



Asia Cyber
University

Algoritma & Pemrograman 1

Model Data pada C++

2

Pengertian

- Tipe Data :
- Jenis data berdasarkan isi dan sifatnya
- Logika : Kotak Kosong dibuat khusus untuk suatu benda dengan jenis tertentu
- Variabel :
 - Tempat untuk menyimpan data dengan tipe tertentu yang isinya bisa diubah-ubah sesuai dengan tipenya.
 - Setiap Variabel hanya bisa menyimpan satu buah nilai.
 - Dapat diberi nama sesuka hati.
- Konstanta :

Suatu Variabel yang ditentukan nilai standarnya (default) dari awal dan nilainya tidak berubah.

Variable

- Berbeda dengan pendeklarasian variabel di bahasa pemrograman lain, dalam C++ sebelum mendeklarasikan variabel, hal pertama yang harus dideklarasikan adalah tipe data yang akan digunakan untuk menampung data.
- Format penulisannya adalah :

Tipe_data pengenalan = nilai ;

Contoh : int a;
 float nomor;
 int a = 10;
 char s = 'a';

- Jika hendak mendeklarasikan beberapa variabel sekaligus dengan tipe data yang sama, dapat dilakukan dengan 2 cara, yaitu :
 - int a;
 - int b;
 - int c;

} int a,b,c;

Variable

- Contoh Program :

```
// bekerja dengan variabel
#include <iostream>
Using namespace std;

int main ()
{
    // inisialisasi variabel :
    int a, b;
    int hasil;

    // proses :
    a = 5;
    b = 2;
    a = a + 1;
    hasil = a - b;

    cout << hasil;           // cetak hasilnya :
    return 0;                // menghentikan program :
}
```

Konstanta

- Konstanta mirip dengan variable, namun memiliki nilai tetap.
- Konstanta dapat berupa nilai Integer, Float, Karakter dan String. Pendeklarasian konstanta dapat dilakukan dengan 2 cara :

1. Menggunakan (#define)

- Pendeklarasian dengan `#define` tanpa diperlukan adanya tanda `=` untuk memasukkan nilai ke dalam pengenal dan juga tanpa diakhiri dengan tanda semicolon(`;`).

- Format penulisannya adalah :

`#define pengenal_nilai`

Contoh : `#define phi 3.14159265`

`#define Newline '\n'`

`#define lebar 100`

2. Menggunakan (const)

Contoh : `const int lebar = 100;`

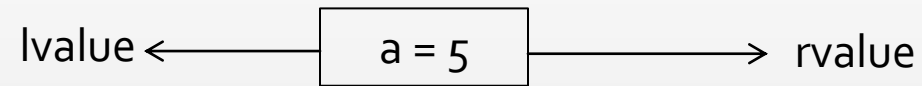
`const char tab = '\t';`

`const zip = 1212; //Tanpa tipe data, otomatis menjadi tipe int.`

Operator

- **Operator Assign (=)**

Operator (=), akan memberikan nilai ke dalam suatu variable.



- Keterangan :

- Artinya memberikan nilai 5 ke dalam variable a.
- lvalue harus selalu berupa variable,
- rvalue dapat berupa variable, nilai, konstanta, hasil operasi atau kombinasinya.

- **Operator Aritmatika (+, -, *, /, %)**

Operator % = Modulus, yaitu untuk mengetahui sisa hasil bagi.

Misal : $a = 11 \% 3$, maka isi variable $a = 2$ karena sisa hasil bagi 11 dan 3 adalah 2.

Operator

- **Operator Majemuk (+=, -=, *=, /=, %=, <<=, >>=, &=, |=)**

- ❖ Menyederhanakan perhitungan dengan format penulisan Operator Majemuk

Contoh :

`a += 5` sama artinya dengan `a = a+5`

`a *= 5` sama artinya dengan `a = a*5`

`a /= 5` sama artinya dengan `a = a/5`

`a %= 5` sama artinya dengan `a = a % 5`

- **Operator Penaikan dan Penurunan (++ dan --)**

- ❖ Operator penaikan (++) akan menaikkan atau menambahkan 1 nilai variable. Sedangkan operator (--) akan menurunkan atau mengurangi 1 nilai variable.

- ❖ Misalnya :
`a++;`
`a+=1;`
`a=a+1;`

Operator

- **Operator Relasional (`==`, `!=`, `>`, `<`, `>=`, `<=`)**
 - ❖ Yang dihasilkan dari operator ini bukan berupa sebuah nilai, yang berupa bilangan bool yaitu benar atau salah.

Operator	Keterangan
<code>==</code>	Sama Dengan
<code>!=</code>	Tidak Sama Dengan
<code>></code>	Lebih besar dari
<code><</code>	Kurang dari
<code>>=</code>	Lebih besar dari atau sama dengan
<code><=</code>	Kurang dari atau sama dengan

Contoh :

(`7==5`) hasilnya adalah false

(`5>4`) hasilnya adalah true

(`5<5`) hasilnya adalah false

Operator

- **Operator Logika (!, &&, ||)**

- ❖ Operator logika juga digunakan untuk memberikan nilai atau kondisi true dan false.
- ❖ Biasanya operator logika dipakai untuk membandingkan dua kondisi.
- ❖ Operator Logika &&

Contoh : `((5==5) && (3>6))` mengembalikan nilai false, karena `(true && false)`

Operator Logika NOT (!)

Contoh : `!(5==5)` akan mengembalikan nilai false, karena `!(true)`.

Operator Kondisional

- Operator Kondisional (?)

- ❖ Format penulisan operator kondisional adalah :

kondisi ? hasil1 : hasil2

Jika kondisi benar maka yang dijalankan adalah hasil1 dan jika kondisi salah, maka akan dijalankan hasil2.

Contoh : $7 == 5 ? 4 : 3$ hasilnya adalah 3, karena 7 tidak sama dengan 5

$5 > 3 ? a : b$ hasilnya adalah a, karena 5 lebih besar dari 3

```
#include <iostream.h>
void main()
{
    int m = 26, n = 82;
    int min = m < n ? m : n;
    cout<<"Bilangan terkecil adalah "<<min<<endl;
}
```

Outputnya :

Bilangan terkecil adalah 26

Type Data Dasar

- Bilangan Bulat (Integer, Short, Long)
 - ✓ Digunakan untuk data angka yang tidak mengandung angka dibelakang koma.
Misalnya 3, 21, 78

Type Data	Byte	Jangkauan	Deklarasi
Integer	2	-32.768, ..., 32.767	int, signed int signed
Short	1	-128, ..., 127	short short int
Long	4	-2.147.435.648, ..., 2.147.435.647 (2.1 Milyar)	long, long int signed long
Unsigned Int	2	0, ..., 65.535	unsigned int unsigned
Unsigned Long	4	0, ..., 4,294,967,595	unsigned long unsigned long int

Catatan : 1 Byte = 8 Bit, 2 Byte = 16 Bit, 4 Byte = 32 Bit

Type Data Dasar

- Bilangan Bulat (Integer, Short, Long)
 - ✓ Unsigned digunakan untuk data yang positif
 - ✓ Contoh :

```
#include <iostream>
int main()
{
    int x; //mendeklarasikan variabel x dengn tipe data int
    x=3; //melakukan assigment terhadap variabel x
    cout <<"Nilai x adalah "<<x;
    return 0;
}
```

Type Data Dasar

- Bilangan Desimal (Floating Point)
 - ✓ Data bilangan yang mengandung angka dibelakang koma.
 - ✓ Terdiri dari :
 - Single Precision (32 bit atau 4 byte)
 - Double Precision (64 – 80 bit atau 8 - 10 byte)
 - ✓ Menggunakan notasi Exponensial "e" atau "E"
- Contoh : $3.14E5 = 3.14 \times 10^5$

Type Data	Byte	Jangkauan	Deklarasi	Jumlah Digit
Float	4	$3.4 \times E^{-38}, \dots, 3.4 \times E^{+38}$	float	5 – 7
Double	8	$1.7 \times E^{-308}, \dots, 1.7 \times E^{+308}$	double	15 - 16
Long Double	10	$3.4 \times E^{-4932}, \dots, 1.1 \times E^{+4932}$	long double	19

Tipe Data Dasar

- Bilangan Desimal (Floating Point)

✓ Contoh :

```
#include <iostream>
using namespace std;
int main()
{
    float a;
    int b;
    a = 29.51;    //melakukan assigment terhadap variabel a
    cout << "Nilai a adalah "<<a;
    b = int(a);   //merubah nilai float ke integer
    cout << "\nNilai b adalah "<<b;
    return 0;
}
```

Type Data Dasar

- Karakter & String

- ✓ Merepresentasikan data berupa karakter (char)
- ✓ String dinyatakan dengan pointer dari tipe char yaitu **char***.
- ✓ Karakter digunakan untuk menyimpan sebuah karakter
- ✓ Karakter adalah turunan dari integer (menggunakan kode *ASCII*).
- ✓ String adalah gabungan dari karakter

Tipe Data	Byte	Jangkauan	Deklarasi
Character	1	-128, ..., 127	char
Unsigned Char	1	0, ..., 255	unsigned char

- ✓ Contoh :
char kata = 'B';
char kata[] = "Belajar";
char* kata = "Belajar";

```
char kata[4];  
  
kata[0] = 'c';  
kata[1] = '+';  
kata[2] = '+';  
kata[3] = '\0';
```

Tipe Data Dasar

- Karakter & String

Contoh :

```
#include <iostream>
using namespace std;
int main()
{
    char Karakter ='A';
    char* nama ="Rudi";
    char Jurusan[15]="Informatika";
    cout <<Karakter<<endl;
    cout <<nama<<endl;
    cout <<Jurusan<<endl;
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    char c = 'A';
    int k;

    //konversi string ke int
    k = (int) c;

    cout <<"Nilai konfersi adalah <<k;
    return 0;
}
```


Table ASCII

TABEL ASCII (0 - 127),
karakter dasar ASCII dalam 7 bit (bit ke-8 / paling kiri = 0)

Kode ASCII	Char	Kode ASCII	Char	Kode ASCII	Char	Kode ASCII	Char	Kode ASCII	Char
0	NULL	32	Spasi	58	:	91	[123	{
1		33	!	59	;	92	\	124	
2		34	"	60	<	93]	125	}
3		35	#	61	=	94	^	126	~
4		36	\$	62	>	95	_	127	□
5		37	%	63	@	96			
6		38	&	64	a				
7	bell	39	'						
8		40	(65	A	97	a		
9	tab	41)	66	B	98	b		
10	line feed	42	*	67	C	99	c		
11	home	43	+	68	D	100	d		
12	form feed	44	,	69	E	101	e		
13	Carriage return	45	-	70	F	102	f		
14		46	.	71	G	103	g		
15		47	/	72	H	104	h		
16				73	I	105	i		
17				74	J	106	j		
18		48	0	75	K	107	k		
19		49	1	76	L	108	l		
20		50	2	77	M	109	m		
21		51	3	78	N	110	n		
22		52	4	79	O	111	o		
23		53	5	80	P	112	p		
24		54	6	81	Q	113	q		
25		55	7	82	R	114	r		
26		56	8	83	S	115	s		
27		57	9	84	T	116	t		
28	cursor ke kanan			85	U	117	u		
29	ke kiri			86	V	118	v		
30	ke atas			87	W	119	w		
31	ke bawah			88	X	120	x		
				89	Y	121	y		
				90	Z	122	z		

Tipe Logika (Boolean)

- Merepresentasikan data-data yang hanya mengandung dua buah nilai, yaitu nilai logika (boolean) terdiri dari :

- True : Nilai benar (nilai 1)
- False : Nilai salah (nilai 0).

Operator	Arti
&&	And
	Or
!	Not

- Operator Logika biasa digunakan untuk menghubungkan ungkapan suatu relasi

Operand 1	Operand 2	Hasil	
			&&
False	False	False	False
False	True	True	False
True	False	True	False
True	True	True	True

Tipe Logika (Boolean)

- Merepresentasikan data-data yang hanya mengandung dua buah nilai, yaitu nilai logika

(boolean) terdiri dari :

- True : Nilai benar (nilai 1)
- False : Nilai salah (nilai 0).

Operator	Arti
&&	And
	Or
!	Not

- Operator Logika biasa digunakan untuk menghubungkan ungkapan suatu relasi

Operand 1	Operand 2	Hasil	
			&&
False	False	False	False
False	True	True	False
True	False	True	False
True	True	True	True

Terimakasih