

## JURNAL 13

Nama: Mohammad Fathurrohman

Nim: 2211104070

Kelas: SE 0603

### Design\_Pattern\_Implementation

Source Code:

PusatDataSingleton.js

```
1 class PusatDataSingleton {
2   constructor() {
3     if (PusatDataSingleton._instance) {
4       return PusatDataSingleton._instance;
5     }
6     this.dataTersimpan = [];
7     PusatDataSingleton._instance = this;
8   }
9
10  static getDataSingleton() {
11    if (!PusatDataSingleton._instance) {
12      PusatDataSingleton._instance = new
PusatDataSingleton();
13    }
14    return PusatDataSingleton._instance;
15  }
16
17  getSemuaData() {
18    return this.dataTersimpan;
19  }
20
21  printSemuaData() {
22    this.dataTersimpan.forEach((data, index)
=> {
23      console.log(`${index + 1}. ${data}`);
24    });
25  }
26
27  addSebuahData(input) {
28    this.dataTersimpan.push(input);
29  }
30
31  hapusSebuahData(index) {
32    if (index >= 0 && index < this.
dataTersimpan.length) {
33      this.dataTersimpan.splice(index, 1);
34    } else {
35      console.log("Index tidak valid.");
36    }
37  }
38 }
39
40 module.exports = PusatDataSingleton;
41
```

Main.js

```

1  const PusatDataSingleton = require(
    "./pusatDataSingleton");
2
3  const data1 = PusatDataSingleton.
    getDataSingleton();
4  const data2 = PusatDataSingleton.
    getDataSingleton();
5
6  data1.addSebuahData("Mohammad Fathurrohman");
7  data1.addSebuahData("Mfathurrohman");
8  data1.addSebuahData("Vaninside (Asprak)");
9  data1.addSebuahData("rizqiirz (Asprak)");
10
11 console.log("\nData dari data2:");
12 data2.printSemuaData();
13
14 data2.hapusSebuahData(2);
15
16 console.log("\n
    Data 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

```

Output:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603> cd 13 Design Pattern Implementation
PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603\13_Design_Pattern_Implementation> cd JURNAL
PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603\13_Design_Pattern_Implementation\JURNAL> node main.js

Data dari data2:
1. Mohammad Fathurrohman
2. Mfathurrohman
3. Vaninside (Asprak)
4. rizqiirz (Asprak)

Data dari data1 setelah penghapusan:
1. Mohammad Fathurrohman
2. Mfathurrohman
3. rizqiirz (Asprak)

Jumlah data di data1: 3
Jumlah data di data2: 3
PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603\13_Design_Pattern_Implementation\JURNAL>

```

Penjelasan:

Kode ini menggunakan **pola desain Singleton** melalui class `PusatDataSingleton`. Tujuannya adalah agar semua bagian program yang memanggil class ini selalu mengakses **satu objek yang sama**, bukan membuat objek baru setiap kali dipanggil.

Di dalam class terdapat array `dataTersimpan` yang menyimpan data. Terdapat method untuk menambah data (`addSebuahData`), menampilkan semua data (`printSemuaData`), mengambil seluruh data (`getSemuaData`), dan menghapus data berdasarkan indeks (`hapusSebuahData`).

Pada bagian Main, objek `data1` dan `data2` sebenarnya merujuk ke **instance yang sama**. Ketika data ditambahkan lewat `data1`, lalu ditampilkan lewat `data2`, hasilnya tetap muncul karena kedua variabel tersebut mengakses data yang sama. Setelah salah satu data dihapus lewat `data2`, perubahan itu juga terlihat ketika data ditampilkan lewat `data1`.

Artinya, semua perubahan terhadap data terjadi secara **terpusat** karena hanya ada **satu instance** dari `PusatDataSingleton` yang digunakan bersama oleh seluruh program.