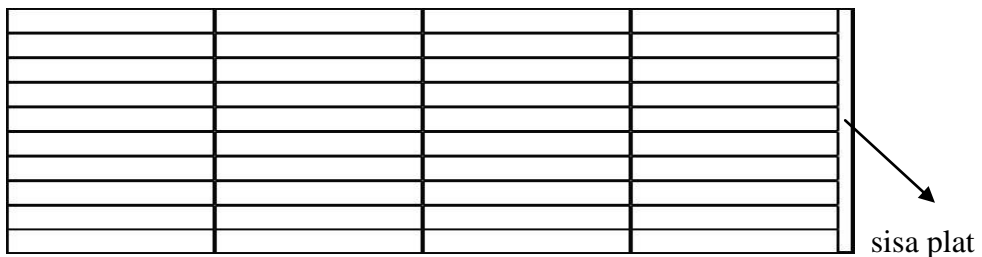


II. TINJAUAN PUSTAKA

2.1 Pemotongan

Masalah pemotongan dan pengepakan muncul dengan nama yang berbeda sesuai dengan literatur yang telah dipelajari. Meskipun beberapa makalah membahas masalah yang ada sebelum tahun 1961, Gilmore dan Gomory tersebut membuat suatu artikel yang membahas mengenai pemrograman dengan pendekatan satu dimensi mengenai masalah pemotongan yang pertama dan praktis.

Pada bagian ini, masalah pemotongan dan pengepakan serta berbagai masalah yang terkait diselidiki untuk mengeksplorasi formulasi dan teknik solusi yang dapat digunakan dalam masalah rencana pemotongan. Gambar 2 adalah rencana plat yang akan dipotong



Gambar 2. Rencana plat yang akan dipotong

2.2. Masalah Pemotongan dan Pengepakan Dua Dimensi

Mengingat deskripsi masalah, pemotongan plat bisa dianggap sebagai pemotongan dua dimensi atau berbagai masalah seperti berikut :

1. Dua dimensi dengan 2 tahap potongan pola
2. Semua item kecil harus ditugaskan ke pilihan penyimpanan
3. Benda besar adalah kelompok yang dibentuk oleh benda – benda kecil
4. Item kecil dari ukuran yang berbeda

Solusi untuk masalah pemotongan dan pengepakan dapat diklasifikasikan menjadi dua pendekatan yaitu, orientasi pola dan orientasi objek (*Dyckhoff 1990, dan Sweeney 1991*).

Seperti terlihat pada tabel sebagian besar metode solusi menggunakan pendekatan pola orientasi dalam memecahkan masalah. Pada pendekatan pola berorientasi berbeda dengan objek, pertama kali adalah membuat pola dan menempatkan objek besar dan item kecil dari beberapa pola – pola ini. Tabel 1 menunjukkan berbagai kategori solusi pendekatan dalam masalah pemotongan dan pengepakan.

Tabel 1. Pendekatan yang berbeda dalam menyelesaikan masalah pemotongan dan pengepakan (*Adaptasi dari Dyckhoff 1990 dan Sweeney 1991*).

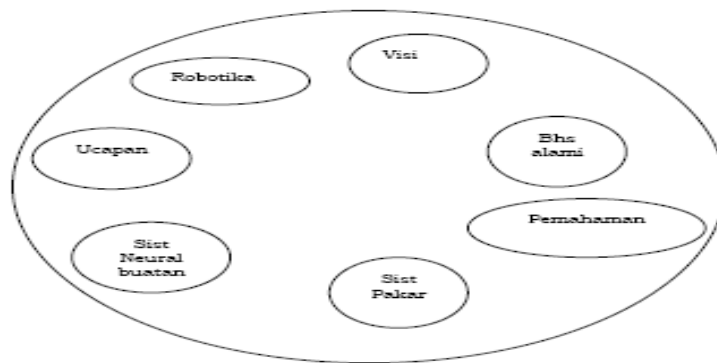
Pendekatan orientasi objek		Pendekatan orientasi pola		
Metode tepat	Perkiraan algoritma	Prosedur sequential heuristik	Pola Tunggal	Pola Persekutuan
Cabang dan terikat, program dinamis (Gloden, 1976)	Bin algoritma pengepakan (Coffman, 1984)	Heuristik (Haessler, 1971)	Bunttil algoritma (Hin xman, 1980; Dowsland, 1985; Ternoetal., 1987)	LP Berbasis dan heuristic umum (Hin xman, 1980; Stadler, 1988; Farley, 1988)

Pada setiap pola yang dihasilkan dalam masalah pemotongan dua dimensi jauh lebih sulit dibandingkan dengan masalah pemotongan satu dimensi (*Dyckhoff 1990*).

Pola yang dibuat menghasilkan sebuah proses adalah sub masalah dalam masalah penyimpanan. Masalah pemotongan plat tidak memiliki stok ukuran tetap, tujuannya adalah untuk menghasilkan pola berbeda yang akan mencakup semua item permintaan. Pada intinya masalah generasi pola atau pengelompokkan item kecil. Bahkan masalah pola generasi merupakan masalah pengelompokkan khusus sebagai kendala geometris.

2.3 Sistem Pakar

Sistem pakar (ES) adalah Sebuah program komputer yang dirancang untuk menyelesaikan masalah seperti layaknya seorang pakar (*human expert*). Sebuah sistem pakar memecahkan masalah domain dengan keahlian yang spesifik (misalnya lapangan) dan tidak dapat diterapkan untuk memecahkan masalah umum. Bagian ini memberikan tinjauan singkat sistem pakar tentang karakteristik, aplikasi dan komponen utama. Untuk menanggapi suatu masalah melibatkan beberapa ahli pada bidangnya (*Feigenbaum, 1979*).



Gambar. 3 Bagian Dari Intelegensi Buatan (*www.google.2011*)

Ada banyak manfaat yang dapat diperoleh dengan mengembangkan sistem pakar antara lain:

1. Masyarakat awam non pakar dapat memanfaatkan keahlian didalam bidang tertentu tanpa kehadiran langsung seorang pakar.
2. Meningkatkan produktivitas kerja yaitu bertambah efisiensi pekerjaan tertentu serta hasil solusi kerja.
3. Penghematan waktu dalam menyelesaikan masalah yang beragam.
4. Memberikan penyederhanaan solusi untuk kasus yang kompleks dan berulang-ulang.

5. Pengetahuan dari seorang pakar dapat didokumentasikan tanpa ada batas waktu.
6. Memungkinkan penggabungan berbagai bidang pengetahuan dari berbagai pakar untuk dikombinasikan.

Selain banyak manfaat yang diperoleh, ada juga kelemahan pengembangan sistem pakar, yaitu :

1. Daya kerja dan produktivitas manusia menjadi berkurang karena semuanya dilakukan secara otomatis oleh system.
2. Pengembangan perangkat lunak sistem pakar lebih sulit dibandingkan dengan perangkat lunak konvensional.

2.4 Aplikasi dan Karakteristik Sistem Pakar

Sistem pakar telah diterapkan dalam berbagai masalah yang menafsirkan, mengidentifikasi, memprediksi, mendiagnosis, merancang, perencanaan, pemantauan, pengujian, pelatihan dan pengendalian (*Badiru, 1992*).

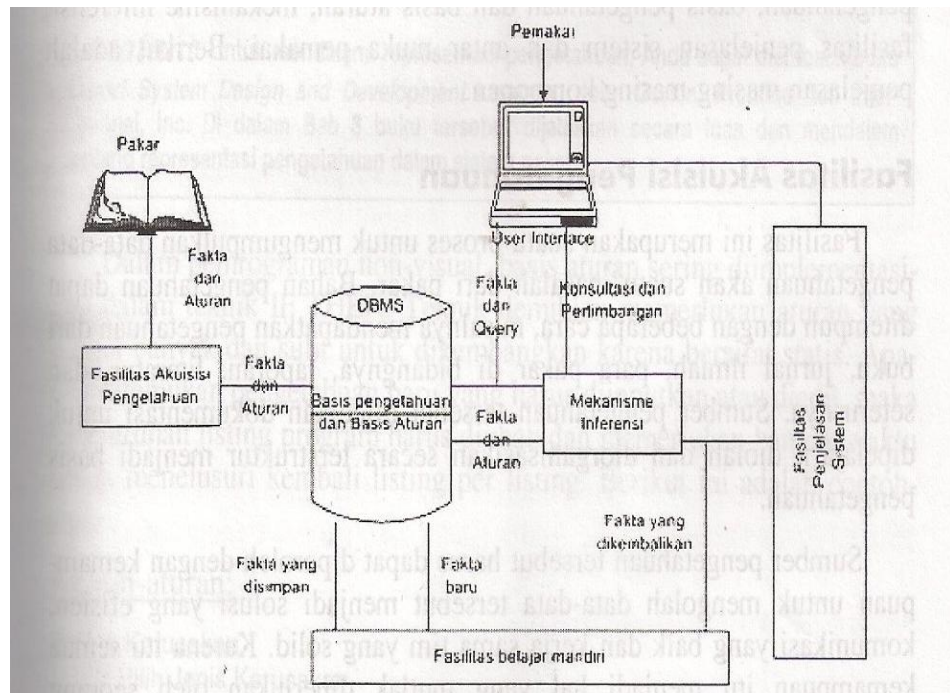
Aplikasi sistem pakar termasuk bisnis, pendidikan, manufaktur dan hukum. Penggunaan naik sistem pakar dari kebutuhan untuk mengkompensasi beberapa keterbatasan keahlian manusia. Keterbatasan ini termasuk kelangkaan keahlian manusia, fisik dan mental batas, tidak konsisten. Dari sisi manajemen terhadap sistem pakar menyediakan beberapa hal (*Badiru, 1992 ; Mc Cart, 1991*) :

1. Bantuan untuk ahli : sistem pakar dapat membantu para ahli membuat, memeriksa dokumen dan mengambil keputusan.

2. Ahli penggantian atau replikasi : sejak keahlian manusia terbatas, organisasi ingin mencari ahli pengetahuan dan menggunakan pengetahuan dalam tidak adanya ahli.
3. Analisis sensitivitas : sistem pakar dapat memfasilitasi pengambilan keputusan dengan menyediakan kemungkinan hasil dari program yang berbeda penyelesaian.
4. Pelatihan : sistem pakar dapat digunakan untuk personil yang kurang berpengalaman.
5. Peningkatan kinerja : sistem pakar dapat meningkatkan probabilitas, frekuensi dan konsistensi pembuatan keputusan yang baik. Mereka juga memberikan keputusan yang objektif tanpa reaksi dan emosional.
6. Kontrol biaya : setelah dikembangkan, sistem pakar menyediakan real time, biaya rendah dan ahli tingkat solusi oleh non ahli.

2.5 Komponen dasar sistem pakar

Suatu sistem pakar mempunyai ciri dan karakteristik tertentu. Hal ini juga harus didukung oleh komponen-komponen sistem pakar yang mampu menggambarkan tentang ciri dan karakteristik tersebut. Komponen penting dalam sistem pakar adalah akuisisi pengetahuan, basis pengetahuan dan basis aturan, mekanisme inferensi, fasilitas penjelasan program dan antar muka pengguna yang merupakan suatu kesatuan yang tidak dapat dipisahkan. Sedangkan fasilitas belajar mandiri merupakan komponen yang mendukung sistem pakar sebagai suatu keahlian tingkat lanjutan.



Gambar. 4 Struktur bagan sistem pakar (Andi,2009)

Fasilitas akuisisi pengetahuan merupakan suatu proses mengumpulkan data pengetahuan akan suatu masalah dari pakar. Setelah proses akuisisi dilakukan ,dapat dipresentasikan menjadi basis pengetahuan dan basis aturan yang selanjutnya dikumpulkan, dikodekan, diorganisasikan dan digambarkan dalam bentuk rancangan lain menjadi bentuk yang sistematis. Semua bentuk representasi data bertujuan untuk menyederhanakan data sehingga mudah dimengerti dan mengefektifkan proses pengembangan program. Mekanisme inferensi adalah bagian dari sistem pakar yang melakukan penalaran dengan menggunakan isi daftar aturan berdasarkan urutan dan pola tertentu. Selama proses konsultasi antar pakar dan pengguna, mekanisme inferensi menguji aturan satu demi satu sampai kondisi aturan itu benar. Secara umum ada dua teknik utama yang digunakan dalam mekanisme inferensi untuk pengujian

aturan, yaitu penalaran maju (*forward reasoning*) dan penalaran mundur (*reverse reasoning*).

Fasilitas penjelasan sistem merupakan bagian dari sistem pakar yang memberikan penjelasan tentang bagaimana program dijalankan, apa yang harus dijelaskan kepada pengguna tentang suatu masalah, memberikan rekomendasi kepada pemakai, mengakomodasi kesalahan pemakai dan menjelaskan bagaimana suatu masalah terjadi. Fasilitas penjelasan sistem harus mampu menjelaskan bagaimana harus memeriksa sekering yang putus atau bagaimana memeriksa aki motor, sehingga pengguna dapat mengerti dengan jelas apa yang dilakukannya. Dalam sistem pakar, fasilitas penjelasan sistem sebaiknya diintegrasikan kedalam tabel basis pengetahuan dan basis aturan karena hal ini lebih memudahkan perancangan sistem.

Antar muka pemakai memberikan fasilitas komunikasi antara pengguna dan sistem, memberikan berbagai fasilitas informasi dan berbagai keterangan yang bertujuan untuk membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi. Pada umumnya antarmuka pengguna juga berfungsi untuk memasukkan pengetahuan baru kedalam basis pengetahuan sistem pakar, menampilkan fasilitas penjelasan sistem dan memberikan tuntunan penggunaan sistem secara menyeluruh langkah demi langkah sehingga pengguna mengerti apa yang harus dilakukan terhadap sistem.

2.6 Tahapan pengembangan sistem pakar

Terdapat 6 tahap atau fase dalam pengetahuan sistem pakar seperti (Andi,2009) :

1. Identifikasi

Tahap ini merupakan tahap penentuan hal yang penting sebagai dasar dari permasalahan yang akan dianalisis. Tahap ini merupakan tahap untuk mengkaji dan membatasi masalah yang akan diimplementasikan kedalam sistem. Setiap masalah yang diidentifikasi harus mencari 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 sistem. Konseptualisasi juga menganalisis data-data penting yang harus didalami bersama dengan pakar dibidang permasalahan tersebut.

3. Formalisasi

Apabila tahap konseptualisasi telah selesai dilakukan, maka ditahap formalisasi konsep-konsep tersebut diimplementasikan secara formal, misalnya memberikan kategori sistem pakar yang akan dibangun, mempertimbangkan beberapa faktor pengambilan keputusan seperti

keahlian manusia, kesulitan dan tingkat kesulitan yang mungkin terjadi, dokumentasi kerja, dan sebagainya.

4. Implementasi

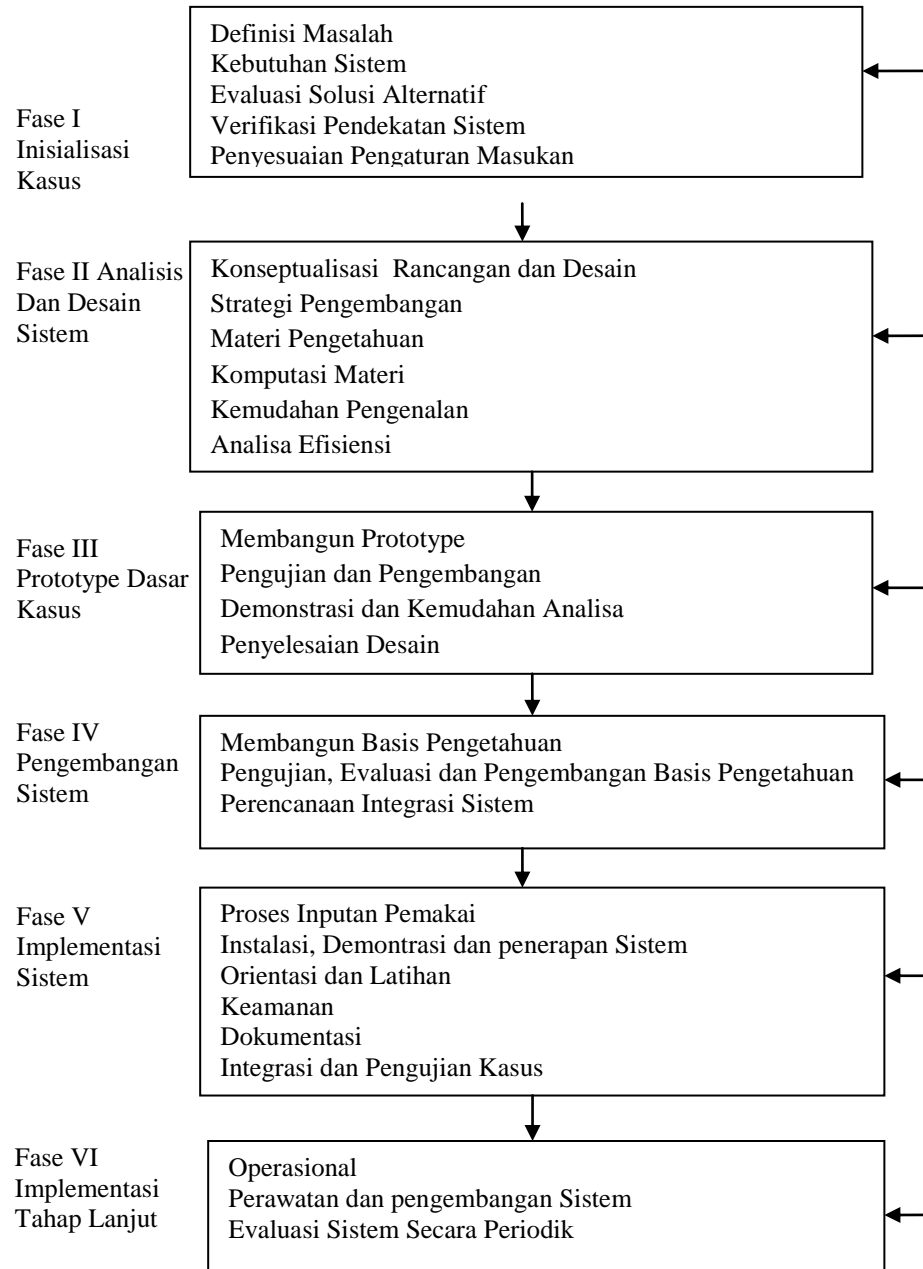
Apabila pengetahuan sudah diformalisasikan secara lengkap, maka tahap implementasi dapat dimulai dengan membuat garis besar masalah kemudian memecahkan masalah kedalam modul-modul.

5. Evaluasi

Sistem pakar yang selesai dibangun, perlu untuk dievaluasi untuk menguji dan menemukan kesalahannya. 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 pengembangan sistem yang berguna adalah proses dokumentasi sistem dimana didalamnya tersimpan semua hal penting yang dapat menjadi tolak ukur pengembangan sistem dimasa mendatang termasuk didalamnya adalah kamus pengetahuan masalah yang diselesaikan.



Gambar. 5 Fase pengembangan sistem pakar (Andi,2009)

Dari titik penggunaan, sistem pakar biasanya memberikan jawaban dalam domain yang spesifik (*field atau daerah*) dan memiliki karakteristik sebagai berikut (McCart 1991; Sumichrast 1995; Tzafestas et al 1993) :

1. Fitur yang paling membedakan dari sistem pakar adalah kemampuannya untuk melakukan seperti orang ahli dalam suatu domain yang spesifik.

Sebuah sistem pakar dapat bertindak seperti orang ahli dalam memecahkan masalah, memberikan nasihat atau informasi. Kinerja dari sistem pakar tergantung bagaimana domain pengetahuan tertentu ditangkap, diatur dan diakses dalam sistem pakar. Beberapa sistem pakar yang canggih memiliki kemampuan untuk belajar dan memperluas pengetahuan mereka sendiri sebagai dasar informasi baru ditambah.

2. Sistem pakar sering termasuk user *multi interface* meminta pertanyaan yang saling berhubungan, menerima input dan menampilkan hasil pengolahan. Banyak sistem pakar juga memiliki kemampuan untuk menjelaskan alasan yang diikuti dalam mencapai solusi. Kemampuan sistem pakar untuk menjelaskan alasan adalah salah satu keuntungan dari sistem pakar dibandingkan dengan pemodelan intelegen yang lain.
3. Banyak masalah ditangani oleh sistem pakar dapat dirumuskan sebagai ruang pemecahan sebuah masalah, sistem ini melakukan pencarian dengan pengolahan sumber daya yang terbatas dan mendapatkan solusi yang terbaik dalam setiap permasalahan. Metode pencarian exhaustive tidak digunakan karena membutuhkan memori dan waktu proses yang lebih. Sistem pakar menggunakan informasi khusus dari domain untuk memandu pencarian dan mengurangi pencarian waktu. Informasi seperti ini disebut informasi heuristik dan metode pencarian yang disebut heuristik. Heuristik tidak selalu mencakup setiap situasi yang sedemikian mungkin, sistem pakar terkadang gagal untuk mendapatkan solusi yang baik.

4. Pengetahuan yang memecahkan masalah secara kompleks, masalah dunia nyata biasanya pasti tidak tepat dan tidak lengkap. Sistem pakar seperti manusia sering memiliki kemampuan untuk menghadapi ketidakpastian.

a. Struktur Sistem Pakar

Seperti yang telah didefinisikan diatas, sebuah sistem pakar mensimulasikan pengambilan keputusan dan pemecahan masalah dalam sebuah domain tertentu dan diharapkan untuk tampil sebanding dengan ahli tingkat. Dalam pengambilan fakta dan menerapkan heuristik efisien, pengetahuan dalam sistem pakar harus diorganisir dalam format yang mudah diakses. Selanjutnya, para peneliti telah menemukan bahwa banyak masalah membutuhkan penalaran yang identik, pengolahan dan berbeda hanya didalam domain spesifik yang diperlukan pengetahuan untuk pemecahan masalah. Untuk alasan ini, sistem pakar biasanya disusun dalam tiga komponen (*Badiru 1992*) :

1. Pengetahuan dasar : variabel ini berisi uraian deklaratif ahli informasi, pemecahan masalah peraturan, prosedur atau data intrinsik yang diperlukan untuk pemecahan masalah domain.
2. Interferensi mesin : mekanisme kontrol genetik yang memecahkan masalah dengan menginterpretasikan pengetahuan aksiomatik dalam basis pengetahuan.
3. Memori kerja : interferensi catatan masalah saat ini dalam ruang kerja yang disebut memori kerja. Memori kerja mungkin juga sebagai tujuan

perolehan pengetahuan (akuisisi pengetahuan modul) dan interaksi umum dengan pengguna (*user interface*) (Nikolopoulos, 1992).

Pembagian basis pengetahuan dan inferensi engine memiliki keuntungan dari fleksibilitas dasar pengetahuan biasa diganti dengan yang baru. Namun inferensi engine dan basis pengetahuan yang tidak sepenuhnya independent. Basis ini merupakan pengetahuan yang harus memperhitungkan built in control mekanisme inferensi engine. Kinerja sistem pakar sangat dipengaruhi oleh struktur dari basis pengetahuan atau secara khusus oleh skema representasi pengetahuan. Bagian berikut membahas akuisisi, representasi pengetahuan dan mekanisme penalaran diadaptasi dalam sebuah inferensi engine serta keprihatinan khusus untuk masalah melibatkan beberapa ahli.

b. Pengetahuan elisitasi dan representasi

Dasar pengetahuan mempertahankan pengetahuan pakar yang relevan untuk memecahkan masalah domain. Pengetahuan itu ditangkap melalui pengetahuan proses akuisisi. Pengetahuan yang ditangkap harus dikonversi atau dikodekan untuk representasi pengetahuan formalisme yang dapat disimpan dan dimanipulasi oleh komputer. Beberapa penelitian menggunakan “ akuisisi pengetahuan “ istilah untuk menyertakan elisitasi pengetahuan dan representasi sebagai pengetahuan elisitasi saja, selain itu sebagai representasi pengetahuan (Nikolopoulos, 1997).

Bagian kreatif menekankan perolehan pengetahuan yang “memberi kesempatan untuk memperoleh pengetahuan mandiri yang menghasilkan pendekatan baru dalam memecahkan masalah domain yang dipertimbangkan“ (*Badiru,1992*).

Untuk menghindari kebingungan, semakin sempit dan konsisten didefinisikan dalam jangka elisitasi pengetahuan yang digunakan dalam bagian ini. Penelitian masalah dan teknik untuk memunculkan sebuah pengetahuan secara terus – menerus.

Salah satu definisi pengetahuan adalah “suatu kesadaran, pemahaman, hubungan fakta, representasi dan beberapa proses dari subyek”.

Berdasarkan definisi ini, tidak ada yang jelas representasi sistem pakar untuk memiliki kesabaran tersebut. Untuk sistem komputer pengetahuan mungkin berarti “menyimpan, mengambil, menciptakan fakta, hubungan, representasi dan proses yang berkaitan dengan subjek tertentu” (*Sumichrast,1995*)

Ada banyak skema representasi dalam sistem pakar yaitu (*Badiru 1992; Sumichrast 1995*) :

1. Jaringan semantik : jaringan semantik terdiri dari kumpulan node dan busur. Node dihubungkan dengan busur untuk membentuk hubungan objek. Membawa busur notasi yang menunjukkan jenis hubungan antara node. Jaringan semantik tidak memiliki struktur definitif formal yang diperlukan untuk implementasi dalam pengaturan operasional.

2. Aturan : aturan yang paling umum dan serbaguna representasi semua skema. Aturan menggunakan template " jika X maka Y " untuk mewakili pengetahuan. X merupakan permis atau masukan, sedangkan Y adalah kesimpulan, output atau konsekuen. Misalnya " JIKA tidak aktif dan B tidak tersedia BERAKHIR menggunakan metode pemotongan C ". Pendahuluan ini biasanya berisi beberapa klausal terkait penghubung logika. Keuntungan menggunakan aturan (a) fleksibilitas aturan individu dapat dengan mudah ditambahkan, dihapus atau diubah (b) kemudian interpretasi, (c) kemampuan untuk mewakili interaksi antara deklaratif dan pengetahuan prosedural. Namun aturan tidak selalu memberikan efisien yang berarti mewakili pengetahuan. Sebagai contoh pengetahuan prosedural tidak efisien direpresentasikan sebagai aturan. Bahkan ketika aturan efisien merupakan pengetahuan dan jumlah aturan yang diperlukan untuk beberapa masalah tantangan dalam merancang strategi pencarian yang efisien.

3. Proposional logika atau logika predikat : logika proposional adalah dasar logika yang digunakan untuk menentukan apakah suatu proposisi yang diberikan benar atau salah.

Predikat logika menambahkan kemampuan untuk menentukan hubungan tertentu dan membuat generalisasi tentang proposisi. Logika proposional memiliki kelebihan kesederhanaan, keringkasan dan modularitas. Namun kesulitan yang mewakili prosedural dan heuristik pengetahuan dalam mengelola pengetahuan basis besar karena struktur organisasi terbatas.

4. Lain – lain : skema representasi lainnya seperti kasus, script, bingkai dan obyek atribu nilai. Pengetahuan representasi sangat penting untuk kinerja sistem ahli. Metode representasi pengetahuan yang berbeda memiliki kekuatan yang berbeda dan keterbatasan. Dalam prakteknya, sistem berbasis aturan yang paling mudah untuk diterapkan. Rulebased sistem telah diterima performanya secara keseluruhan tetapi efisiensi menurun dengan meningkatnya volume pengetahuan. Pemilihan representasi skema harus dipertimbangkan dengan aplikasi yang spesifik.

c. Penalaran dalam sistem pakar

Pengetahuan disimpan dalam basis pengetahuan dengan penalaran secara logis untuk menghasilkan sebuah solusi. Sebuah sistem pakar dan mekanisme kontrol biasanya tertanam dalam inferensinya. Disajikan beberapa konsep penting dan model inferensinya :

1. *Deduktif dan penalaran induktif* : induktif dan deduksi berguna untuk konsep sistem pakar. Penggunaan yang menarik merupakan kesimpulan tentang anggota kelas tertentu dan informasi umum tentang kelas tersebut, sebaliknya harus menarik kesimpulan umum berdasarkan data yang spesifik.
2. *Modus ponens* : alasan logis yang memungkinkan kesimpulan dapat ditarik berdasarkan aturan dan fakta, ini adalah nama resmi untuk memungkinkan ketika aturan premis aturan benar yang berlaku untuk sebuah kesimpulan. Sebagai contoh mengingat aturan “jika A benar

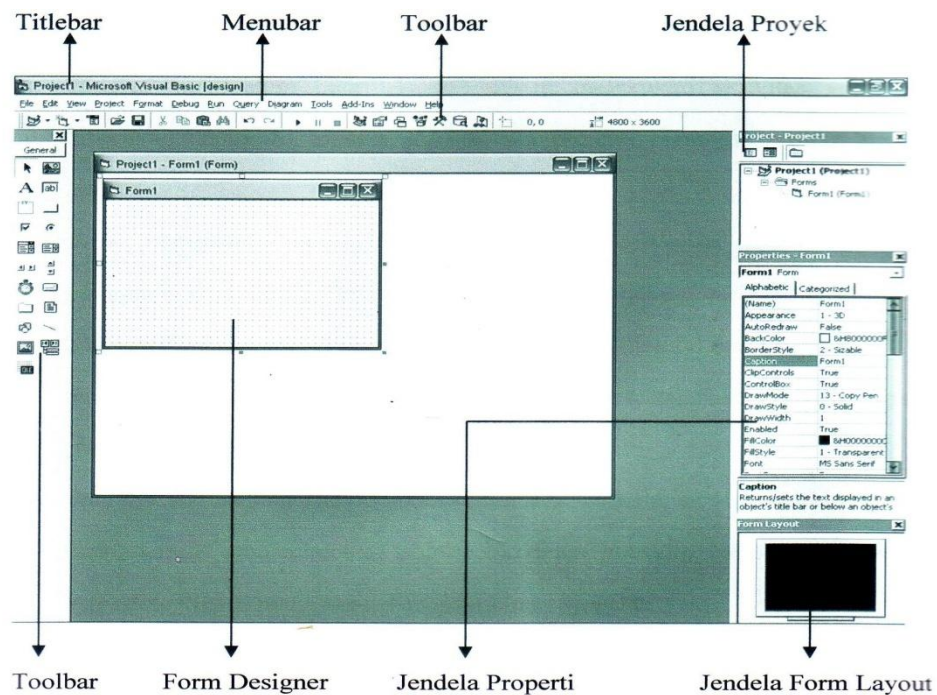
maka B benar” dan fakta “A benar”, ponens modus memungkinkan kesimpulan dari “B adalah benar” bisa ditarik.

3. *Modulus tollens* : kebalikan dari dari modus ponens, bahwa jika premis dari aturan harus benar untuk aturannya. Jika kesimpulannya adalah palsu premis tersebut adalah salah, misalnya “jika A benar maka B benar” dan fakta “B adalah palsu, modus tollens “adalah palsu”.
4. *Forward chaining* : suatu mekanisme kontrol untuk menyimpulkan fakta baru dari fakta yang diberikan. Sebuah rantai maju dimulai dengan fakta – fakta yang diketahui dan menyimpulkan fakta baru sampai dengan tujuan tercapai atau tidak ada fakta – fakta baru yang dapat disimpulkan.
5. *Backward chaining* : mekanisme kontrol yang dimulai dari tujuannya. Sebuah proses mundur karena diawali dengan solusi potensial dan pencarian fakta – fakta yang mendukung untuk sebuah solusi. Banyak sistem pakar menggunakan lebih dari satu penalaran dan metode kontrol. Misalnya, aturan dasar sistem dapat menggunakan forward chaining dan backward chaining.

2.7. Microsoft Visual Basic

Microsoft Visual Basic merupakan salah satu aplikasi pemrograman visual yang memiliki bahasa pemrograman yang cukup populer dan mudah untuk dipelajari. Basis bahasa pemrograman yang digunakan dalam Visual Basic adalah BASIC (*Beginners All-purpose Symbolic Instruction Code*) yang merupakan salah satu bahasa pemrograman

tingkat tinggi yang sederhana dan mudah dipelajari. Dengan Visual Basic kita dapat membuat program aplikasi GUI (*Graphical User Interface*) atau program yang memungkinkan pengguna komputer berkomunikasi menggunakan grafik atau gambar 8.



Gambar.6 Tampilan awal menu *Visual Basic*

Visual Basic merupakan bahasa pemrograman yang mudah digunakan sehingga banyak yang menggunakan. Dibandingkan dengan bahasa pemrograman lain semisal pascal dengan menggunakan Visual Basic dalam penulisan kode program mampu menambahkan sendiri.

Kelebihan dari *Visual Basic* (VB) adalah kemampuannya untuk mengkompilasi program dalam bentuk *Native code* yaitu optimasi pada saat prosesor mengkompilasi dan menjalankan program tersebut.

Keuntungan yang didapat dari *native code* adalah kecepatannya dalam

mengakses program , hal ini hanya dapat ditemui pada aplikasi-aplikasi yang dikompilasikan dengan bahasa pemrograman C++. Selain itu VB juga menyediakan fasilitas antar muka penulis kode program yang lebih mudah dimengerti dan digunakan sehingga berbagai tipe program dapat dikembangkan didalamnya, misalnya EXE, DLL dan OCX, bahkan program-program berbasis internet.