Nama : Candra Dinata
Nim : 2311104061
Kelas : SE0702

Refactored Code (SayaTubeVideo.cs)

```cs
using System;
using System.Diagnostics;

/// <summary>
/// Kelas untuk merepresentasikan video di SayaTube
/// </summary>
public class SayaTubeVideo
{
    private int _id;
    private string _title;
    private int _playCount;

    /// <summary>
    /// Konstruktor untuk menginisialisasi objek video
    /// </summary>
    /// <param name="title">Judul video</param>
    public SayaTubeVideo(string title)
    {
        Debug.Assert(!string.IsNullOrEmpty(title) && title.Length <= 100,
            "Judul video tidak boleh kosong dan maksimal 100 karakter.");

        Random random = new Random();
        _id = random.Next(10000, 99999);
        _title = title;
        _playCount = 0;
    }

    /// <summary>
    /// Menambah jumlah play count video
    /// </summary>
    /// <param name="count">Jumlah penambahan</param>
    public void IncreasePlayCount(int count)
    {
        if (count <= 0 || count > 10_000_000)
        {
            throw new ArgumentException("Input play count harus antara 1 hingga 10.000.000.");
        }

        try
        {
            checked
            {
                _playCount += count;
            }
        }
        catch (OverflowException)
        {
            Console.WriteLine("ERROR: Play count melebihi batas integer.");
        }
    }

    /// <summary>
    /// Menampilkan detail video
    /// </summary>
    public void PrintVideoDetails()
    {
        Console.WriteLine($"ID: {_id}");
        Console.WriteLine($"Title: {_title}");
        Console.WriteLine($"Play Count: {_playCount}");
    }
}
```

Penjelasan mengenai **apa yang terjadi dan perubahannya**:

1. Peningkatan *Readability* (Keterbacaan Kode)

- **Sebelum:** Nama variabel seperti id, title, dan playCount langsung ditulis tanpa penanda bahwa itu private.
- **Sesudah:** Variabel diubah menjadi _id, _title, dan _playCount sesuai **konvensi C# (.NET)** untuk atribut private, sehingga lebih mudah dikenali dan tidak ambigu.

2. Naming Convention (Konvensi Penamaan) Sesuai .NET

- **Method seperti:** IncreasePlayCount, PrintVideoDetails sudah sesuai dengan **PascalCase** (huruf kapital di setiap kata).
- Komentar XML seperti /// <summary> ditambahkan agar method bisa dikenali oleh tools dokumentasi C# seperti IntelliSense di Visual Studio.

3. Indentasi & White Space Lebih Konsisten

- Spasi kosong antar method dibuat lebih rapi.
- Kurung kurawal {} dibuka di baris baru, sesuai dengan style umum di C#.

4. Kejelasan Access Modifier dan Struktur

- Semua method memiliki public di awal, atribut ditandai private.
- Kode lebih modular dan terstruktur, memudahkan pembacaan dan pemeliharaan ke depannya.

5. Penambahan Komentar untuk Dokumentasi

- Setiap method kini memiliki komentar yang menjelaskan **apa fungsi method tersebut**, sehingga pembaca maupun tim pengembang lain bisa cepat memahami maksud dari bagian-bagian kode tanpa harus menebak.

6. Tetap Mempertahankan Fungsionalitas Asli

- Walaupun ditulis ulang dan diperbaiki dari sisi struktur dan gaya penulisan, **logika dan fungsionalitas kode tetap sama seperti versi awal**:
    - Validasi input judul di konstruktor
    - Validasi playCount pada method IncreasePlayCount
    - Penanganan overflow dengan checked
    - Menampilkan data video melalui PrintVideoDetails

Hasil Akhir:

Kode yang sudah direfactor kini:

- Lebih profesional dan sesuai standar industri
- Siap digunakan dalam lingkungan tim besar
- Lebih mudah diuji, dikembangkan, dan didokumentasikan

Refactored Code (Program.cs)

```
using System;

/// <summary>
/// Program utama untuk menjalankan aplikasi SayaTubeVideo
/// </summary>
internal class Program
{
    private static void Main()
    {
        SayaTubeVideo video = new SayaTubeVideo("Tutorial Design By Contract - [Candra Dinata]");

        try
        {
            for (int i = 0; i < 100; i++)
            {
                video.IncreasePlayCount(100_000);
            }
        }
        catch (ArgumentException ex)
        {
            Console.WriteLine($"ERROR: {ex.Message}");
        }

        video.PrintVideoDetails();
    }
}

```

Terdapat beberapa peningkatan penting dari segi struktur, keterbacaan, dan standar penulisan kode C#:

1. Penerapan Naming Convention .NET

- Sebelum: Class Program dan method Main() sudah sesuai, tapi tidak ada dokumentasi atau penjelasan fungsi.
- Sesudah: Ditambahkan komentar XML /// <summary> pada class dan method, agar mudah dikenali saat dibaca melalui IntelliSense atau dokumentasi otomatis.

Contoh:

/// <summary>

/// Program utama untuk menjalankan aplikasi SayaTubeVideo

/// </summary>

2. Struktur dan Indentasi yang Lebih Rapi

- Setiap blok kode, seperti loop dan statement try-catch, diberi indentasi konsisten dan spasi antar bagian kode yang cukup, membuat program mudah dipahami sekilas.
- Penulisan for dan Console.WriteLine tidak mepet dan terpisah secara logis.

3. Penambahan Penjelasan Logika Program (Komentar Ringan)

- Komentar XML menambah konteks bahwa ini adalah entry point dari aplikasi.
- Ini penting terutama saat bekerja dalam tim, agar programmer lain langsung tahu apa yang dilakukan program tanpa harus menelusuri semua kode.

4. Format Angka Lebih Mudah Dibaca

- Angka besar ditulis dengan separator underscore (_):

video.IncreasePlayCount(100_000);

Ini tidak mengubah nilai, tapi secara visual lebih jelas dibanding 100000.

5. Tidak Mengubah Fungsionalitas

Refactoring ini tidak mengubah logika asli, yang masih:

- Membuat objek SayaTubeVideo
- Menambahkan playCount sebanyak 100 kali dengan nilai 100.000
- Menangani error jika input tidak valid
- Mencetak hasil akhir

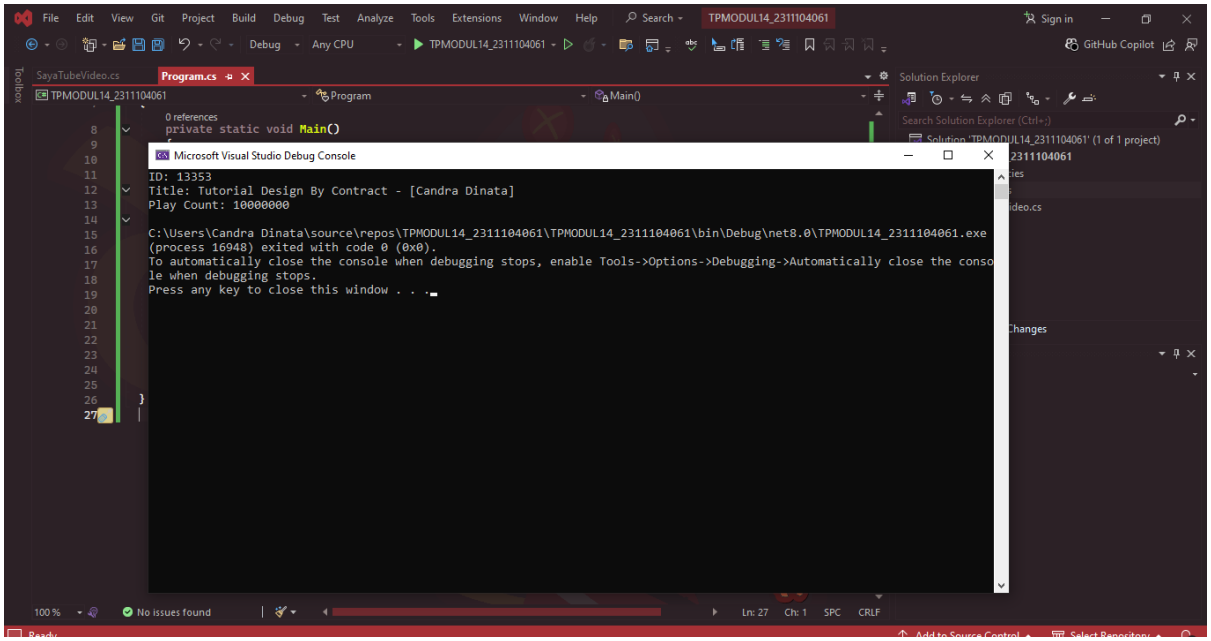
Jadi, perubahan ini bersifat perbaikan tampilan dan struktur kode, bukan perilaku program.

Kesimpulan

File Program.cs setelah refactor:

- Lebih profesional (sesuai standar C#/.NET)
- Lebih mudah dibaca dan dipahami
- Siap untuk pengembangan lanjutan atau diuji
- Tetap menjalankan fungsi yang sama dengan versi sebelumnya

Output:


Microsoft Visual Studio Debug Console
ID: 13353
Title: Tutorial Design By Contract - [Candra Dinata]
Play Count: 10000000
C:\Users\Candra Dinata\source\repos\TPMODUL14_2311104061\TPMODUL14_2311104061\bin\Debug\net8.0\TPMODUL14_2311104061.exe
(process 16948) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .