

## Journal Mandiri Modul 13

Nama: Satria Ramadhan

Nim: 2311104026

Kelas: SE-07-01

### 1. Source Code

#### a. Index.js

```
13_Design_Pattern_Implementation > JM_Modul_13 > JS index.js > ...
6
7 // Tambah data ke data1
8 data1.AddSebuahData("Viona Aziz Syahputri");
9 data1.AddSebuahData("Satria Ramadhan");
10 data1.AddSebuahData("Rizky Naufal Alghifari");
11
12 // Cetak data dari data2 (harus sama seperti data1)
13 console.log("Data2 setelah penambahan di Data1:");
14 data2.PrintSemuaData();
15
16 // Hapus nama asisten dari data2
17 data2.HapusSebuahData(2); // Index ke-2 (asisten)
18
19 // Cetak kembali dari data1 (asisten seharusnya sudah terhapus)
20 console.log("\nData1 setelah penghapusan di Data2:");
21 data1.PrintSemuaData();
22
23 // Tampilkan jumlah data di masing-masing instance
24 console.log(`\nJumlah data di data1: ${data1.GetSemuaData().length}`);
25 console.log(`Jumlah data di data2: ${data2.GetSemuaData().length}`);
26
```

#### b. singleton.js

```
13_Design_Pattern_Implementation > JM_Modul_13 > JS 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       new PusatDataSingleton();
14     }
15     return PusatDataSingleton._instance;
16   }
17
18   AddSebuahData(input) {
19     this.DataTersimpan.push(input);
20   }
21
22   HapusSebuahData(index) {
23     if (index >= 0 && index < this.DataTersimpan.length) {
24       this.DataTersimpan.splice(index, 1);
25     }
26   }
27
28   GetSemuaData() {
29     return this.DataTersimpan;
30   }
31
32   PrintSemuaData() {
33     console.log("Isi Data:");
34     this.DataTersimpan.forEach((item, i) => {
35       console.log(`${i + 1}. ${item}`);
36     });
37   }
38 }
39
40 module.exports = PusatDataSingleton;
```

### 2. Output

```
>_ pwsh JM_Modul_13 main ?12 ~1 -28 102ms 22.14.0 61%
>> node index.js
Data2 setelah penambahan di Data1:
Isi Data:
1. Viona Aziz Syahputri
2. Satria Ramadhan
3. Rizky Naufal Alghifari

Data1 setelah penghapusan di Data2:
Isi Data:
1. Viona Aziz Syahputri
2. Satria Ramadhan

Jumlah data di data1: 2
Jumlah data di data2: 2
```

### 3. Penjelasan

Tugas Jurnal Modul 13 ini menggunakan design pattern Singleton dengan Node.js. Dalam program ini dibuat sebuah class bernama PusatDataSingleton yang hanya bisa memiliki satu objek di seluruh program. Class ini menyimpan data dalam bentuk list string dan memiliki fungsi untuk menambah, menghapus, dan mencetak data. Dua variabel bernama data1 dan data2 dibuat dengan memanggil fungsi GetDataSingleton(), sehingga keduanya mengakses instance yang sama. Ketika data ditambahkan melalui data1, maka data2 juga langsung memiliki isi yang sama, begitu juga saat data dihapus. Hal ini menunjukkan bahwa Singleton digunakan untuk memastikan semua bagian program mengakses dan memodifikasi data yang sama, sehingga cocok untuk kebutuhan seperti konfigurasi global atau penyimpanan sementara yang konsisten.