BAB II

TINJAUAN PUSTAKA

2.1 *Artificial Intelligence*

2.1.1 Sejarah Singkat

Kecerdasan Buatan (*artificial intelligence*) merupakan inovasi baru di bidang ilmu pengetahuan. Mulai ada sejak muncul komputer modern, yakni pada 1940 dan 1950. Kemampuan mesin elektronika baru menyimpan sejumlah besar info, memproses dengan kecepatan sangat tinggi menandingi kemampuan manusia. Ilmu pengetahuan komputer ini khusus ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer. Pada sistem ini memperlihatkan sifat-sifat khas yang dihubungkan dengan kecerdasan dalam kelakuan yang sepenuhnya dapat menirukan beberapa fungsi otak manusia, seperti pengertian bahasa, pengetahuan, pemikiran, pemecahan, dan masalah.

Pentingnya kecerdasan buatan menjadi nyata bagi negara-negara yang berperan sejak tahun 1970. Para pemimpin negara yang mengakui potensialnya kecerdasan buatan mengharap mendapat persetujuan jangka panjang untuk sumber-sumber yang memerlukan dana intensif. Jepang adalah yang pertama kali melakukan itu. Negara ini mengembangkan program yang sangat berambisi dalam penelitian kecerdasan buatan. Sebagai bidang ilmu pengetahuan komputer, kecerdasan buatan sebenarnya sudah mulai diselidiki pada 1930-an dan 1940-an. Pada saat itu, banyak cendekiawan mengembangkan ide-ide baru mengenai komputasi.

Logika matematika menjadi bidang aktif dari penyelidikan kecerdasan buatan, karena sistem logika deduktif telah berhasil diimplementasikan dalam program-program komputer. Seorang ahli matematika bernama Alan Turing, yang memiliki sumbangan besar dalam pengembangan teori kemampuan penghitungan (*computability*), mengusulkan tes untuk melihat bisa atau tidaknya mesin memberikan respon terhadap seangkaian pertanyaan (agar mesin dapat dikatakan cerdas). Uji yang dilakukan adalah dengan mengukur kinerja (*performance*) mesin cerdas. Uji Alan Turing menjadi dasar bagi banyak strategi yang digunakan dengan menilai program-program kecerdasan buatan.

Pada awalnya, kecerdasan buatan hanya ada di universitas-universitas dan laboratorium penelitian, serta hanya sedikit produk yang dihasilkan dan dikembangkan. Menjelang akhir 1970-an dan 1980-an, mulai dikembangkan secara penuh dan hasilnya berangsur-angsur dipublikasikan di khalayak umum. Permasalahan di dalam kecerdasan buatan akan selalu bertambah dan berkembang seiring dengan laju perkembangan zaman menuju arah globalisasi dalam setiap aspek kehidupan manusia, yang membawa persoalan-persoalan yang semakin beragam pula.

Program kecerdasan buatan lebih sederhana dalam pengoperasiannya, sehingga banyak membantu pemakai. Program konvensional dijalankan secara prosedural dan kaku, rangkaian tahap solusinya sudah didefinisikan secara tepat oleh pemrogramnya. Sebaliknya, pada program kecerdasan buatan untuk mendapatkan solusi yang memuaskan dilakukan pendekatan trial and error, mirip seperti apa yang dilakukan oleh manusia.

2.1.2 Pengertian *Artificial Intelligence*

Adapun beberapa definisi *Artificial Intelligence* adalah sebagai berikut :

Definisi *artificial intelligent* menurut Jog [1] : ”*Artificial Intelligence didefinisikan sebagai suatu mesin atau alat pintar (biasanya adalah komputer) yang dapat melakukan suatu tugas yang bilamana tugas tersebut dilakukan oleh manusia akan dibutuhkan suatu kepintaran untuk melakukannya”.*

Definisi *artificial intelligent* menurut Kus [2] : “*Kecerdasan buatan* atau *artificial intelligence merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia”.*

Definisi *artificial intelligent* menurut Sar [3] : “*Kecerdasan buatan (artificial intelligence adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berprilaku cerdas seperti manusia”.*

Kecerdasan buatan dilihat dari berbagai sudut pandang adalah sebagai berikut :

1. Sudut pandang Kecerdasan (Intelligence)

   Kecerdasan buatan adalah bagaimana membuat mesin yang “cerdas” dan dapat melakukan hal-hal yang sebelumnya dapat dilakukan oleh manusia.

2. Sudut pandang Penelitian

Studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dilakukan oleh manusia.

Domain penelitian adalah sebagai berikut :

a. Mundane task

   i. Persepsi (vision and speech)

   ii. Bahasa alami (understanding, generation and translation)

   iii. Pemikiran yang bersifat commonsense

   iv. Robot control

b. Format task

   i. Permainan atau games

   ii. Matematika (geometri, logika, kalkulus, integral, pembuktian)

c. Expert task

   i. Analisis financial

   ii. Analisis medical

   iii. Analisis ilmu pengetahuan

   iv. Rekayasa (desain, pencarian, kegagalan, perencanaan, manufaktur)

3. Sudut pandang Bisnis

   Kumpulan peralatan yang sangat powerful dan metodologis dalam menyelesaikan masalah-masalah bisnis.

4. Sudut pandang Pemrograman (*Programming*)

   Kecerdasan buatan termasuk didalamnya adalah studi tentang pemrograman simbolik, pemecahan masalah, proses pencarian (*search*).

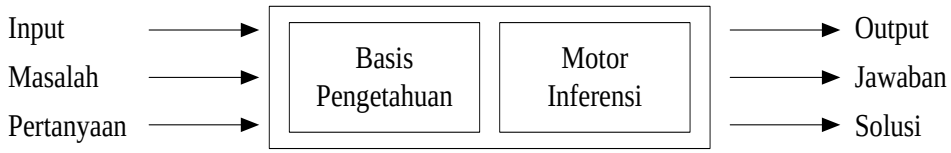
2.1.3 Tujuan *Artificial Intelligance*

Tujuan dari kecerdasan buatan menurut Winston dan Prendergast :

1. Membuat mesin menjadi lebih pintar (tujuan utama)
2. Memahami apa itu kecerdasan (tujuan ilmiah)
3. Membuat mesin lebih bermanfaat (tujuan *entrepreneurial*)

Dua bagian utama yang dibutuhkan untuk aplikasi kecerdasarn buatan (Gambar 2.1 Proses Motor Inferensi) adalah :

a. Basis Pengetahuan (*Khowledge Base*) berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
b. Motor Inferensi (*Inference Engine*) adalah kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar 2. 1 Proses Motor Inferensi

2.1.4 Lingkup Utama Artificial Intelligence

1. Sistem Pakar (*Expert System*)

Komputer sebagai sarana untuk menyimpan pengetahuan para pakar sehingga komputer memiliki keahlian menyelesaikan permasalahan dengan meniru keahlian yang dimiliki pakar.

2. Pengolahan bahasa alami (*Natural Language Processing*)

User dapat berkomunikasi dengan computer menggunakan bahasa sehari-hari, missal bahasa inggris, bahasa Indonesia atau pun bahasa daerah lainnya.

3. Pengenalan ucapan (*Speech recognition*)

Manusia dapat berkomunikasi dengan komputer menggunakan suara.

4. Robotika dan sistem sensor

Contohnya sistem sensor, seperti sistem vision, sistem *tactile* dan sistem pemrosesan sinyal jika dikombinasikan dengan *Artificial Intelligence*, dapat dikategorikan ke dalam suatu sistem yang luas yang disebut sistem robotika.

5. Computer Vision

Menginterpretasikan gambar atau objek tampak melalui komputer.

6. Intelligent Computer Aided Instruction

Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.

7. Game Playing

Permainan dilakukan dengan menggunakan sekumpulan aturan, pencarian ruang, teknik untuk menentukan *alternative* dalam menyimak problema ruang merupakan sesuatu yang rumit, teknik tersebut disebut dengan *Heuristic* dan permainan merupakan bidang yang menarik dalam studi *heuristic*.

2.2 Sistem Pakar

2.2.1 Pengertian Sistem Pakar

Ketika hendak membuat suatu keputusan yang komplek atau memecahkan masalah, seringkali kita meminta nasehat atau berkonsultasi dengan seorang pakar atau ahli. Seorang **pakar** adalah seseorang yang mempunyai pengetahuan dan pengalaman spesifik dalam suatu bidang; misalnya pakar komputer, pakar uji tak merusak, pakar politik dan lain-lain. Semakin tidak terstruktur situasinya, semakin mengkhusus (dan mahal) konsultasi yang dibutuhkan.

Sistem Pakar (*Expert System*) adalah usaha untuk menirukan seorang pakar. Biasanya Sistem Pakar berupa perangkat lunak pengambil keputusan yang

mampu mencapai tingkat performa yang sebanding seorang pakar dalam bidang problem yang khusus dan sempit. Ide dasarnya adalah: kepakaran ditransfer dari seorang pakar (atau sumber kepakaran yang lain) ke komputer, pengetahuan yang ada disimpan dalam komputer, dan pengguna dapat berkonsultasi pada komputer itu untuk suatu nasehat, lalu komputer dapat mengambil inferensi (menyimpulkan, mendeduksi, dll.) seperti layaknya seorang pakar, kemudian menjelaskannya ke pengguna tersebut, bila perlu dengan alasan-alasannya. Sistem Pakar malahan terkadang lebih baik unjuk kerjanya daripada seorang pakar manusia!

Kepakaran (*expertise*) adalah pengetahuan yang ekstensif (meluas) dan spesifik yang diperoleh melalui rangkaian pelatihan, membaca, dan pengalaman. Pengetahuan membuat pakar dapat mengambil keputusan secara lebih baik dan lebih cepat daripada non-pakar dalam memecahkan problem yang kompleks. Kepakaran mempunyai sifat berjenjang, pakar top memiliki pengetahuan lebih banyak daripada pakar yunior.

Tujuan Sistem Pakar adalah untuk mentransfer kepakaran dari seorang pakar ke komputer, kemudian ke orang lain (yang bukan pakar). Proses ini tercakup dalam rekayasa pengetahuan (*knowledge engineering*) yang akan dibahas kemudian.

Sistem pakar yang terkenal :

MYCIN

Dibuat oleh Edward Shortlife of Standford University tahun 70-an. Sistem pakar medical yang bisa mendiagnosa penyakit infeksi dan merekomendasi pengobatan. MYCIN membantu dokter mengidentifikasi pasien yang menderita penyakit. Dokter duduk di depan komputer dan memasukkan data pasien: umur, riwayat kesehatan, hasil laboratorium dan informasi terkait lainnya. Dengan informasi ini ditambah pengetahuan yang sudah ada dalam komputer, MYCIN mendiagnosa selanjutnya merekomendasi obat dan dosis yang harus dimakan. MYCIN sebagai penasehat medis, tidak dimaksudkan untuk mengantikan kedudukan seorang dokter. Tetapi membantu dokter yang belum berpengalaman dalam penyakit tertentu. Juga untuk membantu dokter dalam mengkonfirmasi diagnosa dan terapi yang diberikan kepada pasien apakah sesuai dengan diagnosa dan terapi yang ada dalam basis pengetahuan yang sudah dimasukkan ke dalam MYCIN, karena MYCIN dirancang oleh dokter-dokter yang ahli di bidang penyakit tersebut. Kesimpulan : sistem pakar seperti MYCIN bisa digunakan sebagai bahan pembanding dalam pengambilan solusi dan pemecahan masalah. Keputusan terakhir atas pengobatan tersebut tetap menjadi tanggung jawab dokter.

2.2.2 Tujuan Sistem Pakar

Tujuan dari sistem pakar adalah untuk memindahkan kemampuan (*transferring expertise*) dari seorang ahli atau sumber keahlian yang lain ke dalam

komputer dan kemudian memindahkannya dari komputer kepada pemakai yang tidak ahli (bukan pakar). Proses ini meliputi empat aktivitas yaitu:

1. Akuisi pengetahuan (*knowledge acquisition*) yaitu kegiatan mencari dan mengumpulkan pengetahuan dari para ahli atau sumber keahlian yang lain.
2. Representasi pengetahuan (*knowledge representation*) adalah kegiatan menyimpan dan mengatur penyimpanan pengetahuan yang diperoleh dalam komputer. Pengetahuan berupa fakta dan aturan disimpan dalam komputer sebagai sebuah komponen yang disebut basis pengetahuan.
3. Inferensi pengetahuan (*knowledge inferencing*) adalah kegiatan melakukan inferensi berdasarkan pengetahuan yang telah disimpan didalam komputer.
4. Pemindahan pengetahuan (*knowledge transfer*) adalah kegiatan pemindahan pengetahuan dari komputer ke pemakai yang tidak ahli.

2.2.3 Bentuk Sistem Pakar

1. Mandiri merupakan sistem pakar yang murni berdiri sendiri, tidak digabung dengan perangkat lunak lain, bisa dijalankan pada komputer pribadi dan mainframe.
2. Terkait atau tergabung merupakan sistem pakar hanya bagian dari program yang lebih besar. Program tersebut biasanya menggunakan teknik algoritma konvensional tapi bisa mengakses sistem pakar yang ditempatkan sebagai subrutin, yang bisa dimanfaatkan setiap kali dibutuhkan.

3. Terhubung adalah sistem pakar yang berhubungan dengan software lain. Misalnya spreadsheet, DBMS, program grafik. Pada saat proses inferensi, sistem pakar bisa mengakses data dalam spreadsheet atau DBMS atau program grafik bisa dipanggil untuk menayangkan output visual.
4. Sistem Mengabdi Merupakan bagian dari komputer khusus yang diabdikan kepada fungsi tunggal. Sistem tersebut bisa membantu analisa data radar dalam pesawat tempur atau membuat keputusan intelejen tentang bagaimana memodifikasi pembangunan kimiawi.

2.2.4 Manfaat dan Keterbatasan Sistem Pakar

2.2.4.1 Manfaat Sistem Pakar

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli
2. Bisa melakukan proses secara berulang secara otomatis

3 Menyimpan pengetahuan dan keahlian para pakar

4 Mampu mengambil dan melestarikan keahlian para pakar (terutama yang termasuk keahlian langka)

5 Mampu beroperasi dalam lingkungan yang berbahaya

6 Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian. Pengguna bisa merespon dengan jawaban ’tidak tahu’ atau ’tidak yakin’ pada satu atau lebih pertanyaan selama konsultasi dan sistem pakar tetap akan memberikan jawaban.

7 Tidak memerlukan biaya saat tidak digunakan, sedangkan pada pakar manusia memerlukan biaya sehari-hari.

8 Dapat digandakan (diperbanyak) sesuai kebutuhan dengan waktu yang minimal dan sedikit biaya

9 Dapat memecahkan masalah lebih cepat daripada kemampuan manusia dengan catatan menggunakan data yang sama.

10 Menghemat waktu dalam pengambilan keputusan

11 Meningkatkan kualitas dan produktivitas karena dapat memberi nasehat yang konsisten dan mengurangi kesalahan

12 Meningkatkan kapabilitas sistem terkomputerisasi yang lain. Integrasi Sistem Pakar dengan sistem komputer lain membuat lebih efektif, dan bisa mencakup lebih banyak aplikasi .

13 Mampu menyediakan pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman. Fasilitas penjelas dapat berfungsi sebagai guru.

2.2.4.2 Keterbatasan Sistem Pakar

Metodologi Sistem Pakar yang ada tidak selalu mudah, sederhana dan efektif. Berikut adalah keterbatasan yang menghambat perkembangan Sistem Pakar:

a. Pengetahuan yang hendak diambil tidak selalu tersedia.

b. Kepakaran sangat sulit diekstrak dari manusia.

c. Pendekatan oleh setiap pakar untuk suatu situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.

d. Adalah sangat sulit bagi seorang pakar untuk mengabstraksi atau menjelaskan langkah mereka dalam menangani masalah

e. Pengguna Sistem Pakar mempunyai batas kognitif alami, sehingga mungkin tidak bisa memanfaatkan sistem secara maksimal.

f. Sistem Pakar bekerja baik untuk suatu bidang yang sempit.

g. Banyak pakar yang tidak mempunyai jalan untuk mencek apakah kesimpulan mereka benar dan masuk akal.

h. Istilah dan jargon yang dipakai oleh pakar dalam mengekspresikan fakta seringkali terbatas dan tidak mudah dimengerti oleh orang lain.

i. Pengembangan Sistem Pakar seringkali membutuhkan perekayasa pengetahuan (knowledge engineer) yang langka dan mahal.

j. Kurangnya rasa percaya pengguna menghalangi pemakaian Sistem Pakar.

k. Transfer pengetahuan dapat bersifat subyektif dan bias.

2.2.5 Area Permasalahan Aplikasi Sistem Pakar

1. Interpretasi

Pengambilan keputusan dari hasil observasi, diantaranya pengawasan, pengenalan ucapan, analisis citra, interpretasi sinyal, dan beberapa analisis kecerdasan.

2. Prediksi

   Memprediksi akibat-akibat yang dimungkinkan dari situasi-situasi tertentu, diantaranya peramalan, prediksi demografis, peralaman ekonomi, prediksi lalu lintas, estimasi hasil, militer, pemasaran, atau peramalan keuangan.

3. Diagnosis

   Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati, diantaranya medis, elektronis, mekanis, dan diagnosis perangkat lunak.

4. Desain

   Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu dan kendala-kendala tertentu, diantaranya layout sirkuit, perancangan bangunan.

5. Perencanaan

   Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu, diantaranya perencanaan keuangan, komunikasi, militer, pengembangan politik, *routing* dan manajemen proyek.

6. Monitoring

   Membandingkan tingkah laku suatu sistem yang teramati dengan tingkah laku yang diharapkan darinya, diantaranya *Computer Aided Monitoring System.*

7. *Debugging* dan *repair*

   Menentukan dan mengimplementasikan cara-cara untuk mengatasi malfungsi, diantaranya memberikan resep obat terhadap suatu kegagalan.

8. Instruksi

   Melakukan instruksi untuk diagnosis, *debugging* dan perbaikan kinerja.

9. Kontrol

   Mengatur tingkah laku suatu *environment* yang kompleks seperti kontrol terhadap interpretasi-interpretasi, prediksi, perbaikan, dan monitoring kelakuan sistem

10. Seleksi

    Mengidentifikasi pilihan terbaik dari sekumpulan (*list*) kemungkinan.

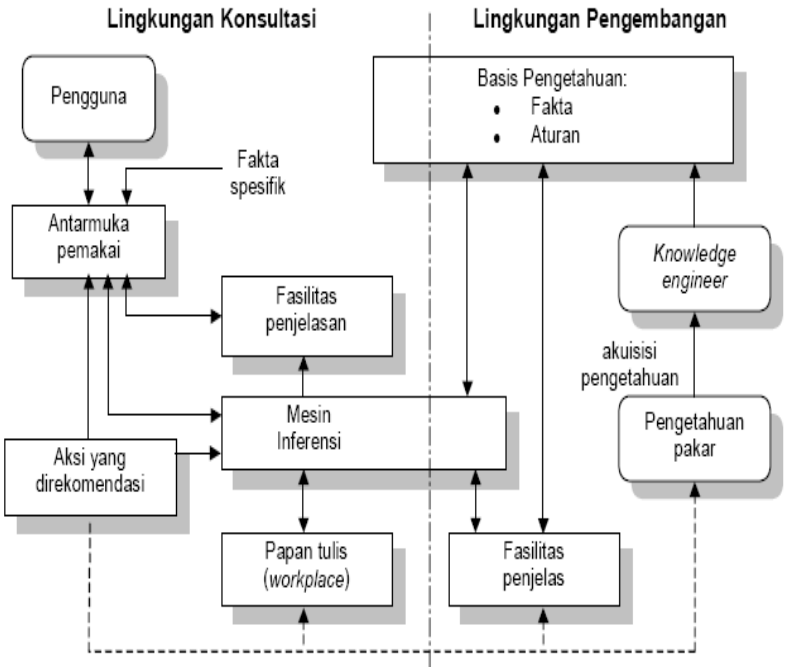
11. Simulasi

    Pemodelan interaksi antara komponen-komponen sistem.

2.2.6 Komponen Sistem Pakar

Sistem pakar terdiri dari dua bagian pokok, yaitu lingkungan pengembangan (*development environment)* dan lingkungan konsultasi (*consultation environment*).

Lingkungan pengembangan digunakan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seorang yang bukan ahli untuk berkonsultasi (Gambar 2.2 Struktur Sistem Pakar).



Gambar 2. 2 Struktur Sistem Pakar

Papan Tulis *(Blackboard/Workplace),* adalah memori/lokasi untuk bekerja dan menyimpan hasil sementara. Biasanya berupa sebuah basis data. Antarmuka Pemakai *(User Interface).* Sistem Pakar mengatur komunikasi antara pengguna dan komputer. Komunikasi ini paling baik berupa bahasa alami, biasanya disajikan dalam bentuk tanya-jawab dan kadang ditampilkan dalam bentuk gambar atau grafik. Antarmuka yang lebih canggih dilengkapi dengan percakapan (*voice communication*).

Subsistem Penjelasan *(Explanation Facility).* Kemampuan untuk menjejak (*tracing*) bagaimana suatu kesimpulan dapat diambil merupakan hal yang sangat penting untuk transfer pengetahuan dan pemecahan masalah.

Komponen subsistem penjelasan harus dapat menyediakannya yang secara interaktif menjawab pertanyaan pengguna, Misalnya:

1. “Mengapa pertanyaan tersebut anda tanyakan?”

2. “Seberapa yakin kesimpulan tersebut diambil?”

3. “Mengapa alternatif tersebut ditolak?”

4. “Apa yang akan dilakukan untuk mengambil suatu kesimpulan?”

5. “Fakta apalagi yang diperlukan untuk mengambil kesimpulan akhir?”

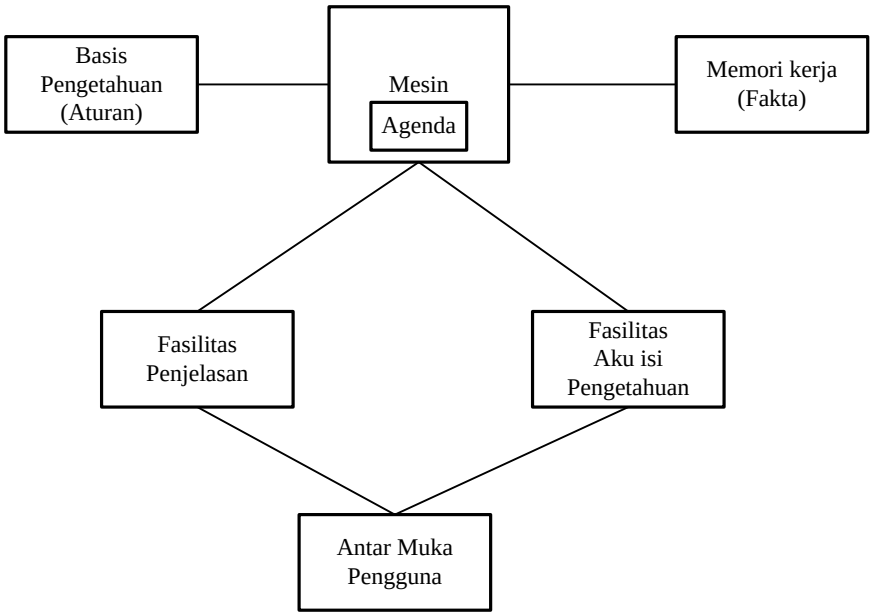
Sistem Penghalusan Pengetahuan *(Knowledge Refining System)*. Seorang pakar mempunyai sistem penghalusan pengetahuan artinya, mereka bisa menganalisa sendiri performa mereka, belajar dari pengalaman, serta meningkatkan pengetahuannya untuk konsultasi berikutnya. Pada Sistem Pakar, swa-evaluasi ini penting sehingga dapat menganalisa alasan keberhasilan atau kegagalan pengambilan kesimpulan, serta memperbaiki basis pengetahuannya.

2.2.7 Arsitektur Sistem Pakar

Sistem pakar memiliki beberapa komponen utama, yaitu antarmuka pengguna (*user interface*), basis data sistem pakar (*expert system database*), fasilitas akuisisi pengetahuan (*knowledge acquisition facility*), dan mekanisme inferensi (*inference mechanism*). Selain itu ada satu komponen yang hanya ada pada beberapa sistem pakar, yaitu fasilitas penjelasan (*explanation facility*) (Martin dan Oxman,1988).

Ada 4 tipe penjelasan yang digunakan dalam sistem pakar (Gambar 2.3 Arsitektur Sistem Pakar), yaitu (Schnupp, 1989):

1. penjelasan mengenai jejak aturan yang menunjukkan status konsultasi.
2. Penjelasan mengenai bagaimana sebuah keputusan diperoleh.
3. Penjelasan mengapa sistem menanyakan suatu pertanyaan.
4. Penjelasan mengapa sistem tidak memberikan keputusan seperti yang dikehendaki pengguna.



Gambar 2. 3 Arsitektur Sistem Pakar

Memori kerja dalam arsitektur sistem pakar (Gambar 2.3) merupakan bagian dari sistem pakar yang berisi fakta-fakta masalah yang ditemukan dalam suatu sesi, berisi fakta-fakta tentang suatu masalah yang ditemukan dalam proses konsultasi.

2.2.8 Pemilihan Masalah

Pembuatan Sistem Pakar membutuhkan waktu dan biaya yang banyak. Untuk menghindari kegagalan yang memalukan dan kerugian yang besar, maka dibuat beberapa pedoman untuk menentukan apakah Sistem Pakar cocok untuk memecahkan suatu problem:

a. Biaya yang diperlukan untuk pembangunan Sistem Pakar ditentukan oleh kebutuhan untuk memperoleh solusi. Sehingga harus ada perhitungan yang realistis untuk cost and benefit.

b. Pakar manusia tidak mudah ditemui untuk semua situasi di mana dia dibutuhkan. Jika pakar pengetahuan tersebut terdapat di mana saja dan kapan saja, maka pembangunan Sistem Pakar menjadi kurang berharga.

c. Problem yang ada dapat diselesaikan dengan teknik penalaran simbolik, dan tidak membutuhkan kemampuan fisik.

d. Problem tersebut harus terstruktur dengan baik dan tidak membutuhkan terlalu banyak pengetahuan awam (common sense), yang terkenal sulit untuk diakuisisi dan dideskripsikan, dan lebih banyak berhubungan dengan bidang yang teknis.

e. Problem tersebut tidak mudah diselesaikan dengan metode komputasi yang lebih tradisionil. Jika ada penyelesaian algoritmis yang bagus untuk problem tersebut, maka kita tidak perlu memakai Sistem Pakar.

f. Ada pakar yang mampu memberikan penjelasan tentang kepakarannya serta mau bekerjasama. Adalah sangat penting bahwa pakar yang dihubungi benar-benar mempunyai kemauan kuat untuk ikut berpartisipasi serta tidak merasa pekerjaannya akan menjadi terancam.

g. Problem tersebut mempunyai sekup yang tepat. Biasanya merupakan problem yang membutuhkan kepakaran yang sangat khusus namun hanya membutuhkan seorang pakar untuk dapat menyelesaikannya dalam waktu yang relatif singkat (misalnya paling lama 1 jam).

2.2.9 Rekayasa Pengetahuan (*Knowledge Engineering*)

Proses dalam rekayasa pengetahuan meliputi (Gambar 2.4):

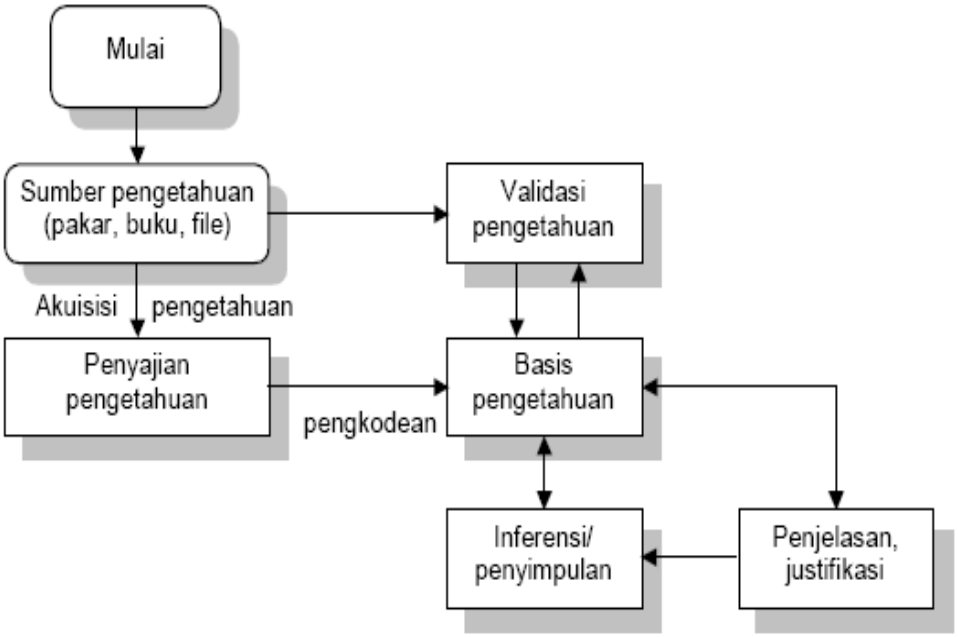
a. Akuisisi pengetahuan, yaitu bagaimana memperoleh pengetahuan dari pakar atau sumber lain (sumber terdokumentasi, buku, sensor, file komputer, dll.).

b. Validasi pengetahuan, untuk menjaga kualitasnya misalnya dengan uji kasus.

c. Representasi pengetahuan, yaitu bagaimana mengorganisasi pengetahuan yang diperoleh, mengkodekan dan menyimpannya dalam suatu basis pengetahuan.

d. Penyimpulan pengetahuan, menggunakan mesin inferensi yang mengakses basis pengetahuan dan kemudian melakukan penyimpulan.

e. Transfer pengetahuan (penjelasan). Hasil inferensi berupa nasehat, rekomendasi, atau jawaban, kemudian dijelaskan ke pengguna oleh subsistem penjelas.



Gambar 2. 4 Proses dalam Rekayasa Pengetahuan

2.2.10 Partisipan Dalam Proses Pengembangan

Pakar, yaitu seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasehat. Pakar menyediakan pengetahuan tentang bagaimana nantinya Sistem Pakar bekerja.

Perekayasa pengetahuan (knowledge engineer), yang membantu pakar untuk menyusun area permasalahan dengan menerjemahkan dan mengintegrasikan jawaban pakar terhadap pertanyaan-pertanyaan dari klien, menarik analogi, serta

memberikan contoh-contoh yang berlawanan, kemudian menyusun basis pengetahuan.

Pengguna, yang mungkin meliputi: seorang klien non-pakar yang sedang membutuhkan nasehat (Sistem Pakar sebagai konsultan atau advisor), seorang siswa yang sedang belajar (Sistem Pakar sebagai instruktur), seorang pembuat Sistem Pakar yang hendak meningkatkan basis pengetahuan (Sistem Pakar sebagai partner), seorang pakar (Sistem Pakar sebagai kolega atau asisten, yang dapat memberikan opini kedua).

Partisipan lain, dapat meliputi: pembangun sistem (system builder), tool builder, staf administrasi dan lain sebagainya.

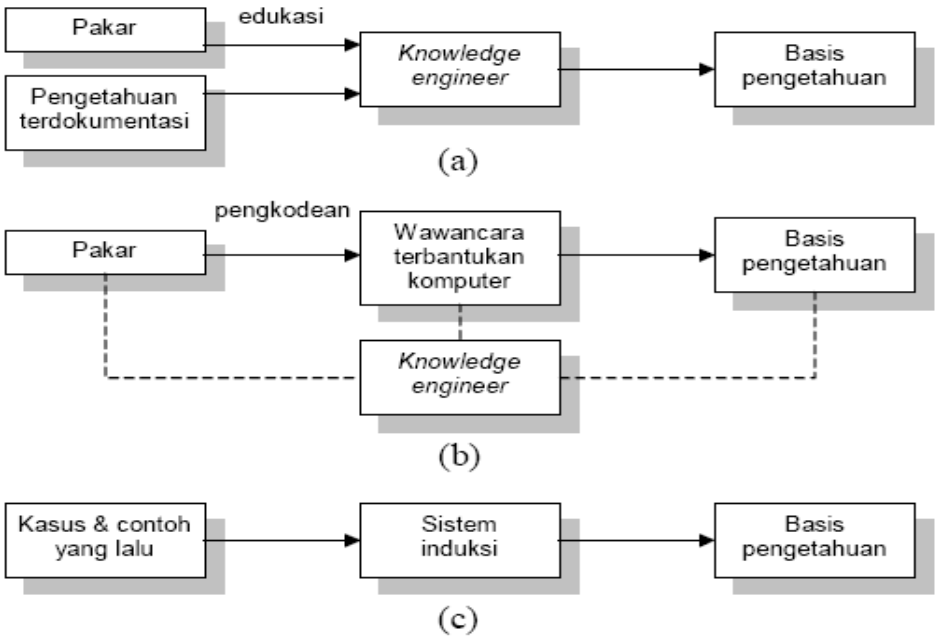
2.2.11 Akuisi Pengetahuan

Dalam proses akuisisi pengetahuan, seorang perekayasa pengetahuan menjembatani antara pakar dengan basis pengetahuan. Perekayasa pengetahuan mendapatkan pengetahuan dari pakar, mengolahnya bersama pakar tersebut, dan menaruhnya dalam basis pengetahuan, dengan format tertentu. Pengambilan pengetahuan dari pakar dapat dilakukan secara (Gambar 2.5):

Manual, di mana perekayasa pengetahuan mendapatkan pengetahuan dari pakar (melalui wawancara) dan/atau sumber lain, kemudian mengkodekannya dalam basis pengetahuan. Proses ini biasanya berlangsung lambat, mahal, serta kadangkala tidak akurat.

Semi-otomatik, di mana terdapat peran komputer untuk: (1) mendukung pakar dengan mengijinkannya membangun basis pengetahuan tanpa (atau dengan sedikit) bantuan dari perekayasa pengetahuan, atau (2) membantu perekayasa pengetahuan sehingga kerjanya menjadi lebih efisien dan efektif.

Otomatik, di mana peran pakar, perekayasa pengetahuan, dan pembangun basis pengetahuan (system builder) digabung. Misalnya dapat dilakukan oleh seorang system analyst seperti pada metode induksi.



Gambar 2. 5 Metode Akuisisi Pengetahuan (a) Manual (b) Akuisisi Terkendali Pakar (c)Induksi

2.2.12 Komponen-komponen Sistem Pakar

1. Basis Pengetahuan (*Knowledge Base*)

Pengetahuan merupakan kemampuan untuk membentuk model mental yang menggambarkan obyek dengan tepat dan mempresentasikannya dalam aksi yang dilakukan terhadap suatu obyek (Martin dan Oxman, 1988).

Pengetahuan dapat diklasifikasikan menjadi tiga, yaitu pengetahuan prosedural (*procedural knowledge*), pengetahuan deklaratif (*declaratif knowlwdge*), dan pengetahuan tacit (*tacit knowledge*). Pengetahuan prosedural lebih menekankan pada bagaimana melakukan sesuatu, pengetahuan deklaratif menjawab pertanyaan apakah sesuatu bernilai salah atau benar, sedangkan pengetahuan tacit merupakan pengetahuan yang tidak dapat diungkapkan dengan bahasa. Basis Pengetahuan merupakan inti program Sistem Pakar dimana basis pengetahuan ini adalah representasi pengetahuan (*Knowledge Representation*) dari seorang pakar.

Pengetahuan dapat dipresentasikan dalam bentuk yang sederhana atau kompleks, tergantung dari masalahnya (Schnupp, 1989).

Ada beberapa model representasi yang penting yaitu : logika (*logic*), jaringan semantik (*semantic nets*), bingkai (*frame*), kaidah produksi (*production rule*).

a. Logika (*logic*)

Logika merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah, dan prosedur yang membantu proses penalaran. Logika merupakan bentuk representasi pengetahuan yang paling tua, yang menjadi dasar dari teknik representasi *high level*.

b. Jaringan Sematik

Merupakan suatu gambaran dari pengetahuan yang memperlihatkan hubungan hirarki dari objek – objek. Objek dipresentasikan dalam bentuk node dan hubungan antara objek dinyatakan oleh garis penghubung beratribut.

c. Bingkai (Frame)

Yaitu blok – blok berisi pengetahuan mengenai objek tertentu, kejadian, lokasi, situasi dari elemen – elemen lain yang menggambarkan objek tersebut secara rinci, dimana rincian objek tersebut disimpan ke dalam sebuah slot yang menggambarkan berbagai atribut dan karakteristik dari objek.

d. Kaidah Produksi

Metode kaidah produksi biasanya dituliskan dalam bentuk jika maka (*if-then*). Kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian, yaitu: pertama jika (premise) dan yang kedua, yaitu muka

(konkulasi). Apabila bagian jika dipenuhi maka bagian muka akan bernilai benar.

2. Basis Data (*Data Base)*

Basis Data adalah bagian yang mengandung semua fakta-fakta, baik fakta awal pada saat sistem mulai beroperasi maupun fakta-fakta yang didapatkan pada saat pengambilan kesimpulan yang sedang dilaksanakan. Dalam praktiknya, Basis data berada di dalam memori komputer. Kebanyakan Sistem Pakar mengandung Basis Data untuk menyimpan data hasil observasi dan data lainnya yang dibutuhkan selama pengolahan.

3. Mesin Inferensi (Inferensi Engine)

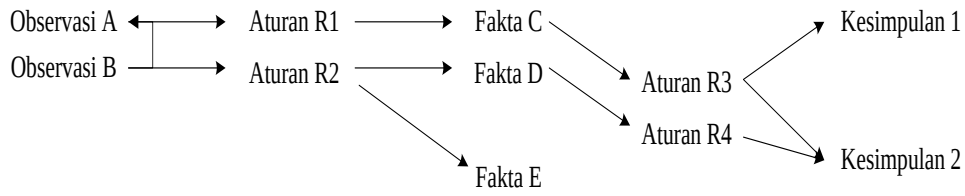
Mesin Inferensi adalah bagian yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang akan menganalisis suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. Secara deduktif mesin inferensi memilih pengetahuan yang relevan dalam rangka mencapai kesimpulan. Dengan demikian sistem ini dapat menjawab pertanyaaan pemakai meskipun jawaban tersebut tidak tersimpan secara eksplisit di dalam basis pengetahuan. Mesin Inferensi memulai pelacakannya dengan mencocokan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.

Dibawah ini ada 2 macam metode inference, yaitu :

a. Forward Chaining (Pelacakan ke Depan)

Pendekatan yang dimotori oleh data (data driven). Dalam pendekatan ini pelacakan dimulai dari informasi masukan dan selanjutnya mencoba menggambarkan kesimpulan. Aturan dalam sistem merepresentasikan aksi-aksi yang harus diambil apabila terdapat suatu kondisi khusus pada item-item dalam memori kerja yang disebut himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil. Aktivitas sistem dilakukan berdasarka siklus mengenal-beraksi (*recognize-act*).

Pelacakan ke depan mencari fakta yang sesuai dari bagian IF dari aturan IF-THEN (Gambar 2.6 menunjukkan proses forward chaining).



Gambar 2. 6 Proses Forward Chaining

b. Backward Chaining

Merupakan penalaran dari node tujuan dan bergerak ke belakang menuju keadaan awal, dalam penalaran ke belakang prosesnya disebut terarah,

menggunakan pendekatan goal-driven, dimulai dari ekspektasi apa yang diinginkan terjadi (*hipotesis*), kemudian mengecek pada sebab-sebab yang mendukung (ataupun kontadiktif) dari ekspektasi tersebut.

Contoh :

R1 : IF suku bunga turun THEN harga obligasi naik

R2 : IF suku bunga naik THEN harga obligasi turun

R3 : IF suku bunga tidak berubah THEN harga obligasi tidak berubah

R4 : IF dolar naik THEN suku bunga turun

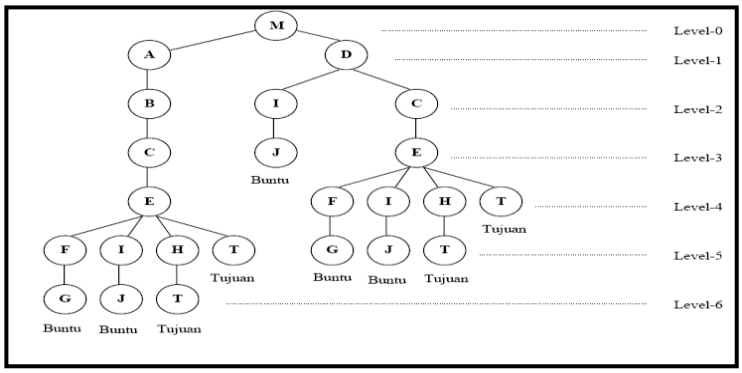
R5 : IF dolar turun THEN suku bunga naik

R6 : IF harga obligasi turun THEN beli obligasi

2.2.13 Pohon Pelacakan

Hampir semua masalah *Artificial Intellegence* ditampilkan dalam bentuk grafik atau jaringan yang berbentuk node dan akar yang disebut pohon pelacakan. Untuk menghindari kemungkinan adanya proses pelacakan suatu node secara berulang, maka digunakan struktur pohon.

Pohon pencarian untuk *graph* keadaan dengan 6 level (Gambar 2.7 Struktur pohon).



Gambar 2.7 Struktur pohon

Pada Gambar 2.7 tersebut sudah tidak terlihat lagi adanya siklus, karena setiap node tidak diperbolehkan memiliki cabang kembali ke node dengan level yang lebih rendah.

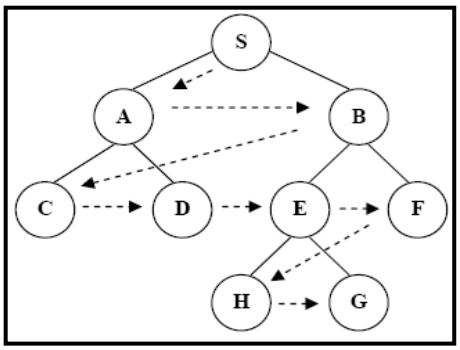
Struktur pohon digunakan untuk menggambarkan keadaan secara hirarki yang terdiri dari beberapa node. Node yang terletak pada level-0 disebut dengan nama aka. Node akar menunjukkan keadaan awal yang biasanya merupakan topic atau objek. Node akar teletak pada level ke nol yang memiliki beberapa percabangan yang terdiri atas beberapa node successor yang disebut dengan nama anak dan merupakan node-node perantara.

Node ini pada gilirannya mempunyai pengganti lagi sebagai anak, bergerak mundur melalui pohon, node ini disebut node pendahuluan, nenek moyang atau orang tua. Node yang tidak mempunyai anak atau tidak mempunyai pengganti disebut node pengganti. Akar yang disilang dan dihubungkan disebut batang.

2.2.13.1 Metode Pencarian dan Pelacakan

A. Pencarian Melebar Pertama (*Breadth-First Search*)

Pada metode *Breadth-First Search* ini adalah semua node pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi node-node pada level n+1. pencarian dimulai dari node akar terus ke level 1 dari kiri ke kanan, kemudian berpindah ke level berikutnya dari kiri ke kanan hingga solusi ditemukan (Gambar 2.8 Metode *Breadth-First Search*).



Gambar 2.8 Metode *Breadth-First Search*

Algoritma :

1. Buat suatu variable *Node_List* dan tetapkan sebagai keadaan awal.
2. Kerjakan langkah-langkah berikut ini sampai tujuan tercapai atau *Node_List* dalam keadaan kosong :
   a) Hapus elemen pertama dari *Node_List*, sebut dengan nama E. Jika *Node_List* kosong, keluar.
   b) Pada setiap langkah yang aturannya cocok dengan E, kerjakan :

i. Aplikasikan aturan tersebut membentuk suatu keadaan baru.

ii. Jika keadaan awal adalah tujuan yang diharapkan, sukses dan keluar.

iii. Jika tidak demikian, tambahkan keadaan awal yang baru tersebut pada akhir *Node_List*.

Keuntungan :

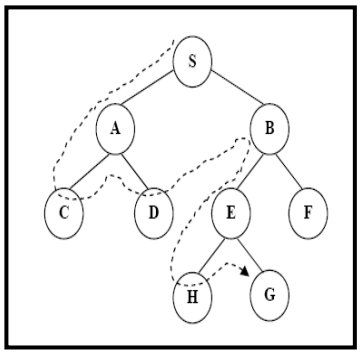
1. Tidak akan menemui jalan buntu
2. Jika ada 1 solusi, maka *breadth – first search* solusi akan menemukannya dan jika ada lebih dari satu solusi, maka solusi minimum akan ditemukan.

Kelemahan :

1. Membutuhkan memori yang cukup banyak
2. Membutuhkan waktu yang cukup lama, karena akan menguji n level untuk mendapatkan solusi pada level yang ke-(n+1).

B. Pencarian Mendalam Pertama (*Depth – First Search*)

Pada metode *Depth – First Search*, Proses pencarian dilakukan pada semua anaknya sebelum dilakukan pencarian ke node-node yang selevel. Pencarian dimulai dari node akar ke level yang lebih tinggi. Proses diulangi terus hingga ditemukan solusi (Gambar 2.9 Metode *Depth – First Search*).



Gambar 2.9 Metode *Depth – First Search*

Algoritma :

1. Jika keadaan awal merupakan tujuan, keluar (sukses).
2. Jika tidak demikian, kerjakan langkah-langkah berikut ini sampai tercapai keadaan sukses atau gagal :
   a) Bangkitkan successor E dari keadaan awal. Jika tidak ada successor, maka akan terjadi kegagalan.
   b) Panggil *Depth – First Search* dengan E sebagai keadaan awal.
   c) Jika sukses berikan tanda sukses. Namun jika tidak, ulangi langkah-2.

Keuntungan :

1. Membutuhkan memori relatif kecil, karena hanya node-node pada lintasan yang aktif saja yang disimpan.
2. Secara kebetulan, metode *Depth – First Search* akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan.

Kelemahan :

1. Memungkinkan tidak ditemukannya tujuan yang diharapkan
2. Hanya mendapat satu solusi pada setiap pencarian

C. Pencarian Terbaik Pertama (*Best First Search)*

Metode Pencarian Terbaik Pertama *(Best First Search)* merupakan kombinasi dari metode *depth first search* dan metode *breadth first search* dengan mengambil kelebihan dari kedua metode tersebut. Pada setiap langkah proses pencarian terbaik pertama, kita memilih node-node dengan menerapkan fungsi heuristik yang memadai pada setiap node atau simpul yang kita pilih dengan menggunakan aturan-aturan tertentu untuk menghasilkan penggantinya. Pada *best first search*, pencarian diperbolehkan mengunjungi node di lebih rendah dan jika ternyata node dilevel lebih tinggi memiliki nilai heuristik lebih buruk.

Untuk mengimplementasikan metode ini, dibutuhkan dua antrian yang berisi node-node, yaitu :

1. *OPEN* yaitu berisi node-node yang sudah dibangkitkan, sudah memiliki fungsi heuristik namun belum diuji. Umumnya berupa antrian berprioritas yang berisi elemen-elemen dengan nilai heuristik tertinggi.
2. *CLOSED* yaitu berisi node-node yang sudah diuji

Algoritma :

1. Tempatkan node awal A pada antrian OPEN

2. Kerjakan langkah-langkah berikut hingga tujuan ditemukan atau antrian OPEN sudah kosong :

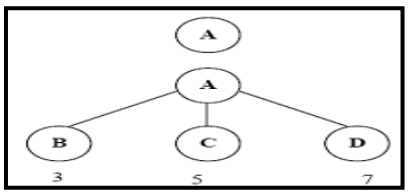
a) Ambil node terbaik dari OPEN (Gambar 2.11 Antrian *OPEN*)

b) Bangkitkan semua successornya

c) Untuk tiap-tiap successor kerjakan :

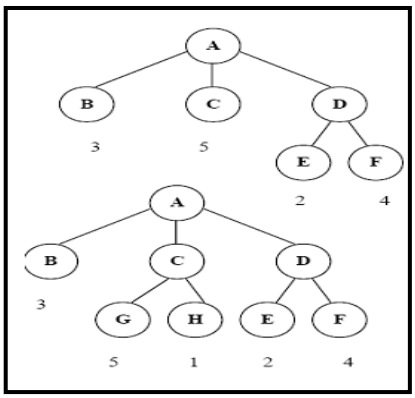
i. Jika node tersebut belum pernah dibangkitkan sebelumnya, evaluasi node tersebut dan masukkan ke OPEN.

ii. Jika node tersebut sudah pernah dibangkitkan sebelumnya, ubah parent jika lintasan baru lebih menjanjikan. Hapus node tersebut dari antrian OPEN.



Antrian *OPEN*

[A]



[D C B]

[C F B E]

[G F B E H]

Gambar 2.10 Antrian *OPEN*

Diasumsikan node dengan nilai yang lebih besar memiliki nilai evaluasi yang lebih baik. Pada keadaan awal, antrian berisi A. Pengujian dilakukan di level pertama, node D memiliki nilai terbaik, sehingga menempati antrian pertama, disusul dengan C dan B. Node D memiliki cabang E dan F yang masing-masing bernilai 2 & 4. Dengan demikian C merupakan pilihan terbaik dengan menempati antrian pertama (Gambar 2.12 Metode *Best First Search*).

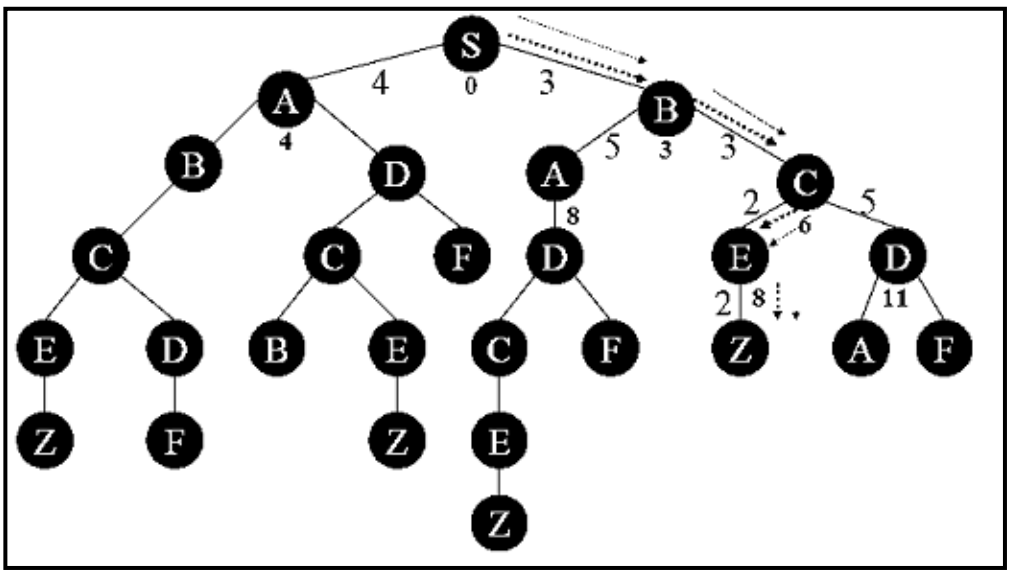
Fungsi Heuristik yang digunakan merupakan prakiraan (estimasi) *cost* dari *initial state* ke *goal state*, yang dinyatakan dengan :

f’ = g + h’

dimana : f’ = prakiraan *cost* dari *initial* ke *goal*

g = *cost* dari *initial state* ke *current state*

h’ = prakiraan *cost* dari *current state* ke *goal state*



Gambar 2.11 Metode *Best First Search*

Keuntungan :

1. Membutuhkan memori yang relatif kecil, karena hanya node-node pada lintasan yang aktif saja yang disimpan.
2. Menemukan solusi tanpa harus menguji lebih banyak node.

Kelemahan :

1. Memungkinkan terjebak pada nilai optima.

2.2.14 Bidang-bidang Sistem Pakar

Ada banyak area atau wilayah yang menjadi daerah kerja *Artificial Intelligence* yaitu jaringan saraf, sistem persepsi, robotic, bahsa ilmiah, sistem pendukung keputusan, sistem informasi berbasis manajemen dan sistem pakar.

Tiap daerah kerja *Artificial Intelligence* memiliki potensi dalam memecahkan masalah, tetapi keunggulan utama dalam bentuk pengetahuan dari pakar manusia secara *heuristic* dalam sistem pakar. *Heuristic* dalam sistem pakar tidak menjamin hasil mutlak sistem kecerdasan buatan lainnya, tetapi menawarkan hasil yang spesifik untuk dimanfaatkan karena sistem pakar berfungsi secara konsisten seperti seorang pakar manusia, menawarkan nasihat kepada pemakai dan menemukan solusi terhadap berbagai permasalahan yang spesifik. Ada berbagai kategori pengembangan sistem pakar, antara lain :

1. Kontrol. Contoh pengembangan banyak ditemukan dalam kasus pasien di rumah sakit, dimana dengan kemampuan sistem pakar dapat dilakukan *control* terhadap cara pengobatan dan perawatan melalui sensor data dan memberikan solusi terapi pengobatan yang tepat bagi si pasien yang sakit.

2. Desain. Contoh sistem pakar untuk membantu mendesain komputer dengan komponen-komponennya.

3. Diagnosis. Contoh sistem pakar ini adalah diagnosis penyakit, kerusakan mesin kendaraan bermotor, kerusakan komponen komputer, dan lain-lain.

4. Instruksi. Instruksi merupakan pengembangan sistem pakar yang sangat berguna dalam bidang ilmu pengetahuan dan pendidikan, dimana system pakar dapat memberikan instruksi dan pengajaran tertentuterhadap suatu topik permasalahan. Contoh pengembangan yaitu system pakar untuk pengajaran bahasa Inggris dan lain-lain.

5. Interpretasi. Sistem pakar yang dikembangkan dalam bidang interpretasi melakukan proses pemahaman akan suatu situasi dari beberapa informasi yang direkam.

6. Monitor. Sistem pakar ini biasanya digunakan pada kemiliteran yaitu menggunakan sensor radar.

7. Perencanaan. Perencanaan banyak digunakan dalam bidan bisnis dan keuangan suatu proyek, dimana system pakar yang melakukan perencanaan suatu pekerjaan berdasarkan jumlah tenaga kerja dan biaya.

8. Prediksi. Biasanya system memberikan simulasi kejadian masa mendatangtersebut, misalnya memprediksi tingkat kerusakan tanaman apabila diserang hama.

9. Seleksi. Sistem pakar dengan seleksi mengidentifikasi pilihan terbaik dari beberapa pilihan kemungkinan solusi.

10. Simulasi. Contoh adalah program PLANT yang menggabungkan antara prediksi dan simulasi, dimana program tersebut mampu menganalisis hama dengan berbagai kondisi suhu dan cuaca.

2.3 Basis Data

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi masalah pada sistem yang memakai pendekatan berbasis berkas.

2.4 Metode Analisis yang digunakan

2.4.1 Flowchart

Flowchart merupakan gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut. Hal tersebut memungkinkan untuk memecah proses menjadi kejadian-kejadian individual atau aktifitas untuk menunjukan secara singkat hubungan diantaranya.

Konstruksi flowchart memungkinkan pengertian lebih baik kepada proses dan pengertian yang lebih baik terhadap proses akan membawa kepada perbaikan pengembangan suatu sistem

2.4.2 Diagram Konteks

Diagram konteks merupakan diagram alir pada tingkat palinng atas yang merupakan penggambaran yang berfungsi untuk memperlihatkan interkasi/hubungan langsung antara sistem dengan lingkungannya. Diagram konteks menggambarkan sebuah sistem berupa sebuah proses yang berhubungan dengan satu atau beberapa entitas/entity.

2.4.3 DFD (Data Flow Diagram)

DFD adalah suatu alat pemodelan yang digunakan untuk memodelkan fungsidari sistem, menggambarkan secara rinci mengenai sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan menunjukan dari dan kemana data mengalir serta penyimpanannya,

Beberapa simbol digunakan di DFD:

1. Kesatuan luar (*external entity*) atau batas sistem (*boundary*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, orhanisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan masi=ukan atau menerima keluaran dari sistem.
2. Arus data (*data flow*) ini mengalir diantara proses (process), sim[panan data (*data store*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. Proses (*process*) merupakan kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkam arus data yang akan keluar dari proses.
4. Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa suatu file atau database di sistemkomputer, suatu arsip atau catatan manual, suatu kotak tempat data di meja seseorang, suatu tabel acuan manual, dan suatu agenda atau buku.
5. Spesisikasi Proses (*Process Specification (PSPEC)*)
6. Spesifikasi proses digunakan untuk menggambarkan semua proses model aliran yang. Nampak pada tingkat akhir penyaringan. Kandungan dari spesifikasi proses dapat termasuk teks naratif, gambaran bahasa desain program (*Program Design Language*) dari algoritma proses, persaaan matematika, tabel, digram, atau bagan.

2.4.4 ERD (Entity Relationship Diagram)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu :

a. Entiti

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entitiini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atributdiwakili oleh simbol *elips*.

c. Hubungan / Relasi

Relasi adalah hubungan antara suatu himpunan entitas dengan himpunan entitas yang lainnya.Relasi direpresentasikan dengan jajaran genjang.

2.4.5 Kamus Data (*Data Dictionary)*

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

Dengan adanya kamus data, analisis sistem dapat mdendefinisikan data yang mengalir di dalam sistem dengan lengkap. Pada tahap perancangan sistem, kamus data dapat digunakan untuk merancang *input*, *output*, dan merancang database program. Kamus data dibuat berdasarkan arus data yang ada.

2.4.6. Perintah-perintah MySQL

Pada MySQL terdapat beberapa perintah. Perintah-perintah pada MySQL ini hampir sama dengan perintah-perintah pada *databaseserver* lainnya. Perintah-perintah MySQL itu antara lain adalah sebagai berikut :

1. *Create database*, digunakan untuk membuat *database* pada *databaseserver*.

   Sintaksnya adalah :

   Create database database_name

   Database_name adalah nama*database* yang akan dibuat.

2. *Use database*, digunakan untuk menunjuk *database* yang akan digunakan.

   Sintaksnya adalah :

   Use database_name

   Database_name adalah nama *database* yang akan digunakan.

3. *Create table*, digunakan untuk membuat tabel pada *database* yang digunakan.

   Sintaksnya adalah :

   Create table table_name

   (

   Column1 column_type column_attributes,

   Column2 column_type column_attributes,

   Primary_key (column_name)

   );

   Table_name adalah nama tabel yang akan dibuat. *Column1* adalah nama kolom yang akan dibuat pada tabel. *Column_type* adalah tipe dari kolom

tersebut , dapatberupa *char, varchar, tinytext, mediumtext, longtext, enum, int, tinyint, mediumint, bigint, float, decimal, time, date, datetime, timestamp, year.*

4. *Insert*, digunakan untuk menambahkan record pada tabel.

   Sintaksnya adalah :

   Insert into table_name(column1, column2,..) values (value1,value2,..)

   Table_name adalah nama tabel yang akan ditambahkan *record*-nya. *Column1, column2*adalah kolom yang akan ditambahkan data. *Value1,value2*adalah data yang akanditambahkan.

5. *Update*, digunakan untuk mengubah record yang sudah ada pada tabel.

   Sintaksnya adalah :

   Update table_name set column1=value1, column2=value2 where column=value

   Table_name adalah nama tabel yang akan dirubah *record*-nya. *Column1,column2* adalah kolom yang akan dirubah data. *Value1,value2*adalah data yangakan digantikan.

6. *Drop table*, digunakan untuk menghapus tabel.

   Sintaksnya adalah :

   Drop table table_name

   Table_name adalah nama tabel yang akan dihapus.

2.4.7 PHP

PHP adalah bahasa pemrograman web atau *scriptinglanguage* yang didesain untuk web. PHP dibuat pertama kali oleh Rasmus Lerdorf, yang pada awalnya

dibuat untuk menghitung jumlah pengunjung pada homepagenya. Diawal Januari 2001, PHP telah dipakai lebih dari 5 juta domain diseluruh dunia, dan akan terus bertambah karena kemudahan aplikasi PHP ini dibandingkan dengan bahasa *Serverside* yang lain.

2.4.8 Website

Pada awalnya internet adalah sebuah proyek yang dimaksudkan untuk menghubungkan para ilmuan dan peneliti di Amerika, namun saat ini telah tumbuh menjadi media komunikasi global yang dipakai semua orang di muka bumi. Pertumbuhan ini membawa beberapa masalah yang penting medasar, diantaranya kenyataan bahwa internet tidak diciptakan pada jaman *graphical user interface* (GUI) seperti saat ini. Internet dimulai pada masa dimana orang masih menggunakan alat-alat akses yang tidak *user-friendly* yaitu terminal berbasis teks serta perintah-perintah *command line* yang panjang-panjang serta sulit diingat, sangat berbeda dengan komputer dewasa ini yang menggunakan klik tombol mouse pada layar grafik berwarna.

Kemudian orang berpikir untuk membuat sesuatu yang lebih baik. Popularitas internet mulai berkembang pesat seperti jamur dimusim penghujan setelah standar baru yaitu HTTP dan HTML diperkenalkan kepada masyarakat. HTTP (Hypertext Transfer Protocol) membuat pengaksesan informasi melalui protokol TCP/IP menjadi lebih mudah dari sebelumnya. HTML (Hypertext Markup Language) memungkinkan orang menyajikan informasi secara visual lebih menarik.