



BAB X

FUNGSI

10.1 DASAR FUNGSI PADA BAHASA C

Fungsi merupakan suatu blok 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 dengan tujuan:

1. Program jadi terstruktur, sehingga mudah dipahami dan mudah dikembangkan.
2. Dapat mengurangi pengulangan kode (duplikasi kode).
3. Mempunyai readability yang tinggi

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.

10.2 FUNGSI PUSTAKA PADA BAHASA C

Fungsi merupakan suatu blok dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya terpisah dari program yang memanggilya.

Fungsi Operasi String (pada header file "string.h")

◆ strcpy()

- Berfungsi untuk menyalin suatu string asal ke variable string tujuan.
- Bentuk umum : **strcpy(var_tujuan, string_asal);**

Contoh :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>

main ()
{
    char a[50],b[50];

    clrscr();
    strcpy(a,"unikom");
    strcpy(b,"dipatiukur");
    puts(a);
    puts(b);
    puts("-----");

    strcpy(b,a);
    puts(a);
    puts(b);

    getch();
}
```

Program 10.1 Contoh strcpy()

◆ strlen()

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

C

contoh :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>

main()
{
    char a[50],b[50];

    clrscr();
    strcpy(a,"unikom");
    strcpy(b,"dipatiukur");
    puts(a);
    puts(b);
    puts("-----");

    printf("%s = %i huruf\n",a,strlen(a));
    printf("%s = %i huruf\n",b,strlen(b));

    getch();
}
```

Program 10.2 Contoh strlen()

◆ strcat()

- Digunakan untuk menambahkan string sumber ke bagian akhir dari string tujuan.
- Bentuk umum : **strcat(tujuan, sumber);**

Contoh :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>

main()
{
    char a[50],b[50];

    clrscr();
    strcpy(a,"unikom");
    strcpy(b,"dipatiukur");
    puts(a);
    puts(b);
    puts("-----");
}
```

```

    strcat (a,b) ;
    puts (a) ;
    puts (b) ;

getch () ;
}

```

Program 10.3 Contoh strcat()

◆ strupr()

- Digunakan untuk mengubah setiap huruf dari suatu string menjadi huruf kapital.
- Bentuk umum : **strupr(string)**;

Contoh :

```

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

main ()
{
    char a[50],b[50];

    clrscr ();
    strcpy (a, "unikom");
    strcpy (b, "dipatiukur");
    puts (a) ;
    puts (b) ;
    puts ("-----");

    strupr (b) ;
    puts (a) ;
    puts (b) ;

getch () ;
}

```

Program 10.4 Contoh strupr()

◆ strlwr()

- Digunakan untuk mengubah setiap huruf dari suatu string menjadi huruf kecil semua.
- Bentuk umum : **strlwr(string)**;

Contoh :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>

main()
{
    char a[50],b[50];

    clrscr();
    strcpy(a,"UNIKOM");
    strcpy(b,"DipAtiUkuR");
    puts(a);
    puts(b);
    puts("-----");

    strlwr(a);
    strlwr(b);
    puts(a);
    puts(b);

    getch();
}
```

Program 10.5 Contoh strlwr()

◆ strcmp()

- Digunakan untuk membandingkan dua buah string.
- Hasil dari fungsi ini bertipe integer dengan nilai :
 - (a) Negative, jika string pertama kurang dari string kedua.
 - (b) Nol, jika string pertama sama dengan string kedua
 - (c) Positif, jika string pertama lebih besar dari string kedua.
- Bentuk umum : **strcmp(string 1, string2);**

Contoh :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
```

```

main()
{
    char a[50],b[50];

    clrscr();
    strcpy(a,"UNIKOM");
    strcpy(b,"DIPATIUKUR");
    puts(a);
    puts(b);
    puts("-----");

    printf("Nilai strcmp(a,b) = %i\n",strcmp(a,b));
    printf("Nilai strcmp(b,a) = %i\n",strcmp(b,a));

getch();
}

```

Program 10.6 Contoh strcmp()

Fungsi Operasi Matematik (header "math.h" dan "stdlib.h")

♦ sqrt()

- Digunakan untuk menghitung akar dari sebuah bilangan.
- Bentuk umum : **sqrt(bilangan);**

Contoh :

```

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

main()
{
    int x,y;

    clrscr();
    x=100;
    y=sqrt(x);
    printf("Akar x=%i\n",y);
    printf("Nilai x=%i\n",x);

getch();
}

```

Program 10.7 Contoh sqrt()

◆ pow()

- Digunakan untuk menghitung pemangkatan suatu bilangan.
- Bentuk umum : **pow(bilangan, pangkat);**

Contoh :

```
#include <stdio.h>
#include <conio.h>
#include <math.h>

main()
{
    int x,y;

    clrscr();
    x=3;
    y=5;
    printf("3 pangkat 3 = %g\n",pow(x,3));
    printf("5 pangkat 2 = %g\n",pow(y,2));

    getch();
}
```

Program 10.8 Contoh pow()

◆ atof()

- Digunakan untuk mengkonversi nilai string menjadi bilangan bertipe double.
- Bentuk umum : **atof(char x);**

Contoh :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <math.h>

main()
{
    float x;
    char angka[3];
```

```

clrscr();

strcpy(angka, "25");
x=atof(angka);

printf("Angka Dalam String = %s\n", angka);
printf("Angka Dalam Float = %.0f\n", x);

getch();
}

```

Program 10.9 Contoh atof()

◆ atoi()

- Digunakan untuk mengkonversi nilai string menjadi bilangan bertipe integer.
- Bentuk umum : **atoi(char x)**;

Contoh :

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <math.h>

main()
{
    int x;
    char angka[3];

    clrscr();

    strcpy(angka, "345");
    x=atof(angka);

    printf("Angka Dalam String = %s\n", angka);
    printf("Angka Dalam Integer = %i\n", x);

    getch();
}

```

Program 10.10 Contoh atoi()

10.3 MEMBUAT FUNGSI PADA BAHASA C

Kita dapat membuat fungsi secara mandiri, yang tentunya, tugas dari fungsi tersebut dapat kita sesuaikan dengan kebutuhan yang kita inginkan. Namun sebelum kita membuat fungsi kita sendiri, ada hal-hal yang perlu diperhatikan, sebagai berikut;

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

```

#include <conio.h>
#include <conio.h>

void id() ; → Deklarasi fungsi

main()
{
  clrscr() ;
  id() ; → Memanggil fungsi

  getch() ;
}

void id()
{
  printf("=====") ;
  printf("==  Rio Yunanto  ==") ;
  printf("==    UNIKOM    ==") ;
  printf("=====") ;
}

```

Badan fungsi yang dibuat

Gambar 10.1 Cara Membuat, Mendeklarasikan, dan Memanggil Fungsi

10.4 PARAMETER FUNGSI

Pada umumnya fungsi memerlukan masukan yang nantinya akan diolah menjadi keluaran fungsi. Pintu masuk masukan inilah yang disebut parameter.



Gambar 10.2 Gambaran Parameter pada Fungsi

Parameter Formal dan Parameter Aktual

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

Contoh :

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

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

Pemanggilan 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 :

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

void hitung(int x, int y);

main()
{
    int a,b;
    clrscr();
    a = 15;
    b = 10;

    printf("Nilai sebelum pemanggilan fungsi\n");
    printf("a = %i b = %i\n\n", a, b);

    hitung(a,b);

    printf("Nilai setelah pemanggilan fungsi\n");
    printf("a = %i b = %i\n\n", a, b);

    getch();
}

void hitung(int x, int y)
{
    x=x*10;
    y=y+10;
    printf("Nilai di akhir fungsi hitung()\n");
    printf("x = %i y = %i\n\n", x, y);
}
```

Program 10.11 Contoh Parameter Call by Value

Pemanggilan Secara Referensi (*Call by Reference*)

Pemanggilan secara Referensi merupakan upaya untuk melewatkan alamat dari suatu variabel ke dalam fungsi. Yang dikirimkan ke fungsi adalah alamat letak dari nilai datanya, bukan nilaidatanya.

Fungsi yang menerima kiriman alamat ini makan menggunakan alamat yang sama untuk mendapatkan nilai datanya.

Perubahan nilai di fungsi akan merubah nilai asli di bagian program yang memanggil fungsi.

Pengiriman parameter secara referensi adalah pengiriman dua arah, yaitu dari fungsi pemanggil ke fungsi yang dipanggil dan juga sebaliknya. Pengiriman secara acuan tidak dapat dilakukan untuk suatu ungkapan.

Contoh :

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

void hitung(int *px, int *py);

main()
{
    int a,b;
    clrscr();
    a = 15;
    b = 10;

    printf("Nilai sebelum pemanggilan fungsi\n");
    printf("a = %i b = %i\n\n", a, b);

    hitung(&a, &b);

    printf("Nilai setelah pemanggilan fungsi\n");
    printf("a = %i b = %i\n\n", a, b);

    getch();
}

void hitung(int *px, int *py)
{
```

```
*px=*px*10;
*py=*py+10;
printf("Nilai di akhir fungsi hitung()\n");
printf("x = %i y = %i\n\n", *px, *py);
}
```

Program 10.12 Contoh Parameter Call by Reference



Try And Error..

LATIHAN - 10

Lat 10.1 :

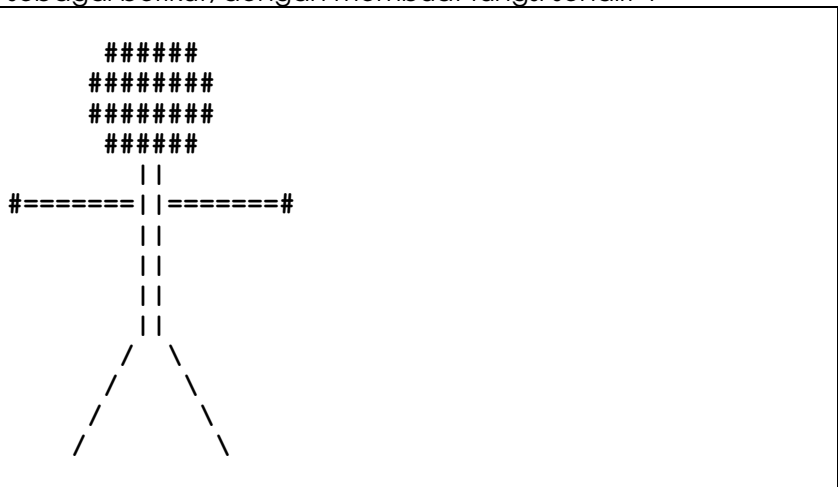
Jelaskan kelebihan parameter by reference dibandingkan dengan parameter by value ?

Lat 10.2 :

Ketika anda membuat program apakah harus membuat fungsi sendiri ? Mengapa ? Apa kelebihan program yang tidak memiliki fungsi sendiri dengan program yang tidak memiliki fungsi sendiri ?

Lat 10.3 :

Tuliskan Kode Program yang dapat menampilkan tampilan sebagai berikut, dengan membuat fungsi sendiri !



Lat 10.4 :

Buatlah fungsi yang dapat menkonversi huruf abjad (A..Z, a..z) menjadi bilangan desimal sesuai standart ASCII !

Lat 10.5 :

Buatlah fungsi yang dapat menkonversi bilangan desimal menjadi bilangan biner !

Lat 10.6 :

Gabungkan Lat 10.4 dan Lat 10.5 untuk membuat program yang dapat menkonversi teks menjadi bilangan biner !