

# Pertemuan 4

## ELEMEN-ELEMEN BAHASA PEMROGRAMAN

### I. Elemen-Elemen Dalam Bahasa Pemrograman

Berikut adalah elemen-elemen pada bahasa pemrograman:

Berikut adalah element-element pada bahasa pemrograman:

1. Aturan Leksikal
2. Type data
3. Exspression
4. Statement
5. Function dan Procesure

#### 1. Aturan Leksikal

1. Token
2. Komentar
3. Identifier
4. Keywords (*Reserved Words*)
5. Operator

##### 1. Token

Elemen tunggal pada bahasa pemrograman, dapat berupa keyword, operator, tanda baca, dll.

##### 2. Komentar

Digunakan untuk memberikan suatu keterangan yang akan menjelaskan isi dari program secara singkat. Komentar hanya sebuah tulisan pada program & tidak akan diproses oleh komputer.

##### Contoh bahasa C :

```
/* Program mencetak hello world  
Oleh: Saya */  
void main() {  
// Cetak hello World  
// Oleh Saya  
printf("Hello World");  
}
```

##### 3. Identifier (pengenal)

- Merupakan elemen nama yang banyak dipakai di dalam program.
- Aturan pemberian nama untuk identifier:
  - Karakter pertama **harus** berupa huruf
  - Karakter berikutnya dapat huruf/digit/garis bawah
  - Panjang boleh berapa saja, tetapi hanya 63 karakter pertama saja yang berarti
  - Huruf kapital/kecil dianggap sama
  - Tidak boleh ada spasi

##### 4. Keywords / Reserved Words

- Kata yang mempunyai makna khusus sebagai kata tercadang (sudah dimengerti komp.)
- Tidak boleh dipakai untuk nama label, tipe, variabel, konstanta, prosedur/fungsi, field dari record, program & unit.
- Contoh : PRINT, OR, RECORD, THEN, dll
- Huruf kapital/kecil dianggap sama

## 5. Operator

Simbol-simbol khusus yang digunakan untuk mengoperasikan suatu nilai data (operand).

### a. Arithmetic Operator

meliputi operator yang digunakan untuk melakukan perhitungan matematis (pow, \*, /, +, -, mod (%))

### b. Assignment Operator

merupakan operator penugasan yang ditandai dengan '=', '+=', '-='. Contoh : C=4 dan C+=3

### c. Bitwise Operator

digunakan untuk operasi bit dan logika.

– Negasi (NOT). Contoh : A=!B

– AND (kedua kondisi bernilai benar). Ditandai dengan simbol '&&'

– OR (jika salah satu kondisi bernilai benar). Ditandai dengan simbol '||'

### e. Relational Operator

- Digunakan untuk mewakili sebuah nilai logika dari suatu persamaan
- Operator : =, >, <, <>, >=, <=

### f. Pointer Operator

- Suatu peubah yang akan dialokasikan hanya pada saat diperlukan, yaitu setelah program dieksekusi
- Variabel pointer akan menunjukkan ke alamat memori yang digunakan untuk menampung data
- Disimbolkan dengan '\*'
- Contoh Pendeklarasian :

Contoh :

```
// more pointers
#include <iostream.h>

int main ()
{
    int value1 = 5, value2 = 15;
    int *p1, *p2;

    p1 = &value1;    // p1 = address of value1
    p2 = &value2;    // p2 = address of value2
    *p1 = 10;        // value pointed by p1 = 10
    *p2 = *p1;       // value pointed by p2 = value pointed by
p1
    p1 = p2;         // p1 = p2 (value of pointer copied)
    *p1 = 20;        // value pointed by p1 = 20

    cout << "value1==" << value1 << "/ value2==" <<
value2;
    return 0;
}
```

Output :  
value1==10 / value2==20

## 2. Tipe data

Pada suatu bahasa pemrograman umumnya telah menyediakan tipe-tipe data yang **sederhana (simple)** maupun yang **terstruktur**.

Berikut adalah tipe data sederhana (simple):

| Tipe         | Pascal                  | Bahasa C                                   | Jangkauan Nilai         |
|--------------|-------------------------|--|-------------------------|
| Integer      | byte                    | char(unsigned char)                        | 0..255                  |
|              | shortint                | char(signed char)                          | -128..127               |
|              | integer                 | Int  | -32768..32767           |
|              | word                    | short(unsigned char)                       | 0..65535                |
|              | longint                 | Long                                       | -2147483648..2147483647 |
| Real/Pecahan | real                    | float                                      | 3.4 E - 38..3.4 E +38   |
|              | double                  | double                                     | 1.7 E - 308..1.7E + 308 |
| String       | String,char(1 karakter) | -  | -                       |
| Boolean      | Boolean                 | Bool(tidak terdapat pada bahasa C standar) | True, false             |

Yang dimaksud dengan **tipe data terstruktur** yaitu tipe data yang dapat menampung lebih dari satu nilai, sebagai berikut:

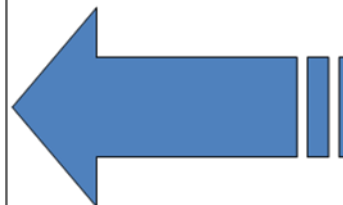
### 1. Array

Yaitu tipe data berindeks yang terdiri dari satu atau lebih elemen/komponen yang memiliki tipe data yang sama (homogen).

BU :        Namvar : ARRAY[awal..akhir] of tipe-data

Contoh : Nilai\_UTS : Array[1..100] of integer

```
/*bahasa C*/
void main()
{
int bil[2][2];
bil[0][0] = 1;
bil[0][1] = 2;
bil[1][0] = bil[0][0] + 2;
bil[1][1] = bil[0][0] + bil[1][0];
}
```



**Contoh** penggunaan array dalam pemrograman Bahasa C++

### 2. Record (Pascal) atau struct (Bahasa C)

Tipe data ini digunakan untuk merepresentasikan kumpulan (set) elemen/komponen yang memiliki satu jenis atau lebih tipe data. Tiap element disebut juga **field** atau **property** atau **attribute**.

Contoh :        Type  
                 Mahasiswa = Record  
                 Nama : String[25];  
                 NIM : String[8]

```

Alamat      : String[50];
End;
VAR
DataMhs    : Mahasiswa;

```

### 3. Expression

- Adalah rumus yang terdiri dari operasi-operasi atau kombinasi operasi.
- Aturan pendahulu:
  - Operator pendahulu yg lebih tinggi dikerjakan lebih dulu
  - Perhitungan dari kiri ke kanan
  - Ekspresi dalam tanda kurung, dikerjakan lebih dulu
- Expression yang paling sederhana yaitu nama variabel. Expression yang lebih kompleks akan melibatkan operator-operator, maupun pemanggilan function atau procedure.

### 4. Statement

#### 1. Simple Statement

a. Assignment Statement Adalah pernyataan penugasan untuk mengisi sebuah variable

| Contoh Pascal   | Contoh bahasa C  |
|---|--|
| <pre> <b>a := 10;</b> <b>b := a * 2;</b> <b>c := c * b;</b> <b>Student[i].id :=</b> <b>12;</b> </pre> | <pre> <b>a = 10;</b> <b>b = a * 2;</b> <b>c = c * b;</b> <b>Student[i].id = 12;</b> </pre> |

b. Statement untuk pemanggilan function/procedure

| Contoh dengan Bahasa C  |
|---|
| <pre> <b>Calculate (a,b);</b> <b>b = GetLength(Student[i].Name);</b> <b>Cetak (a,b);</b> </pre> |

c. Jump Statement

| Fungsi   | Pascal          | Bahasa C        |
|--|-----------------|-----------------|
| Melompat ke statement tertentu                           | <i>goto</i>     | <i>goto</i>     |
| Keluar dari iterative statement                          | <i>break</i>    | <i>break</i>    |
| Melompat ke iterasi selanjutnya pada iterative statement | <i>continue</i> | <i>continue</i> |
| Keluar dari function/procedure                           | <i>exit</i>     | <i>exit</i>     |

2. Compound Statement (kumpulan statement) adalah sekumpulan statement yang terdiri dari statement-statement lain, termasuk juga iteration dan selection statement.

|  |   |
|--|---|
| <pre>{bahasa Pascal} <b>if</b> i = 0 <b>then</b> <b>begin</b>   {compound statement   starts here}   write('Nilai i tidak   boleh nol');   hasil := -1;   {compound statement   ends here} <b>end</b>;</pre> | <pre>/*bahasa C*/ if (i = 0) { //compound statement starts here printf("Nilai i tidak boleh nol"); hasil = -1; //compound statement ends here }</pre> |
|--|---|

3. Iteration Statement  
Digunakan untuk melakukan perulangan sekumpulan statement (compound statement).

| Iteration Statement   | Bahasa Pascal   | Bahasa C   |
|---|---|--|
| Repeat/do loop: Evaluasi kondisi dilakukan di bagian akhir. | <pre><b>repeat</b> (statement); ... (statement); <b>until</b> (kondisi); Contoh: i := 1; <b>repeat</b> i := i + 2; writeln(i); <b>until</b> i &gt;= 10;</pre> | <pre><b>do</b> (statement) <b>while</b> (kondisi); Contoh: i = 1; <b>do</b> {   i = i + 2;   printf("%i\n",i); } <b>while</b> (i &lt; 10);</pre> |

## E. Function dan Procedure

Procedure dan Function disebut juga subroutine, merupakan blok statement yang dapat dipanggil dari lokasi yang berbeda di dalam program.

Yang membedakan antara function dan procedure yaitu:

suatu function jika dijalankan/dipanggil akan mengembalikan suatu nilai.

Contoh :

```
FUNCTION Hitung(Var A,B : integer) : integer;
```

```
VAR
```

```
  N : integer;
```

```
Begin
```

```
  Hitung := A + B;
```

```
End;
```

```
{ program utama }
```

```
VAR
```

```
  X, Y : integer;
```

```
BEGIN
```

```
  Write('Masukkan Nilai X = ');
```

```
  Readln(X);
```

```
  Write('Masukkan Nilai Y = ');
```

```
  Readln(Y);
```

```
  Write('Hasilnya adalah = ',Hitung(X,Y));
```

```
END.
```

Parameter Formal

Variabel Lokal

Variabel Global

Parameter Aktual

Ketika procedure atau function dipanggil, kita dapat melewatkan suatu nilai ke dalam function atau procedure tersebut. Nilai yang dilewatkan disebut juga **argument** atau **parameter** (biasanya ditulis dalam tanda kurung). Ada dua cara melewatkan nilai, yaitu: **Passing by Value (Dilewatkan secara nilai)**

- Jika di dalam pemanggilan suatu prosedur/fungsi, parameter tidak dimaksudkan untuk diubah oleh prosedur/fungsi.
- Parameter yg dilewatkan tdk dpt diubah oleh fungsi/prosedur, sekalipun perubahan telah dilakukan di dalam prosedur/fungsi
- Berguna utk melindungi parameter aktual dari efek perubahan harga yg tdk diinginkan.

## 2. Passing by Reference

- Parameter harus dilewatkan dengan cara menambahkan kata tercadang **VAR** di depan nama parameter di dalam deklarasi prosedur/fungsi

Contoh Passing by value

```
PROGRAM contoh1;
```

```
VAR A : integer;
```

```
PROCEDURE hitung(A : integer);
```

```
BEGIN
```

```
  A := A + 1;
```

```
END;
```

```
BEGIN {program utama}
```

```
  Readln(A);
```

```
  hitung(A);
```

```
  Writeln ('A = ', A);
```

```
END.
```

Contoh Passing by reference

```
PROGRAM contoh2;
```

```
VAR A : integer;
```

```
PROCEDURE hitung(VAR A : integer);
```

```
BEGIN
```

```
A := A + 1;  
END;
```

```
BEGIN {program utama}  
  Readln(A);  
  hitung(A);  
  Writeln ('A = ', A);  
END.
```

Jika Nilai A = 5  
Maka : A = 6