

**RAKTIKUM KONSTRUKSI PERANGKAT LUNAK
TUGAS JURNAL 12**

Performance Analysis Unit Testing & Debugging



**Telkom
University**

disusun Oleh:
Nita Fitrotul Mar'ah
2211104005
SE0601

Dosen Pengampu :
Yudha Islami Sulistya, S.Kom., M.Cs.

**S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

1. MEMBUAT PROJECT GUI BARU

Buka IDE misalnya dengan Visual Studio

- a. Misalnya menggunakan Visual Studio, buatlah project baru dengan nama modul12_NIM
- b. Pastikan project yang dibuat dapat menggunakan GUI (misalnya tipe Windows Form pada Visual Studio).

2. MEMBUAT GUI SEDERHANA DAN

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 “CariNilaiPangkat(int a, int b)” yang menerima dua input dan mengembalikan nilai berupa hasil pangkat a^b dengan melakukan iterasi (tanpa menggunakan library atau fungsi bawaan).
- c. Pada method tersebut terdapat aturan sebagai berikut (berbeda dengan aturan pangkat normal):
 - Apabila input b adalah 0 maka nilai return selalu 1 (walaupun nilai a adalah 0)
 - Apabila input b adalah bilangan negatif, maka nilai return adalah -1
 - Apabila input b lebih dari 10 atau input a lebih dari 100 maka nilai return adalah -2
 - Apabila hasil pangkat melebihi batas maksimal bilangan positif integer (misal dengan checked pada C#) maka nilai return adalah -3
- d. Pada tampilan GUI, pada saat tombol ditekan, maka label output akan menampilkan hasil pangkat dari pemanggilan fungsi “CariNilaiPangkat” dari dua input textbox.

Jawab:

- **Source code**
[Form1.cs](#)

```

using System;
using System.Windows.Forms;

namespace modul12_2211104005
{
    1 reference
    public partial class Form1 : Form
    {
        2 references
        private readonly PangkatCalculator _calculator;

        0 references
        public Form1()
        {
            InitializeComponent();
            _calculator = new PangkatCalculator(); // Menggunakan PangkatCalculator
        }

        0 references
        private void buttonHitung_Click(object sender, EventArgs e)
        {
            try
            {
                int a = int.Parse(textBox1.Text);
                int b = int.Parse(textBox2.Text);

                int result = _calculator.CariNilaiPangkat(a, b);

                switch (result)
                {
                    case -1:
                        labelHasil.Text = "Hasil: " + result;
                        break;
                    case -2:
                        labelHasil.Text = "Hasil: " + result;
                        break;
                    case -3:
                        labelHasil.Text = "Hasil: " + result;
                        break;
                    default:
                        labelHasil.Text = "Hasil: " + result;
                        break;
                }
            }
            catch (FormatException)
            {
                MessageBox.Show("Input tidak valid. Masukkan angka saja.");
            }
        }
    }
}

```

[PangkatCalculator.cs](#)

```

using System;

namespace modul12_2211104005
{
    0 references
    public class PangkatCalculator
    {
        0 references
        public int CariNilaiPangkat(int a, int b)
        {
            // Aturan 1: Jika b == 0, hasil selalu 1 (walaupun a adalah 0)
            if (b == 0) return 1;

            // Aturan 2: Jika b < 0, hasil adalah -1
            if (b < 0) return -1;

            // Aturan 3: Jika a > 100 atau b > 10, hasil adalah -2
            if (a > 100 || b > 10) return -2;

            // Aturan 4: Hitung pangkat dengan aman (tanpa Math.Pow) dan deteksi overflow
            try
            {
                checked
                {
                    int result = 1;
                    for (int i = 0; i < b; i++)
                    {
                        result = checked(result * a);
                    }
                    return result;
                }
            }
            catch (OverflowException)
            {
                return -3; // Overflow terdeteksi
            }

            return -3; // Error lain (jika ada)
        }
    }
}

```

- **Hasil:**

Apabila input b adalah 0 maka nilai return selalu 1 (walaupun nilai a adalah 0)

Apabila input b adalah bilangan negatif, maka nilai return adalah -1

Apabila input b lebih dari 10 atau input a lebih dari 100 maka nilai return adalah -2

- **Penjelasan**

File Form1.cs merupakan bagian dari aplikasi Windows Forms yang menyajikan antarmuka pengguna grafis untuk melakukan perhitungan perpangkatan. Pengguna dapat memasukkan dua nilai (a dan b) melalui TextBox, lalu menekan tombol "Hitung" untuk memulai proses. Aplikasi akan menggunakan objek dari kelas PangkatCalculator untuk menghitung hasil perpangkatan, dan hasilnya akan ditampilkan di Label.

Terdapat beberapa ketentuan dalam perhitungannya:

- Jika $b = 0$, maka hasilnya adalah 1.
- Jika $b < 0$, maka hasilnya adalah -1.
- Jika $a > 100$ atau $b > 10$, maka hasilnya adalah -2.

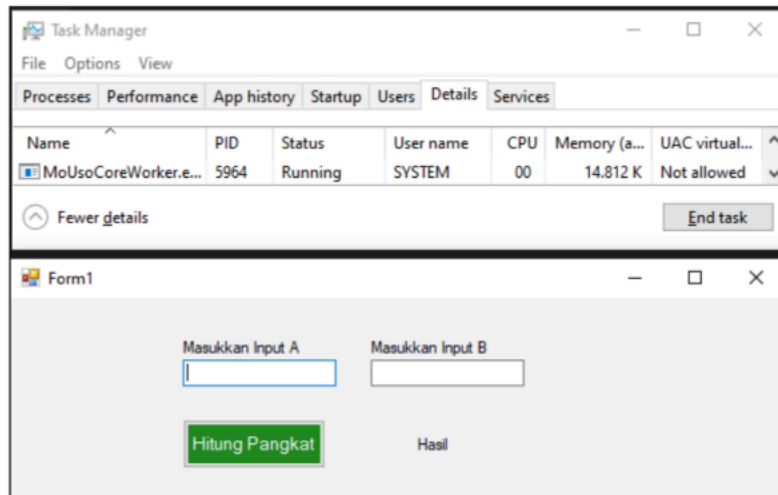
Jika terjadi overflow saat perhitungan (hasil terlalu besar untuk ditampung), maka hasilnya adalah -3.

Semua aturan dan logika ini diatur dalam file PangkatCalculator.cs, yang menggunakan perulangan (loop) dan pengecekan overflow melalui blok checked. Jika pengguna memasukkan nilai yang bukan angka, aplikasi akan menampilkan pesan kesalahan.

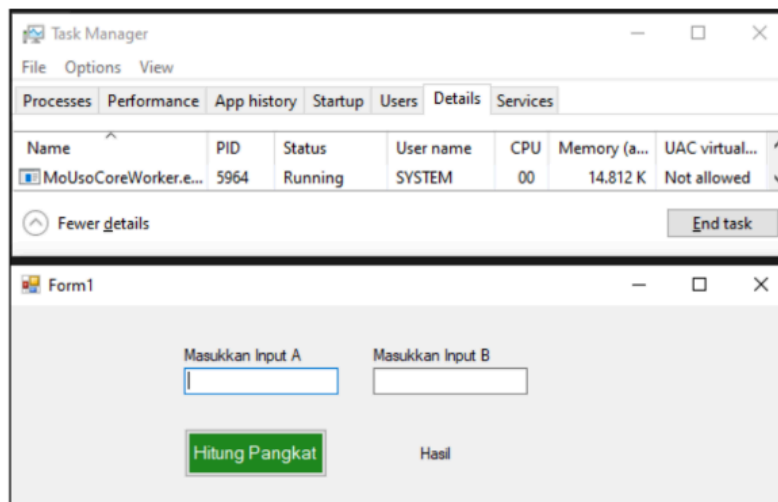
3. MELAKUKAN SOFTWARE PROFILING

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

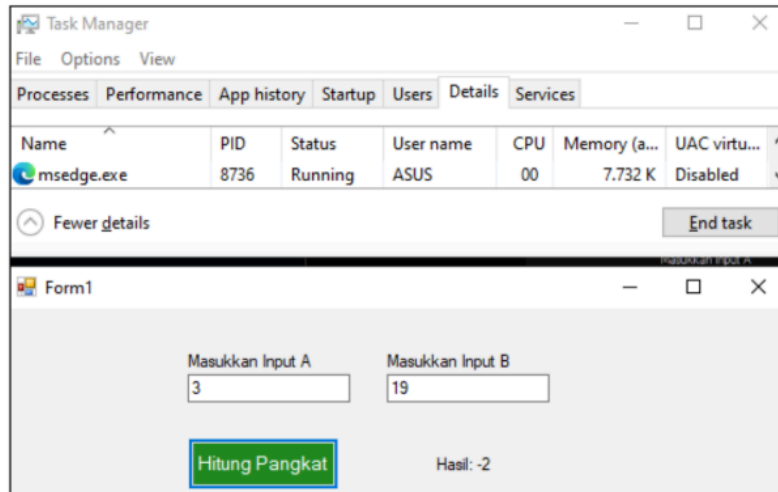
- a. Pada saat program berjalan, catat dan amati CPU usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.



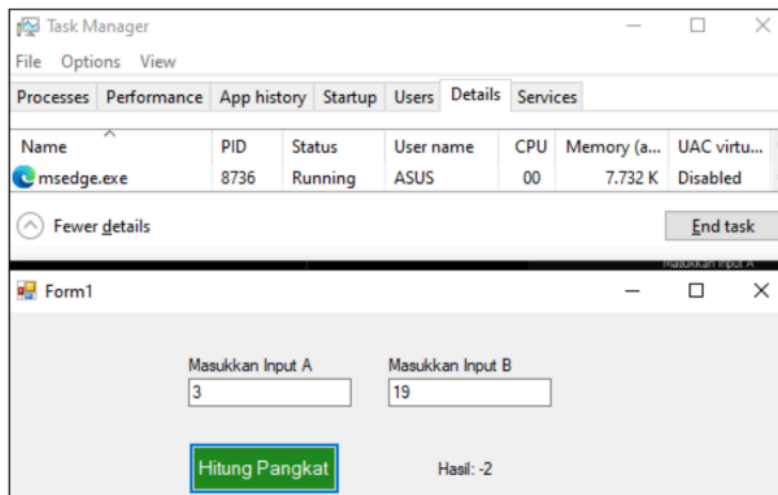
- b. Pada saat program berjalan, catat dan amati memory usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.



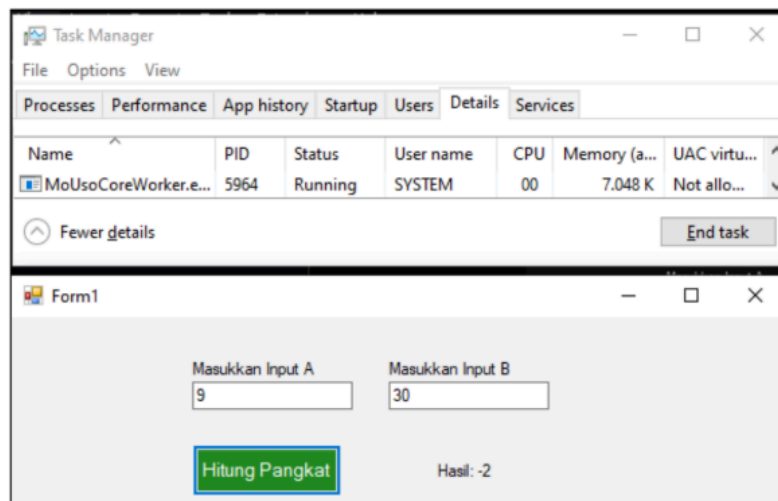
- c. Tambahkan input “3” pada textbox pertama dan “19” pada textbox ketiga, dan tekan tombol button dan catat dan amati memory usage dari aplikasi.



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



- e. Lakukan lagi eksperimen dengan input pertama yaitu “9” dan angka kedua yaitu “30”, laporkan apakah terdapat perubahan di CPU usage dan memory.



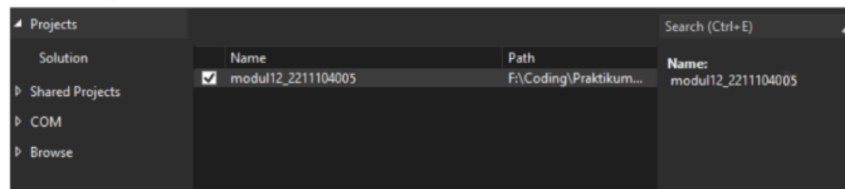
4. MENAMBAHKAN UNIT TESTING

Di dalam project yang sama:

- Buatlah kode unit test untuk menguji method “CariNilaiPangkat” yang dibuat sebelumnya.
- Pastikan kode unit test tersebut memiliki branch coverage yang baik untuk method “CariNilaiPangkat”.
- 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 modul12_2211104005;
using Xunit;

namespace modul12_2211104005.Tests
{
    1 reference
    public class CariNilaiPangkatTests
    {
        7 references
        private readonly PangkatCalculator _calculator;
        0 references
        public CariNilaiPangkatTests()
        {
            _calculator = new PangkatCalculator();
        }

        [Fact]
        0 references
        public void Test_B_EqualToZero_ReturnsOne()
        {
            int result = _calculator.CariNilaiPangkat(0, 0);
            Assert.Equal(1, result);
        }

        [Fact]
        0 references
        public void Test_B_Negative_ReturnsMinusOne()
        {
            int result = _calculator.CariNilaiPangkat(2, -3);
            Assert.Equal(-1, result);
        }

        [Fact]
        0 references
        public void Test_A_GreaterThan100_Or_B_GreaterThan10_ReturnsMinusTwo()
        {
            int result1 = _calculator.CariNilaiPangkat(101, 5);
            int result2 = _calculator.CariNilaiPangkat(5, 11);
            Assert.Equal(-2, result1);
            Assert.Equal(-2, result2);
        }

        [Fact]
        0 references
        public void Test_Overflow_ReturnsMinusThree()
        {
            int result = _calculator.CariNilaiPangkat(200, 5); // 200^5 melebihi batas int
            Assert.Equal(-3, result);
        }

        [Fact]
        0 references
        public void Test_NormalPangkatCalculation()
        {
            int result = _calculator.CariNilaiPangkat(2, 3); // 2^3 = 8
            Assert.Equal(8, result);
        }
    }
}

```

- Hasil testing:

Test	Durati...	Traits	Error Message	Run	Debug
modul12_2211104005.Tests (4)	475 ms				
modul12_2211104005.Tests (4)	475 ms				
CariNilaiPangkatTests (4)	475 ms				
Test_A_GreaterThan100_Or_B_G...	< 1 ms				
Test_B_EqualToZero_ReturnsOne	< 1 ms				
Test_NormalPangkatCalculation	< 1 ms				
Test_B_Negative_ReturnsMinus...	475 ms				
Group Summary					
modul12_2211104005.Tests					
Tests in group: 4					
Total Duration: 475 ms					
Outcomes					
4 Passed					