

## BAB 2. FUNGSI INPUT OUTPUT

### 2.1 Fungsi Input

#### 1. Input Tanpa Format

##### a. Memasukkan Nilai Karakter

Fungsi yang digunakan :

getche() : memasukkan karakter tanpa penekanan Enter

getchar() : memasukkan karakter dengan penekanan Enter

getch() : memasukkan karakter dengan penekanan enter tanpa menampilkan di layar monitor.

##### Contoh :

```
#include <stdio.h>
#include <conio.h>
main()
{
    char C;
    printf("Masukan karakter : "); C = getche();
    printf("\nKarakter yang dimasukkan adalah : %c",C);
}
```

Out : Masukkan karakter : a

Karakter yang dimasukkan adalah : a

##### b. Memasukkan String

Fungsi yang digunakan adalah fungsi gets( ). Fungsi ini menggunakan argument variabel string untuk menerima data.

##### Contoh :

```
#include <stdio.h>
#include <conio.h>
main()
{
    char S[50];
    printf("Masukan String : "); gets(S);
    printf("String yang dimasukkan adalah : %s",S);
}
```

Out : Masukkan String : Bahasa C

String yang dimasukkan adalah : Bahasa C

#### 2. Input Data Terformat

Fungsi yang digunakan adalah **scanf( )**.

Tabel 2.1. Kode-kode format untuk scanf

Kode Format	Kegunaan
%c	Membaca sebuah karakter
%d	Membaca sebuah nilai integer decimal
%i	Membaca sebuah nilai integer decimal

%h	Membaca sebuah nilai short integer decimal
%e	Membaca sebuah data pecahan
%f	Membaca sebuah data pecahan
%g	Membaca sebuah data pecahan
%s	Membaca sebuah data string
%o	Membaca sebuah nilai octal
%x	Membaca sebuah nilai hexadecimal

Contoh input string :

```
#include "stdio.h"
main( )
{
    char S[80];
    printf("Masukkan string : "); scanf("%o[^\n]",S);
    printf("Nilai String adalah : %s \n", S);
}
```

Out : Masukkan string : saya suka kamu  
Nilai string adalah : saya suka kamu

Ket : %[^\n] digunakan menggantikan %s agar semua karakter termasuk spasi akan dibaca sampai ditemui penekan tombol enter, karena %s tidak akan membaca spasi atau tab. Untuk input string tidak perlu menggunakan operator pointer '&', karena pengenalan ini sudah berbentuk suatu pointer.

Contoh input numeric

```
#include <stdio.h>
main ( )
{
    float F;
    int I;
    char C;
    scanf("%4f %4d %c",&F,&I,&C);
    printf("Nilai yang diinput : %5.2f %i dan %c \n",F,I,C);
}
```

Out : Jika nilai yang diinput 12.345678 a  
Nilai yang diinput : 12.30 4567 dan 8

Ket : Pemisah nilai-nilai data yang dimasukkan tergantung dari pemisah kode format yang digunakan. Jika antara kode format digunakan pemisah spasi atau tanpa pemisah maka nilai data yang dimasukkan harus dipisahkan dengan *whitespace character* (blank, tabulasi atau enter).

## 2.2. Fungsi Output

### 1. Output Tanpa Format

Fungsi yang digunakan adalah :

Putch() : untuk menampiulkan karakter

Putchar() : untuk menampilkan nilai karakter.

Puts() : untuk menampilkan string.

#### Contoh :

```
char C, S[10] = "sebuah string
```

```
C = 'A';
```

```
putchar(C); puts(S);
```

**Output :** Asebuah string

### 2. Output dengan format

Fungsi yang digunakan adalah printf( )

Table 2.2. Kode-kode format untuk printf

Kode Format	Kegunaan
%c	Menampilkan sebuah karakter
%s	Menampilkan nilai string
%d	Menampilkan nilai integer decimal
%i	Menampilkan nilai integer decimal
%u	Menampilkan nilai integer decimal tak bertanda
%e	Menampilkan nilai pecahan dalam notasi saintific
%f	Menampilkan nilai pecahan
%g	Pengganti %f atau %e tergantung yg terpendek
%o	Menampilkan sebuah nilai octal
%x	Menampilkan nilai hexadecimal
%p	Menampilkan suatu alamat memory untuk pointer

#### ◆ Mencetak karakter atau string

Format penampilan :

%-Nc : menampilkan 1 karakter dengan lebar N karakter rata kiri..

%s : menampilkan semua karakter di nilai string rata kiri.

%Ns : menampilkan semua karakter rata kanan dengan lebar minimum N posisi.

%-Ns : menampilkan semua karakter rata kiri dengan lebar min N posisi.

%N.Ms : menampilkan rata kanan hanya M buah karakter pertama saja dengan lebar N posisi.

%-N.Ms : menampilkan rata kiri hanya M buah karakter pertama saja dengan lebar N posisi.

%.Ms : menampilkan sejumlah M karakter pertama rata kiri..

#### ◆ Menampilkan Nilai decimal Integer

Kombinasi kode format untuk integer :

%ld, %li : long int  
%hi : short int  
%hu : unsigned short int  
%lu : unsigned long int

Format penampilan :

%i : menampilkan semua digit rata kiri.  
%Ni : menampilkan semua digit rata kanan dengan lebar minimum N posisi.  
%-Ni : menampilkan semua digit rata kiri dengan lebar min N posisi.  
%0Ni : menampilkan rata kanan dengan lebar N digit, blank diisi dengan nol.  
%+Ni : menampilkan tanda positif pada numeric positif.

#### ◆ Menampilkan nilai pecahan

Kombinasi Kode format :

%lf atau %le atau %lg : double  
%Lf , %Le atau %Lg : long double

Format penampilan :

%f : menampilkan semua digit rata kiri.  
%Nf atau % -Nf : menampilkan semua digit rata kanan atau rata kiri dengan lebar minimum N digit.  
%N.Mf : menampilkan semua digit rata kanan dengan lebar min N digit dengan M digit nilai di belakang koma..  
%0Nf : menampilkan rata kanan dengan lebar minimum N digit, blank diisi dengan nol.  
%N.Of : menampilkan semua digit rata kanan dengan lebar min N digit tanpa pecahan.  
%.Mf : menampilkan nilai dengan M digit dibelakang desimal rata kiri.  
# : mencegah pembuangan angka-angka nol di belakang titik decimal.

#### ◆ Menampilkan Nilai Hexadesimal dan Oktal

%#x : menampilkan nilai hexa dengan tampilan diawali 0x  
%#o : menampilkan nilai octal dengan tampilan diawali 0

- ◆ clrscr() : perintah untuk membersihkan layar dan Prototype fungsi terdapat di file judul conio.h
- ◆ gotoxy(kol,bar) : mengatur posisi kursor pada kolom dan baris yang dituju.

#### Contoh :

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

```
void main ()  
{ char A ='A';  
  char B[9]="bahasa C";  
  int C = 1234;  
  int D = -400;
```

```
float E = 123.4567; float F = 12.0;
printf("%-5c\n",A);
printf("%-15s\n",B); printf("%15.3s\n",B);
printf("%3i\n",C); printf("%07d\n",C);printf("%-7d\n",C);printf("%+-3i\n",C);
printf("%+6i\n",D);printf("%-6d\n",D); printf("%u\n",D);
printf("%-15f\n",E); printf("%09.2f\n",E); printf("%15.0f\n",E);
printf("%7g dan %#7g\n",F,F); printf("%#x dan #o\n",C,C);
}
```

**output :**