

# Pemrograman dengan Bahasa C

## Bagian 1

**Yusuf**

**[www.IlmuSoft.com](http://www.IlmuSoft.com)**

4 April 2013

## 1. Program Pertama

Ketikan program berikut dan lakukan kompilasi (gcc <nama\_file> -o <nama\_file>.exe).

```
#include <stdio.h>
int main()
{
    printf("Hello World !");
    return 0;
}
```

Lalu jalankan untuk melihat hasilnya. Jika output dari program adalah tulisan "Hello World !" berarti anda telah sukses membuat program pertama anda. Selamat !

## Penjelasan dari Program Pertama

Sekarang saya akan coba jelaskan maksud dari program di atas.

- **#include <stdio.h>** : ini disebut preprocessor directive yang menyertakan file header stdio untuk keperluan program.
- **int main()** : menandai awal program
- **{** : menandai awal dari tubuh program
- **printf("Hello World !");** : ini merupakan fungsi untuk mencetak karakter
- **return 0;** : artinya program berakhir dengan sukses
- **}** : menandai akhir dari tubuh program

## 2. Struktur Bahasa C

Setiap program yang dibuat dengan bahasa c minimal terdapat satu fungsi main. Di fungsi main inilah program pertama dieksekusi. Program dieksekusi secara urut dari atas ke bawah. Setiap statement dalam bahasa c harus diakhiri dengan titik koma (;). Coba kompilasi ulang program pertama namun sebelumnya hapus dulu tanda titik komanya. Apa yang terjadi ? error.

Bahasa c juga case-sensitive. Statement printf tidak sama dengan Printf.

## 3. Identifier

Identifier adalah sebuah pengenalan. Pengenal dalam bahasa c harus diawali dengan a-z, A-Z atau underscores (\_). Lalu boleh diikuti dengan angka 0-9. Identifier juga tidak boleh mengandung spasi dan simbol apapun.

### Konstanta

Konstanta berasal dari kata constant yang artinya tetap. Cara pendeklarasian konstanta dalam bahasa c adalah sebagai berikut :

```
#define <nama_konstanta> <nilai>
```

Contoh :

```
#define phi 3.14
#define nama "Yusuf"
```

Baris pertama mendeklarasikan konstanta bernama phi dengan nilai 3.14 yang bertipe float sedangkan baris kedua adalah pendeklarasian tipe string. Pada string kita harus mengapit nilai dengan kutip (").

## Variabel

Variabel juga seperti konstanta berguna untuk menyimpan nilai namun nilai variabel bisa berubah-ubah. Cara pendeklarasian variabel adalah sebagai berikut :

```
<tipe_data> <nama_variabel>;
```

Contoh :

```
int jumlah;
float tinggi_badan;
```

Baris pertama mendeklarasikan variabel bertipe integer (bilangan bulat) sedangkan baris kedua bertipe float (bilangan pecahan).

## 4. Tipe Data

Tipe data pada bahasa c terdiri dari tipe data numerik dan non-numerik.

Tipe data numerik : int, short integer dll.

Tipe data non-numerik : char dll.

Contoh pendeklarasian dan pemberian nilai :

```
int a;
a = 10;
char b;
b = 'Z';
```

Meskipun variabel b berisi 'Z' nilai yang sebenarnya disimpan oleh variabel b adalah kode ASCII dari 'Z' yaitu 90 (desimal).

## 5. Operator dan Karakter Escape

Bahasa c mendukung banyak sekali operator untuk keperluan pemrograman. Berikut saya berikan beberapa operator yang tersedia.

Kategori	Operator
Arithmetic	+ - * / %
Logical	&   ^ ! && true false
Relational	== != < > <= >=
Assignment	= += -= *= /= %= &=
Increment Decrement	++ --

Contoh :

```
int a = 4;
int b = 5;
int c;
a++;
c = a * b;
```

hasil akhir dari variabel c adalah 100.

a++ artinya tambahkan nilai a dengan 1. Sehingga baris terakhir artinya  $c = 5 \times 5 = 100$ .

## Karakter Escape

Bahasa c memiliki beberapa karakter khusus yang disebut karakter escape. Karakter ini diawali dengan tanda backslash (\). Misalnya karakter escape \t digunakan untuk tab, karakter \n untuk baris baru, \a untuk membunyikan speaker komputer.

Contoh :

```
#include <stdio.h>
int main()
{
    printf("\tTulisan ini masuk ke dalam\n");
    printf("bunyikan speaker\a");
    return 0;
}
```

## 6. Struktur I/O (Input Output)

Untuk melakukan operasi I/O kita bisa menggunakan fungsi standar bahasa c. Struktur io terdiri dari struktur tidak terformat dan terformat.

### Struktur Input Tak Terformat

#### Input Data Karakter Tak Terformat

Kita bisa menggunakan fungsi berikut untuk mendapatkan input dari keyboard.

getchar() : karakter terlihat dengan enter  
getche() : karakter terlihat tanpa enter  
getch() : karakter tak terlihat tanpa enter

Contoh :

```
#include <stdio.h>
int main() {
    char huruf;
    printf("Ketikkan sebuah karakter : ");
    huruf = getche();
    printf("\nKarakter Anda : %c", huruf);
    return 0;
}
```

## Struktur Input Terformat

### Input data karakter terformat

Pada input terformat kita gunakan scanf() dengan kode format sebagai berikut :

Kode Format	Fungsi
%c	Membaca sebuah karakter
%s	Membaca string
%d	Membaca nilai integer (desimal)
%x	Membaca nilai hexadecimal
%f	Membaca nilai float (pecahan)
%o	Membaca nilai octal

Contoh :

```
#include <stdio.h>
int main() {
    char huruf;
    int angka;
    printf("Ketikkan sebuah huruf : ");
    scanf("%c", &huruf);
    printf("Ketikkan sebuah angka : ");
    scanf("%d", &angka);
    printf("Yang anda masukkan : %c dan %d", huruf, angka);
    return 0;
}
```

Hasil :

Ketikkan sebuah huruf : a (enter)

Ketikkan sebuah angka : 1 (enter)

Yang anda masukkan : a dan 1

## Struktur Output Tak Terformat

Fungsi yang digunakan adalah putchar(char) dan puts(char[]).

Contoh :

```
#include <stdio.h>
int main() {
    char a = 'Y';
    char b[6] = "usuf";
    putchar(a);
    puts(b);
    return 0;
}
```

Hasil :

Yusuf

## Struktur Output Terformat

Selama ini kita sudah sering menggunakan fungsi output terformat yaitu printf(). Kode format yang digunakan sama seperti sebelumnya (lihat kode format scanf()).

Contoh :

```
#include <stdio.h>
int main() {
    char huruf = 'C';
    int jam = 7;
    printf("Saya belajar bahasa %c jam %d malam", huruf, jam);
    return 0;
}
```

Hasil :

Saya belajar bahasa C jam 7 malam