

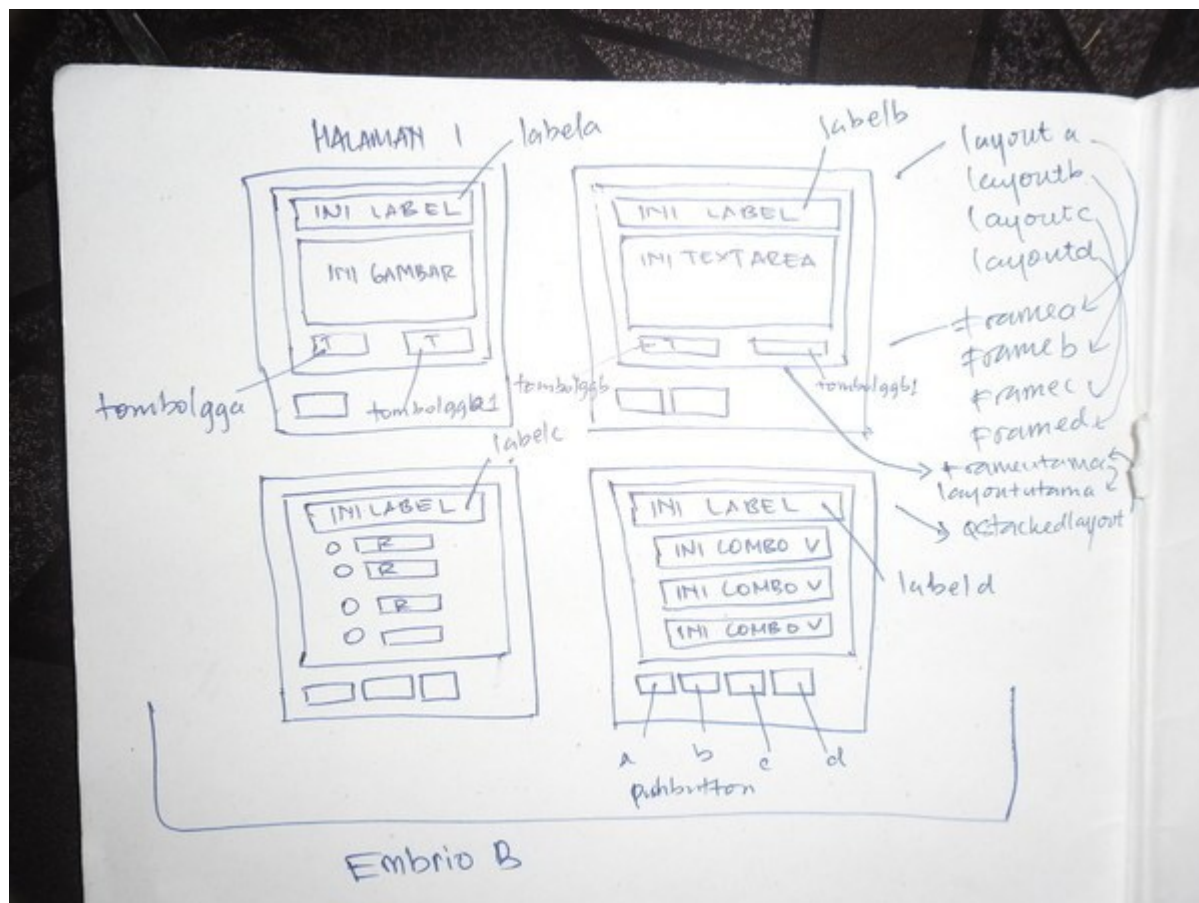
Pemrograman Qt 6 – Membuat Program GUI Sederhana Menggunakan Sebanyak Mungkin Kelas (QStackedLayout, QPushButton, QLabel, QComboBox, dan Lainnya)

Bismillahirrahmanirrahim.

Tulisan ini adalah bentuk PDF dari <http://malsasa.wordpress.com/2013/08/25/pemrograman-qt-6-membuat-program-gui-sederhana-menggunakan-sebanyak-mungkin-kelas-qstackedlayout-qpushbutton-qlabel-qcomboBox-dan-lainnya/>.

Alhamdulillah. Setelah kemarin saya menuai keberhasilan layering sebagaimana yang saya harap, kini saatnya berlatih memperluas program yang sudah bisa layering dengan macam-macam kelas yang saya belum kenal. Ini untuk memperlancar membuat program. Di sini saya membuat satu program dengan 4 layer. Setiap layer diwakili oleh satu tombol. Bedanya, kali ini saya unggah kode sumbernya lengkap di dalam paket ZIP. Silakan diunduh dan dibuka di Qt Creator Anda. Semoga ini bermanfaat.

Mockup



Tidak seperti biasanya, saya kali ini menggambar mockup-nya dahulu di kertas. Ini penting karena akan susah sekali program ditulis kalau harus mengimajinasikan semuanya. Lebih

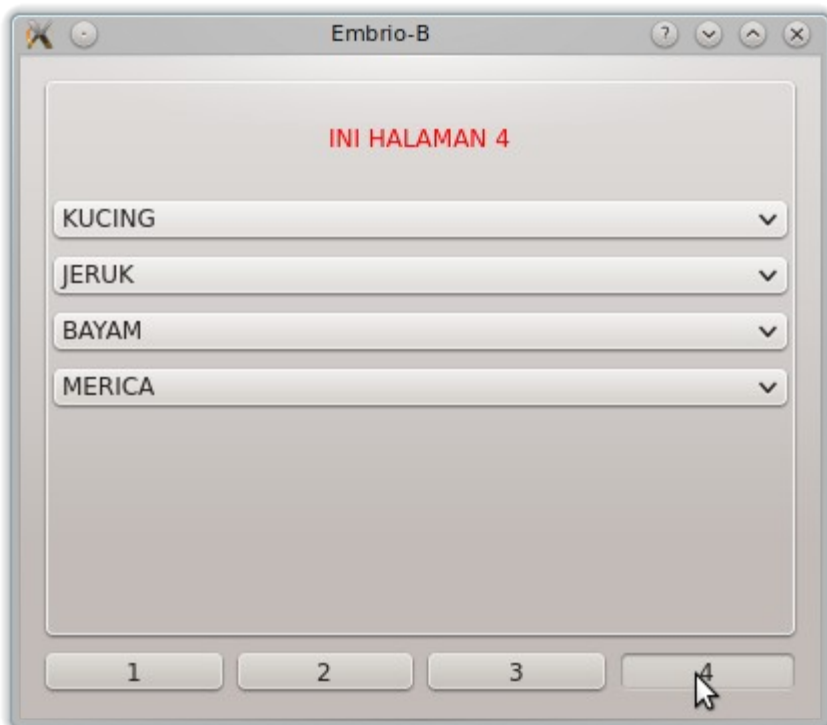
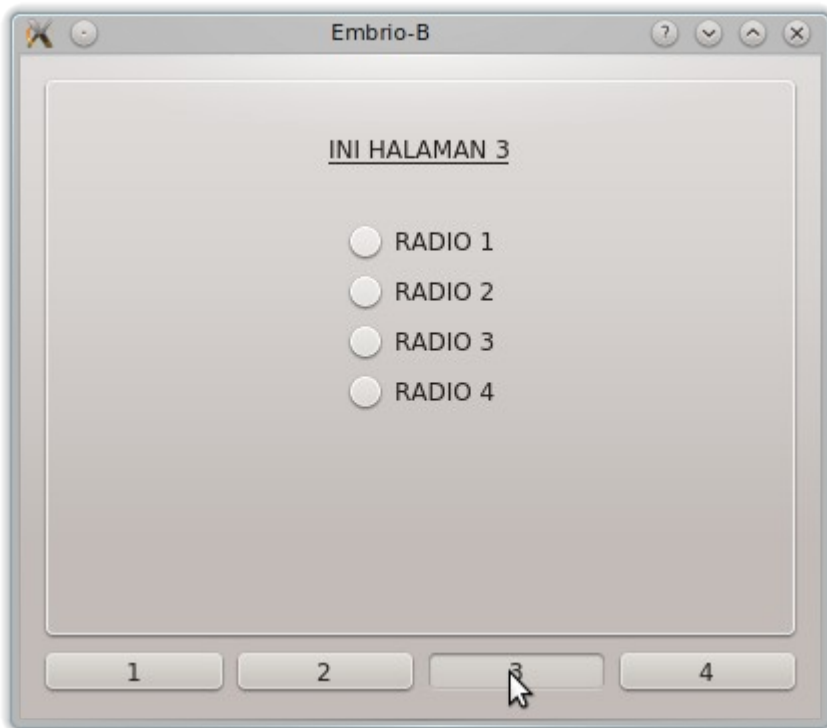
mudah kalau digambar dulu desainnya lalu nanti tinggal menulis kode sesuai gambar.
Lakukan ini juga kala Anda ingin membuat program besar.

Wujud Program yang Diinginkan



Saya cinta <http://ilmoe.com>.





Daftar Kelas Qt yang Dipakai

1. QVBoxLayout
2. QHBoxLayout
3. QStackedLayout
4. QFrame
5. QPushButton
6. QComboBox
7. QLabel
8. QPixmap
9. QRadioButton

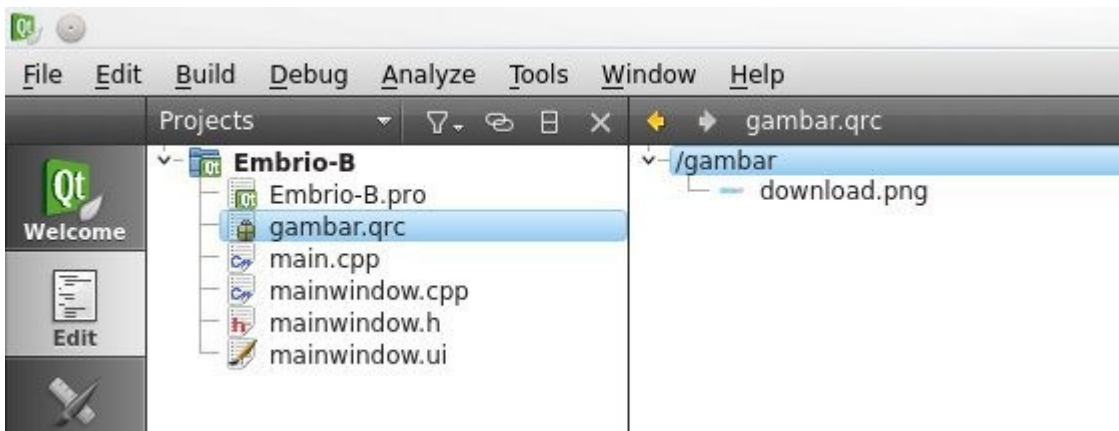
Daftar Method Qt yang Dipakai

1. setReadOnly (baru)
2. setText (baru)
3. setPixmap (baru)
4. addWidget
5. setAlignment (baru)
6. addLayout
7. setContentsMargins (baru)
8. addItem (baru)
9. setLayout
10. setFrameShape
11. setFrameShadow
12. setLayout
13. connect
14. setCurrentIndex

Qt Creator dalam Kode

```
52 //SELESAI DEKLARASI PEMBUATAN OBJEK
53 //deklarasi labela dengan tulisan
54
55
56 labela->setText("<b>INI HALAMAN 1</b>"); //GILA, Qt bisa men-parsing HTML dalam string dalam parameter sekaligus
57 labelb->setText("<i>INI HALAMAN 2</i>"); //Qt benar-benar enak
58 labelc->setText("<u>INI HALAMAN 3</u>"); //tag yang diterima <b> <i> <u> <font>
59 labeld->setText("<font color='red'>INI HALAMAN 4</font>"); //hanya QLabel dan QTextEdit saja, kelas lain tidak menerima HTML
60
61 //deklarasi gambar untuk labelgambar
62
63 labelgambar->setPixmap(QPixmap(":/gambar/download.png")); //ditemukan pada 23 Agustus 2013
64 //sumber: http://qt-project.org/forums/viewthread/15365
65 //ternyata cuma begini saja deklarasinya, gak usah kelas QImage atau QPixmap atau QPixmap sana sekali
66
67 //deklarasi pengisian layouts dengan item-item halaman 1
68
69 layouta->addWidget(labela);
70 layouta->setAlignment(labela, Qt::AlignCenter);
71 layouta->addWidget(labelgambar);
72 layoutf->addWidget(tombolga);
73 layoutf->addWidget(tombolgg);
74 layouta->addLayout(layoutf);
75 layouta->setContentsMargins(1,1,1,55); //ukuran terakhir adalah margin vertikal bawah
76
77 //deklarasi pengisian layout dengan item-item halaman 2
78
79 layoutb->addWidget(labelb);
80 layoutb->setAlignment(labelb, Qt::AlignCenter);
81 layoutb->addWidget(textedit);
82 layoutb->addWidget(tombolgg);
83 layoutb->addWidget(tombolggb);
84
85 //deklarasi pengisian layout dengan item-item halaman 3
86
87 layoutc->addWidget(labelc);
88 layoutc->setAlignment(labelc, Qt::AlignTop);
89 layoutc->setAlignment(labelc, Qt::AlignCenter);
90 layoutc->addWidget(radiob);
91 layoutc->setAlignment(radiob, Qt::AlignTop);
92 layoutc->setAlignment(radiob, Qt::AlignCenter);
93 layoutc->addWidget(radiob);
94 layoutc->setAlignment(radiob, Qt::AlignTop);
```

```
25 QFrame *frame;
26
27 QLabel *labela;
28 QLabel *labelgambar;
29 QLabel *labelb;
30 QLabel *labelc;
31 QLabel *labeld;
32
33 QPushButton *tombola;
34 QPushButton *tombolb;
35 QPushButton *tombolc;
36 QPushButton *tombold;
37
38 QPushButton *tombolga; //gg = gak guna
39 QPushButton *tombolgg;
40 QPushButton *tombolggb;
41 QPushButton *tombolggbi;
42
43 QTextEdit *textedit;
44
45 QRadioButton *radiob;
46 QRadioButton *radiob;
47 QRadioButton *radioc;
48 QRadioButton *radiod;
49
50 QComboBox *comboa;
51 QComboBox *combob;
52 QComboBox *combo;
53 QComboBox *combod;
54
55 QStackedLayout *stack;
56
57 public slots:
58 void gantia();
59 void gantib();
60 void gantid();
61 void gantid();
62
63 private:
64
65 };
66
67 #endif // MAINWINDOW_H
```



Kode Program

mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include

class Dialog : public QDialog
{
    Q_OBJECT

public:
    Dialog();

    QVBoxLayout      *layoututama;
    QVBoxLayout      *layouta;
    QVBoxLayout      *layoutb;
    QVBoxLayout      *layoutc;
    QVBoxLayout      *layoutd;
    QHBoxLayout      *layoute;
    QHBoxLayout      *layoutf;

    QFrame           *frameutama;
    QFrame           *framea;
    QFrame           *frameb;
    QFrame           *framec;
    QFrame           *framed;

    QLabel           *labela;
    QLabel           *labelagambar;
    QLabel           *labelb;
    QLabel           *labelc;
    QLabel           *labeld;

    QPushButton     *tombola;
    QPushButton     *tombolb;
    QPushButton     *tombolc;
    QPushButton     *tombold;

    QPushButton     *tombolgga;           //gg = gak guna
    QPushButton     *tombolgga1;
    QPushButton     *tombolggb;
    QPushButton     *tombolggb1;

    QTextEdit       *textedit;

    QRadioButton    *radioa;
```



```
    QRadioButton    *radiob;
    QRadioButton    *radioc;
    QRadioButton    *radiod;

    QComboBox       *comboa;
    QComboBox       *combob;
    QComboBox       *comboc;
    QComboBox       *combod;

    QStackedLayout  *stack;

public slots:
    void            gantia();
    void            gantib();
    void            gantic();
    void            gantid();

private:

};

#endif // MAINWINDOW_H
```

mainwindow.cpp

```
#include "mainwindow.h" //IDE yang dipakai adalah
Qt Creator
#include "ui_mainwindow.h"
//http://malsasa.wordpress.com

Dialog::Dialog()
{

    //deklarasi semua objek dahulu

    layoututama = new QVBoxLayout; //layout
    layouta = new QVBoxLayout;
    layoutb = new QVBoxLayout;
    layoutc = new QVBoxLayout;
    layoutd = new QVBoxLayout;
    layoute = new QHBoxLayout;
    layoutf = new QHBoxLayout;

    framea = new QFrame; //frame
    frameb = new QFrame;
    framec = new QFrame;
    framed = new QFrame;

    labela = new QLabel; //label
    labelagambar = new QLabel;
    labelb = new QLabel;
    labelc = new QLabel;
    labeld = new QLabel;

    tombola = new QPushButton("1"); //pushbutton
    tombolb = new QPushButton("2");
    tombolc = new QPushButton("3");
    tombold = new QPushButton("4");

    tombolgga = new QPushButton("GGA"); //pushbutton gak
guna
    tombolgga1 = new QPushButton("GGA 1");
    tombolggb = new QPushButton("GGB");
    tombolggb1 = new QPushButton("GGB 1");

    radioa = new QRadioButton("RADIO 1"); //radio
button
    radiob = new QRadioButton("RADIO 2");
    radioc = new QRadioButton("RADIO 3");
    radiod = new QRadioButton("RADIO 4");

    textedit = new QTextEdit("<span style='text-
decoration: underline;'><i><b>INI ADALAH TEXTEDIT</b></i></span>");
//QTextEdit bisa mem-parsing HTML
```

```

textedit->setReadOnly(true);

comboa          = new      QComboBox;           //combo
box
combob          = new      QComboBox;
comboc          = new      QComboBox;
combod          = new      QComboBox;

stack           = new      QStackedLayout;
//primadona kita

//SELESAI DEKLARASI PEMBUATAN OBJEK
//deklarasi labela dengan tulisan

labela->setText("<b>INI HALAMAN 1</b>");           //GILA, Qt bisa
mem-parsing HTML dalam string dalam parameter sekalipun
labelb->setText("<i>INI HALAMAN 2</i>");           //Qt benar-benar
enak
labelc->setText("<span style='text-decoration: underline;'>INI
HALAMAN 3</span>");           //tag yang diterima <b> <i> <span
style='text-decoration: underline;'>
labeld->setText("<span style='color: red;'>INI HALAMAN
4</span>");           //hanya QLabel dan QTextEdit saja, kelas lain tidak
menerima HTML

//deklarasi gambar untuk labelagambar

labelagambar-
>setPixmap(QPixmap(":/gambar/download.png"));           //ditemukan pada
23 Agustus 2013
//sumber: http://qt-project.org/forums/viewthread/15365
//ternyata cuma begini saja deklarasinya, gak usah kelas QImage
atau QPixmap atau QPixmap sama sekali

//deklarasi pengisian layouta dengan item-item halaman 1

layouta->addWidget(labela);
layouta->setAlignment(labela, Qt::AlignHCenter);
layouta->addWidget(labelagambar);
layoutf->addWidget(tombolgga);
layoutf->addWidget(tombolgga1);
layouta->addLayout(layoutf);
layouta->setContentsMargins(1,1,1,55);           //ukuran terakhir
adalah margin vertikal bawah

//deklarasi pengisian layoutb dengan item-item halaman 2

layoutb->addWidget(labelb);
layoutb->setAlignment(labelb, Qt::AlignHCenter);
layoutb->addWidget(textedit);
layoutb->addWidget(tombolggg);

```

```

layoutb->addWidget(tombolggb1);

//deklarasi pengisian layoutc dengan item-item halaman 3

layoutc->addWidget(labelc);
layoutc->setAlignment(labelc, Qt::AlignTop);
layoutc->setAlignment(labelc, Qt::AlignHCenter);
layoutc->addWidget(radioa);
layoutc->setAlignment(radioa, Qt::AlignTop);
layoutc->setAlignment(radioa, Qt::AlignHCenter);
layoutc->addWidget(radiob);
layoutc->setAlignment(radiob, Qt::AlignTop);
layoutc->setAlignment(radiob, Qt::AlignHCenter);
layoutc->addWidget(radioc);
layoutc->setAlignment(radioc, Qt::AlignTop);
layoutc->setAlignment(radioc, Qt::AlignHCenter);
layoutc->addWidget(radiod);
layoutc->setAlignment(radiod, Qt::AlignTop);
layoutc->setAlignment(radiod, Qt::AlignHCenter);
layoutc->setContentsMargins(1,1,1,111); //ini harus ada

//deklarasi pengisian layoutd dengan item-item halaman 4

layoutd->addWidget(labeld);
layoutd->setAlignment(labeld, Qt::AlignHCenter);
layoutd->addWidget(comboa);
layoutd->addWidget(combob);
layoutd->addWidget(comboc);
layoutd->addWidget(combod);
layoutd->setContentsMargins(1,1,1,111);

//begini cara mengisi item-item combo box
//kalau pakai addItem tambah bingung karena
//harus pakai objek List, jika tidak maka gagal

comboa->addItem("KUCING"); //ingat, addItem bukan
addItem
combob->addItem("AYAM");
combob->addItem("SAPI");

combob->addItem("JERUK");
combob->addItem("STROBERI");
combob->addItem("APEL");

comboc->addItem("BAYAM");
comboc->addItem("KANGKUNG");
comboc->addItem("KUBIS");

combod->addItem("MERICHA");
combod->addItem("GARAM");
combod->addItem("BAWANG PUTIH");

```

```

//kemudian deklarasi pemasangan antara frame dengan layout

framea->setLayout(layouta);
frameb->setLayout(layoutb);
framec->setLayout(layoutc);
framed->setLayout(layoutd);

//deklarasi frame agar terlihat wujudnya

framea->setFrameShape(QFrame::StyledPanel);
frameb->setFrameShape(QFrame::StyledPanel);
framec->setFrameShape(QFrame::StyledPanel);
framed->setFrameShape(QFrame::StyledPanel);

framea->setFrameShadow(QFrame::Raised);
frameb->setFrameShadow(QFrame::Raised);
framec->setFrameShadow(QFrame::Raised);
framed->setFrameShadow(QFrame::Raised);

//memasukkan objek sebagai layer

stack->addWidget(framea);
stack->addWidget(frameb);
stack->addWidget(framec);
stack->addWidget(framed);

//mengisi layoute dengan tombol-tombol supaya bisa horizontal
persis mockup

layoute->addWidget(tombola);
layoute->addWidget(tombolb);
layoute->addWidget(tombolc);
layoute->addWidget(tombold);

//memasukkan layout kecil ke dalam layout utama

layoututama->addLayout(stack);
layoututama->addLayout(layoute);

//memasang layoututama ke dalam Dialog

setLayout(layoututama);

//memberi fungsi untuk setiap tombol

connect(tombola, SIGNAL(clicked()), this,
SLOT(gantia())); //A semua
connect(tombolb, SIGNAL(clicked()), this,
SLOT(gantib())); //B semua
connect(tombolc, SIGNAL(clicked()), this,

```

```

SLOT(gantic()));          //C semua
    connect(tombold, SIGNAL(clicked()), this,
SLOT(gantid()));        //D semua
    //cara baca kode ini: kalau tombol c diklik, maka Dialog
memanggil fungsi gantic
    //receiver-nya harus "this", tidak boleh "Dialog", tidak juga
"stack"

}

//deklarasi fungsi-fungsi untuk tombol

void Dialog::gantia()
{
    stack->setCurrentIndex(0);
}

void Dialog::gantib()
{
    stack->setCurrentIndex(1);
}

void Dialog::gantic()
{
    stack->setCurrentIndex(2);          //jika Dialog memanggil
fungsi ini, maka
}
    //stack layer nomor 2 akan maju ke depan

void Dialog::gantid()
{
    stack->setCurrentIndex(3);
}

```

Analisis Kode Program

mainwindow.h

Tidak ada yang berbeda dari posting nomor 5 kemarin. Pokoknya, semua objek harus dideklarasikan di sini dahulu karena akan digunakan dalam layer dan ada fungsi yang memanggil fungsi di luar tubuhnya.

mainwindow.cpp

Di sini banyak yang penting dan berbeda. Yang sama dengan kemarin adalah deklarasi ulang semua kelas yang sudah dideklarasikan tetapi bentuk deklarasi harus pendek namaObjeknya = new NamaKelasnya; tanpa tanda pointer (*) dan tanpa NamaKelas depan. Yang berbeda dari kemarin karena adanya kelas-kelas baru semisal QTextEdit (untuk menulis teks), QLabel, QPixmap, dan lain-lain, di antaranya:

QTextEdit

```
textedit          =    new    QTextEdit ("<u><i><b>INI ADALAH  
TEXTEDIT</b></i></u>");  
textedit->setReadOnly(true);
```

Inilah deklarasi objek QTextEdit. Sama saja dengan kelas yang lain. Ajaibnya, ia menerima HTML di dalam string di dalam parameter dan mem-parsing-nya saat runtime. Ini sangat mengesankan saya bahwa Qt melakukan hal yang sangat memudahkan saya memformat nantinya. Kalau begini, mungkin saya bisa menulis ulang Otodidak dalam Qt dengan memasukkan gambar sekaligus tulisan (dan tulisan itu bisa miring bisa tebal) di dalam satu kotak saja. Qt bisa mem-parsing HTML untuk QTextEdit.

Maksud dari method setReadOnly adalah menjadikan objeknya QTextEdit nanti tidak bisa ditulis lagi. Hanya bisa dibaca saja. Itu kalau diset nilainya true. Perhatikan, apa-apa method yang mengubah perilaku suatu objek, seringkali namanya diawali dengan set. Jadi, kita akan lebih cepat kerja dengan Qt Creator kala mencari method karena tahu tanda-tandanya.

QLabel

```
labela->setText ("<b>INI HALAMAN 1</b>");  
labelb->setText ("<i>INI HALAMAN 2</i>");  
labelc->setText ("<u>INI HALAMAN 3</u>");  
labeld->setText ("<font color='red'>INI HALAMAN 4</font>");
```

Ini pengisian teks untuk QLabel. Digunakanlah method setText milik QLabel. Ajaibnya lagi, parameter string di dalam QLabel menerima HTML dan mem-parsing-nya ketika runtime. Ini pun sangat menguntungkan saya. Saya bisa mewarnai dan membayangkan tulisan dalam

label sesuka saya. Ini akan sangat membantu untuk kita membantu UI dan UX yang baik untuk aplikasi. Ini (wallahu a'lam) belum saya temukan kala menggunakan Java platform (Netbeans).

QPixmap

```
labelagambar->setPixmap(QPixmap(":/gambar/download.png"));
```

Ini adalah tindakan mengisi QLabel dengan gambar. Tentu gambar sudah diimpor terlebih dahulu ke dalam proyek. Lihat posting sebelumnya untuk cara mengimpor gambar. Caranya ternyata dengan method setPixmap miliknya QLabel. Lalu di dalam parameternya dipanggilah QPixmap dengan parameter lagi berupa string, yang string itu berisi path ke gambar yang diimpor. Gambarnya adalah banner <http://ilmoe.com>, website favorit saya.

QVBoxLayout & QHBoxLayout

```
layouta->addWidget(labela);  
layouta->setAlignment(labela, Qt::AlignHCenter);  
layouta->addWidget(labelagambar);  
layoutf->addWidget(tombolgga);  
layoutf->addWidget(tombolgga1);  
layouta->addLayout(layoutf);  
layouta->setContentsMargins(1, 1, 1, 55);
```

Ini meskipun sudah Anda ketahui semuanya, masih penting karena menata objek-objek ke dalam layout. Di sini yang berlaku sebagai layout vertikal adalah layouta dan yang horizontal itu layoutf. Yang vertikal untuk judul dan gambar, yang horizontal untuk dua tombol GG. Lihat foto mockup di atas pada HALAMAN 1.

Yang unik di sini adalah method setAlignment milik QVBoxLayout. Di dalam parameternya ada dua buah isi. Yang pertama adalah objek/widget, dan yang kedua adalah kode alignment (perataan). Isi yang pertama ini adalah labela dan Qt::AlignHCenter. Maksud kode ini adalah jadikan labela itu Center secara horizontal, pada layouta. Seperti ini juga untuk objek-objek lain. Inilah cara kita untuk menata supaya objek-objek dalam layout itu rapi. Kalau tidak begini, lihat saja. Nanti objek-objek Anda bisa awut-awutan letaknya.

Method setContentsMargins ini saya gunakan untuk menarik semua objek ke atas. Karena jika tidak begitu, semua objek terlihat turun mendekati garis paling bawah. Ini tidak rapi. Makanya saya kasih angka 55 supaya ada jarak antara objek paling bawah dengan garis frame bawah. Hasilnya HALAMAN 1 terlihat rapi. Semua objek berada pas di tengah-tengah frame.

QComboBox

```
comboa->addItem ("KUCING");  
comboa->addItem ("AYAM");  
comboa->addItem ("SAPI");  
  
combob->addItem ("JERUK");  
combob->addItem ("STROBERI");  
combob->addItem ("APEL");  
  
comboc->addItem ("BAYAM");  
comboc->addItem ("KANGKUNG");  
comboc->addItem ("KUBIS");  
  
combod->addItem ("MERICA");  
combod->addItem ("GARAM");  
combod->addItem ("BAWANG PUTIH");
```

Inilah cara untuk mengisi tiap-tiap combo box dengan opsi. Hasilnya bisa dilihat pada skrinshot berikut:



Untuk kode-kode setelah baris 118 ini, walhamdulillah sudah saya tulis keterangannya pada posting nomor 5.

Rangkuman

1. Kelas yang bisa meng-handle HTML adalah QLabel dan QTextEdit (yang lainnya belum diketahui). Yang jelas, QPushButton tidak bisa meng-handle HTML.
2. Memasukkan gambar ke dalam QLabel sangat mudah, tidak usah membuat objek baru. Cukup deklarasikan: `label->setPixmap(QPixmap(":/prefix/namagambar.png"));`
3. Ukuran horizontal layer pada semua layer pada QStackedLayout mengikuti ukuran isi layer terpanjang.
4. Kode-kode alignment untuk parameter kedua dalam `setAlignment` milik `QVBoxLayout/QHBoxLayout` adalah `Qt::AlignCenter`, `Qt::AlignHCenter`, `Qt::AlignVCenter`, `Qt::AlignJustify`, `Qt::AlignBottom`, `Qt::AlignRight`, dan `Qt::AlignLeft`. Bisa dipilih sendiri dengan Qt Creator.

Unduh

Kode Sumber: <http://otodidak.freemove.me/tarball/Embrio-B.tar.gz>

Ukuran: 34 KB