

# **DOKUMENTASI APLIKASI DAILY PLANNER**

**Mata Kuliah:**

Pemrograman Berorientasi Objek

**Oleh:**

Firta Aulika Aji Kusuma

24091397029

2024B



**PROGRAM STUDI D4 MANAJEMEN INFORMATIKA  
FAKULTAS VOKASI  
UNIVERSITAS NEGERI SURABAYA  
2025**

## **DAFTAR ISI**

DAFTAR ISI.....	1
BAB I PENDAHULUAN.....	2
1.1 Latar Belakang.....	2
1.2 Tujuan Pembuatan Aplikasi.....	2
1.3 Deskripsi Sistem.....	2
BAB II PERANCANGAN SISTEM.....	3
2.1 Class Diagram.....	3
2.2 Penjelasan Class Diagram.....	3
2.3 Implementasi Program.....	4
BAB III FITUR DAN PENGGUNAAN APLIKASI.....	5
3.1 Fitur Aplikasi.....	5
3.2 Cara Penggunaan Aplikasi.....	5
3.3 Tampilan Aplikasi.....	5
KESIMPULAN.....	6

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Aplikasi Daily Planner dibuat sebagai solusi untuk membantu pengguna dalam mengatur dan memvisualisasikan jadwal kegiatan harian secara terstruktur. Pengelolaan waktu yang kurang baik sering menyebabkan aktivitas menjadi tidak teratur dan tidak efisien. Oleh karena itu, dibutuhkan sebuah aplikasi yang mampu menampilkan jadwal secara visual agar mudah dipahami.

Selain itu, pembuatan aplikasi ini juga bertujuan sebagai penerapan konsep Pemrograman Berorientasi Objek (PBO) ke dalam sebuah proyek nyata menggunakan bahasa pemrograman Python.

## 1.2 Tujuan Pembuatan Aplikasi

Tujuan dari pembuatan aplikasi Daily Planner adalah:

- Menerapkan konsep dasar Pemrograman Berorientasi Objek.
- Membuat aplikasi pengelolaan jadwal harian yang interaktif.
- Melatih perancangan sistem menggunakan class diagram.
- Menghasilkan program yang terstruktur dan mudah dikembangkan.

## 1.3 Deskripsi Sistem

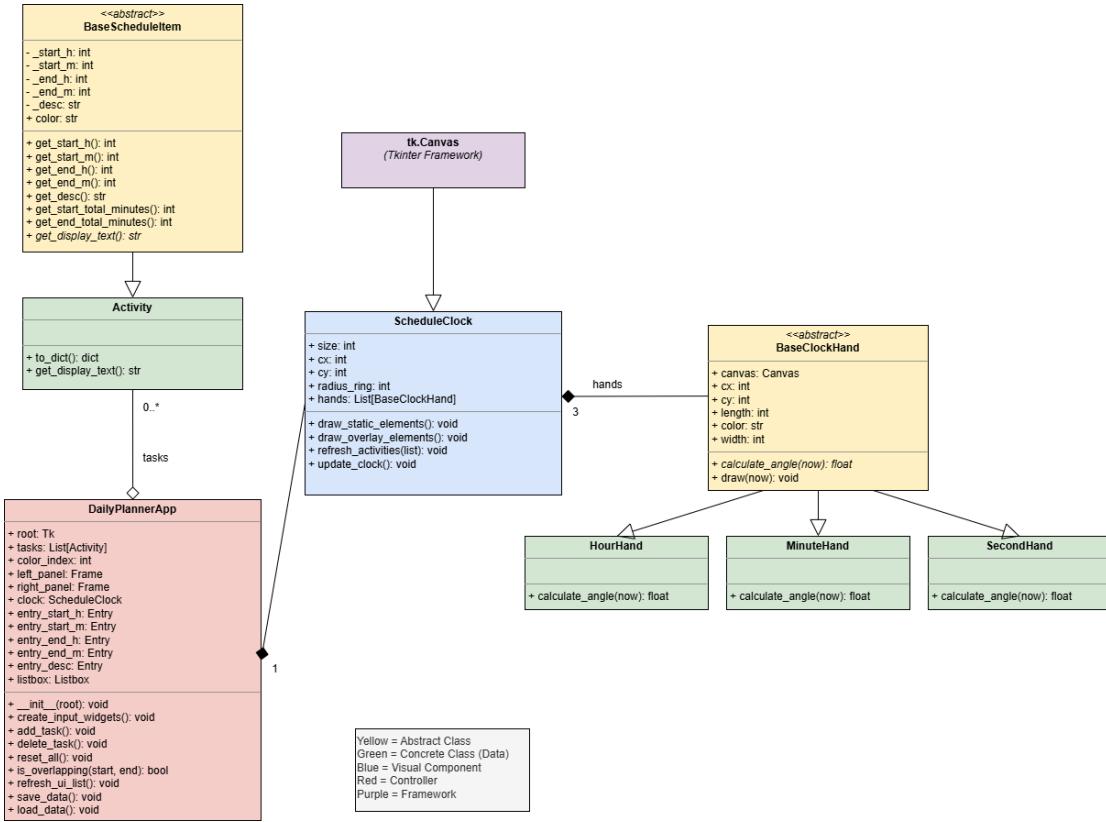
Aplikasi Daily Planner merupakan aplikasi desktop berbasis Python yang memungkinkan pengguna untuk menambahkan aktivitas dengan waktu mulai, waktu selesai, dan deskripsi kegiatan. Aktivitas tersebut divisualisasikan dalam bentuk jam analog sehingga pengguna dapat melihat pembagian waktu secara langsung.

Aplikasi menyediakan fitur penambahan jadwal, penghapusan jadwal, serta penyimpanan data secara otomatis menggunakan file lokal agar data tidak hilang saat aplikasi ditutup.

## BAB II PERANCANGAN SISTEM

### 2.1 Class Diagram

Class diagram digunakan untuk menggambarkan struktur kelas, atribut, method, serta hubungan antar class yang ada dalam aplikasi Daily Planner.



Gambar 1 Class Diagram Aplikasi Daily Planner

### 2.2 Penjelasan Class Diagram

Pada aplikasi Daily Planner, perancangan sistem dilakukan dengan membagi aplikasi ke dalam beberapa class utama dengan struktur hierarki yang jelas. Diagram kelas ini menunjukkan implementasi lengkap dari konsep Object-Oriented Programming.

#### Struktur Utama Kelas:

1. Abstract Base Classes (Warna Kuning):
  - `BaseScheduleItem`: Kelas abstrak yang mendefinisikan struktur dasar untuk item jadwal. Mengimplementasikan encapsulation dengan menggunakan atribut protected (prefix `_`) dan menyediakan metode getter untuk akses terkontrol.
  - `BaseClockHand`: Kelas abstrak untuk sistem jarum jam dengan metode abstrak `calculate_angle()` yang wajib diimplementasikan oleh kelas turunan.
2. Concrete Data Classes (Warna Hijau):
  - `Activity`: Kelas konkret yang mewarisi dari `BaseScheduleItem`, merepresentasikan satu aktivitas dalam jadwal. Mengimplementasikan method `to_dict()` untuk serialisasi data dan override method `get_display_text()`.
  - `HourHand`, `MinuteHand`, `SecondHand`: Kelas konkret turunan dari `BaseClockHand` yang masing-masing mengimplementasikan perhitungan sudut berbeda untuk jarum jam, menit, dan detik.

3. Visual Component (Warna Biru):
  - ScheduleClock: Kelas turunan dari tk.Canvas yang bertanggung jawab untuk visualisasi jam analog dan pie chart jadwal. Menggunakan composition untuk memiliki 3 objek jarum jam.
4. Controller (Warna Merah):
  - DailyPlannerApp: Kelas utama yang mengontrol alur aplikasi, menangani interaksi pengguna, validasi input, dan pengelolaan data.
5. Framework Class (Warna Ungu):
  - tk.Canvas: Kelas dari framework Tkinter yang menyediakan canvas untuk menggambar komponen visual.

#### **Hubungan Antar Kelas:**

1. Inheritance (Panah Putih Segitiga):
  - Activity → BaseScheduleItem
  - HourHand, MinuteHand, SecondHand → BaseClockHand
  - ScheduleClock → tk.Canvas
2. Composition (Diamond Hitam):
  - ScheduleClock → BaseClockHand (multiplicity: 3)
  - DailyPlannerApp → ScheduleClock (multiplicity: 1)
3. Aggregation (Diamond Putih):
  - DailyPlannerApp → Activity (multiplicity: 0..\*)

#### **Implementasi Konsep OOP:**

1. Abstraction: BaseScheduleItem dan BaseClockHand sebagai kelas abstrak yang mendefinisikan interface tanpa implementasi detail.
2. Encapsulation: Atribut protected di BaseScheduleItem dengan akses melalui getter methods.
3. Inheritance: Pewarisan sifat dari kelas induk ke kelas anak untuk menghindari duplikasi kode.
4. Polymorphism: Method calculate\_angle() di-override berbeda di setiap kelas jarum jam.

#### **Alur Data:**

Pengguna input → DailyPlannerApp validasi → Buat objek Activity → Simpan ke list tasks → ScheduleClock visualisasikan → BaseClockHand dan turunannya gambar jarum dengan logika berbeda.

### **2.3 Implementasi Program**

Aplikasi Daily Planner diimplementasikan menggunakan bahasa pemrograman Python dengan library Tkinter sebagai antarmuka grafis. Konsep Pemrograman Berorientasi Objek diterapkan secara konsisten melalui penggunaan class, inheritance, abstraction, encapsulation, dan polymorphism. Data aktivitas disimpan dalam file JSON (firta\_plan.json) untuk menjaga persistensi data. Struktur program dirancang secara modular sehingga setiap class memiliki tanggung jawab yang jelas. Aplikasi mengikuti arsitektur three-tier dengan pemisahan yang jelas antara presentation layer (GUI), business logic layer (validasi dan pengolahan), dan data layer (model dan penyimpanan).

#### **Arsitektur Three-Tier:**

1. Presentation Layer: DailyPlannerApp (GUI), ScheduleClock (visual)
2. Business Logic Layer: Validasi input, overlap detection, sorting
3. Data Layer: BaseScheduleItem, Activity, JSON persistence

Diagram kelas yang diperbarui ini secara akurat merepresentasikan struktur kode yang sebenarnya dan hubungan antar komponen dalam aplikasi Daily Planner.

## BAB III FITUR DAN PENGGUNAAN APLIKASI

### 3.1 Fitur Aplikasi

Aplikasi Daily Planner memiliki beberapa fitur utama yang dirancang untuk membantu pengguna dalam mengatur dan memvisualisasikan jadwal kegiatan harian. Fitur pertama adalah penambahan jadwal kegiatan, di mana pengguna dapat memasukkan waktu mulai, waktu selesai, serta deskripsi aktivitas yang akan dilakukan. Data aktivitas tersebut kemudian disimpan dan dikelola oleh sistem.

Fitur selanjutnya adalah visualisasi aktivitas pada jam analog. Setiap aktivitas yang ditambahkan akan ditampilkan dalam bentuk visual pada jam, sehingga pengguna dapat melihat pembagian waktu secara jelas dan intuitif. Selain itu, aplikasi juga menyediakan fitur penghapusan jadwal untuk menghapus aktivitas yang sudah tidak diperlukan. Data aktivitas disimpan secara lokal menggunakan file, sehingga jadwal tetap tersimpan meskipun aplikasi ditutup.

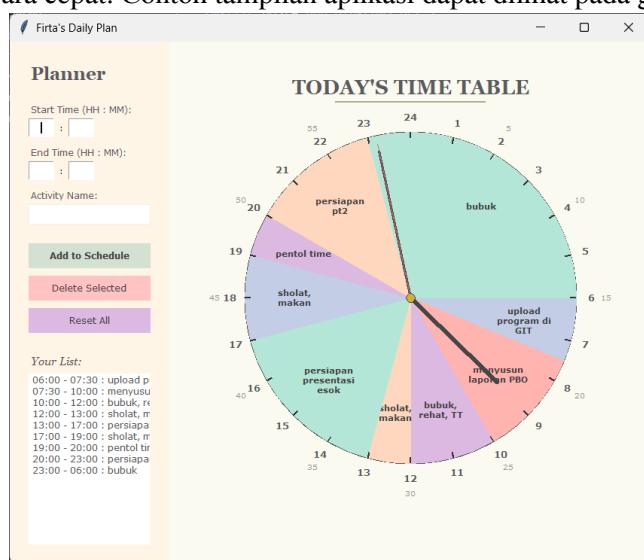
### 3.2 Cara Penggunaan Aplikasi

Untuk menggunakan aplikasi Daily Planner, pengguna dapat menjalankan program melalui file main.py. Setelah aplikasi terbuka, pengguna mengisi waktu mulai dan waktu selesai kegiatan pada kolom yang telah disediakan. Selanjutnya, pengguna memasukkan deskripsi aktivitas sesuai dengan kegiatan yang akan dilakukan.

Setelah semua data diisi, pengguna dapat menekan tombol tambah untuk menyimpan aktivitas. Aktivitas yang telah ditambahkan akan langsung muncul pada daftar kegiatan dan divisualisasikan pada jam analog. Jika pengguna ingin menghapus suatu aktivitas, pengguna dapat memilih aktivitas tersebut dari daftar dan menekan tombol hapus. Proses ini memungkinkan pengguna mengelola jadwal harian dengan mudah dan terstruktur.

### 3.3 Tampilan Aplikasi

Tampilan aplikasi Daily Planner dirancang sederhana dan mudah digunakan. Antarmuka aplikasi terdiri dari area input untuk memasukkan data aktivitas, daftar aktivitas yang telah ditambahkan, serta tampilan jam analog sebagai visualisasi jadwal. Tampilan ini membantu pengguna memahami alur penggunaan aplikasi secara cepat. Contoh tampilan aplikasi dapat dilihat pada gambar berikut.



## **KESIMPULAN**

Berdasarkan hasil perancangan dan implementasi aplikasi Daily Planner, dapat disimpulkan bahwa penerapan konsep Pemrograman Berorientasi Objek dapat membantu dalam membangun aplikasi yang terstruktur dan mudah dipahami. Aplikasi ini berhasil menerapkan konsep abstraction, encapsulation, inheritance, polymorphism, association, dependency, dan composition sesuai dengan perancangan class diagram.