JURNAL
KONSTRUKSI PERANGKAT LUNAK

PERTEMUAN 12

PERFORMANCE ANALYSIS,

UNIT TESTING, DANDEBUGGING

Disusun Oleh :

Muhammad Abdul Aziz

2211104026

SE0601

Asisten Praktikum :

Naufal El Kamil Aditya Pratama Rahman

Imelda

Dosen Pengampu :

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

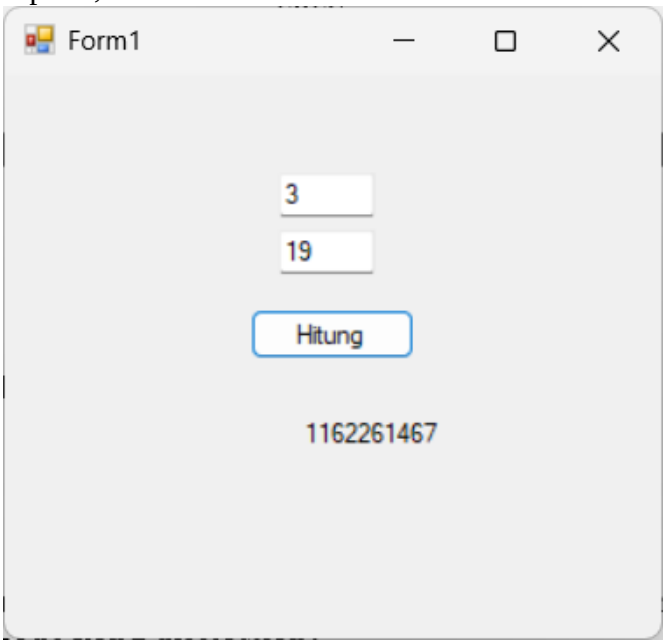
PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

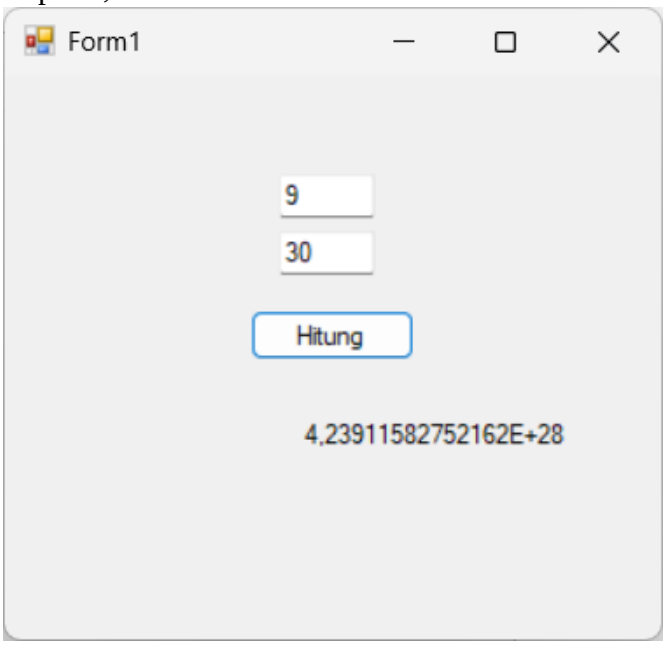
TELKOM UNIVERSITY PURWOKERTO

2025

1. **Screenshoot Hasil Run**
   - Input 3, 19



   - Imput 9, 30



2. **Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).**
   Source Code :

Program.cs

```
l12_2211104026                                  modul12_2211104026.Program
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Threading.Tasks;
5   using System.Windows.Forms;
6
7   namespace modul12_2211104026
8   {
        0 references
9       static class Program
10      {
11          /// <summary>
12          /// The main entry point for the application.
13          /// </summary>
14          [STAThread]
            0 references
15          static void Main()
16          {
17              Application.EnableVisualStyles();
18              Application.SetCompatibleTextRenderingDefault(false);
19              Application.Run(new Form1());
20          }
21      }
22  }
```

Form1.cs

```
dul12_2211104026                                modul12_2211104026.Form1
1   using System;
2   using System.Windows.Forms;
3
4   namespace modul12_2211104026
5   {
        9 references
6       public partial class Form1 : Form
7       {
            7 references
8           public Form1()
9           {
10              InitializeComponent();
11          }
12
13          // Fungsi CariNilaiPangkat
            6 references
14          public int CariNilaiPangkat(int a, int b)
15          {
16              if (b == 0) return 1;
17              if (b < 0) return -1;
18              if (b > 10 || a > 100) return -2;
19
20              try
21              {
22                  checked
23                  {
24                      int hasil = 1;
25                      for (int i = 0; i < b; i++)
26                      {
27                          hasil *= a;
28                      }
29                      return hasil;
```

```
30                  }
31              }
32              catch (OverflowException)
33              {
34                  return -3;
35              }
36          }
37
38          // Event handler untuk button
            1 reference
39          private void btnHitung_Click_1(object sender, EventArgs e)
40          {
41              try
42              {
43                  double a = double.Parse(txtA.Text);
44                  double b = double.Parse(txtB.Text);
45                  double hasil = Math.Pow(a, b);
46                  labelHasil.Text = hasil.ToString();
47              }
48              catch (Exception ex)
49              {
50                  MessageBox.Show("Input tidak valid: " + ex.Message);
51              }
52          }
53      }
54  }
```

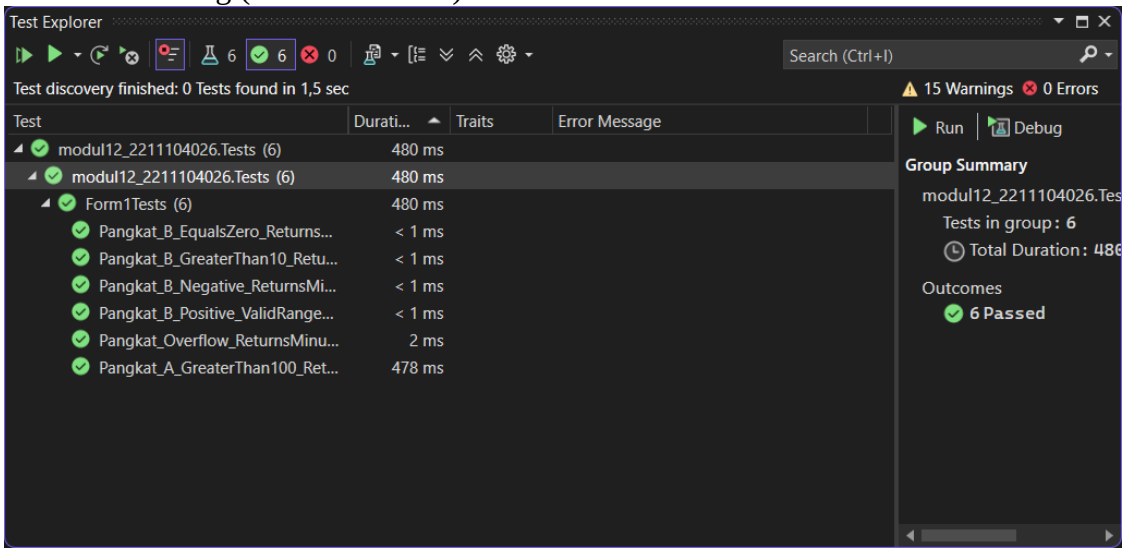
Form1.Designer.cs

```
ul12_2211104026                                  modul12_2211104026.Form1
1   namespace modul12_2211104026
2   {
3       partial class Form1
4       {
5           private System.ComponentModel.IContainer components = null;
6
7           protected override void Dispose(bool disposing)
8           {
9               if (disposing && (components != null))
10              {
11                  components.Dispose();
12              }
13              base.Dispose(disposing);
14          }
15
16          Windows Form Designer generated code
86
87          private System.Windows.Forms.Button btnHitung;
88          private System.Windows.Forms.TextBox txtA;
89          private System.Windows.Forms.TextBox txtB;
90          private System.Windows.Forms.Label lblHasil;
91          private System.Windows.Forms.Label labelHasil;
92      }
93  }
```

UnitTest.cs

```cs
using Microsoft.VisualStudio.TestTools.UnitTesting;
using modul12_2211104026;

namespace modul12_2211104026.Tests
{
    [TestClass]
    0 references
    public class Form1Tests
    {
        private Form1 form;

        [TestInitialize]
        0 references
        public void Setup()
        {
            form = new Form1();
        }

        [TestMethod]
        0 references
        public void Pangkat_B_Positive_ValidRange_ReturnsCorrectResult()
        {
            int result = form.CariNilaiPangkat(2, 3);
            Assert.AreEqual(8, result);
        }

        [TestMethod]
        0 references
        public void Pangkat_B_EqualsZero_ReturnsOne()
        {
            int result = form.CariNilaiPangkat(5, 0);
            Assert.AreEqual(1, result);
        }

        [TestMethod]
        0 references
        public void Pangkat_B_Negative_ReturnsMinusOne()
        {
            int result = form.CariNilaiPangkat(3, -2);
            Assert.AreEqual(-1, result);
        }

        [TestMethod]
        0 references
        public void Pangkat_B_GreaterThan10_ReturnsMinusTwo()
        {
            int result = form.CariNilaiPangkat(2, 11);
            Assert.AreEqual(-2, result);
        }

        [TestMethod]
        0 references
        public void Pangkat_A_GreaterThan100_ReturnsMinusTwo()
        {
            int result = form.CariNilaiPangkat(101, 2);
            Assert.AreEqual(-2, result);
        }

        [TestMethod]
        0 references
        public void Pangkat_Overflow_ReturnsMinusThree()
        {
            int result = form.CariNilaiPangkat(100, 10); // 100^10 = big number, likely overflow
            Assert.AreEqual(-3, result);
        }
    }
}
```

Hasil Unit Tesing (Berhasil Semua)



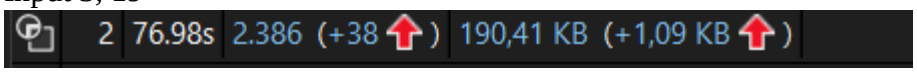
Penjelasan

Aplikasi ini merupakan program Windows Forms yang menghitung nilai pemangkatan dari dua bilangan dengan validasi input dan pengujian unit. Fungsi utama CariNilaiPangkat(int a, int b) akan mengembalikan hasil pangkat jika nilai b berada dalam rentang 1–10 dan a tidak melebihi 100. Fungsi ini juga menangani kondisi khusus, seperti nilai b nol (hasil 1), b negatif (hasil -1), batasan nilai berlebih (hasil -2), dan overflow perhitungan (hasil -3). Untuk memastikan keandalan fungsi, telah dilakukan 6 pengujian unit menggunakan MSTest yang mencakup semua cabang logika. Seluruh tes berhasil dijalankan dan menghasilkan output yang sesuai dengan ekspektasi, menunjukkan bahwa fungsi CariNilaiPangkat telah bekerja dengan benar dan stabil.

3. CPU & Memory Usage

- Tanpa input apapun

- Input 3, 19



- Imput 9, 30