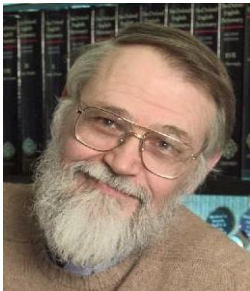


PENGENALAN PROGRAM C++

SEJARAH BAHASA C

C adalah bahasa pemrograman yang dapat dikatakan berada antara bahasa beraras rendah (bahasa yang berorientasi kepada mesin) dan bahasa beraras tinggi (bahasa yang berorientasi pada manusia).

Bahasa beraras tinggi mempunyai komparabilitas yang tinggi antar platform. Karena itu, amatlah mudah untuk membuat program pada berbagai jenis mesin. Berbeda halnya jika menggunakan bahasa beraras rendah, sebab setiap perintahnya bergantung sekali pada jenis mesin.



Brian W. Kernighan

Pencipta C adalah Brian W. Kernighan dan Dennis M. Ritchie pada sekitar tahun 1972. C adalah bahasa pemrograman terstruktur, yang membagi program dalam bentuk sejumlah blok. Tujuannya adalah untuk memudahkan dalam pembuatan dan pengembangan program.

Program yang ditulis dengan menggunakan C mudah sekali untuk dipindahkan dari satu jenis mesin ke jenis mesin lainnya. Hal ini berkat adanya standarisasi bahasa yaitu ANSI (American National Standards Institute) yang dijadikan acuan oleh para pembuat kompiler C.



Bjarne Stroustrup

C++ diciptakan oleh Bjarne Stroustrup di Laboratorium Bell, AT&T pada tahun 1983. Bahasa ini bersifat kompatibel dengan bahasa pendahulunya C. Pada mulanya C++ disebut *a better C*. Nama C++ sendiri diberikan oleh Rick Mascitti tahun 1983. Adapun tanda ++ berasal dari nama operator kenaikan pada bahasa C.

C diambil sebagai landasan dari C++ adalah karena keportabilitas C yang memungkinkan diterapkannya ke berbagai mesin, dari PC hingga mainframe, serta pada pelbagai sistem operasi (DOS, UNIX, VMS dan sebagainya)

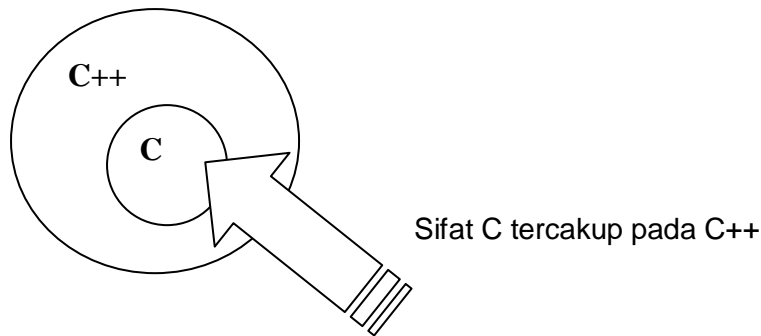


Dennis M. Ritchie

Keistimewaan yang sangat berarti pada C++ adalah karena bahasa ini mendukung pemrograman yang berorientasi obyek (Object Oriented Programming) tetapi sekali lagi C++ hanyalah bahasa yang bersifat hibrid, bukan bahasa murni yang berorientasi obyek.

Tujuan utama pembuatan C++ adalah untuk meningkatkan produktivitas pemrogram dalam membuat aplikasi. PBO dan C++ dapat mengurangi kekompleksitasan, terutama pada program yang besar yang terdiri dari 10.000 baris atau lebih.

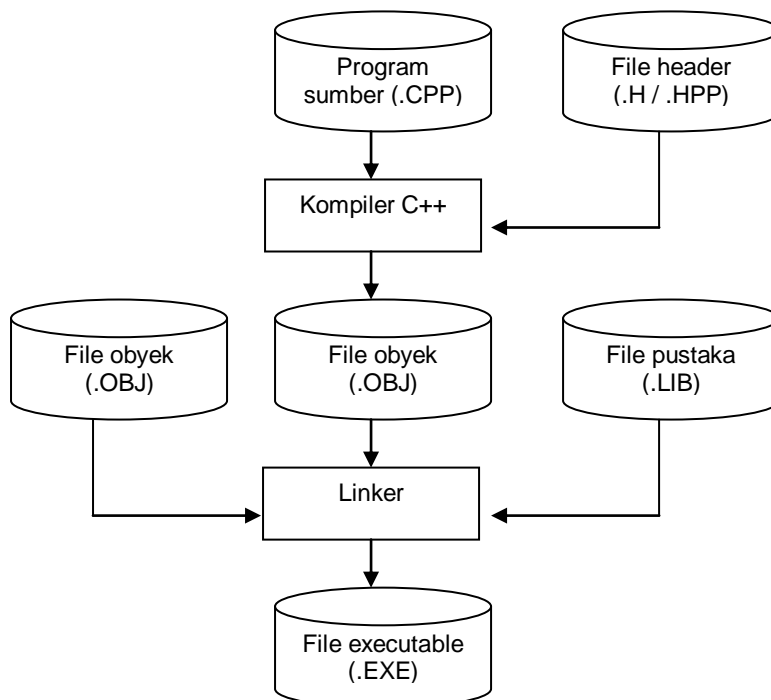
Greg Perry pada tahun 1993 menyatakan C++ dapat meningkatkan produktivitas pemrogram lebih dari dua kali dibandingkan bahasa prosedural seperti C, PASCAL dan BASIC karena kode yang ditulis dengan C++ lebih mudah untuk digunakan kembali pada program-program lain.



Program C++ dapat ditulis menggunakan sembarang editor teks seperti NOTEPAD ataupun menggunakan editor bawaan dari kompilator. Program C++ biasa ditulis dengan nama ekstensi **.CPP**. Agar program ini bisa dijalankan, maka harus dikompilasi terlebih dulu dengan menggunakan kompilator C++.

Pada saat pengkompilasian, program sumber (.CPP) bersama file-file header (berekstensi .H atau .HPP) akan diterjemahkan oleh kompilator C++ menjadi kode obyek (.OBJ). File obyek ini berupa file dalam format biner (berkode 0 dan 1).

Selanjutnya file obyek ini bersama-sama dengan file obyek lain serta file pustaka (.LIB) dikaitkan menjadi satu oleh *linker*. Hasilnya berupa file yang bersifat *executable*. File inilah yang bisa dijalankan dari sistem operasi secara langsung.



KOMPILER C++

Saat ini banyak kompilator bahasa C++ yang beredar di pasaran. Sebagai contoh *Microsoft Corporation* sebagai perusahaan pembuat perangkat lunak yang sangat termashur, mengeluarkan produk kompilator C++ berupa Microsoft C/C++ serta Visual C++. Adapun *Borland International* juga mengeluarkan Turbo C++ dan Borland C++.

Berikut ini contoh program C++ yang sangat sederhana dapat dilihat di bawah ini :

```
#include <iostream.h>
void main()
{
    cout << "Halo, Apa kabar? \n";
}
```

Keterangan :

- Kata *void* yang mendahului `main()` digunakan untuk menyatakan bahwa fungsi ini tidak memiliki nilai balik.
- Program C++ memang tidak pernah lepas dari fungsi. Sebab fungsi adalah salah satu dasar penyusun blok pada C++. Sebuah program C++ minimal mengandung sebuah fungsi yaitu `main()`. Fungsi ini menjadi awal dan akhir eksekusi program. Tanda `()` digunakan untuk mengapit argumen fungsi, yaitu nilai yang akan dilewatkan ke fungsi. Pada fungsi `main()` di atas, tidak ada argumen yang diberikan. Oleh karena itu tidak ada entri di dalam `()`.
- Tanda “{” hingga “}” disebut tubuh fungsi. Lebih spesifik lagi, semua yang terletak di dalam tanda { } disebut blok. Tanda “{” menyatakan awal eksekusi program sedangkan “}” menyatakan akhir eksekusi program. Di dalam tanda { } bisa terkandung sejumlah unit yang disebut pernyataan (*statement*)

PERNYATAAN

Baris `cout << "Halo, Apa kabar? \n";` merupakan contoh dari sebuah pernyataan. Pernyataan tersebut digunakan untuk menampilkan tulisan yang terletak pada sepasang tanda petik ganda ke layar. Dalam hal ini, tulisan yang terletak pada sepasang tanda petik disebut konstanta *string*.

Umumnya pernyataan dapat berupa :

- Instruksi untuk menyuruh komputer melakukan sesuatu.
- Menampilkan tulisan ke layar
- Menghitung suatu operasi aritmatika
- membaca dari keyboard dan sebagainya

Setiap pernyataan harus diakhiri dengan tanda titik koma (;).

MENGENAL COUT

Pengenal `cout` merupakan sebuah obyek didalam didalam C++. Obyek ini disediakan oleh C++ untuk mengarahkan data ke *standard output* (normalnya adalah layar).

Misal : `cout << "Selamat Belajar C++\n";`

Tanda “<<” merupakan sebuah operator yang disebut operator penyisipan atau peletakan. Operator ini akan mengarahkan operand (data) yang terletak di sebelah kanannya ke obyek yang terletak di sebelah kiri.

Di dalam sebuah string, karakter “\” diikuti dengan karakter tertentu menyatakan sebuah karakter tunggal. Pada contoh di atas , “\n” adalah karakter pindah baris (*newline*)

Contoh :

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

```
}

```

Hasil :

```
Selamat Belajar C++
Ternyata Mudah Juga...
```

#INCLUDE <IOSTREAM.H>

Baris `#include <iostream.h>` bukanlah suatu pernyataan. Itulah sebabnya tidak ada tanda titik koma yang diperlukan. Baris tersebut menginstruksikan kepada kompiler untuk menyisipkan file lain misalnya *iostream.h* saat program dikompilasi,

File-file yang berakhiran dengan `.h` disebut file *header*, yaitu file-file yang berisi berbagai deklarasi, seperti fungsi, variabel dan sebagainya. Baris `#include <iostream.h>` perlu disertakan pada program yang melibatkan obyek `cout`. Tanpa kehadiran baris tersebut akan terjadi kesalahan sewaktu program dikompilasi.

`#include` adalah salah satu pengarah *praprosesor* yang tersedia pada C++. Bentuk umum `#include` dapat berupa :

1. `#include <nama file>`

Mengisyaratkan bahwa pencarian file dilakukan pada direktori khusus(direktori file include yang bisa di-set melalui kompiler)

2. `#include "nama file"`

Menyatakan bahwa pencarian nama file dilakukan pertama kali pada direktori kerja. Jika nama file tidak ditemukan maka akan dicari pada direktori yang tercantum pada variabel lingkungan PATH

CLRSCR() UNTUK MENGHAPUS LAYAR

Perintah untuk menghapus layar biasa dilibatkan dalam program. Pernyataan yang digunakan adalah : `clrscr()`; Pernyataan tersebut menggunakan baris `#include <conio.h>`

Contoh :

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    cout << "Halo, Apa kabar ?\n";
    cout << "Selamat Belajar C++\n";
    cout << "Ternyata Mudah Juga...";
}
```

Hasil :

```
Halo, Apa kabar ?
Selamat Belajar C++
Ternyata Mudah Juga...
```

GAYA PENULISAN PROGRAM

C++ memberikan keleluasaan dalam menuliskan bentuk atau gaya program, misalnya dari program di atas dapat dirubah menjadi :

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

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

```
void main() { clrscr(); cout << "Halo, Apa kabar ?\n"; cout << "Selamat Belajar C++\n"; cout << "Ternyata Mudah Juga..."; }
```

Namun penulisan diatas tidak dianjurkan karena program sulit dibaca. Program hendaknya ditulis dalam bentuk yang mudah dibaca oleh siapa saja, bukan dalam bentuk yang paling padat dan yang paling penting adalah konsistensi gaya yang dipilih. Pemberian spasi kosong pada posisi tertentu akan membantu program mudah dibaca.

KOMENTAR

Komentar merupakan bagian yang penting dalam program. Kehadirannya sangat membantu dalam memahami program, karena berupa penjelasan-penjelasan mengenai program atau bagian-bagian program.

Penjelasan dapat berupa :

- Tujuan atau fungsi program
- Saat program dibuat atau direvisi
- Keterangan-keterangan lain tentang kegunaan sejumlah pernyataan dalam program.

Seandainya program tidak ditujukan untuk dibaca orang lain, komentar tetap bermanfaat bagi penulis program itu sendiri. Pada C++, suatu komentar diawali dengan dua tanda garis miring (//) atau dengan menggunakan bentuk /* dan */

Contoh :

```
// -----
// program pencetakan
// dibuat oleh Yudi Irawan
// -----

#include <iostream.h>
#include <conio.h>
void main()
{
  clrscr(); /* pernyataan ini berfungsi untuk
            membersihkan layar */
  cout << "Halo, Apa kabar ?\n";
  cout << "Selamat Belajar C++\n";
  cout << "Ternyata Mudah Juga...";
} // akhir program
```

TUGAS

Buat program C++ untuk menampilkan BIODATA anda