BAHASA PEMROGRAMAN

TIM PENGAJAR PEMROGRAMAN
Departemen Ilmu Komputer IPB

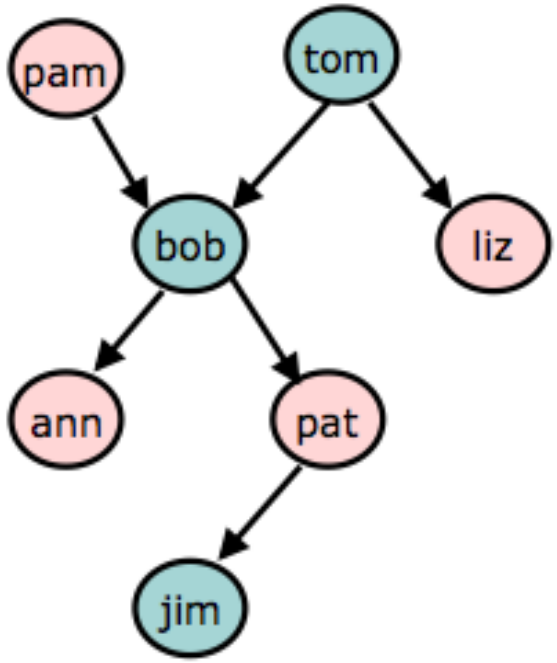
Pertemuan 2
Paradigma Pemrograman
(Logic Programming)

PARADIGMA PEMROGRAMAN

- Paradigma pemrograman adalah bentuk pemecahan masalah mengikuti aliran atau ”genre” tertentu dari program dan bahasa.
- Klasifikasi:

Imperative/ Algorithmic	Declarative		Object-Oriented
	Functional Programming	Logic Programming	
Algol Cobol PL/1 Ada C Modula-3	Lisp Haskell ML Miranda APL	Prolog	Smalltalk Simula C++ Java

PROLOG : Knowledge Representatiom



- Facts: offspring(liz, tom)
- RULE: offspring(Y, X):- parent(X, Y)

LOGIC PROGRAMMING (LP)

- Pemrograman deklaratif, mendeklarasikan tujuan komputasi, bukan menyusun algoritme secara detil. Disebut juga rule-based programming.
- LP dirancang untuk mendeskripsikan properti dari suatu obyek. Hubungan antar obyek dinyatakan dengan aturan if-then (jika-maka).
- LP memiliki mekanisme *built-in* untuk menarik kesimpulan (*inference*) berdasarkan deskripsi properti obyek tersebut.
- Contoh Aplikasi:
  - Artificial intelligence, misalnya MYCIN
  - Database information retrieval, misalnya SQL

FIRST ORDER PREDICATE LOGIC

- Dasar pemrograman logika adalah First Order Predicate Logic, sering disingkat FOPL, dengan versi khusus (penyederhanaan), yang disebut Horn Clause.
- Logika predikat merupakan bentuk notasi lain dari logika proposisi. Contoh fakta dalam logika proposisi: ”joko adalah lelaki”, ”amir adalah lelaki”, ”shinta adalah perempuan”; dalam logika proposisi misalnya dilambangkan dengan P, Q, R $\rightarrow$ sulit menggambarkan hubungan antar obyek.
- Lebih mudah ditulis sebagai:
  LELAKI(joko), LELAKI(amir), PEREMPUAN(shinta).

PREDICATE CALCULUS

- Ada 3 pengertian yang harus dipahami dalam kalkulus proposisi, yaitu:
  1. Terms
  2. Predicates
  3. Quantifiers
- **Term** adalah:
  - constant (konstanta), berupa individu atau konsep tunggal, misalnya 5, joko, dsb.
  - variable, menyatakan suatu individu atau konsep tunggal.
  - function, memetakan n term ke sebuah term. Misalnya f adalah simbol fungsi dan $t_1, \ldots, t_n$ adalah term, maka $f(t_1, \ldots, t_n)$ adalah term.

PREDICATE CALCULUS

- **Predicate** adalah:
  - relation yang memetakan n term ke nilai true (T) atau false (F).
  - Contoh:
    - LOVE(joko, tuty).
    - LOVE(father(joko), joko).

    LOVE adalah relasi, father adalah fungsi.
- **Quantifiers**:
  - Ada 2 jenis, yaitu ”ada” ($\exists$), dan ”untuk semua” ($\forall$).
  - Contoh:
    - ”setiap orang akan mati” ditulis sebagai $(\forall x)(ORANG(x) \rightarrow MATI(x))$
    - $(\forall y \exists x)(ORANG(y) \rightarrow IBU(x, y))$

PREDICATE CALCULUS

Contoh:

- Pernyataan ”x lebih besar dari y” dapat direpresentasikan dalam kalkulus predikat sebagai LEBIHBESAR(x,y). Jadi:

$$\begin{aligned} \text{LEBIHBESAR}(x, y) &= T, \text{ jika x>y} \\ &= F, \text{ selainnya} \end{aligned}$$

- Pernyataan joko mencintai (loves) setiap orang dapat direpresentasikan sebagai: $(\forall x)$ LOVE(joko, x).
- Pernyataan setiap ayah mencintai (loves) anaknya dapat direpresentasikan sebagai: $(\forall x)$ LOVE(ayah(x), x).

QUANTIFIER

Contoh:

- Everyone loves Honey-Bunny : $(\forall x)$LOVE(x, Honey-bunny)
- Someone loves Honey-Bunny : $(\exists x)$LOVE(x, Honey-Bunny)
- Someone loves everybody : $(\exists x)(\forall y)$LOVE(x, y)
- Everybody loves someone : $(\forall x)(\exists y)$LOVE(x,y)

LATIHAN KELAS 1

Tuliskan setiap pernyataan berikut ke dalam kalkulus predikat:

1. setiap mahasiswa memiliki nim.
2. dari semua mahasiswa, ada yang berhenti (DO)
3. tidak semua tanaman memiliki bunga
4. ada gajah yang jantan dan ada yang betina
5. tidak semua mahasiswa itu adalah manusia yang cerdas

HORN CLAUSE

Perhatikan pernyataan: **if** $(P_1 \wedge P_2 \wedge \cdots \wedge P_n)$ **then Q**.

- Dapat juga ditulis sebagai: $Q \leftarrow (P_1 \wedge P_2 \wedge \cdots \wedge P_n)$
  dibaca Q hanya jika P_1 dan P_2 dan . . . dan P_n
- Pernyataan Q akan benar (True) jika semua pernyataan $P_1, P_2 \ldots, P_n$ secara simultan benar.
- Ingat: $A \rightarrow B$ setara (memiliki nilai kebenaran yang sama) dengan $\neg A \vee B$.
- Oleh karena itu, pernyataan implikasi tadi dapat dinyatakan dalam bentuk *disjunctive normal* sebagai:
  $Q \vee \neg P_1 \vee \neg P_2 \vee \cdots \vee \neg P_n$
  Ekspresi ini disebut Horn Clause.

HORN CLAUSE

- Prolog meniru sejauh mungkin bentuk horn clause.
- Pernyataan $Q \leftarrow (P_1 \wedge P_2 \wedge \cdots \wedge P_n)$, dalam Prolog dituliskan sebagai: Q **:-** $P_1, P_2, \ldots, P_n$. Q disebut sebagai head, dan P disebut sebagai body.
- Penarikan kesimpulan dalam Prolog berdasarkan *modus ponens syllogism*:

```
(1) Jika P(x) maka Q(x)
(2) P(a)
(3) Kesimpulan: Q(a)
```

Contoh:

```
(1) Jika orang(X) maka mati(X)
(2) orang(socrates)
(3) Kesimpulan: mati(socrates)
```

- Prolog mengandung ekspresi logika bentuk (1):aturan dan (2):fakta.

PROLOG

- **Term**: constant|variable|structure
- **Constants**: atoms|integers
- **Atoms**: string karakter yang dimulai dengan huruf kecil, menyatakan nama dari obyek atau relasi. Misal: joko, ibu, lelaki
- **Variables**: string karakter yang dimulai dengan huruf besar. Misal: X, Joko
- **Structure**: predikat dengan nama dan sejumlah argumen yang telah *fixed*. Misal: ibu(shinta,X).

TYPES of CLAUSES

- **Rules** (aturan): ekspresi logika ”hanya jika”, Horn clauses.
- **Facts** (fakta): aturan tanpa *body*.

Contoh:

```
habisDibagiDua(X) :- genap(X).
```

Pernyataan tersebut setara dengan pernyataan kalkulus predikat:
$(\forall x)\ (\text{genap}(x) \rightarrow \text{habisDibagiDua}(x))$

CONVERSION OF FOL TO PROLOG

$(\forall x)\ (\forall y)\ (\forall z)\ (\text{father}(x, z) \wedge \text{parent}(z, y) \rightarrow \text{grandfather}(x, y))$

```
grandfather(x,y) :- father(x,z), parent(z,y)
```

Sentence	*Prolog writing*	*Terminology*
$a_1 \wedge a_2 \wedge \ldots \wedge a_n \Rightarrow c$	`c :- a1,a2,...,an.`	*Rule/clause*
`true` $\Rightarrow$ `c`	`c.`	*Unit rule/clause*
`c` $\Rightarrow$ `false`	`:- c`	*Goal*
`The list (a b c ...)`	`[a,b,c,...]`	
`The list (cons X L)`	`[X | L]`	
`The empty list`	`[]`	

CONVERT TO PROLOG FORM

- bill likes icecream : `likes(bill, icecream)`
- bill is tall : `height(bill, tall)`
- john travel to London by train : `travels(john, london, train)`
- if someone needs a bike then they may borrow jane's `borrow(x, bike, jane) :- need(x, bike)`
- all humans are mortal : $(\forall x)\,(\text{human}(x) \rightarrow \text{mortal}(x))$ `mortal(x) :- human(x)`

CONJUNCTION dan DISJUNCTION

- *Conjunction* dari predikat direpresentasikan sebagai deretan struktur yang dipisahkan oleh tanda koma (,).
- *Disjunction* dalam Prolog:
  - Menggunakan tanda titik koma (;) untuk memisahkan struktur
  - Menyusun dalam *clause* yang terpisah
- *Negation*: predikat untuk negasi adalah `not`.

Contoh:

- `sibling(X,Y):- kakak(X,Y) ; adik(X,Y).`
- ```
  sibling(X,Y):- kakak(X,Y).
  sibling(X,Y):- adik(X,Y).
  ```
- `ganjil(X):- not genap(X).`
  ```

LATIHAN KELAS 2

Buatlah klausa Prolog untuk persoalan berikut:

1. `X` adalah kakek dari `Y`.
2. `X` adalah nenek dari `Y`.
3. Faktorial dari 0 adalah 1.
4. Fungsi f didefinisikan sbb:

$$f(x,y) = \begin{cases} x & , x > y \\ y & , \text{selainnya} \end{cases}$$

HOMEWORK

Buat kalkulus predikat dan struktur Prolog untuk masalah berikut (penjelasan detil ada di LMS):

- Bill mengambil payungnya jika hujan
- Hewan buas adalah hewan yang berwarna gelap, berbadan besar, dan gigi bertaring.
- Hewan jinak adalah hewan yang berwarna terang, gigi tak bertaring dan berbadan kecil.
- Penjumlahan dua bilangan.
- Faktorial dari suatu bilangan bulat positif.

Jawaban diketik menggunakan komputer, simpan dalam format PDF, dan kumpulkan melalui LMS-IPB paling lambat Jumat, 4 Maret 2016 pukul 11am.