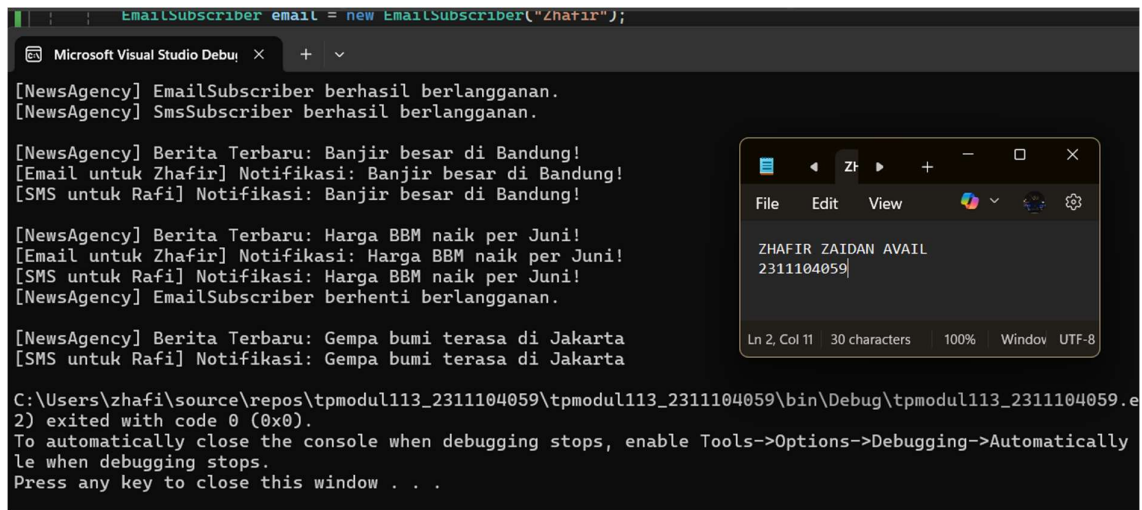


TP MODUL 13

Nama: Zhafir Zaidan Avail

NIM: 2311104059



```
Microsoft Visual Studio Debug Console

[NewsAgency] EmailSubscriber berhasil berlangganan.
[NewsAgency] SmsSubscriber berhasil berlangganan.

[NewsAgency] Berita Terbaru: Banjir besar di Bandung!
[Email untuk Zhafir] Notifikasi: Banjir besar di Bandung!
[SMS untuk Rafi] Notifikasi: Banjir besar di Bandung!

[NewsAgency] Berita Terbaru: Harga BBM naik per Juni!
[Email untuk Zhafir] Notifikasi: Harga BBM naik per Juni!
[SMS untuk Rafi] Notifikasi: Harga BBM naik per Juni!
[NewsAgency] EmailSubscriber berhenti berlangganan.

[NewsAgency] Berita Terbaru: Gempa bumi terasa di Jakarta
[SMS untuk Rafi] Notifikasi: Gempa bumi terasa di Jakarta

C:\Users\zhafi\source\repos\tpmodul113_2311104059\tpmodul113_2311104059\bin\Debug\tpmodul113_2311104059.exe
2) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically
close when debugging stops.
Press any key to close this window . . .
```

Menjelaskan Observer Design Pattern:

a. Contoh Penggunaan Observer

Sebuah aplikasi **cuaca** yang mengirimkan **notifikasi** ke pengguna ketika suhu berubah. Observer digunakan agar setiap device (smartphone, tablet, TV) bisa menerima update cuaca secara otomatis dari server pusat.

b. Langkah-Langkah Implementasi Observer

1. Buat interface Observer (misalnya ISubscriber).
2. Buat interface Subject (misalnya INewsPublisher).
3. Subject menyimpan daftar observer.
4. Observer melakukan Attach() untuk mendaftar, dan Detach() untuk keluar.
5. Subject memanggil Notify() jika data berubah.
6. Observer akan mendapat data baru lewat method Update().

c. Kelebihan & Kekurangan

Kelebihan:

1. Loose coupling (lebih modular)
2. Mudah menambah observer baru
3. Responsif terhadap perubahan data

Kekurangan:

1. Jika terlalu banyak observer, performa menurun
2. Kompleksitas meningkat
3. Debugging antar banyak subscriber sulit

Struktur Proyek:

tpmodul113_2311104059/

- ISubscriber.cs
- NewsAgency.cs
- EmailSubscriber.cs
- SmsSubscriber.cs
- Program.cs

ISubscriber.cs (Interface Observer)

```
public interface ISubscriber
{
    void Update(string message);
}
```

```
}
```

NewsAgency.cs (Subject)

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

public class NewsAgency
{
    private List<ISubscriber> subscribers = new List<ISubscriber>();
    private string latestNews;

    public void Attach(ISubscriber subscriber)
    {
        subscribers.Add(subscriber);
        Console.WriteLine($"[NewsAgency] {subscriber.GetType().Name} berhasil
berlangganan.");
    }

    public void Detach(ISubscriber subscriber)
    {
        subscribers.Remove(subscriber);
        Console.WriteLine($"[NewsAgency] {subscriber.GetType().Name} berhenti
berlangganan.");
    }

    public void PublishNews(string news)
    {
        Console.WriteLine($"\\n[NewsAgency] Berita Terbaru: {news}");
        latestNews = news;
        Notify();
    }

    public void Notify()
    {
        foreach (var subscriber in subscribers)
        {
            subscriber.Update(latestNews);
        }
    }
}
```

EmailSubscriber.cs (Observer 1)

```
using System;

public class EmailSubscriber : ISubscriber
{
    private string name;

    public EmailSubscriber(string name)
    {
        this.name = name;
    }

    public void Update(string message)
    {
        Console.WriteLine($"[Email untuk {name}] Notifikasi: {message}");
    }
}
```

SmsSubscriber.cs (Observer 2)

```
using System;

public class SmsSubscriber : ISubscriber
{
    private string name;

    public SmsSubscriber(string name)
    {
        this.name = name;
    }
}
```

```

    }

    public void Update(string message)
    {
        Console.WriteLine($"[SMS untuk {name}] Notifikasi: {message}");
    }
}

```

Program.cs (Main)

```

using System;

class Program
{
    static void Main(string[] args)
    {
        // Membuat Subject
        NewsAgency newsAgency = new NewsAgency();

        // Membuat Observers
        EmailSubscriber email = new EmailSubscriber("Zhafir");
        SmsSubscriber sms = new SmsSubscriber("Rafi");

        // Menambahkan Observer ke Subject
        newsAgency.Attach(email);
        newsAgency.Attach(sms);

        // Mengirim berita
        newsAgency.PublishNews("Banjir besar di Bandung!");
        newsAgency.PublishNews("Harga BBM naik per Juni!");

        // Menghapus salah satu subscriber
        newsAgency.Detach(email);

        // Berita ketiga, hanya SMS yang terima
        newsAgency.PublishNews("Gempa bumi terasa di Jakarta");
    }
}

```