

Nama: Ahmad Al – Farizi

NIM: 2311104054

- i. Link github: https://github.com/alxfarizi/KPL_AHMAD-AL-FARIZI_2311104054_S1SE-07-02/tree/main/12_Performance_Analysis_Unit_Testing_dan_Debugging/Jurnal
- ii. Hasil running:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
zsh - Modul12_Tests + - - - - X

Modul12_Tests git:(main) x dotnet test
Restore complete (0.8s)
Modul12_Lib succeeded (0.4s) -> /Users/ahmadalfarizi/Jurnal/Modul12_2311104054/Modul12_Lib/bin/Debug/net9.0/Modul12_Lib.dll
Modul12_Tests succeeded (0.3s) -> bin/Debug/net9.0/Modul12_Tests.dll
NUnit Adapter 5.0.0.0: Test execution started
Running all tests in /Users/ahmadalfarizi/Jurnal/Modul12_2311104054/Modul12_Tests/bin/Debug/net9.0/Modul12_Tests.dll
NUnit3TestExecutor discovered 6 of 6 NUnit test cases using Current Discovery mode, Non-Explicit run
NUnit Adapter 5.0.0.0: Test execution complete
Modul12_Tests test succeeded (2.4s)

Test summary: total: 6, failed: 0, succeeded: 6, skipped: 0, duration: 2.4s
Build succeeded in 4.9s
Modul12_Tests git:(main) x
```

iii. Penjelasan source code:

Jurnal/Modul12_2311104054/Modul12_GUI/Views/MainWindow.axaml

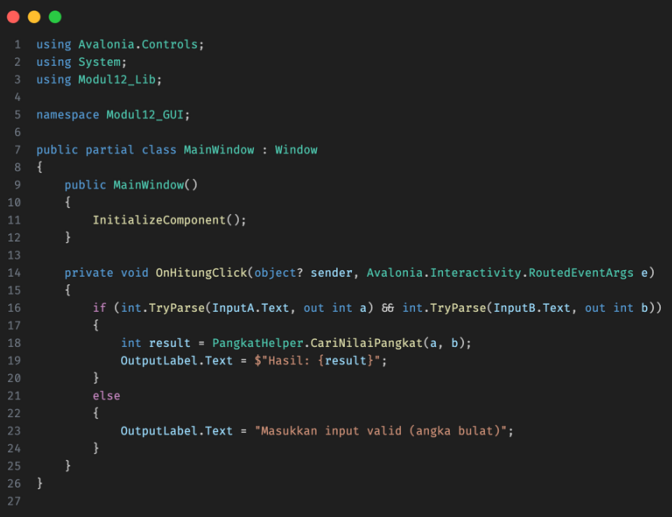
Berisi definisi antarmuka pengguna (GUI) menggunakan markup XAML. Elemen UI seperti input field untuk angka, tombol perhitungan, dan area untuk menampilkan hasil. Layout diatur menggunakan kontainer seperti StackPanel atau Grid untuk menempatkan komponen secara responsif. Tujuan utamanya adalah memungkinkan pengguna memasukkan data dan melihat output secara visual.

A screenshot of a code editor showing XAML code for a WPF window. The code defines a window titled "Cari Nilai Pangkat" with a width of 400 and height of 200. It contains a StackPanel with two text boxes (InputA and InputB) and a button labeled "Hitung Pangkat". The button has a Click event named "OnHitungClick". An output label is also defined.

```
1 <Window xmlns="https://github.com/avaloniaui"
2       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
3       x:Class="Modul12_GUI.MainWindow"
4       Width="400" Height="200"
5       Title="Cari Nilai Pangkat">
6
7     <StackPanel Margin="20" Spacing="10">
8       <TextBox Name="InputA" Watermark="Input a" />
9       <TextBox Name="InputB" Watermark="Input b" />
10      <Button Click="OnHitungClick">Hitung Pangkat</Button>
11      <TextBlock Name="OutputLabel" />
12    </StackPanel>
13 </Window>
14
```

Jurnal/Modul12_2311104054/Modul12_GUI/Views/MainWindow.axaml.cs

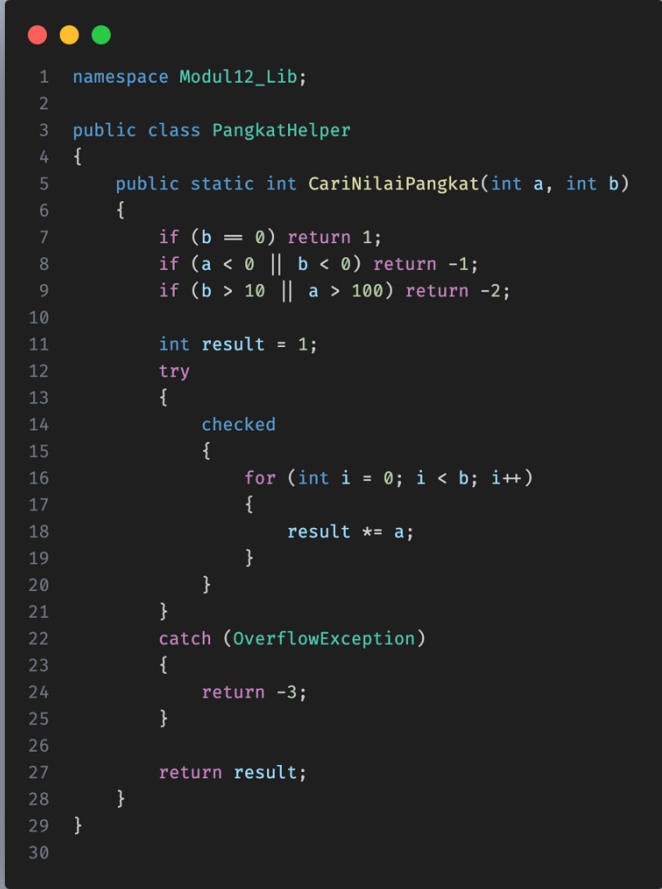
Code behind dari GUI, menghubungkan logika aplikasi dengan tampilan. Pada source code, terdapat event handler seperti OnHitungClicked yang dipicu saat tombol ditekan. Kode ini mengambil input dari UI, memanggil metode dari kelas logika, lalu menampilkan hasil ke UI. Error handling sederhana juga terlihat untuk memastikan aplikasi tidak crash.

A screenshot of a code editor showing C# code for the MainWindow class. The code includes using statements for Avalonia.Controls, System, and Modul12_Lib. It defines a partial class MainWindow : Window with a constructor that calls InitializeComponent(). The OnHitungClick event handler is implemented, which parses the input from InputA and InputB, calls PangkatHelper.CariNilaiPangkat(a, b), and updates the OutputLabel.Text with the result or an error message.

```
1 using Avalonia.Controls;
2 using System;
3 using Modul12_Lib;
4
5 namespace Modul12_GUI;
6
7 public partial class MainWindow : Window
8 {
9     public MainWindow()
10     {
11         InitializeComponent();
12     }
13
14     private void OnHitungClick(object? sender, Avalonia.Interactivity.RoutedEventArgs e)
15     {
16         if (int.TryParse(InputA.Text, out int a) && int.TryParse(InputB.Text, out int b))
17         {
18             int result = PangkatHelper.CariNilaiPangkat(a, b);
19             OutputLabel.Text = $"Hasil: {result}";
20         }
21         else
22         {
23             OutputLabel.Text = "Masukkan input valid (angka bulat)";
24         }
25     }
26 }
27
```

Jurnal/Modul12_2311104054/Modul12_Lib/PangkatHelper.cs

Kelas utilitas berisi metode CariPangkat yang menghitung pangkat suatu bilangan menggunakan loop. Dari source kode, logika dasarnya adalah mengalikan bilangan dasar (base) sebanyak exponent kali. Misalnya, CariPangkat(2, 3) menghasilkan 8. Metode dirancang efisien untuk operasi matematika sederhana dan menjadi inti dari fungsi perhitungan aplikasi.



```
1 namespace Modul12_Lib;
2
3 public class PangkatHelper
4 {
5     public static int CariNilaiPangkat(int a, int b)
6     {
7         if (b == 0) return 1;
8         if (a < 0 || b < 0) return -1;
9         if (b > 10 || a > 100) return -2;
10
11         int result = 1;
12         try
13         {
14             checked
15             {
16                 for (int i = 0; i < b; i++)
17                 {
18                     result *= a;
19                 }
20             }
21         }
22         catch (OverflowException)
23         {
24             return -3;
25         }
26
27         return result;
28     }
29 }
30
```

Jurnal/Modul12_2311104054/Modul12_Tests/Perhitungan.cs

Berisi kelas yang mengorganisasi logika bisnis tambahan, seperti validasi input atau operasi matematika kompleks. Struktur kelas dengan properti atau metode pendukung yang terhubung ke GUI atau PangkatHelper, memastikan pemisahan tanggung jawab antara UI dan logika.

```

1 namespace Modul12
2 {
3     public class Perhitungan
4     {
5         public static int CariNilaiPangkat(int angka, int pangkat)
6         {
7             return (int)Math.Pow(angka, pangkat);
8         }
9     }
10 }
11

```

Jurnal/Modul12_2311104054/Modul12_Tests/PerhitunganTests.cs

Unit test menggunakan framework seperti xUnit/NUnit untuk menguji metode dalam Perhitungan. Menunjukkan skenario pengujian seperti TestHitungPangkat_Positif atau TestHitungPangkat_Nol, yang memverifikasi hasil perhitungan sesuai ekspektasi. Assertion seperti Assert.Equal(8, hasil) digunakan untuk memvalidasi kebenaran logika.

```

1 using NUnit.Framework;
2 using Modul12_Lib;
3
4 namespace Modul12_Tests
5 {
6     public class PerhitunganTests
7     {
8         [Test]
9         public void TestPenjumlahan()
10        {
11            int hasil = Perhitungan.Tambah(2, 3);
12            Assert.That(hasil, Is.EqualTo(5));
13        }
14
15        [Test]
16        public void TestPengurangan()
17        {
18            int hasil = Perhitungan.Kurang(10, 4);
19            Assert.That(hasil, Is.EqualTo(6));
20        }
21
22        [Test]
23        public void TestPerkalian()
24        {
25            int hasil = Perhitungan.Kali(3, 4);
26            Assert.That(hasil, Is.EqualTo(12));
27        }
28    }
29 }
30

```

Jurnal/Modul12_2311104054/Modul12_Tests/CariPangkatTests.cs

Fokus pada pengujian metode CariPangkat di PangkatHelper. Source kode mencakup kasus uji untuk bilangan positif, negatif, dan eksponen nol (misal: CariPangkat(5, 0) harus mengembalikan 1). Pengujian ini memastikan tidak ada bug dalam perhitungan, termasuk penanganan edge case seperti input tidak valid.

```
1  using NUnit.Framework;
2  using Modul12_Lib;
3
4  namespace Modul12_Tests
5  {
6      public class CariNilaiPangkatTests
7      {
8          [Test]
9          public void TestPangkatPositif()
10         {
11             double hasil = Program.CariNilaiPangkat(2, 3);
12             Assert.That(hasil, Is.EqualTo(8));
13         }
14
15         [Test]
16         public void TestPangkatNol()
17         {
18             double hasil = Program.CariNilaiPangkat(5, 0);
19             Assert.That(hasil, Is.EqualTo(1));
20         }
21
22         [Test]
23         public void TestPangkatNegatif()
24         {
25             double hasil = Program.CariNilaiPangkat(2, -2);
26             Assert.That(hasil, Is.EqualTo(0.25));
27         }
28     }
29 }
30
```