

PEMROGRAMAN MODULAR

Tujuan

- Praktikan dapat mengenal struktur prosedur dan fungsi di dalam Pascal
- Praktikan dapat membuat program dengan menggunakan prosedur dan fungsi

Pemrograman modular merupakan salah satu kemudahan yang diberikan oleh bahasa pemrograman Pascal terhadap programmer, karena di sini programmer dapat membagi program yang sedang dibuatnya ke dalam modul-modul program tertentu. Sehingga dapat menghindari penulisan teks program yang sama berkali-kali dan dapat juga memudahkan dalam melakukan pelacakan kesalahan dalam program yang sedang dibuatnya.

Ada 2 jenis pemrograman modular yang dibahas dalam praktikum ini, yaitu :

- PROCEDURE (Prosedur)
- FUNCTION (Fungsi)

PROCEDURE

Deklarasi :

```
Procedure NamaProcedure(DaftarParameter) ;  
    {Deklarasi}  
    Begin  
        .....  
        .....  
    End ;
```

Pemanggilan Procedure :

Suatu procedure bukan merupakan program yang berdiri sendiri sehingga tidak dapat dieksekusi secara langsung. Untuk itu suatu procedure memerlukan sebuah akses pada program utama / modul lain yang berfungsi sebagai pemanggil. Sedangkan cara pemanggilannya adalah dengan : *menuliskan nama procedurenya berikut parameternya (jika ada).*

Hal-hal yang berhubungan dengan procedure :

Ada beberapa hal yang berhubungan dengan penulisan sebuah procedure program yaitu:

- Variabel Global dan variabel Lokal

Variabel Global adalah variabel yang dapat dikenali diseluruh bagian program biasanya variabel ini merupakan variabel yang dideklarasikan pada deklarasi program umum.

Variabel Lokal adalah variabel yang dikenal pada lingkup yang lebih sempit dari pada variabel global.

- Parameter

Ada 2 parameter yang berlaku dalam sebuah procedure, yaitu :

- Parameter Aktual

Yaitu parameter yang dideklarasikan mengikuti pendeklarasian program utama

- Parameter Formal

Yaitu parameter yang dituliskan mengikuti pemanggilan procedure

Aturan antara parameter formal dan parameter aktual :

- Banyaknya parameter aktual harus sama dengan banyaknya parameter formal
- Tiap-tiap parameter aktual harus bertipe sama dengan parameter formal yang bersesuaian
- Tiap-tiap parameter aktual harus diekspresikan yang sesuai dengan parameter formal

Pengiriman parameter :

- Pengiriman parameter secara nilai (by Value)
 - Tiap-tiap parameter dalam procedure utama akan berisi nilai yang dikirim oleh parameter aktual dan bersifat lokal di procedure
 - Pengiriman nilai parameter secara by value merupakan pengiriman searah, artinya hanya dari parameter aktual ke parameter formal
 - Perubahan parameter formal tidak mempengaruhi terhadap nilai parameter aktual

Contoh :

```
procedure      Hitung(A,B,C : integer) ;
begin
    B: = A;
    C: = A + B ;
    Writeln(A,B,C) ;
end;
var           X,Y,Z : integer ;

begin
    readln(X) ; Readln(Y) ; Readln(Z);
    Hitung(X,Y) ; Writeln(X,Y,Z) ;
    readln ;
end.
```

Keterangan :

Misalkan dimasukkan nilai dari $X=5$, $Y=7$ dan $Z=15$ maka nilai ini akan dikirim ke parameter formal pada procedure yaitu A,B,C sehingga nilai awal $A=5$, $B=7$ dan $Z=15$. Setelah melalui proses maka hasil outputnya adalah bahwa nilai $X=5$, $Y=7$ dan $Z=15$. Sebab apapun yang diproses pada procedure tidak akan mempengaruhi nilai dari parameter aktual X,Y,Z .

- Pengiriman parameter secara acuan (by Reference)
 - Merupakan pengiriman dua arah yaitu dari parameter aktual ke parameter formal dan dari parameter formal ke parameter aktual sebagai hasil output dari procedure
 - Perubahan nilai pada parameter formal mempengaruhi pada parameter aktual
 - Dicitrakan dengan penggunaan deklarasi **VAR** , pada parameter procedurenya

Materi ke-4 Praktikum Algoritma dan Pemrograman kelas Matematika

Contoh :

```
Procedure      Hitung(Var  A,B,C : integer) ;
Begin
    B: = A;
    C: = A + B ;
    Writeln(A,B,C) ;
End;
Var            X,Y,Z : integer ;
Begin
    Readln(X) ; Readln(Y) ; Readln(Z);
    Hitung(X,Y,Z) ; Writeln(X,Y,Z) ;
    Readln ;
End.
```

Keterangan :

Misalkan dimasukkan nilai dari $X=5$, $Y=8$ dan $Z=15$ maka nilai ini akan dikirim ke parameter formal pada procedure yaitu A,B,C sehingga nilai awal $A=5$, $B=8$ dan $C=15$. Setelah melalui proses maka hasil outputnya adalah bahwa nilai $X=A=5$, $Y=B=A=5$ dan $Z=C=10$. Sebab apapun yang diproses pada procedure akan mempengaruhi nilai dari parameter aktual X,Y,Z .

Contoh program yang menggunakan procedure :

Materi ke-4 Praktikum Algoritma dan Pemrograman kelas Matematika

```
Program KOMBINASI;
uses crt;
var f1,f2,f3:longint;
    n,r,delta:integer;
    c: real;

Procedure facto( n:integer; var f:longint);
var i:integer;
begin
    f:=1;
    for i:=1 to n do
        f:= f * i;
end;

{=====PROGRAM UTAMA=====}
Begin
    clrscr;
    gotoxy(12,3);
    write('Kita akan menghitung nilai kombinasi dari suatu angka');
    gotoxy(19,4);
    writeln('Yang mempunyai rumus :  $rC_n = n! / (r!(n-r)!)$ ');
gotoxy(20,5);
    writeln('Nilai r harus lebih kecil/sama dengan n');
    writeln;gotoxy(28,6);
    write('Masukkan nilai n :',' ');readln(n);
    writeln;gotoxy(28,8);
    write('Masukkan nilai r :',' ');readln(r);
    facto(n,f1);
    facto(r,f2);
    delta:=(n-r);
    facto(delta,f3);
    c:= f1/(f2*f3);gotoxy(30,10);
    write('Hasilnya :',' ',c:2:0);
    readln;
end.
```

Materi ke-4 Praktikum Algoritma dan Pemrograman kelas Matematika

Hasil Run Program :

Kita akan menghitung nilai kombinasi dari suatu angka

Yang mempunyai rumus : $rCn = n!/(r!(n-r)!)$

Nilai r harus lebih kecil/sama dengan n

Masukkan nilai n : 6

Masukkan nilai r : 2

Hasilnya : 15

FUNCTION

Deklarasi :

```
Function   NamaFungsi(NamaParameter) : TipeData ;  
           {Deklarasi}  
           Begin  
           .....  
           .....  
           End ;
```

Struktur dari fungsi ini hampir sama dengan prosedur hanya saja function harus dideklarasikan beserta tipe datanya dan fungsi ini adalah mengembalikan nilai sehingga Tipe data dalam sebuah function menunjukkan tipe dari data akhir dari deklarasi sebuah fungsi.

Suatu fungsi dapat dipanggil dengan menggunakan beberapa cara, misalkan :

Nama Fungsinya adalah **Rata**, maka ia akan dapat dipanggil dengan menggunakan

```
Rata2 := Rata(vector,Ukuran) ;  
X := Y + Rata(vector,Ukuran) ;  
Write('Nilai rata-ratanya = ', rata(vector,ukuran));
```

Perbedaan prinsip antara procedure dengan function :

- Pada function nilai yang dikirim balik terdapat nama fungsinya (Karena nama fungsi digunakan untuk menampung nama fungsi yang terakhir)
- Function dapat langsung ditampilkan hasilnya. Misal : `writeln(Hitung(X,Y)) ;`

Catatan :

Untuk keterangan yang lain yang meliputi macam variabel, parameter, aturan antar parameter dan cara pengiriman parameter sama dengan yang ada pada procedure, sehingga tidak perlu dituliskan lagi.

Contoh Program yang menggunakan fungsi :

Materi ke-4 Praktikum Algoritma dan Pemrograman kelas Matematika

```
PROGRAM FAKTORIAL;  
  Uses crt;  
  Var i,N,Hsl:integer;  
  
  Function Fakto(N:integer):integer;  
  Var Fak: integer;  
  Begin  
    Fak:= 1;  
    for i:=1 to N do  
      Fak := Fak * i;  
    Fakto:= Fak;  
  end;  
Begin  
  clrscr;  
  write('Faktorial berapa yang anda ingin hitung : ');  
  readln(N);  
  writeln('Hasil perhitungan : ',Fakto(N));  
  readln;  
end.
```

Hasil Run Program :

Faktorial berapa yang anda ingin hitung : 5

Hasil perhitungan : 120

REKURSI

Salah satu keistimewaan yang dimiliki oleh bahasa pemrograman Pascal adalah bahwa pascal dapat melakukan suatu proses yang dinamakan sebagai proses *Rekursi*, yaitu proses untuk memanggil dirinya sendiri. Dalam procedure dan function proses ini bisa berarti proses berulang yang tidak diketahui kapan akan berakhirnya. Contoh paling sederhana dari proses rekursi adalah Proses mencari faktorial dari suatu.

Faktorial N = $N! = N (N-1) (N-2) \dots 3.2.1$

$N! = N (N-1)!$,untuk $N > 0$

$N! = N (N-1) (N-2)!$

Jika ditulis dalam program menjadi :

Faktorial(0) = 1 ;

Faktorial(N) = N (N-1)!

Ini merupakan sebuah hubungan rekurens yang berarti nilai suatu fungsi dengan argument tertentu dapat dihitung dengan menggunakan fungsi yang sama hanya saja dengan argument yang lebih kecil.

Contoh program :

Materi ke-4 Praktikum Algoritma dan Pemrograman kelas Matematika

```
PROGRAM FAKTORIAL;
Uses crt ;
Var i , N , Hsl : integer
Function Fakto(N:integer):integer ;
    Var Fak: integer ;
    Begin
        if (N=0) or (N=1) then
            Fakto:= 1
        else
            Fakto:= N*Fakto(N-1) ;
        end;
    Begin
        clrscr;
        write('Faktorial berapa yang anda ingin hitung : ') ;
        readln(N) ;
        writeln('Hasil perhitungan : ',Fakto(N)) ;
        readln ;
    end.
```

Hasil Run Program :

```
Faktorial berapa yang anda ingin hitung : 5
Hasil perhitungan : 120
```

Dari program diatas maka notasi Fakto(N-1) yang digunakan untuk memanggil program sebelumnya dinamakan sebagai *Pemanggil* atau *rekursi*.