

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 QSql
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 = QSql.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 memanggil 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 menampilkan 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 = QSql.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__':

```

```

#Inisisialisai 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")
#menampilkan isi dari variabel ex
ex.show()
#membuat system exit
sys.exit(app.exec_())

```

2. SOAL!

- 1) Bagaimana cara menghapus record dari sebuah tabel ? beri contoh, buat program, jelaskan, berikan capture hasil running nya
- 2) Bagaimana cara merubah record tertentu dari sebuah tabel ? beri contoh, buat program, jelaskan, berikan capture hasil running nya

3. JAWAB!

- 1) 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

The screenshot shows a Qt5 application window titled "Database Sqlite in Pyqt5". It features a sidebar with buttons: "Buka Database", "Tambah Data or Update Data:", "Nim:", "Nama:", "Jurusan:", "Angkatan:", "Tambah Data", "Update Data", "Cari Data:", and "Hapus Data". The main area contains a large empty text box.

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

The screenshot shows the same application window, but now it displays a table of student records. The 4th record is highlighted in blue.

	nim	nama	jurusan	angkatan
1	180011100011	Park Jihyo	Singer	2018
2	190200011189	Minatozaki Sana	Fashion	2019
3	190411100000	Marcus Rashford	Teknik Informatika	2019
4	190411100073	Achmad Farid Alfa Waid	Teknik Informatika	2019
5	200000000000	Myoi Mina	Gamer Artist	2020

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

	nim	nama	jurusan	angkatan
1	180011100011	Park Jihyo	Singer	2018
2	190200011189	Minatozaki Sana	Fashion	2019
3	190411100000	Marcus Rashford	Teknik Informatika	2019
4	200000000000	Myoi Mina	Gamer Artist	2020

- 2) 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())
```

```

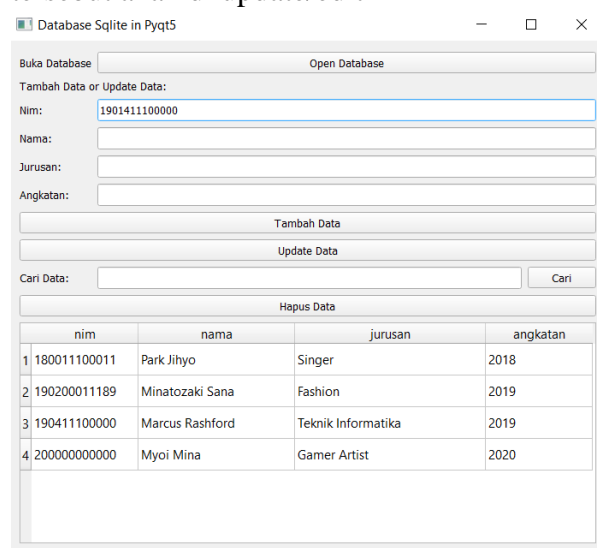
#Mendefinisikan Query
query = QSql.QSqlQuery()
#Menjalankan Perintah Sql
query.prepare("""UPDATE mahasiswa SET
nama=:namaln, jurusan=:jurusanln, angkatan=:angkatanln
WHERE nim =:nimln""")
query.bindValue(":nimln", nimln)
query.bindValue(":namaln", namaln)
query.bindValue(":jurusanln", jurusanln)
query.bindValue(":angkatanln", 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



Database Sqlite in PyQt5

Buka Database

Tambah Data or Update Data:

Nim:

Nama:

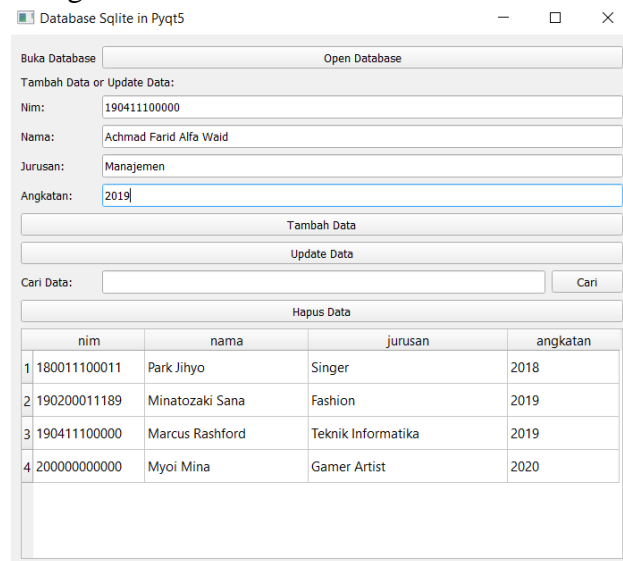
Jurusan:

Angkatan:

Cari Data:

	nim	nama	jurusan	angkatan
1	180011100011	Park Jiho	Singer	2018
2	190200011189	Minatozaki Sana	Fashion	2019
3	190411100000	Marcus Rashford	Teknik Informatika	2019
4	200000000000	Myoi Mina	Gamer Artist	2020

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



Database Sqlite in Pyqt5

Buka Database

Tambah Data or Update Data:

Nim:

Nama:

Jurusan:

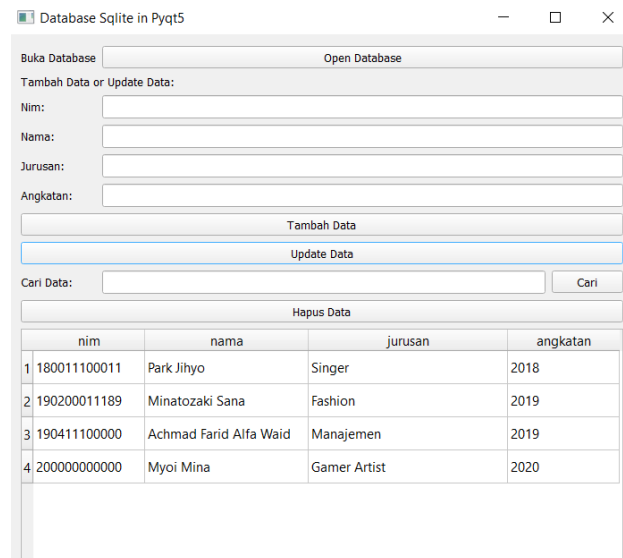
Angkatan:

Cari Data:

Hapus Data

	nim	nama	jurusan	angkatan
1	180011100011	Park Jihyo	Singer	2018
2	190200011189	Minatozaki Sana	Fashion	2019
3	190411100000	Marcus Rashford	Teknik Informatika	2019
4	200000000000	Myoi Mina	Gamer Artist	2020

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.



Database Sqlite in Pyqt5

Buka Database

Tambah Data or Update Data:

Nim:

Nama:

Jurusan:

Angkatan:

Cari Data:

Hapus Data

	nim	nama	jurusan	angkatan
1	180011100011	Park Jihyo	Singer	2018
2	190200011189	Minatozaki Sana	Fashion	2019
3	190411100000	Achmad Farid Alfa Waid	Manajemen	2019
4	200000000000	Myoi Mina	Gamer Artist	2020