

Modul

Dasar Pemrograman
TIP – FTP – UB

Mengapa Modul?

- Ketika program yang dibuat sudah terlalu panjang (ratusan bahkan puluhan ribu baris), sulit untuk memahami jalannya program secara keseluruhan
- Sehingga program tersebut lebih baik dipecah menjadi beberapa bagian (modul, subprogram atau subrutin)
- Memecah program menjadi modul-modul memudahkan dalam mencari kesalahan, memperbaiki serta membuat dokumentasi.
- Pembuatan modul di dalam Pascal dibagi dua yaitu Prosedur dan Fungsi

Prosedur dan Fungsi

- Prosedur dan Fungsi adalah suatu modul program yang terpisah dari program utama, diletakkan dalam blok tersendiri sebagai bagian dari program
- Setiap prosedur diawali dengan *Procedure*, sedangkan Fungsi diawali dengan *Function*

Prosedur dan Fungsi

Prosedur atau Fungsi banyak digunakan pada program terstruktur:

- Merupakan penerapan konsep modular, yaitu memecah program menjadi modul-modul atau subprogram-subprogram yang lebih sederhana.
- Untuk hal-hal yang sering dilakukan berulang-ulang, cukup dituliskan sekali saja dalam modul dan dapat dipanggil atau dipergunakan sewaktu-waktu bila diperlukan

Prosedur

- Prosedur memiliki struktur yang sama dengan struktur program, terdiri dari:
 - nama prosedur,
 - deklarasi-deklarasi
 - deskripsi (*statement atau aksi di dalam* prosedur)
- Semua deklarasi di dalam prosedur bersifat lokal sehingga hanya bisa digunakan oleh prosedur itu saja
- Sedangkan deklarasi di dalam program utama bersifat global sehingga dapat dikenali di seluruh bagian program

Prosedur

Struktur umum prosedur adalah sebagai berikut:

```
Program judul_program;
var
{bagian deklarasi prosedur, sifatnya global}

Procedure Nama_Prosedur; ← - - -
{Jika prosedur tidak memiliki parameter, tanda kurung tidak disertakan}
var
{bagian deklarasi prosedur, sifatnya lokal}
begin
    statement-1;
    statement-2;
    .....
    Statement-n;
end;

{program utama}
begin
    Nama_Prosedur; - - -
end.
```

Prosedur

Apakah harus selalu menggunakan prosedur.....?

Tidak, tergantung kompleksitas program.

- program untuk menampilkan kata "***Hai nama saya Joko***" di layar, maka tidak perlu menggunakan prosedur

Prosedur

Tanpa prosedur :

- seluruh *statement* dituliskan di dalam blok program utama, sehingga ketika program semakin besar maka blok program utama semakin padat

Prosedur

```
Program Persegipanjang;           {Judul Program Utama}
uses crt;
var                               {deklarasi variabel global}
  p,l:byte;                       {p :panjang, l:lebar}
  ls,kel:integer;                 {ls:luas, kel:keliling}
begin
  clrscr;
  writeln('Program Untuk Menghitung luas dan keliling persegipanjang');
  writeln;
  write('Masukan panjang : ');readln(p);
  write('Masukan lebar   : ');readln(l);
  writeln;
  ls:=p*l;                         {luas=panjang x lebar}
  Kel:=(2*p)+(2*l);                {keliling=(2 x panjang) + (2 x lebar)}

  writeln('Luas      : ',ls);
  writeln('Keliling : ',kel);
  readln;
end.
```

Prosedur

Prosedur:

- program utama hanya akan mengeksekusi prosedur-prosedur yang dipanggil, bila untuk sementara kita tidak membutuhkan sebuah prosedur, tidak usah menghapusnya cukup tidak memanggilnya di dalam blok program utama

```

Program PersegiPanjang;                               {judul Program Utama}
uses crt;                                             {deklarasi variabel global}
var                                                  {p :panjang, l:lebar}
    p,l:byte;                                       {ls:luas, kel:keliling}
    ls,kel:integer;

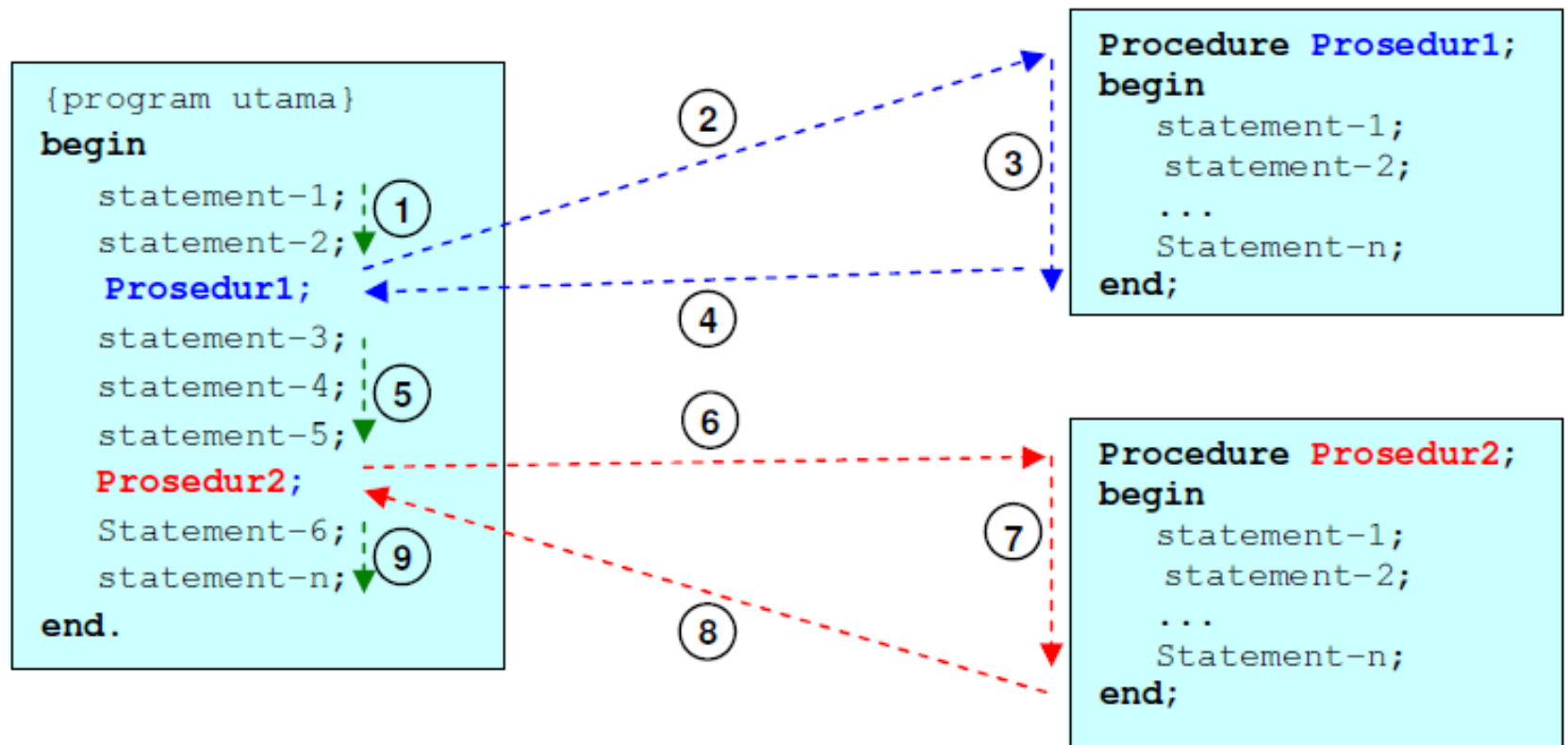
Procedure Hitung_Luas;                               {Prosedur untuk menghitung luas}
begin
    ls:=p*l;
    writeln('Luas      :',ls);
end;

Procedure Hitung_Keliling; ←                        {Prosedur untuk menghitung keliling}
begin
    kel:=(2*p)+(2*l);
    writeln('Keliling :',kel);
end;

{===program utama===}
begin
    clrscr;
    writeln('Program Untuk Menghitung luas dan keliling persegi panjang');
    writeln;
    write('Masukan panjang : ');readln(p);
    write('Masukan lebar   : ');readln(l);
    writeln;
    Hitung_Luas;                                     {Memanggil prosedur Hitung_Luas}
    Hitung_Keliling;                               {Memanggil prosedur Hitung_Keliling}
    readln;
end.

```

Prosedur



Fungsi

- Fungsi tidak jauh berbeda dengan prosedur, yaitu suatu modul program terpisah dari program utama yang diletakkan dalam blok tersendiri yang berfungsi sebagai bagian dari program
- **Namun** fungsi memiliki perbedaan mendasar dengan prosedur yaitu : fungsi memberikan nilai balik (*return*) sedangkan prosedur tidak

Fungsi

```
Function Nama_Fungsi(param_formal:tipe_param,...):tipe_hasil;  
var  
{bagian deklarasi fungsi, sifatnya lokal}  
  
begin  
    statement-1;  
    statement-2;  
    .....  
    Statement-n;  
    Nama_Fungsi:=hasil;  
end;
```

Fungsi

- Untuk mengakses setiap Fungsi kita harus memanggil namanya
- Fungsi juga dapat memiliki parameter
- Tipe hasil di dalam suatu fungsi merupakan tipe nilai balik (*return*) yang diberikan oleh fungsi

Fungsi

```
Program PersegiPanjang;
```

```
uses crt;
```

```
② Function Luas (p, l:integer ):integer;  
begin
```

```
--Luas:=p*l;
```

```
end;
```

```
{===program utama===}
```

```
var
```

```
  pj, lbr:integer;
```

```
begin
```

```
  clrscr;
```

```
  writeln('Program Untuk Menghitung luas persegi panjang');
```

```
  writeln;
```

```
  write('Masukan panjang : ');readln(pj);
```

```
  write('Masukan lebar : ');readln(lbr);
```

```
  write('Luasnya = ', Luas(pj, lbr));
```

```
  readln;
```

```
end.
```

①

③

Prosedur atau Fungsi?

Apakah suatu modul program harus dibuat menjadi fungsi atau prosedur ?

terserah anda, karena semua tergantung dari kebutuhan dan kemampuan seorang programmer

Namun fungsi digunakan bila suatu modul program mengembalikan suatu nilai sedangkan prosedur digunakan untuk menghasilkan sekumpulan aksi

Global and Local Variables

- Variabel global dideklarasikan di dalam program utama dan bisa dipanggil dalam program utama maupun dalam modul
- Variabel lokal dideklarasikan di dalam modul dan hanya dikenali dalam modul tersebut

Global and Local Variables

```
Program PersegiPanjang;           <judul Program Utama>
uses crt;                         <deklarasi variabel global>
var                                <p : panjang, l : lebar>
    p, l : byte;

Procedure Hitung_Luas;           <Prosedur untuk menghitung luas>
var                                <variabel lokal>
    ls : integer;
begin
    ls := p * l;
    writeln('Luas      : ', ls);
end;

Procedure Hitung_Keliling;       <Prosedur untuk menghitung keliling>
var                                <variabel lokal>
    kel : integer;
begin
    kel := (2 * p) + (2 * l);
    writeln('Keliling : ', kel);
end;

<===program utama===>
begin
    clrscr;
    writeln('Program Untuk Menghitung luas dan keliling persegi panjang');
    writeln;
    write('Masukan panjang : '); readln(p);
    write('Masukan lebar   : '); readln(l);
    writeln;
    Hitung_Luas;                  <Memanggil prosedur Hitung_Luas>
    Hitung_Keliling;             <Memanggil prosedur Hitung_Keliling>
    readln;
end.
```

Global and Local Variables

- **p** dan **l** adalah variabel global
- **ls** dan **kel** adalah variabel lokal

Bila variabel lokal dipanggil di luar modul tersebut maka tidak akan dikenali dan terjadi compile error

Error 3: Unknown identifier.

```
uses crt;
var
    p,l:byte;
    <deklarasi variabel global>
    <p :panjang, l:lebar>

Procedure Hitung_Luas;
var
    ls:integer;
    <Prosedur untuk menghitung luas>
    <variabel lokal>

begin
    ls:=p*l;
end;

Procedure Hitung_Keliling;
var
    kel:integer;
    <Prosedur untuk menghitung keliling>
    <variabel lokal>

begin
    kel:=(2*p)+(2*l);
end;

<===program utama===>
begin
    clrscr;
    writeln('Program Untuk Menghitung luas dan keliling persegi panjang');
    writeln;
    write('Masukan panjang : ');readln(p);
    write('Masukan lebar : ');readln(l);
    writeln;
    Hitung_Luas;
    writeln('Luas :',ls);
    <Memanggil prosedur Hitung_Luas>

    Hitung_Keliling;
    writeln('Keliling :',kel);
    <Memanggil prosedur Hitung_Keliling>

    readln;
end.
```

Parameter

- Secara sederhana parameter merupakan variabel yang dituliskan di dalam kurung setelah nama prosedur atau fungsi
- Berdasarkan penulisannya terdapat 2 jenis parameter, yaitu :
 - **parameter formal** (*formal parameter*)
parameter yang disertakan/dituliskan pada saat pendefinisian prosedur/fungsi itu sendiri
 - **parameter aktual** (*actual parameter*)
parameter yang disertakan pada saat pemanggilan prosedur/fungsi tersebut di blok program utama.

Parameter

Berdasarkan tujuannya terdapat 3 jenis parameter formal :

- Parameter masukan (*input parameter*), yaitu : parameter yang nilainya berfungsi sebagai masukan untuk prosedur/fungsi,
- Parameter keluaran (*output parameter*), yaitu : parameter yang berfungsi *untuk* menampung keluaran yang dihasilkan oleh prosedur.
- Parameter masukan/keluaran (*input/output parameter*) adalah parameter *yang* berfungsi sebagai “pintu” masuk dan keluar bagi prosedur tersebut.

```

Program PersegiPanjang;                                {judul Program Utama}
uses crt;

Procedure Hitung_Luas(p,l:integer);                    {Prosedur dengan parameter}
var
  ls:integer;                                           {variabel lokal}
begin
  ls:=p*l;
  writeln('Luas      :',ls);
end;

Procedure Hitung_Keliling(p,l:integer);              {Prosedur dengan parameter}
var
  kel:integer;                                          {variabel lokal}
begin
  kel:=(2*p)+(2*l);
  writeln('Keliling :',kel);
end;

{===program utama===}
var
  pj,lbr:integer;
begin
  clrscr;
  writeln('Program Untuk Menghitung luas dan keliling persegipanjang');
  writeln;
  write('Masukan panjang : ');readln(pj);
  write('Masukan lebar   : ');readln(lbr);
  writeln;
  Hitung_Luas(pj,lbr);                                {Memanggil prosedur Hitung_Luas}
  Hitung_Keliling(pj,lbr);                          {Memanggil prosedur Hitung_Keliling}
  readln;
end.

```


Parameter

- **pj** dan **lbr** adalah parameter aktual/nyata (*actual parameter*)
- **p** dan **l** adalah parameter formal (*formal parameter*).
- Nama parameter aktual dan parameter formal boleh berbeda, yang penting tipe datanya sama
- **p** dan **l** termasuk ke dalam parameter input, dimana **p** dan **l** berfungsi sebagai pintu masuk data bagi prosedur **Hitung_Luas** dan **Hitung_Keliling**
- Data yang masuk ke dalam parameter **p** berasal dari parameter aktual **pj**, dan data yang masuk ke dalam parameter **l** berasal dari parameter aktual **lbr**

Passing Parameter

- Proses pengiriman data dari parameter aktual ke parameter formal disebut dengan transfer parameter (*passing parameter*).
- Nama Parameter aktual dan parameter formal boleh berbeda, tetapi harus memiliki tipe data yang sama selain itu juga jumlah parameter aktual dan parameter formal harus sama
- Di dalam Pascal, parameter dapat dikirim/ditransfer secara nilai (*by value*) atau secara acuan (*by reference*).

Passing by Value

Pengiriman parameter secara nilai (*by value*) mempunyai karakteristik sebagai berikut :

- Data yang dikirim dari parameter aktual di blok program utama ke parameter formal di prosedur adalah nilai dari datanya bukan alamat memori letak dari datanya.
- Prosedur/Fungsi yang menerima nilai ini akan menyimpan data tersebut di alamat memori yang berbeda dari nilai aslinya yang digunakan oleh bagian program yang memanggil fungsi/prosedur tersebut di blok program utama.
- Karena terdapat alamat memori yang berbeda, maka perubahan nilai di fungsi tidak akan merubah nilai asli di bagian program yang memanggil prosedur/fungsi tersebut

```

1  Program PersegiPanjang;           {judul Program Utama}
2  uses crt;

3  Procedure Hitung_Luas(p, l, ls:integer);   {Prosedur dengan parameter}
4  begin
5      ls:=p*l;
6      writeln('p = ',p);
7      writeln('l = ',l);
8      writeln('ls = ',ls);
9  end;

10 {===program utama===}
11 var
12   pj,lbr,luas:integer;
13 begin
14   clrscr;
15   writeln('Program Untuk Menghitung luas persegi panjang');
16   pj:=10;
17   lbr:=6;
18   luas:=0;
19   writeln;
20   Hitung_Luas(pj,lbr,luas);           {Memanggil prosedur Hitung_Luas}
21   writeln;
22   writeln('pj = ',pj);
23   writeln('lbr = ',lbr);
24   writeln('luas = ',luas);
25   readln;
26 end.

```

Passing by Value

- Bila program dijalankan maka hasilnya seperti dibawah ini

```
Program Untuk Menghitung luas persegi panjang
p   = 10
l   = 6
ls  = 60

pj  = 10
lbr = 6
luas = 0
_
```

Passing by Reference

Pengiriman parameter secara acuan (*by reference*) mempunyai karakteristik sebagai berikut :

- Pada saat pengiriman nilai, yang dikirim dari parameter aktual di blok program utama ke parameter formal di prosedur adalah alamat letak dari nilai datanya
- Prosedur/Fungsi yang menerima nilai ini akan menggunakan alamat memori yang sama dengan nilai aslinya yang digunakan oleh bagian program yang memanggil fungsi/prosedur tersebut di blok program utama.
- Karena menggunakan alamat memori yang sama, maka perubahan nilai di prosedur/fungsi akan merubah nilai asli di bagian program yang memanggil prosedur/fungsi tersebut.

```

1  Program PersegiPanjang;           {judul Program Utama}
2  uses crt;                          ②
3  Procedure Hitung_Luas(p,l:integer; var ls:integer);{Prosedur dgn parameter}
4  begin
5  --ls:=p*l;
6     writeln('p  = ',p);
7     writeln('l  = ',l);
8     writeln('ls = ',ls);
9  end;
10 {===program utama===}
11 var
12   pj,lbr,luas:integer;
13 begin
14   clrscr;
15   writeln('Program Untuk Menghitung luas persegi panjang');
16   pj:=10;
17   lbr:=6;
18   luas:=0;
19   writeln;
20   Hitung_Luas(pj,lbr,luas);         ③ {Memanggil prosedur Hitung_Luas}
21   writeln;
22   writeln('pj   = ',pj);
23   writeln('lbr  = ',lbr);
24   writeln('luas = ',luas);
25   readln;
26 end.

```

Passing by Reference

- Bila program dijalankan maka hasilnya seperti dibawah ini

```
Program Untuk Menghitung luas persegi panjang  
p   = 10  
l   = 6  
ls  = 60  
  
pj  = 10  
lbr = 6  
luas = 60
```


Review

- Buatlah program yang terdiri dari 4 prosedur
 1. prosedur untuk menampilkan menu
 2. prosedur untuk memproses pilihan menu
 3. prosedur untuk konversi dari satuan mil ke km (1 mil = 1,609 km)
 4. prosedur untuk konversi dari satuan oz ke gram (1 oz = 28,349 gram)