Pertemuan 13

PEMROGRAMAN

GUI dengan PyQt

KONSEP PyQt

» PyQt adalah modul Python untuk framework aplikasi Qt yang berfungsi untuk menciptakan GUI, yang berjalan di semua platform yang didukung oleh Qt. Modul.
» PyQt5 merupakan modul lengkap Python untuk Qt v5.
» PyQt5 terdapat lebih dari 35 modul ekstensi dan memungkinkan Python untuk digunakan sebagai bahasa pengembangan aplikasi alternatif untuk C++.

Modul PyQt harus melakukan instalasi menggunakan program pip, dengan menuliskan perintah berikut ini:

```
pip install PyQt5
```

Berikut contoh program penggunaan PyQt di python:

Membuat form sederhana dalam PyQt

```
# Membuat Program GUI dalam PyQt

import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel

if __name__ == '__main__':
    a=QApplication(sys.argv)

    form=QWidget()
    form.setGeometry(100,150,350,250)
    form.setWindowTitle('Program GUI dalam PyQt')

    label=QLabel()
    label.setText('Halo Teman, Bagaimana kabarmu?')
    label.move(140,100)
    label.setParent(form)

    form.show()
    a.exec_()
```

Membuat Grid Layout dalam PyQt

```python
# Membuat GridLayout dengan Gaya berorientasi Objek

import sys
from PyQt5.QtWidgets import *

class MainForm(QWidget):
    def __init__(self):
        super() .__init__()
        self.setupUi()

    def setupUi(self):
        self.resize(250,100)
        self.setWindowTitle('Demo GridLayout')
        self.label1=QLabel('Username')
        self.usernameEdit=QLineEdit()
        self.label2=QLabel('Password')
        self.passwordEdit=QLineEdit()
        self.loginButton=QPushButton('Login')
        self.exitButton=QPushButton('Exit')

        grid=QGridLayout()
        grid.addWidget(self.label1,0,0)
        grid.addWidget(self.usernameEdit,0,1,1,2)
        grid.addWidget(self.label2,1,0)
        grid.addWidget(self.passwordEdit,1,1,1,2)
        grid.addWidget(self.loginButton,2,1)
        grid.addWidget(self.exitButton,2,2)

        self.setLayout(grid)

if __name__ == '__main__':
    a=QApplication(sys.argv)
    form=MainForm()
    form.show()
    a.exec_()
```

Membuat Slot untuk menangani Signal

```python
# Membuat Slot untuk Menangani Signal

import sys
from PyQt5.QtWidgets import *

class MainForm(QWidget):
    def __init__(self):
        super().__init__()
        self.setupUi()

    def setupUi(self):
        self.resize(250,100)
        self.setWindowTitle('Demo Signal dan Slot')

        self.label=QLabel('Masukkan nama Anda')
        self.nameEdit=QLineEdit()
        self.haloButton=QPushButton('Halo')

        vbox=QVBoxLayout()
        vbox.addWidget(self.label)
        vbox.addWidget(self.nameEdit)
        vbox.addWidget(self.haloButton)

        self.setLayout(vbox)
        self.haloButton.clicked.connect(self.on_haloButton_clicked)

    def on_haloButton_clicked(self):
        name=self.nameEdit.text()
        QMessageBox.information(self,'Halo','Hai %s, apa kabar?' % name)

if __name__ == '__main__':
    a=QApplication(sys.argv)
    form=MainForm()
    form.show()
    a.exec_()
```

Menampilkan dialog pesan

```python
# Menampilkan dialog pesan

import sys
from PyQt5.QtWidgets import *

class MainForm(QWidget):
    def __init__(self):
        super().__init__()
        self.setupUi()

    def setupUi(self):
        self.resize(350,100)
        self.setWindowTitle('Demo QMessageBox')
        self.informationButton=QPushButton('Informasi')
        self.errorButton=QPushButton('Kesalahan')
        self.warningButton=QPushButton('Peringatan')
        self.confirmButton=QPushButton('Konfirmasi')

        hbox=QHBoxLayout()
        hbox.addWidget(self.informationButton)
        hbox.addWidget(self.errorButton)
        hbox.addWidget(self.warningButton)
        hbox.addWidget(self.confirmButton)

        self.setLayout(hbox)
        self.informationButton.clicked.connect(self.on_informationButton_clicked)
        self.errorButton.clicked.connect(self.on_errorButton_clicked)
        self.warningButton.clicked.connect(self.on_warningButton_clicked)
        self.confirmButton.clicked.connect(self.on_confirmButton_clicked)

    def on_informationButton_clicked(self):
        QMessageBox.information(self,'Informasi','Data telah tersimpan di database')

    def on_errorButton_clicked(self):
        QMessageBox.critical(self,'Kesalahan','File config.ini tidak ditemukan')

    def on_warningButton_clicked(self):
        QMessageBox.warning(self,'Peringatan','Semua data dalam flashdisk akan dihapus')

    def on_confirmButton_clicked(self):
        dialog=QMessageBox.question(self,'Konfirmasi','Hapus baris data ini?')

        if dialog==QMessageBox.Yes:
            print('Anda memilih YES')
        else:
            print('Anda memilih NO')

if __name__ == '__main__':
    a=QApplication(sys.argv)
    form=MainForm()
    form.show()
    a.exec_()
```

Membuat kalkulator

```
# Membuat kalkulator

import sys
from PyQt5.QtWidgets import *

class MainForm(QWidget):
    def __init__(self):
        super().__init__()
        self.setupUi()

    def setupUi(self):
        self.resize(250,200)
        self.setWindowTitle('Demo Kalkulator')

        self.label1=QLabel('Bilangan ke-1')
        self.num1Edit=QLineEdit()
        self.label2=QLabel('Bilangan ke-2')
        self.num2Edit=QLineEdit()
        self.label3=QLabel('Hasil Perhitungan')
        self.resultEdit=QLineEdit()
        self.addButton=QPushButton('Tambah')
        self.subtractButton=QPushButton('Kurang')
        self.mulButton=QPushButton('Kali')
        self.divButton=QPushButton('Bagi')

        vbox=QVBoxLayout()
        vbox.addWidget(self.label1)
        vbox.addWidget(self.num1Edit)
        vbox.addWidget(self.label2)
        vbox.addWidget(self.num2Edit)
        vbox.addWidget(self.label3)
        vbox.addWidget(self.resultEdit)
        vbox.addStretch()

        hbox=QHBoxLayout()
        hbox.addWidget(self.addButton)
        hbox.addWidget(self.subtractButton)
        hbox.addWidget(self.mulButton)
        hbox.addWidget(self.divButton)

```

```
41          mainLayout=QVBoxLayout()
42          mainLayout.addLayout(vbox)
43          mainLayout.addLayout(hbox)
44
45          self.setLayout(mainLayout)
46          self.addButton.clicked.connect(lambda: self.calculate('+'))
47          self.subtractButton.clicked.connect(lambda: self.calculate('-'))
48          self.mulButton.clicked.connect(lambda: self.calculate('*'))
49          self.divButton.clicked.connect(lambda: self.calculate('/'))
50
51      def calculate(self,operator):
52          num1=float(self.num1Edit.text())
53          num2=float(self.num2Edit.text())
54
55          if operator=='+':
56              result=num1+num2
57              operation='penjumlahan'
58          elif operator=='-':
59              result=num1-num2
60              operation='pengurangan'
61          elif operator=='*':
62              result=num1*num2
63              operation='perkalian'
64          else:
65              result=num1/num2
66              operation='pembagian'
67
68          self.label3.setText('Hasil %s' % operation)
69          self.resultEdit.setText(str(result))
70
71  if __name__ == '__main__':
72      a=QApplication(sys.argv)
73      form=MainForm()
74      form.show()
75      a.exec_()
```

Membuat Radio Button

```
# Membuat QRadioButton

import sys
from PyQt5.QtWidgets import *

class MainForm(QWidget):
    def __init__(self):
        super().__init__()
        self.setupUi()

    def setupUi(self):
        self.resize(300,100)
        self.setWindowTitle('Demo QRadioButton')

        self.label=QLabel('Jenis Kemalin:')
        self.radio1=QRadioButton('Laki-laki')
        self.radio2=QRadioButton('Perempuan')
        self.okButton=QPushButton('OK')

        vbox=QVBoxLayout()
        vbox.addWidget(self.label)
        vbox.addWidget(self.radio1)
        vbox.addWidget(self.radio2)
        vbox.addWidget(self.okButton)

        self.setLayout(vbox)
        self.okButton.clicked.connect(self.on_okButton_clicked)

    def on_okButton_clicked(self):
        if self.radio1.isChecked():
            gender=self.radio1.text()
        elif self.radio2.isChecked():
            gender=self.radio2.text()
        else:
            gender=None

        if gender==None:
            QMessageBox.information(self,'Informasi','Anda belum memilih opsi')
        else:
            QMessageBox.information(self,'Informasi','Anda memilih "%s"' % gender)

if __name__ == '__main__':
    a=QApplication(sys.argv)
    form=MainForm()
    form.show()
    a.exec_()
```

Membuat Checkbox

```python
# Membuat QCheckButton

import sys
from PyQt5.QtWidgets import *

class MainForm(QWidget):
    def __init__(self):
        super().__init__()
        self.setupUi()

    def setupUi(self):
        self.resize(250,100)
        self.setWindowTitle('Demo QCheckButton')

        self.label=QLabel('Bahasa pemrograman yang dikuasai:')
        self.checkbox1=QCheckBox('Python')
        self.checkbox2=QCheckBox('PHP')
        self.checkbox3=QCheckBox('Java')
        self.checkbox4=QCheckBox('C++')
        self.okButton=QPushButton('OK')

        vbox=QVBoxLayout()
        vbox.addWidget(self.label)
        vbox.addWidget(self.checkbox1)
        vbox.addWidget(self.checkbox2)
        vbox.addWidget(self.checkbox3)
        vbox.addWidget(self.checkbox4)
        vbox.addWidget(self.okButton)

        self.setLayout(vbox)
        self.okButton.clicked.connect(self.on_okButton_clicked)

    def on_okButton_clicked(self):
        lang=[]
        if self.checkbox1.isChecked(): lang.append('Python')
        if self.checkbox2.isChecked(): lang.append('PHP')
        if self.checkbox3.isChecked(): lang.append('Java')
        if self.checkbox4.isChecked(): lang.append('C++')

        QMessageBox.information(self,'Informasi','Anda memilih "%s"' % str(lang))

if __name__ == '__main__':
    a=QApplication(sys.argv)
    form=MainForm()
    form.show()
    a.exec_()
```

Gunakanlah database yang telah dibuat sesuai servernya.

Menambah isi tabel pada database **mydb** menggunakan SQLite melalui PyQt

```
# Insert data SQLite melalui PyQt

#from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import *
import sys
import sqlite3

class MainForm(QWidget):
    def __init__(self):
        super().__init__()
        self.setupUi()

    def setupUi(self):
        self.resize(250, 100)
        self.setWindowTitle('Insert data produk')
        self.label1=QLabel('kode')
        self.kodeEdit=QLineEdit()
        self.label2=QLabel('nama')
        self.namaEdit=QLineEdit()
        self.label3=QLabel('Harga')
        self.hargaEdit=QLineEdit()
        self.saveButton=QPushButton('Save')
        self.clearButton=QPushButton('Clear')
        self.exitButton=QPushButton('Exit')

        grid=QGridLayout()
        grid.addWidget(self.label1,0,0)
        grid.addWidget(self.kodeEdit,0,1,1,2)
        grid.addWidget(self.label2,1,0)
        grid.addWidget(self.namaEdit,1,1,1,2)
        grid.addWidget(self.label3,2,0)
        grid.addWidget(self.hargaEdit,2,1,1,2)
        grid.addWidget(self.saveButton,3,1)
        grid.addWidget(self.clearButton,3,2)
        grid.addWidget(self.exitButton,3,3)

        self.setLayout(grid)
        self.saveButton.clicked.connect(self.insert_data)
        self.clearButton.clicked.connect(self.on_clearButton_clicked)
        self.exitButton.clicked.connect(self.on_exitButton_clicked)

```

```
    def on_clearButton_clicked(self):
        self.kodeEdit.clear()
        self.namaEdit.clear()
        self.hargaEdit.clear()

    def on_exitButton_clicked(self):
        self.close()

    def insert_data(self):
        try:
            conn = sqlite3.connect("mydb")
            cur = conn.cursor()

            kode = self.kodeEdit.text()
            nama = self.namaEdit.text()
            harga = self.hargaEdit.text()

            data_produk=[kode, nama, harga]
            cur.execute("INSERT INTO produk (kode, nama, harga) VALUES (?, ?, ?)", data_produk)

            conn.commit()
            conn.close()
            print('data berhasil disimpan')
        except conn.Error as e:
            print('data berhasil disimpan')

if __name__ == '__main__':
    a=QApplication(sys.argv)
    form=MainForm()
    form.show()
    a.exec_()
```