

PRAKTIKUM 7

MASUKAN DAN KELUARAN PROGRAM

7.1 TUJUAN PRAKTIKUM

Tujuan Umum

Mahasiswa dapat memahami:

1. Penggunaan perintah Input (masukan) pada sebuah program.
2. Penggunaan perintah Output (keluaran) untuk menampilkan berbagai data pada sebuah program.
3. Penggunaan format cetakan berbagai variabel.

Tujuan Khusus

Mahasiswa dapat :

1. Menuliskan perintah intruksi input pada sebuah program.
2. Mengetahui dan menggunakan berbagai format inputan
3. Menyusun berbagai program yang menginput dan menampilkan berbagai data tipe data.
4. Membaca dan menjelaskan maksud dari suatu program.

7.2 TEORI SINGKAT

7.2.1 Perintah Masukan

Dalam bahasa C proses memasukkan suatu data bisa menggunakan beberapa fungsi pustaka yang telah tersedia. Beberapa fungsi pustaka yang bisa digunakan adalah :

Fungsi **scanf()**

Fungsi pustaka `scanf()` digunakan untuk menginput data berupa data numerik, karakter dan string secara terformat.

Hal-hal yang perlu diperhatikan dalam pemakaian fungsi `scanf()` :

- Fungsi `scanf()` memakai penentu format
- Fungsi `scanf()` memberi pergantian baris secara otomatis
- Fungsi `scanf()` tidak memerlukan penentu lebar field
- Variabelnya harus menggunakan operator alamat &

NO	FORMAT	KETERANGAN	CONTOH
1	%c	Membaca sebuah karakter	<code>scanf("%c", &N);</code>
2	%s	Membaca sebuah string	<code>scanf("%s", &N);</code>
3	%i, %d	Membaca sebuah bilangan bulat (integer)	<code>scanf("%i", &N);</code>

NO	FORMAT	KETERANGAN	CONTOH
4	%f, %e	Membaca sebuah bilangan pecahan (real)	scanf("%f", &N);
5	%o	Membaca sebuah bilangan basis octal	scanf("%o", &N);
6	%x	Membaca sebuah bilangan basis heksadesimal	scanf("%x", &N);
7	%u	Membaca sebuah bilangan tak bertanda	scanf("%u", &N);

Fungsi gets()

- Fungsi gets() digunakan untuk memasukkan data bertipe karakter dan tidak dapat digunakan untuk memasukkan data numerik.
- Harus diakhiri dengan penekanan tombol enter
- Cursor secara otomatis akan pindah baris
- Tidak memerlukan penentu format

Fungsi getchar()

- Fungsi getchar() digunakan untuk mengambil satu karakter dari inputan
- Harus diakhiri dengan penekanan tombol enter
- Karakter yang dimasukkan terlihat pada layar
- Pergantian baris secara otomatis

Fungsi getch() dan getche()

- Fungsi getch() dan getche() digunakan untuk membaca data karakter
- Karakter yang dimasukkan tidak perlu diakhiri dengan penekanan tombol enter
- Tidak memberikan efek pergantian baris secara otomatis
- Jika menggunakan fungsi getch() karakter yang dimasukkan tidak akan ditampilkan pada layar sehingga sering digunakan untuk meminta inputan berupa password.
- Sedangkan pada getche() karakter yang dimasukkan akan ditampilkan pada layar.

7.2.2 Perintah Keluaran

Menampilkan data ke layar monitor

- Menggunakan fungsi **printf()**, **puts()**, dan **putchar()**.
- Fungsi printf() digunakan untuk menampilkan semua jenis data (numerik dan karakter)
- Fungsi puts() digunakan untuk menampilkan data string dan secara otomatis akan diakhiri dengan perpindahan baris.
- Fungsi putchar() digunakan untuk menampilkan sebuah karakter.

Mengatur tampilan bilangan pecahan (float).

Bentuk umum :

```
printf("%m.nf, var);
```

dimana:

- m : menyatakan panjang range
- n : menyatakan jumlah digit di belakang koma.
- var : nilai atau variable yang akan

ditampilkan. Contoh :

```
printf("%5.2f", nilai);
```

artinya variable nilai akan ditampilkan sebanyak 5 digit dengan 2 digit di belakang koma.

7.3 PELAKSANAAN PRAKTIKUM

1. Tuliskan Program 7.1 berikut ini pada editor Dev-C++

Program 7.1 inout1.cpp

```
1 #include <stdio.h>
2
3 main()
4 {
5     char nim[10], nama[30];
6     printf("INPUT DATA MAHASISWA\n");
7     printf("NIM : "); scanf("%s", &nim);
8     printf("NAMA : "); scanf("%s", &nama);
9
10    //tampilkan
11    printf("\nNIM : %s", nim);
12    printf("\nNAMA : %s", nama);
13 }
```

2. Jalankan Program 7.1 di atas dan inputkan NIM dan NAMA Anda masing-masing. Apakah ada hasil yang "tidak sesuai" ?

3. Sekarang ubah Program 7.1 menjadi Program 7.2 berikut ini. Fungsi scanf() untuk menginput NAMA diganti dengan fungsi gets().

Program 7.2 inout2.cpp

```
1 #include <stdio.h>
2 main()
3 {
4     char nim[10], nama[30];
5     printf("INPUT DATA MAHASISWA\n");
6     printf("NIM : "); scanf("%s", &nim);
7     printf("NAMA : "); gets(nama);
8
9     //tampilkan
10    printf("\nNIM : %s", nim);
11    printf("\nNAMA : %s", nama);
12 }
```

4. Lakukan kompilasi dan jalankan Program 7.2 lalu inputkan kembali NIM dan NAMA Anda. Apa yang terjadi? Apakah berhasil?
5. Sekarang tambahkan perintah `fflush(stdin);` setelah perintah untuk menginput NIM. Perhatikan Program 7.3 berikut ini.

Program 7.3 inout3.cpp

```
1  #include <stdio.h>
2  main()
3  {
4      char nim[10], nama[30];
5      printf("INPUT DATA MAHASISWA\n");
6      printf("NIM : "); scanf("%s", &nim);
7      fflush(stdin);
8      printf("NAMA : "); gets(nama);
9
10     //tampilkan
11     printf("\nNIM : %s", nim);
12     printf("\nNAMA : %s", nama);
13 }
```

6. Jalankan Program 7.3 di atas dan inputkan NIM dan NAMA Anda masing-masing. Tuliskan apa yang tercetak di layar! Perintah `fflush(stdin)` berfungsi menghapus buffer I/O di dalam memori. Fungsi dapat ditambahkan setelah perintah inputan.
7. Selanjutnya ditambahkan variabel "nilai" pada Program 7.3 untuk menginput dan menyimpan nilai mahasiswa. Variabel "nilai" dideklarasikan sebagai variabel yang bertipe float (pecahan). Perhatikan Program 7.4 di bawah.

Program 7.4 inout4.cpp

```
1  #include <stdio.h>
2  main()
3  {
4      char nim[10], nama[30];
5      float nilai;
6      printf("INPUT DATA MAHASISWA\n");
7      printf("NIM : "); scanf("%s", &nim);
8      fflush(stdin);
9      printf("NAMA : "); gets(nama);
10     printf("NILAI : "); scanf("%f", &nilai);
11 }
```

```

12 //tampilkan
13 printf("\nNIM : %s", nim);
14 printf("\nNAMA : %s", nama);
15 printf("\nNILAI : %f", nilai);
16 printf("\nNILAI (PEMBULATAN) : %.2f", nilai);
17 }

```

8. Jalankan Program 7.4 dan pada inputan nilai, masukkan beberapa nilai berikut misal : 80, 78.253, 87.243, 90.55.

7.4 LATIHAN

Tuliskan dan jalankan beberapa program berikut ini dan tuliskan hasilnya di tempat yang sudah disediakan.

Program 7.5 lingkaran_in.cpp

```

1  #include <stdio.h>
2  main()
3  {
4      int jari;
5      float luas, keliling;
6
7      printf("Input jari-jari lingkaran : ");
8      scanf("%i", &jari);
9
10     luas = 3.14 * jari * jari;
11     keliling = 2 * 3.14 * jari;
12
13     printf("LUAS dan KELILING LINGKARAN");
14     printf("\nJari-jari = %i", jari);
15     printf("\nLUAS = %.3f", luas);
16     printf("\nKELILING = %.2f", keliling);
17 }

```

Program 7.6 volumebola.cpp

```
1  #include <stdio.h>
2  #define PHI 3.14
3  main()
4  {
5      float jari;
6      float volume;
7
8      printf("PROGRAM MENGHITUNG VOLUME BOLA\n\n");
9      printf("Input jari-jari bola (cm) : ");
10     scanf("%f", &jari);
11
12     //hitung volume
13     volume = 4/3 * PHI * jari * jari * jari;
14
15     printf("\nVolume bola dg jari-jari %.2f cm ");
16     printf("adalah %.3f cm3.", jari, volume);
17 }
```

7.5 TUGAS MANDIRI

Kerjakan soal-soal berikut ini:

1. Buatlah sebuah program Bahasa C untuk menginput panjang alas dan tinggi segitiga (dalam cm). Lalu hitung dan tampilkan luas dan keliling dari segitiga tersebut!
2. Buatlah sebuah program Bahasa C untuk menginput sebuah nilai bilangan bulat yang menyatakan suhu dalam satuan Celcius. Selanjutnya hitung dan tampilkan nilai sudut dalam derajat Fahrenheit dan Reamur!
3. Buatlah sebuah program Bahasa C untuk menginput sisi alas dan sisi tinggi sebuah segitiga siku-siku (dalam cm). Dengan menggunakan rumus Phitagoras, buatlah sebuah program Bahasa C untuk menghitung sisi miring segitiga tersebut dan menampilkannya di layar!
4. Buatlah sebuah program Bahasa C untuk menginput sebuah nilai bilangan bulat positif, lalu tampilkan keterangan "GANJIL" jika bilangan tersebut adalah ganjil dan "GENAP" jika bilangan tersebut adalah genap.