

Referensi :

1. http://www.acm.uiuc.edu/webmonkeys/book/c_guide/index.html
- 2.

KONSEP DASAR BAHASA C

SEJARAH

Bahasa C dikembangkan di Bell Laboratories pada tahun 1972 oleh Dennis Ritchie. Prinsip dasar dan idenya didapat dari bahasa yang sudah ada sebelumnya yaitu bahasa B dan BCPL dan CPL. (*Combined Programming Language*) yang dikembangkan dengan tujuan untuk membentuk suatu bahasa yang mampu untuk pemrograman *high level, machine independent programming* dan masih memungkinkan para pemrogram untuk mengatur perilaku informasinya. Masalah utama dari CPL adalah ukurannya yang terlalu besar untuk digunakan pada beberapa aplikasi. Pada tahun 1967, BCPL (Basic CPL) diciptakan dan dikembangkan oleh Martin Richards, dan kemudian dikembangkan menjadi bahasa B oleh Ken Thompson ditahun 1970, untuk system UNIX pertama pada PDP-7 di Bell Laboratories. Akhirnya pada tahun 1972, bahasa B dikembangkan menjadi bahasa C. *Compiler C* yang pertama didesain dan diimplementasikan oleh Dennis Ritchie dan digunakan di computer Digital Equipment Corporation PDP-11 yang menggunakan system operasi UNIX Bahasa C pertama kali. C adalah "K & R" C, yang merupakan singkatan dari nama penemunya. Selama akhir tahun 1970, C berkembang melalui banyak akademi dan universitas, karena kedekatannya dengan Unix dan ketersediaan dari C compiler. Kepopuleran C menyebabkan berbagai organisasi menggunakan vers C sendiri dan mengakibatkan masalah kompatibilitas. Untuk mengatasi hal ini, pada tahun 1983, American National Standards Institute (ANSI) membentuk sebuah komite untuk menetapkan definisi standar bahasa C, yang kemudian dikenal dengan istilah ANSI Standard C. Penggunaan bahasa C saat ini berkembang dengan standar fungsi library yang banyak.

Alasan-alasan Menggunakan Bahasa C

1. Bahasa C tersedia hampir disemua jenis komputer.
2. Kode bahasa C bersifat *Portable*.
3. Bahasa C hanya menyediakan sedikit Reserved Word. Keandalan C dicapai dengan fungsi-fungsi pustaka.
4. Proses *executable* program dalam bahasa C lebih cepat.
5. Dukungan pustaka yang banyak.
6. C merupakan bahasa terstruktur.
7. Selain bahasa tingkat tinggi, C juga dianggap sebagai bahasa tingkat menengah.
8. C adalah bahasa compiler.

STRUKTUR PROGRAM C

Struktur program C dapat dilihat sebagai kumpulan satu atau lebih fungsi-fungsi. Fungsi yang harus ada adalah `main()`, karena fungsi ini merupakan titik awal dan titik akhir peng-eksekusi program. Diawali dengan tanda `{ }` dan diakhiri dengan `{ }`. Bentuk umumnya :

```
main()
{
    Block statement;
}
```

Bahasa C dikatakan sebagai bahasa pemrograman terstruktur, karena strukturnya menggunakan fungsi-fungsi sebagai program-program bagian (subroutine). Fungsi-fungsi ini dapat ditulis setelah fungsi utama atau diletakkan di file pustaka (library). Jika Fungsi diletakkan di file library dan akan dipakai pada suatu program maka nama *header file*- nya harus disertakan didalam program dengan *preprocessor directive* `#include`.

Contoh program sederhana :

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

HEADER FILE

Merupakan file yang berisi *prototype* (judul, nama, dan sintaks) dari sekumpulan fungsi-fungsi pustaka tertentu, sedangkan fungsi-fungsi pustakanya sendiri disimpan di file pustaka (library file dengan extension .LIB). Cara penulisannya adalah

```
#include <nama header file>
```

File stdio.h akan dicari mulai dari directory tempat header file tersebut di-install. Jika tdk ada akan dilanjutkan mencarinya ke current / working directory.

atau

```
#include "nama header file"
```

File stdio.h akan dicari mulai dari current / working directory, dan jika tdk ada akan dilanjutkan mencarinya ke directory tempat header file tersebut disimpan pada saat menginstall compiler-nya.

RESERVED WORD

Bahasa C standar ANSI mempunyai 32 kata kunci dan turbo C menambahkan 7 kata kunci. Kata kunci ini tidak dapat digunakan sebagai nama variable atau nama fungsi.

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

Ket : tanda * menunjukan tambahan dari turbo C

Bahasa C adalah bahasa yang *case sensitive*. Huruf kecil dibedakan dengan huruf besar dan semua kata kunci ditulis dalam huruf kecil.

IDENTIFIER NAME

Merupakan nama yang dibuat oleh programmer untuk pemberian nama variable, konstanta, fungsi atau label. Identifier name bebas dibuat dengan ketentuan sbb :

1. Panjang minimal 1 karakter, maksimal tidak terbatas, tetapi hanya 32 karakter pertama saja yang diperhatikan.
2. Karakter pertama harus berupa huruf atau garis bawah.
3. Tidak diperbolehkan menggunakan karakter khusus dan blank/spasi.
4. Case sensitive, huruf besar dan kecil dianggap beda.
5. Tidak diperkenankan menggunakan kata kunci.

STATEMENT

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 peng-akhir 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 ({ })

KONSEP TIPE DATA

TIPE DATA

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.

Tipe Data Dasar

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

KONSTANTA

Konstanta adalah suatu nilai yang tidak pernah berubah selama proses program. Dibagi menjadi 4, yaitu :

- a. Konstanta Numerik Integer
 Merupakan nilai numerik bilangan bulat. Sesuai dengan tipe data yang ada maka dapat berupa konstanta numerik integer bertanda diberi tanda, integer tidak bertanda (diberi huruf u atau U pada akhir nilainya), integer panjang bertanda (diberi huruf l atau L diakhir nilainya) dan integer panjang tidak bertanda (diberi huruf ul atau UL diakhir nilainya).
- b. Konstanta numerik pecahan
 Merupakan nilai numerik yang dapat mempunyai nilai pecahan dibelakang titik desimal. Dapat bernilai ketepatan tunggal (float), ketepatan ganda (double), atau ketepatan ganda panjang (long double).
- c. Konstanta Karakter dan String
 Merupakan nilai sebuah karakter yang ditulis diantara tanda petik ganda. Pengertian karakter disini meliputi huruf a-z atau A-Z, digit 0 – 9 karakter khusus dan karakter grafik.
- d. Konstanta Karakter Escape
Escape character banyak digunakan di statemen-statemen untuk menampilkan hasil. Konstanta karakter escape diawali dengan tanda ' \ '.

Karakter Escape	Fungsi
\a	bunyi bel (bell atau alert)
\b	mundur 1 spasi (backspace)
\f	ganti halaman (form feed)
\n	ganti baris baru (new line)

\r	ke kolom pertama, baris yang sama (carriage return)
\t	tabulasi horisontal
\v	tabulasi vertikal
\0	nilai kosong (null)
\'	karakter petik tunggal
\"	karakter petik ganda
\\	karakter backslash

OPERATOR

Macam-macam operator yang tersedia

Operator adalah suatu tanda atau simbol yang digunakan untuk suatu operasi tertentu. Bahasa C menyediakan operator pengerjaan, operator aritmatika, operator tipe, operator hubungan, operator logika, operator bitwise, operator ternary dan operator koma.

Operator Aritmatika (Arithmetic operator)

Operator	Fungsi	Jenjang
*	Perkalian	3
/	Pembagian	3
%	Pembagian modulo (Sisa pembagian)	3
+	Penjumlahan	4
-	Pengurangan	4

Operator aritmatika melibatkan 2 buah operand, terkadang operand yang digunakan berbeda tipenya. Untuk menghindari hal-hal yang tidak diinginkan maka kompilasi C mempunyai pedoman untuk operand yang berbeda tipe :

1. Tipe char akan dikonversikan ke tipe int
2. Tipe float akan dikonversikan ke tipe double
3. Jenjang tertinggi adalah mulai dari long double, double, long int, unsigned int, dan int. ini berarti tipe double dioperasikan dengan tipe int akan menghasilkan tipe double.

Operator Unary (Unary Operator)

Operator unary merupakan operator yang hanya menggunakan sebuah operand saja. Operator-operator unary mempunyai jenjang 2.

Operator	Fungsi
-	Unary minus
++	Increase dgn penambahan nilai 1
--	Decrease dengan pengurangan nilai 1
(tipe)	Cast
sizeof	Ukuran operand dalam byte
!	unary NOT
~	Komplemen 1 (bitwise NOT)
&	Menghasilkan alamat memori operand(operator pointer)
*	Menghasilkan nilai pengenalan alamatnya(operator pointer)

Operator Pengerjaan Aritmatika (Assignment Operator)

Operator pengerjaan digunakan untuk memindahkan nilai dari suatu ungkapan ke suatu pengenalan. Operator pengerjaan mempunyai jenjang 14.

Operator	Contoh	Ekuivalen dengan
=	A = B + C	Mengerjakan B + C ke A
+=	A += 1	A = A + 1
-=	A -= B	A = A - B
*=	A *= B	A = A * B
/=	A /= B	A = A / B
%=	A %= B	A = A % B

Operator Hubungan (Relational Operator)

Operator hubungan digunakan untuk menunjukkan hubungan antara 2 buah operand. Banyak digunakan untuk penyeleksian kondisi dengan statement if, do-while, atau while.

Operator	Fungsi	Jenjang
<	Lebih kecil dari	6
<=	Lebih kecil atau sama dengan	6
>	Lebih besar dari	6
>=	Lebih besar atau sama dengan	6
==	Sama dengan	7
!=	Tidak sama dengan	7

Operator Logika (Logical Operator)

Operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan.

Operator	Fungsi	Jenjang
&&	Logika DAN (AND)	11
	Logika ATAU (OR)	12

Operator Koma (Comma Operator)

Operator koma digunakan untuk menggabungkan beberapa ungkapan dengan proses yang berurutan dari ungkapan sebelah kiri koma ke ungkapan sebelah kanan koma. Operator koma mempunyai jenjang 16.

Operator Bitwise

Operator bitwise digunakan untuk memanipulasi bit-bit nilai data yang ada di memori. Operator-operator ini hanya dapat digunakan untuk tipe data char, int, dan long int.

Operator	Fungsi	Jenjang
<<	Pergeseran bit ke kiri	5
>>	Pergeseran bit ke kanan	5
&	Bitwise AND	8
^	Bitwise XOR (Exclusive OR)	9
	Bitwise OR	10
~	Bitwise NOT	1

Operator Pengerjaan Bitwise

Operator	Contoh	Ekuivalen dengan
<<=	A <<= 2	A = A << 2
>>=	A >>= 2	A = A >> 2
&=	A &= 0x1b	A = A & 0x1b
^=	A ^= 0x1b	A = A ^ 0x1b
=	A = 0x1b	A = A 0x1b

MEMASUKAN DATA

Fungsi-fungsi yang digunakan

Fungsi-fungsi pustaka yang digunakan untuk memasukkan data melalui keyboard, prototypenya ada di file judul `stdio.h` dan `conio.h`. Fungsi-fungsi yang menggunakan file judul `stdio.h` yaitu `gets()` dan `scanf()`. Sedangkan fungsi yang menggunakan file judul `conio.h`, yaitu `getche()`, `getchar()`, dan `getch()`.

Untuk memasukkan nilai karakter tidak terformat digunakan `getchar()`, `getch()`, dan `getche()`, tergantung dari karakteristik masing-masing.

getchar() Sintak: : `int getchar(void);`

Fungsi:

- mengembalikan sebuah karakter (nilai ASCII) berikutnya dari buffer keyboard.
- Karakter ditampilkan di layar monitor
- Menunggu sampai ada ENTER
- Header file ada di `stdio.h`

getch() Sintak: `int getch(void);`

Fungsi

- mengembalikan satu karakter dari buffer keyboard
- karakter tidak ditampilkan di layar monitor (no echo)
- Tidak menunggu sampai ada ENTER
- Cocok untuk membuat password
- Header file ada di `conio.h`

getche() Sintak : `int getche(void)`

Fungsi :

- mengembalikan satu karakter dari keyboard
- Karakter ditampilkan di layar (echo)
- Tidak menunggu sampai ada ENTER
- Header file ada di `conio.h`

gets() Sintak : `char *gets(char *buffer)`

Fungsi:

- membaca string dari keyboard sampai ketemu new-line dan disimpan pada buffer.
- Kemudian new-line di replace dengan null character
- Mengembalikan nilai NULL jika ada error dan mengembalikan argument-nya (buffer) jika sukses.

Untuk meg-input nilai data terformat digunakan perintah `scanf()`, Spesifikasi *format* adalah : "*% type*" dimana *type* bisa diganti dengan salah satu dari sbb:

Kode Format	Fungsi
<code>%c</code>	Membaca sebuah karakter
<code>%s</code>	Membaca nilai string
<code>%d</code>	Membaca nilai desimal integer
<code>%i</code>	Membaca nilai desimal integer
<code>%x</code>	Membaca nilai heksa desimal integer
<code>%o</code>	Membaca nilai oktal integer
<code>%f</code>	Membaca nilai pecahan
<code>%e</code>	Membaca nilai pecahan
<code>%g</code>	Membaca nilai pecahan
<code>%h</code>	Membaca nilai short integer desimal
<code>[...]</code>	Membaca karakter string yg diakhiri dengan karakter yg tidak ada didalam [...]
<code>[^..]</code>	Membaca karakter string yg diakhiri dengan karakter yg ada didalam [..]

Fungsi scanf mengembalikan tipe integer, dimana nilai nya menyatakan jumlah field yang sukses di assigned. Contoh:

```
int x,y,z,w;  
x=scanf("%d %d %d",&y,&z,&w);
```

maka :

- Jika di input dari keyboard 3 buah nilai interger 6 7 8, maka nilai x = 3;
- Jika di input dari keyboard 4 buah nilai interger 6 7 8 9 maka nilai x = 3 (karena 3 nilai yg sukses di-assigned masing-masing ke variabel y, z dan w)

Karakter Space, tab, linefeed, carriage-return, formfeed, vertical-tab, dan newline disebut "*white-space characters*". Contoh :

```
char ss[40];  
scanf("%s",ss);
```

Pada potongan program diatas, jika dimasukkan string "Selamat Pagi Pak" dari keyboard maka yg dimasukkan ke variabel ss hanya "Selamat" saja.

Untuk mengambil string yang diakhiri karakter tertentu (misalnya ENTER), dengan scanf, menggunakan format [^\n]. Menjadi :

```
char ss[40];  
scanf("%[^\n]",ss);
```

MENAMPILKAN HASIL

Fungsi-fungsi umum yang digunakan

Prototype dari fungsi-fungsi untuk menampilkan hasil terdapat pada file judul stdio.h bersifat standar yaitu putchar(), puts(), printf(), fprintf() dan conio.h bersifat tidak standar, dalam arti tidak semua kompiler C menyediakan yaitu clrscr(), gotoxy().

Untuk menampilkan hasil tidak terformat digunakan putchar() untuk menampilkan karakter tidak terformat dan puts() untuk menampilkan string tidak terformat. Maksudnya tidak terformat adalah lebar dan bentuk tampilannya tidak dapat diatur.

Sedangkan untuk hasil terformat digunakan perintah printf dengan spesifikai format sbb: *%[flags][width][.precision] type*;

Kode Format	Fungsi
%c	Menampilkan sebuah karakter
%s	Menampilkan nilai string
%d	Menampilkan nilai desimal integer
%i	Menampilkan nilai desimal integer
%u	Menampilkan nilai desimal integer tidak bertanda
%x	Menampilkan nilai heksa desimal integer
%o	Menampilkan nilai oktal integer
%f	Menampilkan nilai pecahan
%e	Menampilkan nilai pecahan dalam notasi scientific
%g	Sebagai pengganti '%f' atau '%e' tergantung mana yang terpendek
%p	Menampilkan suatu alamat memori untuk pointer

width : menentukan jumlah kolom yang disediakan

precision : menentukan jumlah angka dibelakang koma (untuk bilangan pecahan)

flags dapat diganti sbb:

- none : right justify (rata kanan)
- : left justify (rata kiri)
- +: untuk bilangan dimulai dgn tanda – jika negatip atau + jika positif

CONTOH 1:

```
printf("%6d", 34);          ....34
printf("%-6d", 34);        34....
```

CONTOH 2 :

```
printf("%10s", "GUNDAR");    ...GUNDAR
printf("%-10s", "GUNDAR");   GUNDAR ...
printf("%8.2f", 3.14159);    ....3.14
printf("%-8.3f", 3.14159);   3.141...
printf("%c\n",65);           //akan ditampilkan A
printf("%x\n",'A');           // akan ditampilkan 41
printf("%o\n",65);           // akan ditampilkan 101
printf("%+d\n",34);          // akan ditampilkan +34
printf("%+d\n",-45);         // akan ditampilkan -45
printf("%e\n",3.14);         // akan ditampilkan 3.140000e+000
```

CONTOH 3:

```
#include <stdio.h>
```

```
int main(){
```

```
    char ss[]="Selamat Datang";
    printf("123456789012345678901234567890\n");
    printf("%.10s di Gundar\n",ss);
    printf("%10s di Gundar\n",ss);
    printf("%-10s di Gundar\n",ss);
    printf("%.20s di Gundar\n",ss);
    printf("%20s di Gundar\n",ss);
    printf("%-20s di Gundar\n",ss);
    printf("%20.10s di Gundar\n",ss);
    printf("%-20.10s di Gundar\n",ss);
    return 0;
}
```

Output Program disamping sbb:

```
123456789012345678901234567890
Selamat Da di Gundar
Selamat Datang di Gundar
Selamat Datang di Gundar
Selamat Datang di Gundar
Selamat Datang di Gundar
Selamat Datang di Gundar
Selamat Datang di Gundar
Selamat Da di Gundar
Selamat Da di Gundar
```

putchar() Sintak: int putchar(int c)

Fungsi :

- Menampilkan karakter ke layar monitor pada cursor, kemudian setelah ditampilkan cursor bergerak ke posisi berikutnya.
- Mengembalikan EOF jika error, dan mengembalikan karakter yang ditampilkan jika sukses
- Putchar adalah macro yang sama artinya dengan: putc(c, stdout)
- Header File : stdio.h

putch() Sintak : int putch(int ch)

Fungsi :

- menampilkan karakter ascii di ch di monitor tanpa memindahkan kursor ke posisi berikutnya
- Header file : conio.h
- Mengembalikan EOF jika error, dan mengembalikan karakter yang di tampilkan jika sukses.

puts() Sintak : int puts(const char *str);

Fungsi:

- Menampilkan string str ke layar monitor dan memindahkan kursor ke baris baru.
- Header file: stdio.h
- Mengembalikan nilai non-negative jika sukses dan EOF jika ada error.

CONTOH :

```
puts("Selamat Datang");
puts("Di GUNDAR");
```

Tampilan di layar monitor:

```
Selamat Datang
Di GUNDAR
```


Penempatan kursor

- Layar dapat dihapus dengan menggunakan fungsi: ***clrscr()***;
- Kursor dapat dipindahkan ke posisi manapun di dalam layar monitor dengan menggunakan fungsi : ***gotoxy(col,row)***; dimana col = kolom dan row = baris
- Sebagian dari baris, mulai posisi kursor hingga akhir baris (end of line), dapat dihapus dengan fungsi: ***clreol()***;
- Function prototype untuk fungsi *gotoxy()*, *clrscr()*, *clreol()* pada bahasa C terdapat pada header file : **<conio.h>**