Modul Pertemuan ke 4

CMC101 Topik Dalam Pemrograman

Sumber :

- Rinaldi Munir, Prolog

1. Pemrograman Prolog

Modul ini memberi penekanan kepada aspek pemprograman dengan menggunakan bahasa pemprograman Prolog. Disamping itu dalam modul ini juga akan dibahas tentang perbedaan bahasa Prolog dengan bahasa pemprograman konvensional seperti bahasa Pascal dan C. Dan mekanisme dasar Prolog juga dibahas melalui contoh program yang mudah.

Setelah mempelajari modul ini diharapkan dapat :

1. Mengetahui sejarah ringkas dan kegunaan bahasa pemprograman Prolog.
2. Mengetahui perbedaan di antara bahasa pemprograman Prolog dengan bahasa pemprograman konvensional.
3. Mendapat gambaran umum tentang program Prolog melalui contoh program mudah.

1.1. Sejarah Prolog

Pernahkan anda melihat atau mendengar perkataan Prolog? Prolog adalah singkatan daripada PROgramming in LOGic. Prolog merupakan satu ide yang dicetuskan pada awal 1970an untuk menggunakan logika sebagai bahasa pemprograman. Mereka yang bertanggungjawab dalam pengembangan ide ini ialah Robert Kowalski dari Edinburgh dalam aspek teori dan Colmerauer dari Marseilles dalam aspek implementasi.

Kapankah bahasa Prolog sesuai untuk digunakan? Prolog biasanya dikaitkan dengan berlogika dan merupakan bahasa pemprograman untuk perhitungan simbolik dan tak-berangka. Prolog paling sesuai untuk menyelesaikan masalah yang berkaitan dengan objek dan hubungan antara objek, masalah persamaan corak, masalah penjejakan ke belakang dan masalah yang informasinya tidak lengkap.

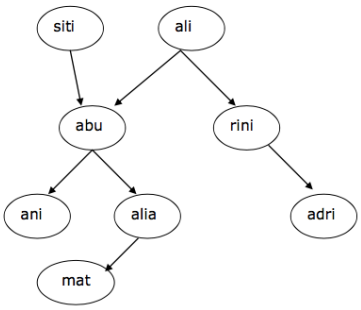
Algoritma dalam Prolog terdiri dari logika dan kontrol. Logika merupakan fakta dan peraturan yang menerangkan apa yang seharusnya dilakukan oleh algoritma. Sedangkan kontrol merupakan cara algoritma bisa diimplementasikan dengan menggunakan

peraturan. Sintaks yang dibentuk dalam Prolog adalah dalam bentuk klausa atau formula First Order Predicate Logic.

1.2. Contoh Program Mudah

Kita akan melihat satu contoh program mudah untuk mendapat gambaran bagaimana program dalam bahasa pemprograman Prolog dikodekan. Gambar 1.1 di bawah menunjukkan contoh satu hubungan keluarga. Coba bayangkan hubungan atau pertalian dalam hubungan masing-masing yang terdiri dari ibubapak, kakek, nenek, adik, akang, teteh, sepupu dan sebagainya. Fakta yang menunjukkan ali ibubapak kepada Abu bisa ditulis dalam Prolog sebagai:

ibubapak(ali,abu).



Gambar 1.1: Pohon Keluarga

Di sini kita memilih ibubapak sebagai nama hubungan: ali dan abu adalah argumen. Disebabkan oleh alasan tertentu yang akan dibicarakan kemudian, kita menulis nama seperti ali dan abu dengan menggunakan huruf kecil. Keseluruhan pohon keluarga dalam Gambar 1.1 bisa diartikan dalam program Prolog sebagai:

```
ibubapak(siti,abu).
ibubapak(ali,abu).
ibubapak(ali,rini).
ibubapak(abu,ani).
ibubapak(abu,alia).
ibubapak(alia,mat).
ibubapak(rini,adri).
```

Setiap pernyataan, contohnya ibubapak(siti,abu) dikenali sebagai klausa dan perlu diakhiri dengan tanda titik. Jadi program ini mengandungi tujuh klausa. Setiap klausa menafsirkan satu fakta mengenai hubungan ibubapak. Sebagai contoh, ibubapak(ali,abu) merupakan satu contoh hubungan ibubapak. ibubapak dikenali sebagai predikat yang mewakili hubungan di antara elemen.

Apabila program ini dihubungkan dengan sistem Prolog, Prolog bisa diajukan dengan beberapa pertanyaan berkaitan dengan hubungan ibubapak. Sebagai contoh: Adakah Abu ibubapak kepada Alia? Pertanyaan ini bisa dihubungkan dengan sistem Prolog dengan mengetik pertanyaan di bawah pada terminal komputer dengan andaian komputer telah dilengkapi dengan penterjemah Prolog.

```
?- ibubapak(abu,alia).
```

Fakta ini telah ada dalam program. Prolog akan memberi jawaban:

```
Yes
```

Pertanyaan seterusnya mungkin:

```
?-ibubapak(rini,alia).
```

Prolog akan memberi jawaban:

```
no
```

karena program tidak menyatakan apa-apa fakta bahwa Rini merupakan ibubapak kepada Alia. Prolog juga memberi jawaban ‘tidak’ kepada pertanyaan:

```
?-ibubapak(ali,ahmad).
```

karena program tidak memiliki nama Ahmad.

Kita juga bisa menanyakan pertanyaan yang lebih menarik. Sebagai contoh: Siapa ibubapak kepada Rini?

```
?-ibubapak(X,rini).
```

Contoh pertanyaan ini agak berbeda dengan contoh pertanyaan di atas. Jawaban yang akan diperoleh bukan saja ‘ya’ atau ‘tidak’ untuk kasus ini. Prolog akan memberitahu kita apakah nilai bagi X supaya pernyataan di atas itu benar. Jadi, jawabannya ialah:

```
X = ali
```

Bagi pertanyaan siapakah anak kepada Abu atau Abu merupakan ibubapak kepada siapa? Bisa dibuat dengan Prolog sebagai:

```
?-ibubapak(abu,X).
```

Sekarang, jika kita meneliti kembali pohon keluarga pada Gambar 1.1, kita akan mendapati Abu merupakan ibubapak kepada Ani dan Alia. Dalam contoh ini terdapat lebih daripada satu penyelesaian. Kalau kita masih ingat, contoh-contoh sebelum ini hanya melibatkan satu penyelesaian saja. Ini agak berbeda dengan contoh sekarang. Apabila pertanyaan ini diajukan kepada sistem Prolog, Prolog akan memaparkan penyelesaian pertama dahulu yaitu:

```
X = ani
```

Bagi mendapatkan penyelesaian seterusnya (dalam kebanyakan implementasi Prolog, kita harus mengetik semikolon (;)) setelah penyelesaian pertama X=ani, Prolog akan memberikan penyelesaian yang kedua.

X = alia

Jika kita meminta penyelesaian lain lagi yaitu dengan mengetik semikolion (;) setelah perkataan alia, maka Prolog akan memberi jawaban 'tidak' karena semua penyelesaian telah habis.

Program kita juga bisa diajukan pertanyaan yang lebih umum: Siapa ibubapak kepada siapa? Formula lain kepada pertanyaan ini bisa ditulis dalam kalimat mudah sebagai:

Cari X dan Y supaya X adalah ibubapak kepada Y.

Ini bisa dinyatakan dalam Prolog sebagai:

?-ibubapak(X,Y).

Sekarang Prolog akan mencari semua pasangan ibubapak-anak satu per satu. Penyelesian akan dipaparkan satu per satu selagi kita memberitahu Prolog yang kita memerlukan penyelesaian lain yaitu dengan mengetik semikolon (;), sehingga semua penyelesaian akan dijumpai. Jawaban yang dipaparkan ialah:

X = siti Y = abu;

X = ali Y = abu;

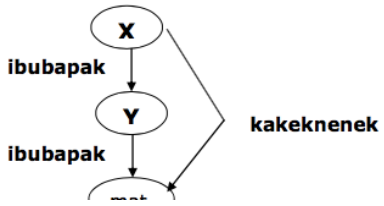
X = ali Y = rini; ....

Contoh program kita juga bisa ditanyakan dengan pertanyaan yang lebih kompleks seperti: Siapa kakek atau nenek kepada Mat? Jika kita perhatikan, program ini tidak mengetahui secara langsung tentang hubungan kakeknenek. Maka pertanyaan yang dikemukakan itu perlu dipecahkan kepada dua langkah yaitu :

(1) Siapa ibubapak kepada Mat? Andaikan Y.

(2) Siapa ibubapak kepada Y? Andaikan X.

Ini dapat digambarkan dalam Gambar 1.2 di bawah.



Gambar 1.2: Hubungan kakeknenek digambarkan sebagai gabungan dua hubungan ibubapak. Maka, pertanyaan ini ditulis dalam Prolog sebagai satu urutan yang terdiri daripada dua pertanyaan mudah:

?-ibubapak(Y,mat),ibubapak(X,Y).

Jawabannya ialah:

X = alia Y = abu

Pertanyaan yang diubah itu bisa dibaca: Cari X dan Y yang memenuhi kebutuhan di bawah:

ibubapak(Y,mat) dan ibubapak (X,Y)

Seterusnya, pertanyaan-pertanyaan lain yang bisa ditanya oleh kita ialah: Adakah Ani dan Alia mempunyai ibubapak yang sama? Ini sekali lagi bisa digambarkan dalam dua langkah:

(1) Siapa ibubapak kepada Ani? Andaikan X.

(2) Adakah X (yang sama) ibubapak kepada Alia?

Pertanyaan yang sesuai dalam Prolog ialah:

?-ibubapak(X,ani),ibubapak(X,alia).

Jawabannya ialah:

X = abu

Contoh lain penggunaan Prolog ialah bagi menerangkan dunia objek dan hubungan antara objek. Sebagai contoh, bagi penerangan dunia Aliya dan Hasan seperti di bawah.

Dunia Aliya dan Hasan

Power_Rangers adalah boneka.
Snoopy adalah boneka
Aliya bermain dengan Snoopy.
Aliya suka setiap boneka yang dia main.
Hasan suka apa yang Aliya suka.

Diterjemah dalam Prolog

```
boneka(power_rangers).
boneka(snoopy).
main(aliya, snoopy)
suka(aliya,X):- boneka(X), main(aliya,X).
suka(hasan,Y):- suka(aliya,Y)..
```

1.3. Fakta dan Peraturan

Fakta dan peraturan merupakan dua perkara yang saling berkaitan dan diperlukan apabila menulis program Prolog. Apakah fakta? Satu fakta mewakili satu unit informasi yang senantiasa dikatakan benar. Contohnya:

Langit itu biru.

Hari sedang hujan.

Sim suka epal.

Sim suka buku.

Siti suka buku.

langit(biru).

hari(hujan).

suka(sim,epal).

suka(sim,buku).

suka(siti,buku).

Sedangkan peraturan? Satu peraturan mengungkap satu hubungan di antara fakta-fakta dengan menggunakan implikasi logika :-. Peraturan mewakili satu penegasan bersyarat (conditional assertion). Kita bisa mewujudkan berbagai jenis hubungan atau predikat dalam program. Contohnya daripada fakta di atas, kita bisa mewujudkan satu hubungan kontrol berdasarkan pada pernyataan:

X dan Y adalah kawan jika Wujud satu Z dengan X suka Z dan Y juga suka Z

Coba kita perhatikan apakah yang dimaksudkan dengan X, Y dan Z ini? Sebenarnya X, Y dan Z merupakan variabel. Variabel ini bisa mewakili atau mengambil sebarang nilai. Jadi, jika kita menganggap sim = X , siti = Y, maka berdasarkan peraturan di atas dan dengan menggunakan dua fakta:

Sim suka buku suka(sim,buku). Siti suka buku suka(siti,buku).

Kita akan mendapati sistem Prolog bisa membuktikan bahwa Sim dan Siti adalah kawan karena keduanya-duanya suka kepada buku (dengan Z = buku). Jadi peraturan yang bisa dihasilkan dalam Prolog ialah:

kawan(sim,siti) :- suka(sim, buku), suka(siti,buku).

1.4. Perluasan Program Menggunakan Peraturan

Contoh program hubungan keluarga bisa diperluaskan dalam berbagai cara. Andaikan kita ingin memasukkan informasi tambahan jeniskelamin bagi setiap orang yang ada. Dalam hubungan ibubapak. Ini bisa dilakukan dengan menambah fakta berikut kepada program kita:

perempuan(siti).

lelaki(ali).

lelaki(abu).

perempuan(rini).

perempuan(ani).

perempuan(alia).

lelaki(mat).

Hubungan yang diperkenalkan di sini ialah lelaki dan perempuan. Hubungan ini merupakan hubungan unari. Hubungan binari ialah hubungan ibubapak yang menerangkan hubungan di antara sepasang objek. Klausa unari yang pertama dibaca sebagai: Siti ialah perempuan.

Perluasan yang pertama kepada program ini ialah dengan memperkenalkan hubungan anak sebagai lawan dari hubungan ibubapak. Kita bisa menakrifkan hubungan anak dengan cara yang sama seperti hubungan ibubapak yaitu dengan memasukkan fakta tentang hubungan anak. Sebagai contoh:

anak(rini,ali).

Walau bagaimanapun, hubungan anak bisa ditakrifkan dengan cara yang lebih indah yaitu dengan menggunakan fakta dalam hubungan ibubapak. Cara alternatif ini berdasarkan pada pernyatan logika di bawah:

Untuk semua X dan Y Y adalah anak kepada X jika

X adalah ibubapak kepada Y

Klausa Prolog yang seimbang dengan pernyataan logika di atas yang masih memberi makna yang sama ialah:

anak(Y,X):- ibubapak(X,Y).

Klausa ini juga bisa dibaca sebagai: Untuk semua X dan Y

Jika X adalah ibubapak kepada Y maka Y adalah anak kepada X

Terdapat perbedaan di antara fakta dan peraturan. Fakta adalah seperti ibubapak(ali,rini) yang senantiasa benar. Sebaliknya, peraturan menetapkan sesuatu perkara itu benar jika syaratnya dipenuhi. Oleh itu, kita katakan bahwa peraturan mempunyai:

bagian syarat (sebelah kanan peraturan) dan bagian kesimpulan (sebelah kiri peraturan).

Bagian kesimpulan juga disebut sebagai kepala klausa dan bagian syarat dikenali sebagai badan klausa. Sebagai contoh:

anak(Y,X):- ibubapak(X,Y).

kepala badan

Jika syarat ibubapak(X,Y) benar, maka kesan logikanya ialah anak(Y,X).

Andaikan kita menanya program kita apakah Rini anak kepada Ali:

?-anak(rini,ali).

Tidak terdapat fakta anak dalam program. Oleh itu, hanya terdapat satu cara untuk mempertimbangkan pertanyaan ini ialah dengan menggunakan peraturan tentang anak.

Peraturan ini perlulah sesuai untuk sebarang objek X dan Y, dengan itu juga sesuai untuk digunakan bagi sesuatu objek yang khusus seperti Rini dan Ali. Untuk mengimplementasikan peraturan bagi Rini dan Ali, Y perlu menggantikan Rini dan X menggantikan Ali.

X = ali dan Y = rini

Setelah penggantian, kita telah menghasilkan satu kasus khusus daripada peraturan umum. Kasus khusus ini ialah:

anak(rini,ali):- ibubapak(ali,rini).

Bagian syarat telah menjadi:

ibubapak(ali,rini).

Sekarang Prolog coba mencari apakah syarat itu betul. Jadi, tujuan awal

anak(rini,ali).

Telah diganti dengan subtujuan:

ibubapak(ali,rini).

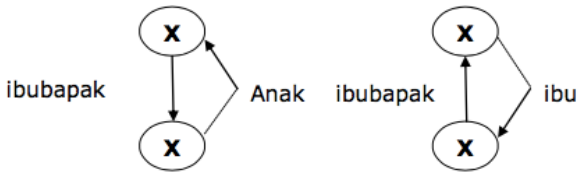
Tujuan baru ini merupakan fakta dalam program Prolog. Ini bermakna, bagian kesimpulan peraturan juga benar, dan Prolog akan memberi jawaban ‘ya’.

Sekarang, kita coba tambahkan hubungan keluarga kepada program. Spesifikasi untuk hubungan ibu bisa dihasilkan berdasarkan pernyataan logika di bawah:

Untuk semua X dan Y, X adalah ibu kepada Y jika X adalah ibubapak kepada Y dan X adalah perempuan.

Ini bisa diterjemahkan kepda Prolog sebagai peraturan sepertii dibawah:

ibu(X,Y) :- ibubapak(X,Y), perempuan(X).



Gambar 1.3: Graf untuk hubungan ibubapak, anak dan ibu

Nod dalam gambar ini menunjukkan objek yaitu argumen hubungan. Lengkuk di antara nod menunjukkan hubungan binari. Juga menunjukkan titik dari argumen pertama ke argumen kedua.

1.5. Makna Deklaratif dan Prosedur bagi Program

Dalam contoh program sebelum ini, yaitu program hubungan keluarga, kita mendapati penyelesaian diperoleh dengan mudah tanpa perlu memahami bagaimana sebenarnya

sistem mendapatkan penyelesaian tersebut. Oleh itu, bagi menangani keadaan ini, anda perlu untuk membedakan dua tingkat makna dalam program Prolog yaitu:

makna deklaratif makna prosedur

Makna deklaratif hanya menumpukan kepada hubungan yang ditakrifkan dalam program. Makna deklaratif akan menentukan apa yang akan menjadi output kepada program. Sebaliknya, makna prosedur menentukan bagaimana output itu didapat yaitu bagaimana sebenarnya hubungan itu dinilai oleh sistem Prolog.

Prolog membenarkan pemprogram mempertimbangkan makna deklaratif tanpa tergantung pada makna prosedur. Pendekatan deklaratif membuatkan pemprograman dalam Prolog lebih mudah dibandingkan dengan bahasa pemprograman berorientasikan prosedur seperti Pascal. Walau bagaimanapun, pendekatan deklaratif tidak selalu mencukupi. Bagi program yang besar, aspek prosedur tidak bisa secara langsung diabaikan oleh pemprogram dikarenakan bahasa prosedur memiliki kecekapan dalam perlaksanaan pemrograman.