

Nama : I Kadek Aditia Pradipta

Nim : 2201010022

Project : Data Pengajuan Sertifikasi

Deskripsi Proyek

Proyek ini bertujuan untuk mengembangkan aplikasi desktop menggunakan bahasa pemrograman Java yang memungkinkan pengguna untuk mengelola data pengajuan sertifikasi mahasiswa. Aplikasi ini memungkinkan pengguna untuk menambahkan, mengedit, menghapus, dan menampilkan data mahasiswa yang mengajukan sertifikasi. Data yang dikelola mencakup NIM (Nomor Induk Mahasiswa), Nama, Sertifikasi, Jenis Kelamin, Agama, Status Pembayaran, dan Jadwal Sertifikasi.

Fitur Utama

1. **Menampilkan Data Mahasiswa:** Aplikasi menampilkan data mahasiswa dalam bentuk tabel yang dapat diperbarui.
2. **Penambahan Data Mahasiswa:** Pengguna dapat menambahkan data mahasiswa baru melalui formulir input.
3. **Pengeditan Data Mahasiswa:** Data mahasiswa yang sudah ada dapat diedit dan disimpan kembali.
4. **Penghapusan Data Mahasiswa:** Data mahasiswa dapat dihapus dari tabel.
5. **Pengelolaan Gambar:** Aplikasi dapat menampilkan gambar mahasiswa berdasarkan NIM.

Teknik Pemrograman Berorientasi Objek (OOP) yang Digunakan

1. **Encapsulation:** Menggunakan kelas untuk mengelompokkan data dan metode yang terkait. Misalnya, kelas `data_mahasiswa` untuk mengelola data mahasiswa.
2. **Inheritance:** Meskipun tidak secara eksplisit digunakan dalam kode yang diberikan, konsep pewarisan dapat diterapkan untuk memperluas fungsionalitas dari kelas-kelas yang ada.
3. **Polymorphism:** Implementasi polimorfisme melalui overloading atau overriding metode dapat diterapkan untuk berbagai tindakan seperti menyimpan dan mengedit data.
4. **Abstraction:** Mengabstraksi detail implementasi dari interaksi pengguna, misalnya dengan metode `simpan`, `rubah`, dan `UpdateJTable` di kelas `data_mahasiswa`.

Penjelasan:

- **Access Modifier:** Atribut dan metode dalam `DataMahasiswaDAO` menggunakan access modifier yang sesuai untuk memastikan keamanan dan fleksibilitas.
- **Constructor:** Constructor `DataMahasiswaDAO()` untuk menginisialisasi objek `conn` menggunakan koneksi dari kelas `Koneksi`.
- **Polymorphism:** Menggunakan list generik `List<Mahasiswa>` untuk menampung objek `Mahasiswa`, memungkinkan fleksibilitas dalam pengelolaan dan pengaksesan data.

Implementasi Teknik dan Metode OOP

- **Access Modifier:** Menggunakan `private` untuk atribut dan `public` untuk metode yang memastikan enkapsulasi data dan memungkinkan akses terkontrol dari luar kelas.

- **Overloading:** Tidak ada contoh overloading di dalam contoh ini, namun bisa diterapkan pada berbagai metode jika diperlukan untuk memungkinkan fungsionalitas tambahan dengan nama yang sama tetapi parameter yang berbeda.
- **Method Inheritance:** Menggunakan inheritance secara implisit dari kelas Object Java, serta memanfaatkan pengelolaan data dengan desain yang terstruktur dalam kelas Mahasiswa.
- **Constructor:** Menggunakan constructor untuk menginisialisasi objek Mahasiswa dan DataMahasiswaDAO, yang memungkinkan inisialisasi data saat objek dibuat.
- **Polymorphism:** Contoh polymorphism terjadi dalam penggunaan objek Mahasiswa dan List<Mahasiswa>, di mana kelas Mahasiswa digunakan secara umum untuk merepresentasikan objek mahasiswa dalam aplikasi.

Penyempurnaan Potensial

- **Validasi Input:** Menambahkan validasi input untuk memastikan data yang dimasukkan sesuai dengan format yang diharapkan.
- **Manajemen Koneksi:** Menyempurnakan manajemen koneksi dengan memastikan koneksi ditutup setelah digunakan untuk mencegah kebocoran sumber daya.
- **Penanganan Kesalahan:** Meningkatkan penanganan kesalahan dengan memberikan pesan yang lebih informatif atau melakukan logging untuk debug lebih lanjut.