

Laporan Praktikum
Mata Pemrograman Berorientasi Objek



Pertemuan 5 Tugas 4
“Polimorphysm”

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

Disusun Oleh :
Muhammad Ottmar Makhtar (2306110)

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

I. Pendahuluan

Praktikum ini bertujuan untuk mengaplikasikan teori yang telah dipelajari di dalam kelas melalui Praktikum dan Teori. Dengan melibatkan mahasiswa dalam proses praktis, diharapkan pemahaman terhadap materi dapat diperkuat. Pada praktikum kali ini, fokus penelitian adalah teori polimorphism dan mengimplementasikan di Visual Studio Code, yang merupakan salah satu konsep dasar yang penting untuk dikuasai. Melalui praktikum ini, mahasiswa diharapkan dapat memahami prinsip-prinsip dasar Pemrograman Berorientasi Objek, serta mampu menerapkan metode yang tepat dalam polimorphism.

II. Alat dan Bahan

1. Alat

- Laptop
- Mouse

2. Bahan

- Materi Polimorphism

III. Langkah Praktikum

1. Melakukan Digitasi di ArcGIS Pro

```
//Class1 Bendera
class bendera {
    constructor (nama) {
        this.nama = nama
    }
    warnaDanbentuk(){
        console.log(`Bendera ${this.nama} memiliki warna dan bentuk`)
    }
}
```

1. **bendera** adalah superclass yang memiliki **constructor** yang menerima parameter nama dan menyimpannya dalam properti `this.nama`. Ini digunakan untuk menyimpan nama bendera.
2. Metode **warnaDanbentuk()** di kelas **bendera** hanya mencetak teks umum: "Bendera `${this.nama}` memiliki warna dan bentuk". Metode ini akan di-override oleh subclass.

```
class indonesia extends bendera {
  warnaDanbentuk() {
    console.log(`Bendera ${this.nama} memiliki warna merah dan putih. bentuknya garis horizontal` )
  }
}
```

3. **indonesia** adalah subclass yang mewarisi dari superclass bendera.
4. Metode **warnaDanbentuk()** di sini meng-override metode dari superclass bendera. Pada subclass indonesia, implementasinya lebih spesifik: mencetak informasi tentang bendera Indonesia, yaitu warnanya merah dan putih serta bentuknya berupa garis horizontal.

```
class jerman extends bendera {
  warnaDanbentuk() {
    console.log(`Bendera ${this.nama} memiliki warna hitam, merah, dan kuning. bentuknya bentuknya garis horizontal`)
  }
}
```

5. **jerman** juga merupakan subclass yang mewarisi dari superclass bendera.
6. Seperti pada subclass indonesia, metode **warnaDanbentuk()** di-override untuk memberikan informasi spesifik tentang bendera Jerman, yaitu warnanya hitam, merah, dan kuning, serta bentuknya juga berupa garis horizontal.

```
let Bendera = [new indonesia('indonesia'), new jerman('jerman')]
Bendera.forEach(bendera => bendera.warnaDanbentuk())
```

7. Array **Bendera** berisi dua objek: satu dari class indonesia dan satu lagi dari class jerman. Keduanya adalah instance dari subclass yang mewarisi dari superclass bendera.
8. **Bendera.forEach(bendera => bendera.warnaDanbentuk())**: Setiap objek bendera dalam array ini akan memanggil metode warnaDanbentuk(). Karena indonesia dan jerman memiliki implementasi metode warnaDanbentuk() masing-masing, hasil yang dicetak akan sesuai dengan subclass mereka.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS
[Running] node "c:\Games\UPI Kampus Serang\Tugas\Semester 3\PBO\PBO Pertemuan 4\latihan1.js"
Bendera indonesia memiliki warna merah dan putih. bentuknya garis horizontal
Bendera jerman memiliki warna hitam, merah, dan kuning. bentuknya bentuknya garis horizontal
```

9. **Inheritance:** Kelas `indonesia` dan `jerman` mewarisi dari kelas `bendera`, sehingga mereka memiliki akses ke properti dan metode yang ada di kelas `bendera`.
10. **Polymorphism:** Meskipun metode `warnaDanbentuk()` dipanggil dengan nama yang sama, hasilnya berbeda tergantung pada subclass yang dipanggil. Subclass `indonesia` memberikan deskripsi tentang bendera Indonesia, dan subclass `jerman` memberikan deskripsi tentang bendera Jerman. Klik "Finish"

IV. Kesimpulan

Ini adalah contoh yang baik dari bagaimana `polymorphism` memungkinkan satu metode dipanggil untuk objek yang berbeda, namun hasilnya disesuaikan dengan implementasi yang spesifik dari subclass masing-masing.