

Pemrograman Qt 5 – QStackedLayout dan QPushButton untuk Layering (setCurrentIndex)

Bismillahirrahmanirrahim.

Tulisan ini adalah bentuk PDF dari <http://malsasa.wordpress.com/2013/08/20/pemrograman-qt-5-qstackedlayout-dan-qpushbutton-untuk-layering-setcurrentindex/>.

Satu bulan saya mencari cara mengganti layer (bahasa teknisnya mengganti indeks dari QStackedLayout) dengan klik pada QPushButton. Kemarin maksimal baru bisa berganti dengan QComboBox. Alhamdulillah, penantian itu usai dan saya senang sekali. Mengapa? Karena jika ini sudah berhasil dilakukan, maka semua proses pengerjaan program yang saya inginkan akan lancar insya Allah. Bagaimana lagi? Program saya sebetulnya hanya berisi hal ini nantinya. Sekarang saatnya saya beberkan programnya.

Wujud Program yang Diinginkan



Penjelasan: pokoknya, tampilan program harus berubah menjadi tampilan lain jika diklik tombol. Di sini, ada frame-frame yang ditumpuk jadi layer. Kalau tombol 1 diklik, maka frame 1 tampil di atas sedangkan frame 2 di bawah. Sebaliknya pula demikian. Ini yang saya sebut layering.

Daftar Kelas Qt yang Dipakai

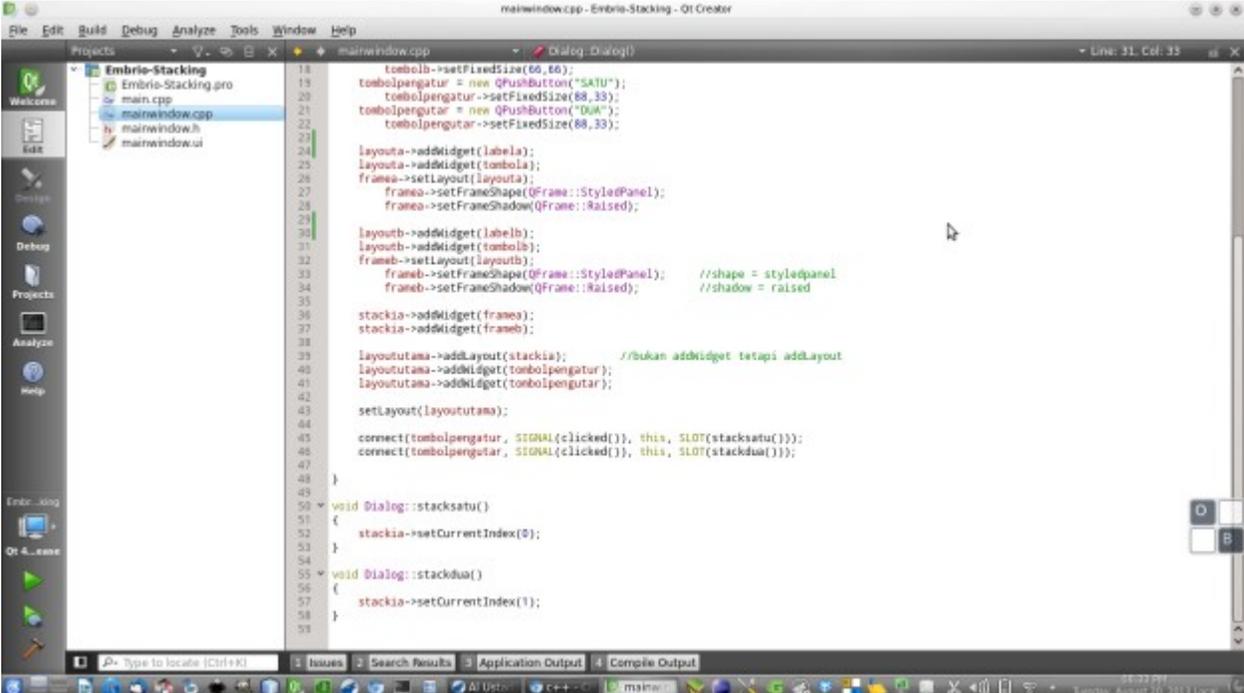
1. QVBoxLayout
2. QFrame
3. QPushButton
4. QStackedLayout (primadona tulisan ini)
5. QLabel

Daftar Method Qt yang Dipakai

1. addWidget
2. setLayout
3. addLayout
4. setCurrentIndex (primadonanya primadona ini)
5. stacksatu
6. stackdua
7. setFrameShape
8. setFrameShadow
9. setFixedSize

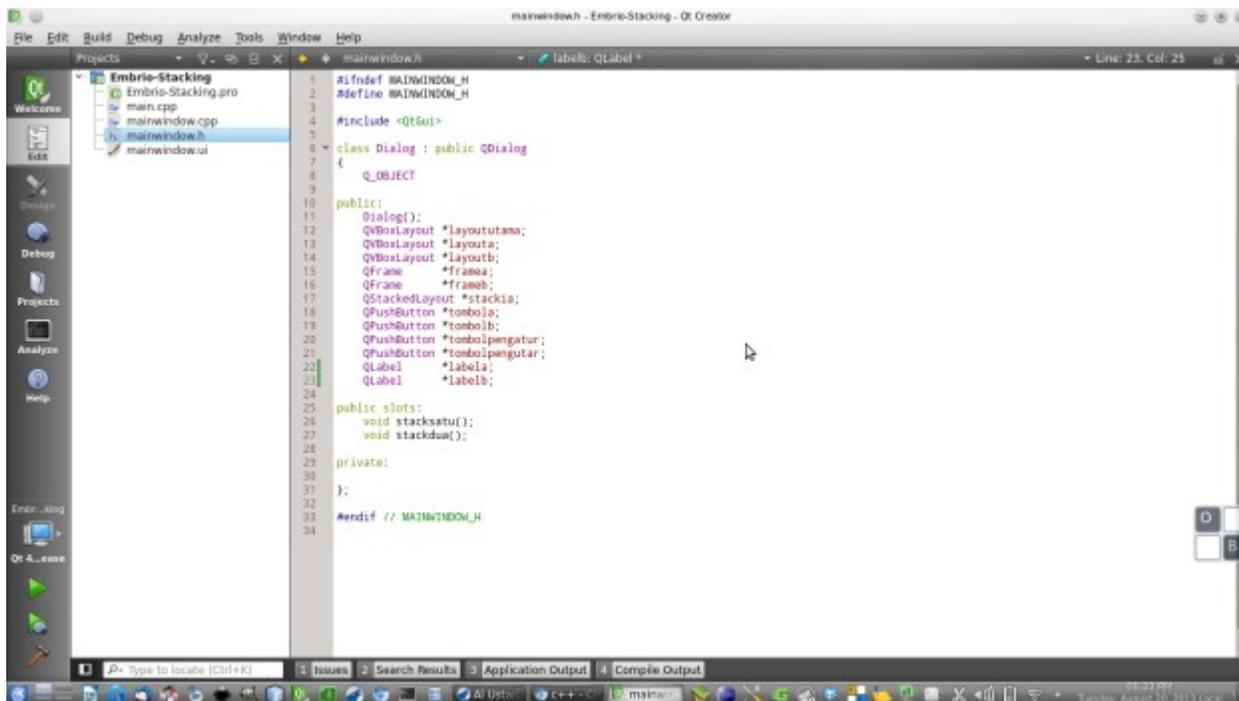
Qt Creator dalam Kode

mainwindow.cpp



```
mainwindow.cpp - Embrio-Stacking - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects Embrio-Stacking mainwindow.cpp Dialog - Dialog()
Embrio-Stacking pro
  main.cpp
  mainwindow.cpp
  mainwindow.h
  mainwindow.ui
18 tombolb->setFixedSize(66, 66);
19 tombolpengatur = new QPushButton("SATU");
20 tombolpengatur->setFixedSize(88, 33);
21 tombolpengutar = new QPushButton("DUA");
22 tombolpengutar->setFixedSize(88, 33);
23
24 layout->addWidget(labela);
25 layout->addWidget(tombola);
26 framea->setLayout(layouta);
27 framea->setFrameShape(QFrame::StyledPanel);
28 framea->setFrameShadow(QFrame::Raised);
29
30 layoutb->addWidget(labelb);
31 layoutb->addWidget(tombolb);
32 frameb->setLayout(layoutb);
33 frameb->setFrameShape(QFrame::StyledPanel); //shape = styledpanel
34 frameb->setFrameShadow(QFrame::Raised); //shadow = raised
35
36 stackia->addWidget(framea);
37 stackia->addWidget(frameb);
38
39 layoututama->addLayout(stackia); //bukan addWidget tetapi addLayout
40 layoututama->addWidget(tombolpengatur);
41 layoututama->addWidget(tombolpengutar);
42
43 setLayout(layoututama);
44
45 connect(tombolpengatur, SIGNAL(clicked()), this, SLOT(stacksaatu()));
46 connect(tombolpengutar, SIGNAL(clicked()), this, SLOT(stackdua()));
47
48 }
49
50 void Dialog::stacksaatu()
51 {
52     stackia->setCurrentIndex(0);
53 }
54
55 void Dialog::stackdua()
56 {
57     stackia->setCurrentIndex(1);
58 }
59 }
```

mainwindow.h



Kode Program

mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QtGui>

class Dialog : public QDialog
{
    Q_OBJECT

public:
    Dialog();
    QVBoxLayout *layoututama;
    QVBoxLayout *layouta;
    QVBoxLayout *layoutb;
    QFrame *framea;
    QFrame *frameb;
    QStackedLayout *stackia;
    QPushButton *tombola;
    QPushButton *tombolb;
    QPushButton *tombolpengatur;
    QPushButton *tombolpengutar;
    QLabel *labela;
    QLabel *labelb;
};

#endif
```

```
        QLabel      *labelb;  
  
public slots:  
    void stacksatu();  
    void stackdua();  
  
private:  
  
};  
  
#endif // MAINWINDOW_H
```

mainwindow.cpp

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

Dialog::Dialog()
{
    layoututama = new QVBoxLayout;
    layouta     = new QVBoxLayout;
    layoutb     = new QVBoxLayout;
    framea      = new QFrame;
    frameb      = new QFrame;
    stackia     = new QStackedLayout;
    labela      = new QLabel("FRAME 1");
    labelb      = new QLabel("FRAME 2");

    tombolbola  = new QPushButton("1");
        tombolbola->setFixedSize(88,88);
    tombolbolb  = new QPushButton("2");
        tombolbolb->setFixedSize(66,66);
    tombolpengatur = new QPushButton("SATU");
        tombolpengatur->setFixedSize(88,33);
    tombolpengutar = new QPushButton("DUA");
        tombolpengutar->setFixedSize(88,33);

    layouta->addWidget(labela);
    layouta->addWidget(tombolbola);
    framea->setLayout(layouta);
        framea->setFrameShape(QFrame::StyledPanel);
        framea->setFrameShadow(QFrame::Raised);

    layoutb->addWidget(labelb);
    layoutb->addWidget(tombolbolb);
    frameb->setLayout(layoutb);
        frameb->setFrameShape(QFrame::StyledPanel);           //shape =
styledpanel
        frameb->setFrameShadow(QFrame::Raised);           //shadow =
raised

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

    layoututama->addLayout(stackia);           //bukan addWidget tetapi
addLayout
    layoututama->addWidget(tombolpengatur);
    layoututama->addWidget(tombolpengutar);

    setLayout(layoututama);

    connect(tombolpengatur, SIGNAL(clicked()), this,
SLOT(stacksatu()));
```

```

        connect (tombolpengutar, SIGNAL(clicked()), this,
SLOT(stackdua()));

}

void Dialog::stacksatu()
{
    stackia->setCurrentIndex(0);
}

void Dialog::stackdua()
{
    stackia->setCurrentIndex(1);
}

```

Analisis Kode Program

HEADER

Kali ini saya tidak bisa mengabaikan HEADER begitu saja. Berkas mainwindow.h menjadi penting pada program ini. Mengapa? Ini berkaitan dengan slot dan fungsi yang dipakai. Pada berkas ini, yang paling penting yang harus diperhatikan adalah deklarasi. Lihat, deklarasi objek-objek ini sangat penting dan harus ada:

```

QVBoxLayout      *layoututama;
QVBoxLayout      *layouta;
QVBoxLayout      *layoutb;
QFrame           *framea;
QFrame           *frameb;
QStackedLayout   *stackia;
QPushButton      *tombola;
QPushButton      *tombolb;
QPushButton      *tombolpengatur;
QPushButton      *tombolpengutar;
QLabel           *labela;
QLabel           *labelb;

```

Ingat, mereka harus dideklarasikan dalam mainwindow.h. Mereka harus ada di bawah lingkupan public:. Jangan dipikirkan antara nama tombol pengatur dengan pengutar. Sama saja. Yang penting lagi, bahkan paling pentingnya adalah ini:

```

public slots:
    void stacksatu();
    void stackdua();

```

Mengapa penting? Karena dua buah slot ini akan dipakai untuk menggerakkan QStackedLayout supaya mengubah indeksnya. Alias, ganti tampilan. Dan slot ini harus dideklarasikan di lingkup public slots:, tidak bisa ditaruh pada private karena slot ini berjalan antarfungsi. Kalau dibuat private, maka ia hanya

berjalan di dalam satu fungsi yang sama. Yang dimaksud antarfungsi adalah fungsi besar Dialog(), stacksatu/stackdua itu sendiri, dan fungsi khusus setCurrentIndex() milik objek QStackedLayout yang semuanya ada di dalam mainwindow.cpp. Intinya, yang diinginkan dengan ini adalah, jika ada tombol diklik maka fungsi stacksatu() dipanggil, dan di dalam fungsi stacksatu() tersebut dipanggillah fungsi setCurrentIndex(). Perhatikan, objek QStackedLayout berada di dalam fungsi Dialog(), tidak di dalam stacksatu(). Ini hanya bisa dilakukan jika stacksatu bertipe public. Demikian juga dengan stackdua().

Oh iya, pada program kali ini, area private: dibiarkan kosong. Sudah, jangan dipikirkan.

CPP

Khusus untuk mainwindow.cpp, ada sesuatu yang sangat berbeda dengan 4 tulisan Qt sebelumnya. Apa yang beda? Deklarasi objek-objek yang dipangkas. Lihat:

```
layoututama = new QVBoxLayout;
layouta      = new QVBoxLayout;
layoutb      = new QVBoxLayout;
framea       = new QFrame;
frameb       = new QFrame;
stackia      = new QStackedLayout;
labela       = new QLabel("FRAME 1");
labelb       = new QLabel("FRAME 2");

tombola      = new QPushButton("1");
tombola->setFixedSize(88,88);
tombolb      = new QPushButton("2");
tombolb->setFixedSize(66,66);
tombolpengatur = new QPushButton("SATU");
tombolpengatur->setFixedSize(88,33);
tombolpengutar = new QPushButton("DUA");
tombolpengutar->setFixedSize(88,33);
```

Yang sangat perlu diperhatikan adalah pergantian pola dari:

```
QNamaKelas *namaObjek = new QNamaKelas;
```

menjadi bentuk deklaras objek yang terpangkas demikian:

```
namaObjek = new QnamaKelas;
```

Perhatikan. Deklarasi awal QNamaKelas tidak dipakai lagi. Juga tidak dipakai lagi tanda pointer (*). Langsung ditulis nama objek yang diinginkan lalu tetap ditulis nama kelas di akhir deklaras (sebelah kanan tanda '='). Mengapa dipakai deklaras model begini? Karena kita sudah mendeklarasikan secara umum pada header. Dan karena saya kemarin satu bulan memakai pola yang pertama untuk deklaras dalam mainwindow.cpp ini, satu bulan pula saya gagal mengerjakan kode ini. Setelah diganti dengan deklaras baru ini, dan ditambahi sedikit kode, alhamdulillah berhasil.

Yang paling penting dari berkas CPP kali ini adalah kode connect ini:

```
connect(tombolpengatur, SIGNAL(clicked()), this, SLOT(stacksatu()));  
connect(tombolpengutar, SIGNAL(clicked()), this, SLOT(stackdua()));
```

Inilah kunci dari berkas CPP ini, yang satu bulan saya cari kode yang benarnya bagaimana. Ternyata beginilah menulis kodenya. Inti kode di atas adalah

```
objek, SIGNAL(fungsi()), objek, SLOT(fungsi())
```

yang diterjemahkan dalam konteks program 5 ini menjadi

```
tombol, SIGNAL(clicked()), this, SLOT(stacksatu())
```

Maksud this adalah kelas yang tulisan this ini ada di dalamnya. Maka this merujuk pada Dialog. Makna kode connect ini adalah jika ada tombol yang terkena perlakuan clicked() (maksudnya, diklik), maka kelas Dialog memanggil fungsi stacksatu(). Itulah maksudnya. Dan ini sangat krusial, tidak boleh diganti kata this menjadi Dialog atau kata Dialog(), atau tidak juga boleh diganti menjadi stackia walaupun aslinya yang ingin dilakukan dengan kode connect() ini hanyalah mengganti indeks milik stackia saja. Nyatanya, harus demikianlah caranya.

Kemudian, kode yang harus ada meskipun kecil sekali adalah ini:

```
void Dialog::stacksatu()  
{  
    stackia->setCurrentIndex(0);  
}  
  
void Dialog::stackdua()  
{  
    stackia->setCurrentIndex(1);  
}
```

Maksud semua potongan kode ini hanyalah bagaimana caranya dari tombol dikirim perintah ke QStackedLayout untuk mengubah indeksinya. Sudah, begitu saja. Hanya saja untuk itu, harus ada fungsi khusus sebagai SLOT yang dipanggil di dalam connect, yang di dalam fungsi khusus itu akan dipanggil fungsi asli milik QStackedLayout sehingga sanggup mengubah indeksinya. Jadi ada fungsi di dalam fungsi. Itu terjadi dalam kode fungsi stacksatu/stackdua di atas. Perhatikan, stackia->setCurrentIndex(1); itulah kodenya. Untuk bisa mengganti setCurrentIndex ini harus kita lakukan demikian panjang pengodean. Namun jika dilihat ulang secara keseluruhan, ealah, ternyata kode kita ini sangatlah pendek. Mudah untuk dilakukan.

Hasil Akhir



Alhamdulillah, berhasil. Kalau sudah bisa begini, semua sisa pengerjaan akan menjadi sangat-sangat ringan.

Rangkuman

1. Fungsi connect untuk menggerakkan QStackedLayout dengan QPushButton harus disandarkan receiver SLOT-nya pada kelas tertinggi yakni Dialog, dengan kata ganti this. Khas OOP.
2. Mengubah nilai setCurrentIndex miliknya QStackedLayout itu dilakukan dengan membuat satu fungsi baru yang di dalam fungsi itu baru dipanggilah setCurrentIndex. Tidak bisa secara langsung setCurrentIndex dipanggil di dalam connect.
3. Deklarasi objek di dalam CPP jika objek itu sudah dideklarasikan di H, maka tidak diulang nama kelasnya di kiri tanda '='.

Catatan Penting

Saya menyebutkan satu bulan rentang waktu pencarian ini adalah dari sini:

1. <http://qt-project.org/forums/viewthread/30461/>
2. <http://stackoverflow.com/questions/17916905/change-qstackedlayout-index-with-qpushbutton-not-qcombobox>