

PROJECT UAS
PEMROGRAMAN VISUAL
TASK MANAGER



Disusun Oleh:
LALU RESTU BAGUS ANUGRAH (F1D022059)

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MATARAM

2025

A. Deskripsi Aplikasi

Aplikasi **Task Manager** adalah sebuah solusi digital sederhana namun efisien yang dirancang untuk membantu pengguna dalam mencatat, memantau, dan mengelola aktivitas atau tugas-tugas harian mereka. Aplikasi ini memungkinkan pengguna untuk mencatat informasi penting terkait setiap tugas, seperti *judul tugas*, *kategori aktivitas*, *tanggal pelaksanaan*, *status kemajuan*, serta *catatan tambahan* yang relevan. Dengan fitur ini, pengguna dapat merencanakan aktivitas secara lebih tertata dan menghindari keterlambatan atau kelupaan terhadap hal-hal yang perlu dikerjakan.

Seluruh data yang dimasukkan oleh pengguna disimpan secara lokal menggunakan database **SQLite**, sehingga pengguna tidak membutuhkan koneksi internet untuk mengoperasikan aplikasi. Selain itu, aplikasi juga menyediakan fitur ekspor ke dalam format **CSV**, yang memungkinkan pengguna menyimpan atau mencetak daftar tugas mereka untuk keperluan dokumentasi atau laporan harian.

Kegunaan aplikasi ini bersifat fleksibel dan universal. Dalam konteks pribadi, aplikasi dapat dimanfaatkan untuk mencatat kegiatan seperti jadwal belajar, daftar belanja, hingga pengingat deadline tugas. Di lingkungan akademik, aplikasi ini dapat menjadi alat bantu mahasiswa atau dosen dalam mengelola tugas perkuliahan atau rencana penelitian. Sedangkan dalam dunia kerja, aplikasi ini bisa dimodifikasi sebagai sistem manajemen tugas sederhana untuk proyek individu maupun tim.

Dengan antarmuka yang intuitif, layout yang rapi, serta fitur yang sesuai kebutuhan sehari-hari, Task Manager diharapkan dapat mendorong produktivitas penggunanya. Proyek ini juga mencerminkan penerapan nyata dari berbagai konsep yang telah dipelajari dalam mata kuliah *Pemrograman Visual*, mulai dari penggunaan **PyQt5**, **desain UI dengan Qt Designer**, **manajemen basis data SQLite**, hingga integrasi fungsi ekspor menggunakan **pandas**.

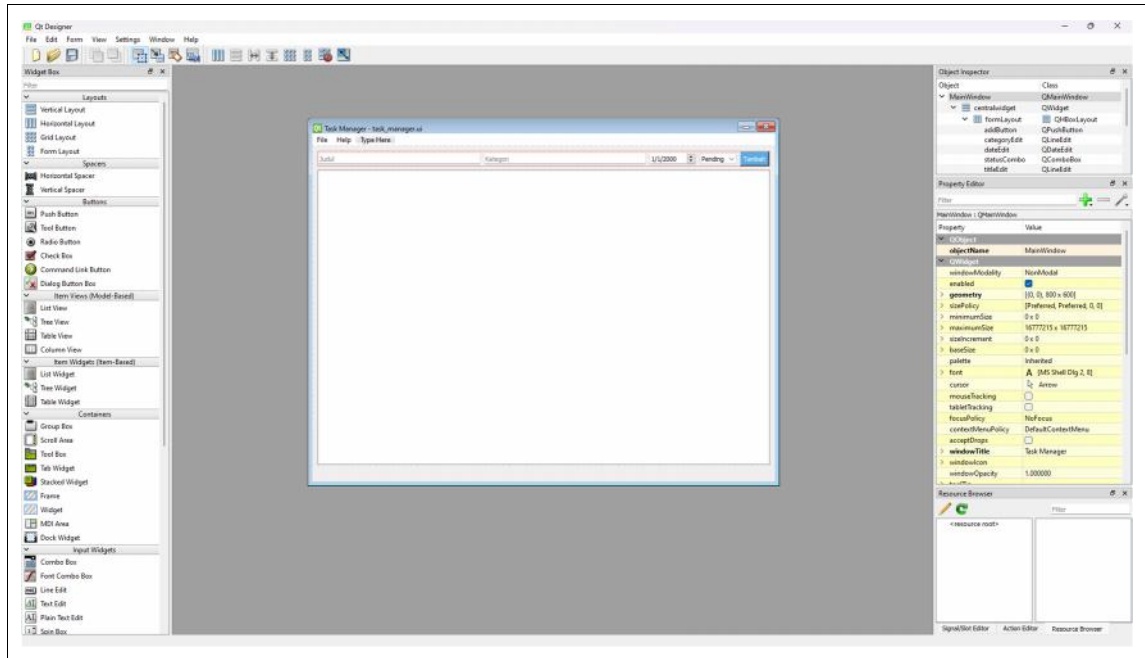
B. Langkah Pengembangan Aplikasi

1. Perencanaan Ide Fitur

Langkah awal dalam pengembangan aplikasi adalah merumuskan ide yang tepat dan sesuai dengan ketentuan tugas. Dalam proyek ini, dipilih tema **Task Manager**, sebuah aplikasi yang memungkinkan pengguna mencatat dan mengatur aktivitas atau tugas harian mereka. Pemilihan ini didasarkan pada pertimbangan bahwa manajemen waktu dan tugas

merupakan kebutuhan universal, baik dalam konteks akademik, profesional, maupun kehidupan pribadi. Selain itu, ide ini juga memberikan fleksibilitas dalam desain serta peluang untuk menerapkan berbagai fitur interaktif dan komponen dari PyQt5 secara kreatif dan kontekstual.

2. Desain Antarmuka denn Qt Designer



Setelah menentukan ide, tahap selanjutnya adalah merancang antarmuka pengguna (UI) menggunakan Qt Designer. Desain dimulai dengan pembuatan *form input* yang berisi kolom isian seperti judul tugas, kategori aktivitas, tanggal, dan status. Komponen-komponen ini dimasukkan menggunakan QLineEdit, QDateEdit, dan QComboBox. Selanjutnya, elemen utama visual yaitu QTableWidgetItem ditambahkan sebagai media untuk menampilkan daftar tugas yang telah disimpan. Pengaturan tata letak dilakukan dengan menggunakan QVBoxLayout dan QHBoxLayout agar tampilan tertata rapi dan responsif. Selain itu, QMenuBar ditambahkan untuk menyediakan opsi menu seperti "File" dan "Help", sementara QStatusBar digunakan untuk menampilkan informasi identitas mahasiswa di bagian bawah aplikasi secara permanen.

3. Pembuatan Database SQLite

Untuk menyimpan data tugas secara lokal, digunakan sistem basis data SQLite yang ringan dan terintegrasi langsung dalam aplikasi Python tanpa perlu server eksternal. Basis data dirancang dengan satu tabel utama bernama tasks, yang memiliki enam kolom: id,

title, category, date, status, dan notes. Kolom id digunakan sebagai primary key yang di-*autoincrement*, sedangkan lima kolom lainnya menyimpan informasi tugas yang diinputkan oleh pengguna. Struktur ini memastikan bahwa data yang disimpan terorganisasi dengan baik dan mudah dikelola.

4. Integrasi UI dengan Logika Program

Setelah antarmuka selesai, tahap berikutnya adalah menyatukan tampilan dengan logika program menggunakan Python dan PyQt5. File .ui dimuat melalui `uic.loadUi`, dan setiap komponen UI dihubungkan ke fungsi-fungsi Python melalui *signals and slots*. Tombol “Tambah” diatur untuk menjalankan fungsi `addTask()`, yang akan menyimpan data ke dalam SQLite. Menu “Export CSV” diatur untuk menjalankan `exportCSV()`, yaitu proses ekspor data ke file .csv menggunakan library `pandas`. Selain itu, status bar secara otomatis menampilkan nama dan NIM mahasiswa, memberikan kesan profesional dan personal pada aplikasi.

5. Uji Coba dan Debugging

Langkah terakhir adalah melakukan pengujian aplikasi untuk memastikan semua fitur berjalan sesuai rencana. Proses ini melibatkan menjalankan aplikasi secara berulang, mengisi data dengan berbagai skenario, dan memperhatikan apakah ada error atau perilaku yang tidak diinginkan. Aplikasi juga diuji untuk memastikan stabilitasnya dalam lingkungan sistem operasi berbeda, serta ketahanannya terhadap kesalahan input pengguna. Hasil akhirnya adalah aplikasi yang berjalan stabil, responsif, dan siap digunakan oleh pengguna akhir.

C. Penjelasan Fungsi-fungsi Utama

1. `createDB()`

`createDB()` Fungsi ini bertugas untuk memastikan bahwa database SQLite yang dibutuhkan oleh aplikasi tersedia dan telah memiliki struktur tabel yang sesuai. Ketika aplikasi pertama kali dijalankan, `createDB()` akan membuat file database bernama `tasks.db` (jika belum ada) dan membuat tabel `tasks` dengan kolom-kolom penting seperti `title`, `category`, `date`, `status`, dan `notes`. Fungsi ini merupakan fondasi dari integrasi basis data karena menjamin bahwa data yang dimasukkan pengguna dapat disimpan secara permanen.

2. `loadData()`

mengambil seluruh data tugas dari database dan menampilkannya di tampilan QTableWidget dalam antarmuka pengguna. Dengan menggunakan perulangan, setiap baris data dari database diisi ke dalam baris-baris tabel pada UI, sehingga pengguna dapat melihat daftar tugas yang sudah pernah dimasukkan secara langsung. Fungsi ini biasanya dipanggil saat aplikasi dibuka pertama kali maupun setelah pengguna menambahkan data baru.

3. addTask

Fungsi ini bertanggung jawab untuk mengambil data yang dimasukkan pengguna dari form input dan menyimpannya ke dalam database SQLite. Pertama, fungsi ini akan memvalidasi bahwa kolom “Judul” dan “Kategori” telah diisi. Jika valid, data akan dimasukkan ke tabel tasks di database, kemudian form akan dibersihkan, dan tampilan tabel diperbarui. Fitur ini menjadi jantung aplikasi karena menghubungkan input pengguna langsung ke penyimpanan permanen.

4. exportCSV()

exportCSV() Fungsi ini memberikan kemampuan kepada pengguna untuk menyimpan data tugas mereka ke dalam file CSV, yang dapat digunakan sebagai dokumentasi atau referensi luar aplikasi. Setelah pengguna memilih lokasi penyimpanan, seluruh isi tabel tasks akan diambil dari database, diubah menjadi DataFrame menggunakan pustaka pandas, lalu disimpan ke file .csv. Fungsi ini membantu pengguna membawa data mereka keluar dari aplikasi dalam format yang umum dan mudah dibaca.

5. initUI()

dijalankan saat aplikasi dimulai dan bertanggung jawab untuk menyambungkan elemen-elemen antarmuka pengguna dengan fungsi-fungsi logika program. Di sini dilakukan pengaturan signal-slot, seperti menghubungkan tombol “Tambah” dengan fungsi addTask(), dan menu “Export” ke exportCSV(). Selain itu, fungsi ini juga mengatur teks awal di statusbar, menentukan jumlah kolom pada tabel, serta mengatur header tabel. Secara keseluruhan, initUI() bertindak sebagai pengatur awal yang mempersiapkan antarmuka agar dapat berfungsi dengan benar.

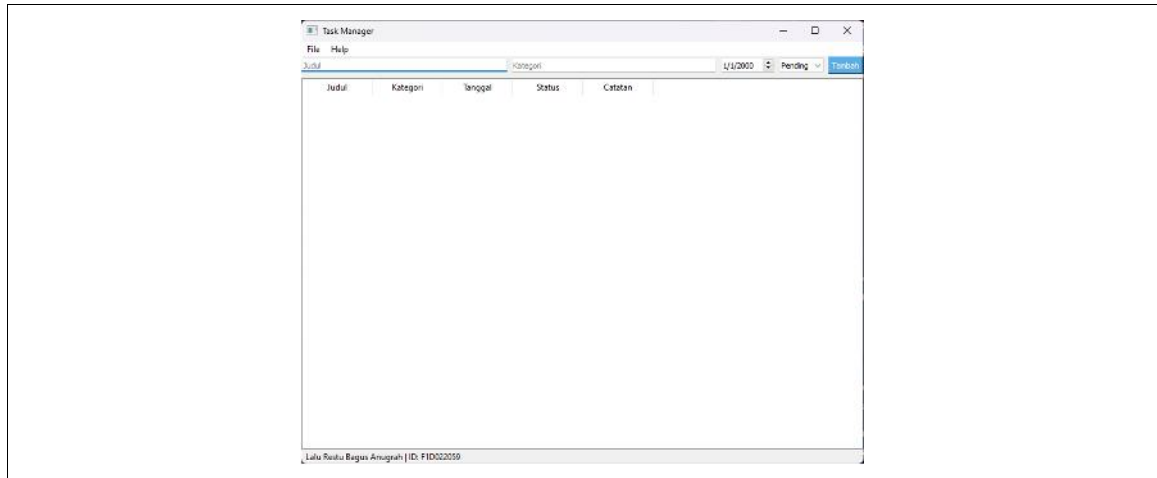
6. showAbout()

Fungsi ini sederhana namun penting sebagai bagian dari elemen informasi dalam aplikasi. Saat pengguna memilih menu “Help About”, fungsi ini akan menampilkan

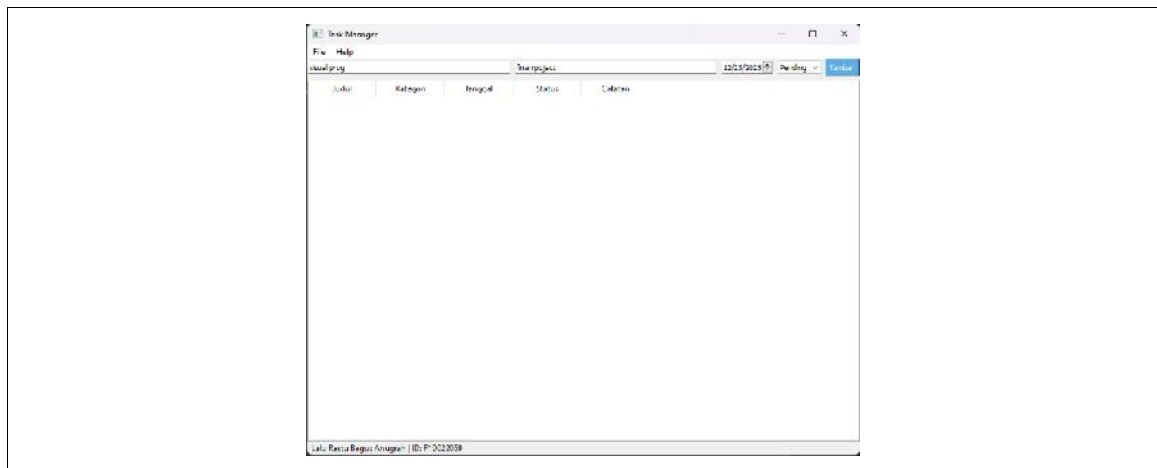
kotak dialog pop-up yang berisi informasi singkat tentang aplikasi, termasuk nama pembuat dan ID mahasiswa. Tujuannya adalah untuk memberikan konteks kepada pengguna tentang siapa pengembang aplikasi dan tujuan umum dari proyek ini.

D. Interface

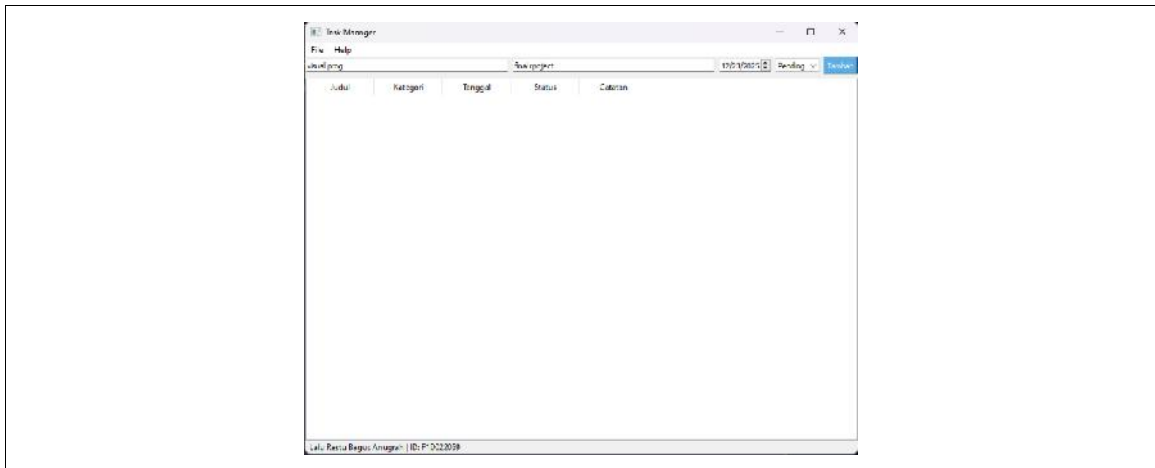
1. Tampilan Utama Aplikasi



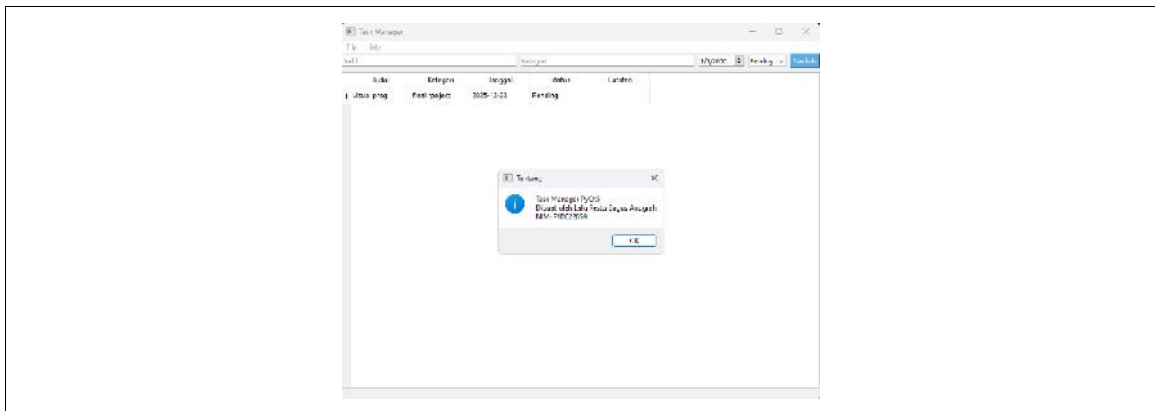
2. Form Input Tugas



3. Tabel data terisi



4. Popup “About”



5. Hasil CSV folder

