

## DOORMACHINE

```
class DoorMachine {  
  constructor() {  
    this.state = "Terkunci";  
    console.log("Pintu terkunci");  
  }  
  
  kunciPintu() {  
    if (this.state === "Terbuka") {  
      this.state = "Terkunci";  
      console.log("Pintu terkunci");  
    } else {  
      console.log("Pintu sudah terkunci");  
    }  
  }  
  
  bukaPintu() {  
    if (this.state === "Terkunci") {  
      this.state = "Terbuka";  
      console.log("Pintu tidak terkunci");  
    } else {  
      console.log("Pintu sudah terbuka");  
    }  
  }  
}  
  
const door = new DoorMachine();  
door.bukaPintu(); // Mengubah state menjadi Terbuka  
door.bukaPintu(); // Mencoba membuka pintu yang sudah terbuka  
door.kunciPintu(); // Mengubah state menjadi Terkunci
```

```
door.kunciPintu(); // Mencoba mengunci pintu yang sudah terkunci
```

```
PS N:\TELKOM UNIVERSITY\SEM 4\KPL_Novita Syahwa Tri Hapsari_2311104007_SE07-01\04_OOP\TP> node .\DoorMachine.js
Pintu terkunci
Pintu tidak terkunci
Pintu sudah terbuka
Pintu terkunci
Pintu sudah terkunci
```

Penjelasan: Kode di atas merupakan implementasi dari mesin status pintu yang dibuat menggunakan JavaScript. Kelas `DoorMachine` memiliki dua status, yaitu "Terkunci" dan "Terbuka". Begitu objek dibuat, pintu akan berada dalam keadaan terkunci, dan pesan "Pintu terkunci" akan ditampilkan di konsol. Metode `bukaPintu()` berfungsi untuk mengubah status pintu menjadi "Terbuka" jika pintu dalam keadaan terkunci. Di sisi lain, metode `kunciPintu()` dipergunakan untuk mengunci kembali pintu yang sebelumnya terbuka. Apabila metode ini dipanggil dalam kondisi yang tidak sesuai—seperti berusaha membuka pintu yang sudah terbuka—maka pesan yang relevan akan ditampilkan tanpa mengubah status pintu. Bagian terakhir dari kode ini mensimulasikan perubahan status dengan memanggil metode yang sudah tersedia.

## KODE POS

```
class KodePos {
    static kodePosDict = {
        batununggal: 40266,
        kujangsari: 40287,
        mengger: 40267,
        wates: 40256,
        cijaura: 40287,
        jatisari: 40286,
        margasari: 40286,
        sekejati: 40286,
        kebonwaru: 40272,
        maleer: 40274,
        samoja: 40273,
    };

    static getKodePos(kelurahan) {
```

```

    return (
        this.kodePosDict[kelurahan.toLowerCase()] || "Kode pos tidak ditemukan"
    );
}

static getListKelurahan() {
    return Object.keys(this.kodePosDict).map(
        (kelurahan) => kelurahan.charAt(0).toUpperCase() + kelurahan.slice(1)
    );
}
}

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

console.log("Daftar Kelurahan yang tersedia:");
console.log(KodePos.getListKelurahan().join(", "));

readline.question("Masukkan nama kelurahan yang di cari: ", (kelurahan) => {
    console.log(`Kode Pos ${kelurahan}: ${KodePos.getKodePos(kelurahan)}`);
    readline.close();
});

```

```

Daftar Kelurahan yang tersedia:
Batununggal, Kujangsari, Mengger, Wates, Cijaura, Jatisari, Margasari, Sekejati, Kebonwaru, Maleer,
Samoja
Masukkan nama kelurahan yang di cari: Mengger
Kode Pos Mengger: 40267
PS N:\TELKOM UNIVERSITY\SEM 4\KPL_Novita Syahwa Tri Hapsari_2311104007_SE07-01\04_OOP\TP>

```

Penjelasan Code: Kode di atas mencari kode pos berdasarkan nama kelurahan. Kelas `KodePos` menyimpan daftar kelurahan dan kode posnya, serta memiliki metode untuk mengambil kode pos dan menampilkan daftar kelurahan. Program menggunakan `readline` untuk menerima input pengguna, lalu menampilkan kode pos yang sesuai atau pesan jika tidak ditemukan.