Laporan Praktikum

Mata Kuliah Pemprograman Berorientasi Objek

Pertemuan 5. Praktikum 5

“polymorphism”

Dosen Pengampu : Willdan Aprizal Aripin, S.Pd., M.Kom.

Disusun Oleh :

Nuril Khairiyah

2300231

PROGRAM STUDI SISTEM INFORMASI KELAUTAN

UNIVERSITAS PENDIDIKAN INDONESIA

2024

I. PENDAHULUAN

Polymorphism merupakan salah satu prinsip dasar dalam Object-Oriented Programming (OOP) yang memungkinkan satu interface (seperti method atau operasi) untuk digunakan oleh berbagai objek dengan cara yang berbeda. Dalam konteks OOP, polymorphism memungkinkan kita untuk menulis kode yang lebih fleksibel dan modular.

II. ALAT DAN BAHAN

- Visual Studio Code
- Chrome
- Laptop

III. PENJELASAN CODE

Dalam kode ini, polymorphis digunakan dengan cara mendefinisikan beberapa kelas turunan yang memiliki metode yang sama, tetapi memberikan implementasi yang berbeda. polymorphis memungkinkan objek dari kelas turunan untuk diperlakukan sebagai objek dari kelas induknya, namun mereka dapat memiliki perilaku yang berbeda sesuai dengan implementasi masing-masing kelas.

Kelas Kapal adalah kelas induk yang mendefinisikan dua metode: infoKapal() dan aktivitasUtama(). Setiap kelas turunan, seperti KapalPenumpang, KapalBarang, KapalPesiar, dan lainnya, mewarisi kelas Kapal dan meng-override metode-metode ini untuk memberikan perilaku yang spesifik sesuai dengan jenis kapal tersebut.

- Kelas Induk: Kapal

Kelas ini mendefinisikan properti dasar sebuah kapal, seperti nama, jenis, panjang, dan lebar. Kelas ini juga memiliki dua method:

infoKapal(): Mengembalikan informasi umum tentang kapal, seperti nama, jenis, dan ukuran.

aktivitasUtama(): Mengembalikan informasi umum tentang aktivitas kapal, yaitu berlayar di laut.

Kelas Kapal berfungsi sebagai kelas induk yang akan diwarisi oleh berbagai jenis kapal dengan perilaku yang lebih spesifik.

- Subclass 1: KapalPenumpang

Properti tambahan: kapasitasPenumpang, yang menyimpan jumlah penumpang yang bisa diangkut kapal.

Overriding infoKapal(): Method ini menambahkan informasi tambahan, yaitu kapasitas penumpang, selain informasi umum dari kelas induk.

Overriding aktivitasUtama(): Method ini menggambarkan aktivitas utama dari kapal penumpang, yaitu mengangkut penumpang.

- Subclass 2: KapalBarang

Properti tambahan: kapasitasBarang, yang menyimpan kapasitas barang dalam ton.

Overriding infoKapal(): Menambahkan informasi kapasitas barang selain informasi umum dari kapal.

Overriding aktivitasUtama(): Menggambarkan aktivitas kapal barang, yaitu mengangkut muatan barang.

- Subclass 3: KapalPesiar

Properti tambahan: fasilitas, yang berisi daftar fasilitas mewah yang tersedia di kapal pesiar.

Overriding infoKapal(): Menambahkan informasi tentang fasilitas yang dimiliki kapal pesiar.

Overriding aktivitasUtama(): Menggambarkan aktivitas utama kapal pesiar, yaitu memberikan layanan fasilitas mewah kepada penumpang.

- Subclass 4: KapalNelayan

Properti tambahan: kapasitasIkan, yang menyimpan kapasitas maksimal ikan yang dapat dibawa oleh kapal.

Overriding infoKapal(): Menambahkan informasi tentang fungsi kapal untuk menangkap ikan.

Overriding aktivitasUtama(): Menggambarkan aktivitas kapal nelayan, yaitu menangkap ikan dengan kapasitas tertentu.

- Subclass 5: KapalSelam

Properti tambahan: kedalamanOperasi, yang menyimpan informasi mengenai kedalaman maksimum kapal selam bisa beroperasi.

Overriding infoKapal(): Menambahkan informasi tentang kedalaman operasi kapal selam.

Overriding aktivitasUtama(): Menggambarkan aktivitas kapal selam, yaitu beroperasi di bawah laut pada kedalaman tertentu.

- Subclass 6: KapalTanker

Properti tambahan: kapasitasMinyak, yang menyimpan informasi mengenai kapasitas minyak.

Overriding infoKapal(): Menambahkan informasi tentang Kapasitas minyak.

Overriding aktivitasUtama(): Menggambarkan aktivitas kapal Tanker, yaitu sedang mengangkut berapa liter minyak.

- Subclass 7: KapalPatroli

Properti tambahan: kecepatanMaksimal, yang menyimpan informasi mengenai kecepatan maksimal kapal.

Overriding infoKapal(): Menambahkan informasi tentang kapal dapat melaju dengan kecepatan maksimal.

Overriding aktivitasUtama(): Menggambarkan aktivitas kapal patroli, yaitu berpatroli dengan kecepatan maksimal.

- Subclass 8: KapalTunda

Properti tambahan: dayaTarik, yang menyimpan informasi mengenai daya Tarik kapal.

Overriding infoKapal(): Menambahkan informasi tentang kapal memiliki daya Tarik sebesar(Ton).

Overriding aktivitasUtama(): Menggambarkan aktivitas kapal Tunda, yaitu sedang menarik kapal lain dengan daya Tarik (Ton).

- Subclass 9: KapalLaut

Properti tambahan: kedalamanOperasi, yang menyimpan informasi mengenai Jarak tempuh kapal.

Overriding infoKapal(): Menambahkan informasi tentang kapal mampu menempuh jarak sejauh.

Overriding aktivitasUtama(): Menggambarkan aktivitas kapal Laut, yaitu kapal sedang berjalan sejauh.

Polymorphism memungkinkan kita untuk memanggil method infoKapal() dan aktivitasUtama() pada objek-objek dari kelas yang berbeda, tetapi setiap kelas akan memberikan implementasi yang berbeda.

Polymorphism terlihat ketika kita menggunakan loop untuk mengakses array daftarKapal. Setiap objek kapal yang berbeda, meskipun berasal dari kelas yang berbeda (seperti KapalPenumpang, KapalBarang, dll.), diperlakukan sebagai objek dari kelas Kapal. Namun, saat metode infoKapal() atau aktivitasUtama() dipanggil, setiap objek memberikan hasil yang berbeda sesuai dengan kelas spesifiknya.

Jadi, dalam polymorphism, metode yang sama dipanggil pada objek yang berbeda, namun hasilnya akan bervariasi sesuai dengan kelas turunan masing-masing. Ini memungkinkan kode untuk lebih fleksibel dan dinamis, karena kita bisa menggunakan objek-objek yang berbeda dalam satu struktur yang sama, tetapi mereka tetap mempertahankan perilaku spesifiknya masing-masing.