

Pengenalan C++

Bab 1

1.1 Sejarah C++

Bahasa C++ diciptakan oleh Bjarne Stroustrup di AT&T Bell Laboratories awal tahun 1980-an berdasarkan C ANSI (American National Standard Institute). Pertama kali, prototype C++ muncul sebagai C yang diperceangkih dengan fasilitas kelas. Bahasa tersebut disebut C dengan kelas (C with class). Selama tahun 1983-1984, C dengan kelas disempurnakan dengan menambahkan fasilitas pembebanlebih operator dan fungsi yang kemudian melahirkan apa yang disebut C++. Symbol ++ merupakan operator C untuk operasi kenaikan, muncul untuk menunjukkan bahwa bahasa baru ini merupakan versi yang lebih canggih dari C.

Borland International merilis compiler Borland C++ dan Turbo C++. Kedua compiler ini sama-sama dapat digunakan untuk mengkompilasi kode C++. Bedanya, Borland C++ selain dapat digunakan dibawah lingkungan DOS, juga dapat digunakan untuk pemrograman Windows.

Selain Borland International, beberapa perusahaan lain juga merilis compiler C++, seperti Topspeed C++ dan Zortech C++.

Contoh Program C :

```
# include <stdio.h>
Main ( )
{
Char pesan [ ] = "Hai, C programmers !" ;
Printf (pesan) ;
Return 0 ;
}
```

Contoh Program C++ :

```
# include <iostream.h>
Main ( )
{
Char pesan [ ] = "Hai, C programmers !" ;
Cout << pesan ;
Return 0 ;
}
```

1.2 TENTANG C++

C++ diciptakan untuk mendukung pemrograman berorientasi pada objek (Object Oriented Programming/OOP) yang tidak dimiliki C. sementara C merupakan bahasa pemrograman terbaik dilingkungannya, bahasa ini tidak

memiliki kemampuan OOP. Reputasi C tidak diragukan lagi dalam menghasilkan program .EXE berukuran kecil, eksekusi yang cepat, antarmuka (interfacing) yang sederhana dengan bahasa lain dan fleksibilitas pemrograman. Apa yang membuat C tampak sukar dipelajari mungkin karena tiadanya pemeriksaan tipe. Sebagai contoh, dapat mencampur bilangan bulat dengan string untuk menghasilkan karakter. Namun, justru dsitu letak fleksibilitas C, dapat mengolah data C sebebas mengolah data dalam bahasa assembly.

1.3 BORLAND C++

Dibandingkan compiler C++ yang lain, Borland C++ memiliki keunggulan terutama dalam hal kecepatan dan efisiensi kompilasi. Disamping itu, Borland C++ mendukung beberapa system operasi yaitu DOS, Windows 16bit (Window 3.0) dan windows 32 bit (Windows NT). Meskipun demikian compiler Borland C++ juga memiliki kelemahan bila dibandingkan compiler C++ yang lain, misalnya : pemrograman dengan Borland C++ terutama yang menyangkut tampilan jauh lebih sulit daripada pemrograman dengan Microsoft Visual C++.

1.4 STRUKTUR BAHASA C++

Program C maupun C++ selalu tersusun dari 4 (empat) bagian utama, yaitu :

1. Bagian komentar yang ditandai dengan symbol // dan pasangan /* ... */
2. Bagian pengarah compiler yang ditandai dengan symbol #
3. Bagian deklarasi
4. Bagian definisi

1.4.1 BAGIAN KOMENTAR

Program yang baik pada umumnya diberi komentar yang akan membantu orang lain maupun pembuat program itu untuk memahami program yang dibuat. Dalam C atau C++ setiap tulisan yang diapit oleh symbol /* ... */ atau setiap baris yang dimulai dengan symbol // dianggap komentar. C++ tidak mengizinkan komentar bersarang (nested comment), namun Borland C++ lebih fleksibel dalam hal ini.

Contoh C++ menggunakan komentar:

```
/* FIRST.CPP */
/* Program C++ pertamaku */
// Praproses
#include <iostream.h>
Void main ( )
{
Char pesan [ ] = "Hello, C++ programmers!" ;
```

```
Cout << pesan ;  
Return 0 ;  
}
```

Pada Borland C++ dapat menggunakan komentar bersarang asalkan opsi cek **Nested comments** pada menu **Options/Compiler/Source** dipilih.

1.4.2 BAGIAN PENGARAH KOMPILER

Contoh program C++ :

```
# include <iostream.h>  
Void main ( )  
{  
Char pesan [ ] = "Hello, C++ programmers!" ;  
Cout << pesan ;  
Return 0 ;  
}
```

Merupakan statement praprosesor, disebut juga pengarah compiler karena berfungsi mengatur proses kompilasi.

IOSTREAM.H merupakan file program yang mengandung deklarasi kelas-kelas yang diperlukan oleh objek **cout**. File-file dengan ekstensi .H yang berisi deklarasi fungsi-fungsi standar C ini, disebut secara umum sebagai file header.

Beberapa pengarah compiler adalah :

1. # define
2. # include
3. # if, # else, # elif, # endif
4. # ifdef, # ifndef

1.4.2.1 PENGARAH KOMPILER # DEFINE

Untuk mendefinisikan suatu pengenalan / konstanta yang nantinya akan digantikan oleh praprosesor saat program dikompilasi.

Contoh Program :

```
# define SIZE 30  
Int array [SIZE] ;  
For (register int i = 0 ; i < SIZE ; i++)  
{  
Cout << array [ i ] ;  
}
```

1.4.2.2 PENGARAH KOMPILER # INCLUDE

Berfungsi membaca file program tertentu dan mengikutsertakan file tersebut dalam proses kompilasi. Nama file yang dimaksud harus diapit symbol ' < ' dan ' > ' atau tanda kutip dua (" ... ").

1.4.2.3 PENGARAH KOMPILER # IF, # ELSE, # ELIF, # ENDIF

Digunakan untuk memilih bagian program yang akan dikompilasi. Kompilasi cari ini disebut kompilasi bersyarat dan program yang baik biasanya memanfaatkan teknik ini.

1.4.2.4 PENGARAH KOMPILER # IFDEF, # IFNDEF

Digunakan juga dalam kompilasi bersyarat. **# Ifdef** dapat dibaca : 'jika didefinisikan' dan **# ifndef** dapat dibaca : 'jika tidak didefinisikan'. Pengarah compiler ini sering digunakan untuk menandai bahwa suatu file sudah diikutsertakan dalam kompilasi

1.4.3 BAGIAN DEKLARASI DAN DEFINISI

Semua program C pada dasarnya tersusun dari rangkaian pemanggilan fungsi yang bekerja atas sekelompok data. Selain pemanggilan fungsi, program C mengandung komponen lain yang disebut statement. Statement C ada dua, yaitu : statement yang tidak dapat dieksekusi / non executable (bila dikompilasi tidak menghasilkan kode objek dan biasanya digunakan untuk mengatur alur program), dan statement yang dapat dieksekusi / executable (bila dikompilasi akan menghasilkan kode objek). Setiap pemanggilan fungsi maupun statement executable dalam C harus diakhiri dengan tanda titik koma (;).

Contoh program C++:

```
# include <iostream.h>
Void main ( )
{
Char pesan [ ] = "Hello, C++ programmers!" ;
Cout << pesan ;
Return 0 ;
}
```

Dalam contoh program C++ diatas, **Return** merupakan contoh statement executable yang menginstruksikan agar suatu fungsi mengembalikan nilai balik tertentu. Contoh statement non executable adalah : **If, else, dan while.**

Main () merupakan contoh fungsi, sedangkan **pesan** adalah contoh data. Baik data maupun fungsi harus dideklarasikan. Data perlu dideklarasikan agar compiler tahu berapa byte memori yang harus disediakan untuk data yang bersangkutan, sedangkan fungsi perlu dideklarasikan agar compiler dapat memeriksa ketepatan pemanggilan fungsi yang bersangkutan. Deklarasi fungsi sering disebut pula prototype fungsi.

1.5 INPUT DAN OUTPUT

Di ANSI C, operasi input dan output dilakukan dengan menggunakan fungsi-fungsi yang ada di header file **stdio.h**. contohnya untuk input dan output ke layer monitor digunakan perintah seperti **printf**, **scanf**, **putch**, dsb. Untuk input dan output ke file digunakan perintah seperti **fread**, **fwrite**, **fputc**, dsb.

C++ mempunyai teknik input dan output yang baru, yaitu : menggunakan **stream**. Header file untuk input dan output stream adalah **iostream.h** dan beberapa file lain, seperti **strstrea.h**, **fstream.h**, dan **constrea.h**.

Stream adalah suatu logika device (peralatan logika) yang menghasilkan dan menerima informasi atau suatu wadah yang digunakan untuk menampung keluaran dan menampung aliran data. Stream adalah nama umum untuk menampung aliran data (contoh : file, keyboard, mouse), maupun untuk keluaran (contoh : layer, printer).

Dalam C++ input berarti membaca dari stream dan output berarti menulis ke stream.

Bentuk Umum Output operator :

Cout << ekspresi ;

Bentuk umum Input operator :

Cin >> variable ;

Dalam C++, menggunakan escape sequences untuk merepresentasikan suatu karakter yang tidak terdapat dalam tradisional symbol. Beberapa diantaranya :

\n : linefeed / baris baru
\b : back space
\“ : petik ganda

Contoh program versi ANSI C :

```
# include <stdio.h>
Void main ( )
{
Int x ;
Printf ( "Masukkan sebuah bilangan : \n" ) ;
Scanf ( " %d ", &x ) ;
Printf ( "Bilangan yang dimasukkan adalah %d\n ", x ) ;
}
```

Contoh program versi C++ :

```
# include <iostream.h>
Void main ( )
{
Int x ;
Cout << "Masukkan sebuah bilangan : " << endl ;
Cin >> x ;
cout << "Bilangan yang dimasukkan adalah " << x << endl ;
}
```

Contoh Program untuk input dan output :

```
# include <iostream.h>
Main ( )
{
Int a ;
Cout << "masukkan suatu bilangan :";
Cin >> a ;
Cout << "nilai tersebut ditambah 1 = ' << a+1 ;
Return 0 ;
}
```