
Sesi/Perkuliahan ke: VII

Tujuan Instruksional Khusus :

1. Mahasiswa dapat menjelaskan pembentukan fungsi tanpa parameter.
2. Mahasiswa dapat menjelaskan mengenai pengiriman parameter dalam fungsi.
3. Mahasiswa dapat menjelaskan cara pemanggilan pada procedure tersarang.
4. Mahasiswa dapat menjelaskan cara fungsi memanggil dirinya sendiri
5. Mahasiswa dapat membuat contoh program sederhana dengan menggunakan fungsi.

Pokok Bahasan : Fungsi

Deskripsi singkat : Dalam pertemuan ini akan mempelajari tentang fungsi tanpa parameter, parameter dalam fungsi, fungsi Standart aritmatika, fungsi tersarang, fungsi Memanggil dirinya sendiri.

Referensi :

1. Anonim, "Algoritma & Pemrograman II", Penerbit Gunadarma, Jakarta, 1990
2. Bryon, Goffried, "Programming with PASCAL", Schaum Series, New York, 1986
3. Prather, Ronald E., "Problem Solving Principle : Programming with Pascal", Prentice Hall, New Jersey, 1982
4. Dumairy, Matematika Terapan untuk Bisnis & Ekonomi Press (BPFE Yogyakarta)
5. Yogiarto H.M, Turbo Pascal 5.0 Penerbit Andi Offset, Yogyakarta.
6. Ediman Lukito, Belajar Sendiri Pemrograman dengan Turbo Pascal 7.0
7. Ir. P. Insap Santosa, Turbo Pascal versi 5.0 dan 5.5, Elexmedia Komputindo

❶ FUNCTION

Blok fungsi hampir sama dengan blok prosedur, hanya fungsi harus dideklarasikan dengan tipenya atau jenis hasilnya. Tipe deklarasi ini menunjukkan tipe hasil dari fungsi.

Pada bahasa Pascal dikenal beberapa fungsi, misalkan : abs, pred, sqrt, sqr, succ dan sebagainya.

Fungsi-fungsi tersebut biasanya dikenal dengan Built in Function. Sedangkan function yang akan bicarakan disini adalah fungsi yang kita buat sendiri.

Berbeda dengan procedure, function merupakan modul program yang menghasilkan suatu kuantitas.

Hal ini dapat dilihat dari bentuk header-nya yang menyebutkan jenis data dari kuantitas yang dihasilkan.

Secara umum bentuk header suatu function adalah :

FUNCTION nama : jenis hasil;

Atau **FUNCTION nama (formal parameter : jenis) : jenis_hasil;**

Contoh :

1. Akan dibuat suatu fungsi dengan nama MAX yang dapat menentukan integer terbesar di antara dua integer.

```
Function MAX (x,y : integer) : integer;
```

```
Begin
```

```
    If x < y then MAX := y ;
```

```
        Else MAX := x;
```

```
End;
```

Selanjutnya kita dapat menggunakan fungsi di atas dalam suatu program, misalnya dengan menyatakan sebagai berikut :

```
P := MAX(a,b);  
Z := MAX(a+b,a*b);  
Q := MAX(MAX(a,b),c);  
.....  
dsb.
```

2. Function LOG (x : real) : real;

Begin

```
LOG := ln (x) / ln (10.0);
```

End;

3. Function POWER (x,y : real) : real;

Begin

```
POWER := exp (y * ln (X))
```

End;

$a^b = \text{POWER} (a,b)$

$b^a = \text{POWER} (b,a)$

$(p + q)^{r/s} = \text{POWER} (p + q, r/s)$

.....

dll

Contoh :

Function Hitung(Var A,B : integer): integer;

Begin

 Hitung := A + B;

End;

Var X,Y : integer;

Begin

 Write('Nilai X ? ');

 Readln(X);

 Write('Nilai Y ? ');

 Readln(Y);

 Writeln;

 Writeln(X, ' + ', Y, ' = ', Hitung(X,Y));

End.

Hasilnya :

 Nilai X ? 2

 Nilai Y ? 3

 2 + 3 = 5

Perbedaan fungsi dengan prosedur adalah :

1. Pada fungsi, nilai yang dikirimkan balik terdapat pada nama fungsinya (kalau pada prosedur pada parameter yang dikirimkan secara acuan). Pada contoh, nama fungsi tersebut adalah Hitung dan nilai yang dikirim balik berada pada nama fungsi tersebut. Sehingga nama fungsi ini harus digunakan untuk menampung hasil yang akan dikirimkan dari fungsi, sebagai berikut :

 Hitung := A + B;

 _____ Nama fungsi yang berisi nilai yang akan dikirimkan

-
2. Karena nilai balik berada di nama fungsi tersebut, maka fungsi tersebut dapat langsung digunakan untuk dicetak hasilnya, sebagai berikut :

```
Writeln(X, ' + ', Y, ' = ', Hitung(X, Y));
```

_____ Nama fungsi yang langsung digunakan untuk ditampilkan hasilnya.

Atau nilai fungsi tersebut dapat juga langsung dipindahkan ke pengenal variabel yang lainnya, sebagai berikut :

```
Hasil := Hitung(X, Y);
```

```
Writeln(X, ' + ', Y, ' + ', Hasil);
```

Sedang pada prosedur, nama prosedur tersebut tidak dapat digunakan langsung, yang dapat langsung digunakan adalah parameternya yang mengandung nilai balik.

REKURSIF

Suatu fungsi atau prosedur dalam bahasa Pascal dapat bersifat rekursif. Artinya, fungsi atau prosedur tersebut dapat memanggil dirinya sendiri. Berikut ini sebuah contoh fungsi dan prosedur yang rekursif.

- ```
1. function faktorial (nilai : integer) : integer;
begin
 if nilai <= 0 then faktorial := 1;
 else faktorial := nilai * faktorial (nilai-1)
end;
```

Var

```
N : integer;
```

---

Begin

Write('Berapa faktorial ? ');

Readln(N);

Writeln(N, ' faktorial = ', faktorial(N):9:0);

End.

$$\begin{aligned} \text{faktorial (4)} &= 4 * \text{faktorial (3)} \\ &\quad 3 * \text{faktorial (2)} \\ &\quad\quad 2 * \text{faktorial (1)} \\ &\quad\quad\quad 1 * \text{faktorial (0)} \\ &\quad\quad\quad\quad 1 \\ &= 4 * 3 * 2 * 1 * 1 \\ &= 24 \end{aligned}$$

2. Bilangan Fibonanci:

$F(0) = 0$

$F(1) = 1$

$F(n) = F(n-1) + F(n-2)$ ; untuk  $n > 1$

Function fibonacci ( n : integer ) : integer;

Begin

If n = 0 then fibonacci := 0

Else

If n := 1 then fibonacci := 1

Else fibonacci := fibonacci (n-1) + fibonacci (n-2);

End;

---

3. Procedure reverse ( num : integer );

Begin

    If num < 10 then write(num)

        Else begin

            Write(num mod 10);

            Reverse(num div 10);

        End;

End;

## Fungsi Standar

### 1. Fungsi standar aritmatika

- Fungsi standar **ABS**

Bentuk umum : **ABS(x)**;

Digunakan untuk memutlakkan suatu nilai yang ditunjukkan oleh argumen x.

Contoh :

Begin

    X:=-2.3;

    Write('Nilai X = ',X,' Nilai mutlaknya = ',Abs(X):3:1);

End.

- Fungsi standar **EXP**

Bentuk Umum : **EXP(x):real**;

Digunakan untuk menghitung nilai pangkat dari bilangan e yaitu sebesar  $e^x$ . Hasilnya berupa nilai real.

- 
- Fungsi standar **LN**  
Bentuk umum : **LN(x):real;**  
Digunakan untuk menghitung nilai logaritma alam (natural logarithm) dari nilai x. Hasilnya berupa nilai real.
  
  - Fungsi standar **INT**  
Bentuk umum : **INT(x:real):real;**  
Digunakan untuk menghasilkan nilai integer dari x. hasil dari fungsi adalah tipe real dengan nilai yang berupa pembulatan ke bawah (nilai pecahan dibuang) dari nilai x.  
Contoh :  
Begin  
    X:=9.99;  
    Write('Nilai yang akan dibulatkan = ',X);  
    Writeln('Nilai pembulatannya = ',Int(X):3:2);  
End.  
Hasil :  
    Nilai yang akan dibulatkan = 9.99  
    Nilai pembulatannya = 9.00
  
  - Fungsi standar **FRAC**  
Bentuk umum : **FRAC(x):real;**  
Digunakan untuk mendapatkan nilai pecahan dari argumen x. Argumen x dapat bernilai real maupun integer dan hasil dari fungsi adalah real.  
Contoh :  
Begin  
    X:=9.99;  
    Write('Nilai X = ',X,' Nilai pecahannya = ',Frac(X):4:3);  
End.  
Hasilnya :    Nilai X = 9.99 Nilai pecahannya = 0.990



- 
- Fungsi standar **SQR**

Bentuk umum : **SQR(x)**;

Digunakan untuk menghitung nilai pangkat kuadrat dari argumen x.

Contoh :

Begin

```
X :=2;
```

```
Write('Nilai X = ',X,' Nilai kuadratnya = ',sqr(x));
```

End.

Hasilnya : Nilai X = 2 Nilai kuadratnya = 4

- Fungsi standar **SQRT**

Bentuk umum : **SQRT(x) : real**;

Digunakan untuk menghitung nilai akar dari argumen x, hasilnya berupa real.

- Fungsi standar **PI, SIN, COS, ARCTAN**

## 2. Fungsi Standar Transfer

Digunakan untuk merubah suatu nilai ke bentuk nilai lain.

- Fungsi standar **CHR**

Bentuk umum : **CHR(x:byte):char**;

Digunakan untuk merubah nilai dari byte x ke bentuk karakter yang sesuai dengan kode ASCII.

Contoh :

```
X := 66;
```

```
Write('Nilai X = ',x,' Nilai karakternya = ',CHR(X));
```

Hasilnya : Nilai X = 66 Nilai karakternya = B

---

- Fungsi standar **ORD**

Bentuk umum : **ORD(x):longint;**

Digunakan untuk merubah nilai x ke bentuk nilai longint yang sesuai dengan kode ASCII, merupakan kebalikan dari fungsi CHR.

- Fungsi standar **ROUND**

Bentuk umum : **ROUND(x:real):longint;**

Digunakan untuk membulatkan nilai dari real x ke nilai longint yang terdekat. Bila nilai pecahan sama dengan atau lebih besar dari 0.5 akan dibulatkan ke atas, sedang kalau lebih kecil dari 0.5 akan dibulatkan ke bawah.

Contoh :

```
Write('10 dibagi 3 hasil pembulatan terdekat ',Round(10/3));
```

```
Writeln('20 dibagi 3 hasil pembulatan terdekat adalah
' ,Round(20/3);
```

Hasilnya :

```
10 dibagi 3 hasil pembulatan terdekat adalah 3
```

```
20 dibagi 3 hasil pembulatan terdekat adalah 7
```

- Fungsi standar **TRUNC**

Bentuk umum : **TRUNC(x:real):longint;**

Digunakan untuk membulatkan nilai dari real x ke nilai longint terkecil. Atau dengan kata lain membulatkan ke bawah.

Contoh :

```
Write('10 dibagi 3 hasil pembulatan terdekat ',Trunc(10/3));
```

```
Writeln('20 dibagi 3 hasil pembulatan terdekat adalah ',Trunc(20/3);
```

Hasilnya :

```
10 dibagi 3 hasil pembulatan terdekat adalah 3
```

```
dibagi 3 hasil pembulatan terdekat adalah 6
```

---

### 3. Fungsi Standar Lainnya

Fungsi standar yang tidak termasuk dalam kelompok pembagian di atas :

- Fungsi standar **Hi, Lo, Swap**
- Fungsi standar **Random**

Bentuk umum : **Random [(range :word)];**

Digunakan untuk menghasilkan angka random berkisar dari nilai lebih besar atau sama dengan nol dan lebih kecil dari satu. Bila range tidak disebutkan, hasil dari fungsi ini adalah real, bila range disebutkan, hasilnya adalah word.

- Fungsi standar **SizeOf**

Bentuk umum : **SizeOf(x):word;**

Digunakan untuk menunjukkan besarnya byte yang digunakan oleh suatu variabel x, hasilnya berupa nilai word.

- Fungsi standar **UPCASE**

Bentuk umum : **UpCase(Ch:char):char;**

Digunakan untuk merubah argumen suatu karakter yang ditunjukkan oleh Ch menjadi bentuk karakter huruf besar (upper case).