FUNGSI INPUT DAN OUTPUT

3

Obyektif :

6. Mengetahui fungsi input dan fungsi output bahasa C
7. Dapat mengerti dan menggunakan fungsi input
8. Dapat mengerti dan menggunakan fungsi output

Fungsi Input

Fungsi-fungsi pustaka yang digunakan untuk memasukkan data melalui keyboard, prototypenya ada di file judul stdio.h dan conio.h. Fungsi-fungsi yang menggunakan file judul stdio.h yaitu gets() dan scanf(). Sedangkan fungsi yang menggunakan file judul conio.h, yaitu getche(), getchar(), dan getch().

Fungsi Input Data Tidak Terformat

Untuk memasukkan nilai karakter tidak terformat digunakan getchar() getch(), getche() dan memasukan nilai string tidak terformat yaitu gets, tergantung dari karakteristik masing-masing.

- getchar()

  Sintak: : int getchar(void):

  Fungsi:

  - mengembalikan sebuah karakter (nilai ASCII) berikutnya dari buffer keyboard.
  - Karakter ditampilkan di layar monitor
  - Menunggu sampai ada ENTER
  - Header file ada di stdio.h

- getch()

  Sintak: int getch(void):

Fungsi

- mengembalikan satu karakter dari buffer keyboard
- karakter tidak ditampilkan di layar monitor (no echo)
- Tidak menunggu sampai ada ENTER
- Cocok untuk membuat password
- Header file ada di conio.h

❖ getche()

Sintak : int getche(void)

Fungsi :

- mengembalikan satu karakter dari keyboard
- Karakter ditampilkan di layar (echo)
- Tidak menunggu sampai ada ENTER
- Header file ada di conio.h

Contoh:

```
#include <stdio.h>
#include<conio.h>
main()
{
        char A;
        printf("Masukan Nilai Sebuah Karakter?");
        A=getche();
        printf("\nNilai yang dimasukan adalah:%c\n",A);
}
```

❖ gets()

Sintak : char *gets(char *buffer)

Fungsi:

- membaca string dari keyboard sampai ketemu new-line dan disimpan pada buffer.
- Kemudian new-line di replace dengan null character

- Mengembalikan nilai NULL jika ada error dan mengembalikan argument-nya (buffer) jika sukses.

Fungsi Input Data Terformat

Untuk meg-input nilai data terformat digunakan perintah scanf(), Spesifikai *format* adalah : *"% type"* dimana *type* bisa diganti dengan salah satu dari sbb:

Kode Format	Fungsi
%c	Membaca sebuah karakter
%s	Membaca nilai string
%d	Membaca nilai desimal integer
%i	Membaca nilai desimal integer
%x	Membaca nilai heksa desimal integer
%o	Membaca nilai oktal integer
%f	Membaca nilai pecahan
%e	Membaca nilai pecahan
%g	Membaca nilai pecahan
%h	Membaca nilai short integer desimal
[…]	Membaca karakter string yg diakhiri dengan karakter yg tidak ada didalam [...]
[^..]	Membaca karakter string yg diakhiri dengan karakter yg ada didalam [..]

Fungsi scanf mengembalikan tipe integer, dimana nilai nya menyatakan jumlah field yang sukses di assigned. Contoh:

```
int x,y,z,w;
x=scanf("%d %d %d",&y,&z,&w);
```

maka :

- Jika di input dari keyboard 3 buah nilai interger 6 7 8, maka nilai x = 3;

- Jika di input dari keyboard 4 buah nilai interger 6 7 8 9 maka nilai x = 3 (karena 3 nilai yg sukses di- assigned masing-masing ke variabel y, z dan w)

Karakter Space, tab, linefeed, carriage-return, formfeed, vertical-tab, dan newline disebut *"white-space characters"* . Contoh :

```
char ss[40];
scanf("%s",ss);
```

Pada potongan program diatas, jika dimasukkan string "Selamat Pagi Pak" dari keyboard maka yg dimasukkan ke variabel ss hanya "Selamat" saja. Untuk mengambil string yang diakhiri karakter tertentu (misalnya ENTER), dengan scanf, menggunakan format [^\n]. Menjadi :

```
char ss[40];
scanf("%[^\n]",ss);
```

Fungsi Output

Prototype dari fungsi-fungsi untuk menampilkan hasil terdapat pada file judul stdio.h bersifat standar yaitu putchar(), puts(), printf(), printf() dan conio.h bersifat tidak standar, dalam arti tidak semua kompiler C menyediakan yaitu clrscr(), gotoxy().

Fungsi Output Hasil Tidak Terformat

Untuk menampilkan hasil tidak terformat digunakan putchar(), putch() dan puts(). Maksudnya tidak terformat adalah lebar dan bentuk tampilannya tidak dapat diatur.

- **putchar( )**

  Sintak: int putchar(int c)

  Fungsi :

  - untuk menampilkan karakter tidak terformat

- Menampilkan karakter ke layar monitor pada cursor, kemudian setelah ditampilkan cursor bergerak ke posisi berikutnya.
- Mengembalikan EOF jika error, dan mengembalikan karakter yang ditampilkan jika sukses
- Putchar adalah macro yang sama artinya dengan: putc(*c*, stdout)
- Header File : stdio.h

❖ **putch( )**

Sintak : int putch(int ch)

Fungsi :

- menampilkan karakter ascii di ch di monitor tanpa memindahkan kursor ke posisi berikutnya
- Header file : conio.h
- Mengembalikan EOF jika error, dan mengembalikan karakter yang di tampilkan jika sukses.

❖ **puts( )**

Sintak : int puts(const char *str);

Fungsi:

- Menampilkan string ke layar monitor dan memindahkan kursor ke baris baru.
- Header file: stdio.h
- Mengembalikan nilai non-negative jika sukses dan EOF jika ada error.

CONTOH :

```
puts(”Selamat Datang”);
puts(”Di GUNDAR”);
```

Tampilan di layar monitor :

Selamat Datang
Di GUNDAR

Penempatan kursor

- Layar dapat dihapus dengan menggunakan fungsi: *clrscr();*
- Kursor dapat dipindahkan ke posisi manapun di dalam layar monitor dengan menggunakan fungsi : *gotoxy(col,row);* dimana col = kolom dan row = baris
- Sebagian dari baris, mulai posisi kursor hingga akhir baris (end of line), dapat dihapus dengan fungsi: *clreol();*
- Function prototype untuk fungsi gotoxy(), clrscr(), clreol() pada bahasa C terdapat pada header file : <conio.h>

Fungsi Output Hasil Terformat

Sedangkan untuk hasil terformat digunakan perintah **printf** dengan spesifikai format sbb: *%[flags][width][.precision] type;*

Kode Format	Fungsi
%c	Menampilkan sebuah karakter
%s	Menampilkan nilai string
%d	Menampilkan nilai desimal integer
%i	Menampilkan nilai desimal integer
%u	Menampilkan nilai desimal integer tidak bertanda
%x	Menampilkan nilai heksa desimal integer
%o	Menampilkan nilai oktal integer
%f	Menampilkan nilai pecahan
%e	Menampilkan nilai pecahan dalam notasi scientific
%g	Sebagai pengganti ‘%f’ atau ‘%e’ tergantung mana yang terpendek
%p	Menampilkan suatu alamat memori untuk pointer

width : menentukan jumlah kolom yang disediakan

precision : menentukan jumlah angka dibelakang koma (untuk bilangan pecahan)

flags dapat diganti sbb:

none : right justify (rata kanan)

- : left justify (rata kiri)

+ : untuk bilangan dimulai dgn
tanda – jika negatip atau +
jika positip

CONTOH 1:

```
printf(“%6d”, 34);          ….34
printf(”%-6d”, 34);         34….
```

CONTOH 2 :

```
printf(“%10s”, “GUNDAR”);       …GUNDAR
printf(“%-10s”, “GUNDAR”);          GUNDAR . . .
printf(“%8.2f”, 3.14159 );      ....3.14
printf(“%-8.3f”, 3.14159 );     3.141...
printf("%c\n",65);    //akan ditampilkan A
printf("%x\n",'A');    // akan ditampilkan 41
printf("%o\n",65);   // akan ditampilkan 101
printf("%+d\n",34);  // akan ditampilkan +34
printf("%+d\n",-45); // akan ditampilkan -45
printf("%e\n",3.14); // akan ditampilkan   3.140000e+000
```

CONTOH 3:

```
#include <stdio.h> Laboratorium Sistem Informasi
int main(){
   char ss[]="Selamat Datang";
   printf("12345678901234567890123456789O\n");
```

```
    printf("%.10s di Gundar\n",ss);
    printf("%10s di Gundar\n",ss);
    printf("%-10s di Gundar\n",ss);
    printf("%.20s di Gundar\n",ss);
    printf("%20s di Gundar\n",ss);
    printf("%-20s di Gundar\n",ss);
    printf("%20.10s di Gundar\n",ss);
    printf("%-20.10s di Gundar\n",ss);
    return 0;
}
```

Output Program di atas sbb:

```
12345678901234567890123456789
Selamat Da di Gundar
Selamat Datang di Gundar
Selamat Datang di Gundar
Selamat Datang di Gundar
     Selamat Datang di Gundar
Selamat Datang       di Gundar
         Selamat Da di Gundar
Selamat Da          di G
```