

## MODUL 8 SUB PROGRAM

---

### 1. Deskripsi Singkat

Pada modul 8 ini akan dipelajari tentang sub program dan transfer parameter. Pemahaman terhadap kedua hal ini menjadi sangat penting dalam pembuatan program tingkat lanjut.

### 2. Tujuan Praktikum

Setelah praktikum pada modul 8 ini diharapkan mahasiswa dapat memahami:

- 1) fungsi dan prosedur, serta dapat membedakan antara fungsi dan prosedur
  - 2) transfer parameter
- dan dapat menggunakannya dalam program.

### 3. Material Praktikum

Kegiatan pada modul 8 ini memerlukan material berupa program Pascal.

### 4. Kegiatan Praktikum

#### A. Konsep dan Bentuk Umum Fungsi dan Prosedur

Sub program atau sering disebut sebagai program kecil merupakan bagian penting dari sebuah program yang berfungsi untuk membantu programmer dalam melakukan efisiensi dengan cara menghindari penulisan kode yang berulang-ulang.

Sub program pada Pascal memiliki begin dan end nya sendiri dan dapat pula memiliki variabelnya sendiri (variabel lokal). Variabel yang dideklarasikan pada sub program tidak dapat digunakan pada program utama. Variabel yang dideklarasikan pada program utama disebut dengan variabel global.

Sub Program pada Pascal dibagi menjadi 2 macam yaitu procedure (prosedur) dan function (fungsi). Sub program dideklarasikan pada bagian kamus program.

#### 1. Prosedur

```
Procedure NamaProsedure[ (parameter) ] ;  
[deklarasi variabel];  
begin  
    statemen-statemen;  
end;
```

Perhatikan program untuk menghitung pangkat suatu bilangan berikut ini.

```
Program Belajar_Prosedur;  
  
var n, hasil: integer;
```

```

begin
  write(' N= '); readln(n);

  hasil:=n*n;

  writeln(' Pangkat 2 dari ',n,' = ',hasil);

  readln;
end.

```

Hasil dari program di atas, misal kita inputkan nilai n sama dengan 3, maka outputnya adalah:

```

N= 3

Pangkat 2 dari 3= 9

```

Jika kita modifikasi program di atas sehingga penghitungan pangkat dilakukan di dalam sebuah procedure, maka akan menjadi seperti di bawah ini. Simpan program di bawah ini dengan nama **Praktikum8A.pas** lalu jalankan. Apakah hasilnya sama dengan program tanpa procedure?

```

Program Belajar_Prosedur;
var n,hasil:integer;
Procedure Pangkat2;
begin
  hasil:=n*n;
end;
begin
  write(' N= ');readln(n);
  pangkat2;
  writeln(' Pangkat 2 dari ',n,' = ',hasil);
  readln
end.

```

Simpan ulang program di atas dengan nama **Praktikum8B.pas**, lalu modifikasilah program tersebut agar angka yang akan dipangkatkan dikirimkan sebagai parameter. Ikuti panduan di bawah ini:

```

Program Belajar_Prosedur;
var n,hasil:integer;
Procedure Pangkat2(...); //definisikan parameter

```

di dalam tanda kurung

```
begin
    hasil:=n*n;    // sesuaikan nama variabel dgn nama parameter
end;
begin
    write(' N= ');readln(n);
    pangkat2(...); //modifikasi pemanggilan prosedur dengan
                    mengirimkan parameter
    writeln(' Pangkat 2 dari ',n,'= ',hasil);
    readln
end.
```

## 2. Fungsi

```
Function NamaFungsi[(parameter)]: tipe_fungsi;
[deklarasi variabel];
begin
    statemen-statemen;
    [NamaFungsi:=nilai_fungsi]
end;
```

Buka kembali program **Praktikum8A.pas**, simpan ulang dengan nama **Praktikum8C.pas**.  
Modifikasi prosedur pada program sehingga menjadi sebuah fungsi, seperti di bawah ini:

```
Program Belajar_fungsi;

var n,hasil:integer;

Function Pangkat2:integer; //Deklarasi fungsi dan return value
begin
    Pangkat2:=n*n; //sesuaikan nama variabel dgn nama parameter
end;
begin
    write(' N= ');readln(n);
    hasil:= Pangkat2;
```

```
writeln(' Pangkat 2 dari ',n,' = ',hasil);  
readln;  
end.
```

Simpan ulang program **Praktikum8C.pas** dengan nama **Praktikum8D.pas**, lalu modifikasilah program tersebut agar angka yang akan dipangkatkan dikirimkan sebagai parameter. Ikuti panduan di bawah ini:

```
Program Belajar_fungsi;  
  
var n,hasil:integer;  
  
    Function Pangkat2(...):integer; //definisikan parameter  
begin  
    Pangkat2:=n*n;  
end;  
begin  
    write(' N= ');readln(n);  
    hasil:=pangkat2(...); //modifikasi pemanggilan fungsi  
    writeln(' Pangkat 2 dari ',n,' = ',hasil);  
    readln  
end.
```

**Kesimpulan dari keempat program di atas:**

- 1) Fungsi dan prosedur baik dengan parameter atau tidak, dapat menyelesaikan persoalan yang sama.
- 2) Tentu ada perbedaan antara fungsi dan prosedur. Fungsi mengembalikan sebuah nilai karenanya dalam pendeklarasiannya fungsi harus memiliki tipe dan dibagian akhir ada statemen yang memberikan nilai pada fungsi tersebut. sedangkan prosedur tidak mengembalikan nilai.
- 3) penggunaan parameter dapat menambah fleksibilitas dari fungsi atau prosedur yang dibuat.

## B. Transfer Parameter

Transfer parameter terbagi 2, yaitu tranfer by value (parameternya saat dideklarasikan tanpa kata kunci var) dan transfer by reference/location jika dideklarasikan dengan kata kunci var.

Contoh Program. Simpan program dengan nama **Praktikum8E.pas**.

```
{Contoh program yang menunjukkan perbedaan transfer parameter  
By Value dan By Location}
```

```
Program Contoh8E;
```

```
var x,y:integer;
```

```
    procedure subpro(a:integer;var b:integer);
```

```
    var c:integer;
```

```
    begin
```

```
        c:=a; a:=b; b:=c;
```

```
    end;
```

```
begin
```

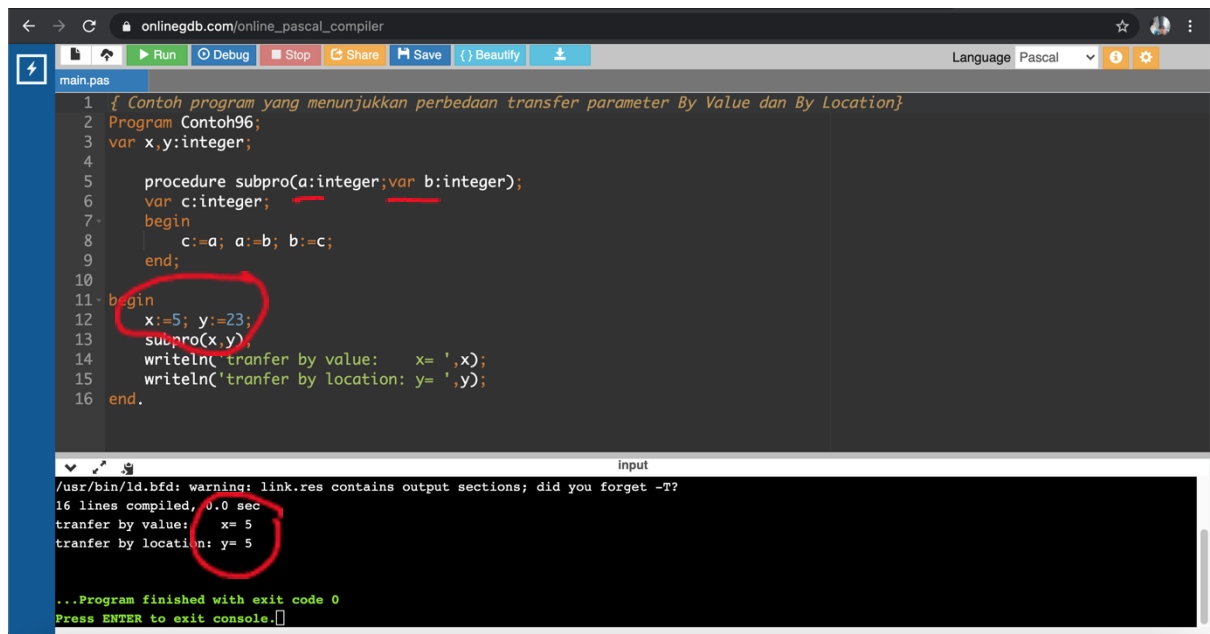
```
    x:=5; y:=23;
```

```
    subpro(x,y);
```

```
    writeln('tranfer by value:    x= ',x);
```

```
    writeln('tranfer by location: y= ',y);
```

```
end.
```



```
onlinegdb.com/online_pascal_compiler
```

```
main.pas
```

```
1 { Contoh program yang menunjukkan perbedaan transfer parameter By Value dan By Location}
```

```
2 Program Contoh96;
```

```
3 var x,y:integer;
```

```
4
```

```
5     procedure subpro(a:integer;var b:integer);
```

```
6     var c:integer;
```

```
7     begin
```

```
8         c:=a; a:=b; b:=c;
```

```
9     end;
```

```
10
```

```
11 begin
```

```
12     x:=5; y:=23;
```

```
13     subpro(x,y);
```

```
14     writeln('tranfer by value:    x= ',x);
```

```
15     writeln('tranfer by location: y= ',y);
```

```
16 end.
```

```
input
```

```
/usr/bin/ld.bfd: warning: link.res contains output sections; did you forget -T?
```

```
16 lines compiled, 0.0 sec
```

```
tranfer by value:    x= 5
```

```
tranfer by location: y= 5
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.
```

Terlihat pada program di atas bahwa parameter aktual x ditransfer by value ke parameter formal a sedangkan parameter aktual y ditransfer by location ke parameter formal b sehingga hasilnya, nilai x tetap tidak berubah sedangkan y nilainya berubah menjadi nilai x sebagaimana dilakukan di dalam prosedur subpro.

Apakah actual parameter dan formal parameter harus bertipe data yang sama? Untuk mengetahui hal tersebut, modifikasi program agar parameter a dan b bertipe real. Jalankan program, lalu apa hasil yang didapatkan?

## 5. Penugasan

1. Simpan tangkapan layar yang menunjukkan source code dan output dari **Praktikum8B.pas**, **Praktikum8D.pas**, dan **Praktikum8E.pas** dalam file docx.

Simpan ulang file tersebut dalam format pdf, dan beri nama dengan format <<kelas>>\_<<nim>>\_modul8.pdf, contoh: **1KS1\_192191234\_modul8.pdf**.

2. Buat program kalkulator dengan fungsi KABATAKU(kali, bagi, tambah, kurang) dimana saat program dijalankan, yang pertama muncul adalah menu seperti di bawah ini:

```
Selamat datang di Kalkulator Sederhana
Silahkan pilih menu berikut:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Keluar
Pilihan Anda:
```

untuk masing-masing menu 1-4, buat dengan menggunakan sub program.

**Simpan program dengan nama kelas\_nim\_Praktikum8F.pas.**

Gabungkan file 1KS1\_192191234\_modul8.pdf dan kelas\_nim\_Praktikum8F.pas menggunakan **zip** dan beri nama dengan format **kelas\_nim\_praktikum8.zip**. Kumpulkan file zip tersebut sebagai responsi melalui Google Classroom sesuai dengan batas waktu yang telah ditetapkan.