

Modul 5 Praktikum PemDesk

Ganjil 2018

Database 2

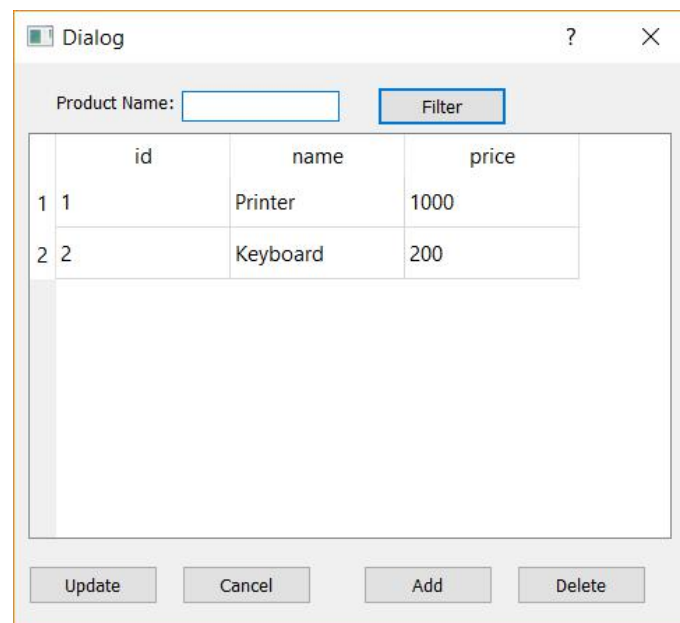
Materi

- Editing dalam table view
- Navigasi per record

Editing Dalam Table View

Tampilan isi tabel dalam table view dapat juga ditambahkan beberapa fasilitas untuk memanajemen record, seperti: Add, Update, dan Delete.

Contoh program yang akan dibuat adalah aplikasi pengedit record pada table view. Tampilan program dapat dilihat pada gambar 1.



Gambar 1. Tampilan Program Edit Table View

Langkah-langkah yang perlu dilakukan adalah:

1. Buat antar muka seperti pada gambar 1 menggunakan Qt Designer.:
 - a. Drag-and-drop widget label, line edit, dan push button pada bagian atas form.
 - b. Tambahkan widget table view di tengah form.
 - c. Tambahkan 4 buah push button pada bagian bawah form.

d. Edit tulisan yang ditampilkan masing-masing widget sesuai dengan gambar 1.

Simpan hasilnya dengan nama `wk8aEditTableView.ui` dan lakukan proses konversi menggunakan `pyuic5` menjadi `wk8aEditTableView.py`.

2. Lanjutkan dengan membuat aplikasi utama seperti yang dapat dilihat pada program 1.
 1. Beri program utama nama: `wk8aEditTableViewMain.py`

```
import sys
from wk8aEditTableView import *
from PyQt5 import QSql

class MyForm(QDialog):
    def __init__(self, parent=None):
        QtWidgets.QWidget.__init__(self, parent)
        self.ui = Ui_Dialog()
        self.ui.setupUi(self)
        self.db = QSql.QSqlDatabase.addDatabase('QSQLITE')
        self.db.setDatabaseName('test.sqlite')
        self.db.open()
        self.model = QSql.QSqlTableModel(self)
        self.model.setTable("product")
        self.model.setEditStrategy(QSql.QSqlTableModel.OnManualSubmit)
        self.model.select()
        self.ui.tableView.setModel(self.model)
        self.ui.pushButton.clicked.connect(self.FilterRecord)
        self.ui.pushButton_2.clicked.connect(self.UpdateRecord)
        self.ui.pushButton_3.clicked.connect(self.CancelAll)
        self.ui.pushButton_4.clicked.connect(self.InsertRecord)
        self.ui.pushButton_5.clicked.connect(self.DeleteRecord)

    def UpdateRecord(self):
        self.model.submitAll()
    def CancelAll(self):
        self.model.revertAll()
    def InsertRecord(self):
        self.model.insertRow(self.ui.tableView.currentIndex().row())
    def DeleteRecord(self):
        self.model.removeRow(self.ui.tableView.currentIndex().row())
        self.model.submitAll()
    def FilterRecord(self):
        self.model.setFilter("name like '"+self.ui.lineEdit.text()+"%'"")

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    myapp = MyForm()
    myapp.show()
    sys.exit(app.exec_())
```

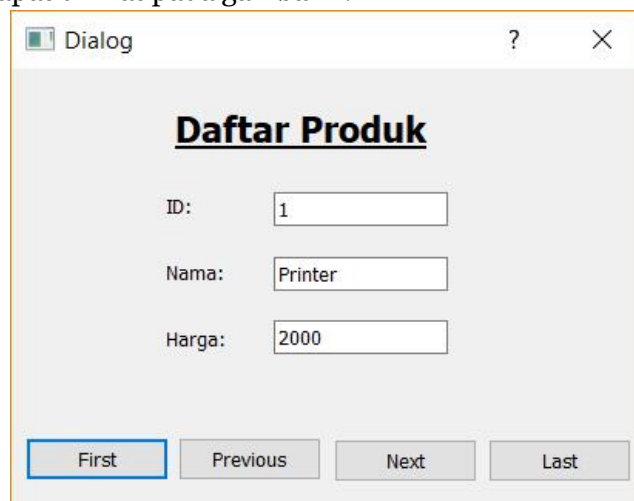
Program 1. Program Edit pada Table View

Program 1 membuat 5 custom slot: 1 slot dikoneksikan dengan 1 tombol. Pengguna dapat melakukan filtering record berdasar kolom nama dengan cara mengetikkan tulisan pada line edit dan kemudian mengklik tombol Filter. Pengguna dapat mengedit informasi pada record dengan mengklik ganda pada cell, kemudian mengetikkan nilai yang baru. Untuk menyimpan secara permanen perubahan tersebut, tombol Update perlu diklik. Untuk menambah record baru, pengguna mengklik pada satu row, kemudian klik tombol Add. Akan muncul baris baru pada table view. Pengguna dapat memasukkan data baru di baris tersebut, lalu klik tombol Update untuk menyimpannya. Ketika tombol Delete diklik, baris aktif akan dihapus dari tabel.

Navigasi per Record

Penampilan data menggunakan table view memang memudahkan pengguna untuk melihat keseluruhan data dengan cepat, tapi kita juga bisa membuat program supaya hanya menampilkan hanya satu record pada form.

Contoh program yang akan dibuat adalah aplikasi navigasi record sederhana. Tampilan program dapat dilihat pada gambar 2.



The image shows a Windows-style dialog box titled "Dialog" with a question mark icon and a close button. The main title inside the dialog is "Daftar Produk" in bold and underlined. Below the title, there are three input fields: "ID:" with the value "1", "Nama:" with the value "Printer", and "Harga:" with the value "2000". At the bottom of the dialog, there are four buttons: "First" (highlighted with a blue border), "Previous", "Next", and "Last".

Gambar 2. Tampilan hasil running program navigasi per record
Langkah-langkah yang perlu dilakukan adalah:

1. Buat antar muka seperti pada gambar 2 menggunakan Qt Designer.:
 - a. Drag-and-drop widget label untuk dijadikan judul Form
 - b. Tambahkan 3 buah label dan 3 buah line edit untuk menampilkan field-field pada record
 - c. Tambahkan 4 buah push button pada bagian bawah form.
 - d. Edit tulisan yang ditampilkan masing-masing widget sesuai dengan gambar 2.

Simpan hasilnya dengan nama wk8bNavigasiRecord.ui dan lakukan proses konversi menggunakan pyuic5 menjadi wk8bNavigasiRecord.py.

2. Lanjutkan dengan membuat aplikasi utama seperti yang dapat dilihat pada program 2. Beri program utama nama: wk8bNavigasiRecordMain.py

```
import sqlite3
from wk8bNavigasiRecord import *
from PyQt5 import QtSql

class MyForm(QtWidgets.QDialog):
    recno = 0
    def __init__(self, parent=None):
        QtWidgets.QWidget.__init__(self, parent)
        self.ui = Ui_Dialog()
        self.ui.setupUi(self)

        self.db = QtSql.QSqlDatabase.addDatabase('QSQLITE')
        self.db.setDatabaseName('test.sqlite')
        self.db.open()
        self.model = QtSql.QSqlQueryModel(self)
        self.model.setQuery("select * from product")
        self.record = self.model.record(0)
        self.ui.lineEdit.setText(str(self.record.value("id")))
        self.ui.lineEdit_2.setText(str(self.record.value("name")))
        self.ui.lineEdit_3.setText(str(self.record.value("price")))
        self.ui.pushButton.clicked.connect(self.dispFirst)
        self.ui.pushButton_2.clicked.connect(self.dispPrevious)
        self.ui.pushButton_3.clicked.connect(self.dispNext)
        self.ui.pushButton_4.clicked.connect(self.dispLast)
    def displayRec(self):
        self.record=self.model.record(MyForm.recno)
        self.ui.lineEdit.setText(str(self.record.value("id")))
        self.ui.lineEdit_2.setText(str(self.record.value("name")))
        self.ui.lineEdit_3.setText(str(self.record.value("price")))
    def dispFirst(self):
        MyForm.recno = 0
        self.displayRec()
    def dispPrevious(self):
        MyForm.recno -= 1
        if MyForm.recno < 0:
            MyForm.recno = 0
        self.displayRec()
    def dispLast(self):
        MyForm.recno = self.model.rowCount()-1
        self.displayRec()
```

```
def dispNext(self):
    MyForm.recno += 1
    if MyForm.recno > self.model.rowCount()-1:
        MyForm.recno = self.model.rowCount()-1
    self.displayRec()

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    myapp = MyForm()
    myapp.show()
    sys.exit(app.exec_())
```

Program 2. Program database sqlite sederhana

Berbeda dengan program-program sebelumnya yang menggunakan QSqlTableModel, program 2 membuat model dengan QSqlQueryModel yang memungkinkan kita untuk mengeksekusi perintah SELECT SQL. Dengan model ini, kita bisa mengakses satu-persatu field hasil dari perintah SELECT dan menset nilainya pada 3 line edit yang tersedia.

Program 2 membuat sebuah variable recno sebagai variable pencatat pada posisi record berapa yang sedang ditampilkan program. Jika pengguna mengklik tombol First, recno diset dengan nilai 0. Jika pengguna mengklik tombol Last, recno diset dengan nilai jumlah record dikurangi 1. Dan jika pengguna mengklik tombol Next atau Previous, recno akan ditambah atau dikurangi dengan nilai 1; selanjutnya nilai recno dicek supaya tidak kurang dari 0 dan tidak lebih dari jumlah maksimum record.

Tugas

1. Modifikasi program 1: Buat 2 filter baru (berarti tambahkan 2 label baru, 2 line edit baru, dan 2 push button baru). Yang pertama untuk id, yang kedua untuk harga.
2. Melanjutkan tugas dari praktikum sebelumnya, tampilkan tabel product, customer, dan pembelian dalam satu form menggunakan 3 buah table view. Pada masing-masing table view, buat fasilitas Update, Cancel, Add, dan Delete.