

TP MODUL 6

Design By Contract And Defensive Programming



Nama :

Dhiemas Tulus Ikhsan (2311104046)

Dosen :

Yudha Islami Sulistya, S.Kom., M.Cs

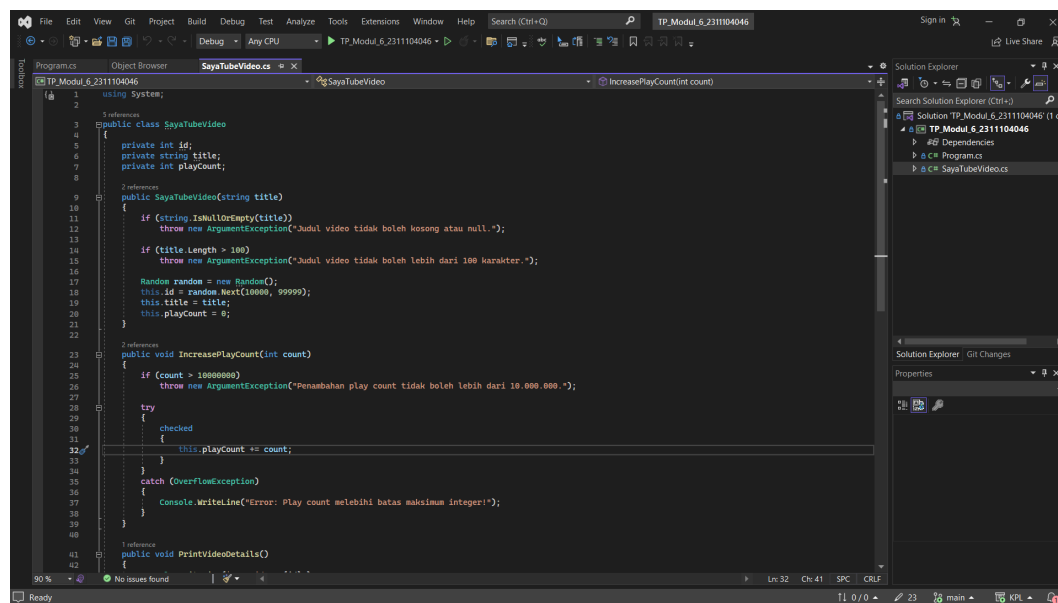
PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

TELKOM UNIVERSITY PURWOKERTO

2025

Penjelasan Program SayaTubeVideo: Design by Contract dan Defensive Programming

Program ini merupakan implementasi class `SayaTubeVideo` yang menerapkan prinsip Design by Contract (DbC) dan Defensive Programming untuk menjaga agar objek video memiliki perilaku yang valid, stabil, dan aman dari input atau pemrosesan yang tidak sesuai.



Class `SayaTubeVideo` menyimpan tiga informasi penting:

- `id`: nilai acak unik antara 10000 sampai 99999,
- `title`: judul video yang wajib diisi dan dibatasi panjangnya,
- `playCount`: penghitung jumlah pemutaran video.

Saat objek dibuat, kontrak awal (pre-condition) langsung ditegakkan dalam konstruktor:

```
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.");
```

Kode di atas merupakan bentuk defensive check agar data yang masuk sesuai ekspektasi: judul harus ada dan tidak terlalu panjang. Hal ini penting agar sistem tidak menerima input aneh yang bisa menimbulkan error di bagian lain program.

Kemudian, `id` dihasilkan secara acak:

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

Metode `IncreasePlayCount(int count)` digunakan untuk menambah jumlah play video. Agar valid, nilai yang ditambahkan tidak boleh lebih dari 10 juta:

```
if (count > 10000000)  
    throw new ArgumentException("Penambahan play count tidak boleh lebih dari  
10.000.000.");
```

Selain itu, pengecekan `checked` digunakan untuk menangkap *overflow* ketika jumlah play melebihi batas maksimum `int`. Jika itu terjadi, maka akan ditangani dengan `catch` untuk menghindari crash:

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

Method `PrintVideoDetails()` hanya menampilkan informasi dari video dengan format yang rapi ke konsol.

Pada bagian `Main()`, program menguji dua hal:

1. Pembuatan dan penambahan play count normal:

```
SayaTubeVideo video1 = new SayaTubeVideo("Tutorial Design By  
Contract – Dhiemas Tulus Ikhsan");  
video1.IncreasePlayCount(5000000);
```

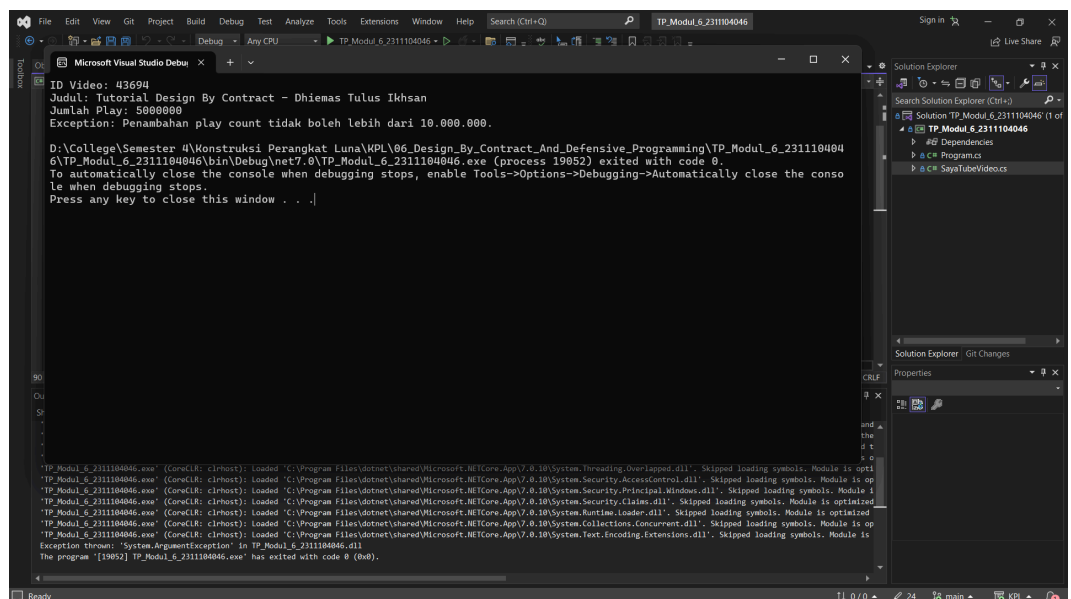
```
video1.PrintVideoDetails();
```

Ini akan berjalan lancar karena play count masih dalam batas aman.

2. Simulasi overflow:

```
SayaTubeVideo video2 = new SayaTubeVideo("Test Overflow");  
for (int i = 0; i < 10; i++)  
{  
    video2.IncreasePlayCount(int.MaxValue);  
}
```

Di bagian ini, program mencoba menambahkan nilai maksimum integer sebanyak 10 kali, yang jelas akan menimbulkan overflow. Namun, dengan pendekatan defensive, program tidak crash dan malah menangkap errornya, lalu mencetak pesan ke konsol.



Kesimpulan

Program ini menunjukkan praktik yang baik dalam pemrograman defensif:

- Validasi input pada konstruktor dan method.
- Penanganan kemungkinan overflow.
- Pemisahan tanggung jawab melalui method khusus.

Penerapan prinsip Design by Contract memperjelas ekspektasi terhadap method (seperti batasan nilai input), sedangkan Defensive Programming membuat program tahan terhadap input tidak valid dan kondisi ekstrem.

