MANAJEMEN RESEP MAKANAN

Dosen pengampu :

Pahrul Irfan, S.Kom., M.Kom.

Disusun oleh :

QHAULAN SYAQHILA (F1D022152)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2025

A. Deskripsi singkat topik aplikasi yang dipilih

Aplikasi ini merupakan aplikasi desktop berbasis GUI (Graphical User Interface) yang dikembangkan menggunakan PyQt5 dan SQLite. Aplikasi ini juga dibuat menggunakan Qt Designer dengan nama file "ResepMakanan.ui" dan dihubungkan ke fungsionalitas Python melalui uic.loadUi. Untuk data-data resep disimpan secara lokal menggunakan database SQLite dengan file bernama "resepmakanan.db" . Fungsinya adalah untuk mencatat, mengelola, dan menyimpan data resep masakan pribadi.

Fitur Utama:

1. Tambah Resep, yang dimana pengguna dapat memasukkan resep baru dengan detail seperti nama resep, kategori (misal: snack/cemilan, minuman, makanan berat, dll), tanggal dibuat, tingkat kesulitan, bahan-bahan, dan catatan tambahan (cara pembuatan).
2. Edit Resep, yang dimana pengguna dapat mengubah langsung isi tabel dengan klik ganda pada data yang ingin diubah.
3. Hapus Resep, yang dimana resep tertentu dapat dihapus dari database dengan konfirmasi terlebih dahulu.
4. Ekspor ke CSV, yang dimana semua data resep dapat diekspor ke file CSV untuk keperluan backup atau ingin membuka dengan Excel.
5. Tampilan Tabel, yang dimana semua data resep ditampilkan dalam bentuk tabel dengan kolom seperti nama, kategori, tanggal, kesulitan, bahan, dan catatan.

B. Penjelasan langkah demi langkah pembuatan proyek

1. Pertama-tama mencari ide aplikasi yang ingin dibuat,
2. Kemudian, membuat tampilannya menggunakan QtDesigner,
3. Lalu, mulai membuat program agar tampilan yang telah dibuat di QtDesigner dapat berjalan/berfungsi setiap fitur-fiturnya.
4. Dan yang terakhir melakukan testing kepada setiap fitur yang ada, jika ada kesalahan kembali memperbaiki codingan hingga sampai selesai.

C. Penjelasan fungsi utama yang digunakan dalam aplikasi

```
import sys
import sqlite3
import csv
from PyQt5 import uic
from PyQt5.QtWidgets import (
```

```
    QApplication, QMainWindow, QTableWidgetItem, QFileDialog, QMessageBox,
QInputDialog
)
```

Script diatas merupakan beberapa library utama yang digunakan untuk membangun aplikasi manajemen resep ini. Library sys digunakan untuk mengelola parameter dan proses sistem, seperti menghentikan aplikasi dengan sys.exit() dan menerima argumen dari command line melalui sys.argv. Kemudian, library sqlite3 digunakan untuk menangani basis data lokal atau yang sering kita sebut database menggunakan SQLite. Dengan library ini, aplikasi dapat menyimpan, membaca, mengubah, dan menghapus data resep secara permanen dalam file database (resepmakanan.db). Library csv digunakan untuk melakukan ekspor data dari aplikasi ke dalam format file CSV, yang memudahkan pengguna menyimpan dan membuka data dengan aplikasi spreadsheet seperti Microsoft Excel.

```
def init_db():
    conn = sqlite3.connect("resepmakanan.db")
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS resep (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            nama TEXT,
            kategori TEXT,
            tanggal TEXT,
            kesulitan TEXT,
            catatan TEXT,
            bahan TEXT
        )
    """)
    conn.commit()
    conn.close()
```

Fungsi diatas berfungsi untuk menginisialisasi database SQLite yang digunakan untuk aplikasi. Jika fungsi ini dijalankan, sistem akan membuka atau membuat file database dengan nama “resepmakanan.db”. Setelah itu, dibuat sebuah objek cursor yang digunakan untuk mengeksekusi perintah SQL. Perintah SQL yang dijalankan adalah “CREATE TABLE IF NOT EXISTS, yang bertujuan untuk membuat tabel bernama resep jika tabel tersebut belum ada sebelumnya. Tabel resep memiliki tujuh kolom: id sebagai primary key yang bernilai unik dan bertambah otomatis, nama untuk menyimpan nama resep, kategori untuk jenis masakan, tanggal sebagai tanggal penambahan resep, kesulitan untuk tingkat kesulitan memasak, catatan untuk informasi tambahan, dan bahan untuk daftar bahan-bahan yang digunakan. Setelah

pembuatan tabel, perubahan disimpan ke dalam database menggunakan commit(), lalu koneksi database ditutup dengan close(). Dengan demikian, fungsi ini memastikan bahwa struktur database yang diperlukan sudah siap digunakan sebelum aplikasi berjalan.

```
def __init__(self):
        super().__init__()
        uic.loadUi("ResepMakanan.ui", self)
        self.resize(1020, 800)

        self.setWindowTitle("Manajemen Resep Masakan")

        # Hubungkan tombol ke fungsi
        self.tambah_btn.clicked.connect(self.tambah_resep)
        self.ekspor_btn.clicked.connect(self.ekspor_csv)
        self.hapus_btn.clicked.connect(self.hapus_resep)

        # Hubungkan klik ganda ke fungsi edit
        self.table.cellDoubleClicked.connect(self.edit_resep)

        # Hubungkan menu Help dan Exit
        self.actionExit.triggered.connect(self.keluar_aplikasi)
        self.actionHelp.triggered.connect(self.tampilkan_bantuan)

        self.create_dock_widget()

        self.load_data()
```

Fungsi diatas digunakan untuk menyambungkan file ui dengan program (python) dengan nama window yaitu Manajemen Resep Masakan. Di fungsi ini juga berguna untuk menyambungkan tombol-tombol (button-button) yang ada pada program seperti tambah resep, ekspor, dan hapus resep, serta action exit, dan action help pada menu bar. Fungsi ini juga digunakan untuk mengubah data, maksudnya jika user ingin mengubah resep dapat di klik 2 kali.

```
def tambah_resep(self):
        nama = self.nama_input.text()
        kategori = self.kategori_input.currentText()
        tanggal = self.tanggal_input.text()
        kesulitan = self.kesulitan_input.currentText()
        bahan = self.bahan_input.toPlainText()
        catatan = self.catatan_input.toPlainText()
```

```
        if not nama or not kategori or not bahan:
             QMessageBox.warning(self, "Peringatan", "Nama, Kategori, dan
Bahan wajib diisi!")
            return

        conn = sqlite3.connect("resepmakanan.db")
        cursor = conn.cursor()
        cursor.execute("""
            INSERT INTO resep (nama, kategori, tanggal, kesulitan, bahan,
catatan)
            VALUES (?, ?, ?, ?, ?, ?)
        """, (nama, kategori, tanggal, kesulitan, bahan, catatan))
        conn.commit()
        conn.close()

        self.clear_inputs()
        self.load_data()
```

Fungsi diatas digunakan untuk menambah resep baru oleh user, yang dimana user diminta untuk memasukkan nama resep, kategori, tanggal, kesulitan, bahan dan catatan/proses pembuatannya tetapi untuk nama, kategori, dan bahan wajib diisi oleh user, jika tidak di isi akan muncul peringatan untuk mengisi ketiga tabel tersebut. Kemudian fungsi ini juga akan menambah resep ke dalam database. Serta fungsi ini akan menghapus data-data yang telah ditambah secara otomatis, jadi user dapat dengan mudah menambah resep baru lagi.

```
def hapus_resep(self):
        selected_row = self.table.currentRow()
        if selected_row < 0:
           QMessageBox.warning(self, "Peringatan", "Pilih resep yang ingin
dihapus.")
            return

        id_item = self.table.item(selected_row, 0)
        nama_item = self.table.item(selected_row, 1)

        if not id_item:
            return

        resep_id = id_item.text()
        nama_resep = nama_item.text() if nama_item else "(Tanpa Nama)"
```

```
        reply = QMessageBox.question(
            self,
            "Konfirmasi",
            f"Yakin ingin menghapus resep '{nama_resep}'?",
            QMessageBox.Yes | QMessageBox.No
        )
        if reply == QMessageBox.Yes:
            conn = sqlite3.connect("resepmakanan.db")
            cursor = conn.cursor()
            cursor.execute("DELETE FROM resep WHERE id = ?", (resep_id,))
            conn.commit()
            conn.close()
            self.load_data()
```

Fungsi diatas digunakan untuk menghapus resep yang ingin dihapus. Fungsi ini akan menghapus resep apabila user menekan salah satu tabel dari baris resep yang akan di hapus. Dan fungsi ini juga akan mengkonfirmasi terlebih dahulu sebelum user menghapus resep. Lalu kemudian pada fungsi ini juga langsung menghapus data resep yang dihapus pada database.

```
def edit_resep(self, row, column):
        try:
            # Ambil ID dari baris yang diklik (kolom 0 disembunyikan tapi
tetap bisa diakses)
            resep_id = int(self.table.item(row, 0).text())

            # Daftar nama kolom di database sesuai index
            kolom_nama = {
                1: "nama",
                2: "kategori",
                3: "tanggal",
                4: "kesulitan",
                5: "bahan",
                6: "catatan"
            }

            if column not in kolom_nama:
               QMessageBox.information(self, "Info", "Kolom ini tidak bisa
diedit langsung.")
                return

            current_value = self.table.item(row, column).text()
```

```
            # Pilih jenis dialog berdasarkan kolom
            if column in [5, 6]:
               new_value, ok = QInputDialog.getMultiLineText(self, "Edit",
f"Ubah {kolom_nama[column].capitalize()}:", text=current_value)
            else:
                new_value, ok = QInputDialog.getText(self, "Edit", f"Ubah
{kolom_nama[column].capitalize()}:", text=current_value)

            if not ok or not new_value.strip():
                return

            # Update hanya kolom yang diedit
            conn = sqlite3.connect("resepmakanan.db")
            cursor = conn.cursor()
            cursor.execute(f"""
                UPDATE resep
                SET {kolom_nama[column]} = ?
                WHERE id = ?
            """, (new_value.strip(), resep_id))
            conn.commit()
            conn.close()

            self.load_data()
                                QMessageBox.information(self,    "Sukses",
f"{kolom_nama[column].capitalize()} berhasil diperbarui.")

        except Exception as e:
                        QMessageBox.critical(self,   "Kesalahan",   f"Terjadi
kesalahan:\n{str(e)}")
```

Fungsi diatas digunakan untuk mengedit resep yang ingin diedit. Fungsi ini akan berjalan ketika user telah menekan tabel yang ingin diedit, dan fungsi ini akan juga akan menampilkan tombol oke atau cancel. Jika user menekan tombol oke maka resep akan di update begitu sebaliknya jika user menekan cancel maka resep tidak melakukan perubahan.

```
def load_data(self):
        self.table.setColumnCount(7)
              self.table.setHorizontalHeaderLabels(["ID",   "Nama",
"Kategori", "Tanggal", "Kesulitan", "Bahan", "Catatan"])
        self.table.setRowCount(0)
```

```
        conn = sqlite3.connect("resepmakanan.db")
        cursor = conn.cursor()
            cursor.execute("SELECT id, nama, kategori, tanggal,
kesulitan, bahan, catatan FROM resep")
        for row_index, row_data in enumerate(cursor.fetchall()):
            self.table.insertRow(row_index)
            for col_index, data in enumerate(row_data):
                        self.table.setItem(row_index, col_index,
QTableWidgetItem(str(data)))

        self.table.setColumnHidden(0, True)
        conn.close()
```

Fungsi ini digunakan untuk mengambil data-data resep pada database dan data-data tersebut akan ditampilkan pada tabel GUI. Pertama-tama fungsi ini akan mengatur tabel mejadi tujuh yaitu id, nama, kategori, kesulitan, bahan dan catatan. Tetapi untuk kolom ID disembunyikan agar tampilannya lebih bersih dan rapi, namun nilainya tetap digunakan secara internal. Jadi pada tabel GUI terdapat 6 tabel yang ditampilkan. Sebelum di masukkan ke dalam tabel GUI, tabel dikosongkan terlebih dahulu agar tidak terjadi duplikasi. Kemudian data diambil dari database dan dimasukkan kedalam tabel GUI.

```
def ekspor_csv(self):
       path, _ = QFileDialog.getSaveFileName(self, "Simpan File", "", "CSV
Files (*.csv)")
        if path:
            conn = sqlite3.connect("resepmakanan.db")
            cursor = conn.cursor()
              cursor.execute("SELECT nama, kategori, tanggal, kesulitan,
bahan, catatan FROM resep")
            data = cursor.fetchall()
            conn.close()

            with open(path, 'w', newline='', encoding='utf-8') as file:
                writer = csv.writer(file)
              writer.writerow(["Nama", "Kategori", "Tanggal", "Kesulitan",
"Bahan", "Catatan"])
                writer.writerows(data)
```

```
            QMessageBox.information(self, "Sukses", "Data berhasil diekspor
ke CSV!")
```

Fungsi ini digunakan untuk mengekpor data-data resep yang berada pada database ke dalam file dengan format CSV agar user dapat melihat melalui spreadsheet pada Microsoft Excel. Pertama-tama user diminta untuk memilih tempat file tersebut akan disimpan. Kemudian, setelah memilih tempat disimpannya file tersebut dan menekan tombol save, maka aplikasi akan membuka koneksi ke database ”resepmakanan.db”. setelah data berhasil diambil, akan muncul kotak dialog yang menginformasikan bahwa data berhasil di ekspor.

```
def clear_inputs(self):
        self.nama_input.clear()
        self.kategori_input.clear()
        self.tanggal_input.clear()
        self.kesulitan_input.setCurrentIndex(0)
        self.bahan_input.clear()
        self.catatan_input.clear()
```

Fungsi ini digunakan untuk mengkosongkan semua field input dalam formulir user setelah data berhasil ditambah. Dan funngsi ini juga memastikan bahwa semua elemen formular kembali ke keadaan awal (kosong) dan siap diisi untuk data berikutnya.

```
def keluar_aplikasi(self):
        reply = QMessageBox.question(
            self,
            "Konfirmasi Keluar",
            "Apakah Anda yakin ingin keluar dari aplikasi?",
            QMessageBox.Yes | QMessageBox.No
        )
        if reply == QMessageBox.Yes:
            QApplication.quit()
```

Fungsi ini digunakan untuk menangani proses keluar dari aplikasi dengan melakukan konfirmasi terlebih dahulu kepada user. Ada dua piliahan yang diberikan oleh aplikasi ini yaitu yes atau no. Jika user menekan tombol yes maka aplikasi akan keluar jita tidak maka aplikasi tetap berjalan.

```
def tampilkan_bantuan(self):
        QMessageBox.information(
            self,
            "Bantuan",
            "Ini adalah aplikasi Manajemen Resep Masakan.\n\n"
            "Panduan Aplikasi sebagai berikut: \n"
            "- Tambah: Menyimpan resep baru.\n"
```

```
            "- Hapus: Menghapus resep yang dipilih.\n"
            "- Ekspor: Menyimpan semua resep ke file CSV.\n"
            "- Klik dua kali pada tabel untuk mengedit data.\n"
            "- Menu File → Exit untuk keluar dari aplikasi."
        )
```

Fungsi ini digunakan untuk menampilkan informasi bantuan kepada user dalam bentuk kotak dialog (message box). Fungsi ini akan menampilkan beberapa panduan saat menggunakan aplikasi Manajemen Resep Makanan ini.

```
def create_dock_widget(self):
        self.dock = QDockWidget("Petunjuk Aplikasi", self)


           dock_text = QTextEdit("Aplikasi ini dapat digunakan untuk memenajemen/menyimpan resep makan dari pengguna. Untuk petunjuk lebih lanjut Anda dapat menakan tombol file di sebelah kiri atas dan menekan tombol bantuan")
        dock_text.setReadOnly(True)
        self.dock.setWidget(dock_text)
        self.addDockWidget(Qt.RightDockWidgetArea, self.dock)
```

Fungsi ini digunakan untuk menambah dock widget ke dalam halaman utapa aplikasi. Donk widget ini berisi petunjuk aplikasi sehingga penggunaka bisa mambaca informasi tambahan pada aplikasi. Dock eidget ini juga akan berisi penjelasan singkat kegunaan aplikasi dan intruksi untuk membuka menu bar yang Dimana di menu bar bantuan terdapat petunjuk lebih lengkap dari aplikasi ini.

```
if __name__ == '__main__':
    init_db()
    app = QApplication(sys.argv)
    window = ResepApp()
    window.show()
    sys.exit(app.exec_())
```

Script diatas digunakan untuk memastikan bahwa code didalam aplikasi ini berjalan saat file di eksekusi langsung. Fungsi init_db() dipanggil terlebih dahulu utntuk memastikan database dan tabel sudah ada atau dibuat jika belum ada. Kemudian, QApplication(sys.argv) membuat aplikasi Qt yang diperlukan untuk menampilkan GUI. Objek ResepApp() (yang merupakan subclass dari QMainWindow) dibuat dan ditampilkan di layar dengan window.show(). Terakhir, sys.exit(app.exec_()) akan menjalankan loop utama Qt dan keluar dari program dengan kode status yang sesuai setelah jendela aplikasi ditutup.

D. Screenshot antarmuka dan fungsionalitas yang relevan