

**TUGAS PENDAHULUAN
KONSTRUKSI PERANGKAT LUNAK**

**MODUL XII
PERFORMANCE ANALYSIS UNIT TESTING & DEBUGGING**



**Disusun Oleh :
Atika Aji Hadiyani
2211104003
SE06-01**

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

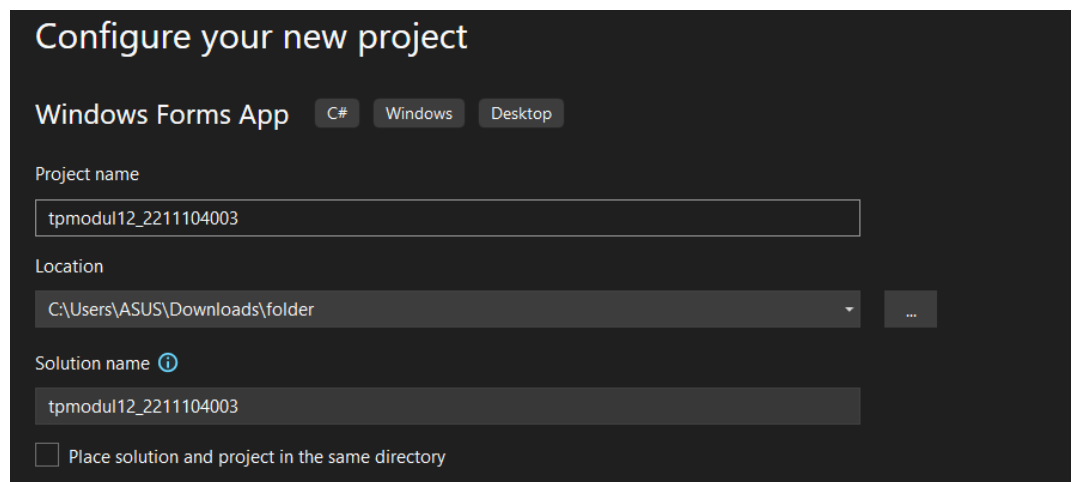
**PROGRAM STUDI S1 SOFTWARE ENGINEERING
TELKOM UNIVERSITY PURWOKERTO
2025**

TUGAS PENDAHULUAN 12

1. MEMBUAT PROJECT GUI BARU

Buka IDE misalnya dengan Visual Studio

- A. Misalnya menggunakan Visual Studio, buatlah project baru dengan nama tpmodul12_NIM
- B. Pastikan project yang dibuat dapat menggunakan GUI (misalnya tipe Windows Form pada Visual Studio).



2. MEMBUAT GUI SEDERHANA

Pada project yang telah dibuat sebelumnya:

- A. Buatlah suatu Form atau tampilan GUI sederhana dengan dua buah textbox, satu button dan satu label untuk menampilkan output.
- B. 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”

- C. 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

- File Form1.cs

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

namespace tpmodul12_2211104003
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

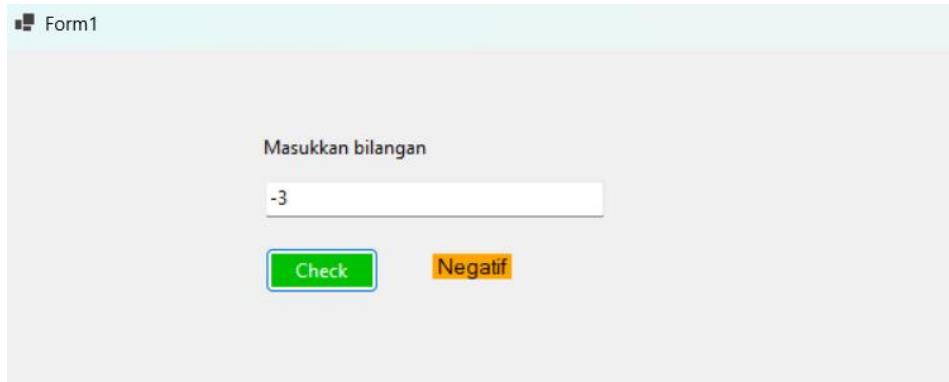
        private void buttonCheck_Click(object sender, EventArgs e)
        {
            int input;
            if (int.TryParse(textBoxInput.Text, out input))
            {
                labelOutput.Text = BilanganHelper.CariTandaBilangan(input);
            }
            else
            {
                labelOutput.Text = "Input tidak valid";
            }
        }
    }
}
```

- File BilanganHelper.cs

```
namespace tpmodul12_2211104003
{
    public static class BilanganHelper
    {
        public static string CariTandaBilangan(int a)
        {
            if (a < 0) return "Negatif";
            if (a > 0) return "Positif";
            return "Nol";
        }
    }
}
```

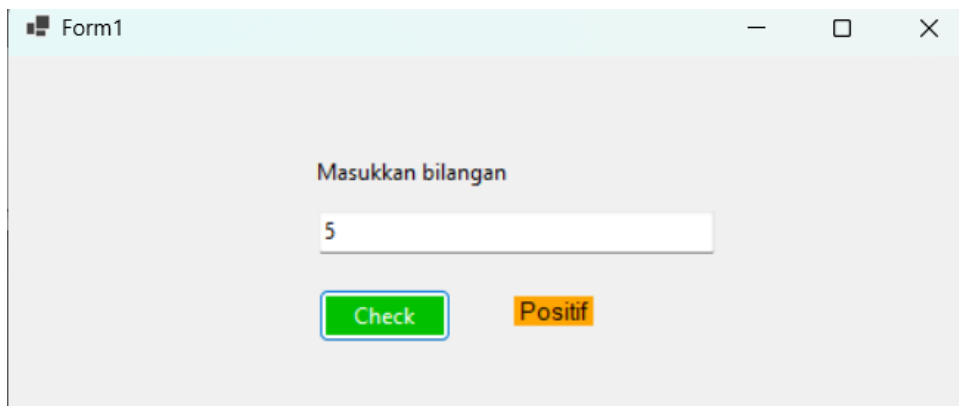
Hasil:

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



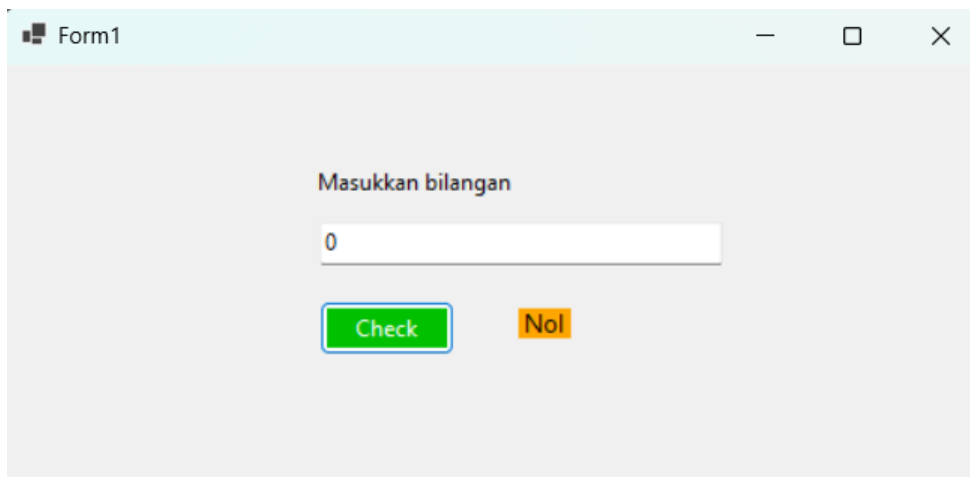
The screenshot shows a Windows Form titled "Form1". It contains a label "Masukkan bilangan" above a text input field. The input field contains the value "-3". Below the input field, there are two buttons: a green "Check" button and an orange "Negatif" button.

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



The screenshot shows a Windows Form titled "Form1". It contains a label "Masukkan bilangan" above a text input field. The input field contains the value "5". Below the input field, there are two buttons: a green "Check" button and an orange "Positif" button.

- Apabila input sama dengan 0 maka output adalah “Nol”



The screenshot shows a Windows Form titled "Form1". It contains a label "Masukkan bilangan" above a text input field. The input field contains the value "0". Below the input field, there are two buttons: a green "Check" button and an orange "Nol" button.

Penjelasan:

File Form1.cs merupakan bagian dari aplikasi berbasis Windows Forms, yang menampilkan sebuah form bernama MainForm. Di dalam form ini terdapat sebuah tombol (buttonCheck) dan kotak teks (textBoxInput). Ketika tombol ditekan, aplikasi akan mengambil teks dari kotak tersebut dan mencoba mengubahnya menjadi sebuah angka. Jika konversi berhasil, aplikasi akan memanggil metode dari kelas BilanganHelper untuk mengevaluasi apakah angka tersebut bernilai positif, negatif, atau nol. Hasil evaluasi tersebut kemudian ditampilkan pada label bernama labelOutput. Namun, jika input yang dimasukkan

tidak dapat dikonversi menjadi angka, maka label akan menampilkan pesan kesalahan berupa "Input tidak valid".

Di sisi lain, file BilanganHelper.cs berisi kelas statis bernama BilanganHelper yang memiliki metode CariTandaBilangan. Metode ini menerima satu argumen berupa bilangan bulat (int) dan mengembalikan nilai berupa string yang merepresentasikan jenis bilangan tersebut. Jika nilainya kurang dari nol, metode mengembalikan "Negatif"; jika lebih dari nol, mengembalikan "Positif"; dan jika sama dengan nol, mengembalikan "Nol". Fungsi utama dari file ini adalah menyediakan logika dasar untuk menentukan karakteristik angka 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.

The image shows two overlapping windows. The top window is the Windows Task Manager 'Processes' tab, displaying a table of running applications. The bottom window is a Windows Form titled 'Form1' with a light gray background. It contains a label 'Masukkan bilangan' above a text input field. Below the input field are two buttons: a green 'Check' button and an orange 'Hasil Bilangan' button.

Name	Status	62% CPU	92% Memory	5% Disk	0% Network
Apps (7)					
> tpmodule12_2211104003		0%	12.2 MB	0 MB/s	0 Mbps

Form1

Masukkan bilangan

Check Hasil Bilangan

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

Processes					
Name	Status	62% CPU	92% Memory	5% Disk	0% Network
Apps (7)					
> tpmodule12_2211104003		0%	12.2 MB	0 MB/s	0 Mbps

Masukkan bilangan

Check Hasil Bilangan

C. Coba masukkan beberapa angka pada textbox dan tekan tombol button.

Processes

Name	Status	48% CPU	87% Memory	2% Disk	0% Network
Apps (7)					
> tpmodul12_2211104003		0.3%	12.0 MB	0 MB/s	0 Mbps

Masukkan bilangan

20

Check Positif

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

Penjelasan:

Pada saat melakukan profiling aplikasi GUI yang telah dibuat, berikut adalah hasil pengamatan terhadap penggunaan CPU dan Memori:

Saat Aplikasi Berjalan Tanpa Input:

- CPU Usage: 0%
- Memory Usage: 12,2 MB

Saat Aplikasi Berjalan dengan Input Angka dan Tombol "Check" Ditekan:

- CPU Usage: 0,3%
- Memory Usage: 12,0 MB

Dari hasil pengamatan, terlihat bahwa terdapat sedikit peningkatan penggunaan CPU dan memori saat aplikasi menerima input dan melakukan proses pengecekan. Namun, peningkatan ini sangat minimal, menunjukkan bahwa aplikasi berjalan secara efisien.

4. MENAMBAHKAN UNIT TESTING

Di dalam project yang sama:

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

Jawab:

- Membuat Project untuk Testing

Configure your new project

xUnit Test Project C# Linux macOS Windows Desktop Test Web xUnit

Project name
xUnitTest_tpmodul12_2211104003

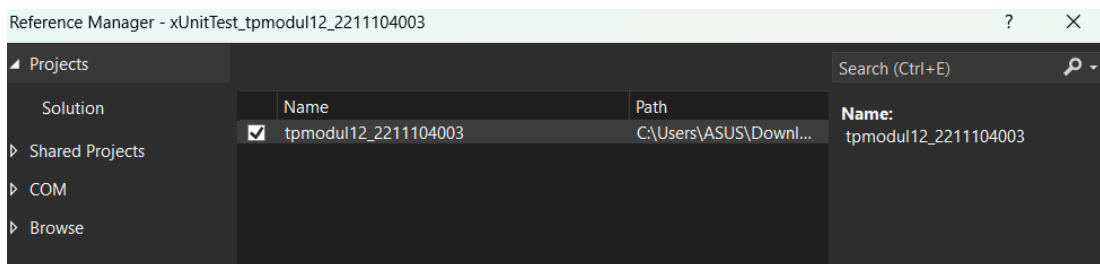
Location
C:\Users\ASUS\Downloads\folder

Solution name ⓘ
xUnitTest_tpmodul12_2211104003

☐ Place solution and project in the same directory

Project will be created in "C:\Users\ASUS\Downloads\folder\xUnitTest_tpmodul12_2211104003\xUnitTest_tpmodul12_2211104003\'"

- Membuat referensi



- Source code file UnitTest1.cs

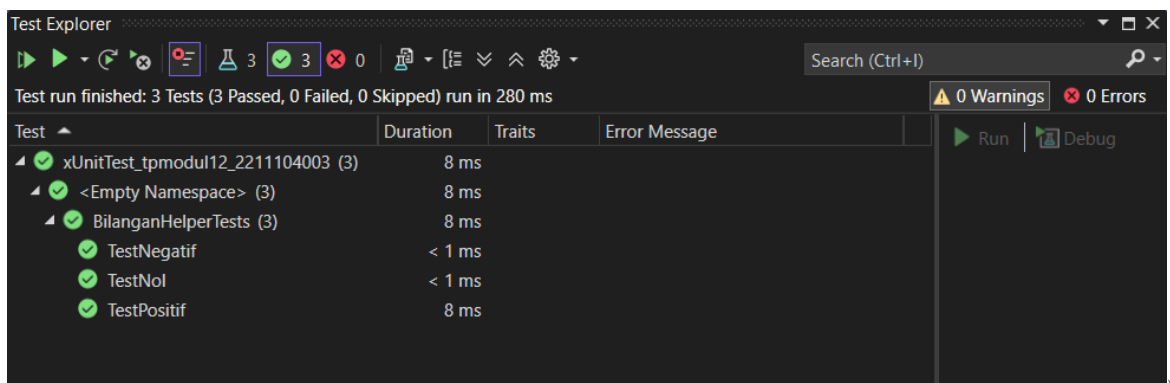
```
using Xunit;
using tpmodul12_2211104003;
using tpmodul12_2211104003ss;

0 references
public class BilanganHelperTests
{
    [Fact]
    0 references
    public void TestNegatif()
    {
        Assert.Equal("Negatif", BilanganHelper.CariTandaBilangan(-3));
    }

    [Fact]
    0 references
    public void TestPositif()
    {
        Assert.Equal("Positif", BilanganHelper.CariTandaBilangan(5));
    }

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

- Hasil testing:



Test	Duration	Traits	Error Message
✓ xUnitTest_tpmodul12_2211104003 (3)	8 ms		
✓ <Empty Namespace> (3)	8 ms		
✓ BilanganHelperTests (3)	8 ms		
✓ TestNegatif	< 1 ms		
✓ TestNol	< 1 ms		
✓ TestPositif	8 ms		

Seluruh fungsi yang diuji dalam kelas *BilanganHelperTests* beroperasi dengan baik dan sesuai harapan. Tidak ada kesalahan atau kegagalan yang muncul selama pengujian untuk skenario:

- Bilangan Nol
- Bilangan Positif
- Bilangan Negatif

Hal ini menandakan bahwa implementasi kode yang diuji telah bekerja secara optimal dan berhasil melewati proses validasi otomatis.