2311104025
Eduardo Bagus Prima Julian
SE0701

Singleton.js

JS PusatDataSingleton.js > PusatDataSingleton > HapusSebuahData

```
class PusatDataSingleton {
  constructor() {
    if (PusatDataSingleton._instance) {
      return PusatDataSingleton._instance;
    }

    this.DataTersimpan = [];
    PusatDataSingleton._instance = this;
  }

  static GetDataSingleton() {
    if (!PusatDataSingleton._instance) {
      PusatDataSingleton._instance = new PusatDataSingleton();
    }
    return PusatDataSingleton._instance;
  }

  GetSemuaData() {
    return this.DataTersimpan;
  }

  PrintSemuaData() {
    if (this.DataTersimpan.length === 0) {
      console.log("Tidak ada data.");
    } else {
      console.log("Data tersimpan:");
      this.DataTersimpan.forEach((data, index) => {
        console.log(`${index + 1}. ${data}`);
      });
    }
  }

  AddSebuahData(input) {
    this.DataTersimpan.push(input);
  }

  HapusSebuahData(index) {
    if (index >= 0 && index < this.DataTersimpan.length) {
      this.DataTersimpan.splice(index, 1);
    } else {
      console.log("Index tidak valid.");
    }
  }
}

module.exports = PusatDataSingleton;
```

Penjelasan:

Class PusatDataSingleton merupakan implementasi dari design pattern Singleton dalam JavaScript, yang memastikan hanya ada satu objek instance dari class ini selama siklus hidup aplikasi. Class ini memiliki atribut DataTersimpan berupa array List<string> yang menyimpan data secara global, serta properti statis_instance untuk menyimpan instance tunggal dari class tersebut. Konstruktor akan mengembalikan instance yang sudah ada jika sebelumnya telah dibuat. Method GetDataSingleton() berfungsi sebagai access point untuk mendapatkan instance Singleton. Class ini juga menyediakan method untuk menambah (AddSebuah Data), menghapus (Hapus Sebuah Data), mencetak (PrintSemua Data), dan mengambil semua data (GetSemua Data). Dengan pola ini, seluruh bagian program yang memanggil instance akan selalu mengakses dan memodifikasi data yang sama secara konsisten.

Main.js

```
JS main.js > ...
1   const PusatDataSingleton = require('./PusatDataSingleton');
2
3   const data1 = PusatDataSingleton.GetDataSingleton();
4   const data2 = PusatDataSingleton.GetDataSingleton();
5
6   data1.AddSebuahData("Muhammad Fathi Rafa");
7   data1.AddSebuahData("Fajar Budiawan");
8   data1.AddSebuahData("Muhammad Mahrus Ali");
9   data1.AddSebuahData("Asisten: Kak Fazza");
10
11  console.log("Cetak dari data2:");
12  data2.PrintSemuaData();
13
14  data2.HapusSebuahData(3);
15
16  console.log("\nCetak dari data1 setelah penghapusan:");
17  data1.PrintSemuaData();
18
19  console.log(`\nJumlah data di data1: ${data1.GetSemuaData().length}`);
20  console.log(`Jumlah data di data2: ${data2.GetSemuaData().length}`);
```

Penjelasan:

implementasi program utama yang menggunakan class Pusat DataSingleton untuk mendemonstrasikan cara kerja design pattern Singleton. Dua variabel data1 dan data2 dibuat dengan memanggil method GetDataSingleton(), namun karena menggunakan pola Singleton, keduanya sebenarnya merujuk ke instance yang sama. Kemudian, data1 digunakan untuk menambahkan empat data, yaitu nama-nama anggota kelompok dan asisten praktikum. Setelah itu, data2 digunakan untuk mencetak seluruh data yang telah dimasukkan, lalu menghapus data keempat (yaitu nama asisten) menggunakan method Hapus Sebuah Data(). Langkah berikutnya

adalah mencetak ulang data melalui data1, dan hasilnya tidak akan mencantumkan nama asisten karena data1 dan data2 mengakses instance yang sama. Terakhir, program menampilkan jumlah elemen yang tersimpan dalam data1 dan data2, yang akan menunjukkan nilai yang sama sebagai bukti bahwa kedua variabel berbagi data yang identik melalui instance Singleton.

Output:

```
PS D:\KPL\KPL_Eduardo Bagus Prima Julian_2311104025_SE0701\13_Design_Pattern\Jurnal> node main.js
Cetak dari data2:
Data tersimpan:
1. Muhammad Fathi Rafa
2. Fajar Budiawan
3. Muhammad Mahrus Ali
4. Asisten: Kak Fazza

Cetak dari data1 setelah penghapusan:
Data tersimpan:
1. Muhammad Fathi Rafa
2. Fajar Budiawan
3. Muhammad Mahrus Ali

Jumlah data di data1: 3
Jumlah data di data2: 3
```