

## BAB 2

### AKUISISI DAN REPRESENTASI PENGETAHUAN

- ✓ Akan dibahas beberapa representasi pengetahuan yang digunakan secara umum untuk Expert System.
- ✓ Representasi pengetahuan merupakan hal yang penting dalam expert system, karena :
  - a. Shell expert system dirancang untuk type representasi pengetahuan tertentu, spt baris dan logika.
  - b. Cara expert system menunjukkan pengetahuan, akan memberikan efek pengembangan, efisiensi, kecepatan dan perawatan system.

#### 2.1. ARTI DARI PENGETAHUAN

- ✓ Pengetahuan mempunyai banyak arti. Sinonim dari **pengetahuan** → data, fakta, informasi.
- ✓ Berdasarkan sumbernya, **pengetahuan dibedakan menjadi 2** :
  1. **Pengetahuan Formal/ Deep Knowledge** : Pengetahuan yang terdapat dalam buku-buku, jurnal, buletin ilmiah. Pengetahuan formal dianggap sebagai pengetahuan yang bersifat umum.
  2. **Pengetahuan non Formal / Surface Knowledge** : Pengetahuan-pengetahuan praktis dalam bidang tertentu, yang diperoleh seorang pakar dari pengalamannya pada bidang tersebut dalam waktu yang cukup lama.
- ✓ **Berdasarkan cara merepresentasikannya**, Pengetahuan dapat diklasifikasikan menjadi :
  - a. **Procedural Knowledge** → Diketahui bagaimana caranya melakukan sesuatu.

Contoh : Diketahui bagaimana caranya menumpahkan pot air.

- b. **Declarative knowledge** → Mengacu pada pengetahuan, bahwa sesuatu benar atau salah.

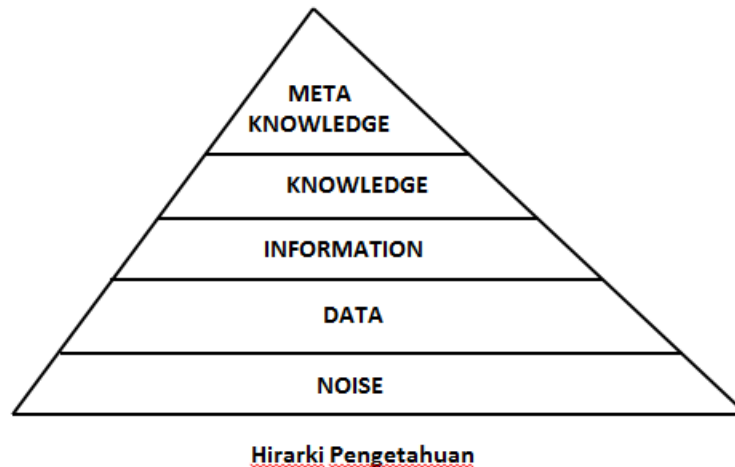
Contoh :

- c. **Tacit knowledge / Unconscious knowledge/Pengetahuan Heuristik** ←  
Karena tdk dapat diekspresikan dengan bahasa.

Contoh :

- Bagaimana memindahkan tangan kita?.  
→ Dengan otot rilek atau kaku  
⇒ Spt apa rilek atau kakunya
- Bagaimana berjalan atau naik sepeda?. → Susah diekspresikan dengan bahasa.

- ✓ Jadi pengetahuan adalah penting dalam expert system.
- ✓ **Analogi ekspresi jenis Wirth** : → Algorithms + Data structures = Programs
- ✓ **Untuk expert system** : → Knowledge + Inference = Expert systems



- **Noise** → berisi item yang tidak menarik dan data yang tidak jelas.
  - **Data** → Berisi informasi yg menarik
  - **Information** → Berisi informasi yang sangat khusus
  - **Knowledge** → Diaktifkan oleh fakta untuk membuat fakta baru atau kesimpulan.
  - → Bentuk fakta dapat berupa data atau informasi. Berdasarkan pada bagaimana fakta dituliskan, expert system menggambarkan kesimpulan dengan menggunakan data atau informasi.
- ✓ **Expert system akan :**
1. Memisahkan data dari noise
  2. Mentransformasi data ke dalam informasi ... atau
  3. Menginformasi data ke dalam pengetahuan.
- ✓ **Contoh** : Perhatikan rangkaian dari 24 angka berikut ini :
- 137178766832525156430015
- Tanpa pengetahuan, seluruh rangkaian tsb mungkin muncul sbg noise.
  - Jika diketahui bahwa rangkaian tsb berarti, maka rangkaian angka tsb adalah data.

## 2.2. REPRESENTASI PENGETAHUAN

- **Ada beberapa metode representasi pengetahuan** yang biasa digunakan :
  1. Metode kalkulus predikat
  2. Bingkai/frame
  3. Jaringan Semantik /Semantic network
  4. Metode kaidah produksi
  5. Representasi logika

### 2.2.1. KALKULUS PREDIKAT

- **Kalkulus Predikat** → Merupakan cara sederhana untuk merepresentasikan pengetahuan secara deklaratif.
- Dalam kalkulus predikat, **pernyataan deklaratif dibagi menjadi 2 bagian:**
  - a. Bagian predikat
  - b. Bagian argument
- Contoh :
  1. **Baju disimpan di lemari** dapat ditulis sebagai berikut :  
**disimpan di (lemari, baju)**

dimana :

disimpan = **Predikat**

lemari, baju = **Argumen**

Dalam Kalkulus predikat, **argument dapat berupa variabel**, misal :

**Amir mencintai Tuti**

Bila Amir = x & Tuti = y, maka bentuk kalkulus predikatnya :

**mencintai (x, y)**

### 2.2.2. BINGKAI/FRAME

- **Bingkai** :
  - ✓ Blok-blok atau potongan-potongan yang berisi pengetahuan mengenai obyek-obyek tertentu, kejadian, lokasi, situasi maupun elemen-elemen lainnya dengan ukuran yang relative besar.
  - ✓ Struktur data yang mengandung semua informasi / pengetahuan yang relevan dari suatu objek.
- **Blok-blok tersebut** menggambarkan → **Obyek-obyek secara lebih rinci**.
- Pengetahuan dalam bingkai dibagi-bagi ke dalam slot atau atribut, yang dapat mendiskripsikan pengetahuan secara deklaratif atau prosedural.
- **CONTOH** :

Bingkai untuk merepresentasikan pengetahuan mengenai gajah

- ✓ Bingkai Gajah

kelas = mamalia

Warna = abu-abu

Ukuran = besar

BINGKAI MOBIL	
Klas	: Transportasi
Pabrik	: Audi
Negara asal	: Jerman
Model	: 3000 turbo
Tipe	: Sedan
Berat	: 500 kg
Jumlah roda	: 4
BINGKAI MESIN	
Ukuran Silinder	: 3.19 inci
Rasio kompresi	: 7,9 – 1
Tenaga	: 140 Hp

### 2.2.3. KAIDAH PRODUKSI

- ✓ **Metode Kaidah Produksi**, biasanya dituliskan dalam bentuk : jika-maka (*if-then*)
- ✓ **Kaidah** ini → dikatakan sebagai hubungan implikasi antara **bagian premise** (jika) dan **bagian konklusi** ( maka).
- ✓ Sebuah kaidah, terdiri dari klausa-klausa.
- ✓ Sebuah Klausa, mirip dengan sebuah kalimat dengan : subyek, kata kerja dan obyek yang menyatakan suatu fakta.
- ✓ Ada sebuah **klausa premise** dan sebuah **klausa konklusi** pada setiap kaidah.
- ✓ Suatu kaidah juga dapat terdiri atas **beberapa premise**, dan juga **lebih dari satu konklusi**.
- ✓ Contoh :
  - a. Jika **hari ini hujan** maka **saya tidak jadi pergi**
  - b. Jika saya lulus **dan** saya diterima di PT maka saya akan beli baju baru **atau** saya akan beli laptop baru.
  - c. Jika rumah saya sudah laku **atau** mobil saya sudah laku maka saya akan segera melunasi hutang
  - d. Gejala hama walang sangit

Kaidah 1 : IF Daun busuk

THEN Terserang hama walang sangit

Kaidah 2 : IF Bulir padi kosong

THEN Terserang hama walang sangit

Kaidah 3 : IF Bulir padi kosong AND Daun busuk

THEN Terserang hama walang sangit

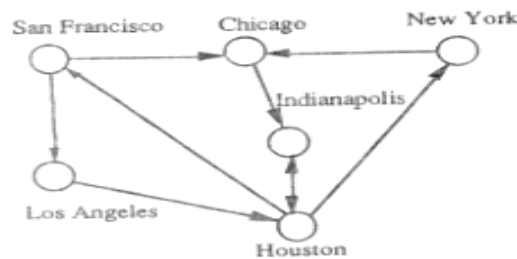
ELSE Tidak terserang hama walang sangit

#### 2.2.4. JARINGAN SEMANTIK

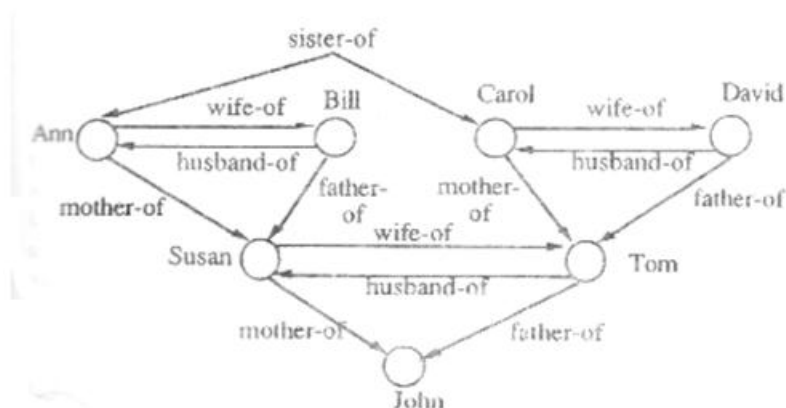
✓ **Semantic Network / Proportional Net/ Jaringan Semantik :**

- Teknik representasi AI klasik yang digunakan untuk informasi proporsional.
- Cara merepresentasikan pengetahuan yang paling tua dan paling mudah.
- Merupakan jaringan data dan informasi yang menunjukkan hubungan antar berbagai obyek, dimana informasi yang terhubung tersebut adalah informasi yang proporsional ( pernyataan yang dapat bernilai benar atau salah).

✓ **Jaringan semantic** → merupakan **graf berarah**, yang terdiri dari simpul/node dan busur/arc yang menghubungkannya.



(a).Jaringan Umum  
( Menunjukkan route pesawat udara antar kota)

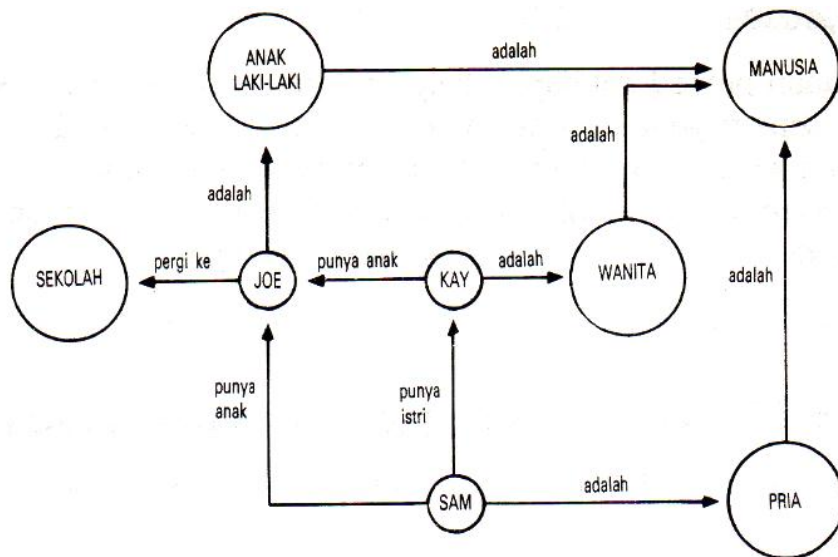


(b).Jaringan Semantik  
(Menunjukkan hubungan antar anggota keluarga)  
Gambar 2.1. Dua Tipe Jaringan

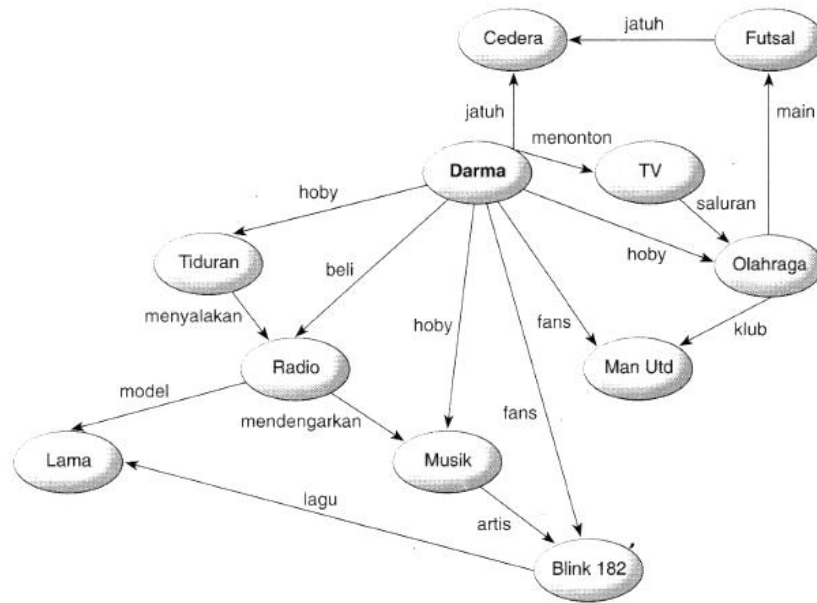


Gambar 2.2. Jaringan Semantik dengan IS-A dan jenis hubungan (AKO)

- ✓ **AKO ( A KIND OF )** → menghubungkan node "generic" ke node "generic",
- IS-A** → menghubungkan node "individual" ke node "generic".







Contoh-contoh Jaringan Semantik

## 2.2.5. LOGIKA DAN JARINGAN

- ✓ Bagian yang penting dalam pemberian alasan adalah → **Pembuatan kesimpulan dari suatu kalimat.**
- ✓ **Logika** : Bentuk representasi pengetahuan yang paling tua.
- ✓ **Proses Logika** → adalah Proses membentuk kesimpulan / menarik kesimpulan berdasarkan fakta yang telah ada.
- ✓ **Input dari Proses Logika** → Premis / fakta-fakta yang diakui kebenarannya, sehingga dengan melakukan penalaran pada proses logika, dapat dibentuk suatu kesimpulan yang benar.
- ✓ **Ada 2 penalaran yang dapat dilakukan** untuk mendapat konklusi :
  1. **Penalaran Deduktif** : Penalaran dimulai dari prinsip umum, untuk mendapatkan kesimpulan yang lebih khusus.

Contoh :

Premis Mayor : Jika bulir padi kosong, berarti terserang hama walang sangit.

Premis Minor : Bulir padi kosong → Konklusi : Padi terserang hama walang sangit.

2. **Penalaran Induktif** : Penalaran yang dimulai dari fakta-fakta khusus untuk mendapatkan kesimpulan umum.

Contoh :

Premis 1 : Penyakit blast adalah penyakit berbahaya

Premis 2 : Penyakit gosong adalah penyakit berbahaya

Premis 3 : Penyakit bercak daun adalah penyakit berbahaya

Konklusi : Penyakit padi adalah penyakit berbahaya.

- ✓ Aplikasi komputer untuk menunjukkan pemberian alasan telah dihasilkan oleh “**Logic Programming**”.
- ✓ **Sylogisme** → adalah suatu cara untuk menunjukkan pengetahuan, contohnya sbb :

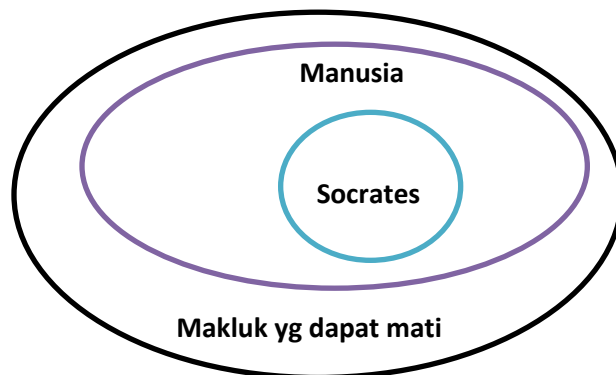
$P_1$  : Semua manusia akan mati

$P_2$  : Socrates adalah manusia

Kesimpulan : Socrates akan mati

- ✓ Cara lain untuk menunjukkan pengetahuan adalah dengan **diagram Venn**.

CONTOH :



- ✓ Dalam bentuk matematika, lingkaran dari diagram venn menunjukkan suatu rangkaian yang merupakan koleksi dari obyek

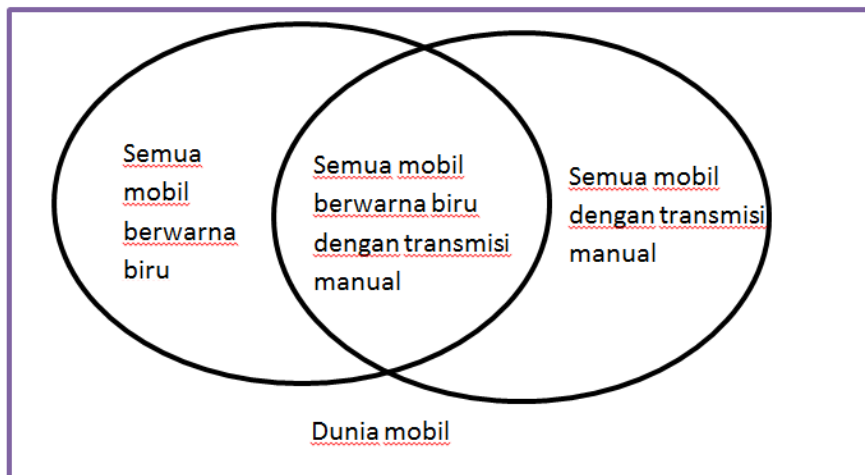
CONTOH :

A = Himpunan dari seluruh mobil berwarna biru.

B = Himpunan semua mobil dengan transmisi manual.

C = Himpunan semua mobil berwarna biru dan dengan transmisi manual

Maka :  $C = A \cap B = \{x | x \in A \text{ dan } x \in B\}$



## 2.5. LOGIKA PROPORSIONAL

- ✓ **Type tertua dan paling sederhana** dari **logika formal / formal logic** adalah → **Silogisme**.
- ✓ Aljabar → adalah logika formal angka. Misal :  
Suatu sekolah mempunyai 25 komputer dengan total memory 60 board.  
Beberapa computer mempunyai 2 memory board dan yang lain mempunyai 4 memori board. Hitung jumlah computer tiap tipe.

JAWAB:

x = banyak computer tipe 1

y = banyak computer tipe 2

Shg :  $x + y = 25$

$2x + 4y = 60$

Maka  $x = 20$ ,  $y = 5$

- ✓ **Logika formal** memungkinkan kita berkonsentrasi pada **pemberian alasan**, tanpa dibingungkan oleh obyek apa yang kita beri alasan.
- ✓ Perhatikan silogisme berikut ini :

$P_1$  : All X are Y

$P_2$  : Z is a X

Conclusion : Z is a Y

Adalah valid dengan apapun yang mengganti X, Y dan Z → Karena arti tidak menjadi masalah dalam logika formal, **hanya bentuk / kemunculan yg penting**.

- ✓ Suatu kalimat yang mempunyai kebenaran disebut : **Pernyataan atau Proposisi**.
- ✓ Kalimat yang tidak dapat ditentukan nilai kebenarannya disebut **Kalimat Terbuka**.
- ✓ Kalimat biasanya diklasifikasikan menjadi 4 tipe :

Tipe	Contoh
Imperatif	Apakah dapat kau katakan padaku?.
Interogasif	Apakah itu?.
Kalimat seru	Itu benar!.
Deklaratif	Bulan maret berumur 31 hari

- ✓ **Kalimat majemuk / Compound Statement** → dibentuk dengan menggunakan hubungan logika pada pernyataan individual. **Hubungan logika/koneksi logika** diantaranya :

Konektif	Arti
$\wedge$	AND, konjungsi
$\vee$	OR, Disjungsi
$\sim$	NOT, Negasi
$\rightarrow$	Jika...maka...; Implikasi / Kondisional
$\leftrightarrow$	Jika hanya jika; Biimplikasi/bikondisional

- ✓  $p \leftrightarrow q \approx (p \rightarrow q) \wedge (q \rightarrow p)$
- ✓ **Tautologi** : Pernyataan gabungan yang selalu benar → Contoh :  $p \vee \sim p$
- ✓ **Kontradiksi** : Pernyataan gabungan yang selalu salah → Contoh :  $p \wedge \sim p$

## QUANTIFIER UNIVERSAL

- ✓ **Logika predikat** dihubungkan dengan **struktur internal** dari suatu kalimat, yaitu dihubungkan dengan **kata khusus** yang disebut **quantifier**, yaitu : Setiap, beberapa, tidak.
- ✓ Universal quantifier ditunjukkan dengan symbol :  $\forall$  diikuti  $\geq 1$  argumen untuk “ domain variabel”.

✓  $\forall \rightarrow$  Dibaca “ Untuk setiap “ atau “ Untuk semua”.

✓ Contoh dalam domain jumlah :

$(\forall x) (x + x = 2x) \dots \rightarrow$  **Dibaca** : “ Untuk setiap  $x$ , kalimat  $x+x = 2x$  adalah benar “.

$\rightarrow$  Bisa ditulis :  $(\forall x) (p)$

✓ Contoh lain :  $p$  menunjukkan kalimat “ Seluruh anjing adalah binatang”, bisa ditulis sbb:

$(\forall x) (p) \equiv (\forall x) (\text{jika } x \text{ adalah anjing} \rightarrow x \text{ adalah binatang})$

✓ Pernyataan kebalikannya : “ **Tidak ada anjing yang binatang** “, ditulis sbb :

$(\forall x) (\text{jika } x \text{ adalah anjing} \rightarrow \sim x \text{ adalah binatang})$

Bisa dibaca :

- Setiap anjing bukan binatang
- Semua anjing bukan binatang

✓ Contoh lain :” Seluruh segitiga adalah polygon “, ditulis sbb:

$(\forall x) (x \text{ adalah segitiga} \rightarrow x \text{ adalah poligon})$

“ Untuk semua  $x$ , jika  $x$  adalah segitiga, maka  $x$  adalah polygon.

✓ Cara yang lebih pendek untuk menuliskan kalimat logika mencakup predikat adalah dengan menggunakan : “ **Predikat Functions**”.

$(\forall x) (x \text{ adalah segitiga} \rightarrow x \text{ adalah poligon})$  bisa ditulis sbb:

$(\forall x) (\text{segitiga}(x) \rightarrow \text{poligon}(x))$

✓ **Fungsi Predikat** biasanya ditulis dalam notasi yang lebih singkat dengan menggunakan huruf besar untuk menunjukkan predikat.

Misal : S = segitiga dan P = polygon , maka :

$(\forall x) ( \text{segitiga}(x) \rightarrow \text{poligon}(x) )$  ditulis dgn lebih singkat sbb :

$(\forall x) ( S(x) \rightarrow P(x) )$

- ✓ Contoh lain : Misal M = fungsi predikat untuk manusia & D=fungsi predikat untuk mati, maka “ Seluruh manusia akan mati “ dapat ditulis sbb:

$(\forall x) ( M(x) \rightarrow D(x) )$

## EKSISTENSI QUANTIFIER

- ✓ **Eksistensial quantifier** : Menjelaskan suatu pernyataan yang benar untuk minimal 1 anggota domain.

Dituliskan dengan  $\exists$  diikuti  $\geq 1$  argumen.

- ✓ Contoh :

$(\exists x) ( x.x = 1 ) \rightarrow \text{dibaca} : \text{Ada beberapa } x, \text{ yang jika dikalikan dengan } x \text{ sendiri } =1$

Contoh lain :

$(\exists x) ( \text{gajah}(x) \wedge \text{nama}(\text{Clyde}) ) \rightarrow \text{Dibaca} : \text{Ada beberapa gajah dengan nama Clyde.}$

- ✓ Contoh lain :

- $(\forall x) ( \text{gajah}(x) \rightarrow \text{empat} - \text{kaki}(x) ) \rightarrow \text{Semua gajah mempunyai 4 kaki}$
- $(\exists x) ( \text{gajah}(x) \wedge \text{tiga} - \text{kaki}(x) ) \rightarrow \text{Ada gajah yang mempunyai 3 kaki}$

Contoh	Arti
(1a) $(\forall x) (P)$	Semua gajah adalah binatang mamalia
(1b) $(\exists x) (\sim P)$	Ada gajah yang bukan binatang mamalia
(2a) $(\exists x) (P)$	Ada gajah yang binatang mamalia
(2b) $(\forall x) (\sim P)$	Semua gajah bukan binatang mamalia

✓ (1b) negasi dari (1a) dan (2b) negasi dari (2a)

## QUANTIFIER DAN SET/JARINGAN

✓ **Quantifier** dapat digunakan untuk menentukan **set /jaringan** atas himpunan semesta S sbb :

Set expression	Logical Equivalent
$A=B$	$\forall x (x \in A \leftrightarrow x \in B)$
$A \subseteq B$	$\forall x (x \in A \rightarrow x \in B)$
$A \cap B$	$\forall x (x \in A \wedge x \in B)$
$A \cup B$	$\forall x (x \in A \vee x \in B)$
$A'$	$\forall x (x \in S \mid \sim(x \in A))$

✓ Contoh :

$E \subset M \rightarrow$  Seluruh gajah adalah mamalia.

✓ Bentuk logika dan himpunan dari hukum de Morgan :

Set	Logic
$(A \cap B)' = A' \cup B'$	$\sim(p \wedge q) \equiv \sim p \vee \sim q$
$(A \cup B)' = A' \cap B'$	$\sim(p \vee q) \equiv \sim p \wedge \sim q$





## 2.6. MEMILIH TEKNIK REPRESENTASI PENGETAHUAN

**Ada 4 kriteria dalam memilih teknik representasi pengetahuan:**

- a. **Kemampuan representasi** → Teknik yang dipilih harus mampu merepresentasikan semua jenis pengetahuan yang akan dimasukkan ke dalam sistem pakar.
- b. **Kemudahan dalam penalaran** → Teknik yang dipilih harus mudah diproses untuk memperoleh kesimpulan.
- c. **Efisiensi proses akuisisi** → teknik yang dipilih harus membantu pemindahan pengetahuan dari pakar ke dalam computer.
- d. **Efisiensi proses penalaran** → teknik yang dipilih harus dapat diproses dengan efisien untuk mencapai kesimpulan.

## 2.7. AKUISISI PENGETAHUAN

- **Akuisisi pengetahuan** juga merupakan hal yang penting, selain teknik representasi pengetahuan.
- **Teknik Akuisisi pengetahuan**, biasanya dikerjakan oleh seorang **Knowledge Engineer** : Orang yang memiliki latar belakang pengetahuan tentang computer dan mengerti cara pengembangan sistem pakar.
- Dalam tahap Akuisisi pengetahuan, → knowledge engineer berusaha menyerap pengetahuan.
  - ✓ Pengetahuan diperoleh dari seorang pakar, buku, jurnal ilmiah, dll.
  - ✓ Pengetahuan dari pakar dapat diperoleh dengan wawancara.
  - ✓ Pengetahuan yang diperoleh harus selengkap mungkin ( Krn akan mempengaruhi kemampuan sistem pakar yang akan dibuat).

- Dalam **melakukan wawancara dengan pakar**, ada beberapa hal yang harus diperhatikan :
  - a. Pertanyaan yang diajukan harus spesifik, agar penjelasan yang diberikan pakar dapat terarah dan rinci, sehingga seluruh informasi yang berhubungan dengan pertanyaan tersebut dapat diperoleh.
  - b. Beri pakar kesempatan untuk menjelaskan dengan caranya sendiri, jangan menghindari metode penyajian yang diberikan pakar, jangan memaksa pakar untuk menjelaskan dengan cara yang tidak biasa digunakannya.
  - c. Jangan memotong penjelasan dari pakar. Hindari pengajuan fakta, yang menyebabkan pakar ragu akan informasi yang diberikan.
  - d. Gunakan alat perekam untuk menghindari kehilangan informasi dari hasil wawancara.
  - e. Perhatikan cara pakar memanfaatkan pengetahuan dalam menyelesaikan masalah.

### **Teknik-teknik lain untuk memperoleh pengetahuan dari pakar:**

#### **1. Observasi**

→ Melihat langsung pakar menyelesaikan masalah di lapangan,

#### **2. Diskusi masalah**

→ Menggali data, pengetahuan dan prosedur yang dibutuhkan pakar untuk menyelesaikan suatu masalah.

#### **3. Deskripsi Masalah**

→ Pakar mendeskripsikan masalah pada setiap kategori solusi dalam domain permasalahan.

#### **4. Analisa Permasalahan**

→ Memberikan beberapa persoalan kepada pakar untuk menyelesaikan rangkaian penalarannya.

5. **Tatacara perbaikan** → Pakar memberikan beberapa masalah untuk diselesaikan oleh Knowledge Engineer, dan pakar memperbaiki cara penyelesaian tersebut berdasarkan aturan dari hasil wawancara
6. **Tatacara Pengujian** → Pakar mengevaluasi dan mengkritik prototype kaidah dan struktur pengendalian dari sistem yang dibangun.
7. **Tatacara Validasi** → Knowledge Engineer membentuk prototype sistem pakar berdasarkan hasil penyelesaian masalah yang diajukan, lalu diuji oleh pakar lain.

## 2.8. KETIDAKPASTIAN

- Dalam kenyataan sehari-hari, pakar sering berurusan dengan fakta-fakta yang tak menentu dan tidak pasti.
- Sistem pakar juga harus bisa menangani masalah kekurangpastian / ketidakpastian tsb.
- Teknik-teknik yang digunakan untuk menangani ketidakpastian adalah :
  1. Certainty factor / Faktor kepastian
  2. Probabilitas.
  3. Fuzzy set Theory / Teori gugus tidak pasti.

## FAKTOR KEPASTIAN / CERTAINTY FACTOR (CF)

- Faktor Kepastian (CF) :
  - ✓ Nilai untuk mengukur keyakinan pakar
  - ✓ Diperoleh dari pengurangan nilai kepercayaan / measure of belief (MB) oleh nilai ketidakpercayaan / measure of disbelief (MD) →  

$$CF[h,e] = MB[h,e] - MD[h,e]$$

$$CF[h,e] \quad : \text{Faktor kepastian}$$

MB[h,e] : Ukuran kepercayaan/tingkat keyakinan terhadap hipotesis h, jika diberikan / dipengaruhi evidence (fakta) e (antara 0-1)

MD[h,e] : Ukuran ketidakpercayaan/tingkat ketidakyakinan terhadap hipotesis h, jika diberikan / dipengaruhi evidence e (antara 0-1)

✓ ..

- Tujuan penggunaan Faktor Kepastian (CF) :
  - ⇒ Untuk mengolah ketidakpastian dari fakta dan gejala, dengan menghindarkan keperluan data dan perhitungan yang besar
- Rule dengan “premise majemuk” yang dihubungkan dengan operator “DAN” atau “ATAU”, yang masing-masing mempunyai nilai CF sendiri-sendiri, maka :
  - ✓ Nilai CF gabungan untuk rule dengan penghubung “DAN” → adalah CF yang terkecil.
  - ✓ Nilai CF gabungan untuk rule dengan penghubung “ATAU” → adalah CF yang terbesar.

CONTOH :

1. Jika saya mempunyai uang lebih (CF=0,4)

**dan** tidak turun hujan (CF=0,7)

**maka** saya akan pergi memancing (CF gabungan = 0,4 ( yg terkecil))

2. Jika saya terima bonus (CF=0,5)

**atau** Amin membayar hutangnya ke saya (CF=0,8)

maka malam ini saya nonton (CF = 0,8 (yang terbesar) )

- Untuk CF dari gabungan 2 buah rule dirumuskan sebagai berikut :
$$CF_{gabungan} = CF(x) + CF(y) - CF(x) * CF(y)$$

CONTOH :

Rule 1 : Jika hewan berbulu

dan hewan menyusui

maka hewan tersebut tergolong mamalia ( $CF=0,6$ )

Rule 2 : Jika hewan tergolong mamalia

dan hewan tersebut makan daging

maka hewan tersebut jenis karnivora ( $CF = 0,8$ )

Maka  $CF_{gabungan} = 0,6+0,8 - 0,6*0,8 = 0,92$ .

Ada 3 jenis selang Faktor Kepastian (CF) yang biasa digunakan:

- a. Nilai 0 untuk pernyataan yang salah, nilai 1 untuk pernyataan yang benar,
- b. Selang  $0 - 1 \rightarrow$  Nilai 0 berarti salah mutlak, nilai 1 berarti benar mutlak.  
Selang nilai  $0 < CF < 1$  menunjukkan derajat kepastian.
- c. Selang  $(-1) - 1 \rightarrow -1$  berarti salah mutlak & nilai 1 berarti benar mutlak.  
Selang nilai  $-1 < CF < 0$  menunjukkan derajat kesalahan & Selang nilai  $0 < CF < 1$  menunjukkan derajat kebenaran.