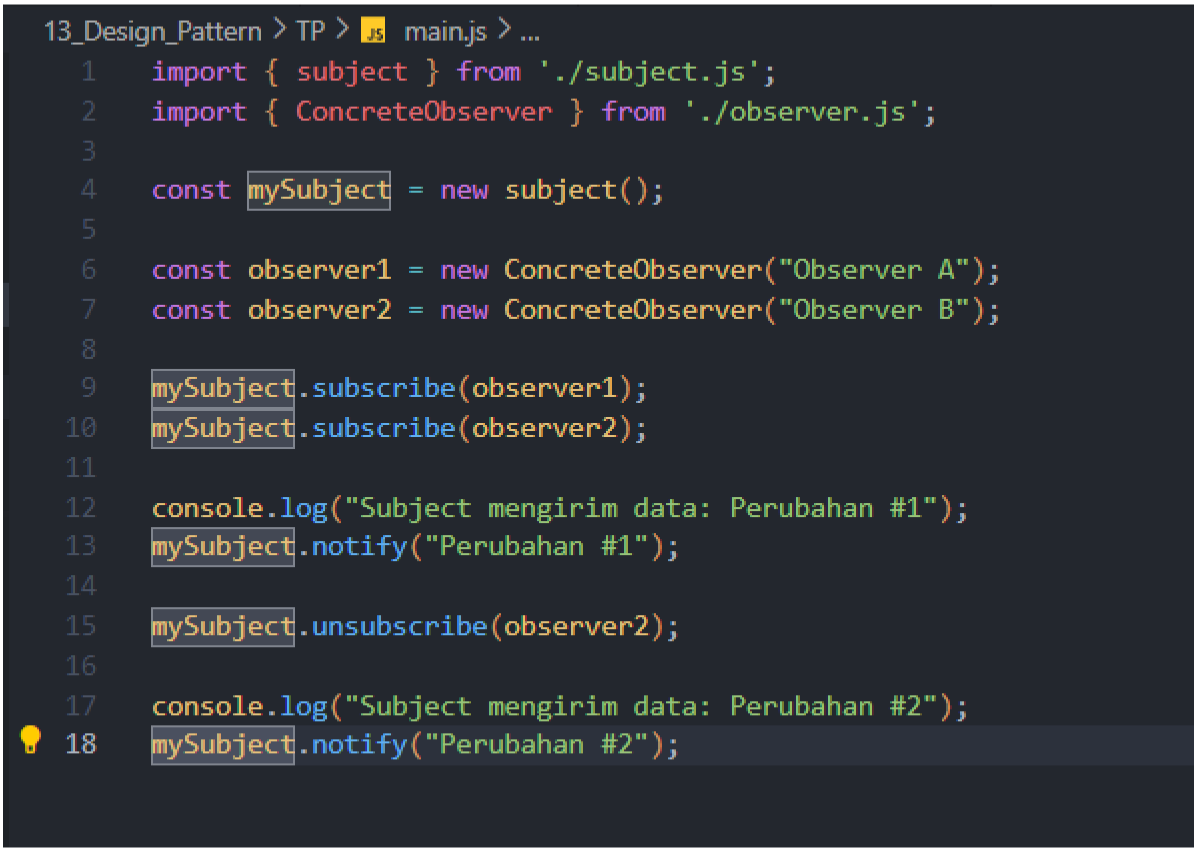
Kelvin Ferdinan_2311104009

https://github.com/mispipin/KPL_Kelvin-Ferdinan_2311104009_s1se0701

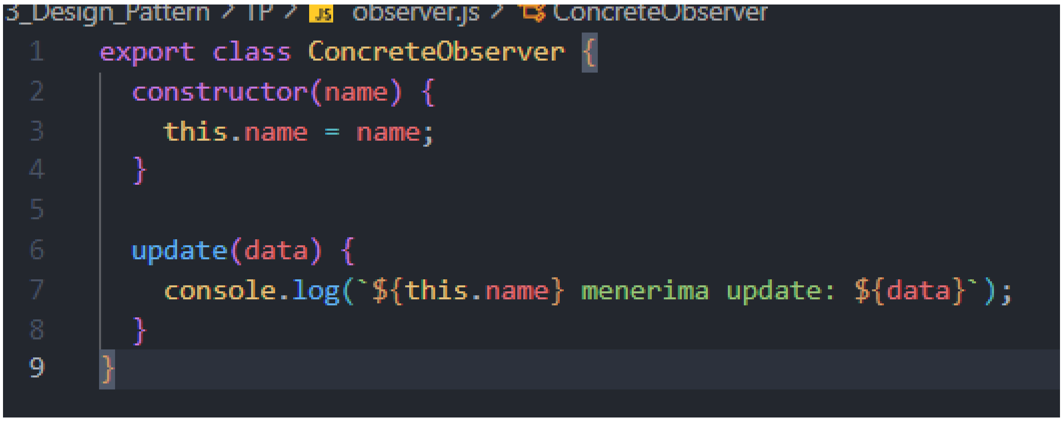
1. Observer pattern cocok digunakan ketika satu objek (subject) perlu memberi tahu banyak objek lain (observers) saat terjadi perubahan, tanpa perlu tahu siapa observers-nya.
2. Cara implementasi observer pattern
   a. Buat interface atau abstract class observer
   b. Buat interface atau abstract class subject
   c. Implementasikan concrete observer
   d. Implementasikan concrete subject
3. Kelebihan:
   a. Low coupling: Subject tidak perlu tahu detail implementasi observers.
   b. Scalable: Mudah menambah atau menghapus observers tanpa mengubah subject.
   c. Realtime update: Cocok untuk sistem event-driven atau notifikasi otomatis.

main.js



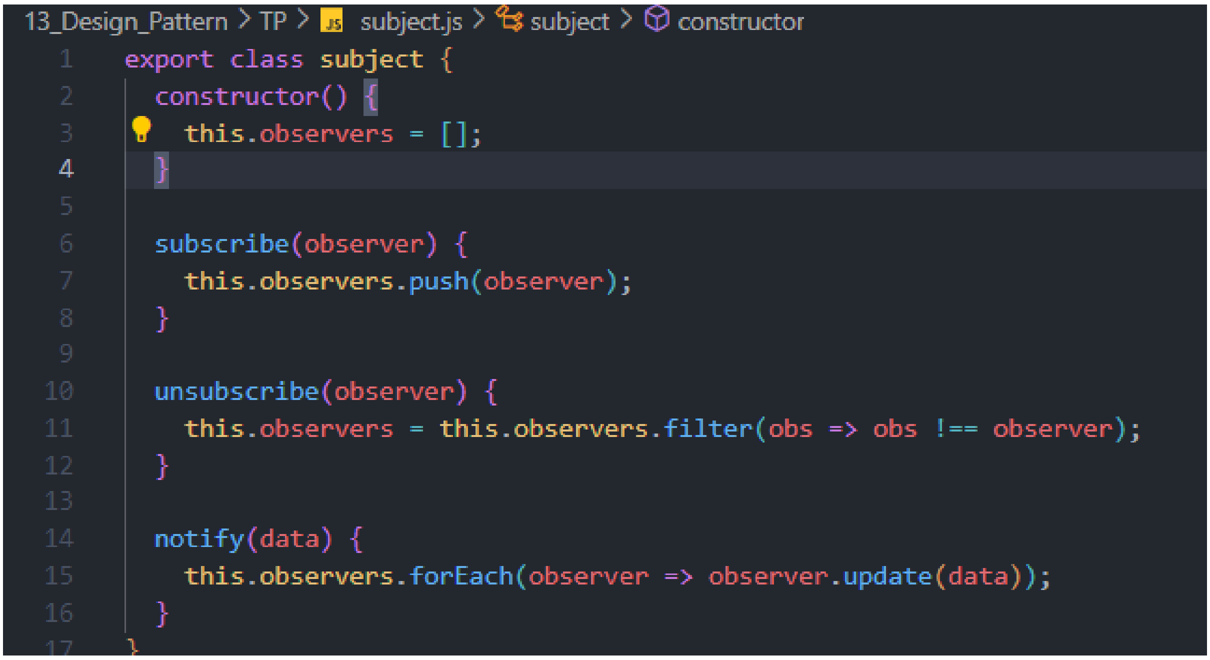
```javascript
import { subject } from './subject.js';
import { ConcreteObserver } from './observer.js';

const mySubject = new subject();

const observer1 = new ConcreteObserver("Observer A");
const observer2 = new ConcreteObserver("Observer B");

mySubject.subscribe(observer1);
mySubject.subscribe(observer2);

console.log("Subject mengirim data: Perubahan #1");
mySubject.notify("Perubahan #1");

mySubject.unsubscribe(observer2);

console.log("Subject mengirim data: Perubahan #2");
mySubject.notify("Perubahan #2");
```

Obserever.js



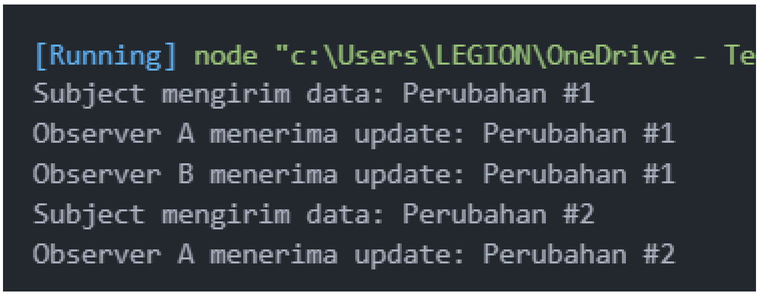
```javascript
export class ConcreteObserver {
  constructor(name) {
    this.name = name;
  }

  update(data) {
    console.log(`${this.name} menerima update: ${data}`);
  }
}
```

Subject.js



```javascript
export class subject {
  constructor() {
    this.observers = [];
  }

  subscribe(observer) {
    this.observers.push(observer);
  }

  unsubscribe(observer) {
    this.observers = this.observers.filter(obs => obs !== observer);
  }

  notify(data) {
    this.observers.forEach(observer => observer.update(data));
  }
}
```

Output:



```
[Running] node "c:\Users\LEGION\OneDrive - Te
Subject mengirim data: Perubahan #1
Observer A menerima update: Perubahan #1
Observer B menerima update: Perubahan #1
Subject mengirim data: Perubahan #2
Observer A menerima update: Perubahan #2
```