

Mukadimah

1.1 Pengenalan Bahasa C++

Bahasa C merupakan bahasa pemrograman tingkat menengah. Pada tahun 1972 bahasa C pertama kali dirancang oleh [Dennis M Ritchie](#) di Bell laboratories. Kemudian tahun 1978 Dennis dan [Brian W. Kernighan](#) mempublikasikan bahasa C melalui [The C Programming Language](#) sehingga bahasa C dikenal banyak orang. Selanjutnya pada tahun 1989, akhirnya bahasa C distandardisasi [ANSI](#) (American National Standard Institute) sehingga bahasa C menjadi bahasa pemrograman standar hingga saat ini dan bisa dibuat kompilernya pada beberapa platform yang berbeda-beda.

Bahasa C dikatakan sebagai bahasa pemrograman terstruktur, fungsional karena strukturnya menggunakan fungsi-fungsi sebagai program-program bagian (subroutine/ module). Fungsifungsi selain fungsi utama disebut subroutine/ module dan ditulis setelah fungsi utama (main) atau diletakkan pada file pustaka (library). Jika fungsi-fungsi diletakkan pada file pustaka dan akan dipakai disuatu program, maka nama file header-nya harus dilibatkan dalam program menggunakan preprocessor directive `#include`.

Kemudian bahasa C dikembangkan oleh Bjarne Stroustrup at Bell Labs menjadi bahasa C++. Pada bulan Oktober 1985 munculah buku *The C++ Programming Language* yang membahas tentang bahasa pemrograman itu langsung dari penciptanya sendiri. Bahasa C++ mengalami dua tahap evolusi.

- Pertama, dirilis oleh [AT&T Laboratories](#), dinamakan [cfront](#). C++ versi ini hanya berupa kompiler yang menterjemahkan bahasa C++ menjadi bahasa C untuk dieksekusi
- Kedua, [Borland International Inc.](#) mengembangkan kompiler C++ menjadi sebuah kompiler yang mampu mengubah C++ langsung menjadi bahasa mesin (assembly). Tahun 1990, C++ mulai diarahkan ke pengembangan [Pemrograman Berorientasi Obyek](#).

Beberapa keunggulan C++ dibandingkan dengan bahasa C adalah sebagai berikut.

Object-oriented programming

Bahasa pemrograman ini sangat mendukung pemrograman berorientasi obyek yang melihat permasalahan secara obyek dan bukan prosedural.

Portability

Kita dapat mengkompilasi C++ kode yang sama di hampir semua jenis komputer dan sistem operasi tanpa membuat perubahan apapun. C++ adalah bahasa pemrograman yang paling sering digunakan di dunia.

Brevity

Karena bahasa C++ merupakan bahasa tingkat tinggi, maka bahasa yang ditulis dengan bahasa C++ termasuk ringkas dan pendek dibandingkan bahasa-bahasa sejamannya pada waktu itu. Bahasa C++ termasuk bahasa pemrograman tua yang sudah mendukung berbagai macam kata kunci yang mampu menyingkat proses penulisan kode program.

Modular programming

Tubuh program pada bahasa C++ dapat terdiri dari beberapa file source code yang disusun secara terpisah dan kemudian dihubungkan secara bersama-sama. Kemampuan ini jelas menghemat waktu karena tidak perlu mengkompilasi ulang aplikasi yang lengkap ketika membuat satu perubahan, tetapi hanya file yang berisi perubahan itu saja. Selain itu, karakteristik ini memungkinkan kita untuk menghubungkan kode C++ dengan kode yang dibuat oleh bahasa lain, seperti bahasa Assembly dan C dan dapat digunakan kembali (reuseable).

C Compatibility

C++ sangat backward compatible dengan bahasa C, sehingga aplikasi / kode program yang ditulis dengan bahasa C dapat digabungkan dengan bahasa C++ dengan sangat mudah, bahkan hampir tidak perlu mengubah kodenya.

Speed

Kode yang dihasilkan dari kompilasi C++ sangat efisien, karena C++ mendukung prinsip dualitas bahwa dia mendukung bahasa tingkat tinggi dan bahasa tingkat rendah sehingga dapat mengurangi ukuran hasil kompilasi dari bahasa itu.

1.2 Pengantar Qt Creator

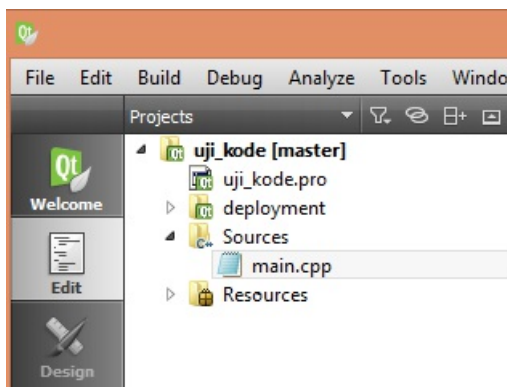
Qt Creator merupakan cross-platform C++ integrated development environment yang merupakan bagian dari Qt SDK. Qt Creator mempunyai debugger dalam bentuk visual dan layout GUI serta tempat perancangan form. Teks editornya mempunyai fasilitas syntax highlighting dan autocompletion. Qt Creator menggunakan compiler C++ dari kumpulan compiler GNU pada Linux dan FreeBSD. Pada Windows Qt Creator dapat menggunakan MinGW¹ atau [MSVC](#) yang sudah build-in di dalam install.

bahasa C agar dapat dipahami oleh bahasa mesin (assembler) pada komputer. Aplikasi ini dapat di unduh secara gratis.

Project Qt Creator menggunakan format cross platform project (.pro) untuk mengizinkan tim developer untuk share project yang mempunyai platform-platform yang berbeda-beda dan menggunakan common tool untuk implementasi dan debugging program. Sebuah project dapat meliputi:

file-file yang digroup secara bersama-sama, langkah-langkah build program, form-form dan file-file resource, dan pengaturan untuk menjalankan aplikasi.

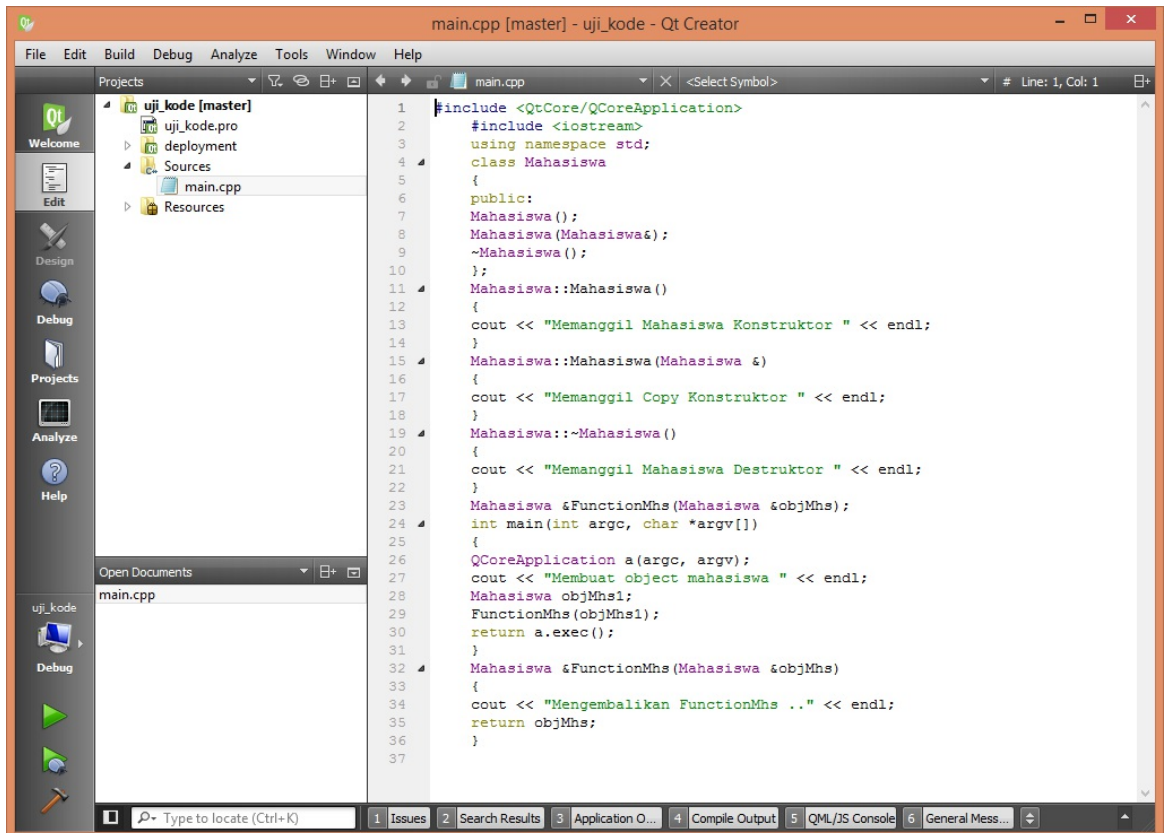
Projek dapat dibuat secara manual atau diimport dari file projek yang sudah ada. Jika projeknya dibuat secara manual, maka sebuah file-file akan digenerate oleh Qt Creator, tergantung dari tipe file yang dimiliki. Seperti Jika filenya adalah sebuah GUI application, maka Qt Creator men-generate sebuah file kosong yang berektensi `.ui` yang akan dimodifikasikan melalui Qt Designer. Qt Creator diintegrasikan dengan sistem cross-platform untuk mem-build secara otomatis: qmake dan CMake. Projek yang tidak menggunakan qmake atau CMake dapat diimport-kan, dan Qt Creator dapat meng-ignore sistem build.



Editor Qt Creator mempunyai sebuah code editor yang telah terintegrasi dengan Qt Designer untuk mendesain dan membangun aplikasi GUI dari Qt widgets. Karena Qt Creator adalah sebuah Integrated Development Enviroment (IDE), maka Qt Creator memisahkan antara text editor untuk build dan editor untuk menjalankan (run) aplikasi-aplikasi. Qt Creator bukan hanya bisa membaca text file biasa, akan tetapi juga bisa membaca file C++ dan bahasa QML.

Keunggulan Code Editor Qt Creator

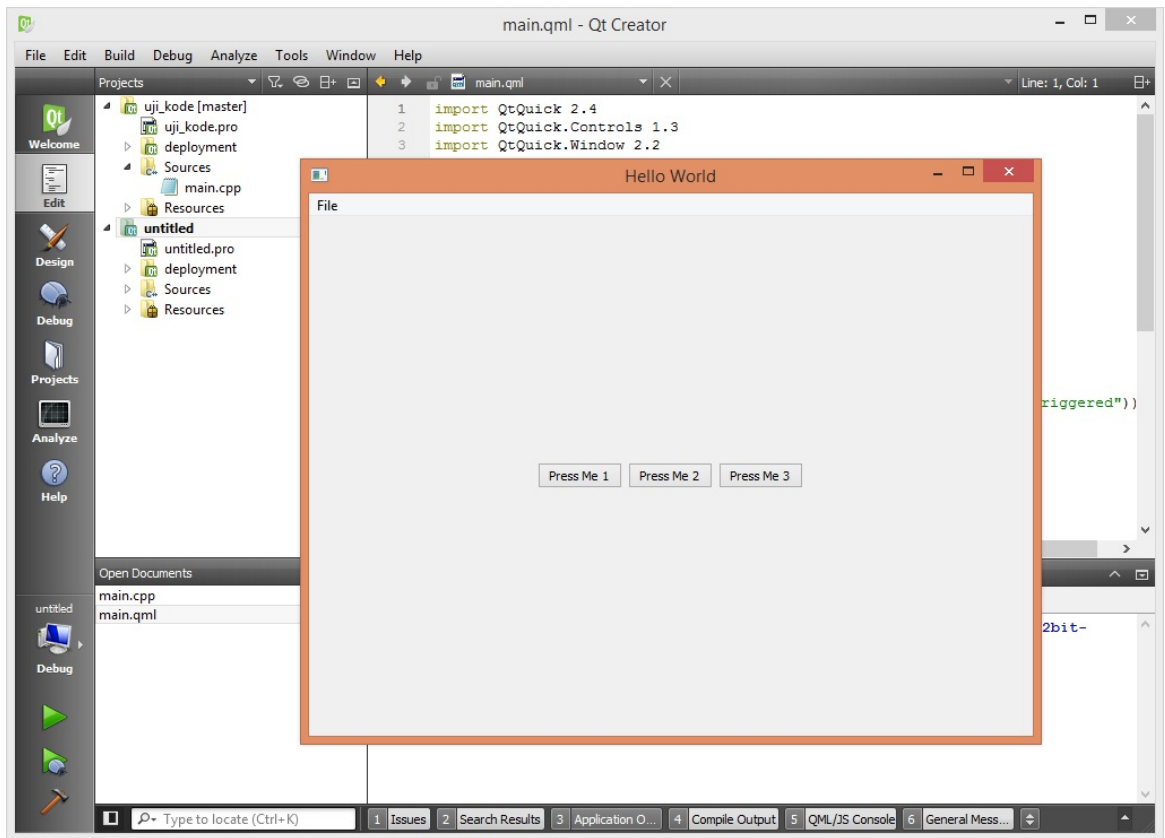
- Dapat menulis code dengan format yang benar.
- Mengantisipasi apa yang akan programmer tulis dan code yang komplit.
- Menampilkan baris-baris yang error dan pesan-pesan warning.
- Memandu programmer secara semantik untuk menulis classes, functions, dan symbols.
- Menyediakan fasilitas bantuan context-sensitive pada classes, functions, dan symbols.
- Me-rename symbol-symbol dengan langkah intelligent, sehingga simbol-simbol yang lain dengan nama yang sama tidak ter-rename.
- Menampilkan lokasi function, class yang dideklarasikan atau yang dipanggil



UI Designer

Qt Creator menyajikan dua buah editor visual: Qt Designer dan Qt Quick Designer. Qt Designer merupakan sebuah tool untuk mendesain dan membangun aplikasi GUI dari Qt widgets. Widgets dan forms yang dibentuk dengan Qt Designer terintegrasi dengan code program, Qt signals dan mekanisme slots, sehingga kita dengan mudah memberikan nilai-nilai dan properti-properti pada elemen-elemen grafik. Semua properti-properti yang diatur pada Qt Designer dapat diubah secara dinamik melalui/di dalam code.

Qt Quick Designer digunakan untuk membangun secara mudah animasi-animasi dengan menggunakan sebuah bahasa pemrograman yang dikenal dengan [Qt Meta-Object Language \(QML\)](#). Dalam QML, sebuah user interface dispesifikasikan sebagai sebuah pohon (tree) dari objects dengan properti-properti. Kamu menggunakan teks editor visual untuk menciptakan items, screens, dan aplikasi, serta mendefinisikan perubahan action-action pada komponennya. Dapat digunakan Qt atau JavaScript untuk mengimplementasikan logik aplikasi.



Bahasa yang di dukung

Anda dapat menggunakan code editor menulis code dalam Qt C++ atau bahasa pemrograman QML, javascript bahkan HTML5. Syntax highlighting juga disajikan untuk banyak bahasa pemrograman yang lain.

Platform

Qt Creator men-support untuk membangun dan menjalankan aplikasi-aplikasi Qt untuk desktop environments (Seperti Windows, Linux, FreeBSD dan Mac OS) Selain itu juga bisa dijalankan pada mobile devices (seperti Android, windows 8 dan iOS). Ketika sebuah aplikasi dibangun untuk mobile device yang bisa mengkoneksi ke Personal Computer (PC), maka Qt Creator men-generate sebuah package instalasi, menginstall package tersebut pada device, dan menemukuskannya.

Tools

Qt Creator diintegrasikan dengan kumpulan tool-tool yang bermanfaat dan membantu, seperti version control systems dan Qt Simulator. Qt Creator menggunakan command line client version control system untuk mengakses repositories ([Git](#), [subversion](#), [Perforce](#), [CVS](#), [Mercurial](#)).

Debuggers

Qt Creator tidak mempunyai debugger. Qt Creator mempunyai plugin debugger yang bekerja sebagai interface antara Qt Creator core dan external native debuggers

Debuggers adalah:

- GNU Symbolic Debugger (gdb)
- Microsoft Console Debugger (CDB)

- internal Java Script debugger

Dapat menghubungkan mobile devices dengan PC dan memproses debug yang berjalan pada devices.

1.3 Mengetahui macam - macam Teknologi User Interface (UI) pada QT

Interaksi antara pengguna dengan logic software dinamakan User Interface disingkat dengan UI. UI ini berwujud bisa sebuah window, bisa tombol, bisa sebuah text area dan lain sebagainya. Inilah komponen User Interface. Sebagai seorang programmer (pembuat program aplikasi), terlebih programmer yang menggunakan Qt, maka anda akan disediakan beragam UI yang bisa anda pilih sesuai kebutuhan. Ada QtWidget, QtQuick dan QtWebKit. Ketiganya dapat anda pilih sesuai kebutuhan sebagai UI program anda. Ingin tahu bedanya? Mari kita simak.

Sama seperti Visual Studio yang menyediakan beragam Amazing User Interface tingkat tinggi sampai pengguna bingung memilihnya, Qt menyediakan tiga UI yang dapat kita gunakan. Anda pun bisa menggabungkan ketiganya. :)

Qt Creator, sebuah Editor Qt adalah contoh dari perpaduan multiple teknologi UI ini. Coba amati Qt Creator yang selalu anda gunakan tersebut.

Qt Creator menggunakan teknologi QtWidget sebagai User Interface pada menu dan kotak dialog nya. Coba perhatikan kembali Welcome Screen dari Qt Creator, lihat, tampilannya berbeda bukan, bahkan button nya sangat berbeda dan tidak biasa, ini karena UI untuk Welcome Screen menggunakan teknologi QtQuick. Lalu dimana letak penggunaan QtWebKit?

Yup, perhatikan Help Documentation nya, wow, ini seperti halaman web yang menyatu dengan software nya bukan? Ya, teknologi QtWebKit digunakan dalam pembangunan Help Documentation ini.

A. Teknologi User Interface QtQuick

QtQuick merupakan salah satu Teknologi UI dari Qt yang menggunakan QML dan JavaScript sebagai penyusun UI nya. Mirip seperti XAML yang dikembangkan Microsoft, QML merupakan teknologi binding dari Qt yang memfasilitasi pengguna dengan visual canvas dan rendering engine nya. Teknologi UI ini sangat cocok sekali untuk Hardware Acceleration seperti OpenGL pada VGA driver kita.

Jangan salah, bila anda menggunakan QtQuick versi 2, maka memang butuh OpenGL yang disupport oleh VGA anda. OpenGL sekarang begitu terkenal, banyak games – games yang mulai menargetkan OpenGL karena begitu flexibel dan mudah. Tapi sayangnya, OpenGL ini tidak terdapat pada VGA driver bawaan OS.

Jadi jangan heran, saat anda install ulang komputer (bahkan Windows 8.1 sekalipun) anda akan menemukan bahwa tidak ada OpenGL pada VGA driver anda. Cara terbaik adalah periksa Motherboard anda dan cari driver VGA yang sesuai dengan Motherboard anda. Biasanya gratis.

Animation, Transition, Visual Effect, Shader Effect dan lain – lain merupakan fasilitas yang dapat anda kembangkan saat menggunakan QtQuick sebagai User Interface (UI) aplikasi anda.

B. Teknologi User Interface QtWidget

QtWidget merupakan tradisional User Interface element yang biasanya terdapat dalam desktop environment. Bila anda pengguna linux, maka UI ini merupakan bagian dari KDE. Tapi jangan salah, QtWidget sangat dinamis untuk Windows dan Mac OS. Sehingga bila anda menggunakan QtWidget sebagai UI anda maka tampilannya mirip sekali dengan UI pada OS anda.

Bila anda pengguna Windows 8.1 seperti saya dengan Amazing Flat Designnya, maka QtWidget ini menyesuaikan

dengan UI OS.

Semua standar komponen untuk aplikasi seperti button, textarea, menu dan lain – lain terdapat pada QWidget ini. Sehingga sangat cocok sekali untuk anda yang gemar membuat aplikasi tradisional standar.

Bila kita membuat aplikasi dengan QWidget, maka saat memulai project, akan muncul pemilihan Base Class, ada tiga yaitu Class QWidget, Class QMainWindow, dan Class QDialog.

Perbedaan dari ketiga Base Class di atas adalah berikut ini:

- `*QWidget` merupakan base class untuk semua GUI element pada QWidget User Interface. Coba lah eksplorasi dan bedakan ketiganya :)
- `*QDialog` merupakan sebuah window yang biasanya digunakan untuk mengejutkan pengguna, seperti saat window dialog muncul ketika pengguna harus memasukan input dengan benar atau hal – hal yang lain, tampilan dari QDialog tidak berbeda dengan QWidget, anda bisa menggunakan salah satu.
- `*QMainWindow`, nah, ini adalah sebuah class yang sangat unik, karena menggunakan feature *built in* yang sangat populer seperti status bar, toolbar, dan menu bar. Cobalah membuat aplikasi QWidget dengan QMainWindow sebagai base class nya, pasti secara otomatis akan ditampilkan menubar, toolbar, dan statusbar.

C.Teknologi User Interface QtWebkit

Tahukah anda bahwa web programing adalah kegiatan yang paling berkembang di dunia saat ini? Tahukah anda bahwa web koding seperti html, css, js sangat populer dan mudah digunakan dan mulai merambah ke teknologi desktop seperti Html5 dan Css3?

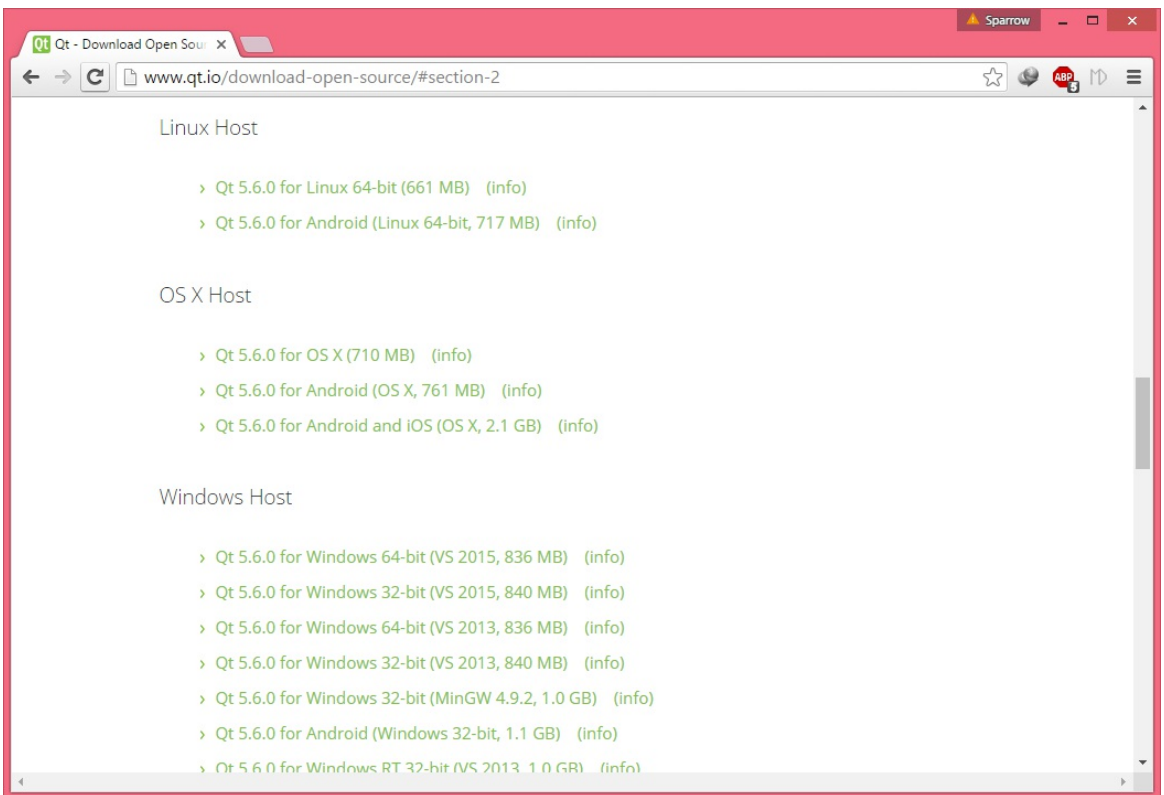
Lalu kenapa tidak digunakan dalam pemrograman desktop? Ya, dengan menggunakan User Interface QtWebkit ini anda bisa membuat sebuah desktop application dengan menggunakan koding web. Unik bukan?

Teknologi QtWebkit menampilkan web content melalui QML, sedangkan C++ API digunakan untuk interaksi dengan web content tersebut.

Perlu diperhatikan bersama bahwa pemilihan teknologi adalah biasa, so, tetaplah berkreasi, berikut kita kutipkan beberapa perbandingan antar tiga teknologi UI dari Qt Help Documentation.

1.4 Install Qt Creator

Pada tutorial ini kita akan menginstall Qt pada ubuntu 14.04 atau Windows 8 dengan menggunakan versi terbaru dari Qt Creator yang dapat di unduh di [halaman](#) Qt cCreator.



A. Install Qt Creator di Ubuntu 14.04

1. Download

Kunjungi website Qt untuk mendownload Qt Crator sesuai dengan versi sistem operasi yang digunakan baik itu 64-bit atau 32 bit. atau juga dapat di download dengan menggunakan command line di linux dengan mengetikan.

Contoh:

```
wget http://download.qt.io/official_releases/online_installers/qt-unified-linux-x86-online.run
```

jika menggunakan sistem operasi berbasis 64 bit

```
wget http://download.qt.io/official_releases/qt/5.6/5.6.0/qt-opensource-linux-x64-5.6.0.run
```

2. Install

Atur permissi, jalankan installer dan ikuti perintah berikut ini untuk menginstall Qt Creator secara lengkap.

```
chmod +x qt-opensource-linux-x64-5.6.0.run  
./qt-opensource-linux-x64-5.6.0.run
```

3. Install g++

Buka terminal untuk menginstal g++:

```
sudo apt-get install build-essential
```

4. Konfigurasi Kompiler

Buka Qt Creator klik tool > Options. Klik build & run dan pilih tab Kit. Konfigurasi kompilasi jika belum terdeteksi secara otomatis.

5. Install Pustaka OpenGL

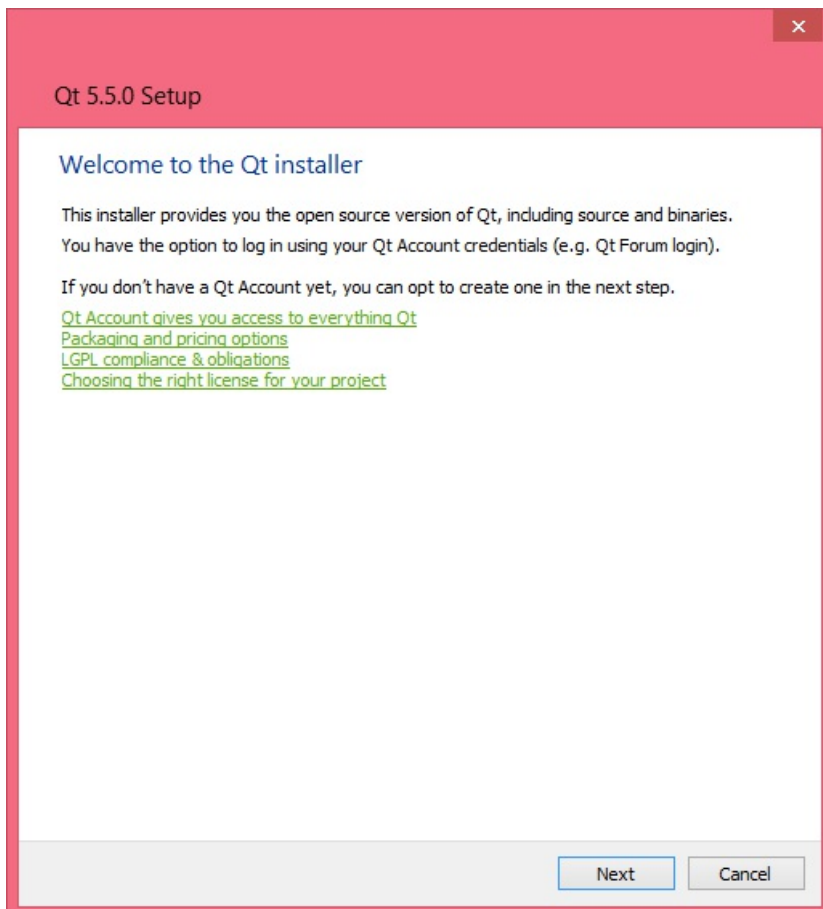
Jalankan Perintah berikut ini menginstall Pustaka OpenGL:

```
sudo apt-get install mesa-common-dev
```

Install Qt di Windows

Anda bisa mendownload di halaman download Qt creator dan memilih versi dari Aplikasi yang Anda butuhkan baik 64 bit atau 32 bit. Sesuaikan dengan sistem operasi yang Anda miliki.

1. Jika Anda telah mendownload Qt creator maka qt-opensource-windows-x86-mingw492-5.5.0.exe. Disini penulis menggunakan versi 32 bit jika sistem operasi Anda 64 bit maka gunakanlah 64 bit walaupun dapat menggunakan versi 32 bit.



2. Pilih **Next** dan akan muncul halaman Qt Account jika anda tidak ingin mendaftarkan diri dapat di lewati dengan memilih **skip**.

×

← Qt 5.5.0 Setup

Qt Account – Your unified login to everything Qt

Please log in to Qt Account

Login

[Forgot password?](#)

Need a Qt Account?

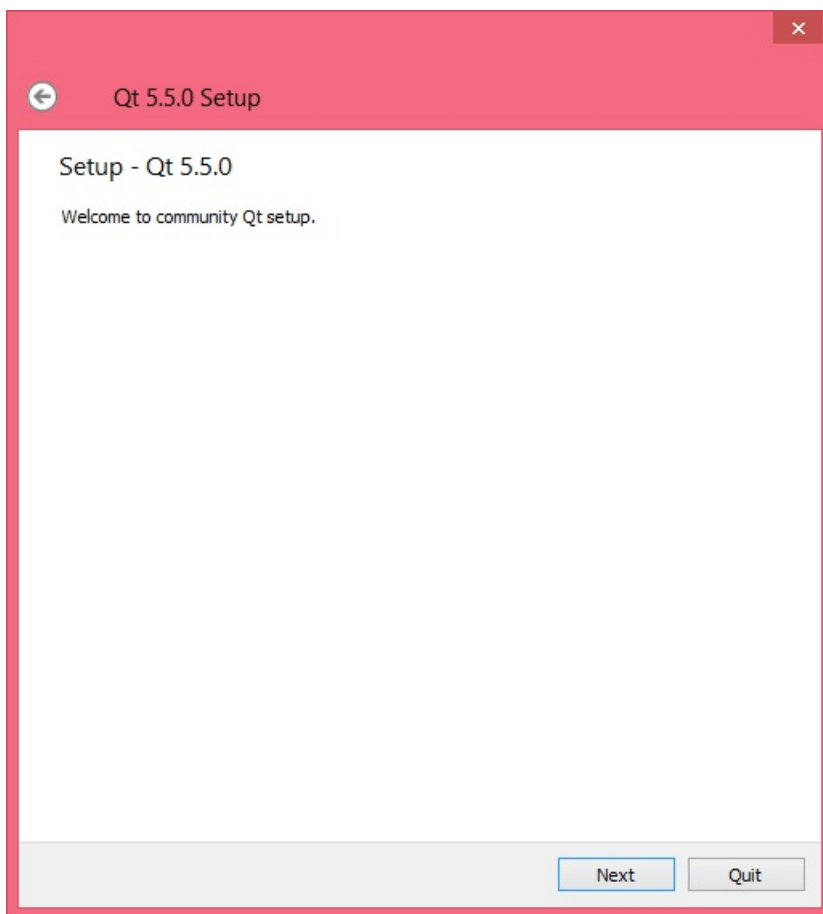
Sign-up

☐ I accept the [service terms](#).

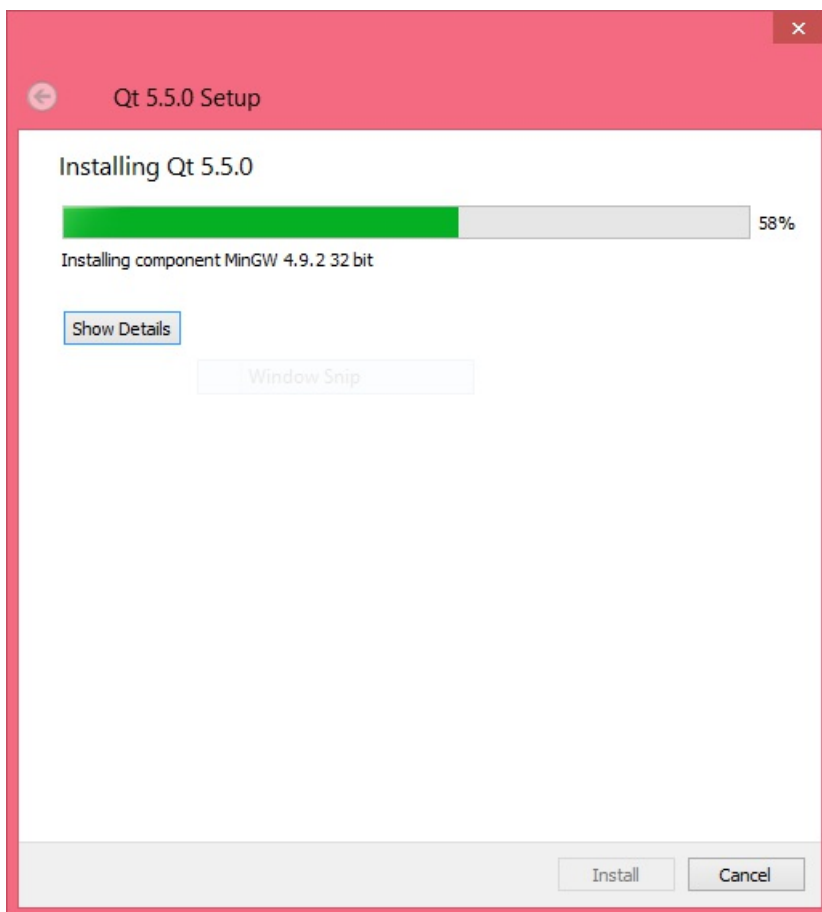
Skip

Cancel

3. Masuk ke halaman Setup terus **next** saja.

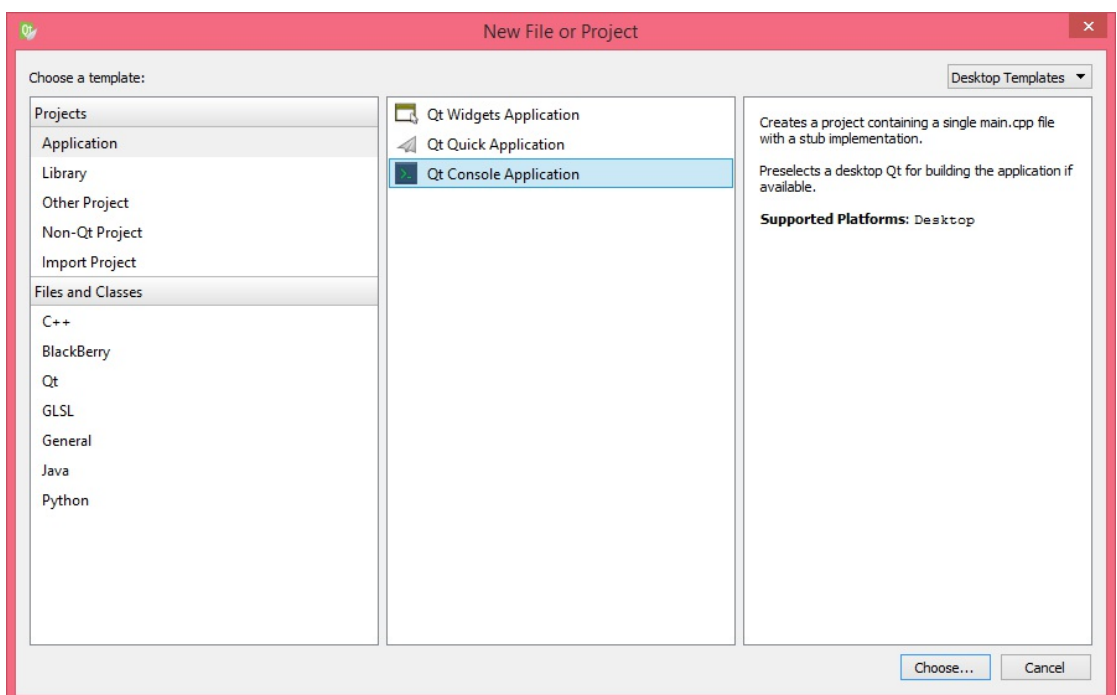


4. Installer akan menginstall aplikasi sampai selesai apabila telah selesai maka klik finish untuk mengakhiri proses pemasangan aplikasi.

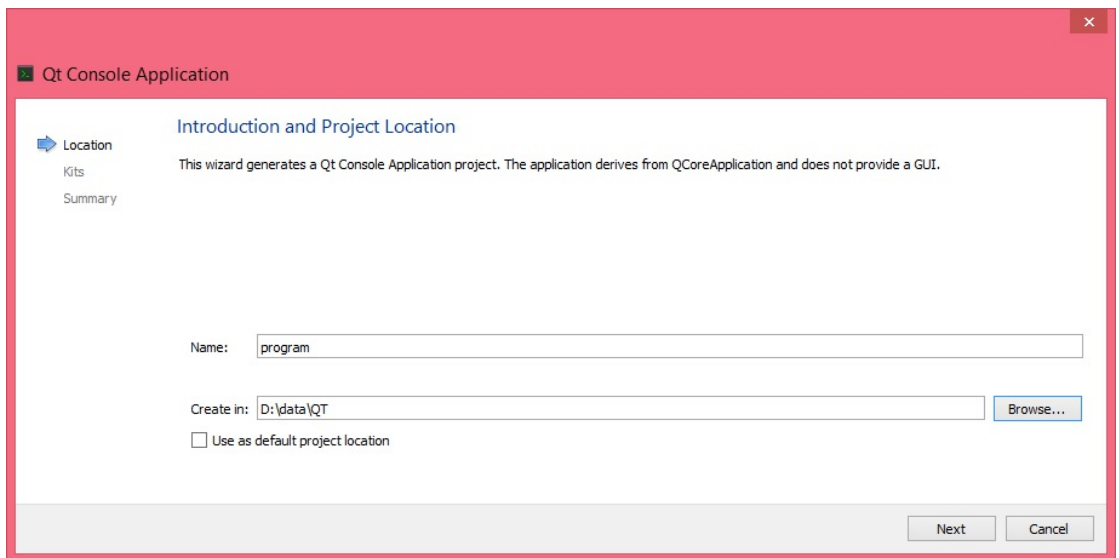


1.5 Program Console Pertama dengan Qt Creator

1. Untuk mencoba membuat aplikasi dengan Qt Creator maka kita perlu dengan membuat menu file > new Project dan pilih project application > Qt console application



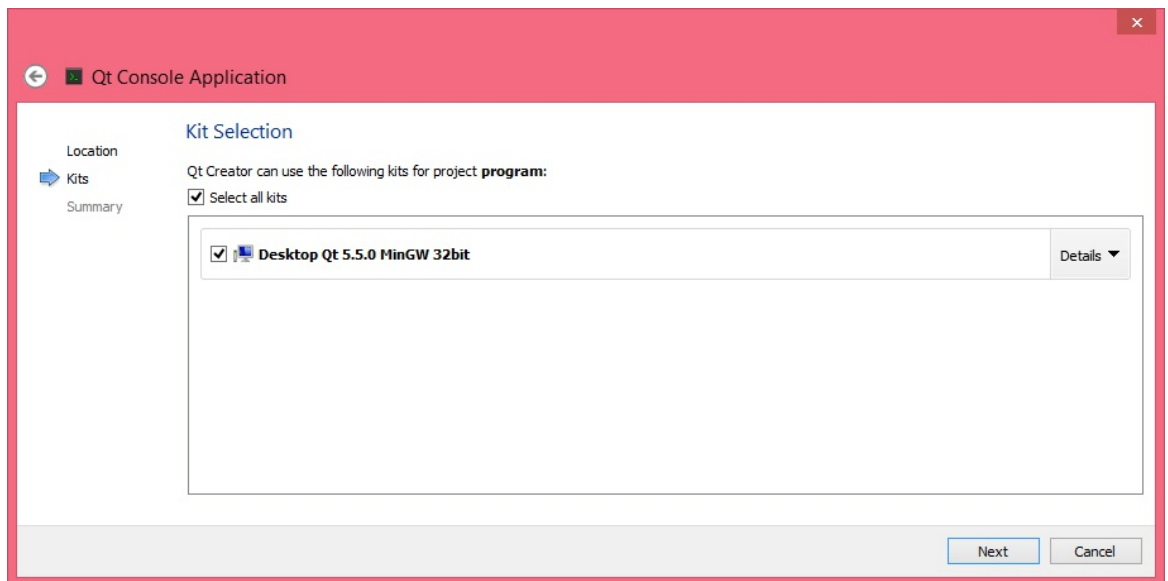
2. kemudian beri nama dengan Program yang akan kita buat dan direktori tempat aplikasi yang kita buat.



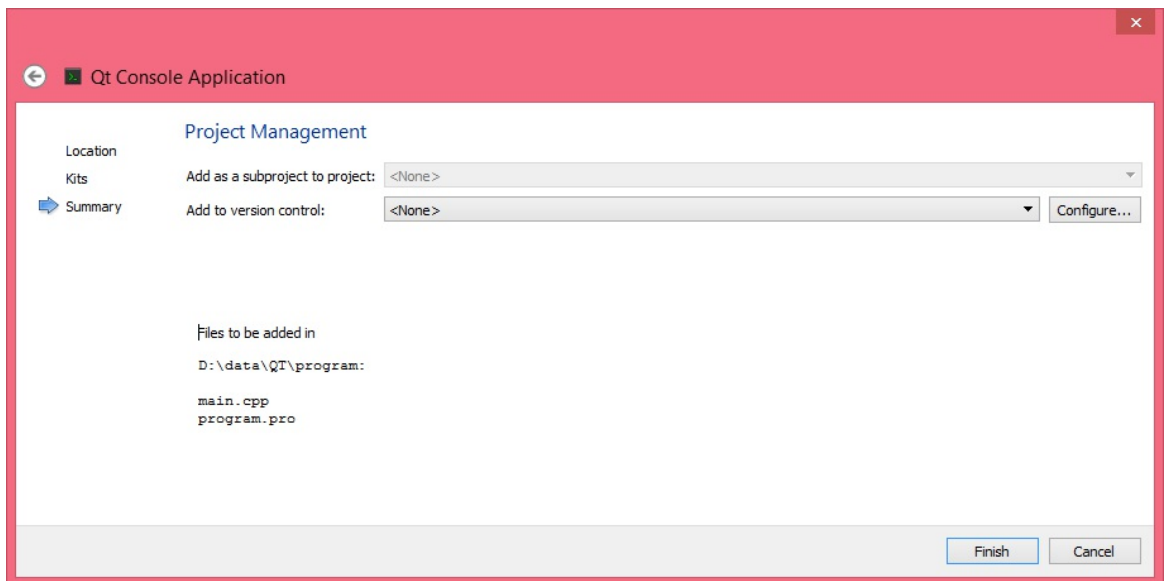
3. Klik Next, kemudian pilih compiler yang akan kita gunakan. Disini penulis menggunakan MinGW sebagai compilernya.

T> **Tips** T> T> Simulator atau compiler yang lengkap terdiri dari

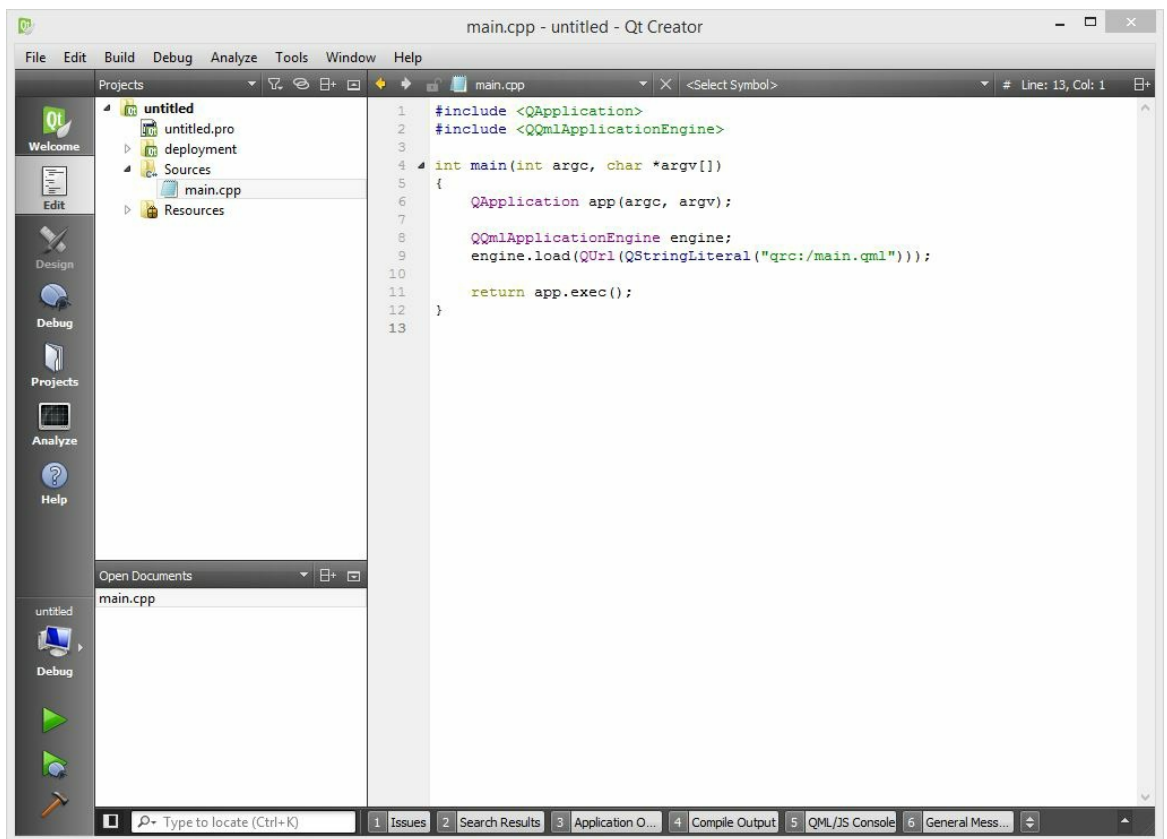
- Qt Simulator MingGW 4.4
- Qt Simulator VS 2008, 2010, 2011, 2012 2013, 2014
- Android SDK dan NDK



1. Kemudian pilih jenis sub version yang akan kita gunakan, jika Anda tidak menggunakan sub version maka pilih none pada add to subversion.



1. Apabila di lakukan dengan benar maka akan muncul Qt Editor sebagai berikut ini.



Program Bahasa C/C++ tidak mengenal aturan penulisan di kolom/baris tertentu, jadi bisa dimulai dari kolom/baris manapun. Namun demikian, untuk mempermudah pembacaan program dan untuk keperluan dokumentasi, sebaiknya penulisan program di bahasa C/C++ diatur sedemikian rupa sehingga mudah dan enak dibaca. Berikut adalah struktur dasar program yang dibuat dengan bahasa C++:

```
#include <header>
using namespace std;
int main(int argc, char *argv[])
{
    deklarasi variabel;
    deklarasi konstanta;
    perintah âperintah;
    //komentar
    return 0;
}
```

Penjelasan :

1. #include

`#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_file>
```

Bagian tersebut tidak diakhiri dengan tanda semicolon, karena bentuk tersebut bukanlah suatu bentuk pernyataan, tetapi merupakan preprocessor directive. Baris tersebut menginstruksikan kepada kompiler untuk menyisipkan file lain dalam hal ini file yang berakhiran .h (file header) yaitu file yang berisi C++ standard library. Pada C++ ekstensi .h tidak dituliskan.

Beberapa contoh pengikutsertaan berkas adalah:

- `#include <iostream>` : diperlukan pada program yang melibatkan objek `cout` dan `cin`
- `#include <conio>` : diperlukan bila melibatkan `clrscr()` , yaitu perintah untuk membersihkan layar dan fungsi `getch()` untuk menerima sembarang input keyboard dari user.
- `#include <iomanip>` : diperlukan bila melibatkan `setw()` yang bermanfaat untuk mengatur lebar dari suatu tampilan data.
- `#include <math>` : diperlukan pada program yang menggunakan operasi `sqrt()` yang bermanfaat untuk operasi matematika kuadrat.

2. using namespace std;

Semua elemen standard C++ library dinyatakan dalam apa yang disebut namespace, namespace tersebut bernama std. Jadi artinya untuk mengakses semua fungsionalitas std kita menuliskan bahwa kita menggunakan namespace std.

3. int main ()

Program C++ terdiri dari satu atau lebih fungsi, dan di antara salah satunya harus ada fungsi main dan hanya boleh ada satu main pada tiap program C++. Setiap program C++ akan dan pasti akan memulai eksekusi programnya pada fungsi main ini, meskipun main bukan fungsi yang pertama ditulis di program. 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.

Tanda { dan pada akhir program terdapat tanda }. Tanda { harus ada pada setiap awal dari sebuah fungsi dan tentu saja harus diakhiri dengan tanda }. Tanda ini digunakan untuk menunjukkan cakupan(scope) dari sebuah fungsi, dimana untuk menunjukkan fungsi ini dimulai dan berakhir.

4. Komentar

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

1. */ Komentar anda diletakkan di dalam ini bisa mengapit lebih dari satu baris /*
2. *// Komentar anda diletakkan disini (hanya bisa sebaris)*

Programmer sering sekali memasukkan komentar di dalam code agar program lebih mudah dibaca. Komentar juga membantu orang lain untuk membaca dan mengerti isi dari code. Komentar tidak menyebabkan komputer melakukan suatu instruksi ketika program dijalankan.

5. Tanda Semicolon (;)

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

9. return 0

Pernyataan return menyebabkan fungsi utama untuk menyelesaikan kegiatannya lalu mengembalikan hasil dari fungsi utama. Kode kembalian biasanya angka 0 atau 1. Jika angka yang dikembalikan 0 berarti program berakhir dengan tidak ada error, sedangkan jika 1 maka program berakhir dengan error.

Contoh 1. Struktur program C++

Untuk lebih jelasnya silahkan coba ketik program berikut pada project baru.

```
#include <QtCore/QCoreApplication>
#include <iostream>
using namespace std;
int main (int argc, char *argv [])
{
    QCoreApplication a (argc, argv);
    cout<<"Hello World"<<endl;
    cout<<"Selamat Belajar C/C++ ";
    cout<<"enter my World";
    return a.exec ();
}
```

Kemudian jalankan dengan menekan tombol Run (CTRL + R)

A> {linenos=off} A> Hello world A> A> Selamat belajar C/C++ enter my world

Tampilan Hello World diakhiri dengan tanda enter baru kemudian dilanjutkan dengan tulisan berikutnya yaitu Selamat Belajar C/C++ enter my World. Artinya perintah `endl` merupakan perintah untuk memberi tanda enter. Sedangkan untuk tulisan Selamat Belajar C/C++ dan tulisan enter my World yang pada source code terpisah dengan perintah `cout`, pada tampilan hasil program tetap sama dan tidak ada enter diantaranya. Hal ini karena tidak ada perintah untuk menampilkan enter diantara kedua kalimat tersebut. Penulisan pada kode tidak akan mempengaruhi hasil output program.

-
1. minGW adalah salah satu aplikasi yng digunakan untuk mengkompile↵