

Laporan Praktikum
Mata Kuliah Pemograman Berorientasi Objek



Pertemuan 5 Tugas ke 4
“polymorphysm”

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

Disusun Oleh :
Tegar Dzaki Hakim (2306110)

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA

2024

Tujuan

Memahami dan mengimplementasikan konsep **polymorphism** menggunakan **class** pada JavaScript dengan membuat superclass dan subclass yang menerapkan metode yang sama namun menghasilkan output berbeda.

Dasar Teori

- **Polymorphism** : Merupakan konsep dalam pemrograman berorientasi objek yang memungkinkan objek berbeda untuk merespons terhadap metode yang sama namun dengan perilaku yang berbeda.
- **Inheritance (Pewarisan)**: Konsep dalam OOP di mana satu **class** (subclass/child class) dapat mewarisi properti dan metode dari **class** lainnya (superclass/parent class).
- **Class dan Object**: Class adalah template atau cetak biru untuk membuat objek. Objek adalah instance dari class.

Alat dan Bahan

- Laptop atau komputer dengan sistem operasi yang mendukung.
- Browser atau editor teks (seperti VS Code) yang mendukung JavaScript.
- Node.js (opsional untuk menjalankan JavaScript di luar browser).

Langkah-Langkah Percobaan

1. Buat Superclass dan Subclass:

Kita membuat superclass sebagai dasar yang nantinya akan diwariskan oleh subclass. Subclass dapat meng-override metode yang ada di superclass untuk menampilkan perilaku yang berbeda, sehingga menunjukkan polymorphism.

2. Konstruktor Superclass dan Subclass:

Setiap superclass memiliki sebuah konstruktor yang menerima beberapa parameter untuk mendefinisikan atribut dasar. Subclass akan memanggil konstruktor superclass menggunakan `super()` dan menambahkan properti spesifik milik subclass tersebut.

3. Penerapan Polymorphism:

Dalam metode `deskripsi` pada setiap superclass dan subclass, kita menunjukkan bagaimana polymorphism bekerja. Ketika metode yang sama dipanggil pada objek yang berbeda (instance dari subclass), hasil yang diberikan akan berbeda sesuai dengan subclass tersebut.

6. Hasil Percobaan dan Kode Program:

Berikut adalah kode JavaScript lengkap dengan subclass dan metode yang menggunakan polymorphism:

```
```javascript
// Superclass Bendera
class Bendera {
 constructor(namaNegara, warnaUtama) {
 this.namaNegara = namaNegara;
 this.warnaUtama = warnaUtama;
 }
}
```

```

 }

 deskripsiBendera() {
 return `Bendera negara ${this.namaNegara} memiliki warna utama
 ${this.warnaUtama}.`;
 }
}

// Subclass Bendera
class BenderaIndonesia extends Bendera {
 constructor() {
 super("Indonesia", "Merah Putih");
 }

 deskripsiBendera() {
 return `${super.deskripsiBendera()} Ini adalah bendera dari Indonesia.`;
 }
}

class BenderaPerancis extends Bendera {
 constructor() {
 super("Perancis", "Biru Putih Merah");
 }

 deskripsiBendera() {
 return `${super.deskripsiBendera()} Ini adalah bendera dari Perancis.`;
 }
}

// Superclass Kendaraan
class Kendaraan {

```

```
 constructor(jenis, kecepatanMaks) {
 this.jenis = jenis;
 this.kecepatanMaks = kecepatanMaks;
 }

 deskripsiKendaraan() {
 return `Ini adalah kendaraan jenis ${this.jenis} dengan kecepatan maksimum
 ${this.kecepatanMaks} km/jam.`;
 }
}

// Subclass Kendaraan
class Mobil extends Kendaraan {
 constructor() {
 super("Mobil", 180);
 }

 deskripsiKendaraan() {
 return `${super.deskripsiKendaraan()} Kendaraan ini beroda empat.`;
 }
}

class Motor extends Kendaraan {
 constructor() {
 super("Motor", 120);
 }

 deskripsiKendaraan() {
 return `${super.deskripsiKendaraan()} Kendaraan ini beroda dua.`;
 }
}
```

// Superclass Makanan

```
class Makanan {
 constructor(nama, rasa) {
 this.nama = nama;
 this.rasa = rasa;
 }

 deskripsiMakanan() {
 return `Makanan ini adalah ${this.nama} yang memiliki rasa ${this.rasa}.`;
 }
}
```

// Subclass Makanan

```
class NasiGoreng extends Makanan {
 constructor() {
 super("Nasi Goreng", "Gurih");
 }

 deskripsiMakanan() {
 return `${super.deskripsiMakanan()} Makanan ini populer di Indonesia.`;
 }
}
```

```
class Sushi extends Makanan {
 constructor() {
 super("Sushi", "Asin dan Asam");
 }

 deskripsiMakanan() {
 return `${super.deskripsiMakanan()} Makanan ini berasal dari Jepang.`;
 }
}
```

```
 }
}
```

```
// Superclass JenisKelamin
```

```
class JenisKelamin {
 constructor(kelamin) {
 this.kelamin = kelamin;
 }

 deskripsiJenisKelamin() {
 return `Jenis kelamin ini adalah ${this.kelamin}.`;
 }
}
```

```
// Subclass JenisKelamin
```

```
class Pria extends JenisKelamin {
 constructor() {
 super("Pria");
 }

 deskripsiJenisKelamin() {
 return `${super.deskripsiJenisKelamin()} Biasanya memiliki peran laki-laki.`;
 }
}
```

```
class Wanita extends JenisKelamin {
```

```
 constructor() {
 super("Wanita");
 }

 deskripsiJenisKelamin() {
```

```
 return `${super.deskripsiJenisKelamin()} Biasanya memiliki peran perempuan.`;
 }
}
```

// Superclass HewanPeliharaan

```
class HewanPeliharaan {
 constructor(nama, suara) {
 this.nama = nama;
 this.suara = suara;
 }

 deskripsiHewanPeliharaan() {
 return `Hewan peliharaan ini adalah ${this.nama} dan mengeluarkan suara ${this.suara}.`;
 }
}
```

// Subclass HewanPeliharaan

```
class Kucing extends HewanPeliharaan {
 constructor() {
 super("Kucing", "Meong");
 }

 deskripsiHewanPeliharaan() {
 return `${super.deskripsiHewanPeliharaan()} Kucing sering dipelihara di rumah.`;
 }
}
```

class Anjing extends HewanPeliharaan {

```
 constructor() {
 super("Anjing", "Guk Guk");
 }
}
```

```
}

deskripsiHewanPeliharaan() {
 return `${super.deskripsiHewanPeliharaan()} Anjing sering dijadikan penjaga rumah.`;
}
}
```

// Polymorphism in Action

// Bendera

```
let indonesia = new BenderaIndonesia();
let perancis = new BenderaPerancis();
console.log(indonesia.deskripsiBendera());
console.log(perancis.deskripsiBendera());
```

// Kendaraan

```
let mobil = new Mobil();
let motor = new Motor();
console.log(mobil.deskripsiKendaraan());
console.log(motor.deskripsiKendaraan());
```

// Makanan

```
let nasiGoreng = new NasiGoreng();
let sushi = new Sushi();
console.log(nasiGoreng.deskripsiMakanan());
console.log(sushi.deskripsiMakanan());
```

// Jenis Kelamin

```
let pria = new Pria();
let wanita = new Wanita();
console.log(pria.deskripsiJenisKelamin());
```



```
console.log(wanita.deskripsiJenisKelamin());

// Hewan Peliharaan
let kucing = new Kucing();
let anjing = new Anjing();
console.log(kucing.deskripsiHewanPeliharaan());
console.log(anjing.deskripsiHewanPeliharaan());
...
```

## 7. Penjelasan Kode:

- Superclass `Bendera`: Memiliki metode `deskripsiBendera()` yang memberikan deskripsi tentang bendera dari suatu negara. Subclass `BenderaIndonesia` dan `BenderaPerancis` mewarisi metode ini dan menambahkan detail lebih lanjut untuk masing-masing negara.
- Superclass `Kendaraan`: Memiliki metode `deskripsiKendaraan()`, yang memberikan deskripsi dasar tentang kendaraan. Subclass `Mobil` dan `Motor` menambahkan detail spesifik tentang jenis kendaraan.
- Superclass `Makanan`: Subclass `NasiGoreng` dan `Sushi` menambahkan informasi tentang makanan yang berasal dari Indonesia dan Jepang, mewarisi metode `deskripsiMakanan()` dari superclass.
- Superclass `JenisKelamin`: Subclass `Pria` dan `Wanita` menampilkan informasi tentang jenis kelamin dengan menambahkan deskripsi dari superclass `deskripsiJenisKelamin()`.
- Superclass `HewanPeliharaan`: Subclass `Kucing` dan `Anjing` menambahkan deskripsi spesifik terkait suara dan peran dari masing-masing hewan peliharaan.

## Kesimpulan

Konsep polymorphism memungkinkan metode yang sama digunakan pada berbagai objek, namun hasil yang dikeluarkan berbeda sesuai dengan class yang memanggil metode tersebut. Dalam contoh di atas, metode seperti `deskripsiBendera`, `deskripsiKendaraan`, dll., memiliki perilaku yang berbeda saat dipanggil oleh subclass yang berbeda, meskipun nama metodenya sama.