

**TUGAS PENDAHULUAN  
KONSTRUKSI PERANGKAT LUNAK**

**MODUL XII**

**PERFORMANCE ANALYSIS UNIT TESTING & DEBUGGING**



**Disusun Oleh :**

Nadia Putri Rahmانيar / 2211104012

S1 SE-06-01

**Dosen Pengampu :**

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

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

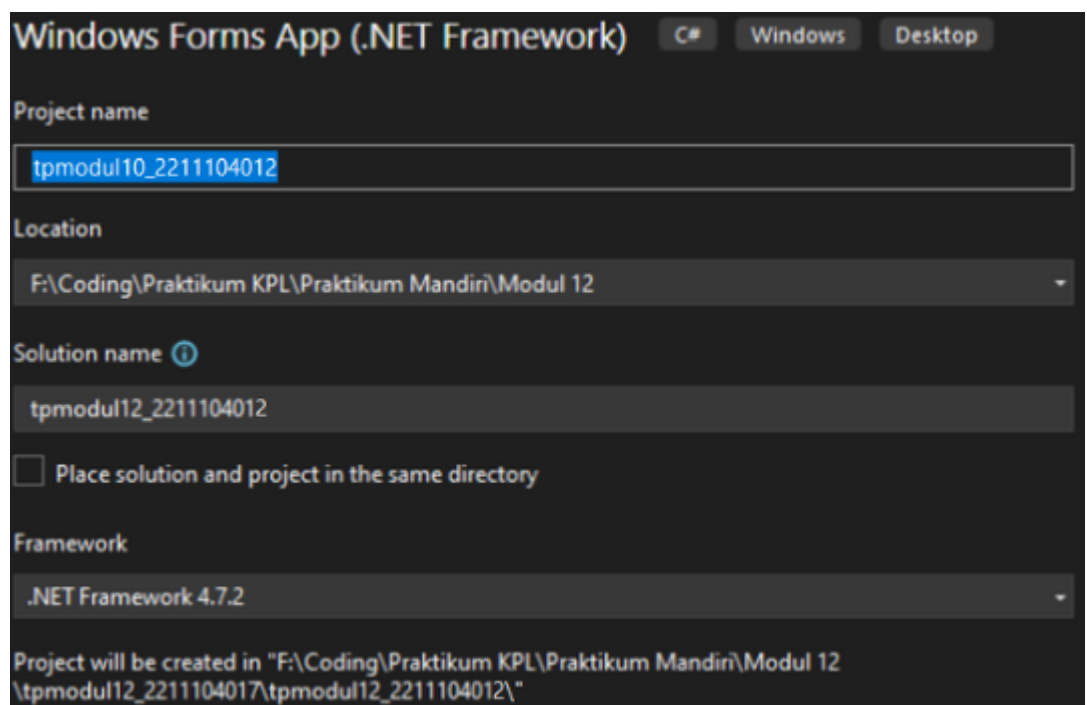
**2025**

## 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;

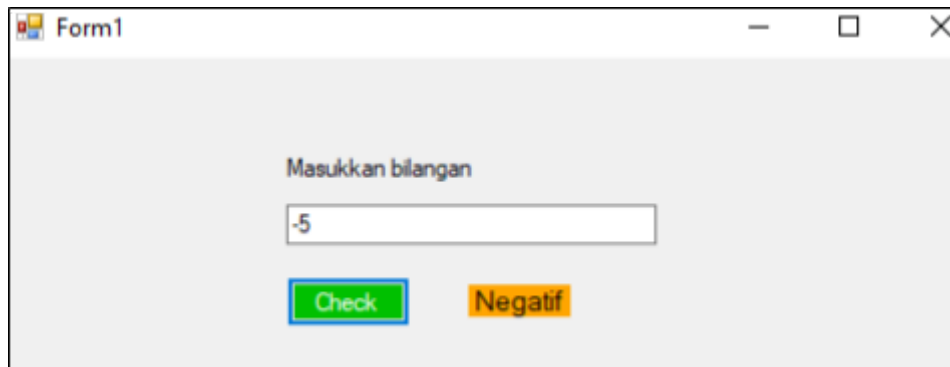
namespace tpmodul12_2211104012
{
    3 references
    public partial class MainForm : Form
    {
        1 reference
        public MainForm()
        {
            InitializeComponent();
        }
        1 reference
        private void buttonCheck_Click(object sender, EventArgs e)
        {
            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_2211104012
{
    4 references
    public static class BilanganHelper
    {
        4 references | 3/3 passing
        public static string CariTandaBilangan(int a)
        {
            if (a < 0) return "Negatif";
            if (a > 0) return "Positif";
            return "Nol";
        }
    }
}
```

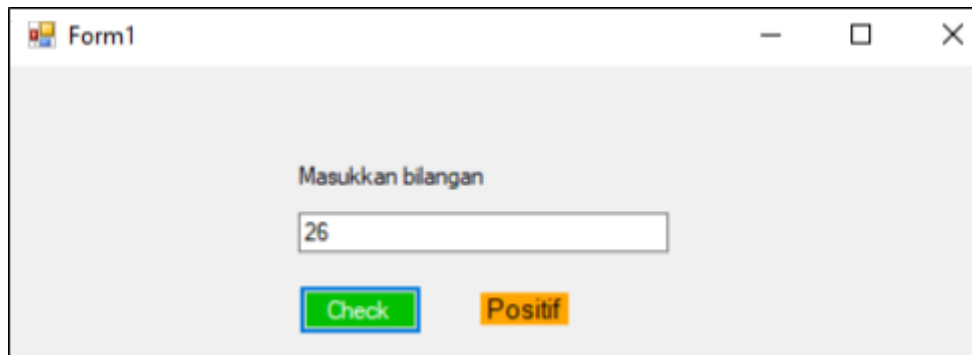
## Hasil:

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



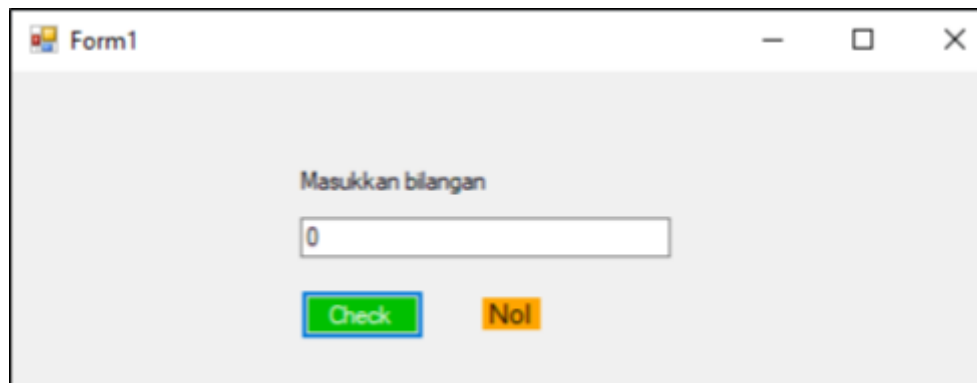
The screenshot shows a Windows Form titled 'Form1'. It contains a label 'Masukkan bilangan' (Enter a number) above a text box. The text box contains the value '-5'. Below the text box are two buttons: a green 'Check' button and an orange 'Negatif' button.

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



The screenshot shows a Windows Form titled 'Form1'. It contains a label 'Masukkan bilangan' (Enter a number) above a text box. The text box contains the value '26'. Below the text box 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' (Enter a number) above a text box. The text box contains the value '0'. Below the text box are two buttons: a green 'Check' button and an orange 'Nol' button.

## Penjelasan:

File Form1.cs merupakan komponen utama dari sebuah aplikasi Windows Forms, di mana sebuah formulir (MainForm) dilengkapi dengan tombol (buttonCheck) dan kotak teks (textBoxInput). Saat tombol tersebut diinteraksikan, aplikasi akan mengambil masukan teks dari kotak teks dan berupaya mengonversinya menjadi nilai numerik. Apabila konversi berhasil, aplikasi akan memanfaatkan metode yang disediakan oleh BilanganHelper untuk mengidentifikasi apakah angka tersebut positif, negatif, atau nol, kemudian menampilkan hasilnya pada label (labelOutput). Sebaliknya, jika masukan tidak valid (bukan angka), label akan memperlihatkan

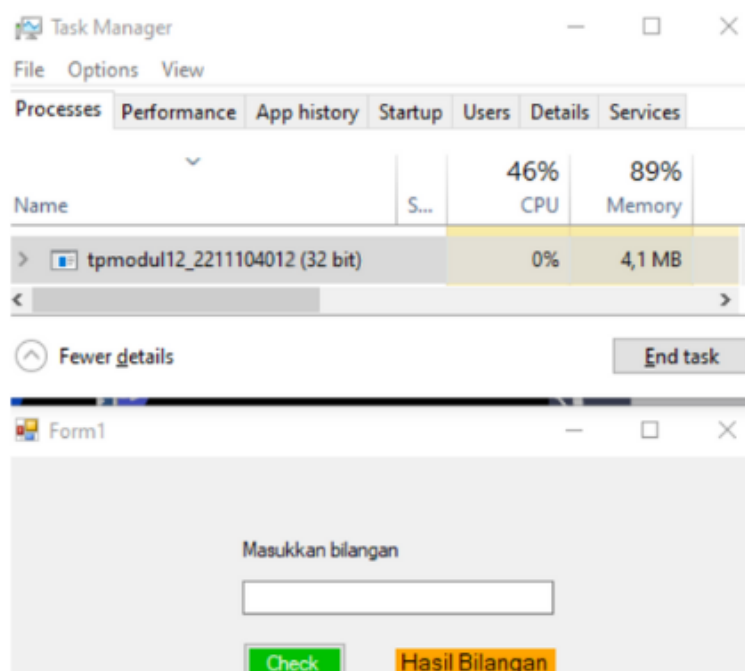
pesan "Input tidak valid".

Sementara itu, berkas BilanganHelper.cs mendefinisikan sebuah kelas statis bernama BilanganHelper, yang di dalamnya terdapat metode CariTandaBilangan. Metode ini menerima sebuah argumen berupa bilangan bulat (int a) dan mengembalikan representasi tekstual dari tanda bilangan tersebut. Jika angka yang diberikan kurang dari nol, metode akan mengembalikan "Negatif". Apabila lebih dari nol, ia akan mengembalikan "Positif". Dan jika sama dengan nol, akan mengembalikan "Nol". Berkas ini secara efektif berfungsi sebagai modul logika sederhana yang bertanggung jawab untuk memverifikasi tanda bilangan yang dimasukkan oleh pengguna.

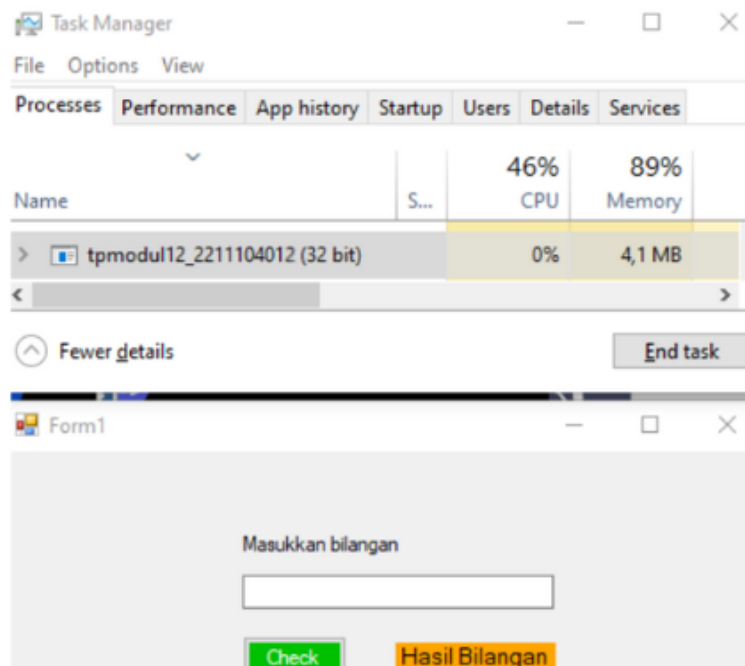
### 3. MELAKUKAN SOFTWARE PROFILING

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

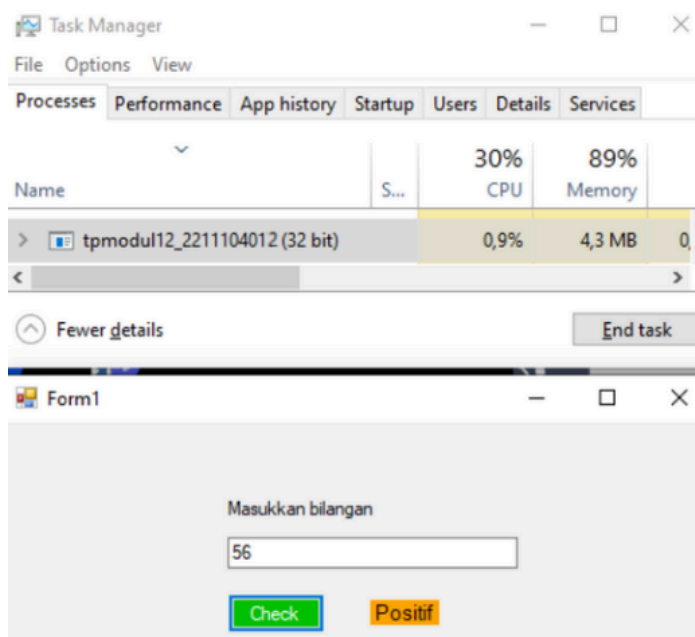
- A. Pada saat program berjalan, catat dan amati CPU usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.



- 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).

**Penjelasan:**

Dari hasil pemantauan kinerja aplikasi GUI yang telah dikembangkan, teramati pola penggunaan CPU dan memori sebagai berikut:

- Ketika aplikasi dalam kondisi idle (berjalan tanpa adanya interaksi pengguna), tercatat penggunaan CPU sebesar 0% dan konsumsi memori

sekitar 4,1 MB.

- Sebaliknya, saat aplikasi menerima masukan berupa angka dan tombol "Check" diaktifkan, terjadi sedikit kenaikan pada penggunaan CPU menjadi 0,9% dan memori sebesar 4,3 MB.

Peningkatan penggunaan CPU dan memori ini, meskipun ada, tergolong sangat kecil. Hal ini mengindikasikan bahwa aplikasi beroperasi dengan efisien, dengan sumber daya sistem yang dimanfaatkan secara optimal bahkan saat ada aktivitas pemrosesan input.

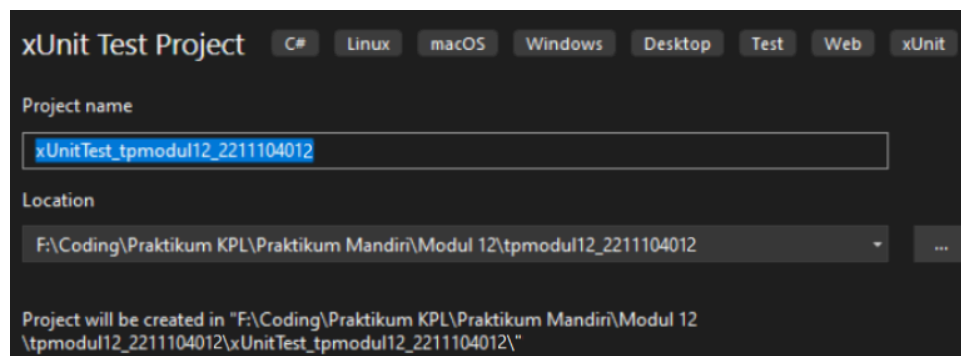
#### 4. MENAMBAHKAN UNIT TESTING

Di dalam project yang sama:

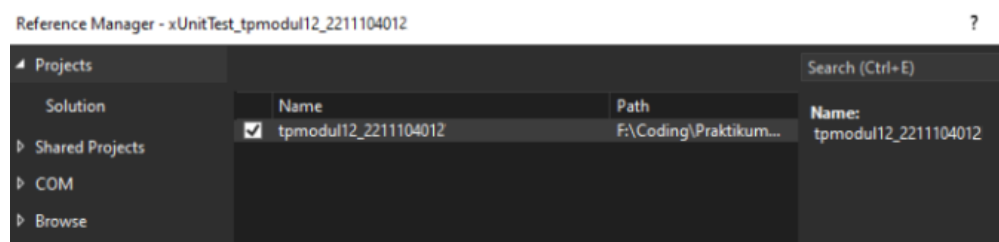
- A. Buatlah kode unit test untuk menguji method “CariTandaBilangan” yang dibuat sebelumnya.
- B. Pastikan kode unit test tersebut memiliki branch coverage yang baik untuk method “CariTandaBilangan”.
- C. Jalankan kode unit test yang dibuat dan lampirkan hasil unit testing yang dilakukan.

**Jawab:**

- Membuat Project untuk Testing



- Membuat referensi



- Source code file UnitTest1.cs

```
using Xunit;
using tpm modul12_2211104012;

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 Explorer

Test run finished: 3 Tests (3 Passed, 0 Failed, 0 Skipped) run in 1,1 sec

Test	Durati...	Traits	Error Mess...
✓ xUnitTest_tpm modul12_2211104012 (3)	539 ms		
✓ <Empty Namespace> (3)	539 ms		
✓ BilanganHelperTests (3)	539 ms		
✓ TestNol	< 1 ms		
✓ TestPositif	< 1 ms		
✓ TestNegatif	539 ms		

**Group Summary**

xUnitTest\_tpm modul12\_2211104012

Tests in group: 3

⌚ Total Duration: 539 ms

**Outcomes**

✓ 3 Passed

Seluruh fungsionalitas yang dievaluasi dalam kelas BilanganHelperTests terbukti beroperasi sesuai dengan hasil yang diharapkan. Selama proses pengujian, tidak ditemukan adanya kesalahan atau kegagalan pada skenario-skenario yang meliputi:

- Pengujian untuk bilangan nol
- Pengujian untuk bilangan positif
- Pengujian untuk bilangan negatif

Kesimpulan ini mengindikasikan bahwa implementasi kode yang telah diuji telah berfungsi secara akurat dan berhasil melewati tahap validasi otomatis.