Knowledge Based System

Informatics Engineering Study Program
School of Electrical Engineering and Informatics

Institute of Technology Bandung

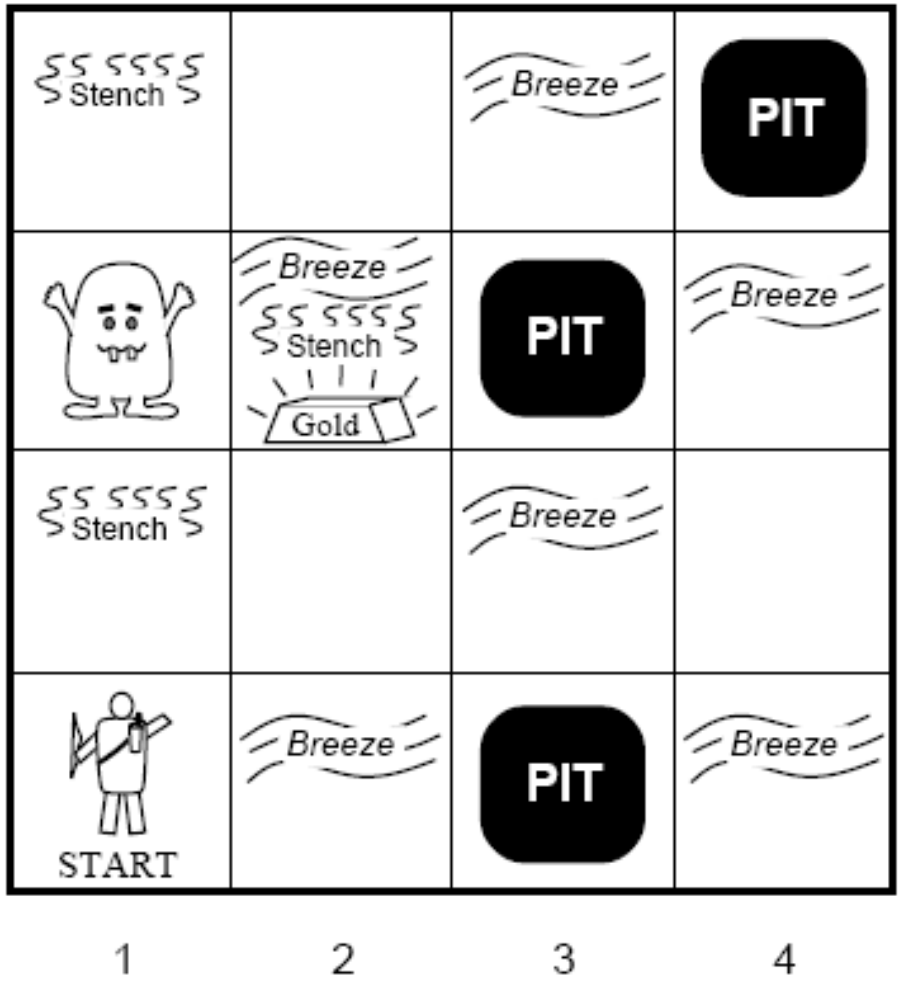
Overview

- Review
- Knowledge-based System
  - Fundamental properties of logical reasoning
    - Facts+knowledge➔conclusion
    - Guarantee: correct conclusion from correct facts
  - Knowledge representation
  - Knowledge reasoning
  - KBS development: declarative approach, problem
  - KBS vs ES vs conventional program
  - KBS: interactive, embedded, basic vs general architecture, ill-structured problem
  - Metode pemecahan masalah: klasifikasi vs konstruksi
- KBS Examples
- Knowledge Engineering

Review: Simple Problem Solving Agent

- Problem Solving Agent handles finite states (i.s – goal)
- States in path finding problem: agent locations
  e.g.: in Arad, in Bucharest
- States in CSP: set variables X_i with values from domain D_i
  e.g.: {}, {WA=red, NT=green, Q=red, SA=blue, NSW=green, V=red, T=green}
- Local search: hill climbing, simulated annealing, GA
- Informed search enables problem solving agents to perform well (with admissible heuristics)
  - This knowledge is very specific and inflexible
  - General knowledge and reasoning → knowledge –based agent

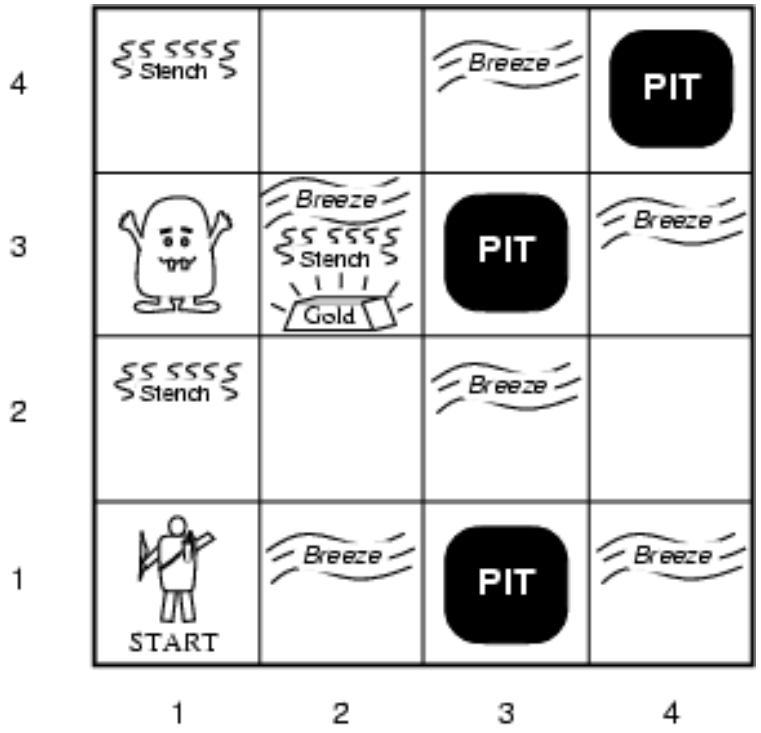
The Wumpus World



- Cave with rooms
- Wumpus eats anyone who enters its room
- Wumpus can be shot by an agent, but the agent has only one arrow
- Pit will trap anyone, except for the wumpus
- Agent can find gold heap

The Wumpus World: Task Environment

- Performance measure
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- Environment
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square



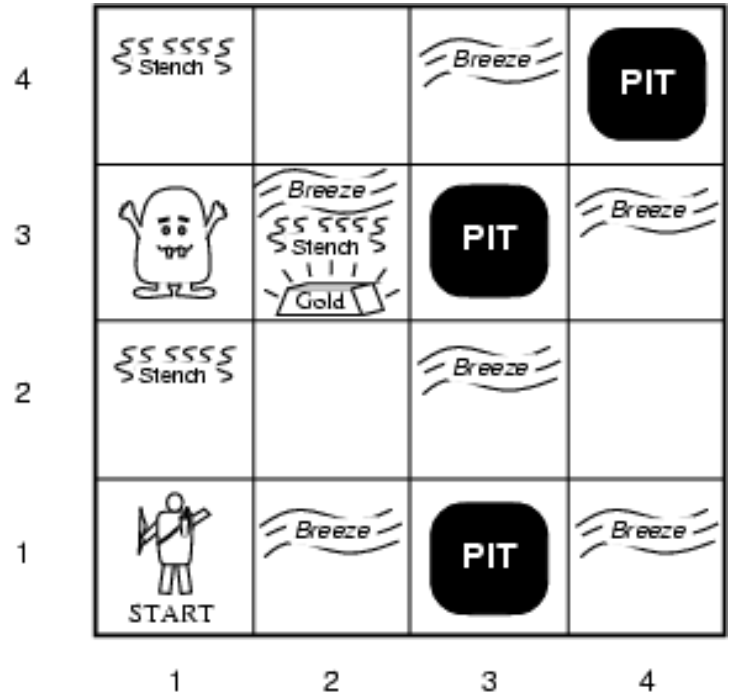
- Sensors: Stench, Breeze, Glitter, Bump, Scream
- Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot

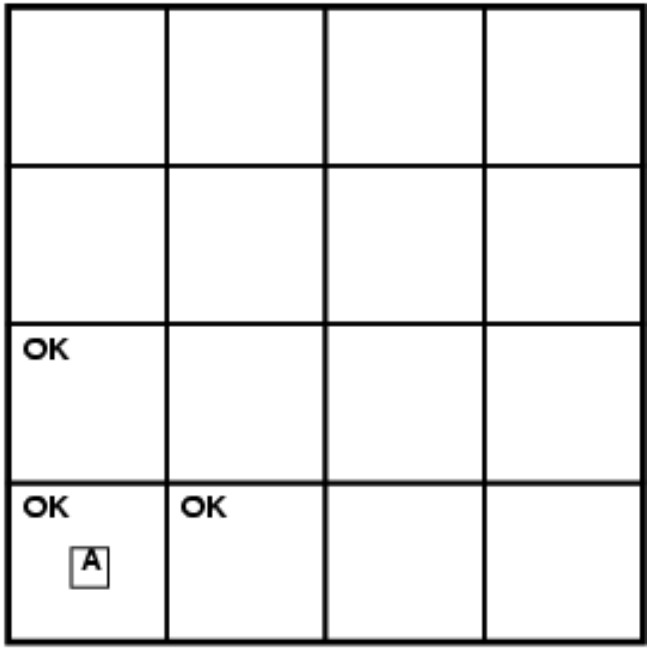
Wumpus world characterization

- Fully Observable ?
- No – only local perception
- Deterministic ?
- Yes – outcomes exactly specified
- Episodic ?
- No – sequential at the level of actions
- Static ?
- Yes – Wumpus and Pits do not move
- Discrete ?
- Yes
- Single-agent ?
- Yes – Wumpus is essentially a natural feature

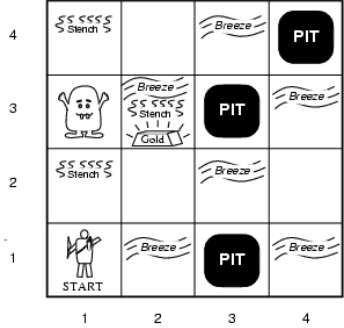
Exploring a wumpus world

[1,1] : OK (safe)
Percept [1,1] : [None, None, None, None, None]
No stench in [1,1] : No wumpus in [1,2] and [2,1]
No breeze in [1,1]: No pit in [1,2] and [2,1]
Action: forward to [2,1]





Exploring a wumpus world (2)



Percept [2, 1] : [None, Breeze, None, None, None]
No stench in [2,1] : No wumpus in [3,1] and [2,2]
Breeze in [2,1]: there must be a pit in [3,1] or [2,2]
Set action: go back to [1,1] and forward to [1,2]

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

(a)

A = *Agent*
B = *Breeze*
G = *Glitter, Gold*
OK = *Safe square*
P = *Pit*
S = *Stench*
V = *Visited*
W = *Wumpus*

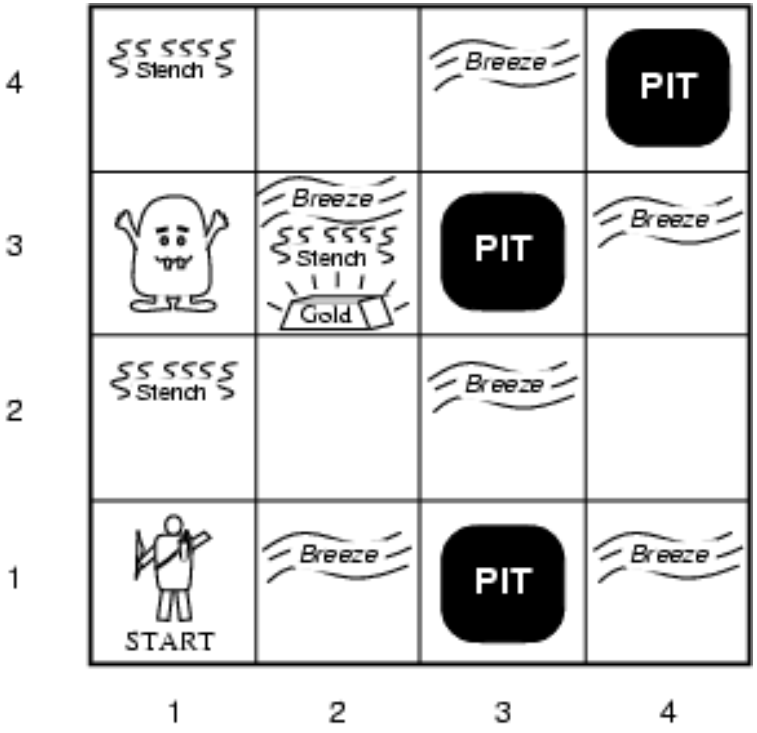
1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 A B OK	3,1 P?	4,1

(b)

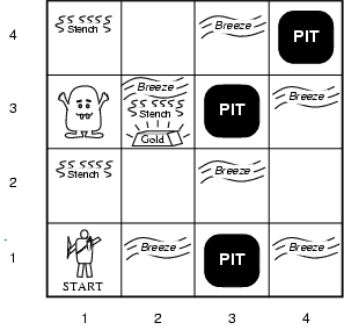
Exploring a wumpus world (3)

Percept [1,2] : [Stench, None, None, None, None]
Stench in [1,2] : there must be a wumpus in [1,3] or [2,2] or [1,1]
No wumpus in [1,1] and No stench in [2,1] → wumpus in [1,3]
No breeze in [1,2]: No pit in [1,3] and [2,2] → pit in [3,1] and [2,2] OK
Set action: go to [2,2]

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1



Exploring a wumpus world (4)



Percept [2,2] : [None, None, None, None, None]
No stench in [2,2] : No wumpus in [2,3] and [3,2]
No breeze in [2,2]: No pit in [2,3] and [3,2]
Set action: go to [2,3]

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Percept [2,3] : [Stench, Breeze, Glitter, None, None]
Action: Grab

The Wumpus World: Summary

- Fundamental properties of logical reasoning
  - In each step, the agent draws a conclusion from available information
  - Conclusion is **guaranteed** to be correct if the available information is correct
- Knowledge-based agent

Simple Knowledge Based Agent

- Agent design: declarative approach
- TELL KB what it needs to know
- ASK itself what to do -- answers should follow from the KB

```
function KB-AGENT(percept) returns an action
    static: KB, a knowledge base
            t, a counter, initially 0, indicating time

    TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
    action ← ASK(KB, MAKE-ACTION-QUERY(t))
    TELL(KB, MAKE-ACTION-SENTENCE(action, t))
    t ← t + 1
    return action
```

Intro to Knowledge Representation

- Instead of thinking about all the ways a world could be, we're going to work in a language of expressions that describe those sets

- It's one way of representing knowledge

Intro to Knowledge Representation (2)

- A language (to represent knowledge/ information)

  a set of syntactic and semantic conventions that makes it possible to describe things, and a way of manipulating expression in language

- Syntax: a description of what you're allowed to write down, what the expressions are, that are legal in a language.
- Semantic: which is some story about what those expressions mean.
- In short: Syntax is form and semantics is content.

Intro to Knowledge Representation (3)

- Examples:
  - Map → symbols, interpretation of symbols to represent real geographic condition
  - Natural Languages → collection of symbols to explain things
- Objectives of selection KR:
  - Processing → as simple as possible
  - Represent real-world problems into more comprehensible problems

Intro to Knowledge Representation (4)

- The representation should be:
  - Suitable for problem domain
    - Decision tree for classification
    - Skeletal construction for construction
    - Rule for all problem domain
  - Suitable for the tasks (inference)
    - Decision tree including interview process
    - Probability model for decision with uncertainty
  - Suitable for users (man or machine)
    - Semantic network for user, rule for machine

Intro to Knowledge Representation (5)

- Requirements of knowledge representation:
  - No contradiction
  - Each symbol must be unique
  - Explain certain objects, relations and attributes
  - Efficient manipulation in computer system

- Several examples → application oriented
  - *Logic: robotics*
  - *Production rules*: expert systems
  - *Semantic network, frame*: structured object representation → story understanding system
    - Information extraction

Reasoning: Deduction, Induction ?

1. All men are mortal
2. If it is raining then the streets are wet
3. This swan is white;
4. That swan is white;
5. Socrates is a man
6. Every swan that I've ever seen is white;
7. It is raining
8. The sun The sun has risen every day so far

1. Socrates is mortal
2. All swans are white
3. The sun will rise tomorrow
4. The streets are wet

Knowledge Representation Manipulation (3)

- Abduction

  1. If a person has a cold, then he has a runny nose;

     Jack has a runny nose;

     Therefore Jack has a cold

  - possibility of wrong conclusion
  - practical reasoning → diagnosis

Review: Generic Knowledge-based Agent

- Knowledge base is the central component of knowledge based agent / system → KBS
- Knowledge base: background knowledge
  - Input: percept; Output: action

  function KB_AGENT(percept) → action

  TELL(KB,percept) {assert percepts}

  action ← ASK (KB) {reasoning}

  TELL (KB, action) {assert action}
- Percepts and reasoning results (including action) is stored in working memory

Knowledge-based Agent Development

- A knowledge-based agent can be built by declarative approach
  - Starting with an empty knowledge-base
  - Agent designer can TELL sentences one by one until the agent knows how to operate in its environment
- Problem:
  - The designers cannot anticipate all possible situations
  - The designers cannot anticipate all changes over time
  - The designers have no idea about the solution

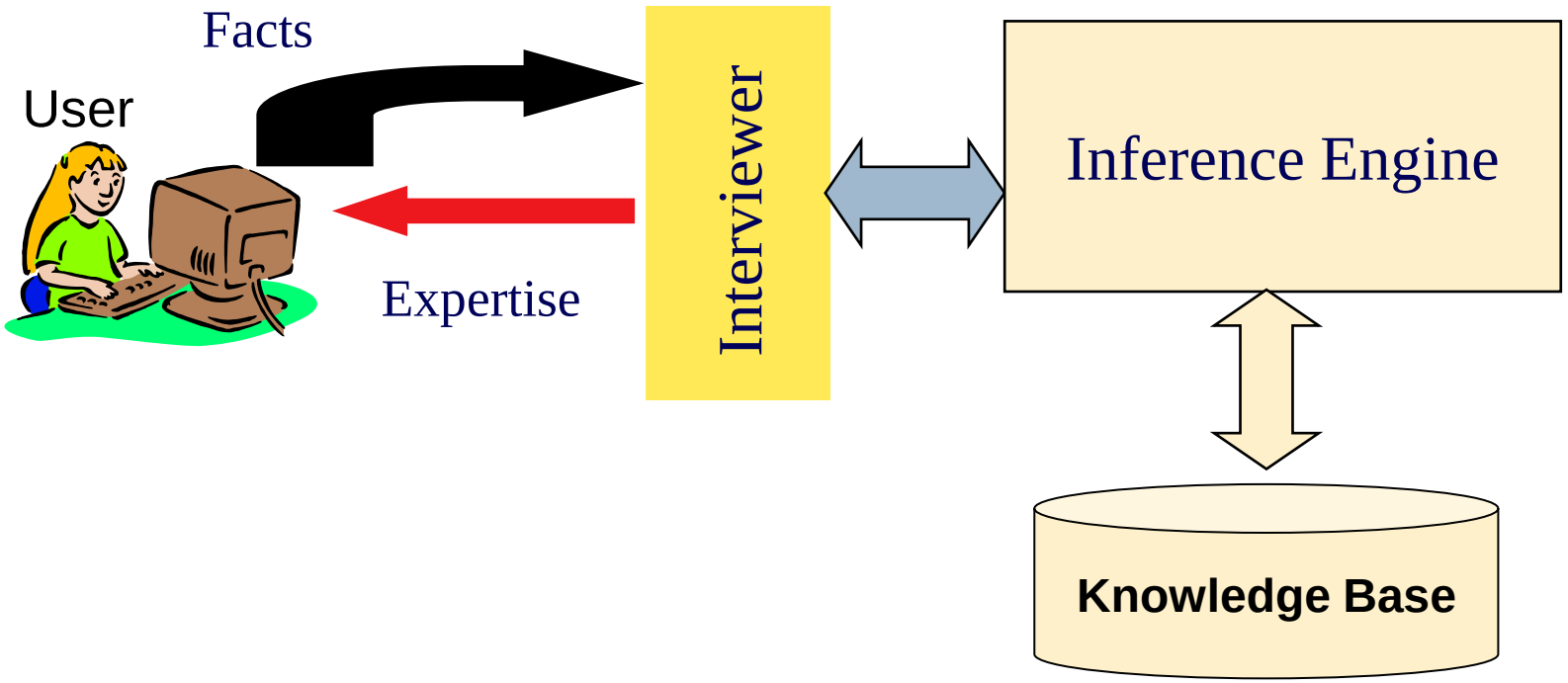
Knowledge-based System (KBS/SBP)

- Sistem yang melakukan task dengan mengaplikasikan pengetahuan dalam representasi simbolik
- SBP vs sistem pakar
- Sistem pakar:
  - sistem komputer yang meniru kemampuan pengambilan keputusan pakar pada domain tertentu
  - Sumber pengetahuan sistem pakar: pakar manusia
  - Domain sistem pakar: persoalan dunia nyata

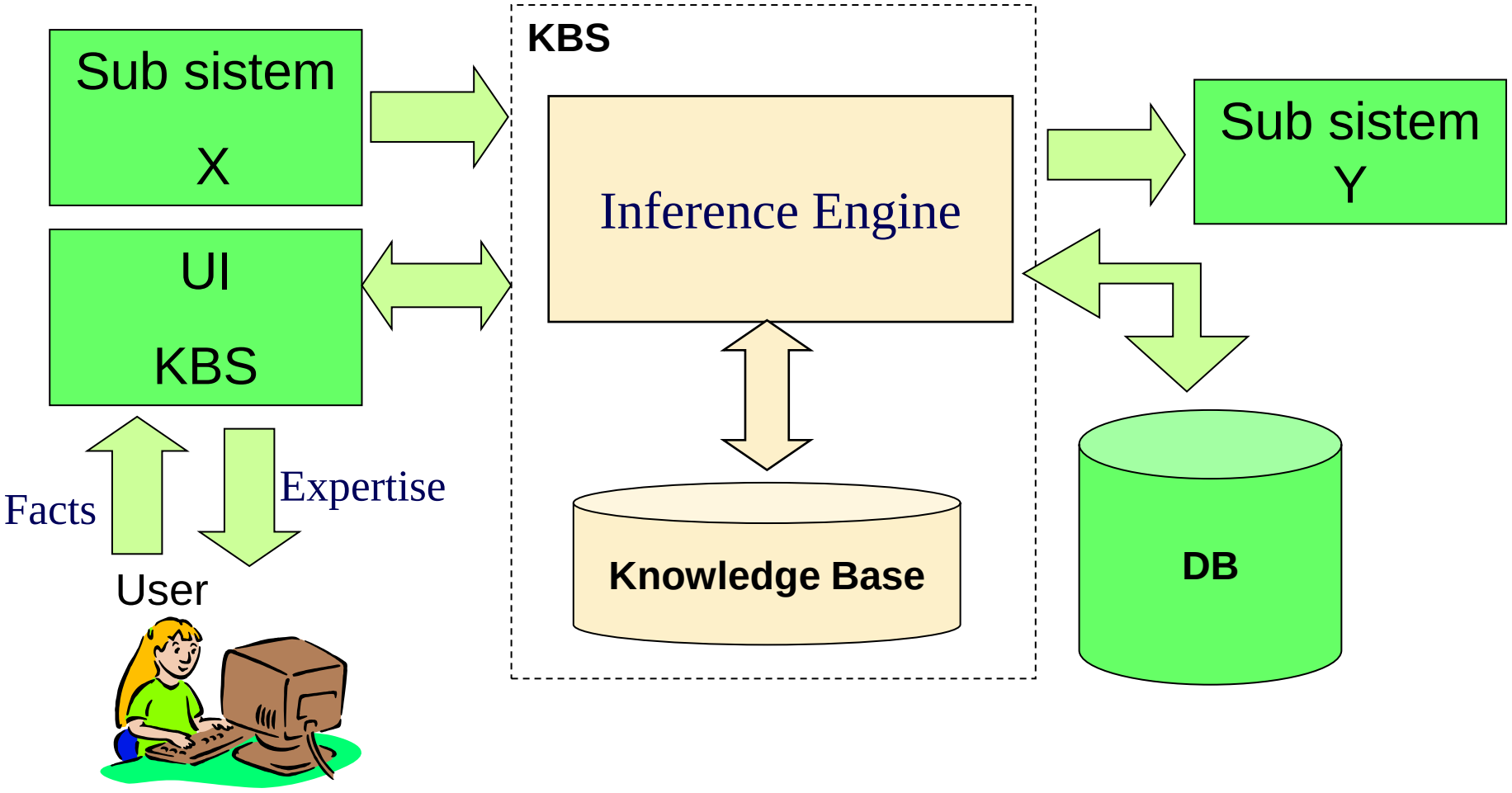
KBS vs Program Konvensional

Program Konvensional	KBS
algoritma + data Contoh: Penghitungan IPK	**metode pemecahan masalah + domain knowledge + data** Contoh: diagnosis penyakit, diagnosis kerusakan mobil
Programmer menentukan apa yang harus dilakukan dan urutan yang harus dilakukan	Pakar menentukan aksi, urutan ditentukan oleh interpreter

Interactive KBS

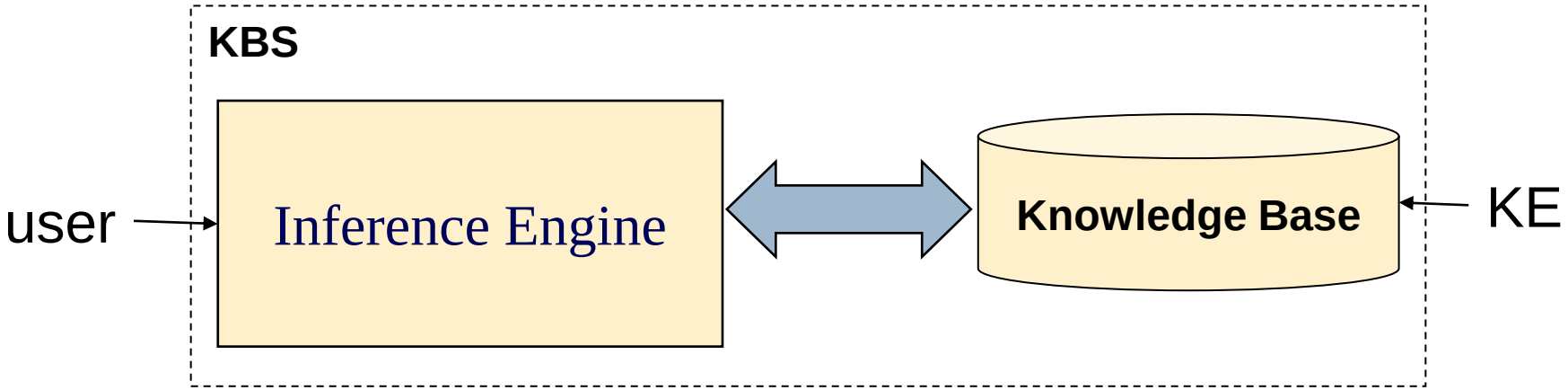


Embedded KBS

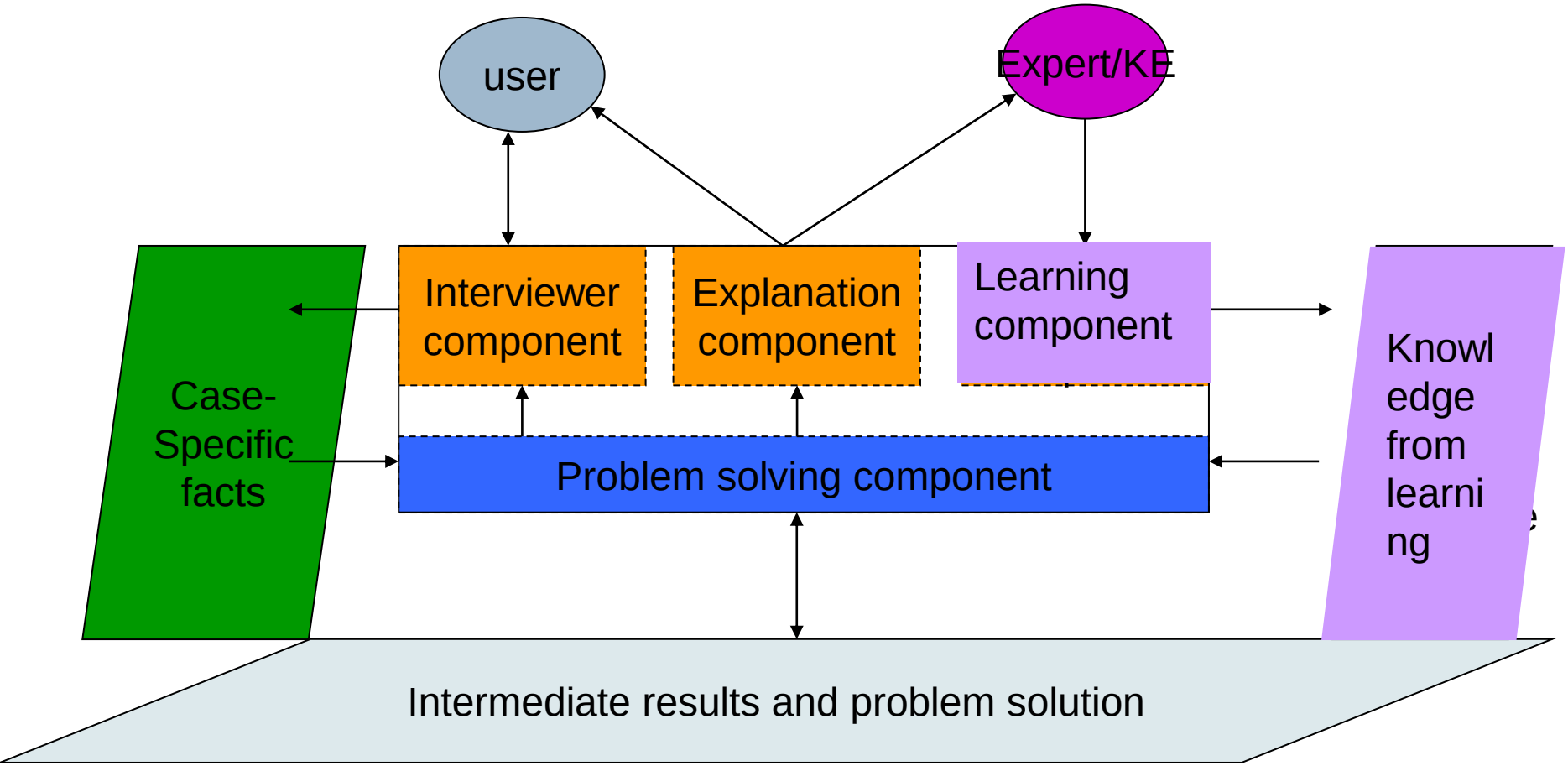


Arsitektur Dasar KBS

- Arsitektur: modul-modul program + hubungan antar modul
- Arsitektur Dasar:



Arsitektur Umum KBS



UTS 2009/2010

- Pada arsitektur umum knowledge-based system, terdapat komponen domain specific expert knowledge, case-specific facts, dan intermediate results. Jelaskanlah minimal 2 perbedaan ketiga komponen tersebut !

Perbedaan domain specific expert knowledge, case-specific facts, dan intermediate results

	DSEK	CSF	IR
Sumber isi	Hasil akuisisi pengetahuan dari pakar atau learning	Persepsi lingkungan (masukan dari user)	Hasil inferensi DSEK, persepsi, IR
Isi	Pengetahuan/pola, fakta domain	Fakta	Fakta
Sifat	Statik	Dinamik	Dinamik
Komponen terkait	Komponen KA, Mesin inferensi	Interviewer, Mesin inferensi	Mesin Inferensi

Domain KBS

- Ill-structured/ill-defined/messy problem
  - Problem: well formed vs ill-structured
  - Well formed → solusi: program konvensional
    Contoh: problem matematika/sains
  - Ill-structured → solusi: SBP
    Contoh: rencana liburan
- Domain-well bounded
  - terbatas dan spesifik

Ill-structured problem: Contoh Ekstrim

Travel agent's questions	Responses
Can I help you ?	I'm thinking about going somewhere
Where do you want to go?	I'm not sure where to go
Any particular destination?	I just like to travel; destination's not important
How much can you afford ?	I don't have enough money to go
Can you get some money?	I don't know how to get the money
When do you want to go?	I must go soon.

Ill-structured problem: Karakteristik

Responses	Characteristic
I'm thinking about going somewhere	Goal not explicit
I'm not sure where to go	Solution space unbounded
I just like to travel; destination's not important	Problem states not discrete
I don't have enough money to go	Intermediate states difficult to achieve
I don't know how to get the money	State operator unknown
I must go soon.	Time constraint

Problem Characteristics

	Well-formed Problem	Ill-structured Problem
Goal	Explicit	Not explicit
Solution space	bounded	unbounded
Solution	Exact/certain	Uncertain
Problem states	Discrete	Not discrete
State operator	Explicit, deterministic	Unknown

Problem Category

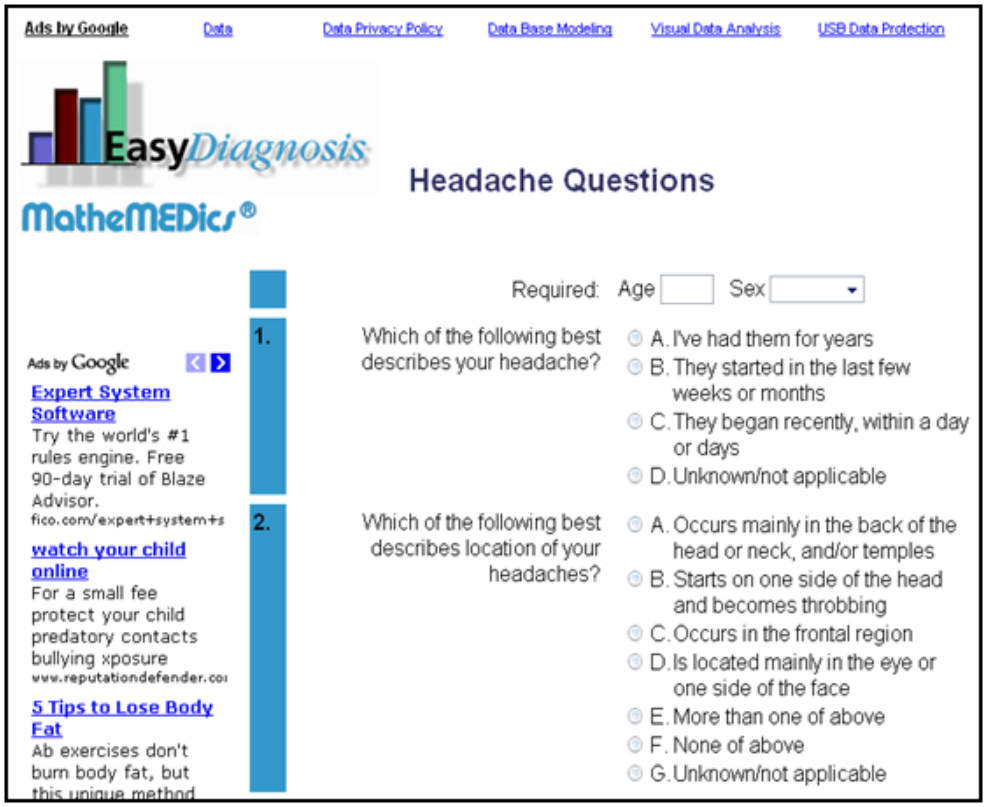
- Kelas masalah → metode pemecahan masalah → representasi dan inferensi
- Kategori metode pemecahan masalah:
  - Klasifikasi → classifier
    - Solusi dipilih dari set kelas masalah yang sudah didefinisikan
    - Pemetaan set observasi ke set solusi
  - Konstruksi
    - Solusi disusun dari elemen solusi

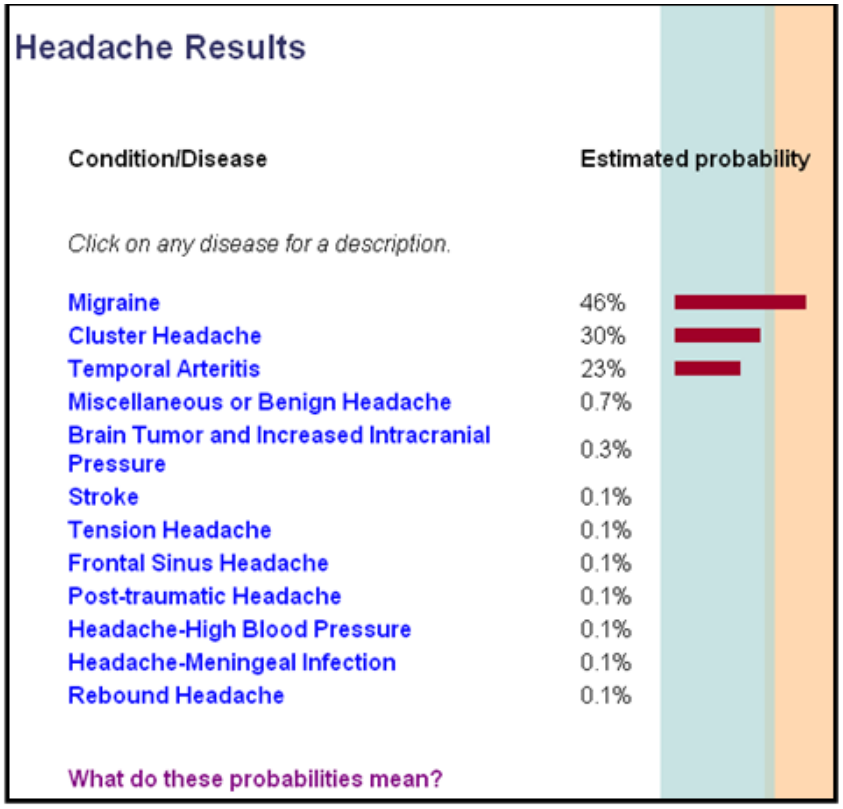
KBS Examples

Contoh Aplikasi

- Kesehatan: BAL2000, LISA, ISABEL, CTSHIV, DxPlain, MedWeaver, The Analyst, FuzzyFluid, Casnet, PUFF, Centaur, EasyDiagnosis, CLEM, VIE-PNN
- Lingkungan: ESS-WWTP, CREWS, CORMIX, HITERM, GCES, Oncologic
- Jaringan: NIDES, AudES, eXpert-BSM, Expert Advisor, Online ES (listrik)
- ITS: ActiveMath, TEST, ELM-ART, SID2002 Math ES, Chest
- Komputer/HW: DART, PEARL, PDAmum
- Manajemen: DXMAS, CESA, FINEVA
- Permainan: FRES, Rogomatic
- Geologi: PROSPECTOR II, DAS
- Pertanian: EXSEL, HABES, DSS4Ag
- Biologi: RIH, PSORTb
- NASA: Weather ES, SHINE
- Lainnya: TTA (teroris), ACAS-PRO (kartu kredit), USLIMITS 2, CATD-RT, HWYCON, SHYSTER (hukum)

EasyDiagnosis Medical Expert System



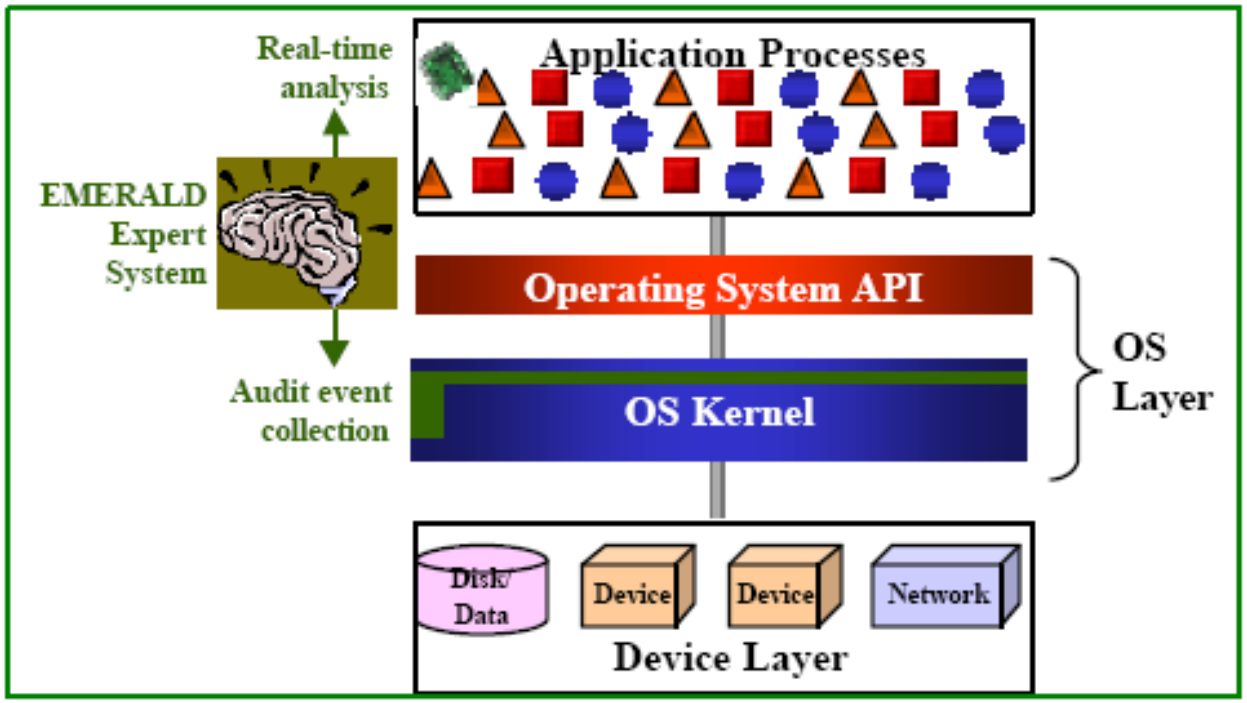


Green Chemistry Expert System (GCES)

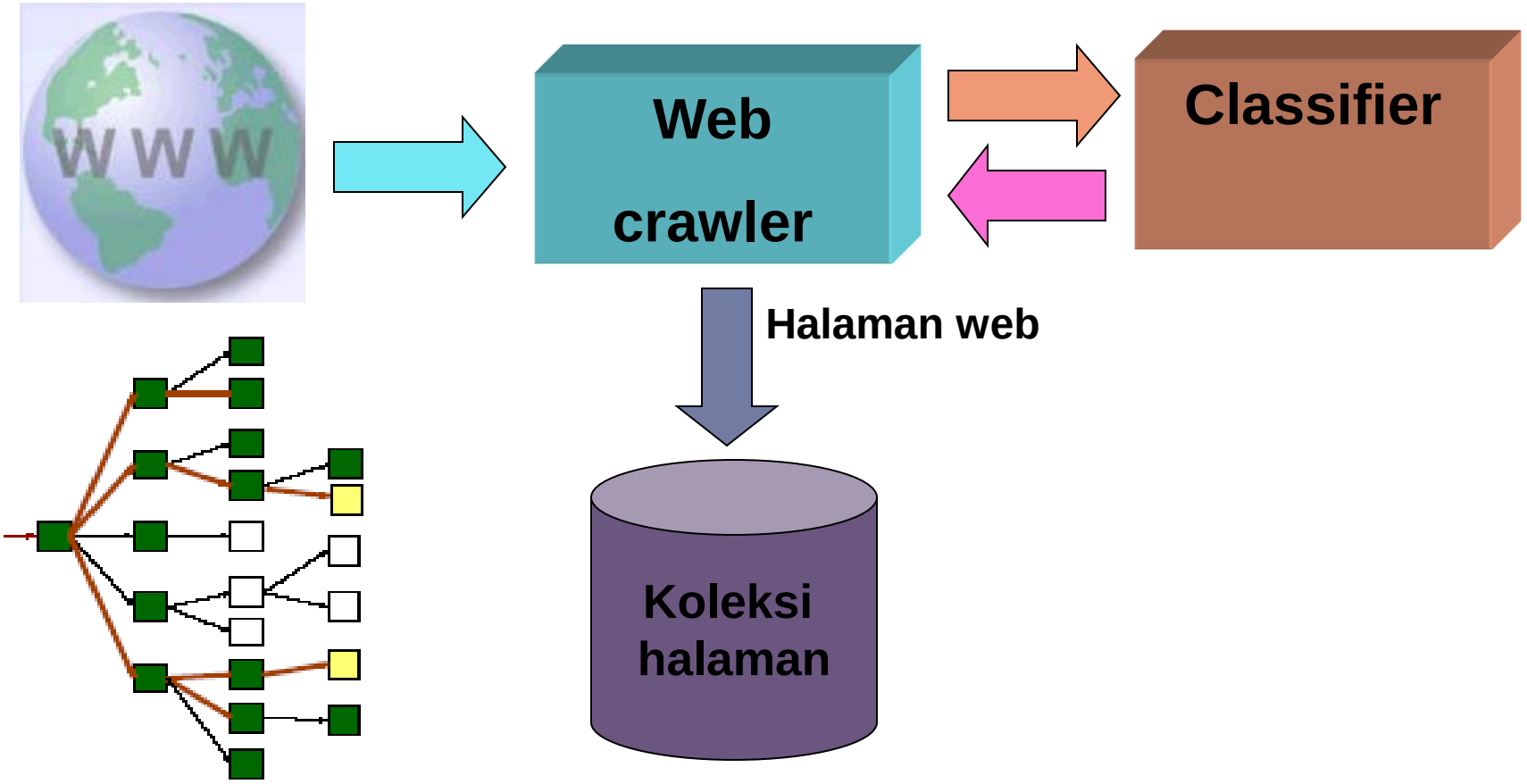
- Developer: EPA (*Evironmental Protection Protection Agency*) Amerika Serikat
  - MS Access, DBMS
- untuk menilai substansi yang berbahaya dalam reaksi kimia sehingga polusi dapat dicegah
- http://www.epa.gov/greenchemistry/pubs/gces.html

eXpert-BSM

- Intrusion Detection Solution for Sun Solaris
- Output: hasil analisis dan alert adanya intrusi pada audit trail dari Sun Solaris
- Sub sistem Emerald ES

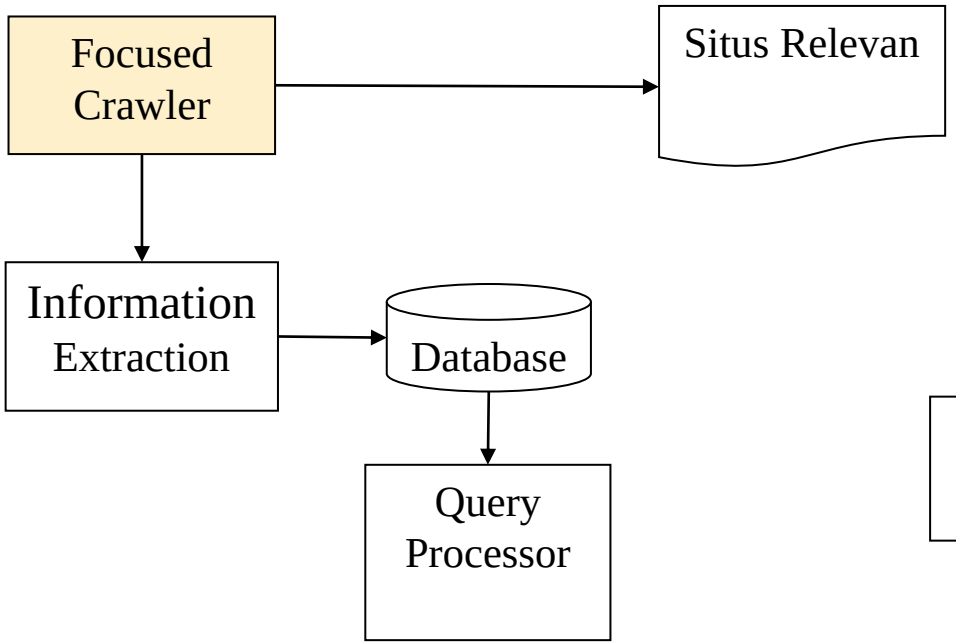


Focused Crawler Domain X



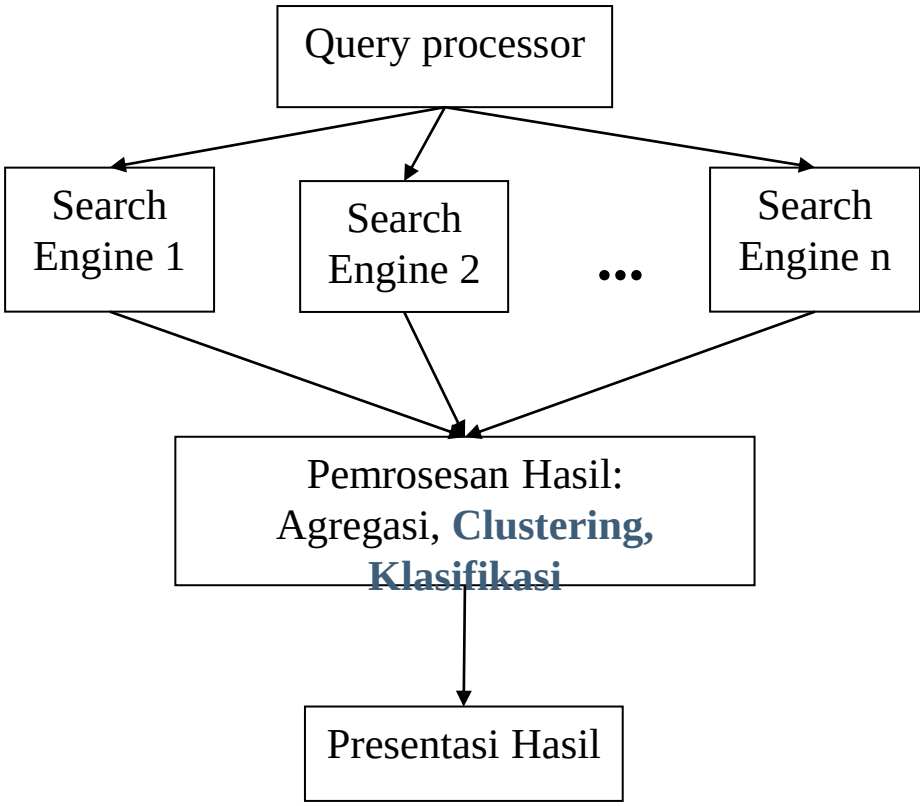
Vertical Search Engine

Search Engine: Architecture



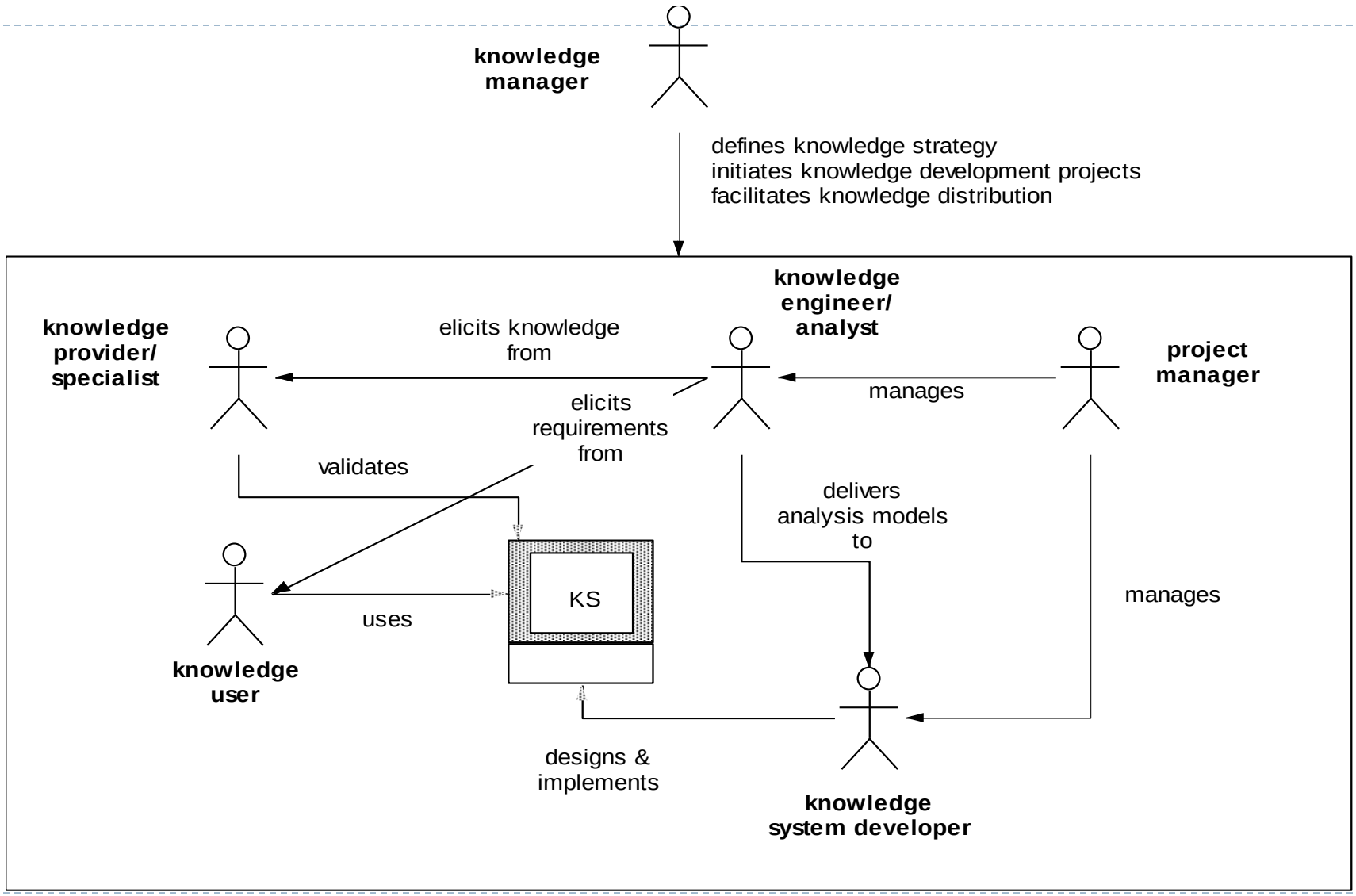
Vertical search engine

Meta search engine



Knowledge Engineering

KBS Developer



Rekayasa Pengetahuan

- Akuisisi pengetahuan dalam suatu domain dari satu atau lebih sumber non-elektronik dan konversinya ke dalam suatu bentuk yang dapat digunakan oleh komputer untuk memecahkan persoalan yang umumnya hanya dapat dipecahkan oleh pakar domain tersebut.

Akuisisi Pengetahuan (KA)

- KA=knowledge ***elicitation*** + ***representation***
- knowledge elicitation
  - Proses ekstraksi pengetahuan domain dan strategik dari pakar
  - Interview antara KE dan pakar
  - a cyclical process
- Knowledge representation
  - Proses merepresentasikan pengetahuan hasil ekstraksi ke suatu bentuk formal

Task dalam Knowledge Elicitation

- Pada setiap iterasi:
  - ***collect*** knowledge (e.g. from expert)
  - determine ***key concepts*** in problem domain
  - establish ***relationships*** between various concepts in problem domain
  - decide h***ow knowledge is represented*** in KBS
  - determine what knowledge needs to be collected in the next cycle

Tahapan Akuisisi Pengetahuan

- Identification
  - Identifikasi karakteristik masalah
- Conceptualization
  - Menemukan konsep2 untuk merepresentasikan pengetahuan
- Formalization
  - Design struktur untuk mengorganisasikan pengetahuan
- Implementation
  - Formulasi pengetahuan ke bentuk runnable program
- Testing
  - Validasi pengetahuan

Teknik Akuisisi Pengetahuan

- Manual:
  1. Interview
  2. Observasi
  3. Intuitive: tukar peran Knowledge Engineer dan pakar
- Otomatis:
  - Menggunakan tools untuk memfasilitasi akuisisi
  - Tools untuk pakar
  - Tools machine learning

Keywords

- Knowledge based system, expert system
- KBS: interactive, embedded
- Inference engine, knowledge base
- Component: Interviewer, explanation, knowledge acquisition, learning
- Classification, construction
  - Automatic text summarization: classification vs construction
- Knowledge engineering, knowledge acquisition
- Knowledge elicitation, knowledge representation

Exercise

- KBS ? Klasifikasi atau konstruksi ?
  - Sistem prediksi penghasilan seorang pekerja (≥$50K, <$50K) dengan melihat rencana kerja dan jumlah jam kerja, serta rate per jam yang berlaku.
  - Sistem penyusun menu makan siang dengan memilih paket menu yang tersedia.
  - Sistem penyusun menu makan siang dengan memilih makanan utama (nasi/kentang), lauk (ayam/daging/ikan/telur), sayur (sop/tumis/lalap), buah (jeruk/apel/melon), dan minuman (air putih/jus/soda).
  - Sistem pemberi nilai jawaban essay berdasarkan persentase kemunculan kata kunci yang telah ditentukan oleh pemberi soal.
  - Focused crawler yang menentukan apakah suatu halaman web relevan untuk suatu domain tertentu dengan melihat pola kemunculan kata yang ada pada halaman web tersebut.

Solution

- Sistem prediksi penghasilan seorang pekerja (≥$50K, <$50K) dengan melihat rencana kerja dan jumlah jam kerja, serta rate per jam yang berlaku. → **Bukan Klasifikasi**
- Sistem penyusun menu makan siang dengan memilih paket menu yang tersedia. → **Klasifikasi; Kelas: paket menu**
- Sistem penyusun menu makan siang dengan memilih makanan utama (nasi/kentang), lauk (ayam/daging/ikan/telur), sayur (sop/tumis/lalap), buah (jeruk/apel/melon), dan minuman (air putih/jus/soda). → **Konstruksi; Elemen solusi: makanan utama, lauk, sayur, buah, minuman**
- Sistem pemberi nilai jawaban essay berdasarkan persentase kemunculan kata kunci yang telah ditentukan oleh pemberi soal. → **Bukan KBS**
- Focused crawler yang menentukan apakah suatu halaman web relevan untuk suatu domain tertentu dengan melihat pola kemunculan kata yang ada pada halaman web tersebut. → **Klasifikasi; Kelas: relevan, tidak relevan**