

## PROYEK AKHIR SEMESTER GASAL



NAMA : SITI AFRIANI  
KELAS : X PPLG 1  
NIS : 258733

JUDUL PROYEK : APLIKASI “JADWAL TUGAS”

PENGEMBANGAN PERANGKAT LUNAK DAN GIM  
**SMK NEGERI 1 KANDEMAN**  
TAHUN PELAJARAN 2025/2026

## DESKRIPSI PROYEK

Proyek saya adalah membuat program manajemen jadwal tugas yang berbasis console yang memungkinkan pengguna untuk menambah tugas, melihat jadwal tugas, serta menghapus tugas. Program ini sederhana, interaktif, dan cocok untuk mempelajari dasar struktur data, validasi input, dan kontrol alur program.

1. Melihat daftar tugas yang sudah tersimpan.
2. Menambah tugas baru.
3. Menghapus tugas tertentu berdasarkan nomor.
4. Mengelola data tugas secara sederhana.

Proyek ini mengimplementasikan **konsep dasar pemrograman** seperti:

1. **Struktur Dasar Program C#** : Struktur umum program C# selalu dimulai dengan class dan metode Main().
2. **Variabel dan Tipe Data** : Digunakan untuk menyimpan nilai agar bisa digunakan Kembali
3. **Perulangan (Looping)** : Digunakan agar program mengulang perintah tertentu selama kondisi masih benar.
4. **Percabangan (Kondisi / Control Flow)** : Digunakan untuk mengambil keputusan berdasarkan kondisi tertentu.
5. **Koleksi Data (List)** : Digunakan untuk menyimpan banyak data sekaligus.
6. **Input dan Output** : Program berinteraksi dengan pengguna melalui console.
7. **Validasi dan Konversi Data** : Untuk memastikan input angka dari pengguna valid, digunakan int.TryParse().
8. **Operasi CRUD Sederhana** : Program ini menerapkan operasi CRUD (Create, Read, Delete) pada daftar tugas.

## RINGKASAN TEORI C# YANG DIGUNAKAN

### 1. Struktur Dasar Program C#

Semua program C# diawali dengan kelas (class) dan memiliki metode Main() sebagai titik awal eksekusi program.

```
class Program
{
    static void Main()
```

- class Program Menandakan satu blok program.
- static void Main() Titik mulai program dijalankan.
- Semua logika aplikasi ditulis di dalam metode Main().

### 2. Variabel dan Tipe Data

Program menggunakan beberapa tipe data dasar:

```
List<string> todoList = new List<string>();
bool running = true;
string pilihan;
```

```
List<string> todoList = new List<string>();
bool running = true;
```

- List<string> Koleksi data yang menyimpan banyak elemen bertipe teks (string).
- bool Tipe data logika (bernilai true atau false).
- string Tipe data untuk teks.

### 3. Struktur Perulangan (Looping)

Program menggunakan while loop untuk menampilkan menu terus-menerus sampai pengguna memilih keluar

```
while (running)
{
```

- Selama running bernilai true, program akan terus berulang dan menampilkan menu utama.

### 4. Struktur Percabangan (Switch-Case)

Digunakan untuk memilih perintah yang akan dijalankan sesuai input pengguna.

```
case "1": break;
case "2": break;
case "3": break;
case "4": break;
default : break;
```

- switch memeriksa nilai variabel pilihan, lalu mengeksekusi blok kode tertentu yang sesuai.

### 5. Koleksi Data (List)

Program menggunakan List<string> untuk menyimpan daftar tugas.

```
List<string>
```

- Digunakan untuk menyimpan banyak tugas.

### 6. Input dan Output

Untuk berinteraksi dengan pengguna.

Input digunakan: untuk menerima data dari pengguna, sedangkan output menampilkan hasil.

Contoh

```
Console.Write();
string pilihan = Console.ReadLine();
```

- untuk menampilkan teks
- untuk membaca input pengguna

### 7. Validasi dan Konversi Data

Program memastikan input dari pengguna valid sebelum diproses.

```
if (int.TryParse(Console.ReadLine(), out int nomor))
```

- TryParse() digunakan untuk mengonversi teks ke angka dengan aman tanpa menyebabkan error.  
Ini menunjukkan konsep penanganan input dan validasi data.

## 8. Operasi CRUD Sederhana

Operasi digunakan untuk Fungsi Implementasi

```
todoList.Add(tugasBaru)

todoList.RemoveAt(nomor - 1);

for (int i = 0; i < todoList.Count; i++)
{
    Console.WriteLine($"{i + 1}. {todoList[i]}");
}
```

- |          |                      |                              |
|----------|----------------------|------------------------------|
| ▪ Create | Menambah tugas       | todoList.Add(tugasBaru)      |
| ▪ Delete | Menghapus tugas      | todoList.RemoveAt(nomor - 1) |
| ▪ Read   | Melihat daftar tugas | perulangan for               |

## 9. Operator

Operator digunakan untuk melakukan perhitungan atau operasi logika.

jenis Operator Simbol Contoh:

```
nomor > 0 && nomor <= todoList.Count
i < todoList.Count
```

1. Aritmatika → +, ++
2. Perbandingan → ==, !=, <, <=, >
3. Logika → &&
4. Penugasan → =
5. Indexer → []

## PERANCANGAN PROGRAM

Rancangan Tampilan Aplikasi Daftar tugas

```
==== APLIKASI JADWAL TUGAS ====
1. Lihat Daftar Tugas
2. Tambah Tugas
3. Hapus Tugas
4. Keluar

Pilih menu (1-4): |
```

## IMPLEMENTASI PROGRAM

Kode program utama

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        List<string> todoList = new List<string>();
        bool running = true;

        while (running)
        {
            Console.Clear();
            Console.WriteLine("==== APLIKASI JADWAL TUGAS ====");
            Console.WriteLine("1. Lihat Daftar Tugas");
            Console.WriteLine("2. Tambah Tugas");
            Console.WriteLine("3. Hapus Tugas");
            Console.WriteLine("4. Keluar");
            Console.Write("\nPilih menu (1-4): ");

            string pilihan = Console.ReadLine();
            Console.Clear();

            switch (pilihan)
            {
                case "1":
                    Console.WriteLine("=== DAFTAR TUGAS ===");
```

```

        if (todoList.Count == 0)
        {
            Console.WriteLine("Belum ada tugas.");
        }
        else
        {
            for (int i = 0; i < todoList.Count; i++)
            {
                Console.WriteLine($"{i + 1}. {todoList[i]}");
            }
            Console.WriteLine("\nTekan ENTER untuk kembali...");
            Console.ReadLine();
            break;
        }

    case "2":
        Console.Write("Masukkan nama tugas: ");
        string tugasBaru = Console.ReadLine();
        if (!string.IsNullOrEmpty(tugasBaru))
        {
            todoList.Add(tugasBaru);
            Console.WriteLine("Tugas berhasil ditambahkan!");
        }
        else
        {
            Console.WriteLine("Nama tugas tidak boleh kosong.");
        }
        Console.WriteLine("\nTekan ENTER untuk kembali...");
        Console.ReadLine();
        break;

    case "3":
        Console.WriteLine("=== HAPUS TUGAS ===");
        for (int i = 0; i < todoList.Count; i++)
        {
            Console.WriteLine($"{i + 1}. {todoList[i]}");
        }

        Console.Write("\nMasukkan nomor tugas yang ingin dihapus: ");
        if (int.TryParse(Console.ReadLine(), out int nomor) && nomor
            > 0 && nomor <= todoList.Count)
        {
            todoList.RemoveAt(nomor - 1);
            Console.WriteLine("Tugas berhasil dihapus!");
        }
        else
        {
            Console.WriteLine("Nomor tidak valid.");
        }
        Console.WriteLine("\nTekan ENTER untuk kembali...");
        Console.ReadLine();
        break;

    case "4":
        running = false;
        Console.WriteLine("Keluar dari aplikasi...");
        break;

    default:
        Console.WriteLine("Pilihan tidak valid.");
        Console.ReadLine();
        break;
}

```

```

    }
}

```

## PENJELASAN KODE PROGRAM

### 1. Deklarasi List dan Variabel

```

List<string>
bool running = true

```

- digunakan untuk menyimpan daftar tugas.
- menjaga menu tetap berulang.

### 2. Perulangan Menu

```

while (running)

```

- membuat menu tampil terus sampai user memilih keluar.

### 3. Percabangan Switch

Menentukan aksi berdasarkan pilihan user:

```

"1" → tampilkan daftar tugas
"2" → tambah tugas
"3" → hapus tugas
"4" → keluar

```

### 4. Menambah Tugas

```

todoList.Add(tugasBaru);

```

### 5. Menghapus Tugas

Memvalidasi nomor menggunakan:

```

int.TryParse(...)
todoList.RemoveAt(nomor - 1);

```

### 6. Menampilkan Tugas

Menggunakan perulangan for

```

for (int i = 0; i < todoList.Count; i++)
{

```

## Tampilan Output Hasil Program

=== DAFTAR TUGAS ===

1. NAHASA INDONESIA

2. PAI

3. BJAWA

4. PIPAS

Tekan ENTER untuk kembali...

## LAMPIRAN

### Kode Program Lengkap

```
Program.cs
project-akhir-semester
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  using System;
8  using System.Collections.Generic;
9
10 class Program
11 {
12     static void Main()
13     {
14         List<string> todoList = new List<string>();
15         bool running = true;
16
17         while (running)
18         {
19             Console.Clear();
20             Console.ForegroundColor = ConsoleColor.Yellow;
21             Console.WriteLine("==== APLIKASI JADWAL TUGAS ====");
22             Console.WriteLine("1. Lihat Daftar Tugas");
23             Console.WriteLine("3. Hapus Tugas");
24             Console.WriteLine("4. Keluar");
25             Console.Write("\nPilih menu (1-4): ");
```



```
Program.cs* X
project-akhir-semester Program
26
27 string pilihan = Console.ReadLine();
28 Console.Clear();
29
30 switch (pilihan)
31 {
32     case "1":
33         Console.WriteLine("=== DAFTAR TUGAS ===");
34         if (todoList.Count == 0)
35         {
36             Console.WriteLine("Belum ada tugas.");
37         }
38         else
39         {
40             for (int i = 0; i < todoList.Count; i++)
41             {
42                 Console.WriteLine($"{i + 1}. {todoList[i]}");
43             }
44             Console.WriteLine("\nTekan ENTER untuk kembali...");
45             Console.ReadLine();
46             break;
47         }
48     case "2":
49         Console.WriteLine("Masukkan nama tugas: ");
50         string tugasBaru = Console.ReadLine();
51         if (!string.IsNullOrEmpty(tugasBaru))
52         {
53             todoList.Add(tugasBaru);
54             Console.WriteLine("Tugas berhasil ditambahkan!");
55             Console.WriteLine("\nTekan ENTER untuk kembali...");
56             Console.ReadLine();
57         }
58         break;
59 }
```

```
Program.cs* X
project-akhir-semester Program
56
57     }
58     else
59     {
60         Console.WriteLine("Nama tugas tidak boleh kosong.");
61     }
62     Console.WriteLine("\nTekan ENTER untuk kembali...");
63     Console.ReadLine();
64     break;
65 }
66 case "3":
67     Console.WriteLine("=== HAPUS TUGAS ===");
68     for (int i = 0; i < todoList.Count; i++)
69     {
70         Console.WriteLine($"{i + 1}. {todoList[i]}");
71     }
72     Console.WriteLine("\nMasukkan nomor tugas yang ingin dihapus: ");
73     if (int.TryParse(Console.ReadLine(), out int nomor) && nomor > 0 && nomor <= todoList.Count)
74     {
75         todoList.RemoveAt(nomor - 1);
76         Console.WriteLine("Tugas berhasil dihapus!");
77     }
78     else
79     {
80         Console.WriteLine("Nomor tidak valid.");
81     }
82     Console.WriteLine("\nTekan ENTER untuk kembali...");
83     Console.ReadLine();
84     break;
85 }
```

```
Program.cs* X
project-akhir-semester Program
77
78     }
79     else
80     {
81         Console.WriteLine("Nomor tidak valid.");
82     }
83     Console.WriteLine("\nTekan ENTER untuk kembali...");
84     Console.ReadLine();
85     break;
86 }
87 case "4":
88     running = false;
89     Console.WriteLine("Keluar dari aplikasi...");
90     break;
91 default:
92     Console.WriteLine("Pilihan tidak valid.");
93     Console.ReadLine();
94     break;
95 }
96 }
97 }
98 }
99 }
```

ScreenShot Tampilan Program

=== DAFTAR TUGAS ===

1. MATEMATIKA
2. BAHASA INDONESIA
3. PIPAS
4. SENI BUDAYA
5. INFORMATIKA

Tekan ENTER untuk kembali...

|