BAB 3
MENGGUNAKAN PROGRAM PROLOG

3.1 Pengenalan

Bab ini akan membahas software dan mengekodekan program Prolog. Contoh-contoh program mudah juga dikemukakan untuk membantu memberikan kefahaman dari mulai program yang ditulis, proses pertanyaan bisa dilakukan dengan memasukkan tujuan yang seimbang terhadap fakta dan aturan. Prolog seterusnya akan menilai pertanyaan yang dibuat oleh pengguna untuk mendapatkan output.

3.2 Objektif

Di akhir bab ini anda dapat:
1. Menjelaskan tentang penulisan program Prolog.
2. Mempelajari jenis-jenis pertanyaan dalam Prolog.
3. Mengetahui bagaimana Prolog menilai suatu pertanyaan.

3.3 Kebutuhan Software

Sebelum kita memulai program Prolog, setiap kita perlu dilengkapi dengan software Prolog. Dalam modul ini, kita akan menggunakan software swi-prolog versi 5.1 dan software Notepad sebagai file dukungan terhadap software swi-prolog. Software Notepad akan digunakan sebagai file penyunting. Program Prolog akan dikodekan pada Notepad dan kemudiannya akan diterjemah dan di-run-kan pada software swi-prolog. Software swi-prolog versi 5.1 ini merupakan salah satu contoh penterjemah Prolog yang bias diperoleh secara gratis dari internet.

3.4 Penulisan Program Prolog

Seperti yang telah dibicarakan dalam Bab sebelumnya, program Prolog mengandung klausa. Klausa bisa terdiri daripada fakta dan peraturan. Fakta akan senantiasa benar. Peraturan pula akan benar berdasarkan kepada syarat yang diberi. Klausa mengandung kepala dan badan. Badan perlu dibuat supaya klausa adalah benar.

Fakta mempunyai kepala tetapi tidak mempunyai badan. Manakala peraturan memiliki kepala yang disebut sebagai kesimpulan (gol) dan

badan yang dikenali sebagai syarat atau sub-gol. Badan dalam peraturan ini dipisahkan dengan tanda koma. Jika salah satu sub-gol adalah palsu maka kesemua gol adalah palsu.
Apabila satu sistem prolog dimulai, maka tanda berikut akan muncul pada skrin:

?-

Seterusnya kita membuat satu program Prolog dengan menggunakan software Notepad sebagai software penyunting. Andaikan program yang hendak ditulis seperti berikut: Program di atas perlu disimpan dan diberi nama file. Andaikan ia disimpan dalam direktori c dan diberi nama program1. Kita seterusnya perlu me-run-kan program tersebut menggunakan software swi-prolog dengan mengetik:

?- consult('c:program1').
yes

Penterjemah Prolog akan berinteraktif dengan menyatakan *yes* bahwa file telah diakses dan jika tidak terdapat komentar pada program. Seandainya terdapat komentar apakah komentar sintaks atau komentar logika, software Prolog akan menyatakan *no* pada skrin.

Untuk menulis program secara langsung, satu file perlu dirujuk mewakili kata kunci.
Contohnya:

?- consult(user).
boneka(barbie).
boneka (tubies).
main(farzanah, tubies).
suka(farzanah, X):-boneka(X), main(farzanah, X).
suka(suzi,Y):-suka(farzanah,Y).
<ctrl-z><enter>
yes

Setelah satu program telah dibuat, pengguna bisa menanyakan beberapa pertanyaan *(queries)* dengan memasukkan gol yang sesuai. Bagian seterusnya membicarakan pertanyaan yang bisa dilakukan dalam software Prolog.

ibubapak(ali,ahmad).
ibubapak(alia, siti).
ibubapak(abu,kassim).

3.5 Pertanyaan*(Queries)*

Dengan satu pertanyaan menimbulkan penterjemah Prolog bekerja. Pengguna bertanya satu pertanyaan dan Prolog akan mencoba menjawab pertanyaan mengikut informasi (aturan dan peraturan) yang ada.
Pertanyaan dalam Prolog adalah untuk menilai satu gol dan mencari informasi menggunakan deduksi logika. Contoh-contoh pertanyaan yang bisa diajukan bagi program di atas ialah :

Untuk mengetahui sama ada 'tubies adalah boneka'

?- boneka(tubies).

Untuk mengetahui 'apakah kesukaan farzanah ?'

?- suka(farzanah, X).
X = barbies
yes

Untuk mengetahui 'apakah kesukaan suzi ?'

?- suka(suzi, Y).
Y = barbies
yes

Bagaimana Prolog dapat mencari jawaban bagi:
1. suzi suka Y jika farzanah suka Y.
2. farzanah suka Y jika Y adalah boneka dan farzanah bermain dengan Y.
3. barbies adalah boneka dan farzanah bermain dengan barbies.

Dari keadaan di atas, kita dapatkan katakan bahwa suzi suka barbies karena suzi suka apa yang farzanah suka dan juga dinyatakan bahwa farzanah suka barbies. Maka, suzi suka barbies. Tetapi, bagaimanakah keadaan itu dikodekan dalam Prolog?

Pernyataan 'suzi suka Y jika farzanah suka Y' bisa dikodekan dalam bentuk peraturan karena pernyataan ini menunjukkan keadaan yang bersyarat.

suka(suzi,Y) :- suka(farzanah,Y).

Bagi pernyataan 'farzanah suka Y jika Y adalah boneka dan farzanah bermain dengan Y' dikodekan dalam Prolog sebagai :

suka(farzanah,Y) :- boneka(Y), bermain(farzanah,Y).

Bagi pernyataan ketiga yaitu 'barbies adalah boneka' dan 'farzanah bermain dengan barbies' bisa ditulis sebagai fakta dalam Prolog sebagai :

boneka(barbies).
bermain(farzanah,barbies).

3.5.1 Jenis Pertanyaan

Pertanyaan dalam Prolog bisa dikategorikan kepada 3 jenis yaitu:
1. Pertanyaan mencari
2. Pertanyaan pengesahan
3. Pertanyaan tindakan

Pertanyaan mencari mengandungi satu atau lebih variabel dan informasi pertanyaan ialah untuk memenuhi gol dengan menyediakan nilai kepada variabel. Contoh:

?- suka (farzanah, X).

bisa dibaca sebagai "Cari semua nilai X yang gol bagi pertanyaan *suka(farzanah,X)* adalah benar. Jawabannya adalah barbies dengan variabel X dibebankan nilai barbies untuk membuat pertanyaan di atas menjadi benar.

Pertanyaan pengesahan bertujuan mencari pengesahan apakah satu gol (tiada variabel) adalah benar atau palsu. Contoh:

?-suka(farzanah,barbies).

Pertanyaan tindakan meminta sistem mengendalikan beberapa tindakan seperti:

- mengendalikan input-output ?- consult(c:namafile).
- memberhentikan sistem ?- halt.
- memulakan kesalahan mengetik ?- trace.

3.6 Contoh Program : Pohon Keluarga

Jika kita masih ingat, contoh program pohon keluarga telah diterangkan dengan ringkas dalam Bab sebelumnya. Dalam bab ini, contoh program keluarga ini akan dibicarakan dengan lebih terperinci yang melibatkan penulisan fakta dan peraturan. Fakta yang menerangkan hubungan keluarga ialah:

ibubapak(X,Y). % X adalah ibubapak Y.

Tanda % mewakili komen, yaitu setiap baris komen mesti dimulai dengan tanda %. Prolog tidak akan menterjemahkan baris yang mengandungi tanda %. Andaikan terdapat beberapa fakta dalam program kita seperti di bawah.

```
ibubapak(aliya, amin).
ibubapak(ahmad,amin).
ibubapak(ahmad,liza).
ibubapak(amin,ana).
```

ibubapak(amin,tina).
ibubapak(tina,aris).

Berdasarkan kepada fakta-fakta di atas, kita bisa mengajukan beberapa pertanyaan seperti:

?-ibubapak(ahmad,liza). % adakah ahmad ibubapak liza
yes
?-ibubapak (ahmad, aris). % adakah ahmad ibubapak aris
no
*?-ibubapak(*X, *liza).* % siapakah ibubapak liza
X=ahmad
yes
?-ibubapak(amin, X). % siapakah anak kepada amin
X=ana ; % untuk mendapatkan jawaban lain tekan ;
X=tina
yes
?-ibubapak(X,Y). % senaraikan X dan Y dengan X adalah ibubapak Y
X=aliya
Y=amin;
X=ahmad
Y=amin;
X=ahmad
Y=liza; % dan seterusnya
?-ibubapak(X,Y),ibubapak(Y,aris). % siapakah kakek aris
X=amin
Y=tina
Yes.

Coba kita tuliskan pertanyaan atau gol untuk mendapatkan penyelesaian kepada 'siapakah cucu ahmad'. Kodekan semua fakta di atas ke dalam Notepad dan seterusnya run-kan program tersebut dengan menggunakan penterjemah swi-prolog dan ajukan semua contoh pertanyaan di atas serta bandingkan jawaban yang diberikan oleh penterjemah Prolog dengan jawaban kita.

Berdasarkan kepada fakta ibubapak, kita juga bisa memperkenalkan hubungan seperti mamah, ayah, teteh, akang dan kakek/nenek dengan menggunakan peraturan karena kita tahu bahwa program Prolog terdiri daripada klausa dan klausa ini pula terdiri daripada fakta dan peraturan.

Andaikan kita ingin menambahkan fakta jeniskelamin seperti di bawah kepada contoh program pohon keluarga yang dibicarakan sebelum ini.

perempuan(aliya).
perempuan(liza).
perempuan(ana).
lelaki(ahmad).
lelaki(amin).
lelaki(aris).

Jadi, hubungan seperti mamah, ayah, teteh, akang, kakek/nenek bisa dibuat dengan menggunakan peraturan. Silahkan perhatikan contoh-contoh peraturan di bawah.

1. mamah(X,Y):-ibubapak(X,Y), perempuan(X).

Untuk semua X dan Y,
X adalah mamah kepada Y jika
X adalah ibubapak Y dan X adalah perempuan

?-mamah(aliya,X).
X=amin
yes

2. ayah(X,Y):-ibubapak(X,Y),lelaki(X).

Untuk semua X dan Y,
X adalah ayah kepada Y jika
X adalah ibubapak Y dan X adalah lelaki

?-ayah(ahmad,X).
X=amin;
X =liza
yes.

3. akang(X,Y):-ibubapak(Z,X),ibubapak(Z,Y), lelaki(X).

Untuk semua X dan Y,
X adalah akang kepada Y jika
Z adalah ibubapak kepada X dan
Z adalah ibubapak kepada Y dan
X adalah lelaki

?-akang(ahmad,Y).
X=liza
yes

4. teteh(X,Y):-ibubapak(Z,X),ibubapak(Z,Y),perempuan(X).

Untuk semua X dan Y,
X adalah teteh kepada Y jika
Z adalah ibubapak kepada X dan
Z adalah ibubapak kepada Y dan
X adalah perempuan

?-teteh(ana,Y).
Y=tina
yes

5. kakek(X,Z):- ibubapak(X,Y),ibubapak(Y,Z),lelaki(X).

Untuk semua X dan Z,
X adalah kakek kepada Z jika
X adalah ibubapak kepada Y dan
Y adalah ibubapak kepada Z dan
X adalah lelaki

?-kakek(ahmad, Z).
Z=ana;
Z=tina
yes

6. keturunan(Y,X):-ibubapak(X,Y).

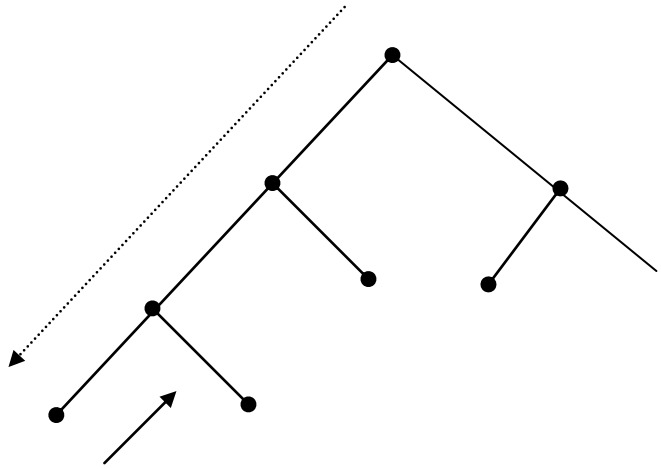
Untuk semua X dan Y,
Y adalah keturunan X jika X adalah ibubapak Y.

3.7 Bagaimana Prolog Menilai Pertanyaan

Penterjemah Prolog menilai suatu pertanyaan melalui :
1. Penyatuan Kitar Jadi-semula *(Recursive Cycle Of Unification)*
2. Penilaian sub-gol

Apabila pengguna mengetik satu pertanyaan, penterjemah Prolog mencari melalui klausa-klausa, kepala satu klausa yang bisa disatukan dengan pertanyaan. Apabila satu klausa yang dapat disatukan dengan pertanyaan ditemui, klausa tersebut diaktifkan dan setiap sub-gol dinilai dahulu. Sebaliknya, jika penterjemah Prolg gagal untuk memenuhi satu sub-gol (klausa untuk sub-gol tidak ada), prolog akan balik kembali *(backtracks)* kepada sub-gol yang berhasil pada yang terakhir sekali.

Apabila penterjemah Prolog balik kembali, itu akan membatalkan *(undo)* jalan benar yang telah dibuat oleh sub-gol. Kemudian ia akan coba pada set jalan baru. Lihatlah gambar di bawah yang menggambarkan proses balik kembali.



Perhatikan contoh program Prolog di bawah.

```
ayah(ali,wahid).
ayah(ali,abdullah).
akang(wahid,abdullah).
akang(jalil,ali).
kakek(U,V):-
        akang(U,B),
        ayah(B,V).
```

Andaikan kita ingin mengajukan pertanyaan 'jalil merupakan kakek kepada siapa' dan dikodekan pertanyaan ini dalam Prolog sebagai :

```
?-kakek(jalil,W).
```

Bagaimanakah penterjemah Prolog menyelesaikan gol ini? Perhatikan langkah-langkah penjejakan yang dilaksanakan oleh penterjemah Prolog untuk mendapatkan penyelesaian pertanyaan atau gol di atas.

Pertanyaan aktif **Klausa program**

1. *?-kakek(jalil,W).*

- Pertanyaan ini adalah di atas *'timbunan*'.
- Penterjemah mencari klausa yang padan dalam pangkalan data.

Pertanyaan aktif **Klausa program**

2. *?-kakek(jalil,W).* ⟶ *kakek(U,V):-*
   akang(U,B),
   ayah(B,V).

Penyatuan adalah wajar di mana *U = jalil dan V = W*

Pertanyaan aktif **Klausa program**

3.

?-kakek(jalil,W). ⟶ *kakek(jalil,W):-*
akang(jalil,B),
ayah(B,W).

4.

Penterjemah menilai isi klausa

- Setiap sub-gol diletak di atas *'timbuna*n' mengikut giliran dan dinilai.
- Jika semua sub-gol adalah benar, gol dipenuhi.
- Sub-gol pertama diletakkan di atas *'timbunan'.*

Pertanyaan aktif **Klausa program**

?-kakek(jalil,W). ⟶ *kakek(jalil,W):-*
akang(jalil,B),
ayah(B,W).

akang(wahid,abdullah).
?-akang(jalil,B). ⟶ *akang(jalil,ali).*

Penyatuan gagal!!
Oleh itu, pergi ke klausa akang dan coba penyatuan.
Coba sehingga semua klausa akang diperiksa.

Pertanyaan aktif **Klausa program**

5.

?-kakek(jalil,W) ⟶ *kakek(jalil,W):*
akang(jalil,B),
ayah(B,W).

akang(wahid,abdullah).
?-akang(jalil,B). ⟶ *akang(jalil,ali).*

Pertanyaan menyatukan klausa dengan *B=ali*

6.
Setiap kejadian *B* dalam aturan 'kakek' akan ditukar kepada ali.

Pertanyaan aktif		**Klausa program**
?-kakek(jalil,W)	→	*kakek(jalil,W):-* *akang(jalil,ali),* *ayah(ali,W).*
		akang(wahid,abdullah).
?-akang(jalil,ali).	→	*akang(jalil,ali).*

Sekarang, sub-gol akang berhasil.
Penterjemah kini menempatkan sub-gol kedua di atas *'timbunan'.*

Pertanyaan aktif		**Klausa program**
7.		
?-kakek(jalil,W)	→	*kakek(jalil,W):-* *akang(jalil,ali),* *ayah(ali,W).*
		akang(wahid,abdullah).
?-akang(jalil,ali).	→	*akang(jalil,ali).*
		ayah(ali,wahid).
?-ayah(ali,W).	→	*ayah(ali,abdullah).*

Sekarang, penterjemah coba menyatukan sub-gol *ayah(ali,W)* dengan klausa *ayah(ali,wahid).* Ini berhasil dengan *W=wahid*

Pertanyaan aktif		**Klausa program**
8.		
?-kakek(jalil,wahid)	→	*kakek(jalil,wahid):-* *akang(jalil,ali),* *ayah(ali,wahid).*
		akang(wahid,abdullah).
?-akang(jalil,ali).	→	*(jalil,ali).*
		ayah(ali,wahid).
?-ayah(ali,wahid).	→	*(ali,abdullah).*

Pada tahap ini, gol dipenuhi setelah mendapat jawaban pertama *W=wahid* Jika pengguna menolak jawaban tersebut dan menanyakan jawaban lain dengan mengetik ; maka *W=abdullah*

JIKA penterjemah disuruh balik kembali dan mencari jawaban baru
MAKA

- Pertanyaan aktif yang terkini akan menyebabkan kegagalan pada lokasi terkini dalam pangkalan data klausa.
- Semua jalan dijalankan melalui keberhasilan klausa yang terakhir yang *undone.*
- Penterjemah akan mulai menilai lagi pertanyaan yang sama pada klausa setelahnya.

Sub-gol ayah berada di atas 'timbunan' supaya ia digagalkan. Sekarang, dipilih klausa ayah kedua untuk menyatukan sub-gol.

	Pertanyaan aktif		**Klausa program**
9.			
	?-kakek(jalil,W)	→	*kakek(jalil,W):-* *akang(jalil,ali),* *ayah(ali,W).*
			akang(wahid,abdullah).
	?-akang(jalil,ali).	→	*akang(jalil,ali).*
			ayah(ali,wahid).
	?-ayah(ali,W).	→	*ayah(ali,abdullah).*

Sekarang, sub-gol *ayah(ali,W)* bersatu dengan klausa *ayah(ali,abdullah).* dengan *W=abdullah*

	Pertanyaan aktif		**Klausa program**
10.			
	?-kakek(jalil,abdullah)	→	*kakek(jalil,abdullah):-* *akang(jalil,ali),* *ayah(ali,abdullah).*
			akang(wahid,abdullah).
	?-akang(jalil,ali).	→	*akang(jalil,ali).*
			ayah(ali,wahid).
	?-ayah(ali,abdullah).	→	*ayah(ali,abdullah).*

Gol dipenuhi!! W=abdullah

Jika pengguna menanyakan pertanyaan yang lain,

Pertanyaan aktif **Klausa program**

11.

?-kakek(jalil,W) → *kakek(jalil,W):-*
akang(jalil,ali),
ayah(ali,W).

akang(wahid,abdullah).
akang(jalil,ali).

?-akang(jalil,ali). →

ayah(ali,wahid).
ayah(ali,abdullah).

?-ayah(ali,W). → *…….* .

Pertanyaan ayah gagal dan dibuang dari *'timbuna*n'.

Pertanyaan aktif **Klausa program**

12.

?-kakek(jalil,W) → *kakek(jalil,W):-*
akang(jalil,B),
ayah(B,W).

akang(wahid,abdullah).
akang(jalil,ali).

?-akang(jalil,B). → …….

Sub-gol akang gagal.

Pertanyaan aktif **Klausa program**

13.

→ *kakek(jalil,W):-*
akang(jalil,B),
ayah(B,W).

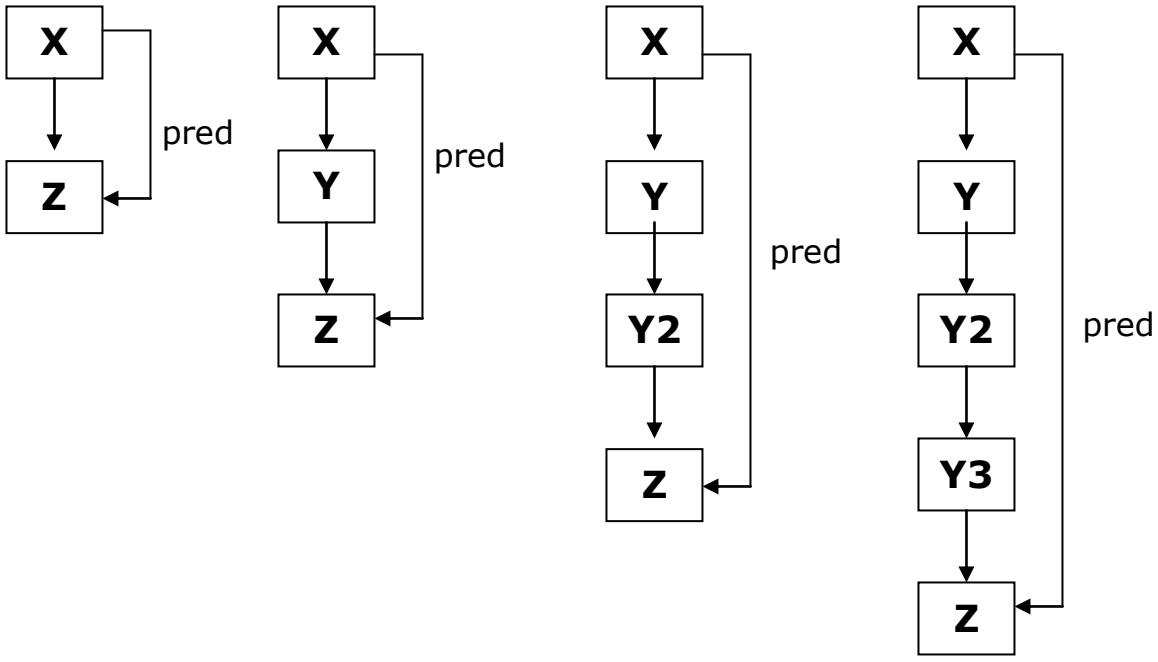
?-akang(jalil,B). → …….

Pertanyaan kakek gagal. Penterjemah membalas TIDAK.

3.8 Peraturan Jadi-Kembali *(Recursive)*

Peraturan jadi-kembalii bermaksud satu fungsi memanggil dirinya sendiri. Peraturan ini selalu digunakan dalam pemprograman Prolog. Contoh di bawah menerangkan penggunaan peraturan jadi-kembali bagi menerangkan hubungan *predecesso*r. Bagi menulis peraturan ini, terdapat dua perkara yang perlu dipertimbangkan iaiatu :

- X mungkin *predecessor* secara langsung kepada Z. Contoh, X adalah ibubapak kepada Z
- X mungkin *predecessor* secara tidak langsung kepada Z. Contohnya terdapat satu rantaian ibubapak di antara X dan Z.



Oleh itu, terdapat 2 jenis peraturan yang menerangkan perhubungan *predecesso*r.

Peraturan 1

predecessor(X,Z):-
    ibubapak(X,Z).
Untuk semua X dan Z,
    X adalah predecessor Z jika
    X adalah ibubapak Z.

Peraturan 2

```
predecessor(X,Z):-
        ibubapak(X,Y),
        predecessor(Y,Z).
Untuk semua X dan Z,
        X adalah predecessor Z jika
        terdapat Y dengan
                (1) X adalah ibubapak Y dan
                (2) Y adalah predecessor Z.
```

3.9 Kesimpulan

Sekarang anda telah mempelajari cara-cara untuk menulis dan program dan pertanyaan dalam Prolog. Kita akan lebih memahami program Prolog jika membuat program sendiri dengan berpedoman kepada contoh program yang disediakan dalam bab ini. Diharuskan juga untuk mencoba program jadi-kembali karena peraturan ini sangat penting dalam membuat program aplikasi kecerdasan buatan yang menggunakan bahasa pemprograman Prolog.

3.10 Latihan

Takrifkan pohon keluarga anda dan dapatkan hubungan seperti kakek, nenek, sepupu, anak sepupu, kakak-beradik dan lain-lain hubungan. Seterusnya kodekan hubungan ini ke dalam bahasa pemprograman Prolog dan ajukan gol atau pertanyaan untuk mewakili hubungan yang telah anda takrifkan.