

## **TUGAS JURNAL MODUL 12**



**Disusun Oleh :**

**Pradana Argo Pangestu**

**2311104079**

**Kelas :SE-07-02**

**Dosen :**

**Yudha Islami Sulistya**

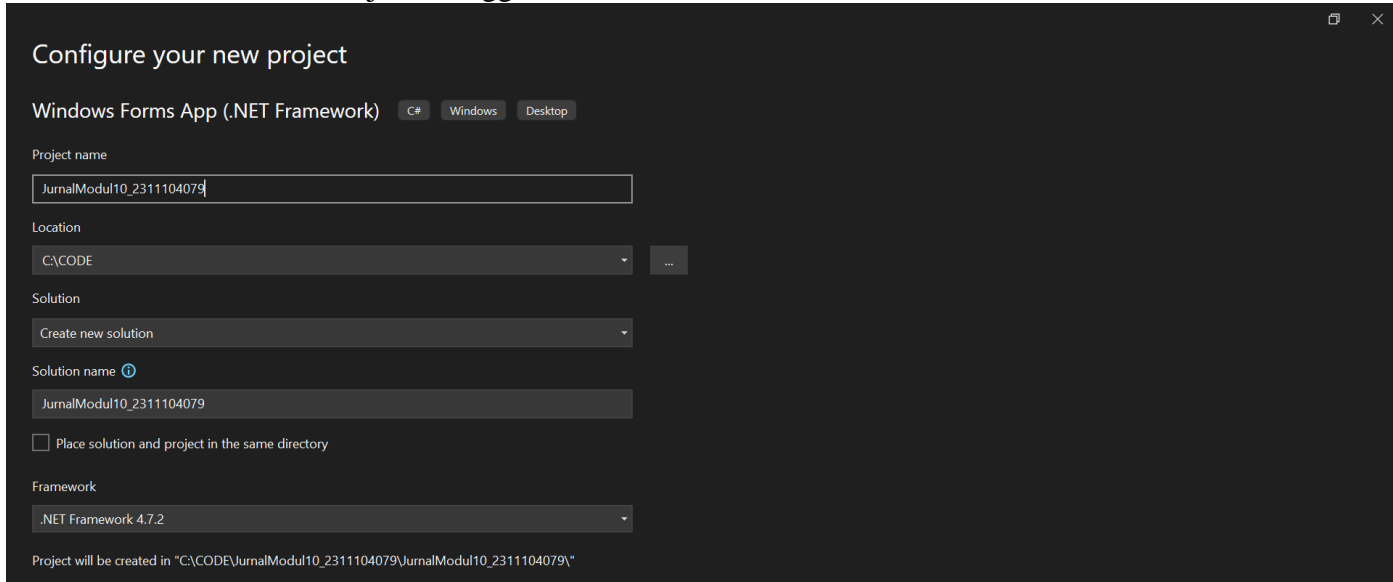
**PROGRAM STUDI SOFTWARE ENGINEERING  
DIREKTORAT KAMPUS PURWOKERTO  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## I. Link Github

[https://github.com/Pradana123/KPL\\_PradanaAP\\_2311104079/tree/main/12\\_Performance Analysis Unit Testing dan Debbuging](https://github.com/Pradana123/KPL_PradanaAP_2311104079/tree/main/12_Performance_Analysis_Unit_Testing_dan_Debbuging)

### I. Ss Output :

#### 1. Membuat New Project Menggunakan Blank Solution



Configure your new project

Windows Forms App (.NET Framework) C# Windows Desktop

Project name  
JurnalModul10\_2311104079

Location  
C:\CODE

Solution  
Create new solution

Solution name ⓘ  
JurnalModul10\_2311104079

☐ Place solution and project in the same directory

Framework  
.NET Framework 4.7.2

Project will be created in "C:\CODE\JurnalModul10\_2311104079\JurnalModul10\_2311104079\"

#### 2. Menambahkan textbox sebanyak 2 di “Form.cs [Design]”



Form1

menampilkan hasil output

Hitung

#### 4. Menambahkan syntax di class Form1.cs

```
1. using System;
2. using System.Collections.Generic; using System.ComponentModel; using
   System.Data;
3. using System.Drawing; using System.Linq; using System.Text;
4. using System.Threading.Tasks; using System.Windows.Forms;
5. namespace JurnalModul12_2311104079
6. {
7.     public partial class Form1 : Form
8.     {
9.         public Form1()
10.        {
11.            InitializeComponent();
12.        }
13.        private void lbloutput_Click(object sender, EventArgs e)
14.        {
15.        }
16.
17.        private void Form1_Load(object sender, EventArgs e)
18.        {
19.
20.        }
21.
22.        private void txtExponent_TextChanged(object sender, EventArgs e)
23.        {
24.
25.        }
26.
27.        private void txtBase_TextChanged(object sender, EventArgs e)
28.        {
29.
30.        }
31.
32.        public int CariNilaiPangkat(int a, int b)
33.        {
34.            if (b == 0) return 1;
35.            if (b < 0) return -1;
36.            if (b > 10 || a > 100) return -2;
37.
38.            try
39.            {
40.                checked
41.                {
42.                    int hasil = 1;
43.                    for (int i = 0; i < b; i++)
44.                    {
45.                        hasil *= a;
46.                    }
47.                    return hasil;
48.                }
49.            }
50.            catch (OverflowException)
```

```
51.     {
52.     return -3;
53.     }
54.     }
55.     private void btnHitung_Click(object sender, EventArgs e)
56.     {
57.         int a, b;
58.         bool isValid = int.TryParse(txtBase.Text, out a); bool isValid =
           int.TryParse(txtExponent.Text, out b);
59.         if (!isValid || !isValid)
60.         {
61.             lblOutput.Text = "Input tidak valid!"; return;
62.         }
63.         int hasil = CariNilaiPangkat(a, b); lblOutput.Text = $"Hasil: {hasil}";
```

#### Penjelasan Singkat:

Program ini merupakan **aplikasi GUI berbasis Windows Forms** yang dirancang untuk menghitung **pangkat dari dua bilangan bulat: basis (a) dan eksponen (b)**, dengan aturan khusus yang ditetapkan.

#### Komponen Antarmuka:

- txtBase: TextBox untuk memasukkan nilai basis (**a**).
- txtExponent: TextBox untuk memasukkan nilai eksponen (**b**).
- btnHitung: Tombol untuk memulai proses perhitungan.
- lblOutput: Label untuk menampilkan hasil akhir atau pesan kesalahan.

#### Alur Eksekusi:

1. Pengguna mengisi nilai pada txtBase dan txtExponent.
2. Ketika tombol "**Hitung**" ditekan:
  - Program membaca dan memvalidasi bahwa kedua input adalah bilangan bulat.
  - Kemudian memanggil method CariNilaiPangkat(int a, int b) untuk menghitung nilai  $a^b$ .
  - Hasil atau pesan kesalahan ditampilkan di lblOutput.

#### Logika Method CariNilaiPangkat:

Method ini bersifat **public** dan menerima dua parameter int a (basis) dan int b (eksponen). Hasil dikembalikan sebagai int, dengan aturan khusus:

- Jika  $b == 0$ , maka hasilnya **1** (aturan matematika).
- Jika  $b < 0$ , maka mengembalikan **-1** (kode kesalahan: eksponen negatif tidak didukung).
- Jika  $b > 10$  atau  $a > 100$ , maka mengembalikan **-2** (kode kesalahan: melebihi batas input).
- Jika perhitungan menghasilkan nilai yang melebihi batas int, maka terjadi **overflow**, ditangani dengan blok checked. Jika terjadi OverflowException, maka dikembalikan nilai **-3**.

**Tujuan Desain:**

Dengan pendekatan ini, program dapat memberikan **umpan balik yang jelas** kepada pengguna:

- Hasil perhitungan yang benar.
- Kode kesalahan jika input tidak sesuai aturan.
- Deteksi overflow untuk mencegah crash atau hasil salah.

## 5. Menambah Project Baru menggunakan Unit Test Project

—

□

×

### Configure your new project

Unit Test Project (.NET Framework) C# Windows Test

Project name

Location

...

Solution name ⓘ

☐ Place solution and project in the same directory

Framework

Project will be created in "C:\CODE\UnitTestProjectJurnal12\UnitTestProjectJurnal12\"

## 6. Menambah syntax Class PangkatTest.cs

```
1. using System;
2. using Microsoft.VisualStudio.TestTools.UnitTesting; using JurnalModul12_2311104079;
3. [TestClass]
4. public class PangkatTests
5. {
6.     Form1 form = new Form1();
7.     [TestMethod]
8.     public void TestPangkatB0()
9.     {
10.         Assert.AreEqual(1, form.CariNilaiPangkat(0, 0));
11.     }
12.     [TestMethod]
13.     public void TestPangkatNegatif()
14.     {
15.         Assert.AreEqual(-1, form.CariNilaiPangkat(5, -2));
16.     }
17.     [TestMethod]
18.     public void TestBesarDari10()
19.     {
20.         Assert.AreEqual(-2, form.CariNilaiPangkat(5, 11));
21.     }
22.     [TestMethod]
23.     public void TestLebih100()
24.     {
25.         Assert.AreEqual(-2, form.CariNilaiPangkat(101, 3));
26.     }
27.     [TestMethod]
28.     public void TestOverflow()
29.     {
30.         Assert.AreEqual(-3, form.CariNilaiPangkat(9, 30));
31.     }
32.     [TestMethod]
33.     public void TestNormal()
34.     {
```

Penjelasan Singkat:

File **PangkatTests.cs** berisi **kode unit test otomatis** yang ditulis menggunakan **framework MSTest**. Tujuannya adalah untuk memastikan bahwa method **CariNilaiPangkat** dalam kelas **Form1** berjalan **sesuai dengan aturan dan spesifikasi** yang telah ditentukan.

### Isi dan Tujuan Pengujian:

Kelas ini terdiri dari **enam metode pengujian**, yang masing-masing menguji satu kondisi logika tertentu:

#### 1. **TestPangkatB0**

- Menguji jika eksponen **bernilai 0**.
- Hasil yang diharapkan: **1** (karena  $a^0 = 1$  dalam matematika).

#### 2. **TestPangkatNegatif**

- Menguji jika eksponen **bernilai negatif**.
- Hasil yang diharapkan: **-1** (kode kesalahan karena eksponen negatif tidak didukung).

#### 3. **TestBesarDari10**

- Menguji jika eksponen **lebih dari 10**.
- Hasil yang diharapkan: **-2** (melebihi batas maksimum eksponen).

#### 4. **TestLebih100**

- Menguji jika basis **lebih dari 100**.
- Hasil yang diharapkan: **-2** (melebihi batas maksimum nilai basis).

#### 5. **TestOverflow**

- Menguji perhitungan yang menghasilkan angka sangat besar hingga terjadi **overflow**.



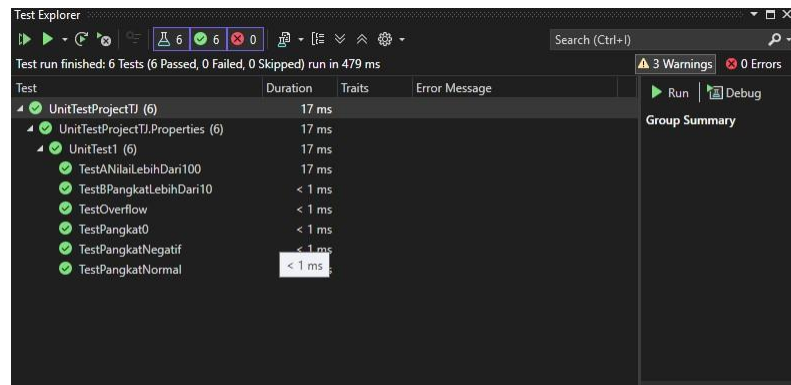
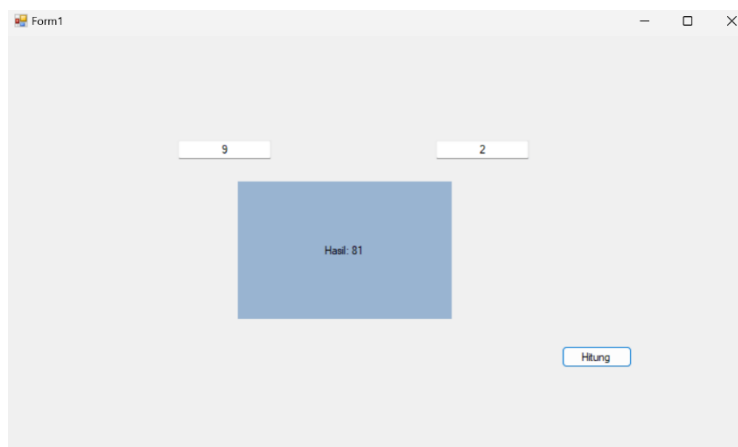
- Hasil yang diharapkan: **-3** (menandakan terjadi overflow saat menghitung pangkat).

## 6. TestNormal

- Menguji perhitungan dalam kondisi **normal**, seperti  $3^3$ .
- Hasil yang diharapkan: **27**, sesuai perhitungan  $3 \times 3 \times 3$ .

### I. Hasil Running

#### 1. Hasil Running



## II. Kesimpulan

Pengembangan aplikasi GUI berbasis Windows Forms untuk menghitung nilai pangkat telah berhasil dilakukan dengan baik. Fungsi utama `CariNilaiPangkat` mampu menangani berbagai kondisi input secara akurat, termasuk kasus eksponen nol, eksponen negatif, batas nilai maksimum, dan overflow.

Validasi input serta penanganan kesalahan dirancang dengan logika yang jelas, sehingga memberikan umpan balik yang informatif bagi pengguna. Selain itu, unit testing dengan `MSTest` telah berhasil menguji semua cabang logika dalam fungsi tersebut, yang menjamin keandalan dan stabilitas program.

Unit test ini juga memastikan bahwa aplikasi tetap berfungsi dengan benar meskipun terjadi perubahan pada kode di masa depan. Secara keseluruhan, aplikasi ini tidak hanya memenuhi aspek fungsional, tetapi juga menunjukkan kualitas kode yang baik melalui penerapan praktik pengujian yang sistematis dan menyeluruh.