

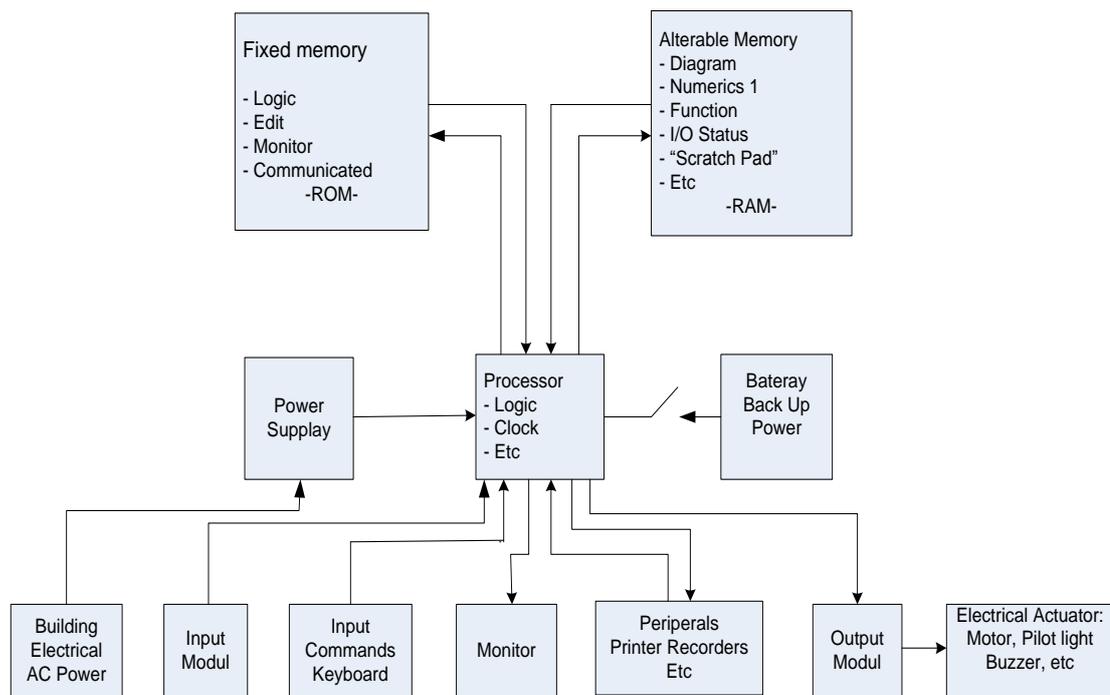
## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem Komputer**

Komputer adalah alat elektronik yang dapat menerima input data, mengolah data, dan memberikan informasi menggunakan suatu program yang tersimpan dalam memori komputer, menyimpan program dan hasil pengolahan dan bekerja secara otomatis. (Jogiyanto ; 2002 : 2)

Dalam bentuk yang paling sederhana komputer terdiri dari 5 bagian fungsional yang independent, yaitu : input, memori, aritmatik dan logika, output dan kontrol unit. Unit input menerima informasi yang dikodekan dari operator manusia, peralatan elektromagnetik seperti keyboard ke display atau dari komputer yang lain dengan menggunakan line komunikasi digital. Informasi yang diterima disimpan dalam memori untuk selanjutnya dipakai sebagai referensi atau digunakan segera oleh rangkaian aritmatika dan logika untuk menjalankan operasi yang diinginkan. Langkah - langkah untuk memproses suatu informasi tergantung program yang tersimpan di dalam memori. Dan pada akhirnya hasil dari proses akan dikirim keluar melalui output unit. Seluruh dari rangkaian pekerjaan ini dikontrol oleh kontrol unit. Struktur Central Processing Unit (CPU) computer ditunjukkan oleh gambar II.1. (Jogiyanto ; 2002 : 4)



**Gambar II.1 Blok Diagram Operasi CPU**

**Sumber : Jogyanto (2002 : 4)**

## II.2. Artificial Intelligence (AI)

*Artificial Intelligence* (AI) atau kecerdasan buatan adalah suatu ilmu pengetahuan dan teknologi yang mempelajari cara membuat komputer melakukan sesuatu seperti yang dilakukan oleh manusia. Kecerdasan buatan (AI) sebagai sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan oleh manusia . (Sri Kusuma Dewi ; 2003 : 1)

*Artificial Intelligence* (AI) memiliki tujuan untuk menciptakan komputer-komputer dapat berfikir lebih cerdas dan membuat mesin lebih berguna. Dorongan utama dari *Artificial Intelligence* (AI) adalah pengembangan fungsi normal

komputer yang digabungkan dengan kecerdasan manusia, seperti memberi alasan, menarik kesimpulan, belajar dan memecahkan masalah.

Lebih detailnya, pengertian kecerdasan buatan dapat dipandang dari berbagai sudut pandang, antara lain :

1. Sudut pandang kecerdasan

Kecerdasan buatan akan membuat mesin menjadi cerdas (mampu berbuat seperti apa yang dilakukan oleh manusia).

2. Sudut pandang penelitian

Kecerdasan buatan adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.

*Domain* yang sering dibahas oleh para peneliti meliputi :

1. *Mundane Task*

- a. Persepsi (*vision & speech*)
- b. Bahasa alami (*understanding, generation & translation*)
- c. Pemikiran yang bersifat *commonsense*
- d. *Robot control*

2. *Formal Task*

- a. Permainan/*Games*
- b. Matematika (geometri, logika, kalkulus integral, pembuktian)

3. *Expert Task*

- a. Analisis finansial
- b. Analisis Medikal
- c. Analisis Ilmu Pengetahuan

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

### 3. Sudut pandang bisnis

Kecerdasan Buatan adalah kumpulan peralatan yang sangat *powerfull* dan metodologis dalam menyelesaikan masalah-masalah bisnis.

### 4. Sudut pandang pemrograman

Kecerdasan Buatan meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*), dan pencarian (*searching*)

(Sri Kusuma Dewi; 2003 : 45)

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

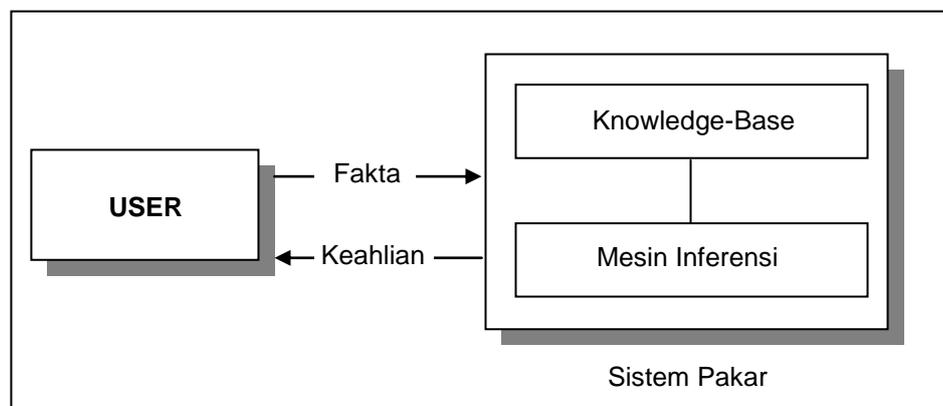
- a. Basis Pengetahuan (*Knowledge Base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- b. motor inferensi (*Inference Engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.

## II.3. Sistem Pakar

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

pakar sudah digunakan untuk berbagai macam sistem yang menggunakan teknologi sistem pakar itu. Teknologi sistem pakar ini meliputi bahasa sistem pakar, program dan perangkat keras yang dirancang untuk membantu pengembangan dan pembuatan sistem pakar.

*Knowledge* dalam sistem pakar mungkin saja seorang ahli, atau *knowledge* yang umumnya terdapat dalam buku, majalah dan orang yang mempunyai pengetahuan tentang suatu bidang. Istilah sistem pakar, *sistem knowledge-base*, atau sistem pakar *knowledge-base*, sering digunakan dengan arti yang sama. Kebanyakan orang menggunakan istilah sistem pakar karena lebih singkat, bahkan walau belum benar-benar pakar, hanya menggunakan *knowledge* secara umum.



**Gambar II.2 Konsep Dasar Fungsi Sistem Pakar**

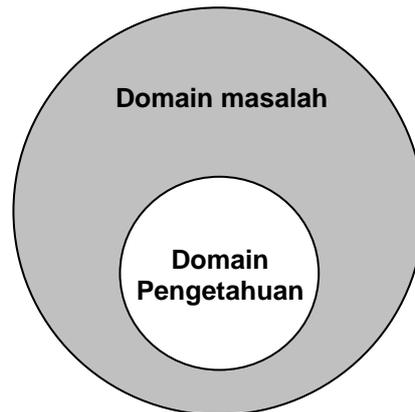
**Sumber: Sri Kesuma Dewi (2003: 5)**

Gambar II.2 menggambarkan konsep dasar suatu sistem pakar *knowledge-base*. Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Bagian dalam sistem pakar terdiri dari dua komponen utama, yaitu *knowledge base* yang berisi *knowledge* dan *mesin inferensi* yang menggambarkan kesimpulan. Kesimpulan tersebut merupakan respons dari sistem pakar atas permintaan pengguna.

Penggunaan sistem *knowledge base* (basis pengetahuan) juga dirancang untuk aksi pemandu cerdas seorang ahli. Pemandu cerdas dirancang dengan teknologi sistem pakar karena memberikan banyak keuntungan terhadap pengembangannya. Semakin banyak *knowledge* yang ditambahkan untuk pemandu cerdas maka sistem tersebut akan semakin baik dalam bertindak sehingga semakin menyerupai pakar sebenarnya. Pengembangan suatu pemandu cerdas merupakan permulaan bagi pengembangan sistem pakar yang lebih lengkap lagi.

Suatu *knowledge* dari sistem pakar bersifat khusus untuk satu domain masalah saja. Domain masalah adalah bidang atau ruang lingkup yang khusus, seperti kedokteran, keuangan, bisnis, ilmu pengetahuan atau teknik. Sistem pakar menyerupai kepakaran manusia yang secara umum dirancang untuk menjadi pakar dalam satu domain masalah saja.

*Knowledge* dari sistem pakar tentang penyelesaian masalah yang khusus disebut dengan domain *knowledge* dari suatu pakar. Sebagai contoh, sistem pakar kedokteran yang dirancang untuk mendiagnosis infeksi penyakit akan mempunyai suatu uraian *knowledge* tentang gejala-gejala penyakit yang disebabkan oleh infeksi penyakit. Dalam kasus ini *domain knowledge*-nya adalah bidang kedokteran yang terdiri dari *knowledge* tentang penyakit, gejala, dan cara pengobatan. Gambar II.3 menggambarkan hubungan antara domain masalah dan domain *knowledge*. Dapat dilihat juga bahwa domain *knowledge* secara keseluruhan merupakan bagian dari domain masalah.



**Gambar II.3 Hubungan Antara Domain Knowledge Dan Domain Masalah**  
**Sumber: Sri Kesuma Dewi (2003: 5)**

Secara umum Sistem Pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli (Muhammad Arhami, 2005). Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Dengan sistem pakar ini, orang awampun dapat menyelesaikan masalah yang cukup rumit, yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang sangat berpengalaman. (Sri Kusumadewi, 2003).

### **II.3.1 Sejarah Sistem Pakar**

Perkembangan *Artificial Intelligence* merupakan terobosan baru dalam dunia komputer. *Artificial Intelligence* berkembang setelah perusahaan General Electric menggunakan komputer pertama kali di bidang bisnis. Pada tahun 1956, istilah *Artificial Intelligence* mulai dipopulerkan oleh John McCarthy sebagai suatu tema ilmiah di bidang komputer yang diadakan di Dartmouth College.

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

Setelah GPS, ternyata *Artificial Intelligence* banyak dikembangkan dalam bidang permainan atau *game*. Misalnya program permainan catur oleh Shannon (1955) dan program untuk pengecekan masalah oleh Samuel (1963). Banyak juga ahli yang mengimplementasikan *Artificial Intelligence* dalam bidang bisnis dan matematika.

Pada tahun 1972, Newell dan Simon memperkenalkan Teori Logika secara konseptual yang kemudian berkembang pesat dan menjadi acuan pengembangan sistem berbasis kecerdasan buatan lainnya.

Buchanan dan Feigenbaum juga mengembangkan bahasa pemrograman DENDRAL. Pada tahun 1978. Bahasa pemrograman ini dibuat untuk badan antariksa Amerika Serikat, yaitu NASA, dan digunakan untuk penelitian kimia di planet Mars.

Pada tahun 1976, yaitu 2 tahun sebelum DENDRAL, sebenarnya program sistem pakar sudah dikembangkan secara modern, yaitu MYCIN yang dibuat oleh Shortliffe dengan bahasa pemrograman LISP. Program MYCIN menyimpan ± 500 basis pengetahuan dan basis aturan untuk mendiagnosis penyakit manusia. Program ini juga mengimplementasikan metode penelusuran dan pemecahan

masalah, serta mengembangkan berbagai teori penting dalam kecerdasan buatan seperti metode *certainty factor*, teori probabilitas dan teorema fuzzy.

Dewasa ini program MYCIN menjadi acuan penting untuk pengembangan sistem pakar secara modern karena di dalamnya telah terintegrasi semua komponen standar yang dibutuhkan oleh sistem pakar itu sendiri.

### **II.3.2 Pengertian Sistem Pakar**

Ada beberapa pengertian tentang sistem pakar, antara lain:

1. Menurut Durkin: Sistem pakar adalah suatu program komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh seorang pakar.
2. Menurut Ignizio: Sistem pakar adalah suatu model dan prosedur yang berkaitan, dalam suatu domain tertentu, yang mana tingkat keahlian dapat dibandingkan dengan keahlian seorang pakar.
3. Menurut Giarratano dan Riley: Sistem pakar adalah suatu sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar.

### **II.3.3 Fitur-Fitur Sistem Pakar**

Sistem pakar harus memiliki fitur berikut:

1. Keahlian: Sebagaimana telah dideskripsikan di atas, pakar dibedakan berdasarkan tingkat keahlian mereka. Sistem pakar harus memiliki keahlian yang akan memungkinkan sistem membuat keputusan tingkat pakar. Sistem harus menampilkan performa pakar dan kekuatan yang cukup.

2. *Pertimbangan simbolik*: Dasar pemikiran kecerdasan tiruan adalah menggunakan pertimbangan simbolik daripada perhitungan matematis. Hal ini juga berlaku untuk sistem pakar. Jadi, pengetahuan harus direpresentasikan secara simbolik, dan mekanisme pertimbangan primer juga harus simbolik. Mekanisme pertimbangan simbolik biasa menyertakan backward chaining dan forward chaining, yang akan dideskripsikan pada bagian selanjutnya.
3. *Deep knowledge* (kedalaman pengetahuan): Berkaitan dengan tingkat keahlian dalam basis pengetahuan. Basis pengetahuan harus berisi pengetahuan kompleks yang tidak mudah diperoleh dari nonpakar.
4. *Self-knowledge*: Sistem pakar harus dapat menganalisa pertimbangannya sendiri dan menjelaskan mengapa dicapai suatu kesimpulan. Kebanyakan pakar memiliki kemampuan pembelajaran yang sangat kuat yang memungkinkan mereka memperbarui secara konstan pengetahuannya. Sistem pakar juga harus dapat belajar dari keberhasilan dan kegagalannya dan dari sumber pengetahuan lain.

Pengembangan sistem pakar dibagi menjadi dua generasi. Kebanyakan sistem pakar generasi pertama menggunakan aturan jika-maka untuk merepresentasikan dan menyimpan pengetahuannya. Sistem pakar generasi kedua jauh lebih fleksibel dalam mengadopsi banyak representasi pengetahuan dan metode pertimbangan. Sistem tersebut dapat mengintegrasikan jaringan dengan inferensi berbasis-aturan untuk mencapai performa keputusan yang lebih tinggi. Tabel dibawah ini menyediakan perbandingan antara sistem konvensional dan sistem pakar.

**Tabel II.1 Fitur Sistem Pakar**

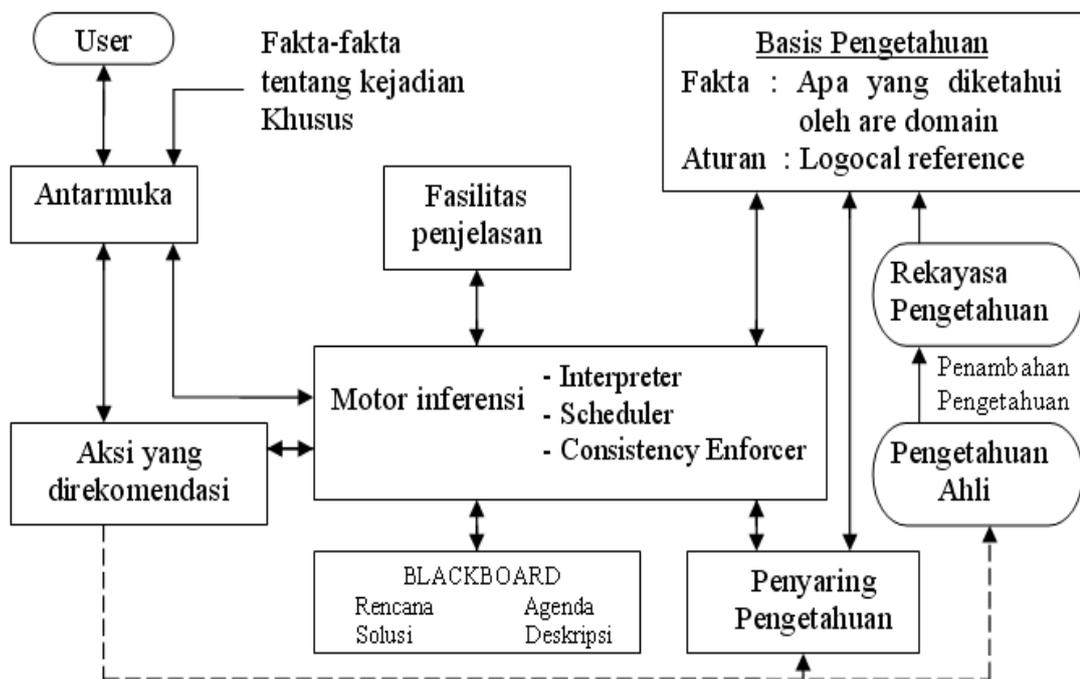
Sistem Konvensional	Sistem Pakar
Informasi dan pengolahannya biasanya digabungkan dalam satu program berurutan.	Basis pengetahuan secara nyata dipisahkan dari mekanisme pengolahan (inferensi) (misalnya, aturan pengetahuan dipisahkan dari kontrol).
Program tidak melakukan kesalahan (programmer atau pengguna yang membuat kesalahan)	Program dapat melakukan kesalahan.
Biasanya tidak menjelaskan mengapa data input diperlukan atau bagaimana kesimpulan dihasilkan.	Penjelasan adalah bagian dari sebagian besar ES.
Memerlukan semua data input. Berfungsi dengan tidak tepat jika ada data yang hilang kecuali jika telah dirancang demikian.	Tidak memerlukan semua fakta awal. Biasanya dapat tiba pada kesimpulan yang masuk akal sekalipun ada fakta yang hilang.
Perubahan dalam program sangat membosankan (kecuali dalam DSS).	Perubahan dalam aturan mudah dilakukan.
Sistem beroperasi hanya jika telah lengkap.	Sistem dapat beroperasi dengan hanya sedikit aturan (seperti dalam prototipe pertama).
Eksekusi dilakukan pada basis algoritma langkah demi langkah.	Eksekusi dilakukan menggunakan heuristik dan logika.
Manipulasi efektif pada database besar.	Manipulasi efektif pada basis pengetahuan besar.
Representasi dan penggunaan data.	Representasi dan penggunaan pengetahuan.
Efisiensi biasanya menjadi tujuan utama. Efektivitas penting hanya untuk DSS.	Efektivitas adalah tujuan utama.
Mudah menangani data kuantitatif	Mudah menangani data kualitatif
Menggunakan representasi data numerik	Menggunakan representasi pengetahuan simbolik dan numerik.
Menyerap, memperbesar, dan mendistribusikan akses ke data atau informasi numerik.	Menyerap, memperbesar, dan mendistribusikan akses ke penilaian atau pengetahuan.

Sumber: Sri Kesuma Dewi (2003: 5)

### II.3.4 Struktur Sistem Pakar

Struktur sistem pakar terdiri dari 2 bagian pokok, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi

(*consultation environment*). Lingkungan pengembangan sebagai pembangun sistem pakar baik dari segi pembangun komponen maupun basis pengetahuan. Lingkungan konsultasi digunakan oleh seorang yang bukan ahli untuk berkonsultasi. Struktur sistem pakar dapat dilihat seperti pada Gambar II.1.



**Gambar II.4 Struktur Sistem Pakar**  
**Sumber : Kristanto Andri (2004 :36)**

Ppenjelasan Gambar II.1 adalah sebagai berikut :

1. Subsistem penambahan pengetahuan. Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Pengetahuan itu bisa berasal dari ahli buku, basisdata, penelitian dan gambar.
2. Basis pengetahuan berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.

3. Motor inferensi (*inference engine*). Program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan dan *blackboard*. Serta digunakan untuk memformulasikan konklusi. Ada 3 elemen utama dalam motor inferensi, yaitu:
  - a. *Interpreter*: Mengeksekusi item-item agenda yang terpilih dengan menggunakan aturan-aturan dalam basis pengetahuan yang sesuai.
  - b. *Scheduler*: Akan mengontrol agenda.
  - c. *Consistency enforcer*: Akan berusaha memelihara kekonsistenan dalam merepresentasikan solusi yang bersifat darurat.
4. *Blackboard*. Merupakan area dalam memori yang digunakan untuk merekam kejadian yang sedang berlangsung termasuk keputusan sementara. Ada 3 tipe keputusan yang dapat direkam, yaitu:
  - a. Rencana: Bagaimana menghadapi masalah.
  - b. Agenda: Aksi-aksi yang potensial yang sedang menunggu untuk dieksekusi.
  - c. Solusi: Calon aksi yang akan dibangkitkan.
5. Antarmuka. Digunakan untuk media komunikasi antara user dan program.
6. Subsistem penjelasan. Digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif melalui pertanyaan:
  - a. Mengapa suatu pertanyaan ditanyakan oleh sistem pakar?
  - b. Bagaimana konklusi dicapai?
  - c. Mengapa ada alternatif yang dibatalkan?

- d. Rencana apa yang digunakan untuk mendapatkan solusi?
7. Sistem penyaring pengetahuan. Sistem ini digunakan untuk mengevaluasi kinerja sistem pakar itu sendiri untuk melihat apakah pengetahuan-pengetahuan yang ada masih cocok untuk digunakan dimasa mendatang.
8. Subsistem penambahan pengetahuan. Bagian ini digunakan untuk memasukkan pengetahuan, mengkonstruksi atau memperluas pengetahuan dalam basis pengetahuan. Pengetahuan itu bisa berasal dari ahli buku, basisdata, penelitian dan gambar.
9. Basis pengetahuan berisi pengetahuan-pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.

### **II.3.5 Bidang-Bidang Masalah yang Cocok untuk Sistem Pakar**

Sistem pakar dapat diklasifikasikan dalam beberapa cara. Satu cara adalah berdasarkan area persoalan umum yang ditanganinya. Misalnya, diagnosa dapat didefenisikan sebagai “menyimpulkan kegagalan sistem dari observasi”. Deskripsi singkat tiap kategori adalah sebagai berikut:

1. Sistem interpretasi menyimpulkan deskripsi situasi dari observasi. Katagori ini termaksud pengawasan, speech understanding, analisis citra, interpretasi sinyal, dan banyak jenis analisis kecerdasan. Sistem interpretasi menjelaskan data yang telah diobservasi dengan menetapkan arti simbolik yang mendeskripsikan situasi.

2. Sistem prediksi menyertakan peramalan cuaca, prediksi demografi, peramalan ekonomi, prediksi lalu lintas, perkiraan panen, dan peramalan militer, pemasaran, dan keuangan.
3. Sistem diagnostik menyertakan diagnosis medis, elektronik, mekanik, dan perangkat lunak. Sistem diagnostik biasanya menghubungkan ketidakteraturan kelakuan kelakuan dengan penyebab dasarnya.
4. Sistem desain mengembangkan konfigurasi objek yang memenuhi batasan persoalan desain. Persoalan tersebut mencakup layout sirkuit, desain bangunan, dan layout pabrik. Sistem desain mengontruksi objek dalam berbagai hubungan satu sama lain dan memverifikasi bahwa konfigurasi ini sesuai dengan batasan yang dinyatakan.
5. Sistem perencanaan fokus pada persoalan perencanaan jangka pendek dan panjang pada area seperti manajemen proyek, routing, komunikasi, pengembangan produk, aplikasi militer, dan perencanaan keuangan.
6. Sistem pengawasan membandingkan observasi kelakuan sistem dengan standar yang tampaknya penting untuk keberhasilan pencapaian tujuan. Fitur penting ini berhubungan dengan kecacatan potensial dalam rencana tersebut. Terdapat banyak sistem pengawasan dengan bantuan komputer untuk topik dari kontrol lalu lintas udara hingga tugas manajemen fiskal.
7. Sistem debugging mengandalkan pada kemampuan perencanaan, desain, prediksi untuk membuat spesifikasi atau rekomendasi untuk membetulkan persoalan diagnosis.

8. Sistem perbaikan mengembangkan dan mengeksekusi rencana untuk mengelola perbaikan persoalan diagnosis tertentu. Sistem tersebut menggabungkan kemampuan debugging, perencanaan, dan eksekusi.
9. Sistem instruksi menggabungkan subsistem diagnosis dan debugging yang secara spesifik menangani kebutuhan siswa. Biasanya, sistem tersebut mulai dengan membangun deskripsi hipotetik tentang pengetahuan siswa yang menginterpretasikan kelakuannya. Sistem kemudian mendiagnosis kelemahan pengetahuan siswa dan mengidentifikasi perbaikan yang diperlukan untuk mengatasi kekurangan tersebut. Akhirnya, sistem merencanakan interaksi tutorial yang dimaksudkan untuk memberikan perbaikan pengetahuan bagi siswa.
10. Sistem kontrol secara adaptif mengatur keseluruhan kelakuan sistem. Untuk melakukan hal ini, sistem kontrol harus berulang kali menginterpretasikan situasi terbaru, memprediksi masa depan, mendiagnosa penyebab persoalan yang terantisipasi, merumuskan rencana pemulihan, dan mengawasi eksekusinya untuk memastikan keberhasilan.

**Tabel II.2 Bidang Sistem Pakar**

Kategori	Persoalan yang Ditangani
Sistem interpretasi	Menyimpulkan deskripsi situasi dari observasi
Sistem prediksi	Menyimpulkan kemungkinan konsekuensi dari suatu situasi
Sistem diagnostik	Menyimpulkan kegagalan sistem dari observasi
Sistem desain	Mengonfigurasi objek dengan batasan
Sistem perencanaan	Mengembangkan rencana untuk mencapai tujuan
Sistem pengawasan	Membandingkan observasi rencana, memunculkan pengecualian
Sistem debugging	Menyarankan pemulihan untuk kegagalan
Sistem perbaikan	Mengeksekusi rencana untuk mengelola pemulihan yang disarankan.
Sistem instruksi	Mendiagnosis, men-debug, dan memperbaiki performa

	siswa
Sistem kontrol	Menginterpretasikan, memprediksi, memperbaiki, dan mengawasi kelakuan sistem

**Sumber : Aziz Farid (2004 : 39)**

### **II.3.6 Manfaat dan Kemampuan Sistem Pakar**

Ribuan sistem pakar saat ini digunakan di hampir setiap industri dan area fungsional. Wong dan Monaco (1995) menyediakan ulasan komprehensif dan analisis literatur tentang sistem pakar dalam bisnis. Eom (1996) melakukan survei komprehensif tentang sekitar 440 sistem pakar operasional dalam bisnis. Surveinya mengungkapkan bahwa banyak *Expert System* (ES) yang memiliki pengaruh sangat besar terutama pada penyusutan waktu pelaksanaan tugas dari hari ke jam, menit, atau detik, dan adanya manfaat sistem tak terhitung antara lain peningkatan kepuasan konsumen, peningkatan kualitas produk dan layanan, pengambilan keputusan yang akurat dan konsisten.

Manfaat ES potensial yang utama yaitu:

#### 1. Meningkatkan Output dan produktivitas

ES dapat bekerja lebih cepat daripada manusia. Misalnya, XCON memungkinkan DEC meningkatkan throughput VAX yang mengkonfigurasi urutan empat kali lipat.

#### 2. Menurunkan waktu pengambilan keputusan

Dengan menggunakan rekomendasi yang disediakan ES, manusia dapat membuat keputusan lebih cepat. Misalnya, perwakilan American Express membuat keputusan persetujuan biaya dalam waktu kurang dari lima detik, dibandingkan dengan sekitar tiga menit sebelum implementasi ES. Sifat ini

penting dalam mendukung garis depan pembuat keputusan yang harus membuat keputusan cepat pada saat berinteraksi dengan pelanggan.

3. Meningkatkan kualitas Proses dan Produk

ES dapat meningkatkan kualitas dengan menyediakan nasihat yang konsisten dan mengurangi ukuran dan tingkat kesalahan. Misalnya, XCON mengurangi tingkat kesalahan dari 35 persen ke 2 persen (pada awalnya, bahkan lebih sedikit lagi pada rilis berikutnya) sehingga meningkatkan kualitas minikomputer.

4. Mengurangi *Downtime*

Banyak ES operasional yang digunakan untuk mendiagnosis kegagalan dan menyarankan perbaikan. Dengan menggunakan ES dimungkinkan mengurangi downtime mesin secara signifikan. Misalnya, pada oil rig. Sistem ini menghemat sejumlah besar uang bagi perusahaan dengan mengurangi downtime secara signifikan.

5. Menyerap Keahlian Langka

Kelangkaan keahlian menjadi nyata dalam situasi dimana tidak terdapat cukup pakar untuk suatu tugas, pakar ada yang akan pensiun atau keluar dari pekerjaan, atau keahlian diperlukan di sejumlah area geografis yang luas. Contoh yang dideskripsikan dalam sketsa pembukaan adalah contoh bagus untuk hal tersebut.

6. Fleksibilitas

ES dapat menawarkan fleksibilitas dalam industri jasa dan manufaktur.

7. Operasi Peralatan yang Lebih Mudah

Sistem pakar membuat peralatan kompleks menjadi lebih mudah dioperasikan. STEAMER adalah ES awal yang ditujukan untuk melatih karyawan yang tidak berpengalaman untuk mengoperasikan mesin kapal kompleks. Contoh lain adalah ES yang dikembangkan untuk Shell Oil Company untuk melatih orang yang menggunakan rutin program komputer kompleks.

#### 8. Eliminasi Kebutuhan Peralatan yang Mahal

Sering manusia harus mengandalkan peralatan mahal untuk mengawasi dan mengontrol. ES dapat menjalankan tugas yang sama dengan instrumen berbiaya rendah. Karena kemampuannya untuk menginvestigasi informasi yang disediakan oleh instrumen dengan lebih cepat dan menyeluruh.

#### 9. Operasi di Lingkungan Berbahaya

Banyak tugas harus dilakukan dalam lingkungan berbahaya. ES memungkinkan manusia menghindari lingkungan tersebut. ES memungkinkan pekerja menghindari lingkungan panas, lembab, atau beracun, misalnya reaktor daya nuklir yang mengalami kegagalan. Fitur ini sangat penting dalam konflik militer.

#### 10. Aksesibilitas ke Pengetahuan dan *Help Desk*

Sistem Pakar menjadikan pengetahuan dapat diakses sehingga membebaskan pakar dari pekerjaan rutin. Orang dapat bertanya pada sistem dan mendapatkan nasihat yang berguna. Satu area penerapan adalah dukungan help desk (Dryden, 1996). Lihat Website Exsys Corporation ([www.exsys.com](http://www.exsys.com)), [www.inference.com](http://www.inference.com), dan Ginesys Corporation ([www.ginesys.com](http://www.ginesys.com)). Lainnya adalah dukungan call center yang saat ini

menggunakan sistem cerdas berbasis-web (Thomas, dkk.,1997; Orzech, 1998) (lihat [www.clarify.com](http://www.clarify.com)).

11. Kemampuan untuk Bekerja dengan Informasi yang Tidak Lengkap/Tidak Pasti Kebalikan dengan sistem komputer konvensional, sistem pakar, seperti pakar manusia, dapat bekerja dengan data, informasi, atau pengetahuan yang tidak lengkap, tidak tepat, dan tidak pasti. Pengguna dapat merespon dengan “don’t know” atau “*not sure*” terhadap satu atau lebih pertanyaan sistem selama konsultasi, dan ES masih akan dapat menghasilkan suatu jawaban, sekalipun mungkin bukan jawaban pasti.

12. Kelengkapan Pelatihan

ES dapat menyediakan pelatihan. Orang yang baru bekerja dengan ES akan semakin berpengalaman. Fasilitas penjelasan dapat pula berfungsi sebagai perangkat pengajaran, karena itu catatan dan penjelasan dapat disisipkan ke dalam basis pengetahuan.

13. Peningkatan Pemecahan Masalah

ES dapat meningkatkan pemecahan masalah dengan memungkinkan integrasi penilaian pakar tingkat atas ke dalam analisa. Misalnya, ES yang disebut *statistical Navigator* dikembangkan untuk membantu orang yang baru belajar menggunakan paket komputer yang kompleks secara statistik.

14. Meningkatkan Proses Pengambilan Keputusan

ES menyediakan umpan balik yang cepat tentang konsekuensi keputusan, memfasilitasi komunikasi antara pembuat keputusan dalam suatu tim, dan memungkinkan respons cepat terhadap perubahan yang tidak terprediksi

dalam lingkungan sehingga menghasilkan pemahaman yang lebih baik pada situasi pengambilan keputusan.

#### 15. Meningkatkan kualitas keputusan

ES dapat diandalkan. ES tidak akan lelah atau merasa bosan, atau sakit, atau mogok, dan membantah bosnya. ES secara konsisten memerhatikan detail dan tidak meremehkan informasi relevan dan solusi potensial, karena itu ES membuat lebih sedikit kesalahan. ES juga menyediakan rekomendasi yang sama untuk persoalan yang berulang.

#### 16. Kemampuan untuk Memecahkan Persoalan Kompleks

Suatu saat ES dapat menjelaskan persoalan kompleks yang solusinya berada di luar kemampuan manusia. Beberapa ES telah dapat memecahkan persoalan di mana lingkup pengetahuan yang diperlukan melebihi orang manapun. Hal ini memungkinkan pembuat keputusan memperoleh kontrol pada situasi yang rumit dan meningkatkan operasi sistem kompleks.

#### 17. Transfer Pengetahuan ke Lokasi Terpencil

Salah satu manfaat potensial ES yang terbesar adalah kemudahan transfernya di sepanjang batas internasional. Contoh transfer tersebut adalah perawatan mata ES (untuk diagnosis dan rekomendasi perawatan) yang dikembangkan di Rutgers University bekerja sama dengan World Health Organization. Program tersebut telah diimplementasikan di Mesir dan Algeria, di mana penyakit mata serius telah menyebar, tetapi spesialis mata sangat langka. Program PC tersebut berbasis aturan dan dapat dioperasikan oleh perawat, asisten dokter,

atau praktisi umum. Web tersebut digunakan secara luas untuk menyebarkan informasi pada pengguna di lokasi terpencil.

#### 18. Peningkatan Sistem Informasi yang Lain

Sistem pakar sering menyediakan kemampuan cerdas untuk sistem informasi lain.

### II.3.7 Keterbatasan Sistem Pakar

Metode ES yang tersedia mungkin tidak langsung dan efektif, bahkan untuk banyak aplikasi dalam kategori umum. Persoalan-persoalan berikut telah memperlambat penyebaran komersial ES:

1. Pengetahuan tidak selalu siap tersedia.
2. Akan sulit mengekstrak keahlian dari manusia.
3. Pendekatan tiap pakar pada suatu penilaian situasi mungkin berbeda tetapi benar.
4. Sulit, bahkan bagi pakar berkemampuan tinggi, untuk mengikhtisarkan penilaian situasi yang baik pada saat berada dalam tekanan waktu.
5. Pengguna sistem pakar memiliki batasan kognitif alami.
6. ES bekerja dengan baik hanya dalam domain pengetahuan sempit.
7. Kebanyakan pakar tidak memiliki sarana mandiri untuk memeriksa apakah kesimpulannya masuk akal.
8. Kosakata atau jargon yang digunakan pakar untuk menyatakan fakta dan hubungan acapkali terbatas dan tidak dipahami oleh pakar lain.
9. Sering sekali dibutuhkan bantuan dari *knowledge engineer* yang langka dan mahal yang merupakan suatu fakta yang menjadikan konstruksi ES mahal.

10. Kurangnya kepercayaan pada bagian pengguna akhir menjadi penghalang penggunaan ES.
11. Transfer pengetahuan adalah subjek terhadap sekumpulan bias perseptual dan penilaian.
12. Sistem pakar mungkin tidak mampu mencapai kesimpulan. Misalnya, XCON yang dikembangkan sepenuhnya pada awalnya tidak dapat memenuhi sekitar dua persen pesanan yang ditujukan padanya. Lebih lagi, sistem pakar, sebagaimana halnya pakar manusia, terkadang menghasilkan rekomendasi yang tidak tepat.

### **II.3.8 Komponen Sistem Pakar**

Komponen sistem pakar terdiri dari Antar Muka Pemakai, Basis Pengetahuan dan Motor Inferensi.

#### **a. Antar Muka Pemakai (*User Interface*)**

Antarmuka Pemakai memberikan fasilitas komunikasi antara pemakai dan sistem, memberikan berbagai fasilitas informasi dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi.

Pada umumnya, antarmuka pemakai juga berfungsi untuk menginputkan pengetahuan baru ke dalam basis pengetahuan sistem pakar, menampilkan fasilitas penjelasan sistem dan memberikan tuntunan penggunaan sistem secara menyeluruh langkah demi langkah sehingga pemakai mengerti apa yang harus dilakukan terhadap sistem.

Syarat utama membangun antarmuka pemakai adalah kemudahan dalam menjalankan sistem. Semua kesulitan dalam membangun suatu program harus disembunyikan, yang ditampilkan hanyalah tampilan yang interaktif, komunitatif dan kemudahan pakai.

**b. Basis Pengetahuan (*Knowledge Base*)**

Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah, tentu saja di dalam domain tertentu. Ada 2 bentuk pendekatan basis pengetahuan yang sangat umum digunakan, yaitu :

a. Penalaran Berbasis Aturan (*Ruled Based Reasoning*).

Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk *IF-THEN*. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan si pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu, bentuk ini juga digunakan apabila dibutuhkan penjelasan tentang jejak (langkah-langkah) dalam Pencapaian solusi yang diharapkan sebelumnya.

b. Penalaran Berbasis Kasus (*Case Based Reasoning*).

Pada penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan diturunkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini digunakan apabila user menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Selain itu, bentuk ini juga digunakan apabila kita telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

### c. Motor Inferensi (*Inference Engine*)

Motor atau mekanisme inferensi adalah bagian dari sistem pakar yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi antar sistem dan pemakai, mekanisme inferensi menguji aturan satu demi satu sampai kondisi aturan itu benar.

## II.3.9. Tahapan Pembuatan Sistem Pakar

Terdapat 6 tahap atau fase dalam pengembangan sistem pakar seperti digambarkan pada gambar. Penjelasan berikut merupakan penjelasan secara garis besar tentang fase-fase pengembangan tersebut.

### 1. Identifikasi

Tahap ini merupakan tahap penentuan hal-hal penting sebagai dasar dari permasalahan yang akan dianalisis. Tahap ini merupakan tahap untuk mengkaji dan membatasi masalah yang akan diimplementasikan dalam sistem. Setiap masalah yang akan diidentifikasi harus dicari solusi, fasilitas yang akan dikembangkan, penentuan jenis bahasa pemrograman dan tujuan yang ingin dicapai dari proses pengembangan tersebut. Apabila proses identifikasi masalah dilakukan dengan benar maka akan dicapai hasil yang optimal.

### 2. Konseptualisasi

Hasil identifikasi masalah dikonseptualisasikan dalam bentuk relasi antar data, hubungan antar pengetahuan dan konsep-konsep penting dan ideal yang akan diterapkan dalam suatu sistem. Konseptualisasi juga menganalisis data-data

penting yang harus didalami bersama dengan pakar di bidang permasalahan tersebut. Hal ini dilakukan untuk memperoleh konfirmasi hasil wawancara dan observasi sehingga hasilnya dapat memberikan jawaban pasti bahwa sasaran permasalahan tepat, benar dan sudah selesai.

### 3. Formalisasi

Apabila tahap konseptualisasi telah selesai dilakukan. Maka ditahap ini formalisasi konsep-konsep tersebut diimplementasikan secara formal, misalnya memberikan kategori sistem yang akan dibangun, mempertimbangkan beberapa faktor pengambilan keputusan seperti keahlian manusia, kesulitan dan tingkat kesulitan yang mungkin terjadi, dokumentasi kerja, dan sebagainya.

### 4. Implementasi

Apabila pengetahuan sudah diformalisasikan secara lengkap, maka tahap implementasi dapat dimulai dengan membuat garis besar masalah kemudian memecahkan masalah ke dalam modul-modul. Untuk memudahkan maka harus diidentifikasi:

- a. Apa saja yang menjadi inputan.
- b. Bagaimana prosesnya digambarkan dalam bagan alur dan basis aturannya.
- c. Apa saja yang menjadi output atau hasil dan kesimpulannya.

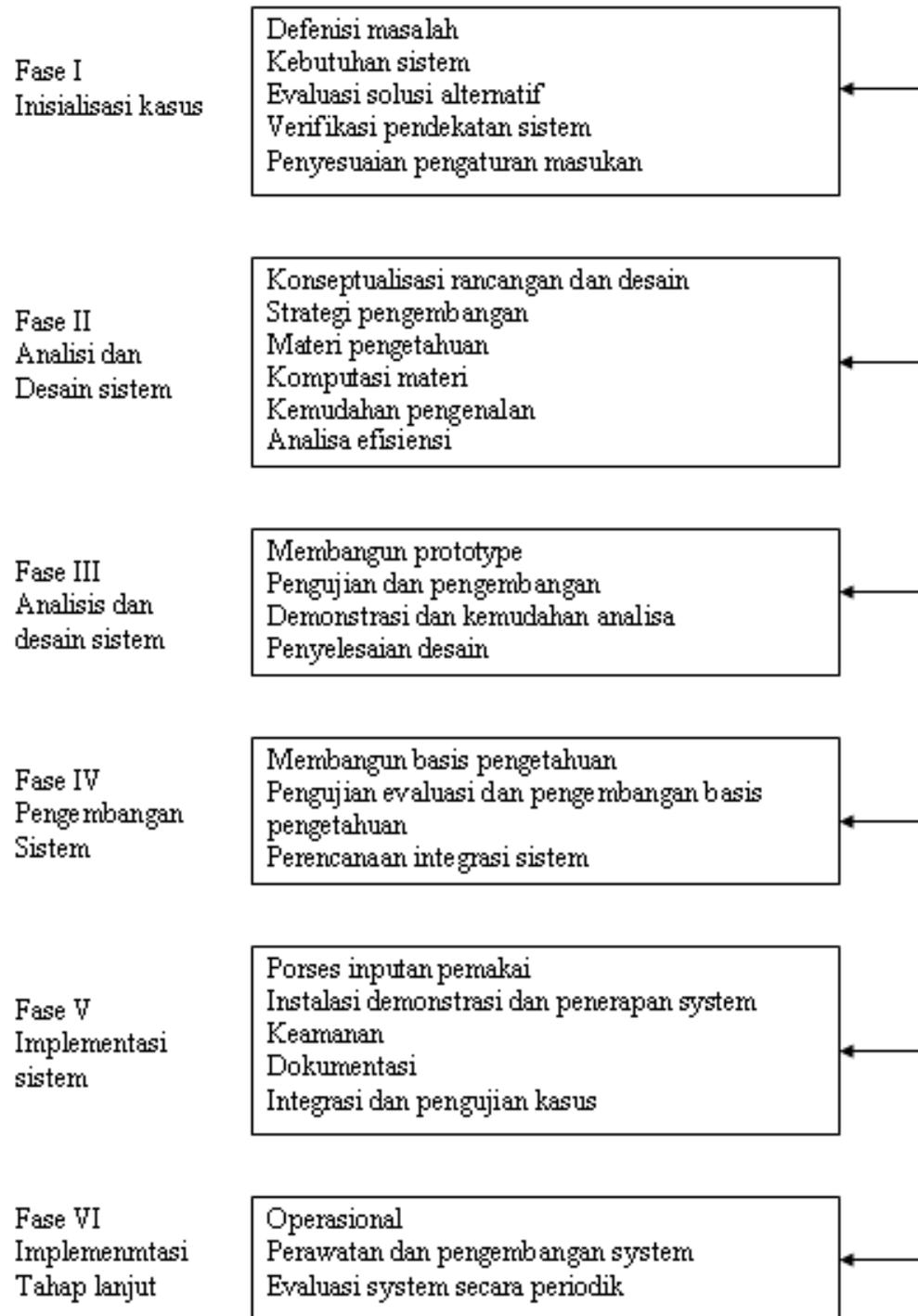
Sesudah itu semuanya diubah dalam bahasa yang mudah dimengerti oleh komputer dengan menggunakan tahapan fase seperti gambaran fase pengembangan sistem pakar

## 5. Evaluasi

Sistem pakar yang selesai dibangun, perlu untuk dievaluasi untuk menguji dan menemukan kesalahannya. Hal ini merupakan yang umum dilakukan karena suatu sistem belum tentu sempurna setelah selesai pembuatannya sehingga proses evaluasi diperlukan untuk penyempurnaannya. Dalam evaluasi akan ditemukan bagian-bagian yang harus dikoreksi untuk menyamakan permasalahan dan tujuan akhir pembuatan sistem.

## 6. Pengembangan sistem

Pengembangan sistem diperlukan sehingga sistem yang dibangun tidak menjadi usang dan investasi sistem tidak sia-sia. Hal ini pengembangan sistem yang paling berguna adalah proses dokumentasi sistem ini dimana di dalamnya tersimpan semua hal penting yang dapat menjadi tolak ukur pengembangan sistem di masa mendatang termasuk di dalamnya adalah kamus pengetahuan masalah yang diselesaikan.



**Gambar II.5 Tahap Pembuatan Sistem Pakar**

**Sumber : Aziz Farid (2004 : 179)**

## II.4. Representasi Pengetahuan

Pada bagian sebelumnya kita telah dapat merepresentasikan suatu masalah dalam ruang keadaan. Dalam menyelesaikan masalah tersebut, tentu saja dibutuhkan pengetahuan yang cukup. Tidak hanya itu, sistem juga harus memiliki kemampuan untuk menalar. Basis pengetahuan dan kemampuan untuk melakukan penalaran merupakan bagian terpenting dari sistem yang menggunakan kecerdasan buatan. Meskipun suatu sistem memiliki banyak pengetahuan, namun tidak memiliki kemampuan untuk menalar, tentu akan menjadi percuma saja. Demikian pula sebaliknya, apabila suatu sistem memiliki kemampuan yang sangat handal untuk menalar, namun basis pengetahuan yang dimilikinya tidak cukup, maka solusi yang diperolehpun menjadi tidak maksimal.

### II.4.1 Logika

Logika adalah bentuk representasi pengetahuan yang paling tua. Pada dasarnya proses logika adalah proses membentuk kesimpulan atau menarik suatu inferensi berdasarkan fakta yang telah ada. Input dari proses logika berupa premis atau fakta-fakta yang diakui kebenarannya sehingga dengan melakukan penalaran pada proses logika dapat dibentuk suatu inferensi atau kesimpulan yang benar pula.



**Gambar II.6 Proses Logika**  
**Sumber : Aziz Farid (2004 : 181)**

Ada 2 penalaran yang dapat dilakukan untuk mendapat konklusi:

1. Penalaran Deduktif. Penalaran dimulai dari prinsip umum untuk mendapatkan konklusi yang lebih khusus. Contoh:

Premis mayor : Jika hujan turun saya tidak akan berangkat kuliah.

Premis minor : Hari ini hujan turun.

Konklusi : Hari ini saya tidak akan berangkat kuliah.

2. Penalaran Induktif. Penalaran dimulai dari fakta-fakta khusus untuk mendapatkan kesimpulan umum. Contoh:

Premis -1 : Aljabar adalah pelajaran yang sulit.

Premis -2 : Geometri adalah pelajaran yang sulit.

Premis -3 : Kalkulus adalah pelajaran yang sulit.

Konklusi : Matematika adalah pelajaran yang sulit.

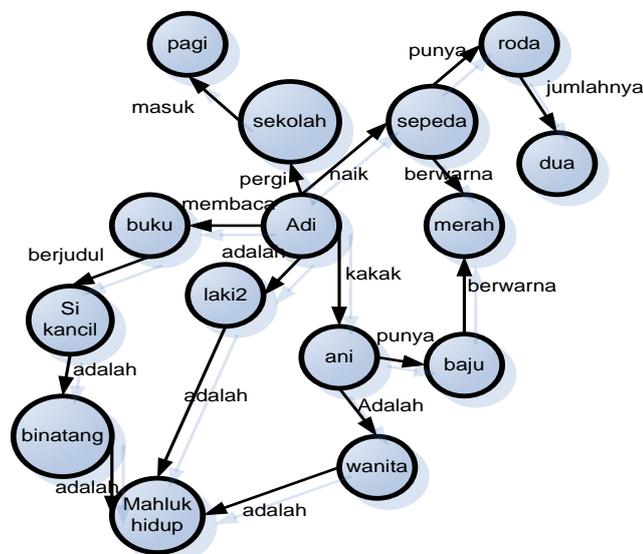
Pada penalaran induktif ini, munculnya premis baru bisa mengakibatkan gugurnya konklusi yang sudah diperoleh. Sebagai contoh, misalkan muncul premis-4 pada contoh di atas:

Premis -4 : Optika adalah pelajaran yang sulit.

Premis tersebut, menyebabkan konklusi: "Matematika adalah pelajaran yang sulit", menjadi salah. Hal itu disebabkan 'Optika' bukan merupakan bagian dari 'Matematika'. Sehingga apabila digunakan penalaran induktif, sangat dimungkinkan adanya ketidakpastian.

## II.4.2 Jaringan Semantik

Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan obyek dan informasi tentang obyek-obyek tersebut. Objek disini berupa benda atau peristiwa. Antara 2 obyek dihubungkan oleh arc yang menunjukkan hubungan antar objek. Gambar II.7 merupakan contoh representasi pengetahuan dengan menggunakan jaringan semantik.

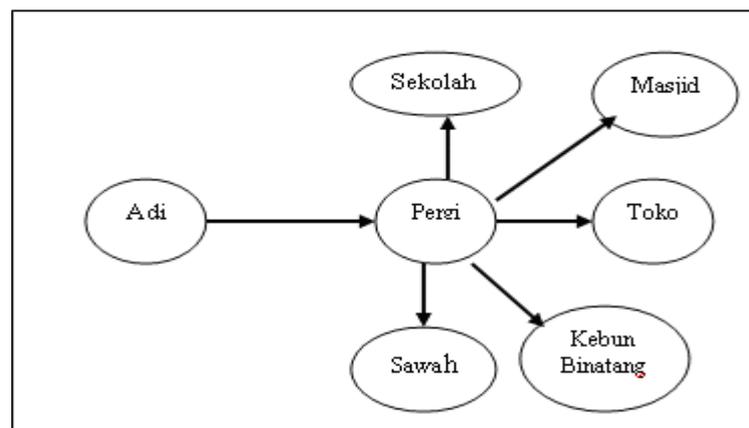


**Gambar II.7 Contoh Jaringan Semantik**

**Sumber : Aziz Farid (2004 : 185)**

Salah satu kelebihan dari jaringan semantik adalah bisa mewariskan. Sebagai contoh pada gambar 2.9 ada garis yang menghubungkan antara adi dengan laki-laki, dan laki-laki mahluk hidup, sehingga apabila ada pertanyaan: Apakah Adi mahluk hidup? Maka kita bisa merunut garis dari mahluk hidup, kemudian ke laki-laki, dan akhirnya ke Adi, sehingga terbukti bahwa Adi adalah mahluk hidup.

Sistem jaringan semantik ini selalu tergantung pada jenis masalah yang akan dipecahkan. Jika masalah itu bersifat umum, maka hanya memerlukan sedikit rincian. Jika ternyata masalah itu banyak melibatkan hal-hal lain, maka di dalam jaringan awalnya diperlukan penjelasan yang lebih rinci lagi. Gambar II.8 menunjukkan rincian dari node awal Adi, apabila Adi hendak pergi ke berbagai tempat. Node Adi dihubungkan dengan node baru, yaitu pergi.

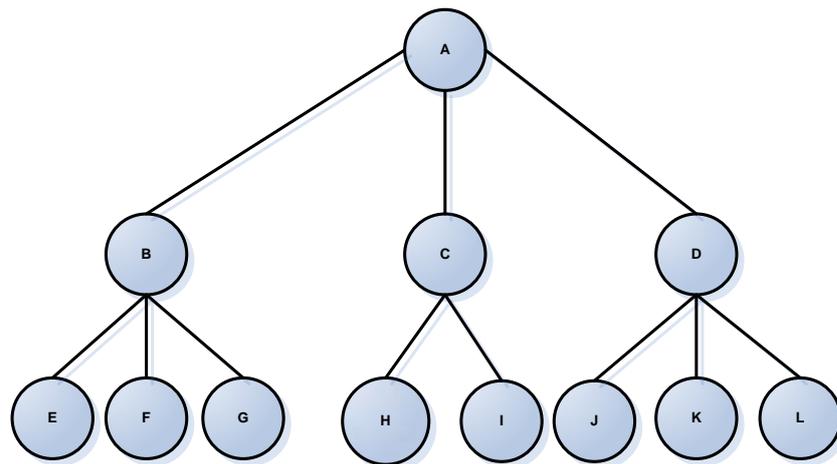


**Gambar II.8 Perluasan Jaringan Semantik**

**Sumber : Aziz Farid (2004 : 186)**

### II.4.3 Pohon

Pohon merupakan struktur penggambaran pohon secara hirarkis. Struktur pohon yang terdiri dari node-node yang menunjukkan objek, dan busur (arc) yang menunjukkan hubungan antar obyek (Gambar II.9).

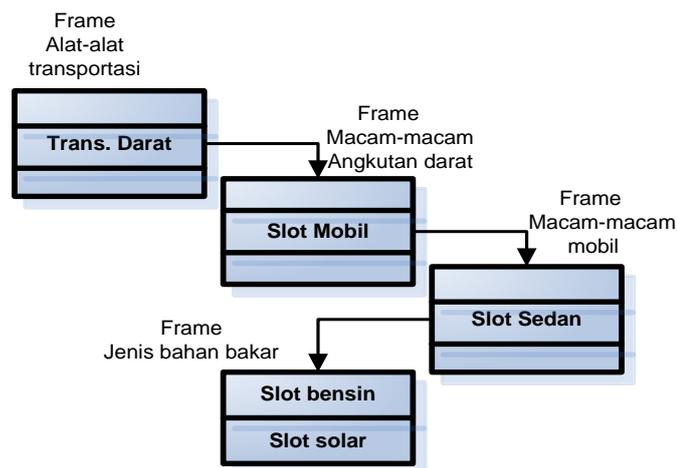


**Gambar II.9 Struktur Pohon**

**Sumber : Aziz Farid (2004 : 183)**

#### **II.4.4 Frame**

Frame merupakan kumpulan pengetahuan tentang suatu obyek tertentu, peristiwa, lokasi, situasi dan lain lain. Frame memiliki slot yang menggambarkan rincian, atribut dan karakteristik obyek. Frame biasanya digunakan untuk mempresentasikan pengetahuan yang didasarkan pada karakteristik yang sudah dikenal, yang merupakan pengalaman-pengalaman. Dengan menggunakan frame, sangatlah mudah untuk membuat inferensi obyek, peristiwa atau situasi baru, karena frame menyediakan basis pengetahuan yang ditarik dari pengalaman.



**Gambar II.10 Frame**

**Sumber : Aziz Farid (2004 : 189)**

Pada Gambar II.10 menunjukkan alat-alat transportasi. Frame tersebut memiliki 3 slot, yaitu alat-alat transportasi di udara, di darat, dan di laut. Ada beberapa slot yang bernilai tetap dan pula tidak tetap (prosedural). Slot yang bernilai tetap misalkan jumlah roda pada sedan (4). Jenis slot lainnya bersifat prosedural, artinya slot yang memungkinkan penambahan informasi baru yang bisa ditambahkan pada aturan IF. Misalnya informasi tentang kecepatan perjalanan, pengisian tangki bahan bakar, atau pemakaian bahan bakar tiap KM.

#### **II.4.5 Naskah (*Script*)**

*Script* adalah skema representasi pengetahuan yang sama dengan frame, yaitu merepresentasikan pengetahuan berdasarkan karakteristik yang sudah dikenal sebagai pengalaman-pengalaman. Perbedaannya, frame menggambarkan obyek sedangkan *script* menggambarkan urutan peristiwa. Dalam menggambarkan urutan peristiwa, *script* menggunakan slot yang berisi informasi

tentang orang, obyek, dan tindakan-tindakan yang terjadi dalam suatu peristiwa.

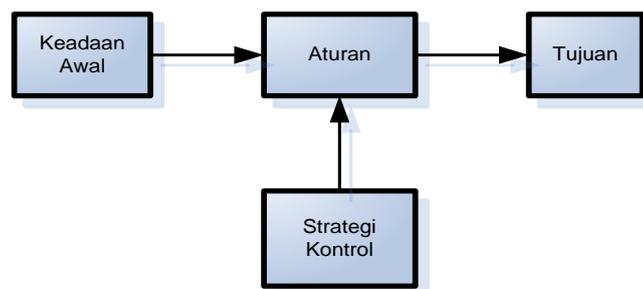
Elemen-elemen script meliputi:

1. Kondisi input, yaitu kondisi yang harus dipenuhi sebelum terjadi atau berlaku suatu peristiwa dalam script.
2. *Track*, yaitu variasi yang mungkin terjadi dalam suatu script.
3. *Prop*, yaitu berisi obyek-obyek pendukung yang digunakan selama peristiwa terjadi.
4. *Scene*, yaitu adegan yang dimainkan yang menjadi bagian dari suatu peristiwa.
5. *Role*, yaitu peran yang dimainkan oleh seseorang dalam peristiwa.
6. Hasil, yaitu kondisi yang ada setelah urutan peristiwa dalam *script* terjadi.

#### II.4.6 Sistem Produksi

Sistem produksi secara umum terdiri dari komponen-komponen sebagai berikut (Gambar II.11):

1. Ruang keadaan, yang berisi keadaan awal, tujuan dan kumpulan aturan yang digunakan untuk mencapai tujuan.
2. Strategi Kontrol, yang berguna untuk mengarahkan bagaimana proses pencarian akan berlangsung dan mengendalikan arah eksplorasi.



**Gambar II.11 Sistem Produksi**

**Sumber : Aziz Farid (2004 : 193)**

Aturan menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Aturan produksi dituliskan dalam bentuk jika-maka (*if-then*). Aturan *if-then* menghubungkan anteseden (*antecedent*) dengan konsekuensi yang diakibatkannya. Berbagai struktur aturan *if-then* yang menghubungkan obyek atau atribut adalah sebagai berikut :

JIKA premis MAKA konklusi

JIKA masukan MAKA keluaran

JIKA kondisi MAKA tindakan

JIKA anteseden MAKA konsekuen

JIKA data MAKA hasil

JIKA tindakan MAKA tujuan

Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Masukan mengacu pada data yang harus tersedia sebelum keluaran dapat diperoleh. Kondisi mengacu pada keadaan yang harus berlaku sebelum tindakan dapat diambil. Anteseden mengacu pada situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan. Tindakan mengacu pada kegiatan yang harus dilakukan sebelum hasil dapat diharapkan (Hanifah,1998).

Aturan dapat diklasifikasikan menjadi dua, yaitu aturan derajat pertama (*first order rule*) dan aturan meta (*meta rule*). Aturan derajat pertama adalah aturan sederhana yang terdiri dari anteseden dan konsekuen misalnya:

JIKA bersin-bersin dan pusing

MAKA terserang penyakit flu

Aturan meta adalah aturan yang anteseden atau konsekuennya mengandung informasi tentang aturan yang lain.

Misalnya:

Aturan I:

JIKA mengalami kehilangan kesadaran yang berlangsung sangat singkat, sehingga aktivitas yang sedang berjalan terhenti.

DAN terkadang disertai dengan mata yang menatap kosong atau gerakan mioklonik dari sekelompok otot mata atau wajah, otomatisme, kehilangan tonus otot (sehingga barang yang dipegang bisa terjatuh atau bila sedang berdiri bisa terjatuh)

DAN serangan berakhir dengan diikuti oleh pulihnya kesadaran

DAN berlangsung beberapa detik sampai setengah menit, dan dapat berlangsung puluhan kali dalam sehari

MAKA mengalami tipe sawan lena

Aturan 2:

JIKA tipe sawan umum

ATAU tipe sawan mioklonik

ATAU tipe sawan lena

ATAU tipe sawan tonik-klonik

DAN EEG tidak menunjukkan adanya kelainan fokal

DAN penyebab tidak diketahui

DAN awitan berhubungan dengan usia

MAKA terkena epilepsi idiopatik umum.

Dalam contoh di atas, konsekuen dari aturan 1, yaitu mengalami sawan lena, menjadi anteseden bagi aturan 2.

Ada beberapa ciri-ciri dari sistem produksi yang menjadikannya berbentuk lebih khusus dan menarik dari representasi pengetahuan untuk sistem pakar, yaitu:

### 1. *Expressiveness dan Intuitiveness*

Pengalaman dalam pekerjaan keahlian manusia mengindikasikan pengulangan berat dari tema ” *well, in the case of so on and so I usually do such and such* ”. Tema ini memetakan keadaan alami dalam bentuk IF...THEN dari aturan produksi. Aturan-aturan produksi secara esensial menyatakan kepada kita bahwa apa yang dilakukan dalam situasi tertentu. Karena banyak sistem pakar yang diorganisasikan berkaitan dengan nasihat/saran apa yang harus dilakukan, maka sifat dari sistem produksi ini adalah merupakan suatu hal khusus secara alami representasi pengetahuan.

### 2. *Simplicity*

Struktur seragam dari sintaks IF...THEN dalam sistem berbasis pengetahuan memberikan suatu kesederhanaan yang menarik untuk representasi pengetahuan. Ciri ini meningkatkan suatu keadaan yang dapat dibaca dari aturan produksi dan komunikasi antara berbagai bagian dari program tunggal. Aturan produksi dengan sintaksnya yang tepat mengarah kepada pendokumentasian diri.

### 3. *Modularity dan Modifiability*

Dengan kealamiannya, aturan produksi mengkodekan bentuk diskret informasi yang secara umum tidak berhubungan dengan aturan produksi yang lain,

kecuali jika ada suatu aturan produksi eksplisit yang menghubungkan mereka. Informasi dapat diperlakukan sebagai suatu kumpulan fakta yang independent yang mana boleh ditambahkan atau dihapus dari sistem dengan secara esensial tidak mengakibatkan "side effect" (efek sampingan) yang mengganggu. Ciri modular sistem produksi menunjukkan kenaikan perbaikan dan setelan yang bagus dari sistem produksi dengan tidak mengalami pengurangan kinerja.

#### 4. Pengetahuan Intensif (*Knowledge Intensive*)

Catatan bahwa tiga bagian struktur dari interpretasi aturan, basis pengetahuan, dan memori kerja (*working memory*) memberikan suatu pemisahan yang efektif dari basis pengetahuan yang berasal dari interpretasi aturan atau mesin pengetahuan yang berasal dari interpretasi aturan atau mesin inferensi. Jadi mesin inferensi menjadi tujuan umum dan kerjasama yang efektif, secara prinsip, pada berbagai macam basis pengetahuan. Basis pengetahuan disusun dari aturan produksi, yang pada gilirannya yang sangat utama adalah "pengetahuan murni" (*pure knowledge*) karena membutuhkan isi, bukan kontrol atau informasi pemrograman. Karena setiap aturan produksi ekuivalen untuk suatu ringkasan dan kejelasan kalimatnya maka masalah semantik diselesaikan dengan struktur representasi.

## II.5. Rule Based Expert System

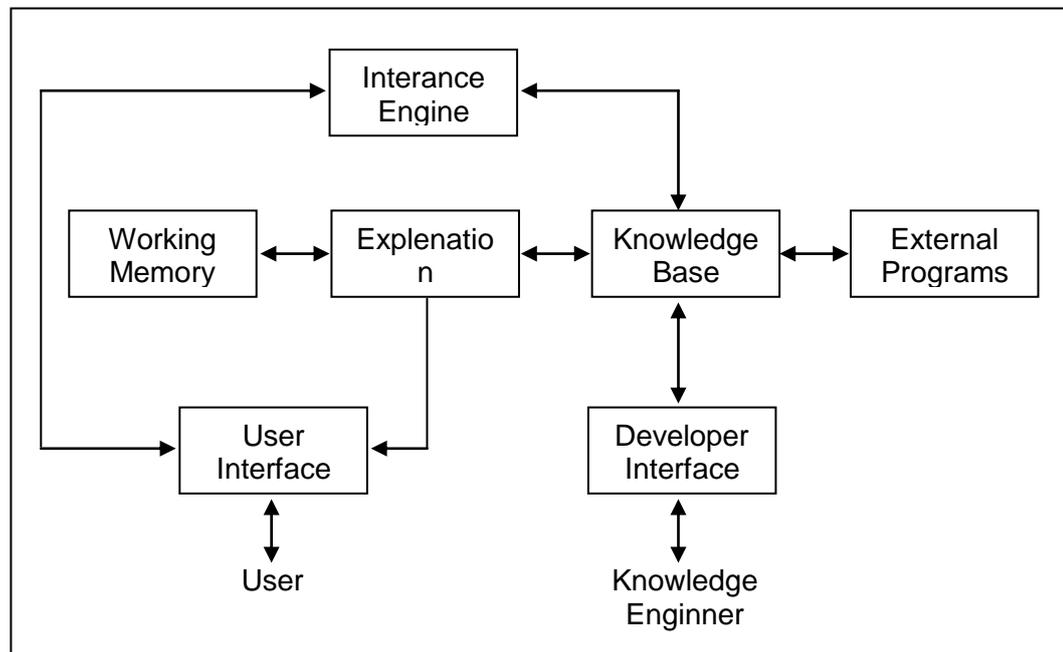
Sistem pakar berbasis aturan (*rule based expert system*) adalah system pakar yang menggunakan kaidah atau aturan (*rule*) untuk mempresentasikan

pengetahuan di dalam basis pengetahuannya. Suatu Rule terdiri atas 2 bagian, yaitu :

1. *Antecedent*, yaitu bagian yang mengekspresikan situasi atau *premise* (Pernyataan berawalan IF).
2. *Qonsequen*, yaitu bagian yang menyatakan suatu tindakan tertentu atau konklusi yang diterapkan jika situasi atau *premise* bernilai benar (Pernyataan berawalan THEN). Misalnya : IF hari mendung THEN akan turun hujan.

Konsekuensi atau konklusi pada bagian THEN akan dinyatakan benar jika bagian IF pada sistem tersebut juga benar atau sesuai dengan aturan tertentu..

Berikut ini merupakan gambar arsitektur sistem berbasis aturan.



**Gambar II.12 Arsitektur Sistem Berbasis Aturan**  
**Sumber: Sri Kesuma Dewi (2003: 5)**

**Keterangan:**

## a. Knowledge Base

Reperensi pengetahuan dari seseorang atau beberapa pakar yang diperlukan untuk memahami, memformulasikan dan memecahkan masalah. Dalam hal ini digunakan untuk memecahkan masalah-masalah yang terjadi pada komputer. Knowledge base ini terdiri dari dua elemen dasar, yaitu fakta dan rules.

## b. Inference Engine

Merupakan otak dari permasalahan sistem pakar yang mengandung mekanisme fungsi berpikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini yang menganalisis suatu masalah tertentu dan kemudian mencari solusi atau kesimpulan yang terbaik.

## c. Working Memory

Merupakan tempat penyimpanan fakta-fakta yang diketahui dari hasil menjawab pertanyaan.

## d. User /Developer Interface

Semua software pengembangan sistem pakar memberikan interface yang berbeda bagi user dan developer. User akan berhadapan dengan tampilan yang sederhana dan mudah, sedangkan developer akan berhadapan dengan editor dan *source code* waktu mengembangkan program. Pada bagian ini user bisa melihat dan berinteraksi dengan sistem. Biasanya dalam bentuk display teks ataupun grafik yang interaktif.

## e. Explanation Facility

Merupakan subsistem yang bertanggung jawab untuk menyediakan *explanation* (penjelasan) dari proses *reasoning* sistem yang memberikan penjelasan saat mana user mengetahui apakah alasan yang diberikan sebuah solusi.

f. External Programs

Berbagai program yang berfungsi sebagai pendukung dalam pembuatan sistem pakar seperti database, spreadsheets, algorithms dan yang lainnya.

Adapun keunggulan sistem pakar berbasis aturan, yaitu :

- a. Meningkatkan output dan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari manusia.
- b. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
- c. Mampu menangkap kepakaran yang sangat terbatas.
- d. Dapat beroperasi di lingkungan yang berbahaya.
- e. Memudahkan akses ke pengetahuan.
- f. Handal, sistem pakar tidak pernah menjadi bosan dan kelelahan atau sakit. Sistem pakar juga secara konsisten melihat semua detil dan tidak akan melewatkan informasi yang relevan dan solusi yang potensial.
- g. Meningkatkan kapabilitas sistem terkomputerisasi yang lain. Integrasi sistem pakar dengan sistem komputer lain membuat lebih efektif, dan mencakup lebih banyak aplikasi.
- h. Mampu bekerja dengan informasi yang tidak lengkap atau tidak pasti. Berbeda dengan sistem komputer konvensional, sistem pakar dapat bekerja dengan

informasi yang tidak lengkap. Pengguna dapat merespon dengan “tidak tahu atau “tidak yakin” pada satu atau lebih pertanyaan selama konsultasi, dan sistem pakar akan tetap memberikan jawabannya.

- i. Mampu menyediakan pelatihan bagi pengguna pemula yang bekerja dengan sistem pakar dan akan memberikan pengalaman. Fasilitas penjelasan akan berfungsi sebagai guru.
- j. Meningkatkan kemampuan problem solving, karena mengambil sumber pengetahuan dari banyak pakar.
- k. Meniadakan kebutuhan perangkat yang mahal.
- l. Akan bersifat fleksibel.
- m. Ekspresi yang alamiah (*natural*)
- n. Bagian pengendalian yang terpisah dengan pengetahuan.
- o. Mudah melakukan ekspansi sistem.
- p. Menggunakan pengetahuan yang relevan.
- q. Dapat menggunakan pengetahuan heuristik.
- r. Dapat menggunakan pengetahuan yang *uncertainty*.

Sedangkan kelemahan sistem pakar adalah :

- a. Pengetahuan yang hendak diambil tidak selalu tersedia.
- b. Kepakaran sangat sulit diekstrak dari manusia.
- c. Pendekatan oleh setiap pakar untuk setiap situasi atau problem bisa berbeda-beda, meskipun sama-sama benar.
- d. 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 mengecek apakah kesimpulan mereka benar dan masuk akal.
- h. Istilah dan jargon yang dipakai oleh pakar dalam mengekspresikan fakta seringkali terbatas dan tidak mudah dimengertai oleh orang lain.
- i. Pengembangan sistem pakar seringkali membutuhkan perekayasaan pengetahuan (*knowledge engineer*) yang langka dan mahal.
- j. Kurangnya rasa percaya pengguna menghalangi pemakaian sistem pakar.
- k. Transfer pengetahuan dapat bersifat subyektif dan bias.

## **II.6. Forward Chaining dan Backward Chaining**

Forward chaining adalah suatu metode penyelesaian masalah yang digunakan untuk mendapatkan solusi dari suatu problem berdasarkan kondisi yang ada, atau suatu proses yang memulai pencarian dari premis atau data menuju pada konklusi (data-driven). Cara kerjanya adalah inference engine menyalakan atau memilih *rule-rule* dimana bagian premis-nya cocok dengan informasi yang ada pada bagian *working memory*.

Gambar berikut ini menunjukkan bagaimana cara kerja Metode Forward Chaining.



kesalahan pemakaian peralatan dan mengemudi kendaraan curian. Selanjutnya Anda menguji fakta yang mendukung hipotesis lanjut ini. Apakah botol minuman yang Anda buang keluar melalui jendela, kecepatan 100 dalam 30 mil perjam dalam zona cepat, lampu belakang yang rusak atau plat mobil yang Anda gunakan mengidentifikasi mobil yang Anda kemudi adalah mobil curian? Dalam kasus ini setiap bagian dari fakta mendukung hipotesis lanjut dan bisa jadi semuanya benar. Sembarang atau semua hipotesis lanjut ini merupakan alasan yang mungkin untuk membuktikan hipotesis kerja yang diinginkan polisi.

Hal ini akan sangat menolong Anda dalam memvisualisasikan forward dan backward chaining dalam istilah suatu path melalui suatu ruang permasalahan, dimana keadaan lanjut berhubungan dengan hipotesis lanjut bagi backward chaining atau konklusi lanjut untuk forward chaining. Hal ini tentu saja memungkinkan untuk melakukan diagnosis dalam sistem forward chaining dan perencanaan dalam backward chaining. Secara khusus penjelasan (*explanation*) dimudahkan dalam backward chaining karena sistem ini dapat dengan mudah menjelaskan secara tepat tujuan apa yang dicoba untuk dipenuhi. Dalam forward chaining penjelasan tidak dimudahkan karena sub tujuannya tidak secara eksplisit diketahui sehingga ditemukan.

**Tabel II.3 Beberapa karakteristik Forward dan Backward chaining**

<b>Forward chaining</b>	<b>Backward chaining</b>
Perencanaan monitoring, kontrol	Diagnosis
Disajikan untuk masa depan	Disajikan untuk masa lalu
Antecedent ke konsekuen	Konsekuen ke antecedent
Data memandu, penalaran dari bawah ke atas	Tujuan memandu, penalaran dari atas ke bawah

Bekerja ke depan untuk mendapatkan solusi apa yang mengikuti fakta	Bekerja ke belakang untuk mendapatkan fakta yang mendukung hipotesis
<i>Breadth first search</i> dimudahkan	<i>Depth first search</i> dimudahkan
<i>Antecedent</i> menentukan pencarian	<i>Konsequent</i> menentukan pencarian
Penjelasan tidak difasilitasi	Penjelasan difasilitasi

**Sumber: Sri Kesuma Dewi (2003: 5)**

## II.7. Konsep Dasar Bahasa Pemrograman

### II.7.1. Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 merupakan bahasa pemrograman berbasis GUI (Graphical User Interface) atau program yang memungkinkan pemakai komputer berkomunikasi dengan komputer tersebut dengan menggunakan modus grafik atau gambar, disebut juga dengan *Event-driven programming* (Program Terkendali Kejadian), artinya program menunggu sampai adanya respon dari pemakai berupa kejadian tertentu, seperti : tombol, diklik, menu dipilih, dan lain-lain.

Microsoft Visual Basic 6.0 berawal dari bahasa BASIC (*Begginers All-Purpose Symbolic Instruction Code*), di DartMounth Collage, Amerika Serikat, pada awal tahun 1960-an. Sistem tersebut sekarang sudah jarang digunakan. Di era Windows, Microsoft menciptakan Visual Basic yang terus mengalami penyempurnaan hingga terciptanya Microsoft Visual Basic 6.0 ini.

Visual Basic pada dasarnya adalah bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah atau intruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu.

Sejak dikembangkannya pada tahun 80-an, Visual Basic telah mencapai versinya yang ke-6. Visual basic adalah juga merupakan sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis windows.

Visual Basic juga merupakan event-driven programming (pemrograman kendali kejadian) artinya kejadian menunggu sampai adanya respon dari pemakai yaitu berupa event/kejadian tertentu (tombol diklik, menu dipilih dan lain-lain). Versi Microsoft Visual Basic versi 6.0 yaitu dari hasil pengembangan versi sebelumnya.

## II.7.2. Mengenal Elemen Visual Basic

### 1. Menu Bar

Menu Bar pada Visual Basic ini terdiri dari beberapa pilihan yang berfungsi untuk melakukan tugas dan kegiatan tertentu. Menu Bar tersebut terdiri dari File, Edit, View, Project, Format, Debug, run, Query, diagram, Tools, Add-Ins, Window dan Help.



**Gambar II.14 Tampilan Menu Bar Microsoft Visual Basic**  
**Sumber: Adi Kurniadi (2000: 12)**

**Tabel II.4 Menu Bar dan fungsinya**

Nama	Keterangan
File	Kelompok perintah yang berfungsi untuk mengatur suatu file, seperti New, Save, Open dan sebagainya.
Edit	Kelompok perintah untuk pengeditan, baik pengeditan objek, komponen maupun kode pada Code Editor.

	Contohnya Cut, Paste, Align.
<u>V</u> iew	Perintah-perintah untuk mengaktifkan bagian-bagian dari Integrated Development environment VB6.
<u>P</u> roject	Berfungsi untuk manajemen proyek serta pendukungnya.
<u>F</u> ormat	Kelompok perintah untuk melakukan proses format.
<u>D</u> ebug	Untuk pengaturan dalam pencarian kesalahan.
<u>R</u> un	Berfungsi untuk menangani proses kompilasi program, seperti Ran, build, Step Over, Debug dan seterusnya.
<u>Q</u> uery	Kumpulan perintah untuk mengakses data yang di perlukan pada Aplikasi Database. Menu ini adalah menu bar pada versi 6.0.
<u>D</u> iagram	Menyediakan perintah untuk pengaturan diagram. Diagram merupakan menu baru versi 6.0 ini sangat menolong dalam proses perencanaan.
<u>T</u> ools	Sebagai penyediaan perlengkapan tambahan program.
<u>A</u> dd_Ins	Untuk pengaturan program tambahan bagi VB 6.0.
<u>W</u> indow	Pengaturan Windows yang sedang terbuka.
<u>H</u> elp	Menyediakan informasi untuk menolong pemakai dalam menggunakan VB 6.0.

**Sumber: Adi Kurniadi (2000: 12)**

## 2. Control Menu

Control Menu merupakan salah satu bagian dalam baris judul (*Title Bar*) yang di gunakan untuk memanipulasi jendela Visual Basic. Control Menu ini berada di pojok kiri atas dari jendela Visual Basic. Adapun isi dari Control Menu adalah sebagai berikut:

- a. Restore : Untuk mengubah ukuran jendela keukuran sebelumnya.
- b. Move : Untuk memindahkan letak jendela.
- c. Size : Untuk mengubah ukuran jendela.

- d. Minimize : Untuk meminimalkan ukuran jendela.
- e. Maximize : Untuk memaksimalkan ukuran jendela.
- f. Close : Untuk menutup jendela.

### 3. Toolbar

Toolbar merupakan cara cepat didalam menjalankan suatu perintah, karena tombol-tombol yang berada dalam Toolbar tersebut mewakili suatu perintah tertentu. Toolbar pada Visual Basic terdiri dari Toolbar Standard, Toolbar Debug, Toolbar Edit dan sebagainya. Toolbar Standard merupakan salah satu Toolbar yang aktif pada saat menjalankan Visual Basic. Toolbar tersebut adalah sebagai berikut:



**Gambar II.15 Tampilan Toolbar Standard pada Visual Basic**

**Sumber: Adi Kurniadi (2000: 12)**

**Tabel II.5 Toolbar dan fungsinya**

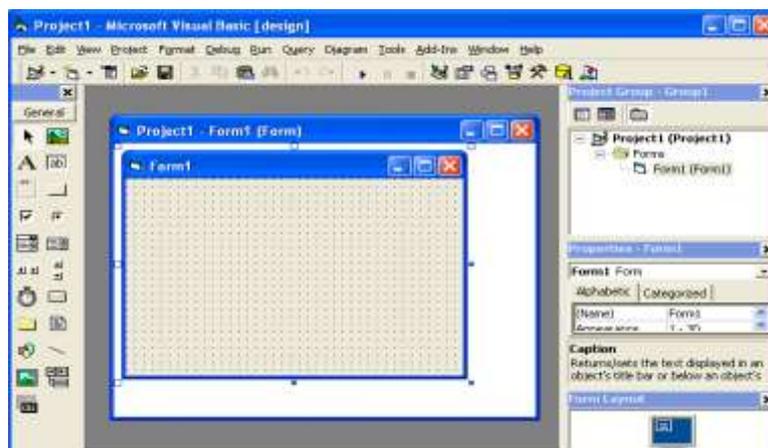
<b>Tombol</b>	<b>Nama</b>	<b>Fungsi</b>
	Add Project	Menambahkan Project baru pada project yang ada.
	Add Item	Menambahkan komponen atau object kedalam form.
	Menu Editor	Menampilkan menu editor untuk merubah tampilan menu
	Open Project	Mengaktifkan/membuka project yang pernah di buat
	Save Project	Menyimpan project yang sudah di buat pada komputer
	Cut	Memotong elemen yang di pilih pada layar, umumnya untuk di pindahkan
	Copy	Menduplikat/copy elemen yang dipilih

	Paste	Menyalin elemen yang di pilih dengan perintah Cut atau Copy
	Paste	Menyalin elemen yang di pilih dengan perintah Cut atau Copy
	Undo	Mengembalikan perintah terakhir
	Redo	Mengembalikan perintah yang di lakukan dengan Undo
	Start	Menjalankan project yang di buat dalam Visual Basic
	Break	Menghentikan sementara project yang di jalankan
	End	Mengakhiri/menghentikan running program
	Project Expoler	Menampilkan/mengaktifkan jendela project Explorer
	Propetis window	Menampilkan/mengaktifkan jendela properties
	From Layout window	Menampilkan/mengaktifkan jendela Form Layout
	Object Browser	Menampilkan/mengaktifkan jendela Object Browser.
	Toolbox	Menampilkan/mengaktifkan jendela Toolbox
	Data View window	Menampilkan/mengaktifkan jendela Data View
	Visual Component Manager	Menampilkan/mengaktifkan jendela View Component Manager

**Sumber: Adi Kurniadi (2000: 12)**

#### 4. Jendela Form

Jendela Form (Form Window) merupakan jendela utama dalam bekerja dengan Visual Basic. Karena pada Form ini akan menempatkan beberapa objek, misalnya label atau teks, gambar, tombol-tombol perintah dan sebagainya. Semua objek yang ditempatkan pada form akan dijalankan atau di tampilkan pada layar Window.



**Gambar II.16 Tampilan Jendela Form**  
**Sumber: Adi Kurniadi (2000: 12)**

#### 5. Project Explorer

Project Explorer merupakan salah satu jendela pada Visual Basic yang berisi semua file dalam aplikasi yang akan di buat, misalnya form, modul, class dan sebagainya.

#### 6. Jendela Propertis

Properties berisi semua informasi tentang sifat dari sebuah objek. Setiap objek dapat berbeda sifatnya sesuai dengan kebutuhan dari objek tersebut.

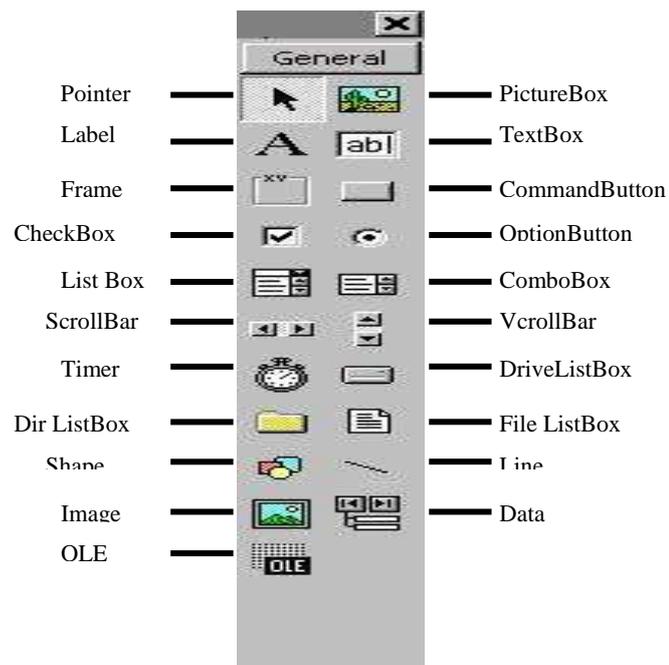
Properties tersebut antara lain, nama objek, warna, ukuran, posisi dan lain sebagainya.

## 7. Form Layout

Form Layout merupakan jendela yang menggambarkan posisi form dalam layar monitor.

## 8. Toolbox

Toolbox merupakan salah satu fasilitas pada Visual Basic yang berisi beberapa objek atau kontrol yang di butuhkan untuk membentuk suatu program aplikasi. Control merupakan suatu objek yang akan menjadi *interface* (penghubung) antara program aplikasi dan user-nya, dan kesemuanya harus di letakkan di dalam jendela form.



**Gambar II.17 Toolbox Control**

Sumber: Adi Kurniadi (2000: 12)

**Tabel II.6 Toolbox dan Fungsi**

Nama	Fungsi
Pointer	Ini bukan kontrol melainkan petunjuk kontrol yang terpasang pada form
Picture Box	Untuk menampilkan gambar statis maupun gambar aktif dari sumber di luar dirinya
Text Box	Untuk membuat area teks di mana teksnya bisa di ubah oleh pemakai
Frame	Untuk mengelompokkan beberapa kontrol. Kontrol ini harus di pasang terlebih dahulu sebelum kontrol yang di lingkupinya
Command Batton	Untuk membuat sebuah tombol pelaksanaan perintah
Check Box	Untuk membuat kotak cek yang mudah pemakaiannya. Bisa di gunakan untuk pemilihan dua keadaan (Benar atau salah, misalnya) dari banyak pilihan sekaligus
Option Button	Untuk pemilihan dua keadaan yang banyak pilihan. Namun hanya pilihan saja yang bisa di aktifkan pada satu saat.
Combo Box	Untuk menghasilkan kontrol yang merupakan kombinasi dari listbox dan textbox, daftar pilihan yang bisa digulung
Hscroll Bar	Untuk pengulungan dengan jangka lebar dengan indikasi posisi pemilihan dalam posisi horizontal
Vscroll Bar	Untuk penggulungan dengan jangka lebar dengan indikasi posisi pemilihan dalam posisi vertikal
Timer	Untuk menghitung waktu event dalam interval yang ditentukan
DriveList Box	Untuk menampilkan disk drive yang di miliki Computer
DirList Box	Untuk menampilkan direktori dan path
FileLise Box	Untuk menampilkan sebuah daftar file

Shape	Untuk memasang kontrol yang mampu menghasilkan saran agar pemakai bisa menggambarkan berbagai bentuk seperti oval, lingkaran dan lain-lain
Line	Untuk menggambarkan garis dengan berbagai piasinya
Image	Untuk menampilkan gambar bitmap, icon maupun metafile pada form. Kontrol Picture Box menyediakan lebih banyak fasilitas dibanding kontrol ini
Data	Menyediakan sarana akses data dalam suatu database
OLE	Untuk menghasilkan proses link dan embed objek antar aplikasi

**Sumber: Adi Kurniadi ( 2000 : 12)**

## II.8. Microsoft SQL Server 2005

### II.8.1 Mengenal Microsoft SQL Server 2005

Menurut Jogianto (1990;3) “Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.”

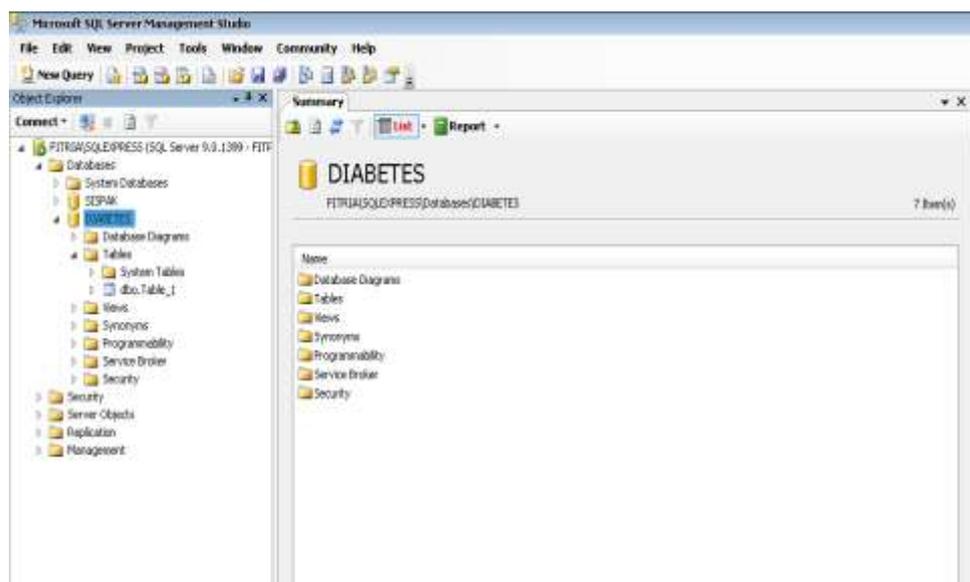
Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu di organisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*).

Dalam satu file terdapat record-record yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu record terdiri dari field yang saling berhubungan menunjukkan bahwa field tersebut dalam satu

pengertian yang lengkap dan direkam dalam satu record. Dari pengertian diatas dapat disimpulkan bahwa basis data mempunyai beberapa kriteria penting, yaitu :

1. Bersifat data oriented dan bukan program oriented.
2. Dapat digunakan oleh beberapa program aplikasi
3. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya.
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah
5. Dapat digunakan dengan cara-cara yang berbeda.

*SQL Server 2005* atau disebut juga dengan *SQL Server 9* merupakan pengembangan dari *SQL Server* versi 2000. *SQL Server 2005* merupakan aplikasi database produk *Microsoft* yang memiliki fitur-fitur baru untuk membuat suatu *platform* yang sempurna, khususnya dalam mengolah data yang berskala besar dan aplikasi *e-commerce*.



**Gambar II.18** Tampilan *SQL Server 2005*

## II.8.2. Tipe Data SQL Server 2005

Langkah pertama sebelum merancang sebuah tabel adalah menentukan tipe data pada setiap *field* yang akan digunakan. Tipe data digunakan untuk menjamin integritas data karena ada perubahan atau penambahan data yang harus disamakan dengan tipe data dari tabel.

**Tabel II.7 Tipe Data SQL Server 2005**

Kategori	Tipe Data	Jenis	Penjelasan
<i>Exact Numerics</i>	<i>Integer</i>	<i>BigInt</i>	Tipe data bilangan bulat 8 <i>byte</i>
		<i>Int</i>	Tipe data bilangan bulat 4 <i>byte</i>
		<i>SmallInt</i>	Tipe data bilangan bulat 2 <i>byte</i>
		<i>TinyInt</i>	Tipe data bilangan bulat 1 <i>byte</i>
	<i>Bit</i>	<i>Bit</i>	Tipe data bilangan bulat dengan nilai 0 / 1
	<i>Decimal dan Numeric</i>	<i>Decimal (n,p)</i>	Tipe data bilangan pecahan
		<i>Numeric (n,p)</i>	Tipe data <i>numeric</i> fungsinya sama dengan tipe data <i>decimal</i>
	<i>Monetary</i>	<i>Money</i>	Tipe data untuk nilai uang berukuran 8 <i>byte</i>
		<i>SmallMoney</i>	Tipe data untuk nilai uang
	<i>Floating point</i>	<i>Float</i>	Tipe data untuk bilangan <i>floating point</i> .
<i>Real</i>		Tipe data untuk bilangan yang tidak dapat dibulatkan	
<i>Date dan Time</i>	<i>Datetime</i>	Tipe data tanggal dan jam dengan akurasi 3,33 milidetik.	
	<i>Smalldatetime</i>	Tipe data tanggal dan jam dengan ketelitian sampai 1 menit.	
<i>Character String</i>	<i>Char (n)</i>	Tipe data dengan nilai <i>string</i> dengan panjang 1 s/d 8000 <i>byte</i> .	
	<i>Varchar (n)</i>	Tipe data dengan nilai <i>string</i> dengan batas maksimal 8000 <i>byte</i> .	
	<i>Text</i>	Tipe data semua jenis data berupa <i>text</i>	
<i>Unicode Character String</i>	<i>Nchar[(n)]</i>	Tipe data yang mempunyai ukuran panjang tetap sesuai yang ditentukan.	
	<i>Nvarchar[(n max)]</i>	Tipe data yang mempunyai nilai panjang tidak tetap	
	<i>Ntext</i>	Tipe data yang mempunyai	

			panjang tidak tetap.
<i>Binary String</i>	<i>Binary[(n)]</i>		Tipe data <i>binary</i> dengan panjang tetap
	<i>Varbinary[n max]</i>		Tipe data <i>binary</i> dengan panjang tidak tetap
	<i>Image</i>		Image merupakan tipe data <i>binary</i> dengan panjang tidak tetap
Tipe Data Lainnya		<i>Cursor</i>	Tipe data yang digunakan untuk variabel yang berisi referensi ke suatu kursor
		<i>Timestamp</i>	Tipe data yang berisi nilai urutan waktu
		<i>Uniqueidentifier</i>	Tipe data dengan bentuk bilangan <i>hexadecimal</i> dengan ukuran 16 <i>byte</i>
		<i>SQL Variant</i>	Tipe data yang berisi semua data yang diperbolehkan <i>SQL Server 2005</i> kecuali <i>text</i> , <i>ntext</i> , <i>timestamp</i> , dan <i>sql variant</i>
		<i>Table</i>	Tipe data yang dipakai untuk menyimpan hasil tabel
		<i>XML</i>	Tipe data dengan format dokumen berbasis teks khusus untuk menyimpan metode

Sumber: Yuswanto (2009 : 70)

## II.9. Penyakit Jantung

Jantung, yang dalam bahasa latin disebut *cor*, merupakan sebuah rongga organ berotot yang memompa darah melalui pembuluh darah karena kontraksi berirama yang berulang. Istilah *kardiak* berasal dari bahasa yunani, *cardia*, yang berarti yang berhubungan dengan jantung. Ia merupakan salah satu organ yang berperan dalam sistem peredaran darah. Masing-masing bagian jantung, kanan dan kiri, memiliki ruang sebelah atas (*atrium*) yang berfungsi untuk mengumpulkan darah dan ruang sebelah bawah (*ventrikel*) yang berfungsi untuk mengeluarkan darah. Agar darah hanya mengalir dalam satu arah, *ventrikel*

memiliki satu katup pada jalan masuk dan satu katup pada jalan keluar. Dalam pengertian lain, jantung adalah sebuah otot yang memompa darah keseluruh tubuh. Jantung sangat penting bagi manusia karena selalu diperlukan untuk memompa darah keseluruh tubuh sehingga tubuh mendapatkan oksigen dan sari makanan yang diperlukan untuk metabolisme tubuh.

Penyebab terjadinya serangan jantung ada dua yaitu:

#### 1. Faktor Yang Tidak Dapat Diubah

##### 1. Genetis

Riwayat penyakit jantung dalam keluarga dapat menjadi salah satu penyebab penyakit jantung.

##### 2. Usia dan Gender

Masa usia 45 tbagi laki-laki, merupakan sebuah masa yang rawan terhadap terjadinya serangan jantung.

#### 2. Faktor Yang Dapat Diubah

##### 1. Tekanan darah Tinggi (*Hipertensi*)

Tekanan darah tinggi adalah suatu peningkatan tekanan darah di dalam *arteri*.

##### 2. Stres

Tidak bisa dipungkiri, stress juga mempunyai dampak negatif bagi kehidupan manusia.

##### 3. Perokok berat

Tembakau dan asap tembakau mengandung dahan-bahan kimia yang menyebabkan kerusakan pada dinding-dinding pembuluh darah.

#### 4. Kolesterol Tinggi

Kolesterol atau kadar lemak dalam darah ditimbulkan oleh konsumsi makanan sehari-hari.

#### 5. Obesitas

Obesitas adalah kelebihan berat badan yang diakibatkan oleh penimbunan lemak tubuh yang berlebihan.

#### 6. Diabetes Mellitus

Diabetes mellitus atau penyakit gula darah alias kencing manis adalah suatu kondisi ketika kadar gula didalam darah melebihi batas normal karena tubuh tidak dapat melepaskan atau menggunakan insulin secara cukup.

#### 7. kebiasaan dan Berlebihan mengonsumsi alkohol

Alkohol adalah minuman yang mengandung *etanol*. *Etanol* adalah bahan psikoaktif dan mengonsumsinya dapat menyebabkan penurunan kesadaran.