

Nama : Juan Axl Ronaldo Zaka Putra

NIM : 220411100066

Kelas : Pemrograman Desktop C

### **P13: Read Delete**

## **CODING DAN PENJELASAN**

### **1. Import sys dan PyQt6**

```
import sys

from PyQt6.QtSql import QSqlDatabase, QSqlQuery, QSqlQueryModel
from PyQt6.QtWidgets import (
    QApplication,
    QWidget,
    QLabel,
    QLineEdit,
    QPushButton,
    QDialog,
    QTableView,
    QFormLayout,
    QHBoxLayout,
    QVBoxLayout,
    QGridLayout,
)
```

Import sys, PyQt6.QtSql agar dapat terhubung dan memanipulasi database. Dan PyQt6.QtWidgets agar dapat menggunakan widget-widget yang dibutuhkan.

### **2. Fungsi db\_open()**

```
def db_open():
    global db
    db = QSqlDatabase.addDatabase("QSQLITE")
    db.setDatabaseName("mahasiswadb.db")
    if not db.open():
        print("db connection failed")
    else:
        print("db connection success")
```

Fungsi db\_open() digunakan untuk koneksi ke database. Inisiasi database pada variable db dengan QSqlDatabase dan menambahkan database dengan QSQLITE karena database

yang akan digunakan adalah SQLite, lalu set nama database dengan file database yang telah dibuat sebelumnya, dan buka database dengan `db.open()`.

### 3. Fungsi `set_window()`

```
def set_window():
    global layoutContainer
    layoutContainer = QVBoxLayout()

    btnInsert = QPushButton("insert")
    btnInsert.setStyleSheet("background-color: green")
    btnInsert.clicked.connect(lambda:insert())
    layoutContainer.addWidget(btnInsert)

    btnDelete = QPushButton("delete")
    btnDelete.setStyleSheet("background-color: red")
    btnDelete.clicked.connect(lambda:delete(tableView))
    layoutContainer.addWidget(btnDelete)

    layoutContainer.addLayout(form)
    window.setLayout(layoutContainer)

    window.setGeometry(100, 100, 350, 400)
    window.setWindowTitle("P13")
    window.show()
    sys.exit(app.exec())
```

Fungsi `set_window()` digunakan untuk mengeksekusi aplikasi dan menampilkan window. Didalam fungsi ini terdapat inisiasi `layoutContainer` dengan `QVBoxLayout()` untuk wadah layout utama atau terluar. Inisiasi dua button untuk insert dan delete yang di connect ke fungsi `insert()` dan `delete()` lalu kedua button tersebut ditempatkan ke dalam `layoutContainer`. Menambahkan layout form ke `layoutContainer` dan set layout pada window ke `layoutContainer`. Berikutnya, mengatur ukuran window, judul window, menampilkan window, dan eksekusi aplikasi.

#### 4. Fungsi db\_query()

```
def db_query():  
    global query  
    query = QSqlQuery(db)  
    query.prepare("select * from mahasiswa")  
    query.exec()  
  
    global model  
    model = QSqlQueryModel()  
    model.setQuery(query)  
    model.layoutChanged.emit()  
  
    global tableView  
    tableView = QTableView()  
    tableView.reset()  
    tableView.setModel(model)  
    tableView.resizeColumnsToContents()  
  
    global form  
    form = QFormLayout()  
    form.addWidget(tableView)
```

Fungsi db\_query() digunakan untuk melakukan query ke database untuk menampilkan isi tabel mahasiswa pada database. Inisiasi variable query dengan QSqlQuery(db) untuk melakukan query ke database, prepare SQL untuk menampilkan data dalam tabel, lalu eksekusi query. Inisiasi model dengan QSqlQueryModel untuk menampilkan data yang telah di query. Inisiasi tableView dengan QTableView() untuk menampilkan data yang telah di query dalam bentuk tabel dan setModel pada model yang sebelumnya sudah dibuat. Inisiasi form dengan QFormLayout() untuk menggunakan layout form, lalu form diberi widget tableView.

## 5. Fungsi insert():

```
def insert():
    nama, ok = QInputDialog.getText(window, 'Insert', 'Input nama:')
    nim, ok = QInputDialog.getText(window, 'Insert', 'Input NIM:')
    alamat, ok = QInputDialog.getText(window, 'Insert', 'Input
alamat:')

    query.prepare("insert into mahasiswa (nama, nim, alamat) values
(:nama, :nim, :alamat)")
    query.bindValue(":nama", nama)
    query.bindValue(":nim", nim)
    query.bindValue(":alamat", alamat)
    query.exec()

    query.prepare("select * from mahasiswa")
    query.exec()

    model.setQuery(query)

    tableView.setModel(model)

    form.addWidget(tableView)
    layoutContainer.removeItem(layoutContainer.itemAt(2))
    layoutContainer.addLayout(form)
```

Fungsi insert() untuk menambahkan data pada tabel mahasiswa dalam database. Didalam fungsi ini menampilkan input dialog untuk menginputkan data yang akan di insert. Setelah data diinputkan maka akan dilakukan query insert dengan prepare dan bindValue pada value yang akan diinsert. Barikutnya tabel ditampilkan ulang agar data yang baru tampil pada tabel.

## 6. Fungsi delete()

```
def delete(tableView):  
    id = ""  
    rows = sorted(set(index.row() for index in  
tableView.selectedIndexes()))  
    for row in rows:  
        id = tableView.model().data(tableView.model().index(row, 0))  
  
    query.prepare("delete from mahasiswa where idMhs = :id")  
    query.bindValue(":id", id)  
    query.exec()  
  
    query.prepare("select * from mahasiswa")  
    query.exec()  
  
    model.setQuery(query)  
  
    tableView.setModel(model)  
  
    form.addWidget(tableView)  
    layoutContainer.removeItem(layoutContainer.itemAt(2))  
    layoutContainer.addLayout(form)
```

Fungsi delete() untuk menghapus data pada tabel mahasiswa dalam database. Didalam fungsi ini mengambil semua nilai idMhs dari setiap baris data. Saat baris data pada tabel di click lalu click button delete maka data yang di click tersebut akan terhapus. Data dalam tabel tersebut bisa dihapus karena pada fungsi ini dilakukan query untuk menghapus data berdasarkan id setelah id didapatkan dengan cara click baris data pada tabel dan query dijalankan setelah button delete di click. Berikutnya tabel ditampilkan ulang agar data yang sudah dihapus hilang pada tabel.

## 7. Eksekusi

```
app = QApplication(sys.argv)  
window = QWidget()  
db_open()  
db_query()  
set_window()
```

Inisiasi variable app dengan QApplication(sys.argv) dan window dengan QWidget(). Menjalankan fungsi db\_open() untuk koneksi ke database, db\_query() untuk menampilkan data, dan menjalankan fungsi set\_window() untuk menjalankan aplikasi atau menampilkan window.

## HASIL RUNNING



