

# Pemrograman Qt 8

## QVBoxLayout, QHBoxLayout, QGridLayout dan Sifat-Sifatnya

Ade Malsasa Akbar

1 Desember 2013

### Ringkasan

Bismillahirrahmanirrahim. Kelebihan pemrograman visual adalah adanya layout yang secara otomatis akan mengatur penataan apa pun objek yang kita tuliskan di jendela. Pendek kata, kita tidak perlu lagi mengatur ukuran matematis (X,Y) dari setiap objek GUI satu per satu karena sudah diotomatisasi oleh layout. Di Java SWING, saya menemukan FreeLayout, GridLayout, BorderLayout, AbsoluteLayout, dan lain-lain. Di Qt Framework ini, saya menemukan QVBoxLayout, QHBoxLayout, QGridLayout, QFormLayout, dan lain-lain pula. Saya akan membahas 3 layout Qt yang sudah saya gunakan dalam latihan-latihan saya kemarin dan ini bisa Anda pakai dalam program apa pun. Penting, bisa diletakkan lebih dari 1 jenis layout dalam 1 jendela. Semoga bermanfaat untuk Anda.

# Daftar Isi

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Teori</b>   | <b>4</b>  |
| 1.1      | Manfaat Nyata Layout . . . . .                                       | 4         |
| 1.2      | Ke Mana Arah Tulisan Ini? . . . . .                                  | 5         |
| <b>2</b> | <b>Pemrograman</b>   | <b>6</b>  |
| 2.1      | Menggunakan Layout Horizontal . . . . .                              | 6         |
| 2.2      | Kode . . . . .   | 6         |
| 2.2.1    | mainwindow.cpp . . . . .   | 6         |
| 2.2.2    | mainwindow.h . . . . .   | 6         |
| 2.2.3    | Qt Creator dan Kode . . . . .  | 7         |
| 2.2.4    | Hasil Layout Horizontal . . . . .                                    | 8         |
| <b>3</b> | <b>Menggunakan Layout Vertikal</b>                                   | <b>9</b>  |
| 3.1      | Kode . . . . .   | 9         |
| 3.1.1    | mainwindow.cpp . . . . .   | 9         |
| 3.1.2    | mainwindow.h . . . . .   | 9         |
| 3.1.3    | Qt Creator dan Kode . . . . .  | 9         |
| 3.1.4    | Hasil Layout Vertikal . . . . .                                      | 10        |
| <b>4</b> | <b>Menggunakan Grid Layout</b>                                       | <b>10</b> |
| 4.1      | Kode . . . . .   | 11        |
| 4.1.1    | mainwindow.cpp . . . . .   | 11        |
| 4.1.2    | mainwindow.h . . . . .   | 11        |
| 4.1.3    | Hasil Grid Layout . . . . .  | 12        |
| <b>5</b> | <b>Modifikasi Terhadap QHBoxLayout, QVBoxLayout, dan QGridLayout</b> | <b>13</b> |
| 5.1      | HBoxLayout . . . . .   | 13        |
| 5.1.1    | Penjelasan Kode QHBoxLayout . . . . .                                | 15        |
| 5.2      | VBoxLayout . . . . .   | 16        |
| 5.2.1    | Penjelasan QVBoxLayout . . . . .                                     | 19        |
| 5.3      | GridLayout . . . . .   | 20        |
| 5.3.1    | Qt Creator dan Kode . . . . .  | 22        |
| 5.3.2    | Penjelasan GridLayout . . . . .                                      | 22        |
| <b>6</b> | <b>Rangkuman</b>   | <b>23</b> |
| <b>7</b> | <b>Unduh Kode Sumber</b>   | <b>24</b> |
| <b>8</b> | <b>Tentang Dokumen Ini</b>   | <b>24</b> |
| <b>9</b> | <b>Tentang Penulis</b>   | <b>24</b> |

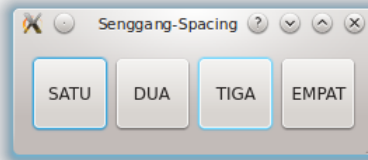
## Daftar Gambar

|    |  |    |
|----|--|----|
| 1  | Contoh Manfaat <i>Layout</i> . . . . .                                   | 4  |
| 2  | Contoh Penerapan <i>Layout</i> Vertikal . . . . .                        | 4  |
| 3  | Qt Creator Menunjukkan Kode <code>QHBoxLayout</code> . . . . .           | 7  |
| 4  | <code>QHBoxLayout</code> dengan 4 Tombol . . . . .                       | 8  |
| 5  | Maximized <code>QHBoxLayout</code> . . . . .                             | 8  |
| 6  | <code>QVBoxLayout</code> . . . . .                                       | 10 |
| 7  | Maximized <code>QVBoxLayout</code> . . . . .                             | 10 |
| 8  | Rencana <code>QGridLayout</code> . . . . .                               | 11 |
| 9  | Hasil <code>QGridLayout</code> . . . . .                                 | 12 |
| 10 | Maximized <code>QGridLayout</code> . . . . .                             | 12 |
| 11 | <code>QHBoxLayout</code> Asli . . . . .                                  | 13 |
| 12 | <code>QHBoxLayout</code> Hasil Modifikasi . . . . .                      | 14 |
| 13 | <code>QVBoxLayout</code> Asli . . . . .                                  | 16 |
| 14 | <code>QVBoxLayout</code> Asli Ketika Maximized . . . . .                 | 16 |
| 15 | <code>QVBoxLayout</code> Setelah Dimodifikasi . . . . .                  | 17 |
| 16 | <code>QVBoxLayout</code> Maximized Setelah Dimodifikasi . . . . .        | 18 |
| 17 | <code>QGridLayout</code> . . . . .                                       | 20 |
| 18 | Maximized <code>QGridLayout</code> . . . . .                             | 20 |
| 19 | Kode <code>QGridLayout</code> Modifikasi dalam Qt Creator . . . . .      | 22 |
| 20 | Parameter <code>addWidget</code> dari <code>QGridLayout</code> . . . . . | 23 |

# 1 Teori

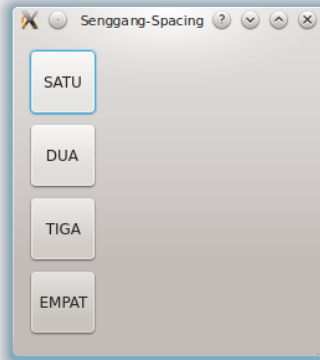
## 1.1 Manfaat Nyata Layout

Misalkan Anda ingin membuat program dengan empat tombol berjajar ke samping seperti gambar ini:



Gambar 1: Contoh Manfaat *Layout*

maka tidak perlulah Anda menulis kode peletakan x sekian y sekian untuk masing-masing tombol dan sampai berjam-jam memosisikan ukurannya biar simetris seperti gambar. Tidak perlu. Cukup buat objek layout horizontal lalu masukkan kode tombol-tombol ke dalam layout itu. Otomatis si layout horizontal akan mengatur tombol-tombol Anda berjajar ke samping. Setiap Anda masukkan satu tombol ke dalam layout, ia akan diletakkan persis di sebelah kanan tombol sebelumnya. Otomatis. Begitu pula dengan kebutuhan penataan tombol yang vertikal, atau kebutuhan penataan yang seperti keramik-keramik lantai (grid), atau malah ingin bebas menentukan ukuran secara manual semuanya, juga bisa. Gambar ini contoh program yang memakai layout vertikal:



Gambar 2: Contoh Penerapan *Layout* Vertikal

Nah, sangat praktis, bukan? Layout di Qt (maupun di framework lain) itu berguna sekali. Dan apakah hanya untuk tombol? Tentu tidak. Seluruh objek GUI (QWidget) seperti QComboBox akan diperlakukan sama seperti tombol. Maka jadi makin praktislah memrogram GUI dengan layout di Qt. Pasti lebih praktis.

## **1.2 Ke Mana Arah Tulisan Ini?**

Kita akan latihan saja kali ini. Pokoknya kita akan:

1. memilih dan membuat layout,
2. memasang tombol ke dalam layout,
3. melihat hasilnya secara normal dan maximized,
4. memodifikasi ukuran-ukuran yang tersedia pada layout, dan
5. menggabung 3 jenis layout dalam 1 jendela.

## 2 Pemrograman

### 2.1 Menggunakan Layout Horizontal

Ini dilakukan dengan kelas QHBoxLayout. Kita akan membuat program seperti gambar pertama di atas.

### 2.2 Kode

#### 2.2.1 mainwindow.cpp

---

```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 Dialog::Dialog()
5 {
6     QHBoxLayout *layoututama = new QHBoxLayout;
7     QPushButton *tombolsatu = new QPushButton("SATU");
8     QPushButton *tomboldua = new QPushButton("DUA");
9     QPushButton *tomboltiga = new QPushButton("TIGA");
10    QPushButton *tombolempat = new QPushButton("EMPAT");
11
12    tombolsatu->setFixedSize(55,55);
13    tomboldua->setFixedSize(55,55);
14    tomboltiga->setFixedSize(55,55);
15    tombolempat->setFixedSize(55,55);
16
17    layoututama->addWidget(tombolsatu);
18    layoututama->addWidget(tomboldua);
19    layoututama->addWidget(tomboltiga);
20    layoututama->addWidget(tombolempat);
21
22    setLayout(layoututama);
23
24 }
```

---

#### 2.2.2 mainwindow.h

---

```
1 #ifndef MAINWINDOW_H
2 #define MAINWINDOW_H
3
4 #include
5
6 class Dialog : public QDialog
7 {
8     Q_OBJECT
```

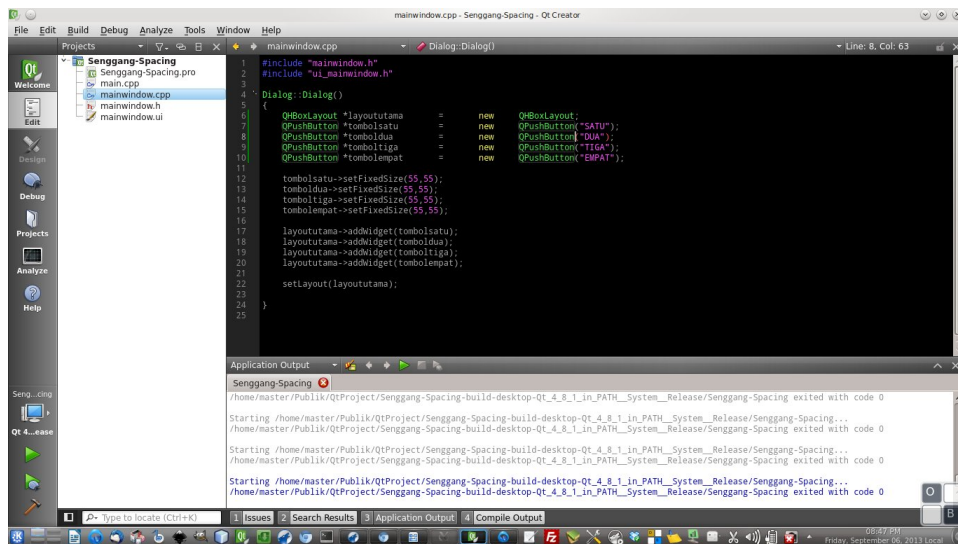
```

9
10 public:
11     Dialog();
12
13 private:
14
15 };
16
17 #endif // MAINWINDOW_H
18 }

```

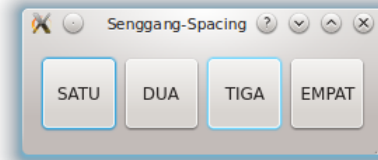
---

### 2.2.3 Qt Creator dan Kode

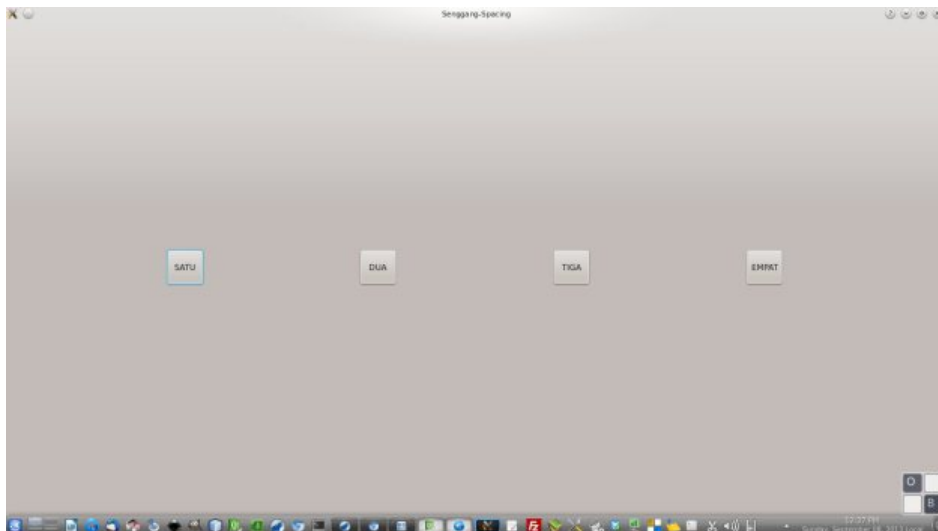


Gambar 3: Qt Creator Menunjukkan Kode QHBoxLayout

#### 2.2.4 Hasil Layout Horizontal



Gambar 4: QHBoxLayout dengan 4 Tombol



Gambar 5: Maximized QHBoxLayout

Perhatikan, tombol-tombol itu jadi renggang dan otomatis centered ketika jendela maximized. Ini sifat QHBoxLayout yang pertama.



## 3 Menggunakan Layout Vertikal

Ini dilakukan dengan kelas QVBoxLayout. Kita akan membuat program seperti gambar dua.

### 3.1 Kode

#### 3.1.1 mainwindow.cpp

---

```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 Dialog::Dialog()
5 {
6     QVBoxLayout *layoututama = new QVBoxLayout;
7     QPushButton *tombolsatu = new QPushButton("SATU");
8     QPushButton *tomboldua = new QPushButton("DUA");
9     QPushButton *tomboltiga = new QPushButton("TIGA");
10    QPushButton *tombolempat = new QPushButton("EMPAT");
11
12    tombolsatu->setFixedSize(55,55);
13    tomboldua->setFixedSize(55,55);
14    tomboltiga->setFixedSize(55,55);
15    tombolempat->setFixedSize(55,55);
16
17    layoututama->addWidget(tombolsatu);
18    layoututama->addWidget(tomboldua);
19    layoututama->addWidget(tomboltiga);
20    layoututama->addWidget(tombolempat);
21
22    setLayout(layoututama);
23
24 }
```

---

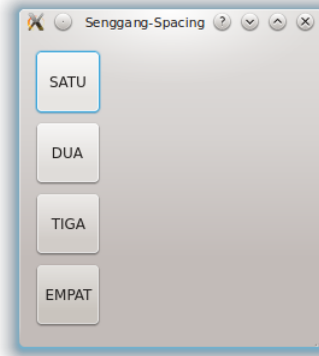
#### 3.1.2 mainwindow.h

Sama dengan yang horizontal.

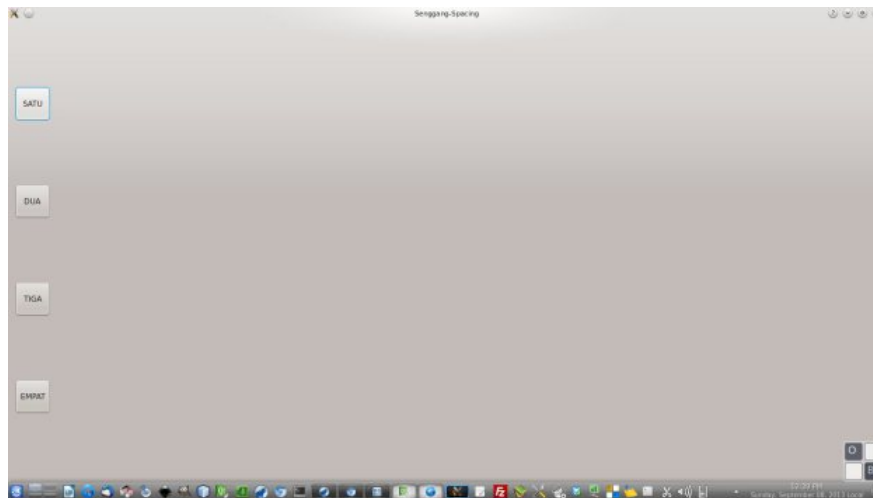
#### 3.1.3 Qt Creator dan Kode

Sama saja dengan yang horizontal, bedanya cuma pemakaian QVBoxLayout di sini.

### 3.1.4 Hasil Layout Vertikal



Gambar 6: QVBoxLayout



Gambar 7: Maximized QVBoxLayout

Perhatikan, tombol-tombol itu jadi renggang, centered on Y axis, dan align left ketika jendela maximized. Align left ini ciri dari QVBoxLayout yang default.

## 4 Menggunakan Grid Layout

Grid Layout di Qt Framework adalah jalan kebebasan yang sangat praktis untuk membangun GUI yang *semau gue*. Ini jenis layout paling fleksibel yang

dapat Anda gunakan untuk aplikasi yang kompleks. Ini dilakukan dengan menggunakan kelas `QGridLayout`. Kita akan buat seperti gambar ini:



Gambar 8: Rencana `QGridLayout`

## 4.1 Kode

### 4.1.1 `mainwindow.cpp`

Sama saja dengan di atas, hanya ganti `QVBoxLayout` dengan `QGridLayout`.

### 4.1.2 `mainwindow.h`

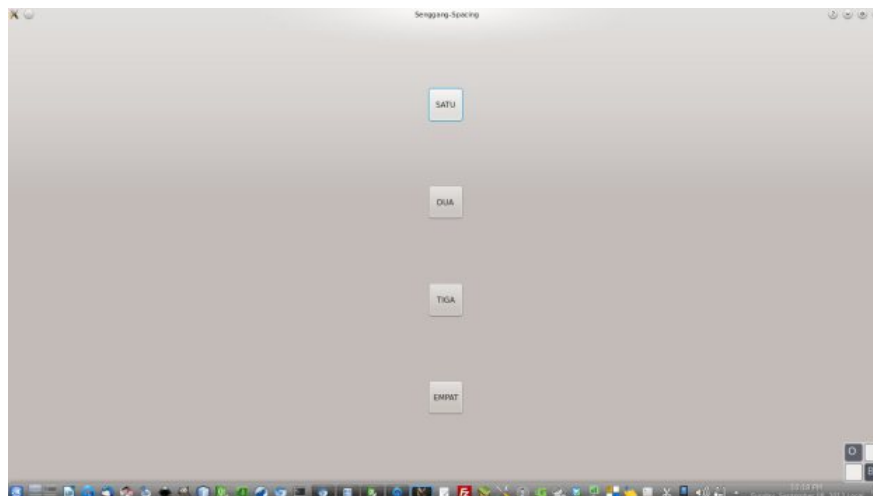
Sama juga dengan header sebelumnya.

### 4.1.3 Hasil Grid Layout

Sekilas mirip QVBoxLayout karena default sudah vertikal. Ini belum kelihatan manfaat aslinya QGridLayout.



Gambar 9: Hasil QGridLayout



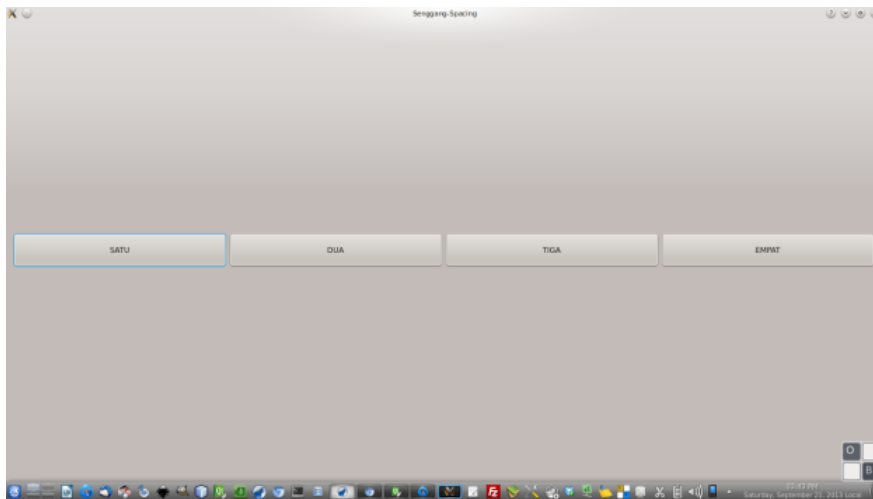
Gambar 10: Maximized QGridLayout

Perhatikan, sekilas memang mirip QVBoxLayout walau sudah maximized. Bedanya, ia di center in Y dan center in X. Sedangkan QVBoxLayout defaultnya cuma center in X.

## 5 Modifikasi Terhadap QHBoxLayout, QVBoxLayout, dan GridLayout

Sampai saatnya bagian terpenting yakni modifikasi. Ini tujuannya untuk mengetahui sifat-sifat dari kelas layout yang bersangkutan. Jika tidak dimodifikasi, kita tahunya hanya begitu doang QHBoxLayout, QVBoxLayout, dan GridLayout. Padahal ada seabreg kemampuan dari layout tersebut yang bisa jadi sangat kita perlukan tapi kita tidak tahu.

### 5.1 QHBoxLayout



Gambar 11: QHBoxLayout Asli

Sifatnya memanjang ke samping secara simetris kalau di-maximize, tidak memanjang ke atas.

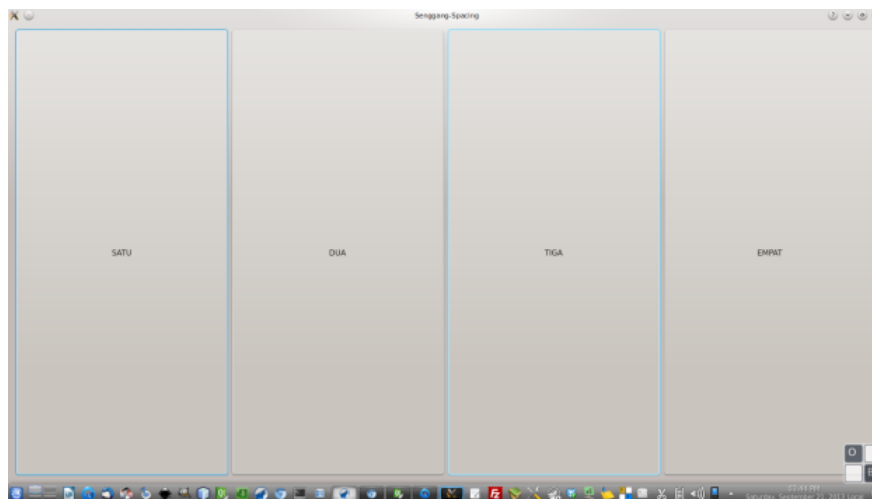
```
1 //mainwindow.cpp
2 #include "mainwindow.h"
3 #include "ui_mainwindow.h"
4
5 Dialog::Dialog()
6 {
7     QHBoxLayout *layoututama = new QHBoxLayout;
8     QPushButton *tombolsatu = new QPushButton("SATU");
9     QPushButton *tomboldua = new QPushButton("DUA");
10    QPushButton *tomboltiga = new QPushButton("TIGA");
11    QPushButton *tombolempat = new QPushButton("EMPAT");
12
13    tombolsatu->setMinimumSize(55,55);
```

```

14     tomboldua->setMinimumSize(55,55);
15     tomboltiga->setMinimumSize(55,55);
16     tombolempat->setMinimumSize(55,55);
17
18     layoututama->addWidget(tombolsatu);
19     layoututama->addWidget(tomboldua);
20     layoututama->addWidget(tomboltiga);
21     layoututama->addWidget(tombolempat);
22
23     setLayout(layoututama);
24 }

```

Listing 1: Kode QHBoxLayout Asli



Gambar 12: QHBoxLayout Hasil Modifikasi

Sifatnya memanjang ke samping dan ke atas simetris kalau di-maximize, mirip bagian modem di Otodidak.

```

1 //mainwindow.cpp
2 #include "mainwindow.h"
3 #include "ui_mainwindow.h"
4
5 Dialog::Dialog()
6 {
7     QHBoxLayout *layoututama = new QHBoxLayout;
8     QPushButton *tombolsatu = new QPushButton("SATU");
9     QPushButton *tomboldua = new QPushButton("DUA");
10    QPushButton *tomboltiga = new QPushButton("TIGA");
11    QPushButton *tombolempat = new QPushButton("EMPAT");
12

```

```

13     tombolSATU->setMinimumSize(55,55);
14     tombolDUA->setMinimumSize(55,55);
15     tombolTIGA->setMinimumSize(55,55);
16     tombolEMPAT->setMinimumSize(55,55);
17
18     tombolSATU->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::
    Expanding);
19     tombolDUA->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::
    Expanding);
20     tombolTIGA->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::
    Expanding);
21     tombolEMPAT->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::
    Expanding);
22
23     layoutUTAMA->addWidget(tombolSATU);
24     layoutUTAMA->addWidget(tombolDUA);
25     layoutUTAMA->addWidget(tombolTIGA);
26     layoutUTAMA->addWidget(tombolEMPAT);
27
28     setLayout(layoutUTAMA);
29 }

```

---

Listing 2: Kode QHBoxLayout Hasil Modifikasi

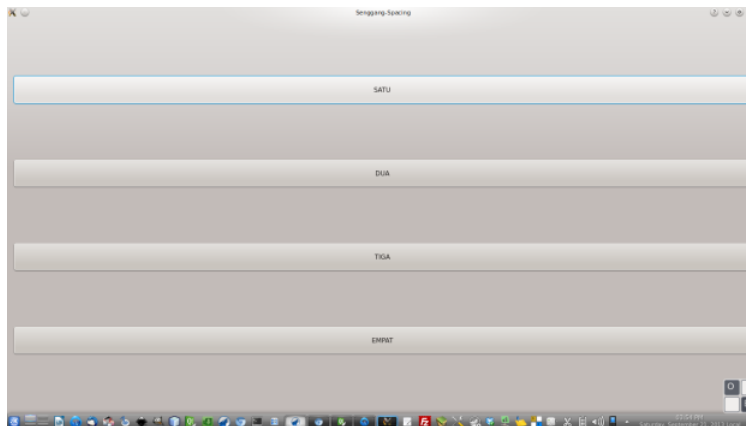
### 5.1.1 Penjelasan Kode QHBoxLayout

Saya ganti fungsi `setFixedSize` dengan `setMinimumSize` sehingga ketika jendela di-maximize, si tombol ikut membesar mengikuti ukuran jendela. Saya tambahkan fungsi `setSizePolicy` pada masing-masing tombol sehingga bisa `expanding` ke atas juga mengikuti besarnya ukuran jendela.

## 5.2 QVBoxLayout



Gambar 13: QVBoxLayout Asli



Gambar 14: QVBoxLayout Asli Ketika Maximized

Memanjang ke samping tetapi tidak ke atas sekaligus ketika di-maximize.

```
1 //mainwindow.cpp
2 #include "mainwindow.h"
3 #include "ui_mainwindow.h"
4
5 Dialog::Dialog()
```

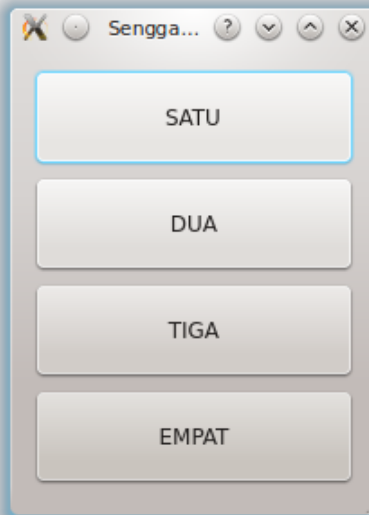


```

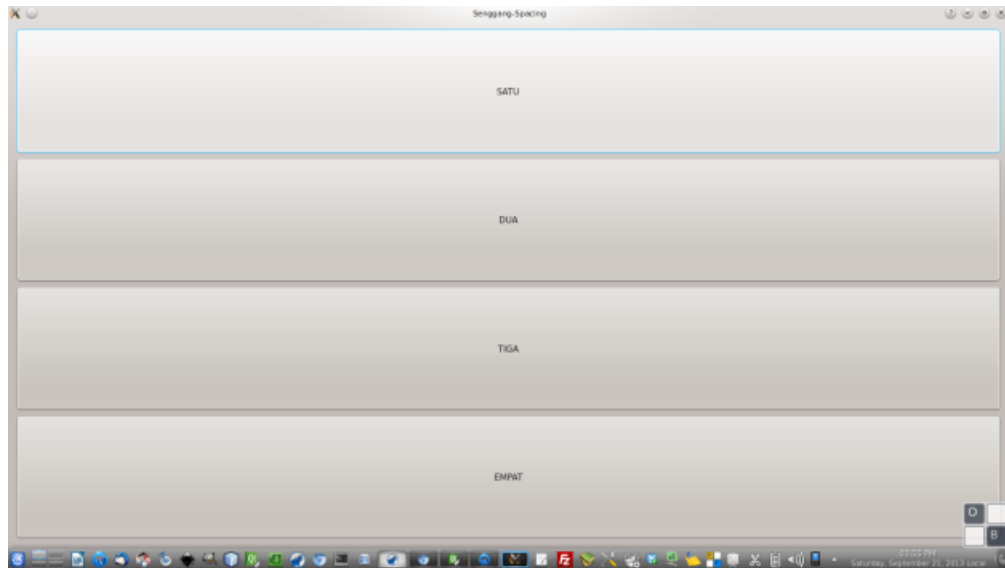
6  {
7      QVBoxLayout *layoututama      =      new      QVBoxLayout;
8      QPushButton *tombolsatu       =      new      QPushButton("SATU");
9      QPushButton *tomboldua        =      new      QPushButton("DUA");
10     QPushButton *tomboltiga        =      new      QPushButton("TIGA");
11     QPushButton *tombolempat       =      new      QPushButton("EMPAT");
12
13     tombolSATU->setMinimumSize(55,55);
14     tombolDUA->setMinimumSize(55,55);
15     tombolTIGA->setMinimumSize(55,55);
16     tombolempat->setMinimumSize(55,55);
17
18     layoututama->addWidget(tombolsatu);
19     layoututama->addWidget(tomboldua);
20     layoututama->addWidget(tomboltiga);
21     layoututama->addWidget(tombolempat);
22
23     setLayout(layoututama);
24 }

```

Listing 3: Kode QVBoxLayout Asli



Gambar 15: QVBoxLayout Setelah Dimodifikasi



Gambar 16: QVBoxLayout Maximized Setelah Dimodifikasi

```
1 //mainwindow.cpp
2 #include "mainwindow.h"
3 #include "ui_mainwindow.h"
4
5 Dialog::Dialog()
6 {
7     QVBoxLayout *layoututama = new QVBoxLayout;
8     QPushButton *tombolsatu = new QPushButton("SATU");
9     QPushButton *tomboldua = new QPushButton("DUA");
10    QPushButton *tomboltiga = new QPushButton("TIGA");
11    QPushButton *tombolempat = new QPushButton("EMPAT");
12
13    tombolsatu->setMinimumSize(55,55);
14    tomboldua->setMinimumSize(55,55);
15    tomboltiga->setMinimumSize(55,55);
16    tombolempat->setMinimumSize(55,55);
17
18    tombolsatu->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::
19    Expanding);
20    tomboldua->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::
21    Expanding);
22    tomboltiga->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::
23    Expanding);
24    tombolempat->setSizePolicy(QSizePolicy::Expanding, QSizePolicy::
25    Expanding);
26}
```

```
23     layoututama->addWidget(tombolsatu);
24     layoututama->addWidget(tomboldua);
25     layoututama->addWidget(tomboltiga);
26     layoututama->addWidget(tombolempat);
27
28     setLayout(layoututama);
29 }
```

---

Listing 4: QVBoxLayout Hasil Modifikasi

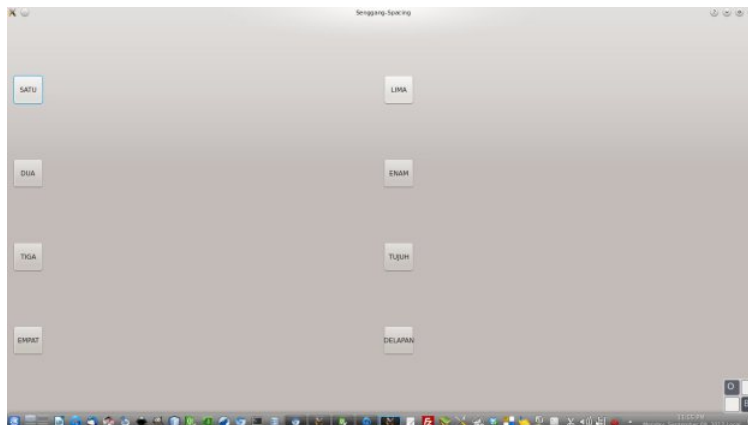
### 5.2.1 Penjelasan QVBoxLayout

- Saya ganti fungsi `setFixedSize` dengan `setMinimumSize` sehingga ketika jendela di-maximize, si tombol ikut membesar mengikuti ukuran jendela.
- Saya tambahkan fungsi `setSizePolicy` pada masing-masing tombol sehingga bisa `expanding` ke atas juga mengikuti besarnya ukuran jendela.

### 5.3 QGridLayout



Gambar 17: QGridLayout



Gambar 18: Maximized QGridLayout

Perhatikan, inilah sisi kekuatan QGridLayout. Kita bisa mengatur tombol-tombol sehingga membentuk kolom dan baris. Perhatikan juga sifatnya kalau di-maximize.

---

```
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 Dialog::Dialog()
5 {
6     QGridLayout *layoututama = new QGridLayout;
```

```

7   QPushButton *tombolsatu = new QPushButton("SATU");
8   QPushButton *tomboldua = new QPushButton("DUA");
9   QPushButton *tomboltiga = new QPushButton("TIGA");
10  QPushButton *tombolempat = new QPushButton("EMPAT");
11  QPushButton *tombollima = new QPushButton("LIMA");
12  QPushButton *tombolenam = new QPushButton("ENAM");
13  QPushButton *tomboltujuh = new QPushButton("TUJUH");
14  QPushButton *tomboldelapan = new QPushButton("DELAPAN");
15
16  //ukuran tombol
17
18  tombolsatu->setFixedSize(55,55);
19  tomboldua->setFixedSize(55,55);
20  tomboltiga->setFixedSize(55,55);
21  tombolempat->setFixedSize(55,55);
22
23  tombollima->setFixedSize(55,55);
24  tombolenam->setFixedSize(55,55);
25  tomboltujuh->setFixedSize(55,55);
26  tomboldelapan->setFixedSize(55,55);
27
28  //mengatur peletakan di grid
29
30  layoututama->addWidget(tombolsatu);
31  layoututama->addWidget(tomboldua);
32  layoututama->addWidget(tomboltiga);
33  layoututama->addWidget(tombolempat);
34
35  layoututama->addWidget(tombollima, 0, 1);
36  layoututama->addWidget(tombolenam, 1, 1);
37  layoututama->addWidget(tomboltujuh, 2, 1);
38  layoututama->addWidget(tomboldelapan, 3, 1);
39
40  //pengaturan untuk penataan
41
42  layoututama->setAlignment(tombolsatu, Qt::AlignLeft);
43  layoututama->setAlignment(tomboldua, Qt::AlignLeft);
44  layoututama->setAlignment(tomboltiga, Qt::AlignLeft);
45  layoututama->setAlignment(tombolempat, Qt::AlignLeft);
46
47  layoututama->setAlignment(tombollima, Qt::AlignLeft);
48  layoututama->setAlignment(tombolenam, Qt::AlignLeft);
49  layoututama->setAlignment(tomboltujuh, Qt::AlignLeft);
50  layoututama->setAlignment(tomboldelapan, Qt::AlignLeft);
51
52  setLayout(layoututama);
53
54 }

```

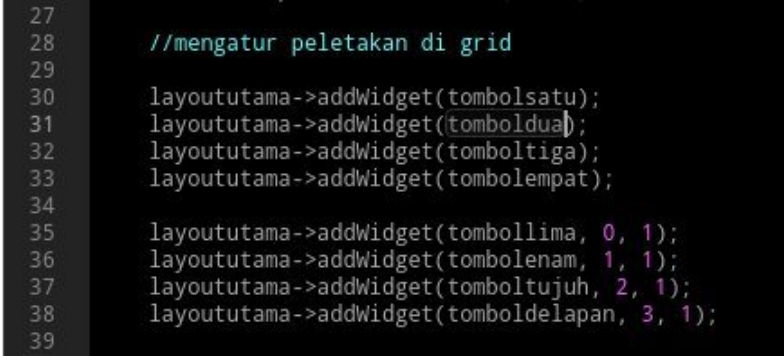
---

Listing 5: Kode Program QGridLayout Modifikasi

Kode paling pentingnya:

```
1 //mengatur peletakan di grid
2
3 layoututama->addWidget(tombolsatu);
4 layoututama->addWidget(tomboldua);
5 layoututama->addWidget(tomboltiga);
6 layoututama->addWidget(tombolempat);
7
8 layoututama->addWidget(tombollima, 0, 1);
9 layoututama->addWidget(tombolenam, 1, 1);
10 layoututama->addWidget(tomboltujuh, 2, 1);
11 layoututama->addWidget(tomboldelapan, 3, 1);
12 }
```

### 5.3.1 Qt Creator dan Kode



```
27
28 //mengatur peletakan di grid
29
30 layoututama->addWidget(tombolsatu);
31 layoututama->addWidget(tomboldua);
32 layoututama->addWidget(tomboltiga);
33 layoututama->addWidget(tombolempat);
34
35 layoututama->addWidget(tombollima, 0, 1);
36 layoututama->addWidget(tombolenam, 1, 1);
37 layoututama->addWidget(tomboltujuh, 2, 1);
38 layoututama->addWidget(tomboldelapan, 3, 1);
39
```

Gambar 19: Kode QGridLayout Modifikasi dalam Qt Creator

### 5.3.2 Penjelasan QGridLayout

Yang perlu diperhatikan, QGridLayout itu punya fungsi `addWidget` yang berbeda bentuk dari `addWidget` pada kelas-kelas yang lain. Mengapa bisa berbeda? Ya karena sifat QGridLayout itu sendiri yang membuat penataan tombol kita (nantinya tidak cuma tombol) bisa membentuk kolom-kolom dan baris-baris. Seperti ini bentuk khususnya:



Gambar 20: Parameter addWidget dari QGridLayout

Jadi, dengan QGridLayout, kita mengatur objek-objek GUI kita (tombol) berdasarkan kolom dan baris. Bisa dikata, posisi x dan y. Oleh karena itu, kode saya di atas yang berbunyi:

---

```
1 layoututama->addWidget(tombollima, 0, 1);
2 layoututama->addWidget(tombolenam, 1, 1);
3 layoututama->addWidget(tomboltujuh, 2, 1);
4 layoututama->addWidget(tomboldelapan, 3, 1);
```

---

maksudnya:

- letakkanlah objek tumbollima pada baris ke-0 kolom ke-1
- letakkan objek tombolenam, pada baris ke-1 kolom ke-1, dan seterusnya.

## 6 Rangkuman

- QGridLayout adalah layout paling fleksibel yang bisa kita gunakan untuk membuat antarmuka sebebas mungkin.
- Untuk membuat tombol bisa membesar mengikuti ukuran jendela, bisa digunakan setSizePolicy dengan parameter **QSizePolicy::Expanding** pada kedua tempatnya.

## 7 Unduh Kode Sumber

Program kali ini bernama Senggang-Spacing.

- Alamat: <http://otodidak.freesever.me/tarball/Senggang-Spacing.tar.gz>
- Ukuran: 3 KB

## 8 Tentang Dokumen Ini

Dokumen ini adalah versi PDF dari posting asli <http://malsasa.wordpress.com/2013/08/30/pemrograman-qt-8-qvboxlayout-qhboxlayout-qgridlayout-dan-sifat-sifatnya/>. Dokumen ini disusun ulang dengan L<sup>A</sup>T<sub>E</sub>X dengan antarmuka Gummi di atas Ubuntu 12.04. Dokumen ini adalah bagian dari usaha kecil memperbanyak panduan Linux dalam Bahasa Indonesia. Dokumen ini tidaklah bebas dari kesalahan yang membuat penulis memohon maaf dan mengajak Anda untuk menulis dokumen yang lebih baik.

Penulis berusaha membuat versi PDF ini semirip mungkin dengan versi HTML. Namun penulis menemukan bahwa kode-kode sumber dalam dokumen ini bisa di saling-tempel dengan sempurna pada Evince PDF Viewer dan tidak sempurna (membutuhkan sedikit penyuntingan tambahan) pada Okular PDF Viewer. Maka penulis menyarankan Anda untuk menggunakan Evince atau menulis sendiri tiap-tiap kode sumber yang ada atau mengunduh langsung versi paket kode sumbernya pada alamat yang tersebut di atas. Penulis memohon maaf atas ketidaknyamanan ini.

Catatan teknis keterbatasan: listings, fontenc, [T1], accsupp, minted (ada bug pada Gummi), Pygments.

## 9 Tentang Penulis

Penulis adalah warga Forum Ubuntu Indonesia dan aktif di forum-forum Linux lain di Indonesia. Penulis mendukung pendidikan perangkat lunak legal (terutama FOSS) untuk masyarakat. Penulis menyediakan buku-buku panduan Linux untuk pemula maupun ahli untuk diunduh secara gratis<sup>1</sup>. Penulis bisa dihubungi via SMS di nomor 0896 7923 7257.

---

<sup>1</sup><http://malsasa.wordpress.com/pdf>