

# CATATAN SINGKAT TENTANG BAHASA PROLOG

Pada bagian ini akan dibahas hal-hal berikut :

1. Bahasa PROLOG
2. Struktur Data (objek yang ditangani)
3. Pemroses Bahasa
4. Struktur Program PROLOG
5. Karakter yang Dikenal
6. Satuan Leksikal
7. Operator
8. Aturan Transformasi

## BAHASA PROLOG

Nama PROLOG adalah singkatan dari PROgramming in LOGic. Bahasa ini pertama kali dikembangkan oleh Alain Colmerauer dan P. Roussel di Universitas Marseilles Perancis pada tahun 1972.

Bahasa PROLOG:

- dikembangkan berdasarkan Kalkulus Predikat Orde Pertama.
- merupakan salah satu bahasa pemrograman logika.
- termasuk yang banyak digunakan dan sangat menonjol secara komersial.
- sebenarnya difokuskan pada pemrograman skala sedang, karena tidak memadai untuk pemrograman skala besar, kecuali jika tersedia komputer yang khusus dirancang untuk menjalankan PROLOG (seperti yang dilakukan di Jepang: berupa “Mesin PROLOG”).
- dikenal sebagai bahasa pemrograman yang banyak digunakan pada domain permasalahan Intelektual Buatan.

## STRUKTUR DATA (OBJEK YANG DITANGANI)

Program PROLOG terdiri dari fakta dan aturan. Sedangkan query dapat berada di dalam atau di luar program.

Struktur data PROLOG terdiri dari struktur data primitif berupa struktur data simbolik dan struktur data kompleks berupa list.

## PEMROSES BAHASA

Pemroses dari bahasa PROLOG dapat berupa pemroses:

- yang interaktif, berbentuk interpreter, sehingga query bersifat eksternal (berada di luar program)
- yang tidak interaktif, berbentuk compiler, sehingga query dapat bersifat eksternal maupun internal (berada di dalam program).

Pemroses bahasa PROLOG dilengkapi dengan **Mesin Inferensi** yang melakukan komputasi yes/no berdasarkan masukan program P dan goal berupa query Q. Mekanisme kerja Mesin Inferensi tersebut akan dijelaskan pada bagian eksekusi program PROLOG.

## STRUKTUR PROGRAM PROLOG

Program PROLOG dapat diorganisasikan dengan dua cara sebagai berikut.

Organisasi cara pertama, fakta dan aturan dikelompokkan secara **terpisah**. Seluruh fakta dikelompokkan tersendiri, demikian juga aturan dikelompokkan tersendiri, tetapi penulisannya fakta mendahului aturan. Cara ini dapat dilakukan jika tidak ada predikat yang sekaligus berbentuk fakta dan aturan.

Organisasi cara kedua, fakta dan aturan dikelompokkan secara **tidak terpisah**. Pengelompokan didasarkan hanya atas nama predikat yang sama, atau lebih luasnya: didasarkan atas sekelompok fakta dan aturan yang mewakili suatu persoalan. Tetapi dalam penulisannya tetap fakta mendahului aturan.

Cara Pertama	Cara Kedua
<pre>/* FAKTA */ /* Spesifikasi Fakta ke-1 */ .....  /* Spesifikasi Fakta ke-n */ .....  /* ATURAN */ /* Spesifikasi Aturan ke-1 */ .....  /* Spesifikasi Aturan ke-n */ .....</pre>	<pre>/* FAKTA */ /* Spesifikasi Fakta ke-1 */ ..... /* Spesifikasi Fakta ke-n */ .....  /* ATURAN */ /* Spesifikasi Aturan ke-1 */ ..... /* Spesifikasi Aturan ke-n */ .....  /* FAKTA DAN ATURAN */ /*Spesifikasi Fakta &amp; Aturan ke-1 */ ..... /*Spesifikasi Fakta &amp; Aturan ke-n */ .....</pre>

Contoh: Pengorganisasian yang baik.

Cara Pertama	Cara Kedua
PROGRAM POHON KELUARGA INGGRIS	PROGRAM ARITMATIKA
<pre> /* FAKTA */ /* pria(X) benar, jika X adalah pria */ pria(philip). pria(charles). pria(williams).  /* ayah(X,Y) benar, jika X adalah ayah dari Y */ ayah(philip,charles). ayah(charles,williams). ayah(charles,harry).  /* ATURAN */ /* anakLelaki(X,Y) benar, jika Y adalah ayah dari X dan X adalah pria dan X tidak sama dengan Y*/     anakLelaki(X,Y) :- ayah(Y,X),                         pria(X), X &lt;&gt; Y. </pre>	<pre> /* FAKTA */ /* suksesor(X,Y) benar, jika Y adalah suksesor dari X */ suksesor(0,1). suksesor(1,2). suksesor(2,3). suksesor(3,4).  /* ATURAN */ /* max2(X,Y,Z) benar, jika Z adalah nilai maksimum dari 2 bilangan X dan Y */     max2(X,Y,X) :- X &gt;= Y.     max2(X,Y,Y) :- X &lt; Y.  /* FAKTA DAN ATURAN */ /* plus(X,Y,Z) benar, jika Z adalah hasil penjumlahan dari X dan Y */ /* Basis */ plus(0,X,X). /* Rekurens */ plus(X,Y,Z) :- suksesor(V,X),                 plus(V,Y,W),                 suksesor(W,Z). </pre>

## KARAKTER YANG DIKENAL dalam Bahasa PROLOG

Terdiri dari:

1. Karakter A s/d Z
2. Angka 0 s/d 9
3. Karakter khusus: + - \* / < > = [ ] | , ; . \_ !

## SATUAN LEKSIKAL

Satuan leksikal adalah token dasar dari bahasa PROLOG, yang dibentuk dari himpunan karakter yang absah. Satu satuan leksikal harus dituliskan dalam satu baris. Satu satuan leksikal dapat dipisahkan oleh satu atau lebih karakter blank. Satuan leksikal bahasa PROLOG dapat dibedakan menjadi yang berikut ini.

SATUAN LEKSIKAL		ATURAN PENULISAN	CONTOH
<b>Komentar</b>		diawali dengan /* dan diakhiri dengan */	/* Ini adalah komentar. */
<b>Identifier</b>	Bentuk Umum	Identifier: huruf diikuti dengan satu/lebih huruf, angka, dan garis bawah. Karakter :- menyatakan penghubung logik implikasi (if-then). Karakter , menyatakan penghubung logik AND. Karakter . menyatakan akhir kalimat. Jika ingin menggunakan penghubung logik OR dalam kalimat, maka pemecahannya dengan cara membuat fakta atau aturan dengan nama predikat yang sama.	pria(philips). pria(charles). pria(williams).  anakLelaki(X,Y) :- ayah(Y,X), pria(X), X<> Y.
	Predikat	Nama-predikat harus diawali dengan huruf kecil.	pria anakLelaki
	Konstanta	Nama-konstanta harus diawali dengan huruf kecil.	philips charles williams
	Variabel	Nama-variabel harus diawali dengan huruf besar atau garis bawah _. Garis bawah digunakan untuk menyatakan variabel tak bernama, yaitu variabel yang tidak pernah terikat pada suatu nilai tertentu sehingga selalu dapat digantikan oleh nilai apapun.	X            Y Anak  anakLelaki(X,Y) :- ayah(Y,X), pria(X), X <> Y.  suka(_,kue) :- enak(kue).
<b>Literal Numerik</b>		bilangan integer atau real, dengan bilangan dasar 10 (desimal)	6 27 2.5 45.9
<b>Karakter</b>		dituliskan diantara dua tanda petik tunggal ‘	‘A’ ‘a’ ‘\$’
<b>String</b>		dituliskan diantara dua tanda petik ganda “	“A” “A adalah karakter” “williams adalah anak charles”
<b>Symbol</b>		kumpulan karakter yang diawali dengan huruf kecil dan diakhiri dengan karakter blank	A a_adalah_karakter williams_anak_charles

## OPERATOR

**Operator Aritmatika**, berbentuk fungsi.

No	Bentuk Umum	Arti	PROLOG	TURBO PROLOG
1	plus(X,Y,Z)	$Z = X + Y$	$Z \text{ is } X + Y$	$Z = X + Y$
2	minus(X,Y,Z)	$Z = X - Y$	$Z \text{ is } X - Y$	$Z = X - Y$
3	kali(X,Y,Z)	$Z = X * Y$	$Z \text{ is } X * Y$	$Z = X * Y$
4	bagi(X,Y,Z)	$Z = X / Y$	$Z \text{ is } X / Y$	$Z = X / Y$
5	div(X,Y,Z)	$Z = X \text{ div } Y$	$Z \text{ is } X \text{ div } Y$	$Z = X \text{ div } Y$
6	mod(X,Y,Z)	$Z = X \text{ mod } Y$	$Z \text{ is } X \text{ mod } Y$	$Z = X \text{ mod } Y$

**Operator Relasional**, berbentuk predikat infix.

No	Bentuk Umum	Arti	PROLOG	TURBO PROLOG
1	lt(X,Y)	$X < Y$	$X < Y$	$X < Y$
2	le(X,Y)	$X \leq Y$	$X \leq Y$	$X \leq Y$
3	gt(X,Y)	$X > Y$	$X > Y$	$X > Y$
4	ge(X,Y)	$X \geq Y$	$X \geq Y$	$X \geq Y$
5	eq(X,Y)	$X = Y$	$X == Y$	$X = Y$
6	ne(X,Y)	$X \neq Y$	$X \neq Y$	$X \neq Y$
7	asg(X,Y)	$X := Y$	$X \text{ is } Y$	$X = Y$

## ATURAN TRANSFORMASI

Aturan transformasi dari notasi deklaratif/lojik ke notasi bahasa PROLOG adalah sebagai berikut :

Notasi Lojik	Notasi PROLOG
<b><u>Spesifikasi</u></b> atau <b><u>Komentar</u></b>  { <spesifikasi predikat > }  { <komentar> }	 /* <spesifikasi-predikat> */  /* <komentar> */
<b><u>Fakta</u></b>  <nama-predikat> ( <parameter> ) .	<b>/* FAKTA */</b>  <nama-predikat> ( <parameter> ) .
<b><u>Aturan</u></b>  <nama-predikat> ( <parameter> ) ← <nama-predikat-1> ( <parameter-1> ) , <nama-predikat-2> ( <parameter-2> ) , ... <nama-predikat-n> ( <parameter-n> ) .	<b>/* ATURAN */</b>  <nama-predikat> ( <parameter> ) :- <nama-predikat-1> ( <parameter-1> ) , <nama-predikat-2> ( <parameter-2> ) , ... <nama-predikat-n> ( <parameter-n> ) .
<b><u>Query</u></b> <i>{simple query}</i> <nama-predikat ( <parameter> ) ?  <i>{conjunctive query}</i> <nama-predikat-1> ( <parameter-1> ) , <nama-predikat-2> ( <parameter-2> ) , ... <nama-predikat-n> ( <parameter-n> ) ?	<b>/* SIMPLE QUERY */</b> <nama-predikat ( <parameter> ) .  <b>/* CONJUNCTIVE QUERY */</b> <nama-predikat-1> ( <parameter-1> ) , <nama-predikat-2> ( <parameter-2> ) , ... <nama-predikat-n> ( <parameter-n> ) .

**Contoh Program 1.1:** cara penulisan fakta, aturan, dan query eksternal dalam Interpreter PROLOG (standar).

### PROGRAM POHON KELUARGA INGGRIS

```
/* FAKTA */
/* ayah(X,Y) benar, jika X adalah ayah dari Y */
ayah(philips,charles).
ayah(charles,williams).

/* pria(X) benar, jika X adalah pria */
pria(philips).
pria(charles).
pria(williams).

/* ATURAN */
/* anakLelaki(X,Y) benar, artinya X adalah anak lelaki dari Y,
   jika Y adalah ayah dari X dan X adalah pria dan X bukan Y. */
anakLelaki(X,Y) :- ayah(Y,X), pria(X).
```

### QUERY (EKSTERNAL )

```
? ayah(philips,charles)
yes

? ayah(philips,W)
yes
W=charles

? ayah(V,W)
yes
V=philips, W=charles
V=charles, W=williams

? pria(harry)
no

? pria(Z)
yes
Z=Philips
Z=Charles
Z=Williams

? anakLelaki(W,philips)
yes
W=charles
```

#### Catatan:

Pada query (eksternal):

Simbol “?” menyatakan bahwa sistem siap menerima query.

Sesudah itu, pemakai dapat mengetikkan query yang perlu dijawab oleh sistem.

Jawaban sistem dituliskan di bawah query tersebut.

### PROGRAM POHON KELUARGA INGGRIS

#### DOMAINS

dnama = symbol

#### PREDICATES

/\* FAKTA \*/

/\* ayah(X,Y) benar, jika X adalah ayah dari Y \*/

ayah(dnama,dnama)

/\* pria(X) benar, jika X adalah pria \*/

pria(dnama)

/\* ATURAN \*/

/\* anakLelaki(X,Y) benar, artinya X adalah anak lelaki dari  
Y, jika Y adalah ayah dari X dan X adalah pria. \*/

anakLelaki(dnama,dnama)

#### CLAUSES

/\* FAKTA \*/

/\* ayah(X,Y) benar, jika X adalah ayah dari Y \*/

ayah(philips,charles).

ayah(charles,williams).

ayah(charles,harry).

/\* pria(X) benar, jika X adalah pria \*/

pria(philips).

pria(charles).

pria(williams).

/\* ATURAN \*/

/\* anakLelaki(X,Y) benar, artinya X adalah anak lelaki dari  
Y, jika Y adalah ayah dari X dan X adalah pria. \*/

anakLelaki(X,Y) :- ayah(Y,X), pria(X).

#### QUERY (EKSTERNAL)

Goal: pria(andrew)

no

Goal: pria(Z)

yes

Z=philips

Z=charles

Z=williams

Goal: anakLelaki(williams,philips)

no

#### Catatan:

1. **Domains:** mendeklarasikan tipe data yang digunakan dalam program, untuk menghindari enumerasi fakta yang banyak dan ada “aturan/rumus” nya semisal integer.
2. **Predicates:** mendeklarasikan seluruh predikat (beserta parameternya) yang digunakan dalam program.
3. **Clauses:** mendefinisikan realisasi fakta dan aturan dari predikat yang dideklarasikan pada Predicates.
4. **Pada query** (eksternal):
  - Simbol GOAL menyatakan bahwa sistem siap menerima query.
  - Sesudah itu, pemakai dapat mengetikkan query yang perlu dijawab oleh sistem.
  - Jawaban sistem dituliskan di bawah query tersebut.