

BAB I

PENDAHULUAN

PENGERTIAN

Bahasa pemrograman adalah notasi yang digunakan untuk menulis program (komputer). Bahasa ini dibagi menjadi tiga tingkatan yaitu bahasa mesin, bahasa tingkat rendah dan bahasa tingkat tinggi.

Bahasa mesin (*machine language*) berupa *micro-instruction* atau *hardwire*. Programnya sangat panjang dan sulit dipahami. Di samping itu sangat tergantung pada arsitektur mesin. Keunggulannya adalah prosesnya sangat cepat dan tidak perlu interpreter atau penterjemah

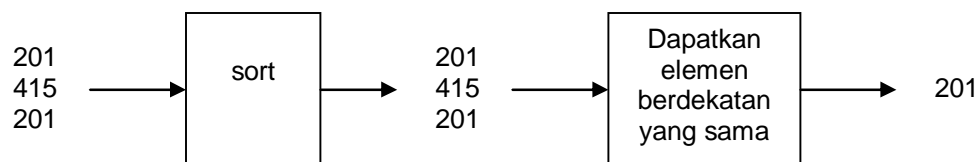
Bahasa tingkat rendah (*low level language*) berupa *macroinstruction* (*assembly*). Seperti halnya bahasa mesin, bahasa tingkat rendah tergantung pada arsitektur mesin. Programnya panjang dan sulit dipahami walaupun prosesnya cepat. Jenis bahasa tingkat ini perlu penterjemah berupa *assembler*.

Bahasa tingkat tinggi (*high level language*) menyerupai struktur bahasa manusia sehingga mudah dipahami. Bahasa ini tidak tergantung pada arsitektur mesin tetapi memerlukan penterjemah berupa *compiler* atau *interpreter*.

STRUKTUR PROGRAM

Struktur suatu program tidak bersifat unik, tetapi jawaban atau keluaran dari suatu program adalah unik. Program haruslah efisien dan mempunyai jawaban yang unik. Dalam menyusun suatu program, berbagai pendekatan dilakukan, diantaranya adalah top-down. Pendekatan top-down dalam pemrograman ini adalah dekomposisi suatu problem menjadi beberapa sub-problem sederhana yang dapat dipecahkan secara independent. Berikut adalah contoh penyusunan program menggunakan pendekatan tersebut untuk mencari elemen yang sama dari suatu list (201 415 201).

Dekomposisi pertama: urutkan data dan dapatkan elemen berdekatan yang sama.



Dekomposisi kedua:

```
for each element in the list do
  if the element appears anywhere earlier in the list
    then write out the element
for each element in the list do
  if the element is in the table
    then write out the element
  else insert the element into the table
for each element x in the list do
  if find (x)
    then write (x)
  else insert (x)
```

STRUKTUR SINTAKSIS

Sintaks suatu bahasa menunjukkan bagaimana bahasa disusun untuk membentuk suatu program. Sintaks suatu bahasa ditentukan dengan menggunakan notasi tertentu yang disebut dengan *context-free grammar* atau *simply grammar*; misalnya **BNF** (Backus-Naur Form), **EBNF** (Extended BNF), atau **syntax charts**. Berikut adalah contoh notasi dalam bentuk BNF, EBNF, dan diagram.

BNF

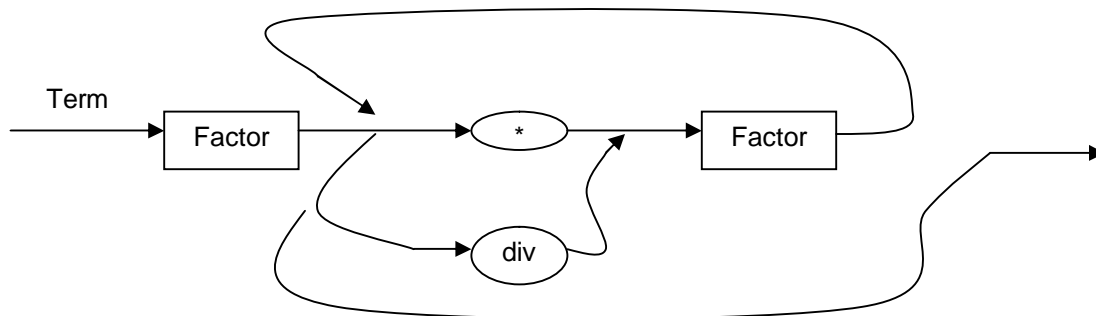
```
<Real-number> ::= <digit-sequence>.<digit-sequence>
<digit-sequence> ::= <digit> | <digit-sequence>
<digit> ::= 0|1|2|3|4|5|6|7|8|9
<expression> ::= <expression>+<term>|<expression>-<term>|<term>
<term> ::= <term>*<factor>|<term> div <factor>|<factor>
<factor> ::= (<expression>)|<variable>|<constant>
```

EBNF

```
Expression ::= term { ('+'|'-')term }
Term ::= factor { ('*'|'div')factor }
factor ::= ('('expression')'|variable|constant
catatan: {} = pengulangan
```

SYNTAX CHARTS:

```
Term ::= Factor { (* | div) Factor }
```



NOTASI EKSPRESI

Suatu ekspresi dinotasikan dalam tiga cara, yaitu infix (operator berada di antara dua operand), postfix (operator di belakang operand), dan prefix (operator di depan operand). Berikut adalah contoh notasi suatu ekspresi:

1. Infix: $(b*b) - ((4*a)*c)$
2. Postfix: $bb*4a*c^-$: $b*b-4*a*c$
3. Prefix: $- * bb**4ac$

FUNGSI

Komponen dari deklarasi fungsi adalah : nama fungsi, parameter, dan aturan untuk menyimpan hasil suatu fungsi. Sintaks suatu fungsi secara umum adalah:

```
fun<name>(<formal parameter>) <body>;
```

Berikut adalah beberapa contoh deklarasi fungsi:

```
fun successor (n) = n+1;  
fun absolute(n)=if n=0 then n else 0-n;  
fun or(x,y)=if x=true then true else y;
```

Salah satu hal penting dalam pemrograman adalah fungsi yang bersifat rekursif (*recursive*), yaitu fungsi yang memanggil dirinya sendiri. Misalnya pada perhitungan faktorial suatu bilangan, dapat dituliskan sebagai:

```
fun factorial(n)= if n=0 then 1 else n*factorial(n-1);
```

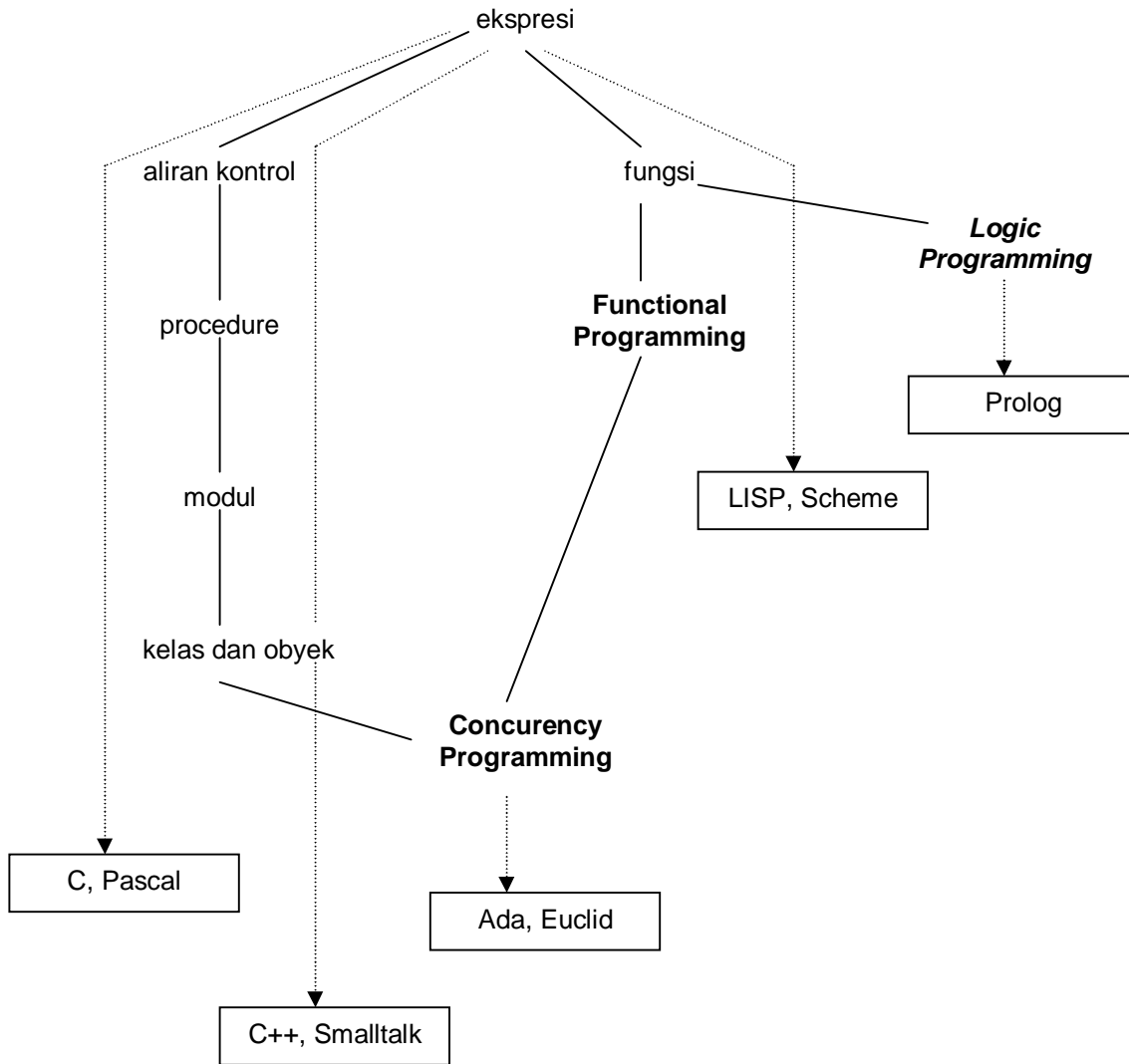
KELOMPOK BAHASA PEMROGRAMAN

Secara umum, bahasa pemrograman dapat dikelompokkan berdasarkan struktur pemrogramannya, terutama dalam mengimplementasikan suatu ekspresi. Jika setiap ekspresi selalu dinyatakan dalam bentuk fungsi, maka bahasa pemrograman ini disebut sebagai *functional programming*. Bahasa pemrograman yang termasuk dalam kelompok ini antara lain adalah **LISP** dan **Scheme**. Sedangkan jika ekspresi dinyatakan dan ditulis secara deskriptif yang diproses secara logika maka bahasa pemrograman yang digunakan disebut sebagai *logic programming*. Bahasa pemrograman **Prolog** termasuk dalam kelompok ini.

Bahasa pemrograman diproses oleh komputer berdasarkan aliran kontrolnya. Jika aliran kontrol dinyatakan secara prosedural, maka bahasa pemrograman ini dikelompokkan sebagai *programming with assignment*. Banyak sekali bahasa pemrograman yang termasuk ke dalam kelompok ini antara lain adalah **Basic**, **Pascal**, **C**, **Modula-2**, **Fortran**, dan sebagainya. Jika aliran kontrol tidak bersifat unik maka pemrograman demikian dikelompokkan sebagai *concurrency programming*. Bahasa pemrograman kelompok ini antara lain adalah **Ada** dan **Euclid**.

Kelompok lain bahasa pemrograman yang akhir-akhir ini sangat populer adalah *object oriented programming*, dimana data dan prosedur dikemas ke dalam satu obyek. **C++** dan **smalltalk** masuk ke dalam kelompok bahasa pemrograman ini. Bahasa pemrograman ini berkembang hingga muncul teknik pemrograman yang disebut sebagai

visual programming, yang merupakan metode lain untuk mengimplementasikan pemrograman berorientasi obyek. Secara garis besar, pengelompokan bahasa pemrograman dapat digambarkan sebagai berikut.



Dalam buku ini akan dibahas tiga kelompok bahasa pemrograman, yaitu *functional programming*, *logic programming*, dan *object oriented programming*.