

## JURNAL 14

Nama: Mohammad Fathurrohman

Nim: 2211104070

Kelas: SE 0603

## Clean Code

Source Code:

PusatDataSingletonRefactored.js

```
1  /**
2   * Singleton class yang menyimpan data global
3   * aplikasi
4   */
5  class PusatDataSingleton {
6    constructor() {
7      if (PusatDataSingleton._instance) {
8        return PusatDataSingleton._instance;
9      }
10     this.storedData = [];
11     PusatDataSingleton._instance = this;
12   }
13
14   /**
15    * Mengembalikan instance singleton
16    * @returns {PusatDataSingleton}
17    */
18   static getInstance() {
19     if (!PusatDataSingleton._instance) {
20       PusatDataSingleton._instance = new
21       PusatDataSingleton();
22     }
23     return PusatDataSingleton._instance;
24   }
25
26   /**
27    * Mengembalikan seluruh data
28    * @returns {string[]}
29    */
30   getAllData() {
31     return this.storedData;
32   }
33
34   /**
35    * Menampilkan semua data ke console
36    */
37   printAllData() {
38     this.storedData.forEach((item, index) => {
39       console.log(`${index + 1}. ${item}`);
40     });
41   }
42
43   /**
44    * Menambahkan sebuah data ke dalam list
45    * @param {string} input
46    */
47   addData(input) {
48     this.storedData.push(input);
49   }
50
51   /**
52    * Menghapus data berdasarkan index
53    * @param {number} index
54    */
55   removeData(index) {
56     if (index >= 0 && index < this.storedData.
57     length) {
58       this.storedData.splice(index, 1);
59     } else {
60       console.log("Index tidak valid.");
61     }
62   }
63
64   module.exports = PusatDataSingleton;
```

MainRefactored.js

```

1  const PusatDataSingleton = require(
    "./pusatDataSingletonRefactored");
2
3  // Inisialisasi dua variabel singleton
4  const data1 = PusatDataSingleton.getInstance
    ();
5  const data2 = PusatDataSingleton.getInstance
    ();
6
7  // Tambahkan data
8  data1.addData("Mohammad Fathurrohman");
9  data1.addData("Vaninside");
10 data1.addData("rizqiiirz");
11
12 // Cetak data dari data2
13 console.log("\n== Data Awal ==");
14 data2.printAllData();
15
16 // Hapus salah satu data
17 data2.removeData(1);
18
19 // Cetak ulang dari data1
20 console.log("\n== Setelah Penghapusan ==");
21 data1.printAllData();
22
23 // Cek jumlah data
24 console.log(`\nJumlah data di data1: ${data1.
    getAllData().length}`);
25 console.log(`Jumlah data di data2: ${data2.
    getAllData().length}`);

```

Output:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603> cd 14_Clean_Code
PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603\14_Clean_Code> cd JURNAL
PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603\14_Clean_Code\JURNAL> node mainRefactored.js

== Data Awal ==
1. Mohammad Fathurrohman
2. Vaninside
3. rizqiiirz

== Setelah Penghapusan ==
1. Mohammad Fathurrohman
2. rizqiiirz

Jumlah data di data1: 2
Jumlah data di data2: 2
PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603\14_Clean_Code\JURNAL>

```

Penjelasan:

- **Class PusatDataSingleton:**
- Menyimpan data dalam array storedData.

- Hanya dibuat **satu instance** (singleton), tidak peduli berapa kali dipanggil.
- Punya fungsi:
  - `addData()` untuk menambah data.
  - `removeData()` untuk menghapus data berdasarkan indeks.
  - `printAllData()` untuk menampilkan data.
  - `getAllData()` untuk mengambil semua data.
- **Bagian Utama (Main):**
- `data1` dan `data2` mengambil instance yang sama.
- Menambah data lewat `data1`.
- Menampilkan data lewat `data2`.
- Menghapus data lewat `data2`.
- Data ditampilkan ulang lewat `data1` dan hasilnya berubah, karena semua akses mengarah ke **satu data yang sama**.