DISUSUN OLEH : ADAM MUKHARIL BACHTIAR, S.Kom.

Sekilas Tentang C dan C++

Sebuah bahasa pemrograman dapat dikategorikan ke dalam tiga level bahasa, yaitu:

1. Bahasa tingkat tinggi
2. Bahasa tingkat menengah
3. Bahasa tingkat rendah.

Semakin tinggi level suatu bahasa maka bahasa pemrograman tersebut akan mudah untuk dipelajari karena dekat dengan bahasa manusia. Sebaliknya, semakin rendah level suatu bahasa pemrograman maka makin sulit untuk dipelajari karena dekat dengan bahasa mesin.

Bahasa C (dibuat oleh Brian W. Kernighan dan Dennis M. Ritchie) merupakan bahasa Intermediate yang artinya adalah bahasa tersebut bisa dikatakan sebagai High Level Language, di mana para programmer diberikan sederetan sintaks (aturan penulisan) yang dapat dimengerti oleh manusia. Akan tetapi bahasa C juga dapat digolongkan sebagai Low Level Language karena pada bahasa C disediakan pula sintak dalam bentuk bahasa Assembly (di mana kita ketahui bahwa bahasa Assembly merupakan salah satu Low Level Language).

Bahasa C menyediakan beberapa komponen yang disediakan oleh perangkat lunak tersebut agar seorang programmer dapat dengan mudah mengimplementasikan kodenya. Adapun komponen bahasa C terdiri dari:

1. Editor
2. Interpreter
3. Compiler
4. Debugging.

Berlanjut ke C++, apabila berbicara C++ biasanya tidak akan lepas dari C sebagai bahasa pendahulunya. C++ diciptakan satu dekade setelah C. Diciptakan oleh Bjarne Stroustrup pada tahun 1983. Bahasa ini kompatibel dengan bahasa C. keistimewaan yang sangat berarti pada C++ karena bahasa ini mendukung pemrograman yang berorientasi objek (Object Oriented Programming – OOP).

Semua bahasa mempunyai kelemahan atau kelebihan sendiri-sendiri. Begitu juga dengan bahasa C dan C++. Adapun kelebihannya adalah sebagai berikut:

1. Banyak memiliki operator untuk mengolah/memanipulasi data.
2. Bahasa C termasuk sebagai bahasa terstruktur sehingga program dapat lebih mudah dipahami atau dikembangkan.
3. Kecepatan eksekusi tinggi.

Dan beberapa kelemahannya adalah sebagai berikut:

1. Banyaknya operator atau cara penulisan program kadang menimbulkan kebingungan para pemakainya.
2. Perlu adanya ketelitian dalam penulisan program karena C dan C++ bersifat Case Sensitive (Membedakan antara huruf kapital dan huruf kecil).

Editor

Editor adalah sebuah fasilitas yang disediakan oleh bahasa C dan C++ untuk menuliskan kode yang telah didesain oleh programmer. Editor yang disediakan pada umumnya sama dengan yang disediakan oleh perangkat lunak lainnya yang menangani file **TEXT**. Akan tetapi untuk beberapa editor terbaru model TEXT ini dapat dibedakan antara keyword, variabel, dan sebagainya.

Interpreter

Kebanyakan perangkat lunak bahasa pemrograman menyediakan fasilitas Interpreter. Demikian juga dengan bahasa C. Interpreter digunakan untuk membaca kode yang telah ditulis oleh programmer untuk diterjemahkan oleh Interpreter C dan C++. Sehingga sekumpulan kode yang telah ditulis dapat berjalan sesuai dengan sintaks yang telah ditentukan oleh bahasa C dan C++.

Compiler

Tidak semua perangkat lunak bahasa pemrograman menyediakan fasilitas Compiler. Tetapi untuk bahasa C dan C++, fasilitas ini disediakan. Compiler digunakan untuk mentranslator sekumpulan kode yang telah ditulis sesuai dengan sintak yang ditentukan oleh bahasa C ke bentuk yang lain. Dalam hal ini diubah ke

dalam bentuk bahasa Assembly sehingga selanjutnya akan menghasilkan suatu file execute (.exe). di mana file tersebut dapat berdiri sendiri tanpa memerlukan perangkat lunak lainnya.

Debugging

Bahasa C dan C++ menyediakan fasilitas debugging yang dapat digunakan untuk menelusuri setiap kode yang telah ditulis. Sehingga programmer dapat mengetahui perubahan dan pengaruh serta kesalahan setiap kode per baris yang telah ditulis. Fasilitas debug yang disediakan antara lain:

1. Debug per baris.
2. Debug per prosedur/fungsi.
3. Debug per breakpoint.

Software yang Dapat Digunakan

Ada beberapa software yang dapat digunakan untuk membuat program dengan bahasa C dan C++, antara lain:

1. Turbo C++
2. Borland C++
3. Dev-C++
4. GCC.

Untuk software nomor 1 sampai 3 digunakan pada sistem operasi windows sedangkan untuk software nomor 4 digunakan di sistem operasi Linux.

Dev-C++

Untuk membuat sebuah program, seorang programmer membutuhkan sebuah editor untuk menuliskan sintaks programnya beserta compiler untuk mengubah sintaks tersebut menjadi sebuah file executable. Sekarang sudah banyak sekali software yang menggabungkan antara editor dengan compilernya. Bahkan untuk membuat sebuah file executable, seorang programmer tinggal mengklik satu buah tombol.

Salah satu software yang bisa digunakan untuk membangun sebuah program C dan C++ yaitu Dev-C++. Salah satu kelebihan dari software ini adalah keterbukaan sumber (open source) sehingga orang umum bisa mengupgrade aplikasi ini serta software ini bersifat freeware (**gratis**). Sehingga tidak aneh apabila software ini sangat digemari dibandingkan software-software lain yang berbayar (biasanya lumayan mahal). Selain itu software ini juga memungkinkan untuk menambahkan library-library yang bukan bawaan dari software ini. Akan tetapi software ini juga memiliki kelemahan yaitu ada beberapa prosedur dan fungsi yang ada pada software lain dihilangkan dalam Dev-C++ ini.

Instalasi Dev-C++

Untuk dapat menggunakan aplikasi Dev-C++, kita harus menginstal terlebih dahulu aplikasi ini ke dalam hard disk. Adapun langkah-langkah penginstalasian aplikasi ini adalah sebagai berikut:

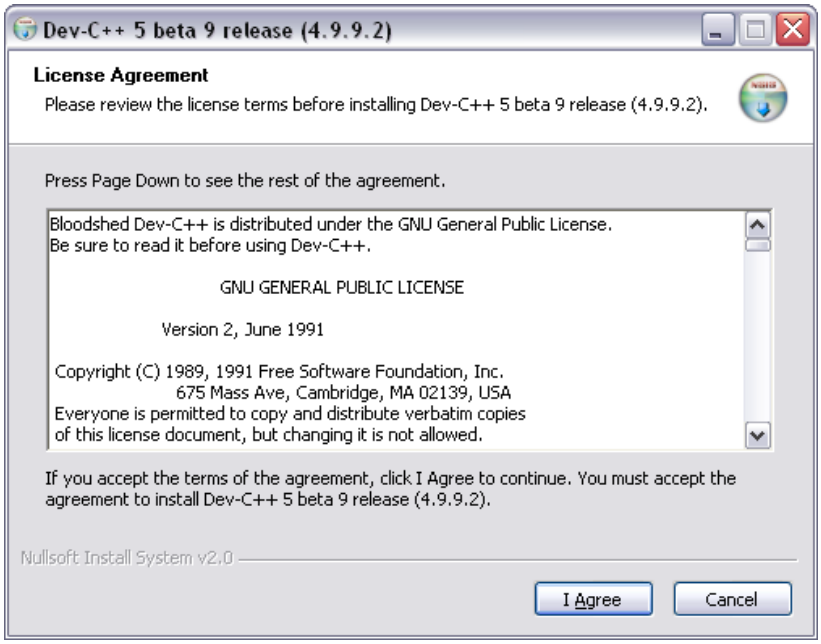
1. Download terlebih dahulu aplikasi Dev-C++ dari alamat http://www.bloodshed.net/dev/devcpp.html.
2. Setelah itu aktifkan filenya sehingga terlihat tampilan sebagai berikut:

Gambar 1.1 Tampilan pertama instalasi Dev-C++

3. Tunggu sampai proses instalasi berlanjut. Pilih bahasa yang diinginkan lalu klik OK.

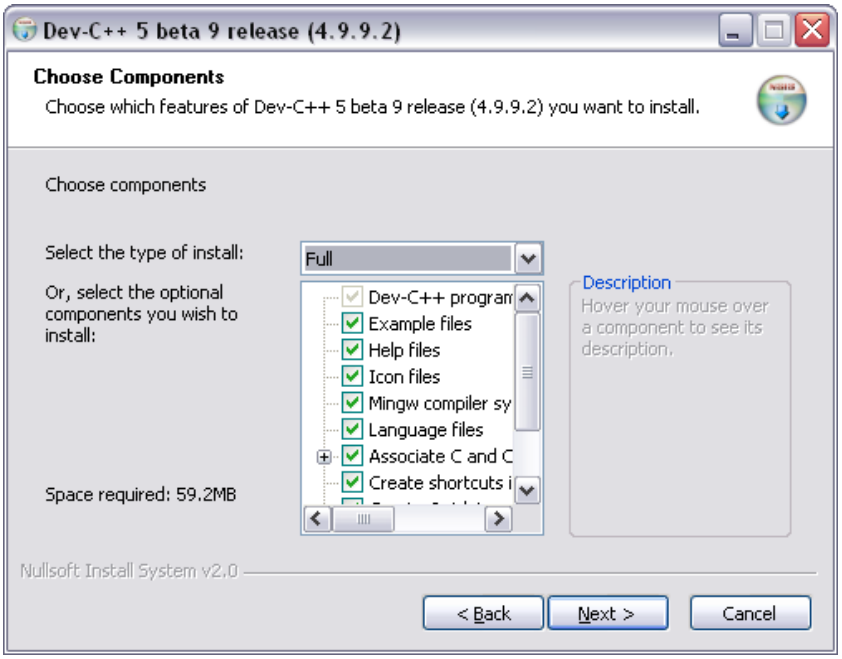
Gambar 1.2 Tampilan pemilihan bahasa

4. Selanjutnya akan tampil layar License Agreement lalu klik I Agree.



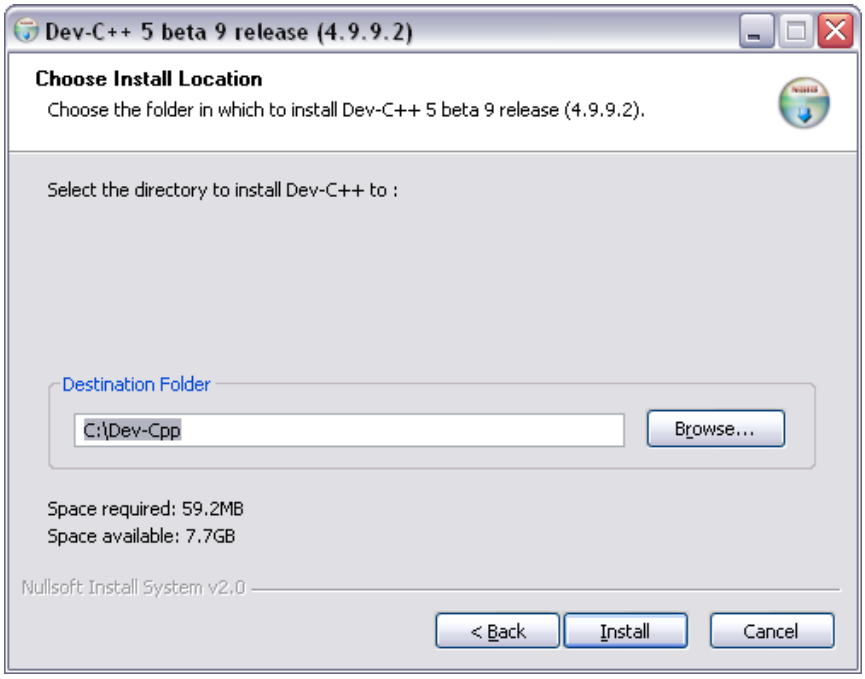
Gambar 1.3 Tampilan license agreement

5. Selanjutnya kita bisa memilih komponen yang ingin diinstal. Pilih tipe instalan Full lalu klik Next.



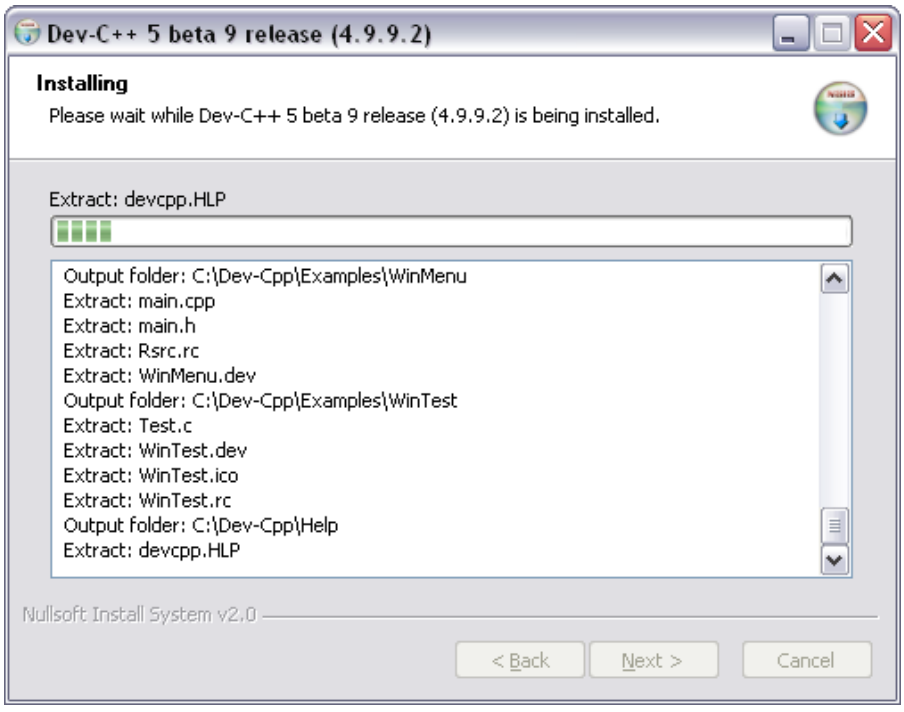
Gambar 1.4 Tampilan komponen

6. Pilih direktori untuk anda menginstalkan aplikasi Dev-C++ lalu klik Install.



Gambar 1.5 Tampilan lokasi instalasi

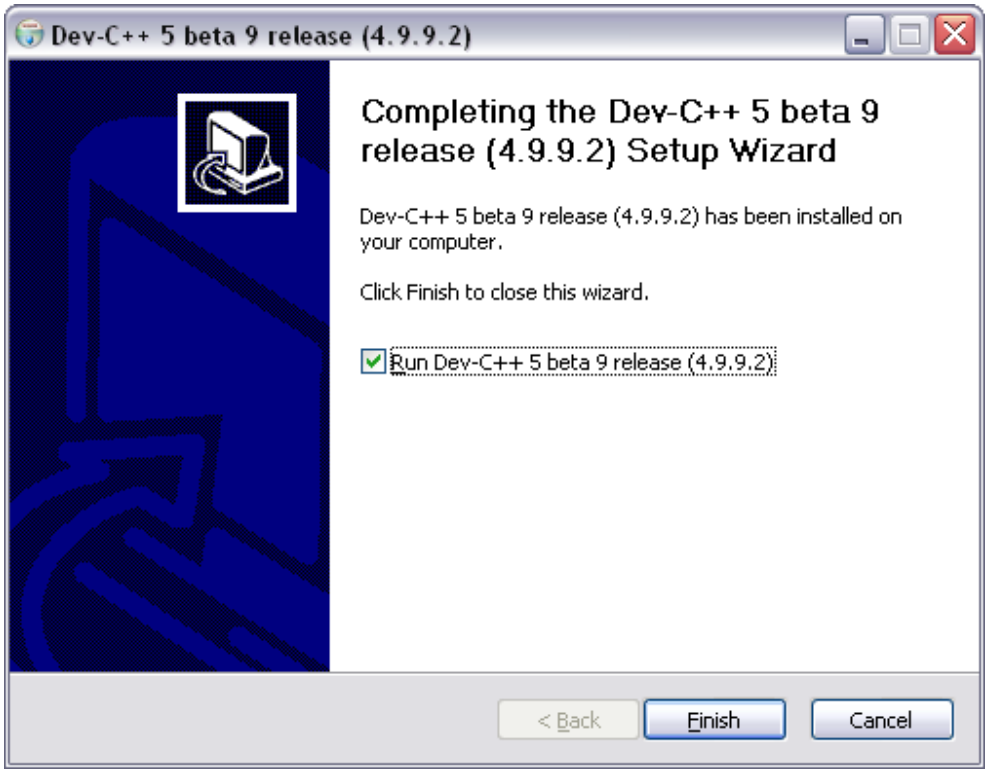
7. Tunggu sampai proses instalasi selesai kemudian akan timbul layar yang berisi pertanyaan apakah proses instalasi ini untuk semua pengguna klik Yes untuk pertanyaan tersebut.



Gambar 1.6 Tampilan proses instalasi

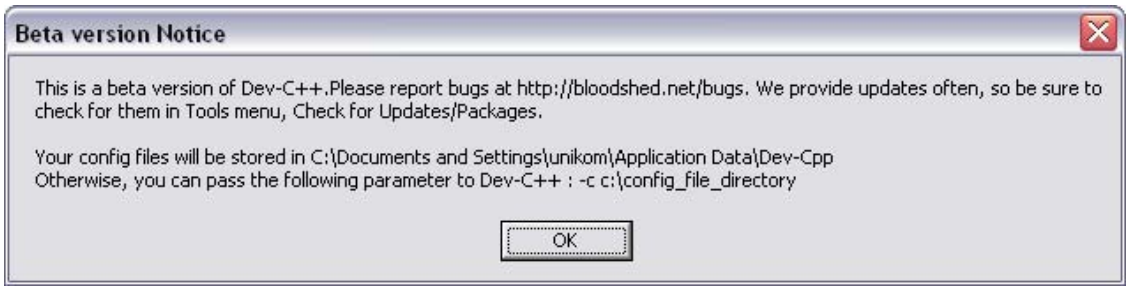
Gambar 1.7 Tampilan pertanyaan instalasi

8. Pada layar terakhir anda bisa langsung menjalankan aplikasi Dev-C++ dengan cara mengklik combo box yang disediakan lalu klik Finish.



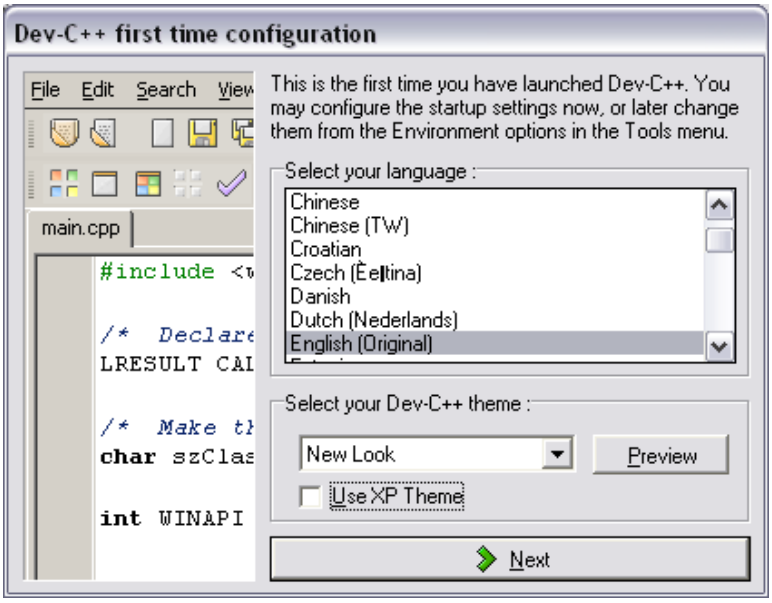
Gambar 1.8 Tampilan selesai instalasi

9. Proses instalasi akan dilanjutkan dengan proses konfigurasi. Langkah pertama kita akan diberi tahu versi berapa dari Dev-C++ yang kita gunakan. Klik OK.



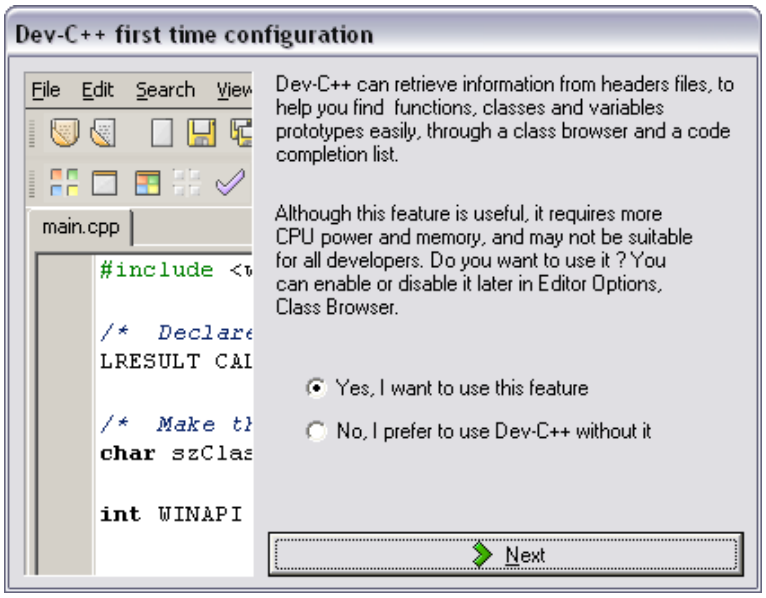
Gambar 1.9 Tampilan versi Dev-C++

10. Pada form pertama pilih bahasa yang akan digunakan di aplikasi Dev-C++. Pilih English (original). Kemudian pilih theme yang akan digunakan. Kita juga bisa memilih XP theme agar tampilannya lebih lembut. Lalu klik Next.



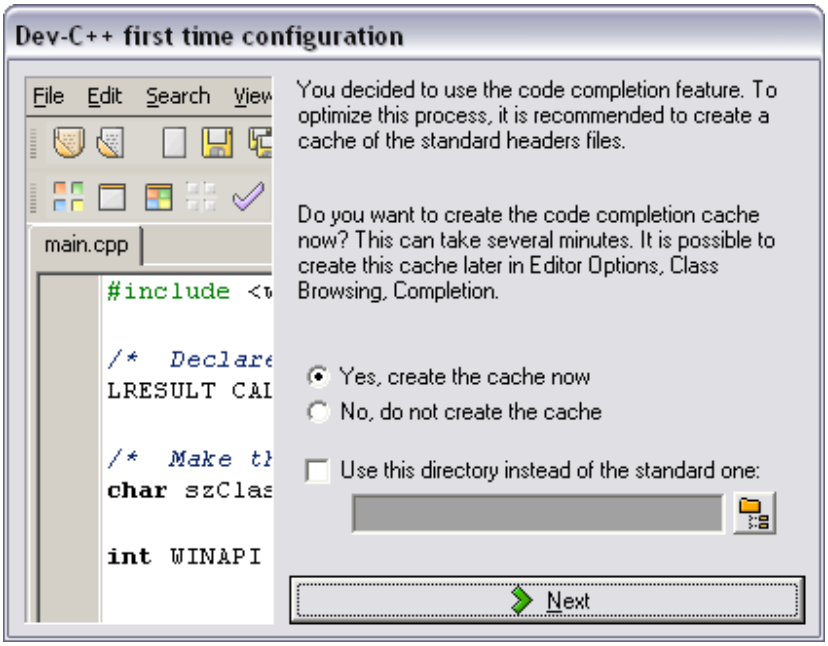
Gambar 1.10 Tampilan form pertama konfigurasi

11. Di form kedua kita bisa memilih untuk menggunakan fitur bantuan code completion (untuk menemukan class, function, dan lain-lain dengan cepat) dengan syarat memori yang digunakan cukup untuk menjalankan fitur ini. Klik Next.



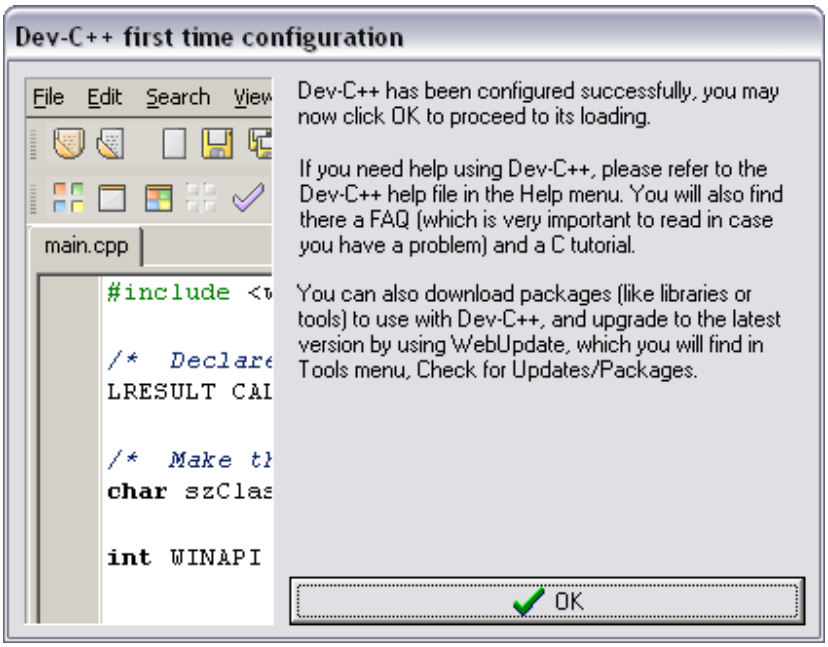
Gambar 1.11 Tampilan form kedua konfigurasi

12. Di form ketiga kita bisa membuat cache (ruang khusus) untuk menampung fitur code completion lalu klik Next. Tunggu sampai proses konfigurasi selesai.



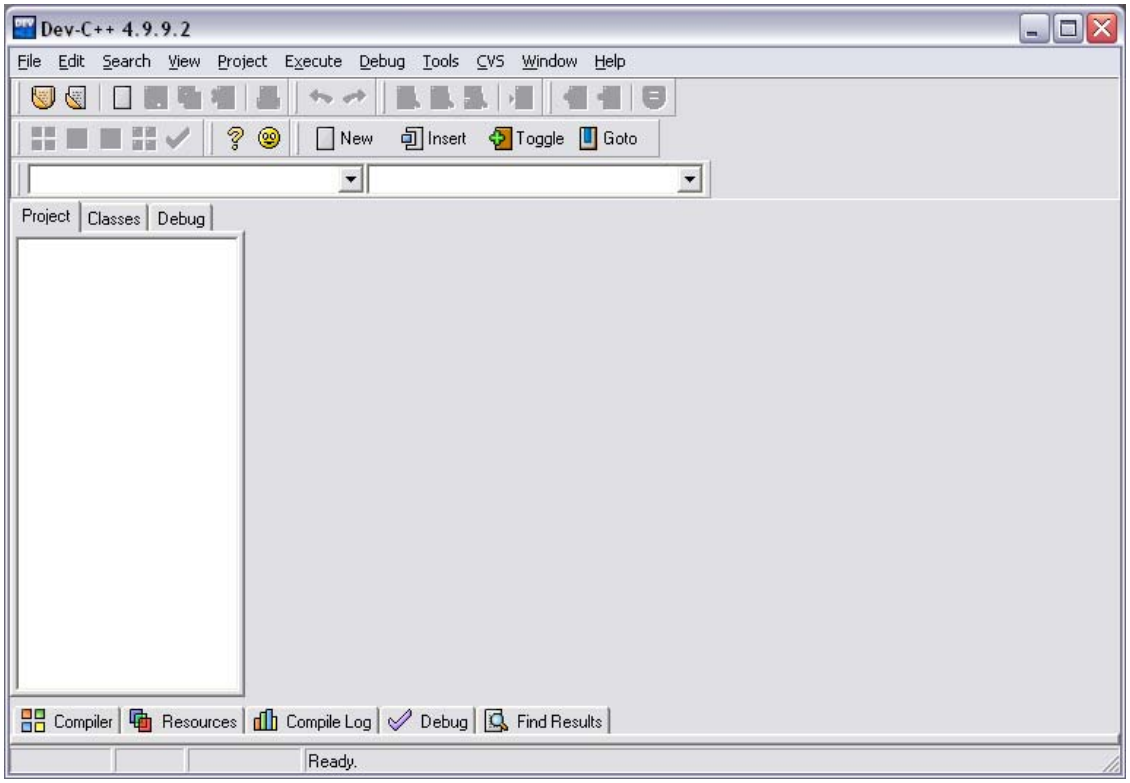
Gambar 1.12 Tampilan form ketiga konfigurasi

13. Pada form terakhir akan diberi tahu bahwa proses konfigurasi telah selesai kemudian klik OK.



Gambar 1.13 Tampilan form terakhir konfigurasi

14. Setelah itu aplikasi Dev-C++ bisa digunakan.



Gambar 1.14 Tampilan jendela kerja Dev-C++

Kerangka Program C dan C++

Sebelum masuk ke dalam bahasa c dan c++ ada baiknya mempelajari terlebih dahulu tentang kerangka program yang ada di bahasa c dan c++. Setiap program yang ditulis menggunakan bahasa c atau c++, program tersebut akan memiliki minimal sebuah fungsi utama yang dipanggil dengan nama **main()**. Tapi tidak menutup kemungkinan apabila seorang pengguna membuat fungsi di luar dari fungsi main().

Kerangka pertama yang akan dibahas kerangka bahasa c. Adapun kerangka bahasa c adalah sebagai berikut:

```
1  /*
2     Nama Program
3     Nama File    : …….c
4     NIM          : ……
5     Nama Pembuat : ……
6  */
7
8  # include <stdio.h> //atau #include “stdio.h”
9  # include <conio.h> //biasanya diganti dengan conio2.h
10
11 //prototipe fungsi
12 Tipe_data nama_fungsi1(parameter1, parameter2, ...);
13 Tipe_data nama_fungsi2(parameter1, parameter2, ...);
14
15 //Implementasi fungsi
16 Tipe_data nama_fungsi1(parameter1, parameter2, ...)
17 {
18    Pernyataan_yang_akan_dilakukan;
19    ...
20 }
21
22 Tipe_data nama_fungsi2(parameter1, parameter2, ...)
23 {
24    Pernyataan_yang_akan_dilakukan;
25    ...
26 }
27
28 //Fungsi Utama
29 int main(int argc, char argc[])
30 {
31    Pernyataan_yang_akan_dilakukan;
32    ...
33    return 0;
34 }
```

Penggalan kerangka di atas bukan bentuk baku dari bahasa c. Sebagai contoh untuk fungsi di luar fungsi utama tidak harus ada. Apabila dirasa cukup menuliskan pernyataan-pernyataan di dalam fungsi utama maka membuat fungsi tambahan tidaklah menjadi suatu kewajiban. Letak dari fungsi pun tidak harus seperti pada penggalan kerangka di atas. Kita bisa saja meletakkan fungsi utama sebelum fungsi-fungsi tambahan.

Untuk melihat perbedaan antara bahasa c dan bahasa c++ bisa dilihat pada penggalan kerangka bahasa c++ berikut ini:

```
1   /*
2      Nama Program
3      Nama File    : …….cpp
4      NIM          : ……
5      Nama Pembuat : ……
6   */
7
8   # include <iostream.h> //atau #include “stdio.h”
9
10  //prototipe fungsi
11  Tipe_data nama_fungsi1(parameter1, parameter2, ...);
12  Tipe_data nama_fungsi2(parameter1, parameter2, ...);
13
14  //Implementasi fungsi
15  Tipe_data nama_fungsi1(parameter1, parameter2, ...)
16  {
17     Pernyataan_yang_akan_dilakukan;
18     ...
19  }
20
21  Tipe_data nama_fungsi2(parameter1, parameter2, ...)
22  {
23     Pernyataan_yang_akan_dilakukan;
24     ...
25  }
26
27  //Fungsi Utama
28  int main(int argc, char argc[])
29  {
30     Pernyataan_yang_akan_dilakukan;
31     ...
32     return 0;
33  }
```

Kalau diperhatikan lebih dekat maka perbedaan antara kedua bahasa tersebut tidaklah terlalu signifikan. Perbedaan yang paling mendasar adalah penggunaan file header yang terletak setelah kata include. Untuk bentuk umum dan peletakkan fungsinya pun sama. Perbedaan akan terlihat jelas ketika seorang programmer mulai untuk merancang sebuah fungsi (terutama dalam hal input dan output).

File .H (Header File)

File header (file dengan ekstensi .h) adalah file yang berisi fungsi-fungsi dan telah dikompilasi sebelumnya sehingga bisa digunakan dalam pembangunan sebuah program. Untuk memanggil file header tersebut kita bisa menggunakan dua buah cara yang akan dijelaskan dalam poin-poin berikut:

1. #include <……..h>

   Cara pemanggilan yang pertama biasanya digunakan untuk memanggil file header bawaan (yang bukan dibuat sendiri oleh programmernya) yang sudah tersedia di dalam aplikasi Dev-C++.

2. #include "……..h"

   Cara pemanggilan yang kedua ini digunakan untuk memanggil file header yang dibuat oleh programmernya sendiri atau file header yang bukan file bawaan dari aplikasi Dev-C++. Tapi tidak akan mempengaruhi jalannya program apabila kita menggunakan cara yang kedua ini untuk memanggil file header bawaan.

Sebagai contoh dalam bahasa c kita menggunakan file header stdio.h. file header ini dipanggil apabila seorang programmer ingin menggunakan fungsi printf dan scanf. Dalam bahasa c++ kita memanggil file header iostream.h agar kita bisa menggunakan fungsi cout dan cin (baca:si in). apabila ingin menggunakan fungsi-fungsi yang belum ada di file header yang digunakan maka kita perlu untuk memanggil file header lain yang mengandung fungsi yang kita gunakan.

C++ Klasik dan C++ Modern

Pada sub bab terakhir ini akan dibahas perbedaan antara c++ klasik dengan c++ modern. Perbedaan yang paling tampak adalah pada kompiler c++ lama masih menggunakan namespace global, sedangkan untuk c++ modern yang digunakan adalah namespace std. Di Dev-C++ yang digunakan adalah bentuk c++ modern. Untuk lebih jelasnya kita bisa lihat dalam penggalan kerangka bahasa c++ klasik di bawah ini:

```
1  # include <iostream.h>
2
3  int main(int argc, char argc[]){
```

```
4       ...
5       return 0;}
```

Selanjutnya perhatikan untuk bahasa c++ modern di bawah ini:

```
1   # include <iostream>
2
3   using namespace std;
4
5   int main(int argc, char argc[])
6   {
7       ...
8       return 0;
9   }
```

Kalau diperhatikan, bahasa c++ modern tidak mengakhiri file headernya dengan ekstensi .h dan juga menambahkan satu baris perintah yaitu **using namespace std**. hal ini dilakukan agar kita tidak perlu lagi untuk membubuhkan kata std::<nama_fungsi> pada setiap fungsi yang akan digunakan. Selebihnya dalam penulisan fungsi dan lain-lain tidak mengandung perbedaan yang berarti.