

Bagian I: Deklarasi dan Konstruktor Kelas YoutubeVideo

```
class YoutubeVideo
{
    private readonly int _id;
    private readonly string _title;
    public int PlayCount { get; set; }

    public YoutubeVideo(string title)
    {
        Contract.Requires(title.Length < 100 &&!string.IsNullOrEmpty(_title),
            "TIDAK ADA JUDUL FILM YANG LEBIH DARI 100 HURUF !");
        Random random = new Random();
        _id = random.Next(10000, 100000);
        _title = title;
        PlayCount = 0;
        Contract.Ensures(_title.Length < 100&&!string.IsNullOrEmpty(_title));
    }
    // ...
}
```

Penjelasan:

- **Field Deklarasi:** Menyimpan ID, judul, dan jumlah pemutaran (PlayCount).
 - **Konstruktor:**
 - Menggunakan Contract untuk memastikan judul tidak kosong dan kurang dari 100 karakter.
 - Menghasilkan ID secara acak dan menginisialisasi PlayCount ke nol.
 - Menggunakan Contract.Ensures untuk menjamin kondisi tertentu setelah konstruksi.
-

Bagian II: Invariant Kontrak

```
[ContractInvariantMethod]
protected void ObjectVariant()
{
    Contract.Invariant(PlayCount >= 0);
}
```

Penjelasan:

- **Contract.Invariant:** Menjamin bahwa nilai PlayCount selalu bernilai nol atau lebih, sehingga tidak pernah negatif.
-

Bagian III: Metode IncreasePlayCount

```
public void IncreasePlayCount(int time)
{
    Contract.Requires(time < 10000000, "tindakan bomleh lebih dari 10m perinput  
yh");
    try
    {
        checked
        {
            PlayCount += time;
        }
    }
    catch (OverflowException e)
    {
        Console.WriteLine($"sebaiknya jangan dibesarbesarkan -> {e}");
        throw;
    }
}
```

Penjelasan:

- **Validasi Input:** Menggunakan Contract.Requires untuk memastikan nilai time tidak melebihi batas tertentu.
 - **Penambahan Aman:** Menggunakan blok checked untuk mendeteksi overflow ketika menambahkan nilai ke PlayCount.
 - **Penanganan Error:** Jika terjadi overflow, ditangani dengan menampilkan pesan dan melempar exception kembali.
-

Bagian IV: Metode PrintVideoDetails

```
public void PrintVideoDetails()
{
    Console.WriteLine($"_id: {_id}, TITLE: {_title}, PLAY_COUNT:
{PlayCount}");
}
```

Penjelasan:

- Menampilkan detail video (ID, judul, dan jumlah pemutaran) ke konsol.
-

Bagian V: Main Program

```
class Program
{
    static void Main()
    {
        var youtubeVideo = new YoutubeVideo("Tutorial Design By Contract -
jauharfz");
        youtubeVideo.PrintVideoDetails();
        for (int i = 0; i < 1000; i++)
        {
            youtubeVideo.IncreasePlayCount(10000000 - 1);
            Console.WriteLine($"{i + 1}.UPDATE_PLAY_COUNT:
{youtubeVideo.PlayCount}");
        }
    }
}
```

Penjelasan:

- **Instansiasi:** Membuat objek YoutubeVideo dengan judul yang ditentukan.
 - **Menampilkan Detail Awal:** Memanggil PrintVideoDetails untuk menampilkan informasi awal.
 - **Loop Update:** Melakukan perulangan 1000 kali untuk menambahkan nilai ke PlayCount, kemudian menampilkan pembaruan setiap iterasi.
-