

**Laporan Praktikum**  
**“Mata Kuliah Pemrograman Berorientasi Objek”**



**Pertemuan 5**  
**“POLYMORPHISM”**

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

**Disusun Oleh :**  
M. FAHRI  
(2308465)

**PROGRAM STUDI SISTEM INFORMASI KELAUTAN**  
**UNIVERSITAS PENDIDIKAN INDONESIA**  
**2024**

## **I. Pendahuluan**

Pada praktikum ini, saya membuat program sistem kapal yang dapat menampilkan informasi tentang tiket, barang, tempat pendaratan, pelabuhan, dan penumpang. Sistem ini dibuat menggunakan HTML, CSS, dan JavaScript

## **II. Tujuan**

Tujuan dari praktikum ini adalah untuk membuat sistem kapal yang dapat menampilkan informasi tentang tiket, barang, tempat pendaratan, pelabuhan, dan penumpang. ikasi ArcGISPro

## **III. Metode dan penjelasan code**

Pada praktikum ini, kami menggunakan metode pembuatan sistem kapal dengan menggunakan HTML, CSS, dan JavaScript. Kami membuat file HTML untuk menampilkan informasi, file CSS untuk mengatur tampilan, dan file JavaScript untuk membuat sistem kapal dapat berinteraksi dengan pengguna.

### **HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Sistem Kapal</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" href="style.css">
</head>
```

*Dokumen HTML ini adalah deklarasi awal dari sebuah sistem kapal yang menggunakan bahasa Inggris, karakter UTF-8, dan lebar layar perangkat yang sesuai, serta menghubungkan file CSS Bootstrap dan file CSS yang dibuat sendiri untuk mengatur tampilan dan gaya dokumen.*

```
<body>
```

```
<div class="container py-5">
```

```
<div class="header text-center mb-4">
```

```
<h1 class="display-4">Sistem Kapal</h1>
```

*ini memuat bagian header yang terdiri dari judul "Sistem Kapal" dengan ukuran font besar dan berat font bold, yang ditempatkan di dalam sebuah div dengan kelas "container" dan "py-5" yang berfungsi sebagai wadah untuk isi dokumen.*

```
</div>
```

```
<div class="row text-center">
```

```
<div class="col-md-6 mb-3">
```

```
<button class="btn btn-primary w-100" id="btn-tiket">Tampilkan Info  
Tiket</button>
```

```
</div>
```

```
<div class="col-md-6 mb-3">
```

```
<button class="btn btn-primary w-100" id="btn-barang">Tampilkan Info  
Barang</button>
```

```
</div>
```

```
<div class="col-md-6 mb-3">
```

```
<button class="btn btn-primary w-100" id="btn-tempat-endaratan">Tampilkan Info  
Tempat Pendaratan</button>
```

```
</div>
```

```
<div class="col-md-6 mb-3">
```

```
<button class="btn btn-primary w-100" id="btn-pelabuhan">Tampilkan Info  
Pelabuhan</button>
```

```
</div>
```

*ini memuat empat tombol dengan kelas "btn btn-primary" yang berfungsi untuk menampilkan informasi tiket, barang, tempat pendaratan, dan pelabuhan, yang ditempatkan di dalam sebuah div dengan kelas "row" dan "text-center" yang berfungsi sebagai wadah untuk tombol-tombol.*

```
<div class="col-12 d-flex justify-content-center mb-3">
```

```
<button class="btn btn-primary" id="btn-penumpang">Tampilkan Info  
Penumpang</button>
```

```
</div>
</div>
<div id="hasil" class="mt-4"></div>
</div>
```

*ini memuat sebuah tombol dengan kelas "btn btn-primary" yang berfungsi untuk menampilkan informasi penumpang, yang ditempatkan di dalam sebuah div dengan kelas "col-12", "d-flex", "justify-content-center", dan "mb-3" yang berfungsi sebagai wadah untuk tombol, dan diikuti oleh sebuah div dengan id "hasil" yang berfungsi sebagai wadah untuk menampilkan hasil.*

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"></script>
<script src="kelas.js"></script>
</body>
</html>
```

*ini diakhiri dengan dua tag script yang menghubungkan file JavaScript Bootstrap dan file JavaScript yang dibuat sendiri, yang berfungsi untuk menambahkan fungsionalitas dan interaktivitas pada halaman web, dan diakhiri dengan tag </body> dan </html> yang menandai akhir dari dokumen HTML.*

## JavaScript

```
class Kapal {
    constructor(namaKapal, jenisKapal) {
        this.namaKapal = namaKapal;
        this.jenisKapal = jenisKapal;
    }

    tampilkanInfo() {
        return `Nama Kapal: ${this.namaKapal}, Jenis Kapal: ${this.jenisKapal}`;
    }
}
```

```
tampilkanInfoDetail() {  
    throw new Error("Method 'tampilkanInfoDetail' harus diimplementasikan");  
}  
}
```

```
class Tiket extends Kapal {  
    constructor(namaKapal, jenisKapal, hargaTiket, jumlahTiket) {  
        super(namaKapal, jenisKapal); // Memanggil constructor dari superclass  
        this.hargaTiket = hargaTiket;  
        this.jumlahTiket = jumlahTiket;  
    }  
}
```

```
tampilkanInfoDetail() {  
    return `${this.tampilkanInfo()}, Harga Tiket: ${this.hargaTiket}, Jumlah Tiket:  
    ${this.jumlahTiket}`;  
}  
}
```

```
class Barang extends Kapal {  
    constructor(namaKapal, jenisKapal, namaBarang, jumlahBarang) {  
        super(namaKapal, jenisKapal);  
        this.namaBarang = namaBarang;  
        this.jumlahBarang = jumlahBarang;  
    }  
}
```

```
tampilkanInfoDetail() {
```

```
        return `${this.tampilkanInfo()}, Nama Barang: ${this.namaBarang}, Jumlah  
Barang: ${this.jumlahBarang}`;  
    }  
}
```

```
class TempatPendaratan extends Kapal {  
    constructor(namaKapal, jenisKapal, namaTempatPendaratan,  
lokasiTempatPendaratan, kapasitasTempatPendaratan) {  
        super(namaKapal, jenisKapal);  
        this.namaTempatPendaratan = namaTempatPendaratan;  
        this.lokasiTempatPendaratan = lokasiTempatPendaratan;  
        this.kapasitasTempatPendaratan = kapasitasTempatPendaratan;  
    }  
  
    tampilkanInfoDetail() {  
        return `${this.tampilkanInfo()}, Nama Tempat Pendaratan:  
${this.namaTempatPendaratan}, Lokasi: ${this.lokasiTempatPendaratan}, Kapasitas:  
${this.kapasitasTempatPendaratan}`;  
    }  
}
```

```
class Pelabuhan extends Kapal {  
    constructor(namaKapal, jenisKapal, namaPelabuhan, lokasiPelabuhan) {  
        super(namaKapal, jenisKapal);  
        this.namaPelabuhan = namaPelabuhan;  
        this.lokasiPelabuhan = lokasiPelabuhan;  
    }  
  
    tampilkanInfoDetail() {
```

```
        return `${this.tampilkanInfo()}, Nama Pelabuhan: ${this.namaPelabuhan}, Lokasi:
        ${this.lokasiPelabuhan}`;
    }
}
```

```
class Penumpang extends Kapal {
```

```
    constructor(namaKapal, jenisKapal, namaPenumpang, umurPenumpang) {
        super(namaKapal, jenisKapal);
        this.namaPenumpang = namaPenumpang;
        this.umurPenumpang = umurPenumpang;
    }
```

```
    tampilkanInfoDetail() {
        return `${this.tampilkanInfo()}, Nama Penumpang: ${this.namaPenumpang},
        Umur Penumpang: ${this.umurPenumpang}`;
    }
}
```

```
document.getElementById('btn-tiket').addEventListener('click', function() {
    let tiket = new Tiket("Kapal A", "Feri", 50000, 200);
    document.getElementById('hasil').innerHTML = tiket.tampilkanInfoDetail();
});
```

```
document.getElementById('btn-barang').addEventListener('click', function() {
    let barang = new Barang("Kapal B", "Kargo", "Pakaian", 1000);
    document.getElementById('hasil').innerHTML = barang.tampilkanInfoDetail();
});
```

```
document.getElementById('btn-tempat-endaratan').addEventListener('click',
function() {
```

```

    let tempatPendaratan = new TempatPendaratan("Kapal C", "Penumpang",
"Pelabuhan Merak", "Banten", 3000);

    document.getElementById('hasil').innerHTML =
tempatPendaratan.tampilkanInfoDetail();

});

document.getElementById('btn-pelabuhan').addEventListener('click', function() {

    let pelabuhan = new Pelabuhan("Kapal D", "Penumpang", "Pelabuhan Tanjung
Priok", "Jakarta");

    document.getElementById('hasil').innerHTML = pelabuhan.tampilkanInfoDetail();

});

document.getElementById('btn-penumpang').addEventListener('click', function() {

    let penumpang = new Penumpang("Kapal E", "Penumpang", "John Doe", 30);

    document.getElementById('hasil').innerHTML = penumpang.tampilkanInfoDetail();

});

```

#### ***Baris 1-5: Kelas Kapal***

- *class Kapal { : Membuat sebuah kelas baru bernama Kapal.*
- *constructor(namaKapal, jenisKapal) { : Membuat sebuah konstruktor untuk kelas Kapal yang menerima dua parameter: namaKapal dan jenisKapal.*
- *this.namaKapal = namaKapal; : Mengatur nilai atribut namaKapal dari kelas Kapal dengan nilai yang diterima dari parameter namaKapal.*
- *this.jenisKapal = jenisKapal; : Mengatur nilai atribut jenisKapal dari kelas Kapal dengan nilai yang diterima dari parameter jenisKapal.*
- *} : Menutup konstruktor.*

#### ***Baris 6-8: Metode tampilkanInfo()***

- *tampilkanInfo() { : Membuat sebuah metode baru bernama tampilkanInfo() pada kelas Kapal.*
- *return Nama Kapal: \${this.namaKapal}, Jenis Kapal: \${this.jenisKapal}; : Mengembalikan sebuah string yang berisi informasi tentang kapal, termasuk namaKapal dan jenisKapal.*
- *} : Menutup metode.*



**Baris 9-11: Metode `tampilkanInfoDetail()`**

- `tampilkanInfoDetail()` { : Membuat sebuah metode baru bernama `tampilkanInfoDetail()` pada kelas `Kapal`.
- `throw new Error("Method 'tampilkanInfoDetail' harus diimplementasikan");` : Membuat sebuah error yang menunjukkan bahwa metode `tampilkanInfoDetail()` harus diimplementasikan pada kelas turunan.
- `}` : Menutup metode.

**Baris 12-18: Kelas `Tiket`**

- `class Tiket extends Kapal` { : Membuat sebuah kelas baru bernama `Tiket` yang merupakan turunan dari kelas `Kapal`.
- `constructor(namaKapal, jenisKapal, hargaTiket, jumlahTiket)` { : Membuat sebuah konstruktor untuk kelas `Tiket` yang menerima empat parameter: `namaKapal`, `jenisKapal`, `hargaTiket`, dan `jumlahTiket`.
- `super(namaKapal, jenisKapal);` : Memanggil konstruktor dari kelas `Kapal` dengan parameter `namaKapal` dan `jenisKapal`.
- `this.hargaTiket = hargaTiket;` : Mengatur nilai atribut `hargaTiket` dari kelas `Tiket` dengan nilai yang diterima dari parameter `hargaTiket`.
- `this.jumlahTiket = jumlahTiket;` : Mengatur nilai atribut `jumlahTiket` dari kelas `Tiket` dengan nilai yang diterima dari parameter `jumlahTiket`.
- `}` : Menutup konstruktor.

**Baris 19-22: Metode `tampilkanInfoDetail()` pada kelas `Tiket`**

- `tampilkanInfoDetail()` { : Membuat sebuah metode baru bernama `tampilkanInfoDetail()` pada kelas `Tiket`.
- `return `${this.tampilkanInfo()}`, Harga Tiket: ${this.hargaTiket}, Jumlah Tiket: ${this.jumlahTiket};` : Mengembalikan sebuah string yang berisi informasi tentang tiket, termasuk `hargaTiket` dan `jumlahTiket`.
- `}` : Menutup metode.

**Baris 23-29: Kelas `Barang`**

- `class Barang extends Kapal` { : Membuat sebuah kelas baru bernama `Barang` yang merupakan turunan dari kelas `Kapal`.
- `constructor(namaKapal, jenisKapal, namaBarang, jumlahBarang)` { : Membuat sebuah konstruktor untuk kelas `Barang` yang menerima empat parameter: `namaKapal`, `jenisKapal`, `namaBarang`, dan `jumlahBarang`.
- `super(namaKapal, jenisKapal);` : Memanggil konstruktor dari kelas `Kapal` dengan parameter `namaKapal` dan `jenisKapal`.

- *this.namaBarang = namaBarang; : Mengatur nilai atribut namaBarang dari kelas Barang dengan nilai yang diterima dari parameter namaBarang.*
- *this.jumlahBarang = jumlahBarang; : Mengatur nilai atribut jumlahBarang dari kelas Barang dengan nilai yang diterima dari parameter jumlahBarang.*
- *} : Menutup konstruktor.*

***\*\*Baris 30-33: Metode tampilkanInfoDetail***

## Style.css

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
}

.container {
    max-width: 800px;
    margin: 40px auto;
    padding: 20px;
    background-color: #fff;
    border: 1px solid #ddd;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.header {
    background-color: #333;
    color: #fff;
    padding: 20px;
    text-align: center;
    border-bottom: 1px solid #ddd;
}
```

```
.header h1 {  
    margin: 0;  
}
```

```
.button-group {  
    margin-top: 20px;  
}
```

```
.button-group button {  
    margin-right: 10px;  
    padding: 10px 20px;  
    border: none;  
    border-radius: 5px;  
    background-color: #4CAF50;  
    color: #fff;  
    cursor: pointer;  
}
```

```
.button-group button:hover {  
    background-color: #3e8e41;  
}
```

```
#hasil {  
    margin-top: 20px;  
    padding: 20px;  
    background-color: #f9f9f9;  
    border: 1px solid #ddd;  
    border-radius: 10px;  
}
```

```
#hasil h2 {  
    margin-top: 0;  
}
```

```
#hasil ul {  
    list-style: none;  
    padding: 0;  
    margin: 0;  
}
```

```
#hasil li {  
    padding: 10px;  
    border-bottom: 1px solid #ddd;  
}
```

```
#hasil li:last-child {  
    border-bottom: none;  
}
```

```
#hasil li:hover {  
    background-color: #f0f0f0;  
}
```

#### ***Baris 1-2: Pengaturan font dan warna latar belakang***

- *body { : Membuat sebuah blok CSS untuk elemen body.*
- *font-family: Arial, sans-serif; : Mengatur font yang digunakan pada elemen body menjadi Arial dan sans-serif.*
- *background-color: #f0f0f0; : Mengatur warna latar belakang elemen body menjadi #f0f0f0.*

#### ***Baris 3-10: Pengaturan container***

- *.container { : Membuat sebuah blok CSS untuk elemen dengan kelas container.*
- *max-width: 800px; : Mengatur lebar maksimum elemen container menjadi 800px.*

- *margin: 40px auto; : Mengatur margin atas dan bawah elemen container menjadi 40px dan margin kiri dan kanan menjadi auto.*
- *padding: 20px; : Mengatur padding elemen container menjadi 20px.*
- *background-color: #fff; : Mengatur warna latar belakang elemen container menjadi #fff.*
- *border: 1px solid #ddd; : Mengatur border elemen container menjadi 1px solid #ddd.*
- *border-radius: 10px; : Mengatur radius border elemen container menjadi 10px.*
- *box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); : Mengatur bayangan elemen container menjadi 0 0 10px rgba(0, 0, 0, 0.1).*

#### **Baris 11-14: Pengaturan header**

- *.header { : Membuat sebuah blok CSS untuk elemen dengan kelas header.*
- *background-color: #333; : Mengatur warna latar belakang elemen header menjadi #333.*
- *color: #fff; : Mengatur warna teks elemen header menjadi #fff.*
- *padding: 20px; : Mengatur padding elemen header menjadi 20px.*
- *text-align: center; : Mengatur teks elemen header menjadi rata tengah.*
- *border-bottom: 1px solid #ddd; : Mengatur border bawah elemen header menjadi 1px solid #ddd.*

#### **Baris 15: Pengaturan judul header**

- *.header h1 { : Membuat sebuah blok CSS untuk elemen h1 di dalam elemen header.*
- *margin: 0; : Mengatur margin elemen h1 menjadi 0.*

#### **Baris 16-20: Pengaturan grup tombol**

- *.button-group { : Membuat sebuah blok CSS untuk elemen dengan kelas button-group.*
- *margin-top: 20px; : Mengatur margin atas elemen button-group menjadi 20px.*

#### **Baris 21-25: Pengaturan tombol**

- *.button-group button { : Membuat sebuah blok CSS untuk elemen button di dalam elemen button-group.*
- *margin-right: 10px; : Mengatur margin kanan elemen button menjadi 10px.*
- *padding: 10px 20px; : Mengatur padding elemen button menjadi 10px 20px.*
- *border: none; : Mengatur border elemen button menjadi none.*
- *border-radius: 5px; : Mengatur radius border elemen button menjadi 5px.*

- *background-color: #4CAF50; : Mengatur warna latar belakang elemen button menjadi #4CAF50.*
- *color: #fff; : Mengatur warna teks elemen button menjadi #fff.*
- *cursor: pointer; : Mengatur cursor elemen button menjadi pointer.*

**Baris 26-27: Pengaturan hover tombol**

- *.button-group button:hover { : Membuat sebuah blok CSS untuk elemen button di dalam elemen button-group ketika dihover.*
- *background-color: #3e8e41; : Mengatur warna latar belakang elemen button menjadi #3e8e41 ketika dihover.*

**Baris 28-34: Pengaturan hasil**

- *#hasil { : Membuat sebuah blok CSS untuk elemen dengan id hasil.*
- *margin-top: 20px; : Mengatur margin atas elemen hasil menjadi 20px.*
- *padding: 20px; : Mengatur padding elemen hasil menjadi 20px.*
- *background-color: #f9f9f9; : Mengatur warna latar belakang elemen*

## **I. Kesimpulan**

Pada praktikum ini, kami berhasil membuat sistem kapal yang dapat menampilkan informasi tentang tiket, barang, tempat pendaratan, pelabuhan, dan penumpang. Sistem ini dapat berinteraksi dengan pengguna dan menampilkan informasi yang relevan. Kami juga belajar tentang cara membuat sistem kapal menggunakan HTML, CSS, dan JavaScript..