

**TUGAS PENDAHULUAN
KONSTRUKSI PERANGKAT LUNAK**

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



Disusun Oleh :

Rizky Hanifa Afania

2211104017

SE-06-01

Dosen Pengampu :

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

PROGRAM STUDI S1 SOFTWARE ENGINEERING

TELKOM UNIVERSITY PURWOKERTO

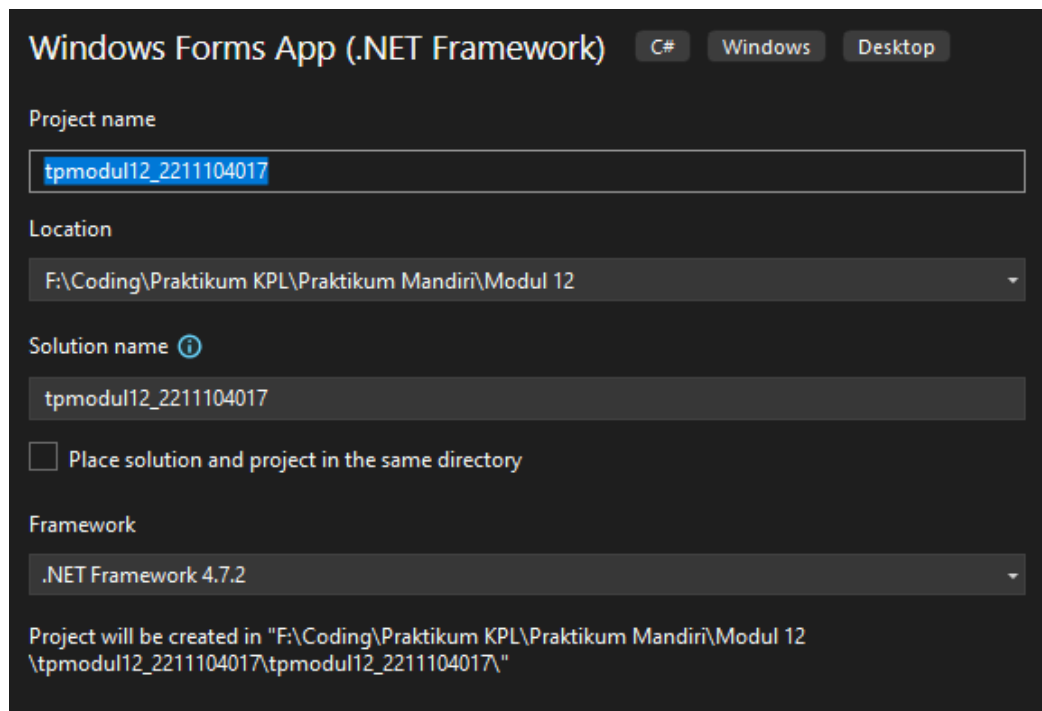
2025

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:

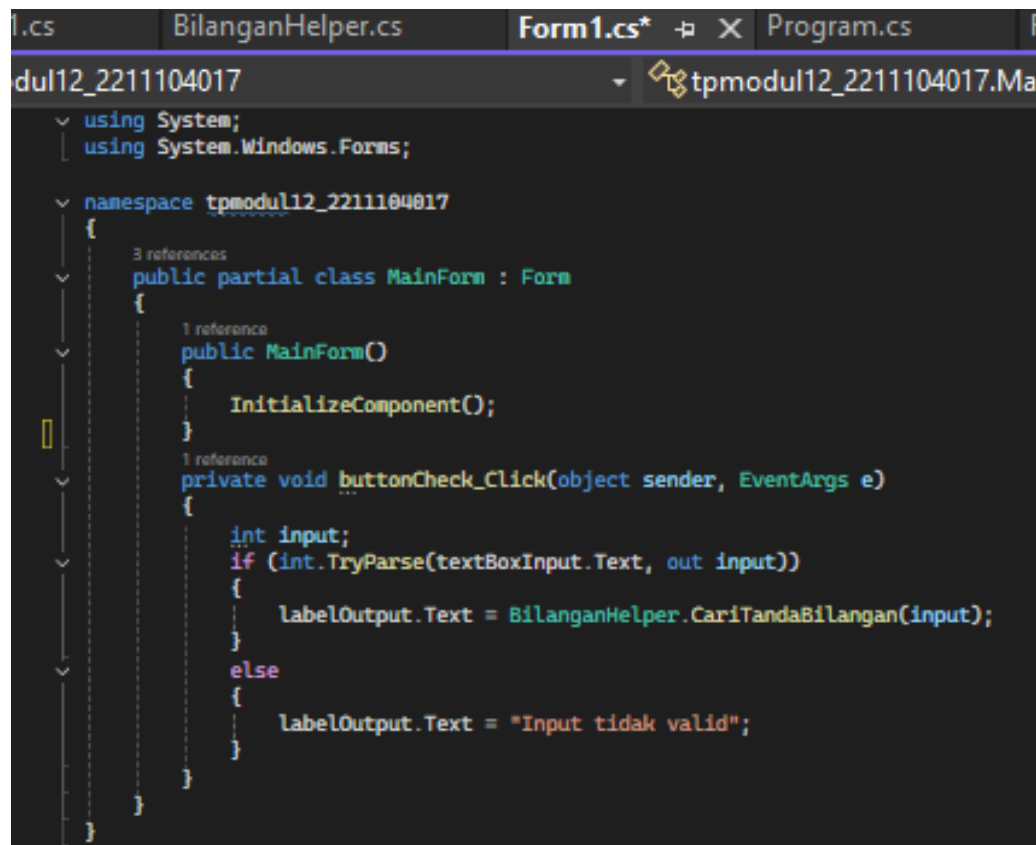
- A. Buatlah suatu Form atau tampilan GUI sederhana dengan dua buah textbox, satu button dan satu label untuk menampilkan output.
- B. Tambahkan satu method dengan nama "CariTandaBilangan(int a)" yang menerima satu input dan mengembalikan nilai string dengan aturan sebagai berikut:
 - Apabila input a 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

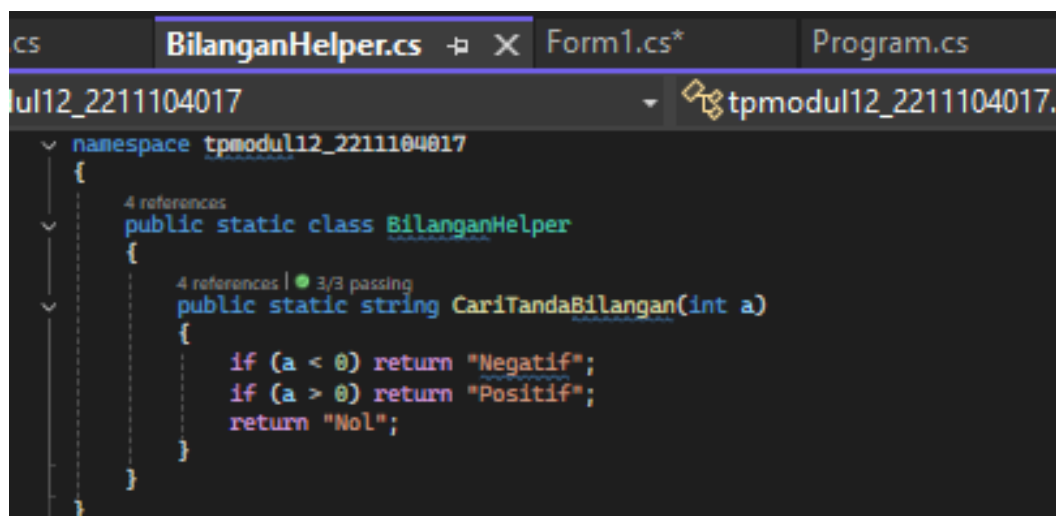


```
using System;
using System.Windows.Forms;

namespace tpmodul12_2211104017
{
    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

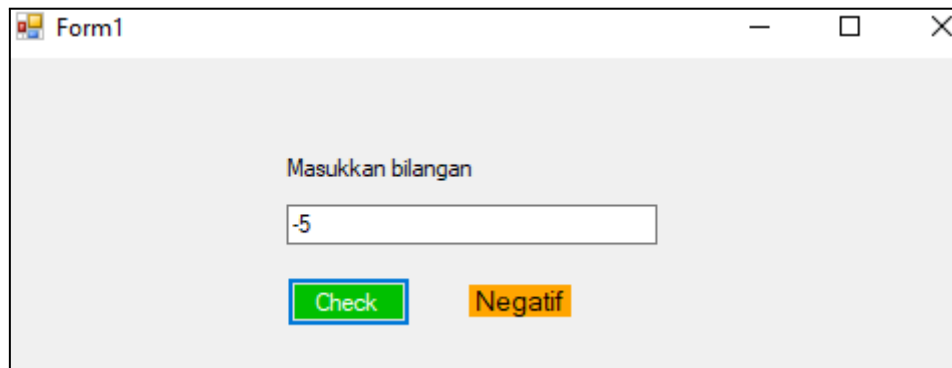


```
using System;

namespace tpmodul12_2211104017
{
    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";
        }
    }
}
```

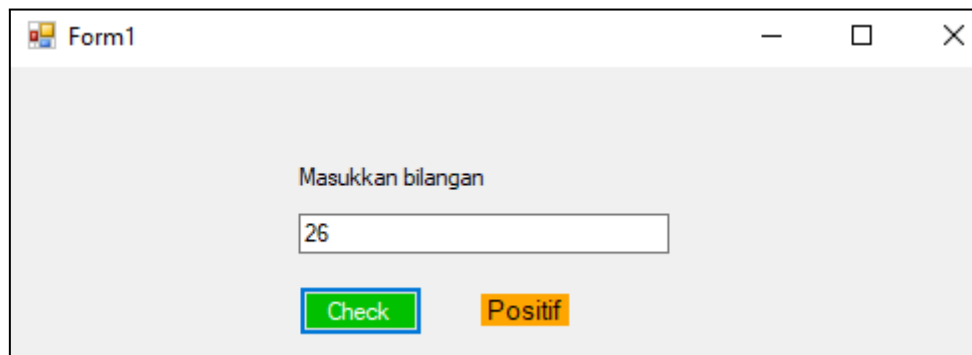
Hasil:

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



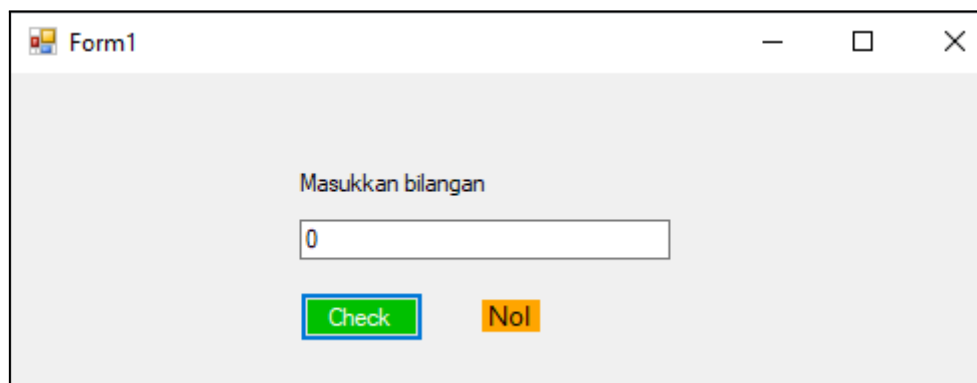
The screenshot shows a Windows Form titled 'Form1'. It contains a label 'Masukkan bilangan' above a text input field. The input field contains the value '-5'. Below the input field, there are two buttons: a green 'Check' button and an orange 'Negatif' button.

- 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 '26'. 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:

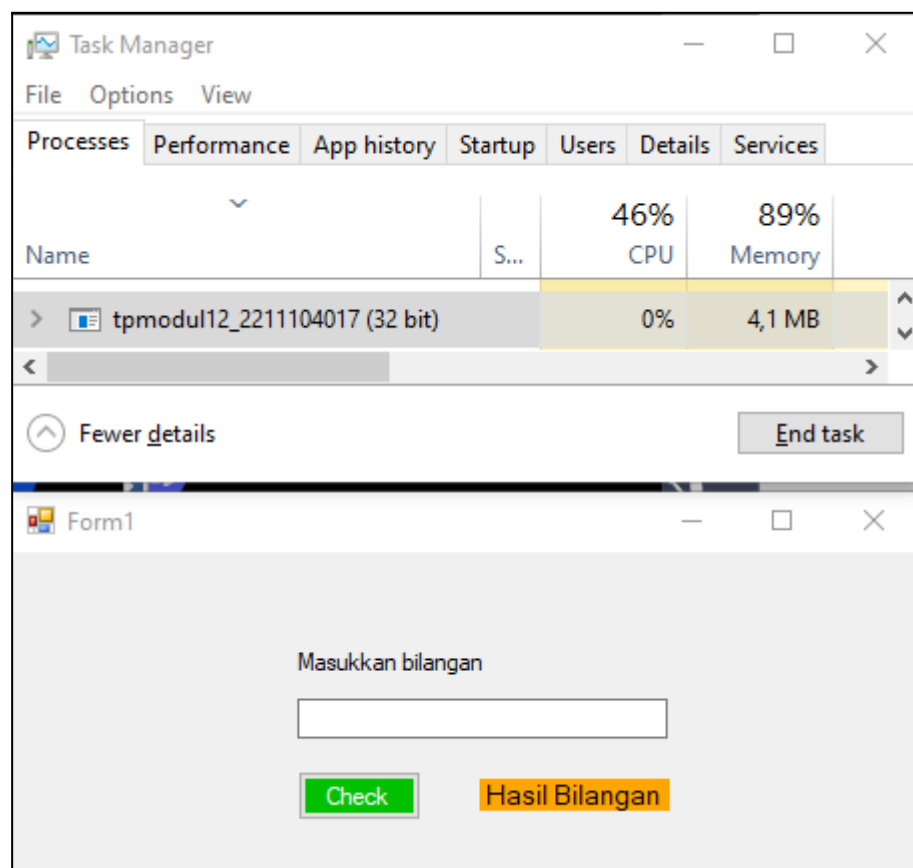
File Form1.cs adalah bagian dari aplikasi Windows Forms yang terdapat sebuah form bernama MainForm yang memiliki sebuah tombol (buttonCheck) dan sebuah kotak teks (textBoxInput). Ketika tombol diklik, aplikasi akan membaca teks dari kotak teks dan mencoba mengonversinya menjadi angka. Jika berhasil, aplikasi akan menggunakan metode dari BilanganHelper untuk menentukan apakah angka tersebut positif, negatif, atau nol, dan hasilnya ditampilkan pada label (labelOutput). Jika input tidak valid (bukan angka), label akan menampilkan pesan "Input tidak valid".

Pada file BilanganHelper.cs, terdapat kelas statis bernama BilanganHelper dengan metode CariTandaBilangan. Metode ini menerima satu parameter berupa angka (int a) dan mengembalikan string yang menggambarkan tanda bilangan tersebut. Jika angka kurang dari nol, akan mengembalikan "Negatif". Jika lebih dari nol, mengembalikan "Positif". Jika sama dengan nol, mengembalikan "Nol". File ini berfungsi sebagai pengelola logika sederhana untuk mengecek tanda bilangan yang diinput pengguna.

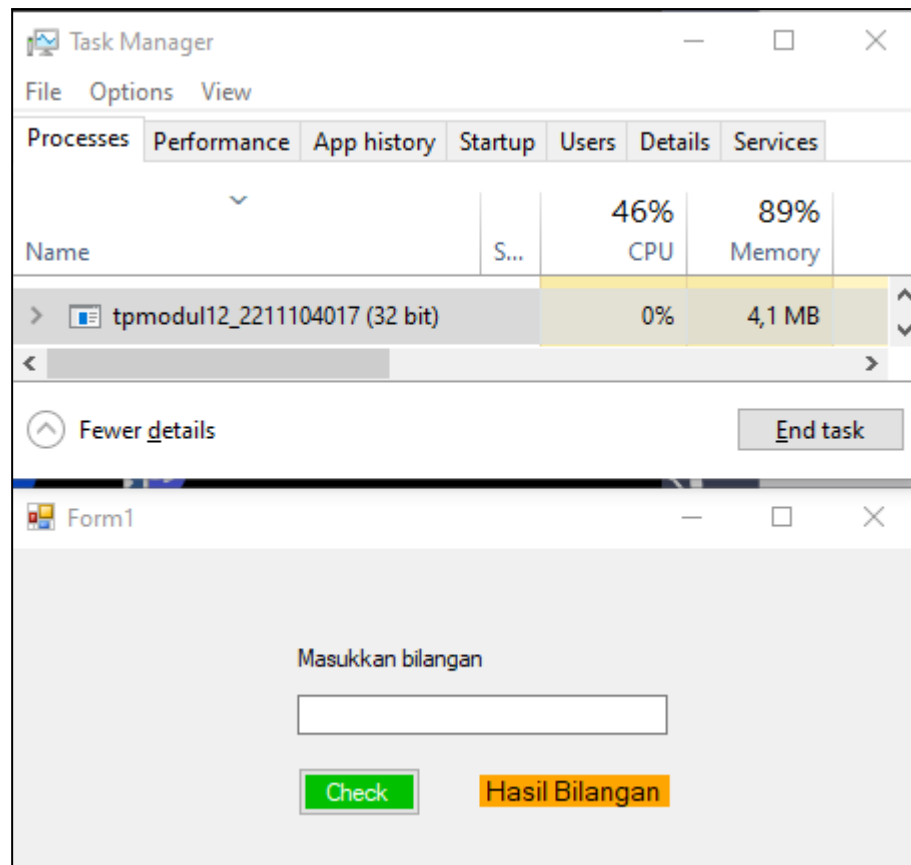
3. MELAKUKAN SOFTWARE PROFILING

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

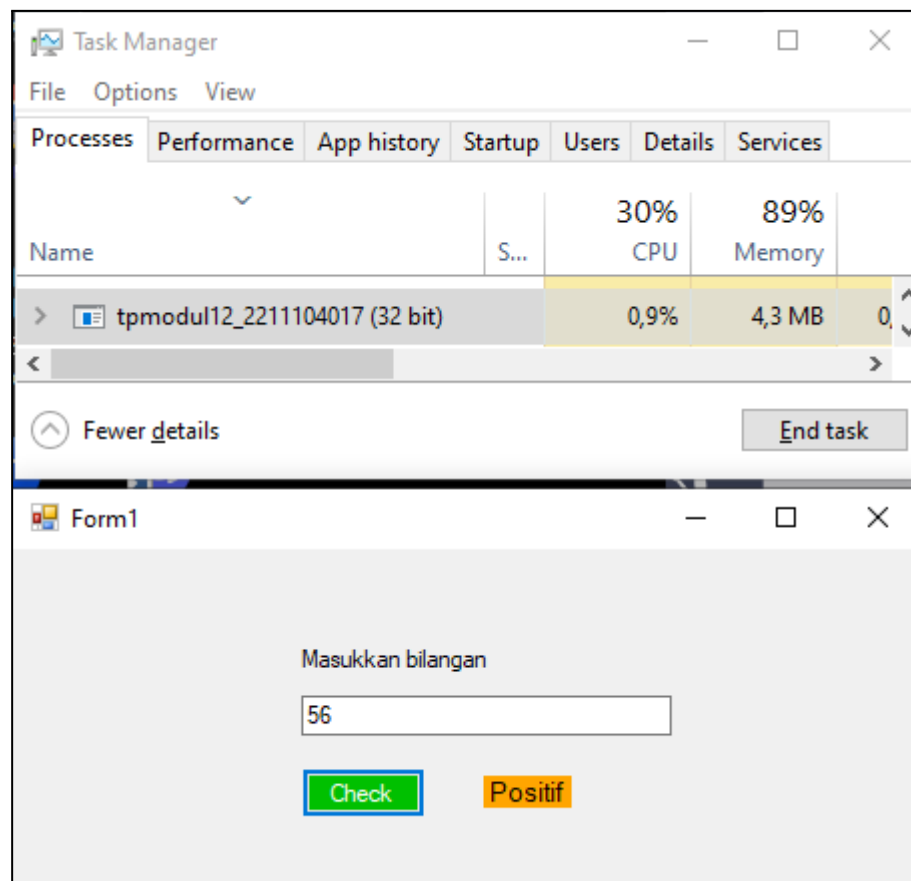
- A. Pada saat program berjalan, catat dan amati CPU usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.



- 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:

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: 4,1 MB

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

- CPU Usage: 0,9%
- Memory Usage: 4,3 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.

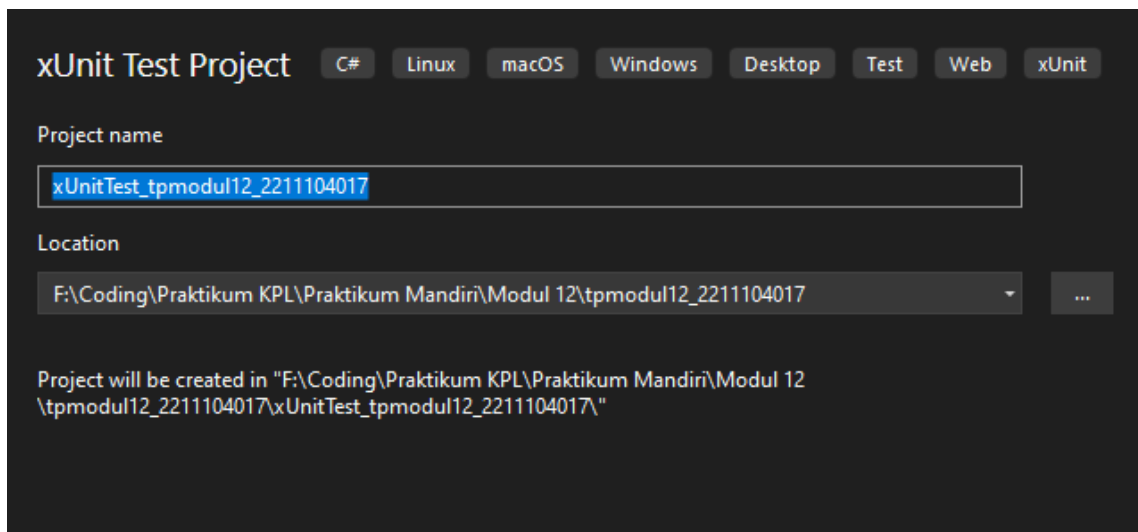
4. MENAMBAHKAN UNIT TESTING

Di dalam project yang sama:

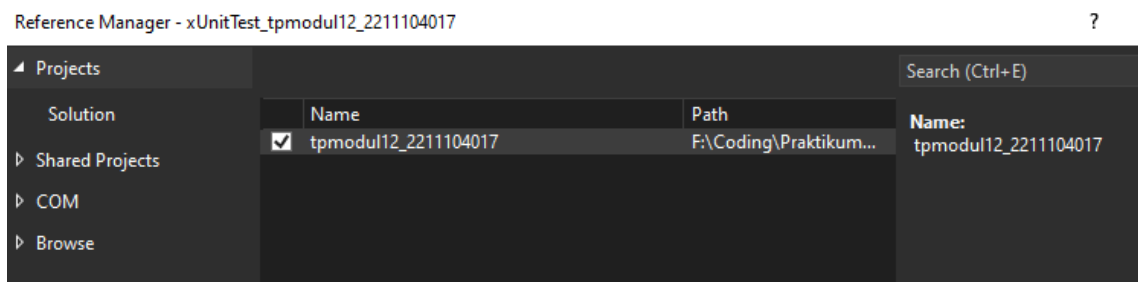
- A. Buatlah kode unit test untuk menguji method “CariTandaBilangan” yang dibuat sebelumnya.
- B. Pastikan kode unit test tersebut memiliki branch coverage yang baik untuk method “CariTandaBilangan”.
- C. Jalankan kode unit test yang dibuat dan lampirkan hasil unit testing yang dilakukan.

Jawab:

- Membuat Project untuk Testing



- Membuat referensi



- Source code file UnitTest1.cs

```

xUnitTest_tpmodul12_2211104017
1  using Xunit;
2  using tpmodul12_2211104017;
3
4  0 references
5  public class BilanganHelperTests
6  {
7      [Fact]
8      public void TestNegatif()
9      {
10         Assert.Equal("Negatif", BilanganHelper.CariTandaBilangan(-5));
11     }
12
13     [Fact]
14     public void TestPositif()
15     {
16         Assert.Equal("Positif", BilanganHelper.CariTandaBilangan(10));
17     }
18
19     [Fact]
20     public void TestNol()
21     {
22         Assert.Equal("Nol", BilanganHelper.CariTandaBilangan(0));
23     }
24 }

```

- Hasil testing:

Test Explorer

Test Explorer

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

Test	Durati...	Traits	Error Mess...
✔ xUnitTest_tpmodul12_2211104017 (3)	539 ms		
✔ <Empty Namespace> (3)	539 ms		
✔ BilanganHelperTests (3)	539 ms		
✔ TestNol	< 1 ms		
✔ TestPositif	< 1 ms		
✔ TestNegatif	539 ms		

Group Summary

xUnitTest_tpmodul12_2211104017

Tests in group: 3

Total Duration: 539 ms

Outcomes

✔ 3 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.