

Nama : abdul roni_ SE063 _2211104080

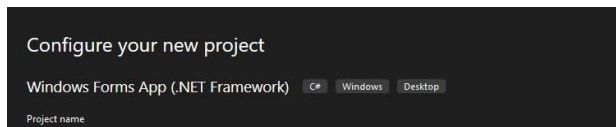
1. MEMBUAT PROJECT GUI BARU

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).

Bukti :

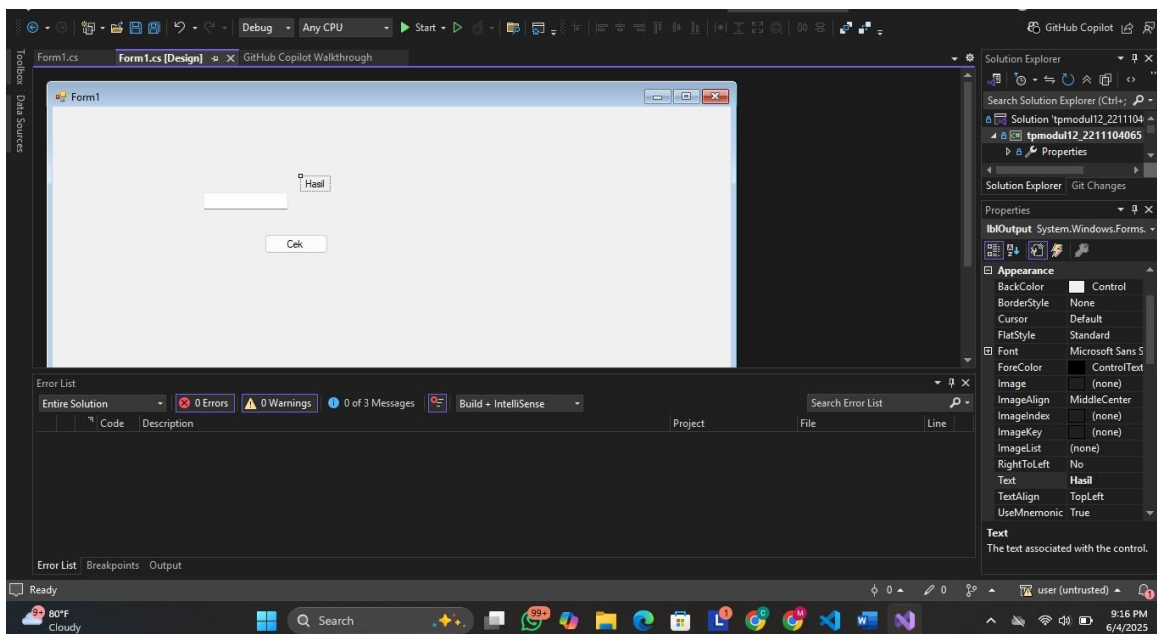


2. MEMBUAT GUI SEDERHANA DAN

Pada project yang telah dibuat sebelumnya:

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

Bukti



B. Tambahkan satu method dengan nama "CariTandaBilangan(int a)" yang menerima satu input dan mengembalikan nilai string dengan aturan sebagai berikut:

i. Apabila input a kurang dari 0 maka output adalah

"Negatif" ii. Apabila input a lebih dari 0 maka output adalah

"Positif" iii. Apabila input sama dengan 0 maka output

adalah "Nol" **Form1.cs**

```
using System; using
System.Collections.Generic; using
System.ComponentModel; using
System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace tpmodul12_2211104080
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // Method untuk mencari tanda bilangan
        public string CariTandaBilangan(int a)
        {
            if (a < 0)
                return "Negatif";
            else if (a > 0)
                return "Positif";
            else
```

```

        return "Nol";
    }

    // Event handler untuk button click      private void
button1_Click(object sender, EventArgs e)
    {
        // Mendapatkan input dari TextBox dan mengonversi menjadi integer
int input = Convert.ToInt32(txtInput.Text);

        // Memanggil method CariTandaBilangan dan mendapatkan hasil
string result = CariTandaBilangan(input);

        // Menampilkan hasil pada Label
lblOutput.Text = result;
    }

    private void txtInput_TextChanged(object sender, EventArgs e)
    {

    }

    private void lblOutput_Click(object sender, EventArgs e)
    {

    }
}

```

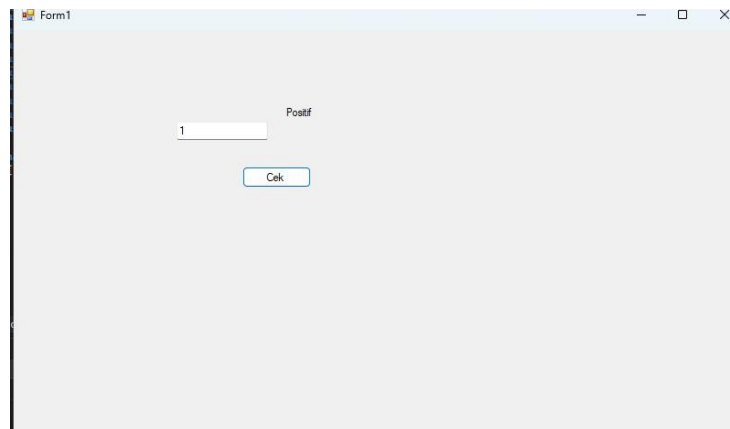
```
}
```

Penjelasan singkat source code:

Kode di atas adalah implementasi dari aplikasi Windows Forms yang memiliki satu form dengan sebuah TextBox, Button, dan Label. Aplikasi ini bertujuan untuk memeriksa tanda bilangan yang dimasukkan pengguna. Ketika tombol **Button** ditekan, aplikasi akan mengambil nilai dari **TextBox** (dikonversi menjadi integer), kemudian memanggil method `CariTandaBilangan` untuk menentukan apakah angka tersebut "Negatif", "Positif", atau "Nol". Hasilnya kemudian ditampilkan di **Label**. Method `CariTandaBilangan` mengecek kondisi angka yang diberikan dan mengembalikan hasil berupa string yang sesuai dengan kondisi angka tersebut. Event handler lainnya (untuk `TextChanged` dan `lblOutput_Click`) tidak memiliki implementasi aksi spesifik dalam kode ini.

C. Pada tampilan GUI, pada saat tombol ditekan, maka label output akan menampilkan hasil pangkat dari pemanggilan fungsi “`CariTandaBilangan`” dari input textbox yang diberikan.

Positif



Negatif

Form1

Negatif

-7

Cek

No!

Form1

Nol

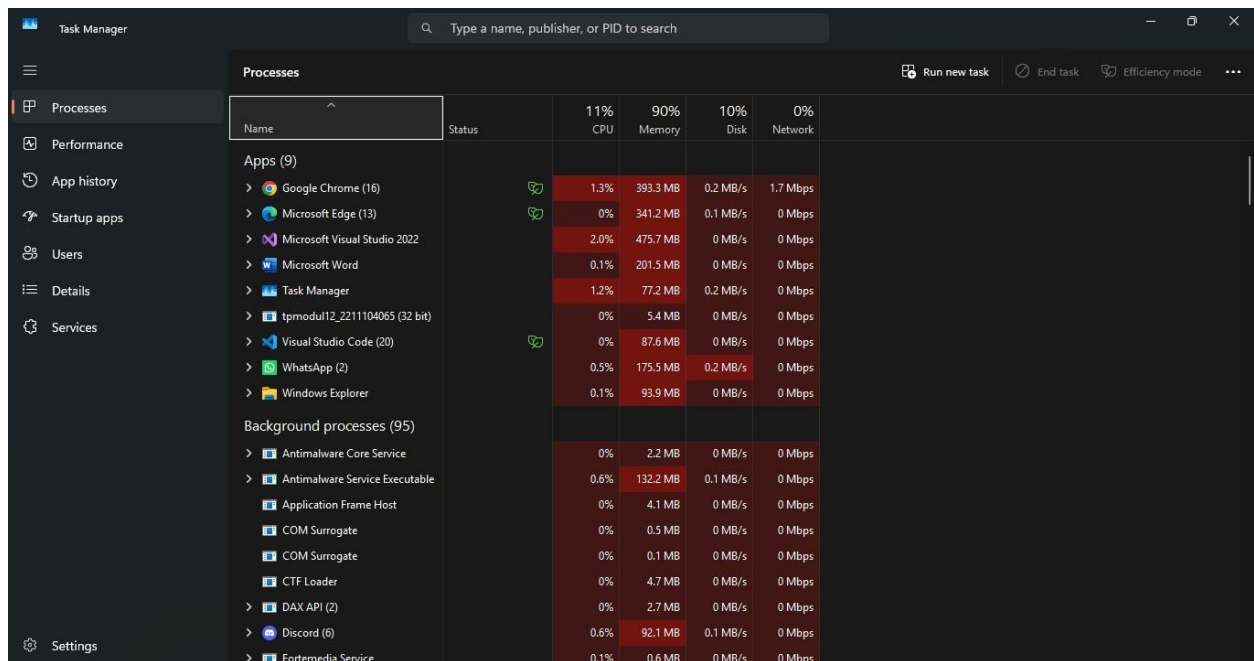
0

Cek

3. MELAKUKAN SOFTWARE PROFILING

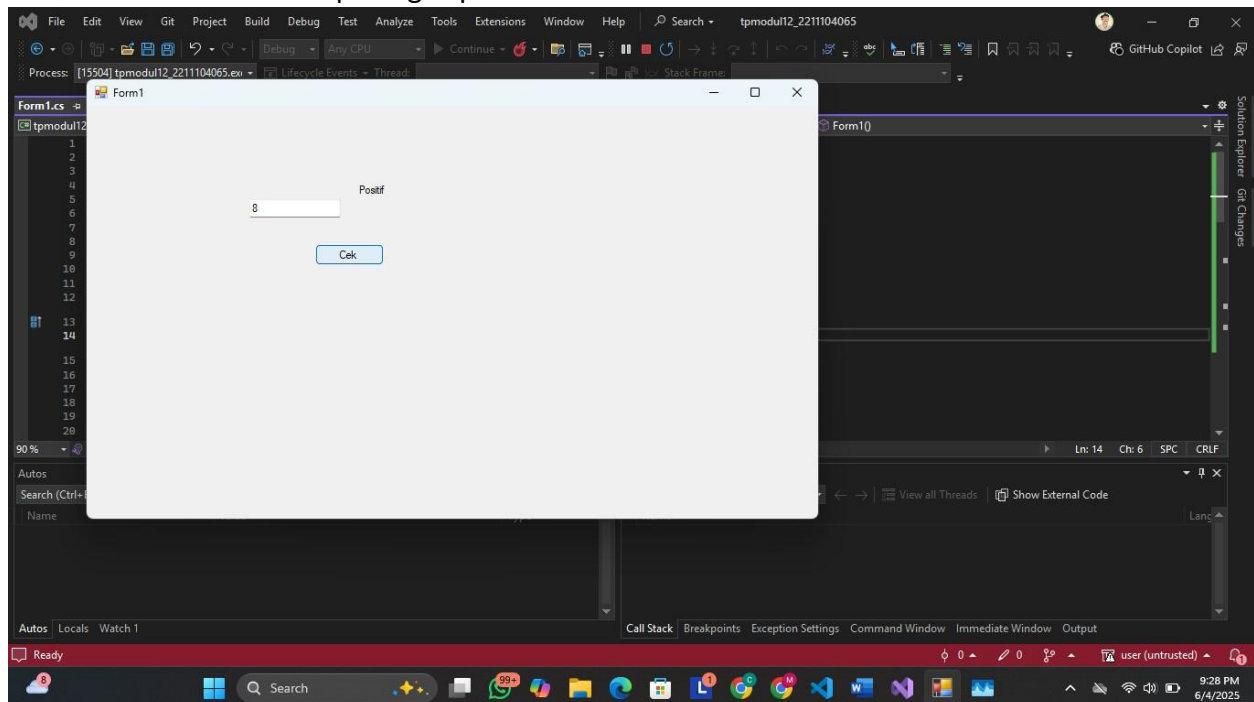
Jalankan project yang dibuat sebelumnya dan jalankan profiling tools (misal dari visual studio, task manager atau sejenisnya):

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

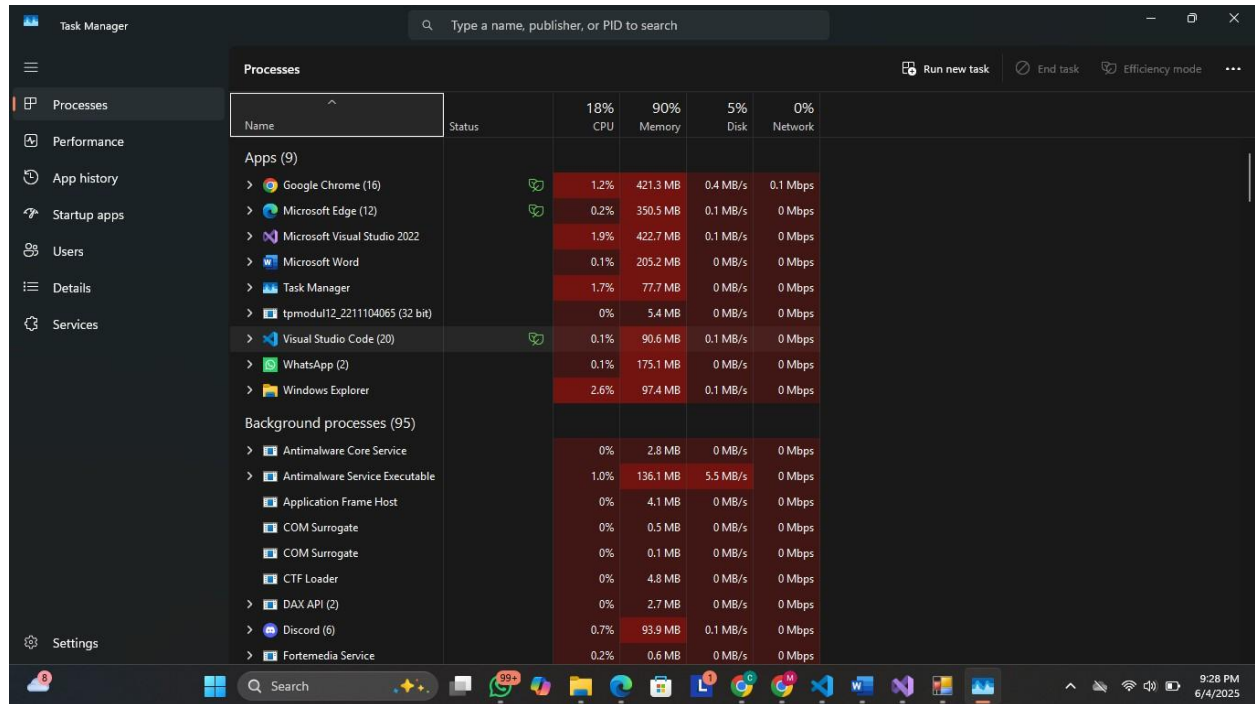


Name	Status	11% CPU	90% Memory	10% Disk	0% Network
Apps (9)					
Google Chrome (16)		1.3%	393.3 MB	0.2 MB/s	1.7 Mbps
Microsoft Edge (13)		0%	341.2 MB	0.1 MB/s	0 Mbps
Microsoft Visual Studio 2022		2.0%	475.7 MB	0 MB/s	0 Mbps
Microsoft Word		0.1%	201.5 MB	0 MB/s	0 Mbps
Task Manager		1.2%	77.2 MB	0.2 MB/s	0 Mbps
tpmodul12_2211104065 (32 bit)		0%	5.4 MB	0 MB/s	0 Mbps
Visual Studio Code (20)		0%	87.6 MB	0 MB/s	0 Mbps
WhatsApp (2)		0.5%	175.5 MB	0.2 MB/s	0 Mbps
Windows Explorer		0.1%	93.9 MB	0 MB/s	0 Mbps
Background processes (95)					
Antimalware Core Service		0%	2.2 MB	0 MB/s	0 Mbps
Antimalware Service Executable		0.6%	132.2 MB	0.1 MB/s	0 Mbps
Application Frame Host		0%	4.1 MB	0 MB/s	0 Mbps
COM Surrogate		0%	0.5 MB	0 MB/s	0 Mbps
COM Surrogate		0%	0.1 MB	0 MB/s	0 Mbps
CTF Loader		0%	4.7 MB	0 MB/s	0 Mbps
DAX API (2)		0%	2.7 MB	0 MB/s	0 Mbps
Discord (6)		0.6%	92.1 MB	0.1 MB/s	0 Mbps
Fortimedia Service		0.1%	0.6 MB	0 MB/s	0 Mbps

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



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



The screenshot shows the Windows Task Manager interface. The 'Processes' tab is selected, displaying a list of running applications and background processes. The columns show the process name, status, CPU usage, memory usage, disk usage, and network usage. The total CPU usage is 18%, and total memory usage is 90%.

Name	Status	18% CPU	90% Memory	5% Disk	0% Network
Apps (9)					
Google Chrome (16)	Running	1.2%	421.3 MB	0.4 MB/s	0.1 Mbps
Microsoft Edge (12)	Running	0.2%	350.5 MB	0.1 MB/s	0 Mbps
Microsoft Visual Studio 2022	Running	1.9%	422.7 MB	0.1 MB/s	0 Mbps
Microsoft Word	Running	0.1%	205.2 MB	0 MB/s	0 Mbps
Task Manager	Running	1.7%	77.7 MB	0 MB/s	0 Mbps
tpmodul12_2211104065 (32 bit)	Running	0%	5.4 MB	0 MB/s	0 Mbps
Visual Studio Code (20)	Running	0.1%	90.6 MB	0.1 MB/s	0 Mbps
WhatsApp (2)	Running	0.1%	175.1 MB	0 MB/s	0 Mbps
Windows Explorer	Running	2.6%	97.4 MB	0.1 MB/s	0 Mbps
Background processes (95)					
Antimalware Core Service	Running	0%	2.8 MB	0 MB/s	0 Mbps
Antimalware Service Executable	Running	1.0%	136.1 MB	5.5 MB/s	0 Mbps
Application Frame Host	Running	0%	4.1 MB	0 MB/s	0 Mbps
COM Surrogate	Running	0%	0.5 MB	0 MB/s	0 Mbps
COM Surrogate	Running	0%	0.1 MB	0 MB/s	0 Mbps
CTF Loader	Running	0%	4.8 MB	0 MB/s	0 Mbps
DAX API (2)	Running	0%	2.7 MB	0 MB/s	0 Mbps
Discord (6)	Running	0.7%	93.9 MB	0.1 MB/s	0 Mbps
Fortemedia Service	Running	0.2%	0.6 MB	0 MB/s	0 Mbps

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”.

UnitTest1.cs


```

using Microsoft.VisualStudio.TestTools.UnitTesting; using
tpmodul12_2211104080;

namespace tpmodul12_2211104080.Tests
{
    [TestClass]
    public class Form1Tests
    {
        // Unit Test untuk memeriksa apakah nilai negatif menghasilkan output
        "Negatif"
        [TestMethod]
        public void TestCariTandaBilangan_Negatif()
        {
            // Arrange: Membuat objek Form1
            Form1 form = new Form1();

            // Act: Memanggil method CariTandaBilangan dengan input -5
            string result = form.CariTandaBilangan(-5);

            // Assert: Memastikan hasil yang dikembalikan adalah "Negatif"
            Assert.AreEqual("Negatif", result);
        }

        // Unit Test untuk memeriksa apakah nilai positif menghasilkan output
        "Positif"
        [TestMethod]
        public void TestCariTandaBilangan_Positif()
        {
            // Arrange: Membuat objek Form1
            Form1 form = new Form1();

            // Act: Memanggil method CariTandaBilangan dengan input 5
            string result = form.CariTandaBilangan(5);

```

```

            // Assert: Memastikan hasil yang dikembalikan adalah "Positif"
            Assert.AreEqual("Positif", result);
        }

```

```

        // Unit Test untuk memeriksa apakah nilai 0 menghasilkan output "Nol"
        [TestMethod]
        public void TestCariTandaBilangan_Nol()
        {
            // Arrange: Membuat objek Form1
            Form1 form = new Form1();

```

```

        // Act: Memanggil method CariTandaBilangan dengan input 0
        string result = form.CariTandaBilangan(0);

        // Assert: Memastikan hasil yang dikembalikan adalah "Nol"
        Assert.AreEqual("Nol", result);
    }
}
}

```

Penjelasan Kode :

Kode di atas adalah unit test yang menguji method CariTandaBilangan yang ada di dalam kelas Form1 pada project Windows Forms. Tiga metode pengujian disediakan:

1. **TestCariTandaBilangan_Negatif**: Menguji apakah method CariTandaBilangan mengembalikan string "Negatif" ketika diberikan input angka negatif (-5).
2. **TestCariTandaBilangan_Positif**: Menguji apakah method CariTandaBilangan mengembalikan string "Positif" ketika diberikan input angka positif (5).
3. **TestCariTandaBilangan_Nol**: Menguji apakah method CariTandaBilangan mengembalikan string "Nol" ketika diberikan input angka 0.

Setiap test dilakukan dengan mengikuti pola **Arrange-Act-Assert**, yaitu menyiapkan objek yang diperlukan (Arrange), menjalankan method yang diuji (Act), dan memverifikasi hasil yang diperoleh dengan nilai yang diharapkan (Assert). Jika hasilnya sesuai dengan yang diharapkan, maka unit test tersebut berhasil.

C. Jalankan kode unit test yang dibuat dan lampirkan hasil unit testing yang dilakukan.

