

**LAPORAN UJIAN TENGAH SEMESTER
PEMOGRAMAN VISUAL
SEMESTER GENAP 2025**

Manajemen Film dan Buku



**Disusun Oleh :
Ida Ayu Vinaya Anindya (F1D022124)**

Deskripsi Singkat:

Aplikasi Movie and Book Management merupakan aplikasi yang dirancang dengan memanfaatkan PyQt Designer untuk desain antarmuka, lalu dikembangkan lebih lanjut menggunakan bahasa pemrograman Python. Aplikasi ini dirancang untuk membantu pengguna mengelola daftar buku dan film yang telah rilis di internet ataupun bioskop. Dalam aplikasi ini, pengguna berperan sebagai administrator atau pengguna biasa yang dapat menambahkan entri baru dengan mengisi informasi seperti judul, tahun rilis, nama penulis atau sutradara, sinopsis, serta memberikan penilaian atau *rating*. Data yang telah dimasukkan kemudian bisa dilihat oleh pengguna lain, dan tersedia fitur pencarian serta penyaringan berdasarkan waktu atau jenis media.

Langkah Pembuatan Aplikasi :

1. Desain Antar Muka

Bagian desain antarmuka ini dirancang menggunakan PyQt Design yang kemudian dikonversi dengan bantuan pyuic5. Pada bagian, digunakan beberapa jenis widget seperti horizontal layout, plain text, text edit, combo box, check box dan table widget. Setiap elemen ini memiliki fungsi masing-masing dalam desain antarmuka, untuk lebih jelasnya, sebagai berikut :

a. Horizontal Layout

Bagian horizontal layout digunakan untuk mengatur bagian widget dalam bentuk horizontal. Dalam desain aplikasi ini, horizontal layout digunakan untuk membagi dua bagian widget yaitu bagian untuk menampilkan dengan menggunakan table widget dan bagian form untuk memasukan informasi terkait dengan film atau buku.

b. Plain Text

Bagian plain text digunakan untuk menampung teks dalam jumlah banyak. Dalam aplikasi ini, plain text digunakan dalam memberikan informasi pada form mengenai informasi apa saja yang harus dimasukkan terkait film atau buku (judul, tahun rilis, penulis atau sutradara, sinopsi, serta penilaian dari film atau buku tersebut).

c. Text Edit

Bagian text edit digunakan untuk memasukkan informasi dalam bentuk teks pendek ataupun panjang, text edit dapat digunakan sebagai input text. Dalam aplikasi ini text edit dipakai untuk memasukkan informasi mengenai film atau buku pada bagian formnya. Selain itu text edit juga berperan sebagai search untuk memfilter kata berdasarkan judul.

d. Combo Box

Bagian combo box digunakan untuk memilih satu opsi dari beberapa pilihan yang tersedia. Pada aplikasi ini combo box digunakan dalam filtering, yaitu pengguna diminta memilih data yang dimasukkan tersebut contohnya seperti, film atau buku yang ada pada bagian form. Adapun, filtering dari form yang telah dimasukkan, yaitu pengguna diminta untuk memilih tampilan berdasarkan terbaru, film atau buku.

e. Check Box

Bagian check box digunakan untuk memilih salah satu opsi dari beberapa pilihan. Check box dan combo box memiliki fungsi yang sama

namun pada check box bentuknya dicentang sedangkan pada combo box bentuknya seperti *dropdown*. Pada aplikasi ini digunakan dalam form pada bagian penilaian atau rating.

f. Push Button

Bagian push button digunakan untuk mengklik tombol yang nantinya data yang dimasukkan akan terkirim dan tersampaikan pada bagian table widget. Pada aplikasi ini digunakan dalam menyimpan dan menghapus kembali form seperti awal.

g. Table Widget

Bagian table widget digunakan untuk menampilkan data yang telah dimasukkan dalam bentuk tabel. Pada aplikasi ini digunakan dalam menampilkan data hasil form.

2. Pengembangan Bahasa Python

Dalam pengembangan dengan bahasa python, desain antarmuka sebelumnya akan ditambahkan beberapa fungsi untuk menjalankan program tersebut misalnya pada bagian push button, dan table widgetnya. Pada aplikasi ini bahasa python digunakan untuk menunjang beberapa fitur penting dalam manajemen film dan buku secara praktis. Beberapa fitur yang ditambahkan antara lain adalah tombol "Add" untuk mengosongkan form *input* agar siap digunakan kembali, dan tombol "Save" yang digunakan untuk menyimpan data input ke dalam tampilan tabel. Selain itu, terdapat juga fitur pencarian melalui kolom LineEdit dan fitur *fitering* menggunakan ComboBox untuk mempermudah pengguna dalam melihat data berdasarkan kategori tertentu seperti jenis media (Film atau Buku) dan data terbaru.

3. Interaksi Komponen

Komponen dalam aplikasi ini berinteraksi antara satu dengan lainnya melalui sistem slot dan signal dari PyQt. Contohnya, jika pengguna memilih jenis data pada QComboBox, maka proses *filter_data()* akan secara otomatis memproses dan menampilkan data relevan pada tabel. Demikian pula jika pengguna mengetikkan kata kunci pada kolom pencarian, sistem akan merespons langsung melalui proses *search_data()* yang mengfilter data berdasarkan teks yang ditekan. Interaksi lainnya adalah ketika tombol "Save" diklik, semua input dari komponen seperti QTextEdit, QComboBox, dan QCheckBox akan dikumpulkan, diproses, dan ditampilkan dalam tabel sebagai satu entri baru.

4. Penyimpanan Sementara

Penyimpanan data dalam aplikasi ini dilaksanakan secara sementara melalui struktur list Python, dimana semua data yang dimasukkan akhirnya disimpan dalam bentuk dictionary ke dalam list *self.data_list*. Masing-masing dictionary mengandung informasi lengkap seperti judul, jenis, penulis atau sutradara, tahun, dan rating. Sementara belum ada penggunaan penyimpanan permanen seperti penggunaan database atau file luar, metode ini memungkinkan data tetap terkelola dengan baik selama aplikasi dibuka, serta memungkinkan fleksibilitas di pengolahan serta manipulasi data secara langsung.

Penjelasan Fungsi Aplikasi

1. Class Ui_MainWindow

Kelas `Ui_MainWindow` merupakan inti dari antarmuka aplikasi yang menyimpan semua komponen logika dan visual utama. Di dalam kelas ini terdapat `setupUi()` yang digunakan untuk mengatur tata letak (layout) elemen-elemen GUI seperti tabel, tombol, input form, combo box, dll. Kelas ini juga mengelola koneksi antara aksi pengguna (misalnya permainan tombol klik atau perubahan teks) dengan fungsi-fungsi tertentu melalui sistem signal dan slot PyQt. Semua fitur dan interaksi antarkomponen diatur dan dikelola di dalam kelas ini.

2. Dev Save_Data

Fungsi `save_data()` yang berfungsi untuk menyimpan data-data yang telah dimasukkan pengguna melalui form input. Fungsi ini akan mendapatkan nilai dari beberapa komponen seperti judul (`QTextEdit`), jenis (`QComboBox`), penulis/sutradara, tahun, dan rating yang dipilih menggunakan `QCheckBox`. Setelah semua koleksi, fungsi ini akan menyimpannya ke dalam list `self.data_list` dalam bentuk dictionary, kemudian memanggil fungsi `show_data()` untuk menampilkannya ke dalam tabel. Fungsi ini juga akan memanggil `add_data()` untuk membersihkan form jika data berhasil disimpan.

3. Dev Add_Data

Fungsi `add_data()` dipakai untuk membersihkan atau mengosongkan semua form input agar siap digunakan lagi untuk memasukkan entri baru. Fungsi ini akan membersihkan konten dari `QTextEdit`, mengembalikan `QComboBox` ke posisi default, dan membuat semua `QCheckBox` tidak dipilih. Fitur ini sangatlah berguna untuk memudahkan pengguna ketika ingin menambahkan entri baru dengan cepat tanpa perlu melakukan penghapusan input sebelumnya secara manual.

4. Dev Show_Data

Fungsi `show_data()` memiliki tugas utama untuk menampilkan seluruh data yang tersimpan ke dalam `QTableWidget`. Fungsi ini akan terlebih dahulu mengosongkan isi tabel, kemudian mengisi ulang dengan data yang ada di list `self.data_list`. Setiap entri akan dimasukkan ke dalam baris tabel secara terstruktur berdasarkan urutan dan kolom yang telah ditentukan, seperti judul, tipe, penulis/sutradara, tahun, dan rating.

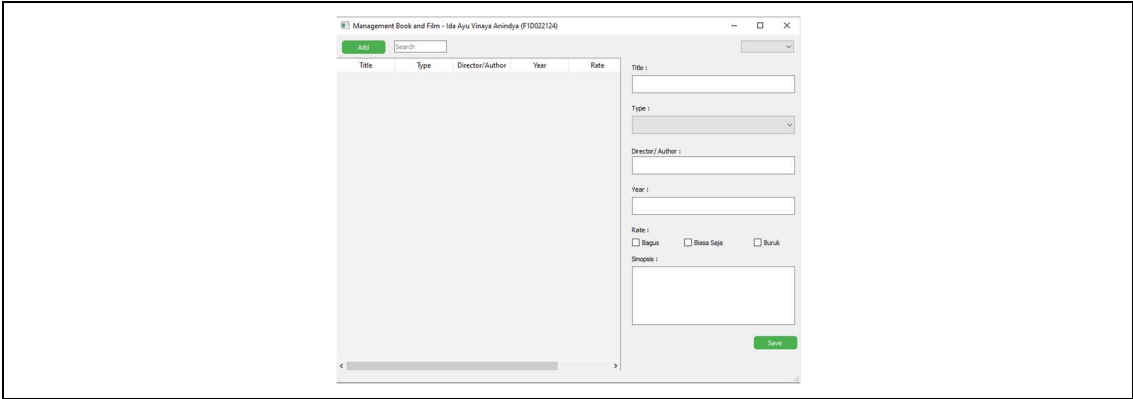
5. Dev Search_Data

Fungsi `search_data()` digunakan untuk penyaringan dan penampilkannya data berdasarkan kata kunci yang diperlukan pengguna memasukkannya ke dalam ruang pencarian. Fungsi ini akan membandingkan input teks dengan segmen judul dari tiap entri pada `self.data_list`, dan hanya akan menampilkan data yang menyimpan kata kunci tersebut. Proses ini dilakukan real-time, jadi pengguna dapat melihat hasil pencarian secara langsung ketika mengetik.

6. Dev Filter_Data

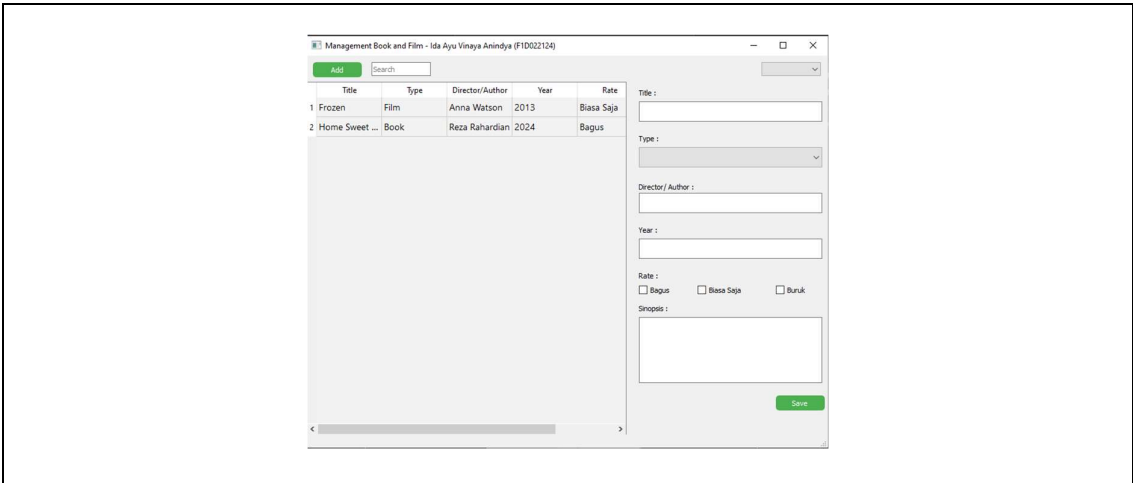
Fungsi `filter_data()` digunakan untuk menyaring tampilan data berdasarkan pilihan yang dipilih pada `QComboBox` filter. Pilihan yang tersedia antara lain "Terbaru", "Film's", dan "Book's". Jika pengguna memilih "Terbaru", maka data akan diurutkan berdasarkan tahun secara menurun. Jika memilih "Film's" atau "Book's", maka hanya data dengan tipe tersebut yang akan ditampilkan. Fungsi ini memudahkan pengguna dalam melihat data secara lebih spesifik sesuai kebutuhan.

Desain Interface



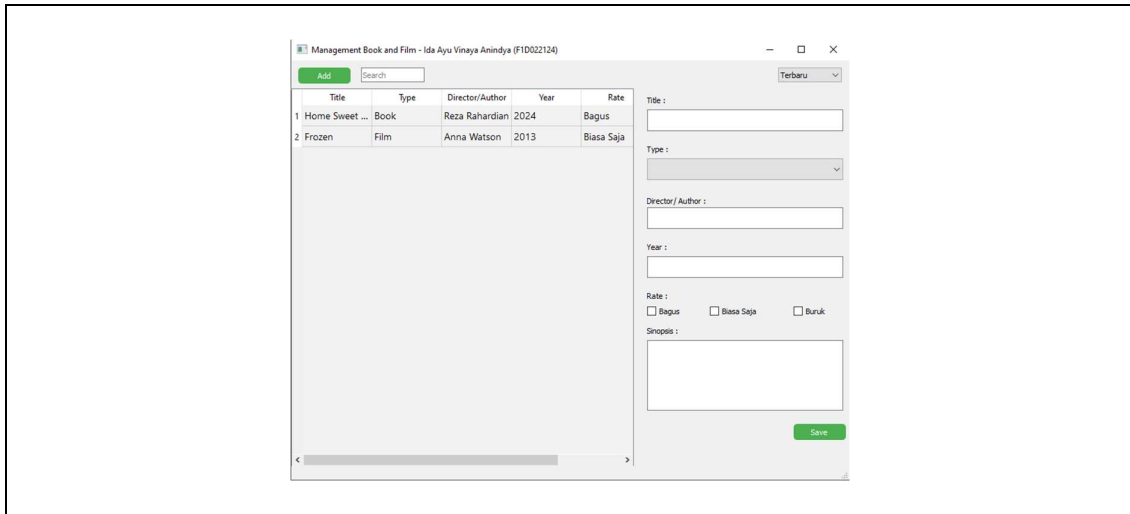
Gambar 1

Pada Gambar 1 merupakan tampilan utama dari aplikasi manajemen film dan buku. Gambar 1 diatas menampilkan bagian form di sebelah kanan dan tabel widget untuk menampilkan data input terkait film dan buku.



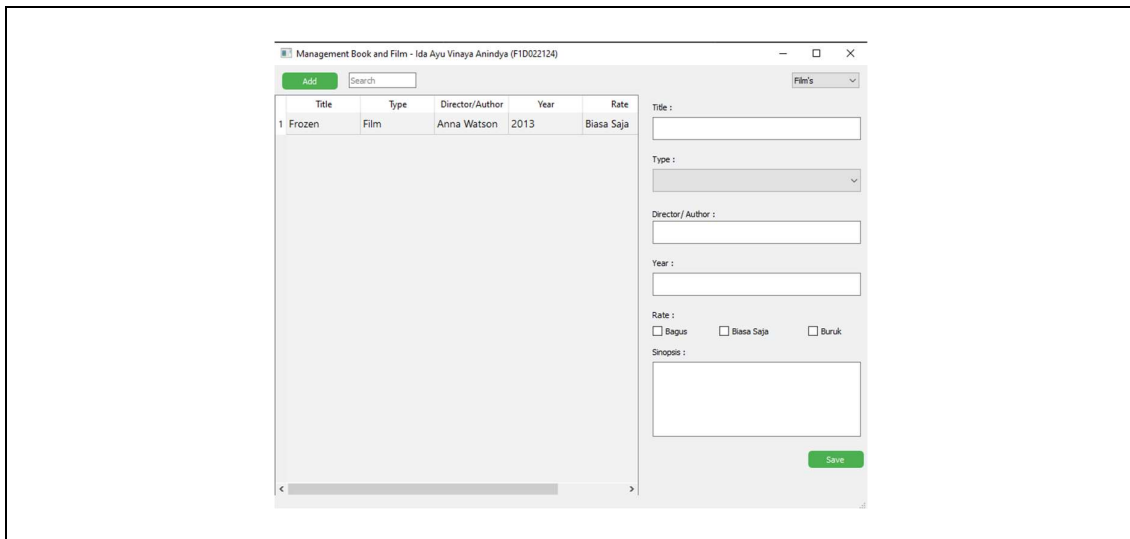
Gambar 2

Pada Gambar 2 memperlihatkan tampilan saat data yang telah diinputkan berhasil ditampilkan dalam tabel widget di sisi kiri aplikasi. Untuk mengembalikan form input ke kondisi awal atau kosong, pengguna dapat menekan tombol "Add" yang berfungsi untuk mereset seluruh isian form.



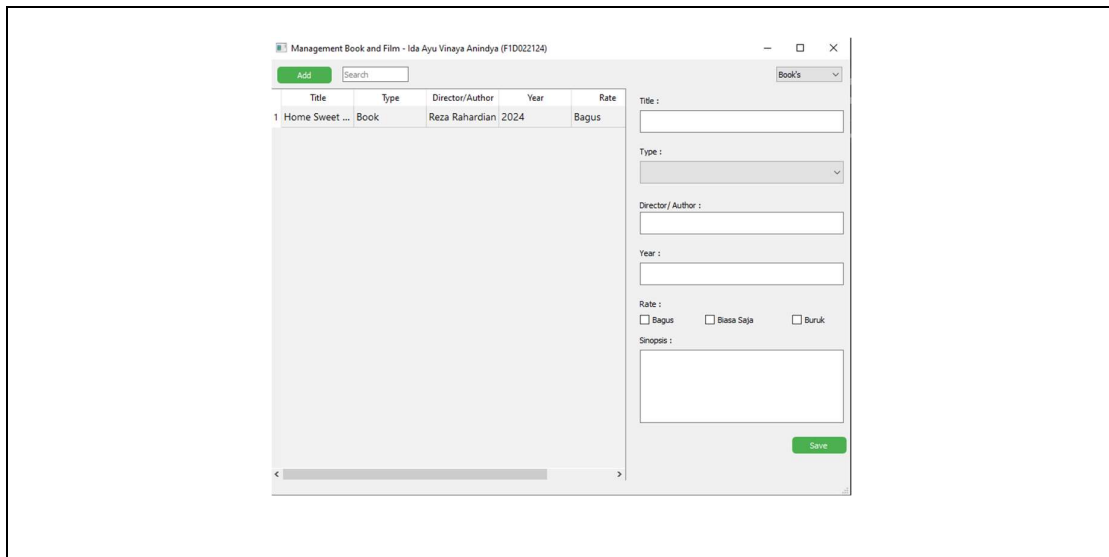
Gambar 3

Pada Gambar 3 memperlihatkan tampilan saat data yang telah diinputkan dilakukan filtering berdasarkan data terbaru dari film dan buku yang baru saja diinputkan pada bagian sisi kanan dari widget.



Gambar 4

Pada Gambar 4 memperlihatkan tampilan saat data yang telah diinputkan dilakukan filtering berdasarkan data film saja dari film dan buku yang baru saja diinputkan pada bagian sisi kanan dari widget.



Gambar 5

Pada Gambar 5 memperlihatkan tampilan saat data yang telah diinputkan dilakukan filtering berdasarkan data buku saja dari film dan buku yang baru saja diinputkan pada bagian sisi kanan dari widget.

Link Github : <https://github.com/F1D022124/pv25-miniproject-BookorMovieManager.git>