

**Laporan Praktikum**  
**Mata Kuliah Pemrograman 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**

### **1. 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.

### **2. 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.

### **3. 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.

### **4. 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.

#### 5. 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.

#### 6. 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.

#### 7. Polymorphism

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