

# Sistem Berbasis Pengetahuan

2024



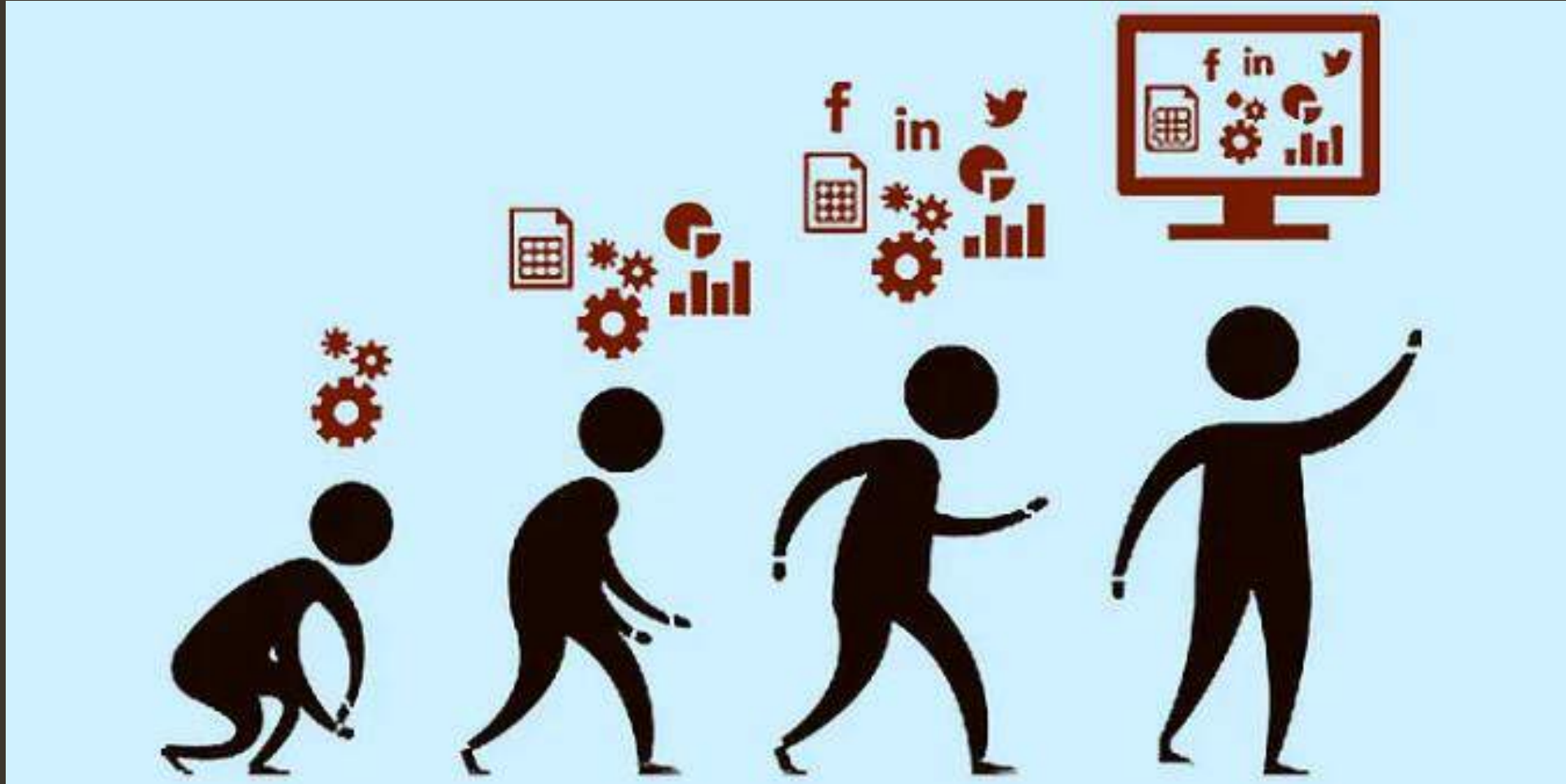
Prodi D-IV Sistem Informasi Bisnis  
Jurusan Teknologi Informasi



**Renungkan....**

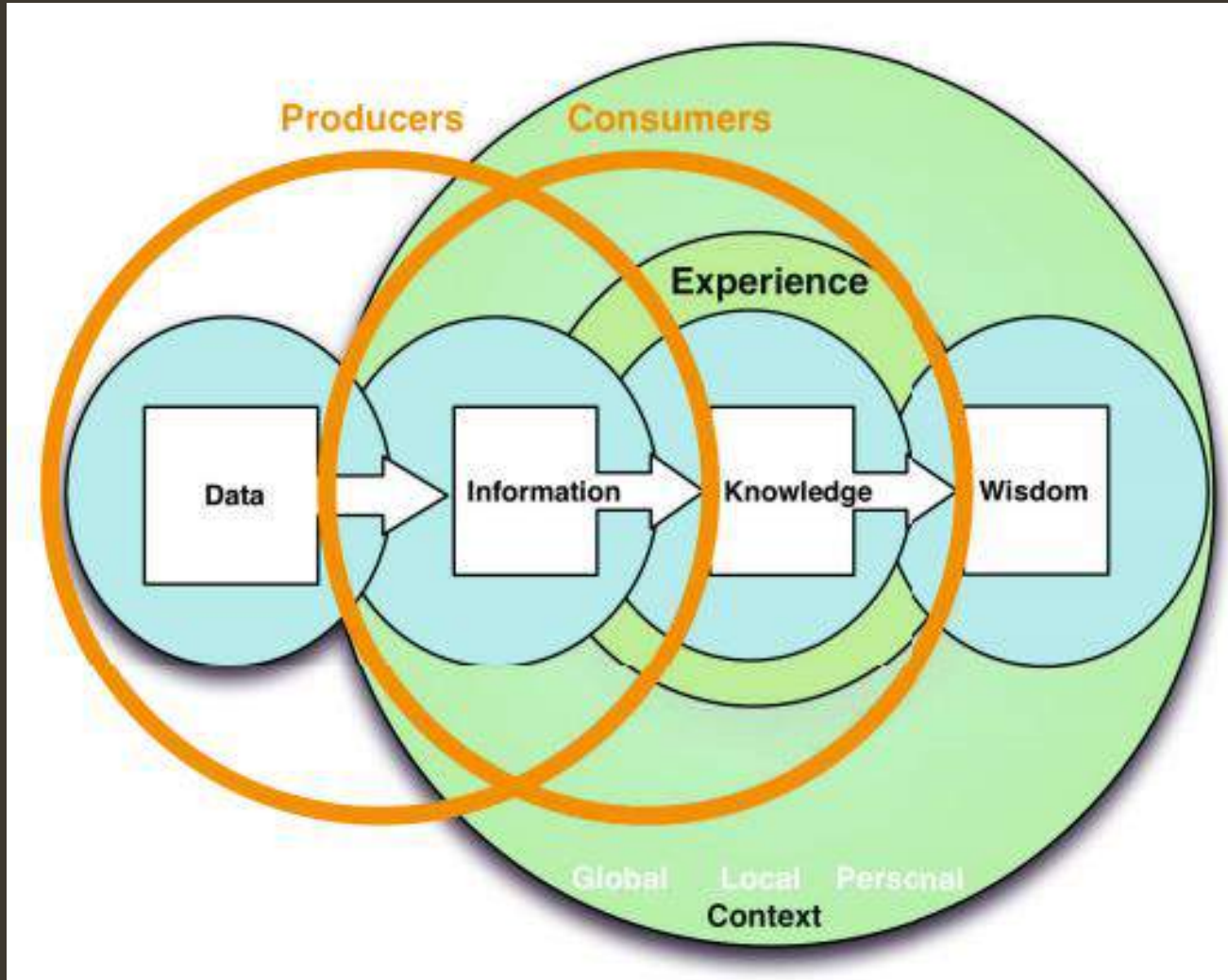
<https://www.youtube.com/watch?v=TnssuFikoiw>

# Pendahuluan



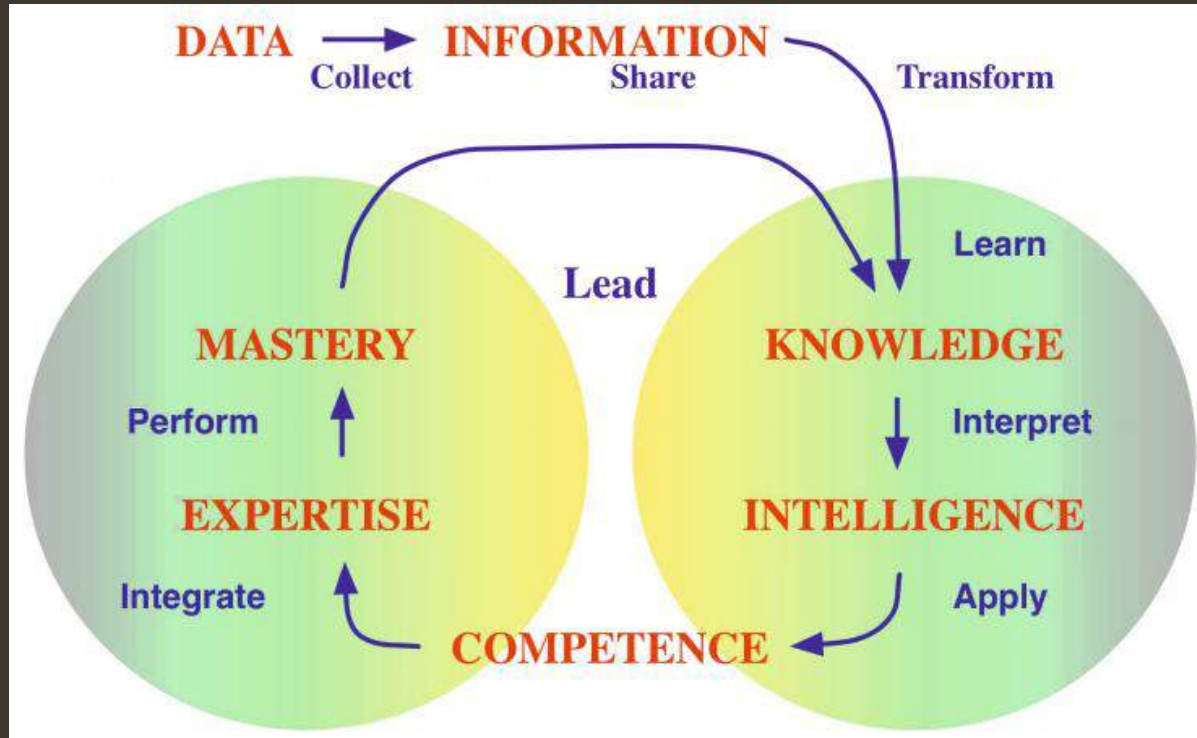
Everything should be made as simple as possible , but not simpler -- Albert Einstein

# Pendahuluan



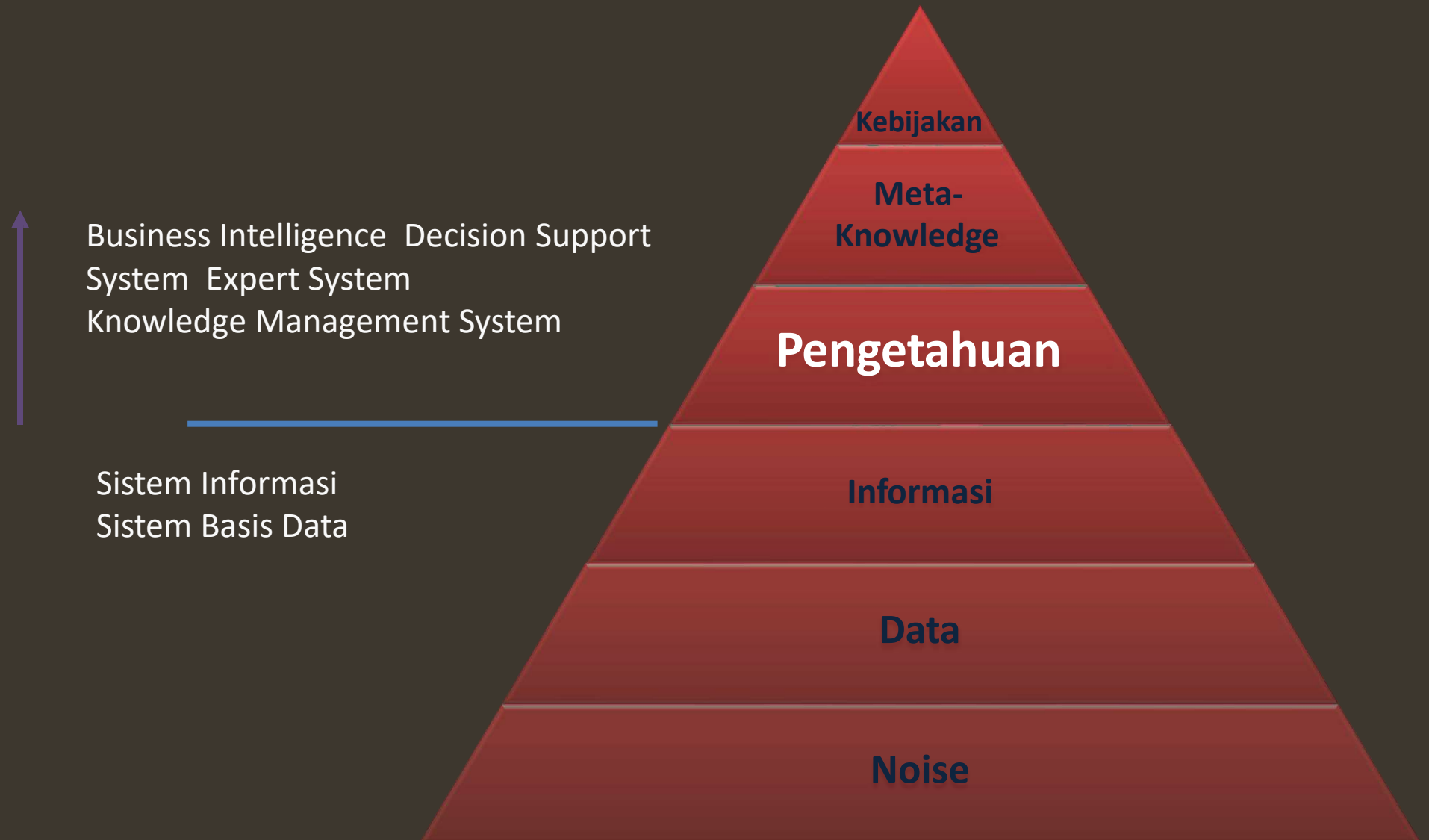
- Data adalah fakta mentah yang belum diolah. Dalam sistem berbasis pengetahuan, data menjadi input awal untuk pengolahan informasi.
- Informasi adalah data yang diolah dan diberi makna. Dalam sistem berbasis pengetahuan, informasi sering kali dihasilkan melalui pemrosesan data oleh algoritma.
- Pengetahuan muncul dari pemahaman mendalam terhadap informasi. Dalam sistem berbasis pengetahuan, pengetahuan direpresentasikan dalam bentuk aturan atau model.
- Kebijakan adalah penerapan pengetahuan dengan bijak untuk mengambil keputusan yang tepat. Dalam sistem berbasis pengetahuan, kebijakan mengandalkan penggunaan pengetahuan yang relevan.

# Pendahuluan

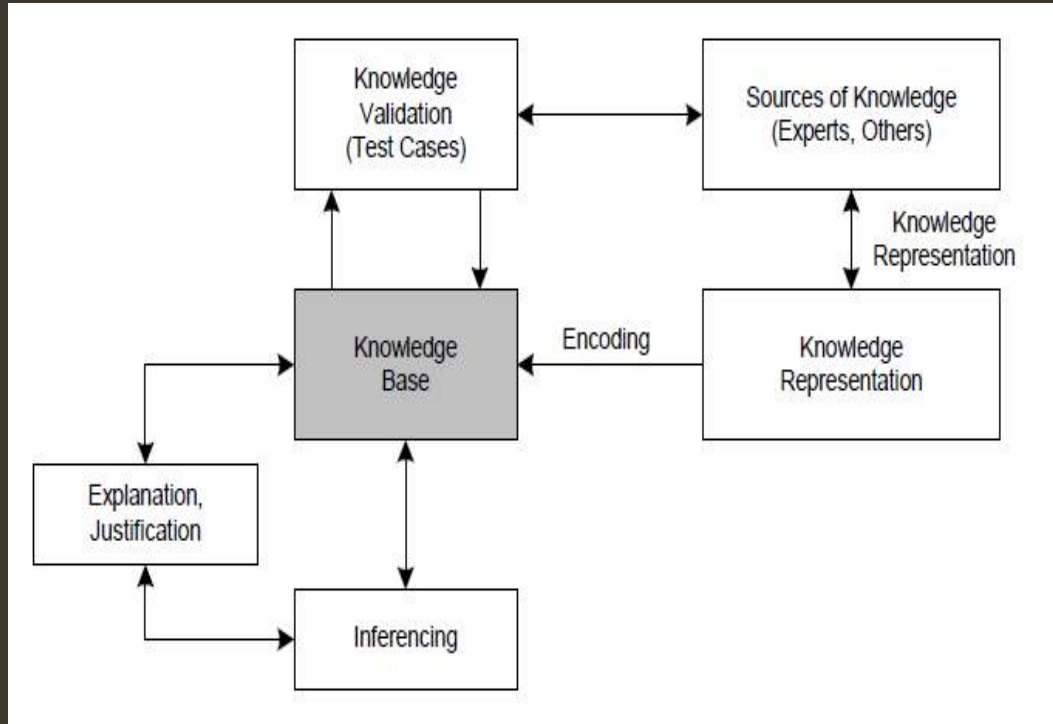


Tujuan SBP adalah membuat komputer mampu meniru kemampuan manusia dalam memecahkan masalah yang kompleks dengan memanfaatkan pengetahuan yang tersimpan. Manajemen pengetahuan mengandalkan pemrosesan data oleh mesin. Sistem informasi dimanfaatkan untuk mengatur informasi, sistem informasi, dan aplikasi manajemen pengetahuan

# Hierearki Pengetahuan



# Rekayasa Pengetahuan



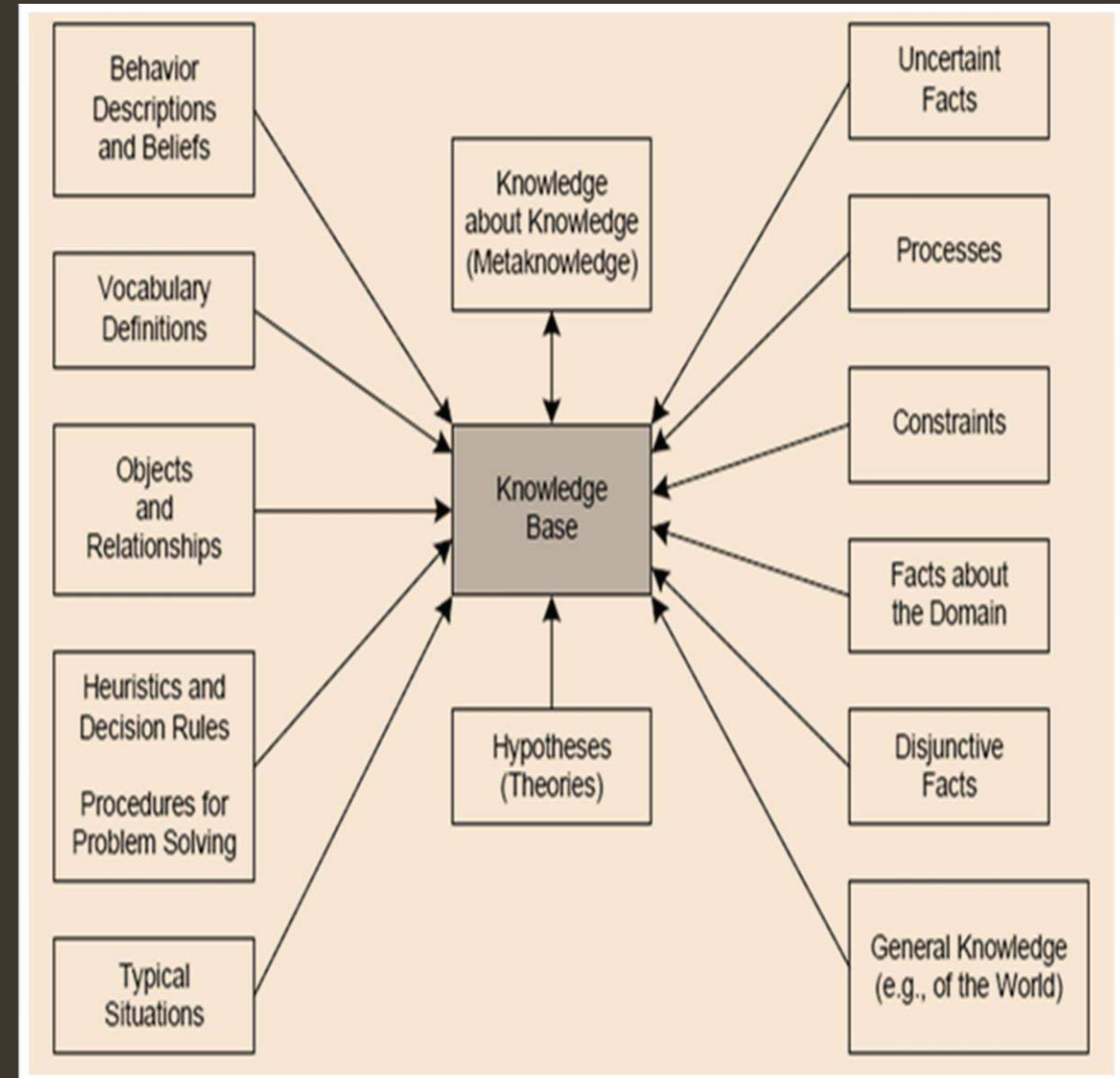
- **Validasi Pengetahuan:** Memastikan keakuratan, konsistensi, dan relevansi pengetahuan sistem
- **Sumber Pengetahuan:** Asal-usul pengetahuan dalam sistem, dari ahli, publikasi ilmiah, data empiris, atau pengalaman praktis
- **Representasi Pengetahuan:** Cara pengetahuan direpresentasikan agar dimengerti dan digunakan oleh mesin. Ini bisa berupa aturan logika, basis data, jaringan semantik, atau model lain yang memungkinkan pemahaman dan pemanfaatan yang efektif.
- **Inferensi:** Proses membuat keputusan atau penarikan kesimpulan berdasarkan pengetahuan sistem. Dengan aturan logika atau metode lainnya, sistem bisa mengekstrak informasi baru atau memecahkan masalah kompleks.
- **Penjelasan dan Justifikasi:** Kemampuan sistem menjelaskan alasan di balik keputusan atau kesimpulan yang dibuat. Ini meningkatkan transparansi dan kepercayaan pengguna terhadap sistem. Dengan penjelasan yang tepat, pengguna dapat lebih memahami proses inferensi dan menerima hasilnya.



# Sumber Pengetahuan

Terdapat dua sumber Pengetahuan :

1. **Terdokumentasi**; contoh : database, buku, jurnal, sumber online, dsb
2. **Tidak Terdokumentasi**; Sumber tidak terdokumentasi sering kali dijumpai akan tetapi sulit didokumentasikan karena berada dalam pikiran manusia; contoh : pengalaman, kesaksian lisan, intuisi, insight, dsb.





# Klasifikasi Pengetahuan

Berdasarkan Teknik pemecahan masalah

## *Compiled Knowledge*

Pengalaman pakar terhadap suatu domain masalah yang biasanya diperoleh dari buku, catatan, dan standar prosedur

## *Quantitative Knowledge*

Pengetahuan yang berasal dari teknik dalam teori matematika atau metode numerik

## *Qualitative Knowledge*

Pengetahuan yang diperoleh dari aturan praktis, teori perkiraan, dan model sebab akibat. Pengetahuan kualitatif ada 2 jenis:

1. Pengetahuan Prosedural : bagaimana cara menyelesaikan sesuatu, Contoh: cara membuat kue
2. Pengetahuan Deklaratif: berhubungan dengan atribut dari suatu benda atau domain masalah, Contoh: buah jeruk berbentuk bulat dan warna orange

# Kategori Pengetahuan

## *Procedural vs Declarative Knowledge*

Declarative Knowledge – pengetahuan dari fakta

Procedural Knowledge – pengetahuan tentang bagaimana terjadinya “sesuatu”

## *Tacit vs Explicit Knowledge*

Tacit – tidak mudah diartikulasikan

Explicit – mudah diartikulasikan

## *General vs Specific Knowledge*

# Tingkatan Pengetahuan

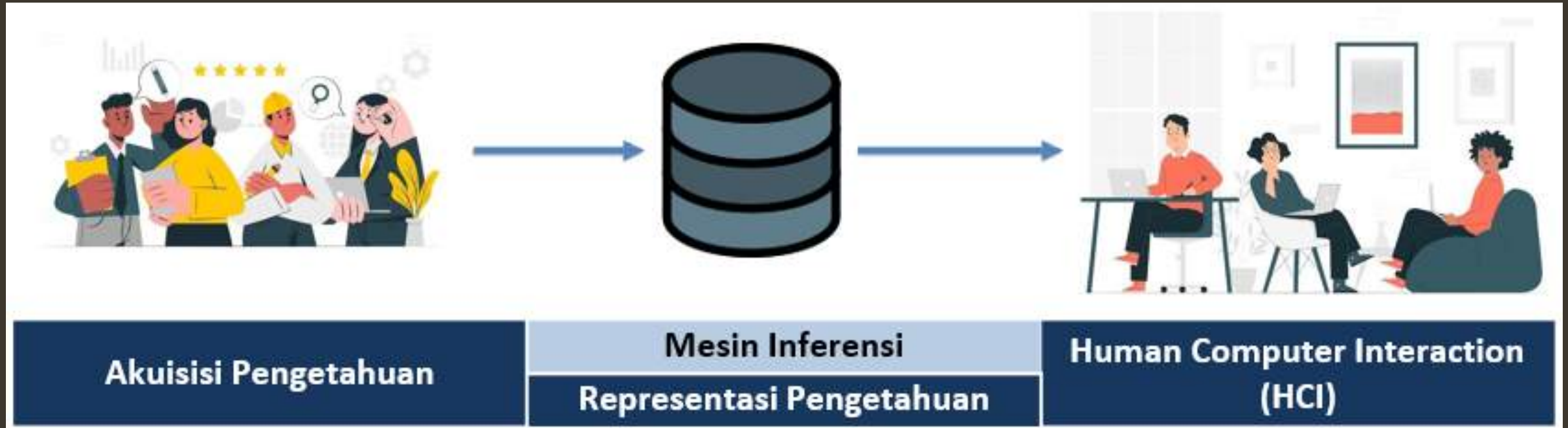
## *SHALLOW KNOWLEDGE (PENGETAHUAN DANGKAL)*

Mengacu pada representasi informasi level permukaan yang berhubungan dengan situasi yang sangat spesifik. Salah satu contohnya aturan IF-THEN

## *DEEP KNOWLEDGE (PENGETAHUAN DALAM)*

Mengacu pada struktur dalam dan penyebab dari suatu sistem dan mempertimbangkan interaksi diantara komponen-komponen sistem. Berbasis secara penuh pada integrasi, hubungan erat kesadaran manusia yang melibatkan emosi, akal sehat, intuisi dan lain-lain. Jenis pengetahuan ini sulit untuk dikomputerisasikan

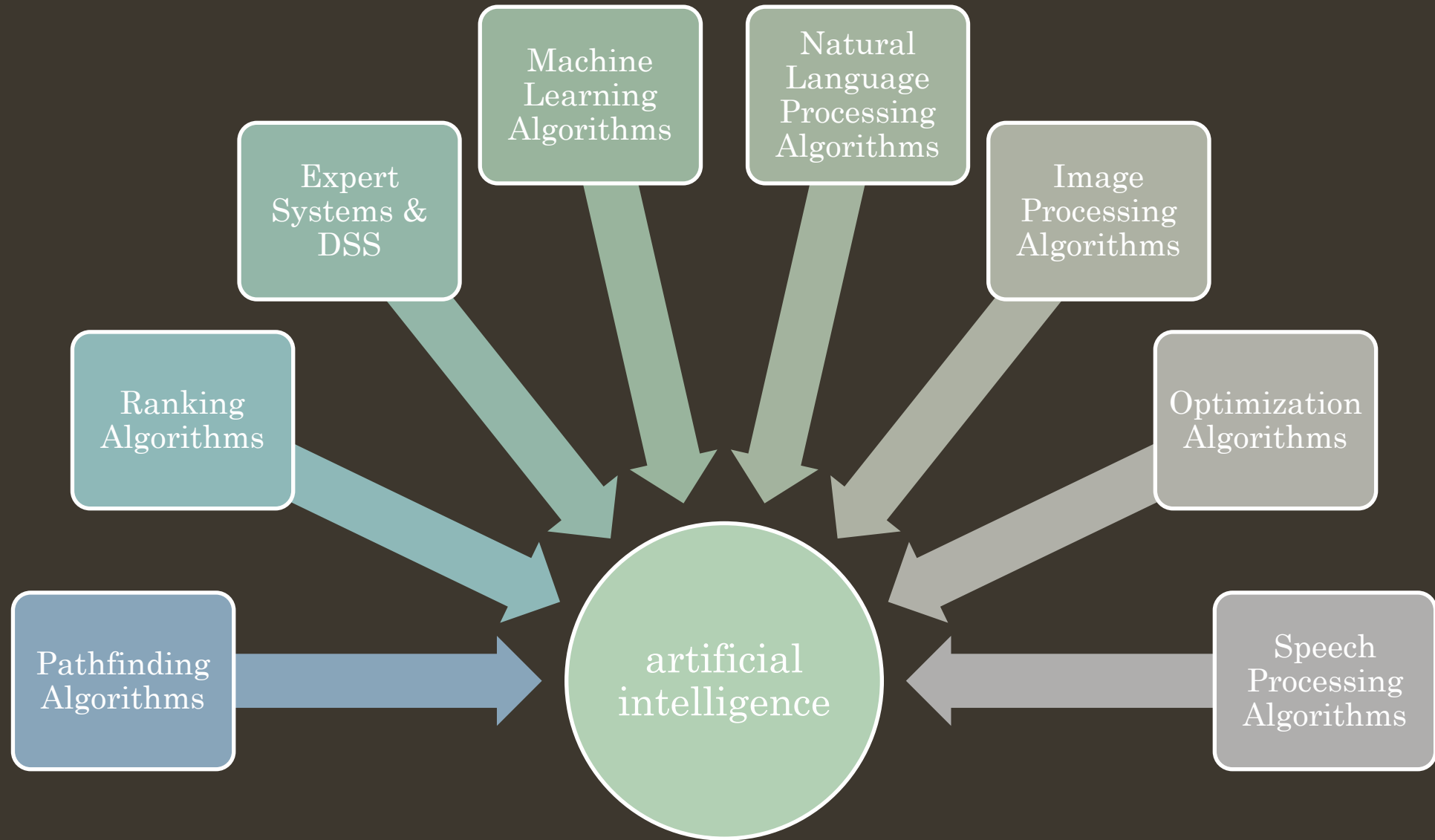
# Arsitektur SBP



# Kecerdasan Buatan

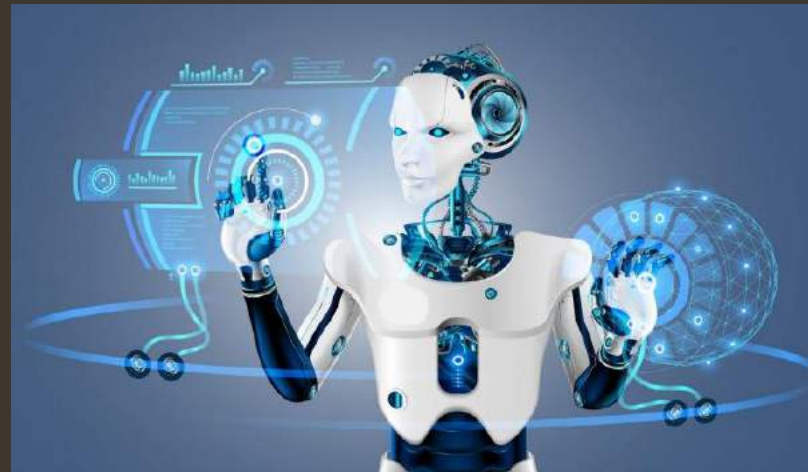
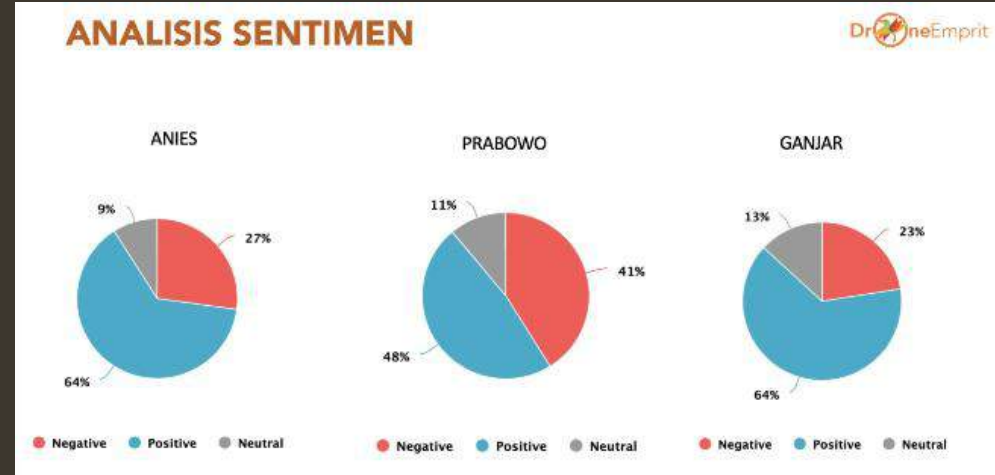
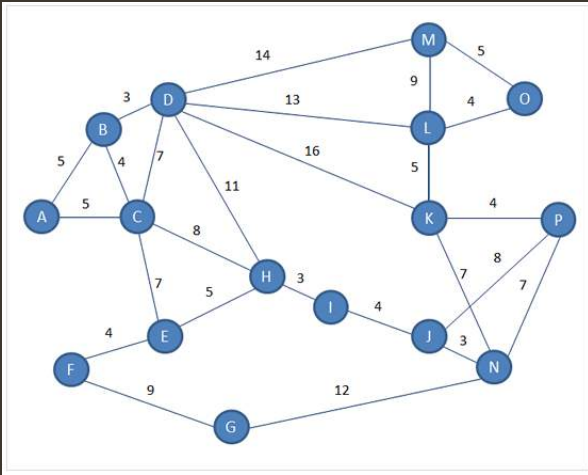
Kecerdasan buatan (AI) dan Sistem Berbasis Pengetahuan (SBP) memiliki hubungan erat dalam konteks pengembangan sistem yang mampu melakukan pemrosesan dan pengambilan keputusan secara cerdas. Kecerdasan buatan adalah bidang ilmu komputer yang bertujuan untuk menciptakan mesin atau program komputer yang dapat melakukan tugas yang membutuhkan kecerdasan manusia. Salah satu cabang utama dari kecerdasan buatan adalah sistem berbasis pengetahuan (SBP), yang fokus pada penerapan pengetahuan manusia dalam bentuk aturan, fakta, dan strategi untuk memungkinkan mesin melakukan penalaran, pengambilan keputusan, dan pemecahan masalah yang kompleks.

# Kecerdasan Buatan





# Penerapan Kecerdasan Buatan





# **Akuisisi dan Validasi Knowledge, Induksi Rule, Case Base Reasoning, dan Komputasi Syaraf**

Tim Teaching Sistem Berbasis Pengetahuan  
D-IV Sistem Informasi Bisnis  
Genap 2024



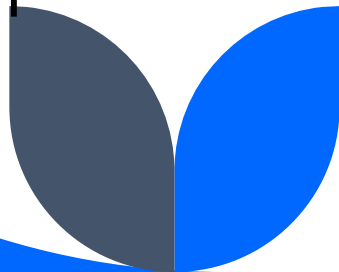
# Tujuan Pembelajaran

- Mahasiswa memahami Akuisisi dan Validasi Knowledge
- Mahasiswa memahami Induksi Rule, Case Based Reasoning, dan Komputasi Syaraf

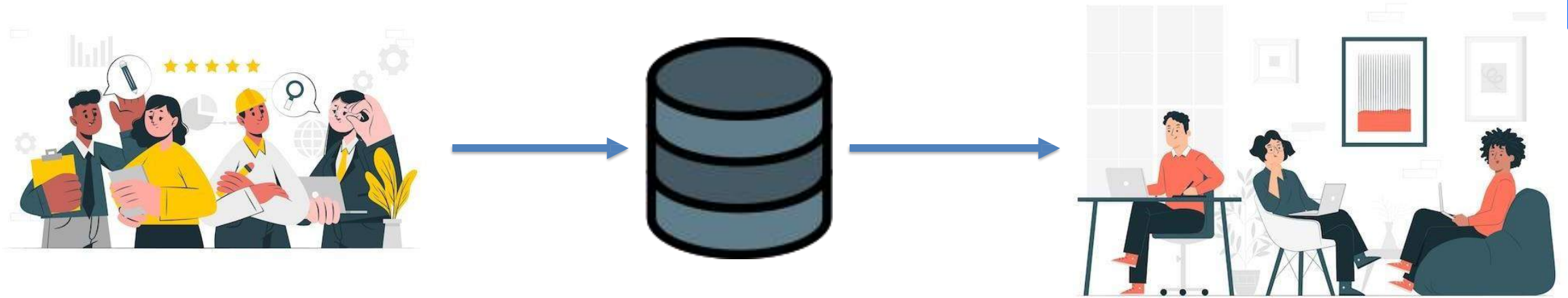
# Apa itu Pengetahuan (Knowledge)?



- **Informasi** yang diperoleh dari **kumpulan data**
- **Kumpulan data** dapat diambil polanya
- **Pola dari kumpulan data** adalah salah satu **bentuk pengetahuan**
- Pengetahuan yang diperoleh akan **merepresentasikan** keadaan yang telah terjadi atau akan terjadi di masa depan



# INTI SISTEM PAKAR



**Akuisisi Pengetahuan**

**Mesin Inferensi**  
**Representasi Pengetahuan**

**Human Computer  
Interaction (HCI)**

# Akuisisi dan Validasi Pengetahuan





# APA ITU AKUISISI PENGETAHUAN



- Mengumpulkan informasi dari berbagai sumber/ pakar untuk kemudian disimpan dalam sistem komputer
- Sumber pengetahuan: pakar/ experts, buku, laporan teknikal, database, dll
- Siapa yang melakukannya?



*Knowledge Engineer*



Pakar

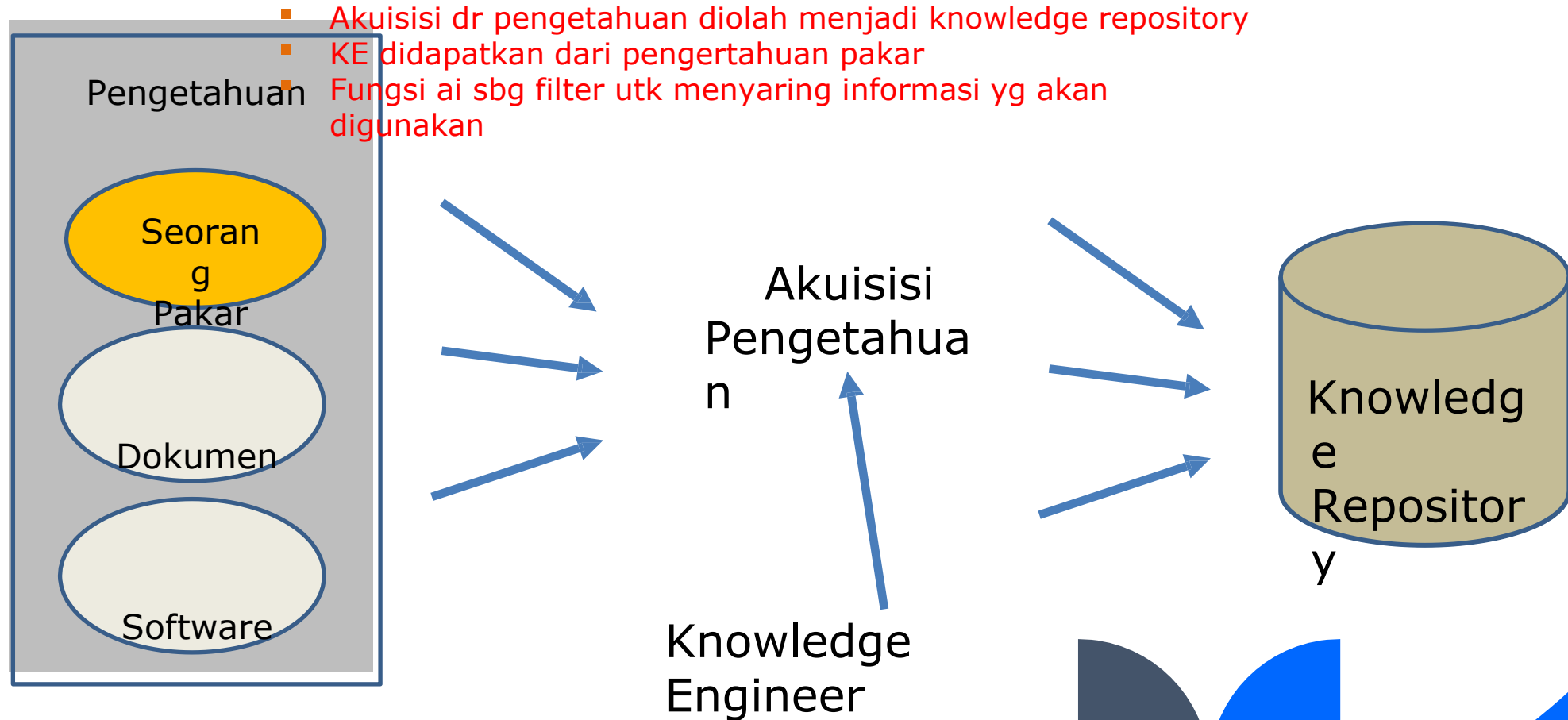
# Akuisisi Pengetahuan



- Mengambil informasi sesuai kebutuhan yang sudah disaring sebelumnya
- bottleneck
- Akuisisi Pengetahuan merupakan proses ekstraksi, strukturisasi, dan mengorganisasikan pengetahuan dari satu sumber atau lebih.
- Proses ini merupakan suatu proses yang penting, namun seringkali menjadi “bottleneck” yang membatasi pengembangan sistem pakar dan sistem AI yang lain.



# AKUISISI PENGETAHUAN – What is it?



# (Perekayasaan Pengetahuan)



- Knowledge Engineering (KE) adalah seni dalam membawa prinsip-prinsip dan perkakas (tools) dari penelitian AI yang berhubungan dengan permasalahan aplikasi yang sulit yang dibutuhkan pakar pengetahuan guna dicari solusinya.
- KE dibedakan dalam 2 perspektif :
  - Dalam arti Sempit : akuisisi, representasi, validasi pengetahuan, explanasi dan maintenance
  - Dalam arti luas : proses penyusunan dan penanganan sistem AI
- KE istilahnya sama dengan akuisisi



# Tujuan KE

- Manuyusun program yang memiliki modul- modul alami, sehingga penambahan dan perubahan dapat dibuat dalam satu modul tanpa mempengaruhi kerja modul yang lain.



# Proses-Proses KE

- Knowledge Acquisition (Akuisisi Pengetahuan)
- Knowledge Validation (Validasi Pengetahuan)
- Knowledge Representation (Representasi/ penyajian Pengetahuan)
- Inferensi (Penarikan kesimpulan)
- Explanation and Justification (Kemampuan menerangkan dan memberi pertimbangan)





# Akuisisi Pengetahuan

- Mengakuisisi pengetahuan dari pakar, buku-buku, dokumen-dokumen, sensor, file-file komputer.
- Pengetahuan dapat berupa :
  - Permasalahan khusus
  - Prosedur penyelesaian masalah
  - Pengetahuan umum (misal, pengetahuan bisnis)
  - Meta pengetahuan (pengetahuan tentang pengetahuan)

# APA SAJA YANG DI AKUISISI

Akuisisi  
Pengetahuan

- What is it?
- What are the issues?
- What are the solutions?
- What techniques are use?

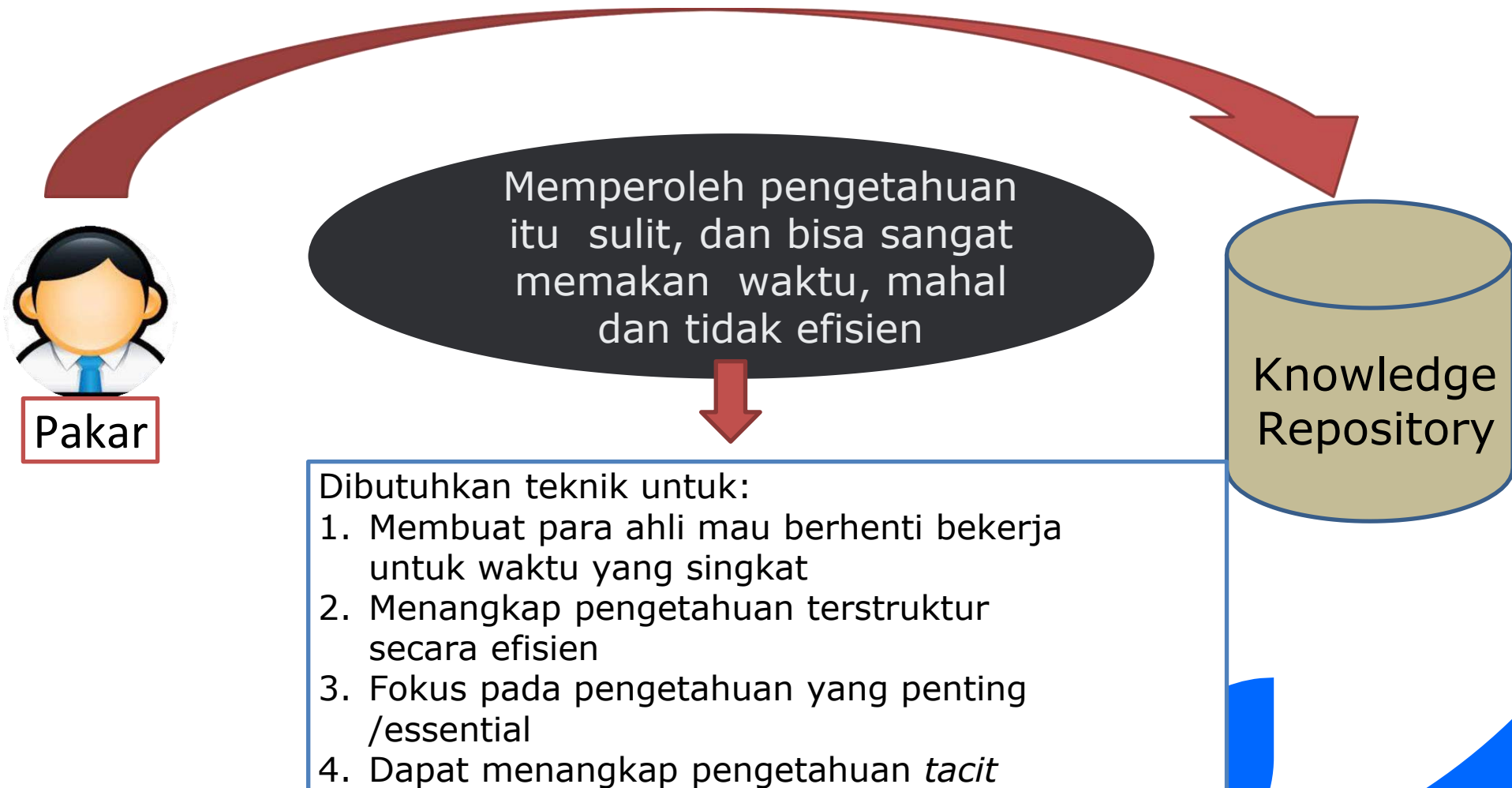
# TUJUAN AKUISISI PENGETAHUAN



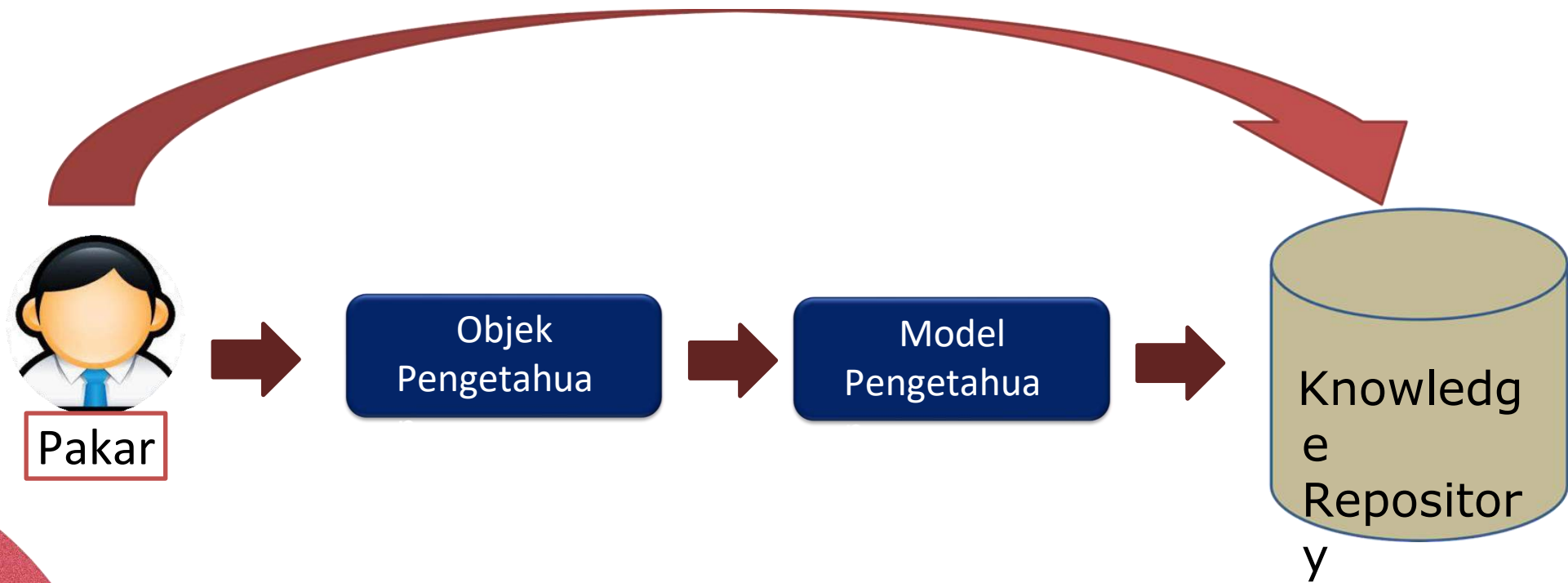
- Supaya bisa diimplementasikan ke komputer
- 1. Merumuskan pengetahuan (*Knowledge Base*) sehingga dapat diorganisasikan kedalam komputer
- 2. Mendapatkan pengetahuan, fakta – aturan, model dan cara pemecahan masalah



# AKUISISI PENGETAHUAN - *Isu*



# DARI PAKAR KE REPOSITORI



# Validasi Pengetahuan

- Pengetahuan harus divalidasi dan diuji sehingga kualitasnya dapat diterima

# Representasi Pengetahuan

- Pengetahuan yang diperoleh diorganisasikan dalam suatu aktifitas yang meliputi menyiapkan sebuah “peta pengetahuan” dan menjadikan pengetahuan dalam basis pengetahuan





# Kesulitan dalam Akuisisi pengetahuan

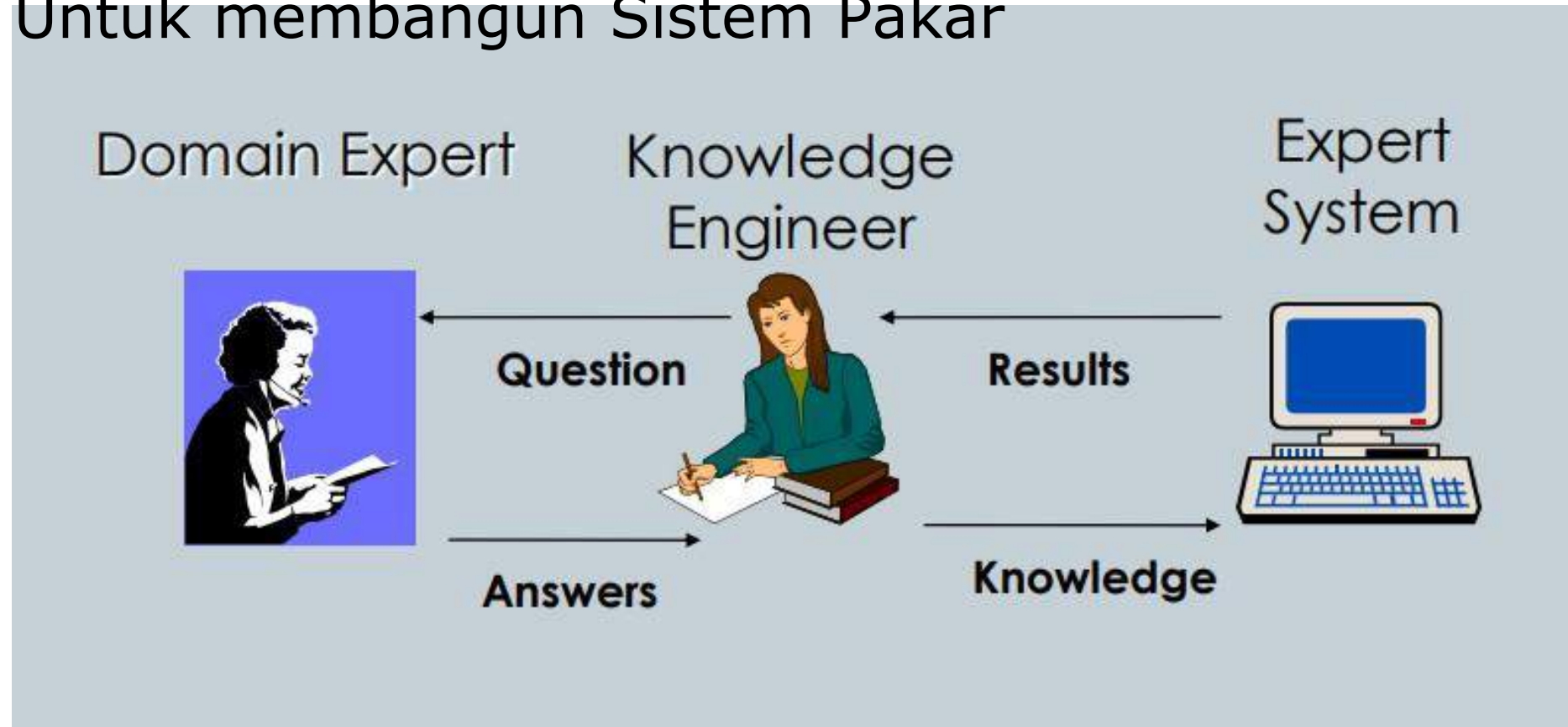


- Masalah dalam mentransfer pengetahuan
  - Mengekspresikan pengetahuan
  - Transfer ke mesin
  - Jumlah partisipan
  - Strukturisasi pengetahuan
  - Alasan lain
- Ketrampilan perekayasa pengetahuan
  - Komputer skills
  - Kemampuan belajar dengan cepat
  - Berpikir logis, dll



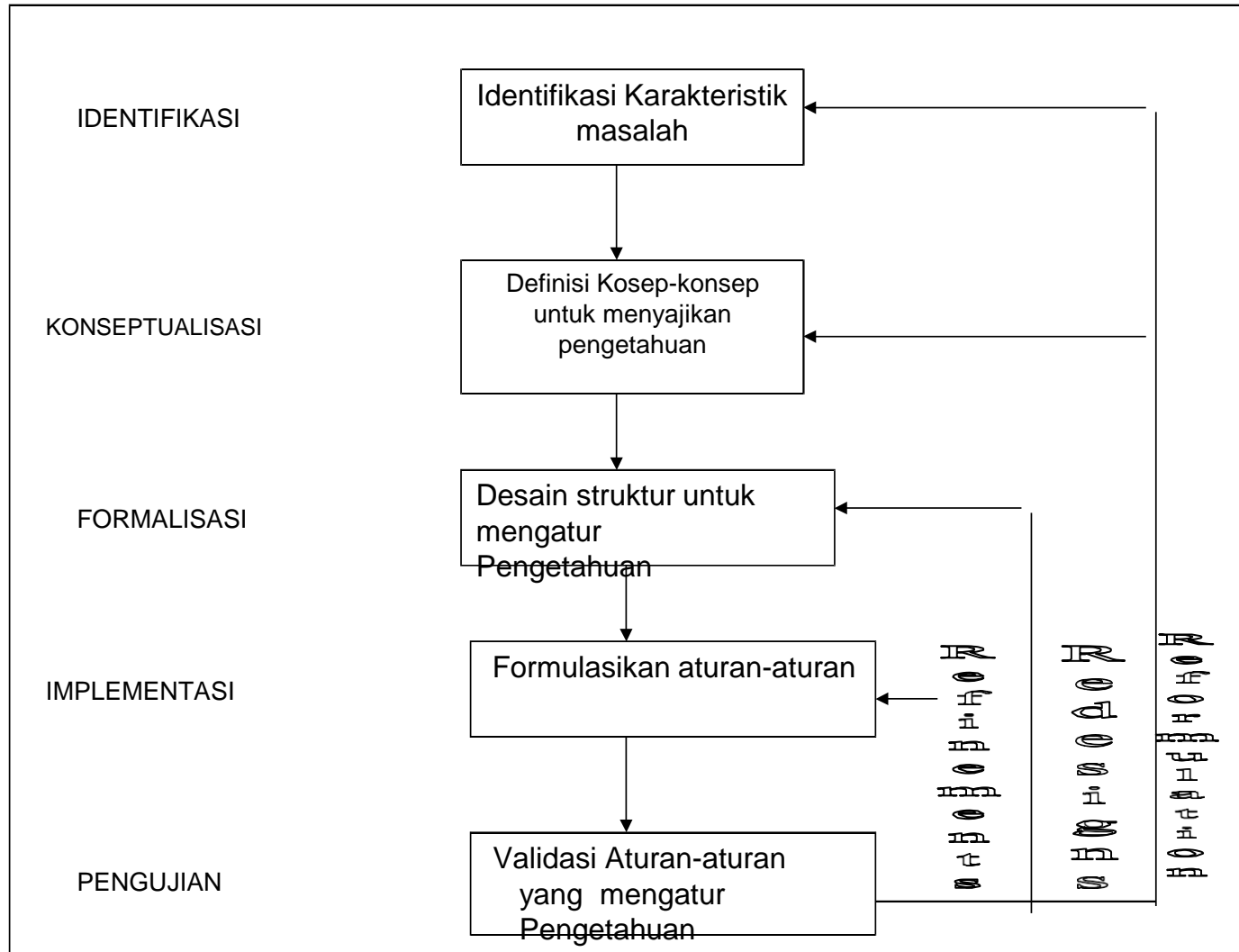
# PROSES AKUISISI PENGETAHUAN

Untuk membangun Sistem Pakar



# Langkah-Langkah Proses Akuisisi Pengetahuan

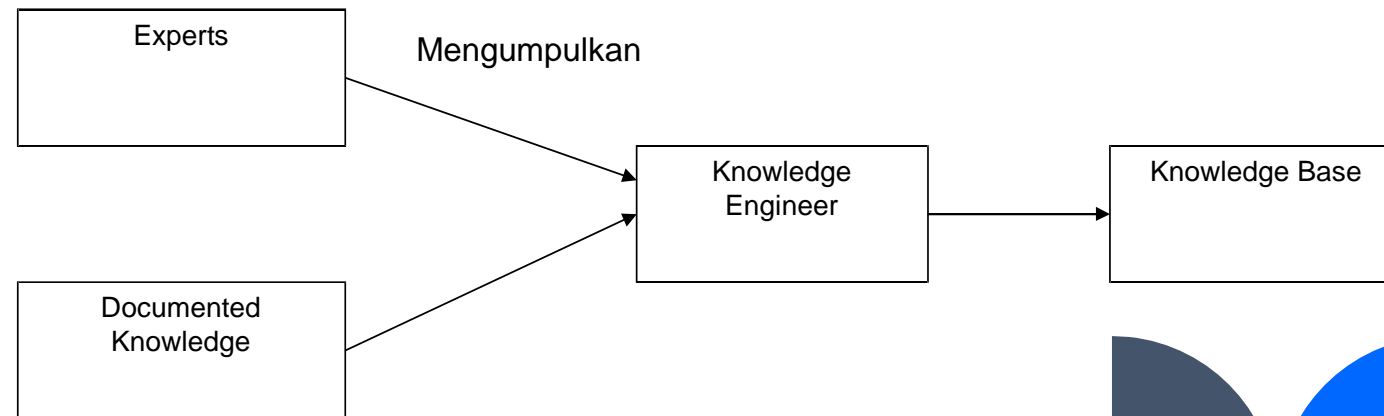
▪ Ada 5 tahapan



# Metode-Metode Akuisisi Pengetahuan



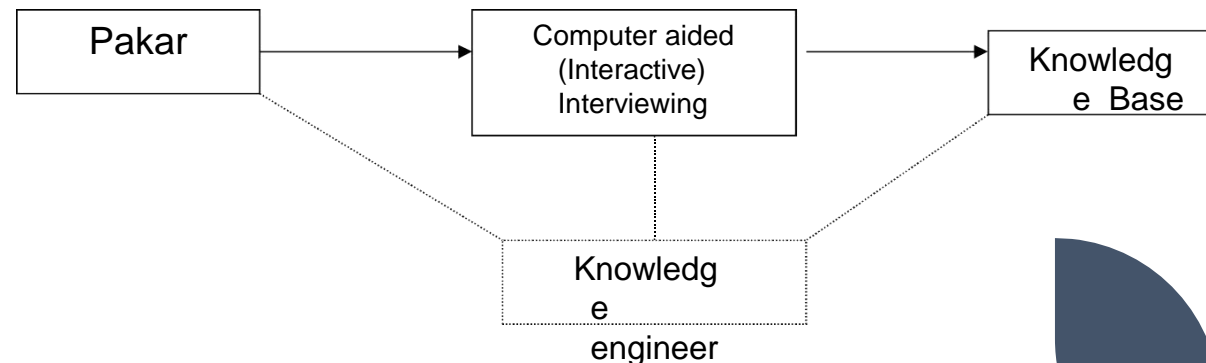
- **Metode Manual**
  - Meliputi beberapa macam wawancara, observasi, dll
  - Perekayasa pengetahuan mengumpulkan pengetahuan dari pakar dan sumber lain



# Metode-Metode Akuisisi Pengetahuan



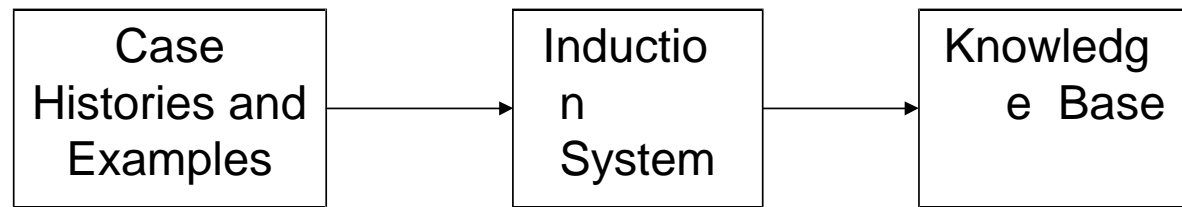
- **Metode Semi otomatis**
  1. Mengharapkan bantuan pakar dengan membiarkan mereka membangun basis pengetahuan dengan sedikit atau tanpa bantuan perekayasa pengetahuan
  2. Mengharapkan bantuan perekayasa pengetahuan dengan membiarkan mereka mengeksekusi hal-hal tertentu sehingga lebih efisien/ efektif



# Metode-Metode Akuisisi Pengetahuan



- Metode Otomatis
  - Peran pakar dan/atau perekayasa pengetahuan diminimalkan atau dikurangi
  - Contoh : Sistem induksi



# Validasi dan Verifikasi Basis Pengetahuan

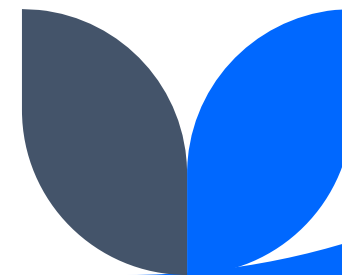
- Kalo validasi hasil dari yg ada di sistem apakah sesuai dg requirement ato ga
- Klo verifikasi adalah org yg akan menguji sistem tersebut (yg bagian testing), yang nguji knowledge endngineernya



- Untuk **mengontrol akuisisi pengetahuan** diperlukan **evaluasi, validasi, dan verifikasi**.
- **Evaluasi** dimaksudkan untuk melihat apakah sistem dapat digunakan, efisien, dengan biaya yang efektif.
- **Validasi** adalah bagian dari evaluasi yang menilai performa sistem (misal; membandingkan dengan kepakaran). Validasi mengacu pada pembuatan sistem yang “benar” (*building the right system*), yaitu performa sistem memiliki akurasi yang dapat diterima.
- **Verifikasi** mengacu pada pembuatan kebenaran sistem (*building the system rights*), yaitu substansi sistem dapat diimplementasikan dengan benar pada spesifikasinya.

# Ukuran-Ukuran Validitas

- **Accuracy** : sebaik apa sistem merefleksikan kenyataan, kebenaran pengetahuan dalam knowledge base
- **Adaptability** : Kemungkinan untuk dikembangkan dimasa yang akan datang, perubahan
- **Adequacy** (or Completeness) : porsi dari pengetahuan yang penting dalam knowledge base
- **Appeal** : Sebaik apa kesesuaian basis pengetahuan dengan intuisi dan simulasinya secara praktis.





# Ukuran-Ukuran Validitas

- **Breadth** : Seberapa baik, ruang lingkup pengetahuan dicover (dijangkau).
- **Depth** : Derajat kelengkapan (detil) pengetahuan.
- **Face validity** : kredibilitas pengetahuan
- **Generality** : Kapabilitas sistem untuk digunakan dengan permasalahan lain yang memiliki kemiripan.
- **Precision** : Kapabilitas sistem untuk mereplkasi parameter sistem tertentu; konsistensi nasihat, jangkauan dari variabel-variabel dalam basis pengetahuan.



# Ukuran-Ukuran Validitas



- **Realism** : Akuntabilitas untuk relasi dan variabel- variabel yang relevan; realitas yang mirip
- **Robustness** : Sensitivitas dari konklusi untuk struktur model
- **Sensitivity** : Tekanan terhadap perubahan- perubahan dalam basis pengetahuan yang didasarkan pada kualitas output
- **Technical and operational validity** : validitas dari asumsi- asumsi, konteks, batasan, dan kondisi serta ketepatan ukuran.



# PENGETAHUAN MANUAL DAN SEMI OTOMATIS



- ✓ Lamban dan Mahal
- ✓ Sulit untuk validasi pengetahuan yang diperoleh
- ✓ Seringkali terdapat korelasi lemah diantara laporan verbal dan perilaku mental
- ✓ Dalam situasi tertentu pakar tak mampu menyediakan keseluruhan hal tentang bagaimana suatu keputusan tersebut dibuat.
- ✓ Kualitas suatu sistem banyak tergantung pada kualitas pakar dan Knowledge Engineer
- ✓ Pakar tidak memahami teknologi Sistem pakar
- ✓ Dalam banyak kasus, Knowledge Engineer tidak memahami sifat dasar bisnis

# SOLUSI



- ❖ Diperlukan pengembangan metode akuisisi pengetahuan yang dapat mengurangi atau menghilangkan kebutuhan dari 2 partisipan.
- ❖ Metode ini disebut Akuisisi Pengetahuan yang dibantu oleh komputer atau Akuisisi Pengetahuan **OTOMATIS**

## TUJUAN:

1. Meningkatkan produktivitas pekerjaan Knowledge Engineer (mengurangi biaya) Mengurangi level keterampilan yang diperlukan oleh knowledge Engineer
2. Menghilangkan atau mengurangi kebutuhan akan Pakar  
Menghilangkan atau mengurangi kebutuhan akan Knowledge Engineer Meningkatkan Kualitas pengetahuan yang didapatkan.



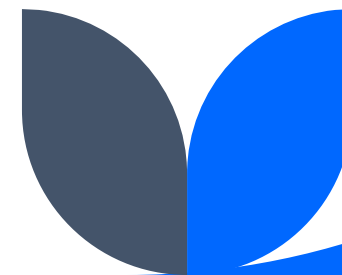
# INDUKSI RULE

# INDUKSI RULE OTOMATIS



- Induksi rule : ad ri bhs manusia itu maka kita merubah bhs manusia tsb ke bhs mesin agar dapat dipahami

- ❖ Induksi -> proses reasoning dari hal khusus ke hal umum
- ❖ Rule dihasilkan oleh program komputer dari contoh kasus
- ❖ Menyediakan contoh-contoh masalah (*training set/* kumpulan pelatihan)
- ❖ Contoh Algoritma Induksi Rule Otomatis: ID3



# APAKAH ITU ID3?



- ▶ Pertama kali ID3 mengkonversi matriks pengetahuan ke dalam pohon keputusan.
- ▶ Atribut yang tidak relevan dihilangkan dan atribut yang relevan diorganisasi dalam cara yang efisien.



# Contoh Kasus

- Bmkg yg menentukan siklus cuaca dalam beberapa hari berikutnya menggunakan data terkait

No.	Kelembaban (%)	Suhu (°C)	Keadaan Langit	Hujan
1	85	25	Cerah	Ya
2	90	24	Berawan	Ya
3	78	26	Hujan Ringan	Ya
4	88	22	Cerah	Ya
5	80	27	Hujan Lebat	Tidak
6	70	30	Cerah	Tidak
7	75	29	Berawan	Tidak
8	82	28 ↓	Cerah	Tidak



# Keterangan-Contoh Kasus



1. Memilih Atribut Pemisah: ID3 memilih atribut yang paling informatif untuk memisahkan data. Misalkan kita mulai dengan atribut kelembaban.
2. Membagi Data Berdasarkan Atribut Pemisah: Kita membagi dataset menjadi subset berdasarkan nilai-nilai atribut kelembaban. Misalnya, kita dapat membagi data menjadi dua subset: (1) kelembaban  $\leq 80\%$  dan (2) kelembaban  $> 80\%$ .
3. Rekursif Memilih Atribut Lain: Proses ini diulangi untuk setiap subset yang dihasilkan. Kita bisa memilih atribut suhu untuk subset pertama (kelembaban  $\leq 80\%$ ) dan atribut keadaan langit untuk subset kedua (kelembaban  $> 80\%$ ).
4. Membuat Pohon Keputusan: Proses ini berlanjut secara rekursif sampai kita memiliki subset homogen atau tidak ada atribut lagi yang tersisa untuk dipilih. Pada titik ini, kita menambahkan simpul terminal ke pohon keputusan dengan label prediksi (hujan atau tidak hujan).

# KEUNTUNGAN DARI INDUKSI RULE (1)



- ❖ Dapat digunakan untuk hal-hal yang lebih rumit dan lebih menguntungkan secara komersial
- ❖ Pembangun (builder) tidak harus menjadi Knowledge Engineer.
- ❖ Builder bisa berasal dari seorang pakar atau analis sistem
- ❖ Sehingga menghemat waktu dan biaya, serta menyelesaikan kesulitan yang berhubungan dengan Knowledge Engineer yang merupakan orang luar yang tidak Familiar dengan "bisnis"



# KEUNTUNGAN DARI INDUKSI RULE (2)



- ❖ Mesin induksi juga menawarkan kemungkinan mendeduksi pengetahuan baru.
- ❖ Dimungkinkan mendaftar semua faktor yang mempengaruhi keputusan, tanpa pemahaman akan dampaknya
- ❖ Rule yang dihasilkan dapat dikaji ulang oleh pakar dan bila diperlukan bisa dimodifikasi
- ❖ Keuntungan besar dari induksi rule adalah ia meningkatkan proses pemikiran dari sang pakar



# KESULITAN DALAM IMPLEMENTASI (1)



- ❖ Beberapa program induksi yang dapat menghasilkan rule, mungkin tidak mudah bagi orang untuk memahaminya, disebabkan cara program mengklasifikasikan atribut masalah dan propertinya mungkin tidak sesuai dengan cara manusia melakukan hal itu.
- ❖ Program Induksi rule tidak memilih atribut. Sang pakar tetap harus memilih atribut mana yang signifikan; misal, faktor penting untuk menyetujui pinjaman.



# RESULTAN DALAM IMPLEMENTASI (2)



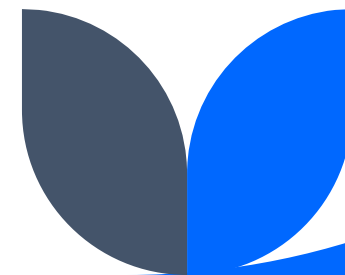
- ❖ Proses pencarian dalam induksi rule berdasarkan algoritma khusus yang menghasilkan pohon keputusan yang efisien, yang mengurangi jumlah pertanyaan yang harus ditayangkan sebelum suatu kesimpulan dapat dicapai.
- ❖ Metode ini baik hanya untuk rule-base (berbasis aturan), masalah berjenis klasifikasi, khususnya pada jenis ya atau tidak (namun demikian masalah ini masih dapat diurai atau dipilah sehingga menjadi kategori klasifikasi)
- ❖ Jumlah atribut harus cukup kecil, jika lebih dari 15 mungkin diperlukan mainframe. Saat ini perkembangan hardware dan software yang meningkat jauh dapat mengatasi permasalahan jumlah atribut



# KESEKUTUPAN DALAM IMPLEMENTASI (3)



- ▶ “Kecukupan” jumlah contoh-contoh yang diperlukan bisa sangat besar
- ▶ Kumpulan contoh harus di-“bersihkan” sebagai contoh kasus-kasus yang merupakan pengecualian dari rule harus dihilangkan dulu
- ▶ Metode initerbatas pada situasi tertentu (**deterministik**)
- ▶ Masalah utama dengan metode ini adalah pembangun (builder) tak tahu lebih jauh, apakah **jumlah contoh-contoh sudah mencukupi** dan apakah **algoritma sudah cukup baik**.





# Case Base Reasoning

# CASE BASE REASONING

- ▶ Pendekatan pembangunan Sistem Pakar dengan mengakses pengalaman-pengalaman penyelesaian masalah untuk memperkirakan solusi dalam penyelesaian masalah di masa depan.
  - Case base reasoning: membentuk database sesuai dg permasalahan yg ada tdi
- ▶ Koleksi dari kasus-kasus historis dan penyelesaiannya membentuk Knowledge Base.





# CASE BASE REASONING

- ▶ Pengambilan keputusan memanggil kembali kasus-kasus sebelumnya yang mungkin identik dengan kasus baru, walaupun dalam banyak hal tidak bisa identic.
- ▶ Kasus-kasus sebelumnya mungkin menunjukkan sedikit kemiripan dengan kasus baru, tetapi walaupun sedikit hal itu tetap saja berguna.
- ▶ Akuisisi Knowledge sangat mudah dilakukan, karena data historis berada dalam file dan hanya memerlukan sedikit verifikasi dari pakar.





# Komputasi Syaraf

# KOMPUTASI SYARAF



Data Historis digunakan untuk menurunkan solusi ke masalah yang baru

Bekerja dalam domain yang cukup sempit menggunakan pendekatan pengenalan pola

Diperlukan sejumlah besar kasus-kasus historis

Akuisisi Pengetahuan menjadi sederhana, karena kasus-kasus historis

dan penyelesaiannya biasanya tersedia dalam database perusahaan.

Pakar berperan hanya selama validasi dan verifikasi



# Bagaimana Memilih Metode Yang Sesuai? (1)



**Mencapai Tujuan dari Sistem Akuisisi Pengetahuan yang Ideal :**

- ➔ Mengarahkan interaksi dengan pakar tanpa intervensi Knowledge Engineer
- ➔ Dapat diaplikasikan untuk domain masalah yang tidak terbatas atau paling tidak, banyak kelas
- ➔ Kemampuan tutorial untuk menghasilkan kebutuhan akan pelatihan awal bagi sang pakar



# Bagaimana Memilih Metode Yang Sesuai? (2)



**Mencapai Tujuan dari Sistem Akuisisi Pengetahuan yang Ideal :**

- ➔ Kemampuan untuk menganalisis pekerjaan yang sedang berlangsung untuk mendeteksi ketidakkonsistenan dan kesenjangan didalam pengetahuan
- ➔ Kemampuan untuk menggabungkan bermacam-macam sumber pengetahuan
- ➔ Antarmuka manusia (Misal, percakapan biasa) yang membuat penggunaan sistem terasa nyaman dan menarik
- ➔ Kemampuan dalam antarmuka secara mudah dengan berbagai Tool Sistem Pakar yang berbeda dan sesuai dengan domain perusahaan



**TERIMA KASIH....**

# Sistem Berbasis Pengetahuan

2024

## Pengenalan Sistem Pakar Berbasis Pengetahuan dan Representasi Pengetahuan



Prodi D-IV Sistem Informasi Bisnis  
Jurusan Teknologi Informasi



# Pengetahuan

- Kunci utama dari sistem pakar
- Analogi
  - Algoritma + Struktur Data = Program
  - Pengetahuan + Inferensi = Sistem Pakar



# Representasi Pengetahuan

- Metode yang digunakan untuk mengodekan pengetahuan dalam sebuah sistem pakar
- Dimaksudkan untuk
  - Menangkap sifat-sifat penting problema
  - Membuat informasi itu dapat diakses oleh prosedur pemecahan problema

# Representasi Pengetahuan

❖ Dua bagian dasar sistem kecerdasan buatan (menurut Turban) :

## ❖ Basis pengetahuan

- ❖ fakta tentang objek-objek dalam domain yang dipilih
- ❖ hubungan di antara domain-domain tersebut

## ❖ Inference Engine

- ❖ Merupakan sekumpulan prosedur
- ❖ Menguji basis pengetahuan dalam menjawab suatu pertanyaan, menyelesaikan masalah, atau membuat keputusan

❖ Karakteristik representasi pengetahuan

- ❖ Dapat diprogram dengan bahasa komputer dan disimpan dalam memori
- ❖ Fakta dan pengetahuan lain yang terkandung di dalamnya dapat digunakan untuk melakukan penalaran

# Model Representasi Pengetahuan

- Pengetahuan dapat direpresentasikan dalam bentuk yang sederhana atau kompleks, tergantung dari masalahnya.
- Beberapa model representasi pengetahuan
  - Logika (*logic*)
  - List & Tree
  - Jaringan semantik (*semantic nets*)
  - Bingkai (*frame*)
  - Tabel Keputusan (*decision table*)
  - Pohon Keputusan (*decision tree*)
  - Naskah (*script*)
  - Kaidah/Sistem produksi (*production rule*)

# Logika

- Suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah, dan prosedur yang membantu penalaran.
- Komputer harus dapat menggunakan proses penalaran deduktif dan induktif ke dalam bentuk yang sesuai dengan manipulasi komputer, yaitu logika simbolik atau matematika
- Disebut Logika Komputasional
  - Logika Proporsional
  - Logika Predikat

# Penalaran Deduktif

- Bergerak dari penalaran umum menuju ke konklusi khusus
- Atau pernyataan premis dan inferensi
  - Premis Mayor
  - Premis Minor
  - Konklusi
- Contoh
  - **Premis mayor** : Jika hujan turun saya tidak akan kuliah
  - **Premis minor** : Pagi ini hujan turun
  - **Konklusi** : Oleh karena itu pagi ini saya tidak akan kuliah

# Penalaran Induktif

- Bergerak dari masalah khusus ke masalah umum
- Menggunakan sejumlah fakta atau premis yang mantap untuk menarik 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
- Konklusi tidak selalu mutlak, dapat berubah jika ditemukan fakta baru
- Contoh
  - **Premis 4** : AI adalah pelajaran yang sulit

# Logika Proporsional

- Proposisi merupakan suatu statemen atau pernyataan yang menyatakan benar (TRUE) atau salah (FALSE)

Operator	Simbol
AND	$\wedge, \cdot$
OR	$\vee, +$
NOT	$\sim, \neg$
IMPLIES	$\rightarrow$
Bi-implikasi	$\Leftrightarrow, \leftrightarrow$

# Logika Proporsional

A	B	$\sim A$	$\sim A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \Leftrightarrow B$
T	T	F	F	T	T	T	T
T	F	F	F	F	T	F	F
F	T	T	T	F	T	T	F
F	F	T	T	F	F	T	T



# Resolusi

- Resolusi digunakan untuk melakukan inferensi pada logika proposisi
- Resolusi adalah suatu aturan untuk melakukan inferensi yang dapat berjalan secara efisien dalam suatu bentuk khusus yaitu *conjunctive normal form* (CNF)

# CNF

- Langkah-langkah untuk mengubah suatu kalimat (konversi) ke bentuk CNF :
  - Hilangkan implikasi dan ekuivalensi
    - $x \rightarrow y$  menjadi  $\neg x \vee y$
    - $x \leftrightarrow y$  menjadi  $(\neg x \vee y) \wedge (\neg y \vee x)$
  - Kurangi lingkup semua negasi menjadi satu negasi saja
    - $\neg(\neg x)$  menjadi  $x$
    - $\neg(x \vee y)$  menjadi  $(\neg x \wedge \neg y)$
    - $\neg(x \wedge y)$  menjadi  $(\neg x \vee \neg y)$
  - Gunakan aturan assosiatif dan distributif untuk mengkonversi menjadi *conjunction of disjunction*
    - Assosiatif :  $(A \vee B) \vee C$  menjadi  $A \vee (B \vee C)$
    - Distributif :  $(A \wedge B) \vee C$  menjadi  $(A \vee C) \wedge (B \vee C)$
  - Buat satu kalimat terpisah untuk tiap-tiap konjungsi

# Contoh Resolusi

- Diketahui basis pengetahuan (fakta-fakta yang bernilai benar) :
  1.  $P$
  2.  $(P \wedge Q) \rightarrow R$
  3.  $(S \vee T) \rightarrow Q$
  4.  $T$
- Tentukan kebenaran  $R$

# Penyelesaian Contoh Resolusi

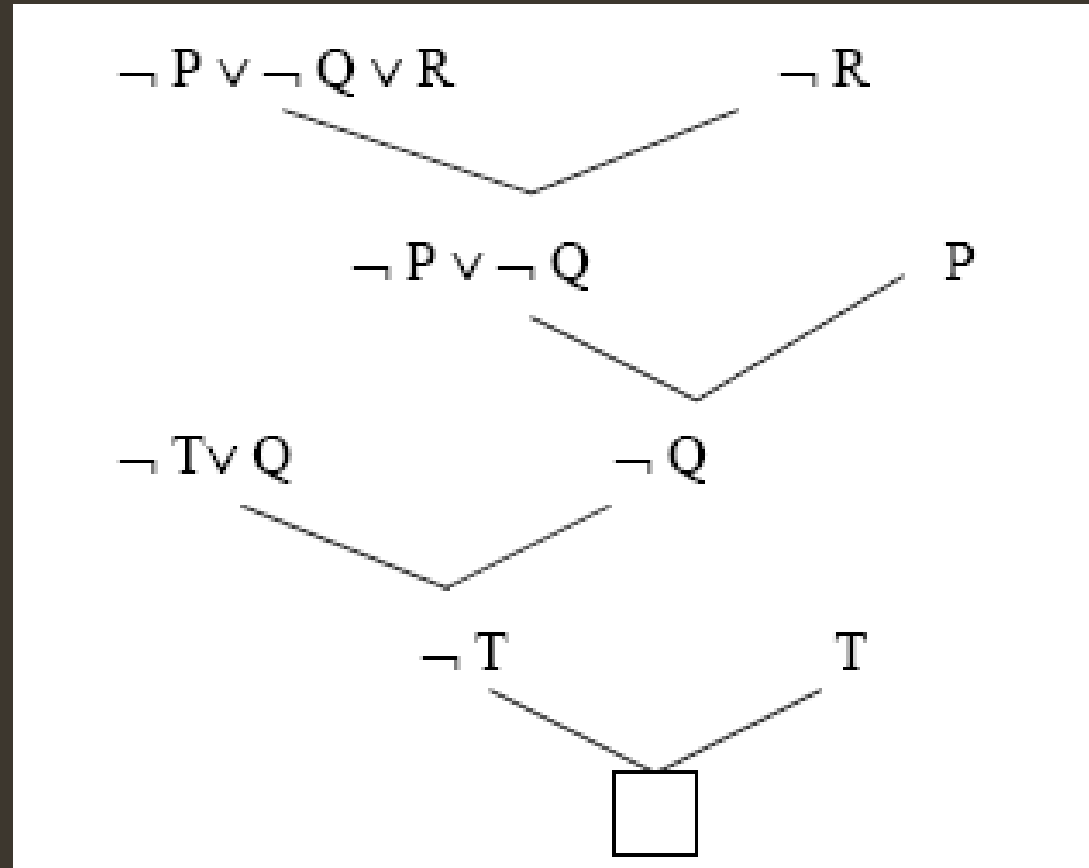
- Ubah ke dalam bentuk CNF

Kalimat	Langkah-langkah	CNF
1. P	Sudah merupakan bentuk CNF	P
2. $(P \wedge Q) \rightarrow R$	<ul style="list-style-type: none"><li>■ Menghilangkan implikasi : <math>\neg (P \wedge Q) \vee R</math></li><li>■ Mengurangi lingkup negasi : <math>(\neg P \vee \neg Q) \vee R</math></li><li>■ Gunakan asosiatif : <math>\neg P \vee \neg Q \vee R</math></li></ul>	$\neg P \vee \neg Q \vee R$
3. $(S \vee T) \rightarrow Q$	<ul style="list-style-type: none"><li>■ Menghilangkan implikasi : <math>\neg (S \vee T) \vee Q</math></li><li>■ Mengurangi lingkup negasi : <math>(\neg S \wedge \neg T) \vee Q</math></li><li>■ Gunakan distributif : <math>(\neg S \vee Q) \wedge (\neg T \vee Q)</math></li></ul>	$(\neg S \vee Q)$ $(\neg T \vee Q)$
4. T	Sudah merupakan bentuk CNF	T

# Penyelesaian Contoh Resolusi

- Tambahkan Kontradiksi pada tujuannya, sehingga fakta-fakta dalam CNF menjadi :

1.  $P$
2.  $\neg P \vee \neg Q \vee R$
3.  $\neg S \vee Q$
4.  $\neg T \vee Q$
5.  $T$
6.  $\neg R$



# Penyelesaian Contoh Resolusi

Contoh bila diterapkan dalam kalimat :

- P : Ani anak yang cerdas
- Q : Ani rajin belajar
- R : Ani akan menjadi juara kelas
- S : Ani makannya banyak
- T : Ani istirahatnya cukup

Kalimat yang terbentuk :

- Ani anak yang cerdas
- Jika ani anak yang cerdas dan ani rajin belajar, maka ani akan menjadi juara kelas
- Jika ani makannya banyak atau ani istirahatnya cukup, maka ani rajin belajar
- Ani istirahatnya cukup

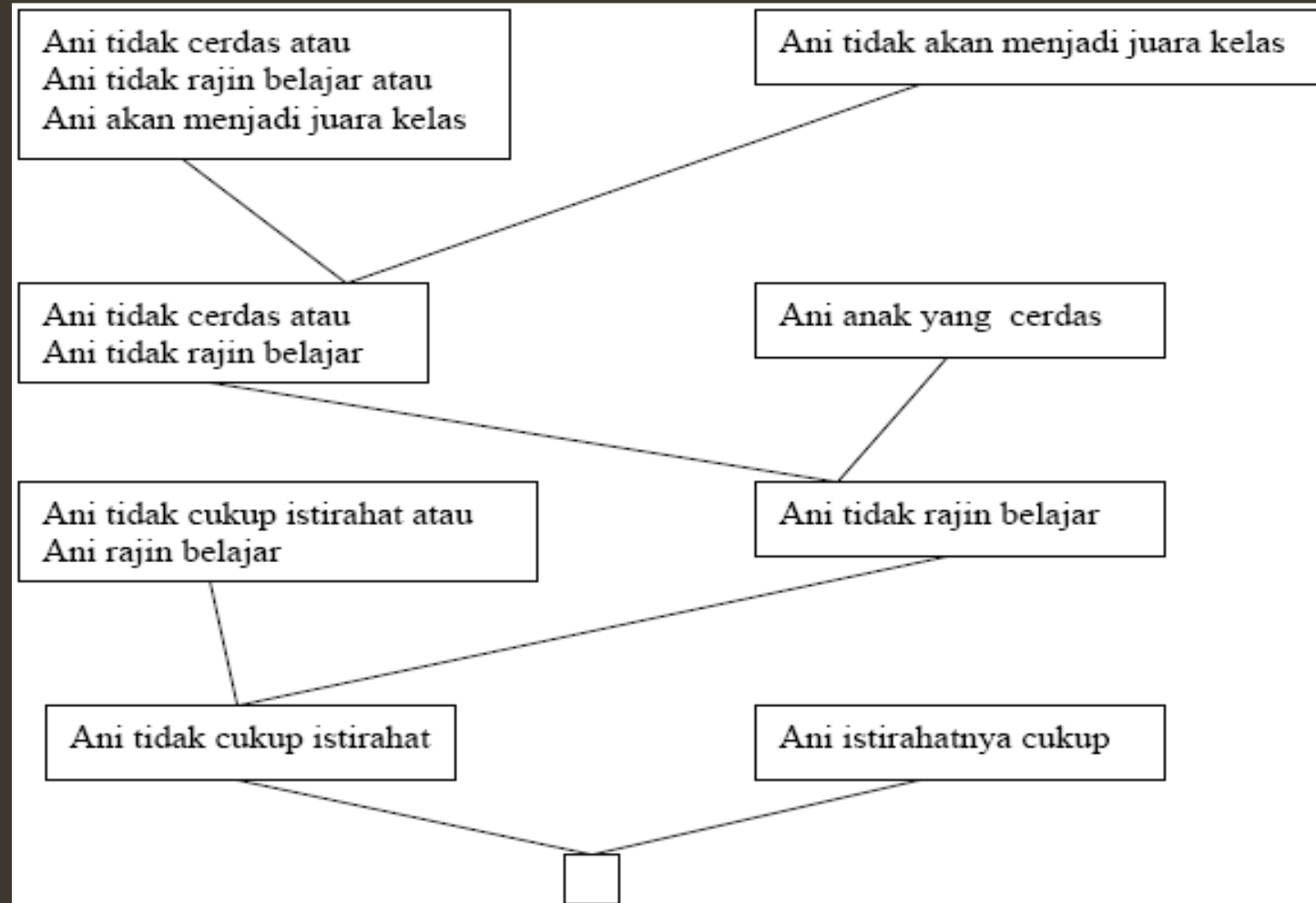
Setelah dilakukan konversi ke bentuk CNF, didapat :

Fakta ke-2 : Ani tidak cerdas atau ani tidak rajin belajar atau ani akan menjadi juara kelas

Fakta ke-3 : Ani tidak makan banyak atau ani rajin belajar

Fakta ke-4 : Ani tidak cukup istirahat atau ani rajin belajar

# Penyelesaian Contoh Resolusi



# Logika Predikat

- Suatu logika yang lebih canggih yang seluruhnya menggunakan konsep dan kaidah proporsional yang sama
- Disebut juga kalkulus predikat, yang memberi tambahan kemampuan untuk merepresentasikan pengetahuan dengan sangat cermat dan rinci
- Memungkinkan memecah statemen ke dalam bagian komponen yang disebut obyek, karakteristik obyek atau beberapa keterangan obyek
- Suatu proposisi atau premis dibagi menjadi dua bagian
  - Argumen (atau obyek)
    - Individu atau obyek yang membuat keterangan
  - Predikat (keterangan)
    - Keterangan yang membuat argumen dan predikat



# Logika Predikat

- Dalam suatu kalimat, predikat dapat berupa kata kerja atau bagian kata kerja
- **PREDIKAT** (individu[obyek]1, individu[obyek]2)
- Misalnya proposisi:
  - Mobil berada dalam garasi
- Dinyatakan menjadi
  - Di dalam (mobil, garasi)
  - Di dalam = produk (keterangan)
  - Mobil = Argumen (obyek)
  - Garasi = Argumen (obyek)

# Logika Predikat

- Contoh lain
  - Proposisi : Rojali suka Juleha
  - Kalkulus Predikat : SUKA (Rojali, Juleha)
- Proposisi : Pintu Terbuka
- Kalkulus Predikat : BUKA (pintu)
- Proposisi : Sensor cahaya aktif
- Kalkulus Predikat : AKTIF (sensor cahaya)

# Logika Predikat

- Pengetahuan diekspresikan dalam kalkulus predikat yang dapat dimanipulasi agar dapat diinferensi/dinalar
- Pangkalan pengetahuan dibentuk dengan menggunakan variabel sebagai simbol-simbol untuk merancang obyek
- misalnya
  - $x = \text{Rojali}$
  - $y = \text{Juleha}$
  - Maka proposisinya menjadi  $\text{Suka}(x,y)$

# Logika Predikat

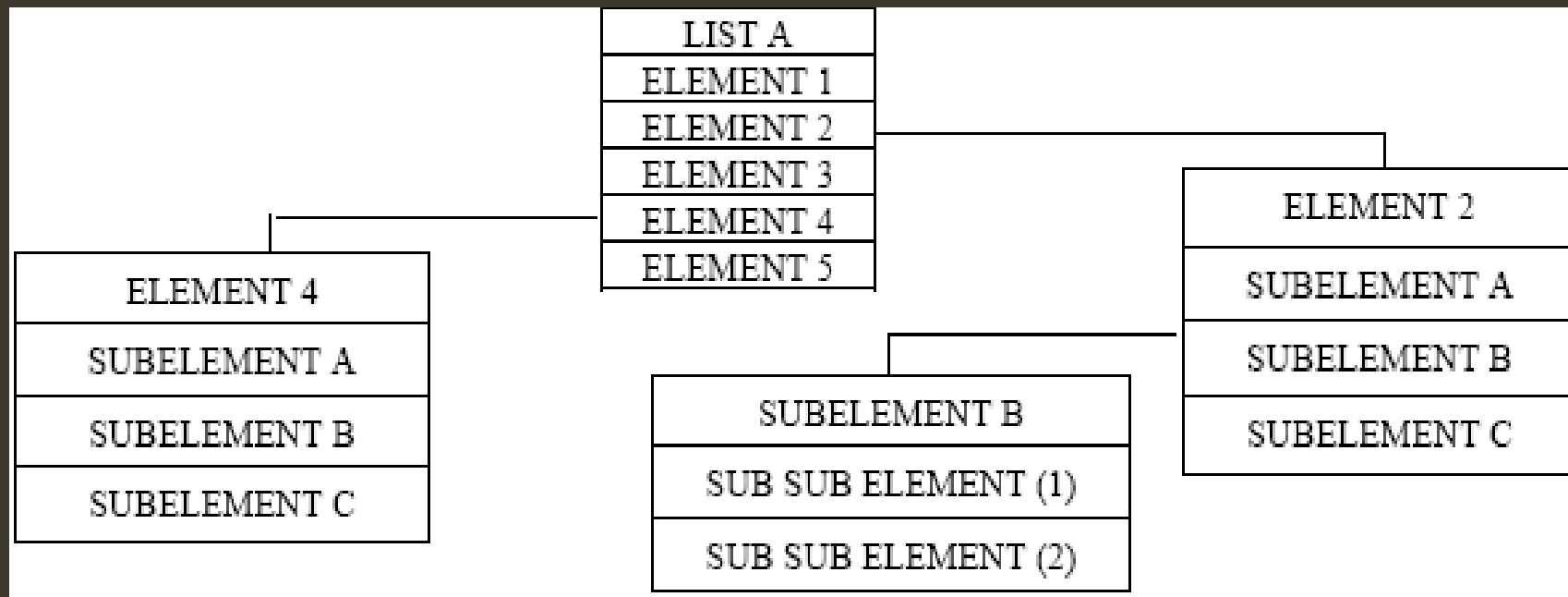
- Predikat kalkulus membolehkan penggunaan simbol untuk mewakili fungsi-fungsi
- Misalnya
  - ayah(Juleha) = Jojon
  - ibu(Rojali) = Dorce
- Fungsi dapat digunakan bersamaan dengan predikat
- Misalnya predikat berikut menjelaskan bahwa Jojon dan Dorce adalah berteman
- $\text{teman}(\text{ayah}(\text{Juleha}), \text{ibu}(\text{Rojali})) = \text{teman}(\text{Jojon}, \text{Dorce})$

# Pengukuran Kuantitas

- Adalah simbol yang memungkinkan untuk menyatakan suatu rangkaian atau cakrawala variabel dalam suatu ekspresi logika
- Dua pengukuran kuantitas, yaitu:
  - Kuantitas universal ( $\forall$ )
    - Untuk semua
  - Kuantitas eksistensial ( $\exists$ )
    - Ada / terdapat
- Contoh:
  - Semua sapi berkaki empat
  - $(\forall x)[\text{Sapi}(x), \text{berkaki empat}(x)]$
  - Beberapa sapi berwarna putih
  - $(\exists x)[\text{Sapi}(x), \text{berwarna putih}(x)]$

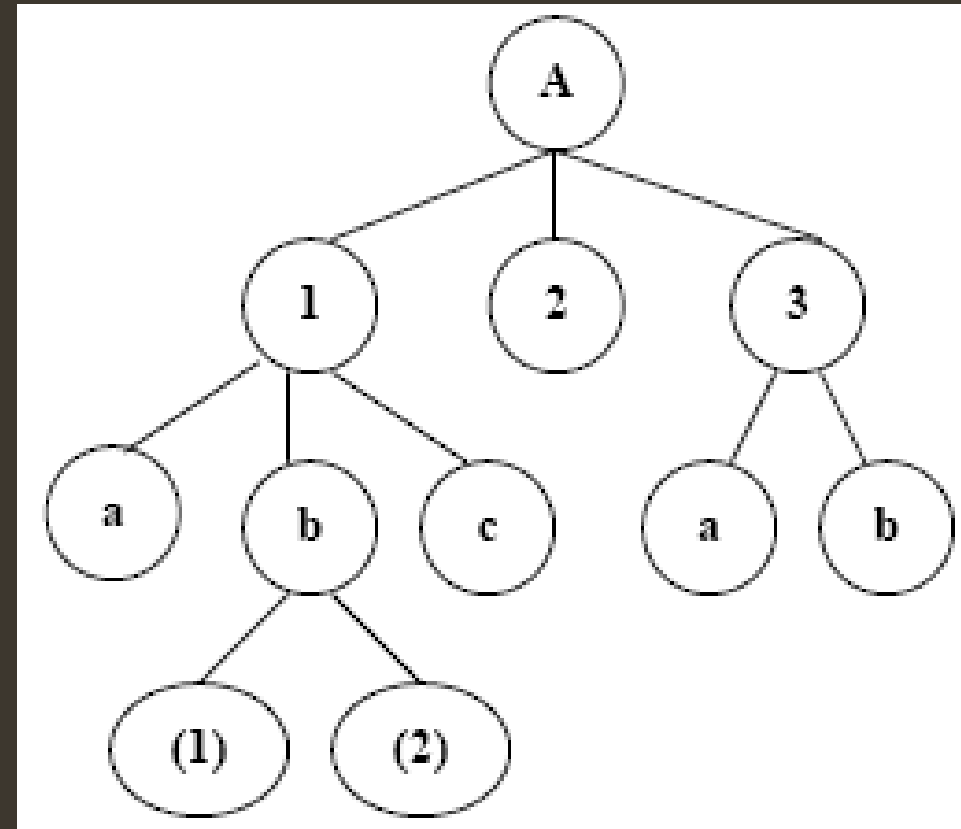
# List

- Struktur sederhana untuk representasi pengetahuan
- Daftar dari rangkaian materi yang terkait
- List digunakan untuk objek yang dikelompokkan, dikategorikan atau digabungkan



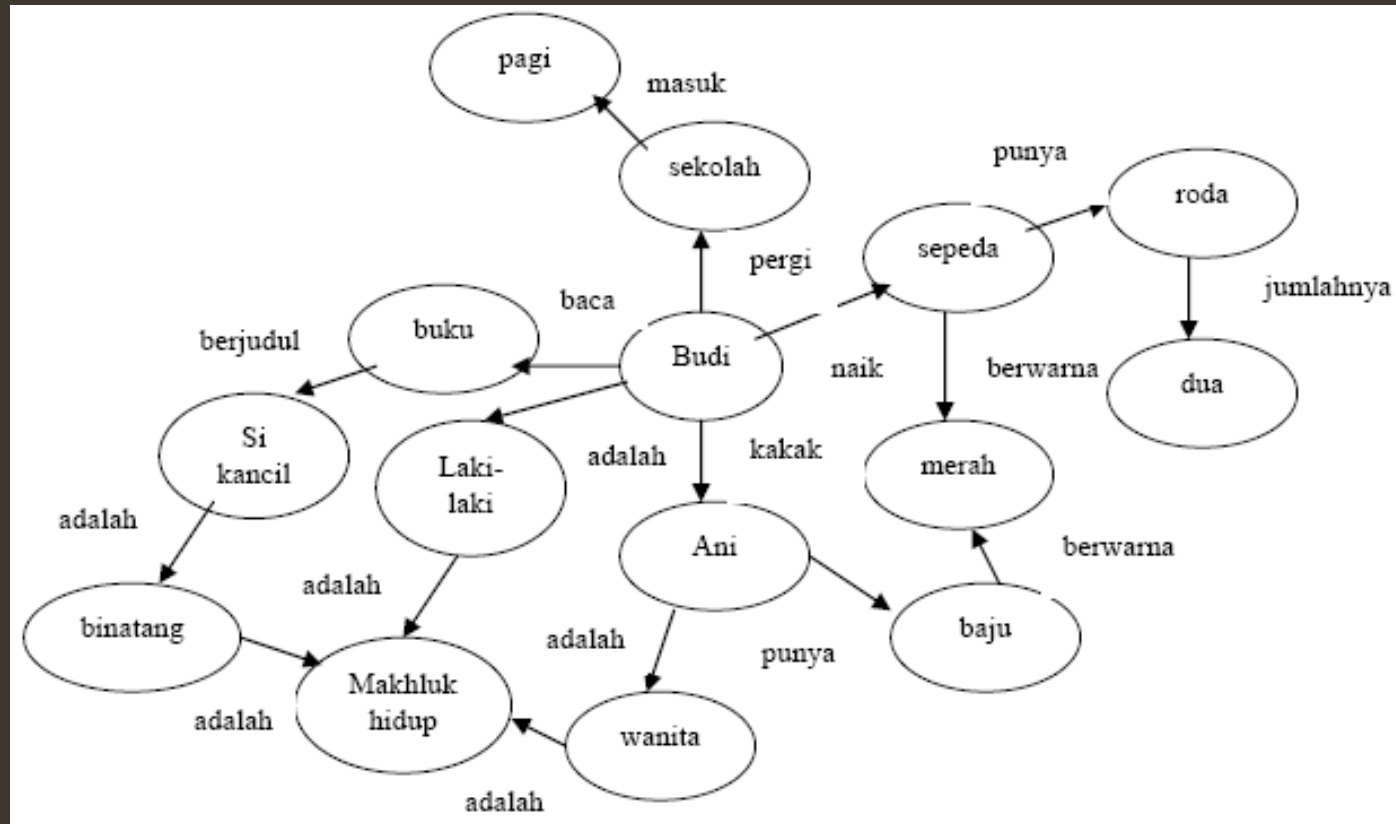
# Tree / Pohon

- Struktur sederhana untuk representasi pengetahuan
- Merupakan struktur grafik hirarki



# Jaringan Semantik

- Struktur sederhana untuk representasi pengetahuan grafis yang menunjukkan hubungan antar berbagai objek





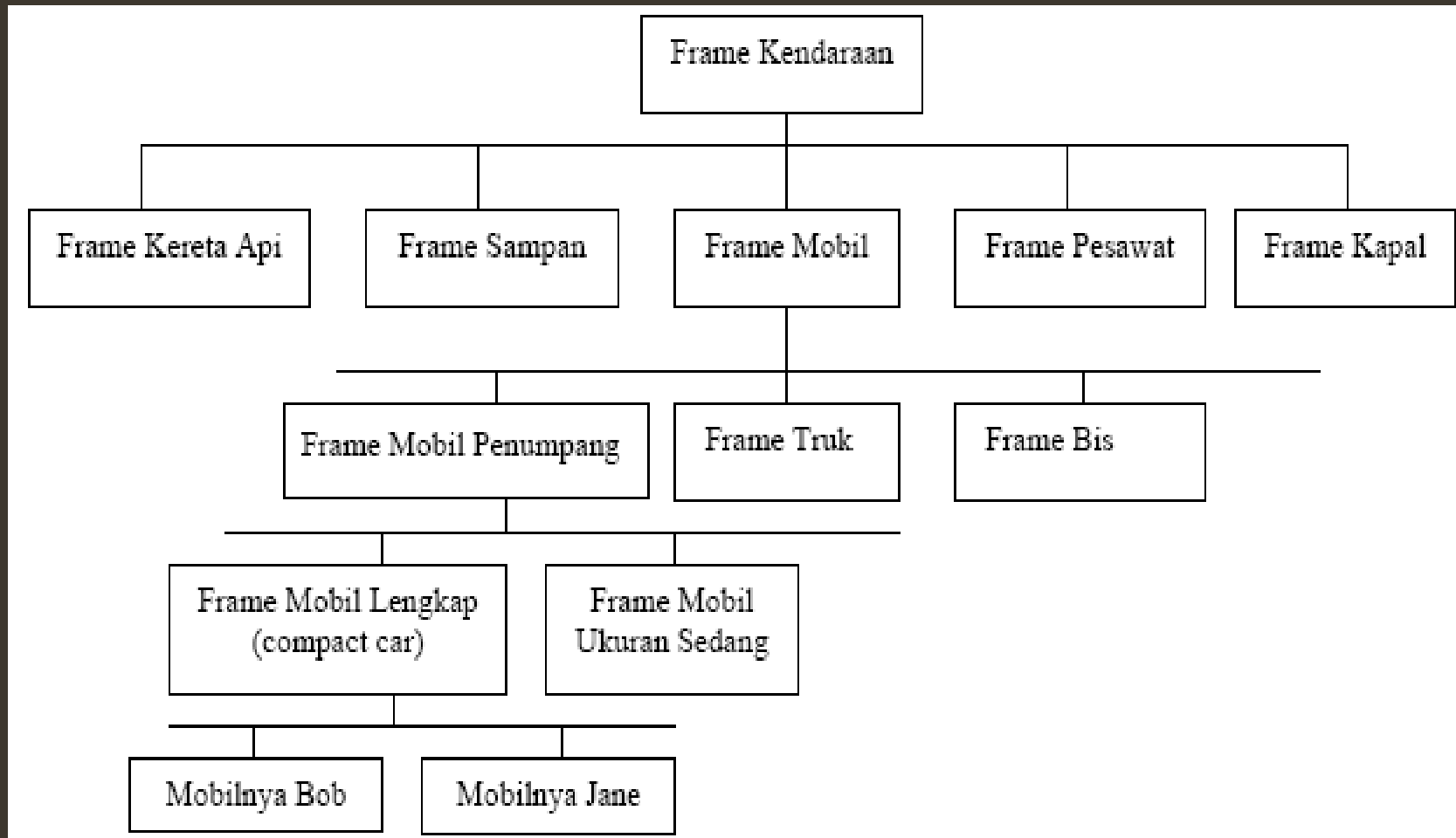
# Frame

- Frame → kumpulan pengetahuan tentang suatu objek tertentu, peristiwa, lokasi, situasi, berdasarkan pengalaman
- Frame → memiliki slot yang menggambarkan rincian (atribut) dan karakteristik objek.
- Hirarki Frame → susunan hirarki dari frame memungkinkan pewarisan frame

# Frame

<b>Frame Mobil</b>
Class : Transportasi
Nama pabrik : Audi
Negara : Jerman
Model : 5000 Turbo
Tipe : Sedan
Bobot : 3300 lb
Ukuran dasar roda : 105,8 inchi
Jumlah pintu : 4 (default)
Transmisi : 3-speed otomatis
Jumlah roda : 4 (default)
Mesin : (referensi kerangka mesin) <ul style="list-style-type: none"><li>• Tipe : in-line, overhead cam</li><li>• Jumlah silinder : 5</li></ul>
Akselerasi <ul style="list-style-type: none"><li>0-60 : 40,4 detik</li><li>¼ mil : 17,1 detik, 85 mph</li><li>Jarak gas : rata-rata 22 mpg</li></ul>
<b>Frame Mesin</b>
Kaliber silinder : 3,19 inci
Tak silinder : 3,4 inci
Rasio kompresi : 7,8 : 1
Sistem bahan bakar : injeksi dengan pertukaran turbo
Tenaga : 140 HP
Torsi : 160/ft/LB

# Hirarki Frame



# Hirarki Frame

Parent Frame Nama : Compact Car		Child Frame Nama : Mobilnya Jane	
Slot	Facets	Slot	Facets
Pemilik	Cek daftar registrasi	Pemilik	Jane
Warna	Daftar per manufaktur	Warna	Biru
No silinder		No silinder	6
Range	4 atau 6		
Jika dibutuhkan	Tanya pemilik		
Buatan		Buatan	Honda
Daftar range	Semua manufaktur		
Jika dibutuhkan	Tanya pemilik	Model	Accord
Model	Gunakan hubungan frame		
Model (tahun)		Model (tahun)	1992
Range	1950 – 2001		
Jika dibutuhkan	Tanya pemilik		

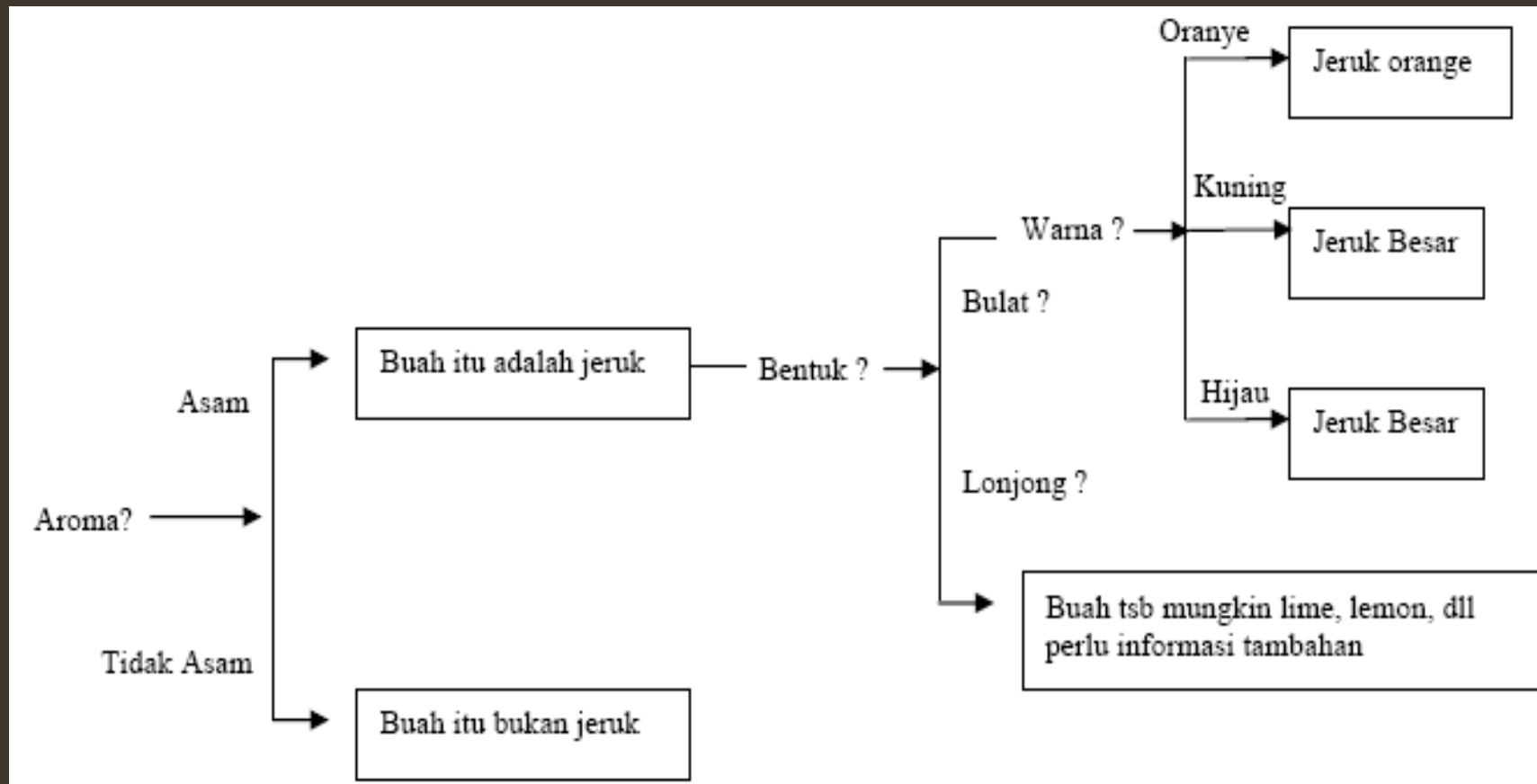
## Tabel Keputusan (Decision Table)

- Tabel keputusan → dalam format tabel
- Tabel dibagi 2 → bagian pertama untuk atribut & bagian kedua untuk nilai & kesimpulan

Atribut								
Bentuk	Bulat	Bulat	Bulat	Bulat	Lonjong	Lonjong	Lonjong	Lonjong
Aroma	Asam	Asam	Manis	Manis	Manis	Manis	Asam	Manis
Warna	Kuning	Oranye	Kuning	Merah	Kuning	Kuning	Oranye	Hijau
Rasa	Asam	Manis	Manis	Manis	Manis	Manis	Asam	Manis
Kulit	Kasar	Kasar	Halus	Halus	Halus	Halus	Halus	Halus
Kesimpulan								
Anggur	X							
Jeruk		X						
Apel			X	X				
Pisang					X			
Pir						X		X

# Pohon Keputusan (Decision Tree)

- Pohon keputusan → mudah dikonversi ke dalam bentuk aturan (rule)



# Naskah (Script)

- Naskah → sama dengan frame, bedanya menggambarkan urutan peristiwa
- Elemen script meliputi :
  1. Kondisi input → kondisi yang harus dipenuhi
  2. Track → variasi yang mungkin terjadi
  3. Prop → berisi objek-objek pendukung
  4. Role → peran yang dimainkan oleh seseorang
  5. Scene → adegan yang dimainkan
  6. Hasil → kondisi yang ada setelah urutan peristiwa dalam script terjadi.

# Sistem/Aturan Produksi (Production Rules)

- Representasi pengetahuan yang berupa aturan (rule) :
  - Antecedent → mengekspresikan situasi / premis (berawalan if)
  - Konsekuen → menyatakan tindakan tertentu jika premis benar (berawalam THEN)
- Contoh :
  - IF lalulintas pagi ini padat
  - THEN saya naik sepeda motor saja



# Sistem/Aturan Produksi (Production Rules)

Aturan dapat ditulis dalam beberapa bentuk :

## 1. IF premis THEN kesimpulan

**Jika** pendapatan tinggi **MAKA** pajak yang harus dibayar juga tinggi

## 2. Kesimpulan IF premis

Pajak yang harus dibayar tinggi **JIKA** pendapatan tinggi

## 3. Inclusion of ELSE

**IF** pendapatan tinggi **OR** pengeluaran tinggi, **THEN** pajak yang harus dibayar tinggi **ELSE** pajak yang harus dibayar rendah

## 4. Aturan yang lebih kompleks

**IF** rating kredit tinggi **AND** gaji lebih besar dari \$30,000 **OR** aset lebih dari \$75,000 **AND** sejarah pembayaran tidak miskin **THEN** pinjaman diatas \$10,000 disetujui dan daftar pinjaman masuk kategori "B"

# Sistem/Aturan Produksi (Production Rules)

2 metode penalaran yang menggunakan aturan :

- *Forward Reasoning* (penalaran maju)
  - Pelacakan dimulai dari keadaan awal (informasi atau fakta yang ada) dan kemudian dicoba untuk mencocokkan dengan tujuan yang diharapkan
  - Gunakan jika jumlah keadaan awal lebih kecil daripada tujuan & kejadian itu berupa fakta baru

## Sistem/Aturan Produksi (Production Rules)

- *Backward Reasoning* (Penalaran mundur)
  - Penalaran dimulai dari tujuan atau hipotesa, baru dicocokkan dengan keadaan awal atau fakta-fakta yang ada.
  - Jika jumlah keadaan awal lebih banyak daripada tujuan
  - Jika kejadian itu berupa query

# Contoh *Forward Reasoning*

- R1 : IF suku bunga turun THEN harga obligasi naik
- R2 : IF suku bunga naik THEN harga obligasi turun
- R3 : IF suku bunga tidak berubah THEN harga obligasi tidak berubah
- R4 : IF dolar naik THEN suku bunga turun
- R5 : IF dolar turun THEN suku bunga naik
- R6 : IF harga obligasi turun THEN beli obligasi

Apabila diketahui bahwa dolar turun, apa keputusan yang diambil, apakah akan membeli obligasi atau tidak

- ***Forward Reasoning*** :
  - Dari fakta dolar turun,
  - berdasarkan Rule 5, diperoleh konklusi suku bunga naik.
  - Dari Rule 2, suku bunga naik menyebabkan harga obligasi turun.
  - Dengan Rule 6, jika harga obligasi turun, maka kesimpulan yang diambil adalah membeli obligasi.

# Contoh *Backward Reasoning*

- R1 : IF suku bunga turun THEN harga obligasi naik
- R2 : IF suku bunga naik THEN harga obligasi turun
- R3 : IF suku bunga tidak berubah THEN harga obligasi tidak berubah
- R4 : IF dolar naik THEN suku bunga turun
- R5 : IF dolar turun THEN suku bunga naik
- R6 : IF harga obligasi turun THEN beli obligasi
- Apabila diketahui bahwa dolar turun, apa keputusan yang diambil, apakah akan membeli obligasi atau tidak
- ***Backward Reasoning :***
  - Dari solusi yaitu membeli obligasi
  - dengan menggunakan Rule 6 diperoleh anteseden harga obligasi turun
  - Dari Rule 2 dibuktikan harga obligasi turun bernilai benar jika suku bunga naik bernilai benar .
  - Dari Rule 5 suku bunga naik memang bernilai benar karena diketahui fakta dolar turun.