

**TUGAS JURNAL  
KONSTRUKSI PERANGKAT LUNAK**

**Pertemuan 13**

**CLEAN CODE**



**Disusun Oleh :**

**Ahmad Junaidi / 2211104002**

**SE-06-01**

**Asisten Praktikum:**

**Naufal El Kamil Aditya Pratama Rahman**

**Imelda**

**Dosen Pengampu :**

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

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## TUGAS JURNAL

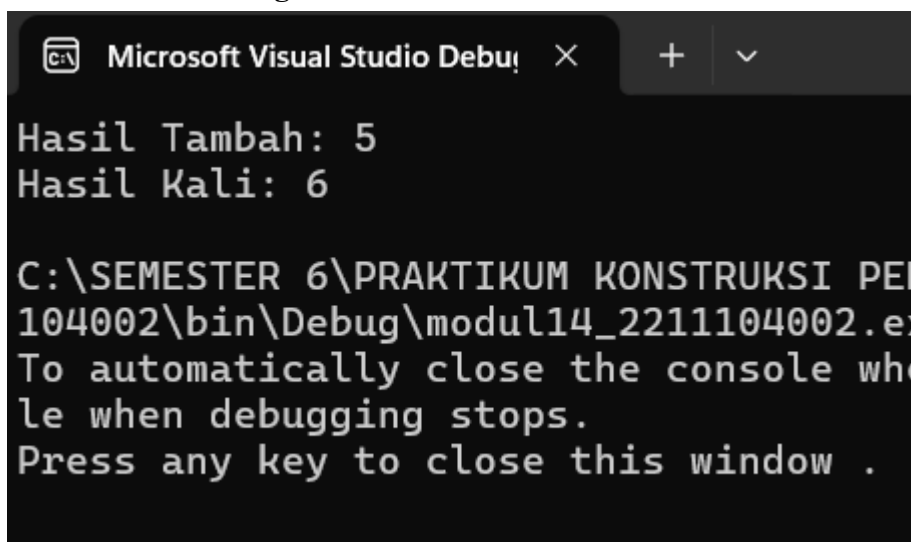
### 1. MEMBUAT PROJECT MODUL

- Copy salah satu folder tugas jurnal yang dimiliki sebelumnya (dari modul 2 sampai modul 13), kemudian rename folder hasil copy-paste tersebut dengan modul14\_NIM (coba pilih tugas pendahuluan yang paling sederhana)
- Misalnya menggunakan Visual Studio, bukalah project/folder yang di-copy sebelumnya.

SAYA MEMILIH TUGAS PENDAHULUAN YANG TP 14

### 2. Screenshoot Hasil Run

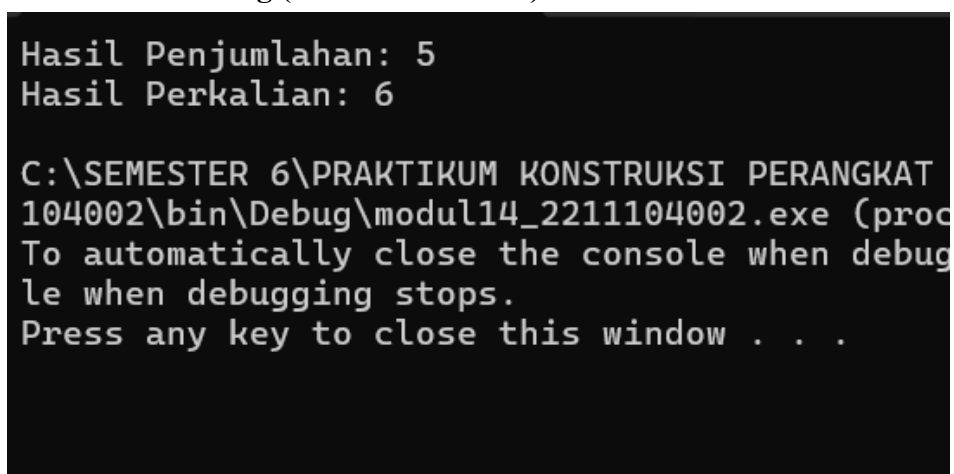
Sebelum Refactoring



```
Microsoft Visual Studio Debug Console
Hasil Tambah: 5
Hasil Kali: 6

C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT
104002\bin\Debug\modul14_2211104002.exe
To automatically close the console when debugging stops.
Press any key to close this window . . .
```

Setelah Refactoring (Standar C# .NET)



```
Hasil Penjumlahan: 5
Hasil Perkalian: 6

C:\SEMESTER 6\PRAKTIKUM KONSTRUKSI PERANGKAT
104002\bin\Debug\modul14_2211104002.exe (proc
To automatically close the console when debug
le when debugging stops.
Press any key to close this window . . .
```

### 3. Source Code: Sebelum dan Sesudah Refactoring

Sebelum Refactoring

Aljabar.cs

```
modul14_2211104002 AljabarApp.Aljabar
1  using System;
2
3  namespace AljabarApp
4  {
5      0 references
6      class Aljabar
7      {
8          1 reference
9          public int tambah(int x, int y)
10         {
11             return x + y;
12         }
13         1 reference
14         public int kali(int x, int y)
15         {
16             return x * y;
17         }
18     }
19 }
```

### Program.cs:

```
modul14_2211104002 AljabarApp.Program
1  using System;
2
3  namespace AljabarApp
4  {
5      0 references
6      class Program
7      {
8          0 references
9          static void Main(string[] args)
10         {
11             Aljabar a = new Aljabar();
12             Console.WriteLine("Hasil Tambah: " + a.tambah(2, 3));
13             Console.WriteLine("Hasil Kali: " + a.kali(2, 3));
14         }
15     }
16 }
```

Setelah Refactoring (Standar C# .NET)

Aljabar.cs

```
modul14_2211104002 AljabarApp.Aljabar
1 using System;
2
3 namespace AljabarApp
4 {
5     /// <summary>
6     /// Berisi metode matematika dasar.
7     /// </summary>
8     2 references
9     public class Aljabar
10    {
11        /// <summary>
12        /// Menambahkan dua bilangan bulat.
13        /// </summary>
14        0 references
15        public int Add(int number1, int number2)
16        {
17            return number1 + number2;
18        }
19
20        /// <summary>
21        /// Mengalikan dua bilangan bulat.
22        /// </summary>
23        0 references
24        public int Multiply(int number1, int number2)
25        {
26            return number1 * number2;
27        }
28    }
29 }
```

## Program.cs

```
modul14_2211104002 AljabarApp.Program
1 using System;
2
3 namespace AljabarApp
4 {
5     /// <summary>
6     /// Kelas utama untuk menampilkan operasi aljabar.
7     /// </summary>
8     0 references
9     public class Program
10    {
11        0 references
12        public static void Main(string[] args)
13        {
14            Aljabar calculator = new Aljabar();
15
16            Console.WriteLine("Hasil Penjumlahan: " + calculator.Add(2, 3));
17            Console.WriteLine("Hasil Perkalian: " + calculator.Multiply(2, 3));
18        }
19    }
20 }
```

## penjelasan:

Pada Jurnal Modul 14 ini, saya melakukan proses refactoring terhadap kode program C# sederhana yang sebelumnya telah dibuat pada modul sebelumnya. Refactoring

dilakukan dengan mengacu pada standar penulisan kode .NET agar kode lebih rapi, mudah dibaca, dan sesuai praktik terbaik. Langkah pertama yang saya lakukan adalah memperbaiki naming convention pada variabel, method, dan objek. Misalnya, method tambah diubah menjadi Add, dan kali menjadi Multiply, agar sesuai dengan penamaan PascalCase yang umum digunakan dalam bahasa C#. Selain itu, variabel lokal seperti x dan y diganti menjadi number1 dan number2 agar lebih deskriptif dan mudah dipahami.

Saya juga menata kembali white space dan indentation pada setiap baris kode agar struktur kode terlihat lebih teratur, menggunakan indentasi 4 spasi sesuai standar. Komentar dalam format XML (///) ditambahkan pada class dan setiap method untuk memberikan penjelasan mengenai fungsionalitasnya, sehingga dokumentasi kode menjadi lebih lengkap. Terakhir, saya melakukan uji coba run program sebelum dan sesudah refactoring, dan memastikan bahwa program tetap berjalan sesuai fungsinya, yaitu menampilkan hasil penjumlahan dan perkalian dua bilangan. Melalui refactoring ini, kualitas kode menjadi lebih baik dan lebih mudah untuk dikembangkan di masa mendatang.