



POLITEKNIK STATISTIKA STIS

For Better Official Statistics

SUB PROGRAM

Nori Wilantika, S.S.T., M.T.I.



- Di saat program kita sudah menjadi besar kita akan mengalami kesulitan dalam mengatur kode program, jika semua kode tersebut disatukan.
- Untuk mengatasi hal tersebut, kita bisa menggunakan subprogram untuk membuat program kita terbagi menjadi beberapa bagian yang masing-masing lebih kecil dan lebih mudah dikelola.
- Subprogram dibentuk dengan mengelompokkan sejumlah perintah untuk menyelesaikan tugas tertentu.
- Subprogram diperlukan jika kelompok perintah tersebut kerap kali digunakan di tempat lain dalam program.
- Pemecahan program menjadi subprogram yang lebih kecil juga akan mempermudah kita jika ingin membuat program yang serupa, kita hanya perlu menyalin, atau memakai subprogram yang sudah ada di dalam program kita yang baru

```

PROGRAM Scores1(INPUT,OUTPUT);
CONST
    NumberOfClasses = 6;
VAR
    Score :ARRAY[1..NumberOfClasses] OF REAL;
    Average, SumOfScores :REAL;
    Index      :INTEGER;
BEGIN
    { Read the scores array }
    { ----- }
    FOR Index := 1 TO NumberOfClasses DO
        BEGIN
            WRITE('Enter score for class #', Index, ': ');
            READLN(Score[Index])
        END;
    { Calculate the sum }
    { ----- }
    SumOfScores := 0;
    FOR Index := 1 TO NumberOfClasses DO
        SumOfScores := SumOfScores + Score[Index];
    { Calculate the average }
    { ----- }
    Average := SumOfScores / NumberOfClasses;
    { Display Results }
    { ----- }
    WRITELN;
    WRITELN('Sum of scores = ', SumOfScores:0:2);
    WRITELN('Average of scores = ', Average:0:2);
    WRITELN;
    WRITELN('Press ENTER to continue..');
    READLN
END.

```



```

BEGIN
    ReadScores;
    GetAverage;
    DisplayResults
END.

```

- Program yang besar dan rumit dapat dipecah-pecah menjadi kecil-kecil sehingga menjadi lebih sederhana.
- Dapat dikerjakan lebih dari satu orang dengan koordinasi yang lebih mudah.
- Lebih mudah mencari kesalahan.
- Dapat dimodifikasi tanpa mengganggu program secara keseluruhan.
- Untuk hal yang sering dilakukan cukup ditulis satu kali dan saat diperlukan cukup dipanggil.
- Mempermudah dokumentasi.

- Dua buah bentuk subprogram yang umum dalam bahasa pemrograman prosedural umumnya dikenal prosedur dan fungsi.
- Subprogram dalam bahasa Pascal diimplementasikan dengan **prosedur dan fungsi**.

1. Prosedur
2. Fungsi
3. Parameter
4. Jangkauan Variabel



Prosedur



- Secara definisi, prosedur sangat mirip dengan program, yaitu terdiri dari:
 1. Bagian header atau Judul, yaitu **NAMA_PROSEDUR**
 2. Bagian blok atau badan program, yang terdiri dari bagian dekarasi dan bagian statement.

- Deklarasi

```
Procedure Nama_Prosedur (parameter);  
Deklarasi variabel;  
Begin  
...  
    statemen-statemen;  
...  
End;
```

Nama_Prosedur seperti
nama sebuah variabel.

Parameter dan Variabel
opsional.

End pada procedur diakhir
titik koma, bukan titik.

- Prosedur tidak mengembalikan nilai.

Judul Program.
Program tidak memiliki variabel dan konstanta.
Deklarasi hanya berisi **Prosedur**.

```
PROGRAM Procedures1(OUTPUT);
{ ----- Beginning of Procedure ----- }
PROCEDURE DrawLine;
CONST
    Dash = '-';
    LineLength = 20;
VAR
    Counter :INTEGER;
BEGIN
    FOR Counter := 1 TO LineLength DO
        WRITE(Dash);
    WRITELN
END;
{ ----- End of Procedure ----- }
{ ----- Main program ----- }
BEGIN
    WRITELN;
    DrawLine;
    WRITELN('** THIS IS A TEST **');
    Drawline
END.
```

Judul prosedur harus valid.
Bagian deklarasi terdiri dari 2 konstanta dan 1 variabel.
Bagian statement terdiri dari sekumpulan task yang akan dijalankan.

Pada main program, prosedur dipanggil 2 kali.

Output:

```
-----
** THIS IS A TEST **
-----
```

Program Judul;

Procedure Bintang;

Begin

 write('*');

End;

Var i : integer;

Begin

for i:=1 **to** 9 **do** bintang;

 writeln;

 bintang;

 write('Judul');

 bintang;

 writeln;

 bintang;

End.

Seperti apakah output program ini?

Judul

*



Fungsi



- Fungsi adalah subprogram yang dapat **mengembalikan nilai**
- Sama seperti prosedur, fungsi terdiri dari:
 1. Bagian header atau Judul, yaitu **NAMA_FUNGSI**
 2. Bagian blok atau badan program, yang terdiri dari bagian dekarasi dan bagian statement.

- Deklarasi

```
Function Nama_fungsi (parameter) :return type;  
Deklarasi variabel ;  
Begin  
...  
    statemen-statemen;  
    Nama_fungsi := return_value  
End;
```

Sebuah fungsi didefinisikan seperti prosedur. Perbedaannya adalah bahwa tipe fungsi (yakni tipe nilai yang akan dikembalikan) harus dinyatakan dalam judul fungsi.

Nilai yang dikembalikan saat fungsi ini dipanggil.

```
Program Judul;
```

```
Function Luas_Lingkaran(r:real):real;  
Begin  
    Luas_lingkaran:=3.14*r*r;  
End;
```

```
Var luas: real;
```

```
Begin  
    luas:= Luas_lingkaran(10);  
    write('Luas= ',luas:0:2);  
End.
```

Seperti apakah output program ini?

Luas = 314.00



Parameter



```

PROGRAM Procedures1(OUTPUT);
{ ----- Beginning of Procedure ----- }
PROCEDURE DrawLine;
CONST
    Dash = '-';
    LineLength = 20;
VAR
    Counter :INTEGER;
BEGIN
    FOR Counter := 1 TO LineLength DO
        WRITE(Dash);
    WRITELN
END;
{ ----- End of Procedure ----- }
{ ----- Main program ----- }
BEGIN
    WRITELN;
    DrawLine;
    WRITELN('** THIS IS A TEST **');
    Drawline
END.

```

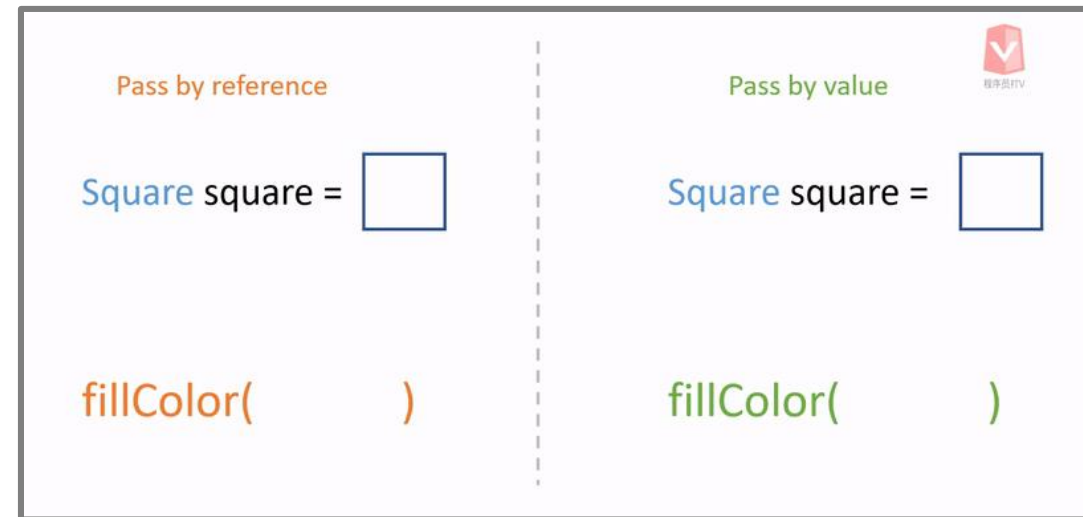
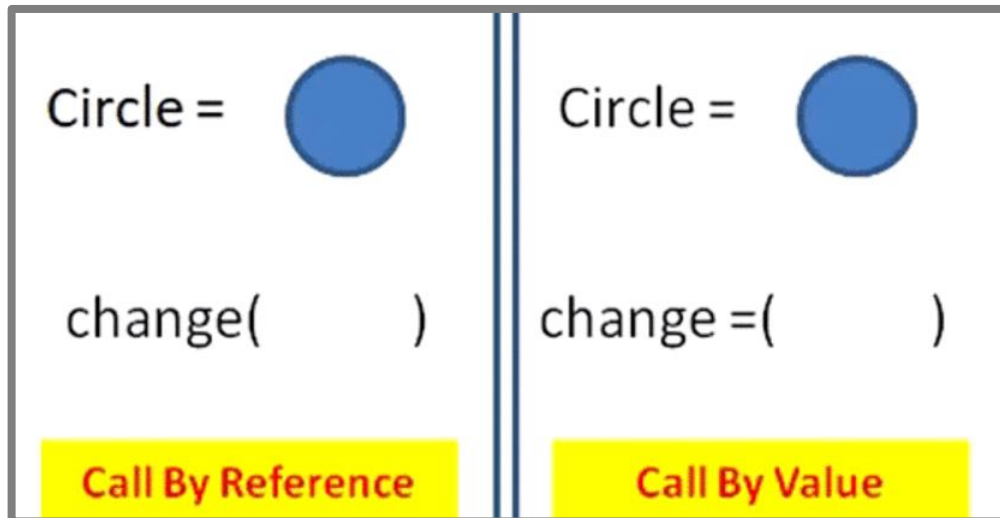
```

PROGRAM Procedures2(OUTPUT):
VAR
    Len          :INTEGER;
    TestSentence  :STRING;
{ ----- Beginning of Procedure ----- }
PROCEDURE DrawLine(LineLength :INTEGER);
CONST
    Dash = '-';
VAR
    Counter :INTEGER;
BEGIN
    FOR Counter := 1 TO LineLength DO
        WRITE(Dash);
    WRITELN
END;
{ ----- End of Procedure ----- }
{ ----- Main program ----- }
BEGIN
    WRITE('Please enter a sentence: ');
    READLN(TestSentence);
    Len := LENGTH(TestSentence);
    WRITELN;
    DrawLine(Len);
    WRITELN(TestSentence);
    Drawline(Len)
END.

```

- Formal
 - Parameter dimana fungsi/prosedur dideklarasikan
- Aktual
 - Parameter dimana fungsi/prosedur digunakan

- By Value: yang dikirim ke modul lain atau ke program utama adalah nilai datanya.
- By Location / By Reference: yang ditransfer ke modul lain atau program utama adalah alamat memorinya.

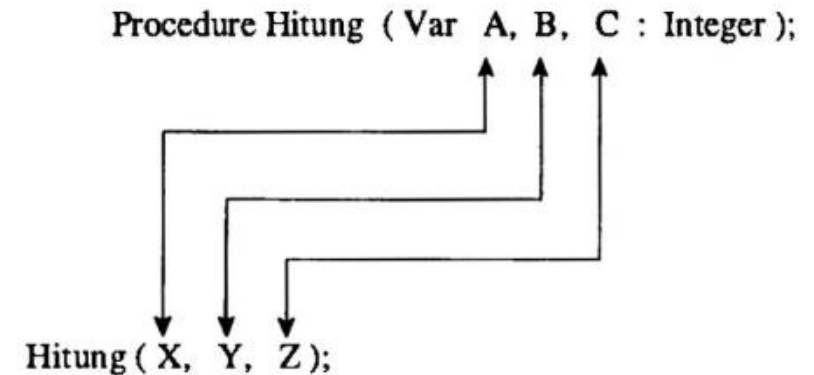


```
Procedure Hitung(Var A,B,C : Integer);
Begin
    C := A +B;
End; (*Procedure Hitung*)
```

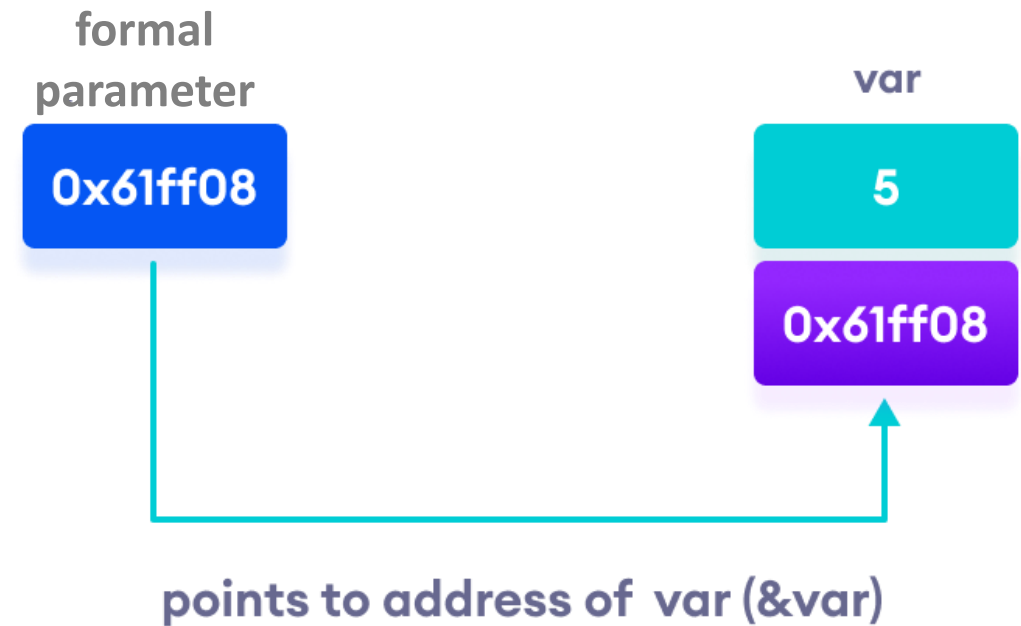
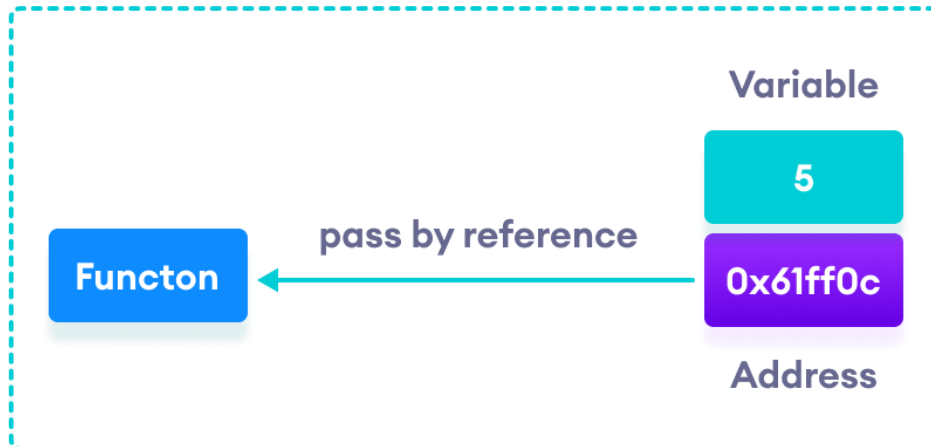
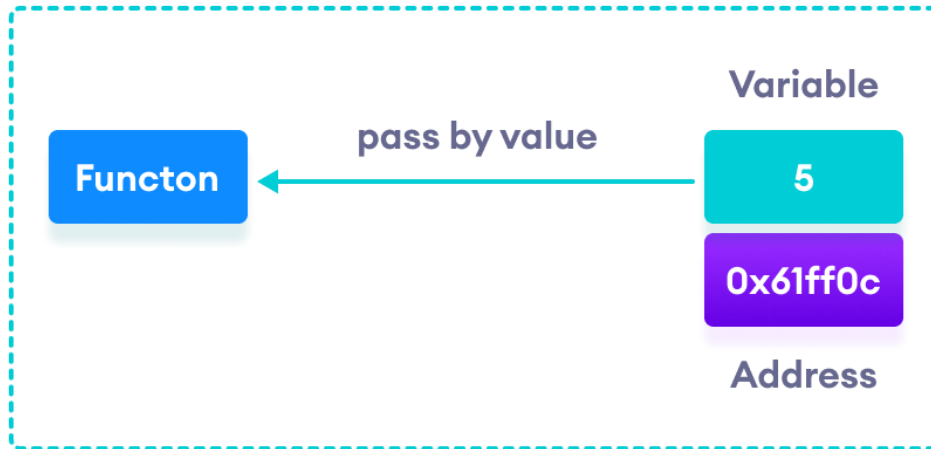
Maka prosedur **Hitung** di atas bisa kita panggil sbb:

```
Var
    X, Y ,Z:Integer;
Begin
    X := 2; Y:= 3;
    Hitung (X, Y, Z);
    Writeln ('X = ', X, ' Y = ', Y, ' Z = ', Z )
End.
```

Hubungan antara parameter formal dengan parameter aktual dengan passed by reference pada contoh tersebut adalah sebagai berikut :



- Cara kerja Pass by Reference mirip seperti pointer



Program Tranfer1;

```
Procedure tukar(x,y:integer);  
  Var z: integer;  
Begin  
  z:=x; x:=y; y:=z;  
End;
```

```
Var a,b:integer;  
Begin  
  a:=2; b:=9;  
  tukar(a,b);  
  writeln(' a= ',a,' b:= ',b);  
End.
```

a=2, b=9

x=2, y=9

z=2, x=9, y=2

a=2, b=9

Program Tranfer2;

```
Procedure tukar(var x,y:integer);  
  Var z: integer;  
Begin  
  z:=x; x:=y; y:=z;  
End;
```

```
Var a,b:integer;  
Begin  
  a:=2; b:=9;  
  tukar(a,b);  
  writeln(' a= ',a,' b:= ',b);  
End.
```

a=2, b=9

x=2, y=9

z=2, x=9, y=2

a=9, b=2

Seperti apakah output masing-masing program?



Variable Scope



- Nilai di dalam modul program atau subprogram Pascal sifatnya adalah lokal, artinya hanya dapat digunakan pada modul atau subprogram yang bersangkutan saja, tidak dapat digunakan pada modul atau subprogram yang lainnya.
- Sehingga jika suatu variabel dideklarasikan didalam suatu prosedur, maka variabel tersebut adalah **variabel lokal**, yang hanya berlaku di dalam prosedur tersebut.
- Cakupan/lingkup variabel local terbatas pada fungsi/prosedur tempat dideklarasikan.
- Parameter termasuk ke jenis variabel lokal.

- Agar suatu variabel bisa berlaku untuk keseluruhan program atau artinya bahwa variabel tersebut bisa diakses oleh program utama maupun subprogram manapun, maka variabel tersebut harus dideklarasikan di program utama sehingga menjadi **variabel global**.
- Variabel global adalah variabel yang dideklarasikan di *main program* (biasanya di baris paling atas program) dan dapat diakses dari unit program mana pun.
- Ruang lingkupnya meliputi seluruh program setelah dideklarasikan.
- Disarankan tidak banyak menggunakan identifier global karena:
 - Jika program semakin besar, kecenderungan error semakin besar.
 - Sulit melacak bila terjadi kesalahan.
 - setiap fungsi dapat mengubah nilai variabel tersebut sehingga sulit terjaga nilainya.

- Apakah output program berikut saat dijalankan?

```
(* Procedure Kuadrat*)  
Procedure Kuadrat;  
Var  
X, Y : Integer;  
Begin  
Write ('Nilai X ? ');  
Readln (X);  
Y := X * X;  
End;  
(* Program Utama *)  
Begin  
Kuadrat;  
Writeln ('Nilai Y = ' ,Y);  
End.
```


- Seperti apakah jangkauan tiap-tiap variabel?

```
Program Jangkauan;  
Var x,y: integer;  
  Function fungsi1(x:real):real;  
  Var y: real;  
  Begin  
    ...  
  End;  
  procedure pro;  
  var x,y,z:integer;  
  begin  
    ...  
  end;  
Var a,b:integer;  
Begin  
  ...  
End.
```

- Ruang lingkup variabel adalah unit program di mana ia dideklarasikan.
- Variabel global dapat diakses di unit program apa pun.
- Variabel lokal tidak dapat diakses di luar unit program di mana ia dideklarasikan.
- Setiap subprogram dapat dipanggil dari unit program mana pun selama didefinisikan sebelum dipanggil.



Latihan



```

const
  xmin = -10;
  xmax = 10;
  ymin = -10;
  ymax = 10;

function code(x: integer; y: integer): byte;
var c : byte;
begin
  c := 0;
  if (x < xmin) then c := c or 8
  else if (x > xmax) then c := c or 1;
  if (y < ymin) then c := c or 4
  else if (y > ymax) then c := c or 2;
  code := c;
end;

```

1. Jika perintah
`writeln(code(-12,-12)
and code(20, 5));`
dijalankan, angka berapakah
yang akan dicetak?
2. Perintah
`writeln(code(-12,-12)
or code(20,5));` akan
mencetak angka
berapakah?

3. Apakah actual parameter dan formal parameter harus bertipe data yang sama? Jelaskan!
4. Coba deklarasikan sebuah variabel global dengan nama X. Lalu deklarasikan kembali variabel dengan nama yang sama (yaitu X) di sebuah prosedur. Apakah jangkauan dari variabel X tersebut? Jelaskan!



POLITEKNIK STATISTIKA STIS

For Better Official Statistics

TERIMA KASIH

