***1. Perintah Output***

Perintah Output adalah perintah yang digunakan untuk mengeluarkan hasil proses komputer sehingga bisa dibaca oleh si pemakai (user). Adapun media yang digunakan untuk menampilkan hasil output tadi bisa berupa monitor atau dicetak ke kertas melalui printer. Salah satu fungsi untuk menampilkan output dalam Turbo C++ adalah :

a.printf().

Fungsi printf() digunakan untuk mencetak data baik berupa teks, numerik, konstanta maupun variabel. Contoh penggunaan printf() bisa dilihat pada contoh program berikut :



Bisa dilihat dari contoh diatas, bahwa perintah printf() bisa diikuti dengan tanda %s, %f, %d dan lain-lain. Tanda % tersebut disebut dengan penentu format (format specifier)  Fungsi printf() digunakan untuk menampilkan semua jenis data (numeric dan karakter)

Perintah lain untuk menampilkan output adalah dengan puts() dan putchar().

b. Fungsi puts() digunakan untuk menampilkan data string dan secara otomatis akan diakhir dengan perpindahan baris.

c. Fungsi putchar() digunakan untuk menampilkan sebuah karakter.berikut adalah contoh penggunaan putchar() dan puts()

[](http://2.bp.blogspot.com/-8jBMB2lWdW0/Uu5aO2VsrZI/AAAAAAAAAC0/56pPIMMEKDA/s1600/input%2B2.jpg)

**d.  Fungsi cout**

Penggunaan**cout** stream dhubungkan dengan operator overloaded  << (Sepasang tanda *"less than"*).  Contoh     :

cout << "Nama Saya Afandi"; *// Tampilkan Output Nama Saya Afandi*

cout << 1000;               *// Tampilkan angka 1000 pada layar*

cout << x;                 *// Tampilkan variabel x on pada layar*

Operator << dikenal sebagai *insertion operator,* dimana berfungsi untuk menginput data yang mengikutinya. Jika berupa string, maka harus diapit dengan kutip ganda ("), sehingga membedakannya dari variable. Contoh

cout << "Hello";      *// Tampilkan Hello pada layar*

cout << Hello;        *// Tampilkan variabel Hello pada layar*

Operator *insertion* (<<) dapat digunakan lebih dari 1 kali dalam kalimat yang sama, Contoh  :

cout << "Hallo, " << "nama " << "Saya Afandi";

Contoh diatas akan menampilkan  **Hallo nama Saya Afandi sentence** pada layar monitor. Manfaat dari pengulangan penggunaan operator insertion (<<) adalah untuk menampilkan kombinasi dari satu variabel dan konstanta atau lebih, contoh                      :

cout << "Hello I am " << age << " years old and my zipcode is " << zipcode;

Misalkan variable age = 24 dan  variable zipcode = 90064 maka output yang dihasilkan :

Hello, I am 24 years old and my zipcode is 90064

Contoh :

cout << "Baris pertama \n ";
cout << "Baris kedua.\nBaris ketiga.";

Output :

Baris pertama.
Baris kedua.
Baris ketiga.

Selain dengan karakter new-line, dapat juga menggunakan manipulator **endl,** contoh    :

cout << "Baris pertama" << endl;
cout << "Baris kedua" << endl;

Output :

Baris pertamaBaris kedua

***2. Perintah Input***
Setiap bahasa pemrograman tidak akan bisa digunakan secara fleksibel jika tidak memiliki perintah input. Perintah input adalah sebuah perintah dalam bahasa program yang mampu meneruskan nilai dari operator untuk diproses oleh komputer. Perintah input memerlukan perangkat keras input, biasanya adalah keyboard. Dalam Turbo C++, terdapat tiga perintah input yaitu scanf(), getche(), getch() dan gets().

***a. Fungsi scanf()***
Bentuk umum dari fungsi scanf() adalah sebagai berikut :



Penggunaan scanf() biasanya dikombinasikan dengan perintah printf(). Perintah printf() disini berfungsi sekedar menampilkan keterangan tentang apa yang harus diinputkan, sehingga operator bisa langsung mengerti harus memasukkan data apa. Untuk penentu format pada scanf(), dapat dilihat pada tabel berikut :
Tabel 3.3 Tabel Penentu Format scanf()



Selain itu, penggunaan scanf() juga harus menyertakan tanda ‘&’ pada awal nama variabel. Tanda ‘&’ disini befungsi sebagai operator alamat (address operator).
Contoh penggunaan scanf() sebagai berikut :



Jika terdapat beberapa proses input (memasukkan data) sekaligus, maka sebaiknya ditambahkan fungsi fflush(stdin); setelah fungsi scanf(). Fungsi fflush(stdin) berfungsi menghapus buffer di dalam alat I/O.

Scanf() juga bisa digunakan untuk menginputkan beberapa data sekaligus dalam satu baris asalkan jumlah dan tipe penentu format sesuai dengan variabel yang akan diinputkan. Data yang akan dimasukkan dapat dipisahkan dengan spasi, tab atau tanda pemisah lain seperti koma (,), garis hubung(-), atau titik dua(:). Pemisah data dalam input yang digunakan harus sama dengan pemisah data dalam scanf().



**b.Fungsi getche()**

    Fungsi input getche() memiliki sifat yang sedikit berbeda dari scanf(). Perbedaan tersebut antara lain :
  1. Bila dalam *scanf()* jumlah karakter data yang diinputkan boleh bebas, maka dalam *getche()*hanya sebuah karakter yang bisa diterima.
  2. Bila *scanf()* membutuhkan tombol RETURN/ENTER untuk mengakhiri input, maka dalam getche() tidak membutuhkannya. Input dianggap selesai begitu kita memasukkan satu karakter dan secara otomatis akan melanjutkan ke baris perintah berikutnya.

 getche() merupakan singkatan dari get character and echo yang artinya ‘menerima sebuah karakter kemudian     tampilkan’. Input yang diterima getche() akan disimpan ke dalam variabel karakter yang sebelumnya harus sudah dideklarasikan.

 **c.Fungsigetch().**

Satu-satunya perbedaan antara getche() dan getch() adalah getche()akan menampilkan karakter yang kita ketikkan, sedangkan getch() tidak akan menampilkan, melainkan hanya menyimpannya dalam memori saja, jadi apa yang kita ketikkan tidak akan muncul dilayar sebelum kita memberikan perintah untuk mencetak nilai tersebut.
  a. Fungsi getch() dan getche() digunakan untuk membaca data karakter.
  b. Karakter yang dimasukkan tidak perlu diakhiri dengan penekanan tombol enter.
  c. Tidak memberikan efek pergantian baris secara otomatis
 d. Jika menggunakan fungsi getch() karakter yang dimasukkan tidak akan ditampilkan pada layar sehingga sering digunakan untuk meminta inputan berupa password.
 e. Sedangkan pada getche() karakter yang dimasukkan akan ditampilkan pada layar



***d. Fungsi gets()***
  a. Fungsi gets() digunakan untuk memasukkan data bertipe karakter dan tidak dapat digunakan untuk          memasukkan data numerik.
  b. Harus diakhiri dengan penekanan tombol enter
  c. Cursor secara otomatis akan pindah baris

  d. Tidak memerlukan penentu format



**e. Fungsi cin**

Menangani standard input pada C++ dengan menambahkan overloaded operator *extraction* (>>) pada **cin** stream. Harus diikuti bengan variable yang akan menyimpan data. Contoh       :

int umur;
cin >> umur;

Contoh diatas mendeklarasikan variabel age dengan tipe int dan menunggu input dari cin (keyborad) untuk disimpan di variabel umur.

**cin** akan memproses input dari keyboard sekali saja dan tombol ENTER harus ditekan.

Contoh :

1. // contoh program input dan output
2. #include <iostream.h>
3. int main ()
4. {
5. int i;
6. cout << "Silahkan masukan nilai: ";
7. cin >> i;
8. cout << "Nilai adalah " << i;
9. cout << " Nilai jika di kali 5 adalah: " << i\*5 << ".**\n**";
10. return 0;
11. }

Output :

**Silahkan masukan nilai:**2
**Nilai jika di kali 5 adalah: 10.**

**cin** juga dapat digunakan untuk lebih dari satu input    :

cin >> a >> b;

Equivalen dengan          :

cin >> a;
cin >> b;

Dalam hal ini data yang di input harus 2, satu untuk variabel **a** dan lainnya untuk variabel **b** yang penulisannya dipisahkan dengan :  spasi, tabular atau *newline*.

**Input dan Output pada C++**

Berbicara tentang input dan ouput data dalam C++ tidak jauh dari penggunaan keyboard untuk input dan tampilan layar monitor untuk output. Berikut adalah pengenalan input dan output pada C++.

**INPUT(memasukkan data)**

**A. Cin**
Dalam C++, perintah cin digunakan untuk menginput suatu nilai dari suatu piranti masukan (keyboard) untuk selanjutnya diproses oleh program. Sintaknya adalah:

cin >> variabel;
contohnya:
cout << “Masukkan suatu bilangan : ”;
cin >> bil;
cout << “Anda memasukkan bilangan “ << bil “\n”;

**B. getch()**

Perintah getch() berfungsi sama seperti cin (perintah input), akan tetapi getch() khusus untuk input berupa karakter. Disamping itu getch() dapat membaca input berupa spasi atau tab, sedangkan cin tidak bisa.
Sintaksnya:

Variabel = getch();

Fungsi ini juga dapat digunakan apabila tidak diinginkan penekanan ENTER ketika input data karakter.

**C. getche()**
Perintah getche() kegunaannya sama dengan getch(), bedanya adalah:

* getch() tidak menampilkan karakter yang diiinput
* getche() menampilkan karakter yang diinput

fungsi getch() dan getche() sama-sama membutuhkan file header conio.h .Berikut ini contoh penggunaan getch() dan getche()

char karakter;
cout << “masukkan sebuah karakter : “;
karakter = getch();
cout << “Anda mengetik karakter : ” << karakter;
cout << “masukkan sebuah karakter : “;
karakter = getche();
cout << “Anda mengetik karakter : ” << karakter;

**OUTPUT(menampilkan data)**

**Cout**

Dalam c++, perintah cout digunakan untuk menampilkan suatu informasi ke piranti output (layar). Contoh-contoh penggunaannya telah banyak diberikan di bab-bab sebelumnya. Berikut ini perintah-perintah tambahan untuk mengatur tampilan output.

endl
Perintah ini berfungsi sama dengan \n (ganti baris)
Contoh:

cout << “Hallo” << endl;

setw()
Perintah ini digunakan untuk mengatur lebar tampilan data (rata kanan).
Contoh:

A = 123;
B = 98;
C = 1;
cout << “Nilai A = “ << setw(6) << A << endl;
cout << “Nilai B = “ << setw(6) << B << endl;
cout << “Nilai C = “ << setw(6) << C << endl;

Bandingkan bila tanpa menggunakan setw().

setfill()
Perintah setfill digunakan untuk menambahkan suatu karakter tertentu pada field yang kosong pada suatu data. Perhatikan contoh berikut ini

A = 123;
B = 98;
cout<< “Nilai A = “<< setw(6) << setfill(‘\*’) << A << endl;
cout<< “Nilai B = “<< setw(6) << setfill(‘.’) << B << endl;

Perintah setfill() biasanya digunakan setelah penggunaan setw().