

// bikin 4 kelas dengan polymorphism

// kelas pertama, yaitu Kapal

```
class Kapal {  
    constructor(nama, jenis, kapasitas, PT){  
        this.nama = nama,  
        this.jenis = jenis,  
        this.kapasitas = kapasitas,  
        this.PT = PT;  
    }  
    infoKapal(){  
        return `Transportasi yang biasa digunakan untuk menyebrang antar pulau adalah kapal laut.  
        \nKapal dapat mengangkut berbagai jenis benda mati ataupun hidup, seperti kapal ${this.nama} yang  
        berada dibawah naungan ${this.PT} berjenis ${this.jenis} ini dapat menampung sebanyak  
        ${this.kapasitas} penumpang.`  
    }  
    theSecretInformationOfKapal(){  
        return `${this.nama} hanya beroperasi di Selat Sunda saja, yaitu antara Pelabuhan Merak, Banten  
        dengan Bakauheni, Lampung di kategori kapal Regular dan selalu bersandar di Dermaga 2 dan 3.`  
    }  
}
```

// kelas kedua, yaitu Tiket dengan polymorphism menggunakan kelas Kapal

```
class Tiket extends Kapal {  
    constructor(nama, jenis, kapasitas, PT, hargaTiket, kategoriTiket){  
        super(nama, jenis, kapasitas, PT)  
        this.hargaTiket = hargaTiket,  
        this.kategoriTiket = kategoriTiket;  
    }  
    infoTiket(){  
        return `Harga tiket dari kapal ${this.nama} ini adalah ${this.hargaTiket} dan terdapat 2 kategori  
        tiketnya, yaitu ${this.kategoriTiket}.`  
    }  
    informasiTiketTambahan(){  
        return `Tiket hanya dapat dibeli melalui aplikasi Ferizzi.`  
    }  
}
```

```
}  
}
```

// kelas ketiga, yaitu Rute dengan polymorphism menggunakan kelas Kapal

```
class Rute extends Kapal {  
    constructor(nama, jenis, kapasitas, PT, tujuan, durasiPelayaran){  
        super(nama, jenis, kapasitas, PT)  
        this.tujuan = tujuan,  
        this.durasiPelayaran = durasiPelayaran;  
    }  
    infoRute(){  
        return `Kapal ${this.nama} berlayar di Selat Sunda tujuan ${this.tujuan} dengan durasi  
pelayaran ${this.durasiPelayaran}.`  
    }  
}
```

// kelas keempat, yaitu Jadwal dengan polymorphism menggunakan kelas Rute

```
class Jadwal extends Rute {  
    constructor(nama, jenis, kapasitas, PT, tujuan, durasiPelayaran, keberangkatan){  
        super(nama, jenis, kapasitas, PT, tujuan, durasiPelayaran)  
        this.keberangkatan = keberangkatan;  
    }  
    infoJadwal(){  
        return `Kapal ${this.nama} memiliki ${this.keberangkatan} jadwal keberangkatan setiap 4 kali  
dalam sehari.`  
    }  
}
```

// menampilkan kelas kapal

```
let dataKapal = new Kapal ("Virgo 18", "Ferry", 1000, "PT. Jemla Ferry")  
console.log(dataKapal.infoKapal())  
document.getElementById("objek").innerHTML = dataKapal.infoKapal()  
console.log(dataKapal.theSecretInformationOfKapal())
```

```
document.getElementById("objek").innerHTML = dataKapal.theSecretInformationOfKapal()
```

```
// menampilkan kelas tiket
```

```
let dataTiket = new Tiket ("Virgo 18", "Ferry", 1000, "PT. Jemla Ferry", "29.000/orang", "Regular dan VIP")
```

```
console.log(dataTiket.infoTiket())
```

```
document.getElementById("objek").innerHTML = dataTiket.infoTiket()
```

```
console.log(dataTiket.informasiTiketTambahan())
```

```
document.getElementById("objek").innerHTML = dataTiket.informasiTiketTambahan()
```

```
// menampilkan kelas rute
```

```
let dataRute = new Rute ("Virgo 18", "Ferry", 1000, "PT. Jemla Ferry", "Merak, Banten - Bakauheni, Lampung", "1 jam 45 menit")
```

```
console.log(dataRute.infoRute())
```

```
document.getElementById("objek").innerHTML = dataRute.infoRute()
```

```
// menampilkan kelas jadwal
```

```
let dataKeberangkatan = new Jadwal ("Virgo 18", "Ferry", 1000, "PT. Jemla Ferry", "Merak, Banten - Bakauheni, Lampung", "1 jam 45 menit", 3)
```

```
console.log(dataKeberangkatan.infoJadwal())
```

```
document.getElementById("objek").innerHTML = dataKeberangkatan.infoJadwal()
```