

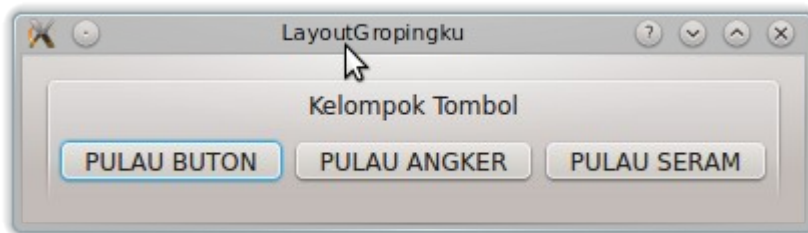
Pemrograman Qt – Membuat Dialog, GroupBox, Layout, dan Tombol dengan Menulis Kode (Bukan Designer)

Bismillahirrahmanirrahim.

Tulisan ini adalah bentuk PDF dari <http://malsasa.wordpress.com/2013/07/02/pemrograman-qt-membuat-dialog-groupbox-layout-dan-tombol-dengan-menulis-kode-bukan-designer/>.

Saya sebenarnya lebih suka cara drag and drop akan tetapi karena sudah kesulitan, maka cara terakhir adalah hard coding. Hard coding (begitu orang menyebutnya) adalah membuat aplikasi GUI tidak dengan drag and drop melainkan menulis kodenya semua dari nol. Kedengarannya mengerikan tetapi nyatanya tidak. Saya ingin mendemonstrasikan sedikit contoh hard coding (walaupun sebetulnya ini belum masuk hard level) untuk membuat satu dialog, di atasnya diberi layout, lalu di atas layout itu diberi tombol. Tetap kita lakukan ini dengan Qt Creator, dengan bahasa C++, dan di atas Ubuntu. Semoga ini bermanfaat.

Wujud Program yang Ingin Dibuat



Alur Kode

Saya nggak berani bilang ini algoritma. Namun inilah alur pikir sederhana dari program ini.

1. Membuat sebuah dialog (QDialog).
2. Mengisi dialog itu dengan groupbox (QGroupBox).
3. Mengisi groupbox dengan layout (QHBoxLayout).
4. Membuat tombol (QPushButton).
5. Memasukkan tombol itu ke dalam layout.
6. Otomatis, jika ada widget (entah button entah apa) masuk ke dalam layout, yang layout itu sendiri ada dalam groupbox, dan si groupbox ada di dalam Dialog, maka sempurna lah tampilan Dialog dengan tombol.

Urut-urutan di atas harus dipenuhi dalam hard coding di Qt karena jika tidak, maka hanya muncul jendela saja tanpa ada isinya. Atau malah hanya muncul pesan galat. Memang terlihat mbulet tetapi saja janji insya Allah akan mudah dengan penjelasan.

Kode Pembuatan Widget Qt (QPushButton, QHBoxLayout, QGroupBox)

Sebelum saya beberkan kodenya secara keseluruhan, saya tunjukkan dulu bagaimana hard coding untuk tombol, groupbox, dan layout agar Anda merasa juga bahwa sama sekali ini tidak hard.

Kode untuk Membuat Tombol (PushButton)

```
QPushButton *tombol = new QPushButton("PULAU BUTON");
```

Mudah dipahami. Maksudnya adalah membuat objek bernama tombol (dengan sifat pointer karena pakai *), yang diturunkan dari kelas QPushButton, dengan teks PULAU BUTON. Bisa saja tidak memakaikan sifat pointer (tanda *), tetapi nanti objek tombol tidak bisa menunjuk (tanda -> kepada fungsi). Ingat, nama kelas untuk tombol adalah QPushButton bukan QButton. Ingat, C++ adalah bahasa pemrograman yang berbasis objek. Jadi, sedikit-sedikit kita akan bertemu dengan objek. Oh iya, nama objek di sini bebas Anda ubah.

Kode untuk Membuat GroupBox

```
QGroupBox *groupbox = new QGroupBox(tr("Horizontal Layout"));
```

Maksudnya adalah membuat objek bernama groupbox dari kelas QGroupBox, dengan teks judul Horizontal Layout. Buat apa sih, groupbox? Ini berguna untuk mengelompokkan layout agar dalam satu dialog/jendela, bisa ada lebih dari satu layout. Latar belakang saya memakai groupbox ini akan dijelaskan di bawah. Sekali lagi, nama objek boleh Anda tentukan sendiri.

Kode untuk Membuat Layout

```
HBoxLayout *layout = new QHBoxLayout;
```

Maksudnya, membuat objek layout berbasis BoxLayout yang bertipe horizontal, yang bernama layout. Buat apa layout itu? Oke, gunanya besar sekali untuk mengatur secara otomatis item-item GUI yang kita buat baik tombol atau lainnya. Sebagai informasi, layout dalam Qt itu sangatlah banyak jenisnya. Hanya saja, saya di sini pilih salah satunya saja yakni BoxLayout yang tipenya horizontal. Sekali lagi, Anda bebas mengganti nama objek. Cukup jelas, bahwa sampai layout pun dianggap objek dengan C++.

Kode Program

Saatnya membeberkan seluruh kode yang saya miliki. Selayang pandang, ada 5 berkas saja yang digunakan dalam program ini yakni:

main.cpp
mainwindow.cpp
mainwindow.h
mainwindow.ui
DialogPertama.pro

Yang akan Anda utak-atik hanyalah mainwindow.cpp dan sedikit di mainwindow.h. Fokus hanya di dua berkas ini. Ah, jangan bingung dengan nama DialogPertama.pro. Kalau Anda buat proyek Qt baru dengan nama BotolMizone maka namanya jadi BotolMizone.pro. Berkas pro ini adalah konfigurasi proyek Qt kita. Otomatis diciptakan oleh Qt Creator. Kebetulan saja saya menamainya DialogPertama.

main.cpp

```
#include <QtGui/QApplication>
#include "mainwindow.h"

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    Dialog w;
    w.show();

    return a.exec();
}
```

mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QDialog>
#include <QGroupBox>
#include <QPushButton>

class QGroupBox;
class QPushButton;

class Dialog : public QDialog
{
    Q_OBJECT

public:
    Dialog();
};
```

```
private:
    void createHorizontalGroupBox();

    QGroupBox *horizontalGroupBox;
    QPushButton *tombol;
};

#endif // MAINWINDOW_H
```

mainwindow.cpp

```
#include <QtGui>
#include "mainwindow.h"
#include "ui_mainwindow.h"

Dialog::Dialog()
{
    createHorizontalGroupBox();
    QVBoxLayout *mainLayout = new QVBoxLayout;
    mainLayout->addWidget(horizontalGroupBox);
    setLayout(mainLayout);
}

void Dialog::createHorizontalGroupBox()
{
    horizontalGroupBox = new QGroupBox(tr("Ini Horizontal Layout"));
    QHBoxLayout *layout = new QHBoxLayout;
    QPushButton *tombol = new QPushButton("PULAU BUTON");
    QPushButton *timbil = new QPushButton("PULAU SERAM");
    QPushButton *tambal = new QPushButton("PULAU ANGKER");
    layout->addWidget(tombol);
    layout->addWidget(timbil);
    layout->addWidget(tambal);
    horizontalGroupBox->setLayout(layout);
}
```

```
1 #include <QtGui>
2 #include "mainwindow.h"
3 #include "ui_mainwindow.h"
4
5 Dialog::Dialog()
6 {
7     createHorizontalGroupBox();
8     QVBoxLayout *mainLayout = new QVBoxLayout;
9     mainLayout->addWidget(horizontalGroupBox);
10    setLayout(mainLayout);
11 }
12
13
14 void Dialog::createHorizontalGroupBox()
15 {
16     horizontalGroupBox = new QGroupBox(tr("Ini Horizontal Layout"));
17     QHBoxLayout *layout = new QHBoxLayout;
18     QPushButton *tombol = new QPushButton("PULAU BUTON");
19     QPushButton *tambil = new QPushButton("PULAU SERAM");
20     QPushButton *tambal = new QPushButton("PULAU ANGKER");
21     layout->addWidget(tombol);
22     layout->addWidget(tambil);
23     layout->addWidget(tambal);
24     horizontalGroupBox->setLayout(layout);
25 }
```

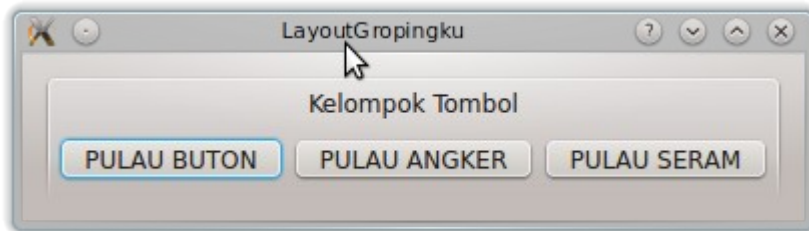
mainwindow.ui

Abaikan isinya karena tidak dipakai.

DialogPertama.pro

Abaikan isinya karena tidak dipakai.

Hasil



Penjelasan Kode

Di dalam main.cpp, kode yang perlu diperhatikan hanya 2 baris yakni

```
Dialog w;  
w.show();
```

Maksudnya adalah membuat objek bernama w dari kelas Dialog. Kelas Dialog ini dijelaskan di dalam mainwindow.h. Inilah pemrograman C++, beberapa berkas pemrograman saling terkait dan saling memanggil. Maka, Anda akan temukan di bagian atas main.cpp ada #include "mainwindow.h".

Di dalam **mainwindow.h**, kode yang perlu diperhatikan adalah

```
class Dialog : public QDialog
```

karena kita memakai Dialog menggantikan MainWindow. Kalau Anda baru membuat proyek baru, pasti kode di dalam mainwindow.h isinya adalah deklarasi kelas MainWindow. Di sini kita menggantinya dengan Dialog.

```
public:  
    Dialog();
```

karena Dialog ini dipakai di main.cpp untuk membuat objek bernama w.

```
private:  
    void createHorizontalGroupBox();  
  
    QGroupBox *horizontalGroupBox;  
    QPushButton *tombol;
```

karena fungsi createHorizontalGroupBox() dipakai di mainwindow.cpp (sedangkan mainwindow.cpp memanggil mainwindow.h ini), horizontalGroupBox adalah objek yang dibuat dari kelas QGroupBox yang objek ini dipakai di mainwindow.cpp, dan objek tombol yang dibuat dari kelas QPushButton yang objek ini juga dipakai di mainwindow.cpp.

```
#include <QDialog>
#include <QGroupBox>
#include <QPushButton>
```

karena di dalam mainwindow.h dipakailah tiga kelas ini yakni QDialog, QGroupBox, dan QPushButton. Enaknya mendeklarasikan ini di atas pada mainwindow.h adalah jika mainwindow.cpp memanggil objek QDialog, maka tidak usah dideklarasikan. Cukup deklarasikan #include "mainwindow.h" dalam mainwindow.cpp karena header inilah yang bertugas memanggil ketiga pustaka tersebut.

Di dalam mainwindow.cpp, kode yang perlu diperhatikan hanya dua blok yakni:

```
//BLOK PERTAMA
Dialog::Dialog()
{
createHorizontalGroupBox();
QVBoxLayout *mainLayout = new QVBoxLayout;
mainLayout->addWidget(horizontalGroupBox);
setLayout(mainLayout);
}
```

karena berkas .cpp yang satu ini adalah inti pemrograman kita. Semua kode sebelumnya hanyalah pelengkap, sebetulnya. Blok pertama ini tujuannya memberikan pada dialog, sebuah layout QVBoxLayout bertipe vertikal. Layout ini, objeknya diberi nama mainLayout. Mengapa? Karena layout kedua nanti akan diletakkan di atas layout pertama ini.

Hal penting lainnya adalah mekanisme pemasukan objek ke dalam layout dan pemasukan layout ke dalam Dialog. Pertama-tama kita harus menciptakan objek layout entah apa tipenya (kalau dalam kode ini QVBoxLayout). Kedua, kita point (->) objek tadi kepada method/fungsi bernama addWidget, dengan memasukkan objek layout kedua ke dalam passing parameter-nya (simpler, passing parameter itu isi dari () setelah kata addWidget). Ingat, method addWidget ini akan sangat sering dipakai dalam hard coding GUI karena memang pembuatan GUI di Qt ini isinya bikin objek lalu tempelkan (addWidget), bikin objek lagi tempelkan (addWidget) lagi, dan begitu terus. Ketiga, kita tempelkan (setLayout) layout yang sudah berisi objek-objek tadi ke dalam parent object (dalam blok ini, Dialog). Jadi, mekanismenya adalah nanti tombol masuk ke groupBox, groupBox masuk ke layout ini, dan layout ini masuk ke Dialog maka muncullah tampilan.

```
//BLOK KEDUA
void Dialog::createHorizontalGroupBox()
{
horizontalGroupBox = new QGroupBox(tr("Ini Horizontal Layout"));
QHBoxLayout *layout = new QHBoxLayout;
QPushButton *tombol = new QPushButton("PULAU BUTON");
QPushButton *timbang = new QPushButton("PULAU SERAM");
QPushButton *tambal = new QPushButton("PULAU ANGKER");
layout->addWidget(tombol);
layout->addWidget(timbang);
layout->addWidget(tambal);
horizontalGroupBox->setLayout(layout);
}
```

```
}
```

lihat, mudah sekali. Kode pada blok kedua inilah yang paling penting dalam pemrograman kali ini. Sebetulnya, andai bisa membuat objek tanpa deklarasi header, maka semua kode yang sebelumnya tidak dibutuhkan. Lihat, di sini terjadi pembuatan objek, lalu penempelan objek ke dalam layout, lalu menempelkan layout ke dalam groupbox. Cuma itu yang dilakukan dengan hard coding GUI di Qt kita kali ini.

Bisa dipahami dari kode ini, pembuatan GUI dengan hard coding ini dilakukan dengan cara membuat objek lalu menempelkan (addWidget) objek tersebut ke dalam layout lalu memasukkan layout tadi ke dalam groupbox (kalau pakai groupbox).

Unduh Source Code Proyek Qt Ini

Ukuran: 4 KB

Tautan: <http://otodidak.freemove.me/tarball/DialogPertama.zip>

Tentang Tulisan Ini

Tulisan ini asalnya adalah dari kode yang diberikan pada laman <http://qt-project.org/doc/qt-4.8/layouts-basiclayouts.html>.

Saya ingin merealisasikan ide dasar saya: aplikasi GUI yang jika satu tombol diklik, tampilannya keseluruhan berubah dengan ganti jumlah tombol dan ganti tampilan gambar. Mirip Ubiquity, tetapi dilengkapi kemampuan pindah tampilan ke penataan yang lain.

Bukti sulitnya realisasi adalah pertanyaan saya yang belum sukses:

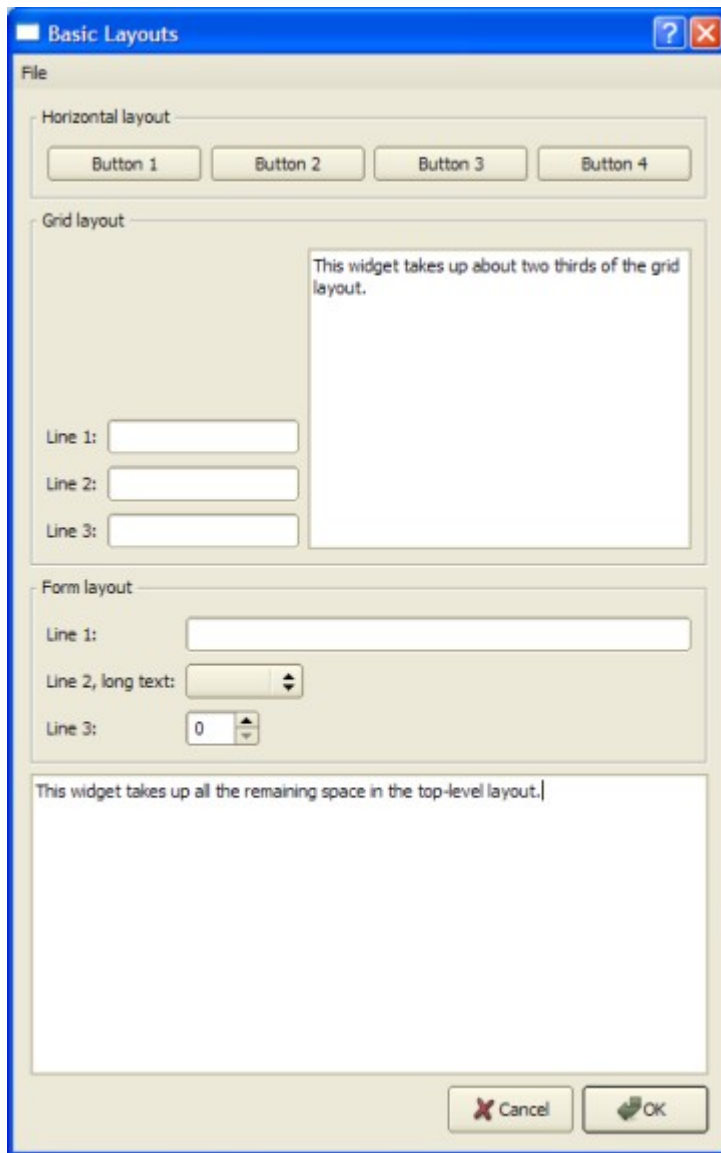
<http://qt-project.org/forums/viewthread/28330/> (Image Slider)

<http://qt-project.org/forums/viewthread/28582/> (GUI Layering and Contents Changed by Button)

Realisasi terdekat yang saya kira adalah dengan

1. membuat sendiri tombol-tombol di dalam satu jendela
2. mengisi setiap tombol dengan kode yang bisa membuat objek-objek GUI
3. jika satu tombol diklik, maka tampilan berubah sesuai kode dan tampilan itu bisa dipanggil lagi kalau si tombol diklik.

itu semua dilakukan dengan membuat sendiri semuanya dan memakai layout. Dan akhirnya saya temukan laman tersebut dan menemukan gambar demo layout ini.



Dan tulisan ini adalah ikatan agar saya tidak lupa cara memrogramnya karena saya pelupa.