TUGAS PENDAHULUAN

1. Kode Pos
   a) Source code
      - KodePos.js

```
You, 3 days ago | 1 author (You)
class KodePos {
  constructor() {
    this.data = {
      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.data[kelurahan] || "Kode pos tidak ditemukan";
  }
}

module.exports = KodePos;
```

      - Main.js

```
You, 3 days ago | 1 author (You)
const KodePos = require("./tp");

const kodePos = new KodePos();
console.log("Kode Pos Batununggal:", kodePos.getKodePos("Batununggal"));
console.log("Kode Pos Kujangsari:", kodePos.getKodePos("Kujangsari"));
```

   b) Output

```
node main.js
Kode Pos Batununggal: 40266
Kode Pos Kujangsari: 40287
pwsh TP main ≡ ?1 85ms
```

   c) Penjelasan

Kelas KodePos dalam JavaScript digunakan untuk menyimpan dan mengambil kode pos berdasarkan nama kelurahan. Saat objek KodePos dibuat, ia akan memiliki sebuah atribut data yang berisi daftar pasangan **nama kelurahan dan kode pos** yang sudah ditentukan. Metode getKodePos(kelurahan) digunakan untuk mencari kode pos dari kelurahan yang diberikan sebagai parameter. Jika kelurahan tersebut ada dalam daftar, metode ini akan mengembalikan kode pos yang sesuai. Namun, jika kelurahan tidak ditemukan dalam data, metode ini akan mengembalikan string "Kode pos tidak ditemukan". Kelas ini diekspor menggunakan module.exports sehingga dapat digunakan di file lain dalam proyek berbasis **Node.js**. Dengan struktur ini, KodePos berfungsi sebagai sistem pencarian kode pos sederhana yang dapat digunakan dalam berbagai aplikasi.

2. DoorMachine

a) Source Code

- DoorMachine.js

```
You, 3 days ago | 1 author (You)
class DoorMachine {
  constructor() {
    this.state = "Terkunci"; // State awal
  }

  unlock() {
    if (this.state === "Terkunci") {
      this.state = "Terbuka";
      console.log("Pintu tidak terkunci");
    } else {
      console.log("Pintu sudah terbuka");
    }
  }

  lock() {
    if (this.state === "Terbuka") {
      this.state = "Terkunci";
      console.log("Pintu terkunci");
    } else {
      console.log("Pintu sudah terkunci");
    }
  }
}

module.exports = DoorMachine;
```

- Main.js

```
You, 3 days ago | 1 author (You)
const DoorMachine = require("./tp");

const door = new DoorMachine();
console.log("State Awal:", door.state);
door.unlock();
door.lock();
```

b) Output

```
pwsh TP2 main ≡ ?1 17ms
> node main.js
State Awal: Terkunci
Pintu tidak terkunci
Pintu terkunci
pwsh TP2 main ≡ ?1 84ms
>
```

c) Penjelasan

Kelas DoorMachine dalam JavaScript berfungsi sebagai **simulasi mesin pintu** yang dapat dikunci dan dibuka. Saat objek DoorMachine dibuat, ia memiliki **state awal "Terkunci"**. Metode unlock() digunakan untuk membuka pintu jika dalam keadaan terkunci, yang akan mengubah state menjadi **"Terbuka"** dan menampilkan pesan **"Pintu tidak terkunci"**. Jika pintu sudah

terbuka, maka metode ini hanya akan mencetak **"Pintu sudah terbuka"** tanpa mengubah state. Sebaliknya, metode lock() digunakan untuk mengunci pintu jika dalam keadaan terbuka, mengubah state kembali menjadi **"Terkunci"**, dan mencetak **"Pintu terkunci"**. Jika pintu sudah terkunci, metode ini hanya akan mencetak **"Pintu sudah terkunci"**. Kelas ini diekspor menggunakan module.exports agar bisa digunakan dalam proyek berbasis **Node.js**. Dengan pendekatan ini, kode dapat digunakan sebagai **state machine sederhana** yang mengatur status pintu dengan aturan logis.