



**TUGAS PENDAHULUAN**  
**KONSTRUKSI PERANGKAT LUNAK**  
**MODUL 14**

**D**  
**I**  
**S**  
**U**  
**S**  
**U**  
**N**

**OLEH :**  
**TIURMA GRACE ANGELINA – 2311104042**  
**S1-SE07-02**

**Dosen :**  
**YIS**

**Prodi S1 Rekaya Perangkat Lunak**  
**Direktorat Kampus Purwokerto**

## 1. REFACTORING DENGAN STANDAR CODE

TPMOD6

Class SayaTubeVideo.cs

```
using System;

public class SayaTubeVideo
{
    private int id;
    private string title;
    private int playCount;

    public SayaTubeVideo(string title)
    {
        if (string.IsNullOrEmpty(title))
            throw new ArgumentException("Judul video tidak boleh kosong atau null.");

        if (title.Length > 100)
            throw new ArgumentException("Judul video tidak boleh lebih dari 100karakter.");

        Random random = new Random();
        this.id = random.Next(10000, 99999);
        this.title = title;
        this.playCount = 0;
    }

    public void IncreasePlayCount(int count)
    {
        if (count > 10000000)
            throw new ArgumentException("Penambahan play count tidak boleh lebih dari 10.000.000.");

        try
        {
            checked
            {
                this.playCount += count;
            }
        }
        catch (OverflowException)
        {
            Console.WriteLine("Error: Play count melebihi batas maksimum integer!");
        }
    }
}
```

```

    }

    public void PrintVideoDetails()
    {
        Console.WriteLine($"ID Video: {id}");
        Console.WriteLine($"Judul: {title}");
        Console.WriteLine($"Jumlah Play: {playCount}");
    }
}

```

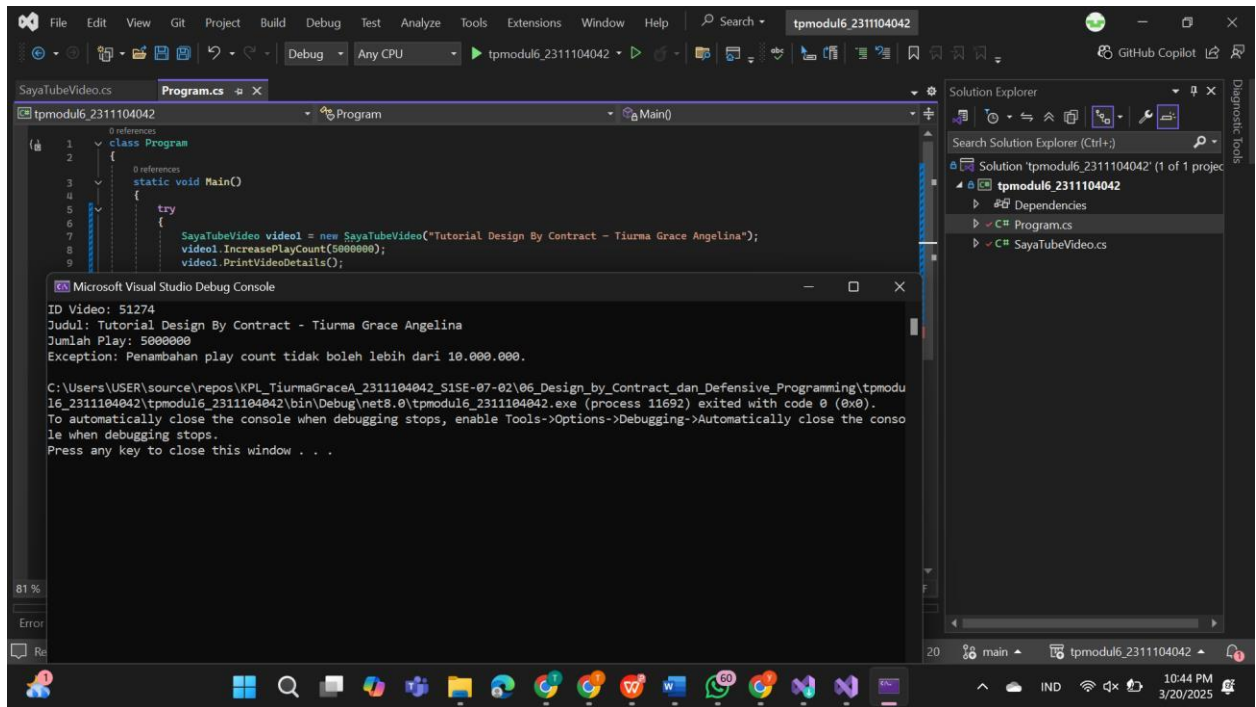
Class program.cs

```

class Program
{
    static void Main()
    {
        try
        {
            SayaTubeVideo video1 = new SayaTubeVideo("Tutorial Design By Contract –  
Tiurma Grace Angelina");
            video1.IncreasePlayCount(5000000);
            video1.PrintVideoDetails();

            SayaTubeVideo video2 = new SayaTubeVideo("Test Overflow");
            for (int i = 0; i < 10; i++)
            {
                video2.IncreasePlayCount(int.MaxValue);
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Exception: {ex.Message}");
        }
    }
}

```



## Penjelasan Implementasi Kode: Sebelum & Sesudah Refactoring

### 1. Sebelum Refactoring (TP Modul 6)

Kode yang dibuat pada TP Modul 6 masih fungsional, namun **belum sepenuhnya mengikuti standar penulisan kode C# (.NET)**. Berikut beberapa poin permasalahan pada versi awal:

#### 1. Naming Convention:

- Nama variabel id, title, dan playCount terlalu umum dan tidak deskriptif.
- Nama method PrintVideoDetails() cukup baik, namun bisa lebih ringkas.

#### 2. Indentasi dan Spasi Putih:

- Tidak semua method diberi spasi pemisah, membuat kode terlihat padat dan kurang terbaca.

#### 3. Komentar:

- Tidak ada komentar sama sekali, padahal penting untuk menjelaskan tujuan method atau logika tertentu.

#### 4. Deklarasi Variabel:

- Nama variabel tidak menjelaskan konteksnya, misalnya id sebaiknya menjadi videoid.

Contoh potongan kode:

```
private int id;

private string title;

public void PrintVideoDetails()
{
    Console.WriteLine($"ID Video: {id}");
    Console.WriteLine($"Judul: {title}");
    Console.WriteLine($"Jumlah Play: {playCount}");
}
```

## 2. Setelah Refactoring (TP Modul 14 - Clean Code)

Refactoring Clean Code – Penjelasan Detail

### A. Naming Convention

#### i. Variable / Property / Attribute

Standar yang digunakan\*\*:

- Gunakan `camelCase` untuk variabel lokal.
- Gunakan `PascalCase` untuk property dan field (jika public), sesuai standar C#.

Sebelum	Sesudah	Penjelasan	
`id`	`videoid`	Nama `id` terlalu umum. `videoid` lebih jelas konteksnya.	
`title`	`videoTitle`	Nama `title` diubah agar menjelaskan konteks atribut.	
`playCount`	`playCount`	Sudah sesuai (karena ini private dan cukup jelas).	

#### ii. Method / Function / Procedure

Standar yang digunakan:

- Gunakan `PascalCase` (huruf kapital di awal setiap kata).
- Nama harus menjelaskan fungsi/metode tersebut secara ringkas & jelas.

Sebelum	Sesudah	Penjelasan
<code>`PrintVideoDetails()`</code>	<code>`PrintDetails()`</code>	Disingkat agar ringkas, tapi tetap jelas fungsinya.
<code>`IncreasePlayCount()`</code>	<code>`IncreasePlayCount()`</code>	Sudah sesuai standar dan menjelaskan maksudnya.

## B. White Space dan Indentation

Standar yang digunakan:

- Gunakan indentasi 4 spasi (default Visual Studio).
- Beri 1 baris kosong antar method, antar block logika, dan setelah deklarasi constructor agar kode tidak bertumpuk.

Sebelum	Sesudah
Method menumpuk satu sama lain	Setiap method dipisah oleh 1 baris kosong
Tidak ada baris kosong di dalam blok kode	Ada baris kosong antar bagian logika penting

Contoh:

SEBELUM

```
public SayaTubeVideo(string title)
{ if (string.IsNullOrEmpty(title)) throw ...; Random r = new Random(); ... }
```

// SESUDAH

```
public SayaTubeVideo(string title)
{
    if (string.IsNullOrEmpty(title))
        throw new ArgumentException("...");

    Random random = new Random();
    this.videoid = random.Next(10000, 99999);
    ...
}
```

### C. Variable / Attribute Declarations

Standar yang digunakan:

- Nama harus deskriptif, bukan singkatan umum.
- Hindari nama seperti `a`, `x`, `data`, `temp` tanpa konteks.

Sebelum	Sesudah	Penjelasan	
`id`	`videoId`	Menjelaskan bahwa itu ID video, bukan ID lainnya.	
`title`	`videoTitle`	Menjelaskan ini adalah title milik video.	
`count`	`count`	Masih boleh karena konteksnya jelas di parameter method.	

### D. Comments (Komentar)

Standar yang digunakan:

- Tambahkan komentar di atas method untuk menjelaskan fungsinya.
- Tambahkan komentar pendek untuk bagian logika yang penting atau rawan error.

Contoh:

```
// Constructor untuk inisialisasi objek video
```

```
public SayaTubeVideo(string title)
```

```
{
```

```
    ...
```

```
}
```

```
// Method untuk menambahkan jumlah play dengan validasi
```

```
public void IncreasePlayCount(int count)
```

```
{
```

```
    // Validasi batas penambahan
```

```
    if (count > 100000000)
```

```
        throw new ArgumentException("...");
```

```
    // Cegah integer overflow
```

```
    try
```

```
    {
```

```
        checked
```

```
        {
```

```
            this.playCount += count;
```

```
        }
```

```

    }
    catch (OverflowException)
    {
        Console.WriteLine("Error: overflow play count!");
    }
}
...

```

```

using System;

public class SayaTubeVideo
{
    // Atribut private dengan nama yang lebih deskriptif
    private int videoId;
    private string videoTitle;
    private int playCount;

    // Constructor untuk inisialisasi objek video
    public SayaTubeVideo(string title)
    {
        // Validasi: judul tidak null/kosong dan <= 100 karakter
        if (string.IsNullOrEmpty(title))
            throw new ArgumentException("Judul video tidak boleh kosong atau null.");

        if (title.Length > 100)
            throw new ArgumentException("Judul video tidak boleh lebih dari 100 karakter.");

        Random random = new Random();
        this.videoId = random.Next(10000, 99999);
        this.videoTitle = title;
        this.playCount = 0;
    }

    // Method untuk menambahkan jumlah play
    public void IncreasePlayCount(int count)
    {
        // Validasi: penambahan maksimal 10 juta
        if (count > 10000000)
            throw new ArgumentException("Penambahan play count tidak boleh lebih dari 10.000.000.");
    }
}

```



```

        try
        {
            checked
            {
                this.playCount += count;
            }
        }
        catch (OverflowException)
        {
            Console.WriteLine("Error: Play count melebihi batas maksimum
integer!");
        }
    }

    // Method untuk menampilkan detail video
    public void PrintDetails()
    {
        Console.WriteLine($"ID Video   : {videoId}");
        Console.WriteLine($"Judul     : {videoTitle}");
        Console.WriteLine($"Jumlah Play : {playCount}");
    }
}

```

```

using System;

class Program
{
    static void Main()
    {
        try
        {
            SayaTubeVideo video1 = new SayaTubeVideo("Tutorial Design By
Contract – Tiurma Grace Angelina");
            video1.IncreasePlayCount(5000000);
            video1.PrintDetails();

            Console.WriteLine();

            SayaTubeVideo video2 = new SayaTubeVideo("Test Overflow");

            // Loop untuk menguji overflow integer

```

```

        for (int i = 0; i < 10; i++)
        {
            video2.IncreasePlayCount(int.MaxValue);
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Exception: {ex.Message}");
    }
}
}

```

Output :

