

# Tugas Pendahuluan Modul 4

Nama : Salman Alfarisi

Kelas : S1SE-07-01

NIM : 2311104036

1.

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

  getKodePos(kelurahan) {
    return this.tabelKodePos[kelurahan] || "Kode Pos tidak ditemukan";
  }
}

// Fungsi main untuk menampilkan tabel kode pos
function main() {
  const kodePos = new KodePos();
  console.log("Kelurahan".padEnd(15) + "Kode Pos");
  console.log("=====");
  for (let kelurahan in kodePos.tabelKodePos) {
    console.log(kelurahan.padEnd(15) + kodePos.getKodePos(kelurahan));
  }
}

// Panggil fungsi main
main();
```

Penjelasan Kode:

- a. Class KodePos{}  
Mendefinisikan kode sebuah kelas yang bernama KodePos.
- b. Constructor  
Method khusus dalam kelas yang otomatis dipanggil saat objek dibuat.

- c. `this.tabelKodePos {}`  
property objek yang berupa objek javascript berisi pasangan kelurahan: kode\_pos
- d. method `getKodePos(Kelurahan)`
  - parameter kelurahan: Input nama kelurahan yang akan dicari kode posnya.
  - menggunakan `this.tabelKodePos[kelurahan]`:
    - Jika nama kelurahan ditemukan, akan mengambil kode pos yang digunakan.
    - Jika tidak ditemukan, akan mengembalikan "Kode pos tidak ditemukan" menggunakan operator OR (`||`).
- e. Function `main {}`  
untuk menampilkan daftar kode pos
- f. `const kodePost = new KodePos()`  
Membuat objek baru dari kelas `KodePos`, sehingga bisa mengakses datanya.
- g. `console.log("kelurahan".padEnd(15) + "Kode Pos");`  
Menampilkan header tabel dengan `padEnd(15)` agar rata kiri.
- h. `for (let kelurahan in kodePos.tabelKodePos)`  
`{console.log(kelurahan.padEnd(15) + kodePos.getKodePos(kelurahan));`
  - Loop `for...in`: Mengulang setiap kelurahan dalam `tabelKodePos`.
  - Menampilkan nama kelurahan dan kode posnya menggunakan `padEnd(15)` agar tampil rapi.
- i. `Main()`  
Memanggil fungsi `main()` sehingga program berjalan.

#### OUTPUT

Kelurahan	Kode Pos
=====	
Batununggal	40266
Kujangsari	40287
Mengger	40267
Wates	40256
Cijaura	40287
Jatisari	40286
Margasari	40286
Sekejati	40286
Kebonwaru	40272
Maleer	40274
Samoja	40273

2.

```
class DoorMachine {
  constructor() {
    this.state = "Terkunci"; // State awal adalah "Terkunci"
    this.showState();
  }

  // Method untuk mengunci pintu
  kunciPintu() {
    if (this.state === "Terbuka") {
      this.state = "Terkunci";
      this.showState();
    } else {
      console.log("Pintu sudah terkunci.");
    }
  }

  // Method untuk membuka pintu
  bukaPintu() {
    if (this.state === "Terkunci") {
      this.state = "Terbuka";
      this.showState();
    } else {
      console.log("Pintu sudah terbuka.");
    }
  }

  // Method untuk menampilkan state saat ini
  showState() {
    if (this.state === "Terkunci") {
      console.log("Pintu terkunci");
    } else {
      console.log("Pintu tidak terkunci");
    }
  }
}

// Fungsi utama untuk menjalankan simulasi
function main() {
  const pintu = new DoorMachine();
  pintu.bukaPintu(); // Pintu tidak terkunci
  pintu.kunciPintu(); // Pintu terkunci
  pintu.kunciPintu(); // Pintu sudah terkunci
  pintu.bukaPintu(); // Pintu tidak terkunci
}

// Jalankan program
main();
|
```

a. Constructor () {}

- This.state = "Terkunci";  
state awal adalah Terkunci
- This.showState();  
memanggil metode showState.

b. Metode kunciPintu(){}

- Jika state adalah Terbuka maka state berubah menjadi Terkunci, lalu memanggil showState
- Dan jika kondisi tersebut tidak terpenuhi maka akan menampilkan Pintu sudah terkunci

c. Method bukaPintu(){}

- Jika state adalah Terkunci maka state berubah menjadi Terbuka lalu memanggil showState.
- Dan jika kondisi tersebut tidak terpenuhi maka akan menampilkan Pintu sudah Terbuka