Zulfa M.A.I

2311104010

TP

Project ini merupakan hasil refactoring dari project Modul 13 yang menerapkan **Observer Pattern** menggunakan JavaScript. Dalam kode ini, terdapat tiga kelas utama, yaitu WeatherStation, TemperatureDisplay, dan TemperatureLogger.

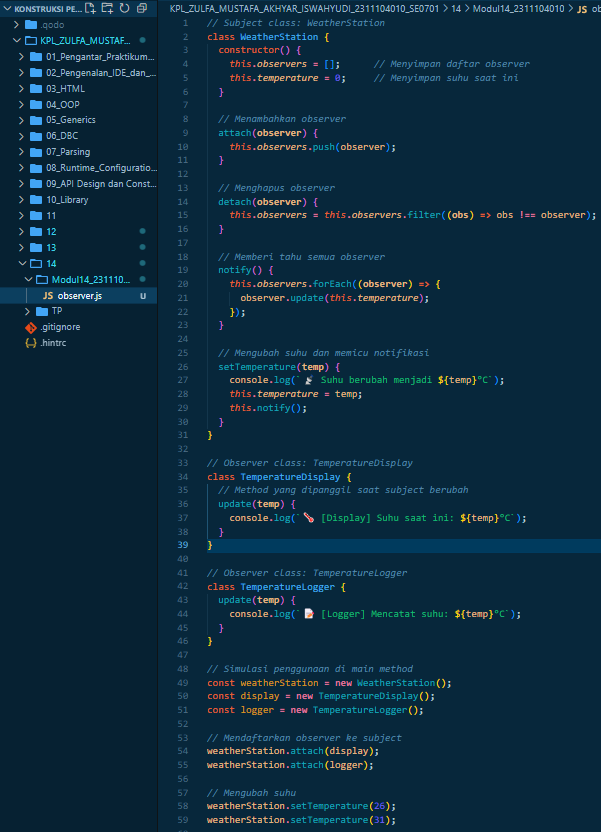
Disini saya ambil dari code Observer dari TP ke-13 saja, berikut refactoring yang dilakukan:

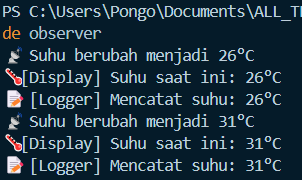
|  |  |
| --- | --- |
| **Aspek** | **Deskripsi Refactoring** |
| **Naming** | Menggunakan camelCase untuk method dan atribut |
| **Indentation** | Konsisten 2 spasi, rapi |
| **Whitespace** | Ada jarak antar method untuk keterbacaan |
| **Comments** | Komentar singkat menjelaskan fungsi tiap bagian |
| **Emoji opsional** | Ditambahkan untuk visual feedback saat simulasi dijalankan |

Kelas WeatherStation bertugas sebagai **subject**, yaitu objek pusat yang menyimpan data suhu. Ketika suhu diperbarui melalui method setTemperature(), objek ini akan memanggil notify() yang bertugas memberi tahu semua observer tentang perubahan tersebut.

Kelas TemperatureDisplay dan TemperatureLogger adalah dua **observer** yang masing-masing menampilkan dan mencatat suhu terbaru. Keduanya memiliki method update() yang akan dipanggil oleh subject saat ada perubahan.

Struktur ini memungkinkan pengembangan lebih lanjut yang mudah, seperti menambahkan observer lain tanpa harus mengubah class WeatherStation.





TP

Pada tugas ini, saya menggunakan proyek kalkulator sederhana berbasis HTML dan JavaScript sebagai bahan refactoring. Kalkulator ini dirancang untuk menerima input angka, menampilkan ekspresi matematika, serta melakukan perhitungan dasar seperti penjumlahan.

Saya pilih Tugas Jurnal minggu ke-4 tentang HTML + JS pada kalkulator sederhana:  
**Penjelasan refactor:**

* **Naming convention**:
  + Variabel dan fungsi memakai camelCase, konstan memakai PascalCase (tidak ada di sini).
  + Fungsi diberi nama deskriptif seperti displayNumber, addOperator, calculateResult, resetCalculator.
* **Indentasi dan whitespace**:
  + Konsisten menggunakan 2 spasi untuk indentasi (bisa juga 4 spasi kalau kamu mau, yang penting konsisten).
  + Baris kosong yang cukup untuk memisahkan fungsi dan blok kode agar mudah dibaca.
* **Komentar**:
  + Tambahkan komentar fungsi dan penjelasan variabel global supaya mudah dipahami.
  + Hindari komentar berlebihan yang hanya mengulang kode.
* **Deklarasi variabel**:
  + Deklarasi di awal, dengan konstanta (const) dan variabel (let) sudah tepat.
* **Perbaikan kecil**:
  + Cek agar tidak bisa memasukkan operator dua kali berturut-turut (misal 1++2) dengan fungsi isOperatorLastChar().
  + Hindari error dengan cek kondisi sebelum evaluasi.

Contoh refactoring file gui kalkulator yagn sudah di clean-code:





