TP MODUL 14

CLEAN CODE

Nama :

Dhiemas Tulus Ikhsan (2311104046)

Dosen :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

TELKOM UNIVERSITY PURWOKERTO

2025

1. Tujuan Praktikum

   Pada Tugas Pendahuluan Modul 14 ini bertujuan untuk menerapkan proses refactoring terhadap kode program C# agar lebih rapi, terstruktur, mudah dibaca, dan sesuai dengan konvensi penulisan standar (coding standard) bahasa C# (PascalCase, camelCase, dan indentasi).

2. Deskripsi Program

   Program ini terdiri dari dua kelas utama:
   - Program: sebagai main class yang menjalankan objek.
   - SayaTubeVideo: class yang merepresentasikan video dan memiliki atribut videoId, videoTitle, dan playCount. Terdapat juga method IncreasePlayCount(int count) dan PrintDetails().

3. Refactoring Dengan Standar Code

   a. Naming Convention (Konvensi Penamaan)

      Sebelum refactoring, penamaan variabel dan method di program masih kurang konsisten dan terlalu umum. Misalnya, variabel bernama id dan title digunakan untuk menyimpan informasi video, tetapi nama-nama tersebut tidak menjelaskan konteksnya secara spesifik. Method untuk mencetak detail video juga dinamai PrintVideoDetails(), yang terlalu panjang dan kurang efisien.

      Setelah refactoring, semua penamaan disesuaikan dengan standar .NET:
      - Variabel id diganti menjadi videoId agar lebih deskriptif dan tidak ambigu.
      - Variabel title diubah menjadi videoTitle supaya lebih jelas bahwa itu adalah judul video.
      - Method PrintVideoDetails() diubah menjadi PrintDetails(), mengikuti gaya penulisan PascalCase dan agar lebih ringkas.

      Secara umum:
      - Variabel dan atribut menggunakan camelCase.
      - Method dan class menggunakan PascalCase.

      Perubahan ini menjadikan kode lebih mudah dibaca dan dipahami oleh programmer lain, serta konsisten dengan standar industri.

   b. White Space dan Indentation

      Sebelum refactoring, beberapa bagian kode ditulis tanpa jarak antar method, dan ada ketidakkonsistenan dalam indentasi (spasi/tab) yang membuat struktur kode kurang rapi saat dibaca.

      Setelah refactoring:
      - Setiap method dipisahkan oleh satu baris kosong agar kode lebih terstruktur.
      - Indentasi diatur secara konsisten menggunakan 4 spasi untuk setiap blok kode, sesuai standar Visual Studio dan .NET.
      - Penempatan kurung kurawal ({}) diperhatikan agar seragam dan rapi.

Dengan perubahan ini, program menjadi lebih bersih secara visual, lebih profesional, dan nyaman saat dibaca maupun dikembangkan lebih lanjut.

c. Penulisan dan Penggunaan Atribut

Sebelum refactoring, penamaan atribut masih terlalu umum seperti id dan title. Meskipun fungsinya sederhana, nama tersebut bisa membingungkan jika program dikembangkan lebih besar.

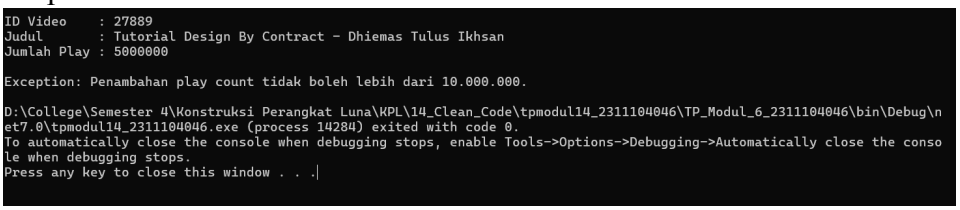
Setelah refactoring, atribut ditulis dengan nama yang lebih menjelaskan fungsinya, seperti videoId dan videoTitle. Penulisan ini menjadikan atribut lebih deskriptif, dan sangat berguna terutama saat bekerja dalam tim atau membaca ulang kode setelah waktu yang lama.

Selain itu:

- Konstruktor kini memvalidasi input judul video agar tidak null, kosong, atau terlalu panjang (lebih dari 100 karakter).
- Method IncreasePlayCount memiliki batas maksimal penambahan 10.000.000 dan menangani overflow dengan blok try-catch dan checked, untuk keamanan nilai integer.

Atribut playCount diinisialisasi di konstruktor dengan nilai 0.

d. Output



e. Kesimpulan

Kode sebelum refactoring masih menggunakan penamaan yang terlalu umum, struktur tidak konsisten, dan belum semuanya mengikuti standar penulisan profesional. Setelah refactoring, kode menjadi jauh lebih rapi, jelas, dan sesuai dengan konvensi .NET. Hal ini penting untuk menjaga keterbacaan, kemudahan debugging, dan kesiapan untuk proyek dalam skala besar atau tim.