

Muhammad Fathi Rafa

2311104022

Jurnal modul 13

Singleton.js

```
1 class PusatDataSingleton {
2   constructor() {
3     if (PusatDataSingleton._instance) {
4       return PusatDataSingleton._instance;
5     }
6
7     this.DataTersimpan = [];
8     PusatDataSingleton._instance = this;
9   }
10
11   static GetDataSingleton() {
12     if (!PusatDataSingleton._instance) {
13       PusatDataSingleton._instance = new PusatDataSingleton();
14     }
15     return PusatDataSingleton._instance;
16   }
17
18   GetSemuaData() {
19     return this.DataTersimpan;
20   }
21
22   PrintSemuaData() {
23     if (this.DataTersimpan.length === 0) {
24       console.log("Tidak ada data.");
25     } else {
26       console.log("Data tersimpan:");
27       this.DataTersimpan.forEach((data, index) => {
28         console.log(`${index + 1}. ${data}`);
29       });
30     }
31   }
32
33   AddSebuahData(input) {
34     this.DataTersimpan.push(input);
35   }
36
37   HapusSebuahData(index) {
38     if (index >= 0 && index < this.DataTersimpan.length) {
39       this.DataTersimpan.splice(index, 1);
40     } else {
41       console.log("Index tidak valid.");
42     }
43   }
44 }
45
46 module.exports = PusatDataSingleton;
47
```

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 (`AddSebuahData`), menghapus (`HapusSebuahData`), mencetak (`PrintSemuaData`), dan mengambil semua data (`GetSemuaData`). Dengan pola ini, seluruh bagian program yang memanggil instance akan selalu mengakses dan memodifikasi data yang sama secara konsisten.

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("Eduardo Bagus");
8  data1.AddSebuahData("Mahrus Ali");
9  data1.AddSebuahData("Asisten: Kak Andi");
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}`);
21
```

Penjelasan:

implementasi program utama yang menggunakan class PusatDataSingleton 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 HapusSebuahData(). 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_MuhammadFathiRafa_2311104022_SE0701\13_Design_Pattern\Jurnal_13_2311104022> nc
Cetak dari data2:
Data tersimpan:
1. Muhammad Fathi Rafa
2. Eduardo Bagus
3. Mahrus Ali
4. Asisten: Kak Andi

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

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