

observer.js

File ini berisi implementasi utama dari pola **Observer**, yaitu:

- **Subject** (yang diamati / Publisher)
- **Observer** (yang mengamati / Subscriber)

```
// Subject (Publisher)
class Subject {
  constructor() {
    this.observers = [];
    this.state = null;
  }

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

  setState(newState) {
    this.state = newState;
    this.notify();
  }

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

// Observer (Subscriber)
class Observer {
  constructor(name) {
    this.name = name;
  }

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

module.exports = { Subject, Observer };
```

- **Modular Design:** Dipisahkan dari index.js agar reusable dan tidak bercampur dengan logika server.
- **Observer List Management:** Subject menyimpan daftar observer yang akan diberi notifikasi saat terjadi perubahan state.

- **Notification Mechanism:** Ketika `Subject.setState()` dipanggil, semua observer otomatis menerima notifikasi.

`index.js` => Menjadi entry point aplikasi, menjalankan server Express dan menguji pola Observer.

```
const express = require('express');
const { Subject, Observer } = require('./observer');

const app = express();
const PORT = 3000;

// Membuat publisher dan observer
const subject = new Subject();

const observerA = new Observer('Observer A');
const observerB = new Observer('Observer B');

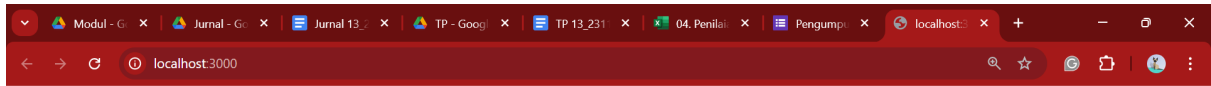
subject.attach(observerA);
subject.attach(observerB);

// Route utama
app.get('/', (req, res) => {
  subject.setState('Data diperbarui!');
  res.send('Observer telah diberi notifikasi.');
```

- **Initialization:** Membuat objek Subject dan dua Observer, lalu mendaftarkannya ke Subject.
- **Trigger & Test:** Mengubah state Subject melalui request HTTP untuk memicu notifikasi ke semua observer.
- **Integration Layer:** Menjembatani observer pattern dengan antarmuka web (via Express).

Ouput

```
Server berjalan di http://localhost:3000  
Observer A menerima update: Data diperbarui!  
Observer B menerima update: Data diperbarui!
```



Observer telah diberi notifikasi.