ELEMEN DASAR C++

HIMPUNAN KARAKTER
Himpunan karakter pada C++ terdiri huruf, digit maupun simbol-simbol lainnya (termasuk spasi dan karakter kontrol).
Huruf, contoh : A s/d Z dan a s/d z
Digit, contoh : 0 s/d 9
Simbol, contoh : + - * % / dan sebagainya

C++ mempunyai cara untuk menyatakan karakter-karakter yang tidak mempunyai kode tombol (seperti karakter tombol) misalnya \n.

PENGENAL(IDENTIFIER)
Pengenal adalah suatu nama yang biasa dipakai dalam pemrograman untuk menyatakan :
- variabel
- konstanta bernama
- tipe data
- fungsi
- label
- obyek
- pengenal-pengenal lain yang didefinisikan oleh pemrogram

Suatu pengenal dapat berupa satu atau beberapa karakter :
- huruf
- digit
- garis bawah
- berawalan dengan huruf atau garis bawah

panjang maksimal nama pengenal pada C++ tergantung oleh kompiler yang digunakan, misalnya Borland C++ memperkenankan nama pengenal hingga 32 karakter, sedangkan Turbo C++ menjamin nama yang signifikan hingga 31 karakter.

Disarankan agar pemberian nama pengenal menggunakan kata yang berarti dan mudah dibaca, misal : *gaji_pegawai* lebih baik daripada *g* ataupun *gajipegawai*. Pada C++, huruf kecil dan huruf kapital pada suatu pengenal tidak dianggap sama. Sifat ini dikenal dengan nama *case sensitive.* Itulah sebabnya pengenal seperti *NAMA* dan *nama* ataupun *Nama* menyatakan tiga pengenal yang berbeda.
Contoh :
Nama pengenal yang benar :
- nama
- NAMA
- nama_barang
- kuartal_2

Nama pengenal yang salah :
- 2semester (tidak boleh diawali angka)
- nama-barang (tanda minus tidak diperkenankan)
- #barang (simbol # tidak boleh digunakan)
- nama barang (tidak boleh mengandung spasi)

KATA KUNCI

Kata kunci (*keyword*) adalah pengenal sistem yang mempunyai makna khusus bagi kompiler. Kegunaan dari golongan ini tidak dapat diubah. Karena itu, kata kunci tidak dapat digunakan sebagai pengenal yang dibuat oleh pemrogram.

Contoh :

asm	else	operator	template
auto	enum	private	this
break	extern	protected	typedef
case	float	public	union
char	for	register	unsigned
class	friend	return	virtual
const	goto	short	void
continue	if	signed	volatile
default	inline	sizeof	while
delete	int	static	
do	long	struct	
double	new	switch	

TIPE DATA

Ukuran memori yang diperlukan untuk masing-masing tipe data sangat bergantung pada perangkat keras dari komputer yang digunakan. Karena itu jangkauan bilangan dari masing-masing tipe data juga bisa berlainan antara satu jenis mesin dengan mesin lain.

Tabel ukuran berbagai tipe dasar :

TIPE DATA	UKURAN MEMORI	JANGKAUAN NILAI	JUMLAH DIGIT PRESISI
char	1 byte	-128 hingga +127	-
Int	2 byte	-32768 hingga +32767	-
long	4 byte	-2.147.438.648 hingga 2.147.438.647	-
float	4 byte	$3.4x10^{-38}$ hingga $3.4x10^{+38}$	6 – 7
double	8 byte	$1.7x10^{-308}$ hingga$1.7x10^{+308}$	15 – 16
long double	10 byte	$3.4x10^{-4932}$ hingga$1.1x10^{+4932}$	19

VARIABEL DAN KONSTANTA

Data pada C++ tersusun dari variabel dan konstanta. Variabel digunakan dalam program untuk menyimpan suatu nilai dan nilai yang ada padanya dapat diubah selama eksekusi program berlangsung. Sedangkan konstanta menyatakan nilai yang tetap misalnya nilai phi = 3.14.

Variabel yang digunakan dalam program harus didefinisikan terlebih dahulu.
Bentuk definisi variabel : *tipe daftar_variabel;*
Contoh :

```
int jumlah;
float harga_per_unit, total_harga;
```

Jika suatu variabel diisi dengan nilai diluar jangkauannya, maka nilai yang akan tersimpan akan diubah sesuai dengan jangkauannya. Misalnya bila suatu variabel bertipe int diberi nilai 75000, yang tersimpan dalam variabel tersebut adalah 9464. (mirip dengan logika timbangan manual).

Bentuk pernyataan yang digunakan untuk memberikan nilai ke variabel yang telah dideklarasikan adalah : *variabel = nilai;*. Pernyataan ini disebut juga dengan pernyataan penugasan .

Variabel dan Konstanta Bertipe char

Bentuk pendefinisian variabel bertipe char adalah : *char huruf;* dalam hal ini variabel huruf dapat menampung sebuah karakter. Untuk menuliskan sebuah konstanta bertipe char, karakter perlu ditulis di dalam tanda petik tunggal, misalnya : ‘y’, ‘Y, ‘%’, ‘$’ dan sebagainya.

Karakter yang ditulis dalam bentuk \ mempunyai arti tersendiri. Karakter-karakter khusus seperti ini biasa disebut *escape sequence characters*

KARAKTER	KETERANGAN
\0	Karakter ber-ASCII nol (karakter null)
\a	Karakter bell
\b	Karakter backspace
\f	Karakter formfeed (ganti halaman)
\n	Karakter newline (pindah baris)
\r	Karakter carriage return (ke awal baris) tanpa linefeed
\t	Karakter tab horizontal
\v	Karakter tab vertical
\\	Karakter \
\’	Karakter ‘
\”	Karakter “
\?	Karakter ?
\000	Karakter yang nilai oktalnya adalah 3 digit oktal
\xhh	Karakter yang nilai heksadesimalnya hh (dua digit heksadesimal)

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
 char kode;
 clrscr;
 cout << "\t*************************\n";
 cout << "\tProgram : Cetak\n";
 cout << "\tDibuat oleh : Yudi Irawan\n";
 cout << "\tTanggal : 04/04/2009\n";
 cout << "\t*************************\n";
 kode ='4';
 cout << "Kode Anda : " << kode;
}
```

Hasil :

```
        *************************
        Program : Cetak
        Dibuat oleh : Yudi Irawan
        Tanggal : 04/04/2009
        *************************
Kode Anda : 4
```

Variabel dan Konstanta Bertipe int

Variabel bertipe int didefinisikan dengan bentuk : *int bilangan;*. Sebuah konstanta bertipe int adalah bilangan bulat yang terletak antara minus 32768 hingga plus 32767 dan tidak mengandung titik desimal.

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
 int bilangan;
 clrscr;
 bilangan = 32763;
 cout <<"Isi bilangan =" <<bilangan;
 bilangan = -44;
 cout <<"\nIsi bilangan =" <<bilangan;
}
```

Hasil :

```
Isi bilangan =32763
Isi bilangan =-44
```

Variabel dan Konstanta Bertipe int

Apabila diinginkan, untuk memproses bilangan bulat yang nilainya lebih besar daripada int, Anda dapat menggunakan tipe long. Suatu variabel bertipe long didefinisikan dengan cara : *long gaji;*.

Pada contoh di atas, gaji didefinisikan bertipe long. Dengan demikian variabel ini dapat menampung nilai ratusan juta. Adapun konstanta bertipe long biasa ditulis dengan akhiran L, misal : gaji = 2000000L;

Namun jika nilai konstanta melebihi 65.535 dengan sendirinya akan diinterpretasikan sebagai tipe long, sekalipun tanda L tidak diberikan.

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
 long tunjangan, gaji ;
 clrscr;
 tunjangan  = 1000000;
 gaji = 2000000L;
 cout <<"Tunjangan Rp. =" <<tunjangan;
 cout <<"\nGaji Rp. =" <<gaji;
}
```

Hasil :

```
Tunjangan Rp. =1000000
Gaji Rp. =2000000
```

Variabel dan Konstanta Bertipe float, double dan long double

Seandainya diinginkan untuk memproses bilangan yang mengandung nilai pecahan, Anda bisa menggunakan tipe *float, double* atau *long double*. Ketiga tipe yang berhubungan dengan bilangan pecahan ini mempunyai perbedaan dalam hal :

- Kepresisian data
- Jangkauan nilai yang dicakup

misalnya :

```
float panjang;
double phi;
```

```
long double total;
```

Bilangan 3,14159 biasa disebut sebagai konstanta bilangan pecahan. Nilai bilangan pecahan juga dapat dinyatakan dalam bentuk eksponensial, misal :
2756.3 dapat ditulis menjadi 2.7563E+3 atau 2.7563e+3
sedangkan 0.0123 dapat ditulis 1.23E-2 atau 1.23e-2
Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
 float a,b,c,d;
 clrscr;
 a = 2756.3;
 b = 2.7563e+3;
 c = 0.0123;
 d = 1.23e-2;
 cout <<"Bilangan a =" <<a;
 cout <<"\nBilangan b =" <<b;
 cout <<"\nBilangan c =" <<c;
 cout <<"\nBilangan d =" <<d;
}
```

Hasil :

```
Bilangan a =2756.3
Bilangan b =2756.3
Bilangan c =0.0123
Bilangan d =0.0123
```

Adakalanya dalam penulisan program, variabel langsung diberi nilai awal setelah didefinisikan, misal :

```
int jumlah;
jumlah = 10;
```

Pernyataan di atas sebenarnya dapat disingkat menjadi : int jumlah = 10;
Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
  int alas=10;
 float phi = 3.14;
 float hasil = alas*phi;
 clrscr;
 cout <<"Alas = " <<alas;
 cout <<"\nNilai phi = " <<phi;
 cout <<"\nHasil kali = " <<hasil;
}
```

Hasil :

```
Alas = 10
Nilai phi = 3.14
Hasil kali = 31.4
```

PEMODIFIKASI TIPE UNSIGNED DAN SIGNED

Pemodifikasi tipe *unsigned* diterapkan pada data bertipe bilangan bulat dan menyebabkan nilai yang terkandung didalamnya selalu bernilai positif, sedangkan tipe *signed* merupakan default dari tipe data dasar, yang menyatakan data bisa bernilai positif maupun negatif.

PEMODIFIKASI TIPE	PERSAMAAN	JANGKAUAN NILAI
signed char	Char	-128 s/d 127
signed int	Int	-32.768 s/d 32.767
signed short int	Short, signed short	-32.768 s/d 32.767
signed long int	Long, long int, signed long	-2.147.483.648 s/d 2.147.483.687
unsigned char	Tidak ada	0 s/d 255
unsigned int	Unsigned	0 s/d 65.535
unsigned short	Unsigned short	0 s/d 65.535
unsigned long int	Unsigned long	0 s/d 4.294.967.295

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
 int x;
 clrscr;
 unsigned int y;
 x=1;
 cout <<"Nilai x = " <<x;
 y=-1;
 cout <<"\nNilai y = " <<y;
}
```

Hasil :

```
Nilai x = 1
Nilai y = 65535
```

Tampak bahwa sekalipun nilai yang diberikan ke x dan y sama-sama bernilai -1 tetapi yang tersimpan pada kedua variabel berbeda, karena y didefinisikan sebgai unsigned yang hanya menampung bilangan positif.

KONSTANTA OKTAL DAN HEKSADESIMAL

Selain dalam bentuk desimal (basis 10), konstanta bilangan bulat dapat disajikan dalam bentuk sistem oktal (basis 8) ataupun sistem heksadesimal (basis16).

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
 int x,y;
 clrscr;
 x=010; // bilangan oktal
 y=0x10; // bilangan heksadesimal
 cout <<"Nilai 10 oktal = " <<x <<" desimal";;
 cout <<"\nNIlai 10 heksadesimal = " <<y <<" desimal";
}
```

Hasil :

```
Nilai 10 oktal = 8 desimal
Nilai 10 heksadesimal = 16 desimal
```

KONSTANTA STRING

Konstanta string merupakan deretan karakter yang diawali dan diakhiri dengan tanda petik ganda (“), misal : “Selamat Belajat C++”. Konstanta ini berbeda dengan kontanta karakter (yang diawali dengan tanda petik tunggal), misal : “A” tidak sama dengan ‘A’.

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
 clrscr;
 cout <<"\"Wingardium Leviosa..\", kata Harry Potter\n";
}
```

Hasil :

```
"Wingardium Leviosa..", kata Harry Potter
```

KONSTANTA BERNAMA

C++ memungkinkan pendefinisian suatu konstanta bernama. Hal ini dilakukan dengan menggunakan kata kunci *cons.* Bentuk penulisan : *const tipe_data nama_konstanta = nilai;*
misal : const float PHI = 3.141592;

Berbeda dengan variabel, suatu konstanta bernama tidak dapat diubah setelah didefinisikan. Menurut tradisi konstanta bernama ditulis dengan huruf kapital. Pada pendefinisian kontanta bernama yang bertipe *int*, kata kunci int boleh tidak ditulis,
misal : const int MAX = 20; dapat ditulis conts MAX = 20;

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
 const float PHI = 3.141592;
 float a=2*PHI;
 clrscr;
 cout <<"Nilai 2 PHI = " <<a;
}
```

Hasil :

```
Nilai 2 PHI = 6.28318
```

TUGAS

1. Buat program untuk menampilkan tulisan sebagai berikut :
   “Anda diundang rapat hari Jum’at / pukul 14.00 WIB”

2. Buat program untuk mencari luas segitiga jika diketahui alas=10 dan tinggi=20
3. Buat program dengan konstanta untuk mencari luas dan keliling lingkaran jika diketahui jari-jari lingkaran=100