

TUGAS PENDAHULUAN

MODUL 4

2311104035

Reza Irawan

1. KodePos.js

```
const readline = require('readline');

class KodePos {
  constructor() {
    this.dataKodePos = {
      Batununggal: 40266,
      Kujangsari: 40287,
      Mengger: 40267,
      Wates: 40256,
      Cijaura: 40287,
      Jatisari: 40286,
      Margasari: 40286,
      Sekejati: 40286,
      Kebonwaru: 40272,
      Maleer: 40274,
      Samoja: 40273,
    };
  }

  getKodePos(kelurahan) {
    const formattedKelurahan = kelurahan.charAt(0).toUpperCase() +
      kelurahan.toLowerCase().slice(1);
    return this.dataKodePos[formattedKelurahan] ?
      this.dataKodePos[formattedKelurahan] : "Kode pos tidak ditemukan";
  }

  listKelurahan() {
    console.log("Daftar Kelurahan");
    for(const kelurahan in this.dataKodePos) {
      console.log(`- ${kelurahan}`);
    }
  }
}

const kodePos = new KodePos();

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout,
});

kodePos.listKelurahan();

rl.question('\nMasukkan nama kelurahan untuk mencari kode pos: ', (input) => {
  console.log(`Kode Pos ${input}:`, kodePos.getKodePos(input));
  rl.close();
});
```

Class KodePos menyimpan data kelurahan dan kode pos dalam objek. Tersedia metode getKodePos untuk mencari kode pos secara case-insensitive dan listKelurahan untuk menampilkan daftar kelurahan. Program menggunakan modul readline untuk membaca input terminal, menampilkan daftar kelurahan, lalu meminta input nama kelurahan. Jika ditemukan, kode pos akan ditampilkan.

Output:

```
PS D:\TUGAS KULIAH SEM 4\KPL_PRKTIKUM\04_OOP> node "d:\TUGAS KULIAH SEM 4\KPL_PRKTIKUM\04_OOP\TP_OOP_2311104035\KodePos.js"
• Daftar Kelurahan
- Batununggal
- Kujangsari
- Mengger
- Wates
- Cijaura
- Jatisari
- Margasari
- Sekejati
- Kebonwaru
- Maleer
- Samoja

Masukkan nama kelurahan untuk mencari kode pos: Wates
Kode Pos Wates: 40256
PS D:\TUGAS KULIAH SEM 4\KPL_PRKTIKUM\04_OOP>
```

2. DoorMachine.js

```
class DoorMachine {
  constructor() {
    this.state = 'Terkunci';
    this.showState();
  }

  lock() {
    this.state = 'Terkunci';
    this.showState();
  }

  unlock() {
    this.state = 'Terbuka';
    this.showState();
  }

  showState() {
    if (this.state === 'Terkunci') {
      console.log('Pintu terkunci');
    } else if (this.state === 'Terbuka') {
      console.log('Pintu tidak terkunci');
    }
  }
}

const door = new DoorMachine();

door.unlock();
door.lock();
```

Class DoorMachine.js merepresentasikan mesin pintu dengan dua status: "Terkunci" dan "Terbuka". Saat objek dibuat, status awalnya adalah "Terkunci" dan akan mencetak "Pintu terkunci". Metode yang tersedia:

lock() → Mengubah status menjadi "Terkunci" dan mencetak "Pintu terkunci".

unlock() → Mengubah status menjadi "Terbuka" dan mencetak "Pintu tidak terkunci".

showState() → Menampilkan status pintu berdasarkan kondisi saat ini.

Outputnya:

```
PS D:\TUGAS KULIAH SEM 4\KPL_PRKTIKUM\04_OOP> node "d:\TUGAS KULIAH SEM 4\KPL_PRKTIKUM\04_OOP\TP_OOP_2311104035\DoorMachine.js"
• Pintu terkunci
Pintu tidak terkunci
Pintu terkunci
PS D:\TUGAS KULIAH SEM 4\KPL_PRKTIKUM\04_OOP>
```