

Jurnal Modul 13
Konstruksi Perangkat Lunak
PRODI REKAYASA PERANGKAT LUNAK

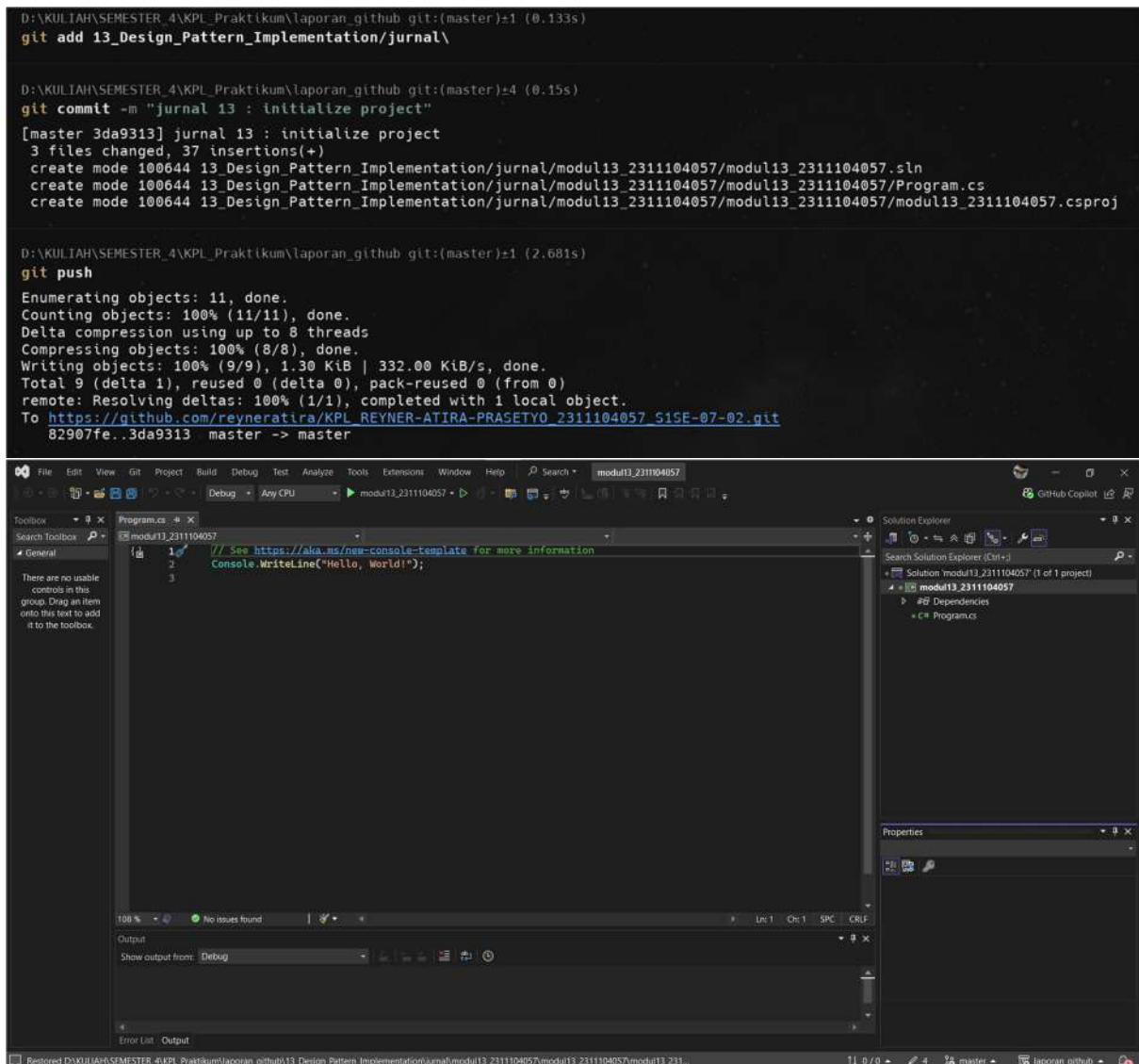


Nama :
Reyner Atira Prasetyo
2311104057
S1SE-07-02

DIREKTORAT TELKOM UNIVERSITY
PURWOKERTO

Jurnal Modul 13

1. Initialize project dan push



The image shows a terminal window and the Visual Studio IDE. The terminal window displays the following commands and output:

```
D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)+1 (0.133s)
git add 13_Design_Pattern_Implementation\jurnal\

D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)+4 (0.15s)
git commit -m "jurnal 13 : initialize project"

[master 3da9313] jurnal 13 : initialize project
3 files changed, 37 insertions(+)
create mode 100644 13_Design_Pattern_Implementation\jurnal\modul13_2311104057\modul13_2311104057.sln
create mode 100644 13_Design_Pattern_Implementation\jurnal\modul13_2311104057\modul13_2311104057\Program.cs
create mode 100644 13_Design_Pattern_Implementation\jurnal\modul13_2311104057\modul13_2311104057\modul13_2311104057.csproj

D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)+1 (2.681s)
git push

Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.30 KiB | 332.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/reynera1ra/KPL_REYNER-ATIRA-PRASETYO_2311104057_515E-07-02.git
82907fe..3da9313 master -> master
```

The Visual Studio IDE shows the project structure in the Solution Explorer, with the file `modul13_2311104057` selected. The Properties window shows the project's configuration. The Output window shows the command `Console.WriteLine("Hello, World!");` being executed.

2. Penjelasan tentang Singleton

A. Contoh Penggunaan Singleton :

1. Logging System – Untuk memastikan hanya satu instance logger yang mencatat seluruh proses aplikasi.
2. Database Connection – Untuk menjaga hanya satu koneksi database yang aktif di seluruh aplikasi.

B. Langkah Implementasi Singleton

1. Buat konstruktor privat untuk mencegah pembuatan objek dari luar kelas.
2. Buat atribut statik privat untuk menyimpan instance tunggal.
3. Buat method publik statik (`GetInstance()`) untuk mengembalikan instance yang sudah dibuat atau membuatnya jika belum ada.

C. Kelebihan dan Kekurangan Singleton

Kelebihan:

1. Kontrol Global atas instance yang sama.
2. Efisiensi Memori – hanya satu objek dibuat.
3. Mudah Diakses – cukup panggil instance secara global.

Kekurangan:

1. Sulit Diuji (Testing) karena sifat global dan dependensi tersembunyi.
2. Tidak Fleksibel – sulit digunakan di konteks multi-threading tanpa pengaturan tambahan.
3. Melanggar prinsip Single Responsibility – karena menangani logika pembuatan dan penyimpanan instance.

3. Membuat class Singleton



```
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

4 references
class PusatDataSingleton
{
    private static PusatDataSingleton _instance;
    6 references
    public List<string> DataTersimpan { get; private set; }

    // Monstruktur privat
    1 reference
    private PusatDataSingleton()
    {
        DataTersimpan = new List<string>();
    }

    0 references
    public static PusatDataSingleton GetDataSingleton()
    {
        if (_instance == null)
        {
            _instance = new PusatDataSingleton();
        }
        return _instance;
    }

    0 references
    public List<string> GetSemuaData()
    {
        return DataTersimpan;
    }

    0 references
    public void PrintSemuaData()
    {
        foreach (var data in DataTersimpan)
        {
            Console.WriteLine(data);
        }
    }

    0 references
    public void AddSebuahData(string input)
    {
        DataTersimpan.Add(input);
    }

    0 references
    public void MapusSebuahData(int index)
    {
        if (index >= 0 && index < DataTersimpan.Count)
        {
            DataTersimpan.RemoveAt(index);
        }
    }
}
```

Hasil : setiap instance objek PusatDataSingleton dapat mengakses dan mengubah data yang sama.

4. Implementasi di main Program

```
3 class Program
4 {
5     0 references
6     static void Main(string[] args)
7     {
8         // Inisialisasi data1 dan data2 dari Singleton
9         PusatDataSingleton data1 = PusatDataSingleton.GetDataSingleton();
10        PusatDataSingleton data2 = PusatDataSingleton.GetDataSingleton();
11
12        // Tambahkan nama anggota dan asisten
13        data1.AddSebuahData("Anggota 1: Aldi");
14        data1.AddSebuahData("Anggota 2: Budi");
15        data1.AddSebuahData("Asisten: Kak Fikri");
16
17        // Print data menggunakan data2
18        Console.WriteLine("Data melalui data2:");
19        data2.PrintSemuaData();
20
21        // Hapus data asisten pada data2
22        data2.HapusSebuahData(2); // index ke-2 (asisten)
23
24        // Print ulang dari data1
25        Console.WriteLine("\nData melalui data1 setelah penghapusan:");
26        data1.PrintSemuaData();
27
28        // Hitung jumlah data
29        Console.WriteLine($"Jumlah data di data1: {data1.GetSemuaData().Count}");
30        Console.WriteLine($"Jumlah data di data2: {data2.GetSemuaData().Count}");
31    }
32 }
```

Hasil yang diharapkan :

1. Data yang ditambahkan lewat data1 juga muncul saat diprint lewat data2.
2. Data yang dihapus lewat data2 juga hilang saat diprint lewat data1.
3. Jumlah data dari data1 dan data2 sama karena mereka merujuk ke instance yang sama (prinsip Singleton).

```
Microsoft Visual Studio Debug Console
Data melalui data2:
Anggota 1: Aldi
Anggota 2: Budi
Asisten: Kak Fikri

Data melalui data1 setelah penghapusan:
Anggota 1: Aldi
Anggota 2: Budi

Jumlah data di data1: 2
Jumlah data di data2: 2

D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github\13_Design_Pattern_Implementation\jurnal\modul13_2311104057\modul13_2311104057\bin\Debug\net8.0\modul13_2311104057.exe (process 21184) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

PUSH

```
D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±2 (0.081s)
git add 13_Design_Pattern_Implementation/jurnal\

D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±3 (0.146s)
git commit -m "jurnal 13 : implementasi singleton pattern"
[master 1b2ee48] jurnal 13 : implementasi singleton pattern
2 files changed, 87 insertions(+), 2 deletions(-)
create mode 100644 13_Design_Pattern_Implementation/jurnal/modul13_2311104057/modul13_2311104057/PusatDataSingleton.cs

D:\KULIAH\SEMESTER_4\KPL_Praktikum\laporan_github git:(master)±1 (2.372s)
git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.44 KiB | 1.44 MiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/reyneraatira/KPL_REYNER-ATIRA-PRASETYO_2311104057_SISE-07-02.git
3da9313..1b2ee48 master -> master
```