

Praktikum Representasi Pengetahuan

Sejarah Prolog

- Prolog singkatan dari **Programming in Logic**.
- Dikembangkan oleh Alain Colmenraurer dan P.Roussel di Universitas Marseilles Perancis, tahun 1972.
- Prolog populer di Eropa untuk aplikasi *artificial intelligence*, sedangkan di Amerika peneliti mengembangkan aplikasi yang sama, yaitu LISP.

Perbedaan Prolog dengan Bahasa Lainnya

- Bahasa Pemrograman yang Umum (Basic, Pascal, C, Fortran):
 - o diperlukan algoritma/prosedur untuk memecahkan masalah (*procedural language*)
 - o program menjalankan prosedur yang sama berulang-ulang dengan data masukan yang berbeda-beda.
 - o Prosedur dan pengendalian program ditentukan oleh programmer dan perhitungan dilakukan sesuai dengan prosedur yang telah dibuat.
- Bahasa Pemrograman Prolog :
 - o *Object oriented language* atau *declarative language*.
 - o Tidak terdapat prosedur, tetapi hanya kumpulan data-data objek (fakta) yang akan diolah, dan relasi antar objek tersebut membentuk aturan yang diperlukan untuk mencari suatu jawaban
 - o Programmer menentukan tujuan (goal), dan komputer menentukan bagaimana cara mencapai tujuan tersebut serta mencari jawabannya.
 - o Dilakukan pembuktian terhadap cocok-tidaknya tujuan dengan data-data yang telah ada dan relasinya.
 - o Prolog ideal untuk memecahkan masalah yang tidak terstruktur, dan prosedur pemecahannya tidak diketahui, khususnya untuk memecahkan masalah non numerik.
 - o Prolog bekerja seperti pikiran manusia, proses pemecahan masalah bergerak di dalam ruang masalah menuju suatu tujuan (jawaban tertentu).
 - o Contoh : Pembuatan program catur dengan Prolog

Aplikasi Prolog :

- Sistem Pakar (Expert System)
 - Program menggunakan teknik pengambilan kesimpulan dari data-data yang didapat, layaknya seorang ahli.
 - Contoh dalam mendiagnosa penyakit
- Pengolahan Bahasa Alami (Natural Language Processing)
 - Program dibuat agar pemakai dapat berkomunikasi dengan komputer dalam bahasa manusia sehari-hari, layaknya penterjemah.
- Robotik

Prolog digunakan untuk mengolah data masuk yang berasal dari sensor dan mengambil keputusan untuk menentukan gerakan yang harus dilakukan.

- Pengenalan Pola (Pattern Recognition)
Banyak digunakan dalam *image processing*, dimana komputer dapat membedakan suatu objek dengan objek yang lain.
- Belajar (Learning)
Program belajar dari kesalahan yang pernah dilakukan, dari pengamatan atau dari hal-hal yang pernah diminta untuk dilakukan.

Fakta dan Relasi

- Prolog terdiri dari kumpulan data-data objek yang merupakan suatu fakta.
- Fakta dibedakan 2 macam :
 - o Menunjukkan relasi.
 - o Menunjukkan milik/sifat.
- Penulisannya diakhiri dengan tanda titik “.”
- Contoh :

Fakta	Prolog
Slamet adalah ayah Amin	ayah (slamet, amin).
Anita adalah seorang wanita	wanita (anita).
Angga suka renang dan tenis	suka(angga, renang). dan suka(angga,tenis).
Jeruk berwarna jingga	jingga(jeruk).

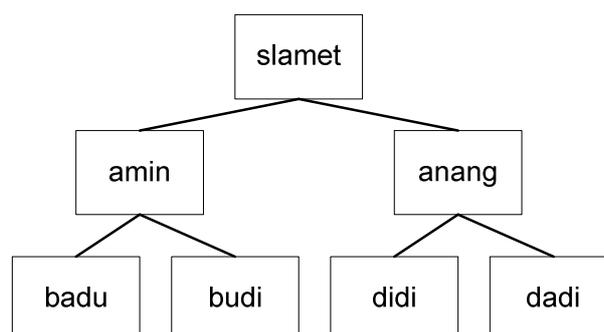
Aturan (“Rules”)

- Aturan adalah suatu pernyataan yang menunjukkan bagaimana fakta-fakta berinteraksi satu dengan yang lain untuk membentuk suatu kesimpulan.
- Sebuah aturan dinyatakan sebagai suatu kalimat bersyarat.
- Kata “if” adalah kata yang dikenal Prolog untuk menyatakan kalimat bersyarat atau disimbolkan dengan “:-”.
- Contoh :

Fakta dan Aturan	Prolog
F : Tino suka apel A : Yuli suka sesuatu yang disukai Tino	suka(tino, apel). suka(yuli,Sesuatu) :- suka(tino,Sesuatu).

- Setiap aturan terdiri dari kesimpulan(kepala) dan tubuh.
- Tubuh dapat terdiri dari 1 atau lebih pernyataan atau aturan yang lain, disebut *subgoal* dan dihubungkan dengan logika “and”.
- Aturan memiliki sifat *then/if conditional*
“Kepala(head) benar jika tubuh (body) benar”.

- Contoh : Silsilah keluarga :



Fakta dan Aturan	Prolog
<p>F : Slamet adalah ayah dari Amin</p> <p>F : Amin adalah ayah dari Badu</p> <p>F : Slamet adalah kakek dari Badu</p> <p>A :</p> <p>Slamet adalah benar kakek Badu jika Slamet adalah benar ayah Amin dan Amin adalah benar ayah Badu.</p>	<p>ayah(slamet,amin).</p> <p>ayah(amin,badu).</p> <p>kakek(slamet,badu).</p> <p>kakek(slamet,amin) :- ayah(slamet,amin) and ayah(amin,badu).</p>
<p>F : Amin adalah saudara kandung Anang</p> <p>F : Amin mempunyai ayah Slamet</p> <p>F : Anang mempunyai ayah Slamet.</p> <p>A :</p> <p>Amin adalah benar saudara kandung Anang, jika Amin mempunyai ayah Slamet dan Anang juga mempunyai ayah Slamet.</p>	<p>saudara_kandung(amin,anang).</p> <p>ayah(slamet,amin).</p> <p>ayah(slamet,anang).</p> <p>saudara_kandung(amin,anang) :- ayah(slamet,amin) and ayah(slamet,anang).</p>

Pertanyaan (“Query”)

- Setelah memberikan data-data berupa fakta dan aturan, selanjutnya kita dapat mengajukan pertanyaan berdasarkan fakta dan aturan yang ada.
- Penulisannya diawali simbol “?-“ dan diakhiri tanda “.”.
- Contoh :

Pertanyaan	Prolog dan Jawaban Program
Apakah Tini suka boneka ?	?- suka(tini,boneka). Yes (jika faktanya Tini suka boneka) No.....(jika tidak sesuai fakta).
Apakah yang disukai Tini ?	?- suka(tini,Apa). Apa=boneka
Siapakah yang suka boneka ?	?- suka(Siapa,boneka).

	Siapa=tini
Dari contoh silsilah keluarga di atas : Siapakah kakek Budi ?	?- kakek(Siapa,budi). Siapa=slamet
Siapakah cucu Slamet ?	?- kakek(slamet,Cucu). Cucu=budi ; Cucu=badu
	(jika kemungkinan ada lebih dari satu jawaban gunakan tanda “;” pada akhir setiap jawaban).

Predikat (“Predicate”)

- Predikat adalah nama simbolik untuk relasi.
- Contoh : ayah(slamet,amin).
Predikat dari fakta tersebut ditulis : ayah(simbol,simbol).

dimana **ayah** adalah nama **predikat**, sedangkan **slamet** dan **amin** adalah menunjukkan **argumen**.

Sebuah predikat dapat tidak memiliki atau memiliki argumen dengan jumlah bebas. Jumlah argumen suatu predikat disebut **aritas** (arity).

ayah(nama) aritas-nya 1

ayah(nama1,nama2) aritasnya 2

- Syarat-syarat penulisan nama predikat :
 - o Harus diawali dengan huruf kecil dan dapat diikuti dengan huruf, bilangan atau garis bawah.
 - o Panjang nama predikat maksimum 250 karakter.
 - o Tidak diperbolehkan menggunakan spasi, tanda minus, tanda bintang dan garis miring.

Variabel

- Variabel adalah besaran yang nilainya dapat berubah-ubah.
- Tata cara penulisan variabel :
 1. Nama variabel harus diawali huruf besar atau garis bawah(_)
 2. Nama variabel dapat terdiri dari huruf, bilangan, atau simbol dan merupakan kesatuan dengan panjang maksimum 250 karakter.
 3. Nama variabel hendaknya mengandung makna yang berkaitan dengan data yang dinyatakannya.

- Contoh : dari silsilah di atas :

?- ayah(slamet,**Anak**).

Anak=budi ;

Anak=badu

No

Dari query di atas akan dicari siapakah anak dari ayah yang bernama Slamet. Karena mempunyai relasi yang sama (yaitu ayah), **variabel Anak** akan mencari nilai dari konstanta suatu fakta/aturan yang sepadan.

Tanda “;” digunakan bila terdapat kemungkinan ada lebih dari satu jawaban.

“No” berarti tidak ada lagi kemungkinan jawaban.

- Contoh : dari silsilah di atas :

?- ayah(slamet,**X**),ayah(**X**,**Y**).

X=amin

Y=budi ;

X=amin

Y=badu ;

X=anang

Y=didi ;

X=anang

Y=didi

No

Contoh : Silsilah Keluarga

%% FAKTA */

%%orang tua */

ayah(slamet,amin).

ayah(slamet,anang).

ayah(amin,budi).

ayah(amin,badu).

ayah(anang,didi).

ayah(anang,dadi).

ayah(santoso,bu_amin).

ayah(supardi,bu_anang).

ibu(bu_slamet,amin).

ibu(bu_slamet,anang).

ibu(bu_amin,budi).

ibu(bu_amin,badu).

ibu(bu_anang,didi).

ibu(bu_anang,dadi).

ibu(bu_santoso,bu_amin).

ibu(bu_santoso,bu_anang).

%% ATURAN */

%% Kakek adalah kakek Cucu */

```
kakek(Kakek,Cucu) :-  
    ayah(Ayah,Cucu),  
    ayah(Kakek,Ayah).  
kakek(Kakek,Cucu) :-  
    ibu(Ibu,Cucu),  
    ayah(Kakek,Ibu).
```

%% Nenek adalah nenek Cucu */

```
nenek(Nenek,Cucu) :-  
    ayah(Ayah,Cucu),  
    ibu(Nenek,Ayah).  
nenek(Nenek,Cucu) :-  
    ibu(Ibu,Cucu),  
    ibu(Nenek,Ibu).
```

%% Nama1 adalah saudara kandung Nama2

```
saudara_kandung>Nama,Name) :-  
    ayah(Ayah>Nama),  
    ayah(Ayah>Name),  
    ibu(Ibu>Nama),  
    ibu(Ibu>Name),  
    Nama \= Name.
```

%% Sdr1 adalah saudara sepupu Sdr2

```
saudara_sepupu(Sdr1,Sdr2) :-  
    ayah(Ayah1,Sdr1),  
    ayah(Ayah2,Sdr2),  
    saudara_kandung(Ayah1,Ayah2).
```

```
saudara_sepupu(Sdr1,Sdr2) :-  
    ayah(Ayah,Sdr1),  
    ibu(Ibu,Sdr2),  
    saudara_kandung(Ayah,Ibu).
```

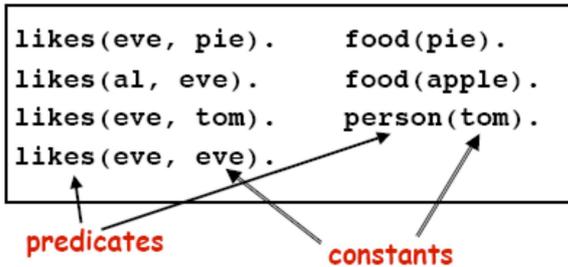
```
saudara_sepupu(Sdr1,Sdr2) :-  
    ibu(Ibu,Sdr1),  
    ayah(Ayah,Sdr2),  
    saudara_kandung(Ibu,Ayah).
```

```
saudara_sepupu(Sdr1,Sdr2) :-  
    ibu(Ibu1,Sdr1),  
    ibu(Ibu2,Sdr2),  
    saudara_kandung(Ibu1,Ibu2).
```

Dengan menggunakan SWI Prolog buatlah praktikum seperti di bawah ini !

Praktikum 1 : Fakta dan Query

Facts



Queries (Asking Questions)

likes(eve, pie).	food(pie).
likes(al, eve).	food(apple).
likes(eve, tom).	person(tom).
likes(eve, eve).	

<pre> ?-likes(al, eve). yes ?-likes(al, pie) no ?-likes(eve, al). no ?-likes(person, food). no </pre> <p style="color: red; margin-left: 20px;">query</p> <p style="color: red; margin-left: 20px;">answer</p>	<pre> ?-likes(al, who). who=eve ?-likes(eve, W). W=pie ; W=tom ; W=eve ; no </pre> <p style="color: red; margin-left: 20px;">variable</p> <p style="color: red; margin-left: 20px;">answer with variable binding</p> <p style="color: red; margin-left: 20px;">force search for more answers</p>
--	--

Prolog, CS314 Fall 2007 © BGRyder/Borgida

Harder Queries

likes(eve, pie).	food(pie).
likes(al, eve).	food(apple).
likes(eve, tom).	person(tom).
likes(eve, eve).	


```

?-likes(A, B).
A=eve, B=pie ; A=al, B=eve ; ...
?-likes(D, D).
D=eve ; no
?-likes(eve, W), person(W).
W=tom
?-likes(al, V), likes(eve, V).
V=eve ; no
                
```

and

Harder Queries

```
likes(eve, pie).    food(pie).
likes(al, eve).    food(apple).
likes(eve, tom).   person(tom).
likes(eve, eve).
```

same binding

```
?-likes(eve,W),likes(W,V).
W=eve,V=pie ; W=eve,V=tom ; W=eve,V=eve
?-likes(eve,W),person(W),food(V).
W=tom,V=pie ; W=tom,V=apple or
?-likes(eve,V),(person(V);food(V)).
V=pie ; V=tom ; no
?-likes(eve,W),\+likes(al,W).
W=pie ; W=tom ; no not
```

Prolog, CS314 Fall 2007 © BGRyder/Borgida

Rules

```
likes(eve, pie).    food(pie).
likes(al, eve).    food(apple).
likes(eve, tom).   person(tom).
likes(eve, eve).
```

What if you want to ask the same question often? Add a *rule* to the database:

```
rule1:-likes(eve,V),person(V). |
```

```
?-rule1.
yes
```

Rules

```
likes(eve, pie).      food(pie).
likes(al, eve).      food(apple).
likes(eve, tom).     person(tom).
likes(eve, eve).
rule1:-likes(eve,V),person(V).
rule2(V):-likes(eve,V),person(V).
```

```
?-rule2(H).
H=tom ; no
?-rule2(pie).
no
```

Note rule1 and rule2 are just like any other predicate!

Praktikum 2 : Contoh Queen Victoria

Queen Victoria Example

```
male(albert).      a fact
female(alice).    Facts are put in a file.
male(edward).
female(victoria).
parents(edward,victoria,albert).
parents(alice,victoria,albert).
?- [family].      loads file
yes
?- male(albert).  a query
yes
?- male(alice).
no
?- parents(edward,victoria,albert).
yes
?- parents(bullwinkle,victoria,albert).
no
```

cf Clocksin
and Mellish

Queen Victoria Example, cont.

- Problem: facts alone do not make interesting programs possible. Need variables and deductive rules.

```
?-female(X).           a query or proposed fact
X = alice ;           ; asks for more answers
X = victoria ;       if user types <return> then
                    no more answers given
no when no more answers left, return no
```

- Variable X has been unified to all possible values that make female(X) true.
 - Performed by pattern match search
- Variables are capitalized, predicates and constants are lower case

Queen Victoria Example, cont.

```
sister_of(X,Y):-
    female(X),parents(X,M,F),parents(Y,M,F).
                    a rule
```

```
?- sister_of(alice,Y).
Y = edward
?- sister_of(alice, victoria).
no
```

Example

```
?-sister_of(X,Y):
    female(X),parents(X,M,F),parents(Y,M,F).
?-sister_of(alice,Y).
Y = edward
?-sister_of(X,Y).
X = alice
Y = edward ;
X = alice
Y = alice ;
no
```

```
(1)male(albert).
(2)female(alice).
(3)male(edward).
(4)female(victoria).
(5)parents(edward,victoria,albert).
(6)parents(alice,victoria,albert).
```

Example shows

- subgoal order of evaluation
- argument invertability
- backtracking
- computation in rule order

What's wrong here?

Example

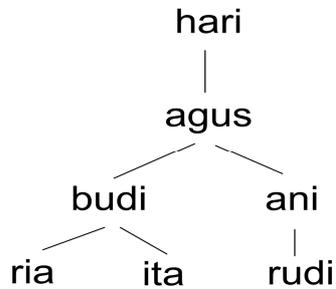
```
sis(X,Y):-female(X),parents(X,M,F),
           parents(Y,M,F),\+(X==Y).
?-sis(X,Y). last subgoal disallows X,Y to have same
value
X=alice
Y=edward ;
no
```

= means *unifies with*

== means *same in value*

Praktikum 3 : Silsilah Keluarga

- Cobalah representasi pengetahuan dari silsilah keluarga berikut ini



Database Pengetahuan

anaklaki(hari, agus). → dibaca sebagai anak laki-laki dari hari adalah agus

anaklaki(agus, budi).

anaklaki(ani, rudi).

anakperempuan(agus, ani).

anakperempuan(budi, ria).

anakperempuan(budi, ita).

Lakukan uji coba seperti dibawah ini :

anaklaki(OT,AL).

Hasil : OT=hari AL=agus

OT=agus AL=budi

OT=ani AL=rudi

Lakukan uji coba seperti dibawah ini :

anaklaki(P,budi), anakperempuan(P,D). → untuk menanyakan saudara perempuan dari budi

Hasil : P=agus D=ani

Kita dapat menambahkan hubungan antara knowledge, misalnya antara anak laki dan anak perempuan di-extend menjadi hubungan orangtua

Database Pengetahuan

orangtua(C,P):-anaklaki(P,C).

orangtua(C,P):-anakperempuan(P,C).

Lakukan uji coba seperti dibawah ini :

orangtua(budi,P).

Hasil : P=agus

Recursive description of relationship

- Jika ingin mencari turunan, berikut cara merepresentasikannya:

Database Pengetahuan

turunan(OT,TR):-orangtua(TR, OT). → anak

turunan(OT,TR):-orangtua(TR, P), orangtua(P, OT). → cucu

turunan(OT,TR):- orangtua(TR, P), orangtua(P,GP),orangtua(GP, OT). → cicit

- Fakta di atas dapat diekspresikan dengan proses rekursif sbb:

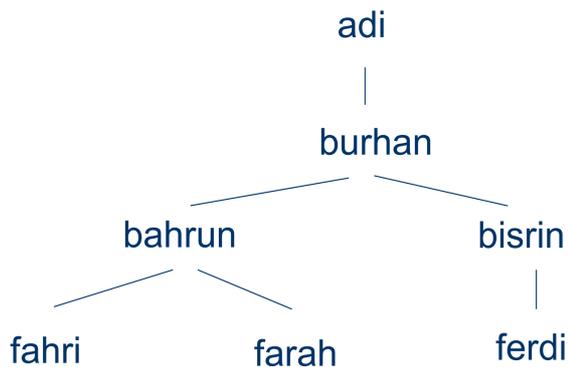
Database Pengetahuan

turunan(OT,TR):-orangtua(TR,OT). → anak

turunan(OT,TR):- orangtua(TR,P),turunan(OT,P).

Tugas Praktikum :

1. Dalam sebuah organisasi perusahaan terdapat tree sebagai berikut:

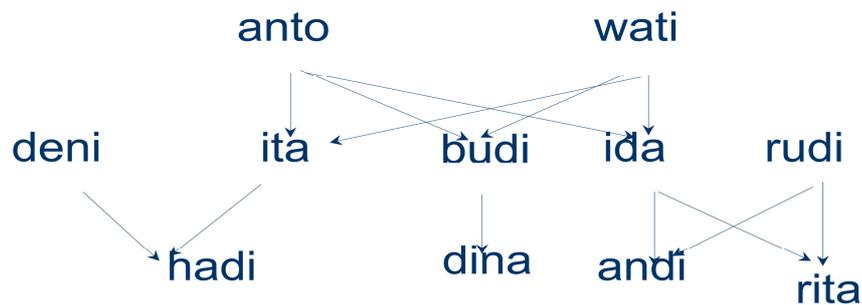


Dari tree tersebut kita dapat membaca bahwa Burhan adalah **bawahan langsung** dari Adi, sebaliknya Adi adalah **atasan langsung** dari Burhan. Fahri dan Farah adalah **anak buah** dari Bahrn, sementara Fahri, Farah, Bahrn, Ferdi, Bisrin secara keseluruhan adalah anak buah dari Burhan.

- Dengan menggunakan sintaks dalam bahasa Prolog, buatlah representasi pengetahuan dari fakta tersebut di atas. (Dari definisi **bawahan langsung**).
- Dengan menggunakan sintaks dari definisi bawahan langsung di atas, terjemahkan untuk **atasan langsung**.
- Bagaimana kita membuat pertanyaan **siapa bawahan langsung dari Burhan**

Dengan menggunakan deskripsi secara rekursif buatlah sintak untuk merepresentasikan fakta **anak buah**.

2. Dari pohon keluarga di bawah ini, tuliskan suatu program dalam bahasa Prolog yang menyatakan predikat **anak, perempuan, laki-laki**. Kemudian dari predikat ini buatlah relasi **orang tua, ayah, ibu, saudara laki-laki, saudara perempuan, paman, bibi, kakek, nenek**.



3. Dengan representasi secara logika di bawah ini jawablah pertanyaan-pertanyaan di bawahnya

```

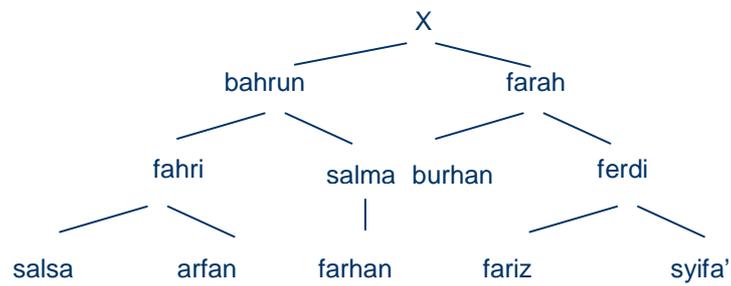
besar(kuda).
besar(sapi).
besar(gajah).
kecil(kelinci).
kecil(tikus).
coklat(kuda).
putih(sapi).
abuabu(gajah).
abuabu(tikus).
putih(kelinci).
terang(X):-putih(X).
terang(X):-abuabu(X).
  
```

Jelaskan hasil dari query berikut ini?

```

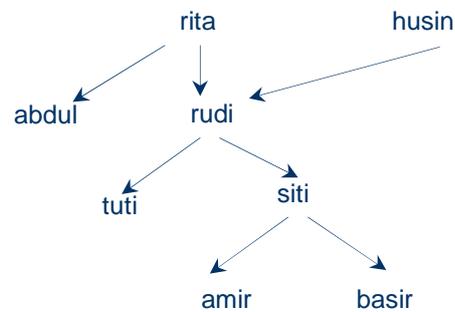
putih(Y),besar(Y).
abuabu(Y),kecil(Y).
  
```

4. Seorang raja X dengan silsilah keturunannya seperti di bawah ini sedang mencari siapa saja dari keturunannya yang bisa menggantikan dirinya untuk menjadi raja. Tentu saja ada syarat untuk menjadi calon raja, yaitu dia adalah **keturunan laki-laki atau keturunan laki-laki dari keturunan laki-laki**. Dengan menggunakan representasi logika dan deskripsi secara rekursif bantulah raja X untuk **mencari siapa saja dari keturunannya yang memungkinkan untuk menggantikan dirinya**.



5. Dari tree famili di bawah ini, formulasikan dalam bahasa Prolog pertanyaan tentang:

- Siapa orang tua Basir.
- Siapa nenek Siti.
- Apakah Tuti mempunyai anak.



***** Selamat Mengerjakan *****