

TJ MODUL 6

Design By Contract And Defensive Programming



Nama :

Dhiemas Tulus Ikhsan (2311104046)

Dosen :

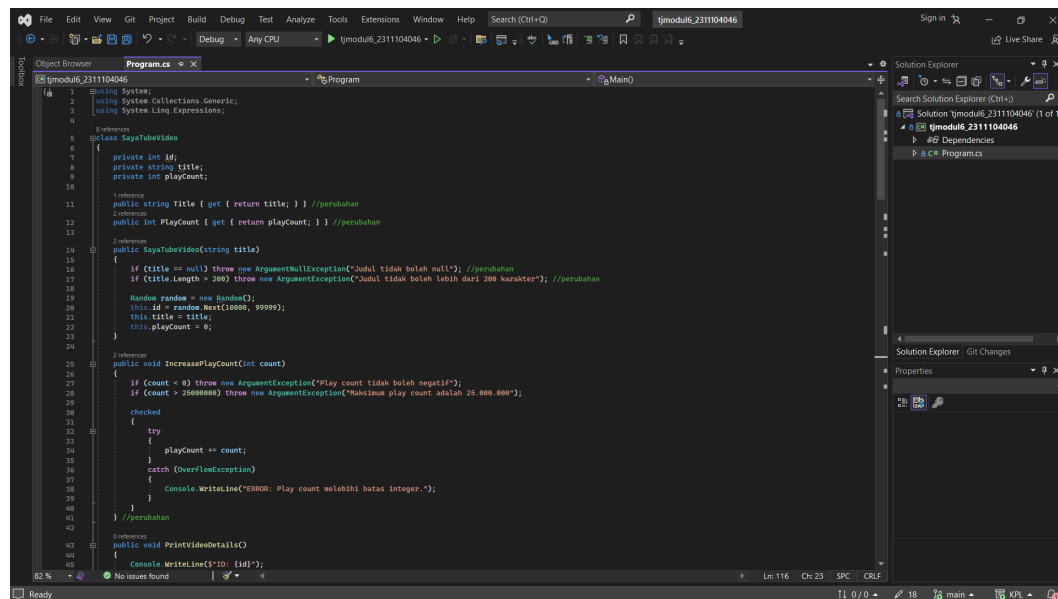
Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

TELKOM UNIVERSITY PURWOKERTO

2025

Penjelasan Program - Design by Contract dan Defensive Programming



Program ini terdiri dari dua class utama, yaitu 'SayaTubeVideo' dan 'SayaTubeUser'. Keduanya menerapkan prinsip Design by Contract dan Defensive Programming* untuk menjaga validitas data dan stabilitas aplikasi.

Class SayaTubeVideo

Class ini merepresentasikan sebuah video dengan properti ID, judul (title), dan jumlah pemutaran (playCount).

Pada konstruktor, diterapkan defensive programming untuk memastikan bahwa judul tidak null dan tidak melebihi 200 karakter:

```
if (title == null) throw new ArgumentNullException("Judul tidak boleh null");  
if (title.Length > 200) throw new ArgumentException("Judul tidak boleh lebih dari 200 karakter");
```

ID video digenerate secara acak antara 10000 hingga 99999. Jumlah play count diinisialisasi ke 0.

Fungsi 'IncreasePlayCount(int count)' akan menambah jumlah play count dengan validasi:

- Nilai 'count' tidak boleh negatif.
- Maksimum yang boleh ditambahkan adalah 25.000.000.
- Blok 'checked' digunakan untuk mendeteksi overflow jika penambahan play count melebihi batas integer ('int.MaxValue').

```
if (count < 0) throw new ArgumentException("Play count tidak boleh negatif");  
if (count > 25000000) throw new ArgumentException("Maksimum play count
```

adalah 25.000.000");

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

Fungsi `PrintVideoDetails()` menampilkan ID, judul, dan play count dari video.

Class `SayaTubeUser`

Class ini merepresentasikan pengguna yang memiliki daftar video yang telah diunggah. Properti pentingnya:

- `Username` (maksimal 100 karakter dan tidak boleh null)
- `uploadedVideos` (list video yang ditambahkan user)

Pada konstruktor, validasi input juga diterapkan untuk menjaga konsistensi data:

```
if (username == null) throw new ArgumentNullException("Username tidak boleh null");
```

```
if (username.Length > 100) throw new ArgumentException("Username tidak boleh lebih dari 100 karakter");
```

Fungsi `AddVideo(SayaTubeVideo video)` memiliki validasi tambahan:

- Tidak menerima video null.
- Tidak menerima video dengan `PlayCount` lebih dari batas integer maksimum.

```
if (video == null) throw new ArgumentNullException("Video tidak boleh null");
```

```
if (video.PlayCount >= int.MaxValue) throw new ArgumentException("Play count tidak boleh lebih dari batas maksimum integer");
```

Fungsi `GetTotalVideoPlayCount()` akan mengakumulasi semua play count dari video yang telah ditambahkan.

Fungsi `PrintAllVideoPlaycount()` menampilkan maksimal 8 video yang diunggah user:

```
int maxPrint = Math.Min(8, uploadedVideos.Count);
for (int i = 0; i < maxPrint; i++)
{
    Console.WriteLine($"Video {i + 1} judul: {uploadedVideos[i].Title}");
}
```

Main Program

Dalam fungsi `Main`, dibuat objek `SayaTubeUser` bernama `"Dhiemas Tulus"` lalu ditambahkan 10 video dengan judul yang telah didefinisikan. Semua ini dibungkus dalam blok `try-catch` untuk menangani error yang mungkin terjadi selama runtime.

Setelah semua video ditambahkan dan ditampilkan, program menguji batas error dengan memanggil `IncreasePlayCount()` dengan dua nilai yang tidak valid:

- `25000001` → melebihi batas maksimum yang diizinkan.
- `-5` → nilai negatif.

Blok `try-catch` khusus disiapkan untuk masing-masing uji coba ini:

```
try
{
    testVideo.IncreasePlayCount(25000001); // Harus error
}
catch (Exception e)
{
    Console.WriteLine($"ERROR: {e.Message}");
}
```

Hasilnya, pesan kesalahan ditampilkan dan program tidak mengalami crash. Inilah kekuatan dari defensive programming: program tetap aman walau inputnya tidak valid.

