

# Fungsi (**Function**)

Modul **Praktikum**

C++

**Dasar Pemrograman Komputer**

Semester Genap 2017/2018

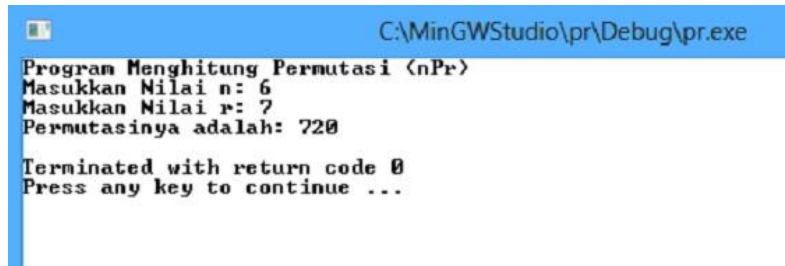
## MODUL 6 FUNGSI

### A. Tujuan

Setelah mempelajari bab ini diharapkan mahasiswa akan mampu :

- Memahami Fungsi.
- Mendeklarasikan dan mendefinisikan fungsi.
- Menyelesaikan masalah menggunakan fungsi.
- Memahami metode pemanggilan fungsi.
- Membuat fungsi rekursif.

### B. Dasar Teori



```
C:\MinGWStudio\pr\Debug\pr.exe
Program Menghitung Permutasi (nPr)
Masukkan Nilai n: 6
Masukkan Nilai r: 7
Permutasinya adalah: 720
Terminated with return code 0
Press any key to continue ...
```

Dalam pembuatan program diatas terdapat suatu rumus permutasi. Agar program lebih efisien dipisahkan antara rumus dengan program utama, sehingga dalam program utam hanya ada perintah memanggil rumus untuk memproses suatu variable. Hal ini dalam pemrograman disebut sebagai fungsi. Fungsi merupakan blok dari kode yang dirancang untuk melakukan tugas khusus. Tujuan pembuatan fungsi adalah:

- Program menjadi terstruktur
- Dapat mengurangi duplikasi kode
- Fungsi dapat di panggil dari program atau fungsi yang lain

Pada umumnya fungsi memerlukan masukan yang disebut parameter atau argument. Hasil akhir fungsi akan berupa nilai(nilai balik fungsi). Bentuk umum sebuah fungsi adalah:

```
TypeFungsi namaFungsi (parameter)
{
//Tubuh fungsi
}
```

Misalkan: Dalam bahasa C++

```
void Ckali(int angka)
{
    int a=angka*angka;
    cout<<"Hasil Perkalian:"<<a<<endl;
}
```

TypeFungsi berfungsi untuk menentukan tipe keluaran fungsi yang dapat berupa salah satu tipe data C++ yang berlaku misalnya char atau int. Default tipe yang tidak disebutkan dianggap sebagai int.

```
Contoh:
Ckali() {
    return (0);
}
```

### PROTOTYPE FUNCTION

Sebuah program C++ dapat terdiri dari banyak fungsi. Salah satu fungsi tersebut harus bernama main(). Jika fungsi yang lain di tuliskan setelah fungsi main(), sebelum fungsi main harus ditambahkan prototype function agar fungsi tersebut dapat dikenal. Hal tersebut tidak berlaku jika penulisannya dilakukan sebelum fungsi main(). Misal:

```

#include <stdio.h>
//prototype function
void hitung(int angka, int bilangan);
void tulis(char);
int tampil(int angka[],char huruf);
//fungsi main
void main()
{
int array[3]={1,2,3};
char huruf='D';
//memanggil fungsi
hitung(2,3);
tulis('A');
tampil(array,huruf);
}
//inisialisasi fungsi hitung
void hitung(int a, int b)
{
//Blok fungsi
printf("Angka adalah: %d\n",a);
printf("Angka adalah: %d\n",b);
}
//inisialisasi fungsi tulis
void tulis(char a)
{
//blok fungsi
printf("huruf adalah: %c\n",a);
} //inisialisasi fungsi tampil
int tampil(int x[], char a) {
//blok fungsi
return(a);
}

```

## Call by Value vs Call by Reference

Ada dua cara pemanggilan fungsi yaitu: (1) Call by Value (2) Call by Reference

### (1) Call By Value

Pemanggilan fungsi dengan melewati nilai parameter variable dalam definisi fungsi. Fungsi tersebut akan mengolah nilai yang dikirimkan kepadanya. Misalnya:

```

float bayar = 5000;

float tax= calcSalesTax(bayar); dapat
juga merupakan suatu konstanta:
hitung(2, 3);
tulis ("A");

```

### (2) Call By Reference

Pemanggilan dengan referensi merupakan suatu usaha melewati alamat dari suatu variable ke dalam fungsi. Dengan perubahan dilakukan di dalam fungsi untuk mengubah variabel diluar fungsi. Misal:

```

A bernilai 20 □ X juga bernilai 20
B bernilai 30 □ Y juga bernilai 30

```

Selanjutnya nilai X dan Y diproses tetapi nilai A dan B akan tetap( tidak berubah). Metode Call By Reference menggunakan operator pointer "&" dan "\*" untuk mengirimkan alamat variable ke suatu fungsi.

### **Fungsi Rekursif**

Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri artinya fungsi tersebut dipanggil di dalam tubuh fungsi itu sendiri. Parameter yang dilewatkan berubah sebanyak fungsi itu dipanggil.

## C. Latihan Praktikum

### 1. Latihan 1

Jelaskan maksud dari program berikut ini:

```
#include <iostream.h>
void Rtukar (int *,int*);
void Vtukar (int,int);
main()
{
    int a,b;
    a=99; b=11;
    //call by reference
    cout<<"Nilai sebelum pemanggilan fungsi "<<endl;
    cout<<"Nilai a= " <<a<<" " Nilai b= "<<b<<endl;
    Rtukar(&a,&b);
    cout<<"Nilai setelah pemanggilan fungsi Rtukar: "<<endl;
    cout <<"Nilai a="<<a<<" " Nilai b="<<b<<endl<<endl;
    //kembalikan ke nilai semula
    a=99;
    b=11;
    //call by value
    cout<<"\nNilai sebelum pemanggilan fungsi : "<<endl;
    cout<<"Nilai a="<<a<<" " Nilai b="<<b<<endl<<endl;
    Vtukar(a,b);
    cout<<"Nilai setelah pemanggilan fungsi Vtukar: "<<endl;
    cout<<"Nilai a=" <<a<<" " Nilai b=" <<b<<endl;
    return 0;
}
void Rtukar (int *px,int *py)
{
    int temp;
    temp = *px;
    *px=*py;
    *py=temp;
    cout<<"Nilai pada fungsi dengan Call by Value: "<<endl;
    cout<<"Nilai px="<<*px<<" " Nilai py="<<*py<<endl<<endl;
}
```

```

void Vtukar (int px,int py)
{
    int temp;

    temp = px;

    px=py;

    py=temp;

    cout<<"Nilai pada fungsi dengan Call by Value: "<<endl;

    cout<<" Nilai px="<<px<<" Nilai py="<<py<<endl<<endl;

}

```

## 2. Latihan 2

Mencari nilai terbesar dari suatu data

```

//fungsi
#include<stdio.h>
//deklarasi fungsi
int cari(int n1, int n2);
void prin (int m);

//main function
main()
{
    int i=5;
    int j=7;
    int k;
    k = cari(i,j);
    prin(k);
    return 0;
}
//<-- function definition here

int cari(int a, int b)
{
    if (a>b) return a;
    else return b;
}

void prin (int m)
{
    printf("Bilangan terbesar adalah: %d\n",m);
}

//<-- end function

```

## 3. Latihan 3

Menentukan pajak pembelian

```

#include <iostream.h>

//deklarasi fungsi
int calcSalesTax(float purchase, float taxRate=0.1);

//main function
main()
{
    float purchase;
    cout<<"Masukkan pembelian : ";
    cin>>purchase;

    float tax=calcSalesTax(purchase);
    cout<<"Pajak pembelian adalah : "<<tax<<endl;
    return 0;
}

int calcSalesTax(float purchase, float taxRate)
{
    return purchase * taxRate;
}

```

## 4. Latihan 4

Membedakan program call by reference dan call by value

```

/*aplikasi rekursif pada faktorial*/
#include <iostream.h>

int faktorial(int a)
{
    if(a==1)
        return 1;
    else
        return a*faktorial(a-1);
}

void main()
{
    int x;
    cout<<"Mencari Nilai Faktorial\n";
    cout<<"Masukkan nilai n=";
    cin>>x;
    cout<<"Faktorial " <<x<<"!="<<faktorial(x)<<endl;
}

```

## 5. Latihan 5

Program menghitung factorial dengan fungsi rekursif

```

/*aplikasi rekursif pada fibonacci*/
#include <iostream.h>

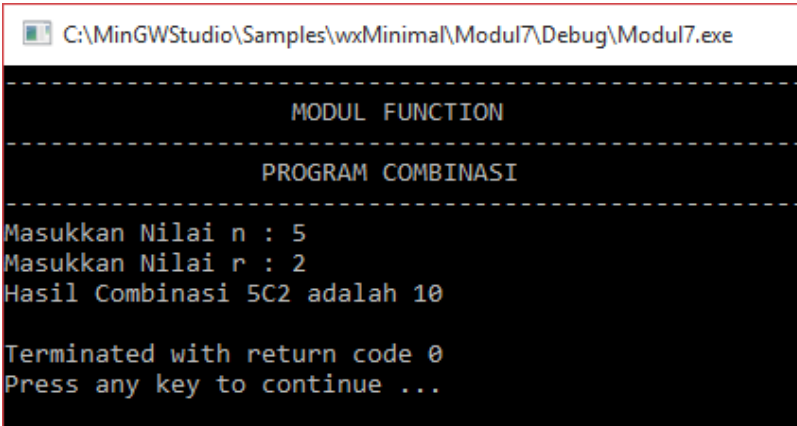
long fibo(long a)
{
    if(a==1 && a==2)
        return 1;
    else
        return fibo(a-1) + fibo(a-2);
}

void main()
{
    int x;
    cout<<"Mencari Nilai Fibonacci\n";
    cout<<"Masukkan nilai n=";
    cin>>x;
    cout<<"Fibonacci " <<x<<"="<<fibo(x)<<endl;
}

```

## D. Tugas Praktikum

1. Program fungsi untuk menampilkan kombinasi  $nCr$



```

C:\MinGWStudio\Samples\wxMinimal\Modul7\Debug\Modul7.exe
-----
MODUL FUNCTION
-----
PROGRAM COMBINASI
-----
Masukkan Nilai n : 5
Masukkan Nilai r : 2
Hasil Kombinasi 5C2 adalah 10

Terminated with return code 0
Press any key to continue ...

```

2. Buat sebuah program untuk menghitung luas permukaan, selimut, volume tabung seperti berikut ini:



```
C:\MinGWStudio\Samples\wxMinimal\Modul7\Debug\Modul7.exe
-----
MODUL FUNCTION
-----
PROGRAM COMBINASI
PROGRAM TABUNG
-----
Masukkan Nilai Jari-Jari Tabung : 32
Masukkan Nilai Tinggi Tabung : 10
-----
Nilai Luas Tabung      : 8440.32
Nilai Selimut Tabung  : 2009.6
Nilai Volume Tabung   : 32153.6
-----
Terminated with return code 0
Press any key to continue ...
```

## E. Tugas Rumah

1. Buatlah sebuah program untuk mencari nilai terbesar dan nilai terkecil dari data-data yang telah diinputkan oleh user. Berikut contoh tampilan program.

```
C:\MinGWStudio\Samples\wxMinimal\Modul7\Debug\Modul7.exe
-----
MODUL FUNCTION
-----
MENENTUKAN BATAS
-----
Maskkan data : 3
Masukkan Data Lagi?(y/n) : y
Maskkan data : 9
Masukkan Data Lagi?(y/n) : y
Maskkan data : 5
Masukkan Data Lagi?(y/n) : y
Maskkan data : 11
Masukkan Data Lagi?(y/n) : n

Data terbesar : 11

Data terkecil : 3

Terminated with return code 0
Press any key to continue ...
```

2. Buatlah sebuah program untuk mencari rata-rata dan simpangan baku (standar deviasi) dari data-data yang telah diinputkan. Berikut contoh tampilan program.

```
C:\MinGWStudio\Samples\wxMinimal\Modul7\Debug\Modul7.exe
-----
MODUL FUNCTION
-----
PROGRAM STANDAR DEVIASI
-----
Maskkan data : 4
Masukkan Data Lagi?(y/n) : y
Maskkan data : 6
Masukkan Data Lagi?(y/n) : y
Maskkan data : 2
Masukkan Data Lagi?(y/n) : n

  Nilai Rata-rata : 4

  Simpangan Baku  : 2

Terminated with return code 0
Press any key to continue ...
```

3. Buatlah sebuah program untuk menampilkan data yang sering diinputkan oleh user sebanyak beberapa kali. Berikut contoh tampilan program.

```
C:\MinGWStudio\Samples\wxMinimal\Modul7\Debug\Modul7.exe
-----
MODUL FUNCTION
-----
MENENTUKAN DATA YANG SERING DIINPUTKAN
-----
Maskkan data : 4
Masukkan Data Lagi?(y/n) : y
Maskkan data : 5
Masukkan Data Lagi?(y/n) : y
Maskkan data : 3
Masukkan Data Lagi?(y/n) : y
Maskkan data : 4
Masukkan Data Lagi?(y/n) : y
Maskkan data : 2
Masukkan Data Lagi?(y/n) : y
Maskkan data : 1
Masukkan Data Lagi?(y/n) : y
Maskkan data : 4
Masukkan Data Lagi?(y/n) : n

-----
Data yang sering diinputkan adalah 4
-----

Terminated with return code 0
Press any key to continue ...
```