

Python + Qt = PyQt

Stop using terminal and start loving GUI

Me

You may know me as `Fakhri || FAI || f4r4w4y`.

A software developer that loves experimenting with tech stuff.

GUI ? Terminal ? 🤔

GUI = **G**raphical **U**ser **I**nterface.

Terminal menggunakan CLI = **C**ommand **L**ine **I**nterface.

Qt vs PyQt ?

Its a GUI framework berbasis C++ dan QML (Qt Modelling Language).

Berbeda dengan Qt itu sendiri, PyQt merupakan sebuah Python wrapper bagi Qt agar kita bisa membuat tampilan berbasis Qt dengan menggunakan bahasa Python.

Why Qt ?

Because its cute-y (?) :v

Dikarenakan banyaknya juga vendor yang menggunakan Qt dan memang memiliki kapabilitas untuk membuat UI yang bagus maka Qt menjadi salah satu GUI framework yang harus diperhitungkan saat pemilihan GUI framework.

But what about PySide2 ?

<https://www.learnpyqt.com/tutorials/pyqt5-vs-pyside2/>

Seperti yang dijelaskan di link atas, sebenarnya keduanya sama sama sebuah wrapper python untuk Qt Framework.

Perbedaan intinya adalah PyQt dibuat oleh Phil Thompson dari [Riverbank Computing Ltd](#) sementara PySide merupakan library asli buatan Qt nya sendiri.

Jadi, which side you want to choose ?

Why not both ?

As proposed in the learnpyqt.com, kita sebenarnya bisa seakan menggunakan 2 framework diatas dengan menggunakan wrapper tambahan.

Dan terlebih sudah ada github repo yang persis menggunakan teknik serupa, yaitu [QtPy](#) (its funny isn't it ? :v)

Installation (PyQt)

Semuanya sudah tertera pada link berikut

<https://pythonbasics.org/install-pyqt/>

Hello World

The old classical way of teaching how to program :D

```
from PyQt5.QtWidgets import *  
  
app = QApplication([])  
  
w = QWidget(windowTitle='Hello World')  
w.show()  
  
app.exec_()
```

Hello World (full version)

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel
from PyQt5.QtGui import QIcon

def window():
    app = QApplication(sys.argv)
    widget = QWidget()
    widget.setWindowTitle("PyQt5 Example")
    widget.show()
    sys.exit(app.exec_())

if __name__ == '__main__':
    window()
```

QWidget vs QMainWindow vs QDialog

referenced from stackoverflow.com

A QWidget is the base class for all drawable classes in Qt.
Any QWidget-based class can be shown as a window by showing it when it has no parent.

A QDialog is based on QWidget, but designed to be shown as a window.
It will always appear in a window, and has functions to make it
work well with common buttons on dialogs (accept, reject, etc.).

QMainWindow is designed around common needs for a main window to have.
It has predefined places for a menu bar, a status bar, a toolbar, and other widgets.
It does not have any built-in allowances for buttons like QDialog does.

Button

```
button1 = QPushButton(widget)
button1.setText("Button1")
button1.clicked.connect(fungsi_apa_aja)
```

`fungsi_apa_aja` bisa terletak dimana saja asalkan bisa dipanggil oleh file python bersangkutan [ini dinamakan `slot`].

Dan triggering function atau pada contoh `clicked` merupakan sebuah `signal`.

MessageBox

```
QMessageBox.critical(  
    self,  
    "Failed",  
    "Please input only an integer value for first and fourth index.",  
)
```

Bisa untuk mengoutputkan sebuah popup alert entah itu success alert atau failed alert pada aplikasinya.

And many more

Bisa dilihat saja referensi di internet untuk contoh lainnya, terlebih lagi dokumentasi nya sudah sangat lengkap untuk melihatkan contoh dan cara penggunaannya.

Development Option(s)

1. Handcode all the UI together with business logic code
2. Create the UI in Qt Designer and integrate with business logic code

Qt Designer (optional)

Berguna untuk membuat UI dengan mudah tanpa perlu mengkhawatirkan kodingan, karena kodingannya bisa autogenerated dari hasil UI yang dibuat.

Installation (Qt Designer)

Untuk windows =

<https://stackoverflow.com/questions/30222572/how-to-install-qt designer>

Untuk linux =

<https://gist.github.com/ujjwal96/1dcd57542bdaf3c9d1b0dd526ccd44ff>

Untuk mac os = `~_(\ツ)_/\`` (i dont have any of 'em)

pyuic

Gunakan pyuic untuk mengenerate kodingan pyqt dari layout yang telah dibuat pada Qt Designer sebelumnya, menggunakan command berikut.

```
pyuic5 design_nya.ui -o kodingan_nya.py -x
```

opsi `-x` digunakan untuk menghasilkan kodingan yang bisa dijalankan sekaligus memperlihatkan hasil layout yang sudah dibuatnya.

Reference

[Written tutorial from learnpyqt](#)

[Written tutorial from pythonbasics.org](#)

[Mastering PyQt by Alan D Moore Codes](#)

[PyQt5 Tutorial by Tech With Tim](#)

Thanks...

Silahkan tanya bila ada yang ingin ditanyakan.

Mohon maaf bila ada kesalahan.

