

# Pengenalan Bahasa C++



## Overview

---

Berbicara tentang C++ tak lepas dari C, sebagai bahasa pendahulunya. C adalah bahasa pemrograman yang dapat dikatakan berada antara bahasa beraras rendah (bahasa yang berorientasi pada mesin) dan bahasa beraras tinggi (bahasa yang berorientasi pada manusia). Seperti diketahui bahasa tingkat tinggi mempunyai kompatibilitas yang tinggi antar *platform*.



## Tujuan

---

- ☒ Mengetahui sejarah bahasa C++
- ☒ Mengenal struktur dasar bahasa C++
- ☒ Mengenal elemen elemen C++
- ☒ Memahami tujuan penggunaan bahasa C++

### 3.1 Sejarah C++

Tahun 1978, Brian W. Kernighan & Dennis M. Ritchie dari AT & T Laboratories mengembangkan bahasa B menjadi bahasa C. Bahasa B yang diciptakan oleh Ken Thompson sebenarnya merupakan pengembangan dari bahasa BCPL ( Basic Combined Programming Language ) yang diciptakan oleh Martin Richard.

Sejak tahun 1980, bahasa C banyak digunakan pemrogram di Eropa yang sebelumnya menggunakan bahasa B dan BCPL. Dalam perkembangannya, bahasa C menjadi bahasa paling populer diantara bahasa lainnya, seperti PASCAL, BASIC, FORTRAN.

Tahun 1989, dunia pemrograman C mengalami peristiwa penting dengan dikeluarkannya standar bahasa C oleh *American National Standards Institute* (ANSI). Bahasa C yang diciptakan Kernighan & Ritchie kemudian dikenal dengan nama ANSI C.

Mulai awal tahun 1980, Bjarne Stroustrup dari AT & T Bell Laboratories mulai mengembangkan bahasa C. Pada tahun 1985, lahirlah secara resmi bahasa baru hasil pengembangan C yang dikenal dengan nama C++. Sebenarnya bahasa C++ mengalami dua tahap evolusi. C++ yang pertama, dirilis oleh AT&T Laboratories, dinamakan **cfont**. C++ versi kuno ini hanya berupa kompiler yang menterjemahkan C++ menjadi bahasa C

sebuah kompiler yang mampu mengubah C++ langsung menjadi bahasa mesin (assembly). Sejak evolusi ini, mulai tahun 1990 C++ menjadi bahasa berorientasi obyek yang digunakan oleh sebagian besar pemrogram professional.

### 3.2 Struktur Bahasa C++

#include <file include>	-----	Header
Main()	-----	Fungsi
{		
Pernyataan program;	-----	Isi Program
}		

Contoh 1	Hasil
<pre>// Program Pertama saya di C++ #include &lt;iostream.h&gt; int main () {     cout &lt;&lt; "Hello World!";     return 0; }</pre>	Hello World!

Sisi kiri merupakan *source code*, yang dapat diberi nama *hiworld.cpp* dan sisi kanan adalah hasilnya setelah di-kompile dan di-eksekusi.

Program diatas merupakan salah satu program paling sederhana dalam C++, tetapi dalam program tersebut mengandung komponen dasar yang selalu ada pada setiap pemrograman C++. Jika dilihat satu persatu :

#### **//Program Pertama saya di C++**

Baris ini adalah komentar. semua baris yang diawali dengan dua garis miring (//) akan dianggap sebagai komentar dan tidak akan berpengaruh terhadap program. Dapat digunakan oleh programmer untuk menyertakan penjelasan singkat atau observasi yang terkait dengan program tersebut.

#### **#include <iostream.h>**

Kalimat yang diawali dengan tanda (#) adalah *preprocessor directive*. Bukan merupakan baris kode yang dieksekusi, tetapi indikasi untuk kompiler. Dalam kasus ini kalimat **#include <iostream.h>** memberitahukan preprocessor kompiler untuk menyertakan header file standard **iostream**. File spesifik ini juga termasuk library deklarasi standard I/O pada C++ dan file ini disertakan karena fungsi-fungsinya akan digunakan nanti dalam program.

#### **int main ()**

Baris ini mencocokkan pada awal dari deklarasi fungsi **main**. fungsi **main** merupakan titik awal dimana seluruh program C++ akan mulai dieksekusi. Diletakan diawal, ditengah atau diakhir program, isi dari fungsi main akan selalu dieksekusi

pertama kali. Pada dasarnya, seluruh program C++ memiliki fungsi **main**.

**main** diikuti oleh sepasang tanda kurung () karena merupakan fungsi. pada C++, semua fungsi diikuti oleh sepasang tanda kurung () dimana, dapat berisi argumen didalamnya. Isi dari fungsi **main** selanjutnya akan mengikuti, berupa deklarasi formal dan dituliskan diantara kurung kurawal ({}), seperti dalam contoh.

**cout << "Hello World";**

Intruksi ini merupakan hal yang paling penting dalam program contoh. **cout** merupakan standard output stream dalam C++ (biasanya monitor). **cout** dideklarasikan dalam header file `iostream.h`, sehingga agar dapat digunakan maka file ini harus disertakan.

Perhatikan setiap kalimat diakhiri dengan tanda semicolon (;). Karakter ini menandakan akhir dari instruksi dan harus disertakan pada setiap akhir instruksi pada program C++ manapun.

**return 0;**

Intruksi **return** menyebabkan fungsi **main()** berakhir dan mengembalikan kode yang mengikuti instruksi tersebut, dalam kasus ini **0**. Ini merupakan cara yang paling sering digunakan untuk mengakhiri program.

Tidak semua baris pada program ini melakukan aksi. Ada baris yang hanya berisi komentar (diawali //), baris yang berisi instruksi untuk preprocessor kompiler (Yang diawali #), kemudian baris yang merupakan inisialisasi sebuah fungsi (dalam kasus ini, fungsi **main**) dan baris yang berisi instruksi (seperti, **cout <<**), baris yang terakhir ini disertakan dalam blok yang dibatasi oleh kurung kurawal ({} ) dari fungsi **main**.

Struktur program dapat dituliskan dalam bentuk yang lain agar lebih mudah dibaca, contoh :

```
int main ()
{
    cout << " Hello World ";
    return 0;
}
```

Atau dapat juga dituliskan :

```
int main () { cout << "Hello World"; return 0; }
```

Dalam satu baris dan memiliki arti yang sama dengan program-program sebelumnya. pada C++ pembatas antar instruksi ditandai dengan semicolon (;) pada setiap akhir instruksi.

### Komentar

Komentar adalah bagian dari program yang diabaikan oleh kompiler. Tidak melaksanakan aksi apapun. Mereka berguna untuk memungkinkan para programmer untuk memasukkan catatan atau deskripsi tambahan mengenai program tersebut. C++ memiliki dua cara untuk menuliskan komentar :

```
//      Komentar baris
/*      Komentar Blok */
```

Komentar baris, akan mengabaikan apapun mulai dari tanda (//) sampai akhir dari baris yang sama. Komentar Blok, akan mengabaikan apapun yang berada diantara tanda /\* dan \*/.

### File include dasar input output pada C++

➤ *#include <stdio.h> / #include "stdio.h"*

Stdio = Standart Input Output

Untuk menjalankan fungsi input dan cetak

Input : scanf

Cetak : printf

➤ *#include <iostream.h> / #include "iostream.h"*

Iostream = Input Output Stream

Untuk menjalankan fungsi input dan cetak

Input : cin

Output : cout

➤ *#include <conio.h> / #include "conio.h"*

Conio = Console Input Output

Untuk menjalankan fungsi getch dan clrscr

Getch : untuk menahan tampilan

Clrscr : untuk membersihkan layar

## **Compile File**

Menterjemahkan program dari bahasa manusia kedalam bahasa yang dimengerti komputer

Klik **project – compile** atau

**Alt + F9**

## **Menjalankan File**

Klik **debug – Run** atau

**Ctrl + F9**

## **3.3 Elemen Dasar C++**

Elemen-elemen C++ terdiri dari :

### **3.3.1 Himpunan Karakter**

Himpunan karakter pada C++ terdiri dari huruf, digit maupun simbol-simbol lainnya (termasuk spasi, karakter kontrol).



#### **Huruf**

A B C D E F G H I J K L M N O P Q R S T U V X Y Z

A b c d e f g h i j k l m n o p q r s t u v w x y z



#### **Digit**

0 1 2 3 4 5 6 7 8 9



#### **Simbol dan Lain-lain**

\_ - + \* dan sebagainya

### **3.3.2 Pengenal (Identifier)**

Pengenal adalah suatu nama yang biasa dipakai dalam pemrograman untuk menyatakan :



Variabel



Fungsi



Konstata



Label



Tipe data



Obyek

Serta hal-hal lain yang dideklarasikan atau didefinisikan oleh pemrogram.

### 3.3.3 Penamaan Pengenal

Suatu pengenal berupa satu atau beberapa karakter :

- ⇒ Huruf
- ⇒ Garis bawah ( \_ )
- ⇒ Digit

Dan berawalan dengan huruf atau garis bawah. Disarankan agar pemberian nama pengenal menggunakan kata yang berarti dan mudah dibaca.

### 3.3.4 Huruf Kecil dan kapital Berbeda

Pada C++, huruf kecil dan huruf kapital pada suatu pengenal dianggap berbeda. sifat ini dikenal dengan istilah *case sensitive*. Sebab itu pengeanal **NAMA**, **Nama** dan **nama** menyatakan tiga pengenal yang berbeda.

### 3.3.5 Kata Kunci

Kata kunci (*keyword*) adalah pengenal sistim yang mempunyai makna khusus bagi kompiler. Kata kunci tidak dapat dirubah. Karena itu kata kunci tidak dapat digunakan sebagai pengenal.

### 3.3.6 Tipe Data

Tipe data dasar pada C++ meliputi :

- ⇒ Char
- ⇒ Float
- ⇒ Long
- ⇒ Int
- ⇒ double
- ⇒ short
- ⇒ long double

Tipe data yang berhubungan dengan bilangan bulat adalah **char**, **int**, **short** dan **long**. Sedangkan lainnya berhubungan dengan bilangan pecahan.

### **3.3.7 Variabel dan Konstanta**

Data pada C++ tersusun dari :

↳ Variabel

↳ Konstanta

Variabel digunakan dalam program untuk menyimpan suatu nilai, nilai yang ada padanya dapat diubah selama eksekusi program berlangsung. Sementara konstanta adalah kebalikan variabel yang sudah memiliki ketetapan dan nilainya tidak dapat diubah (statis) selama eksekusi program berlangsung.