

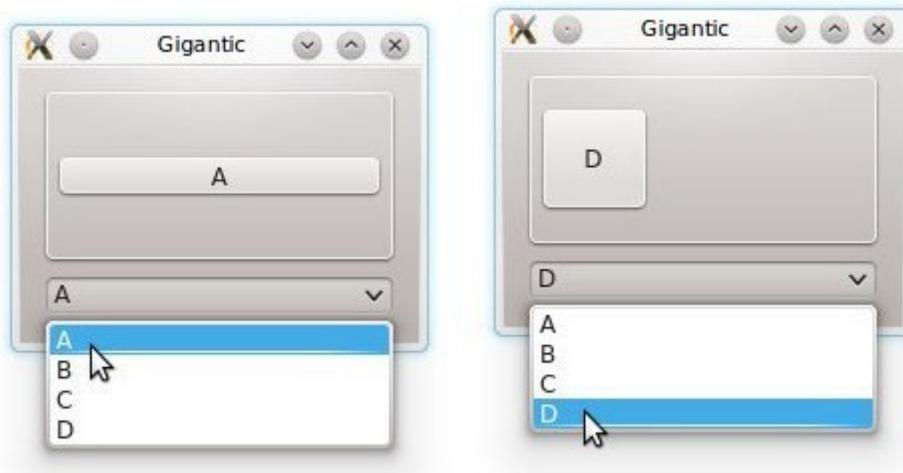
Pemrograman Qt 4 – QStackedLayout dan QComboBox untuk Window Layering

Bismillahirrahmanirrahim.

Tulisan ini adalah bentuk PDF dari <http://malsasa.wordpress.com/2013/08/03/pemrograman-qt-4-qstackedlayout-dan-qcombobox-untuk-window-layering/>.

Saya ingin membuat program yang bisa layering. Saya tidak tahu istilah resmi untuk ini tetapi bayangkanlah aplikasi GUI yang jika satu tombol diklik, maka sebagian atau seluruh isi jendela berubah jadi susunan lain. Anda sering melihatnya di aplikasi yang tidak seperti Word, tidak seperti Excel, tidak seperti Notepad, tidak seperti Firefox. Mereka statis. Anda sering melihatnya ketika Anda menginstal aplikasi di Windows. Jika Anda klik Next/Back, sebagian isi jendela berubah. Yang seperti itu yang saya inginkan. Kali ini saya berhasil membuatnya dengan memanfaatkan QStackedLayout dan QComboBox. Sayang, saya hanya bisa memakai QComboBox bukan QPushButton, padahal yang saya mau adalah QPushButton. Tetapi tidak apa. Ini insya Allah menambah pengetahuan kita dalam membuat aplikasi Qt.

Wujud Program yang Diinginkan



Pokoknya, tampilan dari satu jendela berubah total jika saya memilih salah satu opsi tampilan dari 4 yang ada. Sederhana saja. Saya hanya memasukkan tombol dalam setiap QFrame yang masing-masing QFrame itu bisa dipanggil ke atas (makanya saya kasih nama layering) dengan combo box.

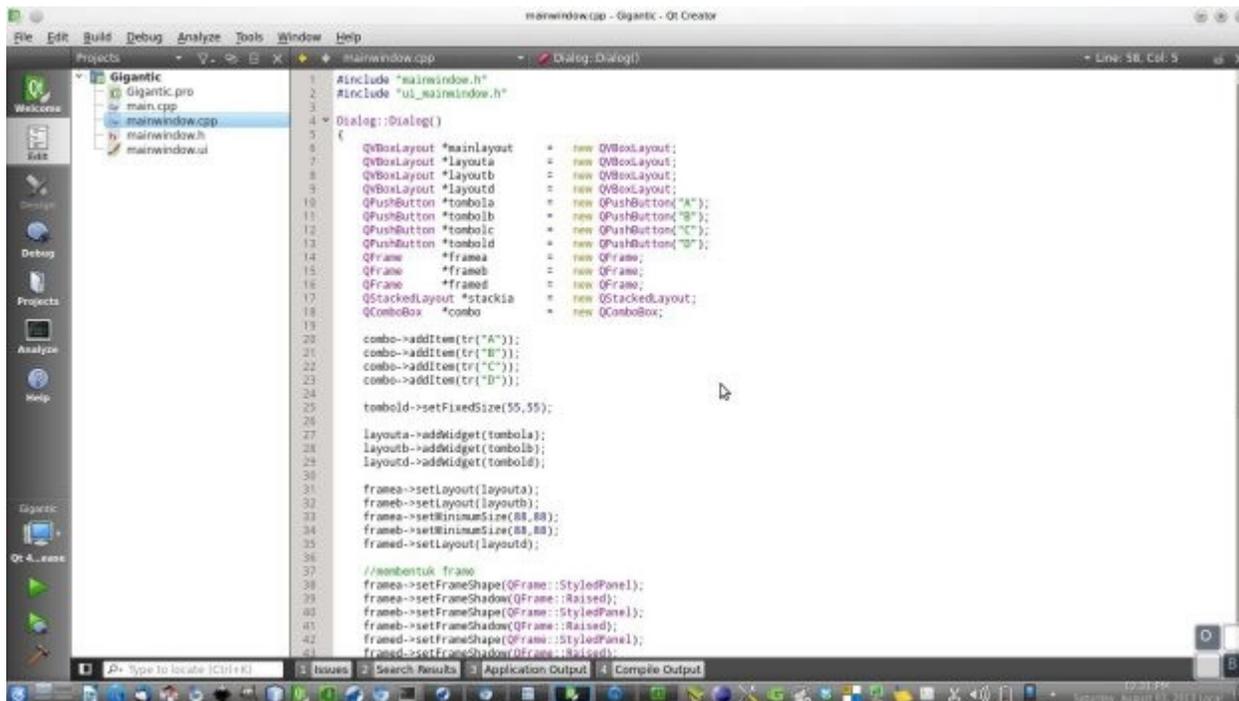
Daftar Kelas Qt yang Dipakai

1. QDialog = dipakai untuk membuat jendela tertinggi
2. QVBoxLayout = dipakai untuk membuat kontainer yang menampung setiap blok/grup dari tombol
3. QPushButton = dipakai untuk membuat tombol
4. QFrame = dipakai untuk menggantikan QGroupBox dalam hal menampung tombol-tombol jadi satu blok/grup.
5. QStackedLayout = dipakai untuk membentuk layer-layer tampilan yang setiap layer-nya bisa berisi widget apa saja.
6. QComboBox = dipakai untuk mewakili masing-masing layer dan untuk switching antarlayer.

Daftar Method Qt yang Dipakai

1. addWidget -> dipakai oleh QVBoxLayout, QStackedLayout
2. setMinimumSize -> dipakai oleh QFrame
3. setLayout -> dipakai oleh QFrame dan QDialog
4. setMaximumWidth -> dipakai oleh QPushButton
5. setFrameShadow -> dipakai oleh QFrame untuk membentuk bayangan; opsi-opsinya adalah QFrame::Raised, QFrame::Sunken, dan QFrame::Plain
6. addItem -> dipakai oleh QStackedLayout untuk membuat opsi-opsi yang bisa diklik. Persis saat Anda memilih font di Word.
7. setFrameShape -> dipakai oleh QFrame untuk membentuk wujud panel apakah itu timbul atau tenggelam; opsi-opsinya adalah QFrame::StyledPanel [dipakai oleh default-nya Qt Creator GUI Builder], QFrame::Box
8. connect -> dipakai di antara QComboBox dengan QStackedLayout.

Qt Creator dan Kode



Kode Program

Sama saja dengan ngite kemarin, hanya ada mainwindow.cpp yang paling penting. Jika bingung dengan kode Qt, silakan merujuk ke tulisan Qt sebelum ini.

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

Dialog::Dialog()
{
    QVBoxLayout *mainlayout = new QVBoxLayout;
    QVBoxLayout *layouta = new QVBoxLayout;
    QVBoxLayout *layoutb = new QVBoxLayout;
    QVBoxLayout *layoutd = new QVBoxLayout;
    QPushButton *tombola = new QPushButton("A");
    QPushButton *tombolb = new QPushButton("B");
    QPushButton *tombolc = new QPushButton("C");
    QPushButton *tombold = new QPushButton("D");
    QFrame *framea = new QFrame;
    QFrame *frameb = new QFrame;
    QFrame *framed = new QFrame;
    QStackedLayout *stackia = new QStackedLayout;
    QComboBox *combo = new QComboBox;
```

```
combo->addItem(tr("A"));
combo->addItem(tr("B"));
combo->addItem(tr("C"));
combo->addItem(tr("D"));

tombold->setFixedSize(55,55);

layouta->addWidget(tombola);
layoutb->addWidget(tombolb);
layoutd->addWidget(tombold);

framea->setLayout(layouta);
frameb->setLayout(layoutb);
framea->setMinimumSize(88,88);
frameb->setMinimumSize(88,88);
framed->setLayout(layoutd);

//membentuk frame
framea->setFrameShape(QFrame::StyledPanel);
framea->setFrameShadow(QFrame::Raised);
frameb->setFrameShape(QFrame::StyledPanel);
frameb->setFrameShadow(QFrame::Raised);
framed->setFrameShape(QFrame::StyledPanel);
framed->setFrameShadow(QFrame::Raised);

//mengecilkan ukuran tombol c
tombolc->setMaximumWidth(33);

stackia->addWidget(framea);
stackia->addWidget(frameb);
stackia->addWidget(tombolc);
stackia->addWidget(framed);

mainlayout->addLayout(stackia);

mainlayout->addWidget(combo);
setLayout(mainlayout);

connect(combo, SIGNAL(activated(int)), stackia,
SLOT(setCurrentIndex(int)));
}
```

Analisis Kode Program

Sedikit saja sebenarnya yang istimewa dari program ke-4 ini. Cukup tahu hal ini maka Anda bisa memodifikasi program menjadi lebih istimewa.

SATU

```
combo->addItem(tr("A"));
combo->addItem(tr("B"));
combo->addItem(tr("C"));
combo->addItem(tr("D"));
Gunanya untuk menambahkan opsi-opsi pada combo box.
```

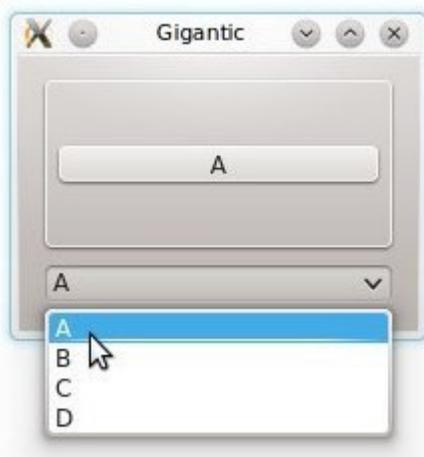
DUA

```
connect(combo, SIGNAL(activated(int)), stackia,
        SLOT(setCurrentIndex(int)));
```

Ini sangat istimewa. Inilah inti dari pemrograman Qt yakni SIGNAL dan SLOT. Method `connect()` ini adalah suatu penghubung antara objek yang mengeluarkan SIGNAL dan objek yang menerimanya yaitu memiliki SLOT. Ini maksudnya jika suatu objek mengeluarkan SIGNAL tertentu maka objek satunya melakukan fungsi tertentu. Itu saja. Dan inilah yang saya pakai untuk mengendalikan layer di dalam `QStackedLayout` dengan menggunakan `QComboBox`. Kita pun pasti akan menggunakannya dalam mengendalikan objek lainnya dengan tombol atau dengan objek lainnya pula. Karena itu, inilah inti dari pemrograman Qt.

```
combo -> objek dari QComboBox
stackia -> objek dari QStackedLayout
```

Hasil Akhir



Rangkuman

1. Layering dilakukan dengan QStackedLayout karena ia mengakomodasi frame-frame sebagai layer dan mampu mengatur susunannya yang kita lihat dengan SLOT `setCurrentIndex()`.
2. Layer-layer di dalam QStackedLayout bisa diatur mana yang terlihat dengan mengaturnya dari QComboBox dengan menghubungkan mereka berdua memakai `connect()`.
3. Sementara yang saya tahu, hanya QComboBox yang mampu mengatur layer dalam QStackedLayout dan QPushButton belum berhasil mengaturnya. Anda bisa melihat buktinya di sini dan di sini. Tampaknya, QStackedLayout ingin selalu diberi masukan berupa integer sementara QPushButton tidaklah seperti QComboBox yang mengeluarkan nilai integer untuk setiap opsi yang kita tentukan. Misalnya yaa, A itu 1, B itu 2, C itu 3, dan seterusnya seperti kode di atas.