

Laporan Praktikum
Mata Kuliah Pemrograman Berorientasi Objek (PBO)



Pertemuan 5. Praktikum 5
“Polymorphism”

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

Disusun Oleh:
Putri Wahyuni
2310271

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

I. PENDAHULUAN

Polimorfisme (Polymorphism) dalam pemrograman berorientasi objek (OOP) adalah konsep di mana objek yang berbeda tipe (subclass) dapat diperlakukan atau dipanggil dengan cara yang sama melalui interface yang sama (method yang sama), namun menghasilkan perilaku yang berbeda-beda sesuai dengan tipe objeknya. Polimorfisme memungkinkan sebuah method atau fungsi memproses objek dari kelas yang berbeda secara seragam, tetapi menghasilkan keluaran atau respons yang spesifik tergantung pada objek yang diproses.

II. ALAT DAN BAHAN

- Laptop
- Aplikasi Visual Studio Code
- Google Chrome

III. PENJELASAN

Berikut adalah penjelasan mengenai kodingannya. Kode ini menunjukkan implementasi polimorfisme dengan menggunakan class dan subclass dalam JavaScript. Berikut langkah-langkah yang terjadi:

1. Class Utama: Sensor

Sensor adalah kelas utama yang memiliki properti dasar berupa nama, lokasi, dan status sensor (misalnya, aktif atau mati).

Ada tiga method di dalamnya:

aktifkan(): Mengubah status sensor menjadi "Aktif".

nonaktifkan(): Mengubah status sensor menjadi "Mati".

getStatus(): Mengembalikan informasi status sensor, termasuk nama, lokasi, dan kondisi saat ini.

2. Subclass: SensorSuhu

Kelas ini mewarisi kelas Sensor dan menambahkan properti tambahan yaitu suhu.

Method tambahan:

ukurSuhu(suhuBaru): Untuk memperbarui nilai suhu yang diukur oleh sensor.

getStatus(): Override (menimpa) method dari kelas induk, menampilkan informasi tambahan tentang suhu saat ini selain status umum sensor.

3. Subclass: SensorKetinggianAir

Ini adalah subclass yang mengukur ketinggian air. Properti tambahan adalah ketinggianAir.

Method tambahan:

ukurKetinggianAir(ketinggianBaru): Memperbarui nilai ketinggian air yang diukur.

getStatus(): Menampilkan status umum sensor sekaligus ketinggian air yang diukur.

4. Subclass: SensorArus

Mengukur kecepatan arus laut. Properti tambahan adalah kecepatanArus.

Method tambahan:

- ukurArus(kecepatanBaru): Memperbarui kecepatan arus yang diukur.
getStatus(): Menampilkan status sensor beserta kecepatan arus saat ini.
5. Subclass: SensorSalinitas
Mengukur salinitas (kadar garam dalam air). Properti tambahan adalah salinitas.
Method tambahan:
ukurSalinitas(salinitasBaru): Memperbarui nilai salinitas yang diukur.
getStatus(): Menampilkan status sensor sekaligus nilai salinitas yang diukur.
6. Subclass: SensorTurbiditas
Mengukur turbiditas atau kekeruhan air. Properti tambahan adalah turbiditas.
Method tambahan:
ukurTurbiditas(turbiditasBaru): Memperbarui nilai turbiditas yang diukur.
getStatus(): Menampilkan status sensor sekaligus nilai turbiditas saat ini.
7. Subclass: SensorTekananAir
Mengukur tekanan air. Properti tambahan adalah tekananAir.
Method tambahan:
ukurTekanan(tekananBaru): Memperbarui nilai tekanan air yang diukur.
getStatus(): Menampilkan status sensor sekaligus nilai tekanan air saat ini.
8. Polimorfisme
tampilkanStatusSensor(sensor): Ini adalah fungsi yang memanfaatkan polimorfisme. Meskipun jenis sensornya berbeda-beda (subclass yang berbeda), fungsi ini memanggil method getStatus() dari objek sensor apapun yang diberikan kepadanya.
Semua sensor (suhu, ketinggian air, arus, dll.) dapat diproses dengan cara yang sama meskipun mereka berasal dari subclass yang berbeda. Metode getStatus() dari masing-masing subclass akan menampilkan informasi spesifik sesuai dengan jenis sensor tersebut.
9. Contoh Penggunaan Polimorfisme
Pada bagian akhir kode, beberapa objek sensor dengan jenis yang berbeda-beda dibuat (misalnya, sensorSuhu, sensorKetinggianAir, dll.). Kemudian, dengan memanfaatkan polimorfisme, method getStatus() dari masing-masing sensor dipanggil menggunakan fungsi tampilkanStatusSensor(). Meskipun setiap sensor merupakan subclass yang berbeda, metode yang sama dapat digunakan karena mereka berbagi struktur dasar dari kelas Sensor.

IV. KESIMPULAN

Kode ini mendemonstrasikan konsep pewarisan, di mana subclass mewarisi properti dan method dari class induk (Sensor), serta polimorfisme, di mana satu fungsi (tampilkanStatusSensor()) dapat digunakan untuk menangani berbagai subclass dengan cara yang seragam namun memberikan keluaran yang berbeda sesuai dengan tipe objeknya.