

NIM : 2201010153

NAMA : I Kadek Andre Alfiana

Project : Data Onepiece

Deskripsi :

Project ini adalah aplikasi desktop yang menampilkan data kru dari seri One Piece. Aplikasi ini memiliki fitur untuk menambah (Baru), mengubah (Ubah), dan menghapus (Hapus) data kru.

Berikut adalah penjelasan mendetail mengenai project ini beserta metode dan teknik pemrograman OOP yang digunakan:

Deskripsi Project

Aplikasi ini adalah sistem manajemen data kru yang menampilkan data seperti ID Kru, Nama, Umur, dan Asal dalam tabel. Pengguna dapat menambah data baru, mengubah data yang sudah ada, dan menghapus data yang tidak diperlukan.

Fitur-Fitur

- Tampilan Tabel : Menampilkan data kru dalam bentuk tabel.
- Form Input : Memungkinkan pengguna untuk memasukkan nama, umur, dan asal kru.
- Tombol Operasi : Terdapat tombol untuk menambah data baru, mengubah data, menghapus data, dan menutup aplikasi.

Metode dan Teknik Pemrograman OOP yang Digunakan

1. Kelas dan Objek

- Penggunaan : Membuat kelas `Kru` yang memiliki atribut seperti `idKru`, `nama`, `umur`, dan `asal`.
- Alasan : Penggunaan kelas dan objek memungkinkan pengelompokan data yang terkait dalam satu entitas, memudahkan pengelolaan dan manipulasi data.

2. Encapsulation (Enkapsulasi)

- Penggunaan : Menggunakan akses modifier `private` untuk atribut kelas `Kru` dan menyediakan metode `getter` dan `setter` untuk akses dan modifikasi data.
- Alasan : Enkapsulasi melindungi data dari akses langsung luar kelas, memastikan integritas data dan memberikan kontrol lebih pada bagaimana data diakses atau diubah.

3. Inheritance (Pewarisan)

- Penggunaan : Mungkin ada kelas dasar `Person` yang memiliki atribut umum seperti `nama` dan `umur`, dan kelas `Kru` mewarisi dari kelas ini.

- Alasan : Mengurangi duplikasi kode dengan mewarisi properti dan metode umum dari kelas dasar.

4. Polymorphism (Polimorfisme)

- Penggunaan : Menggunakan polimorfisme untuk metode yang dapat di-override dalam kelas turunan, seperti metode `tampilkanData` yang dapat diimplementasikan berbeda di kelas `Kru`.
- Alasan : Memungkinkan penggunaan metode yang sama dengan perilaku yang berbeda tergantung pada objek yang memanggilnya, meningkatkan fleksibilitas dan keterbacaan kode.

5. Abstract Classes and Interfaces (Kelas Abstrak dan Antarmuka)

- Penggunaan : Kelas abstrak atau antarmuka untuk mendefinisikan metode umum yang harus diimplementasikan oleh semua kelas turunan.
- Alasan : Memastikan bahwa kelas turunan mengikuti kontrak tertentu, memaksa implementasi metode penting, dan mendukung desain yang lebih modular dan terstruktur.

Alasan Penggunaan Teknik OOP

- Modularitas : Teknik OOP memecah program menjadi modul yang lebih kecil dan lebih mudah dikelola, setiap modul bisa dikembangkan dan diuji secara terpisah.
- Reusability : Penggunaan inheritance memungkinkan kode yang sudah ada untuk digunakan kembali, mengurangi pengulangan kode dan mempercepat pengembangan.
- Maintainability : Struktur OOP membuat kode lebih mudah dipelihara dan dikembangkan di masa depan, karena setiap bagian kode berada dalam kelas yang sesuai dengan tanggung jawabnya.
- Scalability : Aplikasi yang dibangun dengan prinsip OOP lebih mudah diskalakan karena struktur yang jelas dan pemisahan tanggung jawab.

Dengan teknik dan metode ini, aplikasi manajemen data Onepiece dapat dikembangkan dengan lebih efisien, terstruktur, dan mudah dipelihara. Jika ada aspek spesifik yang ingin Anda ketahui lebih dalam, silakan beri tahu saya!