



# JURNAL MODUL 6



Disusun Oleh :  
**Nama : Ganes Gemi Putra**  
**Kelas : SE-07-02**  
**NIM : (2311104075)**

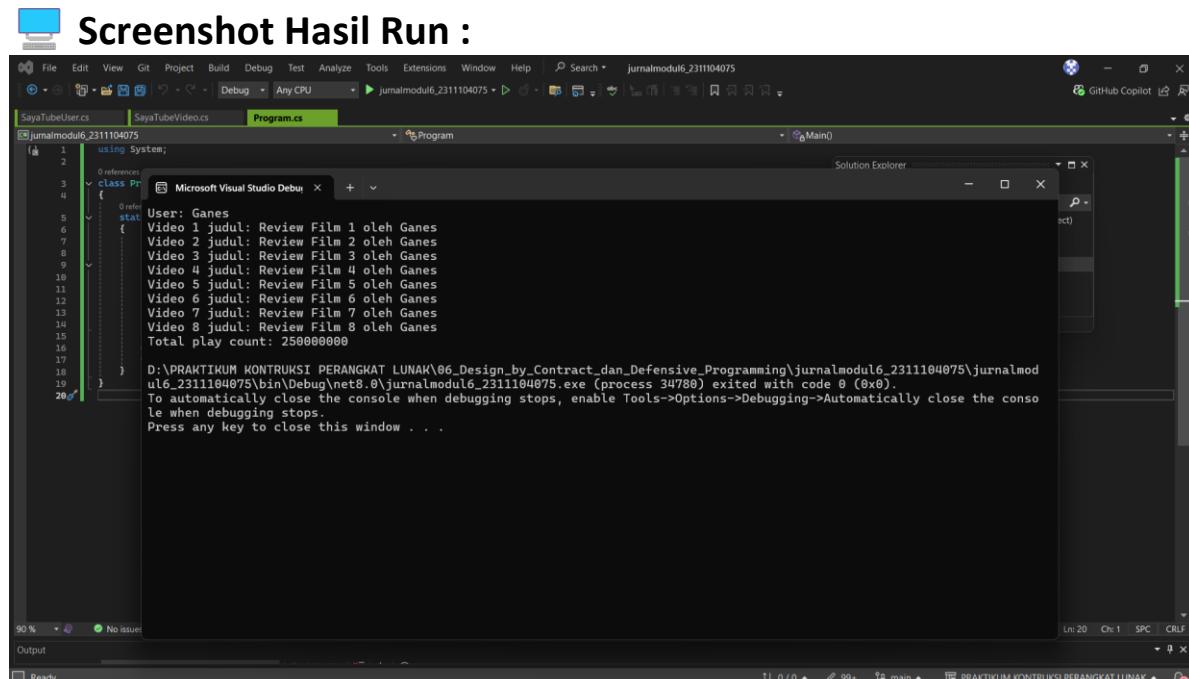
**Dosen : YUDHA ISLAMI SULISTYA**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY**  
**PURWOKERTO**  
**2025**



## Link Repository GitHub :

[https://github.com/GANES2/KPL\\_GANESGEMIPUTRA\\_2311104075\\_SE-07-02/tree/main/06 Design by Contract dan Defensive Programming/Guided06](https://github.com/GANES2/KPL_GANESGEMIPUTRA_2311104075_SE-07-02/tree/main/06%20Design%20by%20Contract%20dan%20Defensive%20Programming/Guided06)

 Screenshot Hasil Run :

The screenshot shows the Microsoft Visual Studio interface during a debugging session. The code editor displays three files: SayaTubeUser.cs, SayaTubeVideo.cs, and Program.cs. The Program.cs file contains the Main() method. The output window shows the execution of the program, specifically the SayaTubeUser class, which prints a list of video reviews and a total play count. The console output is as follows:

```

using System;
class Pt
{
    static User: Ganes
    Video 1 judul: Review Film 1 oleh Ganes
    Video 2 judul: Review Film 2 oleh Ganes
    Video 3 judul: Review Film 3 oleh Ganes
    Video 4 judul: Review Film 4 oleh Ganes
    Video 5 judul: Review Film 5 oleh Ganes
    Video 6 judul: Review Film 6 oleh Ganes
    Video 7 judul: Review Film 7 oleh Ganes
    Video 8 judul: Review Film 8 oleh Ganes
    Total play count: 250000000
}

```

D:\PRAKTIKUM KONTRUKSI PERANGKAT LUNAK\06.Design\_by\_Contract\_dan\_Defensive\_Programming\jurnalmodul6\_2311104075\jurnalmodul6\_2311104075\bin\Debug\net8.0\jurnalmodul6\_2311104075.exe (process 34788) exited with code 0 (0x0).  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .

Gambar 1. Output program saat testing Design by Contract di class SayaTubeUser dan SayaTubeVideo.

### 💡 Penjelasan Singkat Implementasi :

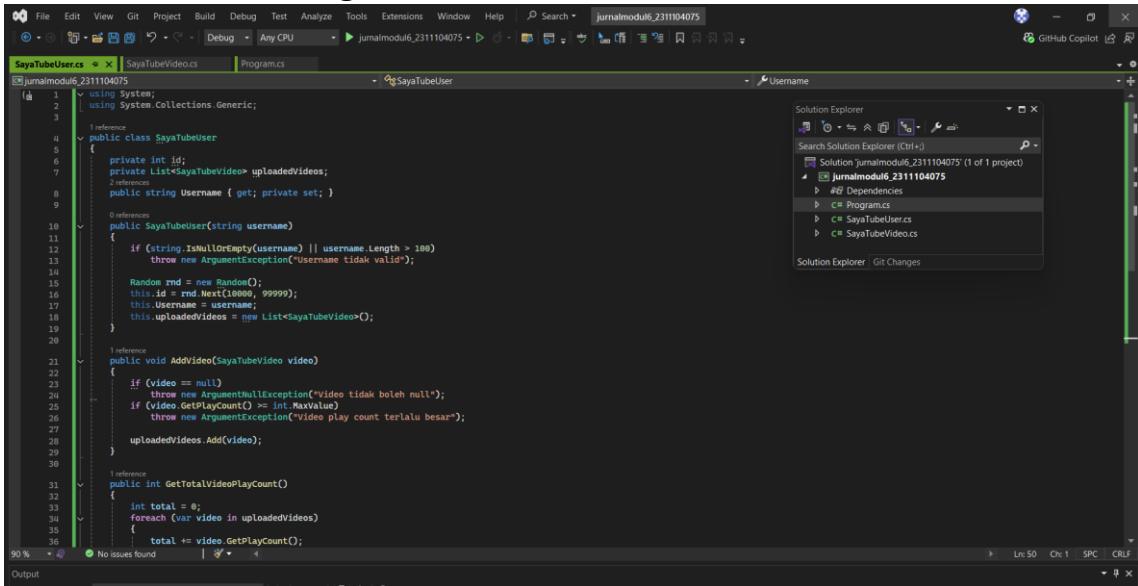
Dalam program ini, saya membuat dua class utama yaitu SayaTubeVideo dan SayaTubeUser. Keduanya menerapkan prinsip *Design by Contract* yang bertujuan untuk memastikan bahwa setiap objek bekerja sesuai dengan aturan yang sudah ditentukan sejak awal, seperti validasi input dan penanganan error.

Di class SayaTubeVideo, saya memastikan bahwa judul video tidak kosong dan panjangnya tidak melebihi 200 karakter. Saat play count ditambahkan, nilainya juga tidak boleh lebih dari 25 juta dan tidak boleh negatif. Selain itu, saya menggunakan checked dan try-catch untuk menghindari overflow agar program tidak berhenti saat nilai play count melebihi batas maksimum integer.

Sementara itu, di class SayaTubeUser, saya melakukan validasi pada nama pengguna agar tidak kosong dan tidak melebihi 100 karakter. Video yang ditambahkan ke daftar juga dicek agar tidak bernilai null dan tidak memiliki play count yang berlebihan. Sebagai bagian dari postcondition, saya membatasi agar hanya maksimal 8 video yang ditampilkan di console saat daftar video dicetak.



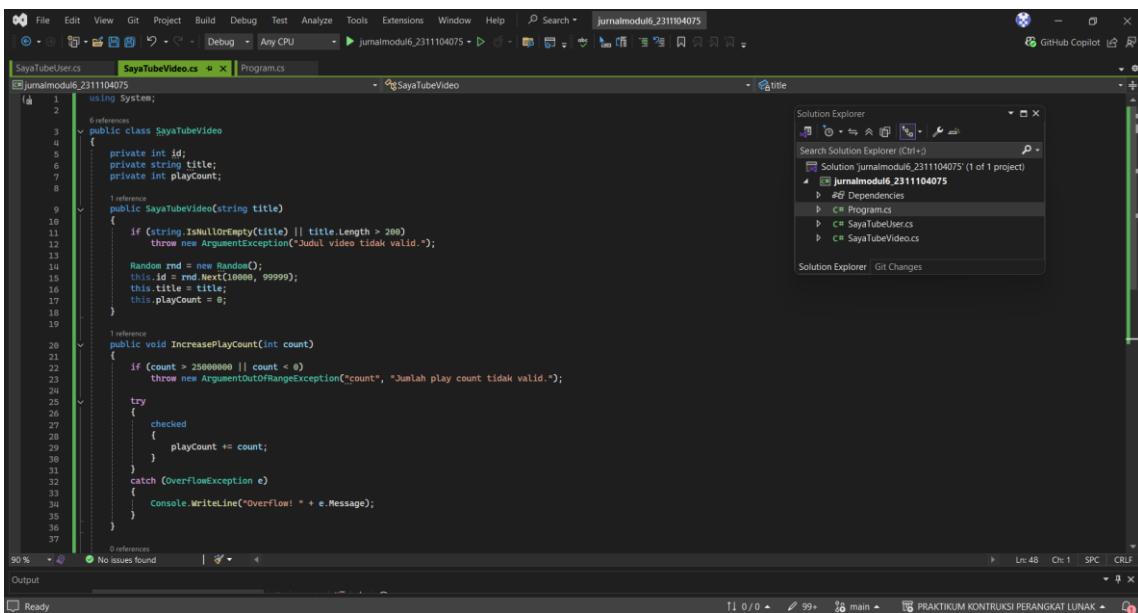
## Screenshot Potongan Kode



```

1  using System;
2  using System.Collections.Generic;
3
4  public class SayTubeUser
5  {
6      private int id;
7      private List<SayTubeVideo> uploadedVideos;
8      private string Username { get; private set; }
9
10     public SayTubeUser(string username)
11     {
12         if (string.IsNullOrEmpty(username) || username.Length > 100)
13             throw new ArgumentException("Username tidak valid");
14
15         Random rnd = new Random();
16         this.id = rnd.Next(100000, 999999);
17         this.Username = username;
18         this.uploadedVideos = new List<SayTubeVideo>();
19     }
20
21     public void AddVideo(SayTubeVideo video)
22     {
23         if (video == null)
24             throw new ArgumentNullException("Video tidak boleh null");
25         if (video.GetPlayCount() > int.MaxValue)
26             throw new ArgumentException("Video play count terlalu besar");
27
28         uploadedVideos.Add(video);
29     }
30
31     public int GetTotalVideoPlayCount()
32     {
33         int total = 0;
34         foreach (var video in uploadedVideos)
35         {
36             total += video.GetPlayCount();
37         }
38     }
39
40 }

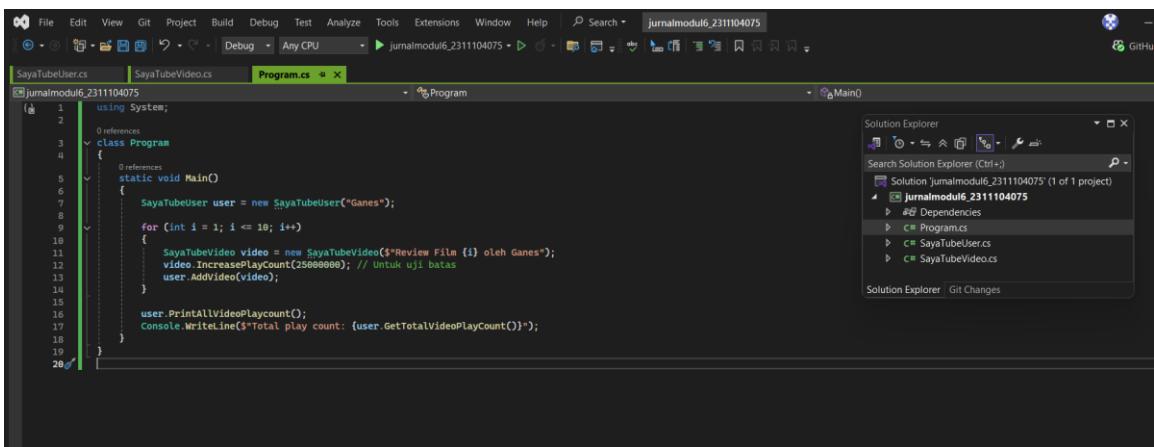
```



```

1  using System;
2
3  public class SayTubeVideo
4  {
5      private int id;
6      private string title;
7      private int playCount;
8
9      public SayTubeVideo(string title)
10     {
11         if (string.IsNullOrEmpty(title) || title.Length > 200)
12             throw new ArgumentException("Judul video tidak valid.");
13
14         Random rnd = new Random();
15         this.id = rnd.Next(100000, 999999);
16         this.title = title;
17         this.playCount = 0;
18     }
19
20     public void IncreasePlayCount(int count)
21     {
22         if (count > 25000000 || count < 0)
23             throw new ArgumentOutOfRangeException("count", "Jumlah play count tidak valid.");
24
25         try
26         {
27             checked
28             {
29                 playCount += count;
30             }
31         }
32         catch (OverflowException e)
33         {
34             Console.WriteLine("Overflow! " + e.Message);
35         }
36     }
37
38 }

```



```

1  using System;
2
3  class Program
4  {
5      static void Main()
6      {
7          SayTubeUser user = new SayTubeUser("Ganes");
8
9          for (int i = 1; i <= 10; i++)
10         {
11             SayTubeVideo video = new SayTubeVideo($"Review Film {i} oleh Ganes");
12             video.IncreasePlayCount(25000000); // Untuk uji batas
13             user.AddVideo(video);
14         }
15
16         user.PrintAllVideoPlayCount();
17         Console.WriteLine($"Total play count: {user.GetTotalVideoPlayCount()}");
18     }
19
20 }

```

Gambar 2. Validasi pada konstruktor dan pengecekan overflow pada play count.

