Nama: Muhammad Rizaldy Akbar

Kelas: SE063

NIM: 2211104065

Saya mengambil salah satu kode dari tugas pendahuluan modul 4

Ini sebelum refactoring

KodePos.js

```
class KodePos {
  constructor() {
    this.kodePos = {
      "Batumunggal": "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.kodePos[kelurahan] | | "Kode pos tidak ditemukan";
 }
}
// Contoh penggunaan
const kodePos = new KodePos();
console.log(kodePos.getKodePos("Batumunggal"));
console.log(kodePos.getKodePos("Kujangsari"));
console.log(kodePos.getKodePos("Samoja"));
console.log(kodePos.getKodePos("Banyumas"));
```

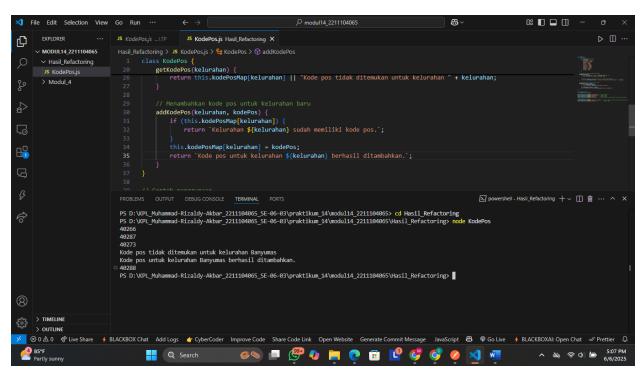
Ini sesudah refactoring

KodePos.js

```
class KodePos {
  constructor() {
    // Peta kelurahan dan kode pos
    this.kodePosMap = {
      "Batumunggal": "40266",
      "Kujangsari": "40287",
      "Mengger": "40267",
      "Wates": "40256",
      "Cijaura": "40287",
      "Jatisari": "40286",
      "Margasari": "40286",
      "Sekejati": "40286",
      "Kebonwaru": "40272",
      "Maleer": "40274",
      "Samoja": "40273"
    };
 }
 // Mendapatkan kode pos berdasarkan kelurahan
 getKodePos(kelurahan) {
    if (typeof kelurahan !== 'string' | | kelurahan.trim() === ") {
      return "Kelurahan harus berupa string yang valid.";
    }
    // Kembalikan kode pos atau pesan error jika kelurahan tidak ditemukan
    return this.kodePosMap[kelurahan] | | "Kode pos tidak ditemukan untuk kelurahan" +
kelurahan;
 }
 // Menambahkan kode pos untuk kelurahan baru
  addKodePos(kelurahan, kodePos) {
    if (this.kodePosMap[kelurahan]) {
      return `Kelurahan ${kelurahan} sudah memiliki kode pos.`;
    this.kodePosMap[kelurahan] = kodePos;
```

```
return 'Kode pos untuk kelurahan ${kelurahan} berhasil ditambahkan.';
  }
}
// Contoh penggunaan
const kodePos = new KodePos();
console.log(kodePos.getKodePos("Batumunggal")); // Output: 40266
console.log(kodePos.getKodePos("Kujangsari")); // Output: 40287
console.log(kodePos.getKodePos("Samoja"));
                                            // Output: 40273
console.log(kodePos.getKodePos("Banyumas")); // Output: Kode pos tidak ditemukan untuk
kelurahan Banyumas
// Menambahkan kode pos baru
console.log(kodePos.addKodePos("Banyumas", "40288")); // Output: Kode pos untuk
kelurahan Banyumas berhasil ditambahkan.
                                                  // Output: 40288
console.log(kodePos.getKodePos("Banyumas"));
```

Output



Setelah refactoring program masih berjalan dengan baik.

Penjelasan

1. Penamaan Variabel:

 kodePos diganti menjadi kodePosMap untuk memberikan nama yang lebih deskriptif. Nama ini menunjukkan bahwa itu adalah peta (map) yang berisi hubungan antara kelurahan dan kode pos.

2. Method addKodePos:

- Ditambahkan method addKodePos untuk memungkinkan penambahan kode pos baru ke dalam peta jika kelurahan belum terdaftar.
- Fungsi ini memeriksa apakah kelurahan sudah ada, jika sudah ada maka akan memberi pesan bahwa kelurahan sudah memiliki kode pos.

3. Validasi Input:

 Menambahkan validasi input di method getKodePos untuk memastikan bahwa input berupa string yang valid. Jika input kosong atau tidak valid, aplikasi akan mengembalikan pesan yang jelas.

4. Komentar:

 Komentar ditambahkan untuk menjelaskan apa yang dilakukan setiap bagian dari kode.

Kelebihan dari Refactoring:

- 1. **Modular dan Terstruktur**: Dengan menambahkan method addKodePos, kode menjadi lebih modular dan memudahkan penambahan fitur baru di masa depan.
- 2. **Validasi Input yang Lebih Baik**: Validasi input lebih jelas, sehingga pengguna mendapatkan feedback yang jelas jika memasukkan kelurahan yang tidak valid.
- 3. **Keterbacaan Kode**: Penggunaan penamaan variabel dan method yang lebih deskriptif membuat kode lebih mudah dipahami.