

# Pertemuan 10

## Fungsi

**Lukman Nulhakim, M.Kom**

STMIK ANTAR BANGSA

*lukman@antarbangsa.ac.id*

# Pokok Bahasan

1. Definisi Fungsi
2. Struktur Fungsi
3. Contoh Fungsi

# Definisi Fungsi

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

Pemanggilan Fungsi dapat berulang-ulang dengan cara memanggil nama fungsinya.

## 10.1. Struktur Fungsi

Sebuah fungsi sederhana mempunyai bentuk penulisan sebagai berikut:

```
nama_fungsi(argumen)
{
  ... 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 '{ }'.

- Fungsi dapat dibedakan menjadi 2.
  1. Fungsi sederhana(tanpa parameter)
  2. Fungsi Berparameter

# Contoh Fungsi Sederhana

Fungsi Membuat Garis dan Judul

```
D:\...\NONAME02.CPP *
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
#include <iomanip.h>

judul()
{
    cout<<" STMIK ANTAR BANGSA "<<endl;
}

garis()
{
    cout<<" ===== " <<endl;
}

main()
{
    clrscr();
    garis();
    judul();
    garis();

    getch();
}
```

```
D:\DATA CAMPUS\BAHAN AJAR\AMIK BSI\PROGRAM\CPP\...
=====
STMIK ANTAR BANGSA
=====
```

Prototype Fungsi adalah menjelaskan kepada kompiller tentang tipe data fungsi, jumlah parameter, tipe data dari masing-masing parameter.

Prototipe fungsi dituliskan di atas blok program utama dan diakhiri dengan tanda qualifier titik koma ( ; ), sedangkan blok program fungsi yang mengandung perintah-perintah atau pernyataan-pernyataan dari program berada di bawah blok program utama yang memiliki keuntungan sebagai berikut:

- Kompiler akan melakukan konversi antara tipe parameter dalam definisi dan parameter fungsi.
- Jika jumlah parameter yang digunakan dalam definisi fungsi dan pada saat pemanggilan fungsi berbeda atau tidak sama, maka akan menunjukkan kesalahan.

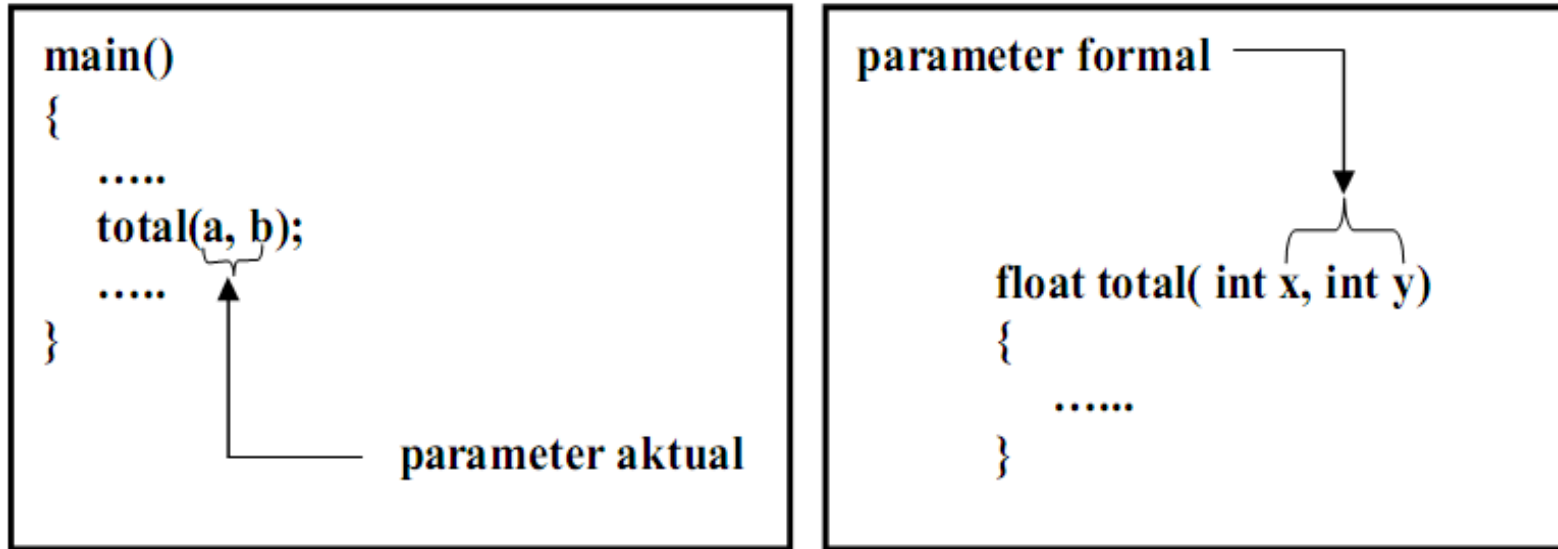
Parameter Fungsi adalah suatu pendefinisian nilai-nilai dari objek-objek yang dideklarasikan pada bagian argumen di fungsi. Nilai-nilai pada objek-objek tersebut didapat dari variabel-variabel yang berada pada program utama

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



# Bentuk penulisan Parameter Fungsi

Bentuk penulisan Parameter Formal dan Parameter Aktual.



Yang Perlu diperhatikan dalam pembuatan Fungsi berparameter :

1. Jumlah Parameter aktual dan parameter formal harus sama
2. Tipe data yang digunakan oleh parameter harus sama
3. Tipe pemanggilan fungsi

# Contoh Program Fungsi

```
//Penggunaan Prototipe pada Fungsi
#include <conio.h>
#include <iostream.h>
#include <string.h> //untuk strcpy

char coment (char ket[30],int n); //prototipe fungsi

main( )
{
    char lagi,c[30];
    int i;
    atas:
    clrscr( );
    {
        cout<<"Masukkan nilai = ";cin>>i;
        coment(c,i); //paramater formal
        cout<<c;
        cout<<"\n\nIngin input lagi [Y/T]: ";cin>>lagi;
    }
    if (lagi=='Y' || lagi=='y')
        goto atas;
    else
        getch( );
}
```

```
//blok program fungsi dengan parameter aktual
char coment (char ket[30],int n)
{
    int a;
    a=n%2;
    if (a==1)
        strcpy(ket,"---Bilangan Ganjil---");
    else
        strcpy(ket,"---Bilangan Genap---");
}
```

# Pemanggilan Fungsi Berparameter

1. Pemanggilan Secara Nilai
2. Pemanggilan Secara Reference
3. Pemanggilan Return

# Pemanggilan Secara Nilai (by value)

Pemanggilan fungsi dengan nilai akan menyebabkan nilai parameter aktual tidak akan berubah, walaupun nilai parameter formalnya berubah.

```
/* ----- */
/* Penggunaan Call By Value */
/* Program Tambah Nilai */
/* ----- */
#include<conio.h>
#include<stdio.h>
#include<iostream.h>

tambah(int m, int n);

main( )
{
    int a, b;
    a = 5;
    b = 9;
    clrscr( );
    cout<<"Nilai Sebelum Fungsi Digunakan ";
    cout<<"\na = "<<a<<" b = "<<b;
    tambah(a,b);
    cout<<"\nNilai Setelah Fungsi Digunakan";
    cout<<"\na = "<<a<<" b = "<<b;
    getch( );
}

tambah(int m, int n)
{
    m+=5;
    n+=7;
    cout<<"\n\nNilai di dalam Fungsi Tambah()";
    cout<<"\nm = "<<m<<" n = "<<n;
    cout<<endl;
```

# Pemanggilan Secara Nilai (by Reference)

Pemanggilan fungsi dengan nilai akan menyebabkan nilai parameter aktual berubah, dengan cara merubah nilai parameter formalnya. Untuk pemanggilan by reference memerlukan sebuah pointer ( \* ) didepan nama parameter formalnya, dan ( & ) untuk nama parameter aktualnya.

```
/* ----- */
/* Penggunaan Call By Reference */
/* Program Tambah Nilai */
/* ----- */

#include<conio.h>
#include<stdio.h>
#include<iostream.h>

tambah(int *c, int *d);

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

tambah(int *c, int *d)
{
    *c+=7;
    *d+=5;
    cout<<endl;
    cout<<"\nNilai di Akhir Fungsi Tambah()";
    cout<<"\nc = "<<*c<<" d = "<<*d;
}
```

Dilakukan untuk melakukan pengiriman suatu nilai atau nilai dari suatu fungsi kepada fungsi yang lain yang memanggilnya.

```
/* Pernyataan Return pd Fungsi */
#include <conio.h>
#include <iostream.h>
#include <stdio.h>

float luas (int r)//fungsi luas lingkaran
{
    return(3.14*r*r);
}

float kel (int r)//fungsi kel lingkaran
{
    return(3.14*2*r);
}
```

```
main( )
{
    int j;
    clrscr( );
    cout<<"Masukkan Jari-jari = ";cin>>j;
    cout<<"Luas lingkaran = "<<luas(j)<<endl;
    cout<<"Keliling lingkaran = "<<kel(j);
    getch( );
}
```



# Contoh Penggunaan Fungsi

Buatlah rancangan perhitungan luas segitiga dengan tampilan seperti berikut :

## Hitung Luas Segitiga

=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=

Input Alas : <input>

Input Tinggi : <input>

=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=

Luas segitiga dengan alas = <output>

dan tinggi = <output> adalah = <output>

Dimana perhitungan luas nya dilakukan di dalam fungsi luas\_segitiga dan penggunaan fungsi garis

```
#include<conio.h>
#include<iostream.h>
#include<stdio.h>
segitiga(float alas,float tinggi); ➔ deklarasi fungsi
main( )
{
float a,t,hasil;
cout<<"MASUKKAN NILAI ALAS SEGITIGA : ";cin>>a;
cout<<"MASUKKAN NILAI TINGGI SEGITIGA : ";cin>>t;
hasil= segitiga(a,t);
printf("\n HASIL LUASNYA : %4.0f",hasil);
getch( );
}
segitiga(float alas,float tinggi) ➔ proses fungsi segitiga
{
return((alas*tinggi)/2);
}
```

**Penjelasannya :** input nilai alas ( a ) dan tinggi ( t ), lalu untuk proses hasil dilakukan pemanggilan fungsi segitiga dan mengirimkan nilai a dan t kedalam fungsi segitiga, dimana ( a ) sebagai alas dan ( t ) sebagai tinggi. Lalu didalam fungsi segitiga di proses hitung luas segitiganya. Lalu dengan fungsi return nilainya di kirim kembali sebagai hasil.



# Buatlah Tampilan Output dari List Code dibawah ini

```
#include<conio.h>
#include<iostream.h>
#include<stdio.h>
siswa(float uts, float uas); //deklarasi fungsi
garis( ) //fungsi garis
{
    printf( "\n\t= = = = = \n");
}
main( )
{
    char nim[10],nama[20],ket[10],lagi;
    float a,b,rata;
    garis( );
    cout<<"\n\t Perhitungan Nilai Siswa "<<endl;
    cout<<"\t" ; garis ( );
    cout<<"\t Masukkan Nim   : "; cin>>nim;
    cout<<"\t Masukkan Nama  : "; cin>>nama;
    cout<<"\t Nilai UTS      : "; cin>>a;
    cout<<"\t Nilai UAS      : "; cin>>b;
```

```
rata = siswa( a , b ) ;
printf("\n\t Nilai Rata-Rata : %3.2f", rata);
if(rata > 59 )
    strcpy(ket, "LULUS");
else
    strcpy(ket, "GAGAL");
cout<<"\n\t Keterangan   : "<<ket<<endl;
getch( );
}

siswa(float uts, float uas) //proses fungsi
siswa
{
    return(( uts + uas)/ 2 );
}
```

# THANK FOR ATTENTION

**Lukman Nulhakim, M.Kom**

STMIK ANTAR BANGSA

*lukman@antarbangsa.ac.id*