

Struktur Dasar Bahasa C

Tipe Data

Dalam bahasa C terdapat lima tipe data dasar, yaitu :

No	Tipe Data	Ukuran	Range (Jangkauan)	Format	Keterangan
1	char	1 byte	- 128 s/d 127	%c	Karakter/string
2	int	2 byte	- 32768 s/d 32767	%i , %d	Integer/bilangan bulat
3	float	4 byte	- 3.4E-38 s/d 3.4E+38	%f	Float/bilangan pecahan
4	double	8 byte	- 1.7E-308 s/d 1.7+308	%lf	Pecahan presisi ganda
5	void	0 byte	-	-	Tidak bertipe

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
void main()
{ int x;
float y;
char z;
double w;
clrscr(); /* untuk membersihkan layar */
x = 10; /* variable x diisi dengan 10 */
y = 9.45; /* variable y diisi dengan 9.45 */
z = 'C'; /* variable z diisi dengan karakter "C" */
w = 3.45E+20; /* variable w diisi dengan 3.45E+20 */
printf("Nilai dari x adalah : %i\n", x); /* Menampilkan isi variable x */
printf("Nilai dari y adalah : %f\n", y); /* Menampilkan isi variable y */
printf("Nilai dari z adalah : %c\n", z); /* Menampilkan isi variable z */
printf("Nilai dari w adalah : %lf\n", w); /* Menampilkan isi variable w */
getch(); }
```

Konstanta

Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Bahasa C juga menyediakan beberapa karakter khusus yang disebut karakter escape, antara lain :

- \a : untuk bunyi bell (alert)
- \b : mundur satu spasi (backspace)
- \f : ganti halaman (form feed)
- \n : ganti baris baru (new line)
- \r : ke kolom pertama, baris yang sama (carriage return)
- \v : tabulasi vertical
- \0 : nilai kosong (null)
- \' : karakter petik tunggal
- \" : karakter petik ganda
- \\ : karakter garis miring

Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program.

Nama dari suatu variable dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :



- Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C bersifat case-sensitive artinya huruf besar dan kecil dianggap berbeda. Jadi antara **nim**, **NIM** dan **Nim** dianggap berbeda.
- Tidak boleh mengandung spasi.
- Tidak boleh mengandung symbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk symbol khusus yang tidak diperbolehkan antara lain : \$, ?, %, #, !, &, *, (,), -, +, =
- Panjangnya bebas, tetapi hanya 32 karakter pertama yang terpakai.

Deklarasi

Deklarasi diperlukan bila kita akan menggunakan pengenal (identifier) dalam program. Identifier dapat berupa variable, konstanta dan fungsi.

- **Deklarasi Variabel**

Bentuk umum pendeklarasian suatu variable adalah :

Nama_tipe nama_variabel;

Contoh :

```
int x;                // Deklarasi x bertipe integer
char y, huruf, nim[10]; // Deklarasi variable bertipe char
float nilai;         // Deklarasi variable bertipe float
double beta;        // Deklarasi variable bertipe double
int array[5][4];    // Deklarasi array bertipe integer
char *p;            // Deklarasi pointer p bertipe char
```

- **Deklarasi Konstanta**

Dalam bahasa C konstanta dideklarasikan menggunakan preprocessor **#define**.

Contohnya : `#define PHI 3.14`
`#define nim "0111500382"`
`#define nama "Sri Widhiyanti"`

- **Deklarasi Fungsi**

Fungsi merupakan bagian yang terpisah dari program dan dapat diaktifkan atau dipanggil dimanapun di dalam program. Fungsi dalam bahasa C ada yang sudah disediakan sebagai fungsi pustaka seperti `printf()`, `scanf()`, `getch()` dan untuk menggunakannya tidak perlu dideklarasikan.

Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer. Bentuk umum deklarasi sebuah fungsi adalah :

Tipe_fungsi nama_fungsi(parameter_fungsi);

Contohnya :

```
float luas_lingkar(int jari);
void tampil();
int tambah(int x, int y);
```

Operator

- **Operator Penugasan**

Operator Penugasan (*Assignment operator*) dalam bahasa C berupa tanda sama dengan ("="). Contoh :

```
nilai = 80;
A = x * y;
```



- **Operator Aritmatika**

Bahasa C menyediakan lima operator aritmatika, yaitu :

- ◆ * : untuk perkalian
- ◆ / : untuk pembagian
- ◆ % : untuk sisa pembagian (modulus)
- ◆ + : untuk penambahan
- ◆ - : untuk pengurangan

- **Contoh Program 1 :**

```
#include "stdio.h"
#include "conio.h"
void main()
{ clrscr(); // untuk membersihkan layar
printf("Nilai dari 9 + 4 = %i", 9 + 4);
printf("Nilai dari 9 - 4 = %i", 9 - 4);
printf("Nilai dari 9 * 4 = %i", 9 * 4);
printf("Nilai dari 9 / 4 = %i", 9 / 4);
printf("Nilai dari 9 \% 4 = %i", 9 % 4);
getch();
}
```

- **Operator Hubungan (Perbandingan)**

Operator Hubungan digunakan untuk membandingkan hubungan antara dua buah operand (sebuah nilai atau variable. Operator hubungan dalam bahasa C :

Operator	Arti	Contoh	
<	Kurang dari	$x < y$	Apakah x kurang dari y
<=	Kurang dari sama dengan	$x <= y$	Apakah x kurang dari sama dengan y
>	Lebih dari	$x > y$	Apakah x lebih dari y
>=	Lebih dari sama dengan	$x >= y$	Apakah x lebih dari sama dengan y
==	Sama dengan	$x == y$	Apakah x sama dengan y
!=	Tidak sama dengan	$x != y$	Apakah x tidak sama dengan y

- **Operator Logika**

Jika operator hubungan membandingkan hubungan antara dua buah operand, maka operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan.

Operator logika ada tiga macam, yaitu :

- ◆ && : Logika AND (DAN)
- ◆ || : Logika OR (ATAU)
- ◆ ! : Logika NOT (INGKARAN)



- **Operator Bitwise**

Operator bitwise digunakan untuk memanipulasi bit-bit dari nilai data yang ada di memori.

Operator bitwise dalam bahasa C :

- ◆ << : Pergeseran bit ke kiri
- ◆ >> : Pergeseran bit ke kanan
- ◆ & : Bitwise AND
- ◆ ^ : Bitwise XOR (exclusive OR)
- ◆ | : Bitwise OR
- ◆ ~ : Bitwise NOT

- **Operator Unary**

Operator Unary merupakan operator yang hanya membutuhkan satu operand saja. Dalam bahasa C terdapat beberapa operator unary, yaitu :

Operator	Arti/Maksud	Letak	Contoh	Equivalen
-	Unary minus	Sebelum operator	A + -B * C	A + (-B) * C
++	Peningkatan dengan penambahan nilai 1	Sebelum dan sesudah	A++	A = A + 1
--	Penurunan dengan pengurangan nilai 1	Sebelum dan sesudah	A--	A = A - 1
sizeof	Ukuran dari operand dalam byte	Sebelum	sizeof(I)	-
!	Unary NOT	Sebelum	!A	-
~	Bitwise NOT	Sebelum	~A	-
&	Menghasilkan alamat memori operand	Sebelum	&A	-
*	Menghasilkan nilai dari pointer	Sebelum	*A	-

Contoh Program 1 :

```

/* Perbedaan operator peningkatan ++ yang diletakkan di depan dan dibelakang
operand */
#include <stdio.h>
#include <conio.h>
void main()
{
int x, nilai;
clrscr();
x = 5;
nilai = ++x; /* berarti x = x + 1; nilai = x; */
printf("nilai = %d, x = %d\n", nilai, x);
nilai = x++; /* berarti nilai = x; nilai = x + 1; */
printf("nilai = %d, x = %d\n", nilai, x);
getch();
}

```



Kata Tercadang (Reserved Word)

Bahasa C standar ANSI memiliki 32 kata tercadang (reserved word) dan Turbo C menambahkannya dengan 7 kata tercadang. Semua *reserved word* tidak boleh digunakan dalam penamaan identifier (variable, nama fungsi dll). Kata Tercadang yang tersedia dalam bahasa C adalah sbb:

*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	

Komentar Program

Komentar program hanya diperlukan untuk memudahkan pembacaan dan pemahaman suatu program (untuk keperluan dokumentasi program). Dengan kata lain, komentar program hanya merupakan keterangan atau penjelasan program. Untuk memberikan komentar atau penjelasan dalam bahasa C digunakan pembatas `/*` dan `*/` atau menggunakan tanda `//` untuk komentar yang hanya terdiri dari satu baris. Komentar program tidak akan ikut diproses dalam program (akan diabaikan).

Contoh Program :

```
#include "stdio.h"
#include "conio.h"
void main()
{
clrscr();                /* Ini untuk membersihkan layar tampilan */
printf("Contoh Penggunaan Komentar"); //komentar tidak ikut diproses
getch();                /* Dan ini untuk menahan tampilan di layar */
}
```

