



UJIAN TENGAH SEMESTER

Pemrograman Berorientasi Object (PBO)

Taofik (nim : 20230803140)

Apa itu Pemrograman Berbasis Objek?

Pemrograman Berbasis Objek (Object-Oriented Programming atau OOP) adalah paradigma pemrograman yang menggunakan konsep "objek" sebagai dasar dalam merancang dan mengembangkan perangkat lunak. Objek ini merupakan entitas yang menggabungkan data (disebut atribut atau properti) dan fungsi (disebut metode) yang beroperasi pada data tersebut.

Prinsip Utama OOP

1. **Encapsulation (Enkapsulasi)** Menggabungkan data dan metode dalam satu unit (objek) dan menyembunyikan detail implementasi dari luar objek. Ini membantu menjaga keamanan data dan mengurangi kompleksitas.
2. **Inheritance (Pewarisan)** Memungkinkan sebuah kelas (class) untuk mewarisi sifat dan perilaku dari kelas lain, sehingga kode dapat digunakan kembali dan mempercepat pengembangan.
3. **Polymorphism (Polimorfisme)** Kemampuan objek untuk mengambil banyak bentuk, misalnya metode yang sama dapat berperilaku berbeda pada objek yang berbeda.
4. **Abstraction (Abstraksi)** Menyembunyikan detail yang tidak perlu dan hanya menampilkan fitur penting dari objek, sehingga memudahkan pengguna dalam berinteraksi dengan objek tersebut.

Kegunaan Pemrograman Berbasis Objek

Pemrograman Berbasis Objek sangat berguna dalam pengembangan perangkat lunak modern karena beberapa alasan berikut:

1. Modularitas

Kode dibagi menjadi objek-objek yang mandiri, sehingga memudahkan pengelolaan dan pengembangan secara terpisah. Ini membuat proyek besar lebih terstruktur dan mudah dipahami.

2. Reusabilitas Kode

Dengan pewarisan dan penggunaan kembali objek, programmer dapat menghindari penulisan ulang kode yang sama, sehingga menghemat waktu dan usaha.

3. Pemeliharaan yang Mudah

Karena kode terorganisir dalam objek yang terpisah, perbaikan bug atau penambahan fitur baru dapat dilakukan tanpa mempengaruhi bagian lain dari program secara signifikan.

4. Skalabilitas

OOP memudahkan pengembangan aplikasi yang kompleks dan besar karena struktur yang jelas dan kemampuan untuk memperluas kelas dan objek.

5. Meningkatkan Produktivitas

Dengan konsep yang jelas dan penggunaan kembali kode, pengembang dapat bekerja lebih cepat dan efisien.

Perbandingan Singkat OOP dengan Paradigma Lain

| Aspek | Pemrograman Berbasis Objek (OOP) | Pemrograman Prosedural |
|-------------------|---|------------------------------|
| Struktur Kode | Berbasis objek dan kelas | Berbasis fungsi dan prosedur |
| Pengelolaan Data | Data dan fungsi digabung dalam objek | Data dan fungsi terpisah |
| Reusabilitas Kode | Tinggi (melalui pewarisan dan polimorfisme) | Rendah |
| Pemeliharaan | Mudah karena modularitas | Sulit jika program besar |
| Abstraksi | Mendukung abstraksi data dan fungsi | Terbatas |