

First Order Logic

Pokok Bahasan

1. Mengapa FOL (First Order Logic)
2. Konsep dasar FOL
3. Sintak dan semantic FOL
4. Konteks penggunaan FOL
5. Rekayasa Pengetahuan dengan FOL
6. Latihan Individu + Tugas Kelompok

Logika Propositional

Logika propositional adalah **declarative**

Logika propositional mengijinkan informasi partial/disjunctive/negated

- (tidak seperti struktur data dan database)

Logika propositional merupakan **compositional**:

- meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$

Arti pada logika propositional **context-independent**

- (tidak seperti natural language, dimana arti tergantung context)

Logika propositional mempunyai kekuatan ekspresif terbatas

- (tidak seperti natural language)

Konsep dasar FOL

- ❑ **Declarative** : menyatakan fakta-fakta terpisah dari mekanisme/prosedur inference.
- ❑ Memungkinkan pernyataan informasi yang **partial / disjunctive / negated**.
- ❑ **Compositional** : “arti” $P \wedge Q$ tergantung arti P dan arti Q.
- ❑ **Context-independent** : arti tidak tergantung konteks.
- ❑ **Unambiguous** : terhadap suatu model, arti sebuah sentence jelas.
- ❑ ...Sayangnya, kurang **expressive**.
Mis.: “Kalau ada jebakan, di kamar sebelah ada hembusan angin” harus dinyatakan dengan $n \times n$ buah sentence propositional logic.

- ❑ **Propositional logic** mengasumsikan bahwa dunia hanya mengandung fakta-fakta.
- ❑ **First order logic (FOL)**, mengasumsikan bahwa dunia bisa mengandung:
 - **Object** : di dalam dunia ada orang, bangunan, buku, UB, ITS, UI, SBY, bilangan, warna, hari, . . .
 - **Relations** : tentang object dalam dunia, ada relasi merah, bulat, cantik, positif, abang dari, lebih besar dari, di atas, terjadi sebelum, . . .
 - **Functions**: fungsi yang menghasilkan object lain seperti ayah dari, babak final dari, satu lebih dari, kaki kiri dari, . . .
- ❑ Hal ini disebut **ontological commitment** dari sebuah logic : apa saja “isi” dunia yang dijelaskan?

- ❑ Ada juga **epistemological commitment**: “kebenaran” apa yang dapat dinyatakan tentang sebuah sentence?
- ❑ Contoh beberapa jenis logic lain :

| Language | Ontological (Isi) | Epistemological (Kebenaran) |
|---------------------|----------------------------------|-------------------------------|
| Propositional logic | facts | true/false/unknown |
| First-order logic | facts, objects, relations | true/false/unknown |
| Temporal logic | facts, objects, relations, times | true/false/unknown |
| Probability theory | facts | degree of belief $\in [0, 1]$ |
| Fuzzy logic | degree of truth $\in [0, 1]$ | known interval value |
| | | |

□ Elemen-elemen dasar FOL:

- **Constants** : KingJohn, 2, UB, ITS, UI, Malang, Depok , . . .
- **Predicates** : Brother , >, Loves, Membenci , Mengajar , . . .
- **Functions** : Sqrt , LeftLegOf , Ayah, . . .
- **Variables** : x , y , a, b, . . .
- **Connectives** : $\wedge \vee \neg \Rightarrow \Leftrightarrow$
- **Equality** : =
- **Quantifiers** : $\forall \exists$

□ Kalimat atomic

- **Definisi atomic sentence :**
predicate(term₁, . . . , term_n)
atau **term**₁ = **term**₂
- **Definisi term :**
function(term₁, . . . , term_n)
atau **constant**
atau **variable**
- Contoh :
 - **Brother** (KingJohn, RichardTheLionheart)
 - **>** (Length(LeftLegOf (Richard)), Length(LeftLegOf (KingJohn)))

❑ Kalimat kompleks

- Kalimat kompleks complex sentence terdiri dari sentence yang digabungkan dengan connective.

- **Definisi complex sentence :**

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$$

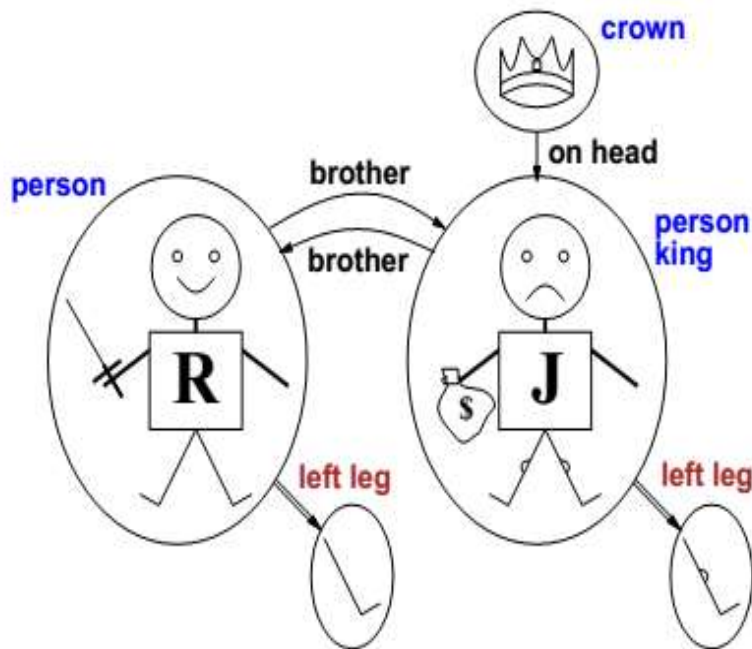
- Contoh :

- $\text{Sibling}(\text{KingJohn}, \text{Richard}) \Rightarrow \text{Sibling}(\text{Richard}, \text{KingJohn})$
- $>(1, 2) \vee \leq(1, 2)$
- $>(1, 2) \wedge \neg >(1, 2)$
- $\text{Belajar}(x, \text{SC}) \Rightarrow \text{Mengerti}(x, \text{AI})$

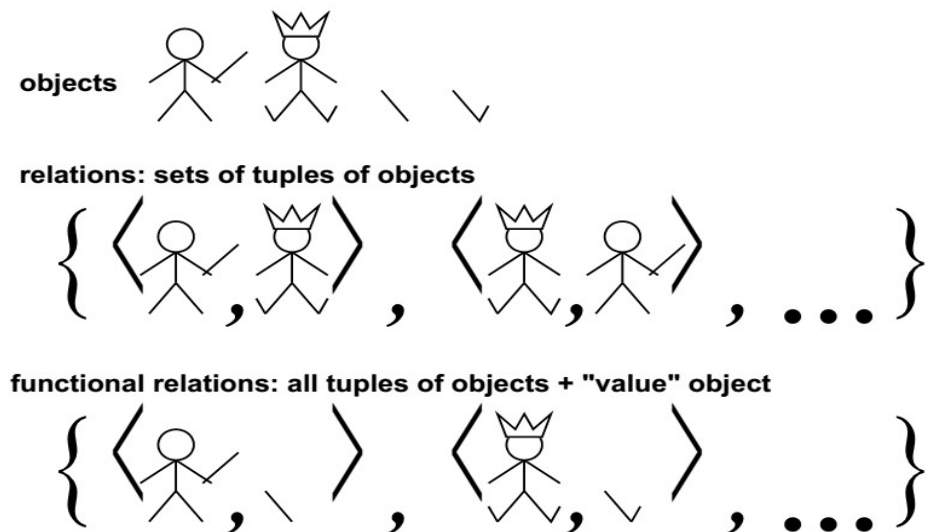
- ❑ Sama halnya dengan. Proposisi Logic (PL), sebuah kalimat FOL bisa juga dikatakan true terhadap sebuah **model**.
- ❑ Namun, sebuah kalimat bisa **diinterpretasikan** banyak cara dalam sebuah model.
- ❑ Model berisi :
 - **Objects** : elemen-elemen di dalam dunia (**domain elements**).
 - **Relations** : hubungan antara elemen-elemen tersebut.
- ❑ Sebuah **interpretasi** mendefinisikan **referent** (“yang dipetakan”)
 - **Constant symbols** → objects
 - **Predicate symbols** → relations
 - **Function symbols** → functional relations

- ❑ **Arti** dari sebuah kalimat FOL :
Kalimat atomik **predicate(term₁, . . . , term_n)** dikatakan bernilai true dalam model m di bawah interpretasi i iff **object** yang di-refer (**term₁, . . . , term_n**) (di bawah i) terhubung oleh **relation** yang di-refer oleh **predicate** (di bawah i) dalam m.

- ❑ Contoh sebuah model:



- ❑ Contoh sebuah model (lebih rinci):



Tanda \forall disebut Universal quantification yang berarti “untuk semua”

- $\forall <variables> <sentence>$

Everyone at NUS is smart:

$$\forall x \text{ At}(x, \text{NUS}) \Rightarrow \text{Smart}(x)$$

- $\forall x P$ bernilai true di sebuah model m iff P bernilai benar dengan x di setiap obyek pada model
- Equivalensi **conjunction** pada **instantiations** pada P

$$\begin{aligned} & \text{At}(\text{KingJohn}, \text{NUS}) \Rightarrow \text{Smart}(\text{KingJohn}) \\ \wedge & \text{At}(\text{Richard}, \text{NUS}) \Rightarrow \text{Smart}(\text{Richard}) \\ \wedge & \text{At}(\text{NUS}, \text{NUS}) \Rightarrow \text{Smart}(\text{NUS}) \end{aligned}$$

Kesalahan umum yang dihindari

\Rightarrow adalah konektivitas utama dengan \forall

Kesalahan umum: menggunakan \wedge sebagai konektivitas utama dengan \forall :

$$\forall x \text{ At}(x, \text{NUS}) \wedge \text{Smart}(x)$$

artinya "Everyone is at NUS and everyone is smart"

Existential Quantification (\exists) : memperlihatkan bahwa hanya beberapa (minimal satu) yang punya properti tertentu atau “tidak semua”

$\exists <variables> <sentence>$

Someone at NUS is smart:

$\exists x \text{ At}(x, \text{NUS}) \wedge \text{Smart}(x)$

$\exists x P$ bernilai benar pada sebuah model m iff P adalah benar dengan x di beberapa obyek pada model

Equivalensi **disjunction** pada **instantiations** pada P

$\text{At}(\text{KingJohn}, \text{NUS}) \wedge \text{Smart}(\text{KingJohn})$

$\vee \text{At}(\text{Richard}, \text{NUS}) \wedge \text{Smart}(\text{Richard})$

$\vee \text{At}(\text{NUS}, \text{NUS}) \wedge \text{Smart}(\text{NUS})$

Kesalahan umum yang dihindari

\wedge adalah konektivitas utama dengan \exists

Kesalahan umum: menggunakan \Rightarrow sebagai konektivitas utama dengan \exists :

$$\exists x \text{ At}(x, \text{NUS}) \Rightarrow \text{Smart}(x)$$

bernilai benar jika **ada seseorang yang tidak di NUS!**

Properti pada quantifiers

$\forall x \forall y$ is the same as $\forall y \forall x$

$\exists x \exists y$ is the same as $\exists y \exists x$

$\exists x \forall y$ is **not** the same as $\forall y \exists x$

$\exists x \forall y \text{ Loves}(x,y)$

- “There is a person who loves everyone in the world”

$\forall y \exists x \text{ Loves}(x,y)$

- “Everyone in the world is loved by at least one person”

Quantifier duality:

$\forall x \text{ Likes}(x, \text{IceCream})$

$\neg \exists x \neg \text{Likes}(x, \text{IceCream})$

$\exists x \text{ Likes}(x, \text{Broccoli})$

$\neg \forall x \neg \text{Likes}(x, \text{Broccoli})$

Equality

$term_1 = term_2$ bernilai benar dalam interpretasi jika dan hanya jika $term_1$ dan $term_2$ merefer ke obyek yang sama

Misal: pendefinisian *Sibling* pada term pada *Parent*:

$$\forall x, y \text{ Sibling}(x, y) \Leftrightarrow [\neg(x = y) \wedge \exists m, f \neg (m = f) \wedge \text{Parent}(m, x) \wedge \text{Parent}(f, x) \wedge \text{Parent}(m, y) \wedge \text{Parent}(f, y)]$$

Penggunaan FOL

- Kalimat “Saya suka coklat” dapat diekspan menjadi “ semua orang suka coklat”, berbentuk:

$$\forall x \text{ Orang}(x) \Rightarrow \text{Suka}(x, \text{Cokelat})$$

- \forall berarti “untuk semua”, disebut *universal quantifier*.
- Juga terdapat *existential quantifier* yang memperlihatkan bahwa hanya beberapa (minimal satu) yang punya properti tertentu, tidak semua:

$$\exists x \text{ Suka}(x, \text{Cokelat})$$

- Dapat dibaca “ada suatu x sedemikian hingga x suka coklat”.
- Catatan:

$$\forall x \text{ Suka}(x, \text{cokelat}) \Rightarrow \exists x \text{ Suka}(x, \text{cokelat}) \text{ adalah benar;}$$

$$\exists x \text{ Suka}(x, \text{cokelat}) \Rightarrow \forall x \text{ Suka}(x, \text{cokelat}) \text{ adalah salah.}$$

Inferensi pada FOL

- *Universal Elimination:*

$\forall v \alpha \Rightarrow \text{SUBSTS}(\{v/g\}, \alpha)$

$\forall x \text{Suka}(x, \text{Cokelat})$, x dapat digantikan oleh Saya, sehingga dapat disimpulkan $\text{Suka}(\text{Saya}, \text{Cokelat})$

- *Existential Elimination:*

$\exists v \alpha \Rightarrow \text{SUBSTS}(\{v/k\}, \alpha)$

$\exists x \text{Suka}(x, \text{Cokelat})$, x dapat digantikan oleh Saya, sehingga dapat disimpulkan $\text{Suka}(\text{saya}, \text{Cokelat})$

- *Existential Introduction:*

$\alpha \Rightarrow \exists v \text{SUBSTS}(\{g/v\}, \alpha)$

Dari $\text{Suka}(\text{saya}, \text{Cokelat})$ dapat disimpulkan $\exists x \text{Suka}(x, \text{Cokelat})$.

Contoh: Hukum Pernikahan

- Pernikahan tidak sah jika kedua mempelai mempunyai hubungan keponakan.
- Wati menikah dengan Andi
- Wati anak kandung Budi
- Budi Saudara Kembar Andi
- Buktikan bahwa pernikahan Andi dan Wati tidak sah!
- Langkah-langkah:
 - Buat kalimat-kalimat dalam *First Order (Predicate) Logic*, berdasarkan pengetahuan awal yang diberikan
 - Gunakan aturan inferensi untuk membuat kalimat baru sampai diperoleh kesimpulan.

1: Membangun Kalimat Awal

- 1: $\forall x,y \text{ Keponakan}(x,y) \wedge \text{Menikah}(y,x)$
 $\Rightarrow \neg \text{Sah}(\text{Menikah}(y,x))$
- 2: $\text{Menikah}(\text{Andi}, \text{Wati})$
- 3: $\text{AnakKandung}(\text{Wati}, \text{Budi})$
- 4: $\text{SaudaraKembar}(\text{Budi}, \text{Andi})$
- 5: $\forall x,y \text{ SaudaraKembar}(x,y) \Rightarrow \text{SaudaraKandung}(y,x)$
- 6: $\forall x,y,z \text{ AnakKandung}(x,y) \wedge \text{SaudaraKandung}(y,z)$
 $\Rightarrow \text{Keponakan}(x,z)$

2: Proses Inferensi

- (5) dan *Universal Elimination*:
7: SaudaraKembar(Budi, Andi) \Rightarrow SaudaraKandung (Budi, Andi)
- (4) , (7) dan Modus Ponens:
8: SaudaraKandung(Budi, Andi)
- (6) dan *Universal Elimination*:
9: AnakKandung(Wati, Budi) \wedge SaudaraKandung(Budi, Andi)
 \Rightarrow Keponakan(wati, Andi)
- (3), (8) dan Pemunculan AND :
10: AnakKandung(Wati, Budi) \wedge SaudaraKandung(Budi, Andi)

2: Proses Inferensi

- (9), (10) dan Modus Ponens:
11: Keponakan(Wati, Andi)
- (1) dan Universal Elimination:
12: Keponakan(Wati, Andi) \wedge Menikah (Andi,Wati)
 $\Rightarrow \neg$ Sah(Menikah(Andi,Wati))
- (11), (2), Pemunculan AND:
13: Keponakan(Wati, Andi) \wedge Menikah (Andi, Wati)
- (12), (13) dan Modus Ponens:
14: \neg Sah(Menikah(Andi, Wati))

Latihan Individu

❑ Ubahlah “Kalimat” dibawah ini menjadi bentuk “FOL” !

- “Ayah adalah orangtua berjenis kelamin laki-laki”.
- “Paman adalah saudara orangtua”.
- “Tidak ada jamur merah yang beracun”.

(Kerjakan 2 dari 3 pilihan yang ada)

❑ Ubahlah “FOL” dibawah ini menjadi bentuk “Kalimat” !

- $\forall x \text{ mahasiswa}(x, \text{IFUNSRI}) \Rightarrow \text{Genius}(x)$
- $\forall x (\text{jamur}(x) \wedge \text{merah}(x)) \Rightarrow \text{beracun}(x)$

(Kerjakan 1 dari 2 pilihan yang ada)

Tugas Kelompok

- ❑ Jelaskan perbedaan antara FOL dan PL?
- ❑ Ubahlah “Kalimat” dibawah ini menjadi bentuk “FOL” !
 - “Cucu adalah anak dari anak saya”.
 - “Paman dan Bibi adalah saudara”.
 - “Ada dua jamur merah”.
 - “Pohon kelapa itu tinggi”.
- ❑ Ubahlah “FOL” dibawah ini menjadi bentuk “Kalimat” !
 - $\exists x \forall t (\text{person}(x) \wedge \text{time}(t)) \Rightarrow \text{can-fool}(x,t)$
 - $(\forall x)(\forall y) \text{above}(x,y) \Leftrightarrow (\text{on}(x,y) \vee (\exists z) (\text{on}(x,z) \wedge \text{above}(z,y)))$
- ❑ Buatlah 1 penggalan puisi bebas yang terdiri minimal 4 baris, kemudian ubahlah dalam bentuk “FOL”.

Universal quantification

- ❑ **Syntax:**
Jika **S** kalimat, \forall **variables S** adalah kalimat
- ❑ **Contoh:**
 - “Semua mahasiswa IFUNSRI adalah Genius”
 - $\forall x \text{ mahasiswa}(x, \text{IFUNSRI}) \Rightarrow \text{Genius}(x)$
- ❑ **Semantics:**
 $\forall x S$ bernilai true dalam model m di bawah interpretasi iff S bernilai true untuk **semua** kemungkinan referent dari x (setiap object di dalam m).
Dengan kata lain, $\forall x S \equiv$ **conjunction** dari semua **instantiation S**:
 $(\text{mahasiswa}(\text{Ani}, \text{IFUNSRI}) \Rightarrow \text{Genius}(\text{Ani})) \wedge$
 $(\text{mahasiswa}(\text{Anto}, \text{IFUNSRI}) \Rightarrow \text{Genius}(\text{Anto})) \wedge$
 .
 .
 .
 $(\text{mahasiswa}(\text{Zaenal}, \text{IFUNSRI}) \Rightarrow \text{Genius}(\text{Zaenal})) \wedge$
 $(\text{mahasiswa}(\text{Zakky}, \text{IFUNSRI}) \Rightarrow \text{Genius}(\text{Zakky}))$

Universal quantification

- ❑ Biasanya, \Rightarrow adalah operator /connective yang digunakan dengan \forall .
- ❑ Masalah yang sering terjadi : menggunakan \wedge sebagai connective untuk \forall :
 $\forall x \text{ mahasiswa}(x, \text{IFUNSRI}) \wedge \text{Genius}(x)$
- ❑ Kalimat ini berarti “Semua orang adalah mahasiswa IFUNSRI dan Genius”.

Existential quantification

❑ Syntax :

Jika **S** kalimat, \exists **variable S** adalah kalimat

❑ Contoh:

- “Ada mahasiswa IFUNSRI yang pintar”
- $\exists x \text{ mahasiswa}(x, \text{IFUNSRI}) \wedge \text{pintar}(x)$

❑ Semantics :

$\exists x S$ bernilai true dalam model m di bawah interpretasi iff S bernilai true untuk **setidaknya** 1 kemungkinan referent dari x (sebuah object di dalam m).

Dengan kata lain, $\exists x S \equiv$ **disjunction** dari semua **instantiation** S :

$(\text{mahasiswa}(\text{Ani}, \text{IFUNSRI}) \wedge \text{pintar}(\text{Ani})) \vee$

$(\text{mahasiswa}(\text{Anto}, \text{IFUNSRI}) \wedge \text{pintar}(\text{Anto})) \vee$

.

.

$(\text{mahasiswa}(\text{Zaenal}, \text{IFUNSRI}) \wedge \text{pintar}(\text{Zaenal})) \vee$

$(\text{mahasiswa}(\text{Zakky}, \text{IFUNSRI}) \wedge \text{pintar}(\text{Zakky}))$

Existential quantification

- ❑ Biasanya, \wedge adalah operator /connective yang digunakan dengan \exists .
- ❑ Masalah yang sering terjadi : menggunakan \Rightarrow sebagai connective untuk \exists :
 $\exists x \text{ mahasiswa}(x, \text{IFUNSRI}) \Rightarrow \text{pintar}(x)$
- ❑ Kalimat ini true jika ada setidaknya 1 orang (object) yang tidak kuliah di IFUNSRI!

Beberapa sifat \forall (For All) dan \exists (There Exist)

- ❑ $\forall x \forall y S$ sama dengan $\forall y \forall x S$, biasa ditulis $\forall x, y S$
- ❑ $\exists x \exists y S$ sama dengan $\exists y \exists x S$, biasa ditulis $\exists x, y S$
- ❑ $\exists x \forall y S$ TIDAK sama dengan $\forall y \exists x S$!
 - $\exists x \forall y \text{ Mencintai } (x, y)$
“Ada (sekurang-kurangnya) seseorang yang mencintai semua orang di dunia.”
 - $\forall y \exists x \text{ Mencintai } (y, x)$
“Semua orang di dunia mencintai sekurang-kurangnya satu orang”.
- ❑ Quantifier bisa dinyatakan dengan yang lain:
 - $\forall x \text{ Doyan}(x, \text{Bakso})$ sama dengan $\neg \exists x \neg \text{Doyan}(x, \text{Bakso})$
 - $\exists x \text{ Doyan}(x, \text{Cilok})$ sama dengan $\neg \forall x \neg \text{Doyan}(x, \text{Cilok})$

“Contoh kalimat” Convert to “FOL”

- ❑ “Ayah adalah orangtua”
 $\forall x, y \text{ Ayah}(x, y) \Rightarrow \text{Orangtua}(x, y)$
- ❑ “Hubungan saudara berlaku simetris”
 $\forall x, y \text{ Saudara}(x, y) \Leftrightarrow \text{Saudara}(y, x)$
- ❑ “Ibu adalah orangtua berjenis kelamin perempuan”
 $\forall x, y \text{ Ibu}(x, y) \Leftrightarrow \text{Orangtua}(x, y) \wedge \text{Perempuan}(x)$
- ❑ “Sepupu adalah anak dari saudara orangtua”
 $\forall x, y \text{ Sepupu}(x, y) \Leftrightarrow \exists ox, oy \text{ Orangtua}(ox, x) \wedge \text{Saudara}(ox, oy) \wedge \text{Orangtua}(oy, y)$

Equality

- ❑ Kalimat **$\text{term}_1 = \text{term}_2$** bernilai true di bawah sebuah interpretasi iff term_1 and term_2 me-refer ke object yang sama.
- ❑ Contoh:
 - **$\text{Ayah}(\text{Anto}) = \text{Abdul}$** adalah satisfiable
 - **$\text{Anto} = \text{Abdul}$** juga satisfiable!
 - **$\text{Anto} = \text{Anto}$** adalah valid.
- ❑ Bisa digunakan dengan negasi untuk membedakan dua term:
 $\exists x, y \text{ Mencintai}(\text{Anto}, x) \wedge \text{Mencintai}(\text{Anto}, y) \wedge \neg(x = y)$
(Anto mendua!)
- ❑ Definisi Sibling:
 $\forall x, y \text{ Sibling}(x, y) \Leftrightarrow (\neg(x = y) \wedge \exists m, f \neg(m = f) \wedge \text{Parent}(m, x) \wedge \text{Parent}(f, x) \wedge \text{Parent}(m, y) \wedge \text{Parent}(f, y))$

Knowledge-based Agent (KBA) dengan FOL

- ❑ Kita bisa menggunakan FOL sebagai KRL (**Knowledge Representation Language**) sebuah KBA.
- ❑ Pertama-tama, kita berikan informasi ke KB (TELL).
- ❑ Kalimat FOL yang ditambahkan ke KB disebut assertion. Contohnya :
 - **TELL(KB,King(John))**
 - **TELL(KB, $\forall x \text{ King}(x) \Rightarrow \text{Person}(x)$)**
- ❑ Lalu, kita bisa memberikan query, atau bertanya, kepada KB (ASK). Contohnya :
 - **ASK(KB,King(John))** jawabannya adalah true.
 - **ASK(KB,Person(John))** jawabannya adalah true.
 - **ASK(KB, $\exists x \text{ Person}(x)$)** jawabannya adalah {x /John}

Substitution

- ❑ Sebuah query dengan **existential variable** bertanya kepada KB: “Apakah ada x sedemikian sehingga . . . ?”
- ❑ Bisa saja jawabannya “ya” atau “tidak”, tetapi akan lebih baik jika jawabannya adalah **nilai (referent) x di mana query bernilai true**.
- ❑ Bentuk jawaban demikian disebut **substitution**, atau binding list: himpunan pasangan variable/term.
- ❑ Untuk kalimat S dan substitution σ , $S\sigma$ adalah hasil “pengisian” S dengan σ :
 - **$S = \text{LebihPintar}(x, y)$**
 - **$\sigma = \{x / \text{Ani}, y / \text{Anto}\}$**
 - **$S\sigma = \text{LebihPintar}(\text{Ani}, \text{Anto})$**
- ❑ $\text{ASK}(\text{KB}, S)$ mengembalikan (satu? semua?) σ sedemikian sehingga $\text{KB} \models S\sigma$.

Knowledge Engineering

- ❑ **Diagnostic** vs. **Causal (model-based)** reasoning penting, mis: diagnosa medis secara AI (dulu diagnostic, sekarang model-based)
- ❑ Proses merancang kalimat-kalimat KRL yang dengan tepat “merepresentasikan” sifat dunia/masalah disebut **knowledge engineering**.
- ❑ “Memrogram” secara deklaratif : pengkodean fakta dan aturan **domain-specific**.
Sedikit jargon :
 - **Agent programmer = knowledge engineer**
- ❑ Mekanisme/proses penjawaban query → inference rule yang **domain-independent**.