

Laporan Tugas Pendahuluan Modul 12

1. From1.cs

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

namespace tpmodul12_2311104064
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        public string CariTandaBilangan(int angkaInput)
        {
            if (angkaInput < 0)
            {
                return "Negatif";
            }
            else if (angkaInput > 0)
            {
                return "Positif";
            }
            else
            {
                return "No!";
            }
        }

        private void cekButton_Click(object sender, EventArgs e)
        {
            try
            {
                string teksDariInput = inputTextBox.Text;

                int angkaUntukDicek =
Convert.ToInt32(teksDariInput);

                string hasilPengecekan =
CariTandaBilangan(angkaUntukDicek);

                outputLabel.Text = hasilPengecekan;
            }
            catch (FormatException)
            {
                outputLabel.Text = "Error: Input harus berupa
angka yang valid!";
            }
            catch (OverflowException)
            {
                outputLabel.Text = "Error: Angka terlalu besar
atau kecil!";
            }
            catch (Exception ex)
            {
            }
        }
    }
}
```

```
        outputLabel.Text = "Error tidak diketahui: " +  
ex.Message;  
    }  
}  
}
```

Penjelasan:

Kode C# ini adalah untuk sebuah aplikasi Windows Forms sederhana yang dibuat dengan Visual Studio. Aplikasi ini memungkinkan pengguna memasukkan angka, dan kemudian program akan menentukan apakah angka tersebut **positif, negatif, atau nol**.

Struktur dan Fungsi Kode

1. Namespace dan Kelas Utama

- Kode ini berada dalam *namespace* `tpmodul12_2311104064`.
- Kelas utamanya adalah `Form1`, yang merupakan turunan dari `System.Windows.Forms.Form`. Ini berarti `Form1` adalah representasi dari jendela aplikasi itu sendiri.
- Kata kunci `partial` menunjukkan bahwa definisi kelas `Form1` mungkin dibagi menjadi beberapa file (biasanya `Form1.cs` untuk logika Anda dan `Form1.Designer.cs` untuk kode yang dibuat otomatis oleh desainer Visual Studio).

2. Konstruktor `Form1()`

- `public Form1()` adalah konstruktor kelas. Metode ini dijalankan secara otomatis ketika jendela `Form1` dibuat.
- `InitializeComponent();` adalah panggilan metode yang sangat penting. Metode ini (yang sebenarnya ada di file `Form1.Designer.cs`) bertanggung jawab untuk membuat dan mengatur semua komponen antarmuka pengguna (GUI) yang telah Anda letakkan di form menggunakan desainer Visual Studio (seperti `TextBox`, `Button`, dan `Label`).

3. Metode `CariTandaBilangan(int angkaInput)`

- Ini adalah fungsi publik yang Anda buat.
- **Tujuan:** Menerima satu argumen berupa bilangan bulat (`angkaInput`) dan mengembalikan sebuah string yang menandakan apakah angka tersebut "Negatif", "Positif", atau "Nol".
- **Logika:**
 - Jika `angkaInput < 0`, ia mengembalikan "Negatif".
 - Jika `angkaInput > 0`, ia mengembalikan "Positif".

4. Metode Event Handler cekButton_Click(object sender, EventArgs e)

- Ini adalah metode privat yang akan dijalankan **ketika tombol bernama cekButton diklik** oleh pengguna. Anda biasanya menghubungkan metode ini ke event Click tombol melalui desainer Visual Studio (misalnya dengan melakukan *double-click* pada tombol di desainer).

- **Logika:**

1. **Mengambil Input:** string teksDariInput = inputTextBox.Text; mengambil teks yang dimasukkan pengguna ke dalam komponen TextBox yang bernama inputTextBox.
2. **Konversi ke Angka:** int angkaUntukDicek = Convert.ToInt32(teksDariInput); mencoba mengubah teks input menjadi bilangan bulat.
3. **Memanggil Fungsi Logika:** string hasilPengecekan = CariTandaBilangan(angkaUntukDicek); memanggil metode CariTandaBilangan dengan angka yang sudah dikonversi untuk mendapatkan hasilnya.
4. **Menampilkan Hasil:** outputLabel.Text = hasilPengecekan; menampilkan string hasil ("Negatif", "Positif", atau "Nol") ke dalam komponen Label yang bernama outputLabel.
5. **Penanganan Kesalahan (Error Handling) dengan try-catch:**
 - try { ... }: Blok kode utama yang mungkin menimbulkan kesalahan dicoba untuk dijalankan.
 - catch (FormatException) { ... }: Jika pengguna memasukkan teks yang tidak bisa diubah menjadi angka (misalnya, huruf atau simbol), kesalahan FormatException akan ditangkap, dan outputLabel akan menampilkan pesan "Error: Input harus berupa angka yang valid!".
 - catch (OverflowException) { ... }: Jika angka yang dimasukkan terlalu besar atau terlalu kecil untuk dapat disimpan dalam tipe data int, kesalahan OverflowException akan ditangkap, dan outputLabel akan menampilkan pesan "Error: Angka terlalu besar atau kecil!".
 - catch (Exception ex) { ... }: Ini adalah blok penangkap kesalahan umum yang akan menangani jenis kesalahan lain yang mungkin tidak terduga. Ia akan menampilkan pesan "Error tidak diketahui: " diikuti dengan pesan kesalahan dari sistem.

2. Hasil Run

The image displays four sequential screenshots of a Java Swing window titled "Form1". Each screenshot shows the state of the application after a user input is processed by a "Cek Tanda" button.

- First Screenshot:** The input field contains "-10". After clicking "Cek Tanda", the output label displays "Negatif".
- Second Screenshot:** The input field contains "10". After clicking "Cek Tanda", the output label displays "Positif".
- Third Screenshot:** The input field contains "0". After clicking "Cek Tanda", the output label displays "Nol".
- Fourth Screenshot:** The input field contains "a". After clicking "Cek Tanda", the output label displays an error message: "Error: Input harus berupa angka yang valid!".