

LAPORAN PRAKTIKUM
Tugas Pendahuluan Modul 13
Design Pattern Implementation



Disusun Oleh:

Berlian Seva Astryana 2311104067

S1SE-07-02

Dosen :

Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

PURWOKERTO

2024

1. Penjelasan Sourcode

```
var subject = new Subject();
```

Baris kode ini membuat objek baru dari kelas Subject. Objek ini bertindak sebagai pusat perubahan status yang nantinya akan memberi tahu semua observer yang terdaftar ketika terjadi perubahan pada state internal-nya.

```
var observerA = new ConcreteObserverA();  
subject.Attach(observerA);
```

Dua baris ini membuat objek observer pertama yaitu ConcreteObserverA, kemudian langsung mendaftarkannya (attach) ke objek subject. Dengan mendaftar sebagai observer, observerA akan menerima notifikasi setiap kali state pada subject berubah.

```
var observerB = new ConcreteObserverB();  
subject.Attach(observerB);
```

Langkah berikutnya adalah membuat observer kedua yaitu ConcreteObserverB, dan juga mendaftarkannya ke dalam daftar observer subject. Sekarang subject memiliki dua observer yang siap menerima notifikasi.

csharp

```
subject.SomeBusinessLogic();  
subject.SomeBusinessLogic();
```

Pemanggilan method SomeBusinessLogic() akan mengubah nilai State dari subject secara acak, lalu memicu pemanggilan Notify(). Ketika Notify() dijalankan, seluruh observer yang terdaftar akan diberi tahu perubahan tersebut melalui method Update() masing-masing. Karena pemanggilan dilakukan dua kali, maka state subject berubah dua kali dan kedua observer akan bereaksi sesuai kondisi yang sudah mereka tentukan.

```
subject.Detach(observerB);
```

Baris ini menghapus observerB dari daftar observer subject. Setelah baris ini dieksekusi, observerB tidak akan lagi menerima notifikasi ketika subject berubah.

```
subject.SomeBusinessLogic();
```

Pemanggilan SomeBusinessLogic() terakhir ini hanya akan memberikan notifikasi kepada observerA, karena observerB sudah tidak lagi terdaftar. Dengan begitu, hanya observerA yang akan bereaksi terhadap perubahan state yang terjadi di subject.

2. Output

```
Subject: Attached an observer.  
Subject: Attached an observer.  
  
Subject: I'm doing something important.  
Subject: My state has just changed to: 7  
Subject: Notifying observers...  
ConcreteObserverB: Reacted to the event.  
  
Subject: I'm doing something important.  
Subject: My state has just changed to: 8  
Subject: Notifying observers...  
ConcreteObserverB: Reacted to the event.  
Subject: Detached an observer.  
  
Subject: I'm doing something important.  
Subject: My state has just changed to: 8  
Subject: Notifying observers...
```