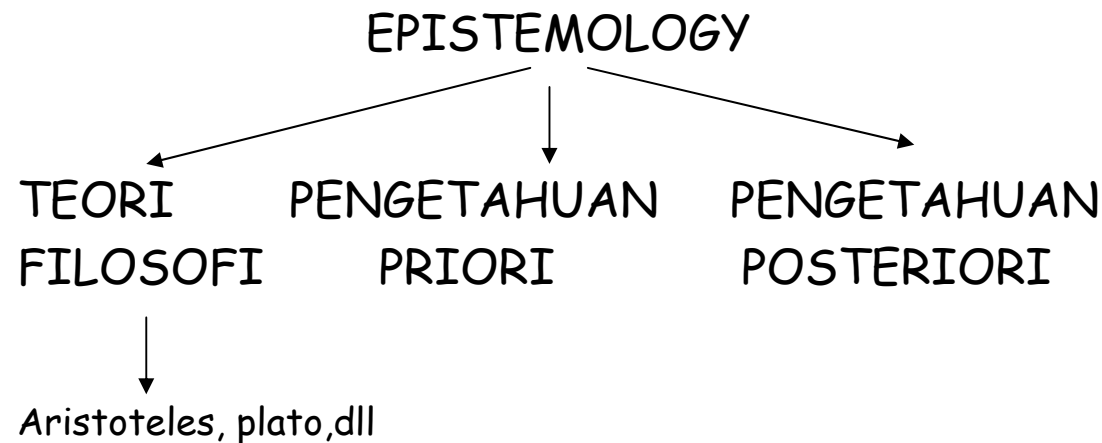


## REPRESENTASI PENGETAHUAN (1)

Representasi pengetahuan merupakan hal penting dalam SP karena:

1. Shell SP didesign untuk type representasi pengetahuan tertentu seperti baris dan logika
2. Akan memberikan efek/akibat pengembangan, efisiensi, kecepatan dan perawatan system.

Study pengetahuan disebut *epistemology*



## **PENGETAHUAN PRIORI**

- ✓ Berasal dari bahasa Latin
- ✓ Berarti yang mendahului (pengetahuan datang sebelumnya dan bebas dari arti)
- ✓ Contoh pernyataan "segalanya memiliki sebab" , " seluruh triangle dalam pesawat mempunyai 180 derajat"
- ✓ Contoh lain pernyataan logika, hukum matematika
- ✓ Disebut secara universal benar dan tidak dapat ditentukan tanpa kontradiksi.

## **PENGETAHUAN POSTERIORI**

- ✓ Adalah pengetahuan yg diperoleh dari arti
- ✓ Kebenaran dari pengetahuannya menggunakan pengalaman.

## **PENGETAHUAN PROSEDURAL**

- ✓ Bagaimana melakukan sesuatu

## **PENGETAHUAN DEKLARATIF**

- ✓ Mengacu pada sesuatu benar atau salah

## **PENGETAHUAN TACIT/UNCONSCIOUS**

✓ Tidak dapat diekspresikan dg bahasa.

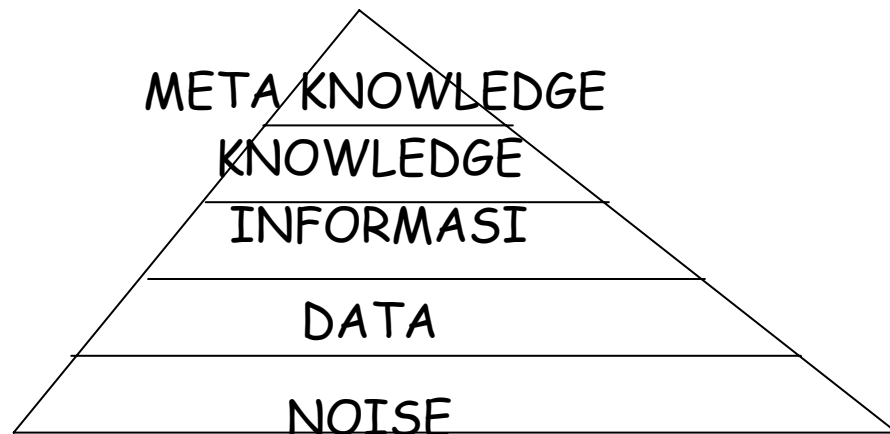
## **ANALOGY      EKSRESI      PENGETAHUAN** **MENURUT WIRTH**

**Algorithms + Data Structures = Programs**

Untuk SP :

**Knowledge + Inference = Expert Systems**

## **HIRARKI PENGETAHUAN :**



SP juga :

1. Memisahkan data dari noise
2. Mentrasformasikan data ke dalam informasi

3. Mentrasformasikan data ke dalam pengetahuan

## **METAKNOWLEDGE**

- ✓ Adalah pengetahuan mengenai beberapa perbedaan domain
- ✓ Menentukan basis pengetahuan mana yg sesuai

## **PRODUKSI**

Teknik representasi pengetahuan mencakup : baris, jaringan semantik, frame, scrips, bahasa representasi pengetahuan (spt KL-1)

## **BNF (BACKUS NAUR FORM)**

- ✓ Format notasi untuk menentukan produksi yaitu metalanguage yaitu untuk menentukan syntax bahasa.
- ✓ Metalanguage diatas bahasa normal (meta berarti diatas)
- ✓ Type bahasa : bahasa natural, bahasa logika, matematika, bahasa komputer

✓ Notasi BNF sederhana : kalimat yg berisi kata benda dan kata kerja diikuti oleh titik

✓ Baris produksi :

**<sentence> ::= <subject><verb><end-mark>**

✓ < > dan ::= merupakan simbol dari metalanguage

✓ ::= berarti "ditentukan sebagai" (sama dg →)

✓ < > simbol nonterminal (yaitu variabel yg menunjukkan bentuk lain)

✓ simbol | berarti atau

Contoh :

**<sentence> → <subject> <verb> <end-mark>**

**<subject> → I | You | We**

**<verb> → left | came**

**<end-mark> → . | ? | !**

Produksinya ?.....

✓ Serangkaian terminal disebut **string**

✓ **Kalimat valid** : jika string didapatkan dari start simbol dg menggantikan nonterminal dg baris definisinya.

✓ **Grammar** : set/rangkaian baris produksi lengkap yg menentukan suatu bahsa secara tidak ambigius.

✓ **Grammar valid** :

**<sentence> → <subject> <verb> <object> <end-mark>**

Contoh :

**<sentence> → <subject phrase> <verb> <object phrase>**

**<subject phrase> → <determiner> <noun>**

**<object phrase> → <determiner> <adjective> <noun>**

**<determiner> → a | an | this | these | those**

**<noun> → man | eater**

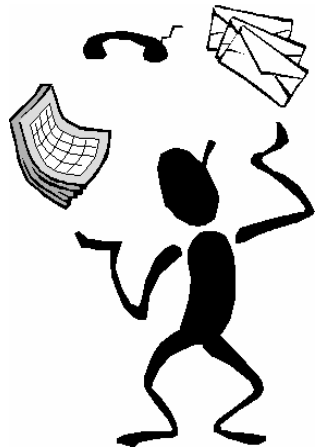
**<verb> → is | was**

**<adjective> → dessert | heavy**

Bagaimana derivative tree ?...

- ✓ < **determine** > digunakan untuk menunjukkan item tertentu
- ✓ **Parse tree** atau **derivation tree** adalah representasi grafik dari kalimat yg diuraikan ke dalam seluruh terminal dan nonterminal yg digunakan untuk mendapatkan kalimat.

## REPRESENTASI PENGETAHUAN (2)



- Representasi Pengetahuan (*Knowledge Representation*) dimaksudkan untuk menangkap sifat-sifat penting masalah dan membuat informasi dapat diakses oleh prosedur pemecahan masalah.
- Bahasa representasi harus dapat membuat seorang programmer mampu mengekspresikan pengetahuan untuk mendapatkan solusi suatu masalah.

- Secara singkat Mylopoulos dan Levesque mengklasifikasikan susunan atau pola representasi menjadi empat kategori :

### *1. Representasi Logika*

Representasi ini menggunakan ekspresi-ekspresi dalam logika formal untuk merepresentasikan basis pengetahuan.

### *2. Representasi Prosedural*

Menggambarkan pengetahuan sebagai sekumpulan instruksi untuk memecahkan suatu masalah. Dalam sistem yang berbasis aturan, aturan if-then dapat ditafsirkan sebagai sebuah prosedur untuk mencapai tujuan pemecahan masalah.

### *3. Representasi Network*

Menyatakan pengetahuan sebagai sebuah graf dimana simpul-simpulnya menggambarkan obyek atau konsep dalam masalah yang dihadapi,

sedangkan lengkungannya menggambarkan hubungan antar mereka. Contohnya adalah jaringan semantik.

#### *4.Representasi Terstruktur*

Memperluas network dengan cara membuat setiap simpulnya menjadi sebuah struktur data kompleks yang berisi tempat-tempat bernama slot dengan nilai-nilai tertentu. Nilai-nilai ini dapat merupakan data numerik atau simbolik sederhana, pointer ke bingkai (*frame*) lain, atau bahkan merupakan prosedur untuk mengerjakan tugas tertentu. Contoh : skrip (*script*), bingkai (*frame*) dan obyek (*object*).

### REPRESENTASI LOGIKA



Representasi logika terdiri dari dua jenis yaitu *Kalkulus proposisional (Propositional logic)* dan *Kalkulus predikatif (Predicate logic)*.

### Kalkulus Proposisional (*Propositional Logic*)

• Proposisi adalah suatu model untuk mendeklarasikan suatu fakta. Lambang-lambang proposisional menunjukkan proposisi atau pernyataan tentang segala sesuatu yang dapat benar atau salah.

#### Lambang-lambang kalkulus proposisional :

1. Lambang pernyataan proposisional  
 $P, Q, R, S, T, \dots$  (disebut sebagai atom-atom)
2. Lambang kebenaran  
benar (*True*) , salah (*False*)
3. Lambang penghubung  
 $\wedge$  (konjungsi),  $\vee$  (disjungsi),  $\sim$  (negasi),  
 $\rightarrow$  (implikasi),  $\leftrightarrow$  (Bi-implikasi),  
 $\equiv$  (equivalen)

Berikut ini adalah tabel kebenaran (*truth value*) lambang penghubung :

P	Q	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

### Equivalen



Suatu kalimat (formula) P dianggap *equivalen* dengan formula Q jika dan hanya jika '*truth value*' dari P sama dengan '*truth value*' dari Q untuk setiap interpretasinya. (ditulis sbg.  $P \equiv Q$ )

Contoh:

$$P \rightarrow Q \equiv \sim P \vee Q$$

P	Q	$\sim P$	$P \rightarrow Q$	$\sim P \vee Q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

- Kalimat-kalimat atau formula dalam kalkulus proposisional dibentuk dari lambang-lambang dasar tersebut.
- Nilai-nilai kebenaran yang dikandung oleh kalimat-kalimat proposisional disebut interpretasi.
- Secara formal, interpretasi diartikan sebagai pemetaan dari lambang-lambang proposisional menuju ke himpunan  $\{T, F\}$  yakni himpunan 'benar-salah'.

- Suatu formula (kalimat) yang mempunyai  $n$  lambang (atom) yang berbeda, mempunyai  $2^n$  interpretasi.
- Interpretasi yang menyebabkan suatu formula bernilai benar dikatakan *satisfy the formula*.
- Suatu formula dikatakan tautology jika dan hanya jika bernilai benar untuk setiap interpretasinya.  
Contoh :  $(A \vee \sim A)$ .
- Suatu formula dikatakan inconsistency jika dan hanya jika bernilai salah untuk setiap interpretasinya.  
Contoh :  $(A \wedge \sim A)$ .
- Suatu formula dikatakan consistent jika tidak inconsistent. Dengan kata lain, suatu formula yang consistent, paling tidak ada satu

interpretasi yang benar. Contoh  $((B \vee C) \wedge \sim C) \vee D$ .

- Jika suatu formula tautology maka consistent, tetapi tidak berlaku sebaliknya.
- Tautology disebut juga *valid formula*
- Inconsistency disebut juga *unsatisfiable formula*
- Consistency disebut juga *satisfiable formula*

Hukum yang berlaku untuk ekspresi proposisional P, Q dan R adalah :

1. Hukum de Morgan :  $\sim(P \vee Q) \equiv (\sim P \wedge \sim Q)$

2. Hukum de Morgan :  $\sim(P \wedge Q) \equiv (\sim P \vee \sim Q)$

3. Hukum distributif :

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$

4. Hukum distributif:

$$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$$

5. Hukum komutatif :  $(P \wedge Q) \equiv (Q \wedge P)$

6. Hukum komutatif :  $(P \vee Q) \equiv (Q \vee P)$

7. Hukum asosiatif :  
 $((P \wedge Q) \wedge R) \equiv (P \wedge (Q \wedge R))$

8. Hukum asosiatif :  
 $((P \vee Q) \vee R) \equiv (P \vee (Q \vee R))$

9. Hukum kontrapositif :  
 $(P \rightarrow Q) \equiv (Q \rightarrow \sim P)$

### Prosedur Pembuktian Teorema

- Suatu formula  $G$  dikatakan sebagai sebuah konsekuensi logis dari formula  $F_1, F_2, \dots, F_n$  jika dan hanya jika setiap interpretasi yang memenuhi  $(F_1 \wedge F_2 \wedge \dots \wedge F_n)$  juga memenuhi  $G$ .

$F_1, F_2, \dots, F_n$  disebut premis  
 $G$  disebut *Goal* dari formula

- Dengan kata lain, formula  $G$  adalah konsekuensi logis dari premis  $F_1, F_2, \dots, F_n$  jika dan hanya jika  $((F_1 \wedge F_2 \wedge \dots \wedge F_n) \rightarrow G)$  adalah Tautology.

- Karena negasi dari suatu *Tautology* adalah *Inconsistency*, maka  $\sim((F_1 \wedge F_2 \wedge \dots \wedge F_n) \rightarrow G)$  adalah Inconsistency.

- Kita tahu bahwa  
 $\sim((F_1 \wedge F_2 \wedge \dots \wedge F_n) \rightarrow G) \equiv$   
 $\sim(\sim(F_1 \wedge F_2 \wedge \dots \wedge F_n) \vee G) \equiv$   
 $(F_1 \wedge F_2 \wedge \dots \wedge F_n) \wedge \sim G$

- Dua Metode Pembuktian Teorema:

1. Metode Langsung (*Direct Method*) membuktikan bahwa  $((F_1 \wedge F_2 \wedge \dots \wedge F_n) \rightarrow G)$  adalah Tautology.
2. Metode *Refutasi* membuktikan bahwa :  $(F_1 \wedge F_2 \wedge \dots \wedge F_n) \wedge \sim G$  adalah Inconsistency.

### Contoh soal:

Buktikan bahwa  $Q$  adalah konsekuensi logis dari premis  $P$  dan  $(P \rightarrow Q)$ !



### Solusi:

1. Metode Langsung, membuktikan bahwa  $((P \wedge (P \rightarrow Q)) \rightarrow Q)$  adalah Tautology.

P	Q	$P \rightarrow Q$	$P \wedge (P \rightarrow Q)$	$(P \wedge (P \rightarrow Q)) \rightarrow Q$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

2. Metode Refutasi, membuktikan bahwa  $(P \wedge (P \rightarrow Q) \wedge \sim Q)$  adalah Inconsistency.

P	Q	$\sim Q$	$P \rightarrow Q$	$P \wedge (P \rightarrow Q)$	$P \wedge (P \rightarrow Q) \wedge \sim Q$
T	T	F	T	T	F
T	F	T	F	F	F
F	T	F	T	F	F
F	F	T	T	F	F

### Rules of Inference (Aturan-aturan Inferensi)

- Pendekatan lain untuk membuktikan teorema yang menggunakan aturan/*rule* (dinamakan *Rules of inference*), adalah dengan cara mendeduksi konsekuensi logis dari premis-premis yang diketahui atau diberikan.

- Beberapa contoh *Rules of Inference* adalah:

1. *Introducing Conjunction*

If F and G then  $(F \wedge G)$

2. *Eliminating Conjunction*

If  $(F \wedge G)$  then F

If  $(F \wedge G)$  then G

3. *Introducing Disjunction*

If F then  $(F \vee G)$

If G then  $(F \vee G)$

4. *Modus Ponens*

If F and  $(F \rightarrow G)$  then G

### 5. Modus Tollens

If  $\sim G$  and  $(F \rightarrow G)$  then  $\sim F$

### 6. Chaining

If  $(F \rightarrow G)$  and  $(G \rightarrow H)$  then  $(F \rightarrow H)$

### 7. Equivalen

If  $F$  and  $(F \equiv G)$  then  $G$

If  $G$  and  $(F \equiv G)$  then  $F$

### Contoh soal:

Bila diberikan premis-premis sebagai berikut:

- (i) *John awakens*
- (ii) *John brings a mop*
- (iii) *Mother is deligthed, if john awakens and cleans his room*
- (iv) *If John brings a mop, then he cleans his room.*

Buktikan dengan *Rules of Inference* (deduksi), dimana *goal*-nya adalah : *Mother is deligthed!*

### Solusi :

Tuliskan premis tersebut sebagai simbol (atom):

$A = \text{John awakens}$

$B = \text{John brings a mop}$

$C = \text{John cleans his room}$

$D = \text{Mother is delighted}$

Goal yang ingin dibuktikan adalah  $D$

Tuliskan premis tersebut sebagai formula:

(1)  $A$

(2)  $B$

(3)  $A \wedge C \rightarrow D$

(4)  $B \rightarrow C$

Deduksi dengan *Rules of Inference*

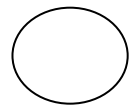
(5)  $C$  (dng. *Modus Ponens* (2) dan (4))

(6)  $A \wedge C$  (dng. *Intro. Conjunction* (1) dan (5))

(7)  $D$  (dng. *Modus Ponens* (3) dan (6))

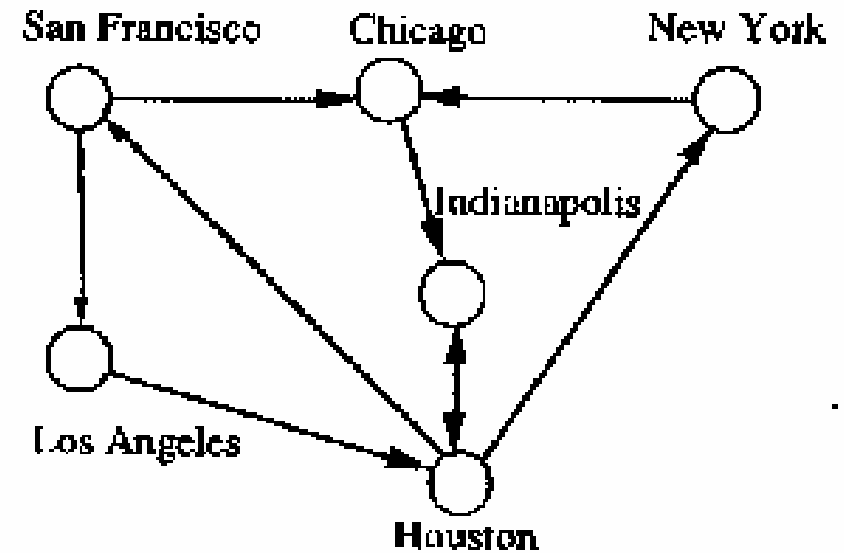
## JARINGAN SEMANTIK

- ✓ Sering disebut **proportional net**
- ✓ Bentuk dari pengetahuan deklaratif krn proporsi trb menunjukkan fakta
- ✓ Proporsi selalu benar atau salah disebut juga "atomic"
- ✓ Merupakan gambaran grafis yg menunjukkan hubungan antar berbagai objek. Yaitu dlm bentuk "nodes" dan "arcs" yg menghubungkannya

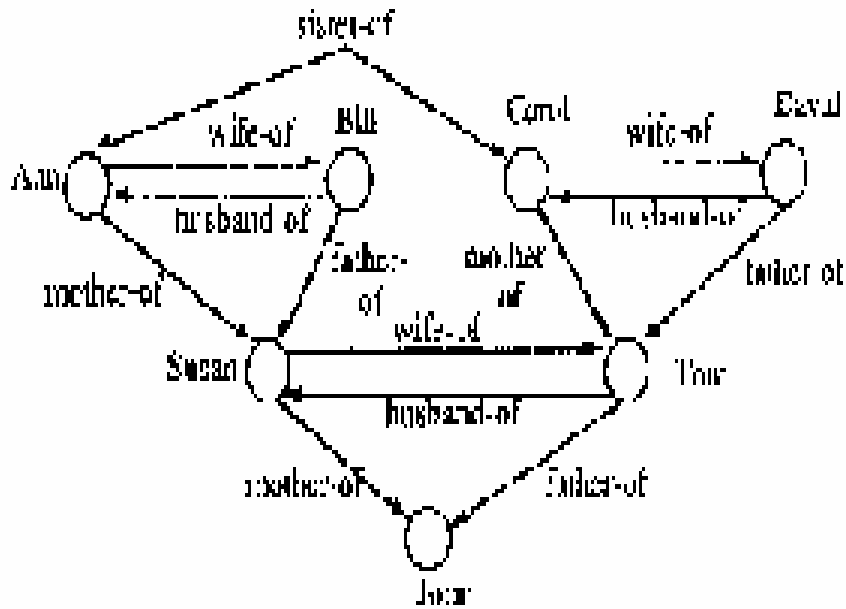


✓ → Nodes disebut juga dgn **objek**, digunakan untuk menunjukkan objek fisik, konsep, situasi

- ✓ → : Links atau edges atau arcs, untuk mengekspesikan suatu relasi
- ✓ Contoh route pesawat terbang (directed graph)



- ✓ Disebut juga **associative nets**, krn node dihubungkan dg yang alin.
- ✓ Bentuk links **IS-A, HAS-A, A-KIND-OF (AKO)**
- ✓ **IS-A** menunjukkan hubungan kelas, pada gbr diatas menunjukkan "jarak dari"
- ✓ **HAS-A** digunakan untuk mengidentifikasi karakteristik atau atribut objek node.
- ✓ **AKO** digunakan untuk menghubungkan satu jenis ke jenis yg lain



- ✓ Salah satu masalah pd jaringan semantik adalah *tidak adanya standar definisi nama link*
- ✓ **Object-attribute-value triple (OAV) atau triplet** digunakan untuk memberi karakter semua pengetahuan dlm jaringan semantik
  - *Object* dapat berupa fisik atau konsepsi
  - *Attribute* adalah karakteristik dari object
  - *Values* adalah ukuran spesifik dari *attribute* dalam situasi tertentu.

**Contoh: O-A-V item**

<u>Object</u>	<u>Attribute</u>	<u>Values</u>
Rumah	Kamar tidur	2, 3, 4 dst.
Rumah	Warna	Putih, Biru, dst.
Kamar tidur	Ukuran	2x3, 3x3, 3x4, dst
Diterima di universitas	Nilai ujian masuk	A, B, C, atau D

**Bahasa PROLOG**

- ✓ Model : Pemrograman Logika
- ✓ Jenis data : Simbolik dan numerik, predikat, list

Kalkulus Predikatif	PROLOG	Arti
$\cap$	,	dan
$\cup$	;	atau
$\leftarrow$	:-	menyebabkan
+	not	tidak, bukan

- ✓ Variabel dinyatakan sebagai string karakter alfanumerik dimulai dengan huruf besar :

likes(X,ana)

- ✓ Contoh :

likes(doni,tina),likes(doni,ana)

likes(doni,ana) :- likes(doni,tina)

not(likes(tina,ana))

- ✓ Contoh :

Proposisi : Mobil berada didalam garasi

Kalkulus predikat : didalam (mobil,garasi)

- ✓ Contoh lain :

1. red is a color

2. Tom is the father of John

3. Tom and Susan are the parents of John

Jawab :

1. color (red)

2. father\_of (Tom,John)

3. parents (Tom,Susan,John)

## **SCHEMATA**

- ✓ Jaringan semantik contoh dari **Shallow knowledge Structure** karena seluruh pengetahuan jaringan semantik diisikan dalam link dan node

- ✓ **Concept schema** : dengan skema tsb kita dapat menunjukkan konsep.

- ✓ Contoh konsep mengenai binatang, setiap orang mempunyai persepsi sendiri mengenai bintang (berkaki 4 atau 2 , berbulu atau bersisik dsb)

- ❖ Banyak pengetahuan yang digunakan dalam proses penalaran kita sehari hari yang sudah pasti dan sudah dikenal dengan baik. Hal ini didasarkan kepada berbagai penampilan situasi dan objek-objek khusus, dan proses yang tak bervariasi.

- ❖ Pengetahuan semacam itu kita sebut pengetahuan *stereotype*.

- ❖ Skema adalah satu metoda pengorganisasian, presentasi dan penggunaan pengetahuan stereotype agar komputer bisa menalar

## BINGKAI (FRAME)

- Dengan menggunakan representasi network, kita melihat pengetahuan diatur dengan menggunakan penghubung antar obyek dalam basis pengetahuan. Selain itu, kita dapat mengatur pengetahuan ke dalam unit-unit yang lebih kompleks yang menggambarkan situasi atau obyek yang rumit dalam domain. Unit-unit ini disebut *bingkai (frame)*.
- Menurut Minsky, bingkai dapat dipandang sebagai struktur data statik yang digunakan untuk merepresentasikan situasi-situasi yang telah dipahami dan stereotip
- Setiap bingkai individual dapat dipandang sebagai sebuah struktur data yang dalam banyak hal serupa dengan "record", dan berisi

informasi yang relevan dengan entitas-entitas stereotip.

- Bingkai mempermudah kita untuk mengatur pengetahuan kita secara **hirarki**.
- Adalah blok pengetahuan yang relatif besar atau kumpulan pengetahuan tentang suatu objek tertentu, peristiwa, lokasi, situasi atau elemen-elemen lainnya.
- Rinciannya diberikan ke dalam slot yang menggambarkan berbagai atribut dan karakteristik objek.
- Frame biasanya digunakan untuk merepresentasikan pengetahuan stereotip atau pengetahuan yang didasarkan kepada karakteristik yang sudah dikenal yang merupakan pengalaman-pengalaman.
- Dalam bentuk fisik, frame merupakan suatu gambaran seperti "garis besar" yang sudah dikategorikan dan sub kategori.

- Slot menggambarkan atribut seperti nama pabrik, model, asa-usul pabrik, jenis mobil, jumlah pintu, mesin dan karakteristik lainnya
- Beberapa slot mempunyai nilai tetap.
- Jenis slot lainnya bersifat prosedural. Hal ini merupakan slot yang memungkinkan penambahan informasi baru yang bisa ditambahkan pada kaidah dasar IF
- Hampir semua sistem artificial intelligence terbuat dari kumpulan frame-frame yang dalam hal ini satu sama lain saling berhubungan. Secara bersama-sama mereka (frame-frame) membentuk suatu hirarki yang dapat digunakan untuk maksud penalaran.
- Untuk menggunakan sistem frame, kita harus membuat program rrame itu sendiri dengan menggunakan bahasa pemograman Al.

(CATT : Contoh mekanik Mobil)

### ***SKRIP (SCRIPT)***

- *Skrip (script)* merupakan representasi terstruktur yang menggambarkan urutan stereotip dari kejadian-kejadian dalam sebuah konteks khusus.
- Skrip mula-mula dirancang oleh Schank dan kelompok risetnya sebagai alat pengorganisasi struktur-struktur *ketergantungan konseptual menjadi* deskripsi khusus.
- Script adalah skema representasi pengetahuan yang sama dengan frame.
- Perbedaannya ialah, frame menggambarkan objek, sedang script menggambarkan urutan peristiwa.
- Dalam menggambarkan urutan peristiwa, script menggunakan serangkaian slot yang berisi informasi tentang orang, objek, dan tindakan-tindakan yang teijadi dalam suatu peristiwa.
- Elemen script yang tipikal termasuk kondisi masukan, prop, role dan scene.

- Kondisi masukkan menggambarkan situasi yang harus dipenuhi sebelum terjadi atau berlaku suatu peristiwa yang ada dalam script.
- Prop mengacu kepada objek yang digunakan dalam urutan peristiwa yang terjadi.
- Role mengacu kepada orang-orang yang terlibat dalam script.
- Scene menggambarkan urutan peristiwa aktual yang terjadi.
- Komponen-komponen skrip adalah
  - ✓ *Kondisi entri* atau deskriptor dunia sekitar kita yang harus benar agar skrip dapat dipanggil. Contoh : dalam hal skrip restoran, ini mencakup restoran yang sedang buka dan pelanggan yang sedang lapar.
- ✓ *Hasil atau fakta yang benar* begitu skrip diakhiri. Misalnya, pelanggan sudah kenyang, dan pemilik restoran memiliki uang yang lebih

banyak (karena pembayaran oleh pelanggan tersebut).

- ✓ *Penyangga* atau apa-apa yang merupakan isi skrip. Di sini meliputi meja, kursi, pelayan, dan menu.
- ✓ *Peran* adalah tindakan yang dilakukan oleh partisipan individual. Misalnya, pelayan yang mengantar pesanan, dan memberikan tagihan pada pelanggan, serta pesanan pelanggan, makan, dan membayar.
- ✓ *Adegan* yang merupakan kejadian yang menunjukkan aspek waktu dari skrip. Di sini dapat berupa : masuk ke restoran, memesan, makan, dan lain-lain.



## Predicate Logic (Predicate Calculus)

Abjad yang menyusun lambang-lambang kalkulus predikatif terdiri dari :

1. Rangkaian huruf, baik huruf kecil maupun huruf besar dari abjad.
2. Rangkaian digit 0,1,...,9.
3. Garis bawah \_.

Lambang-lambang kalkulus predikatif *dimulai dengan huruf dan diikuti oleh sembarang rangkaian karakter yang diperkenankan*

Contoh :

- (1) Tono
- (2) \*\*\*98
- (3) ya2
- (4) ati\_dan\_tita
- (5) YYY
- (6) 9rumah
- (7) "mata"
- (8) yach????
- (9) &kita

Untuk lambang variabel : dimulai dg huruf besar untuk merancang kelas objek atau sifat yang umum

Contoh : Gumam, POHON

Untuk lambang fungsi/konstanta : dimulai dg huruf kecil

Contoh : gumam, pohon

Kalimat dasar dalam kalkulus predikatif adalah predikat yang diikuti dengan istilah yang berada didalam tanda kurung dan dipisahkan oleh koma. Kalimat kalkulus predikatif dibatasi oleh titik/periode.

Contoh : - *likes*(ani,ida)

- *helps*(anton,tono)

Lambang predikat dalam contoh di atas adalah *likes* dan *helps*. Lambang-lambang predikat merupakan lambang yang dimulai dg huruf kecil.

Kalkulus predikatif juga mencakup dua lambang,  $\forall$  dan  $\exists$  yang membatasi arti sebuah kalimat.  $\forall$  merupakan *penguantifikasi universal* yang menunjukkan bahwa suatu kalimat adalah benar *untuk semua nilai variabelnya*. Sedangkan  $\exists$  merupakan *penguantifikasi eksistensial* yang menunjukkan bahwa suatu kalimat adalah benar *untuk suatu nilai tertentu dalam sebuah domain*.

## QUANTIFIER UNIVERSAL ( $\forall$ )

- Menyatakan " untuk setiap" atau "untuk semua"
- Contoh : p menunjukkan kalimat seluruh kucing adalah binatang

$(\forall x) (p) \equiv (\forall x) (\text{if } x \text{ adalah seekor kucing} \rightarrow x \text{ adalah seekor binatang})$

atau

$(\forall x) (x \text{ is a cat} \rightarrow x \text{ is a animal})$

- Negasi :  $(\forall x) (p) \equiv (\exists x) (\text{if } x \text{ is a cat} \rightarrow \sim x \text{ is a animal})$
- Bagaimana kalimat matematika untuk "seluruh segitiga adalah poligon" ?
- Bagaimana Predicate Function ?

$(\forall x) (x \text{ is a triangle} \rightarrow x \text{ is a polygon})$

$(\forall x) (\text{triangle}(x) \rightarrow \text{polygon}(x))$

- Fungsi Predikat dituliskan dg notasi yg lebih singkat dg huruf besar
- Contoh : T = triangle dan P = Polygon  
 $(\forall x) (T(x) \rightarrow P(x))$
- Dapat diinterpretasikan sebagai konjungsi

## EXISTENTIAL QUANTIFIER ( $\exists$ )

- Suatu pernyataan benar untuk minimal satu anggota domain
- Dibaca "there exists", "at least one", "for some", "there is one", "some"

- Contoh :

P=gajah

Q=binatang mamalia

Semua gajah adalah mamalia  $(\forall x) (P(x) \rightarrow Q(x))$

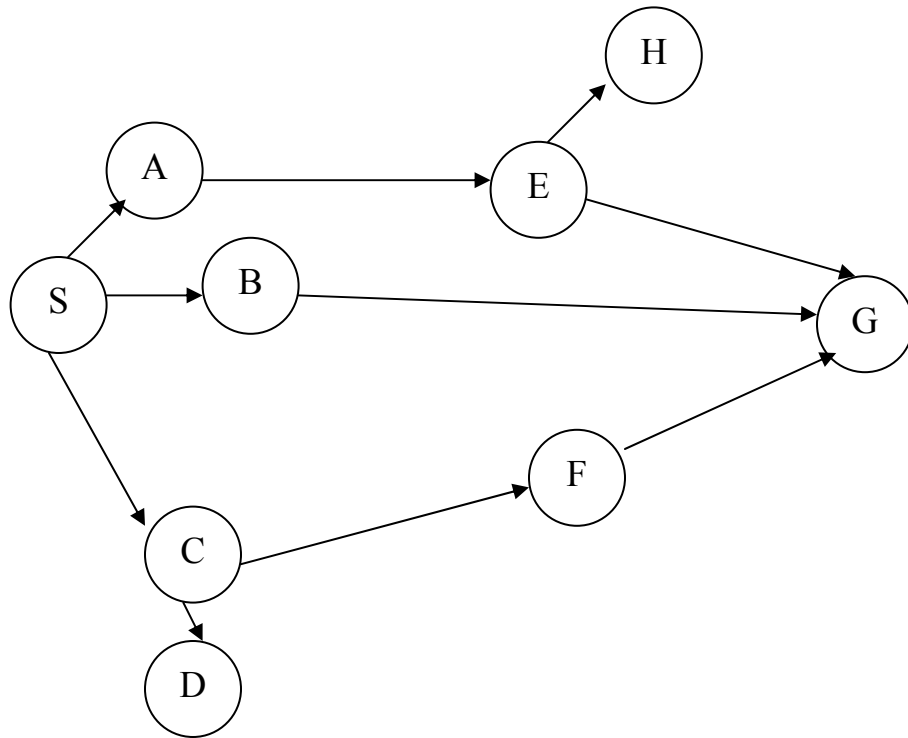
Beberapa gajah bukan mamalia  $(\exists x) (P(x) \rightarrow \sim Q(x))$

Beberapa gajah adalah mamalia  $(\exists x) (P(x) \rightarrow Q(x))$

Set Expression	Logical Equivalent
$A = B$	$\forall x (x \in A \leftrightarrow x \in B)$
$A \subseteq B$	$\forall x (x \in A \rightarrow x \in B)$
$A \cap B$	$\forall x (x \in A \wedge x \in B)$
$A \cup B$	$\forall x (x \in A \vee x \in B)$
$A'$	$\forall x (x \in U \mid \sim (x \in A))$
U (Universe)	T (True)
$\emptyset$ (empty set)	F (False)

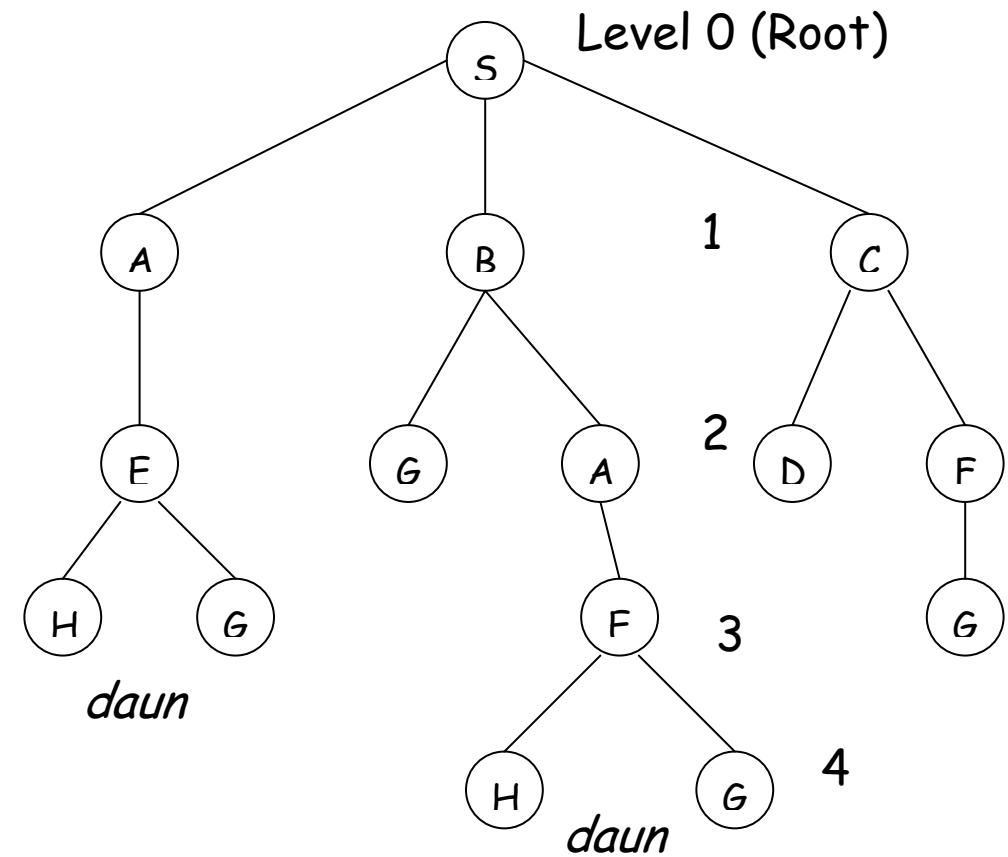
Set	Logic
$(A \cap B)' \equiv A' \cup B'$	$\sim(p \wedge q) \equiv \sim p \vee \sim q$
$(A \cup B)' \equiv A' \cap B'$	$\sim(p \vee q) \equiv \sim p \wedge \sim q$

## STATE GRAPH



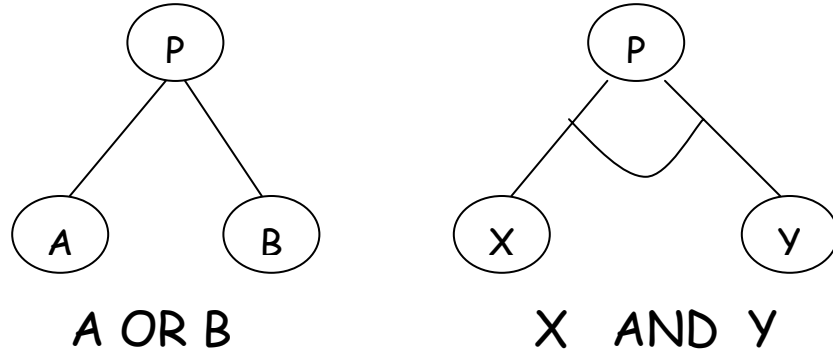
- Peta yg menunjukkan berbagai kota antara kota - kota yg akan dilaluinya agar mencapai kota tujuan yg diinginkan lebih cepat.
- A ... H : Node
- $\longrightarrow$  : Ark/link

## POHON PELACAKAN



- Level  $\rightarrow$  sbg hirarki (menggambarkan kedalaman pohon)
- Node merupakan berbagai keadaan dlm ruang pelacakan. Ark  $\rightarrow$  operatornya.

## POHON AND / OR



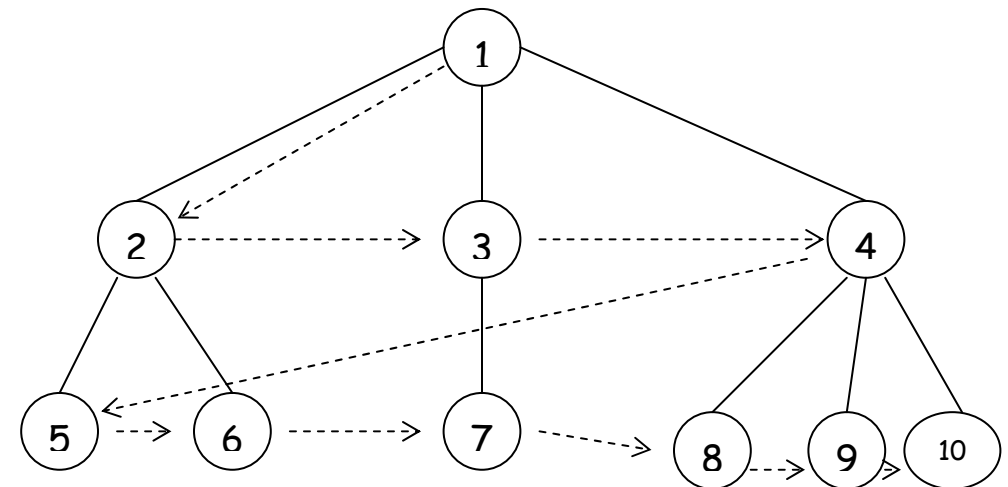
## METODE PELACAKAN

### 1. *BLIND SEARCH (Pelacakan Buta)*

- ✓ Merupakan sekumpulan prosedur yg digunakan dlm melacak ruang keadaan.
- ✓ Menguji seluruh pohon dgn cara yg teratur dg menggunakan semua operator shg menghasilkan suatu solusi.
- ✓ Lebih tepat u/ soal-soal kecil dg beberapa ruang keadaan dan tepat u/ komputer berkecepatan tinggi.

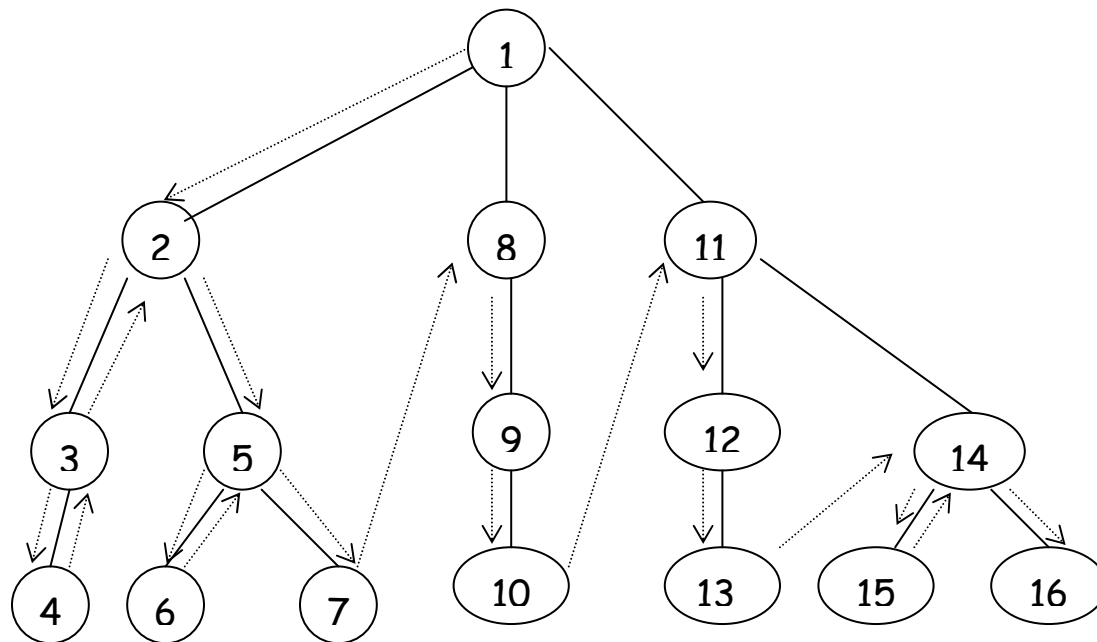
### 2. *BREADTH FIRST (Pelacakan Melebar Pertama)*

- ✓ Menguji semua node dlm pohon pelacakan mulai dari node akar
- ✓ Node yg ada pd setiap tingkat seluruhnya diuji sebelum pindah ke tingkat berikutnya.



### 3. DEPTH FIRST (Pelacakan Pertama Mendalam)

- ✓ Jika keadaan tujuan tidak tercapai maka proses dilakukan dg jalan pelacakan backtrak ke node sebelumnya.
- ✓ Menjamin bisa menemukan solusi *tapi waktu pelacakannya lama.*
- ✓ Masalah utama : sering terjadi penyimpangan arah node tujuan yg sebenarnya.



### 4. HEURISTIC SEARCH

- ✓ Istilah yg berasal dari bahasa Yunani yg berarti " menemukan / menyingkap "
- ✓ Membantu mengurangi wilayah pelacakan yg bisa menimbulkan berbagai alternatif solusi shg dapat membimbing ke tujuan yg diinginkan.

### 5. HILL CLIMBING (Mendaki bukit)

- ✓ Merupakan pelacakan depth first yg memanfaatkan heuristik u/ menentukan jarak yg terpendek atau biaya terendah menuju tujuan yg diinginkan.

### 6. BEST FIRST SEARCH (Pencarian Terbaik Pertama)

- ✓ Kombinasi dari breadth first dan depth first