

PRAKTIKUM PEMROGRAMAN KOMPUTER  
MINGGU 12  
Fungsi

**A. Tujuan praktikum:**

- Mahasiswa mampu membuat program sederhana dengan menggunakan fungsi

**B. Dasar Teori**

**1. PENGERTIAN FUNGSI**

Fungsi merupakan suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya terpisah dari program yang memanggilnya. Fungsi merupakan elemen utama dalam bahasa C karena bahasa C sendiri terbentuk dari kumpulan fungsi-fungsi. Dalam setiap program bahasa C, minimal terdapat satu fungsi yaitu fungsi main(). Fungsi banyak diterapkan dalam program-program C yang terstruktur. Keuntungan penggunaan fungsi dalam program yaitu program akan memiliki struktur yang jelas (mempunyai readability yang tinggi) dan juga akan menghindari penulisan bagian program yang sama. Dalam bahasa C fungsi dapat dibagi menjadi dua, yaitu fungsi pustaka atau fungsi yang telah tersedia dalam Turbo C dan fungsi yang didefinisikan atau dibuat oleh programmer.

**2. BEBERAPA FUNGSI PUSTAKA DALAM BAHASA C**

□ **Fungsi Operasi String (tersimpan dalam header file "string.h")**

◆ **strcpy()**

- Berfungsi untuk menyalin suatu string asal ke variable string tujuan.
- Bentuk umum : strcpy(var\_tujuan, string\_asal);

◆ **strlen()**

- berfungsi untuk memperoleh jumlah karakter dari suatu string.
- Bentuk umum : strlen(string);

*Contoh Program :*

```
#include "iostream.h"
#include "conio.h"
#include "string.h"
void main()
{
    char nama[25];
    strcpy(nama, "Achmad Solichin");
    cout<<"Nama : "<< nama;
    cout<<"Banyaknya karakter nama Anda adalah : " <<strlen(nama);
    getch();
}
```

**3. MEMBUAT FUNGSI SENDIRI**

□ **Deklarasi Fungsi**

Sebelum digunakan (dipanggil), suatu fungsi harus dideklarasikan dan didefinisikan terlebih dahulu.

Bentuk umum pendeklarasian fungsi adalah :

tipe\_fungsi nama\_fungsi(parameter\_fungsi);

Sedangkan bentuk umum pendefinisian fungsi adalah :

Tipe\_fungsi nama\_fungsi(parameter\_fungsi)

```
{
    statement
    statement
    .....
    .....
}
```

#### □ Hal-hal yang perlu diperhatikan dalam penggunaan fungsi :

- ♦ Kalau tipe fungsi tidak disebutkan, maka akan dianggap sebagai fungsi dengan nilai keluaran bertipe integer.
- ♦ Untuk fungsi yang memiliki keluaran bertipe bukan integer, maka diperlukan pendefinisian penentu tipe fungsi.
- ♦ Untuk fungsi yang tidak mempunyai nilai keluaran maka dimasukkan ke dalam tipe void
- ♦ Pernyataan yang diberikan untuk memberikan nilai akhir fungsi berupa pernyataan return.
- ♦ Suatu fungsi dapat menghasilkan nilai balik bagi fungsi pemanggilnya.

#### Contoh Program 1 :

```
float tambah(float x, float y); /* prototype fungsi tambah(), ada titik koma */
void main()
{
    float a, b, c;
    clrscr();
    cout<<"A = ";
    cin>>a;
    cout<<"B = ";
    cin>>b;
    c = tambah(a, b); /* pemanggilan fungsi tambah() */
    cout<<"A + B = "<< c;
    getch();
}
float tambah(float x, float y) /* Definisi fungsi , tanpa titik koma */
{
    return (a+b); /* Nilai balik fungsi */
}
```

#### Contoh Program 2 :

```
/* Program menghitung nilai factorial */
long int faktorial(int N); /* prototype fungsi factorial() */
void main()
{
    int N;
    long int fak;
    cout<<"Berapa factorial ? ";
    cin>>N;
    fak = faktorial(N); /* pemanggilan fungsi factorial() */
    cout<<N<<" factorial = "<< fak;
    getch();
}
long int faktorial(int N) /* definisi fungsi factorial */
{
    int I;
    long int F = 1;
    if(N<=0)
        return(0);
    for(I=2; I<=N; I++)
        F = F * I;
    return(F);
}
```

#### □ Parameter Formal dan Parameter Aktual

- ♦ **Parameter Formal** adalah variabel yang ada pada daftar parameter dalam definisi fungsi.
- ♦ **Parameter Aktual** adalah variabel (parameter) yang dipakai dalam pemanggilan fungsi.

Dalam contoh program pertambahan di atas parameter formal terdapat pada pendefinisian fungsi :

```
float tambah(float x, float y) //parameter formal
{
    return (a+b);
}
```

Sedangkan parameter aktual terdapat pada pemanggilan fungsi :

```

void main()
{
    .....
    c = tambah(a, b); //parameter aktual
    .....
}

```

#### □ Cara Melewatkan Parameter Secara Nilai (*Call by Value*)

- Call by value akan menyalin nilai dari parameter aktual ke parameter formal.
- Yang dikirimkan ke fungsi adalah nilai dari datanya, bukan alamat memori letak dari datanya.
- Fungsi yang menerima kiriman nilai akan menyimpannya di alamat terpisah dari nilai aslinya yang digunakan oleh bagian program yang memanggil fungsi.
- Perubahan nilai di fungsi (parameter formal) tidak akan merubah nilai asli di bagian program yang memanggilnya.
- Pengiriman parameter secara nilai adalah pengiriman searah, yaitu dari bagian program yang memanggil fungsi ke fungsi yang dipanggil.
- Pengiriman suatu nilai dapat dilakukan untuk suatu ungkapan, tidak hanya untuk sebuah variabel, elemen array atau konstanta saja.

*Contoh Program :*

```

void tukar(int x, int y); /* pendeklarasian fungsi */
void main()
{
    int a,b;
    clrscr();
    a = 15;
    b = 10;
    cout<<"Nilai sebelum pemanggilan fungsi\n";
    cout<<"a = "<<a<<" b = "<<b<<"\n\n"; // a dan b sebelum pemanggilan
fungsi
    tukar(a,b); /* pemanggilan fungsi tukar() */
    cout<<"Nilai setelah pemanggilan fungsi\n";
    cout<<"a = "<<a<<" b = "<<b<<"\n\n"; // a dan b setelah pemanggilan
fungsi
    getch();
}
void tukar(int x, int y) /* Pendefinisian fungsi tukar() */
{
    int z; /* variabel sementara */
    z = x;
    x = y;
    y = z;
    cout<<"Nilai di akhir fungsi tukar()\n";
    cout<<"x = "<<x<<" y = "<<y<<"\n\n";
}

```

Kuliah Berseri IlmuKomputer.Com  
 Copyright © 2003 IlmuKomputer.Com  
 Pemrograman Bahasa C dengan Turbo C  
 Achmad Solichin

#### C. CONTOH

Kerjakan contoh berikut untuk lebih memahami fungsi.

1. Program ini merupakan fungsi tidak mengembalikan nilai (void) dan dipanggil secara berulang-ulang sehingga meringankan penulisan program.

```
void box(int length, int width, int height); // box()'s prototype
```

```
int main()
{
    box(7, 20, 4);
    box(50, 3, 2);
    box(8, 6, 9);

    return 0;
}

// Compute the volume of a box.
void box(int length, int width, int height)
{
    cout << "volume of box is " << length * width * height << "\n";
}
```

2. Program ini merupakan fungsi tidak mengembalikan nilai (void) dengan 2 parameter.

```
void main()
{
    cout << "\nNow function test() will be called.\n";
    test( 10, -7.5); // Call
    cout << "\nAnd back again in main()." << endl;
    getch();
}

void test(int arg1, double arg2 ) // Definition
{
    cout << "\nIn function test()."
    << "\n 1. argument: " << arg1
    << "\n 2. argument: " << arg2 << endl;
}
```

3. Ini adalah contoh program menggunakan fungsi yang mengembalikan nilai bertipe double dan memiliki 2 parameter.

```
double area(double, double); // Prototype
void main()
{
    double x = 3.5, y = 7.2, res;
    res = area( x, y+1); // Call
    // To output to two decimal places:
    cout << fixed << setprecision(2);
    cout << "\n The area of a rectangle "
    << "\n with width " << setw(5) << x
    << "\n and length " << setw(5) << y+1
    << "\n is " << setw(5) << res
    << endl;
    getch();
}

// Defining the function area():
// Computes the area of a rectangle.
double area( double width, double len)
{
    return (width * len); // Returns the result.
}
```

4. Fungsi dapat dianggap sebagai ekspresi dan langsung dipanggil pada saat output.

```
// use double data
double box(double length, double width, double height);

int main()
{
    // use the return value of box( ) directly
    cout << "The volume is " << box(10.1, 11.2, 3.3);

    return 0;
}

// This version of box uses double data.
double box(double length, double width, double height)
{
    return length * width * height ;
}
```

#### D. LATIHAN

Untuk setiap soal di bawah ini, buatlah algoritmanya terlebih dahulu.

1. a. Berdasar soal no.2 pada modul 7. Perbaiki program sehingga kita memiliki fungsi yang memiliki parameter berupa tebakan bilangan. Dan fungsi mengembalikan nilai 1 jika benar dan 0 jika salah.  
b. Buatlah prototype fungsi tersebut  
c. Buatlah function main() untuk memanggil fungsi tersebut sama seperti pada modul 7.
2. a. Buat fungsi untuk menampilkan "\*" sebanyak parameter secara mendatar dan diakhiri dengan tanda "\n".  
b. Buatlah prototype fungsi tersebut  
c. Buatlah function main() untuk memanggil fungsi tersebut dengan nilai argumen berupa nilai yang dimulai dari satu hingga 10, setiap perulangan akan bertambah 1.
3. a. Buat fungsi untuk menghitung pangkat suatu bilangan tanpa menggunakan fungsi pow bawaan C++. Parameter fungsi adalah bilangan dan pangkatnya. Misal : pangkat (2,3)  
b. Buatlah prototype fungsi tersebut  
c. Buatlah function main() untuk memanggil fungsi tersebut dengan nilai bilangan yang akan dipangkat berupa masukan dari user kemudian menampilkan hasil dari pangkat.
4. a. Buat fungsi untuk menentukan apakah suatu bilangan bulat bersifat ganjil atau genap. Jika genap maka fungsi menghasilkan nilai 1, dan 0 untuk selainnya.  
b. Buatlah prototype fungsi tersebut  
c. Buatlah function main() untuk memanggil fungsi tersebut dengan nilai bilangan yang akan dicek berupa masukan dari user
5. a. Buatlah sebuah fungsi untuk menghitung nilai pangkat dari bilangan  $x^y$ . Fungsi memiliki parameter x dan y, dan mengembalikan hasil pangkat.  
b. Buatlah prototype fungsi tersebut.  
c. Buatlah function main() untuk memanggil fungsi pangkat dengan nilai x dan y berupa masukan dari user dan fungsi juga menampilkan hasil pangkat.

#### C. Laporan Praktikum

##### Format Laporan Praktikum

1. Tujuan Praktikum (5%)
2. Untuk setiap soal pada latihan , kerjakan algoritma (30%), capture program , beserta output. (30%)
3. Analisa setiap program untuk latihan dan contoh (secara keseluruhan program melakukan apa / apa ide dasar, perintah yang digunakan dan mengapa perintah tersebut dilakukan) (30%)
4. Daftar Pustaka (5%) => buku yang anda pakai untuk belajar

Catatan

- Laporan terlambat setelah 1 minggu tidak akan diberi nilai.

- Keterlambatan akan diberi nilai sebenarnya dan pengurangan 5 untuk setiap harinya.
- Laporan yang sama atau ditengarai sebagai hasil penjiplakan/plagiat maka tidak akan diberi nilai.

@ You are nothing if you don't do something @