

Putra Strata Tandika Setyawan

2311104050

S1SE0702

Jurnal Modul 13

1. Class yang sudah di refactor dengan standar.NET

```
using System;

/// <summary>
/// Mewakili video YouTube dengan kemampuan pelacakan jumlah putar.
/// </summary>
public class SayaTubeVideo
{
    private readonly int _id;
    private readonly string _judul;
    private int _jumlahPutar;

    /// <summary>
    /// Menginisialisasi instance baru dari kelas SayaTubeVideo.
    /// </summary>
    /// <param name="judul">Judul video (maksimal 100 karakter, tidak boleh
    null atau kosong).</param>
    /// <exception cref="ArgumentException">Dilempar ketika judul tidak
    valid.</exception>
    public SayaTubeVideo(string judul)
    {
        // Pra-kondisi: judul tidak boleh null/kosong dan harus ≤ 100
        karakter
        if (string.IsNullOrEmpty(judul))
        {
            throw new ArgumentException("Judul video tidak boleh null atau
            kosong.", nameof(judul));
        }

        if (judul.Length > 100)
        {
            throw new ArgumentException("Judul video tidak boleh lebih dari
            100 karakter.", nameof(judul));
        }

        var acak = new Random();
        _id = acak.Next(10000, 99999);
        _judul = judul;
        _jumlahPutar = 0;
    }

    /// <summary>
    /// Menambah jumlah putar dengan jumlah tertentu.
    /// </summary>
    /// <param name="jumlah">Jumlah penambahan (maksimal 10.000.000 per
    panggilan).</param>
    /// <exception cref="ArgumentOutOfRangeException">Dilempar ketika
    jumlah melebihi batas.</exception>
    public void TambahJumlahPutar(int jumlah)
```

```

{
    // Pra-kondisi: jumlah harus ≤ 10.000.000
    if (jumlah > 10000000)
    {
        throw new ArgumentOutOfRangeException(nameof(jumlah),
            "Jumlah penambahan tidak boleh lebih dari 10.000.000 per
panggilan.");
    }

    try
    {
        checked
        {
            _jumlahPutar += jumlah;
        }
    }
    catch (OverflowException ex)
    {
        Console.WriteLine($"Terjadi overflow pada jumlah putar:
{ex.Message}");
        // Alternatif: bisa diatur ke nilai maksimum int
    }
}

/// <summary>
/// Menampilkan detail video ke konsol.
/// </summary>
public void CetakDetailVideo()
{
    Console.WriteLine($"Detail Video:");
    Console.WriteLine($"- ID: {_id}");
    Console.WriteLine($"- Judul: {_judul}");
    Console.WriteLine($"- Jumlah Putar: {_jumlahPutar:N0}");
}
}

```

Perbaikan dalam Refactoring:

#### A. Konvensi Penamaan

1. **Field:** Diubah menjadi `_camelCase` untuk field private
  - Sebelum: `id`, `title`, `playCount`
  - Sesudah: `_id`, `_judul`, `_jumlahPutar`
2. **Method:** Tetap PascalCase dengan nama lebih deskriptif
  - `CetakDetailVideo()` lebih eksplisit dibanding `Print()`

#### B. Spasi dan Indentasi

1. Menambahkan spasi konsisten:

- Sekitar operator (=, +=)
  - Setelah koma dalam parameter
  - Antara blok kode logis
2. Memperbaiki indentasi pada:
    - Isi method
    - Daftar parameter
    - Blok penanganan exception

#### C. Deklarasi Variabel/Atribut

1. Membuat `_id` dan `_judul` `readonly` karena tidak berubah setelah konstruksi
2. Menggunakan `var` untuk variabel lokal (`acak`) ketika tipe jelas
3. Mengelompokkan deklarasi field yang terkait

#### D. Komentar

1. Menambahkan dokumentasi XML untuk:
  - Kelas
  - Semua method public
  - Parameter
  - Exception yang dilempar
2. Menambahkan komentar jelas untuk:
  - Pengecekan pra-kondisi
  - Bagian logika penting
3. Memperbaiki pesan error dengan detail lebih jelas

2. Program.cs yang sudah di refactor dengan standar.NET

```
using System;

/// <summary>
/// Program utama untuk demonstrasi kelas SayaTubeVideo
/// </summary>
public class Program
{
    /// <summary>
    /// Entry point aplikasi
    /// </summary>
    /// <param name="args">Argumen command line</param>
    public static void Main(string[] args)
    {
        TampilkanHeaderProgram();

        try
        {
            JalankanDemoVideo();
        }
        catch (Exception ex)
        {
            TanganiError(ex);
        }
        finally
        {
            TampilkanPesanAkhir();
        }
    }

    private static void TampilkanHeaderProgram()
    {
        Console.WriteLine("==== Program Demonstrasi SayaTubeVideo ====");
        Console.WriteLine();
    }

    private static void JalankanDemoVideo()
    {
        var video = BuatVideoContoh();

        Console.Write("Ingin menampilkan simulasi overflow? (y/n): ");
        var input = Console.ReadLine()?.ToLower();

        if (input == "y")
        {
            JalankanSimulasiOverflow(video);
        }
        else
        {
            JalankanDemoNormal(video);
        }
    }

    private static SayaTubeVideo BuatVideoContoh()
    {
        const string judulVideo = "Tutorial Design By Contract - Putra  
Strata Tandika Setyawan-2311104050";
    }
}
```

```

        return new SayaTubeVideo(judulVideo);
    }

    private static void JalankanSimulasiOverflow(SayaTubeVideo video)
    {
        Console.WriteLine("\n=== Simulasi Overflow ===");

        try
        {
            for (int i = 0; i < 100000; i++)
            {
                video.IncreasePlayCount(1000000);
            }
        }
        catch (OverflowException ex)
        {
            Console.WriteLine($"Terjadi overflow: {ex.Message}");
        }
    }

    private static void JalankanDemoNormal(SayaTubeVideo video)
    {
        try
        {
            video.IncreasePlayCount(5000);
            video.PrintVideoDetails();
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error saat menambah play count: {ex.Message}");
        }
    }

    private static void TanganiError(Exception ex)
    {
        Console.WriteLine($"Error yang tidak terduga: {ex.Message}");
        Console.WriteLine($"Stack Trace: {ex.StackTrace}");
    }

    private static void TampilkanPesanAkhir()
    {
        Console.WriteLine("\nProgram selesai dijalankan.");
    }
}

```

Perbaikan yang Dilakukan:

#### A. Konvensi Penamaan

##### 1. Method:

- Semua method menggunakan PascalCase
- Nama lebih deskriptif (contoh: `JalankanDemoVideo` bukan `Main`)

## 2. **Variabel:**

- Menggunakan camelCase (judulVideo, input)
- Menggunakan `var` untuk tipe yang jelas

### B. Spasi dan Indentasi

#### 1. Konsisten menggunakan:

- 4 spasi untuk indentasi
- Spasi setelah `if`, `for`, `try`
- Baris kosong antar method

#### 2. Formatting:

- Parameter sejajar
- Blok kode konsisten

### C. Deklarasi Variabel

1. Menggunakan `const` untuk string literal
2. Mengelompokkan deklarasi terkait
3. Menggunakan `var` ketika tipe jelas

### D. Komentar

#### 1. XML Documentation untuk:

- Kelas Program
- Method utama
- Parameter

#### 2. Komentar menjelaskan:

- Alur program
- Penanganan error
- Logika penting

### E. Perbaikan Struktur


1. Memecah method panjang menjadi sub-method

2. Error handling lebih spesifik
3. Logika bisnis terpisah dari UI
4. Menggunakan finally block untuk cleanup

#### F. Tambahan Perbaikan


1. Null check dengan operator ?
2. Penanganan error lebih informatif
3. Penggunaan konstanta untuk magic numbers
4. Alur program lebih terstruktur

### 3. Hasil Outputnya

 D:\Setya\Clone\KPL\_Putra-Strata-Tandika-Setyawan\_2311104050\14\_Clean\_Code\TP\tpmodul14\_2311104050\tpmodul

```
==== Program SayaTubeVideo ====
Ingin menampilkan simulasi overflow? (y/n):
```

Jika pilih y

 D:\Setya\Clone\KPL\_Putra-Strata-Tandika-Setyawan\_2311104050\14\_Clean\_Code\TP\tpmodul14\_2311104050\tpmodul6\_2311104050\bin\Debug\net

```
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
Terjadi overflow: Arithmetic operation resulted in an overflow.
```

Jika pilih n

```
==== Program SayaTubeVideo ====
Ingin menampilkan simulasi overflow? (y/n): n
PID: 28408
Title: Tutorial Design By Contract - Putra Strata Tandika Setyawan-2311104050
Play Count: 5000
Program selesai.
D:\Setya\Clone\KPL_Putra-Strata-Tandika-Setyawan_2311104050\14_Clean_Code\TP\tpmodul14_2311104050\tpmodul6_2311104050\bin\Debug\net8.0\tpmodul6_2311104050.exe (process 10452) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```