

## **PRAKTIKUM 6**

### *Fungsi dan Prosedur*

Tujuan Praktikum :

1. Memperkenalkan kepada mahasiswa konsep pemrograman modular
2. Memperkenalkan kepada mahasiswa pemakaian fungsi/function
3. Memperkenalkan kepada mahasiswa pemakaian prosedur
4. Mempraktekkan pemakaian fungsi
5. Mempraktekkan pemakaian prosedur

**Fungsi (Function)** merupakan blok dari kode yang dirancang untuk melaksanakan tugas khusus. Kegunaan dari fungsi ini adalah untuk:

- ✓ Mengurangi pengulangan penulisan program yang berulang atau sama.
- ✓ Program menjadi lebih terstruktur, sehingga mudah dipahami dan dapat lebih dikembangkan.

Fungsi-fungsi yang sudah kita kenal sebelumnya adalah fungsi `main()`, yang bersifat mutlak, karena fungsi ini program akan dimulai, sebagai contoh yang lainnya fungsi `printf()`, `cout()` yang mempunyai tugas untuk menampilkan informasi atau data kelayar dan masih banyak lainnya.

Terdapat lima konsep dasar mengenai fungsi dalam pemrograman, yaitu:

- Fungsi adalah objek (bagian program/rutin) yang mengerjakan suatu tugas tertentu dan digunakan untuk memodularkan program dengan suatu ciri mengembalikan suatu nilai (return value)
- Fungsi dapat digunakan untuk menghindari penulisan yang sama yang ditulis berulang-ulang.
- Semua variabel yang dideklarasikan dalam fungsi merupakan variabel lokal, yang hanya diketahui dalam fungsi bersangkutan
- Fungsi bisa memiliki parameter yang menyediakan komunikasi antara fungsi dengan bagian yang memanggil fungsi tersebut (Parameter formal dan Parameter aktual).
- Parameter bersifat lokal.

#### **Percobaan 6.1: Penulisan Model Fungsi**

Sebuah fungsi sederhana mempunyai bentuk penulisan sebagai berikut:

```
nama_fungsi(argumen1, argumen2, ...)  
{  
    ... pernyataan / perintah;  
    ... pernyataan / perintah;  
    ... pernyataan / perintah;  
}
```

Keterangan:

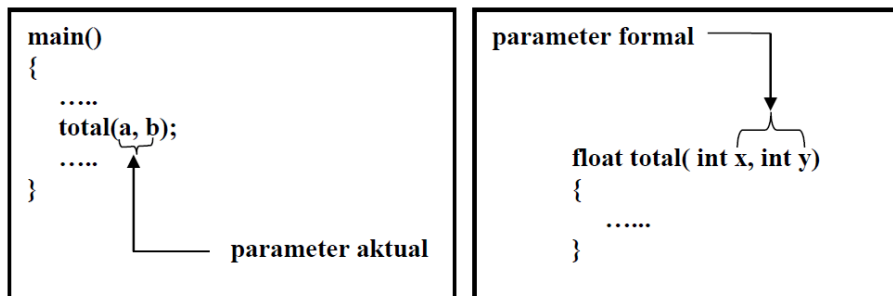
- Nama fungsi, boleh dituliskan secara bebas dengan ketentuan, tidak menggunakan spasi dan nama-nama fungsi yang mempunyai arti sendiri.

- Argumen, diletakan diantara tanda kurung “( )” yang terletak dibelakang nama fungsi. Argumen boleh diisi dengan suatu data atau dibiarkan kosong.
- Pernyataan / perintah, diletakan diantara tanda kurung ‘{ }’.

*Pada pemanggilan sebuah fungsi, cukup dengan menuliskan nama fungsinya.*

Terdapat dua macam para parameter fungsi, yaitu :

- **Parameter formal** adalah variabel yang terdapat pada daftar parameter yang berada didalam definisi fungsi.
- **Parameter Aktual** adalah variabel yang digunakan pada pemanggilan suatu fungsi.



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

void ucapan() { //fungsi void atau prosedur
    cout<<"Program Nilai Kuadrat"<<endl;
    cout<<"-----"<<endl;
}

int pangkat1(int x) { //fungsi
    return ( x * x);
}

int pangkat2(int x); // deklarasi fungsi di bawah main

main() {
    int n;
    ucapan();
    cout<<"Input x : "; cin >> n; cout<<endl;
    cout<<"Pangkat1 = "<< pangkat1(n) << endl;
    cout<<"Pangkat2 = "<< pangkat2(n) << endl;
}

int pangkat2(int x) { // definisi fungsi di bawah main
    return ( x * x);
}
```

### Percobaan 6.2: Pemanggilan dengan nilai (*arguments passed by value*)

```
#include<iostream>
#include<conio.h>
using namespace std;

int tambah(int m, int n) {
    int a,b;
    a= m+5;
    b= n+7;
    cout<<"\n\n Nilai variabel lokal di dalam Fungsi Tambah( )";
    cout<<"\n a = "<<a<<" b = "<<b;
    cout<<endl;
    return m+n;
}

main() {
    int a, b;
    a = 5;
    b = 9;
    cout<<"Nilai Sebelum Fungsi Digunakan ";
    cout<<"\na = "<<a<<" b = "<<b;
    cout<<"\n Hasil return dari Fungsi Tambah( ) = "<<tambah(a,b)<<endl;
    cout<<"\nNilai Setelah Fungsi Digunakan";
    cout<<"\na = "<<a<<" b = "<<b;
    getch();
}
```

### Percobaan 6.3: Pemanggilan dengan acuan (*arguments passed by reference*)

Kadangkala diinginkan sebuah pertukaran nilai yang mempengaruhi nilai variabel pemberinya. Untuk melakukannya, diperlukan sebuah fungsi dengan Pemanggilan dengan nilai.

```
#include<iostream>
#include<conio.h>
using namespace std;

void tambah(int &c, int &d)
{
    c+=7;
    d+=5;
    cout<<endl;
    cout<<"\n Nilai di Akhir Fungsi Tambah()";
    cout<<"\n c = "<<c<<" d = "<<d;
}

main() {
    int a, b;
    a = 4;
    b = 6;
    cout<<" Nilai Sebelum Pemanggilan Fungsi";
    cout<<"\n a = "<<a<<" b = "<<b;
    tambah(a,b);
    cout<<endl;
    cout<<"\n Nilai Setelah Pemanggilan Fungsi";
    cout<<"\n a = "<<a<<" b = "<<b;
    getch();
}
```

### Percobaan 6.4: Fungsi Boolean

```
#include <iostream>
#include <math.h>
using namespace std;

int isPrima(int bil) {
    float sqrtBil = sqrt(float(bil));
    if (bil < 2) return 0; // false
    if (bil == 2) return 1; // true
    if (bil % 2 == 0) return 0; // false
    for (int gjl=3; gjl <= sqrtBil; gjl+=2) {
        if (bil % gjl == 0) return 0; // false
    }
    return 1; // true
}

main() {
    cout << "Bilangan-bilangan prima <100 : " << endl;
    for (int n=1; n < 100; n++) {
        if (isPrima(n))
            cout << n << " ";
    }
    cout << endl;
}
```

### Percobaan 6.5: Fungsi Rekursif

**Fungsi rekursif** adalah suatu fungsi yang **memanggil dirinya sendiri**, artinya fungsi tersebut dipanggil di dalam tubuh fungsi itu sendiri. Fungsi rekursif sangat berguna bila diimplementasikan untuk pekerjaan pengurutan data, atau menghitung nilai factorial suatu bilangan.

```
#include<iostream>
#include<conio.h>
using namespace std;
long factorial (long a) {
    if (a>1)
        return (a* factorial (a-1));
    else
        return (1);
}

main() {
    long l;
    cout<<"tuliskan bilangan : ";cin>>l;
    cout<<"!"<<l<<" = "<<factorial(l);
}
```

### Sumber/Referensi:

- (1) Munir, R. (2012). Algoritma dan Pemrograman. Jilid 1 Bandung: Penerbit Informatika.
- (2) Wirth, N. (1990). Algorithms + Data Structures = Programs. India: Prentice-Hall Of India Pvt. Limited.
- (3) Hubbard, J.R. (1996). Programming With C++, Schaum's outlines Series. USA: McGraw Hill
- (4) Suarga, M.Math., (2012). Algoritma Dan Pemrograman (Edisi 2). Yogyakarta: Penerbit Andi.
- (5) Shalahuddin, M dan AS. Rosa. (2007) Belajar Bahasa Pemrograman dengan C++ dan Java: Penerbit Informatika.
- (6) Joel Adams-Sanford Leestma-Larry Nyhoof, (1995). C++ An Introduction To Computing: Prentice-Hall, Inc
- (7) James P. Cohoon-Jack W.Davidson, McGraw-Hill, (1997). C++ Programming Design.
- (8) Budi Raharjo, (2004). Mengungkap Rahasia Pemrograman Dalam C++: Penerbit Informatika.
- (9) Abdul Kadir, (2003). Pemrograman C++, Andi, Yogyakarta
- (10) Yosua Onesimus Suheru, (2004). Trik Memecahkan Masalah Dengan Tiga bahasa Pemrograman – C++, Pascal dan Visual Basic, Gava Media, Yogyakarta