

PENGENALAN BAHASA C++

Mira Suryani, S.Pd., M.Kom

S-1 Teknik Informatika
Jatinangor, 20 September 2018



From West Java for Indonesia to the World through SDGs

www.unpad.ac.id



Tujuan

- Mahasiswa akan dapat menjelaskan mengenai struktur bahasa C++ yang digunakan dan tata bahasanya
- Mahasiswa memahami hal-hal yang mendasari Bahasa C++ seperti operator dan sintaks dasarnya.
- Mahasiswa mengerti bagaimana membuat program sederhana dengan bahasa C++
- Mahasiswa memahami alur runtunan dan bisa menggunakan dasar pemrograman untuk operasi input dan output



Pokok Bahasan

Pada bagian ini akan dibahas topik-topik tentang:

- Bahasa C dan C++
- Struktur program
- Keyword
- Tipe data
- Operator
- Alur pemrograman dengan pola:

Input-Proses-Output



Kelebihan Bahasa C dan C++

- Bahasa C dan C++ tersedia hampir di semua jenis computer (PC, minikomputer dan mainframe).
- Bahasa C dan C++ hanya menyediakan **sedikit kata-kata kunci**
- Kode bahasa C dan C++ sifatnya adalah **portable dan fleksibel** untuk semua jenis komputer.
- Aplikasi yang ditulis dengan bahasa C untuk suatu komputer tertentu **dapat digunakan di komputer** lain hanya dengan sedikit modifikasi.
- Proses **executable program** bahasa C dan C++ **lebih cepat**
- Dukungan **pustaka** yang **banyak**.
- C dan C++ adalah **bahasa yang terstruktur**.
- C juga mengandung **feature-feature low-level** yang secara normal hanya tersedia pada Assembly atau bahasa mesin
- Program yang ditulis dengan C dikompilasi menjadi program yang kecil dan dieksekusi dengan efisien
- C++ Sudah mendukung **OOP** (*Object Oriented Programming*).
- C++ dapat membuat **aplikasi graphic processor** berkualitas tinggi.



Kekurangan Bahasa C dan C++

- **Banyaknya Operator** serta **fleksibilitas penulisan** program kadang-kadang **membingungkan** pemakai (Pemula).
- Bagi pemula pada umumnya akan **kesulitan menggunakan pointer** dan penerapan konsep OOP.



Susunan penulisan program dengan C++

```
/* Nama program      :  
   Nama              : Mira  
   NPM               :  
   Tanggal buat     :  
   Deskripsi         :  
*****/  
// deklarasi header file / Preprocessor directive  
// deklarasi fungsi / void  
  
main() {  
    /* KAMUS / DEKLARASI*/  
    /* ALGORITMA / DESKRIPSI */  
}
```



Komentar

- Komentar adalah catatan yang ditulis pada kode dengan tujuan sebagai bahan dokumentasi.
- Teks tersebut bukan bagian dari program dan tidak mempengaruhi jalannya program.

- Ada 2 jenis komentar dalam C++ yaitu:

- C++ style komentar satu baris (in line comment)

Awalan : // komentar

- C style beberapa baris (block comment)

Pasangan : /* komentar */



Preprocessor Directives

- *Preprocessor directives* diawali dengan `#include` dan diikuti dengan nama header file (library file)

- Contoh :

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
using namespace std; → untuk mengaktifkan penulisan standar yang ada  
pada header file
```




Contoh :

Pengenal identitas program dan pencetakan Hello Unpad

```
/* Nama program      : hello.cpp
   Nama              : Mira
   NPM              : 1234
   Tanggal buat     : 28 Februari 2017
   Deskripsi        : Pencetakan Hello Unpad 2 kali
   ----- */
```

```
#include <iostream.h>
```

```
using namespace std;
```

```
main() {
```

```
    cout<<"Hello, Unpad." <<endl;
```

```
    cout<<"Hello, " << "Unpad";
```

```
}
```



Identifier (Pengenalan)

- C++ Identifier adalah suatu tanda yang mewakili nama-nama variabel, fungsi, method, class, dsb.
- Contoh dari Identifier adalah : nilai, factorial, Hello dll.
- Pendeklarasian C++ adalah **case-sensitive**.
- Hal ini berarti bahwa Identifier : **Hello** tidak sama dengan **hello**.
- Identifier harus dimulai dengan salah satu huruf atau underscore “_”, dan tidak diawali dengan angka.



Keyword (Kata kunci)

- Kata yang bermakna khusus yang tidak bisa digunakan sebagai pengenal.
- Ada 48 buah keyword dalam Bahasa C++ yaitu:

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



Type Data

- Data adalah fakta atau kenyataan yang tercatat mengenai suatu objek yang direpresentasikan dalam bentuk tulisan, suara, gambar, sinyal atau simbol.
- Pengertian data ini menyiratkan suatu nilai yang bisa dinyatakan dalam bentuk konstanta atau variabel sehingga dikatakan tipe data adalah macam / isi data di dalam suatu variabel dalam bahasa program



Tipe primitive (tipe data dasar)

Tipe data	Byte	Batasan
char	1	Bilangan bulat / ASCII antara -128 s.d. 127
unsigned char	1	Bilangan bulat antara 0 s.d. 255
short	2	Bilangan bulat antara -32.768 s.d. 32.767 (-2^{15} s.d. $2^{15}-1$)
unsigned short	2	Bilangan bulat antara 0 s.d. 65.535 (0 s.d. $2^{16}-1$)
int	4	Bilangan bulat antara -2.147.483.648 s.d. 2.147.483.647 (-2^{31} s.d. $2^{31}-1$)
unsigned int / unsigned	4	Bilangan bulat antara 0 s.d. $2^{32}-1$
long int	4	Bilangan bulat antara -2^{31} s.d. $2^{31}-1$
unsigned long int	4	Bilangan bulat antara 0 s.d. $2^{32}-1$
float	4	Bilangan real antara - 3.4 E+38 s.d. 3.4E+38 (7 digit presisi)
double	8	Bilangan real antara -1.7E+308 s.d. 1.7E+308 (15 digit presisi)
bool	1	true / false → Tidak semua compiler yang support

```
#include <limits.h>
```

```
cout << "Bilangan minimum char: "<<CHAR_MIN<<endl; //??  
cout << "Bilangan minimum char: "<<UCHAR_MAX<<endl; //??  
cout << "Bilangan maximum int : "<<INT_MAX; //??
```



Variabel

- **Variabel** adalah item yang digunakan data untuk menyimpan pernyataan objek. Variabel memiliki **tipe data** dan **nama**.
- Tipe data menandakan tipe nilai yang dapat dibentuk oleh variabel itu sendiri.
- **Nama variabel** harus mengikuti aturan untuk identifier.

Deklarasi : tipe namaVar1, namaVar2 [=nilaiAwal]

contoh :

1. Variable karakter : char var1=10, var2, var3;

2. Variable string : char kata[10]; /* panjang karakter maksimum = 10

3. Variable integer, float

```
int bulat=5; /* berisi bilangan bulat*/
```

```
float bil_real=0.5;
```

Perhatikan :

```
int x=y=5 ; // ERROR
```

seharusnya : int x=5, y=5; tetapi :

```
int x,y;
```

```
x=y=5; // DIBOLEHKAN disebut CHAINED ASSIGNMENT
```



Petunjuk Membuat nama variabel

1. Sangat baik untuk menginisialisasi variabel yang dibuat
2. Gunakan nama yang bersifat menggambarkan deskriptif untuk variabel yang dibuat,
*Misalkan jika ingin mempunyai variabel yang terdiri atas nilai siswa, beri nama dengan nama **grade / nilai** dan jangan hanya beberapa huruf random*

Contoh :

```
double nilai=0.0;
```

```
int num = 10;
```

```
//jangan double a=0.0;
```

contoh : deklarasi variable dan pencetakannya

```
main() {  
    int num;  
    num=10;  
    cout << num << endl;  
}
```



Konstanta

```
const type nama = nilai_konst;
```

Atau

```
#define nama nilai_konst
```

Contoh :

```
const double phi=3.1415;
```

Jenis Konstanta	Contoh
Konstanta desimal	x = 10;
Konstanta hexadesimal	x = 0x1B; x = 0x10;
Konstanta Octal	x = 015;
Konstanta Float	x = 20e6; x = 6.5536E-6;
Konstanta Long	x = 30000L; x = 30000l;
Konstanta karakter	x = 'c';



Konstanta Karakter dengan Backslash '\'

Kode	Karakter	ASCII
\a	Bell	07
\b	Back space	08
\f	Form feed	12
\n	Baris baru	10 dan 13
\r	Carriage Return	13
\0	NULL	00
\\	Backslash	92
\v	Vertical Tab	11
\"	Tanda kutip dua	34
\'	Tanda kutip satu	44
\o	Konstanta Octal	- -
\x	Konstanta Hexadesimal	- -



Contoh : Penulisan angka decimal, octal dan heksadesimal

```
//-----  
  
main{  
    cout << 17 << endl;           // hasil = ?  
    cout << 017 << endl;         //??  
    cout << 0x17;                 //??  
}  
  
//-----
```

Contoh : penulisan string dan penggunaan string.h

```
// Berkas: LiteralString.Cpp  
#include <string.h>  
main(){  
    cout << "abc \n def"<< endl;           // hasil = ??  
    cout << "abc \t def"<< endl;           //??  
    cout << "\"Halo\""<< endl;             //??  
    cout << "Panjang kata Unpad ="<< strlen("Unpad"); //??  
}
```



Operator Aritmatika & Increment / Decrement

Simbol	Fungsi	Contoh Penggunaan
-	Pengurangan	$x = x - 10;$
+	Penjumlahan	$x = x + 10;$
/	Pembagian	$x = x / 10;$
*	Perkalian	$x = x * 10;$
%	Modulo	$x = 11 \% 2;$

```
cout<<"13 % 5 = "<<(13%5)<<endl; // ??  
cout<<"13 / 5 = "<<(13/5 )<<endl; // ??  
cout<<"13.0/5 ="<<(13.0/5); //??
```

Simbol	Fungsi	Contoh Penggunaan
++	Increment	$x++$ artinya $x=x+1$ $++x$ artinya $x=x+1$
--	Decrement	$x--$ artinya $x=x-1$ $--x$ artinya $x=x-1$



Contoh : Increment

```
int m=66;  
m++;  
cout << m;                // ??
```

Contoh : Pre Increment

```
int m=66, n;  
n= ++m;  
cout << "m = " << m << " n = " << n ; // 67, 67
```

Perhatikan :

```
m=m+1                //m=67  
n=m                  //n=67
```

Contoh : Post Increment

```
int m=66, n;  
n= m++;  
cout << "m = " << m << " n = " << n ; // 66, 67
```

Perhatikan :

```
n=m;                //n=66  
m=m+1              //m=67
```



Operator Assignment

Simbol	Keterangan
-=	x = x - y dapat ditulis x -= y
+=	x = x + y dapat ditulis x += y
/=	x = x / y dapat ditulis x /= y
*=	x = x * y dapat ditulis x *= y
%=	x = x % y dapat ditulis x %= y

```
Contoh :
int nilai=50;
nilai += 20;
cout << "nilai = "<<nilai;           // hasil nilai=70

Contoh :
char c='A';           //atau c=65 dan pakai ` bukan `
cout << c++ <<" "<<<int(c)<<endl;    // hasil `A` dan 65
cout << c++ <<" "<<<int(c);         // hasil `B` dan 66

Perhatikan setelah dicetak lalu increment
```



Operator Logika

Simbol	Keterangan
&&	AND
	OR
!	Not

```
Contoh :   int nilai = 70;
           if (nilai>=60 && nilai <=100){
               cout<<"Lulus";
           }
```



Operator Bitwise

Simbol	Keterangan
&	AND
	OR
^	XOR
~	Complement
>>	Shift Right (geser kanan)
<<	Shift Left (geser kiri)

Contoh :

```
int a = 13;           // biner : 1101
int b = 6;           // biner : 0110
cout<<"a & b =" << (a&b); // biner : 0100 = 4
```

X	Y	AND (&)	OR ()	XOR (^)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



Operator Shift

Bentuk umum dari operator shift :

variabel >> nomor posisi bit

variabel << nomor posisi bit

Contoh :

```
int a = 13; // biner : 1101
cout<<"a << 1 ="<<(a<<1); // biner : 11010 = 26
```




Type casting (Pengarah type)

Deklarasi

```
typeData ( var);
```

- Perubahan tipe ke yang lebih besar ukurannya, jika ke yang lebih kecil akan terjadi pemotongan nilai.

Contoh :

```
int jumlah=10, nData=3;
```

```
float rata;
```

```
cout<<(jumlah / nData);           // hasil = 3
```

```
rata= float (jumlah) / nData;     // hasil = 3.333 → hasil bagi dikenakan casting dari int ke float
```

```
cout<<rata;
```



I/O stream manipulator

Manipulator	Keterangan
dec	Mengkonversi menjadi bilangan desimal
hex	Mengkonversi menjadi bilangan hexadesimal
oct	Mengkonversi menjadi bilangan octal
endl	Baris baru atau sama saja dengan \n dan flush stream
ends	Menyisipkan null zero kedalam string
flush	flush output stream
setbase(int n)	Mengkonversi ke base n (0 untuk desimal default, 8 untuk octal , 16 untuk hexadesimal)
resetiosflags(long f)	Menghilangkan format dengan format flag yang terdapat pada tabel format flag)
setiosflags(long f)	setting format sesuai dengan tabel format flag.
setfill(int c)	setting isi dengan karakter
setprecision(int n)	setting floating point dengan n presisi
setw(int n)	setting lebar field



Format flag untuk resetiosflags & setiosflags

Nama format flag	Keterangan
<code>ios::left</code>	Output rata kiri dengan <code>setw()</code>
<code>ios::right</code>	Output rata kanan dengan <code>setw()</code>
<code>ios::scientific</code>	Format output dengan notasi scientific
<code>ios::fixed</code>	Format angka dengan desimal format
<code>ios::dec</code>	Format menjadi bilangan desimal
<code>ios::hex</code>	Format menjadi bilangan hexadesimal
<code>ios::oct</code>	Format menjadi bilangan oktal
<code>ios::uppercase</code>	Merubah semua hexadesimal dan notasi scientific menjadi huruf besar (contoh 1.22E+03)
<code>ios::showbase</code>	Mencetak base prefix (0x untuk hexadesimal atau 0 untuk octal)
<code>ios::showpoint</code>	Mencetak titik untuk menunjukkan presisi



Contoh Menampilkan Biner

```
#include <iostream>
```

```
#include <bitset>
```

```
using namespace std;
```

```
main(){
```

```
    int a =10;
```

```
    cout << "a = " << bitset<6>(a) << endl;
```

```
}
```



Contoh : Penggunaan I/O Manipulator untuk format output

```
#include <iostream.h>
int main(){
    double pi = 3.141592654;
    // Tampilan default: left justified, presisi 6.
    cout << pi << endl;

    // Ubah dg precision 4, lebar field 12, isi dg #
    cout.precision(4);
    cout.width(12);
    cout.fill('#');
    cout << pi << endl;
    // Ubah presisi ke 10
    cout.precision(10);
    cout << pi << endl;
    return 0;
}
```



Contoh : iomanip

```
#include <iostream.h> // basic_iostream manipulation untuk format I/O
#include <iomanip.h> // formatting manipulator
#include <conio.h> // console I/O
main(){
    int num = 37;
    double pi = 3.141592654;
    cout << "hex: " << hex << num << endl;
    cout << "oct: " << oct << num << endl;
    cout << "dec: " << dec << num << endl;
    cout << setw(8) << num <<endl;
    cout << setw(8) << setfill(' ') << num << endl;
    cout << "Pi: " << setprecision(10) << pi << endl;
    getch();
}
```



Input (Masukan) dan Output (Keluaran)

- Contoh : Menggunakan style bahasa C (dengan scanf dan printf)

```
/* Nama program          : tulisInt.c
   Nama                  : Mira Suryani
   NPM                   : 007
   Tanggal buat          : 28/02/2017
   Deskripsi              : Contoh membaca dan menulis nilai
                           integer dgn bahasa C
   *****/
#include <stdio.h>
main() {
    /* KAMUS */
    int a;
    /* ALGORITMA */
    printf("Contoh baca dan tulis, ketik integer :");
    scanf("%d",&a);
    printf("Nilai yang dibaca : %d \n",a);
    return 0;
}
```



Contoh : Menggunakan style C++ (dengan cin dan cout)

```
//Program Baca.cpp
// Deskripsi   : Contoh membaca dan menulis nilai integer
//              dengan bahasa C++
//*****
# include <iostream.h>
main() {
    // KAMUS
        int a;
    // ALGORITMA
        cout << "Contoh baca dan tulis, ketik integer :";
        cin >>a;
        cout << "Nilai yang dibaca : " << a;
}
```




**ANY
QUESTIONS?**



Sesi Berakhir
TERIMA KASIH
