

APLIKASI VISUAL INTEGRATED DEVELOPMENT ENVIRONMENT UNTUK FRAMEWORK QT

Priyo Hutomo¹, Dhinta Darmantoro², Dana Suliyo Kusumo³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

QT adalah framework pemrograman yang bersifat cross-platform untuk bahasa pemrograman C++. Dengan menggunakan QT pengembang perangkat lunak dapat membangun aplikasi untuk platform Windows, Unix, dan Mac OS X dengan menggunakan file source yang sama. Satu hal yang membuat QT menarik, tidak seperti Java, aplikasi QT bersifat native, baik dari sisi tampilan, behaviour, maupun performansi. Dari sisi kelengkapan library yang tersedia, sebenarnya QT dapat disejajarkan dengan Java. Hanya saja karena belum tersedianya IDE yang didesain khusus untuk QT menyebabkan framework ini cukup sulit digunakan, terutama oleh pengembang yang berasal dari lingkungan Windows. Tugas akhir ini bertujuan untuk membuat aplikasi Visual IDE untuk membantu pengembang perangkat lunak membuat aplikasi berbasis QT. Visual IDE dibangun dengan menggunakan framework VCL dari Borland. Sebagaimana layaknya sebuah IDE, Visual IDE yang dibangun memiliki fungsi dasar dalam pembuatan program, seperti: fungsi untuk mengedit source code, mendesain tampilan GUI dan melakukan proses kompilasi secara otomatis. Walaupun Visual IDE harus disadari masih mempunyai beberapa kelemahan, akan tetapi terlepas dari semua kelemahan yang ada Visual IDE telah dapat digunakan untuk membuat program berbasis QT. Hal ini dibuktikan pada uji coba yang telah dilakukan, yaitu membuat aplikasi database sederhana dengan menggunakan library SQLite3.

Kata Kunci : visual IDE, QT, cross-platform, compiler, debugger, SSH

Abstract

QT is a cross-platform programming framework for C++. With QT programmers can build program that run on Windows, Unix, an Mac OS X from single source without any modification. One interesting thing about QT is that the program developed with QT will have native look and feel. QT include a large number of libraries for variety of needs. The range and quality of the library can be directly compared to Java. Unfortunately there is no IDE that designed specifically for QT yet. Without an IDE, programmers, especially that come from Windows background, will have hard time to develop QT program. In my final assignment, I develop a Visual IDE to help programmers to develop QT program. Visual IDE is developed using VCL framework from Borland. Just like any other IDE, this Visual IDE include basic functions for editing the source code, designing GUI, and do compilation process automatically I have to admit that this Visual IDE still have several weakness, but part from all the weakness, the Visual IDE can already be used to develop a QT program. I have proved it on the test that the Visual IDE has been successfully used to develop a simple database program using SQLite3 library.

Keywords : visual IDE, QT, cross-platform, compiler, debugger, SSH

1. Pendahuluan

1.1 Latar belakang

Semenjak di-*release*-nya Windows 95, Microsoft secara meyakinkan berhasil menguasai pasar sistem operasi dunia. Hal ini tidak berubah sampai sekarang dengan masih digunakannya sistem operasi Windows di lebih dari 90% PC baru yang dijual di seluruh dunia. Dominasi Windows yang sangat luar biasa ini menyebabkan sebagian besar pengembang perangkat lunak komersial tidak merasa perlu membuat produknya dapat berjalan di sistem operasi lain. Cukup dengan mendukung Windows, produk yang dikembangkan sudah dapat menjangkau hampir 90% pengguna komputer di seluruh dunia. Selain itu membangun perangkat lunak yang bersifat *cross-platform* masih dianggap sangat rumit dan mahal oleh sebagian besar pengembang.

Walaupun dominasi Windows saat ini masih sangat kuat, tapi dalam satu atau dua tahun terakhir ini penggunaan sistem operasi non-Windows menunjukkan peningkatan yang cukup besar, terutama untuk sistem operasi Mac OS X dari Apple dan sistem operasi Linux. Mac OS X dengan tampilan yang sangat elegan telah berhasil memikat banyak pengguna Windows, terutama dari kalangan anak muda dan para pekerja seni. Hanya saja faktor harga perangkat keras yang relatif mahal, membuat Mac OS X hanya populer di kalangan terbatas. Jika Mac OS X lebih berkonsentrasi untuk merebut pasar di kelas premium, Linux lebih berkonsentrasi di kelas bawah. Dengan menawarkan sistem operasi yang dapat di-*download* secara gratis, Linux dengan cepat menarik banyak pengguna di negara-negara berkembang dan berbagai lembaga sosial dan penelitian. Ditambah lagi dengan adanya kebijakan dari pemerintah di berbagai negara ataupun kota yang mensyaratkan penggunaan sistem operasi Open Source, terutama Linux, di lingkungannya membuat sistem operasi ini menjadi salah satu platform yang harus diperhitungkan oleh para pengembang perangkat lunak.

Munculnya Mac OS X dan Linux sebagai penantang serius dari dominasi Windows, membuat masalah yang cukup pelik bagi para pengembang perangkat lunak karena sekarang mereka dituntut untuk juga menyediakan versi Mac OSX dan Linux dari produk mereka. Masalah utama dalam pengembangan aplikasi yang bersifat *cross-platform* adalah *framework* pemrograman dari setiap sistem operasi yang tidak saling compatible. Hal ini menyebabkan pengembang terpaksa harus secara khusus membangun aplikasi untuk setiap sistem operasi yang akan didukung yang tentunya akan membuat biaya pengembangan dan *maintenance* dari suatu produk menjadi jauh lebih mahal. Untuk mengatasi masalah tersebut, digunakanlah suatu *framework* yang bersifat netral. *framework* ini bekerja diatas *framework native* dari sistem operasi, sehingga dapat menyediakan *interface* yang bersifat netral atau tidak bergantung dengan sistem operasi tertentu.

QT adalah salah satu *framework* yang dikembangkan untuk tujuan tersebut. Dengan QT, pengembang dapat menghasilkan aplikasi yang bersifat native untuk beberapa sistem operasi dengan satu *source code* yang sama. Hanya saja dengan belum tersedianya IDE yang berbasis visual dan secara khusus mendukung QT, menyebabkan banyak pengembang perangkat lunak yang kesulitan untuk

mengadopsi QT. TrollTech, produsen dari QT, saat ini memang sudah menyediakan sebuah *designer* untuk merancang *User Interface* dan juga sudah menyediakan modul integrasi yang memungkinkan pengembang membangun aplikasi QT dengan Microsoft Visual Studio. Hanya saja, selain fakta bahwa Visual Studio tidak tersedia secara gratis, modul integrasi yang disediakan oleh TrollTech masih jauh dari sempurna dan belum terintegrasi dengan baik.

Mengamati kekurangan tersebut diatas kiranya diperlukan penelitian untuk membangun suatu aplikasi Visual IDE yang didesain khusus untuk framework QT. Berkaitan dengan itu, dalam Tugas Akhir ini dipilih topik untuk membangun aplikasi IDE berbasis visual untuk *framework* QT.

1.2 Perumusan masalah

Sampai saat ini belum tersedia Visual IDE yang secara khusus didesain untuk *framework* QT. Hal ini menyulitkan pengembang aplikasi yang berlatar belakang Windows ataupun Mac OS X yang terbiasa menggunakan IDE berbasis visual untuk menggunakan QT. Oleh karena itu dalam tugas akhir ini dilakukan sebuah penelitian untuk mengembangkan sebuah Visual IDE untuk *framework* QT.

1.3 Tujuan

Tujuan dari penelitian Tugas Akhir ini adalah untuk membangun sebuah aplikasi Visual IDE yang dapat memudahkan pengembang membangun perangkat lunak dengan menggunakan *framework* QT.

1.4 Batasan Masalah

- Visual IDE yang akan dikembangkan hanya dapat dijalankan di sistem operasi Windows.
- Visual IDE yang akan dikembangkan hanya akan mendukung fitur-fitur sebagai berikut:
 - (1) Editor
 - (a) *Syntax highlighting* untuk bahasa C/C++.
 - (b) *Search dan replace*.
 - (c) *Context help*.
 - (d) *Code Complete*.
 - (2) Designer
 - (a) *Form designer*.
 - (b) Property dan event editor.
 - (c) Mendukung penggunaan komponen.
 - (3) Compiler
 - (a) Mendukung multi-compiler (Visual C++ 2005, GCC, dan Intel C++).
 - (4) Debugger
 - (a) Mendukung GDB v5.2 atau versi yang lebih tinggi.
 - (b) Break Point.
 - (c) Stepping.

1.5 Metodologi penyelesaian masalah

Berikut ini adalah metodologi penyelesaian masalah yang dipergunakan dalam tugas akhir ini:

- Studi literatur yang bertujuan untuk mempelajari dasar teori mengenai
 - Teknik membangun aplikasi menggunakan *framework* QT.
 - Teknik pembuatan IDE.
 - Teknik pembuatan designer GUI.
- Studi perancangan perangkat lunak yang bertujuan untuk menentukan metodologi pengembangan perangkat lunak yang akan digunakan dan melakukan perancangan sistem.
- Membangun perangkat lunak berdasarkan rancangan yang telah dibuat.
- Pengujian perangkat lunak.
- Analisa terhadap hasil pengujian.
- Pengambilan kesimpulan dan penyusunan laporan.



5. Kesimpulan Dan Saran

5.1 Kesimpulan

Dari hasil desain, pembangunan aplikasi dan uji coba yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut :

- a. Aplikasi Visual IDE telah berhasil dibangun dengan menggunakan framework VCL dari Borland. Tidak digunakannya framework QT dalam pembangunan Visual IDE menyebabkan proses pembangunan Visual IDE menjadi lebih rumit karena dalam proses tersebut tidak dimungkinkan untuk secara langsung memanggil fungsi yang terdapat pada QT. Walaupun demikian metoda ini memastikan bahwa Visual IDE yang telah dibangun bersifat netral sehingga dapat dimodifikasi untuk mendukung *framework* pemrograman yang lain.
- b. Visual IDE yang dibangun telah dilengkapi dengan berbagai fitur untuk mendukung pembangunan program berbasis QT. Fitur-fitur tersebut diantaranya: source editor, GUI designer, compiler, debugger, context help, dan komponen-komponen dasar yang umum digunakan untuk mendesain user interface.
- c. File Source yang dihasilkan oleh Visual IDE dapat dikompilasi ulang di berbagai sistem operasi yang didukung oleh *framework* QT. Dalam ujicoba, file source dapat dikompilasi tanpa masalah di platform Fedora Core 7 dan Mac OSX.
- d. Pengembang pihak ketiga dapat meningkatkan kemampuan Visual IDE dengan cara menambahkan komponen baru. Proses penambahan komponen seperti yang telah diujicobakan dapat dilakukan dengan mudah melalui *wizard* yang tersedia. Dukungan untuk komponen pada Visual IDE masih memiliki kelemahan, yaitu belum mendukung komponen dengan format binary.
- e. Visual IDE yang telah dibangun, tidak dapat secara fleksibel menangani komponen yang membutuhkan GUI designer tersendiri, seperti main menu, *pop-up menu*, *tool-bar*. Hal ini disebabkan tidak adanya modul untuk menangani GUI designer external pada desain penanganan komponen external (pihak ketiga). Ketidak-fleksibelan ini menyebabkan pengembang pihak ketiga tidak dapat mengintegrasikan komponen yang memiliki GUI designer tersendiri ke Visual IDE.
- f. Compiler yang didukung oleh Visual IDE secara desain tidak terbatas. Secara prinsip, karena QT telah menyediakan metoda yang seragam untuk melakukan kompilasi, maka semua compiler yang didukung oleh QT, termasuk yang tidak secara khusus didukung oleh Visual IDE, dapat digunakan walaupun dengan sedikit keterbatasan. Pesan kesalahan dan peringatan dari compiler yang tidak secara khusus didukung tidak dapat dikenali oleh Visual IDE.
- g. Debugger yang digunakan, GDB, tidak dapat bekerja dengan baik pada Windows versi 64 bit. Masalah ini dapat diatasi dengan mengganti GDB

dengan Intel Debugger v10 atau lebih tinggi yang dijalankan pada mode GDB.

- h. Algoritma yang digunakan pada proses transfer file source saat melakukan kompilasi di *remote computer* belum optimal. Dua kelemahan utama yang perlu untuk diperbaiki, diantaranya: program mentransfer file source satu persatu secara sequensial tanpa melakukan proses kompresi terlebih dahulu dan tidak adanya proses pengecekan terhadap file mana saja yang mengalami perubahan mengharuskan Visual IDE untuk mentransfer semua file setiap akan melakukan kompilasi.
- i. Karena keterbatasan waktu fitur replace yang berfungsi untuk mengganti suatu kata atau kalimat secara otomatis, fitur stepping yang berfungsi untuk melakukan eksekusi per baris perintah, dan fitur code complete yang berfungsi untuk memberi saran tentang kata selanjutnya yang perlu ditulis oleh pengguna belum diimplementasikan. Sebaliknya fitur untuk melakukan kompilasi pada remote komputer karena dipandang perlu untuk membuktikan bahwa file source yang dihasilkan dapat dikompilasi pada berbagai platform dan installer untuk Visual IDE untuk memudahkan pengujian ditambahkan.
- j. Terlepas dari beberapa kelemahan yang ada Visual IDE yang telah dibangun sudah dapat digunakan untuk membuat program berbasis QT.

5.2 Saran

- a. Sebelum dapat digunakan secara komersial, Visual IDE yang telah dibangun masih memerlukan banyak perbaikan dan penambahan fitur-fitur. Fitur seperti *code-complete* dan modul untuk pengelolaan file resource sangat perlu dipertimbangkan untuk ditambahkan pada versi selanjutnya.
- b. Penggunaan framework yang juga bersifat cross-platform dalam pembangunan IDE juga perlu dipertimbangkan untuk versi selanjutnya. Dalam hal ini framework CLX, Wx-Widget, Java, ataupun QT sendiri dapat digunakan.
- c. Juga perlu dipertimbangkan untuk menggunakan metoda client-server pada proses kompilasi di *remote computer*. Dengan adanya program yang berfungsi sebagai server di remote computer proses kompilasi menjadi lebih mudah dan cepat karena beberapa tahapan kompilasi, terutama pada tahap mempersiapkan file-file *source* yang akan dikompilasi, dapat dilakukan oleh program server ini.

Daftar Pustaka

- [1] Blanchette, Jasmin, and Mark Summerfield, C++ GUI Programming with QT 3, Prentice Hall, 2004.
- [2] Dalheimer, Matthias K., Programming with QT, O'Reilly, 2002.
- [3] Hollingworth, Jarrot, Bob Swart, Mark Cashman, and Paul Gustavson, Borland C++ Builder 6 Developer's Guide, Sams 2002.
- [4] <http://qt4.digitalfanatics.org>
- [5] <http://www.gnu.org/software/gdb/gdb.html>
- [6] <http://www.mingw.org>
- [7] <http://www.trolltech.com>
- [8] Kolachina, Satya Sai, C++ Builder 6 Developer's Guide, Wordware Publishing, Inc 2003.
- [9] Mecklenburg, Robert, Managing Project with GNU Make, O'Reilly 2005.
- [10] Pappas, Chris H. and William H. Murray III, Visual C++ .NET: The Complete Reference, McGraw-Hill/Osborne, 2002.
- [11] Schildt, Herbert, C++: The Complete Reference, Fourth Edition, McGraw Hill/Osborne, 2003.
- [12] Stallman, Richard M, Eolan Pesch, and Stan Shebs, Debugging with GDB: The GNU Source-Level Debugger, GNU Press 2002.