

## PROCEDURE DAN FUNCTION

Procedure dan Function adalah suatu program terpisah dalam blok sendiri yang berfungsi sebagai sub-program (modul program) yang merupakan sebuah program kecil untuk memproses sebagian dari pekerjaan program utama.

### ❶ PROCEDURE

Prosedur diawali dengan kata cadangan Procedure di dalam bagian deklarasi prosedur. Prosedur dipanggil dan digunakan di dalam blok program yang lainnya dengan menyebutkan judul prosedurnya.

Prosedur banyak digunakan pada program yang terstruktur, karena :

1. Merupakan penerapan konsep program modular, yaitu memecah-mecah program yang rumit menjadi program-program bagian yang lebih sederhana dalam bentuk prosedur-prosedur.
2. Untuk hal-hal yang sering dilakukan berulang-ulang, cukup dituliskan sekali saja dalam prosedur dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan.

Sebagaimana halnya sebuah program, suatu procedure juga memiliki header dan block. Perbedaan bentuknya dengan program hanyalah pada bagian header-nya saja.

Bentuk Umum header suatu procedure adalah :

**PROCEDURE nama;**

Atau

**PROCEDURE nama (formal parameter : jenis);**

Jika kita menggunakan procedure dalam suatu program, maka procedure tersebut harus dituliskan pada bagian deklarasi.

Contoh : Misal akan dibuat suatu procedure untuk menentukan bilangan bulat terbesar diantara tiga bilangan bulat, maka procedure tersebut adalah sebagai berikut :

```
PROCEDURE maksimum;  
VAR max : integer;  
BEGIN  
    IF a > b THEN max := a ELSE max := b;  
    IF c > max THEN max := c;  
    WRITELN(max);  
END.
```

Selanjutnya, di dalam suatu program, procedure ini dapat digunakan dengan bentuk penulisan sebagai berikut :

```
PROGRAM contoh_1;           _____HEADER program utama
VAR a,b,c : integer;
    PROCEDURE maksimum;
    VAR max : integer;
    BEGIN
        IF a>b THEN max := a ELSE max := b;
        IF c>max THEN max := c;
        WRITELN(max);
    END;

BEGIN
    READLN(a,b,c);
    Maksimum
END.
```

\_\_\_\_\_ deklarasi program utama

\_\_\_\_\_ statement program utama

Contoh 2 :

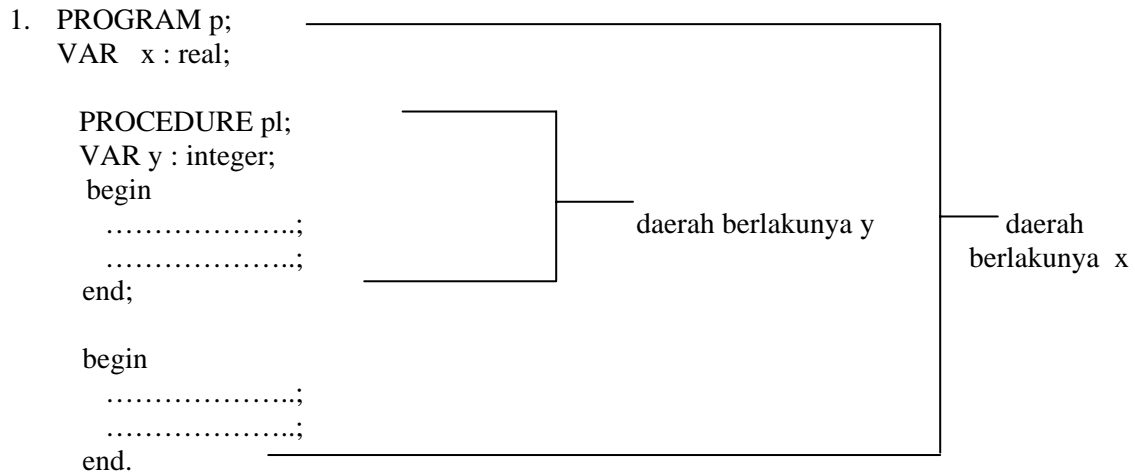
```
PROGRAM CONTOH_2;
VAR p,l,t,vol,panj : real;

PROCEDURE kotak;
VAR v,pl : real;
BEGIN
    v := p * l * t ;
    pl := p + 2.0 * t;
    writeln(v,pl);
END;
BEGIN
    writeln('panjang' : 10,'lebar' : 10,'tinggi' : 10);
    readln(p,l,t);
    kotak;
END.
```

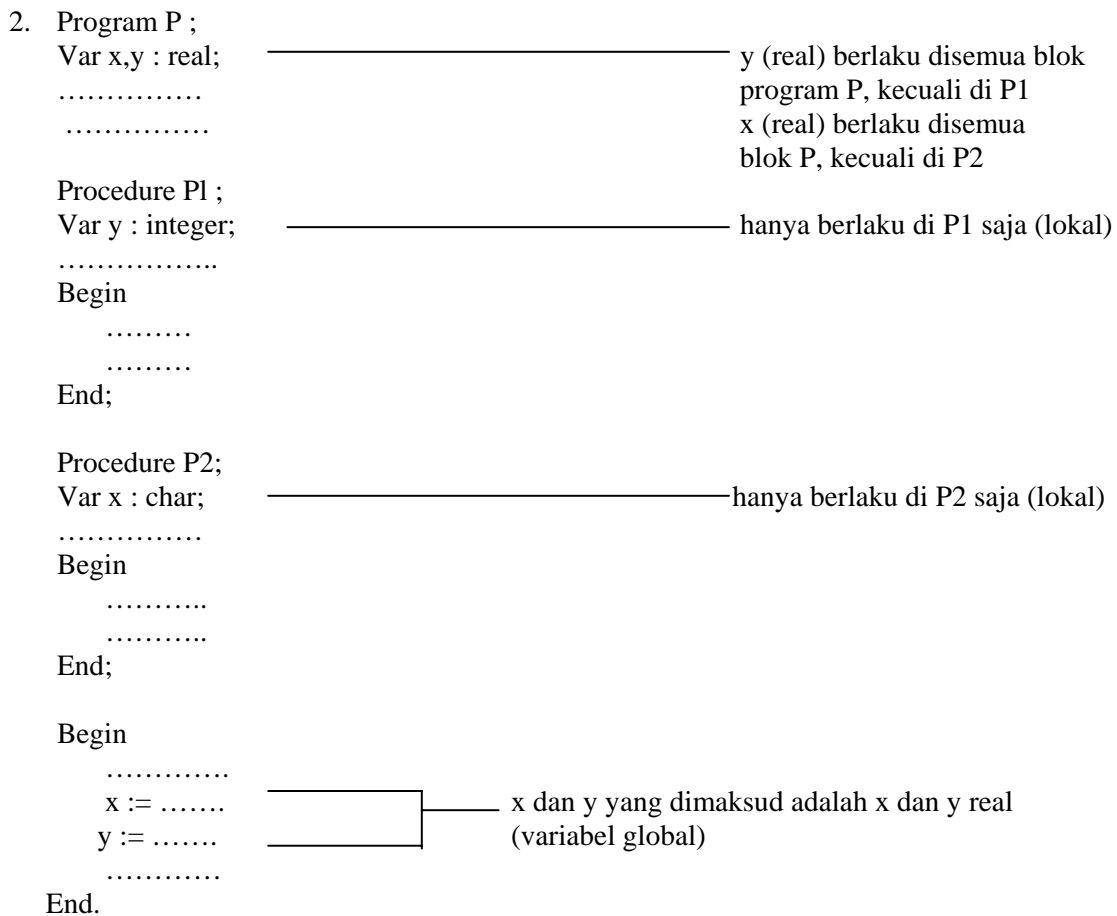
## JANGKAUAN IDENTIFIER

Identifier yang dideklarasikan dalam suatu blok program hanya berlaku pada blok dimana identifier tersebut didefinisikan.

Contoh :



y → variabel global



## PROCEDURE DENGAN PARAMETER

Nilai di dalam suatu modul program Pascal sifatnya adalah lokal, artinya hanya dapat digunakan pada modul atau unit program yang bersangkutan saja, tidak dapat digunakan pada modul atau unit program yang lainnya.

Contoh :

```
Prosedur Tanya_hitung;  
Var X,Y :real;  
Begin  
    Write ('Nilai X ?');  
    Readln(X);  
    Y:=X*X;  
    Writeln('Nilai Y = ',Y:6:2);  
End;  
  
Begin  
    Tanya_Hitung;  
End.
```

Hasilnya :  
Nilai X ? 5  
Nilai Y = 25.00

Keterangan :

Variabel X dan Y sifatnya adalah lokal untuk prosedur Tanya\_hitung, artinya hanya dapat digunakan pada modul itu saja, Pada modul yang lain tidak dapat digunakan, contoh :

```
Prosedur Tanya_hitung;  
Var X,Y :real;  
Begin  
    Write ('Nilai X ?');  
    Readln(X);  
    Y:=X*X;  
End;  
  
Begin  
    Tanya_Hitung;  
    Writeln('Nilai Y = ',Y:6:2);  
End.
```

Hasilnya :  
Error 31: Unknown identifier

Supaya nilai variabel dapat digunakan di modul lainnya, maka dapat dilakukan dengan cara :

**1. Dibuat bersifat global**

Harus dideklarasikan di atas modul yang menggunakannya.

```
Procedure kesatu;  
Begin  
    .....  
    .....  
End; (*akhir dari procedure kesatu.....*)  
  
Var  
    A,B : word;  
  
Procedure kedua;  
Begin  
    .....  
    .....  
End; (*akhir dari procedure kedua*)  
  
Procedure ketiga;  
Begin  
    .....  
    .....  
End; (*akhir dari procedure ketiga*)  
  
Begin  
    .....  
    .....  
End. (*akhir dari modul utama*)
```

Pada contoh diatas, variabel A dan B bersifat global untuk prosedur kedua, ketiga dan utama, tetapi tidak bersifat global untuk prosedur kesatu, sehingga prosedur kesatu tidak dapat menggunakan variabel-variabel tersebut.

```

Var
    A,B : real;

    Procedure kesatu;
    Begin
        .....
        .....
    End; (*akhir dari procedure kesatu.....*)

    Procedure kedua;
    Begin
        .....
        .....
    End; (*akhir dari procedure kedua*)

    Procedure ketiga;
    Begin
        .....
        .....
    End; (*akhir dari procedure ketiga*)

    Begin
        .....
        .....
    End. (*akhir dari modul utama*)
    
```

Pada contoh ini, variabel A dan B bersifat global untuk semua modul.

## 2. Dikirimkan sebagai parameter ke modul yang membutuhkannya.

Parameter yang dikirim dari modul utama ke modul prosedur disebut actual parameter, dan parameter yang ada dan dituliskan pada judul prosedur disebut formal parameter.

### 2.1 Pengiriman parameter secara nilai

Parameter yang dikirimkan secara nilai, maka parameter formal yang ada di prosedur akan berisi nilai yang dikirimkan yang kemudian bersifat lokal di prosedur.

Perhatikan procedure berikut :

```

Procedure konversi;
begin
    f := (5/9) * c + 32;  _____ f dan c variabel global, c sebagai input dan
    
```

```
writeln(c,f);          f sebagai output.  
end;
```

Procedure di atas dapat dipanggil dengan variabel yang berbeda, tetapi penulisannya harus diubah dengan menggunakan parameter sbb:

```
Procedure konversi (var f : real; c : real);  
Begin  
  F := (5/9) * c + 32;  _____ disebut formal parameter  
  Writeln(c,f);  
End;
```

Selanjutnya procedure di atas dapat dipanggil dengan parameter lain, misalnya :

```
konversi (x,y) ;  
  _____  
    actual parameter
```

x dan y disebut sebagai actual parameter.

Pada eksekusinya x akan menggantikan c dan y akan menggantikan f.

x dan y ini dapat berupa :

- konstanta
- variabel
- procedure, atau
- fungsi

Contoh :

```
Procedure Hitung(A,B : integer);  
Var C : integer;  
Begin  
  C := A + B;  
  Writeln('Nilai C = ',C)  
End;
```

```
Var X,Y : integer;  
Begin  
  Write('Nilai X ? ');  
  Readln(X);  
  Write('Nilai Y ? ');  
  Readln(Y);  
  Hitung(X,Y);  
End.
```

Hasilnya :

```
Nilai X ? 2  
Nilai Y ? 3  
Nilai C = 5
```

## 2.2 Pengiriman parameter secara acuan

--

End;

End.

$$X = 2 \quad Y = 3 \quad Z = 5$$

Digunakan untuk mendeklarasikan dimuka judul prosedur terpisah dari bloknya.

End;

End;

End.



prosedur pro1  
prosedur pro2

## Prosedur Standar

Prosedur yang disediakan oleh Turbo Pascal :

1. Prosedur standar **EXIT**  
Digunakan untuk keluar dari suatu blok.
2. Prosedur standar **HALT**  
Digunakan untuk menghentikan proses program baik di program bagian maupun di program utama.
3. Prosedur standar **MOVE**  
Bentuk umum : **MOVE (Var source,dest; count : word);**  
Digunakan untuk menyalin suatu blok sebanyak count byte memori dari blok dimulai byte pertama source dan disalinkan ke byte pertama dest.
4. Prosedur standar **FILLCHAR**  
Digunakan untuk mengisi sejumlah byte nilai ke dalam suatu variabel, sebagai berikut  
**FillChar(x;count :word;ch);**  
X adalah variabel yang dapat bertipe apapun yang akan diisi dengan nilai tipe ordinal Ch sebanyak count byte.

## ② 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;
```

---

```
        ab = POWER (a,b)
        ba = POWER (b,a)
        (p + q)r/s = 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 ? 1
    Nilai Y ? 3

    1 + 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 pengenalan 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.

\_\_\_\_\_ faktorial (4) = 4 \* faktorial (3)  
3 \* faktorial (2)  
2 \* faktorial (1)

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

2. Bilangan Fibonacci:

```
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**

### 1. 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

20 dibagi 3 hasil pembulatan terdekat adalah 6

### 1. 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).