

Laporan Praktikum
Mata Kuliah Pemrograman Berorientasi Objek



Tugas 5
“Polymorphism”

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

Disusun Oleh :
Sattar Putra Hervia
(2309675)

PROGRAM STUDI SISTEM INFORMASI KELAUTAN UNIVERSITAS
PENDIDIKAN INDONESIA
2024

I. PENDAHULUAN

Pemrograman berorientasi objek (OOP) adalah salah satu paradigma yang sangat penting dalam pengembangan perangkat lunak modern. OOP memungkinkan kita untuk membuat program yang lebih modular dan mudah dipelihara dengan cara memisahkan logika ke dalam objek-objek. Dalam laporan ini, kita akan menerapkan konsep OOP dengan menggunakan class dan inheritance untuk membuat program simulasi mobil. Class yang dibuat akan mencakup berbagai jenis mobil dengan menggunakan konsep turunan (*inheritance*) dan polimorfisme.

II. ALAT DAN BAHAN

- Perangkat lunak: Code editor seperti Visual Studio Code
- Bahasa pemrograman: JavaScript
- Komputer atau laptop: Untuk menjalankan dan menguji kode program
- Browser untuk menjalankan JavaScript

III. HASIL DAN PEMBAHASAN

a. Struktur program

Program dimulai dengan membuat class dasar bernama Mobil yang memiliki beberapa properti umum seperti nama mobil, merk, tahun pembuatan, dan jenis mobil. Kemudian dibuat beberapa class turunan yang mewakili jenis mobil tertentu seperti mobil SUV, mobil balap, mobil listrik, dan mobil pickup. Setiap class turunan memiliki karakteristik tambahan yang sesuai dengan tipe mobil tersebut.

Class utama:

```
1 class Mobil {
2   constructor(nama, merk, tahun, jenis) {
3     this.nama = nama;
4     this.merk = merk;
5     this.tahun = tahun;
6     this.jenis = jenis;
7   }
8
9   infoMobil() {
10    return `Mobil ${this.nama} adalah merk ${this.merk}, diproduksi pada tahun ${this.tahun}, dengan jenis ${this.jenis}`;
11  }
12 }
13
```

Class turunan:

- MobilSUV: menambahkan property penggerakRoda
- MobilBalap: menambahkan property kecepatanMaks
- MobilListrik: menambahkan property kapasitasBaterai
- MobilPickUp: menambahkan property kapasitasAngkut

contoh penggunaan class turunan

```
class MobilSUV extends Mobil {  
    constructor(nama, merk, tahun, jenis, penggerakRoda) {  
        super(nama, merk, tahun, jenis);  
        this.penggerakRoda = penggerakRoda;  
    }  
  
    infoMobil() {  
        return `${super.infoMobil()} Mobil ini menggunakan penggerak roda ${this.penggerakRoda}.`;  
    }  
}
```

b. Inheritance dan Polimorfisme

Inheritance memungkinkan kita untuk mewarisi properti dan metode dari class Mobil ke class turunan seperti MobilSUV atau MobilBalap. Dengan begitu, class turunan tetap bisa menggunakan metode infoMobil() dari class induk, namun juga dapat menambahkan atau mengubah informasi spesifik sesuai dengan kebutuhan. Dalam contoh ini, subclass bisa menambahkan atribut tambahan seperti kecepatan maksimum pada MobilBalap atau kapasitas baterai pada MobilListrik.

Polimorfisme diterapkan dengan menggunakan metode yang sama di setiap subclass, yaitu infoMobil(), tetapi menghasilkan output yang berbeda tergantung pada jenis objeknya. Hal ini memungkinkan program menampilkan informasi yang sesuai tanpa harus membuat kode terpisah untuk setiap tipe mobil.

c. Pengujian Program

Setelah mendefinisikan class dan subclass, kita membuat beberapa objek mobil dari setiap jenis dan menyimpannya dalam sebuah array. Kemudian, objek-objek tersebut diakses menggunakan perulangan, dan program akan memanggil metode infoMobil() untuk setiap mobil, menampilkan informasi lengkap sesuai dengan spesifikasinya.

Contoh:

```
// Polimorfisme
const mobilSedan = new Mobil("Sedona", "Toyota", 2020, "sedan");
const mobilSUV1 = new MobilSUV("Ranger", "Ford", 2021, "SUV", "4x4");
const mobilBalap1 = new MobilBalap("Speedster", "Ferrari", 2022, "balap", 300);
const mobilListrik1 = new MobilListrik("Volt", "Tesla", 2023, "listrik", 85000);
const mobilPickUp1 = new MobilPickUp("Truckster", "Isuzu", 2020, "pickup", 2);

const mobilList = [mobilSedan, mobilSUV1, mobilBalap1, mobilListrik1, mobilPickUp1];

// Menampilkan informasi dengan polimorfisme
mobilList.forEach(mobil => {
    console.log(mobil.infoMobil());
});
```

Hasil dari pengujian program menunjukkan bahwa setiap objek mobil dapat menampilkan informasi yang lengkap dan sesuai dengan atribut yang dimiliki masing-masing tipe mobil.

d. Output

- Mobil Sedan: "Sedona", Toyota, 2020
- Mobil SUV: "Ranger", Ford, 2021, Penggerak roda 4x4
- Mobil Balap: "Speedster", Ferrari, 2022, Kecepatan maksimum 300 km/jam
- Mobil Listrik: "Volt", Tesla, 2023, Kapasitas baterai 85000 mAh
- Mobil PickUp: "Truckster", Isuzu, 2020, Kapasitas angkut 2 ton

Setiap mobil dapat menampilkan informasi lengkap sesuai tipe dan atribut yang dimilikinya.

IV. KESIMPULAN

Berdasarkan praktikum ini, dapat disimpulkan bahwa konsep OOP, khususnya *inheritance* dan *polimorfisme*, sangat bermanfaat dalam pembuatan program yang lebih terstruktur dan mudah dipelihara. Dengan menggunakan class induk Mobil, kita dapat membuat class turunan yang lebih spesifik tanpa harus menulis ulang kode yang sama. Konsep polimorfisme juga mempermudah kita dalam memanggil metode yang sama pada objek-objek yang berbeda, menghasilkan keluaran yang sesuai dengan atribut masing-masing.