Praktikum Pemrograman GUI Pertemuan 6

Dasar Materi (Penggunaaan PyQt)

1. Penggunaan Kelas QSpinBox
2. Penggunaan Kelas QdateEdit, QtimeEdit, dan QDateTimeEdit
3. Penggunaan Kelas QCalendarWidget
4. Penggunaan Kelas Qslider & QLCDNumber
5. Penggunaan Kelas QProgressBar

Upload Ke Github masing-masing, materi yang telah diajarkan oleh Dosen Pengampu Praktikum.

 **https://github.com/afandi354**

---

1. **Penggunaan kelas QSpinBox**

   Langkah-langkah :

   a. Buat sebuah file dengan nama **DemoQSpinBox.py**

   b. Ketikan source code berikut pada file **DemoQSpinBox.py**

```
import sys

from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *

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

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

    self.fontLabel = QLabel('Jenis huruf')
    self.fontCombo = QFontComboBox()
    self.fontCombo.setEditable(False)

    self.sizeLabel = QLabel('Ukuran huruf')
    self.sizeSpinBox = QSpinBox()
    self.sizeSpinBox.setRange(8,20)
    self.sizeSpinBox.setSingleStep(1)
    self.sizeSpinBox.setValue(18)

    self.sampleLabel = QLabel('Contoh Teks')
    self.sampleLabel.setFont(QFont('Cambria',18))

    layout = QGridLayout()
    layout.addWidget(self.fontLabel, 0, 0)
    layout.addWidget(self.fontCombo, 0, 1)
    layout.addWidget(self.sizeLabel, 1, 0)
    layout.addWidget(self.sizeSpinBox, 1, 1)
    layout.addWidget(self.sampleLabel, 2, 0, 1, 2)
    #layout.addStretch()
    self.setLayout(layout)

    self.fontCombo.activated.connect(self.changeFont)
    self.sizeSpinBox.valueChanged.connect(self.changeFont)

  def changeFont(self):
    self.sampleLabel.setFont(
      QFont(self.fontCombo.currentText(), self.sizeSpinBox.value())
    )
```

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

    form = MainForm()
    form.show()

    a.exec_()
```

c. Running di terminal atau CMD dengan perintah : **python DemoQSpinBox.py**

d. Screenshot hasil keluaranya.

e. Push *source code* ke github!

---

2. Penggunaan Kelas QdateEdit, QtimeEdit, dan QDateTimeEdit

Langkah-langkah :

a. Buat sebuah file dengan nama **DemoQDateTimeEdit.py**

b. Ketikan source code berikut pada file **DemoQDateTimeEdit.py**

```
import sys

from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *

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

    def setupUi(self):
        self.resize(400, 100)
        self.move(300, 300)
        self.setWindowTitle('Demo QDateTimeEdit')

        self.dateLabel = QLabel('Tanggal')
        self.dateEdit = QDateEdit()
        self.dateEdit.setDisplayFormat('dddd dd/MM/yyyy')
        self.dateEdit.setDate(QDate.currentDate())

        self.timeLabel = QLabel('Waktu')
        self.timeEdit = QTimeEdit()
        self.timeEdit.setDisplayFormat('hh:mm')
        self.timeEdit.setTime(QTime.currentTime())

        self.dateTimeLabel = QLabel('Tanggal dan Waktu')
        self.dateTimeEdit = QDateTimeEdit()
```

```
        self.dateTimeEdit.setDisplayFormat('dddd dd/MM/yyyy hh:mm')
        self.dateTimeEdit.setDateTime(QDateTime.currentDateTime())

        self.okButton = QPushButton('&OK')
        hbox = QHBoxLayout()
        hbox.addStretch()
        hbox.addWidget(self.okButton)

        layout = QGridLayout()
        layout.addWidget(self.dateLabel, 0, 0)
        layout.addWidget(self.dateEdit, 0, 1)
        layout.addWidget(self.timeLabel, 1, 0)
        layout.addWidget(self.timeEdit, 1, 1)
        layout.addWidget(self.dateTimeLabel, 2, 0)
        layout.addWidget(self.dateTimeEdit, 2, 1)
        layout.addLayout(hbox, 3, 0, 1, 2)
        self.setLayout(layout)

        self.okButton.clicked.connect(self.okButtonClick)

    def okButtonClick(self):
        QMessageBox.information(self, 'Informasi Hari Ini',
            'Tanggal : ' + self.dateEdit.date().toString() + '\n' +
            'Waktu : ' + self.timeEdit.time().toString() + '\n' +
            'Datetime : ' + self.dateTimeEdit.dateTime().toString() + '\n')

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

    form = MainForm()
    form.show()

    a.exec_()
```

c. Running di terminal atau CMD dengan perintah : **python DemoQDateTimeEdit.py**

d. Screenshot hasil keluaranya

e. Push *source code* ke github!

---

3. Penggunaan Kelas QCalendarWidget

Langkah-langkah :

a. Buat sebuah file dengan nama **DemoQCalendarWidget.py**

b. Ketikan source code berikut pada file **DemoQCalendarWidget.py**

```
import sys

from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *

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

  def setupUi(self):
    self.resize(400, 100)
    self.move(300, 300)
    self.setWindowTitle('Demo QCalendarWidget')

    self.calendar = QCalendarWidget()
    self.calendar.setGridVisible(True)
    self.calendar.setHorizontalHeaderFormat(QCalendarWidget.LongDayNames)

    self.shortNamesCheck = QCheckBox('Nama hari pendek')

    self.dateEdit = QDateEdit()
    self.dateEdit.setDisplayFormat('dd/MM/yyyy')
    self.dateEdit.setDate(QDate.currentDate())

    self.setButton = QPushButton('Tentukan Tanggal')
    self.getButton = QPushButton('Ambil Tanggal')

    hbox = QHBoxLayout()
    hbox.addWidget(self.dateEdit)
    hbox.addWidget(self.setButton)
    hbox.addWidget(self.getButton)

    layout = QVBoxLayout()
    layout.addWidget(self.calendar)
    layout.addWidget(self.shortNamesCheck)
    layout.addLayout(hbox)
    self.setLayout(layout)

    self.shortNamesCheck.clicked.connect(self.shortNamesCheckClick)
    self.setButton.clicked.connect(self.setButtonClick)
    self.getButton.clicked.connect(self.getButtonClick)

  def shortNamesCheckClick(self):
    if self.shortNamesCheck.isChecked():
      self.calendar.setHorizontalHeaderFormat(QCalendarWidget.ShortDayNames)
    else:
      self.calendar.setHorizontalHeaderFormat(QCalendarWidget.LongDayNames)
```

```
    def setButtonClick(self):
        self.calendar.setSelectedDate(self.dateEdit.date())

    def getButtonClick(self):
        QMessageBox.information(self, 'Informasi',
            'Tanggal aktif: ' + self.calendar.selectedDate().toString())

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

    form = MainForm()
    form.show()

    a.exec_()
```

c. Running di terminal atau CMD dengan perintah : **python DemoQCalendarWidget.py**

d. Screenshot hasil keluaranya

e. Push source code ke github!

---

4. Penggunaan Kelas Qslider & QLCDNumber

Langkah-langkah :

a. Buat sebuah file dengan nama **DemoQSlider.py**

b. Ketikan source code berikut pada file **DemoQSlider.py**

```
import sys

from PyQt5.QtGui import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *

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

    def setupUi(self):
        self.resize(400, 100)
        self.move(300, 300)
        self.setWindowTitle('Demo QSlider dan QLCDNumber')

        self.slider = QSlider(Qt.Horizontal)
        self.slider.setMinimum(-1)
        self.slider.setMaximum(101)
        self.slider.setValue(45)

        self.lcd = QLCDNumber()
        self.lcd.setDigitCount(3)
```

```
        self.lcd.display(45)

        layout = QVBoxLayout()
        layout.addWidget(self.slider)
        layout.addWidget(self.lcd)
        self.setLayout(layout)

        self.slider.sliderMoved.connect(self.sliderMoved)

    def sliderMoved(self):
        self.lcd.display(str(self.slider.value()))

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

    form = MainForm()
    form.show()

    a.exec_()
```

c. Running di terminal atau CMD dengan perintah : **python DemoQSlider.py**

d. Screenshot hasil keluaranya

e. Push *source code* ke github!

---

5. Penggunaan Kelas QProgressBar

Langkah-langkah :

a. Buat sebuah file dengan nama **DemoQProgressBar.py**

b. Ketikan source code berikut pada file **DemoQProgressBar.py**

```
import sys

from PyQt5.QtCore import *
from PyQt5.QtWidgets import *

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


    def initUI(self):

        self.setGeometry(300, 300, 280, 150)
        self.setWindowTitle('Demo QProgressBar')
        self.show()
```

```
        self.pbar = QProgressBar(self)
        self.pbar.setGeometry(30, 40, 200, 25)

        self.btn = QPushButton('Start', self)
        self.btn.move(40, 80)
        self.btn.clicked.connect(self.doAction)

        self.timer = QBasicTimer()
        self.step = 0

    def timerEvent(self, e):
        if self.step >= 100:
            self.timer.stop()
            self.btn.setText('Selesai')
            return

        self.step = self.step + 1
        self.pbar.setValue(self.step)

    def doAction(self, e):
        if self.timer.isActive():
            self.timer.stop()
            self.btn.setText('Start')
        else:
            self.timer.start(100, self)
            self.btn.setText('Stop')


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = MainForm()
    sys.exit(app.exec_())
```

c. Running di terminal atau CMD dengan perintah : **python DemoQProgressBar.py**

d. Screenshot hasil keluaranya

e. Push *source code* ke github!
