Nama : Juan Axl Ronaldio Zaka Putra

NIM : 220411100066

Kelas : Pemrograman Desktop C

**P12: Read Table**

**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,

QInputDialog,

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.

1. Inisiasi app dan window

app = QApplication(sys.argv)

window = QWidget()

Inisiasi variable app dengan QApplication(sys.argv) dan window dengan QWidget().

1. Fungsi run\_window()

def run\_window():

global layoutContainer

layoutContainer = QVBoxLayout()

window.setLayout(layoutContainer)

global form

form = QFormLayout()

form.addWidget(tableView)

layoutContainer.addLayout(form)

btnInsert = QPushButton("Insert")

btnInsert.clicked.connect(insert)

btnInsert.setStyleSheet("background-color: #198754; color: white;")

btnDelete = QPushButton("Delete")

btnDelete.clicked.connect(delete)

btnDelete.setStyleSheet("background-color: #dc3545; color: white")

btnLayout = QHBoxLayout()

btnLayout.addWidget(btnInsert)

btnLayout.addWidget(btnDelete)

layoutContainer.addLayout(btnLayout)

window.setGeometry(100, 100, 350, 400)

window.setWindowTitle("P12: Read Table")

window.show()

sys.exit(app.exec())

Fungsi run\_window() digunakan untuk mengeksekusi aplikasi dan menampilkan window. Didalam fungsi ini terdapat inisiasi layoutContainer dengan QVBoxLayout() untuk wadah layout utama atau terluar. Inisiasi form dengan QFormLayout() untuk menggunakan layout form, lalu form dimasukkan ke layoutContainer. Inisiasi button insert dan delete dan dikoneksikan ke fungsi masing-masing untuk memanipulasi database. Inisiasi btnLayout dengan QHBoxLayout untuk layout button insert dan delete, lalu btnLayout dimasukkan ke layoutContainer. Berikutnya, mengatur ukuran window, judul window, menampilkan window, dan eksekusi aplikasi.

1. Fungsi open\_db()

def open\_db():

global db

db = QSqlDatabase.addDatabase("QSQLITE")

db.setDatabaseName("mahasiswadb.db")

db.open()

if not db.open():

print("db connection error")

else:

print("db connection success")

global query

query = QSqlQuery(db)

query.prepare("SELECT nama, nim, alamat FROM mahasiswa")

query.exec()

global model

model = QSqlQueryModel()

model.setQuery(query)

global tableView

tableView = QTableView()

tableView.setModel(model)

tableView.resizeColumnsToContents()

Fungsi open\_db() digunakan untuk koneksi ke database dan menampilkan isi tabel pada 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().

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.

1. Fungsi insert():

def insert():

nama, ok = QInputDialog.getText(window, 'Insert Data', 'nama: ')

nim, ok = QInputDialog.getText(window, 'Insert Data', 'nim: ')

alamat, ok = QInputDialog.getText(window, 'Insert Data', 'alamat: ')

if nama != '' and nim != '' and 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 nama, nim, alamat FROM mahasiswa")

query.exec()

model.setQuery(query)

tableView.setModel(model)

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.

1. Fungsi delete()

def delete():

nama, ok = QInputDialog.getText(window, 'Delete Data', 'nama: ')

if nama != '':

query.prepare("DELETE FROM mahasiswa WHERE nama = :nama")

query.bindValue(':nama', nama)

query.exec()

query.prepare("SELECT nama, nim, alamat FROM mahasiswa")

query.exec()

model.setQuery(query)

tableView.setModel(model)

Fungsi delete() untuk menghapus data pada tabel mahasiswa dalam database. Didalam fungsi ini menampilkan input dialog untuk menginputkan data yang akan di delete. Setelah data diinputkan maka akan dilakukan query delete dengan prepare dan bindValue pada condition yang akan di delete. Barikutnya tabel ditampilkan ulang agar data yang sudah dihapus hilang pada tabel.

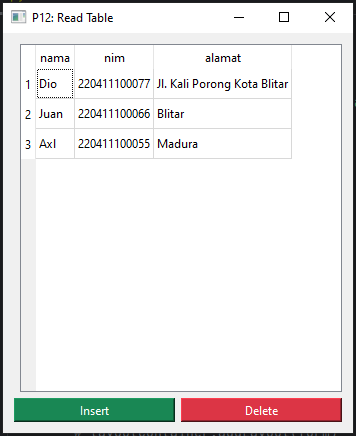
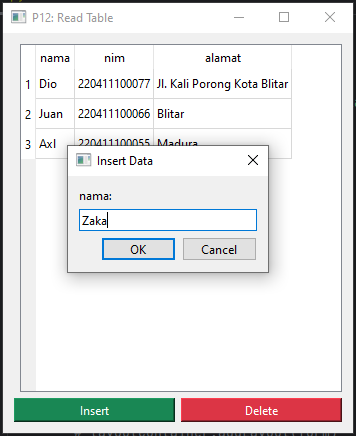
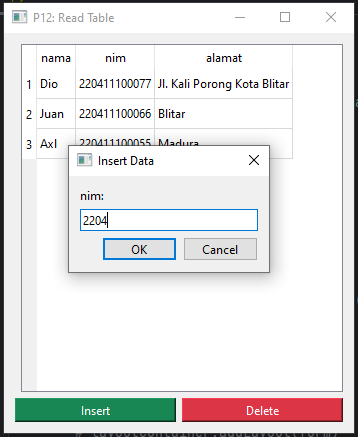
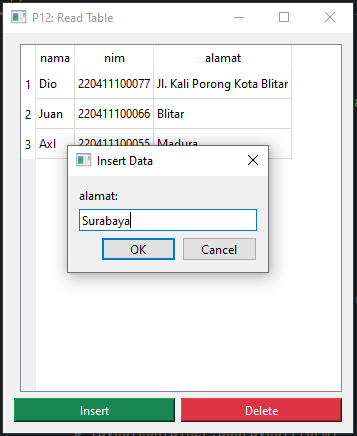
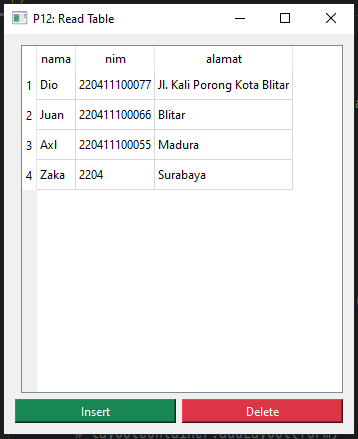
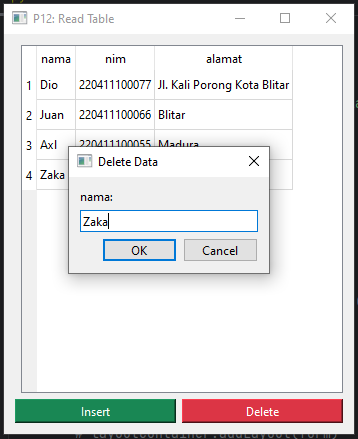
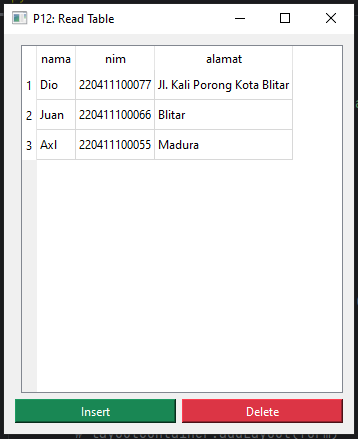
1. Memanggil fungsi

open\_db()

run\_window()

Menjalankan fungsi open\_db() untuk koneksi ke database dan menjalankan fungsi run\_window() untuk menjalankan aplikasi atau menampilkan window.

**HASIL RUNNING**

**      **