

1. Doormachine.js

```
1  class DoorMachine {
2    constructor() {
3      this.state = "Terkunci";
4      this.transition = {
5        Terkunci: { BukaPintu: "Terbuka", KunciPintu: "Terkunci" },
6        Terbuka: { BukaPintu: "Terbuka", KunciPintu: "Terkunci" },
7      };
8    }
9
10   action(action) {
11     console.log(`anda melakukan aksi ${action}`);
12     const nextState = this.transition[this.state][action];
13     // console.log(nextState);
14     if (nextState) {
15       this.state = nextState;
16       console.log(`Pintu ${this.state}`);
17       console.log(`posisi pintu saat ini ${this.state}`);
18     } else {
19       console.log("aksi yang anda coba lakukan tidak dikenali");
20     }
21   }
22 }
23
24 const pintu = new DoorMachine();
25
26 pintu.action("BukaPintu");
27
28 pintu.action("KunciPintu");
29
```

Kode ini membuat simulasi mesin pintu sederhana menggunakan konsep *Finite State Machine* (FSM). Mesin pintu memiliki dua status, yaitu "Terkunci" dan "Terbuka", dan dapat menerima dua aksi, yaitu "BukaPintu" dan "KunciPintu". Objek transition mendefinisikan aturan transisi antar status berdasarkan aksi yang diberikan. Metode action memproses aksi yang diberikan, mengubah status pintu sesuai dengan aturan transisi, dan mencetak pesan yang menunjukkan status pintu saat ini. Jika aksi yang diberikan tidak valid, pesan kesalahan akan dicetak.

2. Kodepos.js

```
1  class KodePos {
2      constructor() {
3          this.data = {
4              Batununggal: 40266,
5              Kujangsari: 40287,
6              Mengger: 40267,
7              Wates: 40256,
8              Cijaura: 40287,
9              Jatisari: 40286,
10             Margasari: 40286,
11             Sekejati: 40286,
12             Kebonwaru: 40272,
13             Maleer: 40274,
14             Samoja: 40273,
15         };
16     }
17
18     getKodePos(namaKota) {
19         return this.data[namaKota] || "kodepos tidak ditemukan";
20     }
21 }
22
23 const kodePos = new KodePos();
24 console.log(kodePos.getKodePos("Kujangsari"));
25 console.log(kodePos.getKodePos("Jatisari"));
26 console.log(kodePos.getKodePos("TidakAda"));
27
```

Kode ini mendefinisikan kelas KodePos yang berfungsi untuk menyimpan dan mengambil informasi kode pos berdasarkan nama kota. Kelas ini memiliki properti data yang berupa objek (dictionary) yang memetakan nama kota ke kode posnya. Metode getKodePos digunakan untuk mencari kode pos berdasarkan nama kota yang diberikan sebagai argumen. Jika nama kota ditemukan dalam objek data, metode ini akan mengembalikan kode pos yang sesuai. Jika tidak ditemukan, metode ini akan mengembalikan string "kodepos tidak ditemukan".

Pada bagian akhir kode, objek kodePos dibuat dari kelas KodePos, dan metode getKodePos dipanggil tiga kali dengan nama kota yang berbeda. Hasil dari setiap pemanggilan metode dicetak ke konsol. Pemanggilan pertama dan kedua berhasil menemukan kode pos untuk nama kota yang diberikan, sedangkan pemanggilan ketiga tidak menemukan kode pos dan mengembalikan string "kodepos tidak ditemukan".