

NAMA : EVI LOLITA APRIYANI

NIM : 1515015128

PROSES LACAK BALIK ADA DI BAWAH

DOMAINS

Disini kita dapat mendeklarasi domain khusus yang dapat digunakan untuk mendeklarasikan struktur data yang tidak didefinisikan oleh standar domain Digunakan sebagai tempat untuk inisialisasi variabel

nama,jender,pekerjaan,benda,alasan,zat = symbol

dimana domain berisi nama,jender,pekerjaan,benda,alasan dan zat bertipe data symbol.symbol dipakai pada predikat yang argumennya memiliki nilai berupa kumpulan karakter yang hanya terdiri dari huruf atau kata tanpa dipisahkan oleh spasi.

umur=integer

domain umur bertipe data integer. integer dipakai pada predikat yang argumennya memiliki nilai berupa bilangan bulat.

PREDICATES

Digunakan sebagai tempat untuk mendeklarasikan predikat yang ada pada bagian Predicates dan mendefinisikannya dalam bagian Clauses

nondeterm orang(nama, umur, jender, pekerjaan)

nondeterm pada section predicates berfungsi untuk memberi tahu compiler bahwa predikat tersebut mempunyai lebih dari satu solusi, predicates dengan nama orang memiliki parameter yang di dalamnya terdapat variabel (nama,umur,jender,pekerjaan) dengan tipe data symbol untuk nama,jender,pekerjaan dan umur bertipe data integer yang sudah di inisialisasikan pada domain

nondeterm selingkuh(nama, nama)

nondeterm pada section predicates berfungsi untuk memberi tahu compiler bahwa predikat tersebut mempunyai lebih dari satu solusi, predicates dengan nama selingkuh memiliki parameter yang di dalamnya terdapat variabel (nama,nama) dengan tipe data symbol yang sudah di inisialisasi pada domain

terbunuh_dengan(nama, benda)

predicates dengan nama terbunuh_dengan memiliki parameter yang di dalamnya terdapat variabel (nama, benda) yang bertipe data symbol yang sudah di inisialisasikan pada domain

terbunuh(nama)

predicates dengan nama terbunuh memiliki parameter yang di dalamnya terdapat variabel(nama) yang bertipe data symbol yang sudah di inisialisasikan pada domain

nondeterm pembunuh(nama)

nondeterm pada section predicates berfungsi untuk memberi tahu compiler bahwa predikat tersebut mempunyai lebih dari satu solusi, predicates dengan nama pembunuh memiliki parameter yang di dalamnya terdapat variabel (nama) dengan tipe data symbol yang sudah di inisialisasikan pada domain

motif(alasan)

predicates dengan nama motif memiliki parameter yang di dalamnya terdapat variabel (alasan) yang bertipe data symbol yang sudah di inisialisasikan pada domain

ternodai(nama, zat)

predicates dengan nama ternodai memiliki parameter yang di dalamnya terdapat variabel (nama,zat) yang bertipe data symbol yang sudah di inisialisasikan pada domain

milik(nama, benda)

predicates dengan nama milik memiliki parameter yang di dalamnya terdapat variabel (nama ,benda) yang bertipe data symbol yang sudah di inisialisasikan pada domain

nondeterm cara_kerja_mirip(benda, benda)

nondeterm pada section predicates berfungsi untuk memberi tahu compiler bahwa predikat tersebut mempunyai lebih dari satu solusi, perdicates dengan nama cara_kerja_mirip memiliki parameter yang di dalamnya terdapat variabel (benda,benda) yang bertipe data symbol yang sudah di inisialisasikan pada domain

nondeterm kemungkinan_milik(nama, benda)

nondeterm pada section predicates berfungsi untuk memberi tahu compiler bahwa predikat tersebut mempunyai lebih dari satu solusi, predicates dengan nama kemungkinan_milik memiliki parameter yang di dalmnya terdapat varibel (nama,benda) yang bertipe data symbol yang sudah di inisialisasikan pada domain

nondeterm dicurigai(nama)

nondeterm pada section predicates berfungsi untuk memberi tahu compiler bahwa predikat tersebut mempunyai lebih dari satu solusi predicates dengan nama di curigai memiliki variabel yang di dalamnya terdapat variabel (nama) yang bertipe data symbol yang sudah di inisialisasikan pada domain

CLAUSES

Digunakan sebagai tempat untuk meletakkan fakta dan aturan

% --- FAKTA --- %

orang(budi,55,m,tukang_kayu).

predikat orang(nama,umur,jender,pekerjaan) /orang nama = budi, umur = 55, jender = m dan pekerjaan = tukang_kayu / orang dengan nama budi umurnya 55 dengan jender m dan pekerjaan sebagai tukang kayu

orang(aldi,25,m,pemain_sepak_bola).

predikat orang(nama,umur,jender,pekerjaan) /orang nama = aldi, umur = 25, jender = m, pekerjaan = pemain_sepak_bola /orang dengan nama aldi umur 25 dengan jender m dan pekerjaan sebagai pemain sepak bola

orang(aldi,25,m,tukang_jagal).

predikat orang(nama,umur,jender,pekerjaan) /orang nama = aldi, umur = 25, jender = m, pekerjaan = tukang_jagal / orang dengan nama aldi umur 25 dengan jender m dan pekerjaan sebagai tukang jagal

orang(joni,25,m,pencopet).

predikat orang(nama,umur,jender,pekerjaan) /orang nama = joni, umur = 25, jender = m, pekerjaan = pencopet / orang dengan nama joni umur 25 dengan jender m dan pekerjaan sebagai pencopet

selingkuh(ina,joni).

predikat selingkuh(nama,nama) /selingkuh nama = ina dan nama = joni /ina selingkuh dengan joni

selingkuh(ina,budi).

predikat selingkuh(nama,nama) /selingkuh nama = ina dan nama = budi /ina selingkuh dengan budi

selingkuh(siti,joni).

predikat selingkuh(nama,nama) /selingkuh nama = siti dan nama = joni /siti selingkuh dengan joni

terbunuh_dengan(siti,pentungan).

predikat terbunuh_dengan(nama,benda)/ terbunuh_dengan nama = siti , benda = pentungan / siti terbunuh dengan pentungan

terbunuh(siti).

predikat terbunuh(nama) /terbunuh nama = siti / siti yang terbunuh

motif(uang).

predikat motif(alasan) /motif alasan = uang / uang adalah motif terbunuhnya siti

motif(cemburu).

predikat motif(alasan) /motif alasan = cemburu / cemburu adalah motif terbunuhnya siti

motif(dendam).

predikat motif(alasan) /motif alasan = dendam / dendam adalah motif terbunuhnya siti

ternodai(budi, darah).

predikat ternodai(nama,zat) /ternodai nama = budi, zat = darah / budi ternodai darah

ternodai(siti, darah).

predikat ternodai(nama,zat) /ternodai nama = siti, zat = darah / siti ternodai darah

ternodai(aldi, lumpur).

predikat ternodai(nama,zat) /ternodai nama = aldi, zat = lumpur / aldi ternodai lumpur

ternodai(joni, coklat).

predikat ternodai(nama,zat) /ternodai nama = joni, zat = coklat / joni ternodai coklat

ternodai(ina, coklat).

predikat ternodai(nama,zat) /ternodai nama = ina, zat = coklat / budi ternodai coklat

milik(budi,kaki_palsu).

predikat milik(nama,benda) /milik nama = budi, benda = kaki_palsu /budi memiliki kaki palsu

milik(joni,pistol).

predikat milik(nama,benda) /milik nama = joni, benda = pistol /joni memiliki pistol

cara_kerja_mirip(kaki_palsu, pentungan).

predikat cara_kerja_mirip(benda,benda) /cara_kerja_mirip benda = kaki_palsu, benda = pentungan / kaki palsu cara kerja mirip dengan pentungan

cara_kerja_mirip(balok, pentungan).

predikat cara_kerja_mirip(benda,benda) /cara_kerja_mirip benda = balok, benda = pentungan / balok cara kerja mirip dengan pentungan

cara_kerja_mirip(gunting, pisau).

predikat cara_kerja_mirip(benda,benda) /cara_kerja_mirip benda = gunting, benda = pisau / gunting cara kerja mirip dengan pisau

cara_kerja_mirip(sepatu_bola, pentungan).

predikat cara_kerja_mirip(benda,benda) /cara_kerja_mirip benda = sepatu_bola, benda = pentungan / sepatu bola cara kerja mirip dengan pentungan

% --- ATURAN --- %

% Nb :

% (:-) jika

% (<>) tidak sama dengan

% (,) dan

% (;) atau

% () mengabaikan variabel yang tidak ingin ditampilkan

% (X,Benda,Senjata,Lakilaki,Pembunuh,Zat) variabel yang merupakan sebuah interpretasi yang digunakan untuk menemukan nilai dari sebuah klausa, Variabel harus diawali dengan huruf besar

kemungkinan_milik(X,sepatu_bola):-

predikat kemungkinan_milik(nama,benda) /kemungkinan_milik nama= X, benda = sepatu_bola

orang(X,_,_,pemain_sepak_bola).

```
predikat orang(nama,umur,jender,pekerjaan) /orang nama = X umur = _, jender = _, pekerjaan =
pemain_sepak_bola
```

➤ X kemungkinan memiliki sepatu bola jika x adalah pemain sepak bola

kemungkinan_milik(X,gunting):-

predikat kemungkinan_milik(nama,benda) /kemungkinan_milik nama= X, benda = gunting

orang(X,_,_,pekerja_salon).

```
predikat orang(nama,umur,jender,pekerjaan) /orang nama = X umur = _, jender = __, pekerjaan =
pekerja_salon
```

- X kemungkinan memiliki gunting jika X seorang pekerja salon

kemungkinan_milik(X,Benda):-

predikat kemungkinan_milik(nama,benda) /kemungkinan_milik nama= X, benda = Benda

milik(X,Benda).

predikat milik(nama,benda) /orang nama = X benda = Benda

➤ X kemungkinan memiliki Benda jika X memiliki benda tersebut

/*****

* dicurigai semua orang yang memiliki senjata yang *

* kerjanya mirip dengan senjata penyebab siti terbunuh. *

*****/

dicurigai(X):-

X di curigai jika

terbunuh_dengan(siti,Senjata) ,

siti terbunuh dengan Senjata

cara_kerja_mirip(Benda,Senjata) ,

Benda yang cara kerjanya mirip Senjata

kemungkinan_milik(X,Benda).

X kemungkinan memiliki Benda tersebut

➤ X dicurigai jika siti terbunuh dengan senjata dan benda yang cara kerjanya mirip senjata dan x kemungkinan memiliki benda itu.

/* * * * * *

* dicurigai laki-laki yang selingkuh dengan siti. *

* * * * * */

dicurigai(X):-

X dicurigai jika

motif(cemburu),

cemburu adalah motifnya dan

orang(X,_m,_),

X adalah orang dengan gender m

selingkuh(siti,X).

siti selingkuh dengan x

➤ X dicurigai jika mempunyai motif cemburu dan X orang yang bergender m dan siti berselingkuh dengan x

/* * * * * *

* dicurigai perempuan yang selingkuh dengan *

* laki-laki yang juga selingkuh dengan siti *

* * * * * */

dicurigai(X):-

X di curigai jika

motif(cemburu),

cemburu adalah motifnya dan

orang(X,_f,_),

X adalah orang dengan jender f dan

selingkuh(X,Lakilaki),

X selingkuh dengan Lakilaki dan

selingkuh(siti,Lakilaki).

siti selingkuh dengan Lakilaki

➤ X dicurigai jika mempunyai motif cemburu dan X orang yang bergender f dan X berselingkuh dengan Lakilaki

/* * * * * * * * * * * * * * * * * * * * * * * * *

* dicurigai pencopet yang mempunyai motif uang. *

* * * * * * * * * * * * * * * * * * * * * * * * */

dicurigai(X):-

X di curigai jika

motif(uang),

uang adalah motifnya dan

orang(X,_,_pencopet).

X adalah orang yang pekerjaannya pencopet

➤ X dicurigai jika uang adalah motifnya dan x adalah seorang pencopet

pembunuh(Pembunuh):-

Pembunuh adalah pembunuh jika

orang(Pembunuh,_,_),

Pembunuh adalah orang dan

terbunuh(Terbunuh),

Terbunuh telah terbunuh dan

Terbunuh <> Pembunuh,

Terbunuh tidak sama dengan Pembunuh /* Bukan bunuh diri */

dicurigai(Pembunuh),

Pembunuh di curigai dan

ternodai(Pembunuh,Zat),

Pembunuh ternodai Zat dan

ternodai(Terbunuh,Zat).

Terbunuh ternodai Zat

- Pembunuh adalah pembunuh jika dia pembunuh orang dan terbunuh adalah terbunuh dan terbunuh tidak sama dengan pembunuh dan pembunuh dicurigai dan pembunuh ternodai zat dan terbunuh ternodai zat

GOAL

sama dengan body dari sebuah aturan, tujuan akhir

pembunuh(X).

X adalah pembunuhnya

Untuk mengetahui terjadinya lacakbalik, akan diamati langkah demi langkah bagaimana Visual Prolog mencari solusi dari goal yang diberikan.

1. Visual Prolog memanggil predikat pembunuh dengan variabel bebas X. Untuk mencoba menjawab panggilan ini, Visual Prolog mencari di program untuk pencocokan. Ditemukan kecocokan dengan klausa pembunuh, dan variabel Apa diunifikasikan dengan variabel Pembunuh
2. Kemudian, Visual Prolog berusaha untuk memenuhi bagian *body* dari *rule*. Untuk melakukannya, Visual Prolog memanggil subgoal yang pertama pada *body* dari *rule* tersebut yaitu orang(Pembunuh,_,_,_), dan mencari pencocokan untuk panggilan ini. Ditemukan pencocokan dengan fakta pertama dari klausa relasi orang. Pada titik ini Pembunuh diikat dengan nilai budi. Kemudian Visual Prolog menset titik lacakbalik pada fakta orang(budi,55,m,tukang_kayu).
3. Dengan Pembunuh diikat pada nilai budi, Visual Prolog membuat panggilan untuk subgoal yang kedua yaitu terbunuh(Terbunuh), dan mencari pencocokan. Dan menemukan dengan fakta yang terbunuh(siti). Pada titik ini Terbunuh diikat dengan nilai siti dan Prolog menset titik lacakbalik pada terbunuh(siti).
4. Dengan Terbunuh diikat pada nilai siti, Visual Prolog membuat panggilan untuk subgoal yang selanjutnya yaitu adalah dicurigai(Pembunuh), dan mencari pencocokan. Karena sebelumnya variable Predikat Pembunuh diikat dengan variable bebas x maka subgol diganti menjadi dicurigai(x). Dan menemukan aturan dicurigai(X): . Kemudian visual prolog Visual Prolog memanggil subgoal yang pertama pada *body* dari *rule* tersebut yaitu terbunuh_dengan(siti,Senjata) , dan mencari pencocokan untuk panggilan ini. Ditemukan pencocokan dengan fakta terbunuh_dengan(siti,pentungan). Pada titik ini Senjata diikat dengan nilai pentungan dan Prolog menset titik lacakbalik pada terbunuh_dengan(siti,pentungan).
5. Dengan Senjata diikat pada nilai pentungan, Visual Prolog membuat panggilan untuk subgoal selanjutnya yaitu cara_kerja_mirip(Benda,Senjata) , dan mencari pencocokan. Karena sebelumnya variable Senjata diikat dengan nilai pentungan maka subgol diganti menjadi yaitu cara_kerja_mirip(Benda,pentungan). Dan menemukan dengan fakta yang cara_kerja_mirip(kaki_palsu, pentungan). Pada titik ini Benda diikat dengan nilai kaki_palsu dan Prolog menset titik lacakbalik pada cara_kerja_mirip(kaki_palsu, pentungan).
6. Dengan Benda diikat pada kaki_palsu, Visual Prolog membuat panggilan untuk subgoal selanjutnya yaitu kemungkinan_milik(X,Benda), dan mencari pencocokan. Dan menemukan dengan aturan yang kemungkinan_milik(X,Benda). Kemudian, Visual Prolog berusaha untuk

memenuhi bagian *body* dari *rule*. Untuk melakukannya, Visual Prolog memanggil subgoalnya yaitu milik(X,Benda). Dan mencari pencocokan untuk panggilan ini. Karena sebelumnya variable Benda diikat dengan nilai kaki_palsu maka subgol diganti menjadi yaitu milik(X,kaki_palsu). Ditemukan pencocokan dengan fakta dari klausa relasi milik. Pada titik ini X diikat dengan nilai budi. Kemudian Visual Prolog menset titik lacakbalik pada fakta milik(budi,kaki_palsu).

7. Dengan X diikat pada budi, Visual Prolog membuat panggilan untuk subgoal selanjutnya yaitu ternodai(Pembunuh,Zat), dan mencari pencocokan. Karena sebelumnya variable Pembunuh diikat dengan nilai budi maka subgol diganti menjadi ternodai(Budi,Zat). Dan menemukan fakta ternodai(Budi,darah) . Pada titik ini Zat diikat dengan nilai darah. Kemudian Visual Prolog menset titik lacakbalik pada fakta ternodai(Budi,darah) .
8. Visual Prolog menemukan kecocokan dengan klausa ternodai berikutnya. Terbunuh kemudian diikat dengan simbol siti . Kemudian Visual Prolog mencoba lagi subgoal yang terakhir dengan variabel Terbunuh diikat pada nilai siti. Panggilannya ternodai(siti,darah). Pencarian kembali dimulai dari bagian atas program karena merupakan panggilan baru dan menemukan kecocokan dengan fakta yang ada dan subgoal terakhir sukses (true).
9. Pada titik ini, *body* dari *rule* pembunuh dapat dipenuhi. Visual Prolog mengembalikan nilai Y pada pemanggil (goal) pembunuh(X) Karena Apa diikat nilainya ke X dan X diikat nilainya budi, maka sekarang X juga diikat nilainya ke pembunuh pada goal tersebut. Visual Prolog kan menampilkan jawaban:

X=budi

1 Solution.

dan program berhenti dengan sukses.