

PERTEMUAN 13

EXPERT SYSTEM (ES)

ES diturunkan dari istilah *Knowledge-Based Expert System*, *Expert System* adalah sistem yang mendapatkan dan menyimpan *Knowledge* manusia ke dalam komputer untuk menyelesaikan permasalahan yang biasanya membutuhkan kepakaran seorang ahli.

Area/bidang kepakaran ini disebut dengan domain, pengembangan *Expert System* terindikasi pada hal-hal di bawah ini:

1. Ketersediaan berbagai tool yang didesain untuk memudahkan pembangunan *Expert System* dan mengurangi biayanya.
2. Penyebarluasan *Expert System* pada ribuan organisasi, beberapa diantaranya menggunakan ratusan atau malah ribuan sistem yang spesifik.
3. Integrasi *Expert System* dengan berbagai CBIS yang lain berkembang makin pesat, khususnya integrasi dengan database dan DSS.
4. Penggunaan *Expert System* semakin meningkat pada berbagai hal, mulai dari sistem help sampai ke aplikasi bidang militer dan ruang angkasa.
5. Penggunaan teknologi *Expert System* sebagai metodologi yang mempermudah pembangunan sistem informasi reguler.
6. Peningkatan penggunaan *object-oriented programming* (OOP) dalam representasi *Knowledge*.
7. Pengembangan sistem utuh memiliki berbagai sumber *Knowledge*, *reasoning*, dan informasi *fuzzy*.
8. Penggunaan *multiple Knowledge base*.

6.1 Konsep Dasar ES.

Kepakaran adalah *Knowledge* yang ekstensif, spesifik yang didapatkan dari training, membaca, dan pengalaman. Berbagai jenis *Knowledge* di bawah ini adalah contoh dari kepakaran:

1. Fakta mengenai area/daerah masalah.
2. Teori mengenai area masalah.
3. Aturan dan prosedur berkaitan dengan area masalah secara umum.
4. *Rules (heuristic)* dari apa yang harus dikerjakan pada situasi masalah tertentu (contoh aturan yang berkaitan dengan penyelesaian masalah).
5. Strategi global untuk menyelesaikan masalah tertentu.
6. *Meta-Knowledge* (*Knowledge* mengenai *Knowledge* itu sendiri).

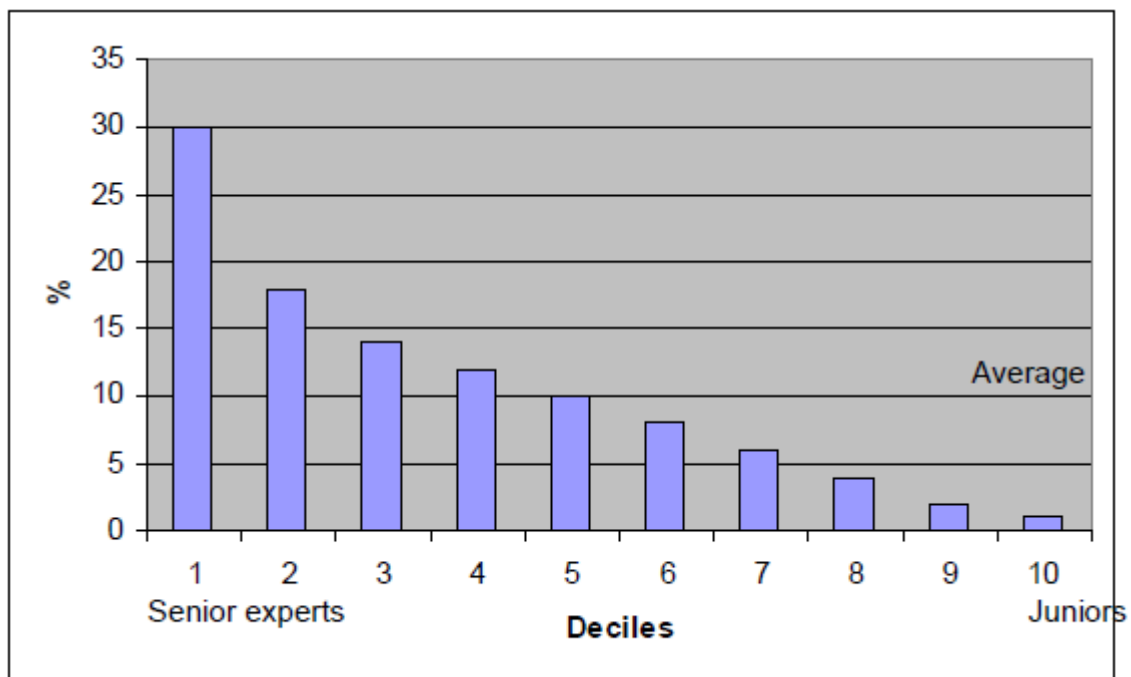
Expert atau kepakaran adalah derajat atau level dari kepakaran, umumnya, kepakaran seorang manusia terdiri dari aktivitas berikut ini:

- ✓ Mengenal dan merumuskan masalah.
- ✓ Menyelesaikan masalah secara cepat dan layak.
- ✓ Menjelaskan solusinya.
- ✓ Belajar dari pengalaman.
- ✓ Me-restrukturisasi *Knowledge*.
- ✓ Breaking rules.
- ✓ Menentukan relevansi.
- ✓ Menguraikan dengan bijak (sesuai dengan keterbatasannya).

Beberapa fakta mengenai kepakaran dapat dilihat berdasarkan fakta bahwa:

1. Kepakaran biasanya diasosiasikan dengan kecerdasan tingkat tinggi tetapi ini tidak mesti berhubungan dengan orang tercerdas.
2. Kepakaran biasanya diasosiasikan dengan kuantitas *Knowledge*.
3. Pakar belajar dari keberhasilan dan kegagalan masa lalu.
4. *Knowledge* dari seorang pakar disimpan dengan baik, diorganisasi, dan dapat dicari lagi dengan cepat.
5. Pakar dapat mengenali pola yang lebih tinggi dari pengalamannya (*excellent recall*).

Di bawah ini adalah diagram distribusi kepakaran:



2 tipe *Knowledge* yang dapat dibedakan adalah: fakta dan prosedur (biasanya rule) yang berkaitan dengan domain permasalahan.

Fitur khas dari *Expert System* adalah kemampuan untuk reasoning, kenyataan bahwa kepakaran disimpan dalam suatu *Knowledge base* dan bahwa program memiliki akses ke *database*, maka komputer diprogram sehingga dapat berinterferensi. Intererencing ini dilakukan

oleh komponen yang disebut *inference engine*, yang di dalamnya terdapat prosedur-prosedur yang berkaitan dengan penyelesaian masalah.

Di bawah ini adalah perbedaan antara *Conventional Systems* dan *Expert System*

Conventional Systems	Expert Systems
Information and its processing are usually combined in one sequential program.	Knowledge base is clearly separated from the processing (inference) mechanism (i.e., knowledge rules separated from the control).
Program does not make mistakes (programming do).	Program may make mistakes.
Do not (usually) explain why input data are needed or how conclusions were drawn.	Explanation is a part of most ES.
Changes in the program are tedious.	Changes in the rules are easy to accomplish.
The system operates only when it is completed.	The system can operate with only a few rules (as the first prototype).
Execution is done on a step-by-step (algorithmic) basis.	Execution is done by using heuristics and logic.
Effective manipulation of large databases.	Effective manipulation of large knowledge bases.
Representation and use of data.	Representation and use of knowledge.
Efficiency is a major goal.	Effectiveness is the major goal.
Easily deal with quantitative data.	Easily deal with qualitative data.
Capture, magnify, and distribute access to numeric data or to information.	Capture, magnify, and distribute access to judgment and knowledge.

Kebanyakan *Expert System* komersial menggunakan sistem yang berbasis rule (*rule-based system*) yaitu *Knowledge* disimpan dalam bentuk rule-rule, yang merupakan prosedur untuk menyelesaikan masalah.

Fitur unik lain dari *Expert System* adalah kemampuan untuk menjelaskan nasehat atau rekomendasi yang diberikan. Penjelasan dan justifikasi ini dilakukan oleh subsistem yang disebut dengan *justifier* atau *explanation subsystem*. Ini menjadikan sistem dapat memeriksa reasoningnya dan menjelaskan operasi-operasi yang dilakukan.

6.2 Struktur ES.

ES dibagi menjadi 2 bagian utama: *development environment* dan *consultation (runtime) environment*. Lingkungan pengembangan digunakan oleh *Expert System* builder untuk membangun komponen dan untuk membawa *Knowledge* ke dalam *Knowledge base*. Lingkungan

konsultasi digunakan oleh orang yang bukan ahli untuk mendapatkan *Knowledge* dan saran setara pakar.

Komponen yang ada dalam *Expert System*, antara lain:

1. *Knowledge acquisition subsystem [human experts, textbooks, research reports]*.
2. *Knowledge base [facts: situation & theory, heuristics or rules]*.
3. *Inference engine [interpreter, scheduler, consistency enforcer]*.
4. *Blackboard (workplace) [plan, agenda, solution]*.
5. *User*.
6. *User interface [friendly, problem oriented]*.
7. *Explanation subsystem (justifier) [what, how, why]*.
8. *Knowledge refining system [analyze performance, learn, improve]*.

Otak dari *Expert System* adalah *Inference engine*, disebut juga *control structure* atau *the rule interpreter* (pada *rule-based ES*). Komponen ini sebenarnya adalah program komputer yang menyediakan metodologi untuk reasoning mengenai informasi dalam *Knowledge base* dan dalam “*blackboard*”, dan digunakan untuk merumuskan kesimpulan. Adapun elemen utama dari *Inference engine* adalah:

1. *Interpreter*

(*rule interpreter* dalam kebanyakan sistem), yang mengeksekusi item agenda yang dipilih dengan mengaplikasikannya pada *Knowledge base rule* yang berhubungan.

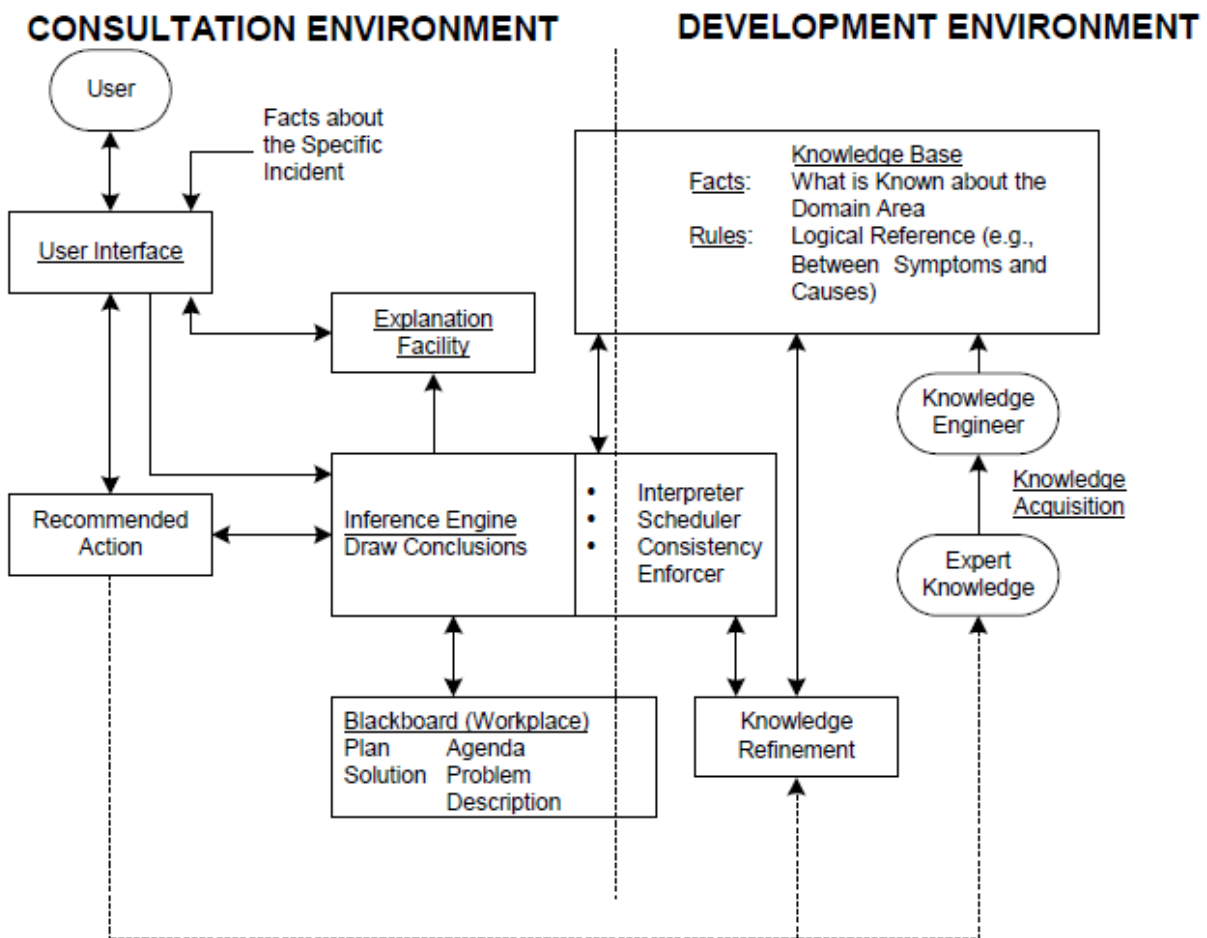
2. *Scheduler*

Adalah komponen yang menjaga kontrol di sepanjang agenda, memperkirakan akibat dari pengaplikasian *inference rule* yang menampilkan prioritas item atau kriteria lain pada agenda.

3. Consistency enforcer

Merupakan komponen yang mencoba menjaga konsistensi representasi solusi yang muncul.

Di bawah ini adalah diagram struktur dari *Expert System*:



6.3 Explanation Subsystem (Justifier)

Kemampuan untuk melacak kebenaran dari kesimpulan yang didapat dari sumber-sumbernya merupakan hal yang krusial baik dalam transformasi kepakaran maupun dalam penyelesaian masalah. Bagian ini dapat melacak kebenaran dan menjelaskan perilaku *Expert System* dengan secara interaktif menjawab pertanyaan seperti ini:

1. Why was a certain question asked by the expert system?

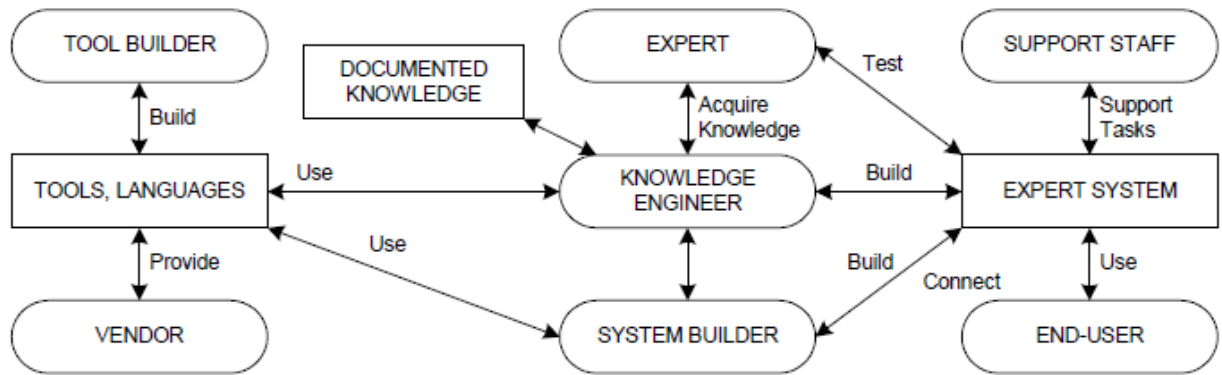
2. Why was a certain alternative rejected?
3. What is the plan to reach the solution?

For example, what remains to be established before a final diagnosis can be determined?

6.4 Elemen Manusia Dalam *Expert System*

Elemen manusia merupakan elemen yang sangat penting dan merupakan salah satu elemen utama dalam sistem ini, elemen manusia merupakan orang-orang yang terlibat dalam pengembangan dan penggunaan *Expert System*, yaitu:

1. Pakar.
2. *Knowledge Engineer*
3. *User*, yang terdiri dari:
 - ✓ Klien yang bukan pakar yang menginginkan nasehat langsung. Disini *Expert System* bertindak sebagai konsultan atau advisor/penasehat.
 - ✓ Pelajar yang ingin belajar. *Expert System* disini bertindak sebagai instruktur.
 - ✓ *Expert System builder* yang ingin meningkatkan *Knowledge base*-nya. Disini *Expert System* bertindak sebagai partner.
 - ✓ Pakar. *Expert System* disini bertindak sebagai kolega atau sebagai asisten.
 - ✓ Pihak lain. Misalnya: system builder, tool builder, vendor, staf pendukung. Lengkapnya dapat dilihat pada bagan di bawah ini:



Adapun bidang permasalahan yang ditangani *Expert System*, dapat dilihat pada tabel berikut, yang disajikan dalam kategori umum dari *Expert System*

Category	Problem Addressed
Interpretation	Inferring situation description from observations
Prediction	Inferring likely consequences of given situations
Diagnosis	Inferring system malfunctions from observations
Design	Configuring objects under constraints
Planning	Developing plans to achieve goal(s)
Monitoring	Comparing observations to plans, flagging exceptions
Debugging	Prescribing remedies for malfunctions
Repair	Executing a plan to administer a prescribed remedy
Instruction	Diagnosing, debugging, and correcting student performance
Control	Interpreting, predicting, repairing, and monitoring system behaviors

Ada beberapa alasan dan berbagai keuntungan potensial yang bisa diperoleh dari *Expert System*, antara lain:

1. Meningkatkan output dan produktivitas.
2. Meningkatkan kualitas.
3. Mengurangi waktu kerusakan (downtime).

Sedangkan permasalahan dan keterbatasan yang menghambat *Expert System* yaitu:

1. *Knowledge* tak selalu tersedia

2. Kepakaran sulit diekstraksi dari manusia.
3. Pendekatan untuk setiap pakar pada situasi tertentu selalu berbeda, dan tak mesti benar.
4. Walaupun pakar tersebut memiliki ketrampilan yang tinggi, sukar untuk mengabstraksikan kepakarannya pada situasi tertentu, apalagi pakar tersebut bekerja di bawah tekanan.
5. User dari *Expert System* memiliki batasan kognitif alamiah.
6. *Expert System* bekerja baik hanya pada domain yang terbatas/sempit.
7. Kebanyakan pakar tak memiliki rasa pengertian pengecekan yang independen walaupun konklusi mereka masuk akal.
8. Kosa kata, atau jargon, yang digunakan pakar untuk mengekspresikan fakta dan relasinya biasanya jarang digunakan dan dimengerti oleh orang lain.
9. Help seringkali dibutuhkan oleh *Knowledge engineer* yang biasanya jarang tersedia dan mahal biayanya – sebuah fakta yang dapat membuat pembangunan *Expert System* lebih banyak memakan biaya.
10. Kendala kepercayaan pada end-user bisa menghalangi penggunaan ES.
11. Transfer *Knowledge* bergantung pada persepsi dan bisa bias dalam prasangka.

6.5 Jenis *Expert System*

Expert System muncul dalam berbagai variasi, adapun jenis-jenis dari *Expert System*, yaitu:

1. Expert System vs. Knowledge-based Systems

Expert System mendapatkan *Knowledge*-nya dari para pakar, sedang KBS dari sumber-sumber terdokumentasi. KBS lebih murah dan lebih cepat dibangun dibandingkan ES.

2. Rule-based Expert Systems

Knowledge direpresentasikan sebagai serangkaian rule-rule (production rules).

3. *Frame-based Systems*

Knowledge direpresentasikan sebagai frame, yaitu representasi dari pendekatan Pemrograman Berbasis Objek (OOP).

4. *Hybrid Systems*

Melibatkan berbagai pendekatan representasi *Knowledge*, paling tidak frame dan rule, tapi biasanya lebih dari itu.

5. *Model-based Systems*

Tersusun di sekitar model yang mensimulasikan struktur dan fungsi dari sistem yang dipelajari. Model digunakan untuk menghitung nilai-nilai, yang dibandingkan dengan sedang diamati. Perbandingan tersebut memicu aksi (jika diperlukan) atau diagnosis lebih lanjut.

Sistem yang diklasifikasikan oleh sifat alamiahnya. Ada 3 jenis, yaitu

- ✓ berhubungan dengan *evidence gathering* (pengumpulan bukti-bukti)
- ✓ *stepwise refinement system*. Sistem ini berhubungan dengan sejumlah besar keluaran dari level-level detil sesudahnya.
- ✓ *Stepwise assembly*, dimana domain subjek dapat mempunyai jumlah yang luar biasa besar keluaran yang mungkin. Jenis khusus dari ini disebut dengan catalog selection. Sistem ini berhubungan dengan masalah seperti pemilihan bahan kimia, baja yang benar.

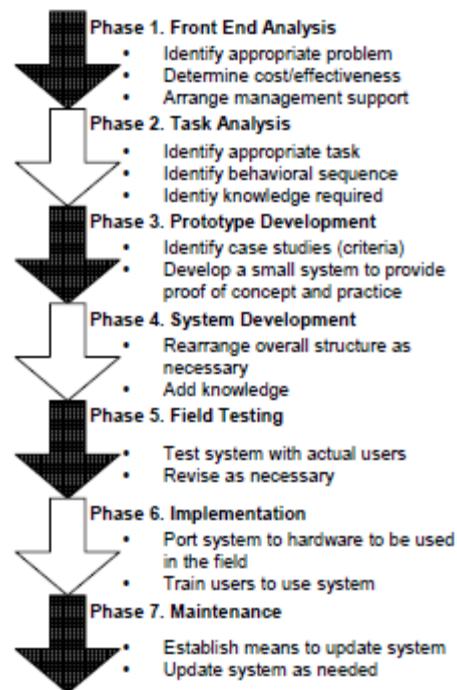
6. Sistem siap pakai (*Off-the-Shelf Systems*)

Sebagai hasil dari produksi massal membuatnya lebih murah dibandingkan dengan sistem yang memenuhi keinginan user (customized system). Sayangnya sistem ini bersifat terlalu umum, dan nasehat/advis yang dihasilkan mungkin tak bernilai pada user yang dihadapkan

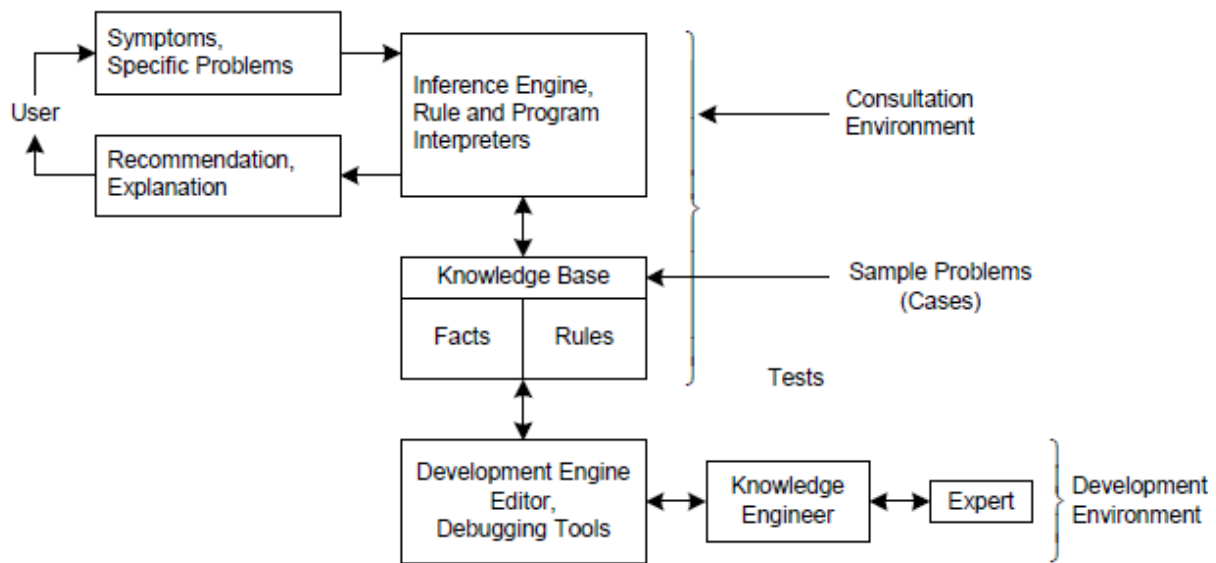
pada situasi yang kompleks. mengontrol proses terkomputerisasi. Sistem selalu menghasilkan respon sesuai waktu yang diperlukan.

6.6 Pengembangan *Expert System*

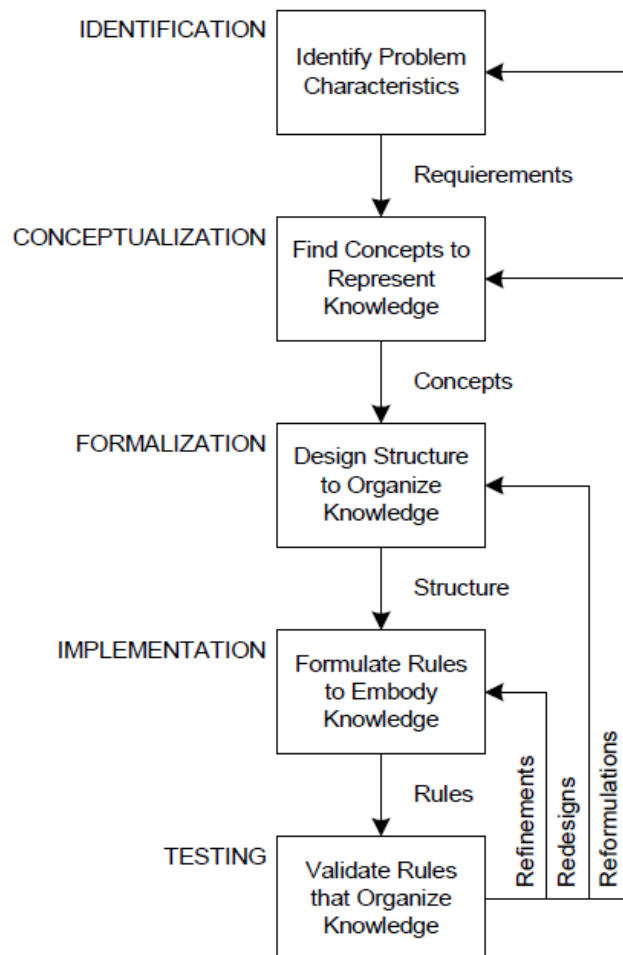
7 langkah yang diperlukan dalam pengembangan *Expert System* dapat dilihat pada diagram di bawah ini:



Membangun *Expert System* yang spesifik dengan suatu shell dapat digambarkan dalam diagram di bawah ini:



Tahapan-tahapan dalam mengakuisisi *Knowledge* digambarkan dalam diagram ini:



Sedangkan proses pengembangan *Expert System* itu sendiri digambarkan sebagai berikut:

