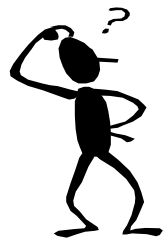


PENGENALAN SISTEM PAKAR

Pengertian A.I

- ✓ Merup. Sub-bid pengetahuan komp. yang ditujukan u/ membuat software (S/W) dan hardware (H/W) yang sepenuhnya bisa menirukan beberapa fungsi otak manusia.
- ✓ Sbg. Cabang sains komp. yang mempelajari otomatisasi tingkah laku cerdas (intelligent)
- *Intelligence/Intelegensia* : seseorang yang pandai melaksanakan pengetahuan yang dimilikinya.

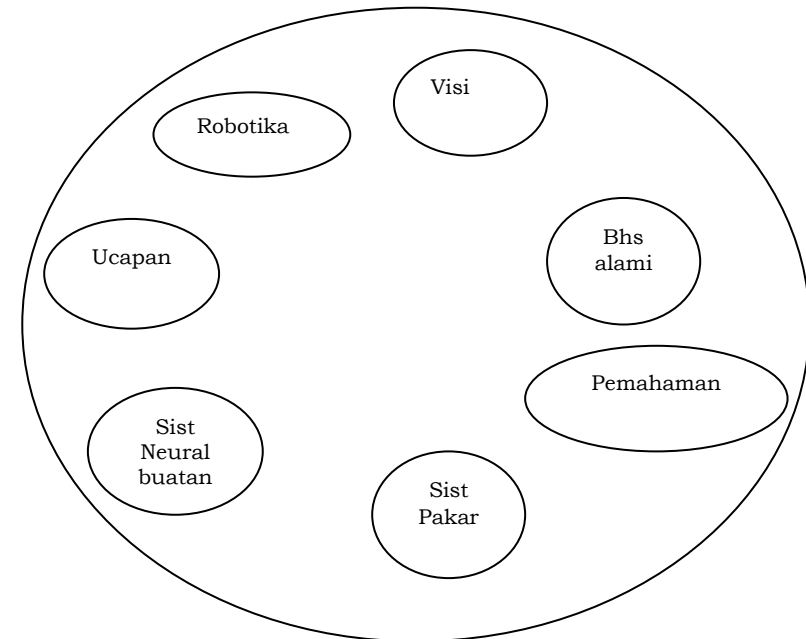


- Mahasiswa/Pelajar ?
- Tukang becak ?
- Bayi ?
- Penjahat ?



dkl : kemampuan manusia u/ memperoleh pengetahuan dan pandai melaksanakannya dalam praktek

- ✓ *Artificial Intelligence* adalah bidang ilmu komputasi yang memungkinkannya u/ memahami, bernalar & bertindak
- ✓ Bagian dari Intelegensi Buatan



Pengertian Sistem Pakar (Expert Sistem)

- Membuat S/W Expert Systems → prog. Sebagai penasehat/konsultan pakar
- Dapat mengumpulkan dan menyimpan pengetahuan seorang/beberapa orang pakar ke dalam komp. → u/ semua orang yang memerlukan
- Tidak u/ menggantikan kedudukan seorang pakar ttp u/ memasyarakatkan pengetahuan & pengalaman pakar tsb.
- Memungkinkan orang lain meningkatkan produktivitas, memperbaiki kualitas keputusan dll.

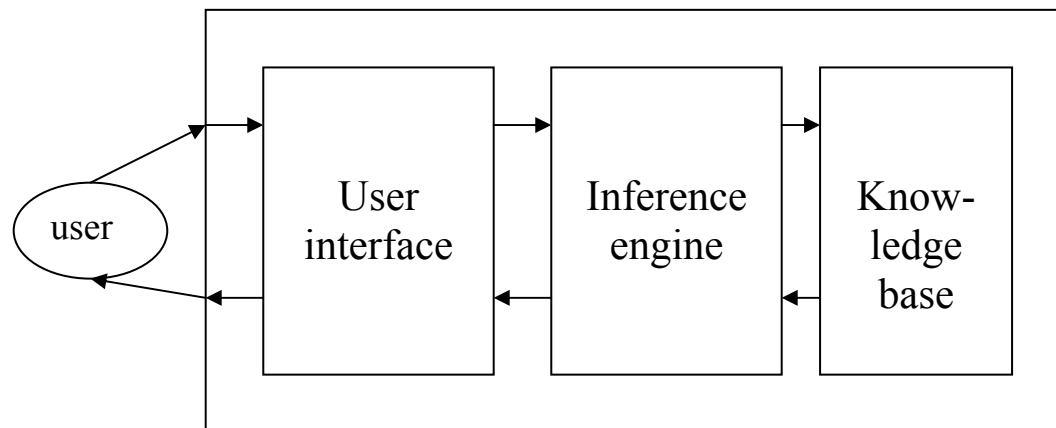
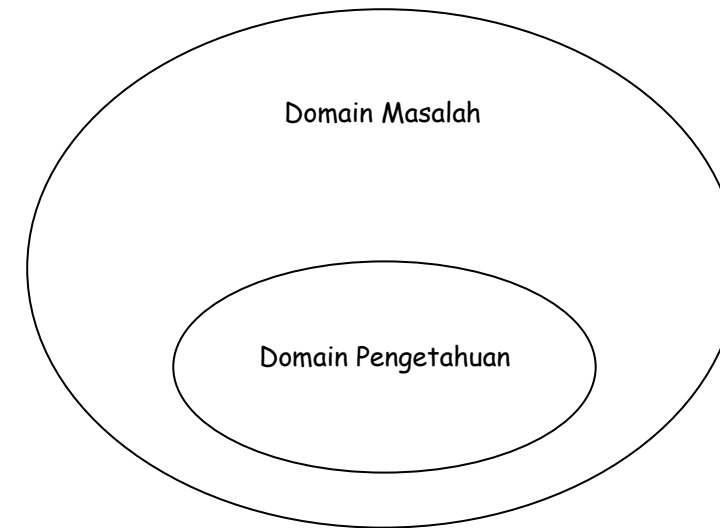


Diagram blok Expert Systems (umum)

- ✓ Knowledge base berisi semua fakta, ide, hubungan
- ✓ Motor inferensi bertugas u/ menganalisis pengetahuan dan menarik kesimpulan berdasarkan knowledge base.
- ✓ S/W user interface berfungsi sbg media pemasukan pengetahuan ke dalam (KB)

Domain Pengetahuan Expert



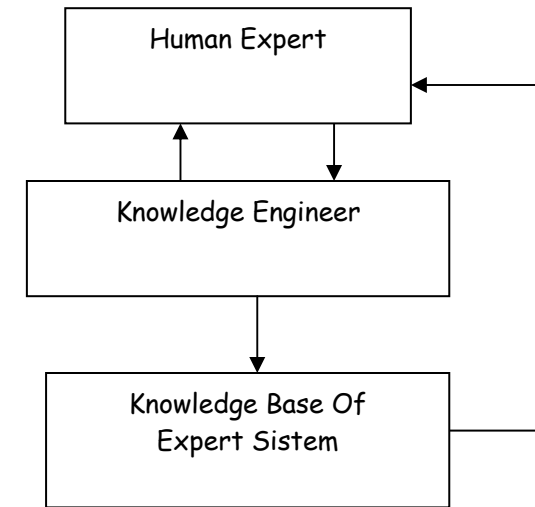
Keuntungan / Kelebihan Sistem Pakar

- Availability-bertambah * Intelligent tutor
- Cost-rendah * Intelligent dB
- Danger-reduced
- Performance
- Multiple expertise
- Reability-bertambah
- Explanation
- Response-cepat
- Steady, unemotional and complete response

KONSEP UMUM SISTEM PAKAR (SP)

*Salah satu metode representasi pengetahuan:
IF..... THEN

*Proses pembuatan SP → *knowledge engineering* yg dilakukan oleh *knowledge engineer*. Selain itu *domain expert* dan *end user*.



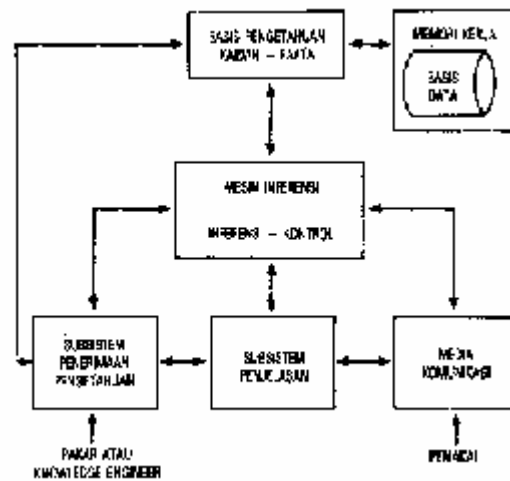
Perkembangan Sistem Pakar

* Tugas *knowledge engineer* adalah memilih S/W & H/W u/ pembuatan SP, membantu mengambil pengetahuan yg dibutuhkan dari pakar domain, serta implementasi pengetahuan pada basis pengetahuan yg benar & efisien

*Tugas pakar domain : meyediakan pengetahuan ttg bid problem yg dihadapi, memahami teknik-teknik pemecahan problema yg dipakai.

*Batasan praktis dari beberapa SP → *casual knowledge*

*SP lebih mudah untuk memprogram dg *Shallow knowledge* yaitu berdasarkan pada pengalaman dan pengetahuan heuristik.



Keterangan

- Basis Pengetahuan
 - inti prog SP
 - representasi pengetahuan dari seorang pakar.
 - Macam-macam

- Mesin Inferensi
 - mekanisme fungsi berfikir dan pola-pola penalaran sistem yg digunakan pakar
 - menganalisa suatu masalah tertentu
 - mencari jawaban atau solusi yg terbaik.
 - Ada 2 pelacakan backward & forward chaining

KARAKTERISTIK SP

- High Performance
- Adequate response time
- Good reliability
- Understandable
- Flexibility

PENGEMBANGAN TEKNOLOGI SP

- Akar SP pada banyak disiplin ilmu "*cognitive science*" yaitu study bagaimana orang memikirkan dlm pemecahan masalah. "*cognitive processor*" yaitu menemukan aturan yg akan diaktifkan.

SP YANG TERKENAL

1. MYCIN

- Dirancang oleh Edward Feigenbaum (Universitas Stanford) th '70 an
- SP medical yg dpt mendiagnosa infeksi bakteri & rekomendasi pengobatan antibiotik

2. DENDRAL

- SP struktur molekular & kimia

3. PROSPECTOR

- Membantu ahli geologi yg mencari & menemukan biji deposit (mineral & batuan)
- Didesign oleh Sheffield Research Institute, akhir '70an

4. XCON (R1)

- SP konfigurasi sistem komputer dasar
- Dikembangkan oleh Digital Equipment Corporation (DEC) dan Carnegie Mellon Universitas (CMU), akhir '70 an
- Untuk sistem komputer DEC VAC 11 1780

5. DELTA

- Didesign & dikembangkan oleh General Electric Company
- SP personal maintenance dg mesin lokomotif listrik diesel.

6. YESMVS

- Didesign oleh IBM awal th '80an
- Membantu operator komputer & mengontrol sistem operasi MVS (multiple virtual storage)

7. ACE

- Didesign & dikembangkan oleh AT&T Bell Lab awal th '80an
- SP troubleshooting pd sistem kabel telpon

KLASIFIKASI APLIKASI SP

1. CONTROL

- Aplikasi komputer yg sangat umum
- Ada 2 jenis kontrol : loop terbuka & tertutup

2. DEGUGGING

- Proses mencari kesalahan & memperbaiki solusi.

3. DESIGN

- Pengumpulan informasi mengenai spesifikasi sistem & produk tertentu
- Untuk merancang sirkuit elektronik, bangunan, dan rumah.

4. DIAGNOSIS

- Untuk mendiagnosa produk atau sistem yg sudah tdk berfungsi.

5. INSTRUKSIONAL

- Untuk membantu dalam proses belajar mengajar

6. INTERPRETASI

- Membantu seorang dlm menafsir & memahami situasi/perspektif suatu peristiwa.

- Contoh : analisa intelegensia, daya tahan, citra dan sinyal

7. PLANNING

- Merumuskan metode, penataan yg dapat mendekatkan pd tujuan.
- Contoh : proyek manajemen, taktik & strategi militer, pemrograman robot

8. PREDIKSI

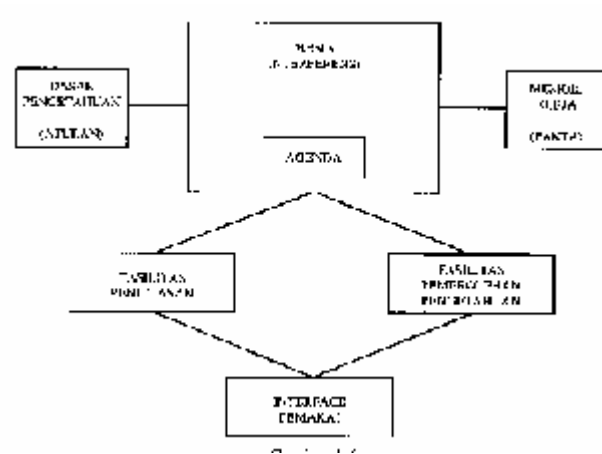
- Meramalkan apa yg terjadi di masa yg akan datang.

9. REPARASI

- Memperbaiki barang yg rusak ke keadaan semula

10. KONFIGURASI

ELEMEN SP



1. User Interface --- kom antara user & SP
2. Explanation Facility --- pemberian alasan pd user
3. Working Memori
4. Inference Engine --- penentuan aturan yg hrs dipenuhi, prioritas aturan yg tercukupi, & prioritas yg tertinggi
5. Agenda --- daftar yg diprioritaskan dari aturan (4)

6. Fasilitas Pemrolehan Pengetahuan --- cara otomatis bagi pemakai untuk memasukkan pengetahuan dlm sistem.

Rangkaian Forward (Forward chaining)

→ merupakan pemberi alasan dari fakta untuk kesimpulan hasil dari fakta

Contoh :

Jika kita melihat bahwa hari ini akan turun hujan sebelum pergi (nyata)

Maka kita harus membawa payung (kesimpulan)

Mis : Programan OPS5, CLIPS

Rangkaian Backward (Backward chaining)

→ Pemberian alasan sebaliknya dari hipotesa, kesimpulan potensial dibuktikan, pada fakta yg mendukung hipotesa

Contoh:

Jika kita tidak melihat keluar dan seseorang masuk dg sepatu basah dan payung.

Hipotesa kita adalah bahwa hari hujan

Mis : EMYCIN

SISTEM PRODUKSI

➤ Salah satu type SP yg paling terkenal adalah system yg berdasarkan pd aturan.

➤ Alasannya :

1. *Modular nature*
2. *Explanation facility*
3. *Similarity to the human cognitive process*

➤ POST

Idenya :

- System matematika & logika merupakan set aturan sederhana untuk menentukan bagaimana mengubah 1 string simbol ke dlm simbol lainnya.
- Yaitu dg input string, kejadian sebelumnya,

➤ ALGORITMA MARKOV

- Merupakan kelompok produksi yg terorder yg diterapkan untuk prioritas ke input string.
- Algoritma akan berakhir dg baik jika:
 - (1). Produksi terakhir tidak dapat diterapkan pada string
 - (2). Suatu produksi yg berakhir dg periode diterapkan.
- Jika input string GABKAB
System produksi $AB \rightarrow HIJ$
Maka hasil akhir GHIJKHIJ
- Karakter $\hat{\ } \rightarrow$ string nol
- Mis $A \rightarrow \hat{\ }$ artinya menghilangkan seluruh kejadian karakter A dlm suatu string
- Karakter tunggal a, b, c, \dots
- Mis $AxB \rightarrow BxA$ artinya mengubah karakter A dan B
- Huruf Yunani α, β

Contoh : Memindahkan huruf pertama string input ke akhir

Aturan 1. $\alpha xy \rightarrow y\alpha x$

2. $\alpha \rightarrow \hat{\alpha}$

3. $\hat{\alpha} \rightarrow \alpha$

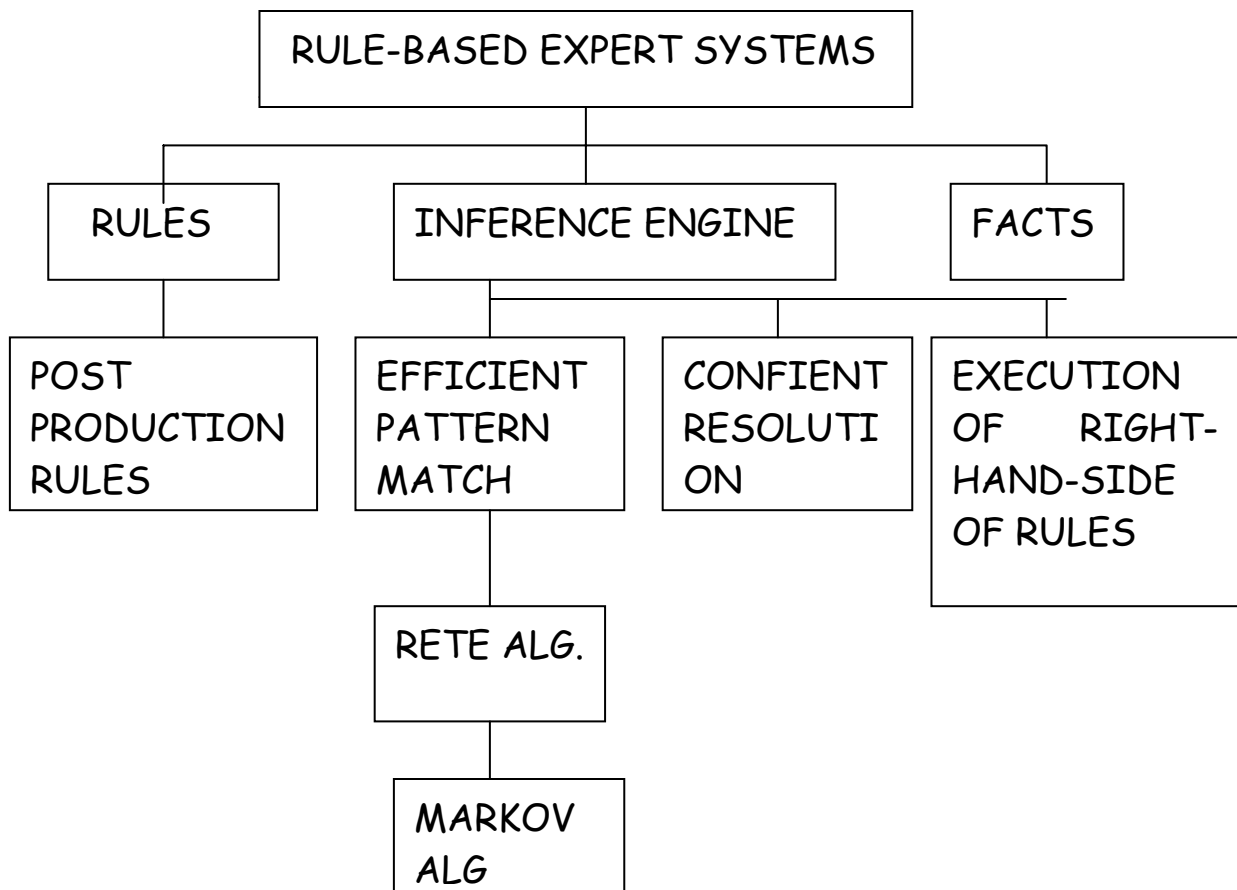
Input ABC

Aturan	Sukses atau Gagal	String
1	G	ABC
2	G	ABC
3	S	αABC
1	S	$B\alpha AC$
1	S	$BC\alpha A$
1	G	$BC\alpha A$
2	S	BCA

➤ ALGORITMA RETE

- Pada algoritma Markov diterapkan aturan/baris prioritas lebih tinggi.

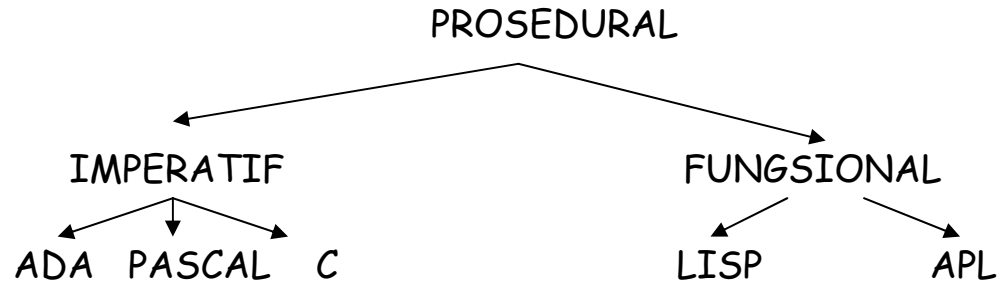
- Masalah timbul jika system mempunyai aturan/baris yang banyak, maka tidak akan efisien.
- Solusinya adalah *algoritma Rete* yang dikembangkan oleh **Charles L.F** di **Carnegie-Mellon University (1979)**
- Yaitu algoritma yg mengetahui tentang seluruh aturan/baris seluruh sistem dan dapat menerapkan suatu baris tanpa harus mencoba setiap baris tanpa berangkai (mencari perubahan dalam gabungan setiap cycle)
- Merupakan gabungan pola yang sangat cepat, yang mendapatkan kecepataannya dengan menyimpan informasi tentang baris dalam jaringan.



SP YANG BERSADARKAN ATURAN MODEREN

KLASIFIKASI PARADIGMA PEMROGRAMAN

1. PARADIGMA PROSEDURAL

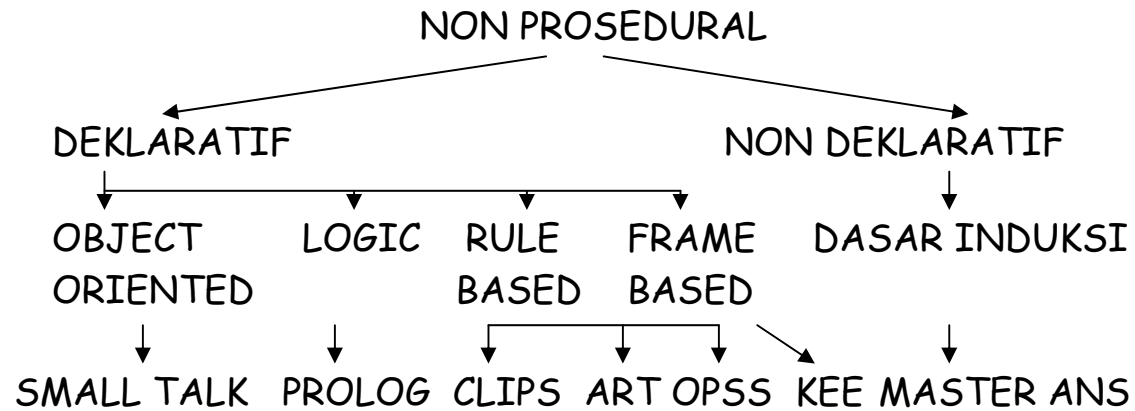


ALGORITMA adalah metode untuk pemecahan masalah dalam sejumlah tahap/langkah tertentu.

- ✓ Implementasi algoritma dalam suatu program disebut *program prosedural*.
- ✓ Pemrograman algoritma (prosedural) dan konvensional untuk program type non-AI.
- ✓ Sinonim untuk pemrograman prosedural adalah prog. Sequential.
- ✓ Pada pemrograman prosedural programmer harus menentukan sesungguhnya bagaimana pemecahan masalah harus di-code-kan.

✓ Pembuat code adalah pemrograman non prosedural.

2. PARADIGMA NON PROSEDURAL



✓ Penekanan pemrograman Non prosedural adalah penentuan apa yg akan diselesaikan dan membiarkan system menentukan bagaimana menyusunnya.

➤ PEMROGRAMAN DEKLARATIF

Memisahkan tujuan dari metode yg digunakan untuk mencapai tujuan.

➤ PEMROGRAMAN OBJECT ORIENTED

Ide : membuat dsign program dg mempertimbangkan data yg digunakan dalam program sebagai objek dan mengimplemnetasikan operasi pada objek tersebut.

➤ PEMROGRAMAN LOGIKA

Pembuktian teori logika dg Logic Theorirt Program (Newell & Simon) pada Darmouth Conference A.I (1956)

Rangkaian backward dapat digunakan untuk mengekspresikan pengetahuan dalam representasi deklaratif maupun kontrol proses pemberian alasan.

Keuntungannya : pembuatannya dapat diproses secara paralel yaitu jika ada beberapa processor dapat bekerja secara simultan.

EXPERT SYSTEM

- ✓ Disebut *pemrograman deklaratif* krn programmer tdk menentukan bagaimana prog. hrs mendapatkan tujuannya pada level algoritma

ANS (ARTIFICIAL NEURAL SYSTEMS)