



TUGAS PENDAHULUAN
KONSTRUKSI PERANGKAT LUNAK
MODUL 13

D
I
S
U
S
U
N

OLEH :
TIURMA GRACE ANGELINA – 2311104042
S1-SE07-02

Dosen :
YIS

Prodi S1 Rekaya Perangkat Lunak
Direktorat Kampus Purwokerto

1. MENJELASKAN SALAH SATU DESIGN PATTERN

a. Contoh kondisi penggunaan design pattern "Observer"

Design pattern Observer cocok digunakan dalam situasi seperti aplikasi media sosial atau platform berita. Misalnya, ketika sebuah akun yang kita ikuti memposting konten baru, semua followers akan mendapatkan notifikasi. Di sini, akun yang diikuti bertindak sebagai Subject, sedangkan followers bertindak sebagai Observer. Ketika Subject berubah (misalnya posting konten baru), semua Observer akan diberi tahu secara otomatis.

b. Langkah-langkah implementasi design pattern "Observer"

Langkah-langkah mengimplementasikan design pattern Observer:

1. Buat interface Observer yang memiliki method `update()` yang akan dipanggil ketika ada perubahan pada Subject.
2. Buat interface Subject dengan method `attach()` untuk menambah observer, `detach()` untuk menghapus observer, dan `notify()` untuk memberitahu semua observer.
3. Implementasikan kelas `ConcreteSubject` yang menyimpan daftar observer dan memanggil `update()` pada setiap observer ketika terjadi perubahan.
4. Implementasikan kelas `ConcreteObserver` yang mendefinisikan tindakan spesifik saat menerima notifikasi.

c. Kelebihan dan kekurangan design pattern "Observer"

Kelebihan:

1. Prinsip "loose coupling" - Subject dan Observer tidak perlu saling mengenal secara detail.
2. Mendukung prinsip "open/closed" - mudah menambahkan Observer baru tanpa mengubah Subject.
3. Memungkinkan komunikasi antara objek-objek saat runtime.

Kekurangan:

1. Dapat menyebabkan memory leak jika Observer tidak di-detach dengan benar.
2. Observer tidak tahu persis apa yang berubah pada Subject.
3. Notifikasi yang tidak tertata dapat menyebabkan perubahan yang tidak terduga atau "update cascade".

2. IMPLEMENTASI DAN PEMAHAMAN DESIGN PATTERN OBSERVER

1. IObserver.cs

```
namespace tpmodul13_2311104042
{
    public interface IObserver
    {
        void Update(ISubject subject);
    }
}
```

2. ISubject.cs

```
namespace tpmodul13_2311104042
{
    public interface ISubject
    {
        void Attach(IObserver observer);
        void Detach(IObserver observer);
        void Notify();
    }
}
```

3. ConcreteSubject.cs

```
using System;
using System.Collections.Generic;
using System.Threading;

namespace tpmodul13_2311104042
{
    public class ConcreteSubject : ISubject
    {
        public int State { get; set; } = 0;

        private List<IObserver> _observers = new List<IObserver>();

        public void Attach(IObserver observer)
        {

```

```
        Console.WriteLine("Subject: Observer ditambahkan.");
        _observers.Add(observer);
    }

    public void Detach(IObserver observer)
    {
        _observers.Remove(observer);
        Console.WriteLine("Subject: Observer dihapus.");
    }

    public void Notify()
    {
        Console.WriteLine("Subject: Memberitahu observer...");
        foreach (var observer in _observers)
        {
            observer.Update(this);
        }
    }

    public void SomeBusinessLogic()
    {
        Console.WriteLine("\nSubject: Saya melakukan sesuatu yang penting.");
        this.State = new Random().Next(0, 10);

        Thread.Sleep(15);

        Console.WriteLine($"Subject: State saya telah berubah menjadi: {this.State}");
        this.Notify();
    }
}
```

4. ConcreteObserverA.cs

```
using System;

namespace tpmodul13_2311104042
{
    public class ConcreteObserverA : IObservable
    {
        public void Update(ISubject subject)
        {
            if ((subject as ConcreteSubject).State < 3)
            {
                Console.WriteLine("ConcreteObserverA: Merespons perubahan state yang kecil");
            }
        }
    }
}
```

5. ConcreteObserverB.cs

```
using System;

namespace tpmodul13_2311104042
{
    public class ConcreteObserverB : IObservable
    {
        public void Update(ISubject subject)
        {
            if ((subject as ConcreteSubject).State >= 3)
            {
                Console.WriteLine("ConcreteObserverB: Merespons perubahan state yang besar");
            }
        }
    }
}
```

6. Program.cs

```
using System;

namespace tpmodul13_2311104042
{
    class Program
    {
        static void Main(string[] args)
        {
            var subject = new ConcreteSubject();

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

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

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

            subject.Detach(observerA);

            subject.SomeBusinessLogic();

            Console.WriteLine("Tekan tombol apa saja untuk keluar...");
            Console.ReadKey();
        }
    }
}
```

The screenshot shows the Visual Studio IDE with the following components:

- Source Code (Program.cs):**

```

14         var observerB = new ConcreteObserverB();
15         subject.Attach(observerB);
16
17         // ... (rest of the code)

```
- Microsoft Visual Studio Debug Console:**

```

Subject: Observer ditambahkan.
Subject: Observer ditambahkan.

Subject: Saya melakukan sesuatu yang penting.
Subject: State saya telah berubah menjadi: 8
Subject: Memberitahu observer...
ConcreteObserverB: Merespons perubahan state yang besar

Subject: Saya melakukan sesuatu yang penting.
Subject: State saya telah berubah menjadi: 7
Subject: Memberitahu observer...
ConcreteObserverB: Merespons perubahan state yang besar
Subject: Observer dihapus.

Subject: Saya melakukan sesuatu yang penting.
Subject: State saya telah berubah menjadi: 3
Subject: Memberitahu observer...
ConcreteObserverB: Merespons perubahan state yang besar
Tekan tombol apa saja untuk keluar...

```
- Solution Explorer:** Shows the project structure for 'tpmod13_2311104042', including dependencies like ConcreteObserverA.cs, ConcreteObserverB.cs, ConcreteSubject.cs, IObservable.cs, IObserver.cs, and Program.cs.
- Output Window:** Shows the execution path and exit code:

```

C:\Users\USER\source\repos\KPL_TiurmaGraceA_2311104042_S1SE-07-02\13_Design_Pattern_Implementation\tpmod13_2311104042\tpmod13_2311104042\bin\Debug\net8.0\tpmod13_2311104042.exe (process 13548) exited with code 0 (0x0).

```
- Error List:** Empty.

Membuat objek ConcreteObserverA sebagai observer pertama. Menghubungkan observerA ke subject menggunakan method Attach(), sehingga observerA akan mendapatkan notifikasi saat terjadi perubahan pada subject.

9. `var observerB = new ConcreteObserverB();`

`subject.Attach(observerB);`

Membuat observer kedua, yaitu ConcreteObserverB. Sama seperti sebelumnya, observerB ditambahkan ke daftar observer dari subject dan akan menerima notifikasi.

10. `subject.SomeBusinessLogic();`

`subject.SomeBusinessLogic();`

Memanggil method SomeBusinessLogic() dua kali. Method ini akan mengubah nilai state secara acak, kemudian memanggil method Notify() yang akan memicu method Update() pada semua observer. ObserverA akan merespons jika State < 3. ObserverB akan merespons jika State >= 3.

11. `subject.Detach(observerA);`

Menghapus observerA dari daftar observer subject.

Setelah ini, observerA tidak akan lagi menerima notifikasi saat state subject berubah.

12. `subject.SomeBusinessLogic();`

Menjalankan kembali logika bisnis setelah observerA dihapus. Kali ini, hanya observerB yang akan menerima notifikasi karena hanya dia yang masih terdaftar.

13. `Console.WriteLine("Tekan tombol apa saja untuk keluar...");`

`Console.ReadKey();`

Menampilkan pesan kepada pengguna dan menunggu input tombol apa saja sebelum menutup program. Ini mencegah jendela console langsung tertutup setelah program selesai dijalankan.

Penjelasan tiap baris kode:

1. `var subject = new ConcreteSubject();`
 - Membuat instance dari kelas `ConcreteSubject` yang akan menjadi subjek yang diobservasi.
2. `var observerA = new ConcreteObserverA(); dan subject.Attach(observerA);`
 - Membuat instance dari `ConcreteObserverA` dan mendaftarkannya ke `subject` menggunakan method `Attach()`.
 - Observer A akan mendapatkan notifikasi ketika state `subject` berubah menjadi kurang dari 3.
3. `var observerB = new ConcreteObserverB(); dan subject.Attach(observerB);`
 - Membuat instance dari `ConcreteObserverB` dan mendaftarkannya ke `subject` menggunakan method `Attach()`.
 - Observer B akan mendapatkan notifikasi ketika state `subject` berubah menjadi lebih dari atau sama dengan 3.
4. `subject.SomeBusinessLogic();`
 - Menjalankan logika bisnis pada `subject` yang akan mengubah state secara acak.
 - Setelah state berubah, method `Notify()` akan dipanggil dan semua observer akan mendapatkan notifikasi melalui method `Update()`.
 - Dijalankan dua kali untuk melihat berbagai kemungkinan perubahan state.
5. `subject.Detach(observerA);`
 - Menghapus `observerA` dari daftar observer pada `subject`.
 - Setelah baris ini dijalankan, `observerA` tidak akan mendapatkan notifikasi lagi.
6. `subject.SomeBusinessLogic();`
 - Menjalankan logika bisnis lagi setelah `observerA` dihapus.
 - Hanya `observerB` yang akan mendapatkan notifikasi.
7. `Console.ReadKey();`
 - Menunggu input dari user untuk mengakhiri program.

Commit

```
C:\WINDOWS\system32\cmd.exe
*****
** Visual Studio 2022 Developer Command Prompt v17.13.0
** Copyright (c) 2022 Microsoft Corporation
*****

C:\Users\USER\source\repos\KPL_TiurmaGraceA_2311104042_S1SE-07-02\13_Design_Pattern_Implementation\tpmod13_2311104042>cd ..

C:\Users\USER\source\repos\KPL_TiurmaGraceA_2311104042_S1SE-07-02\13_Design_Pattern_Implementation>git add tpmod13_2311104042

C:\Users\USER\source\repos\KPL_TiurmaGraceA_2311104042_S1SE-07-02\13_Design_Pattern_Implementation>git commit -m "IMPLEMENTASI DAN PEMAHAMAN DESIGN PATTERN OBSERVER dan up
gitignore"
[main bd8a7fe] IMPLEMENTASI DAN PEMAHAMAN DESIGN PATTERN OBSERVER dan up gitignore
22 files changed, 806 insertions(+)
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/ProjectEvaluation/tpmod13_2311104042.metadata.v9.bin
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/ProjectEvaluation/tpmod13_2311104042.projects.v9.bin
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/ProjectEvaluation/tpmod13_2311104042.strings.v9.bin
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/DesignTimeBuild/.dtbcache.v2
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/FileContentIndex/194aeaab-0963-49cd-abe3-5f0f372be566.vsidx
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/FileContentIndex/56ce3b26-9947-41f8-a153-e89c8f092d3b.vsidx
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/FileContentIndex/614402d1-b3df-40ea-ab15-5efafe5daee8.vsidx
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/FileContentIndex/e14b0363-04ce-413f-aba0-b6f9fa9c0701.vsidx
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/FileContentIndex/ebd5a096-a6ce-48ed-9e88-7c383631712c.vsidx
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/v17/.fudtcache.v2
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/v17/.suo
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/v17/DocumentLayout.backup.json
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/.vs/tpmod13_2311104042/v17/DocumentLayout.json
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042.sln
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042/.gitignore
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042/ConcreteObserverA.cs
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042/ConcreteObserverB.cs
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042/ConcreteSubject.cs
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042/IObserver.cs
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042/ISubject.cs
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042/Program.cs
create mode 100644 13_Design_Pattern_Implementation/tpmod13_2311104042/tpmod13_2311104042/tpmod13_2311104042.csproj

C:\Users\USER\source\repos\KPL_TiurmaGraceA_2311104042_S1SE-07-02\13_Design_Pattern_Implementation>git push origin main
```