

Laporan Praktikum
Mata Kuliah Pemrograman Berorientasi Objek



Pertemuan 5
"Polymorphism"

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

Disusun Oleh :
Dini Dwi Andini
2308802

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

Pertama, kita buat sebuah kelas utama bernama Kapal. Kelas ini memiliki atribut dasar seperti nama, panjang, dan lebar, serta dua metode yaitu berlayar dan berhenti. Setelah itu, kita buat beberapa subclass atau turunan dari kelas Kapal ini, yaitu KapalPenumpang, KapalKargo, KapalIkan, dan KapalPerang. Setiap subclass ini akan memiliki metode yang sama seperti di kelas utama (berlayar dan berhenti), tetapi dengan implementasi yang berbeda-beda, sesuai dengan jenis kapalnya.

1. Kelas `Kapal`:

- Ini adalah kelas utama yang mencakup atribut umum dari sebuah kapal, seperti `nama`, `panjang`, dan `lebar`.
- Kelas ini juga memiliki dua metode, yaitu `berlayar` (untuk menunjukkan aktivitas kapal saat berlayar) dan `berhenti` (untuk menunjukkan aktivitas saat kapal berhenti di pelabuhan).

2. Subclass yang Dibuat:

- Kita membuat empat subclass yaitu:
 - `KapalPenumpang`: Subclass ini merepresentasikan kapal penumpang. Pada metode `berlayar` dan `berhenti`, ada perubahan pesan yang menjelaskan bahwa kapal membawa penumpang.
 - `KapalKargo`: Subclass ini merepresentasikan kapal yang membawa barang atau muatan. Metode `berlayar` dan `berhenti` disesuaikan dengan aktivitas pengangkutan muatan.
 - `KapalIkan`: Subclass ini adalah untuk kapal ikan yang fokus pada aktivitas menangkap ikan dengan jaring. Metode yang sama digunakan, tapi pesan berbeda sesuai konteksnya.
 - `KapalPerang`: Subclass ini merepresentasikan kapal perang dengan persenjataan. Metodenya juga diubah untuk mencerminkan operasional kapal perang.

3. Fungsi `aktivitasKapal`:

- Fungsi ini digunakan untuk menguji polymorphism. Fungsi ini menerima parameter berupa objek kapal dan memanggil metode `berlayar` serta `berhenti` dari objek tersebut.
- Polymorphism terlihat saat fungsi ini dipanggil dengan objek dari berbagai subclass, karena meskipun metode yang dipanggil sama, hasilnya berbeda tergantung pada jenis kapal yang digunakan.

4. Output:

- Setiap kali fungsi `aktivitasKapal` dipanggil dengan objek kapal dari subclass yang berbeda, output yang dihasilkan menunjukkan perilaku yang berbeda-beda sesuai dengan jenis kapal tersebut.

Dengan cara ini, konsep polymorphism diimplementasikan dalam JavaScript melalui inheritance (pewarisan) dan override metode di subclass, membuat metode yang sama memberikan hasil yang berbeda tergantung pada jenis objek yang menggunakannya.