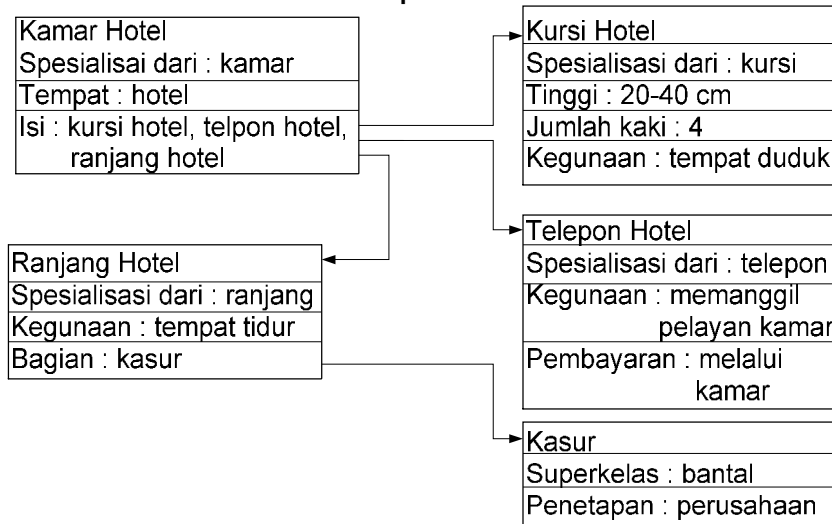


# REPRESENTASI PENGETAHUAN (2)

## 3. Frame

- Frame (Minsky, 1975) dipandang sebagai struktur data static yang digunakan untuk merepresentasikan situasi-situasi yang telah dipahami dan stereotype.
- Frame digunakan untuk merepresentasikan pengetahuan stereotype atau pengetahuan yang didasarkan kepada karakteristik yang sudah dikenal yang merupakan pengalaman masa lalu.
- Frame berupa kumpulan slot-slot (representasi entitas sebagai struktur objek) yang merupakan atribut untuk mendeskripsikan pengetahuan berupa kejadian, lokasi, situasi ataupun elemen-elemen lain. Frame digunakan untuk representasi pengetahuan deklaratif.
- Contoh 1 :  
Frame Pohon
  - Spesialisasi dari : Tumbuhan
  - Jumlah batang : integer (default 1)
  - Jenis kulit : halus
  - Model daun : jenis pohon jarum, berganti daun
  - Bentuk daun : sederhana, berlekuk, campuran
- Frame Pohon Perdu
  - Spesialisasi dari : Pohon
  - Jumlah batang : 3
  - Jenis kulit : halus
  - Model daun : berganti daun
  - Bentuk daun : sederhana, berlekuk

- Contoh 2 : Deskripsi frame untuk kamar hotel.

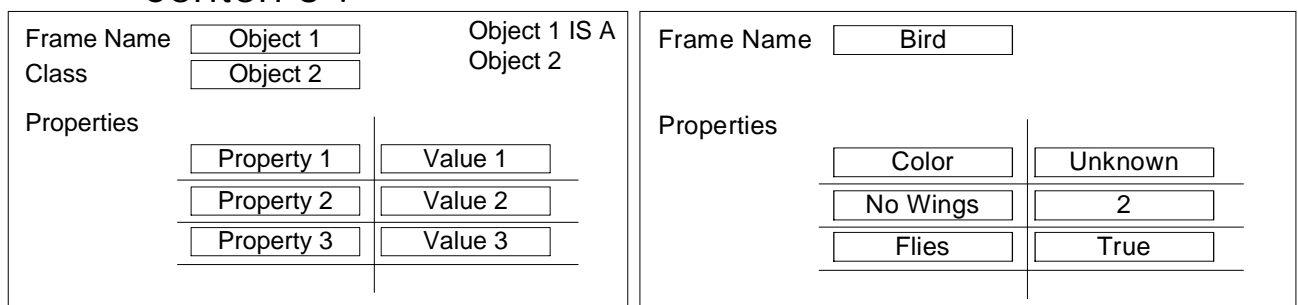


- Setiap frame individual dapat dipandang sebagai struktur data yang mirip record, berisi informasi yang relevan dengan entitas-entitas stereotipe.

Slot-slot dalam frame dapat berisi :

- Informasi identifikasi frame
- Hubungan frame dengan frame lain
- Penggambaran persyaratan yang dibutuhkan frame
- Informasi procedural untuk menggunakan struktur yang digambarkan
- Informasi default frame
- Informasi terbaru.

- Contoh 3 :



Dari contoh 3, terdapat dua elemen dasar, yaitu Slot dan Subslot. Slot merupakan kumpulan atribut / property yang menjelaskan objek yang direpresentasikan oleh frame. Subslot menjelaskan pengetahuan atau prosedur dari atribut pada slot.

Subslot dapat berupa :

- *Value* : menjelaskan tentang nilai dari suatu atribut
- *Default* : nilai yang digunakan jika suatu slot kosong atau tidak dideskripsikan pada frame instansiasi
- *Range* : menandakan jenis dari informasi yang dapat muncul pada slot tersebut (misal 0 sampai 100)
- *If Added* : berisi informasi procedural yang berupa suatu tindakan yang akan dikerjakan jika nilai dari slot diisi (atau berubah)
- *If Needed* : subslot ini digunakan pada kasus dimana tidak ada value pada slot. Suatu prosedur akan dikerjakan untuk memperoleh atau menghitung sebuah value.
- *Other* : slot bisa berisi frame, rule, jaringan semantic ataupun tipe lain dari informasi.

#### 4. Script

- Script (Schank & Abelson, Yale univ) merupakan representasi terstruktur yang menggambarkan urutan stereotip dari kejadian-kejadian dalam sebuah konteks khusus.

- Script mirip dengan frame, perbedaannya : Frame menggambarkan objek, sedangkan Script menggambarkan urutan peristiwa.
- Dalam menggambarkan urutan peristiwa, script menggunakan serangkaian slot yang berisi informasi tentang orang, objek dan tindakan-tindakan yang terjadi dalam suatu peristiwa.
- Elemen script yang tipikal :
  - ◆ Kondisi masukan : menggambarkan situasi yang harus dipenuhi sebelum terjadi 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.
  - ◆ Hasil : kondisi yang ada sesudah peristiwa dalam script berlangsung.
  - ◆ Track : mengacu kepada variasi yang mungkin terjadi dalam script tertentu.
  - ◆ Scene : menggambarkan urutan peristiwa aktual yang terjadi.
- Contoh : Script pergi ke restoran

#### SCRIPT Restoran

Jalur (track) : fast food restoran

Peran (roles) : tamu, pelayan

Pendukung (prop): conter, baki, makanan, uang, serbet, garam, merica, kecap, sedotan, dll

Kondisi masukan : tamu lapar –tamu punya uang

Adegan (scene) 1 : Masuk

- Tamu parkir mobil
- Tamu masuk restoran
- Tamu antri
- Tamu baca menu di list menu dan mengambil keputusan tentang apa yang akan diminta.

Adegan (scene) 2 : Pesanan

- Tamu memberikan pesanan pada pelayan
- Pelayan mengambil pesanan dan meletakkan makanan di atas baki
- Tamu membayar

Adegan (scene) 3 : Makan

- Tamu mengambil serbet, sedotan, garam, dll
- Tamu makan dengan cepat

Adegan (scene) 4 : Pulang

- Tamu membersihkan meja
- Tamu membuang sampah
- Tamu meninggalkan restoran
- Tamu naik mobil dan pulang

Hasil

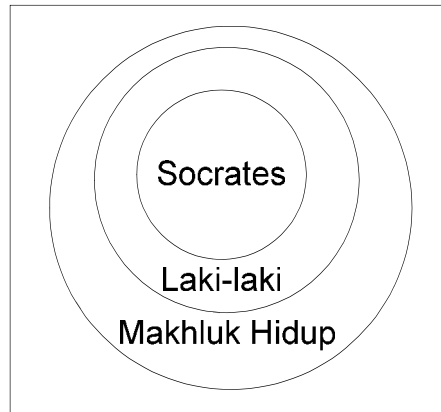
- Tamu merasa kenyang
- Tamu senang
- Tamu kecewa
- Tamu sakit perut

- Keistimewaan Script :

1. Script menyediakan beberapa cara yang sangat alami untuk merepresentasikan "suatu informasi" yang lazim" dengan masalah yang bersumber dari sistem AI dari mula.
2. Script menyediakan struktur hirarki untuk merepresentasikan informasi melalui inklusi subscript dengan sript.

## 5. Logika dan Himpunan

- Representasi pengetahuan dengan symbol logika merupakan bagian dari penalaran eksak.
- Bagian yang paling penting dalam penalaran adalah mengambil kesimpulan dari premis.
- Logika dikembangkan oleh filusuf Yunani, Aristoteles (abad ke 4 SM) didasarkan pada *silogisme*, dengan dua *premis* dan satu *konklusi*.  
Contoh :  
Premis : Semua laki-laki adalah makhluk hidup  
Premis : Socrates adalah laki-laki  
Konklusi : Socrates adalah makhluk hidup
- Cara lain merepresentasikan pengetahuan adalah dengan Diagram Venn.



- Diagram Venn merepresentasikan sebuah himpunan yang merupakan kumpulan objek.
- Objek dalam himpunan disebut elemen.

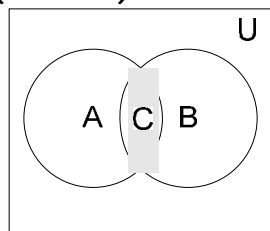
$$A = \{1, 3, 5, 7\}$$

$$B = \{\dots, -4, -2, 0, 2, 4, \dots\}$$

$$C = \{\text{pesawat, balon}\}$$

- Symbol epsilon  $\varepsilon$  menunjukkan bahwa suatu elemen merupakan anggota dari suatu himpunan, contoh :  $1 \varepsilon A$  . Jika suatu elemen bukan anggota dari suatu himpunan maka symbol yang digunakan  $\notin$ , contoh :  $2 \notin A$ .
- Jika suatu himpunan sembarang, misal X dan Y didefinisikan bahwa setiap elemen X merupakan elemen Y, maka X adalah subset dari Y, dituliskan :  $X \subset Y$  atau  $Y \supset X$ .
- Operasi-operasi Dasar dalam Diagram Venn:

- ◆ Interseksi (Irisan)



$$C = A \cap B$$

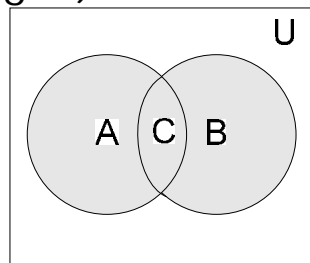
$$C = \{x \in U \mid (x \in A) \wedge (x \in B)\}$$

Dimana :  $\cap$  menyatakan irisan himpunan

| dibaca "sedemikian hingga"

$\wedge$  operator logika AND

- ◆ Union (Gabungan)



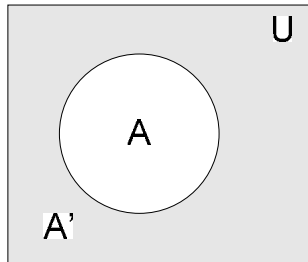
$$C = A \cup B$$

$$C = \{x \in U \mid (x \in A) \vee (x \in B)\}$$

Dimana :  $\cup$  menyatakan gabungan himpunan

$\vee$  operator logika OR

◆ Komplement



$$A' = \{x \in U \mid \sim(x \in A)\}$$

Dimana : ' menyatakan komplement himpunan  
 $\sim$  operator logika NOT

Logika Proposisi

- Disebut juga kalkulus proposisi yang merupakan logika simbolik untuk memanipulasi proposisi.
- Proposisi merupakan pernyataan yang dapat bernilai *benar* atau *salah*.
- Operator logika yang digunakan :

Operator	Fungsi
$\wedge$	Konjungsi (AND/DAN)
$\vee$	Disjungsi (OR/ATAU)
$\sim$	Negasi (NOT/TIDAK)
$\rightarrow$	Implikasi/Kondisional (IF...THEN.../ JIKA...MAKA ...)
$\leftrightarrow$	Equivalensi/Bikondisional (IF AND ONLY IF / JIKA DAN HANYA JIKA) $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

- Kondisional merupakan operator yang analog dengan *production rule*.

Contoh 1 :

“ Jika hujan turun sekarang maka saya tidak pergi ke pasar”

Kalimat di atas dapat ditulis :  $p \rightarrow q$

Dimana :  $p$  = hujan turun

$q$  = saya tidak pergi ke pasar



Contoh 2 :

$p$  = "Anda berusia 21 atau sudah tua"

$q$  = "Anda mempunyai hak pilih"

Kondisional  $p \rightarrow q$  dapat ditulis/berarti :

Kondisional	Berarti
$p \text{ implies } q$	Anda berusia 21 tahun atau sudah tua <i>implies</i> Anda mempunyai hak pilih.
Jika $p$ maka $q$	Jika Anda berusia 21 tahun atau sudah tua, maka Anda mempunyai hak pilih.
$p$ hanya jika $q$	Anda berusia 21 tahun atau sudah tua, hanya jika Anda mempunyai hak pilih.
$p$ adalah (syarat cukup untuk $q$ )	Anda berusia 21 tahun atau sudah tua adalah syarat cukup Anda mempunyai hak pilih.
$q$ jika $p$	Anda mempunyai hak pilih, jika Anda berusia 21 tahun atau sudah tua.
$q$ adalah (syarat perlu untuk $p$ )	Anda mempunyai hak pilih adalah syarat perlu Anda berusia 21 tahun atau sudah tua.

- Tautologi : pernyataan gabungan yang selalu bernilai *benar*.
- Kontradiksi : pernyataan gabungan yang selalu bernilai *salah*.
- Contingent : pernyataan yang bukan tautology ataupun kontradiksi.
- Tabel Kebenaran untuk logika konektif :

$p$	$q$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	F	T	F	F
F	T	F	T	T	F
F	F	F	F	T	T

- Tabel kebenaran untuk negasi konektif :

$p$	$\sim p$
T	F
F	T

## Logika Predikat

- Disebut juga kalkulus predikat, merupakan logika yang digunakan untuk merepresentasikan masalah yang tidak dapat direpresentasikan dengan menggunakan proposisi.
- Logika predikat dapat memberikan representasi fakat-fakta sebagai suatu pernyataan yang mapan (*well form*).
- Syarat-syarat symbol dalam logika predikat :
  - ◆ himpunan huruf, baik huruf kecil maupun huruf besar dalam abjad.
  - ◆ Himpunan digit (angka) 0,1,2,...9
  - ◆ Garis bawah “\_”
  - ◆ Symbol-simbol dalam logika predikat dimulai dengan sebuah huruf dan diikuti oleh sembarang rangkaian karakter-karakter yang diijinkan.
  - ◆ Symbol-simbol logika predikat dapat merepresentasikan variable, konstanta, fungsi atau predikat :

Konstanta : objek atau sifat dari semesta pembicaraan. Penulisannya diawali dengan *huruf kecil*, seperti : pohon, tinggi. Konstanta *true* (benar) dan *false* (salah) adalah symbol kebenaran (*truth symbol*).

Variable : digunakan untuk merancang kelas objek atau sifat-sifat secara umum dalam semesta pembicaraan. Penulisannya diawali dengan *huruf besar*, seperti : Bill, Kate.

Fungsi : pemetaan (mapping) dari satu atau lebih elemen dalam suatu himpunan yang disebut *domain* fungsi ke dalam sebuah elemen unik pada himpunan lain yang disebut *range* fungsi. Penulisannya dimulai dengan *huruf kecil*. Suatu ekspresi fungsi merupakan symbol fungsi yang diikuti argument. Argument adalah elemen-elemen dari fungsi, ditulis diapit tanda kurung dan dipisahkan dengan tanda koma.

Contoh : f(X,Y)  
ayah(david)  
plus(2,3)

Predikat : menamai hubungan antara nol atau lebih objek dalam semesta pembicaraan. Penulisannya dimulai dengan *huruf kecil*, seperti : equals, samadengan, likes, near.

Contoh kalimat dasar :

teman(george,allen)  
teman(ayah\_dari(david),ayah\_dari(andrew))

dimana :

argument : ayah\_dari(david) adalah george  
argument : ayah\_dari(andrew) adalah allen  
predikat : teman

- ◆ Operator logika konektif :  $\wedge, \vee, \sim, \rightarrow, \equiv$ .
- ◆ Logika kalkulus orde pertama mencakup symbol universal quantifier  $\forall$  dan existensial quantifier  $\exists$ .

## *Universal Quantifier*

- Menunjukkan semua kalimat adalah benar untuk semua nilai variabelnya.
- Direpresentasikan dengan symbol  $\forall$  diikuti satu atau lebih argument untuk suatu domain variable.
- Symbol  $\forall$  diinterpretasikan "untuk setiap" atau "untuk semua".

- Contoh 1 :

$$(\forall x) (x + x = 2x)$$

"untuk setiap x (dimana x adalah suatu bilangan), kalimat  $x + x = 2x$  adalah benar."

Contoh 2 :

$$(\forall x) (p) \quad (\text{Jika } x \text{ adalah seekor kucing} \rightarrow x \text{ adalah binatang})$$

Kebalikan kalimat "bukan kucing adalah binatang" ditulis :

$$(\forall x) (p) \quad (\text{Jika } x \text{ adalah seekor kucing} \rightarrow \sim x \text{ adalah binatang})$$

dan dibaca : - "setiap kucing adalah bukan binatang"  
- "semua kucing adalah bukan binatang"

Contoh 3:

$$(\forall x) \quad (\text{Jika } x \text{ adalah segitiga} \rightarrow x \text{ adalah polygon})$$

Dibaca : "untuk semua x, jika x adalah segitiga, maka x adalah polygon"

dapat pula ditulis :  $(\forall x) (\text{segitiga}(x) \rightarrow \text{polygon}(x))$   
 $(\forall x) (T(x) \rightarrow P(x))$

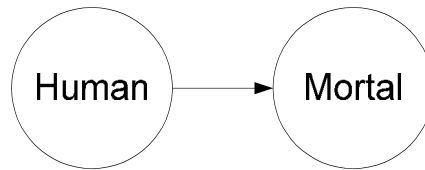
Contoh 4 :

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

Dibaca : "untuk semua x, jika x adalah manusia (human) , maka x melahirkan (mortal)".

Ditulis dalam aturan : IF x adalah manusia  
THEN x melahirkan

Digambar dalam jaringan semantic :



### *Exixtensial Quantifier*

- Menunjukkan semua kalimat adalah benar untuk suatu nilai tertentu dalam sebuah domain.
- Direpresentasikan dengan symbol  $\exists$  diikuti satu atau lebih argument.
- Symbol  $\exists$  diinterpretasikan "terdapat" atau "ada", "paling sedikit satu", "terdapat satu", "beberapa".
- Contoh 1 :

$$(\exists x) (x \cdot x = 1)$$

Dibaca : "terdapat x yang bila dikalikan dengan dirinya sendiri hasilnya sama dengan 1."

Contoh 2 :

$$(\exists x) (\text{gajah}(x) \wedge \text{nama}(\text{Clyde}))$$

Dibaca : "beberapa gajah bernama Clyde".

Contoh 3 :

$$(\forall x) (\text{gajah}(x) \rightarrow \text{berkaki empat}(x))$$

Dibaca : "semua gajah berkaki empat".

Universal quantifier dapat diekspresikan sebagai konjungsi.

$$(\exists x) (\text{gajah}(x) \wedge \text{berkaki tiga}(x))$$

Dibaca : "ada gajah yang berkaki tiga"

Existensial quantifier dapat diekspresikan sebagai disjungsi dari urutan  $a_i$ .  $P(a_1) \vee P(a_2) \vee P(a_3) \dots \vee P(a_N)$

## Quantifier dan Sets

Set Expression	Logical Equivalent
$A = B$	$\forall x (x \in A \leftrightarrow x \in B)$
$A \subseteq B$	$\forall x (x \in A \rightarrow x \in B)$
$A \cap B$	$\forall x (x \in A \wedge x \in B)$
$A \cup B$	$\forall x (x \in A \vee x \in B)$
$\mu$ (universe)	T (True)
$\phi$ (empty set)	F (False)

- Relasi A proper subset dari B ditulis  $A \subset B$ , dibaca "semua elemen A ada pada B", dan "paling sedikit satu elemen B bukan bagian dari A"

- Contoh :

Diketahui : E = elephant

R = reptile

G = gray

F = four legged

D = dogs

M = mammals

Set expression	Berarti
$E \subset M$	"elephant termasuk mammals", tetapi tidak semua mammals adalah elephant
$(E \cap G \cap F) \subset M$	"elephant yang berwarna gray dan memiliki four legged termasuk mammals"
$E \cap R = \phi$	"tidak ada gajah yang termasuk reptile"
$E \cap G \neq \phi$	"beberapa gajah berwarna gray"
$E \cap G = \phi$	"tidak ada gajah yang berwarna gray"

$E \cap G' \neq \phi$	"beberapa elephants tidak berwarna gray"
$E \subset (G \cap F)$	"semua elephants berwarna gray dan memiliki four legged"
$(E \cup D) \subset M$	"semua elephants dan dogs termasuk mammals"
$(E \cap F \cap G) \neq \phi$	"beberapa elephants memiliki four legged dan berwarna gray"

Hukum de Morgan berlaku untuk analogi himpunan dan bentuk logika :

Himpunan	Logika
$(A \cap B) \equiv A' \cup B'$	$\sim(p \wedge q) \equiv p \vee \sim q$
$(A \cup B) \equiv A' \cap B'$	$\sim(p \vee q) \equiv p \wedge \sim q$