Laporan Praktikum

Mata Kuliah Pemrograman Berorientasi Objek

Tugas 4

“Membuat Class dan Methods dengan Polymorphism”

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

Disusun Oleh :
Dzakiya Fikri Murtianingsih
2300313

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

PENJELASAN KODE PROGRAM

Pada tugas ke-4 Mata Kuliah Pemrograman Berorientasi Objek ini, kami telah belajar bagaimana cara membuat class, menambahkan properties, dan membuat methods menggunakan bahasa pemrograman Java Script dengan kode program sebagai berikut:

```
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.`
    }
}

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.`
    }
}

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}.`
    }
}

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.`
    }
}

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()
```

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()

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()

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()

Penjelasan:

- *Class* atau biasa dikenal dengan *blueprint* atau cetak biru untuk objek digunakan untuk mendefinisikan *properties* (atribut) dan *methods* (fungsi) di dalam *class*. Dengan adanya *class*, kita dapat dapat membuat banyak *instance* objek yang memiliki struktur dan perilaku yang sama.
  Contoh 1:

  ```
  class Kapal {
      constructor(nama, jenis, kapasitas, PT){
          this.nama = nama,
          this.jenis = jenis,
          this.kapasitas = kapasitas,
          this.PT = PT;
  }
  ```

  Pada contoh di atas, Kapal berperan sebagai *class* yang nantikan akan diisi oleh berbagai *properties* dan *methods* di dalamnya.
- Pada class, akan terdapat *constructor function* yang digunakan untuk mendefinisikan *properties* dan *methods* yang ingin dimiliki oleh objek yang akan dibuat dengan menggunakan *keyword this* merujuk pada *instance* objek yang sedang dibuat seperti pada contoh 1.

- *Methods* atau disebut juga dengan fungsi adalah fungsi yang didefinisikan di dalam objek. *Methods* digunakan untuk menggambarkan perilaku objek tersebut dan dapat mengakses serta memanipulasi data yang ada dalam objek tersebut, sering kali melalui penggunaan *keyword this* seperti pada contoh di bawah.
  Contoh 2:

  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.`

  }
- Selanjutnya ada *polymorphism*, yaitu digunakan untuk menggunakan atribut pada *class* yang sebelumnya sudah digunakan, kemudian digunakan kembali dan ditambah dengan atribut dari kelas yang baru. Cara menggunakannya dengan menambahkan kata "extends" seperti pada contoh di bawah.
  Contoh 3:

  class Tiket extends *Kapal* {

  constructor(*nama*, *jenis*, *kapasitas*, *PT*, *hargaTiket*, *kategoriTiket*){

  super(nama, jenis, kapasitas, PT)

  this.hargaTiket = hargaTiket,

  this.kategoriTiket = kategoriTiket;

  }

  Class Kapal digunakan kembali untuk mengambil atribut yang ada di dalamnya dan kemudian ditambahkan dengan atribut pada *class* yang baru, yaitu *class* Tiket.
- Untuk menambahkan isi dari atribut yang telah dibuat, dapat dilihat pada contoh di bawah.
  Contoh 4:

  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()

  Let adalah salah satu kata kunci untuk mendeklarasikan variabel, yaitu dataRute. New Rute(...) menunjukkan *instance* baru dari *class* atau *constructor function* yang bernama Rute dengan parameter yang ada di dalam tanda kurung (). Kemudian method infoRute dari objek dataRute dipanggil dengan sintaks "console.log(dataRute.infoRute())". Kode berikutnya digunakan untuk menampilkan informasi di halaman website.