

MODUL VI

FUNGSI DAN PROSEDUR

I. TUJUAN

1. Mampu membuat prosedur dan fungsi dalam bahasa C++
2. Mampu menerapkan konsep prosedur dan fungsi dalam studi kasus nyata
3. Dapat

II. DASAR TEORI

Fungsi dan prosedur adalah suatu bagian dari program yang digunakan untuk menjalankan suatu tugas tertentu dan letaknya terpisah dari bagian program yang menggunakannya. Suatu fungsi atau prosedur dipanggil/digunakan dengan tujuan khusus, yaitu untuk mengerjakan suatu tugas tertentu, dimana tugas-tugas tersebut dapat berupa tugas input (menyimpan hasil ke dalam suatu array atau file) dan/atau output (menampilkan hasil di layar monitor) ataupun melakukan penyeleksian dan perhitungan.

Kegunaan/keuntungan menggunakan fungsi dan prosedur :

- 1) Meningkatkan modularitas program
- 2) Kemudahan dalam menulis dan menemukan kesalahan (debug) program
- 3) Menghemat ukuran program

I. FUNGSI

a. Pengertian Fungsi

Fungsi adalah modul program yang mengerjakan tugas/aktifitas khusus dan mengembalikan/memberikan (return) sebuah nilai yang bertipe sederhana (integer, real, boolean, double dan string).



b. Parameter

Parameter adalah suatu variabel yang berfungsi menampung nilai yang akan dikirimkan ke dalam fungsi atau sebaliknya menampung suatu nilai yang akan dikirimkan oleh fungsi ke bagian yang memanggilnya. Dengan adanya parameter suatu fungsi akan bersifat dinamis. Parameter diberikan pada saat deklarasi fungsi.

Jenis parameter :

1. Parameter masukan, adalah parameter yang digunakan untuk menampung nilai yang akan dijadikan masukan (input) ke dalam suatu fungsi.
2. Parameter keluaran, adalah parameter yang digunakan untuk menampung nilai yang akan dijadikan keluaran (output) yang akan dikirimkan ke bagian yang memanggil fungsi tersebut
3. Parameter masukan/keluaran, adalah parameter yang digunakan untuk menampung nilai yang akan dijadikan masukan (input) ke dalam suatu fungsi selain itu juga menampung nilai yang akan dijadikan keluaran (output) yang akan dikirimkan ke bagian yang memanggil fungsi tersebut

c. Deklarasi Fungsi

```
//definisi dan nama fungsi
tipe_data_keluaran nama_fungsi (argumen1,
argumen2,...)
{
//badan fungsi
.....
.....
return ...;
}
```

Deklarasi fungsi pada Bahasa C++ adalah sebagai berikut.

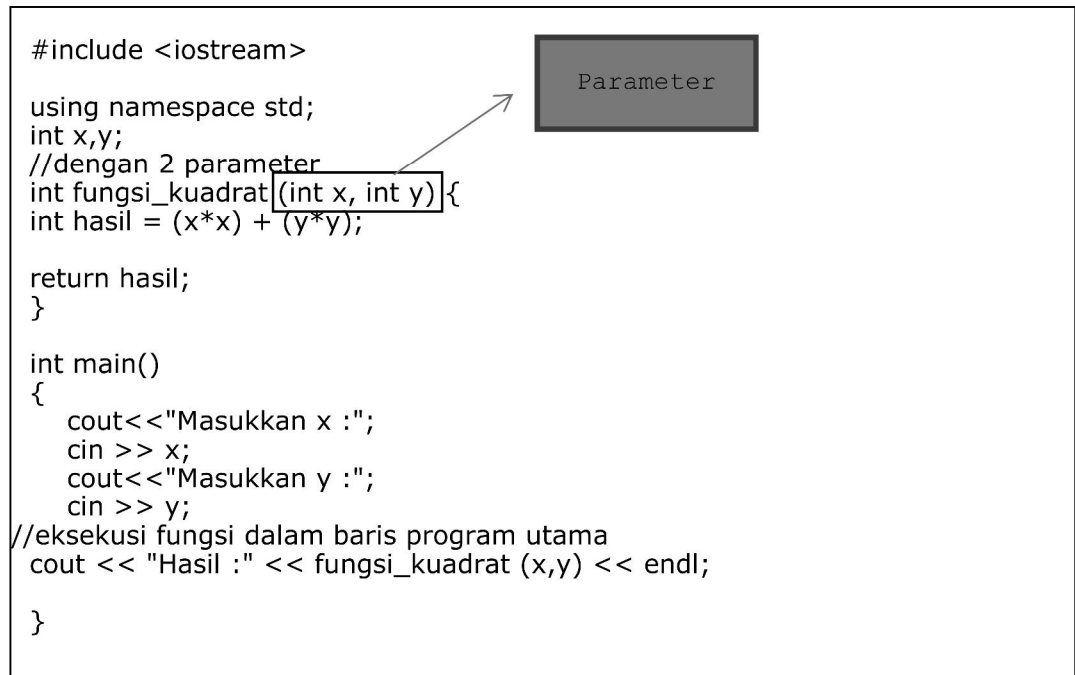
d. Pemanggilan Fungsi

Fungsi yang dipanggil oleh baris program lain yang berada pada badan main() dengan memanggil nama prosedurnya.

- Hilangkan keyword tipe data pada pendeklarasian nama fungsi
- Hilangkan tipe data pada daftar parameter
- Sesuaikan tipe data pada parameter

Contoh :

Berikut merupakan algoritma untuk menghitung fungsi : $f(x) = x^2 + y^2$

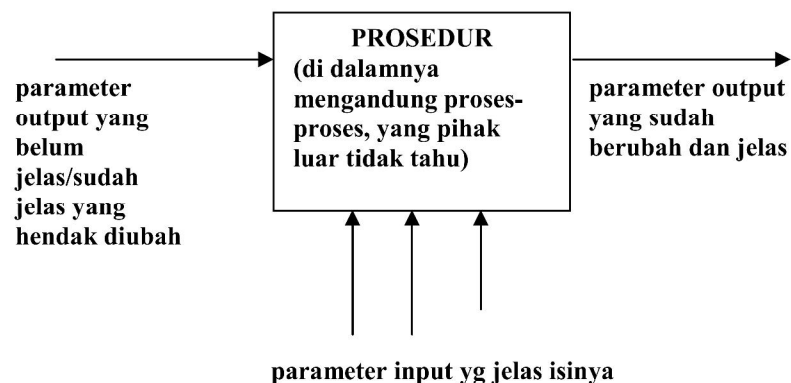


2. PROSEDUR

a. Pengertian Prosedur

Sama halnya dengan fungsi, prosedur (subroutine) adalah modul program yang mengerjakan tugas/aktifitas yang spesifik dan menghasilkan suatu **efek netto**. Efek netto diketahui dengan membandingkan keadaan awal (sebelum) dan keadaan akhir (sesudah) pelaksanaan sebuah prosedur.

Pada dasarnya prosedur membutuhkan input dan output. Input yang standar berasal dari piranti input standar (standard input device), yaitu keyboard. Output yang standar berasal dari piranti output standar (standard output device), yaitu monitor. Hanya ada satu jenis prosedur yang direkomendasikan, yaitu prosedur yang menghasilkan efek netto (*nett effect procedure*). Sama halnya dengan fungsi, prosedur ada yang memiliki parameter atau yang tanpa parameter.



b. Deklarasi Prosedur

Dalam bahasa C/C++ tidak mengenal istilah prosedur, yang dikenal hanya fungsi saja. Prosedur dalam bahasa C/C dikenal juga dengan sebutan fungsi tanpa nilai balik, untuk mendeklarasikan prosedur/fungsi tanpa nilai balik tambahan kita dapat menuliskannya sebagai berikut.

```
//definisi dan nama prosedur
void nama_Prosedur(argumen1, argumen2,...)
{
//badan prosedur
.....
.....
.....
}
```

c. Pemanggilan Prosedur

Prosedur sama dengan fungsi yang dipanggil oleh baris program lain yang berada pada badan main() dengan memanggil nama prosedurnya.

- Hilangkan keyword 'void'
- Hilangkan tipe data pada daftar parameter
- Sesuaikan tipe data pada parameter

Contoh :

Berikut contoh penggunaan prosedur

```
#include <iostream>
using namespace std;

//tanpa parameter
void tampil_nama () {
    cout<< "Nama saya adalah Robert!";
}

//dengan parameter
void tampil_alamat (string x) {
    cout<<x;
}

int main()
{
    //eksekusi prosedur dalam baris program utama
    tampil_nama();
    tampil_alamat ("JI DI Panjaitan");
    return 0;
}
```

d. Perbedaan Fungsi dan Prosedur

Fungsi hanya akan mengembalikan hanya satu nilai ke bagian yang memanggilnya yang artinya sebuah **fungsi hanya mengerjakan satu tugas saja**. Sedangkan prosedur dapat mengembalikan lebih dari satu nilai atau bahkan tidak mengembalikan nilai sama sekali ke bagian yang memanggilnya, yang artinya sebuah **prosedur dapat mengerjakan lebih dari satu tugas**.

III. LATIHAN

I. Kasus :

Membuat program yang akan menghitung luas dan keliling dari persegi berdasarkan masukan dari pengguna

Solusi (dengan fungsi) :

```
#include <iostream>

using namespace std;
int s;
int hitung_keliling (int s) {
    return 4*s;
}
int hitung_luas (int s) {
    return s*s;
}

int masukan;
int main(){
    cout<<"Masukkan Panjang Sisi :";
    cin >> s;
    cout << "Keliling :" << hitung_keliling(s) << endl;
    cout << "Luas :" << hitung_luas (s) << endl;

    return 0;
}
```

Solusi (dengan prosedur) :

```

#include <iostream>

using namespace std;
int s;
void keliling_luas_lingkaran (int s) {
    int luas = s*s;
    int keliling = 4*s;
    cout << "Keliling :" << keliling << endl;
    cout << "Luas :" << luas << endl;
}

int masukan;
int main(){
    cout<<"Masukkan Panjang Sisi :";
    cin >> s;
    keliling_luas_lingkaran(s); //eksekusi prosedur

    return 0;
}

```

2. Kasus :

Menampilkan nilai yang lebih besar dari dua masukan dengan prosedur.

Solusi :

```

#include <iostream>

using namespace std;
int bil1,bil2;
void banding (int a, int b) {
    if (a>b) {
        cout<<a;
    } else if (a==b) {
        cout<< "tidak ada";
    } else {
        cout<<b;
    }
}

int main(){
    cout<<"Masukkan Bilangan 1 :";
    cin >> bil1;
    cout<<"Masukkan Bilangan 2 :";
    cin >> bil2;
    cout <<"Bilangan yang lebih besar adalah :";
    banding (bil1,bil2);
    return 0;
}

```

IV. TUGAS

1. Buatlah program untuk menampilkan nilai bilangan ganjil atau genap dari bilangan yang dimasukkan (gunakan fungsi seleksi ganjil genap)

```
Masukkan bilangan : 5
Bilangan yang anda masukan adalah ganjil
```

2. Buatlah program untuk menghitung luas lingkaran dan fungsi dengan menggunakan fungsi. Jari-jari adalah masukan dari pengguna.
3. Buatlah sebuah kalkulator sederhana untuk melakukan kalkulasi 2 bilangan. Gunakan fungsi atau prosedur dengan tampilan seperti di bawah ini.

KALKULATOR

1. Penjumlahan

2. Perkalian

3. Pembagian

4. Pengurangan

5. Pangkat

Masukkan Pilihan : _

Jika pengguna mengisikan menu 1 maka akan tampil

```
Bilangan pertama : 3
Bilangan kedua   : 4
Hasil penjumlahan: 7
```