

BAB 1 KONSEP DASAR BAHASA C

1.1 SEJARAH DAN STANDAR C

Akar dari bahasa C adalah 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 Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc. (sekarang adalah AT&T Bell Laboratories).

Kepopuleran bahasa C membuat versi-versi dari bahasa ini banyak dibuat untuk komputer mikro. Untuk membuat versi-versi tersebut standar, ANSI (American National Standards Institute) kemudian menetapkan standar ANSI untuk bahasa C. Standar ANSI ini didasarkan dari standar UNIX yang diperluas. Standar ANSI menetapkan sebanyak 32 buah kata-kata kunci (keyword) standar. Ke 32 kata kunci ini adalah :

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

1.2 ALASAN-ALASAN MENGGUNAKAN BAHASA C

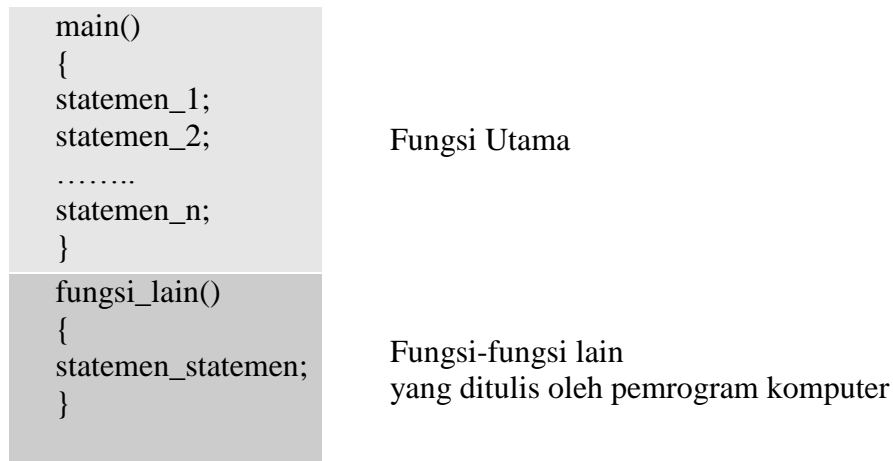
Beberapa alasan dapat dicatat mengapa bahasa C banyak digunakan, diantaranya adalah sebagai berikut :

1. Bahasa C tersedia hampir disemua jenis komputer
2. Kode bahasa C sifatnya adalah portabel
Aplikasi yang ditulis dengan bahasa C untuk suatu komputer tertentu dapat digunakan di komputer lain hanya dengan sedikit modifikasi.
3. Bahasa C hanya menyediakan sedikit kata-kata kunci
4. Proses executable program bahasa C lebih cepat
5. Dukungan pustaka yang banyak
Keandalan bahasa C dicapai dengan adanya fungsi-fungsi pustaka.
6. C adalah bahasa yang terstruktur
Bahasa C mempunyai struktur yang baik sehingga mudah untuk dipahami. C mempunyai fungsi-fungsi sebagai program bagiannya.
7. Selain bahasa tingkat tinggi, C juga dianggap sebagai bahasa tingkat menengah.
Bahasa C mampu menggabungkan kemampuan bahasa tingkat tinggi dengan bahasa tingkat rendah.
8. Bahasa C adalah kompiler
Karena C sifatnya adalah kompiler, maka akan menghasilkan executable program yang banyak dibutuhkan oleh program-program komersial.

1.3 STRUKTUR PROGRAM C

Untuk dapat memahami bagaimana suatu program ditulis, maka struktur dari program harus dimengerti terlebih dahulu, atau sebagai pedoman penulis program (programmer) bagaimana seharusnya program tersebut ditulis.

Struktur dari program C dapat dilihat sebagai kumpulan dari sebuah atau lebih fungsi-fungsi. Fungsi pertama yang harus ada di di program C yang sudah ditentukan namanya, yaitu fungsi main(). Artinya program C minimal memiliki satu fungsi (fungsi main()). Berikut ini adalah struktur dari program C.



Gambar 1.1 Struktur program C

Keterangan :

1. Dimulai dari tanda { hingga tanda } disebut tubuh fungsi / blok.
2. Tanda () digunakan untuk mengapit argumen fungsi, yaitu nilai yang dilewatkan ke fungsi. Pada fungsi main() tidak ada argumen yang diberikan, maka tidak ada entri di dalam ().
3. Kata void menyatakan bahwa fungsi ini tidak memiliki nilai balik.
4. Tanda { menyatakan awal eksekusi program dan tanda } menyatakan akhir eksekusi program.
5. Didalam tanda { } bisa tergantung sejumlah unit yang disebut pernyataan (statemen). Umumnya pernyataan berupa instruksi untuk :
 - memerintah komputer melakukan proses menampilkan string ke layar.
 - menghitung operasi matematika.
 - membaca data dari keyboard.
 - dll.

Berikut ini adalah suatu program C yang sederhana :

```
/*--- Program C Sederhana ---*/
#include <stdio.h>
main()
{ printf("Masukkan Nilai Celcius ? \n"); }
```

1.4 FILE JUDUL / HEADER FILE

File judul (header file) merupakan file yang berisi dengan prototype (judul, nama dan sintak) dari sekumpulan fungsi-fungsi pustaka tertentu. Jadi file ini hanya berisi dengan prototype dari fungsi-fungsi pustaka, sedangkan fungsi-fungsi pustakanya sendiri disimpan di file pustaka (library file dengan extention file .LIB). Misalnya prototype dari fungsi-fungsi pustaka printf() dan scanf() terdapat di file judul stdio.h, sehingga jika fungsi-fungsi ini akan digunakan di program, maka nama file judulnya harus dilibatkan

dengan menggunakan **preprocessor directive #include**. File judul stdio.h berisi prototype fungsi-fungsi pustaka untuk operasi input dan output standar. Ada dua cara untuk melibatkan file judul di suatu program C, yaitu sebagai berikut :

#include <stdio.h> atau #include "stdio.h"

1.5 NAMA PENGENAL

Nama pengenal (Identifier name) adalah nama-nama yang ditentukan sendiri oleh pembuat program. yang nantinya nama tersebut digunakan dalam pemrograman. Fungsinya untuk menyatakan : Variabel, Tipe Data, Konstanta, Fungsi, Label dan Obyek.

Aturan penamaan pengenal :

1. Karakter pertama harus huruf atau garis bawah.
2. Karakter berikutnya boleh huruf, bilangan, atau garis bawah.
3. Panjang maksimal 32 karakter.
4. Nama pengenal tidak boleh sama dengan kata kunci (reserved word)
5. Pengenal bersifat case sensitif.
Huruf kecil dan huruf besar (kapital) pada suatu pengenal tidak dianggap sama :
Contoh : NAMA, Nama, nama = menyatakan pengenal yang berbeda.

Contoh penamaan :

Benar	Salah dan alasannya
Nama	2semester (tidak boleh diawali angka)
NAMA	nama-barang (tanda – tidak diperbolehkan)
Nama_barang	#barang (simbol # tidak diperbolehkan)
Kuartal_2	Nama barang (tidak boleh mengandung spasi)

1.6 KOMENTAR PROGRAM

Komentar merupakan bagian terpenting dari program. Kehadirannya sangat membantu pemrograman ataupun orang lain dalam memahami program, karena berupa penjelasan-penjelasan mengenai program atau bagian-bagian program. Hal ini penjelasannya bisa berupa :

- tujuan / fungsi program
- saat program dibuat atau direvisi
- keterangan-keterangan lain tentang kegunaan sejumlah pernyataan dalam program.

Pada C suatu komentar diawali dengan tanda (/*) dan diakhiri dengan tanda (*/). Semua string atau tulisan yang terletak sesudah tanda /* hingga akhir baris dengan sendirinya akan diperlukan sebagai keterangan. Bagi kompiler hal ini tidak berguna dan akan diabaikan pada saat kompilasi.

Contoh :

```
/* ----- */
/* PROGRAM_1.C */
/* contoh program yang menggunakan komentar */
/* -----*/
```

1.7 STATEMENT

Suatu statemen (statement) adalah pernyataan yang menyebabkan suatu tindakan akan dilakukan oleh komputer. Tindakan tersebut dapat berupa tindakan untuk

menghitung, menampilkan hasil, menerima input data, mengendalikan proses program dan lain-lain. Suatu statemen dibahasa C di tulis dengan diakhiri oleh tanda titik koma (;).

Contoh :

```
X = X + 1;
printf(" Nilai X = %f\n",X);
```

1.8 TIPE DATA

Bahasa C memiliki lima tipe data utama, yaitu karakter (char), integer (int), Floating point (float), double floating(double) dan valualess/kosong (void). Untuk int, float, double dan char dapat dikombinasikan dengan modifier signed, unsigned, long, short, maka hasilnya sbb : Penanda signed tidak perlu dituliskan.

Tipe	Lebar	Jangkauan nilai
Int		
Signed int	16 bit	-32768 s/d 32767
Signed short int	16 bit	-32768 s/d 32767
Signed long int	32 bit	-2147483647 s/d 2147483647
Unsigned int	16 bit	0 s/d 65535
Unsigned short int	16 bit	0 s/d 65535
Unsigned long int	32 bit	0 s/d 4294967295
Float		
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		
Signed char	8 bit	-128 s/d 127
Unsigned char	8 bit	0 s/d 255

1.9 KONSTANTA

Adalah suatu nilai yang tidak pernah berubah selama proses program. Dibagi menjadi :

a. Numeric integer

Dapat berupa integer bertanda dan integer tidak bertanda. Untuk unsigned integer diberi huruf u atau U pada akhir nilai, untuk signed long diberi tanda l atau L dan unsigned long diberi tanda ul atau UL.

Contoh :

43 (signed int), -123 (signed int), -75000L (signed long int), 261062UL (unsigned long int).

b. Numerik pecahan

Merupakan nilai numeric yang dapat mempunyai pecahan dibelakang titik decimal. Bilangan yang diberikan dapat diikuti huruf F untuk float atau huruf L untuk long double.

Contoh :

1.5 (float) , 253.67(float), 12.34E-99 (double), 1000.1F (float), 1.2E+4930 (long double).

c. String dan character.

Merupakan nilai yang ditulis diantara tanda petik ganda untuk string dan tunggal untuk karakter.

Contoh :

“A” string A dan ‘A’ karakter A.

d. Karakter Escape

Karakter ini digunakan untuk menampilkan hasil tertentu dan diawali dengan tanda ‘\’. Berikut table karakter escape :

Karakter	Fungsi
\a	Bunyi bell atau alert
\b	Mundur 1 spasi (backspace)
\f	Form feed (ganti halaman)
\n	New line (ganti baris)
\r	Carriage return (ke kolom pertama, baris yg sama)
\t	Tabulasi horizontal
\v	Tabulasi vertical
\0	Nilai kosong (null)
\'	Tanda petik
\''	Petik ganda
\\	Backslash

e. Konstanta Bernama

Konstanta bernama adalah suatu pengenal yang berhubungan dengan nilai tetap (variabel konstan). C++ memungkinkan pendefinisian suatu konstanta bernama. Hal ini dilakukan dengan menggunakan kata kunci `const`.

Bentuk umum :

```
const tipe_data nama_konstanta = nilai;
```

Contoh :

```
const int bulan = 12;  
const float phi = 3.14285;
```

1.10 VARIABEL

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

a. Mendeklarasikan Variabel

Bentuk umum deklarasi

```
typedata namavariabel;
```

Contoh :

```
int jumlah;  
float rata_rata;  
char nama;
```

b. Pendeklarasian dan penugasan variabel

Nilai awal suatu variabel dapat langsung diberikan pada saat mendeklarasikannya.

Bentuk umum

```
typedata namavariabel = nilai;
```

Contoh :

```
int jumlah = 5;  
float rata_rata = 2.5;  
char karakter = 'A'  
char string[10];
```

1.11 Fungsi Define

#define digunakan untuk mendefinisikan suatu konstanta, makro ataupun nama.

1. Sebagai konstanta

```
#include <stdio.h>  
#define Alamat "jl.mawar no.1 "  
main () { printf(Alamat); }
```

out : jl.mawar no.1

2. sebagai Makro

```
#define kuadrat(x) (x)*(x)  
Maka statement Luas = 3.14 * kuadrat(3);  
Sama artinya Luas = 3.14 * 3 * 3 ;
```

3. Sebagai nama

```
#define pecahan float  
Maka deklarasi variable pecahan derajat;  
Sama dengan float pecahan;
```