

B A H A S A C



LABORATORIUM SISTEM INFORMASI

2009

DAFTAR ISI

Halaman

PENGENALAN PEMROGRAMAN BAHASA C	1
Sekilas Bahasa C	1
Struktur Bahasa C	2
Variable, Statement dan Preprocessor	5
Tipe data Bahasa C	7
OPERATOR BAHASA C	9
Operator Aritmatika	9
Operator Unary	10
Operator Pengerjaan	11
Operator Logika	11
Operator Koma	11
Operator Bitwise	12
FORMAT UNTUK SETIAP TIPE DATA	13
FUNGSI	15
Data Tidak Terformat	15
Data Terformat	17
FUNGSI OUTPUT	18
Hasil Tidak Terformat	19
Hasil Terformat	19
PENGAMBILAN KEPUTUSAN DALAM BAHASA C	23
Statement IF	23
Statement IF... Else	23
Statement Switch Case	26

PERULANGAN DALAM BAHASA C	29
Statement For	29
Statement While	30
Statement Do While	31
STATEMENT CONTINUE DAN GO TO	31
ARRAY DAN DATA TERSTRUKTUR DALAM BAHASA C	33
FUNCTION DALAM BAHASA C	42
POINTER DALAM BAHASA C	54

KATA PENGANTAR

C adalah bahasa yang standar artinya suatu program yang ditulis dengan versi bahasa C tertentu akan dapat di kompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi. Bahasa C tersedia hampir di setiap jenis computer, Kode bahasa C sifatnya adalah portable dan didukung dengan pustaka yang banyak.

Melalui praktikum C, diharapkan mahasiswa dapat merasakan kemudahan dan kelebihan dalam membuat suatu program.

Modul Bahasa C berisi tentang pengertian bahasa C, kemudian diawali mulai dari awal penulisan program, fungsi – fungsi input dan output bahasa C hingga dihasilkan output dari program tersebut.

PENGENALAN PROGRAM DALAM BAHASA C

1

Obyektif :

1. Mengetahui Sejarah Bahasa C
 2. Mengerti tentang Struktur Bahasa C
 3. Mengenai dan dapat menggunakan Variable
-

Sekilas Bahasa C

Akar dari bahasa C adalah BCPL yang dikembangkan oleh Martin Richards pada tahun 1967. Selanjutnya bahasa ini memberikan ide kepada Ken Thompson, yang kemudian mengembangkan bahasa yang disebut bahasa B pada tahun 1970. Perkembangan selanjutnya dari bahasa B adalah bahasa C oleh Dennis Ritchie sekitar tahun 1970-an di Bell

Telephone Laboratories Inc. (sekarang adalah AT&T Bell Laboratories).

Bahasa C pertama kali digunakan di komputer Digital Equipment Corporation PDP-11 yang menggunakan sistem operasi UNIX. C adalah bahasa standar, artinya suatu program ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi. Kepopuleran C membuat versi-versi yang banyak untuk komputer mikro. Untuk menstandarisasikannya maka dibentuk komite pada tahun 1983 yang menetapkan standar ANSI untuk bahasa C.

Hingga saat ini penggunaan bahasa C telah merata di seluruh dunia. Hampir semua perguruan tinggi di dunia menjadikan bahasa C sebagai salah satu mata kuliah wajib. Selain itu, banyak bahasa pemrograman populer seperti PHP dan Java menggunakan sintaks dasar yang mirip bahasa C.

Adapun Karakteristik Bahasa C yaitu:

a. Kelebihan

- Bahasa C tersedia hampir di semua jenis Komputer
- Kode bahasa C sifatnya adalah Portabel (dapat digunakan di semua jenis komputer).
- Bahasa C hanya menyediakan sedikit kata-kata kunci
- Proses executable program lebih cepat
- Dukungan pustaka yang banyak
- C bahasa pemrograman yang terstruktur dan merupakan bahasa tingkat tinggi, namun dapat pula dikatakan sebagai bahasa tingkat menengah karena mempunyai kemampuan seperti bahasa *low level* (tingkat rendah).

b. Kelemahan

- Bahasa C merupakan bahasa yang cukup sulit dimengerti terutama dalam hal sintaks-sintaksnya

Struktur Program Bahasa C

Bahasa C dikatakan sebagai bahasa pemrograman terstruktur, karena strukturnya menggunakan fungsi-fungsi sebagai program-program bagian (sub rutin). Fungsi utama yang pertama kali diproses adalah fungsi yang bernama “**main()**”, selain itu adalah merupakan fungsi-fungsi bagian yang dapat ditulis setelah fungsi “**main()**” atau diletakan di file pustaka (**library**).

Jika fungsi-fungsi yang tersedia pada file pustaka (library) dan akan dipakai di suatu program maka file judulnya (**Header**) nya harus disertakan/ditulis pada program dengan menggunakan preprocessor directive **#include**, contohnya **#include <stdio.h>** yang berarti akan mengambil fungsi-fungsi yang tersedia pada library standar IO, diantaranya **clrscr()**, **printf**, **scanf()**, dll.

Untuk komentar / penjelasan program harus diapit diantara (*/* ...*/*).

*Penulisan program bahasa C bersifat **Case Sensitive** artinya penamaan fungsi, variabel maupun keyword-nya membedakan huruf kecil dan besar.*

Adapun struktur bahasa C sebagai berikut:

#include < file-file header>

/ directive #include digunakan jika mengambil fungsi-fungsi library*/*

main() */* fungsi utama sbg awal dimulainya pemrosesan program */*

```
{           /*blok awal*/
    statemen-statemen;
}           /*blok akhir*/
```

Fungsi-fungsi lain()

```
{
    statemen-statemen;
}
```

Pemakaian #include pustaka (library) yang umum

- conio.h = Tampilan Layar : clrscr(), textcolor(), textbackground(), textattr(), dll.
- stdio.h = standart input/output : clrscr(), printf(), scanf(), puts(), gets(), getch(), cprintf(), dll.
- stdlib.h = standart library (akses file / stream) : fopen(), fclose(), fread(), fwrite(), fprintf(), fscanf(), dll.

- `alloc.h/malloc.h` = pengalokasian memori (memori dinamis) : `malloc()`, `free()`, `calloc()`, dll.

Pemakaian Escape Sequence pada fungsi output (`printf()`, `gets()`, dll) Karakter Escape Sequence sering digunakan untuk menampilkan hasil output, seperti mengganti baris, membunyikan bel, dll).

Karakter Escape	Arti
<code>\a</code>	Bunyi bel
<code>\b</code>	Mundur satu spai
<code>\f</code>	Ganti halaman
<code>\n</code>	Ganti baris baru
<code>\r</code>	Ke kolom awal, baris yang sama
<code>\t</code>	Tabulasi horisontal
<code>\v</code>	Tabulasi vertikal
<code>\0</code>	Null
<code>\'</code>	Petik tunggal
<code>\"</code>	Kutip
<code>\\</code>	Garis miring terbaik

Contoh program sederhana:

```
#include <stdio.h>
main()
{
    clrscr();
    printf("Halo kawan-kawan....!\n"); /* \n : ganti baris baru */
    printf("Saya sedang belajar bahasa C...nih..susah looh...!");
}
```

Variable, Statement dan Preprocessor

Mendeklarasikan **variabel** dapat dituliskan di dalam maupun di luar fungsi. Jika deklarasi variabel ditulis pada luar fungsi, maka variabel tersebut dapat di akses oleh semua fungsi-fungsi yang lain (**Variabel Global**) sedangkan jika di dalam fungsi, maka variabel tersebut hanya dapat diakses oleh fungsi itu sendiri (**Variabel Lokal**). Penamaan variabel tidak boleh mengandung spasi, tidak sama dengan nama keyword, tidak dimulai dengan angka serta tidak mengandung simbol-simbol yang telah ditentukan oleh BAHASA C.

```
char A;
int X, Y, my_var; /* X bertipe integer, Y bertipe integer, my_var
bertipe integer*/
float bil_pecahan=9.56; /* memberikan nilai awal = 9.56 untuk
variabel bil_pecahan */
```

Statement adalah pernyataan yang menyebabkan suatu tindakan dilakukan oleh komputer. Statement dalam bahasa C diakhiri dengan tanda titik koma (;) . Jenis statement diantaranya :

- a. Empty Statement / Null Statement

Statemen kosong adalah statement yang hanya terdiri dari pengakhiran titik koma saja, sehingga tidak melakukan tindakan apapun. Digunakan untuk membuat perulangan kosong yang dimaksudkan untuk memberi jarak ke proses selanjutnya.

b. Expression Statement

Statemen ungkapan merupakan statemen yang dibentuk dari ungkapan yang diakhiri dengan titik koma.

c. Control Statement

Statemen kendali merupakan statemen yang berfungsi untuk mengendalikan proses dari program, dapat berupa proses seleksi kondisi, perulangan atau lompatan. Statemen ini dibentuk dengan menggunakan kata kunci if, switch, do-while, goto, break dan continue.

d. Coumpound Statement/Block Statement

Statemen jamak adalah statemen yang terdiri dari beberapa statemen tunggal yang ditulis diantara tanda kurung kurawal ({ })

Konstanta merupakan nilai yang tidak dapat diubah selama proses dari program. Untuk menentukan konstanta dapat dilakukan dengan menggunakan variabel atau macro.

Jika menggunakan perantara variabel digunakan keyword **const** :

```
const my_konstanta = 306; /* nilai pada variabel my_konstanta
adalah tetap 306 */
```

Jika secara langsung menggunakan macro atau directive **#define**:

```
#define my_konstanta = 306 /* MACRO tidak perlu diakhiri titik
koma (;) */
```

```
float nilai = my_konstanta; /* memberikan variabel nilai dengan
my_konstanta (nilai=306) */
```

Contoh :

```
#include <stdio.h>
#define PI=3.14
/* Perhatikan huruf kecil dan besarnya */
main()
{
    int  a = 2, B, niLAi;
    float bil_pecahan;

    B = 4;
    bil_pecahan = 2.0;
    niLAi = a * (2 + B);
    printf(" Hasil dari %d * (2 + %d) = %d \n",a,b,niLAi);
    printf("Perkalian dari %4.2f * %4.2f = %4.2f \n", bil_pecahan,
    PI, bil_pecahan*PI);

    /* Format specifier untuk %d dan %f akan dijelaskan pada
    sesion berikutnya */
}
```

Tipe Data Bahasa C

Bahasa C menyediakan 5 macam tipe data dasar, yaitu tipe data integer (numerik bulat dideklarasikan dengan int), floating point (numerik pecahan ketepatan tunggal dideklarasikan dengan float), double precision (numerik pecahan ketepatan ganda dideklarasikan dengan double), karakter (dideklarasikan dengan char) dan kosong (dideklarasikan dengan void). Untuk int, float, double dan char dapat dikombinasikan dengan pengubah (*modifier*) signed, unsigned, long, short, maka hasilnya menjadi seperti pada tabel berikut.

Tabel1. Tipe Data Bahasa C

TIPE	Lebar	Jangkauan Nilai
int	16 bit	- 32768 s/d 32767
signed int		
short int		
signed short int		
unsigned int	16 bit	0 s/d 65535
unsigned short int		
long int	32 bit	- 2147483648 s/d 2147483649
signed long int		
unsigned long int	32 bit	0 s/d 4294967296
float	32 bit	3.4E-38 s/d 3.4E+38
double	64 bit	1.7E-308 s/d 1.7E+308
long double	80 bit	3.4E-4932 s/d 3.4E+4932
char	8 bit	- 128 s/d 127
signed char		
unsigned char	8 bit	0 s/d 255