

LAPORAN PRAKTIKUM TP MODUL 12

NIM / Nama: 2311104072 – Jauhar Fajar Zuhair

Bagian I: Pendahuluan

Tugas Pendahuluan Modul 12 ini mempraktikkan konsep Unit Testing pada aplikasi Windows Forms sederhana menggunakan C# dan framework NUnit. Tujuannya adalah membuat sebuah fungsi dalam aplikasi WinForms yang menentukan tanda sebuah bilangan (positif, negatif, atau nol) dan kemudian membuat serangkaian unit test untuk memastikan fungsi tersebut bekerja sesuai harapan dalam berbagai skenario.

Bagian II: Aplikasi Windows Forms (Form1.cs)

Aplikasi utama adalah sebuah Windows Form yang memiliki fungsi untuk menganalisis input numerik.

Fungsi yang Diuji: CariTandaBilangan

Meskipun kode lengkap `Form1.cs` tidak disertakan secara detail di file yang diunggah selain atribut `InternalsVisibleTo`, kita asumsikan (berdasarkan konteks TP dan Unit Test) bahwa terdapat sebuah metode, kemungkinan bernama `CariTandaBilangan`, di dalam kelas `Form1`. Metode ini menjadi target dari unit testing.

```
public string CariTandaBilangan(int number)
{
    if (number > 0)
    {
        return "Positif";
    }
    else if (number < 0)
    {
        return "Negatif";
    }
    else // number == 0
    {
        return "Nol";
    }
}
```

Penjelasan:

- Metode `CariTandaBilangan` menerima sebuah argumen integer (`number`).

- Metode ini mengembalikan string "Positif" jika angka lebih besar dari 0, "Negatif" jika angka kurang dari 0, dan "No!" jika angka sama dengan 0.
 - Metode ini kemungkinan dipanggil oleh event handler (misalnya, saat tombol ditekan) di dalam Form, mengambil input dari `textBoxInput` dan menampilkan hasilnya di `labelHasil`.
-

Bagian III: Unit Tests (`UnitTest1.cs`)

File ini berisi test case yang ditulis menggunakan NUnit untuk memverifikasi fungsionalitas metode `CariTandaBilangan`.

Struktur Test Class:

```
using NUnit.Framework;
using tpmodul12_2311104072; // Namespace dari proyek utama

namespace TestProject_TP
{
    [TestFixture] // Menandakan kelas ini berisi unit test
    public class FindNumberSignTests
    {
        private Form1 _form; // Instance dari Form yang akan diuji

        [SetUp] // Metode ini dijalankan sebelum setiap test case
        public void Setup()
        {
            // Membuat instance baru Form1 untuk setiap test
            _form = new Form1();
        }

        // ... Test Cases ...
    }
}
```

Penjelasan:

- `[TestFixture]` : Menandakan bahwa `FindNumberSignTests` adalah kelas yang berisi test case NUnit.
- `_form` : Variabel privat untuk menampung instance dari `Form1`.
- `[SetUp]` : Metode `Setup` dieksekusi sebelum setiap metode test (`[Test]`) dijalankan. Ini memastikan setiap test berjalan dengan kondisi awal yang bersih (instance `Form1` yang baru).

Test Cases:

1. Test Bilangan Positif:

```
[Test] // Menandakan ini adalah sebuah test case
public void TestPositiveNumber()
{
    // Arrange: Menyiapkan input
    int inputNumber = 10;
    string expected = "Positif";

    // Act: Memanggil metode yang diuji
    string actual = _form.CariTandaBilangan(inputNumber);

    // Assert: Memverifikasi hasil sesuai ekspektasi
    Assert.AreEqual(expected, actual, "Test gagal untuk bilangan positif.");
}
```

Penjelasan:

- **Arrange:** Menyiapkan input 10 dan hasil yang diharapkan "Positif".
- **Act:** Memanggil metode `_form.CariTandaBilangan` dengan input 10.
- **Assert:** Menggunakan `Assert.AreEqual` untuk membandingkan hasil aktual (`actual`) dengan hasil yang diharapkan (`expected`). Test akan gagal jika keduanya tidak sama.

2. Test Bilangan Negatif:

```
[Test]
public void TestNegativeNumber()
{
    // Arrange
    int inputNumber = -5;
    string expected = "Negatif";

    // Act
    string actual = _form.CariTandaBilangan(inputNumber);

    // Assert
    Assert.AreEqual(expected, actual, "Test gagal untuk bilangan negatif.");
}
```

Penjelasan:

- Mirip dengan test positif, namun menggunakan input 5 dan ekspektasi "Negatif".

3. Test Bilangan Nol:

```
[Test]
public void TestZero()
{
```

```

// Arrange
int inputNumber = 0;
string expected = "No1";

// Act
string actual = _form.CariTandaBilangan(inputNumber);

// Assert
Assert.AreEqual(expected, actual, "Test gagal untuk bilangan nol.");
}

```

Penjelasan:

- Menggunakan input 0 dan ekspektasi "No1".

Bagian IV: Menjalankan Test dan Hasil

