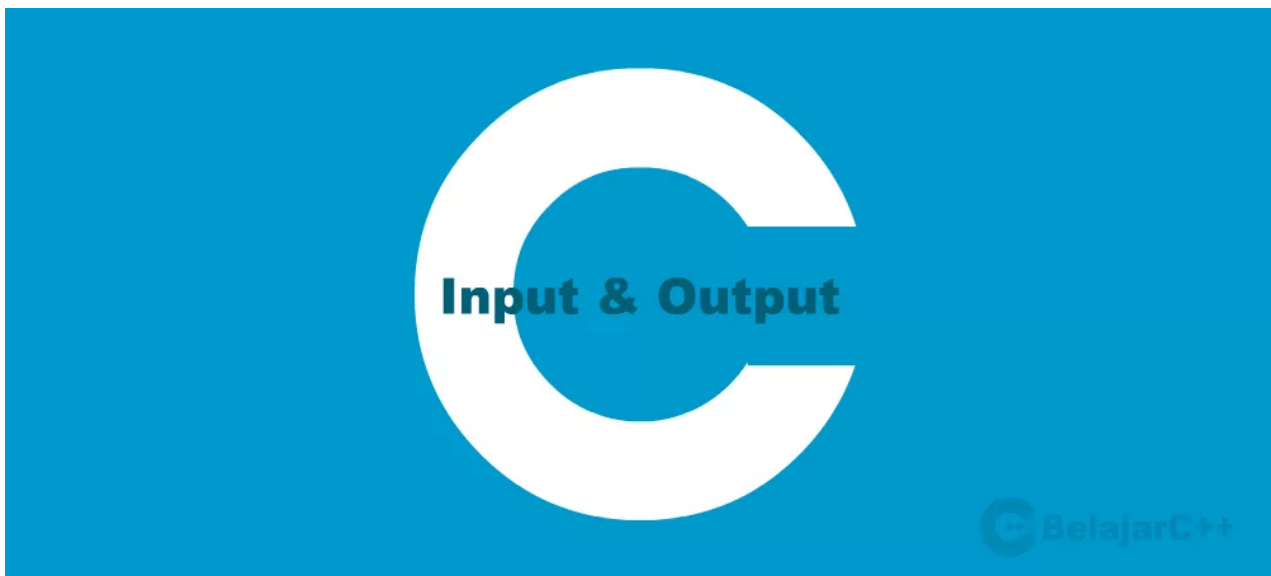


[Home](#) / [Tutorial](#) / [C](#) / Dasar Input Output pada C

Dasar Input Output pada C

Posted on Januari 22, 2018 | Last Modified Januari 22, 2018



Dasar Perintah keluaran (Output) C

Perintah Keluaran (Output) adalah aktivitas dari program ke pengguna, sebuah perintah dari dalam program untuk melakukan aktivitas pemberian informasi kepada pengguna (manusia). Informasi tersebut bisa berbentuk data seperti tulisan, gambar, file, hardcopy, objek dan lain-lain.

Di artikel ini penulis akan jelaskan mengenai dasar perintah keluaran berupa tulisan di dalam layar konsol. Di dalam C++ terdapat 2 cara melakukan output dalam layar konsol yaitu dengan cara C dan C++. dalam bahasa pemrograman C++ kita dapat melakukan perintah keluaran menggunakan keyword cout.

perintah keluaran C seperti printf, puts, putchar dan lain-lain juga tersedia dan dapat digunakan dalam bahasa pemrograman C++.

Macam-Macam Dasar Perintah Keluaran (Output) C

printf()

printf (Print Formatted data to stdout) adalah fungsi yang berasal dari bahasa pemrograman C, fungsi yang paling umum digunakan pada bahasa pemrograman C dalam melakukan perintah keluaran pada layar konsol. printf adalah fungsi yang berasal dari pustaka (file header) <stdio.h>.

Bentuk Penulisan

```
printf("kalimat karakter atau kumpulan karakter  
%Penentu_format_variabel", variabel);
```

Keterangan

di dalam printf tepat di dalam tanda "", anda dapat mengisinya menggunakan karakter, string atau numerik untuk ditampilkan pada layar konsol. Untuk menyertakan variabel dalam penulisan tersebut, kita dapat menyertakan variabel dengan menggunakan penentu format, penentu format digunakan untuk menentukan jenis format apa yang akan digunakan untuk mencetak variabel yang di panggil dan setelah tanda "" tepat setelah koma merupakan daftar urutan identitas variabel, peletakan variabel urut berdasarkan urutan format pada kalimat di dalam tanda "".

Macam – macam Dasar penentu format

Tipe Data	Penentu Format
Integer	%d atau %i
Floating Point :	
– Bentuk Desimal	%f
– Bentuk Berpangkat	%e
– Bentuk Desimal dan Pangkat	%g
Double	%lf
Character	%c
String	%s
Unsigned Integer	%u
Long Integer	%ld

Tipe Data	Penentu Format
Long Unsigned Integer	%lu
Unsigned Hexadecimal Integer	%x
Unsigned Octal Integer	%o

Contoh Penulisan

```
printf("%s mendapatkan nilai %f", nama, nilai);  
printf("%s mendapatkan peringkat %c", nama, grade);
```

untuk tata letak penentu format dan alamat variable, daftar variable harus diurutkan berdasarkan penentu format yang telah disebutkan dalam tanda ("...").

Contoh program

```
#include <stdio.h>  
  
int main()  
{  
    char nama[] = "BelajarCPP", grade = 'A';  
    float nilai = 98;  
  
    printf("%s mendapatkan nilai %f \n", nama, nilai);  
    printf("%s mendapatkan peringkat %c", nama, grade);  
  
    return 0;  
}
```

puts()

puts() adalah fungsi yang mempunyai kepanjangan Put String merupakan standar output sama seperti printf(), fungsi puts() berfungsi untuk menyalin alamat dari variabel string atau string (kalimat) yang disebutkan sebagai argumen untuk dicetak ke layar dalam bentuk string. Puts() Termasuk dalam pustaka <stdio.h>. yang merupakan turunan dari bahasa pemrograman C.

Perbedaan fungsi puts() dan printf()

printf()

Harus menentukan format data untuk tipe data sting, yaitu %s

Untuk mencetak pindah baris memerlukan notasi “\n”

puts()

Tidak perlu menentukan tipe data string karena ini khusus untuk data string

Untuk mencetak pindah bais tidak perlu notasi “\n”, karena sudah diberikan secara otomatis.

Contoh Penulisan

```
puts("Belajar C++");  
puts(nama);
```

Contoh Program

```
#include <stdio.h>  
  
int main()  
{  
    char nama[] = "Belajar C++";  
  
    puts("Anda Sedang belajar di ");  
    puts(nama);  
    return 0;  
}
```

putchar()

fungsi putchar() digunakan untuk menampilkan sebuah karakter ke layar. Penampilan karakter tidak diakhiri dengan pindah baris. Putchar berasal dari pustaka <stdio.h>.

Contoh Penulisan

```
putchar('a');  
putchar(var);
```

fungsi putchar hanya bisa mendapatkan 1 argument dan hanya bisa mencetak satu karakter.

Contoh Program

```
#include <stdio.h>

int main()
{
    char var='a';

    putchar('B');
    putchar('E');
    putchar('L');
    putchar(var);
    putchar('J');
    putchar('A');
    putchar('R');
    putchar('C');
    putchar('+');
    putchar('+');
    return 0;
}
```

Dasar Perintah Masukan (Input) C

Input adalah aktifitas pengguna (manusia) kepada program, yang memungkinkan program untuk menerima data dari pengguna yang biasanya berbentuk text, file, gambar, hardcopy, object dan lain-lain. Data tersebut biasanya akan diolah oleh program dan dikeluarkan dengan perintah keluaran ditujukan untuk pengguna itu sendiri.

Di artikel sebelumnya, pada bahasa pemrograman C++ kita dapat melakukan aktifitas input menggunakan keyword cin. Dan pada artikel ini penulis akan membahas mengenai perintah input pada bahasa pemrograman C.

ada beberapa standartr fungsi yang dapat kita gunakan untuk melakukan aktifitas input informasi dari manusia ke program dalam bahasa pemrograman C, yaitu dengan menggunakan scanf(), gets(), dan getchar() yang merupakan standar fungsi yang biasa digunakan dalam aktifitas input pada layar konsol.

Macam-Macam Dasar Perintah Masukan (Input) C

scanf()

Scanf (Scan/Read Formatted data from stdin) adalah fungsi yang berasal dari bahasa pemrograman C. bisa dikatakan scanf memiliki fungsi berlawanan dengan printf, yaitu scanf berfungsi untuk melakukan perintah masukan melalui layar konsol. Memungkinkan pengguna (manusia) memberikan informasi berupa data text kepada program. Scanf ada fungsi yang berasal dari pustaka <stdio.h>

Bentuk Penulisan

```
scanf("%penentu_format", alamat_variabel);
```

Keterangan

scanf membutuhkan dua argument, yaitu :

- Argument pertama "%penentu_format". Tidak jauh beda dengan printf, scanf membutuhkan penentu format sama seperti penentu format pada printf (untuk dasar penentu format anda bisa melihat pada keterangan printf di atas).
- Argument kedua "alamat_variabel" digunakan untuk memberikan data berupa alamat memori dari variabel untuk penyimpanan data input. Untuk variabel biasa, biasanya kita membutuhkan tanda Address -of Operator "&" untuk menyatakan alamat memori pada variabel tersebut.

Contoh Program

```
#include <stdio.h>

int main( )
{
    char nama[15] = "";

    int kelas = 0;
    float nilai = 0;

    printf("Nama : ");scanf ("%s",nama);
    //Variabel "nama" Tidak menggunakan tanda & karena memanggil
    array tanpa index/penentu elemen
    //akan menghasilkan alamat memori dari elemen pertama array
    printf("Kelas: ");scanf ("%d",&kelas);
    printf("Nilai: ");scanf ("%f",&nilai);

    printf("\nNama : %s \nKelas : %d \nNilai : %f", nama, kelas,
    nilai);
    return 0;
}
```

gets()

Gets (Get String) adalah fungsi yang berasal dari bahasa pemrograman C, bisa dikatakan bahwa gets() merupakan pasangan dari puts() yang memiliki fungsi berlawanan yaitu berfungsi untuk membuat perintah masukan, memungkinkan pengguna (manusia) dan dapat memasukan data atau informasi ke dalam program komputer melalui layar konsol dalam bentuk data text. Mirip seperti scanf tapi gets lebih dikhususkan untuk string. Gets termasuk dalam pustaka <stdio.h>.

Bentuk Penulisan

```
gets(nama_variabel)
```

Perbedaan antara scanf() dan gets()

scanf()	Tidak dapat menerima string yang mengandung spasi atau tab dan dianggap data terpisah
gets()	Dapat menerima string yang mengandung spasi atau tab dan masih dianggap sebagai satu kesatuan data.

Contoh Program

```
#include <stdio.h>

int main()
{
    char nama[20];

    puts("Masukan nama anda :");
    gets(nama);

    puts("\nSenang Berkenalan Dengan Anda, Saudara");
    puts(nama);
    return 0;
}
```

getchar()

Getchar (Get Character from stdin), adalah fungsi yang berasal dari bahasa pemrograman C. merupakan pasangan dari Putchar, dan memiliki fungsi yang berlawanan yaitu, Getchar berfungsi untuk memberikan perintah masukan, memungkinkan pengguna untuk memasukan data informasi berupa satu karakter text melalui layar konsol kepada program komputer. Getchar dikhususkan untuk masukan berupa satu karakter. Getchar termasuk dalam pustaka <stdio.h>.

Bentuk Penulisan

```
variabel = getchar();
```


berbeda seperti fungsi-fungsi sebelumnya, getchar tidak menggunakan argument untuk parameter getchar, tempat dimana data akan disimpan (variabel) ditulis sebelum fungsi getchar dan menggunakan tanda "=" diantara nama variabel dan fungsi getchar().

Contoh Program


```
#include <stdio.h>
int main( )
{
    char var;


    printf("Masukan Sebuah Karakter = ");
    var = getchar( );

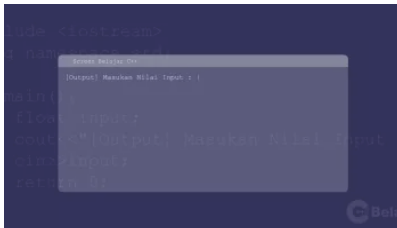
    printf("\nAnda Memasukan karakter \"%c\"", var);
    return 0;
}
```

 Share

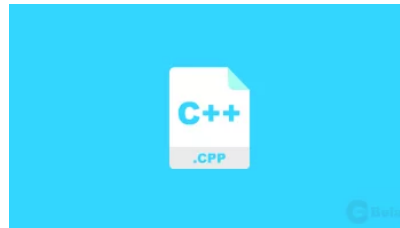
 Tweet

 Share

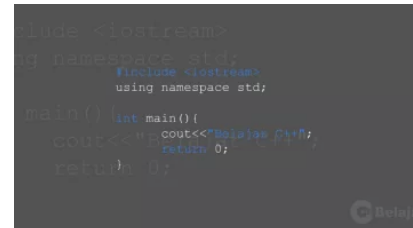
 Pin

 Share
Terkait**Pengertian dan Dasar Input Output C++**

Januari 22, 2018
dalam "C++"

**Pengenalan Bahasa Pemrograman C++**

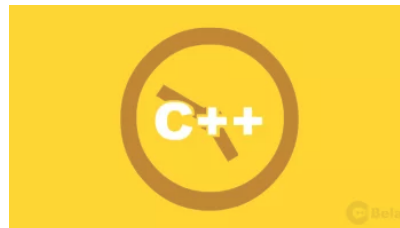
Januari 22, 2018
dalam "C++"

**Struktur dan Bagian-Bagian Dasar C++**

Januari 22, 2018
dalam "C++"

**Pengertian Dan Cara Kerja Komputer**

Januari 21, 2018
dalam "Tutorial"

**Sejarah Bahasa Pemrograman C++**

Januari 22, 2018
dalam "C++"

**Sejarah Bahasa Pemrograman**

Januari 22, 2018
dalam "C++"

