

TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK

MODUL XII
PERFORMANCE ANALYSIS, UNIT TESTING, DAN DEBUGGING



Disusun Oleh:

Aorinka Anendya Chazanah / 2211104013

S1 SE-06-01

Dosen Pengampu:

Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

TUGAS JURNAL

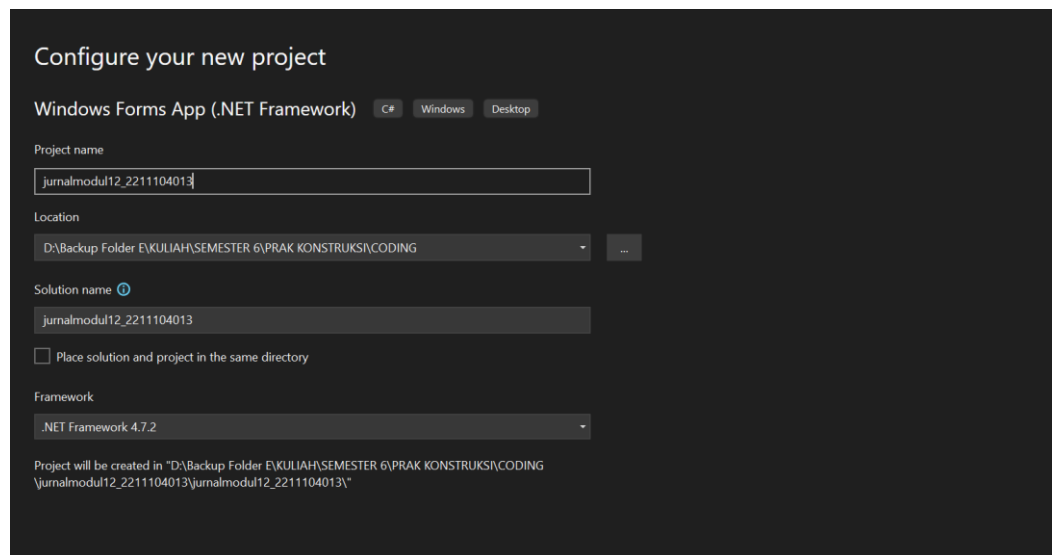
1. MEMBUAT PROJECT GUI SEDERHANA

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 (walapupun 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.

Jawaban

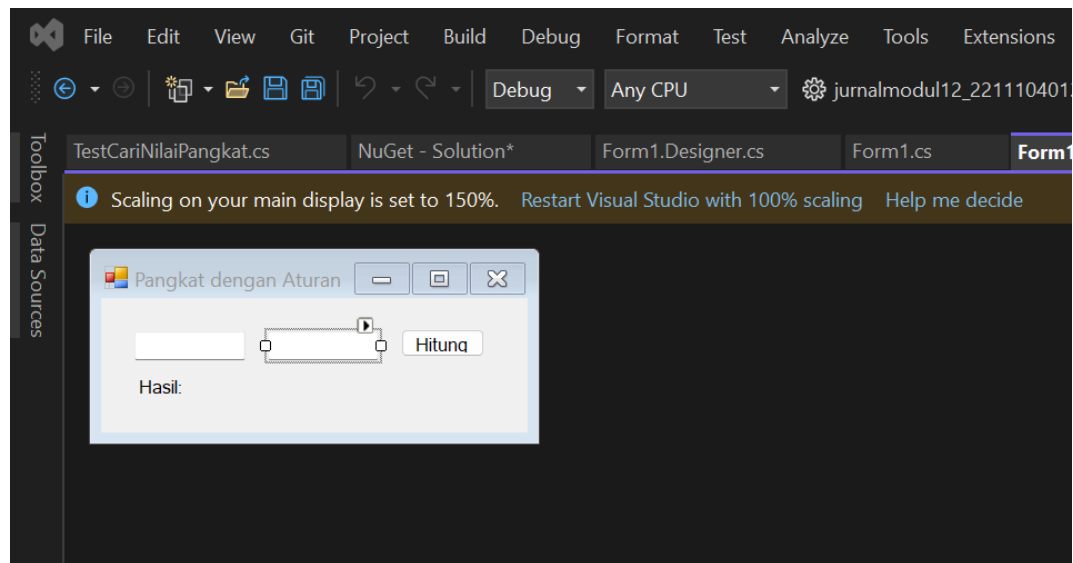
Membuat project baru dengan nama **jurnalmodul12_2211104013**



File Form1.cs

```
1 using System;
2 using System.Windows.Forms;
3
4 namespace jurnalmodul12_2211104013
5 {
6     5 references
7     public partial class Form1 : Form
8     {
9         2 references
10        public Form1()
11        {
12            InitializeComponent();
13        }
14
15        1 reference
16        private void btnHitung_Click(object sender, EventArgs e)
17        {
18            int a, b;
19            if (int.TryParse(txtA.Text, out a) && int.TryParse(txtB.Text, out b))
20            {
21                int hasil = CariNilaiPangkat(a, b);
22                lblOutput.Text = "Hasil: " + hasil.ToString();
23            }
24            else
25            {
26                lblOutput.Text = "Input tidak valid!";
27            }
28        }
29
30        8 references | 6/6 passing
31        public int CariNilaiPangkat(int a, int b)
32        {
33            // Kasus jika b adalah 0, hasilnya selalu 1
34            if (b == 0) return 1;
35
36            // Kasus jika b negatif
37            if (b < 0) return -1;
38
39            // Kasus jika a > 100 atau b > 10, kembalikan -2
40            if (a > 100 || b > 10) return -2;
41
42            // Variabel untuk menyimpan hasil
43            long result = 1;
44
45            // Menggunakan iterasi untuk menghitung pangkat
46            try
47            {
48                for (int i = 0; i < b; i++)
49                {
50                    checked
51                    {
52                        result *= a;
53                    }
54
55                    // Cek overflow secara manual pada tiap langkah
56                    if (result > int.MaxValue || result < int.MinValue)
57                    {
58                        return -2; // Overflow terjadi
59                    }
60                }
61            }
62            catch (OverflowException)
63            {
64                return -2; // Mengembalikan -2 jika overflow
65            }
66
67            return (int)result; // Mengembalikan hasil setelah overflow dicegah
68        }
69    }
70 }
```

Form1.cs (design)



Penjelasan Kode Program

Program ini merupakan aplikasi Windows Form sederhana yang memungkinkan pengguna untuk menghitung hasil perpangkatan dari dua bilangan bulat yang dimasukkan melalui antarmuka. Program terdiri dari form utama bernama Form1, yang memiliki dua kotak input (untuk nilai a dan b), sebuah tombol hitung (btnHitung), dan label untuk menampilkan hasil (lblOutput). Saat tombol diklik, program mencoba mengubah input menjadi bilangan bulat menggunakan `int.TryParse`. Jika input valid, program memanggil metode `CariNilaiPangkat` untuk menghitung hasil a^b dan menampilkan hasilnya. Jika tidak valid, maka akan muncul pesan "Input tidak valid!"

Metode `CariNilaiPangkat` sendiri berisi beberapa logika pengecekan. Jika pangkat b adalah 0, hasilnya langsung dikembalikan sebagai 1. Jika b negatif, maka dikembalikan -1. Bila a lebih dari 100 atau b lebih dari 10, maka hasilnya -2 sebagai tanda bahwa nilai terlalu besar untuk diproses. Untuk menghitung pangkat, digunakan perulangan dan operasi perkalian dalam blok `checked` guna mendeteksi kemungkinan overflow. Jika overflow terdeteksi secara manual atau melalui pengecualian, maka dikembalikan -2. Jika tidak ada masalah, hasil akhir dikembalikan sebagai bilangan bulat. Program ini memastikan perhitungan yang aman dan memberikan kode kesalahan yang jelas jika terjadi kondisi tak normal.

2. MELAKUKAN SOFTWARE PROFILING

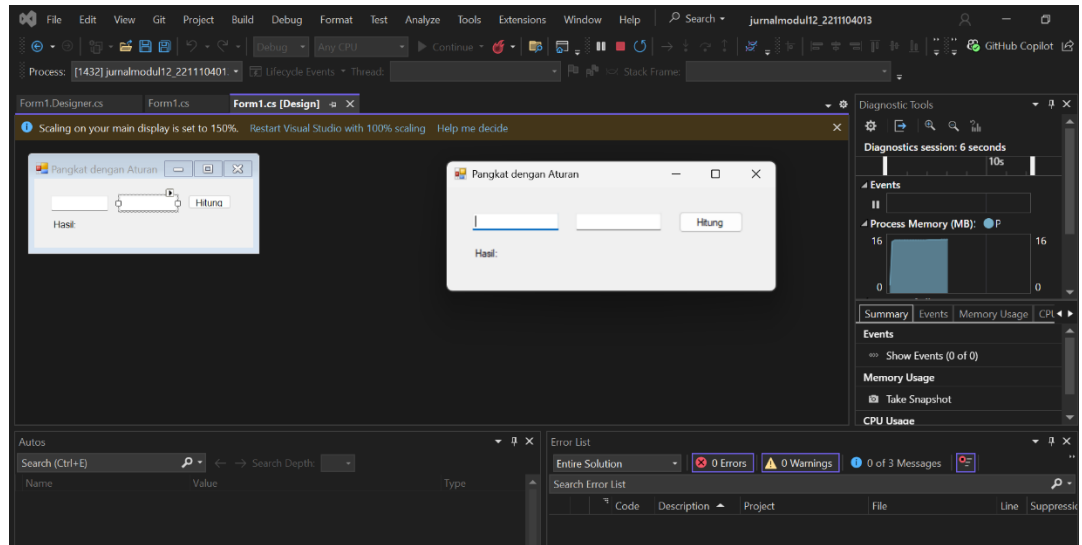
Jalankan project yang dibuat sebelumnya dan jalankan profiling tools (misal dari visual studio, task manager atau sejenisnya):

- Pada saat program berjalan, catat dan amati CPU usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.
- Pada saat program berjalan, catat dan amati memory usage dari aplikasi yang sedang berjalan tanpa melakukan input apapun.
- Tambahkan input "3" pada textbox pertama dan "19" pada textbox ketiga, dan tekan tombol button dan catat dan amati memory usage dari aplikasi.

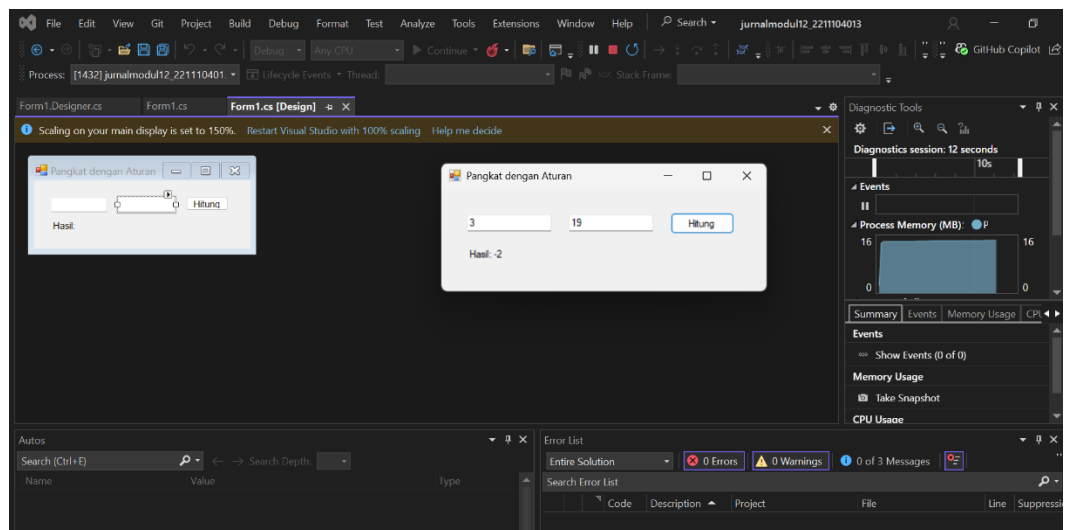
- D. Laporkan apakah terdapat perubahan pada CPU usage dan memory (apabila tidak ada perubahan juga perlu dilaporkan di file docx).
- E. Lakukan lagi eksperimen dengan input pertama yaitu “9” dan angka kedua yaitu “30”, laporkan apakah terdapat perubahan di CPU usage dan memory.

Jawaban

Berikut adalah tampilan ketika program dijalankan, ketika tombol Run pertama kali di tekan Process Memory langsung menunjukkan pada grafik angka 16 seperti gambar berikut di bagian sebelah kanan.



Setelah itu, ketika mencoba menginputkan pada textbox pertama dan textbox kedua hingga output ditampilkan, Process Memory juga tetap menunjukkan grafik angka yang sama, artinya tidak terjadi perubahan apapun.



3. 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.

Jawaban

File TestCariNilaiPangkat.cs

```
1  using System;
2  using Xunit;
3  using jurnalmodul12_2211104013; // Menggunakan namespace project utama
4
5  namespace jurnalmodul12_2211104013.Tests
6  {
7      1 reference
8      public class TestCariNilaiPangkat
9      {
10
11          private Form1 form;
12
13          0 references
14          public TestCariNilaiPangkat()
15          {
16              // Inisialisasi form setiap kali tes dijalankan
17              form = new Form1();
18          }
19
20          [Fact]
21          0 references
22          public void Test_B_Pangkat_0()
23          {
24              // Tes untuk pangkat 0, hasilnya selalu 1
25              Assert.Equal(1, form.CariNilaiPangkat(0, 0));
26              Assert.Equal(1, form.CariNilaiPangkat(5, 0));
27          }
28
29          [Fact]
30          0 references
31          public void Test_B_Negatif()
32          {
33              // Tes untuk b negatif, harus mengembalikan -1
34              Assert.Equal(-1, form.CariNilaiPangkat(2, -1));
35          }
36
37          [Fact]
```

Penjelasan Kode Program Unit Test

Kode program ini merupakan unit test menggunakan framework **xUnit** untuk menguji fungsionalitas dari metode `CariNilaiPangkat` dalam kelas `Form1`. Setiap metode pengujian (ditandai dengan atribut `[Fact]`) menguji kondisi yang berbeda. Misalnya, `Test_B_Pangkat_0` menguji apakah hasil perpangkatan dengan pangkat 0 selalu mengembalikan nilai 1. `Test_B_Negatif` memverifikasi bahwa jika pangkat (b) bernilai negatif, maka metode harus mengembalikan -1. `Test_B_LebihDari10` dan `Test_A_LebihDari100` menguji batasan maksimum dari input, yaitu b tidak boleh lebih dari 10 dan a tidak boleh lebih dari 100, yang keduanya harus mengembalikan -2 jika dilanggar. `Test_Normal` memastikan bahwa perhitungan biasa (misalnya 2^3) menghasilkan output yang benar, yaitu 8. Sementara itu, `Test_Overflow` mengecek apakah perhitungan dengan bilangan besar yang berpotensi overflow mengembalikan

nilai -2 sebagai penanda error. Dengan pengujian ini, kita bisa yakin bahwa fungsi CariNilaiPangkat bekerja dengan benar dalam berbagai situasi.

```
32 [Fact]
33 | 0 references
34 public void Test_B_LebihDari10()
35 {
36     // Tes untuk b > 10, harus mengembalikan -2
37     Assert.Equal(-2, form.CariNilaiPangkat(2, 11));
38 }
39
40 [Fact]
41 | 0 references
42 public void Test_A_LebihDari100()
43 {
44     // Tes untuk a > 100, harus mengembalikan -2
45     Assert.Equal(-2, form.CariNilaiPangkat(101, 5));
46 }
47
48 [Fact]
49 | 0 references
50 public void Test_Normal()
51 {
52     // Tes untuk kasus normal, hasil pangkat yang valid
53     Assert.Equal(8, form.CariNilaiPangkat(2, 3));
54 }
55
56 [Fact]
57 | 0 references
58 public void Test_Overflow()
59 {
60     // Tes untuk overflow, harus mengembalikan -3
61     Assert.Equal(-2, form.CariNilaiPangkat(100000, 5));
62 }
```

Hasil Unit Test

Method CariNilaiPangkat berhasil diuji semua.

The screenshot shows the Test Explorer in Visual Studio. The test run is finished, showing 7 tests passed, 0 failed, and 0 skipped. The tests are listed in a table with columns for Test, Duration, Traits, and Error Message. The tests are all successful, with durations ranging from 1 ms to 662 ms. The right pane shows the test detail summary for the selected test, indicating it passed and showing the source file and line number.

Test	Duration	Traits	Error Message
jurnalmodul12_2211104013 (6)	662 ms		
jurnalmodul12_2211104013.Tests (6)	662 ms		
TestCariNilaiPangkat (6)	662 ms		
Test_A_LebihDari100	1 ms		
Test_B_LebihDari10	1 ms		
Test_B_Negatif	1 ms		
Test_B_Pangkat_0	1 ms		
Test_Normal	657 ms		
Test_Overflow	1 ms		