

Tugas Pendahuluan Modul 14

1. Refactoring Standar Code naming convention, white space, variable, dan comments

```
// Subject.js

/**
 * Kelas Subject berperan sebagai pusat data yang dapat dipantau oleh observer.
 */
class Subject {
  constructor() {
    this._observers = [];
    this._state = 0;
  }

  /**
   * Menambahkan observer ke dalam daftar pemantau.
   * @param {ConcreteObserver} observer - Observer yang ingin dipantau.
   */
  attach(observer) {
    this._observers.push(observer);
  }

  /**
   * Menghapus observer dari daftar pemantau.
   * @param {ConcreteObserver} observer - Observer yang ingin dilepas.
   */
  detach(observer) {
    this._observers = this._observers.filter((obs) => obs !== observer);
  }

  /**
   * Memberi tahu seluruh observer bahwa state telah berubah.
   */
  notify() {
    this._observers.forEach((observer) => observer.update(this));
  }

  /**
   * Mengubah nilai state dan memberi tahu observer.
   * @param {number} newState - Nilai baru dari state.
   */
  setState(newState) {
    this._state = newState;
    this.notify();
  }

  /**
   * Mengambil nilai state saat ini.
   * @returns {number}
   */
  getState() {
    return this._state;
  }
}

module.exports = Subject;
```



```
// ConcreteObserver.js

/**
 * Kelas ConcreteObserver berperan sebagai pemantau perubahan dari Subject.
 */
class ConcreteObserver {
  constructor(name) {
    this._name = name;
  }

  /**
   * Dipanggil ketika Subject mengalami perubahan state.
   * @param {Subject} subject - Objek Subject yang dipantau.
   */
  update(subject) {
    console.log(
      `${this._name} menerima update: State sekarang = ${subject.getState()}`
    );
  }
}

module.exports = ConcreteObserver;
```



```
// main.js

const Subject = require("./Subject");
const ConcreteObserver = require("./ConcreteObserver");

// Membuat instance Subject
const subject = new Subject();

// Membuat observer dan melampirkannya
const observer1 = new ConcreteObserver("Observer 1");
const observer2 = new ConcreteObserver("Observer 2");

subject.attach(observer1);
subject.attach(observer2);

console.log("Set state ke 1");
subject.setState(1);

console.log("Set state ke 2");
subject.setState(2);

// Melepas observer1
subject.detach(observer1);

console.log("Set state ke 3 (Observer 1 sudah dilepas)");
subject.setState(3);
```

Refactoring dilakukan untuk menyesuaikan kode dengan standar penulisan profesional, termasuk penerapan naming convention yang konsisten (menggunakan camelCase untuk variabel dan method, serta prefix _ untuk atribut internal). Seluruh kode diberi komentar deskriptif untuk menjelaskan fungsi tiap method dan class secara ringkas. Struktur whitespace dan indentasi ditata ulang agar lebih rapi dan mudah dibaca. Selain itu, kode dipecah menjadi tiga file terpisah (Subject.js, ConcreteObserver.js, dan main.js) agar sesuai dengan prinsip modularitas dan single responsibility, sehingga memudahkan pengelolaan dan pengujian kode. Refactoring ini memperkuat penerapan Observer Pattern secara lebih bersih dan terstruktur.