

LAPORAN TUGAS PROJEK MATA KULIAH
PEMROGRAMAN VISUAL

StuDate: Student Daily Assignment & Task Entry



Disusun Oleh:

Kayla Mizanti

(F1D022127)

PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2025

A. LATAR BELAKANG

Dalam dunia perkuliahan, mahasiswa sering kali dihadapkan dengan berbagai tugas dari banyak mata kuliah yang memiliki tenggat waktu berbeda-beda. Tidak jarang, mahasiswa mengalami kesulitan dalam mengelola dan mengingat semua tugas tersebut, terutama ketika jumlahnya cukup banyak dan datang secara bersamaan. Kondisi ini dapat menyebabkan keterlambatan pengerjaan, bahkan kelalaian dalam menyelesaikan tugas.

Seiring dengan perkembangan teknologi, kebutuhan akan alat bantu yang dapat mendukung manajemen tugas secara digital menjadi semakin penting. Oleh karena itu, dibutuhkan sebuah aplikasi sederhana yang mampu mencatat, mengelola, dan menampilkan daftar tugas mahasiswa secara terorganisir, efisien, dan mudah digunakan.

Berdasarkan kebutuhan tersebut, dibuatlah aplikasi StuDate (Student Daily Assignment & Task Entry). Aplikasi ini dikembangkan menggunakan PyQt5 sebagai salah satu implementasi pembelajaran pada mata kuliah Pemrograman Visual. StuDate memungkinkan mahasiswa untuk memasukkan data tugas berdasarkan mata kuliah, judul tugas, catatan, tanggal tenggat, serta status prioritas. Selain itu, aplikasi ini dilengkapi fitur pencarian, penyaringan status tugas, serta ekspor data ke dalam format CSV, sehingga pengguna dapat mengelola tugas-tugas mereka secara lebih efektif dan rapi.

Dengan adanya aplikasi StuDate, diharapkan mahasiswa dapat meningkatkan manajemen waktu, menyelesaikan tugas tepat waktu, dan mengurangi risiko keterlambatan atau kelupaan terhadap deadline tugas-tugas akademik mereka.

B. DESKRIPSI APLIKASI

Aplikasi StuDate (Student Daily Assignment & Task Entry) merupakan sebuah sistem manajemen tugas mahasiswa berbasis desktop yang dikembangkan menggunakan framework PyQt5. Aplikasi ini dirancang untuk membantu mahasiswa dalam mengatur dan memantau berbagai tugas perkuliahan secara lebih tertib dan efisien, khususnya dalam menghadapi banyaknya beban akademik yang memiliki deadline berbeda-beda.

StuDate menawarkan fitur utama berupa form input tugas yang mencakup lima data penting, yaitu nama mata kuliah, judul tugas, catatan tambahan, tanggal deadline, dan penanda prioritas. Setiap entri tugas yang ditambahkan secara otomatis akan ditampilkan dalam tabel dinamis yang berisi informasi lengkap termasuk ID, nama mata kuliah, judul tugas, tanggal, status, dan catatan. Tugas yang ditandai sebagai prioritas akan disimbolkan dengan emoji yang ditampilkan sebelum nama mata kuliah di tabel, memberikan penanda visual yang cepat dikenali.

Selain itu, aplikasi ini juga dilengkapi dengan fitur pencarian berdasarkan nama mata kuliah atau judul tugas, serta tombol Paste yang memudahkan pengguna dalam menyalin informasi dari clipboard. Di bawah fitur pencarian, terdapat tiga tombol utama yang berfungsi untuk menambahkan tugas baru, menghapus tugas yang dipilih, dan memfilter tampilan tugas berdasarkan status penggerjaan (*Sudah Dikerjakan* atau *Belum Dikerjakan*). Status tugas secara default akan ditetapkan sebagai *Belum Dikerjakan* saat data baru ditambahkan, namun pengguna dapat langsung mengubahnya melalui tabel sesuai kebutuhan.

Fitur pengeditan langsung pada tabel disediakan secara selektif hanya untuk kolom Status dan Catatan, sehingga pengguna dapat memperbarui informasi tanpa perlu menginput ulang seluruh data. Untuk keperluan dokumentasi atau penyimpanan data, aplikasi juga menyediakan tombol Export ke CSV di bagian kanan bawah, memungkinkan data tugas disimpan dalam format file yang dapat diakses dan dibagikan dengan mudah.

Dari sisi antarmuka, StuDate mengusung desain yang rapi dan user-friendly, dengan komponen-komponen yang tersusun responsif mengikuti ukuran layar. Warna tema yang digunakan memberikan kesan lembut dan fokus, mendukung kenyamanan pengguna saat menggunakan aplikasi dalam jangka waktu lama.

Seluruh data disimpan secara lokal menggunakan database SQLite, menjadikan aplikasi ini ringan, cepat, dan dapat digunakan tanpa koneksi internet. StuDate dikembangkan sebagai proyek tugas mata kuliah Pemrograman Visual dengan memanfaatkan berbagai komponen penting dari PyQt5, seperti QTableWidget, QLineEdit, QDateEdit, QCheckBox, QPushButton, serta fitur clipboard, scroll area, dock widget, dan status bar yang mencerminkan penerapan materi perkuliahan secara menyeluruuh.

C. LANGKAH PEMBUATAN APLIKASI

1. Inisialisasi Proyek dan Struktur Database

Tahap awal dilakukan dengan membuat class utama StuDate yang mewarisi QMainWindow. Di dalam `__init__`, dibentuk koneksi ke SQLite menggunakan `sqlite3.connect("studeate.db")`. Fungsi `create_table()` dipanggil untuk membuat tabel tugas secara otomatis jika belum ada, tanpa membuat database manual di SQLite browser.

- a. id (INTEGER, primary key, auto increment)
- b. matkul (TEXT)
- c. judul (TEXT)
- d. deadline (TEXT)

- e. status (TEXT)
- f. catatan (TEXT)
- g. prioritas (INTEGER; 0 atau 1)

2. Mendesain Antarmuka

Semua tampilan antarmuka dibuat langsung melalui kode Python, menggunakan kombinasi QVBoxLayout dan QHBoxLayout. Komponen yang digunakan meliputi:

- a. QLabel: Judul aplikasi
- b. QLineEdit: Input teks untuk mata kuliah dan judul tugas.
- c. QTextEdit: Untuk catatan tambahan pengguna.
- d. QDateEdit: Pemilih tanggal dengan tampilan kalender.
- e. QCheckBox: Untuk menandai tugas prioritas.
- f. QComboBox: Memfilter status tugas (semua, selesai, belum).
- g. QPushButton: Untuk fungsi Tambah, Hapus, Paste clipboard, dan Export CSV.
- h. QTableWidget: Menampilkan data tugas secara tabel dengan kolom dinamis.
- i. QScrollArea: Membuat tabel bisa di-scroll jika banyak data.
- j. QStatusBar: Menampilkan identitas pembuat aplikasi.
- k. QMenuBar & QAction: Menyediakan menu navigasi tambahan (File, About).

3. Implementasi Fitur

Beberapa fungsi utama yang diimplementasikan agar aplikasi dapat digunakan dengan baik antara lain:

- a. add_task: Menyimpan data ke database berdasarkan input user.
- b. load_data: Mengambil dan menampilkan data dari database ke tabel, termasuk emoji jika tugas diprioritaskan.
- c. delete_task: Menghapus data yang dipilih dari tabel dan database.
- d. handle_cell_click: Memunculkan popup saat kolom status diklik, untuk mengubahnya menjadi "Selesai".
- e. update_cell: Menyimpan perubahan catatan tugas secara langsung saat diedit dari tabel.
- f. clear_inputs: Membersihkan semua input form setelah data ditambahkan.

4. Implementasi Fitur Pencarian dan Filter

- a. search_task: Menyaring isi tabel berdasarkan kata kunci dari input pencarian (mata kuliah/judul).
- b. filter_status: Menampilkan tugas berdasarkan status: semua, belum dikerjakan, atau selesai.

5. Fitur Tambahan untuk Pengguna

- a. Clipboard Paste: Dengan tombol khusus, pengguna bisa menempelkan teks dari clipboard ke field yang sedang aktif (baik QLineEdit maupun QTextEdit).

- b. Menu Bar:

File > Export ke CSV: Untuk menyimpan data sebagai dokumen offline.

About: Menampilkan informasi nama dan NIM pembuat aplikasi.

- c. Status Bar: Menampilkan informasi pembuat aplikasi secara statis di bagian bawah (non-editable).

6. Ekspor Data ke CSV

- a. Dengan menggunakan QFileDialog, pengguna bisa memilih lokasi penyimpanan file .csv.
- b. Data dari tabel akan diekspor dengan header kolom yang sesuai.

D. FUNGSI UTAMA APLIKASI

1. Fungsi Menambah Tugas (add_task) :

Fungsi ini menangani proses memasukkan data baru ke dalam database tugas. Data yang diambil dari input form: nama mata kuliah, judul tugas, catatan, tanggal deadline, dan status prioritas. Setelah divalidasi (tidak kosong), data disimpan dengan status default "Belum Dikerjakan".

2. Fungsi Menampilkan Tabel Data (load_data)

Bertugas mengambil data dari database dan menampilkannya ke dalam QTableWidget. Fitur filter berdasarkan status (Selesai/Belum) dan pencarian (matkul/judul) disatukan dalam logika query. Jika tugas ditandai prioritas, maka simbol api akan ditampilkan di kolom *Mata Kuliah*.

3. Fungsi Menghapus Tugas (delete_task)

Menangani penghapusan data berdasarkan baris yang dipilih pengguna pada tabel. Disertai konfirmasi melalui dialog sebelum data dihapus secara permanen dari database.

4. Fungsi Edit Kolom Status (handle_cell_click)

Jika pengguna mengklik kolom *Status*, akan muncul konfirmasi untuk mengubahnya dari "Belum Dikerjakan" ke "Selesai". Fungsi ini tidak memungkinkan edit langsung dari tabel untuk status, melainkan melalui pop-up.

5. Fungsi Edit Kolom Catatan (update_cell)

Catatan tugas dapat diubah langsung dari tabel tanpa perlu membuka form input. Perubahan langsung disimpan ke database ketika pengguna selesai mengetik.

6. Fungsi Pencarian (search_task)

Input pencarian akan menyaring isi tabel berdasarkan kata kunci pada nama mata kuliah atau judul tugas. Hasil akan ditampilkan secara real-time ketika teks diketik.

7. Fungsi Filter Status (filter_status)

Dropdown filter akan mengubah tampilan tabel untuk menampilkan tugas sesuai status yang dipilih: semua, belum, atau selesai.

8. Fungsi Clipboard (paste_clipboard)

Tombol paste digunakan untuk menempelkan teks dari clipboard ke input aktif, baik QLineEdit atau QTextEdit. Mempermudah pengguna saat ingin menyalin info dari sumber lain.

9. Fungsi Ekspor ke CSV (export_csv)

Memungkinkan pengguna menyimpan seluruh data tugas ke dalam file CSV. Berguna untuk dokumentasi, cetak, atau backup.

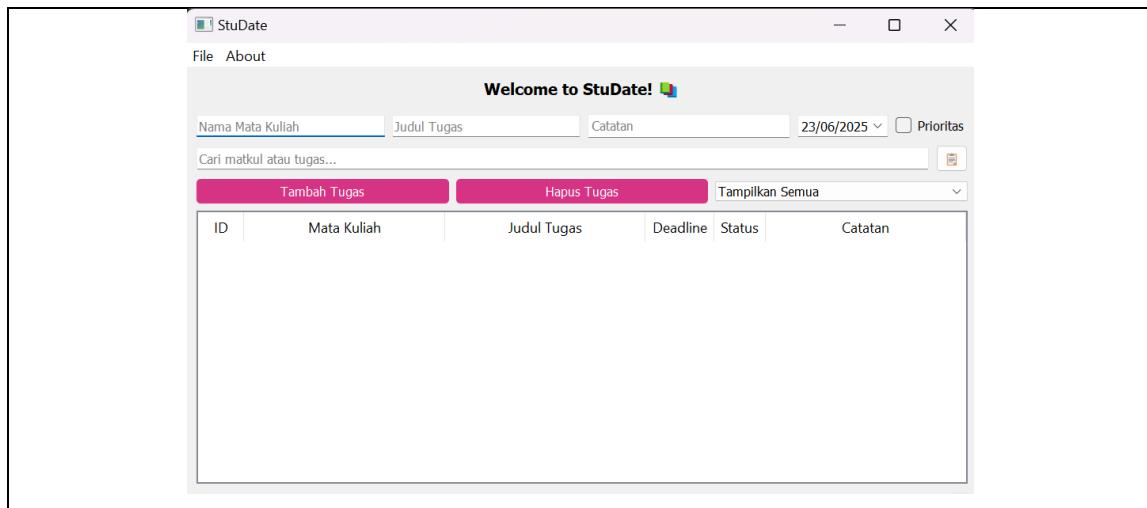
10. Fungsi Menu Bar (setup_menu)

Menambahkan dua fitur penting:

File > Export ke CSV: Shortcut untuk ekspor tanpa perlu menekan tombol di bawah tabel.

About: Menampilkan identitas pembuat aplikasi.

E. TAMPILAN APLIKASI



Welcome to StuDate!

Tambah Tugas Hapus Tugas Tampilkan Semua

ID	Mata Kuliah	Judul Tugas	Deadline	Status	Catatan
1 10	🔥 IOT	PROJEK UAS	2025-06-23	Belum Dikerjakan	buat rangkaian simulasi
2 11	Pemrograman Visual	week 12	2025-06-23	Selesai	mengintegrasikan mat...
3 12	Pemsis	Projek UAS	2025-06-25	Belum Dikerjakan	materi ANOVA
4 13	🔥 TKTI	PRESENTASI AKHIR	2025-06-15	Belum Dikerjakan	BUAT MATERI MATURITY
5 14	🔥 SMART CITY	WINIT	2025-06-30	Belum Dikerjakan	BUAT PITCH DECK
6 15	Logika Fuzzy	buat jurnal	2025-06-29	Belum Dikerjakan	kumpulkan data
7 16	🔥 Logika Fuzzy	jurnal fix	2025-06-30	Belum Dikerjakan	perhitungan sukamoto

Berhasil

Tugas 'PROJEK UAS' berhasil ditambahkan!

OK

Ubah Status

Ubah status menjadi 'Selesai'?

Yes No

Welcome to StuDate!

Tambah Tugas Hapus Tugas Tampilkan Semua

ID	Mata Kuliah	Judul Tugas	Deadline	Status	Catatan
1 10	🔥 IOT	PROJEK UAS	2025-06-23	Belum Dikerjakan	buat rangkaian simulasi
2 11	Pemrograman Visual	week 12	2025-06-23	Selesai	mengintegrasikan mat...
3 12	Pemsis	Projek UAS	2025-06-25	Belum Dikerjakan	materi ANOVA
4 13	🔥 TKTI	PRESENTASI AKHIR	2025-06-15	Belum Dikerjakan	BUAT MATERI MATURITY
5 14	🔥 SMART CITY	WINIT	2025-06-30	Belum Dikerjakan	BUAT PITCH DECK
6 15	Logika Fuzzy	buat jurnal	2025-06-29	Belum Dikerjakan	kumpulkan data
7 16	🔥 Logika Fuzzy	jurnal fix	2025-06-30	Belum Dikerjakan	perhitungan sukamoto

StuDate

Welcome to StuDate! 📚

Tambah Tugas **Hapus Tugas** **Selesai**

ID	Mata Kuliah	Judul Tugas	Deadline	Status	Catatan
1 11	Pemrograman Visual	week 12	2025-06-23	Selesai	mengintegrasikan materi ...
2 12	Pemsis	Projek UAS	2025-06-25	Selesai	materi ANOVA
3 13	🔥 TKTI	PRESENTASI AKHIR	2025-06-15	Selesai	BUAT MATERI MATURITY

StuDate

Welcome to StuDate! 📚

Tambah Tugas **Hapus Tugas** **Tampilkan Semua**

ID	Mata Kuliah	Judul Tugas	Deadline	Status	Catatan
1 15	Logika Fuzzy	buat jurnal	2025-06-29	Belum Dikerjakan	kumpulkan data
2 16	🔥 Logika Fuzzy	jurnal fix	2025-06-30	Belum Dikerjakan	perhitungan sukamoto

StuDate

Welcome to StuDate! 📚

Tambah Tugas **Hapus Tugas** **Tampilkan Semua**

ID	Mata Kuliah	Judul Tugas	Deadline	Status	Catatan
1 10	🔥 IOT	P			
2 11	Pemrograman Visual	w			
3 12	Pemsis	P			
4 13	🔥 TKTI	PRESENTASI AKHIR	2025-06-15	Selesai	BUAT MATERI MATURITY
5 14	🔥 SMART CITY	WINIT	2025-06-30	Belum Dikerjakan	BUAT PITCH DECK
6 15	Logika Fuzzy	buat jurnal	2025-06-29	Belum Dikerjakan	kumpulkan data
7 16	🔥 Logika Fuzzy	jurnal fix	2025-06-30	Belum Dikerjakan	perhitungan sukamoto

Hapus

Pilih tugas yang ingin dihapus.

OK

Welcome to StuDate!

ID	Mata Kuliah	Judul Tugas	Deadline	Status	Catatan
1 10	🔥 IOT	PROJEK UAS	2025-06-23	Belum Dikerjakan	buat rangkaian simulasi
2 11	Pemrograman Visual	week 12	2025-06-23	Selesai	mengintegrasikan mater...
3 12	Pemsis	Projek UAS	2025-06-25	Selesai	materi ANOVA
4 13	🔥 TKTI	PRESENTASI AKHIR	2025-06-15	Selesai	BUAT MATERI MATURITY
5 14	SMART CITY	WINIT	2025-06-30	Belum Dikerjakan	BUAT PITCH DECK
6 15	Logika Fuzzy	buat jurnal	2025-06-29	Belum Dikerjakan	kumpulkan data
7 16	🔥 Logika Fuzzy	jurnal fix	2025-06-30	Belum Dikerjakan	perhitungan sukamoto

Konfirmasi

Yakin ingin menghapus tugas 'WINIT'?

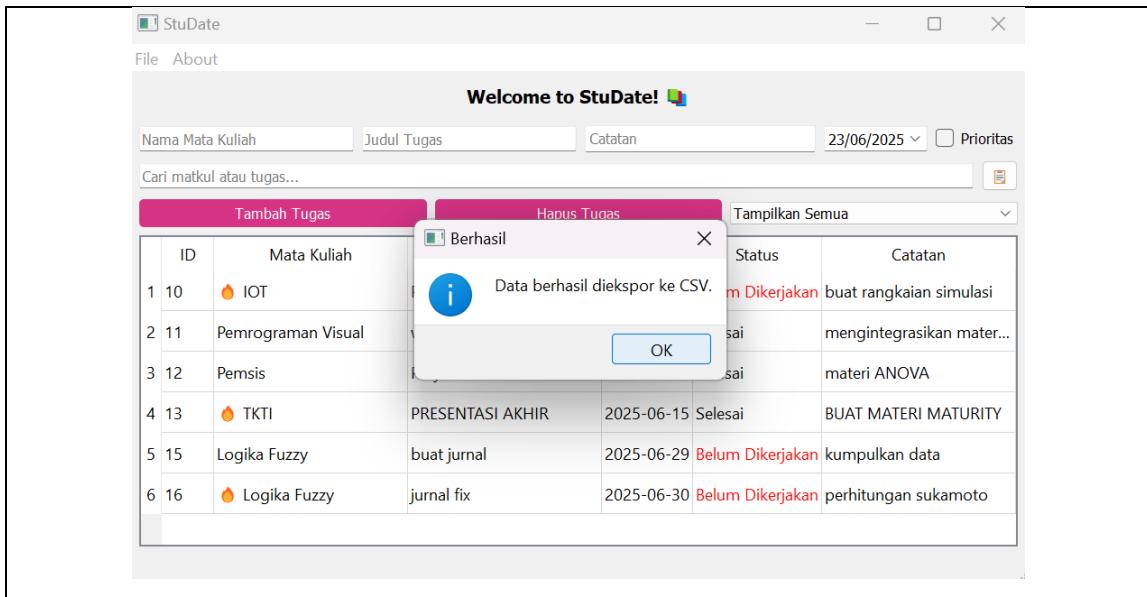
Yes **No**

Tentang

Aplikasi StuDate dibuat oleh Kayla Mizanti
NIM: F1D022127

OK

Export ke CSV



F. CODE PROGRAM

```
import sys, sqlite3, csv
from PyQt5.QtWidgets import (
    QApplication, QMainWindow, QLabel, QLineEdit, QTextEdit, QPushButton,
    QVBoxLayout, QHBoxLayout, QComboBox, QTableWidgetItem, QTableWidget,
    QMessageBox, QCheckBox, QDateEdit, QFileDialog, QScrollArea,
    QStatusBar, QHeaderView, QWidget, QAction, QMenuBar
)
from PyQt5.QtCore import Qt, QDate
class StuDate(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("StuDate")
        self.setGeometry(100, 100, 1000, 600)
        self.conn = sqlite3.connect("studeate.db")
        self.cursor = self.conn.cursor()
        self.create_table()
        self.last_focused_input = None
    # UI setup
    self.main_widget = QWidget()
    self.setCentralWidget(self.main_widget)
    self.setup_menu()
    self.init_ui()
    self.load_data()
    def setup_menu(self):
        menubar = self.menuBar()
        file_menu = menubar.addMenu("File")
        export_action = QAction("Export ke CSV", self)
        export_action.triggered.connect(self.export_csv)
        file_menu.addAction(export_action)
        about_action = QAction("About", self)
        about_action.triggered.connect(lambda: QMessageBox.information(self,
            "Tentang", "Aplikasi StuDate dibuat oleh Kayla Mizanti\nNIM: F1D022127"))
        menubar.addAction(about_action)
    def create_table(self):
        self.cursor.execute("""
            CREATE TABLE IF NOT EXISTS Student (
                ID INTEGER PRIMARY KEY,
                Name TEXT,
                Age INTEGER,
                Gender TEXT,
                Address TEXT,
                Grade TEXT
            )
        """)
```

```

CREATE TABLE IF NOT EXISTS tugas (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    matkul TEXT,
    judul TEXT,
    deadline TEXT,
    status TEXT,
    catatan TEXT,
    prioritas INTEGER
) """
self.conn.commit()
def init_ui(self):
    layout = QVBoxLayout()
    # Title
    title = QLabel("Welcome to StuDate! ")
    title.setAlignment(Qt.AlignCenter)
    title.setStyleSheet("font-size: 20px; font-weight: bold; margin-bottom: 10px;")
    layout.addWidget(title)
    # Form
    form_layout = QHBoxLayout()
    self.matkul_input = QLineEdit(placeholderText="Nama Mata Kuliah")
    self.judul_input = QLineEdit(placeholderText="Judul Tugas")
    self.catatan_input = QTextEdit()
    self.catatan_input.setPlaceholderText("Catatan")
    self.catatan_input.setFixedHeight(30)
    self.date_input = QDateEdit(calendarPopup=True)
    self.date_input.setDate(QDate.currentDate())
    self.prioritas_input = QCheckBox("Prioritas")
    for w in [self.matkul_input, self.judul_input, self.catatan_input,
              self.date_input]:
        w.installEventFilter(self)
    form_layout.addWidget(self.matkul_input)
    form_layout.addWidget(self.judul_input)
    form_layout.addWidget(self.catatan_input)
    form_layout.addWidget(self.date_input)
    form_layout.addWidget(self.prioritas_input)
    layout.addLayout(form_layout)
    # Search & Paste
    search_layout = QHBoxLayout()
    self.search_input = QLineEdit(placeholderText="Cari matkul atau tugas...")
    self.search_input.textChanged.connect(self.search_task)
    self.search_input.installEventFilter(self)
    self.paste_button = QPushButton("")
    self.paste_button.setFixedSize(40)
    self.paste_button.clicked.connect(self.paste_clipboard)
    search_layout.addWidget(self.search_input)
    search_layout.addWidget(self.paste_button)
    layout.addLayout(search_layout)
    # Tombol
    pink = """
QPushButton {
    background-color: #d63384;
    color: white;
    border-radius: 5px;
    padding: 6px 12px;
}
QPushButton:hover {
    background-color: #e83e8c;
}
"""
    button_layout = QHBoxLayout()

```

```

self.add_button = QPushButton("Tambah Tugas")
self.add_button.setStyleSheet(pink)
self.add_button.clicked.connect(self.add_task)
self.delete_button = QPushButton("Hapus Tugas")
self.delete_button.setStyleSheet(pink)
self.delete_button.clicked.connect(self.delete_task)
self.filter_combo = QComboBox()
self.filter_combo.addItem("Tampilkan Semua", "Belum Dikerjakan",
"Selesai")
self.filter_combo.currentTextChanged.connect(self.filter_status)
button_layout.addWidget(self.add_button)
button_layout.addWidget(self.delete_button)
button_layout.addWidget(self.filter_combo)
layout.addLayout(button_layout)
# Tabel
self.table = QTableWidget()
self.table.setColumnCount(6)
self.table.setHorizontalHeaderLabels(["ID", "Mata Kuliah", "Judul Tugas",
"Deadline", "Status", "Catatan"])
self.table.setEditTriggers(QTableWidget.DoubleClicked)
self.table.cellClicked.connect(self.handle_cell_click)
self.table.itemChanged.connect(self.update_cell)
header = self.table.horizontalHeader()
header.setSectionResizeMode(QHeaderView.Stretch)
header.setSectionResizeMode(0, QHeaderView.ResizeToContents)
header.setSectionResizeMode(3, QHeaderView.ResizeToContents)
header.setSectionResizeMode(4, QHeaderView.ResizeToContents)
scroll = QScrollArea()
scroll.setWidgetResizable(True)
scroll.setWidget(self.table)
layout.addWidget(scroll)
# Status bar
self.status = QStatusBar()
self.setStatusBar(self.status)
self.status.showMessage(" Nama: Kayla Mizanti | NIM: F1D022127")
# Set layout
self.main_widget.setLayout(layout)
def eventFilter(self, source, event):
if event.type() == event.FocusIn:
self.last_focused_input = source
return super().eventFilter(source, event)
def paste_clipboard(self):
clipboard = QApplication.clipboard()
if isinstance(self.last_focused_input, QLineEdit):
self.last_focused_input.setText(clipboard.text())
elif isinstance(self.last_focused_input, QTextEdit):
self.last_focused_input.setPlainText(clipboard.text())
def add_task(self):
matkul = self.matkul_input.text().strip()
judul = self.judul_input.text().strip()
catatan = self.catatan_input.toPlainText().strip()
deadline = self.date_input.date().toString("yyyy-MM-dd")
prioritas = 1 if self.prioritas_input.isChecked() else 0
if not matkul or not judul:
QMessageBox.warning(self, "Input Salah", "Matkul dan Judul wajib diisi!")
return
self.cursor.execute("""
INSERT INTO tugas (matkul, judul, deadline, status, catatan, prioritas)
VALUES (?, ?, ?, ?, ?, ?)""",
(matkul, judul, deadline, "Belum Dikerjakan", catatan, prioritas))
self.conn.commit()

```

```

self.clear_inputs()
self.load_data()
QMessageBox.information(self, "Berhasil", f"Tugas '{judul}' berhasil ditambahkan!")
def clear_inputs(self):
    self.matkul_input.clear()
    self.judul_input.clear()
    self.catatan_input.clear()
    self.date_input.setDate(QDate.currentDate())
    self.prioritas_input.setChecked(False)
def load_data(self, filter_status=None, search_term=None):
    self.table.blockSignals(True)
    self.table.setRowCount(0)
    query = "SELECT id, matkul, judul, deadline, status, catatan, prioritas FROM tugas"
    params = []
    if filter_status in ("Belum Dikerjakan", "Selesai"):
        query += " WHERE status = ?"
        params.append(filter_status)
    if search_term:
        clause = "(matkul LIKE ? OR judul LIKE ?)"
        if "WHERE" in query:
            query += " AND " + clause
        else:
            query += " WHERE " + clause
        params.extend([f"%{search_term}%" * 2])
    self.cursor.execute(query, params)
    for row_idx, (id_, matkul, judul, deadline, status, catatan, prioritas) in enumerate(self.cursor.fetchall()):
        self.table.insertRow(row_idx)
        self.table.setItem(row_idx, 0, QTableWidgetItem(str(id_)))
        matkul_display = "🔥 " + matkul if prioritas else matkul
        self.table.setItem(row_idx, 1, QTableWidgetItem(matkul_display))
        self.table.setItem(row_idx, 2, QTableWidgetItem(judul))
        self.table.setItem(row_idx, 3, QTableWidgetItem(deadline))
        status_item = QTableWidgetItem(status)
        if status.lower() == "belum dikerjakan":
            status_item.setForeground(Qt.red)
        else:
            status_item.setForeground(Qt.black)
        self.table.setItem(row_idx, 4, status_item)
        self.table.setItem(row_idx, 5, QTableWidgetItem(catatan))
    self.table.blockSignals(False)
def handle_cell_click(self, row, col):
    if col == 4:
        current_status = self.table.item(row, col).text()
        if current_status.lower() == "belum dikerjakan":
            confirm = QMessageBox.question(
                self, "Ubah Status", "Ubah status menjadi 'Selesai'?", QMessageBox.Yes | QMessageBox.No
            )
            if confirm == QMessageBox.Yes:
                id_ = self.table.item(row, 0).text()
                self.cursor.execute("UPDATE tugas SET status = ? WHERE id = ?", ("Selesai", id_))
                self.conn.commit()
                self.load_data()
                self.filter_combo.currentText() if self.filter_combo.currentText() != "Tampilkan Semua" else None,
                self.search_input.text()
            )

```

```

def update_cell(self, item):
    row = item.row()
    col = item.column()
    if col == 5: # Catatan bisa diedit
        id_ = self.table.item(row, 0).text()
        new_value = item.text()
        self.cursor.execute("UPDATE tugas SET catatan = ? WHERE id = ?", (new_value, id_))
        self.conn.commit()
    def delete_task(self):
        selected = self.table.currentRow()
        if selected < 0:
            QMessageBox.warning(self, "Hapus", "Pilih tugas yang ingin dihapus.")
            return
        id_item = self.table.item(selected, 0)
        judul_item = self.table.item(selected, 2)
        if id_item and judul_item:
            judul_text = judul_item.text()
            confirm = QMessageBox.question(self, "Konfirmasi", f"Yakin ingin menghapus tugas '{judul_text}'?", QMessageBox.Yes | QMessageBox.No)
            if confirm == QMessageBox.Yes:
                self.cursor.execute("DELETE FROM tugas WHERE id = ?", (id_item.text(),))
                self.conn.commit()
                self.load_data()
                QMessageBox.information(self, "Dihapus", f"Tugas '{judul_text}' berhasil dihapus.")
    def export_csv(self):
        path, _ = QFileDialog.getSaveFileName(self, "Simpan CSV", "", "CSV Files (*.csv)")
        if path:
            self.cursor.execute("SELECT id, matkul, judul, deadline, status, catatan FROM tugas")
            rows = self.cursor.fetchall()
            with open(path, 'w', newline='', encoding='utf-8') as f:
                writer = csv.writer(f)
                writer.writerow(["ID", "Mata Kuliah", "Judul Tugas", "Deadline", "Status", "Catatan"])
                writer.writerows(rows)
            QMessageBox.information(self, "Berhasil", "Data berhasil diekspor ke CSV.")
    def search_task(self, text):
        self.load_data()
        self.filter_combo.currentText() if self.filter_combo.currentText() != "Tampilkan Semua" else None,
        text
    )
    def filter_status(self, status):
        self.load_data()
        status if status != "Tampilkan Semua" else None,
        self.search_input.text()
    )
if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = StuDate()
    window.show()
    sys.exit(app.exec_())

```