

TP 14

Nama: Mohammad Fathurrohman

Nim: 2211104070

Kelas: SE 0603

Clean Code

Source Code:

ObserverRefactored.js

```
1  /**
2   * Class Subject: mengelola daftar observer dan status
3   */
4   class Subject {
5     constructor() {
6       this.observers = [];
7       this.state = null;
8     }
9
10    /**
11     * Menambahkan observer
12     * @param {Observer} observer
13     */
14    attach(observer) {
15      this.observers.push(observer);
16    }
17
18    /**
19     * Melepaskan observer
20     * @param {Observer} observer
21     */
22    detach(observer) {
23      this.observers = this.observers.filter((obs) => obs !== observer);
24    }
25
26    /**
27     * Menginformasikan observer tentang perubahan
28     */
29    notify() {
30      this.observers.forEach((observer) => observer.update(this));
31    }
32
33    /**
34     * Mengubah state dan memberi tahu observer
35     * @param {string} newState
36     */
37    setState(newState) {
38      this.state = newState;
39      this.notify();
40    }
41
42    /**
43     * Mengambil nilai state sekarang
44     * @returns {string}
45     */
46    getState() {
47      return this.state;
48    }
49  }
50
51  /**
52   * Class Observer: menerima notifikasi dari Subject
53   */
54  class Observer {
55    constructor(name) {
56      this.name = name;
57    }
58
59    /**
60     * Method yang dipanggil saat Subject berubah
61     */
62    update(subject) {
63      console.log(` ${this.name} menerima perubahan state: ${subject.getState()}`);
64    }
65  }
66
67  module.exports = { Subject, Observer };
68
69
```

MainRefactored.js

```

1  const { Subject, Observer } = require(
    "./observerRefactored");
2
3  // Membuat subject
4  const berita = new Subject();
5
6  // Membuat observer
7  const user1 = new Observer("Fathur");
8  const user2 = new Observer("Vaninside");
9
10 // Mendaftarkan observer
11 berita.attach(user1);
12 berita.attach(user2);
13
14 // Mengubah status
15 console.log("\nStatus awal:");
16 berita.setState("Berita Pagi");
17
18 // Menghapus observer dan ubah lagi
19 berita.detach(user1);
20
21 console.log("\nSetelah pengurangan observer:"
    );
22 berita.setState("Berita Siang");

```

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

Status awal:
Fathur menerima perubahan state: Berita Pagi
Vaninside menerima perubahan state: Berita Pagi

Setelah pengurangan observer:
Vaninside menerima perubahan state: Berita Siang
PS D:\KPL_Mohammad Fathurrohman_2211104070_SE0603\14_Clean_Code\TP> 

```

Penjelasan:

- **Subject:**
Menyimpan **state** dan daftar **observer**.
Saat state berubah, semua observer diberi tahu lewat fungsi `notify()`.
- **Observer:**
Objek yang **menerima notifikasi** dan menampilkan state terbaru dari Subject.

- **Alur Eksekusi:**
 - Membuat Subject bernama berita.
 - Dua observer (user1, user2) didaftarkan.
 - Saat `berita.setState("Berita Pagi")` dipanggil, kedua observer mencetak status.
 - Setelah user1 dilepas, hanya user2 yang menerima notifikasi saat state berubah menjadi "Berita Siang".