

PROCEDURE DAN FUNCTION

❶ PROCEDURE

Prosedur → diawali dengan kata **Procedure**

Prosedure ini digunakan untuk aktivitas yang harus dilakukan lebih dari satu kali dan untuk menghindari penulisan teks program yang sama secara berulang kali. Prosedur ini cukup ditulis sebanyak satu kali namun dapat dipanggil dari bagian program utama.

Pada dasarnya, struktur prosedur sama dengan struktur program yang sudah kita kenal, hanya saja bagian judul (header) ditulis PROSEDURE (bukan lagi PROGRAM)

Prosedur ini dapat disebut juga sebagai SUB ROUTINE.

BU

PROCEDURE nama;

Atau

PROCEDURE nama(formal parameter:jenis);

Untuk memanggil prosedur yang telah kita buat, cukup dengan menuliskan nama dari prosedur yang bersangkutan.

```
C/;
PROCEDURE maksi mum;
VAR max : integer;
BEGIN
  IF a>b THEN max := a ELSE max := b;
  IF c>max THEN max := c;
  WRI TELN(max);
END;
```

```
C/;
PROGRAM contoh_1;
VAR a,b,c : integer;
PROCEDURE maksi mum;
VAR max : integer;
BEGIN
  IF a>b THEN max := a ELSE max := b;
  IF c>max THEN max := c;
  WRI TELN(max);
END;

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

```
C/;
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;
  wri tel n(v, pl);
END;

BEGIN
  wri tel n(' panjang' : 10, ' lebar' : 10, ' ti nggi ' : 10);
  readl n(p, l, t);
  kotak;
END.
```

Parameter dalam Prosedur:

Nilai dalam suatu prosedur sifatnya hanya lokal, artinya hanya dapat digunakan pada prosedur yang bersangkutan.

Dengan demikian dalam suatu program yang menggunakan prosedur terdapat dua istilah **variabel local** dan **variabel global**.

```
1. PROGRAM P1;
VAR x : real;

PROCEDURE pl;
VAR y : integer;
begin
  .....;
  .....;
end;

begin
  .....;
  .....;
end.
```

} daerah y

} daerah x

Var x → variabel global

Var y → variabel lokal

2. Program P2 :

Var x,y : real;

Procedure P1 ;

Var y : integer;

→ y (int) hanya berlaku di P1 saja (lokal)

Begin

.....

End;

Procedure P2;

Var x : char;

→ x (char) hanya berlaku di P2 saja (lokal)

Begin

.....

End;

Begin

.....

x := x dan y adalah x dan y real

y := (variabel global)

.....

End.

PROCEDURE DENGAN PARAMETER

Supaya nilai variabel dalam prosedur dapat digunakan dalam sub program lain atau dalam program utama, dapat dilakukan dua cara; yaitu :

1. Variabel di buat global
2. Variabel di kirim sebagai parameter ke sub program yang membutuhkan.

Contoh :

```
Procedure Tanya_hi_tung;
Var X,Y : real;
Begin
  Write ('Nilai X ?');
  Readln(X);
  Y:=X*X;
  Writeln('Nilai Y = ',Y:6:2);
End;

Begin
  Tanya_Hi_tung;
End.
```

Hasilnya :

Nilai X ? 5

Nilai Y = 25.00

C/:

```
Procedure Tanya_hi_tung;
Var X,Y : real;
Begin
  Write ('Nilai X ?');
  Readln(X);
  Y:=X*X;
End;

Begin
  Tanya_Hi_tung;
  Writeln('Nilai Y = ',Y:6:2);
End.
```

Hasilnya :

Error 31: Unknown identifier

1. Dibuat Variabel bersifat global

```
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*)
```

Variabel yang berada pada prosedur pertama tidak akan dikenal oleh program utama atau prosedur lain. Namun variabel dari program utama (A dan B) akan dikenal oleh prosedur kedua dan prosedur ketiga. Demikian pula variabel yang berada pada pros. Kedua dan ketiga tidak akan dikenal oleh program utama.

```

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*)

```

Dengan meletakkan deklarasi variabel lebih dulu dari pada deklarasi prosedur maka seluruh variabel yang dibuat (A & B) akan dikenali oleh ke tiga prosedur di bawahnya.

```

C/:
Program Prosedur1;
Var C1, C2 : Char;           → Variabel Global

    Procedure RUBAH
    Var T : char;
    Begin
        T := C1;
        C1 := C2;
        C2 := T;
    End;

Begin
    Write('Masukkan dua buah huruf');
    ReadLn (C1, C2);
    WriteLn;
    IF C1 > C2 Then RUBAH
    WriteLn ('Huruf yang anda masukkan adalah ', C1, ' dan ', C2);
End.

```

2. Dikirimkan sebagai parameter ke modul yang membutuhkannya.

- ✓ Parameter yang di kirimkan dari program utama ke sub program di sebut dengan **Parameter Nyata (actual parameter)**
- ✓ Parameter yang dituliskan pada judul prosedur disebut dengan **parameter formal (Formal parameter)**.

Jika suatu parameter akan dikirim dari program utama dan akan diterima oleh sub program, maka tipe datannya harus sama.

Dalam Pascal, parameter dapat dikirimkan secara nilai (*by value*) dan secara acuan (*by reference*)

```

c/:
Procedure konversi ;
begin
    f := (5/9) * c + 32;
    writeLn(c, f);
end;

```

→ penulisan dengan parameter

```

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

```

→ Dipanggil dengan cara :

```

konversi (x,y);
          actual parameter

```

Aturan Penting yang harus diperhatikan dalam menggunakan parameter actual dan parameter formal; yaitu :

- ✗ Jumlah parameter actual pada pemanggilan prosedur harus sama dengan dengan jumlah parameter formal pada deklarasi prosedurnya.
- ✗ Tiap parameter actual dan tipe parameter formal harus bertipe sama

2.1 Pengiriman parameter secara nilai

Bila pengiriman secara nilai, maka parameter formal pada prosedur akan berisi nilai yang dikirim dan bersifat local. Bila hail operasi dalam prosedur mengubah nilai-nilai tersebut, hal itu tidak akan mempengaruhi nilai asli pada program utama.

Contoh :

```
Procedure Hi tung1(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);
  Hi tung1(X, Y);
End.
```

Hasilnya :
Nilai X ? 2
Nilai Y ? 3
Nilai C = 5

Contoh :

```
Procedure Hi tung2(A, B, C : integer);
Begin
  C := A + B;
  A := A + 2; B := B + 4;
  Writeln;
  Writeln('Nilai A = ', A, ' Nilai B = ', B, ' Nilai C = ', C);
End;

Var X, Y, Z : integer;
Begin
  Write('MASUKKAN Nilai X, Y, Z ? ');
  Readln(X, Y, Z);
  Hi tung2(X, Y, Z);

  Writeln('Nilai X = ', X, ' Nilai Y = ', Y, ' Nilai Z = ', Z);
End.
```

Hasilnya :
MASUKKAN NILAI Nilai X, Y, Z ? 3 5 7
Nilai A = 5 Nilai B = 9 Nilai C = 8
Nilai X = 3 Nilai Y = 5 Nilai Z = 7

2.2 Pengiriman parameter secara acuan

Bila pengiriman secara acuan, maka perubahan yang terjadi pada parameter formal akan mempengaruhi nilai parameter nyata.

```
Procedure hitung(Var A,B,C : integer);
```

Menunjukkan pengiriman parameter secara acuan.

Contoh :

```
Procedure Hi tung3(Var A, B, C : integer);
Begin
  C := A + B;
End;

Var X, Y, Z : integer;
Begin
  X := 2; Y := 3;
  Hi tung3(X, Y, Z);

  Writeln('X = ', X, ' Y = ', Y, ' Z = ', Z);
End.
```

Hasilnya :
X = 2 Y = 3 Z = 5

```
Procedure Hi tung4(A, B : integer; Var C : integer);
```

```
Begin
  C := A + B;
  A := A + 2; B := B + 4;
  Writeln;
  Writeln('Nilai A = ', A, ' Nilai B = ', B, ' Nilai C = ', C);
End;

Var X, Y, Z : integer;
Begin
  Write('MASUKKAN Nilai X, Y, Z ? ');
  Readln(X, Y, Z);
  Hi tung4(X, Y, Z);

  Writeln('Nilai X = ', X, ' Nilai Y = ', Y, ' Nilai Z = ', Z);
End.
```

Hasilnya :

MASUKKAN NILAI Nilai X, Y, Z ? 3 5 7
Nilai A = 5 Nilai B = 9 Nilai C = 8
Nilai X = 3 Nilai Y = 5 Nilai Z = 8

Latihan!

Program Latih1

```
Var A, B : integer;

Procedure Hi tung5(Var A, B : integer);
Begin
  Writeln(A, B);
  A := 3; B := 4;
  Writeln(A, B);
End;

Begin
  A := 1; B := 2;
  Writeln(A, B);
  Hi tung5(A, B);
  Writeln(A, B);
End.
```

Program Latih2

```
Var A, B : integer;

Procedure Hi tung6(X : integer; Var Y : integer);
Var A, B : integer;
Begin
  Writeln(X, Y);
  A := 3; B := 4;
  Writeln(A, B);
  X := 3; Y := 4;
  Writeln(X, Y);
End;

Begin
  A := 1; B := 2;
  Writeln(A, B);
  Hi tung6(A, B);
  Writeln(A, B);
End.
```

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** → Digunakan untuk menyalin suatu blok sebanyak count byte memori dari blok dimulai byte pertama source dan disalinkan ke byte pertama dest.
Bentuk umum : **MOVE (Var source, dest; count : word);**
4. Prosedur standar **FILLCHAR** → Digunakan untuk mengisi sejumlah byte nilai ke dalam suatu variabel, sebagai berikut

FillChar(x;count :word;ch);

FUNCTION

Fungsi → diawali dengan kata **Function**

Fungsi ini merupakan suatu sub program yang akan mengembalikan sebuah nilai dengan tipe sederhana (dalam namanya).

Fungsi ini, sama dengan prosedur, cukup ditulis sebanyak satu kali namun dapat dipanggil dari bagian program utama.

Pada dasarnya, struktur prosedur sama dengan struktur program yang sudah kita kenal, hanya saja bagian judul (header) ditulis **FUNCTION**

BU :

```
FUNCTION nama : jenis hasil;
```

Atau

```
FUNCTION nama (formal parameter : jenis ) : jenis_hasil;
```

Contoh :

1. fungsi **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;
```

Contoh →

```
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, Z : integer;
Begin
  Write('Nilai X ? ');
  Readln(X);
  Write('Nilai Y ? ');
  Readln(Y);
  Z := HITUNG(X, Y);
  Writeln;
  Writeln(X, ' + ', Y, ' = ', Z);
End.

```

Hasilnya :

```

Nilai X ? 2
Nilai Y ? 3

1 + 3 = 5

```

Perbedaan fungsi dengan prosedur adalah :

1. Pada fungsi, nilai yang dikirimkan balik terdapat pada nama fungsinya pada prosedur pada parameter yang dikirimkan secara acuan
2. fungsi dapat langsung digunakan untuk dicetak hasilnya, sebagai berikut :

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

Nama fungsi yang langsung digunakan untuk ditampilkan hasilnya.

Atau

```

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

```

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

Pada dasarnya penggunaan parameter formal dan parameter acuan sama dengan pada prosedur. Tapi ada satu hal yang perlu diingat bahwa karena setiap nama fungsi akan menyimpan data, maka jenis data yang akan dihasilkan jangan sampai lupa di ikutsertakan pada setiap fungsi yang dibuat.

REKURSIF

Suatu fungsi atau prosedur dapat bersifat rekursif. → dapat memanggil dirinya sendiri.

```

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)	
1 * faktorial (0)	1
= 4 * 3 * 2 * 1 * 1	
= 24	