

Welcome to Overview TURBO C

Pengenalan C

- Unsur-unsur bahasa C
- Deklarasi variabel & Type Data
- Statement dalam C
- Bentuk umum dari program C
- Tanda / Operator Aritmatik
- Menampilkan Hasil / Output Terformat
- Batch Mode, Interactive Mode, dan File-file Data
- Kesalahan-kesalahan Umum dalam Program C

Sejarah Singkat C

Akar dari bahasa C adalah dari bahasa BCPL yang dikembangkan oleh Martin Richard pada tahun 1967. Bahasa ini memberikan ide kepada Ken Thompson yang kemudian mengembangkan bahasa yang disebut dengan B pada tahun 1970. Perkembangan selanjutnya dari bahasa B adalah bahasa C oleh Dennis Richie sekitar 1970-an di Bell Telephone Laboratories Inc. Bahasa C pertama kali digunakan di komputer Digital Equipment Corporation PDP-11 yang menggunakan sistem operasi UNIX.

C adalah bahasa yang standar, artinya suatu program yang ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi.

Unsur-unsur Bahasa C

Struktur dari program C merupakan kumpulan dari sebuah atau lebih fungsi-fungsi. Fungsi pertama yang harus ada di program C sudah ditentukan namanya yaitu bernama **main ()**. Suatu fungsi di program C dibuka dengan kurung kurawal buka ({) dan ditutup dengan kurung kurawal tutup (}). Di antara kurung kurawal dapat dituliskan statemen-statemen program C.



Struktur Program C

Preprocessor directive

```

main()
{
    statemen-statemen;           Fungsi utama
}

Fungsi_fungsi_lain()
{
    statemen-statemen;           Fungsi fungsi lain yang ditulis pemrogram komputer
}

```

Contoh program C yang sederhana

Berikut ini adalah suatu program C sederhana yang strukturnya mengikuti struktur di atas :

```

/* Program C yang sederhana*/
#include<stdio.h>
main()
{
    float Celcius, Fahrenheit;
    printf("Masukkan Nilai Celcius ?");
    scanf("%f", &Celcius);
    Fahrenheit=Celcius*1.8+32;           /*menghitung konversi*/
    printf("%f celcius adalah %f fahrenheit \n", Celcius, Fahrenheit);
}

```

Jika program ini dijalankan akan didapatkan hasil :

```

Masukkan nilai celcius? 10
    10.000000 celcius adalah 50.000000 fahrenheit

```

Deklarasi Variabel

Variabel adalah suatu pengenalan yang digunakan untuk menyimpan suatu nilai dan nilai dari variabel dapat berubah-ubah selama proses dari program.

Deklarasi Variabel dapat dilakukan dengan menggunakan kata-kata kunci seperti

int, float, double, char, long, register, short, signed dan **unsigned**.



Berikut ini adalah beberapa contoh pendeklarasian variabel-variabel :

```
int x, y      /*mendeklarasikan variabel x dan y tipe integer*/
float a,b    /*mendeklarasikan variabel a dan b tipe floating point*/
char huruf  /*mendeklarasikan variabel huruf tipe karakter*/
double x     /*mendeklarasikan variabel x tipe pecahan ketepatan ganda*/
long int x   /*mendeklarasikan variabel x tipe numerik panjang*/
register x   /*mendeklarasikan variabel x yang nilainya disimpan di register*/
```

Tipe Data

Tipe data dalam program C menyediakan lima macam tipe data dasar, yaitu tipe data integer (nilai numerik bulat yang dideklarasikan int), floating point (nilai numerik pecahan ketepatan tunggal yang dideklarasikan dengan float), double-precision (nilai numerik pecahan ketepatan ganda yang dideklarasikan dengan double), karakter (dideklarasikan dengan char), dan kosong (dideklarasikan dengan void).

Tabel Tipe-tipe Data Dasar

Tipe	Lebar	Jangkauan Nilai	
		Dari	Sampai dengan
int	16 bit	-32768	32767
signed int			
short int			
signed short int			
unsigned int	16 bit	0	65535
unsigned short int			
long int	32 bit	-2147483648	2147483649
signed long int			
unsigned long int	32 bit	0	4294967296
float	32 bit	3.4E-38	3.4E+38
double	64 bit	1.7E-308	1.7E+308
long double	80 bit	3.4E-4932	1.1E+4932
char	8 bit	-128	127
signed char			
unsigned char	8 bit	0	255



Statemen Dalam C

Suatu statemen adalah pernyataan yang menyebabkan suatu tindakan akan dilakukan oleh komputer. Tindakan tersebut misalnya adalah tindakan untuk menghitung, menampilkan hasil, menerima input dan lain-lain. Suatu statemen di bahasa C ditulis dengan diakhiri oleh titik koma (;). Statemen dapat berupa statemen kosong, statemen ungkapan, statemen kendali dan statemen jamak.

Macam-macam Statemen

◆ Statemen Kosong

Disebut juga empty statement atau null statement, adalah statemen yang hanya terdiri dari pengakhir titik koma saja. Artinya, tidak ada tindakan yang akan dilakukan.

◆ Statemen Ungkapan / Expression Statement

Merupakan statemen yang dibentuk dari suatu ungkapan yang diakhiri dengan tanda titik koma (;).

Contoh :

```
fahrenheit = celcius*1.8+32;
++x;
dan sebagainya
```

✦ Statemen Kendali / Control Statement

Merupakan statemen yang fungsinya mengendalikan proses dari program. Statemen ini dibentuk dengan diawali kata-kata kunci **if, switch, for, do-while, goto, break,** dan **continue**.

Contoh :

```
if(N<0) printf("Nilai N tidak boleh negatif!!\n");
```

✦ Statemen Jamak / Compound Statement

Adalah statemen yang terdiri dari gabungan beberapa statemen tunggal yang ditulis berada di antara tanda kurung kurawal ("{" dan "}")



```
Contoh :  
  
if(D==0)  
{  
    x1=-B/(2*A);  
    printf("Dua Akar Real Kemabar:\n");  
    printf("X1=X2=%f \n",X1);  
else if (D>0)  
{  
    x1=(-B+sqrt(D))/(2*A);  
    x2=(-B-sqrt(D))/(2*A);  
    printf("Dua Akar real berlainan:\n");  
    printf("X1 = %f \n",X1);  
    printf("X2 = %f \n",X2);  
}
```

Operator Aritmatika

Bahasa C menyediakan sejumlah operator aritmatika seperti pada tabel di samping.

Operator	Maksud
*	perkalian
/	pembagian
%	pambagian modulus (sisanya)
+	pertambahan
-	pengurangan

Operator * dan /

Operator * digunakan untuk perkalian dan / untuk pembagian.

```
Contoh :  
#include<stdio.h>  
main()  
{  
    int X,Y;  
    float Z;  
  
    X = 7;  
    Y = 3;  
  
    Z=X/Y;  
    Printf("Nilai Z=%f",Z);  
}
```



Operator Modulus (%)

Operator ini hanya digunakan untuk nilai-nilai integer saja, tidak dapat digunakan untuk operasi nilai pecahan (float, double, atau long double). Operator ini akan menghasilkan sisa dari pembagian dari dua buah nilai integer.

Contoh :

```
#include<stdio.h>
Main()
{
    int X,Y,Z;
    X = 7;
    Y = 3;
    Z=X%Y;
    printf("Nilai Z=%d\n",Z);
}
```

Menampilkan Output Terformat

Jika fungsi `outchar()` dan `puts()` menampilkan hasil tanpa dapat diatur bentuknya, karena tidak menggunakan kode format, maka fungsi `printf()` dapat digunakan untuk memformat bentuk dari tampilannya. Fungsi `printf()` dapat digunakan untuk menampilkan hasil yang diatur formatnya, karena fungsi ini menggunakan kode-kode format.

Tabel kode-kode format untuk fungsi `printf()`

Kode Format	Kegunaan
<code>%c</code>	Menampilkan sebuah karakter
<code>%s</code>	Menampilkan nilai string
<code>%d</code>	Menampilkan nilai desimal integer
<code>%i</code>	Menampilkan nilai desimal integer
<code>%u</code>	Menampilkan nilai desimal integer tak bertanda
<code>%x</code>	Menampilkan nilai heksadesimal integer
<code>%o</code>	Menampilkan nilai oktal integer
<code>%f</code>	Menampilkan nilai pecahan
<code>%e</code>	Menampilkan nilai pecahan dalam notasi saintifik
<code>%g</code>	Sebagai pengganti <code>%f</code> atau <code>%e</code> tergantung yang terpendek
<code>%p</code>	Menampilkan suatu alamat memori untuk pointer



Kode-kode format tersebut semuanya ditulis dengan huruf kecil dan tidak benar jika ditulis dengan huruf besar. Misalnya : kode format "%C" merupakan bentuk yang salah.

Batch Mode, Interactive Mode dan Data-data file

- Interactive mode - pemakai saling berhubungan dengan program dan tipe data ketika program sedang dijalankan.
- Batch mode - program meneliti data nya dari suatu file data yang telah disiapkan terlebih dahulu sebagai ganti saling berinteraksi dengan pemakai nya .

Contoh Batch Mode dalam Mengkonversi dari Mil ke Kilometer

```
#include <stdio.h>
#define KMS_PER_MILE 1.609

int
main(void)
{
    double miles,kms;
    scanf("%lf",&miles);
    printf("Jarak dalam Mil adalah : %.2f.\n",miles);
    kms = KMS_PER_MILE*miles;
    printf("Sama dengan %.2f Kilometer.\n", kms);

    return (0);
}
```

Input Redirection

- Contoh di atas menunjukkan penulisan kembali program konversi dari mil ke kilometer dalam bentuk batch.
- Kita mengasumsikan di sini bahwa input standard dihubungkan dengan suatu data batch sebagai ganti keyboard.
- Dalam kebanyakan sistem, asosiasi ini dapat terpenuhi secara relatif dengan mudah melalui input/output redirection yang menggunakan perintah sistem operasi.
- Sebagai contoh, dalam sistem operasi UNIX dan MS.DOS, kita dapat menginstruksikan program kita untuk mengambil file dari mydata dengan menempatkan simbol <mydata di akhir baris perintah> yang menyebabkan compile kita dan program yang terhubung untuk dieksekusi. Jika baris perintah digunakan secara normal metric untuk menjalankan program ini, baris perintah yang baru akan menjadi :
metric <mydata



Program Style *Echo Prints vs. Prompts*

- Dalam contoh program sebelumnya, pernyataan `scanf("%1f",miles);` didapat/kan suatu nilai mil dari baris pertama dalam file.
- Karena input program datang dari suatu file data, tidak usah mendahului statemen ini dengan suatu prompt pesan. Sebagai ganti, kita mengikuti panggilan ke **scanf** dengan statemen

```
printf("Jarak dalam Mil adalah : %.2f.\n",miles);
```

- Statemen `echo prints` atau tampilan nilai hanya tersimpan dalam mil dan menyediakan suatu rekaman data yang digerakkan oleh program.
- Tanpa itu, tidak akan mudah mengetahui nilai `scanf` yang diperoleh dari mil.
- Kapan pun kita mengkonversi suatu program interaktif ke suatu program batch, pastikan kita mengganti masing-masing prompt dengan suatu **echo prints** setelah `scanf`.

Output Redirection

- Kita dapat juga mengalihkan output dari program ke suatu file sebagai ganti layar. Kemudian kita dapat mengirimkan file output ke printer untuk memperoleh suatu salinan output program. Dalam UNIX atau MS DOS, gunakan simbol `>myoutput` untuk mengarahkan kembali output dari layar ke file `myoutput`. Dijalankan dengan baris perintah sebagai berikut:

```
metric>myoutput
```

yang akan menjalankan `compu` dan kode yang terhubung dengan program `metric`, mengambil input program dari keyboard dan menulis program output ke file `myoutput`. Akan tetapi, interaksi dengan program yang berjalan akan sulit karena semua program output termasuk semua pesan prompt, akan dikirim ke file output. Akan lebih baik menggunakan baris perintah :

```
metric <mydata >myoutput
```

yang mengambil input program dari file `mydata` dan mengirim program output ke output file ke file `myoutput`.



Program Konversi Mil ke Km dengan File Bernama

```
#include <stdio.h>
#define KMS_PER_MILE 1.609

int
main(void)
{
    double miles,kms;
    FILE *inp,*outp;

    inp=fopen("b:distance.dat","r");
    outp=fopen("b:distance.out","w");
    fscanf(inp,"%lf",&miles);
    printf(outp,"Jarak dalam Mil adalah : %.2f.\n",miles);
        kms = KMS_PER_MILE*miles;
        printf(outp,"Sama dengan %.2f Kilometer.\n", kms);

    fclose(inp);
    fclose(outp);
    return (0);
}
```

Kontrol Program File Input dan Output

Sebagai alternatif untuk *input output redirection*, C memungkinkan program untuk menamakan secara eksplisit sebuah file dari file yang akan diambil inputnya dan file yang akan dikirim output. Contoh di atas menampilkan sebuah versi konversi jarak yang mengambil input dari data yang bernama `b:distance.dat` dan mengirim hasilnya ke dalam file yang bernama `b:distance.out`

Sebuah program yang memanipulasi spesifik file harus terlebih dahulu mendeklarasikan variabel *file pointer* yang digunakan untuk menyimpan informasi penting untuk memungkinkan akses ke file. Dalam contoh sebelumnya, pernyataan :

```
FILE *inp, *outp;
```

mendeklarasikan bahwa variabel file pointer `inp` dan `outp` akan memegang informasi pengijinan akses ke input program dan file output. Sistem operasi harus mempersiapkan file untuk input atau output sebelum mengijinkan akses. Persiapan ini adalah tujuan dari fungsi *fopen* dalam pernyataan di bawah ini :



```
Inp=fopen("b:distance.dat","r")
outp=fopen("b:distance.out","w");
```

Tugas pertama pernyataan *opens* (mempersiapkan akses) file *b:distance.dat* sebagai sumber dari input program dan menyediakan nilai akses penting dalam variabel file pointer *inp*. "r" dalam perintah pertama *fopen* mengindikasikan bahwa kita ingin membaca atau men-scan data dari file yang terbuka karena perintah tugas kedua pernyataan meliputi "w", mengindikasikan keinginan kita untuk menulis ke dalam *b:distance.out*, *outp* adalah diinisialkan sebagai output file pointer.

Dua pernyataan selanjutnya mendemostrasikan penggunaan dari fungsi *fscanf* dan *fprintf* mengequivalenkan kedua fungsi tersebut dengan fungsi *scanf* dan *printf*.

```
fscanf(inp, "%1f",&miles);
Fprintf(outp, "Jarak dalam mil adalah : %.2fn", miles);
```

Kesalahan-kesalahan Umum dalam C

- **Kesalahan Pemula**
- **Kesalahan Penggunaan**
- **String Errors**
- **Kesalahan Input/Output**

