

**TUGAS PENDAHULUAN  
KONSTRUKSI PERANGKAT LUNAK**

**PERTEMUAN 12**

**PERFORMANCE ANALYSIS, UNIT TESTING, DAN DEBUG**



**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 PENDAHULUAN

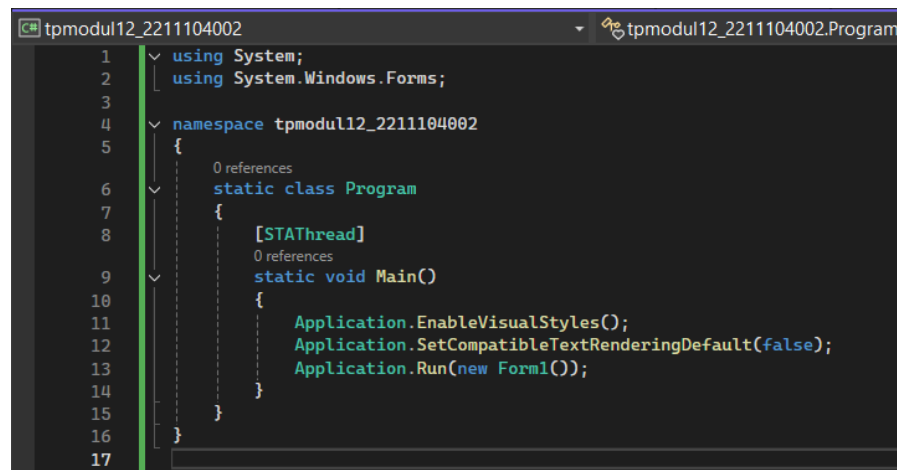
### 1. Screenshot hasil run

S

### 2. Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).

Source Code:

#### - Program.cs

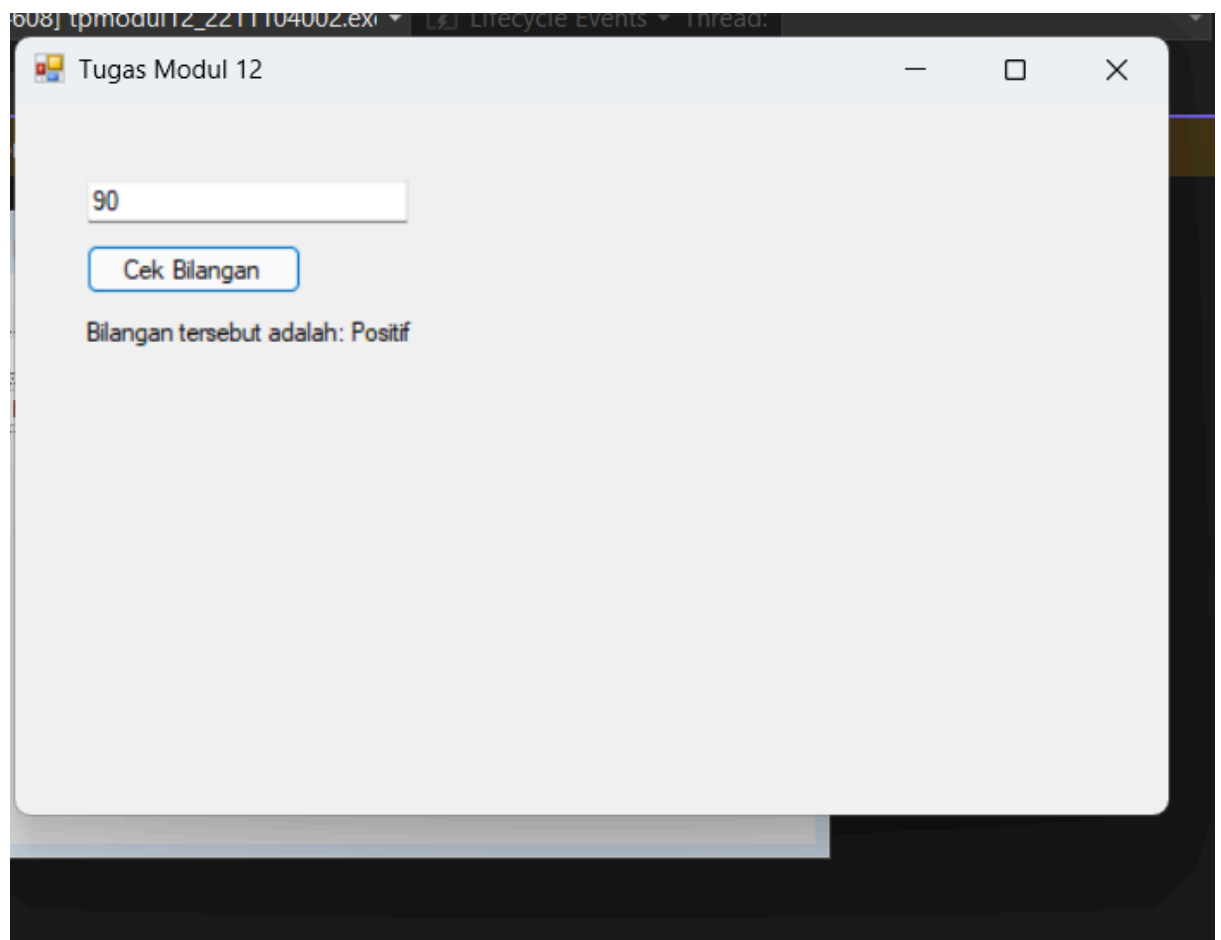
A screenshot of a Visual Studio code editor showing the source code for Program.cs. The file is named 'tpmodul12\_2211104002'. The code is in C# and includes the following: using System; using System.Windows.Forms; namespace tpmodul12\_2211104002 { static class Program { [STAThread] static void Main() { Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new Form1()); } } }

```
1  using System;
2  using System.Windows.Forms;
3
4  namespace tpmodul12_2211104002
5  {
6      0 references
7      static class Program
8      {
9          [STAThread]
10         0 references
11         static void Main()
12         {
13             Application.EnableVisualStyles();
14             Application.SetCompatibleTextRenderingDefault(false);
15             Application.Run(new Form1());
16         }
17     }
```

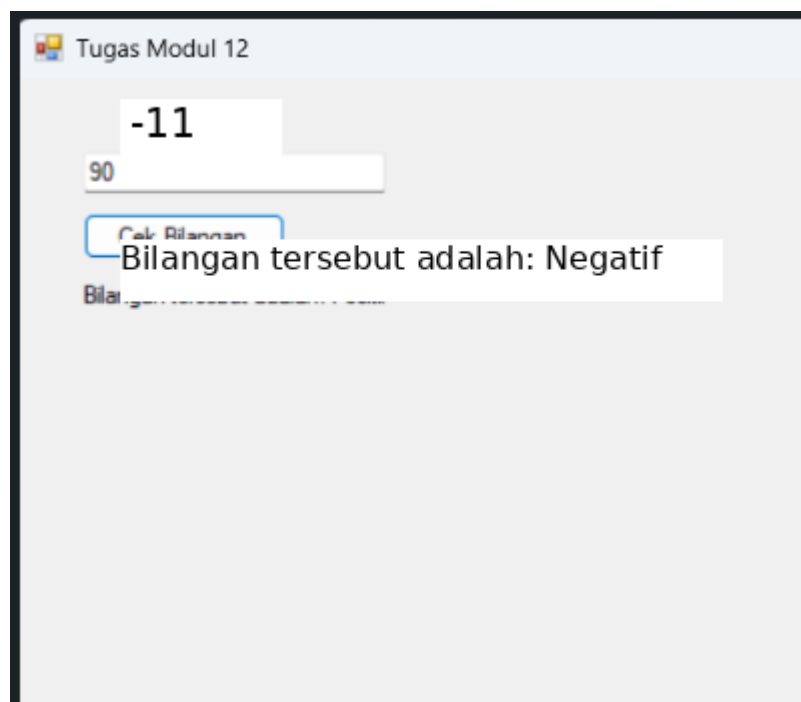
#### - Form.cs

```
tpmodul12_2211104002 tpmodul12_2211104002.Form1
1 using System;
2 using System.Windows.Forms;
3
4 namespace tpmodul12_2211104002
5 {
6     3 references
7     public partial class Form1 : Form
8     {
9         1 reference
10        public Form1()
11        {
12            InitializeComponent();
13        }
14
15        0 references
16        private void btnCek_Click(object sender, EventArgs e)
17        {
18            if (int.TryParse(txtInput.Text, out int nilai))
19            {
20                string hasil = CariTandaBilangan(nilai);
21                lblHasil.Text = $"Bilangan tersebut adalah: {hasil}";
22            }
23            else
24            {
25                lblHasil.Text = "Input tidak valid!";
26            }
27        }
28
29        4 references
30        public string CariTandaBilangan(int a)
31        {
32            if (a < 0)
33                return "Negatif";
34            else if (a > 0)
35                return "Positif";
36            else
37                return "Nol";
38        }
39
40        1 reference
41        private void Form1_Load(object sender, EventArgs e)
42        {
43        }
44    }
45 }
```

Output jika angka positif



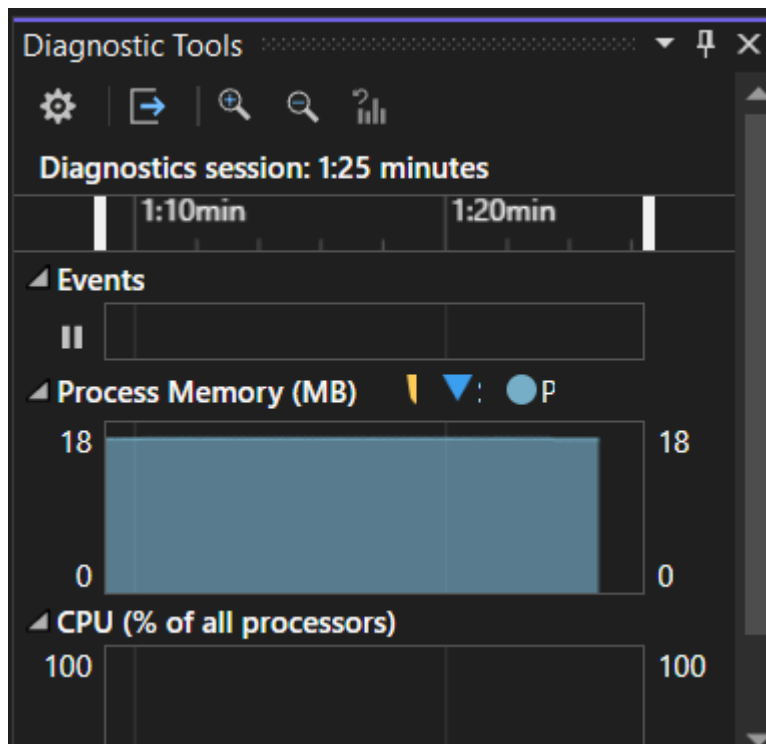
**Output jika angka negatif**



## **Penjelasan:**

Program ini merupakan aplikasi desktop sederhana berbasis Windows Forms (WinForms) yang dibuat menggunakan bahasa C# untuk menyelesaikan tugas praktikum Modul 12. Tujuan utama dari aplikasi ini adalah untuk menentukan apakah suatu bilangan yang dimasukkan oleh pengguna bersifat positif, negatif, atau nol, serta menampilkannya pada antarmuka grafis pengguna (GUI). Selain itu, aplikasi ini juga dilengkapi dengan pengujian otomatis menggunakan unit testing untuk memastikan bahwa logika program bekerja sesuai dengan yang diharapkan. File `Form1.cs` merupakan inti dari antarmuka pengguna. Di dalamnya terdapat sebuah method bernama `CariTandaBilangan(int a)` yang akan mengembalikan string berdasarkan nilai input. Jika nilai kurang dari nol, maka akan mengembalikan "Negatif", jika lebih dari nol akan mengembalikan "Positif", dan jika sama dengan nol akan mengembalikan "Nol". Method ini dipanggil saat pengguna mengklik tombol "Cek Bilangan" yang ada pada form. Nilai input dibaca dari sebuah `TextBox`, lalu hasilnya ditampilkan pada `Label` setelah dilakukan parsing dan pemeriksaan menggunakan `int.TryParse`. File `Form1.Designer.cs` adalah file yang dihasilkan secara otomatis oleh Visual Studio dan berfungsi untuk mendeskripsikan tampilan form. Di dalamnya terdapat deklarasi dan pengaturan elemen-elemen GUI seperti `TextBox`, `Button`, dan `Label`, termasuk letak dan ukurannya di dalam jendela form. Tombol `btnCek` diberi event handler agar saat diklik, fungsi `btnCek_Click` dijalankan. File `Program.cs` berperan sebagai titik masuk utama (entry point) dari aplikasi. Di dalamnya, method `Main()` digunakan untuk menginisialisasi dan menjalankan aplikasi dengan memanggil form utama (`Form1`) menggunakan `Application.Run(new Form1());`. Selain bagian antarmuka, program ini juga dilengkapi dengan unit test yang ditulis dalam file `UnitTest1.cs` pada project test `tpmodul12_2211104002.Tests`. Di dalam file ini, terdapat tiga method pengujian (`Test_Positive`, `Test_Negative`, dan `Test_Zero`) yang masing-masing menguji hasil keluaran dari method `CariTandaBilangan` berdasarkan nilai input tertentu. Misalnya, jika diberikan input -5, maka hasil yang diharapkan adalah "Negatif". Pengujian dilakukan dengan menggunakan framework `MSTest` dan metode `Assert.AreEqual` untuk membandingkan hasil aktual dengan hasil yang diharapkan.

## **Software Profiling**



**Unit Testing**

**Source Code**

```
tpmodul12_NIM.Tests
tpmodul12_2211104002.Tests.UnitTest1

1 using Microsoft.VisualStudio.TestTools.UnitTesting;
2 using tpmodul12_2211104002; // Ganti sesuai nama namespace project kamu
3
4 namespace tpmodul12_2211104002.Tests
5 {
6     [TestClass]
7     public class UnitTest1
8     {
9         [TestMethod]
10        public void Test_Negatif()
11        {
12            var form = new Form1();
13            string hasil = form.CariTandaBilangan(-5);
14            Assert.AreEqual("Negatif", hasil);
15        }
16
17        [TestMethod]
18        public void Test_Nol()
19        {
20            var form = new Form1();
21            string hasil = form.CariTandaBilangan(0);
22            Assert.AreEqual("Nol", hasil);
23        }
24
25        [TestMethod]
26        public void Test_Positif()
27        {
28            var form = new Form1();
29            string hasil = form.CariTandaBilangan(12);
30            Assert.AreEqual("Positif", hasil);
31        }
32    }
33 }
34
```

## Output

tpmodul12_2211104002.Tests (3)	444 ms
TestNegatif	< 1 ms
TestNol	< 1 ms
TestPositif	444 ms

## Penjelasan

Unit test pada program ini berfungsi untuk menguji kebenaran dari method `CariTandaBilangan(int a)` yang terdapat di dalam form utama aplikasi. Metode ini dirancang untuk mengembalikan nilai string berdasarkan tanda suatu bilangan: mengembalikan "Negatif" jika nilai kurang dari nol, "Positif" jika nilai lebih dari nol, dan "Nol" jika nilai sama dengan nol. Pengujian dilakukan menggunakan framework MSTest, yang merupakan framework resmi dari Microsoft untuk keperluan unit testing di lingkungan .NET Framework. File unit test ditempatkan dalam proyek terpisah dengan nama `tpmodul12_2211104002.Tests`, agar terpisah dari kode utama aplikasi. Di dalam file `UnitTest1.cs`, dibuat satu class bernama `UnitTest1` yang ditandai dengan atribut `[TestClass]`, dan berisi tiga method pengujian, masing-masing

diberi atribut [TestMethod]. Setiap method tersebut menguji satu kondisi logika dari method CariTandaBilangan. Misalnya, Test\_Positive() akan menguji apakah input 10 menghasilkan string "Positif", sedangkan Test\_Negative() akan memastikan bahwa input -5 menghasilkan "Negatif", dan Test\_Zero() memastikan bahwa input 0 menghasilkan "Nol". Dalam setiap pengujian, objek Form1 dibuat secara instan, lalu method CariTandaBilangan dipanggil dengan parameter angka tertentu. Hasilnya kemudian dibandingkan dengan hasil yang diharapkan menggunakan Assert.AreEqual(). Jika hasil aktual sesuai dengan yang diharapkan, maka pengujian dinyatakan berhasil (Passed). Dengan adanya unit test ini, pengembang dapat memastikan bahwa logika penentuan tanda bilangan berfungsi dengan benar, serta mempermudah pendeteksian kesalahan (error) saat kode dimodifikasi di masa mendatang. Unit test ini juga mencerminkan praktik pengembangan perangkat lunak yang andal dan dapat diuji (testable).