program sistem pakar telah tersedia dipasaran, meskipun demikian sebagian besar penelitian dan pengembangan pemrograman sistem pakar bukan pada perusahaan-perusahaan pemrograman komersial, akan tetapi terdapat dilaboratorium perguruan tinggi.

Tujuan utama dari sistem pakar bukan untuk menggantikan kedudukan seorang ahli atau pakar, tetapi untuk mensyaratkan pengetahuan dan pengalaman pakar-pakar yang sangat langka itu. Singkatnya banyak masalah yang dipecahkan dan banyak ahli yang menanganinya. Sistem pakar memungkinkan orang lain dapat meningkatkan produktivitasnya, dan sederhananya dapat memperbaiki kualitas keputusan.

Orang-orang yang terlibat dalam pembuatan sistem pakar adalah perancang pengetahuan (*knowledge engineer*), pakar domain problem tertentu, (*domain expert*) dan pemakai akhir atau pemakai sistem pakar (*end user*).

Knowledge engineer merupakan pakar dalam bidang kecerdasan buatan yang bertugas memilih perangkat lunak dan perangkat keras untuk proyek pembuatan sistem pakar membantu mengambil keputusan yang dibutuhkan dari pakar domain, serta mengimplementasikan pengetahuan tersebut ke dalam basis pengetahuan yang benar dan efisien. Pakar domain menyediakan pengetahuan tentang masalah yang dihadapi, umumnya merupakan seorang pekerja yang dipakai serta mengevaluasi suatu solusi. Sedangkan pemakai sistem pakar atau pemakai akhir menentukan batasan-batasan desain utama, jika pemakai akhir tidak puas maka pengembangan sistem pakar tidak berguna.

Contoh-contoh sistem pakar yang terkenal saat ini dalam berbagai domain bidang pengetahuan tertentu antara lain seperti :

1. MYCIN yang berfungsi untuk diagnosa medical.
2. DENDRAL yang berfungsi untuk mengidentifikasi struktur molekul campuran yang tidak dikenal.
3. XCON dan EXCEL yang berfungsi untuk melayani order langganan sistem komputer ke dalam sistem spesifikasi yang lengkap. Selain contoh tersebut diatas masih banyak lagi contoh lain sesuai dengan domain bidang pengetahuan masing-masing seperti PROSPECTOR, EL, SOPHIE, DELTA, FOLIO dan Lain-lain.

2.3 Klasifikasi Sistem Pakar

Berdasarkan kegunaannya, sistem pakar dapat diklasifikasikan menjadi enam jenis yaitu :

2.3.1 Diagnosis

Diagnosis adalah suatu tindakan atau perilaku yang menggunakan bantuan suatu alat dan sistem untuk mempelajari atau mengamati sesuatu hal yang tidak tepat, tidak teratur, sehingga menghasilkan suatu informasi dan membuat inferensi kemungkinan penyebab timbulnya ketidakberesan.

Diagnosis sistem pakar biasanya digunakan untuk merekomendasikan obat untuk orang sakit, kerusakan mesin, kerusakan rangkaian elektronik, dan sebagainya. Prinsipnya adalah menemukan apa masalah atau kerusakan yang

terjadi. Sistem pakar diagnosis merupakan jenis sistem pakar yang paling populer saat ini.

Biasanya sistem pakar diagnosis menggunakan pohon keputusan (*decision tree*) sebagai representasi pengetahuannya. Hal lain dari sistem pakar diagnosis ini adalah basis pengetahuannya bertambah besar secara *eksponensial* dengan kompleksnya permasalahan.

2.3.2 Pengajaran

Sistem pakar ini digunakan untuk mengajar, mulai dari murid sekolah dasar sampai mahasiswa perguruan tinggi. Kelebihan sistem pakar dapat digunakan untuk membuat diagnosa apa penyebab kekurangan dari seorang siswa, kemudian memberikan cara untuk memperbaikinya.

2.3.3 Interpretasi

Sistem pakar interpretasi ini digunakan untuk menganalisa data yang tidak lengkap, tidak teratur, dan data yang kontradiktif misalnya untuk menginterpretasi citra.

2.3.4 Prediksi

Keunggulan dari seorang pakar adalah kemampuannya memprediksi ke depan. Contoh yang mudah kita temui, bagaimana seorang pakar meteorologi memprediksi cuaca besok berdasarkan data-data sebelumnya. Penggunaan sistem pakar prediksi misalnya untuk peramalan cuaca, penentuan masa tanam, dan sebagainya.

2.3.5 Perencanaan

Perencanaan sistem pakar sangat luas mulai dari perencanaan mesin-mesin sampai manajemen bisnis. Penggunaan sistem pakar jenis ini dapat menghemat biaya, waktu, dan material. Contoh penggunaan antara lain yaitu sistem konfigurasi, komputer, tata letak sirkuit.

2.3.6 Kontrol

Sistem kontrol ini digunakan untuk melakukan pengontrolan terhadap kegiatan yang membutuhkan presisi waktu yang tinggi, Misalnya pada industri-industri berteknologi tinggi.

2.4 Sifat-sifat Sistem Pakar

Sistem pakar merupakan program-program praktis yang menggunakan strategi heuristik yang dikembangkan oleh manusia untuk memecahkan problematika-problematika yang spesifik. Karena keheuristikanya itu dan sifatnya yang berdasarkan pada pengetahuan maka umumnya sistem pakar bersifat :

1. Terbuka untuk diperiksa, baik dalam menampilkan langkah-langkah antara (*intermediate step*) maupun dalam menjawab pertanyaan-pertanyaan tentang proses tertentu.
2. Mudah dimodifikasi baik dengan menambah maupun menghapus suatu pengetahuan yang berbasis pengetahuannya.
3. Heuristik dalam menggunakan pengetahuan (yang sering kali tidak sempurna) untuk memperoleh solusi.

Terdapat beberapa alasan mengapa sebuah sistem pakar bersifat terbuka terhadap pemeriksaan antara lain yaitu :

- a. Jika seorang manusia yang pakar seperti dokter atau insinyur menerima rekomendasi dari komputer, mereka harus terpuaskan oleh jawaban yang diberikan.
- b. Bila solusi terbuka terhadap pemeriksaan dapat dievaluasi setiap aspek dan keputusan yang diambil selama proses mendapatkan solusi memungkinkan adanya penambahan sejumlah informasi atau aturan baru untuk mengembangkan kinerjanya. Dalam hal ini informasi mempunyai peranan penting dalam pembaharuan basis pengetahuan.

2.5 Karakteristik Sistem Pakar

Suatu sistem pakar yang ideal memiliki beberapa karakteristik-karakteristik sebagai berikut :

1. Terbatas pada domain keahlian tertentu.

2. Memiliki kemampuan mengolah data yang mengandung ketidakpastian.
3. Dapat mengemukakan rangkaian alasan-alasan yang diberikan dengan cara yang mudah dipahami.
4. Berdasarkan kaidah dan rule tertentu.
5. Dirancang untuk bisa dikembangkan secara bertahap.
6. Pengetahuan dan mekanisme inferensi jelas terpisah.
7. Hasil yang dicapai lebih bersifat memberikan anjuran (advice).
8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pemakai.

2.6 Perangkat Utama Dalam Sistem Pakar

Dalam hal ini terdapat beberapa pihak atau perangkat yang terlibat dalam kegiatan pembuatan sistem pakar antara lain sumber ahli yang merupakan pakar pada domain pengetahuan tertentu, pembuat sistem pakar, alat bantu (tool) pengembangan sistem pakar.

2.6.1 Sumber Ahli

Salah satu kebutuhan yang sangat kritis dan mutlak harus ada untuk masalah sistem pakar berbasis pengetahuan adalah adanya sumber ahli yang akan membantu dalam mendapatkan pengetahuan terbaik yang berasal dari seorang atau banyak ahli pada satu domain bidang pengetahuan tertentu. Sumber ahli tersebut diperlukan dengan alasan-alasan sebagai berikut :

1. Dasar pengetahuan akan lebih komplis dan akurat dari seorang sumber, daripada yang bukan ahli. Banyak pengetahuan yang kompleks dan tujuannya hampir tidak begitu jelas bagi seorang yang bukan ahli.
2. Dasar pengetahuan yang besar mungkin dibangun lebih cepat sedangkan yang bukan ahli tidak membahas secara mendalam dalam mengembangkan setiap objek dan aturan dalam basis pengetahuan.

Untuk memperoleh pengetahuan dari seorang atau lebih ahli dapat dilakukan dengan wawancara dalam hal ini bisa dilakukan beberapa kali hingga pengetahuan yang dibutuhkan sudah dianggap cukup. Proses rekayasa pengetahuan ini memerlukan waktu yang cukup panjang.

Proses merekayasa pengetahuan sebenarnya merupakan proses pembuatan sistem pakar yang intinya merupakan suatu interaksi antara sistem yang dibangun dengan satu atau beberapa dalam domain bidang pengetahuan tertentu.

Melalui wawancara, knowledge engineer harus merumuskan semua kemungkinan solusi. Hal ini untuk menentukan output sistem pakar berbasis pengetahuan yang diminta user. Pekerjaan menyimpulkan atau mempersingkat pengetahuan agar bisa membangun basis pengetahuan dengan menggunakan kaidah if-then. Frame atau skema representasi pengetahuan.

2.6.2 Pembuat Sistem Pakar

Pembuat sistem pakar merupakan orang yang mempunyai latar belakang ilmu komputer atau informatika dan mengenal prinsip-prinsip kecerdasan buatan. Dimana orang tersebut mengetahui dan memahami cara pembuat sistem pakar. Dalam hal ini pembuat sistem pakar mewawancarai para pakar yang menjadi sumber ahli seperti telah diuraikan di atas.

Ada beberapa komponen pengetahuan yang merupakan sumber yang bisa diambil sebagai bahan untuk membangun suatu pengetahuan buatan, yaitu secara umum dapat dikelompokkan sebagai berikut :

1. Fakta, yaitu suatu pernyataan yang menghubungkan beberapa elemen “kebenaran” sesuai dengan domain subjek.
2. Aturan Prosedural, yaitu aturan yang sudah didefinisikan dengan jelas dan berupa suatu urutan langkah dasar serta relasi, relatif terhadap domain.
3. Aturan Heuristik, yaitu aturan umum yang dibuat berdasarkan intuisi atau aturan yang menentukan procedural yang harus dilakukan jika aturan procedural tidak tersedia.

Format (bentuk) pengetahuan akan menurun dan mengarahkan kedalam memilih skema penampilan pengetahuan dalam bentuk kaidah produksi. Untuk itu selama terhadap rekayasa pengetahuan harus berusaha menyempurnakan banyak kaidah yang paling sesuai.

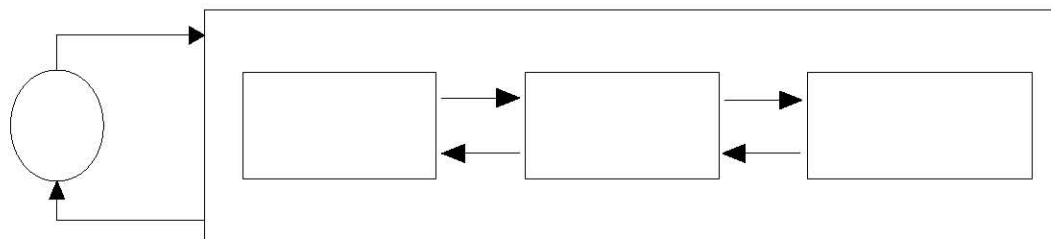
2.6.3 Alat Bantu Pengembangan Sistem Pakar

Dalam hal ini alat bantu pengembangan sistem pakar yang berupa bahasa pemrograman dapat digolongkan kedalam 3 (tiga) jenis yaitu :

1. Bahasa pemrograman umum seperti PROLOG dan LISP, kedua bahasa ini memang dirancang untuk aplikasi *Artificial Intelligence* dan dapat digunakan untuk membuat sistem pakar.
2. Bahasa pemrograman spesialis seperti *Shell* merupakan paket khusus program *Artificial Intelligence*. *Shell expert* sistem atau disebut juga generator adalah paket *software* yang khusus dibuat untuk membantu pembuatan sistem pakar. *Shell* menyediakan kerangka kerja dasar dimana data atau pengetahuan dapat dimasukkan dan dimanipulasi dengan cara yang sudah ditentukan terlebih dahulu. Hampir semua *shell* dan *generator expert* sistem menggunakan skema representasi format kaidah if-then karena kaidah ini sederhana, luwes dan sangat populer.
3. Bahasa pemrograman konvensional seperti BASIC, FORTRAN, C, PASCAL, Bahasa Assembly. Sesuai dengan perkembangan zaman, sistem pakar dapat juga dibangun dengan menggunakan pemrograman berorientasi objek seperti bahasa Delphi dan Visual Basic.

2.7 Komponen Pembuat Sistem Pakar

Sistem pakar ada tiga komponen yang saling berkaitan antara lain yaitu : Basis Pengetahuan (*knowledge base*), Mesin Inferensi (*inference engine*), dan Antarmuka Pemakai (*user interface*).



Gambar II.2 Diagram Sistem Pakar

2.7.1 Basis Pengetahuan

Basis pengetahuan adalah inti dari program sistem pakar dimana basis pengetahuan ini merupakan representasi pengetahuan dari seorang pakar. Basis pengetahuan tersusun atas fakta yang berupa informasi tentang objek dan kaidah (rule) yang merupakan informasi tentang tatacara bagaimana membangkitkan fakta baru dari fakta yang sudah diketahui atau fakta lama.

2.7.2 Mesin Inferensi

Mesin inferensi selain mengontrol aliran dan tahapan inferensi juga merupakan bagian dari sistem pakar yang mendeduksi fakta-fakta baru dari fakta-fakta yang telah ada dengan mempergunakan kaidah-kaidah yang ada. Dimana proses ini menyangkut penjadwalan dan unifikasi. Pada pengontrolan ini mesin inferensi menentukan kaidah mana yang akan diuji terlebih dahulu dan apa yang dilakukan seandainya suatu kaidah sukses atau gagal. Mesin inferensi mengambil fakta yang ada dari basis kaidah statis dan memori yang bekerja atau basis data dinamis kemudian menggunakannya untuk menguji kaidah-kaidah selama proses unifikasi. Ketika kaidah sukses, kaidah tersebut ditambahkan ke memori yang bekerja. Pada beberapa kasus, mungkin terdapat lebih dari satu fakta yang dapat mengunifikasi kaidah yang ada. Oleh karenanya mesin inferensi harus menentukan kaidah mana yang harus dipenuhi.

Adapun mesin inferensi terdiri atas dua komponen yaitu :

2.7.2.1 Strategi Inferensi

Strategi Inferensi adalah berkaitan dengan cara yang digunakan untuk menarik suatu kesimpulan atau suatu keputusan. Dalam strategi ini yang sering digunakan adalah aturan logika yang menyatakan bahwa jika diketahui B bernilai benar maka akan terdapat aturan yang berbunyi, if B then A, dan dapat disimpulkan bahwa A bernilai benar.

Penggunaan aturan logika terdapat beberapa hal yang dapat kita pelajari. Pertama bahwa aturan logika sangat sederhana penalarannya sehingga dapat dengan mudah dimengerti. Kedua, terdapatnya implikasi yang juga sah namun tidak dapat kita lakukan. Contoh suatu logika menyatakan bahwa jika suatu nilai A bernilai salah maka dapat disimpulkan bahwa nilai B juga bernilai salah.

Dengan demikian dapat diambil suatu kesimpulan bahwa aturan logika adalah pembentukan fakta baru dari suatu aturan yang telah diketahui. Adapun penjelasan mengenai metoda-metoda tersebut adalah sebagai berikut :

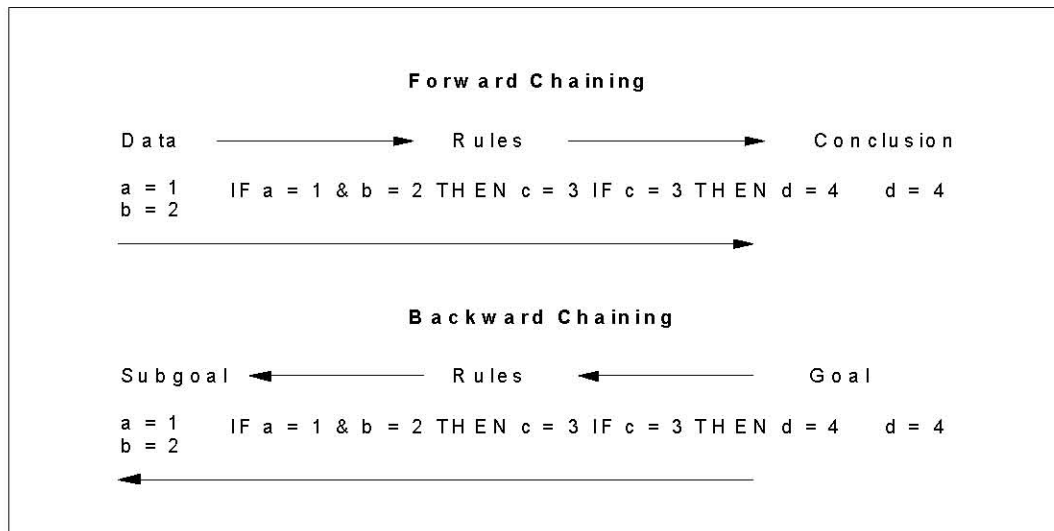
a. Metoda Rangkaian Mundur (*Backward Chaining*)

Pada metoda ini penelusuran dimulai dari hipotesa kemudian barulah mencari informasi untuk memenuhi hipotesa tersebut.

b. Metoda Rangkaian Maju (*Forward Chaining*)

Penelusuran dimulai dengan mencari informasi-informasi kemudian barulah untuk menyimpulkan atau mencari hipotesa berdasarkan informasi yang ada, dalam hal ini fakta-fakta atau aturan-aturan haruslah bernilai benar.

Adapun perbedaan untuk masing-masing metoda tersebut dapat ditunjukkan pada gambar dibawah ini.



Gambar II. 3 Perbedaan antara Forward Chaining dan Backward Chaining

2.7.2.2 Strategi Kontrol

Strategi kontrol adalah berkaitan dengan teknik penelusuran yang di gunakan untuk mencapai keadaan tujuan. Teknik yang biasa digunakan untuk memecahkan masalah dalam sistem pakar adalah telusur mundur dan telusur maju namun kadang juga menggunakan pencarian dengan sistem BFS (*Breadth First Search*) dan sistem DFS (*Depth First Search*). Penggunaan ketiga metoda tersebut

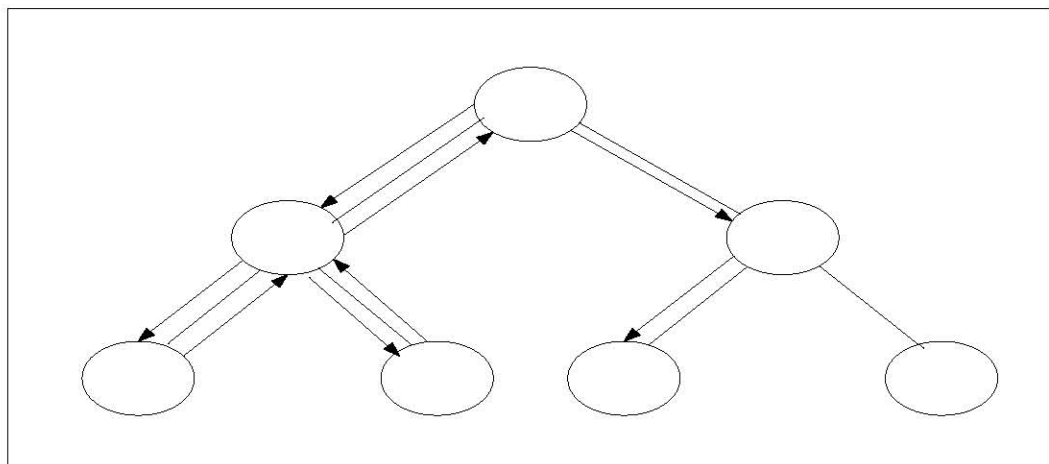
dapat digunakan secara bersama dan mengatasi suatu masalah.

Berikut ini adalah penjelasan mengenai metoda-metoda tersebut antara lain :

1. Metode DFS (*Depth First Search*)

Metoda DFS merupakan teknik penyelesaian problema dengan menelusuri setiap lintasan yang mungkin sampai kedalaman maksimal untuk mencapai suatu konklusi atau tujuan (*goal*) sebelum mencoba lintasan yang lain atau dengan kata lain metoda ini mencari informasi dengan detail terlebih dahulu baru mencari konklusi atau goal.

Sebagai gambaran dari metoda tersebut berikut ini :



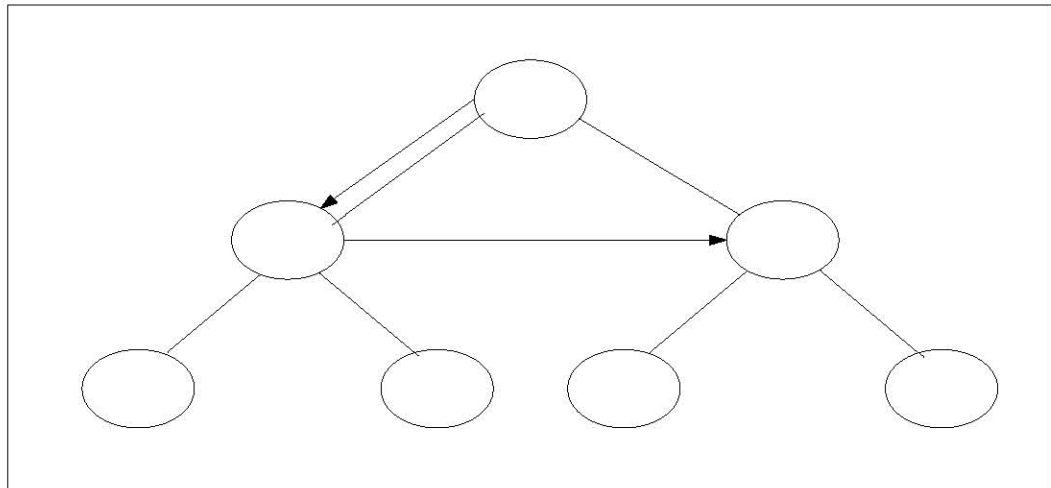
Gambar II. 4 Metoda DFS

Metoda ini akan menelusuri dengan urutan sebagai berikut : A - B - D - B - E - B - A - C - F (*goal*), dalam pencarian ini kita telusuri dari cabang kiri sampai kedalaman tertentu atau sampai goal.

2. Metode BFS (*Breath First Search*)

Metoda BFS adalah melakukan pencarian dengan menelusuri setiap simpul pada level yang sama terlebih dahulu, kemudian turun ke level yang berikutnya sampai ditemukannya suatu goal.

Contoh pencarian dari BFS adalah sebagai berikut :



Gambar II. 5 Metoda BFS

Metoda akan menelusuri dengan urutan sebagai berikut : A - B - C - (goal). Tetapi kebanyakan dalam sistem pakar menggunakan metoda pencarian *Depth First* yaitu dengan menggali sedalam mungkin dan dengan mengikuti rantai aturan yang mengarahkan pertanyaan-pertanyaan yang diberikan oleh sistem.

3. Metode Hill-Climbing

Teknik pencarian dengan menggunakan metode *Hill-Climbing* merupakan teknik pencarian yang sedikit lebih pintar dari DFS dan BFS, karena dalam metoda *Hill-climbing* telah ditambahkan suatu teknik heuristic. Heuristic adalah teknik atau aturan yang menjamin bahwa pencarian yang dilakukan dalam arah yang benar. Dalam metoda *Hill-Climbing* secara formal mencari simpul berikutnya yang notabene jaraknya terdekat ke simpul asal dan pada akhirnya merupakan jarak yang terdekat ke tujuan.

Ketiga metoda tersebut diatas mengalami kelelahan (*exhaustive*) karena sistem mengevaluasi seluruh aturan yang ada sehingga pencarian memakan waktu lama juga memakan memori yang banyak. Tetapi dengan adanya strategi kontrol masalah tersebut dapat diatasi yaitu dengan mengkombinasikan antara telusur mundur, DFS dan lain-lain.

2.7.3 Model Antarmuka (Interface)

Antarmuka (interface) merupakan komponen dalam sistem pakar yang dapat berkomunikasi dengan pengguna. Komunikasi yang dilakukan oleh suatu antarmuka adalah komunikasi yang bersifat timbal balik atau dua arah. Pada sistem pakar yang sederhana pemakai harus dapat menerangkan gambaran dari permasalahannya, oleh sistem pakar kemudian akan memberikan jawaban atau tanggapan yang berupa saran-saran atau kesimpulan.

Dalam antarmuka dari sistem harus ada atau mempunyai suatu fungsi tambahan, dimungkinkan pemakai ingin sistem dapat menerangkan proses pengambilan keputusan, bagaimana cara yang dilakukan oleh sistem untuk sampai pada keputusan tersebut.

2.8 Keuntungan Sistem Pakar

Beberapa keuntungan penerapan sistem pakar adalah sebagai berikut :

- a. Membuat seorang yang awam bekerja seperti layaknya seorang pakar.
- b. Meningkatkan produktifitas akibat meningkatnya kualitas hasil pekerjaan, meningkatnya kualitas pekerjaan ini disebabkan meningkatnya efisiensi kerja.
- c. Menghemat waktu kerja.
- d. Menyederhanakan pekerjaan.
- e. Pengolahan berulang-ulang secara otomatis.
- f. Merupakan arsip yang terpecah dari sebuah keahlian, sehingga bagi pemakai sistem pakar akan seolah-olah berkonsultasi langsung dengan sang pakar, meskipun mungkin sang pakar telah meninggal.
- g. Memperluas jangkauan, dari keahlian seorang pakar. Dimana sebuah sistem pakar yang telah disahkan, akan sama saja artinya dengan seorang pakar yang tersedia dalam jumlah besar (dapat diperbanyak dengan kemampuan yang persis sama). Dapat diperoleh dan dipakai dimana saja.

2.9 Representasi Pengetahuan

Hampir semua *Artificial Intelligence* terbuat dari dua bagian pokok, yaitu pangkalan pengetahuan dan mekanisme inferensi. Pangkalan pengetahuan berisi fakta-fakta tentang objek dalam domain yang ditentukan dan saling berhubungan satu sama lain. Seluruh program AI difokuskan pada domain tertentu. Pangkalan pengetahuan juga bisa berisi pikiran, teori, prosedur praktis dan saling hubungan. Pangkalan pengetahuan membentuk sumber sistem intelegensia dan digunakan oleh mekanisme inferensi untuk melakukan penalaran dan menarik kesimpulan.

Mekanisme inferensi adalah suatu rangkaian prosedur yang digunakan untuk menguji pangkalan pengetahuan dengan cara yang sistematik pada saat menjawab pertanyaan, memecahkan persoalan atau membuat keputusan dalam suatu domain yang telah ditentukan. Cara komputer “berpikir” tentang subjek domain, mekanisme inferensi melakukan pelacakan melalui pangkalan pengetahuan untuk mencari jawaban dan solusinya.

2.9.1 Skema Representasi Pengetahuan

Langkah pertama dalam membuat program *Artificial Intelligence* adalah membangun pangkalan pengetahuan. Agar komputer bisa bertingkah laku seperti seorang intelegensia, maka ia harus mempunyai pengetahuan terlebih dahulu tentang suatu domain tertentu. Sebenarnya pengetahuan itu sudah ada. Apakah ia masih berada dalam otak seorang atau beberapa orang pakar, atau mungkin masih dalam bentuk buku, artikel, memo dan prosedur, yang penting semuanya harus terkodifikasi.

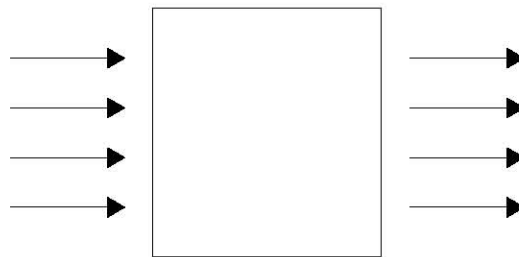
Sebenarnya berbagai representasi skema pengetahuan itu sudah dikembangkan sejak bertahun-tahun yang lalu. Representasi skema ini mempunyai dua karakteristik umum. Pertama, bisa diprogram dengan bahasa komputer yang ada dan disimpan dalam memori. Kedua, dirancang agar fakta-fakta dan pengetahuan lainnya yang terkandung didalamnya bisa digunakan untuk penalaran. Penalaran pengetahuan berisi data struktur yang bisa dimanipulasi oleh sistem inferensi, yang merupakan bagian utama program AI. Sistem inferensi menggunakan teknik pelacakan dan penyocokan pola (*pattern matching*) dalam pangkalan pengetahuan agar bisa menjawab pertanyaan, menarik kesimpulan atau tugas-tugas lainnya yang bersifat intelegensi. Skema representasi pengetahuan

secara umum telah dikategorikan sebagai suatu prosedural. Skema representasi prosedural berhubungan dengan aksi dan prosedur.

Kunci keberhasilan setiap pembuatan program AI terletak pada pemilihan skema representasi pengetahuan yang paling baik dan paling tepat, dan yang paling penting lagi yang sesuai dengan domain pengetahuan dan masalah yang akan dipecahkan. Pemilihan ini akan tergantung kepada rekayasa pengetahuan yang dihasilkan oleh pengalaman yang luas dalam merancang perangkat lunak AI.

2.9.2 Logika

Bentuk representasi pengetahuan yang paling tua adalah logika, yaitu suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah dan prosedur yang membantu proses penalaran. Logika dianggap sebagai bagian filsafat. Pengembangan dan perbaikan proses tersebut umumnya terjadi pada zaman Yunani kuno.



Gambar II. 6 Menggunakan Logika Menuju Penalaran

Gambar II.6 melukiskan bentuk umum proses logika. Pertama, memberikan informasi, membuat statement atau mencatat observasi, bentuk ini merupakan input proses logika dan disebut premis (suatu hal-ihwal yang dianggap benar). Premis ini digunakan oleh proses logika untuk membuat output yang terdiri dari konklusi dan inferensi. Dengan proses tersebut, fakta yang akan diketahui benar akan digunakan untuk memperoleh fakta baru yang juga harus benar.

Dalam hal ini ada dua bentuk dasar penalaran : Deduktif dan Induktif. Kedua bentuk ini digunakan dalam logika untuk membuat inferensi yang diambil dari premis.

2.9.3 Penalaran Deduktif

Jika untuk memperoleh inferensi tertentu kita menggunakan premis umum (kebenaran umum yang dianggap benar), maka proses ini disebut penalaran deduktif atau deduksi. Penalaran ini berangkat dari prinsip umum menuju konklusi khusus.

Proses deduksi umumnya dimulai dari suatu pernyataan premis dan inferensi. Umumnya terdiri dari tiga bagian : **Premis Mayor**, **Premis Minor**, dan **Konklusi**. Untuk maksud penalaran deduktif hampir setiap masalah dan argument dapat diletakkan pada bentuk ini, Misalnya :

Premis mayor : Jika hujan turun saya tidak akan lari pagi.

Premis minor : Pagi ini hujan turun.

Konklusi : Oleh karena itu pagi ini saya tidak akan lari pagi

Agar bisa menggunakan penalaran deduktif, pada umumnya kita harus memformat masalahnya terlebih dahulu. Jika format itu sudah dibuat, maka konklusi harus benar jika premisnya benar. Sekali lagi, seluruh ide dalam soal ini adalah tidak lain terkecuali untuk mengembangkan pengetahuan baru dari pengetahuan yang sudah ada dan diberikan sebelumnya.

2.9.4 Penalaran Induktif

Penalaran induktif adalah kebalikan dari deduktif, yaitu bila deduktif dimulai dari masalah umum menuju ke masalah atau konklusi khusus sedang induktif dimulai dari masalah khusus ke masalah umum, atau dengan kata lain, penalaran induktif menggunakan sejumlah fakta atau premis yang mantap untuk menarik kesimpulan umum. Sebagai contoh dibawah ini akan mencoba menjelaskan proses induktif, pernyataan hanya digunakan untuk mengekspresikan masalah.

Premis : Dioda yang salah menyebabkan peralatan elektronik rusak.

Premis : Transistor rusak menyebabkan elektronik rusak.

Premis : Sirkuit terpadu (IC) rusak menyebabkan peralatan elektronik tidak berfungsi.

Konklusi : Peralatan semi-konduktor rusak, merupakan penyebab utama rusaknya peralatan elektronik.

Sesuatu hal yang sangat menarik tentang penalaran induktif, ialah bahwa konklusi ini tidak pernah final atau mutlak. Konklusi dapat berubah jika fakta baru sudah ditemukan. Jika semua fakta yang mungkin, tidak dimasukkan ke dalam salah satu premis, maka akan selalu timbul ketidakpastian didalam konklusi. Akibatnya, hasil proses penalaran induktif akan selalu mengandung beberapa ukuran ketidakpastian. Walaupun demikian, jika lebih banyak fakta dan premis yang digunakan dalam proses penalarannya, maka ketidakpastian konklusi itu akan berkurang. Semakin banyak pengetahuan yang di miliki semakin banyak inferensi konklusi akan terjadi.

2.10 Metoda Representasi Pengetahuan

Ada berbagai metode representasi pengetahuan yang biasa digunakan yaitu : Metode Kalkulus Predikat, Bingkai (frame), Jaringan Semantic (semantic network), Metode Kaidah Produksi, dan Representasi Logika.

2.10.1 Metoda Kalkulus Predikat

Kalkulus predikat merupakan cara sederhana untuk merepresentasikan pengetahuan secara deklaratif. Dalam kalkulus predikat, pernyataan deklaratif dibagi atas dua bagian yaitu, predikat dan argumen.

Contoh :

Baju disimpan di lemari

Dapat ditulis sebagai berikut :

Disimpan_di(lemari, baju)

Dimana

Disimpan = Predikat

Lemari, baju = Argument

Dalam kalkulus predikat, argument dapat juga berupa variable :

Misalnya :

Budi mengenal Mariana

Bila, Budi = x dan Mariana = y

Maka, bentuk predikat kalkulus adalah :

Mencintai (X,Y)

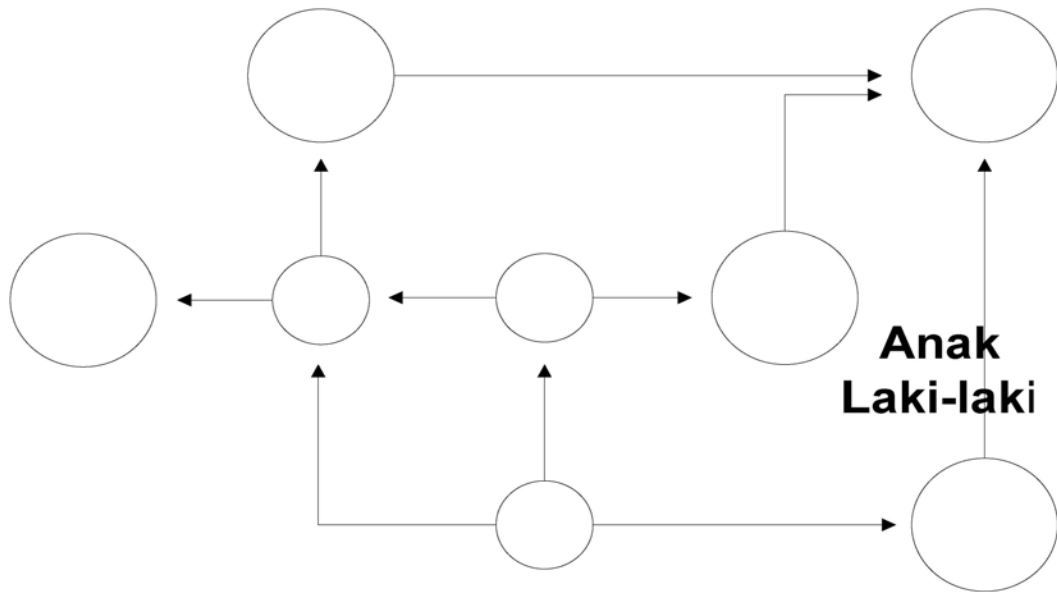
2.10.2 Bingkai

Bingkai adalah blok-blok atau potongan-potongan yang berisi pengetahuan melalui objek-objek khusus, kejadian, lokasi, situasi ataupun elemen-elemen lainnya dengan ukuran yang relatif besar. Blok-blok ini menggambarkan objek-objek tersebut secara lebih rinci. Detail diberikan dalam bentuk rak (slot) yang menggambarkan berbagai atribut dan karakteristik daripada objek tersebut.

Sebuah rak dapat berisi nilai default yaitu nilai yang sudah melekat menjadi ciri dari suatu objek, sebuah rak juga dapat mempunyai nilai yang disebut “*procedural attachment*”, yaitu suatu nilai yang besarnya relatif, misalnya rak *akselerasi* mesin mempunyai nilai 0 – 60 km/jam dalam waktu 4 detik, nilai ini akan berbeda, misalnya waktu dipersingkat menjadi 2 detik, sehingga nilai *akselerasi* mesin relatif terhadap waktu.

2.10.3 Jaringan Semantik

Jaringan semantik merupakan cara representasi pengetahuan yang paling tua dan paling mudah. Cara ini menggunakan gambaran secara grafis dari pengetahuan yang memperlihatkan hubungan hirarkis dan hubungan antara objek-objek dinyatakan oleh garis penghubung berlabel. Contoh jaringan semantik yang mudah ditemukan adalah struktur kepegawaian dari suatu perusahaan atau garis keturunan keluarga.



Gambar II.7 Contoh Jaringan Semantika adalah

2.10.4 Kaidah Produksi

Metode kaidah produksi biasanya dituliskan dalam bentuk jika-maka (if-then). Kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian, yaitu bagian premise (jika) dan bagian konklusi (maka). Apabila bagian premise dipenuhi maka bagian konklusi juga akan bernilai benar.

Sebuah kaidah terdiri dari klausa-klausa. Sebuah klausa mirip sebuah kalimat dengan subyek, kata kerja dan obyek yang menyatakan suatu fakta. Ada sebuah klausa premise dan sebuah klausa konklusi pada setiap kaidah. Suatu kaidah juga dapat terdiri atas beberapa premise dan lebih dari satu konklusi. Antara premise dan konklusi dapat dihubungkan dengan “or” atau “and”.

Contoh :

Jika : hari hujan

Maka : saya tidak jadi pergi

Jika : saya lulus

Dan : saya diterima di perguruan tinggi

Maka : - saya akan beli baju baru

- saya akan beli cd player baru

2.10.5 Representasi Logika

Pengetahuan prosedural atau logika dapat direpresentasikan dalam bentuk predikat logika seperti yang digunakan dalam bahasa PROLOG. Misalnya pernyataan : $a_1, a_2, a_3, \dots, a_n \Rightarrow B$ dapat dipandang sebagai prosedur yang digunakan untuk menghasilkan keadaan yang memenuhi B. pernyataan tersebut dapat direpresentasikan sebagai berikut :

B : - $a_1, a_2, a_3, \dots, a_n$

2.11 Memilih Teknik Representasi Pengetahuan

Dalam memilih teknik representasi pengetahuan terdapat empat kriteria, yaitu :

1. Kemampuan representasi, artinya teknik yang dipilih harus mampu merepresentasikan semua jenis pengetahuan yang akan dimasukkan ke dalam sistem pakar.
2. Kemudahan dalam penalaran, artinya teknik yang dipilih harus mudah diproses untuk memperoleh kesimpulan.
3. Efisiensi proses akuisisi, artinya teknik yang dipilih harus dapat membantu pemindahan pengetahuan dari sumber-sumber pengetahuan kedalam komputer.
4. Efisiensi proses penalaran, artinya teknik yang dipilih harus dapat diproses dengan efisien untuk mencapai kesimpulan.
5. Modularity, artinya sejumlah mana bagian-bagian pengetahuan bisa disimpan secara independent satu sama lain.

2.12 Akuisisi Pengetahuan

Akuisisi pengetahuan merupakan suatu hal yang tak kalah penting dalam membuat suatu sistem pakar. Tahap akuisisi pengetahuan biasanya dikerjakan oleh seorang *knowledge engineer*. *Knowledge engineer* adalah orang yang memiliki latar belakang pengetahuan tentang komputer dan mengerti cara-cara dalam mengembangkan sistem pakar. Pada tahap ini seorang *knowledge engineer* berusaha menyerap pengetahuan yang diperoleh dari seorang pakar atau lebih pakar, dilengkapi dengan buku jurnal ilmiah dan sebagainya. Pengetahuan yang

diperoleh harus lengkap mungkin, sebab hal ini akan sangat berpengaruh pada kemampuan sistem pakar yang akan dibangun.

Pengetahuan yang didapat bisa diperoleh melalui wawancara atau berdialog dengan seorang atau beberapa orang nara sumber. Tapi ada beberapa hal yang harus diperhatikan selama wawancara :

1. Pertanyaan yang diajukan harus spesifik agar penjelasan yang diberikan nara sumber dapat terarah dan terinci sehingga seluruh informasi yang berhubungan dengan pertanyaan tersebut dapat diperoleh.
2. Memberikan kesempatan pada seorang nara sumber untuk menjawab dan menjelaskan permasalahan dengan cara sendiri.
3. Jangan memotong penjelasan dan hindari fakta-fakta yang menyebabkan nara sumber ragu terhadap informasi yang diberikan.
4. Menggunakan alat perekam untuk menghindari kehilangan informasi dari hasil wawancara.
5. Memperhatikan cara nara sumber dalam memanfaatkan pengetahuan untuk menyelesaikan masalah. Dalam hal ini tentunya harus diperhatikan tentang fakta, teori, heuristik ataupun jenis informasi lainnya.

Teknik-teknik lain yang bisa dipergunakan untuk memperoleh pengetahuan dari pakar di antaranya :

1. Observasi, dengan melihat langsung pakar menyelesaikan masalahnya di lapangan.
2. Diskusi Masalah, menggali data, pengetahuan dan prosedur yang dibutuhkan untuk menyelesaikan masalah dari pakar.
3. Deskripsi Masalah, pakar mendeskripsikan masalah pada setiap kategori solusi dalam domain permasalahan.
4. Analisa Permasalahan, memberikan beberapa persoalan kepada pakar untuk memberikan rangkaian pengalamannya.
5. Tata cara Perbaikan, pakar memberikan beberapa masalah untuk di selesaikan oleh knowledge engineer, dan pakar memperbaiki cara penyelesaian tersebut dari hasil wawancara.
6. Tata cara Pengujian, pakar mengevaluasi, mengkritik prototipe kaidah atau aturan dan struktur pengendalian dari sistem yang dibangun.

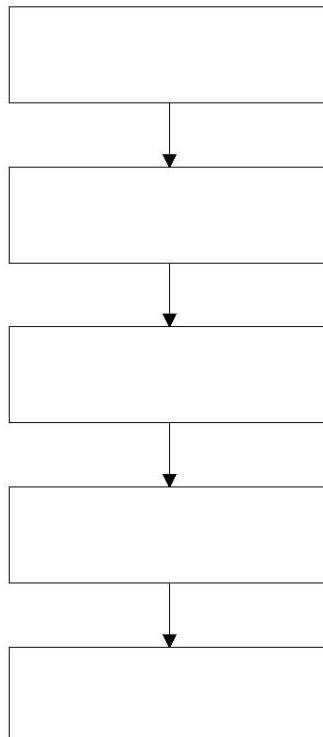
7. Tata cara Validasi, knowledge engineer membentuk prototipe sistem pakar berdasarkan hasil penyelesaian masalah yang diajukan oleh pakar lain.

2.13 Pengembangan Sistem Pakar

Pengembangan suatu sistem pakar ditentukan pada dua fase utama, pertama adalah fase yang termasuk ke dalam identifikasi dan konseptualisasi masalah. Identifikasi mencakup penyeleksian, perolehan buku-buku atau pakar, sumber-sumber pengetahuan dan sumber-sumber lain yang relevan dengan domain permasalahan.

Konseptualisasi akan menentukan konsep dari hasil identifikasi yang akan dikembangkan menjadi sistem pakar. Fase kedua adalah formalisasi, implementasi dan pengetesan. Pada fase ini mulai dibuat arsitektur sistem dan implementasi sistem secara bertahap.

Adapun pengembangan sistem pakar ini terdiri dari lima tahapan yang saling berkesinambungan, yaitu : identifikasi, konseptualisasi, formalisasi, implementasi, dan pengujian. Tahapan dan pengembangan sistem pakar tersebut dapat dilihat pada gambar berikut :



Gambar II.8 Tahapan Pengembangan Sistem Pakar

2.13.1 Identifikasi

Merupakan tahapan untuk mengidentifikasi permasalahan yang akan dikaji, sebaliknya permasalahan yang dikaji sempit tetapi kompleks (mendalam). Pada tahap ini knowledge engineer dapat menentukan masalah secara umum dan memilih sumber-sumber yang digunakan, berdiskusi dengan para pakar untuk menentukan batasan masalah yang dikaji tahapan ini jika dijabarkan secara rinci adalah sebagai berikut adalah :

1. Survey Masalah

Bertujuan untuk mencari pokok-pokok permasalahan yang kira-kira dapat diterima, pada tahap ini kriteria yang diberikan jangan terlalu berat.

2. Pemilihan Kandidat

Tujuan dari tahap ini adalah mempersempit daftar aplikasi yang ditunjukkan pada tahap sebelumnya. Untuk melakukan seleksi dengan membandingkan semua kandidat dengan kriteria semua pilihan berikut ini beberapa contoh kriteria pemilihan yang digunakan :

- a. Apakah tugas yang akan dilakukan memerlukan kemampuan seorang pakar.
- b. Apakah pakar untuk permasalahan tersebut jarang.
- c. Apakah ada keyakinan bahwa solusi menggunakan program konvensional tidak sesuai.
- d. Apakah solusi masalah dapat dihasilkan benar-benar diperlukan.
- e. Apakah solusi tersebut akan tetap berguna untuk beberapa tahun mendatang.

3. Kemungkinan Penerapan Domain

Tahap ini berfungsi untuk melakukan analisa terhadap domain untuk aplikasi sistem pakar.

4. Keberadaan Pakar

Dalam hal ini pakar harus mampu menangani masalah-masalah yang ada pada domain dan mampu memberikan penalaran yang jelas dalam memecahkan masalah.

5. Perhitungan Keuntungan dan Kerugian

Pada tahap ini harus mampu melakukan analisa sejumlah mana keuntungan akan diperoleh dari adanya sistem pakar yang dibuat.

2.13.2 Konseptualisasi

Merupakan tahapan dimana knowledge engineer dan pakar menentukan konsep yang akan dikembangkan menjadi sistem pakar. Dari konsep ini dirinci seluruh unsur yang terlibat dan dikaji hubungan antar unsur serta mekanisme pengendalian yang diperlukan untuk mencapai solusi.

2.13.3 Formalisasi

Merupakan tahap yang menghubungkan antara unsur-unsur yang digambarkan dalam bentuk yang bisa digunakan oleh pakar. Struktur data, teknik inferensi dan alat bantu lainnya termasuk yang ditentukan dalam hal ini.

2.13.4 Implementasi

Tahap ini di masukkan setelah formalisasi hubungan antara unsur-unsur telah ditentukan secara lengkap dan alat membangun yang dipilih telah sesuai. Pada tahap ini knowledge engineer menerjemahkan bentuk hubungan antara unsur kedalam bahasa komputer.

2.13.5 Pengujian

Merupakan tahapan pengujian terhadap sistem pakar yang telah di bangun. Perbaikan selalu dilakukan pada setiap tahapan agar diperoleh sistem yang lengkap dan akurat dalam mengambil keputusan.

2.14 Sekilas Microsoft Visual Basic 6.0

Microsoft Visual Basic adalah bahasa pemrograman yang bekerja dalam lingkup Microsoft Windows. Microsoft Visual Basic bekerja dengan menggunakan objek-objek sebagai komponen pemrogramannya. Setiap objek digambarkan pada layar dan melakukan pengaturan properti terhadap objek yang digambarkan.

Beberapa kemampuan atau manfaat dari Visual Basic di antaranya seperti :

- a. Untuk membuat program aplikasi berbasis Windows.
- b. Untuk membuat objek-objek pembantu program seperti control Active X, file Help, aplikasi internet, dan sebagainya.
- c. Menguji program (*debugging*) dan menghasilkan program akhir berakhiran EXE yang bersifat *executable*, atau dapat langsung dijalankan.

Beberapa dukungan yang diberikan oleh Microsoft Visual Basic adalah

1. Visual Basic memungkinkan programnya untuk mengembangkan antar muka pemakai secara mudah dan tepat.
2. Visual Basic merupakan produk pengembangan dari bahasa basic, sebuah bahasa yang sangat populer terutama dalam lingkungan akademis, meskipun lingkungan pengembangan yang sepenuhnya grafis dan bahasa pemrograman dari Visual Basic dengan intepreter basic awal, keluwesan dan kesederhanaan basic yang semula tetap, sampai batas-batas tertentu.
3. Visual Basic mampu menghasilkan file EXE yang berdiri sendiri, kecuali jika aplikasi yang dibuat memerlukan pengaksesan terhadap basis data yang di maksud dengan berdiri sendiri adalah file yang sudah dikompilasi tersebut tidak lagi memerlukan dinamic link libraries (DLL).
4. Visual Basic menyediakan banyak objek kontrol yang berhubungan dengan pemrograman basis data, sehingga dapat mempermudah dan mempersingkat pemrograman itu sendiri.
5. Visual Basic menyediakan database desktop.

Berikut ini merupakan langkah-langkat pembuatan program dengan Microsoft Visual Basic for Windows adalah :

1. Membuat tampilan antarmuka program. Pada tahap awal, tampilan tidak perlu dibuat sempurna, karena tampilan ini dapat diubah-ubah pada tahap akhir pembuatan program.
2. Membuat kode program. Kode Visual Basic akan diletakan pada kontrol atau form yang akan menggunakan kode tersebut. Kode menjadi milik sebuah kontrol atau form akan dijalankan jika terdapat kejadian terhadap

kontrol atau form tersebut. Misalnya ketika sebuah tombol dan aplikasi ini ditekan, maka dikirim pesan kejadian penekanan tombol dan aplikasi akan memberikan tanggapan dengan memanggil subrutin penekanan tombol yang akan menyebabkan dijalankan kode-kode yang terdapat pada subrutin tersebut.

3. Mengkompilasikan program. Tahap akhir dari pembuatan program adalah mengkompilasikan program sehingga menjadi program yang berdiri sendiri dan dapat dijalankan dalam lingkungan windows tanpa bantuan Visual Basic.