

Input Output Dalam Bahasa C++

Proses input dan output identik dengan penggunaan keyboard untuk input dan tampilan layar monitor untuk output. Dalam bahasa C++ terdapat 2 standar input dan output dasar yang sering digunakan dengan format yang berbeda, yaitu `stdio` dan `iostream`.

1. Input (Memasukkan Data)

Menggunakan `#include <stdio.h>`

Fungsi `stdio.h` yang merupakan singkatan dari "standard input / output header", adalah header di C

library standar yang berisi definisi makro, konstanta, dan deklarasi fungsi dan jenis yang digunakan

untuk berbagai standard operasi input dan output.

a. Fungsi `scanf()`

fungsi `scanf()` berguna untuk memasukkan data baik berupa bilangan, karakter, ataupun kalimat secara terformat. Berikut format-format yang digunakan untuk `scanf()` :

`%c` : Membaca sebuah karakter

`%s` : Membaca sebuah string

`%i`, `%d` : Membaca sebuah bilangan bulat (integer, desimal)

`%f`, `%e` : Membaca sebuah bilangan pecahan (real, float)

`%o` : Membaca sebuah bilangan octal

`%x` : Membaca sebuah bilangan heksadesimal

`%u` : Membaca sebuah bilangan tak bertanda

b. Fungsi `gets()`

fungsi `gets()` berguna untuk memasukkan data bertipe karakter, tanpa penggunaan format seperti `scanf()`, dan tidak dapat untuk menginput data numeric ataupun string.

c. Fungsi `getchar()`

fungsi `getchar` berguna untuk membaca data yang bertipe karakter.

Menggunakan `#include <iostream.h>`

Fungsi `iostream.h` sama dengan `stdio.h` yg merupakan header pada bahasa C++.

a. Fungsi `cin >> var`

fungsi `cin >>` berguna untuk menginput data berupa numerik, string, dan karakter. Var dapat lebih dari satu variabel (`cin >> var >> var`) dan dapat berupa karakter.

b. Format : `cin.get.(var, sizeof(var))`,

penggunaan format ini untuk menginput string lebih dari satu kata atau seperti kita menekan spasi, tetapi akan terjadi kesalahan jika satu program digunakan lebih dari satu kali. Dan untuk menanganinya lebih baik menggunakan format `cin.getline(var, sizeof(var))`.

c. Format : `cin.getline(var, sizeof(var))`.

2. Output (Menampilkan Data)

Menggunakan `#include <stdio.h>`

- a. Menampilkan output ke layar monitor
 - Fungsi `printf()` berguna untuk menampilkan semua jenis data (number, string, dan karakter)
 - Fungsi `puts()` digunakan untuk menampilkan data string dan secara otomatis akan pindah baris (new line).
 - Fungsi `putchar()` digunakan untuk menampilkan sebuah karakter.

- b. Mengatur tampilan output

Bentuk sintaks : `printf("%m,nf", var)`

Keterangan :

m : Menyatakan banyaknya digit angka

n : Menyatakan banyaknya digit angka dibelakang koma

var : variabel bertipe float yang akan ditampilkan

Contoh

`printf("%5.2f", nilai);` → mencetak angka sebanyak 5 digit dan 2 digit di belakang koma

Menggunakan `#include <iostream.h>`

Fungsi `cout <<` dapat digunakan untuk menampilkan semua jenis data (numeric dan karakter) yang dapat berupa nilai, karakter ataupun konstanta.

Contoh Program

Menggunakan `#include <stdio.h>`

```
/* Program Sederhana Input dan Output menggunakan stdio.h */
#include <stdio.h>
#include <conio.h>
main()
{
    int umur;
    char nama[10];
    float tinggi;
    clrscr();
    printf("Masukkan nama : ");
    scanf("%s",&nama); // membaca nilai string
    printf("Masukkan umur : ");
    scanf("%d",&umur); // membaca nilai integer / decimal
    printf("Masukkan tinggi : ");
    scanf("%f",&tinggi); // membaca nilai float / real
    printf("Nilai dari variabel yang anda input : \n");
    printf("Nama : %s\n", nama);
    printf("Umur : %d\n", umur);
    printf("Tinggi : %5.2f", tinggi);
    getch(); }
```

Menggunakan `#include <iostream.h>`

```
/* Program Sederhana Input dan Output menggunakan iostream.h */
#include <iostream.h>
#include <conio.h>
main()
{
    int umur;
    char nama[10];
    float tinggi;
    clrscr();
    cout << "Masukkan nama : ";
    cin >> nama; // membaca nilai string
    cout << "Masukkan umur : ";
    cin >> umur; // membaca nilai integer / decimal
    cout << "Masukkan tinggi : ";
    cin >> tinggi; // membaca nilai float / real
    cout << "Nilai dari variabel yang anda input : \n";
    cout << "Nama : " << nama;
    cout << "\nUmur : " << umur;
    cout << "\nTinggi : " << tinggi;
    getch();
}
```

Jadi :

Untuk membaca atau menampilkan data pada console (layar) :

- dengan library `<stdio.h>` → ***printf & scanf***
- dengan library `<iostream.h>` → ***cout & cin***

Berikutnya kita bahas penerapannya pada struktur runtunan dapat dilihat pada contoh program 1a di bawah ini :

```
#include <iostream>
#include <conio.h>
#include <math.h>
void main () {
    float a,b,c,x1,x2;
    cout<<"masukkan nilai a : ";
    cin>>a;
    cout<<"masukkan nilai b : ";
    cin>>b;
    cout<<"masukkan nilai c : ";
    cin>>c;
    cout<<endl<<endl;
    x1=(b+sqrt(b*b-4*a*c))/2*a*c;
    x2=(b-sqrt(b*b-4*a*c))/2*a*c;
    cout<<"x1="<<x1<<endl<<endl;
    cout<<"x2="<<x2<<endl<<endl;
    getch ();
}
```

Program di atas dimulai dengan identifikasi beberapa variabel yang dibutuhkan, kemudian proses input nilai a, b, dan c oleh user saat program dijalankan. Berikutnya menghitung rumus persamaan x1 dan x2 berdasarkan nilai a,b,c yang telah diinputkan. Terakhir mencetak hasilnya (x1 dan x2) sesuai nilai yang diinputkan oleh user.

Struktur perintah di dalam program tersebut beruntun, berurutan dari atas ke bawah, dan tidak terdapat proses memilih / bercabang dan proses berulang → struktur RUNTUNAN.

Program di atas dapat dikembangkan menjadi berstruktur pemilihan/percabangan, menggunakan perintah **if – else**. Pada struktur pemilihan atau percabangan, terdapat pemeriksaan kondisi / syarat yang harus dipenuhi, yang kemudian akan memilih perintah apa yang akan dilakukan jika syarat tersebut dipenuhi. Perintah tidak lagi dikerjakan secara beruntun seperti pada struktur runtunan, tetapi berdasarkan syarat yang harus dipenuhi.

Contoh program mencari akar persamaan kuadrat di bawah ini (pengembangan dari program 1a di atas), terdapat tambahan perintah untuk memeriksa kondisi yang harus dipenuhi, dan memilih perintah tertentu jika kondisi terpenuhi (**if – else**).

```
//program 1b
#include <iostream>
#include <math.h>
using namespace std;
int main(){
    int a, b, c, D;
    float x1, x2;

    cout<<"Masukan nilai a : ";
    cin>>a;
    cout<<"Masukan nilai b : ";
    cin>>b;
    cout<<"Masukan nilai c : ";
    cin>>c;

    D=(b*b)-(4*a*c);
    if (D>0){
        x1 = (-b + sqrt(D)) / (2*a);
        x2 = (-b - sqrt(D)) / (2*a);

    } else if (D==0){
        x1 = (-b + sqrt(D)) / (2*a);
        x2 = x1; }
    else {
        cout<<"\nAkar Imajiner"<<endl;
        exit(0); }
    cout<<"X1 = "<<x1<<endl;
    cout<<"X2 = "<<x2<<endl;
}
```

Penjelasan program 1b:

- Program di atas menggunakan perintah if untuk menerjemahkan struktur pemilihan masing-masing nilai yang dihasilkan (variabel d).
- Terdapat method sqrt () yang diambil dari modul math, yg berfungsi untuk menghitung akar kuadrat dari suatu bilangan.
- Selain itu terdapat juga fungsi exit() untuk mengentikan program agar tidak mengeksekusi pernyataan setelahnya.
- Modul lainnya yang disertakan adalah iostram untuk menangani input/output program, terdapat 4 variabel dengan perincian sebagai berikut :
 - Empat variabel bertipe integer yaitu : a, b, c, d
 - Dua variabel bertipe float yaitu : x1, x2

Nilai a, b dan c dimasukkan dari keyboard saat program di jalankan. Program akan menghitung nilai diskriminan sesuai rumusnya $D=(b*b)-(4a*c)$. Untuk menentukan akar-akar persamaan kuadrat dari nilai diskriminan.

Contoh : dimasukkan nilai a=1, b=2 dan c= -3, maka program akan menghitung akar-akar persamaan $x^2 + 2x - 3 = 0$

Outputnya :

```
Masukan nilai a : 1
Masukan nilai b : 2
Masukan nilai c : -3
X1 = 1
X2 = -3
-----
```