**TUGAS DELETE AND UPDATE RECORD DATA SQLITE MATAKULIAH PEMROGRAMAN DEKSTOP**

Disusun untuk Memenuhi Tugas Matakuliah Pemrograman Dekstop yang Dibimbing oleh Moh. Kautsar Sophan, S.Kom., M.MT.

****

Disusun oleh:

Achmad Farid Alfa Waid (190411100073)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2020**

1. **Listing Program**

import sys

from PyQt5 import \*

from PyQt5.QtWidgets import \*

from PyQt5.Qt import \*

from PyQt5.QtCore import \*

from PyQt5.QtGui import \*

from PyQt5 import QtSql

import sqlite3

class Mahasiswa(QWidget):

#Membuat fungsi init untuk inisialisasi class Mahasiswa

def \_\_init\_\_(self):

#untuk mengembalikan semua atribut dan method yang ada

super().\_\_init\_\_()

#membuka database

self.OpenDatabase()

#memanggil fungsi Layout yang sudah dibuat agar ditampilkan hasilnya

self.Layout()

def OpenDatabase(self):

#Mendeklarasikan database

db = QtSql.QSqlDatabase.addDatabase('QSQLITE')

#Membuat nama database

db.setDatabaseName('test.db')

#Mengecek Database Apakah sudah terkoneksi atau belum

if db.open():

print('Berhasil membuka Database')

else:

print('Gagal membuka Database!')

def Layout(self):

#Membuat Grid Layout

grid = QGridLayout()

#Membuat label, Line Edit, dan Button yang akan dimasukkan ke dalam layout Grid

open\_database = QLabel("Buka Database")

grid.addWidget(open\_database,0,0)

open\_button = QPushButton("Open Database")

grid.addWidget(open\_button,0,1,1,2)

add\_data = QLabel("Tambah Data or Update Data:")

grid.addWidget(add\_data,1,0,1,0)

add\_nim = QLabel("Nim:")

grid.addWidget(add\_nim,2,0)

self.Nim = QLineEdit(self)

grid.addWidget(self.Nim,2,1,1,2)

add\_nama = QLabel("Nama:")

grid.addWidget(add\_nama,3,0)

self.Nama = QLineEdit(self)

grid.addWidget(self.Nama,3,1,1,2)

add\_jurusan = QLabel("Jurusan:")

grid.addWidget(add\_jurusan,4,0)

self.Jurusan = QLineEdit(self)

grid.addWidget(self.Jurusan,4,1,1,2)

add\_angkatan = QLabel("Angkatan:")

grid.addWidget(add\_angkatan,5,0)

self.Angkatan = QLineEdit(self)

grid.addWidget(self.Angkatan,5,1,1,2)

add\_button = QPushButton("Tambah Data")

grid.addWidget(add\_button,6,0,1,0)

update\_button = QPushButton("Update Data")

grid.addWidget(update\_button,7,0,1,0)

search = QLabel("Cari Data:")

grid.addWidget(search,8,0)

self.Cari = QLineEdit(self)

grid.addWidget(self.Cari,8,1)

search\_button = QPushButton("Cari")

grid.addWidget(search\_button,8,2)

delete\_button = QPushButton("Hapus Data")

grid.addWidget(delete\_button,9,0,1,0)

#Membuat widget table view yang diberi nama "Data" dan akan dimasukkan ke dalam layout Grid

self.tableview = QTableView(self)

self.tableview.setObjectName("Data")

grid.addWidget(self.tableview,10,0,1,0)

#Ketika button di klik akan memanggi fungsi masing - masing

open\_button.clicked.connect(self.tampilData)

add\_button.clicked.connect(self.tambahData)

search\_button.clicked.connect(self.filterData)

delete\_button.clicked.connect(self.hapusData)

update\_button.clicked.connect(self.updateData)

#Layout grid di jadikan layout utama

self.setLayout(grid)

#Fungsi untuk menampilankan data di dalam table

def tampilData(self):

#Membuat Model

model = QSqlQueryModel()

#Mendefinisikan sql

sql = "SELECT \* FROM mahasiswa"

#Mengeksekusi Model Query

model.setQuery(sql)

#Mengeset Data Model Ke table view

self.tableview.setModel(model)

#self.tableview.setWindowTitle(title)

return self.tableview

#Fungsi tambah Data ke dalam tabel mahasiswa

def tambahData(self):

#Mengambil Text inputan

nim = str(self.Nim.text())

nama = str(self.Nama.text())

jurusan = str(self.Jurusan.text())

angkatan = str(self.Angkatan.text())

#Mendefinisikan Query

query = QSqlQuery()

#Menjalankan Perintah Sql

query.prepare("INSERT INTO mahasiswa VALUES ('" + nim + "', '" + nama + "', '" + jurusan + "', '" + angkatan + "')")

#Mengecek apakah query berjalan dengan baik

if query.exec\_():

self.Nim.setText("")

self.Nama.setText("")

self.Jurusan.setText("")

self.Angkatan.setText("")

#Menampilkan Data

self.tampilData()

else:

#Apabila error akan menampilkan errornya ke dalam terminal

print("Insert Error: ", query.lastError().text())

#Fungsi Filter Data

def filterData(self):

#Membuat Model

model = QSqlQueryModel()

#Mengambil Inputan filter

filter\_search = str(self.Cari.text())

sql = "SELECT \* FROM mahasiswa WHERE nim LIKE '%"+str(filter\_search)+"%' OR nama LIKE '%"+str(filter\_search)+"%' OR jurusan LIKE '%"+str(filter\_search)+"%' OR angkatan LIKE '%"+str(filter\_search)+"%'"

self.Cari.setText("")

#Mengeksekusi Model Query

model.setQuery(sql)

#Mengeset Data Model Ke table view

self.tableview.setModel(model)

#mengembalikan nilai table view

return self.tableview

def hapusData(self):

# Mengambil Data Yang Di klik

select = self.tableview.selectedIndexes()[0]

nim = str(self.tableview.model().data(select))

nama = str(self.tableview.model().data(select))

jurusan = str(self.tableview.model().data(select))

angkatan = str(self.tableview.model().data(select))

# Mendefinisikan Query

query = QSqlQuery()

# Menjalankan Perintah Query

query.prepare("DELETE from mahasiswa WHERE (nim = '" + nim + "'OR nama = '" + nama + "'OR jurusan = '" + jurusan + "'OR angkatan = '" + angkatan + "')")

query.exec\_()

if query.exec\_():

print("Hapus Data Success")

self.tampilData()

else:

print("del\_records Error: ", query.lastError().text())

def updateData(self):

# Mengambil Data Yang Di klik

nimln = str(self.Nim.text())

namaln = str(self.Nama.text())

jurusanln = str(self.Jurusan.text())

angkatanln = str(self.Angkatan.text())

#Mendefinisikan Query

query = QtSql.QSqlQuery()

#Menjalankan Perintah Sql

query.prepare("""UPDATE mahasiswa SET nama=:namaln, jurusan=:jurusanln, angkatan=:angkatanln WHERE nim =:nimln""")

query.bindValue(":nimln", nimln)

query.bindValue(":namaln", namaln)

query.bindValue(":jurusanln", jurusanln)

query.bindValue(":angkatanln", angkatanln)

if query.exec\_():

self.Nim.setText("")

self.Nama.setText("")

self.Jurusan.setText("")

self.Angkatan.setText("")

#Menampilkan Data

self.tampilData()

else:

#Apabila error akan menampilkan errornya ke dalam terminal

print("Insert Error: ", query.lastError().text())

if \_\_name\_\_ == '\_\_main\_\_':

#Inisisalisai pyqt

app = QApplication(sys.argv)

#mengatur style di window menjadi style fusion

app.setStyle("fusion")

#membuat variabel ex yang berisi class FormulaMath

ex = Mahasiswa()

#Menentukan ukuran window dan title untuk menampilkan

ex.setGeometry(100,100,700,600)

#membuat judul window

ex.setWindowTitle("Database Sqlite in Pyqt5")

#menampilan isi dari variabel ex

ex.show()

#membuat system exit

sys.exit(app.exec\_())

1. **SOAL!**
2. Bagaimana cara menghapus record dari sebuat tabel ? beri contoh, buat program, jelaskan, berikan capture hasil running nya
3. Bagaimana cara merubah record tertentu dari sebuah tabel ? beri contoh, buat program, jelaskan, berikan capture hasil running nya
4. **JAWAB!**
5. Pertama membuat tombol button hapus terlebih dahulu, dan kemudian menghubungkan button tersebut ke fungsi hapusData

delete\_button = QPushButton("Hapus Data")

grid.addWidget(delete\_button,9,0,1,0)

delete\_button.clicked.connect(self.hapusData)

Kemudian membuat fungsi hapusData yang akan mengeksekusi perintah hapus data record di dalam database.

def hapusData(self):

# Mengambil Data Yang Di klik

select = self.tableview.selectedIndexes()[0]

nim = str(self.tableview.model().data(select))

nama = str(self.tableview.model().data(select))

jurusan = str(self.tableview.model().data(select))

angkatan = str(self.tableview.model().data(select))

# Mendefinisikan Query

query = QSqlQuery()

# Menjalankan Perintah Query

query.prepare("DELETE from mahasiswa WHERE (nim = '" + nim + "'OR nama = '" + nama + "'OR jurusan = '" + jurusan + "'OR angkatan = '" + angkatan + "')")

query.exec\_()

if query.exec\_():

print("Hapus Data Success")

self.tampilData()

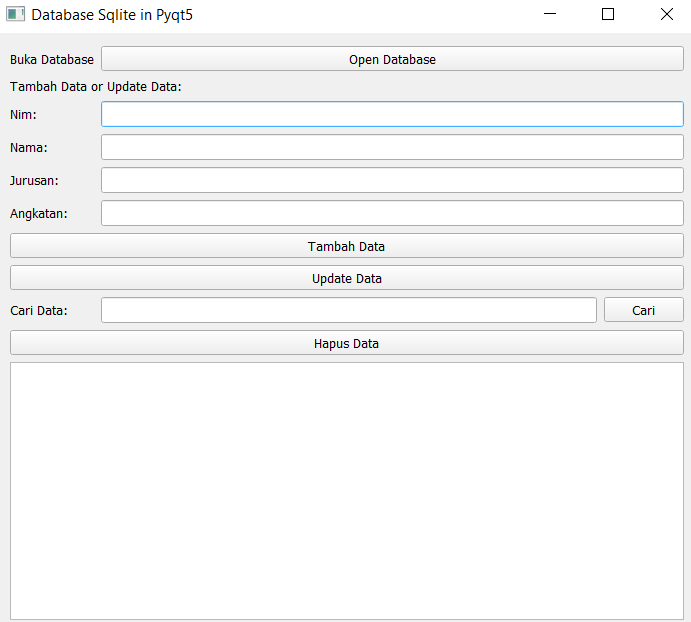
else:

print("del\_records Error: ", query.lastError().text())

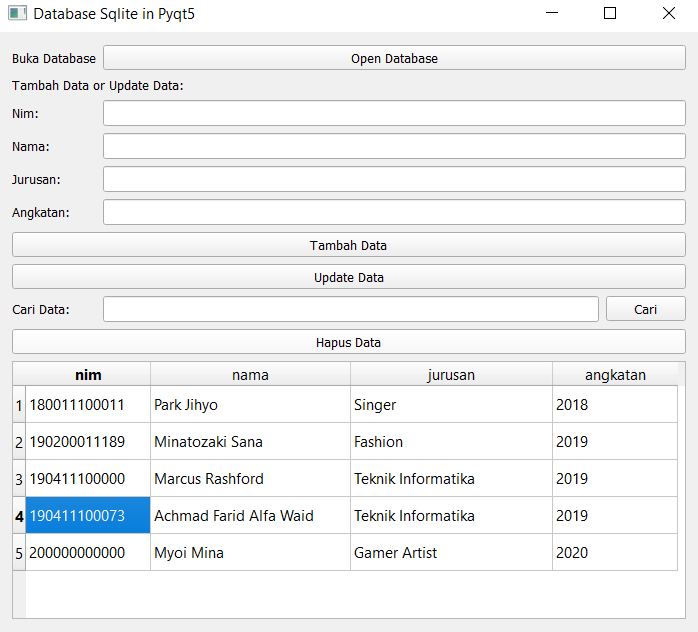
Fungsi ini akan berjalan ketika cursor pada computer memilih salah satu record data, baik itu pada kolom data nim, nama, jurusan, ataupun Angkatan. Jika querynya berhasil maka record data akan terhapus, jika gagal pada terminal akan muncul kesalahan / errornya ada di mana.

**Capture hasil running**

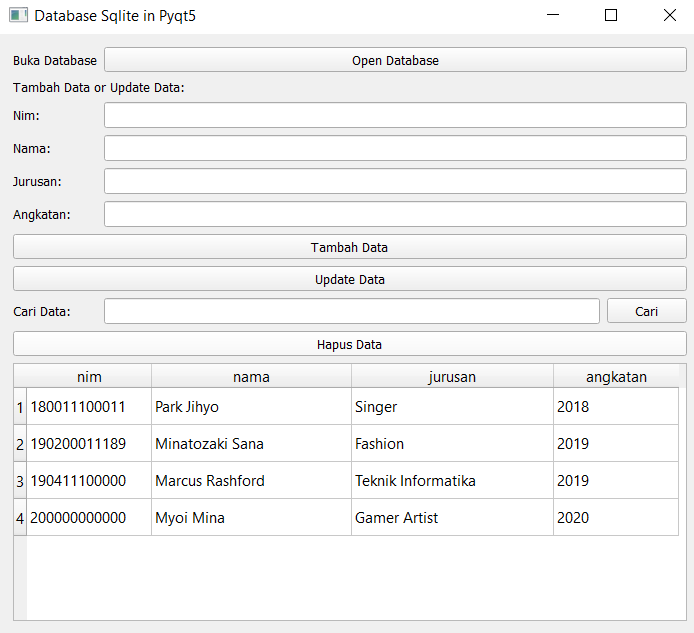
Tampilan awal



Kemudian open database, dan pilih data record yang akan di hapus. Pada contoh ini saya akan menghapus data record ke-4 dengan nim 190411100073.



Kemudian klik button Hapus Data, maka data record tersebut akan terhapus, seperti berikut.



1. Pertama membuat 4 line edit dan tombol button Update Data terlebih dahulu, dan kemudian menghubungkan button tersebut ke fungsi updateData

self.Nim = QLineEdit(self)

grid.addWidget(self.Nim,2,1,1,2)

self.Nama = QLineEdit(self)

grid.addWidget(self.Nama,3,1,1,2)

self.Jurusan = QLineEdit(self)

grid.addWidget(self.Jurusan,4,1,1,2)

self.Angkatan = QLineEdit(self)

grid.addWidget(self.Angkatan,5,1,1,2)

update\_button = QPushButton("Update Data")

grid.addWidget(update\_button,7,0,1,0)

update\_button.clicked.connect(self.updateData)

Kemudian membuat fungsi updateData yang akan mengeksekusi perintah update data pada record tertentu di dalam database.

def updateData(self):

# Mengambil Data Yang Di klik

nimln = str(self.Nim.text())

namaln = str(self.Nama.text())

jurusanln = str(self.Jurusan.text())

angkatanln = str(self.Angkatan.text())

#Mendefinisikan Query

query = QtSql.QSqlQuery()

#Menjalankan Perintah Sql

query.prepare("""UPDATE mahasiswa SET nama=:namaln, jurusan=:jurusanln, angkatan=:angkatanln WHERE nim =:nimln""")

query.bindValue(":nimln", nimln)

query.bindValue(":namaln", namaln)

query.bindValue(":jurusanln", jurusanln)

query.bindValue(":angkatanln", angkatanln)

if query.exec\_():

self.Nim.setText("")

self.Nama.setText("")

self.Jurusan.setText("")

self.Angkatan.setText("")

#Menampilkan Data

self.tampilData()

else:

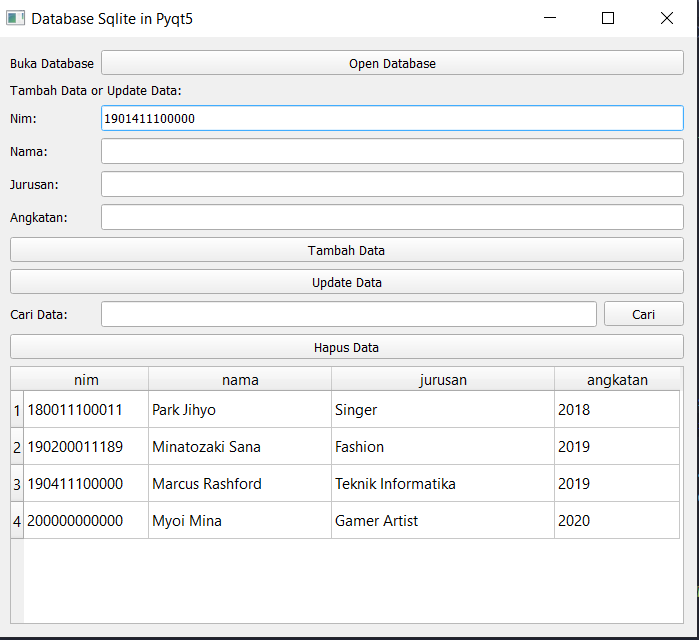
#Apabila error akan menampilkan errornya ke dalam terminal

print("Insert Error: ", query.lastError().text())

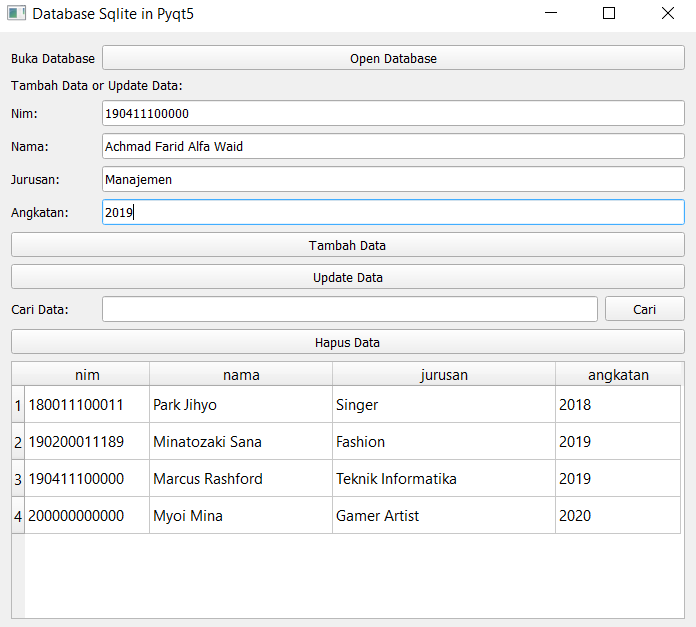
Fungsi ini akan berjalan ketika pada line edit self.Nim sesuai dengan yang ada di record, artinya self.Nim disini sebagai penentu data record mana yang akan di update. Kemudian pada line edit self.Nama, self.Jurusan, self.Angkatan dapat diisi dengan data baru yang nantinya akan di update pada data record terbaru.

**Capture Hasil Program**

Disini saya akan memasukkan nim 190411100000 sebagai penentu bahwa data record tersebut akan di update/edit



Kemudian memasukkan data baru ke dalam line edit Nama, Jurusan, Angkatan sebagai berikut.



Selanjutnya adalah mengklik button Update Data untuk mengeksekusi data baru tersebut, seperti gambar di bawah ini. Yang awalnya bernama Marcus Rashford dan jurusan Teknik Informatika akan berganti dengan hasil inputan data yang baru.

