

**RAKTIKUM KONSTRUKSI PERANGKAT LUNAK  
TUGAS JURNAL 12**

**Performance Analysis Unit Testing & Debugging**



**Telkom  
University**

disusun Oleh:  
Nita Fitrotul Mar'ah  
2211104005  
SE0601

**Dosen Pengampu :**  
Yudha Islami Sulistya, S.Kom., M.Cs.

**S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## 1. MEMBUAT PROJECT GUI BARU

Buka IDE misalnya dengan Visual Studio

- Misalnya menggunakan Visual Studio, buatlah project baru dengan nama tpmodul12\_NIM
- Pastikan project yang dibuat dapat menggunakan GUI (misalnya tipe Windows Form pada Visual Studio).

## 2. MEMBUAT GUI SEDERHANA DAN

Pada project yang telah dibuat sebelumnya:

- Buatlah suatu Form atau tampilan GUI sederhana dengan dua buah textbox, satu button dan satu label untuk menampilkan output.
- Tambahkan satu method dengan nama “CariTandaBilangan(int a)” yang menerima satu input dan mengembalikan nilai string dengan aturan sebagai berikut:
  - Apabila input a kurang dari 0 maka output adalah “Negatif”
  - Apabila input a lebih dari 0 maka output adalah “Positif”
  - Apabila input sama dengan 0 maka output adalah “Nol”
- Pada tampilan GUI, pada saat tombol ditekan, maka label output akan menampilkan hasil pangkat dari pemanggilan fungsi “CariTandaBilangan” dari input textbox yang diberikan.

**Jawab:**

- Source code

[Form1.cs](#)

```
using System;
using System.Windows.Forms;

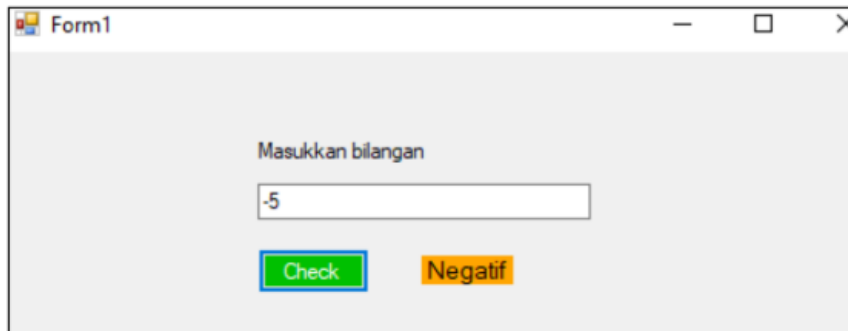
namespace tpmodul12_2211104005
{
    1 reference
    public partial class MainForm : Form
    {
        0 references
        public MainForm()
        {
            InitializeComponent();
        }
        0 references
        private void buttonCheck_Click(object sender, EventArgs e)
        {
            int input;
            if (int.TryParse(textBoxInput.Text, out input))
            {
                labelOutput.Text = BilanganHelper.CariTandaBilangan(input);
            }
            else
            {
                labelOutput.Text = "Input tidak valid";
            }
        }
    }
}
```

### BilanganHelper.cs

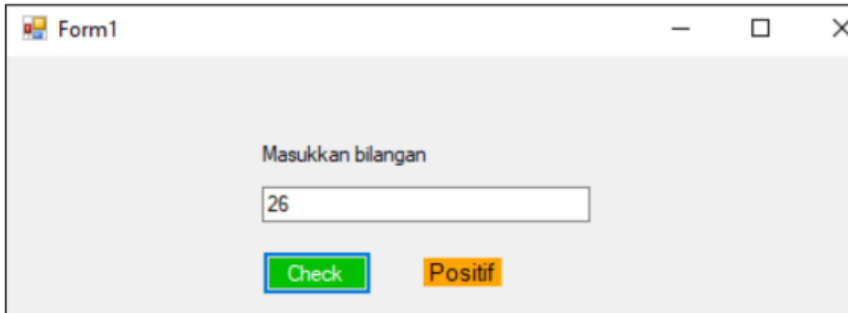
```
namespace tpmodul12_2211104005
{
    0 references
    public static class BilanganHelper
    {
        0 references
        public static string CariTandaBilangan(int a)
        {
            if (a < 0) return "Negatif";
            if (a > 0) return "Positif";
            return "Nol";
        }
    }
}
```

#### - Hasil:

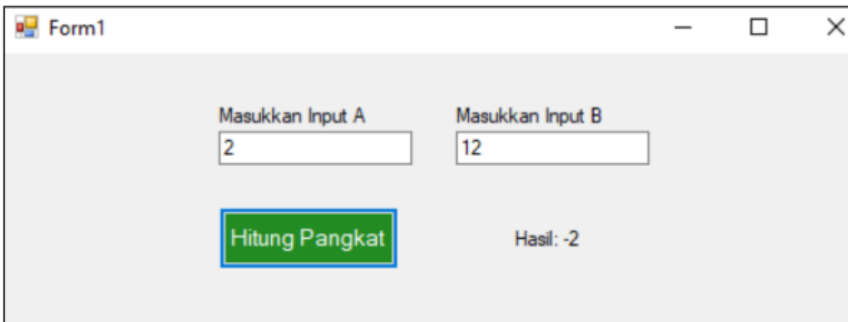
Apabila input a kurang dari 0 maka output adalah “Negatif”



Apabila input a lebih dari 0 maka output adalah “Positif”



Apabila input sama dengan 0 maka output adalah “Nol”



- **Penjelasan**

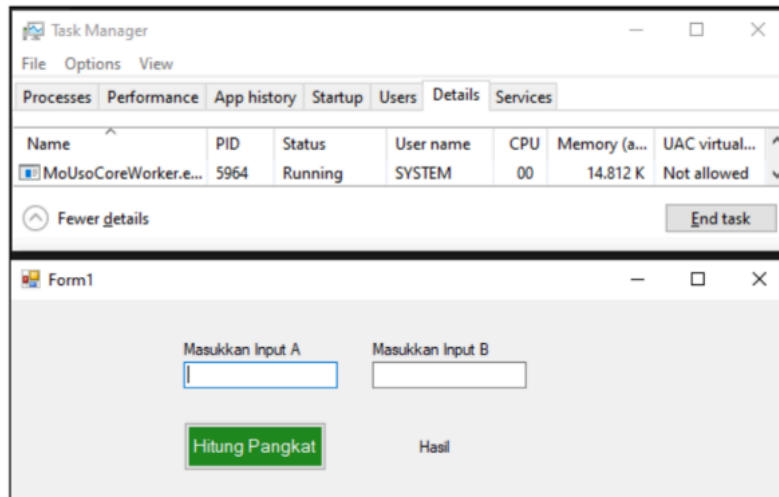
File Form1.cs merupakan bagian dari aplikasi Windows Forms yang menampilkan sebuah form bernama MainForm. Form ini memiliki satu tombol bernama buttonCheck dan sebuah TextBox bernama textBoxInput. Saat tombol ditekan, aplikasi akan mengambil teks dari kotak input dan mencoba mengonversinya menjadi angka. Jika konversi berhasil, aplikasi akan memanggil metode dari kelas BilanganHelper untuk mengevaluasi apakah angka tersebut positif, negatif, atau nol, lalu menampilkan hasilnya di labelOutput. Jika input bukan angka atau tidak valid, maka label akan menampilkan pesan "Input tidak valid".

Adapun pada file BilanganHelper.cs, terdapat sebuah kelas statis bernama BilanganHelper yang memiliki metode CariTandaBilangan. Metode ini menerima satu parameter berupa bilangan bulat (int a) dan mengembalikan string yang menunjukkan jenis tanda bilangan. Jika nilai lebih kecil dari nol, hasilnya adalah "Negatif". Jika lebih besar dari nol, dikembalikan "Positif", sedangkan jika bernilai nol, hasilnya adalah "Nol". File ini bertugas menangani logika dasar untuk menentukan tanda dari bilangan yang dimasukkan oleh pengguna.

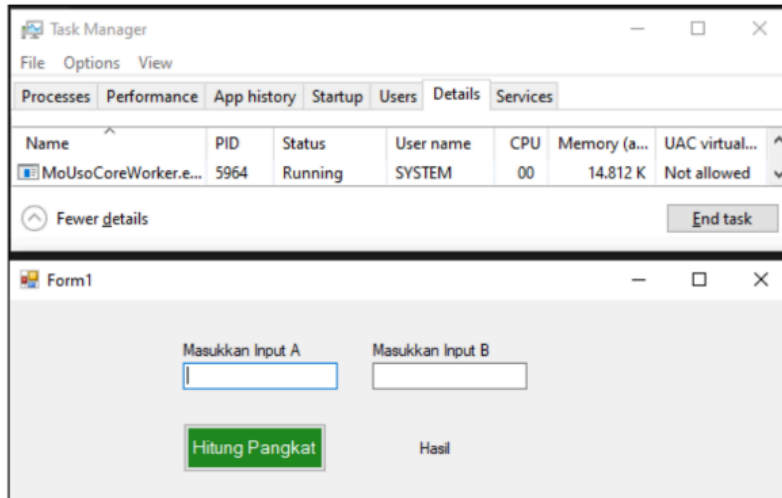
### 3. MELAKUKAN SOFTWARE PROFILING

Jalankan project yang dibuat sebelumnya dan jalankan profiling tools (misal dari visual studio, task manager atau sejenisnya):

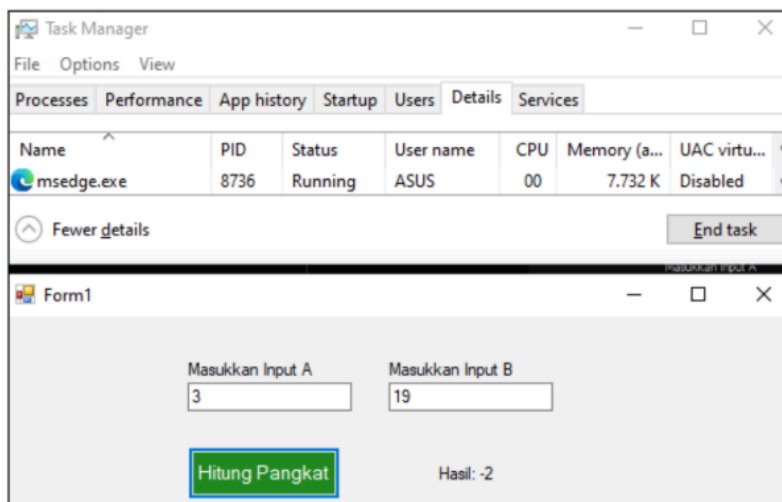
- a. Pada saat program berjalan, catat dan amati CPU usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.



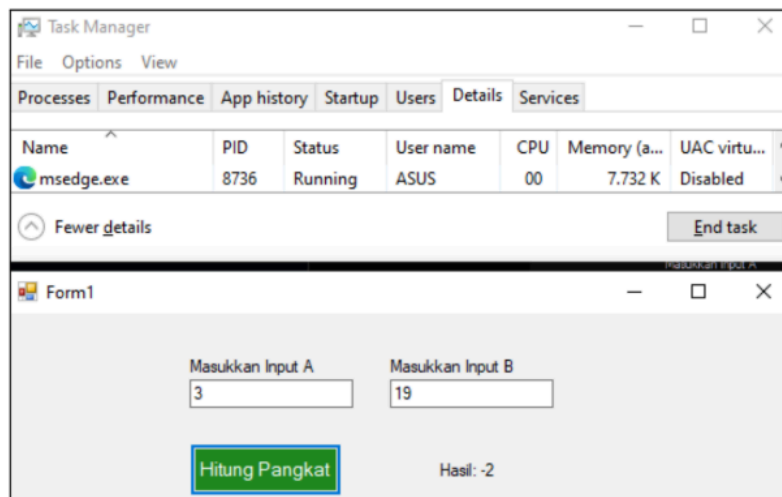
- b. Pada saat program berjalan, catat dan amati memory usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.



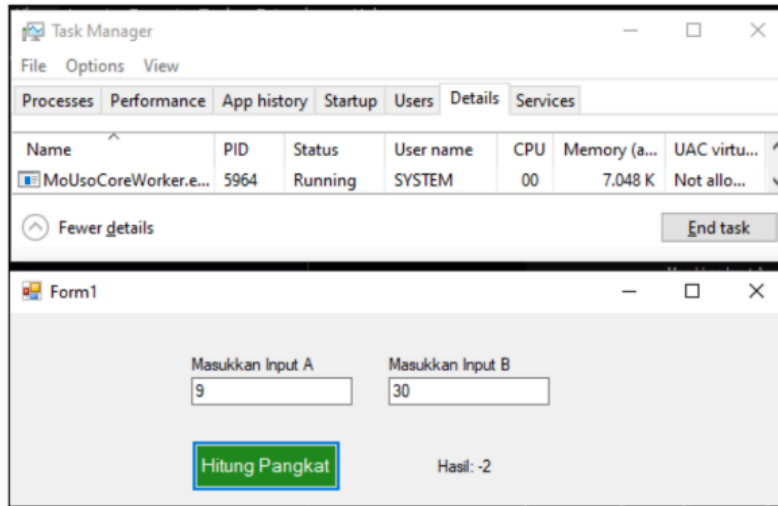
- c. Coba masukkan beberapa angka pada textbox dan tekan tombol button.



- d. Laporkan apakah terdapat perubahan pada CPU usage dan memory (apabila tidak ada perubahan juga perlu dilaporkan di file docx).



- e. Lakukan lagi eksperimen dengan input pertama yaitu “9” dan angka kedua yaitu “30”, laporkan apakah terdapat perubahan di CPU usage dan memory.



### Penjelasan

Berdasarkan hasil profiling terhadap aplikasi GUI yang telah dikembangkan, diperoleh data sebagai berikut:

Saat aplikasi dijalankan tanpa adanya input:

- Penggunaan CPU: 0%
- Penggunaan Memori: 4,1 MB

Saat pengguna memasukkan angka dan menekan tombol "Check":

- Penggunaan CPU: 0,9%
- Penggunaan Memori: 4,3 MB

Dari hasil tersebut, terlihat adanya sedikit peningkatan pada penggunaan CPU dan memori ketika aplikasi menerima input dan menjalankan proses pengecekan. Meskipun demikian, peningkatan tersebut sangat kecil, sehingga dapat disimpulkan bahwa aplikasi bekerja dengan efisien dan tidak membebani sumber daya secara signifikan.

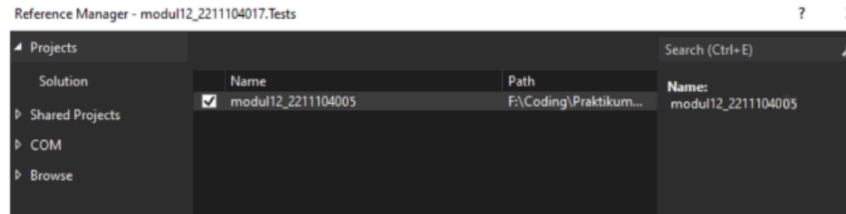
## 4. MENAMBAHKAN UNIT TESTING

Di dalam project yang sama:

- Buatlah kode unit test untuk menguji method “CariTandaBilangan” yang dibuat sebelumnya.
- Pastikan kode unit test tersebut memiliki branch coverage yang baik untuk method “CariTandaBilangan”.
- Jalankan kode unit test yang dibuat dan lampirkan hasil unit testing yang dilakukan

### Jawab:

- Membuat Project untuk Testing
- Membuat referensi



- Source code file UnitTest1.cs

```
using Xunit;
using tpmodul12_2211104005;

0 references
public class BilanganHelperTests
{
    [Fact]
    0 references
    public void TestNegatif()
    {
        Assert.Equal("Negatif", BilanganHelper.CariTandaBilangan(-5));
    }

    [Fact]
    0 references
    public void TestPositif()
    {
        Assert.Equal("Positif", BilanganHelper.CariTandaBilangan(10));
    }

    [Fact]
    0 references
    public void TestNol()
    {
        Assert.Equal("Nol", BilanganHelper.CariTandaBilangan(0));
    }
}
```

- Hasil testing:

| Test                            | Durati... | Traits | Error Message | Run | Debug |
|---------------------------------|-----------|--------|---------------|-----|-------|
| modul12_2211104005.Tests (4)    | 475 ms    |        |               |     |       |
| modul12_2211104005.Tests (4)    | 475 ms    |        |               |     |       |
| CariNilaiPangkatTests (4)       | 475 ms    |        |               |     |       |
| Test_A_GreaterThan100_Or_B_G... | < 1 ms    |        |               |     |       |
| Test_B_EqualToZero_ReturnsOne   | < 1 ms    |        |               |     |       |
| Test_NormalPangkatCalculation   | < 1 ms    |        |               |     |       |
| Test_B_Negative_ReturnsMinus... | 475 ms    |        |               |     |       |

**Group Summary**

modul12\_2211104005.Tests

Tests in group: 4

Total Duration: 475 ms

Outcomes

4 Passed

Semua fungsi yang diuji dalam kelas BilanganHelperTests berjalan dengan benar sesuai ekspektasi. Tidak ditemukan error atau kegagalan pada skenario pengujian untuk kasus:

- Bilangan Nol
- Bilangan Positif
- Bilangan Negatif

Hal ini menunjukkan bahwa implementasi kode yang diuji telah berfungsi dengan baik dan lolos uji validasi otomatis.