

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 12



2411102441215

Nikon Valent Sakaesa

C

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

Laporan Singkat

TUGAS MANDIRI PRAKTIKUM 12

1. Tujuan Praktikum

Tugas mandiri pada Praktikum 12 bertujuan untuk melatih penerapan konsep Pemrograman Berorientasi Objek (PBO) dengan memperhatikan prinsip SOLID, serta membiasakan penggunaan logging sebagai sarana pencatatan proses program. Studi kasus yang digunakan pada praktikum ini adalah validasi data registrasi mahasiswa.

2. Gambaran Umum Program

Program yang dibuat berfungsi untuk melakukan validasi terhadap data registrasi mahasiswa. Validasi dilakukan berdasarkan beberapa kriteria, yaitu jumlah SKS yang diajukan, pemenuhan mata kuliah prasyarat, dan pengecekan bentrokkan jadwal kuliah. Data mahasiswa direpresentasikan dalam class `RegistrationData`, sedangkan proses validasi dijalankan oleh `RegistrationService` yang menerima beberapa aturan validasi dalam bentuk objek.

3. Penerapan Prinsip SOLID

Dalam pengembangan program ini, dilakukan refactoring dengan menerapkan prinsip-prinsip SOLID sebagai berikut:

- Single Responsibility Principle (SRP)
Setiap aturan validasi dipisahkan ke dalam class tersendiri sehingga setiap class hanya memiliki satu fungsi utama.
- Open/Closed Principle (OCP)
Program dirancang agar dapat dikembangkan dengan menambahkan aturan validasi baru tanpa perlu mengubah struktur class utama.
- Dependency Inversion Principle (DIP)
Class `RegistrationService` tidak bergantung langsung pada aturan validasi tertentu, melainkan pada interface `IValidationRule`.

Sebagai perbandingan, class `ValidatorManager` tetap disertakan untuk menunjukkan contoh pendekatan lama yang belum menerapkan prinsip SOLID dengan baik.

4. Penggunaan Logging

Untuk memantau jalannya program, digunakan modul logging sebagai pengganti print. Logging dimanfaatkan untuk mencatat proses validasi sejak awal, menampilkan hasil dari setiap aturan validasi, hingga menentukan apakah proses registrasi dinyatakan berhasil atau gagal. Dengan logging, informasi eksekusi program menjadi lebih terstruktur dan mudah dipahami.

5. Hasil Pengujian

Hasil pengujian menunjukkan bahwa seluruh aturan validasi dapat dijalankan secara berurutan sesuai dengan data yang diuji. Informasi yang ditampilkan melalui logging membantu dalam memahami alur proses validasi dan hasil akhir dari registrasi mahasiswa.

6. Kesimpulan

Berdasarkan hasil praktikum, dapat disimpulkan bahwa penerapan prinsip SOLID dan penggunaan logging pada program validasi registrasi mahasiswa mampu meningkatkan kualitas desain kode. Program menjadi lebih modular, mudah dikembangkan, dan lebih sesuai dengan praktik pemrograman berorientasi objek yang baik.

Hasil Output

```
PS D:\PBO_prak tugas dll> & C:\Users\Valen\AppData\Local\Programs\Python\Python313\python.exe
iri.py"
INFO - Memulai validasi untuk mahasiswa Ani
INFO - Validasi SKS berhasil
INFO - Menjalankan SksLimitRule -> True
INFO - Validasi prasyarat berhasil
INFO - Menjalankan PrerequisiteRule -> True
INFO - Validasi jadwal berhasil
INFO - Menjalankan JadwalBentrokRule -> True
INFO - Validasi registrasi BERHASIL
PS D:\PBO_prak tugas dll>
```

Link GitHub : <https://github.com/2411102441215-Valent/TUGAS-DLL>

Refleksi

Penggunaan *docstring* membantu menjelaskan fungsi dan tujuan setiap class maupun method sehingga memudahkan anggota tim lain untuk memahami alur program. Dengan *docstring*, kode menjadi lebih mudah dibaca dan dipelajari tanpa harus menelusuri seluruh isi program. Selain itu, penerapan *logging* memungkinkan proses eksekusi program dipantau secara jelas tanpa menggunakan banyak print. Logging sangat membantu dalam proses debugging karena kesalahan atau proses penting dapat dilacak dengan cepat. Kombinasi *docstring* dan *logging* membuat Mini Project UAS lebih mudah dikembangkan, dirawat, dan dikerjakan secara kolaboratif.