

UJIAN AKHIR SEMESTER

Mata Kuliah

Pemrograman Berorientasi Objek

Oleh

Azkiya Azimi

24091397046

2024B



PROGRAM STUDI MANAJEMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2025

DAFTAR ISI

HALAMAN JUDUL	1
DAFTAR ISI.....	2
BAB I PENDAHULUAN.....	3
1.1 Latar Belakang	3
1.2 Tujuan Proyek	3
1.3 Ruang Lingkup	3
BAB II PERANCANGAN APLIKASI	4
2.1 Deskripsi Aplikasi	4
2.2 Fitur Utama.....	4
2.3 Teknologi dan Library yang di Gunakan	4
BAB III KONSEP OOP YANG DITERAPKAN.....	6
3.1 Encapsulation	6
3.2 Inheritance	6
3.3 Polymorphism	7
BAB IV DIAGRAM CLASS.....	8
4.1 Gambar dan Penjelasan Diagram Class	8
BAB V ALUR APLIKASI	10
5.1 Alur Umum	10
5.2 Alur Tambah Tugas.....	10
5.3 Alur Reminder dan Export	10
BAB VI TANTANGAN YANG DIHADAPI	11
6.1 Tantangan Teknis	11
6.2 Solusi yang DIterapkan	11
BAB VII IMPLEMENTASI.....	12
7.2 Screenshot Aplikasi	12
BAB VIII PENUTUP	14
8.2 Kesimpulan.....	14
8.2 Saran	14

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital saat ini, pengelolaan jadwal menjadi kebutuhan penting bagi mahasiswa dan profesional. Aplikasi JadwalKu Pro dibuat untuk memenuhi tugas UAS PBO, dengan fokus pada penerapan prinsip OOP menggunakan Python dan Tkinter. Proyek ini terinspirasi dari kebutuhan sehari-hari untuk mengatur tugas dengan fitur kreatif seperti reminder dan quote motivasi.

Aplikasi JadwalKu Pro dibuat sebagai proyek Ujian Akhir Semester (UAS) mata kuliah Pemrograman Berorientasi Objek (PBO). Aplikasi ini dikembangkan menggunakan bahasa pemrograman Python dengan antarmuka grafis (GUI) Tkinter. Proyek ini bertujuan untuk menerapkan konsep-konsep Object Oriented Programming (OOP) dalam sebuah aplikasi nyata. Selain itu, aplikasi ini dilengkapi dengan fitur tambahan seperti pengingat (reminder) dan kutipan motivasi (quote) untuk meningkatkan semangat dan produktivitas pengguna dalam menjalani aktivitas sehari-hari.

1.2 Tujuan Proyek

1. mengembangkan aplikasi berbasis oop menggunakan bahasa pemrograman Python.
2. menerapkan encapsulation, inheritance, dan polymorphism dalam proses pengembangan aplikasi.
3. Membuat dokumentasi lengkap termasuk diagram class dan alur aplikasi.

1.3 Ruang Lingkup

Ruang lingkup proyek ini dibatasi pada pengembangan aplikasi desktop sederhana dengan menggunakan GUI Tkinter. Aplikasi ini tidak menggunakan atau terhubung dengan database eksternal, sehingga penyimpanan data dilakukan secara sederhana sesuai kebutuhan aplikasi. Fokus utama aplikasi adalah pada fitur pengelolaan jadwal dan tugas, baik tugas harian maupun tugas yang bersifat berulang. Proyek ini lebih menekankan pada penerapan konsep OOP dan pembuatan antarmuka pengguna yang mudah digunakan.

BAB II

PERANCANGAN APLIKASI

2.1 Deskripsi Aplikasi

JadwalKu Pro merupakan aplikasi pengelola jadwal berbasis desktop yang dirancang untuk membantu pengguna dalam mengatur dan memantau tugas sehari-hari. Melalui aplikasi ini, pengguna dapat menambahkan tugas baru, mengedit tugas yang sudah ada, serta melihat perkembangan penyelesaian tugas secara visual.

Aplikasi ini menggunakan Tkinter sebagai antarmuka pengguna (Graphical User Interface) sehingga mudah digunakan dan dipahami. Selain itu, fitur pengingat (reminder) dibuat menggunakan konsep threading, yang memungkinkan aplikasi menampilkan notifikasi secara real-time tanpa mengganggu jalannya program utama. Dengan adanya fitur ini, pengguna dapat tetap mendapatkan pengingat tugas tepat waktu saat aplikasi sedang berjalan.

2.2 Fitur Utama

- 1) Tambah tugas dengan kategori, prioritas, dan tipe (harian/berulang).
Pengguna dapat menambahkan tugas baru dengan informasi seperti kategori tugas, tingkat prioritas, serta tipe tugas, baik tugas harian maupun tugas berulang.
- 2) Reminder pop-up dengan quote motivasi.
Aplikasi menyediakan fitur pengingat berupa pop-up yang akan muncul sesuai waktu yang telah ditentukan. Setiap reminder dilengkapi dengan quote motivasi untuk memberikan semangat kepada pengguna.
- 3) Tracking progress dengan progress bar.
Aplikasi menampilkan progress bar yang menunjukkan sejauh mana tugas telah diselesaikan, sehingga pengguna dapat memantau perkembangan aktivitasnya dengan lebih jelas.
- 4) Export data ke CSV.
Data tugas yang telah dimasukkan dapat dieksport ke dalam format CSV. Fitur ini memudahkan pengguna untuk menyimpan, mencetak, atau mengolah data tugas menggunakan aplikasi lain seperti spreadsheet.
- 5) Integrasi kalender sederhana untuk pemilihan tanggal.
Aplikasi dilengkapi dengan kalender sederhana yang digunakan untuk memilih tanggal tugas. Fitur ini membantu pengguna dalam menentukan jadwal dengan lebih akurat dan praktis.

2.3 Teknologi yang Digunakan

- 1) Bahasa Pemrograman
Bahasa pemrograman yang digunakan adalah Python versi 3.x, karena bersifat fleksibel, mudah dipahami, dan sangat mendukung konsep Object Oriented Programming.
- 2) Library dan Modul
Beberapa library dan modul Python yang digunakan antara lain:
 - Tkinter untuk membuat antarmuka pengguna (GUI)
 - tkcalendar untuk fitur pemilihan tanggal

- datetime untuk pengelolaan waktu dan tanggal
 - threading untuk menjalankan reminder secara real-time
 - csv untuk proses ekspor data
 - random untuk menampilkan quote motivasi secara acak
- 3) Tools Pendukung
Git digunakan sebagai tools pendukung untuk mengelola versi kode program dan menyimpan proyek dalam repositori, sehingga memudahkan proses pengembangan dan dokumentasi.

BAB III

KONSEP OOP YANG DITERAPKAN

3.1 Encapsulation

Encapsulation merupakan konsep OOP yang digunakan untuk membatasi akses langsung ke data dalam sebuah objek. Pada aplikasi JadwalKu Pro, konsep encapsulation diterapkan dengan cara membuat atribut bersifat *private* yang ditandai dengan penggunaan tanda garis bawah (underscore) di awal nama atribut.

Sebagai contoh, pada kelas Task, atribut seperti `_name`, `_time`, dan `_priority` tidak dapat diakses atau diubah secara langsung dari luar kelas.

```
class Task:  
    def __init__(self, name, time_str, description, category, priority):  
        self._name = name  
        self._time = datetime.strptime(time_str, "%Y-%m-%d %H:%M")  
        self._description = description  
        self._category = category  
        self._priority = priority  
        self._completed = False  
        self._motivational_quote = self._generate_quote()  
        self._reminder_shown = False
```

Untuk mengakses atau mengubah nilai atribut tersebut, digunakan method getter dan setter, seperti `get_name()` dan `set_name()`.

```
def get_name(self): return self._name  
def get_time(self): return self._time
```

Dengan cara ini, data menjadi lebih aman dan terkontrol, serta mengurangi kemungkinan kesalahan dalam pengelolaan data tugas.

3.2 Inheritance

Inheritance atau pewarisan digunakan untuk membuat kelas baru berdasarkan kelas yang sudah ada. Dalam aplikasi JadwalKu Pro, terdapat kelas induk bernama Task yang menyimpan atribut dan method umum yang digunakan oleh semua jenis tugas.

Kelas DailyTask dan RecurringTask merupakan turunan dari kelas Task. Dengan menggunakan inheritance, kedua kelas tersebut dapat langsung menggunakan atribut umum seperti `_name`, `_time` dan `_priority` tanpa perlu menuliskan ulang kode yang sama.

```
class DailyTask(Task):  
    def __init__(self, name, time_str, description, category, priority,  
                 reminder_minutes=10):  
        super().__init__(name, time_str, description, category, priority)
```

contoh lain pada class RecurringTask:

```
class RecurringTask(Task):
    def __init__(self, name, time_str, description, category, priority,
recurrence_days):
        super().__init__(name, time_str, description, category, priority)
```

Hal ini membuat struktur program menjadi lebih rapi, efisien, dan mudah dikembangkan di kemudian hari.

3.3 Polymorphism

Polymorphism memungkinkan method dengan nama yang sama memiliki perilaku yang berbeda pada setiap kelas. Konsep ini diterapkan pada method `display_info()` dalam aplikasi JadwalKu Pro.

```
def display_info(self):
    return super().display_info() + f"\nReminder: {self._reminder_minutes}
menit sebelumnya"
```

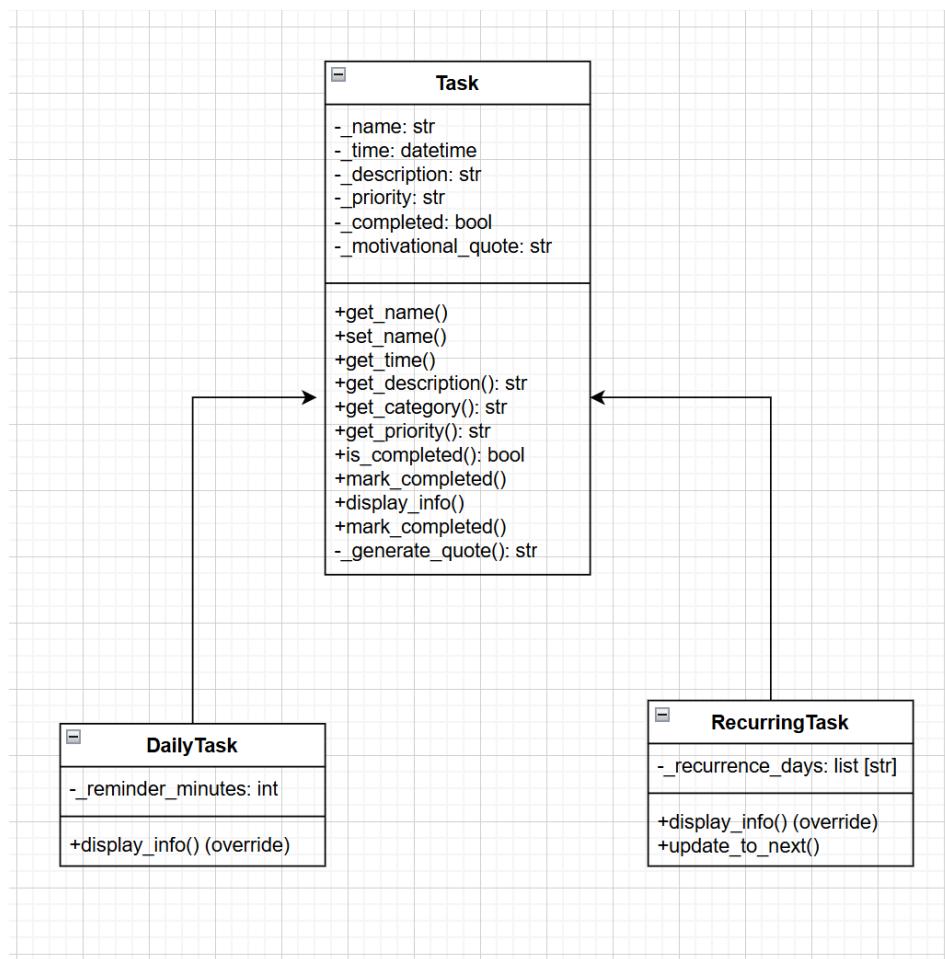
Method `display_info()` didefinisikan pada kelas `Task`, kemudian di override pada kelas `DailyTask` dan `RecurringTask`. Pada kelas `DailyTask`, method ini menampilkan informasi tambahan berupa waktu reminder.

Sementara itu, pada kelas `RecurringTask`, method `display_info()` menampilkan informasi tentang hari pengulangan tugas. Dengan penerapan polymorphism, aplikasi dapat menampilkan informasi tugas yang berbeda sesuai dengan jenis tugasnya tanpa perlu membuat method dengan nama yang berbeda.

```
def display_info(self):
    return super().display_info() + f"\nBerulang: {',
'.join(self._recurrence_days)}"
```

BAB IV

DIAGRAM CLASS



4.1 penjelasan diagram

1) Kelas Task (superclass/kelas induk)

Kelas Task merupakan kelas utama yang menjadi dasar bagi semua jenis tugas dalam aplikasi. Kelas ini menyimpan atribut dan method umum yang dimiliki oleh setiap tugas.

Atribut

Semua atribut pada kelas ini bersifat *private* (ditandai dengan tanda `_`) sebagai penerapan encapsulation, antara lain:

- `_name`: menyimpan nama tugas
- `_time`: menyimpan waktu atau tanggal tugas
- `_description`: menyimpan deskripsi tugas
- `_priority`: menyimpan tingkat prioritas tugas
- `_completed`: menandakan apakah tugas sudah selesai atau belum
- `_motivational_quote`: menyimpan quote motivasi yang ditampilkan saat reminder

m

Method

- Getter dan setter (get_name(), set_name(), dan lainnya) untuk mengakses atribut private
 - Is_completed() untuk mengecek status tugas
 - Mark_competed() untuk menandai tugas sebagai selesai
 - Display_info() untuk menampilkan informasi tugas
 - _generate_quote() untuk menghasilkan quote motivasi secara internal
- 2) Kelas DailyTask(subclass)

Kelas DailyTask merupakan turunan dari kelas Task yang digunakan untuk menangani tugas harian.

Atribut tambahan

- _reminder_minute: menyimpan waktu pengingat sebelum tugas dimulai (dalam menit)

Method

- Display_info() di override untuk menampilkan informasi tambahan berupa pengingat (reminder)

Kelas ini memanfaatkan inheritance untuk menggunakan atribut dan method dari Task, sekaligus menerapkan polymorphism melalui override method display_info().

3) Kelas RecurringTask (Subclass)

Kelas RecurringTask juga merupakan turunan dari kelas Task, yang digunakan untuk tugas berulang.

Atribut tambahan

- _recurrence_days: menyimpan daftar hari pengulangan tugas (misalnya senin, rabu, jumat)

Method

- Display_info() di override untuk menampilkan informasi hari pengulangan
- Update_to_next() untuk memperbarui tugas ke jadwal berikutnya.

BAB V

ALUR APLIKASI

5.1 Gambaran umum alur aplikasi

Alur aplikasi JadwalKu Pro menggambarkan bagaimana pengguna berinteraksi dengan sistem, mulai dari menambahkan tugas, menerima pengingat, hingga mengekspor data tugas. Setiap proses dirancang agar mudah digunakan dan berjalan secara sistematis sesuai dengan konsep pemrograman berorientasi objek.

5.2 Alur tambah tugas

1) Pengguna mengisi form tugas

Pengguna mengisi form yang tersedia pada aplikasi, seperti nama tugas, tanggal dan waktu, kategori, prioritas, serta tipe tugas (harian atau berulang).

2) Validasi input

Sistem melakukan pengecekan terhadap data yang dimasukkan pengguna untuk memastikan semua field telah diisi dengan benar dan sesuai format yang ditentukan.

3) Buat objek task (daily/recurring)

Setelah data dinyatakan valid, sistem akan membuat objek tugas sesuai dengan tipe yang dipilih, yaitu objek DailyTask atau RecurringTask.

4) Menambahkan Tugas ke Daftar Tugas

Objek tugas yang telah dibuat kemudian dimasukkan ke dalam list tugas yang digunakan sebagai penyimpanan data sementara selama aplikasi berjalan.

5) Update tabel dan progress

Sistem memperbarui tabel daftar tugas dan progress bar agar pengguna dapat langsung melihat tugas yang baru ditambahkan beserta status penyelesaiannya.

5.3 Alur reminder dan export

1) Alur reminder

- Sistem menjalankan thread yang secara berkala mengecek waktu, setiap 30 detik.
- Jika waktu tugas sudah mendekati jadwal yang ditentukan, sistem akan menampilkan notifikasi berupa pop-up.
- Pop-up reminder disertai dengan quote motivasi untuk memberikan semangat kepada pengguna.

2) Alur export data

- Pengguna memilih menu atau tombol export data.
- Sistem mengambil seluruh data tugas yang tersimpan dalam list tugas.
- Data tersebut disimpan ke dalam file berformat CSV menggunakan modul csv.writer.
- File CSV dapat dibuka kembali menggunakan aplikasi spreadsheet seperti Microsoft Excel atau LibreOffice Calc.

BAB VI

TANTANGAN YANG DIHADAPI

6.1 Tantangan teknis

Dalam proses pengembangan aplikasi JadwalKu Pro, terdapat beberapa tantangan teknis yang dihadapi, khususnya saat menggabungkan berbagai fitur dalam satu aplikasi berbasis GUI Tkinter.

- 1) Integrasi threading untuk fitur reminder. Jika tidak diatur dengan benar, penggunaan threading dapat menyebabkan antarmuka aplikasi menjadi *freeze* atau tidak merespons. Hal ini terjadi karena Tkinter tidak mengizinkan pembaruan tampilan GUI secara langsung dari thread lain.
- 2) Melakukan validasi input tanggal dan waktu (datetime) dari pengguna. Format input yang tidak sesuai dapat menyebabkan error saat aplikasi dijalankan. Oleh karena itu, diperlukan mekanisme validasi yang baik agar aplikasi tetap berjalan stabil.
- 3) Instalasi library tkcalendar juga menjadi tantangan, terutama ketika aplikasi dijalankan pada environment atau perangkat yang berbeda. Tidak semua environment Python secara default memiliki library tersebut.
- 4) Pengaturan warna pada Treeview. Warna baris atau status tugas tidak langsung muncul karena konfigurasi tag dilakukan berulang di dalam loop, sehingga tampilan tidak konsisten.

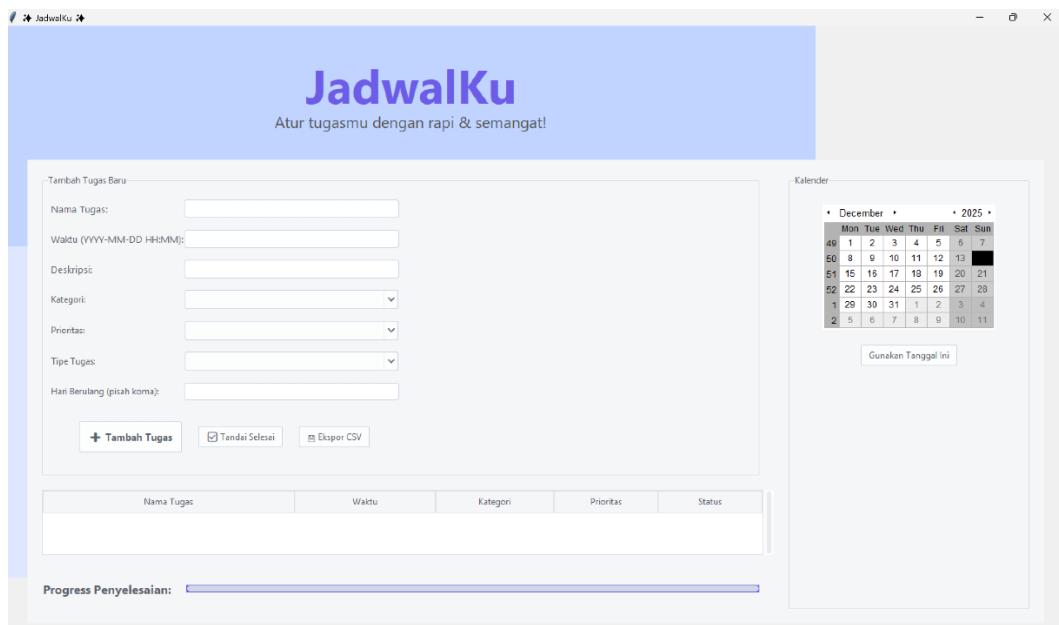
6.2 Solusi yang diterapkan

- 1) Untuk mengatasi tantangan integrasi threading, digunakan method `root.after()` agar proses pembaruan antarmuka pengguna tetap dilakukan di *main thread*. Dengan cara ini, reminder dapat berjalan secara real-time tanpa menyebabkan aplikasi menjadi tidak responsif.
- 2) Validasi input tanggal dan waktu dilakukan menggunakan blok `try-except` saat proses parsing datetime. Jika format input tidak sesuai, sistem akan menampilkan pesan kesalahan kepada pengguna sehingga data yang dimasukkan dapat diperbaiki.
- 3) Masalah instalasi tkcalendar diatasi dengan menyediakan instruksi instalasi yang jelas menggunakan perintah `pip install tkcalendar`. Dengan adanya panduan ini, pengguna atau pengujji aplikasi dapat dengan mudah menyiapkan environment yang dibutuhkan.
- 4) Untuk mengatasi masalah tampilan warna pada Treeview, konfigurasi `tag_configure` dipindahkan ke luar loop pada fungsi `update_task_list`. Dengan cara ini, pengaturan warna hanya dilakukan satu kali dan hasilnya dapat ditampilkan secara konsisten pada setiap pembaruan data.

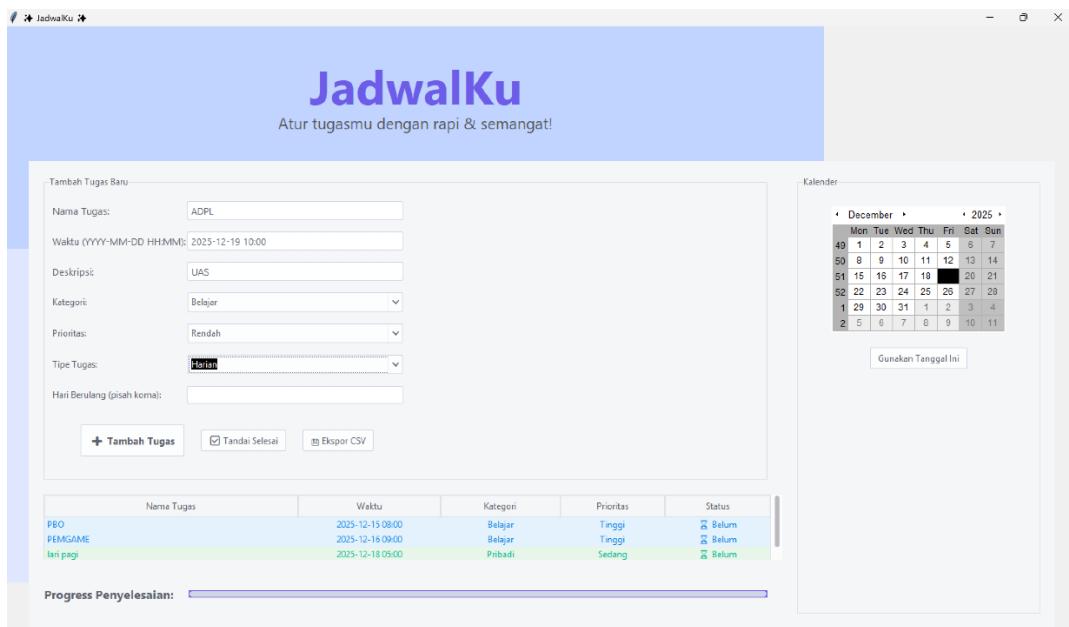
BAB VII

IMPLEMENTASI

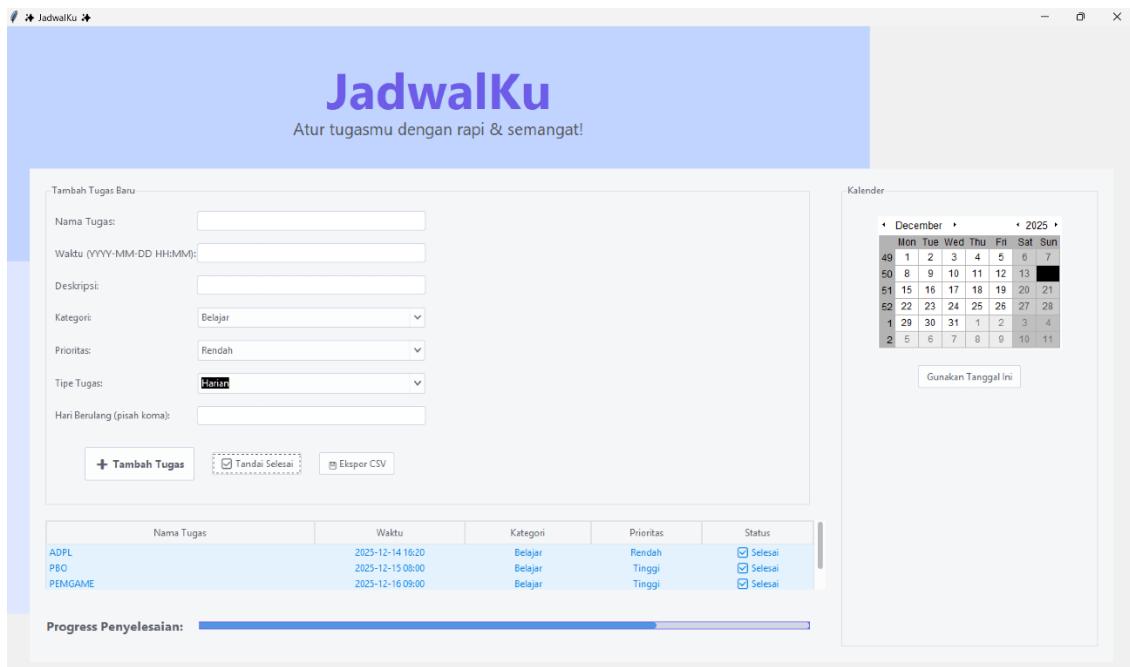
7.1 Screenshot aplikasi



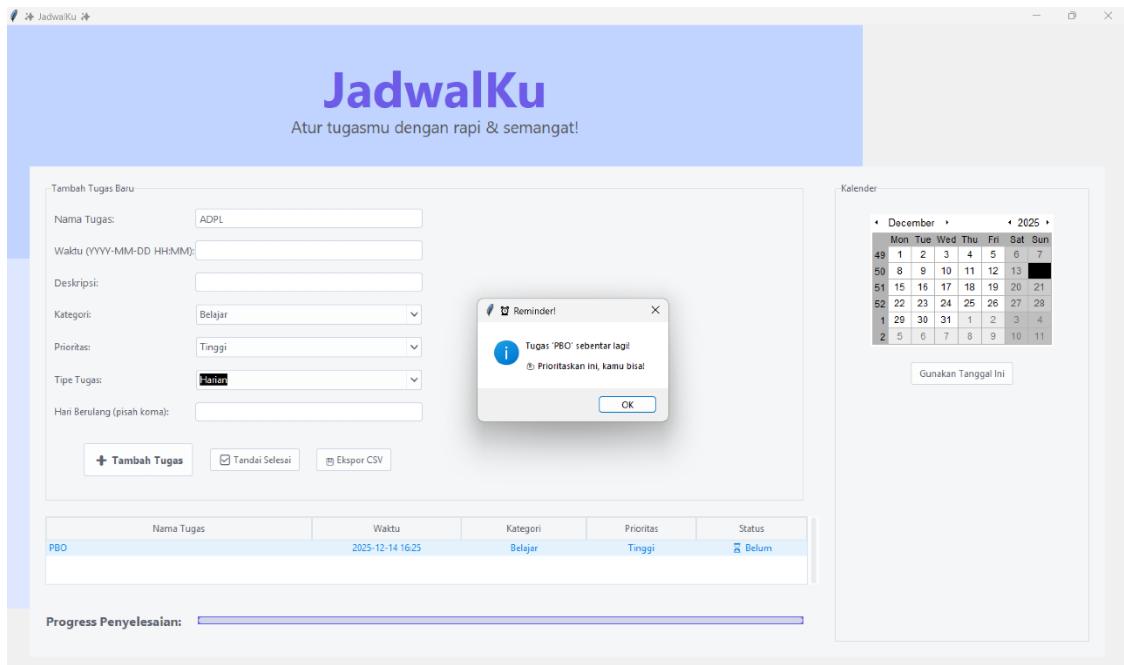
Gambar 1. 1 GUI kosong



Gambar 1. 2 setalah tambah tugas dengan warna kategori



Gambar 1. 3 Progress bar setelah tandai selesai.



Gambar 1. 4 Pop-up reminder.

BAB VIII

PENUTUP

8.1 kesimpulan

Berdasarkan hasil perancangan dan implementasi aplikasi JadwalKu Pro, dapat disimpulkan bahwa proyek ini berhasil menerapkan konsep Object Oriented Programming (OOP) dengan baik. Konsep encapsulation, inheritance, dan polymorphism telah diterapkan secara nyata melalui struktur kelas dan alur program yang terorganisir.

Aplikasi yang dihasilkan bersifat fungsional dan mampu membantu pengguna dalam mengelola jadwal serta tugas sehari-hari. Fitur tambahan seperti reminder dan quote motivasi memberikan nilai lebih pada aplikasi, tidak hanya dari sisi teknis tetapi juga dari sisi kreativitas dan kenyamanan pengguna. Dengan adanya fitur-fitur tersebut, aplikasi menjadi lebih menarik dan bermanfaat untuk meningkatkan produktivitas pengguna.

8.2 Saran

Untuk pengembangan selanjutnya, aplikasi JadwalKu Pro dapat ditingkatkan dengan menambahkan penyimpanan data menggunakan database agar data tugas tidak hilang saat aplikasi ditutup. Selain itu, fitur notifikasi dapat dikembangkan agar tetap berjalan meskipun aplikasi tidak aktif.

Pengembangan antarmuka pengguna juga dapat ditingkatkan agar tampil lebih modern dan responsif. Terakhir, aplikasi ini dapat dikembangkan ke platform lain, seperti aplikasi web atau mobile, sehingga dapat digunakan dengan lebih fleksibel oleh pengguna.

Link Github : https://github.com/Azkiya111/PBO_UAS_JadwalKu