

**TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK**

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



**Disusun Oleh :
Atika Aji Hadiyani
2211104003
SE-06-01**

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

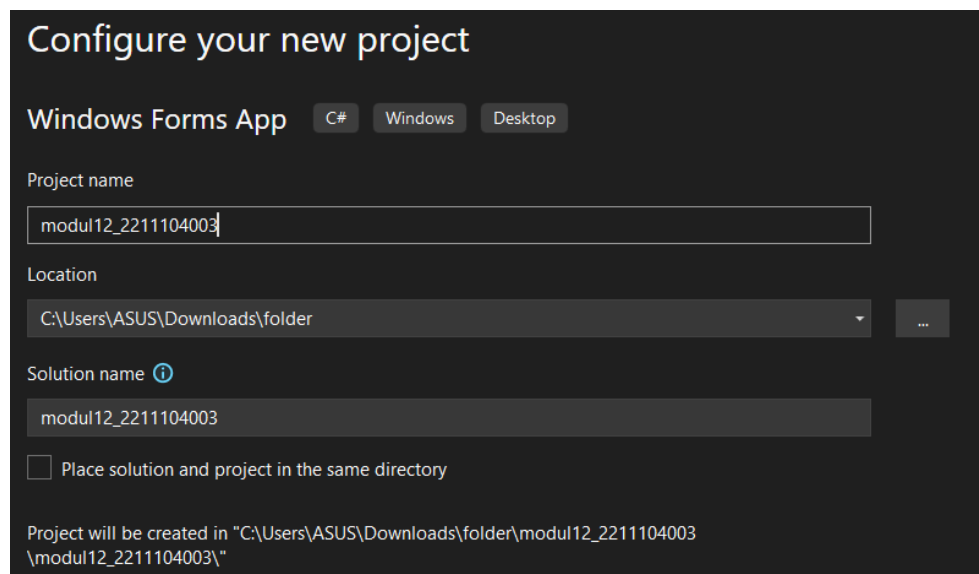
**PROGRAM STUDI S1 SOFTWARE ENGINEERING
TELKOM UNIVERSITY PURWOKERTO
2025**

TUGAS JURNAL 12

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 ab 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

a. File Form1.cs

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

namespace modul12_2211184883
{
    3 references
    public partial class Form1 : Form
    {
        private readonly PangkatCalculator _calculator;

        1 reference
        public Form1()
        {
            InitializeComponent();
            _calculator = new PangkatCalculator(); // Menggunakan PangkatCalculator
        }

        1 reference
        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.");
            }
        }
    }
}
```

b. File PangkatCalculator.cs

```
using System;

namespace modul12_2211104003
{
    2 references
    public class PangkatCalculator
    {
        1 reference
        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)

Form1

Masukkan Input A Masukkan Input B

0 0

Hitung Pangkat Hasil: 1

- 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

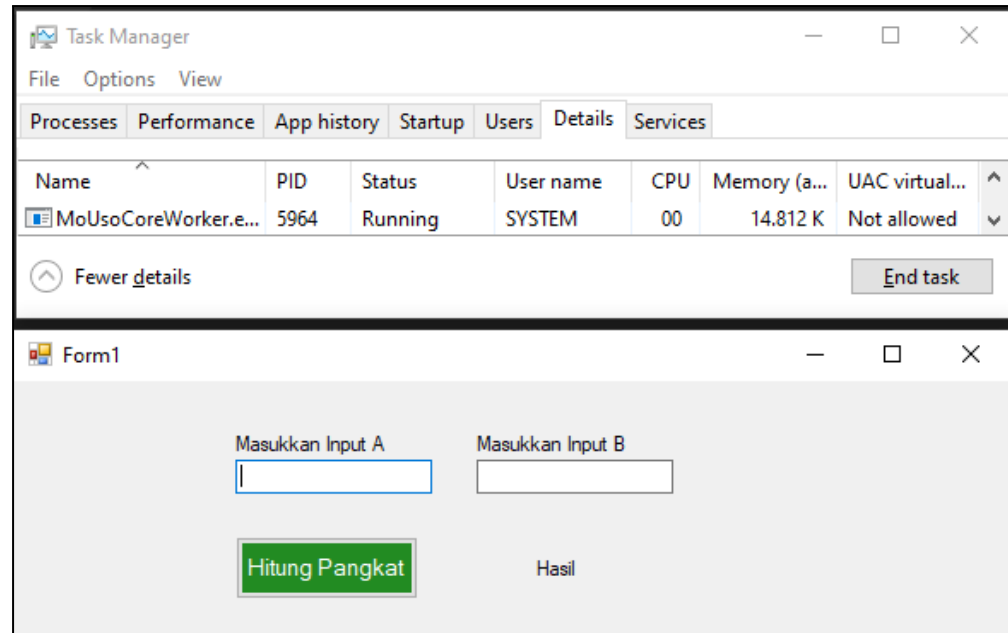
Penjelasan:

Form1.cs merupakan komponen dari aplikasi Windows Forms yang menampilkan antarmuka pengguna untuk melakukan perhitungan eksponen suatu angka. Pengguna dapat mengisi dua nilai numerik (a dan b) ke dalam TextBox, lalu mengklik tombol "Hitung" untuk memulai proses. Perhitungan dilakukan menggunakan objek bernama PangkatCalculator, dan hasilnya akan ditampilkan melalui komponen label. Terdapat beberapa kondisi khusus dalam logika perhitungan: jika nilai b adalah 0, maka hasilnya 1; jika b bernilai negatif, maka hasilnya -1; jika nilai a melebihi 100 atau b lebih besar dari 10, maka hasil dikembalikan sebagai -2. Apabila terjadi overflow (hasil melebihi kapasitas data), maka sistem akan mengembalikan nilai -3. Seluruh logika ini didefinisikan dalam file PangkatCalculator.cs, yang memanfaatkan perulangan (looping) dan mekanisme pengecekan overflow dengan kata kunci *checked*. Jika input yang diberikan bukan angka valid, maka aplikasi akan menampilkan peringatan kesalahan kepada pengguna.

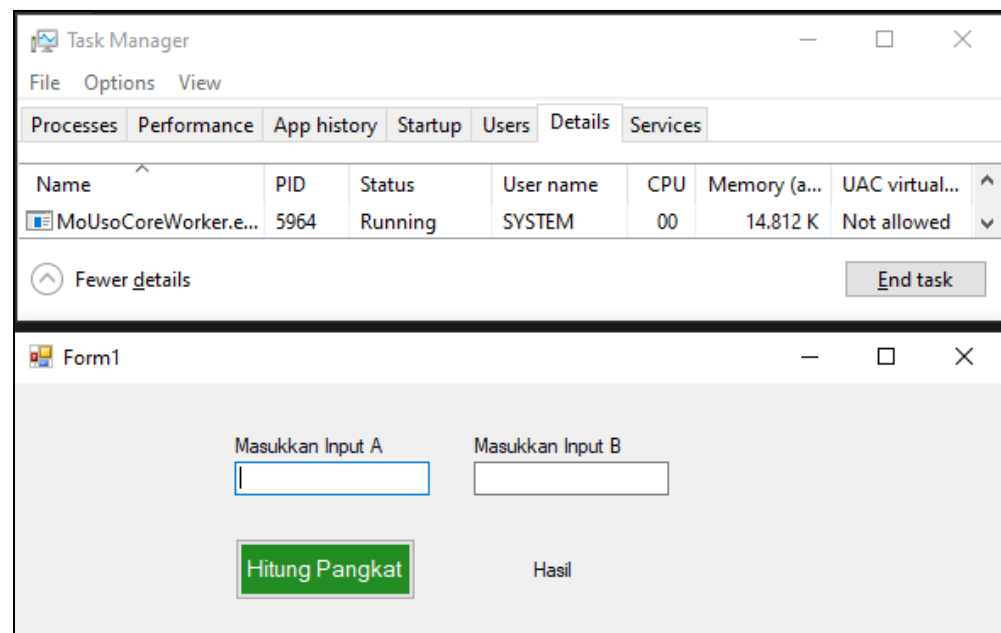
3. MELAKUKAN SOFTWARE PROFILING

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

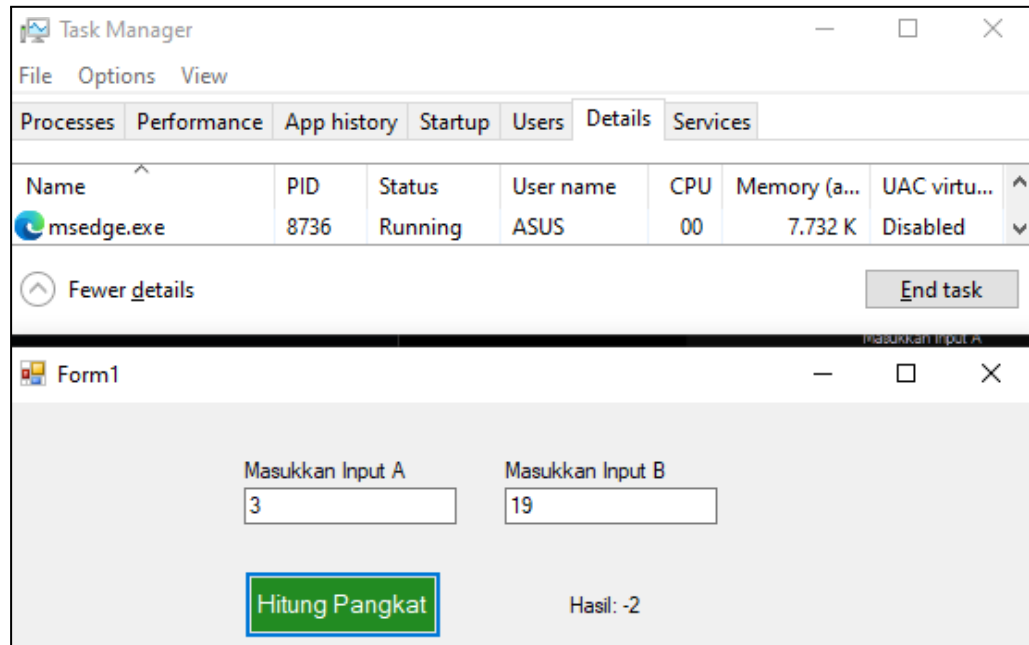
- A. Pada saat program berjalan, catat dan amati CPU usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.



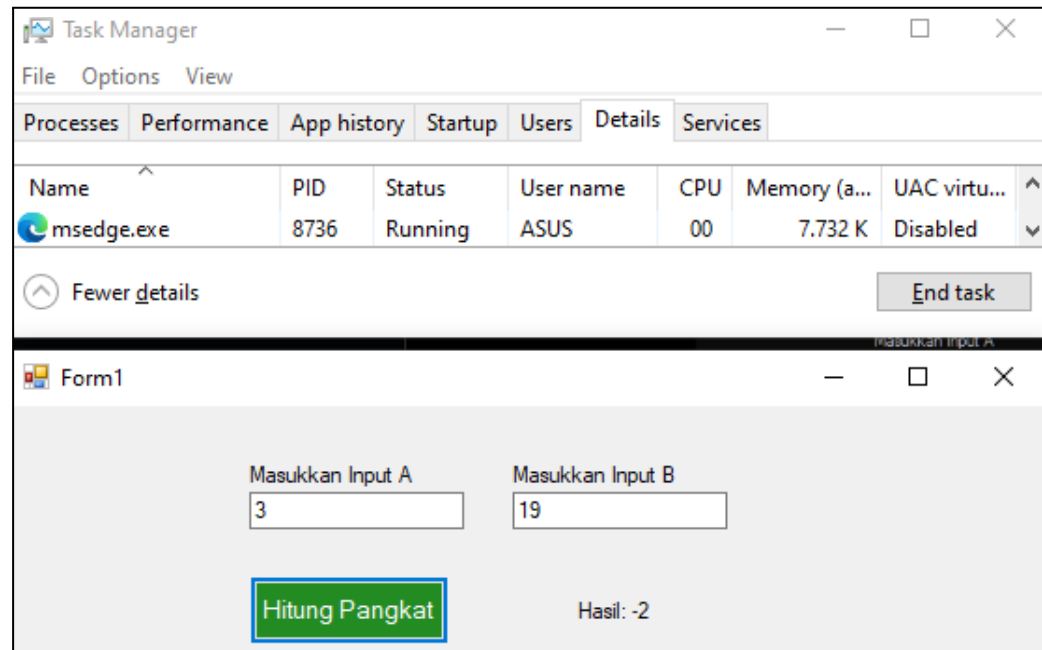
- B. Pada saat program berjalan, catat dan amati memory usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.



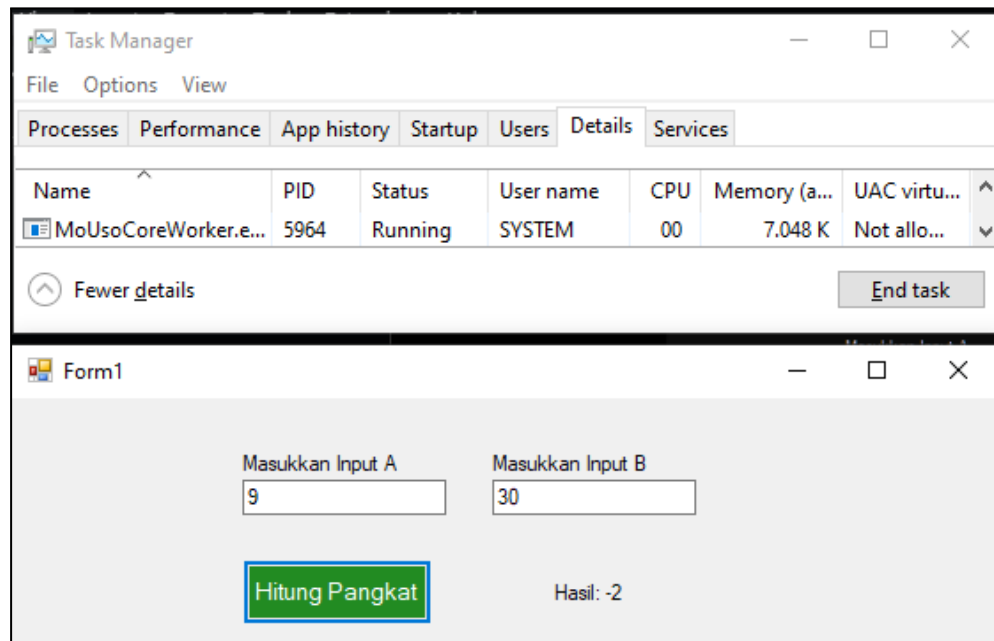
- C. Tambahkan input “3” pada textbox pertama dan “19” pada textbox ketiga, dan tekan tombol button dan catat dan amati memory usage dari aplikasi.



- 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.



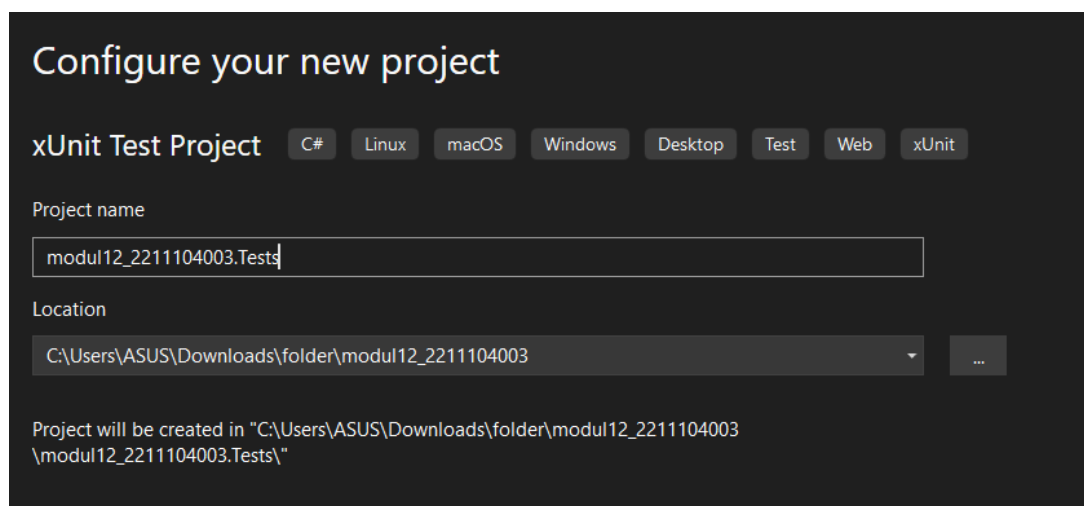
4. MENAMBAHKAN UNIT TESTING

Di dalam project yang sama:

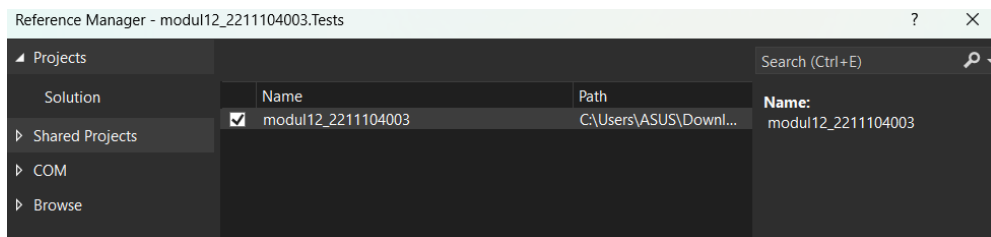
- Buatlah kode unit test untuk menguji method “CariNilaiPangkat” yang dibuat sebelumnya.
- Pastikan kode unit test tersebut memiliki branch coverage yang baik untuk method “CariNilaiPangkat”.
- Jalankan kode unit test yang dibuat dan lampirkan hasil unit testing yang dilakukan.

Jawab:

- Membuat Project untuk Testing



- Membuat referensi



- Source code file UnitTest1.cs

```
using modul12_2211104003;
using Xunit;

namespace modul12_2211104003.Tests
{
    1 reference
    public class CariNilaiPangkatTests
    {
        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 Explorer

Test run finished: 5 Tests (4 Passed, 1 Failed, 0 Skipped) run in 1.8 min

0 Warnings 1 Error

Test	Duration	Traits	Error Message
modul12_2211104003.Tests (4)	1 ms		
modul12_2211104003.Tests (4)	1 ms		
CariNilaiPangkatTests (4)	1 ms		
Test_A_GreaterThan100_Or_B_G...	< 1 ms		
Test_B_EqualToZero_ReturnsOne	< 1 ms		
Test_B_Negative_ReturnsMinus...	< 1 ms		
Test_NormalPangkatCalculation	1 ms		

Run | Debug

Group Summary

modul12_2211104003.Te

Tests in group: 4

Total Duration: 1

Outcomes

4 Passed