Pemrograman Qt 3 – Mendayagunakan QFrame dan QVBoxLayout untuk Membentuk Kolom Lebih Banyak dalam QDialog

Bismillahirrahmanirrahim.

Tulisan ini adalah bentuk PDF dari http://malsasa.wordpress.com/2013/07/07/pemrograman-qt-3-mendayagunakan-qframe-dan-qvboxlayout-untuk-membentuk-kolom-lebih-banyak-dalam-qdialog/.

Kalau kemarin kita *ngite* dengan QGroupBox dan QHBoxLayout, sekarang kita akan mencoba QFrame sebagai ganti QGroupBox baik sebagai kontainer maupun pembuat spasi. Ide dasar program ketiga ini adalah spasi kosong untuk setiap grup widget yang kita buat di dalam QDialog. Oh iya, sampai hari ini kita hanya akan berkutat dengan QDialog tanpa menyentuh QMainWindow. Tentulah, semua hard coding ini tujuannya membuat aplikasi GUI di Linux. Semoga bermanfaat untuk kaum muslimin semuanya.

Wujud Program yang Diinginkan

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. QGridLayout = dipakai untuk membuat kontainer paling besar yang menampung semua objek QVBoxLayout nantinya
4. QPushButton = dipakai untuk membuat tombol
5. QIcon = dipakai untuk memasukkan gambar sebagi ikon tombol, QIcon dipakai sebagai passing parameter
6. QSize = dipakai untuk membuat objek yang menyimpan ukuran, yang di program ini objek berisi ukuran tersebut diterapkan pada logo dalam tombol
7. QFrame = dipakai untuk menggantikan QGroupBox dalam hal menampung tombol-tombol jadi satu blok/grup.

Daftar Method Qt yang Dipakai

1. addWidget -> dipakai oleh QVBoxLayout, QGridLayout
2. setMinimumSize -> dipakai oleh QPushButton
3. setIconSize -> dipakai oleh QPushButton
4. setIcon -> dipakai oleh QPushButton
5. setLayout -> dipakai oleh QFrame dan QDialog
6. setFixedWidth -> dipakai oleh QFrame
7. setFrameShadow -> dipakai oleh QFrame untuk membentuk bayangan; opsi-opsinya adalah QFrame::Raised, QFrame::Sunken, dan QFrame::Plain
8. 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
9. setColumnMinimumWidth -> dipakai oleh QGridLayout untuk menentukan ukuran setiap blok/grup dan menentukan jumlah blok/grup yang bisa dipakai
10. setRowMinimumHeight -> dipakai oleh QGridLayout, idem
11. setLayout -> dipakai oleh QDialog selaku kelas tertinggi untuk memasang QGridLayout ke dalam dirinya.

Peta GUI

Saya jelaskan dulu peta QFrame dalam program ini yang saya susun.

A = QFrame (frameKiri) dengan QVBoxLayout (layoutKontainerKiri)
B = QFrame (frameTengah)
C = QFrame (frameKanan)
D = QFrame (frameSangga)
E = QFrame (frameBawah)
F = QFrame (framePojok)

Kode Program

Sama dengan kegiatan ngite kita kemarin, hanya berkas mainwindow.cpp yang penting.

mainwindow.cpp

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

Dialog::Dialog()
{
    QVBoxLayout *layoutKontainerKanan  = new QVBoxLayout;
    QVBoxLayout *layoutKontainerKiri   = new QVBoxLayout;
    QVBoxLayout *layoutKontainerTengah = new QVBoxLayout;
    QVBoxLayout *layoutKontainerBawah  = new QVBoxLayout;
    QVBoxLayout *layoutKontainerPojok  = new QVBoxLayout;
    QGridLayout *layoutUtama           = new QGridLayout;

    QPushButton *tombolSatu          = new QPushButton("SATU");
    QPushButton *tombolDua           = new QPushButton("DUA");
    QPushButton *tombolTiga          = new QPushButton("TIGA");
    QPushButton *tombolEmpat         = new QPushButton("");

        layoutKontainerKanan->addWidget(tombolSatu);
        layoutKontainerKanan->addWidget(tombolDua);
        layoutKontainerKanan->addWidget(tombolTiga);
        layoutKontainerKanan->addWidget(tombolEmpat);

        QSize size(88,88);
 //menentukan ukuran, bisa dipakai di mana-mana nanti

        tombolEmpat->setMinimumSize(100,100);              //objek
size lalu dimasukkan ke dalam method setIconSize dalam tombolEmpat
        tombolEmpat->setIconSize(size);
 //nggak sangka, dasar OOP; sampai ukuran ikon pun disimpan sebagai
objek
        tombolEmpat->setIcon(QIcon(":/gambar/ubuntu.png")); //logo
ubuntu.png ini jadi seukuran 88x88 betulan dalam tombol [sebelumnya
gagal]

    QPushButton *tombolLima          = new QPushButton("LIMA");
    QPushButton *tombolEnam          = new QPushButton("ENAM");
    QPushButton *tombolTujuh         = new QPushButton("TUJUH");
    QPushButton *tombolDelapan       = new QPushButton("DELAPAN");

        layoutKontainerKiri->addWidget(tombolLima);
        layoutKontainerKiri->addWidget(tombolEnam);
        layoutKontainerKiri->addWidget(tombolTujuh);
```

```
        layoutKontainerKiri->addWidget(tombolDelapan);

    //TOMBOL-TOMBOL DI BARISAN BAWAH
    QPushButton *tombolSembilan      = new QPushButton("SEMBILAN");

        layoutKontainerBawah->addWidget(tombolSembilan);

    QPushButton *tombolSepuluh       = new QPushButton("SEPULUH");

        layoutKontainerPojok->addWidget(tombolSepuluh);

    //FRAME-FRAME PEMUAT TOMBOL
    QFrame *frameKanan = new QFrame;
    frameKanan->setLayout(layoutKontainerKanan);

    QFrame *frameKiri  = new QFrame;
    frameKiri->setLayout(layoutKontainerKiri);

    QFrame *frameTengah = new QFrame;
    frameTengah->setFixedWidth(66);                     //berhasil
mengatur ukuran lebar frame secara galak
//

frameTengah->setFrameShadow(QFrame::Raised);
 //setFrameShadow dan setFrameStyle
//
 frameTengah->setFrameShape(QFrame::StyledPanel);  //kalau
dimatikan, maka semua shadow hilang dan hanya tampak spasi kosong
    frameTengah->setLayout(layoutKontainerTengah);      //jangan lupa
masukkan layout ke dalam frame

    QFrame *frameBawah = new QFrame;
    frameBawah->setLayout(layoutKontainerBawah);

    QFrame *frameSangga = new QFrame;
    frameSangga->setFixedHeight(88);

    QFrame *framePojok  = new QFrame;
    framePojok->setFixedHeight(50);
    framePojok->setLayout(layoutKontainerPojok);

    layoutUtama->addWidget(frameKanan, 1, 3);
    layoutUtama->addWidget(frameTengah, 1, 2);
    layoutUtama->addWidget(frameKiri, 1, 1);
    layoutUtama->addWidget(frameBawah, 3, 1);
    layoutUtama->addWidget(frameSangga, 2, 1);
    layoutUtama->addWidget(framePojok, 3, 3);
    layoutUtama->setColumnMinimumWidth(3, 100);     //menentukan
jumlah kolom dan lebar masing-masingnya
    layoutUtama->setRowMinimumHeight(3, 10);       //menentukan
jumlah baris dan tinggi masing-masingnya
```

```
    setLayout(layoutUtama);
}
```

Analisis Kode Program

Sebenarnya sama saja dengan program sebelumnya. Maka akan saya jelaskan beberapa saja yang terpenting.

SATU

```
QFrame *frameKanan = new QFrame;
 frameKanan->setLayout(layoutKontainerKanan);
```

Masih seperti kemarin, itulah bentuk umum pembuatan objek dari kelas. Di sini yang disebut kelas ialah QFrame sedangkan objeknya frameKanan. Sesudah dibuat, objek dipasangi layout dengan memakai method setLayout.

DUA

```
frameTengah->setFixedWidth(66);
```

Inilah ide dasar program ketiga ini. Method setFixedWidth, sesuai namanya, berguna untuk menentukan ukuran lebar dari frameTengah. Frame yang ini posisinya ada pada baris pertama kolom kedua. Kalau bingung, lihat lagi Peta GUI di atas.

TIGA

```
layoutUtama->addWidget(frameKanan, 1, 3);
layoutUtama->addWidget(frameTengah, 1, 2);
layoutUtama->addWidget(frameKiri, 1, 1);
layoutUtama->addWidget(frameBawah, 3, 1);
layoutUtama->addWidget(frameSangga, 2, 1);
layoutUtama->addWidget(framePojok, 3, 3);
```

Perhatikan, layoutUtama sudah dibentuk dari QGridLayout, bukan layout model lainnya. Kalau dibentuk dari kelas ini, maka dia memiliki bentuk umum passing parameter ala QGridLayout juga. Dan ini spesial karena Grid Layout milik Qt itu fleksibel bisa ke samping dan ke bawah sekaligus. Misalnya saja, kode layoutUtama->addWidget(frameKanan, 1, 3); berarti maksudnya masukkan ke dalam Grid Layout bernama layoutUtama sebuah objek bernama frameKanan, pada baris 1 kolom 3. Mudah sekali, bukan? Begini bentuk umumnya:

```
void QGridLayout::addWidget ( QWidget * widget, int fromRow, int
fromColumn, int rowSpan, int columnSpan, Qt::Alignment alignment =
0 )
```

Bentuk umum ini saya dapat dari dokumentasi internal Qt Creator (tekan F1). Bentuk umum yang njelimet ini, ternyata bisa diterjemahkan jadi kode saya tadi. Bagaimana

memahaminya? Begini caranya:

1. void ini berarti bentuk kode QGridLayout::addWidget dst. adalah fungsi, bukan kelas bukan yang lain. Jadi, tidak perlu dilihat nama QGridLayout-nya kalau mau menggunakan ini. Cukup lihat addWidget dst. saja. Ini artinya fungsi addWidget() di dalam kelas QGridLayout itu cara pakainya demikian. Dan memang berbeda addWidget untuk QGridLayout dengan yang untuk layout lainnya.
2. QWidget * widget diganti dengan frameKanan
3. int fromRow diganti dengan 1
4. int fromColumn diganti dengan 3
5. sisanya tidak diisi tidak apa-apa.

EMPAT

```
layoutUtama->setColumnMinimumWidth(3, 100);
layoutUtama->setRowMinimumHeight(3, 10);
```

Penjelasannya persis seperti nama method-nya. Bentuk umumnya adalah

```
void QGridLayout::setColumnMinimumWidth ( int column, int minSize )
```

yang artinya fungsi setColumnMinimumWidth ini gunanya untuk mengatur ukuran lebar minimum untuk kolom yang ditentukan. Maka, kode saya tadi maksudnya aturlah pada kolom ketiga, ukuran lebar 100 piksel.

LIMA

```
QSize size(88,88);
tombolEmpat->setMinimumSize(100,100);
tombolEmpat->setIconSize(size);
tombolEmpat->setIcon(QIcon(":/gambar/ubuntu.png"));
```

Nah, untuk kode ini, sengaja saya akhirkan penjelasannya. Pertama, dibuat sebuah objek yang menampung suatu dimensi piksel x piksel. Objek itu bernama size yang diturunkan dari kelas QSize. Ini unik, khas OOP. Segalanyaaa… objek. Bahkan sampai yang namanya dimensi pun dibuat sebagai objek. Sesudah itu, tombolEmpat diatur ukuran minimumnya lalu nah. Inilah gunanya. Dengan method milik PushButton yang bernama setIconSize, objek size tadi dimasukkan ke sini. Jadi kita tidak menuliskan setIconSize(88,88) melainkan setIconSize(size). Gunanya untuk baris berikutnya. Method setIcon gunanya untuk mengatur ikon dari tombol. Bisa ditebak, saya memang ingin membuat aplikasi yang berbasis gambar lagi. Begitu gambar dimasukkan, karena tadi sudah diatur dengan setIconSize, maka otomatis nanti kalau dijalankan programnya, ikon yang ditentukan di sini pasti berukuran 88 x 88. Mudah dipahami, bukan? Yang unik dari method terakhir ini adalah setIcon(QIcon(“”));. Perhatikan, ada method di dalam method. Dan di sini dipakailah kelas QIcon. Saya sendiri tidak seberapa paham mengapa harus begini (tetapi saya mengerti juga buat apa?) yang penting bisa masuk gambarnya. Perhatikan lagi, :/gambar/ubuntu.png itu adalah aturan Qt Creator dalam mengimpor gambar. Lihat tulisan pertama saya soal Qt bila belum mengerti soal impor gambar.

ENAM

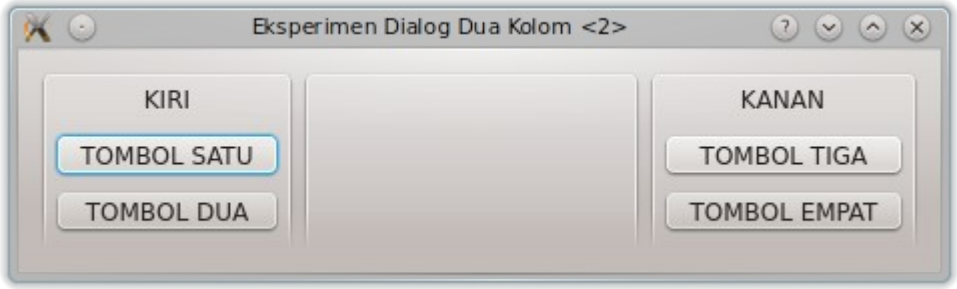
```
frameTengah->setFrameShadow(QFrame::Raised);
frameTengah->setFrameShape(QFrame::StyledPanel);
```

Khusus untuk baris ke-58 kode, saya jadikan komentar. Seandainya diaktifkan, maka nanti frameTengah akan kelihatan border-nya. Bentuk border yang seperti ini keren, tetapi saya tidak inginkan dalam program kali ini. Mungkin nanti pas perlu baru dipakai. Yang penting, perhatikan passing parameter-nya. QFrame::Raised untuk setFrameShadow dan QFrame::StyledPanel untuk setFrameShape. Begitu cara menulis passing parameter-nya. Anda bisa mencarinya dalam dokumentasi internal Qt Creator untuk jenis shadow dan jenis framestyle lainnya.

Hasil Akhir

Ini hasil yang saya inginkan, yakni ada spasi kosong antargrup. Sebagai bonus, kita berhasil membuat tombol dengan gambar sekaligus mengatur sekehendak kita ukurannya.

Bandingkan dengan program kemarin yang masih kelihatan spasi antargrupnya:



Rangkuman

1. Grid Layout adalah yang paling bebas yang bisa dipakai untuk meletakkan objek-objek GUI ke samping dan ke bawah.
2. Untuk mengatur ukuran gambar dalam tombol, kita bisa gunakan QSize untuk menampung ukuran dan memasukkannya ke dalam passing parameter dalam method setIconSize.