

Keg. Pembelajaran 2 : Mengetahui Bahasa C++

1. Tujuan Kegiatan Pembelajaran

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Mengetahui secara baik sintak-sintak dalam bahasa pemrograman C++.
- 2) Menjelaskan struktur program dalam bahasa C++ dengan baik.
- 3) Memahami konstanta dan variable dalam bahasa C++ dengan baik
- 4) Membuat program sederhana dengan bahasa C++ secara benar
- 5) Menggunakan operasi *cout* dan *cin* dengan benar

2. Uraian Materi

a. Mengetahui Bahasa C++

Setelah kita mengetahui sejarah perkembangan bahasa C++, kini kita akan mengetahui lebih jauh tentang apa dan bagaimana bahasa C++. Bentuk Umum bahasa C++ adalah sebagai berikut.

```
# preprocessor directive
void main ()
{
    // Batang Tubuh Program utama
}
```

Gambar 4: Bentuk Umum bahasa C++

Penjelasan :

include Adalah salah satu pengarah *preprocessor directive* yang tersedia pada C++. Preprocessor selalu dijalankan terlebih dahulu pada saat proses kompilasi terjadi. Bentuk umumnya : **# include** <nama_library>, **tidak diakhiri** dengan tanda semicolon, karena bentuk tersebut bukanlah suatu

bentuk pernyataan, tetapi merupakan preprocessor directive. Baris tersebut menginstruksikan kepada kompiler yang menyisipkan file lain, dalam hal ini file yang berakhiran **.h** (file header) yaitu file yang berisi daftar pustaka sebagai deklarasi. Berikut contoh file header al:

- # include <iostream.h> : diperlukan pada program yang melibatkan objek cout
- # include <conio.h> : diperlukan bila melibatkan clrscr(), yaitu perintah untuk membersihkan layar.
- # include <iomanip.h> : diperlukan bila melibatkan setw() yang bermanfaat untuk mengatur lebar dari suatu tampilan data.
- # include <math.h> : diperlukan pada program yang menggunakan operasi sqrt () yang bermanfaat untuk operasi matematika kuadrat.
- **Contoh :**

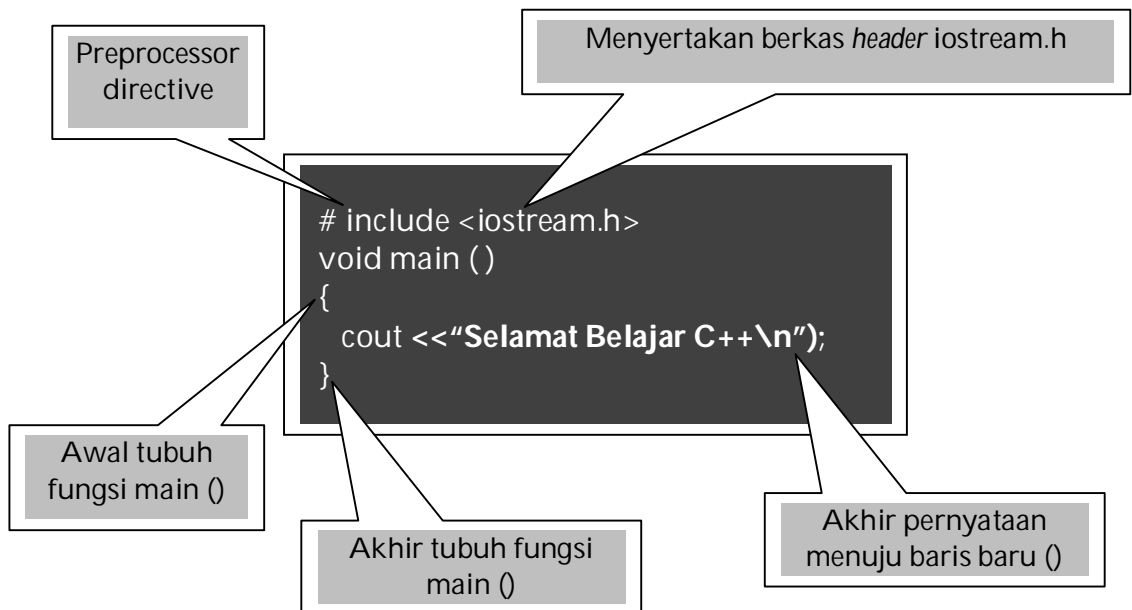
```
# include <iostream.h>
void main ( )
{
    cout << "Selamat Belajar C++\n");
}
```

Gambar 5: Tampilan tulisan C++

Program di atas berfungsi untuk menampilkan tulisan :

"Selamat Belajar C++ " di layar.

Berikut di bawah ini dijelaskan bagian per bagian program hasil eksekusi program tampilan sebagaimana terlihat pada Gambar 4 di atas. Adapun penjelasannya secara rinci perhatikan Gambar 6 berikut di bawah ini.



Gamabr 6. Komponen program dalam C++

- ✚ Berkas **header** (berekstensi.h) adalah berkas yang berisi prototype fungsi, definisi konstanta, dan definisi variable. Prototype fungsi berupa judul suatu fungsi yang dilengkapi dengan argumen dan tipe argumen serta tipe nilai balik untuk fungsi yang memberikan nilai balik. Prototype fungsi harus sudah tersedia sebelum pemanggilan fungsi. Ini merupakan persyaratan selama kompilasi. Fungsi adalah kumpulan kode C yang diberi nama dan ketika nama tersebut dipanggil maka kumpulan kode tersebut dijalankan.
- ✚ **Praprosesor** `#include` adalah suatu perintah yang digunakan untuk mengatur kompiler agar membaca berkas *header* yang disertakan di belakang kata **include** saat pelaksanaan kompilasi.
- ✚ **void main ()** adalah fungsi yang akan dijalankan pertama kali ketika program dieksekusi. `Void main ()` menyatakan bahwa program tidak memberikan nilai balik.

- ✚ Pernyataan di dalam C++ adalah suatu perintah yang dapat berdiri sendiri dalam melaksanakan suatu aksi dan dijalankan ketika program dieksekusi. Setiap pernyataan ditulis dengan diakhiri tanda **titik koma** (;). Pada program di atas, ada satu buah pernyataan, yaitu : **cout** <<"Selamat Belajar C\n";
- ✚ **cout** adalah suatu fungsi yang prototipenya terdapat dalam berkas **iostream.h** Yang berada dalam tanda kurung yang mengikuti **cout** pada program di atas disebut string. String adalah deretan karakter yang ditulis di dalam tanda petik ganda (").
- ✚ \n yang diikuti dengan tanda ; merupakan akhir pernyataan, sekaligus sebagai perintah untuk pindah/ ganti ke baris baru berikutnya.

b. Fungsi main ()

Fungsi ini menjadi awal dan akhir eksekusi program C++. **main** adalah nama judul fungsi. Melihat bentuk seperti itu dapat kita ambil kesimpulan bahwa batang tubuh program utama berada didalam fungsi main(). Berarti dalam setiap pembuatan program utama, maka dapat dipastikan seorang pemrogram menggunakan minimal sebuah fungsi. Pembahasan lebih lanjut mengenai fungsi akan diterangkan kemudian.

Program C++ memang tidak akan pernah lepas dari suatu fungsi/function. Hal ini karena merupakan ciri OOP. Sebuah program C++ minimal memiliki satu fungsi yaitu main(). Fungsi ini merupakan awal program utama. Tulisan main() merupakan nama fungsi, sedangkan bagian yang diapit dengan { dan } disebut blok (tubuh fungsi). Dalam hal ini { merupakan tanda awal blok dan } adalah tanda akhir blok. Seperti halnya dalam Pascal, { dalam Pascal identik dengan BEGIN, sedangkan } identik dengan END. Perintah void bermakna bahwa fungsi main() tidak

mengembalikan nilai/value. Cara penulisan fungsi main() tidak mutlak seperti di atas.

Berikut ini cara penulisan yang lain :

```
#include <iostream.h>
int main()
{
    cout << "Hello world.\n";
    return 0;
}
/* awal program*/
/*tubuh program */
/* akhir program*/
```

c. Komentar

Komentar tidak pernah dicompile oleh compiler. Dalam C++ terdapat 2 jenis komentar, yaitu:

Jenis 1 : /* Komentar anda diletakkan di dalam ini, bisa mengapit lebih dari satu baris */

Jenis 2 : // Komentar anda diletakkan disini (hanya bisa perbaris)

d. Tanda Semicolon

Tanda semicolon " ; " digunakan untuk mengakhiri sebuah pernyataan. Setiap pernyataan harus diakhiri dengan sebuah tanda semicolon.

e. Mengenal cout dan cin (di baca : C out dan C in)

Pernyataan **cout** merupakan sebuah objek di dalam C++, yang digunakan untuk mengarahkan data ke dalam standar output (cetak pada layar)

1) Operator Keluaran cout

Operator keluaran "*cout*" adalah standar output yang ditampilkan di layar, yang diikuti operator penyisipan (<<) dan *newline* ('\n' atau dengan *endl*).

Bentuk umum penulisannya sebagai berikut :

```
cout << " String \n";
```

Jika untuk menampilkan Tulisan "tanggal : " diikuti dengan isi Variabel tgl dan karakter *newline*, pernyataanya menjadi:

```
cout << " Tanggal : " << tgl << ' \n';
```

2) Operator penyisipan cin

Operator masukan cin digunakan untuk membaca data dari standar input (keyboard), dan digunakan meletakkan ke sebuah variabel bernama variabel Var.

Bentuknya :

```
cin >> var
```

Catatan :

" >> " merupakan operator "pengambilan dari "

Contoh :

```
// contoh pengambilan data dengan cin (keyboard)*
// membaca data bertipe int, dan float dari keyboard*
# include <iostream.h>
# include <conio.h>
void main ()
{
    int bil_x;          // def bil bulat
    float bil_y;       // def bil pecahan
    cout << " Masukkan bilangan bulat : ";
```

```

cin >> bil_x;
cout << " masukkan bilangan pecahan : ";
cin >> bil_y;
clrscr();

cout << " bilangan bulat    : " << bil_x << '\n';
cout << " bilangan pecahan : " << bil_y << '\n';
}

```

f. Identifier

Identifier merupakan suatu pengenal atau pengidentifikasi yang kita deklarasikan agar kompiler dapat mengenalinya. Identifier sendiri dapat berupa variabel, konstanta, fungsi, kelas, template maupun namespace.

Identifier yang berfungsi sebagai variabel dan konstanta berfungsi untuk menampung sebuah nilai yang digunakan dalam program. Identifikasi ini dilakukan untuk mempermudah proses penanganan data atau nilai, misalnya untuk memasukkan dan menampilkan nilai.

```

#include <iostream>
void main ( )
{
    char Teks[100];
    int X;
    cout<<"Masukkan sebuah kata   : "; cin>>Teks;
    cout<<"Masukkan sebuah angka : ";   cin>>X;
    cout<<Teks<<endl; //bisa ditulis dengan cout<<X<<'\n';
    cout<<X;
}

```

Gambar 7. Identifier program dalam C++

Hal-hal yang perlu diperhatikan dalam membuat identifier antara lain sebagai berikut:

- 1) Bersifat case sensitive, maka dalam C++ akan membedakan variabel yang ditulis dengan huruf besar dan huruf kecil.
- 2) Identifier tidak boleh berupa angka atau diawali dengan karakter yang berupa angka misalnya :

`long 1000; // SALAH karena identifier berupa angka`

`long 2X; // SALAH karena identifier diawali oleh karakter angka`

`long X2 // BENAR karena identifier tidak diawali oleh angka`

- 3) Identifier tidak boleh mengandung spasi
- 4) Identifier tidak boleh menggunakan karakter-karakter simbol (#,@,?,!,\$, dll)
- 5) Identifier tidak boleh menggunakan kata kunci yang terdapat pada C++.(Misal break,return dll)
- 6) Nama identifier sebaiknya disesuaikan dengan kebutuhannya
- 7) Hindarilah penggunaan nama identifier yang sama dengan identifier yang digunakan oleh C++.

g. Konstanta

Konstanta adalah jenis identifier yang bersifat konstan atau tetap, artinya nilai dari konstanta di dalam program tidak dapat diubah. Ada dua jenis konstanta yang kita kenal, yaitu :

- 1) Konstanta String

String adalah sederetan karakter yang diawali dan diakhir dengan petik ganda (".....")

Contoh : "Selamat Hari Raya"

" a "

↓
Konstanta string

→ beda dengan 'a'

↓
konst char

2) Konstanta Bernama

Sesuatu yang nilainya tetap. Bentuk umum konstanta dapat didefinisikan sebagai berikut :

```
const tipe_data nama_konstanta = nilai
```

Misalnya : `const float Phi = 3.141592`

h. Variabel

Variabel adalah identifier yang mempunyai nilai dinamis, artinya bahwa nilai variabel tersebut dapat kita sesuaikan dengan kebutuhan kita dalam program. Variabel digunakan untuk menyimpan suatu nilai. Nilai ini dapat diubah-ubah selama eksekusi program berlangsung. Dalam program Variabel harus di deklarasikan, yaitu :

- 1) Mengenalkan sebuah pengenalan ke program
- 2) Menentukan jenis /type data yang akan disimpan di variabel

Bentuk pendefinisian variable adalah sebagai berikut :

```
Tipe_data Daftar_Variabel;
```

Daftar : berarti bisa lebih dari satu variabel.

Contoh :

```
int Jumlah;  
float harga_per_unit, total_harga;
```

i. Tipe Data

Tipe data berfungsi untuk mempresentasikan jenis dari sebuah nilai yang terdapat dalam program. Dalam bahasa C++, tipe data dibagi menjadi tiga bagian dasar yaitu tipe dasar, tipe bentukan dan tipe enumerasi.

1) Tipe Data Dasar

Tipe dasar ini digolongkan ke dalam tipe bilangan bulat (integer), bilangan riil (floating-point), tipe logika (boolean), tipe karakter/teks (character/string).

a) Tipe Bilangan Bulat

Tipe ini digunakan untuk data-data angka yang tidak mengandung angka di belakang koma. Tipe data yang termasuk ke dalam kategori ini adalah seperti yang terlihat pada tabel dibawah ini.

Tabel 3. Tipe data bilangan bulat

Tipe Data	Ukuran (dalam bit)	Rentang
Int	16 atau 32	-32.768 sampai 32.767 atau -2,147,483,648 sampai 2,147,483,647
unsigned int	16 atau 32	0 sampai 65.535 atau 0 sampai 4,294,967,295
signed int	16 atau 32	Sama seperti int
short int	16	-32.768 sampai 32.767
unsigned short int	16	0 sampai 65.535
signed short int	16	Sama seperti short in
long int	32	-2.147.483.648 sampai 2.147.483.648
signed long int	32	Sama seperti long int
unsigned long int	32	0 sampai 4,294,967,295

Contoh program yang menggunakan tipe bilangan bulat adalah sebagai berikut:

```

#include <iostream>
void main()
{
    int x;           // Mendeklarasikan variabel x dengan tipe data int
    x = 3;           // Melakukan assignment terhadap variabel x
    cout<<"Nilai x = "<<x;
}

```

Hasil yang diperoleh dari program diatas adalah sebagai berikut: Nilai x = 3

Gambar 8. Tampilan program bil bulat

b) Tipe Bilangan Riil

Tipe ini adalah tipe yang merepresentasikan data-data bilangan yang mengandung angka di belakang koma. Adapun tipe data yang termasuk dalam kategori ini adalah seperti yang ditunjukkan dalam tabel berikut:

Tabel 4. Tipe data Riil (Pecahan)

Tipe Data	Ukuran (dlm bit)	Rentang	Presisi
Float	32	1.2E-38 sampai 3.4E+38	6 digit presisi
Double	64	2.3E-308 sampai 1.7E+308	15 digit presisi
long double	80	3.4E-4932 to 1.1E+4932	19 digit presisi

Contoh program yang menggunakan tipe bilangan bulat adalah sebagai berikut:

```

#include <iostream>
void main()
{
    double y;       // Mendeklarasikan variabel y dengan tipe data double
    y = 222.134;    // Melakukan assignment terhadap variabel x
    cout<<"Nilai y = "<<x;
}

```

Hasil yang diperoleh dari program di atas adalah: Nilai y = 222.134

Gambar 9 . Tampilan Program bil bulat

c) Tipe Logika

Tipe ini merepresentasikan data-data yang hanya mengandung dua buah nilai, yaitu nilai logika (boolean). Nilai logika itu sendiri hanya terdiri dari nilai benar (1) dan salah (0). Dalam bahasa pemrograman, nilai ini umumnya lebih dikenal dengan nilai true (benar) dan false (salah).

d) Tipe String

Tipe ini merepresentasikan data-data yang berupa karakter. Tipe ini dinyatakan dengan tipe char, sedangkan untuk string (=kumpulan karakter) dinyatakan sebagai pointer dari tipe char, yaitu ditulis dengan char*. Dalam C++, tipe karakter diapit oleh tanda petik tunggal ('), sedangkan tipe string diapit oleh tanda petik ganda ("). Adapun tipe data dalam kategori tipe ini adalah seperti berikut:

Tabel 5. Tipe data String

Tipe Data	Ukuran (dlm bit)	Rentang
char	8	-128 sampai 127 atau 0 sampai 255
unsigned char	8	0 sampai 255
signed char	8	-128 sampai 127

Berikuti ini adalah program yang menunjukkan penggunaan tipe data char dan char*.

```

#include <iostream>
void main()
{
    char Karakter = 'Teknik Elektro'; // Mendeklarasikan variabel Karakter dengan nilai 'A'
    char* Teks = "FT"; // Mendeklarasikan variabel Teks dengan nilai "Kata"
    char TEKS[10] = "UNY";
    cout<<Karakter<<endl;
    cout<<Teks<<endl;
    cout<<TEKS<<endl;
    return 0;
}

```

Hasilnya :
 Teknik Elektro
 FT
 UNY

Gambar 10. Penggunaan tipe char.

e) Tipe Data Bentukan

Tipe data bentukan yaitu tipe data yang dibuat sendiri sesuai kebutuhan dalam program yang akan kita buat. Dalam bahasa latin sering disebut *user defined types*. Adapun yang termasuk dalam kategori tipe bentukan ini adalah array (larik), struktur dan enumerasi.

(1) Struktur

Yaitu tipe data bentukan yang menyimpan lebih dari satu variabel bertipe sama maupun berbeda. Berikut bentuk umum pendeklerasian tipe data struktur dalam bahasa C++.

(2) Enumerasi

Yaitu tipe data yang nilainya terbatas pada nilai-nilai yang telah didefinisikan saja. Tipe ini digunakan untuk membentuk tipe data yang nilainya bersifat pasti. Berikut bentuk umum pendeklerasian tipe data enumerasi dalam bahasa C++.

```
Enum nama_file { nilai_1, Nilai_2 }
```

sebagai contoh, di bawah ini dituliskan contoh pendefinisian tipe enumerasi

```
enum HARI { Minggu, Senin, Selasa, Rabu, Kamis, Jum'at, Sabtu};  
enum JENIS_KELAMIN { Pria, wanita};
```

3. Rangkuman

Sebagaimana pada bahasa pemrograman yang lain, Bahasa C++ memiliki bentuk umum struktur pemrograman, meliputi kepala program, tubuh program dan bagian akhir program. C++ memiliki bentuk khusus terkait dengan kepala program. Dalam hal ini melibatkan apa yang dinamakan pengarah *preprocessor directive*. Preprocessor selalu dijalankan terlebih dahulu pada saat proses kompilasi terjadi. Baris tersebut menginstruksikan kepada kompiler yang menyisipkan file lain dalam hal ini file yang berakhiran .h (file header) yaitu file yang berisi daftar pustaka sebagai deklarasi antara lain : #include <iostream.h> ; #include <conio.h>; #include <iomanip.h> ; #include <math.h>, dan sebagainya.

Dalam penyusunan program akan dikenalkan apa yang disebut dengan konstanta dan variable. Konstanta memiliki nilai yang tetap, sebaliknya variable adalah parameter yang memiliki sifat dinamis di saat program sedang dijalankan. Di samping itu, ada yang disebut dengan tipe data. Tipe data dalam pemrograman harus benar-benar diperhatikan, karena akan menentukan proses eksekusi program. Secara umum tipe data dalam pemrograman terdiri dari integer, float, character, string, dan tipe data turunannya.

4. Latihan

- 1) Yang membedakan prinsip penterjemahan antara kompilasi dengan interpreter dalam pemrograman komputer adalah , *kecualai* ;
 - a. Cara pembacaan program
 - b. Tingkat jaminan kerahasiaan program
 - c. Kecepatan eksekusi program
 - d. Penterjemahan kode program .
- 2) Konstanta merupakan suatu nilai yang tidak dapat diubah selama proses program berlangsung. Berikut adalah definisi konstanta , *kecuali*:
 - a. Konstanta nilainya selalu tetap.
 - b. Konstanta harus didefinisikan terlebih dahulu di awal program.
 - c. Konstanta selalu beisi bilangan atau angka.
 - d. Konstanta dapat bernilai integer, pecahan, karakter dan string.
- 3) Pernyataan yang paling tepat adalah :
 - a. Fungsi "void" di depan main() merupakan hal yang wajib harus ada pada C++, jika fungsi ini tidak memberikan nilai balik
 - b. Fungsi "void" di depan main() merupakan hal yang bersifat pilihan saja (option) pada C++, jika fungsi ini memberikan nilai balik
 - c. Fungsi "void" di depan main() merupakan hal yang bersifat option pada C++, jika fungsi ini tidak memberikan nilai balik
 - d. Fungsi "void" di depan main() merupakan hal yang tidak wajib ada, selama fungsi ini tidak memberikan nilai balik
- 4) Pernyataan berikut ini benar adanya, *kecuali*:
 - a. tanda << sebagai operator penyisipan selalu berpasangan dengan cout
 - b. tanda >> sebagai operator pengambilan berpasangan dengan cin
 - c. tanda << selalu harus berpasangan dengan dua tanda petik ganda (" ")
 - d. cout untuk menampilkan data ke standart output (mis: layar)
- 5) Pada C++ disediakan file library, manakah penggunaan library yang paling tepat bentuk pernyataan ini :
 - a. # include <iostream.h> dan # include<conio.h> utk fungsi cout dan setfill()
 - b. # include <iomanip.h> dan # include<math.h> untuk fungsi setw() dan cin
 - c. # include <iostream.h> dan # include<iomanip.h> untuk fungsi cout dan cin
 - d. # include <iostream.h> dan # include<conio.h> utk fungsi cout, cin, clrscr()