

TUGAS PENDAHULUAN KONTRUKSI PERANGKAT LUNAK MODUL 4



**Nama : Ganes Gemi Putra
SE-07-02
(2311104075)**

**DOSEN :
YUDHA ISLAMI SULISTYA**

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

Tujuan Praktikum

1. Memahami konsep Automata dan Table-Driven Construction.
2. Mengimplementasikan state machine pada class sederhana

Kode Program :

1. Class DoorMachine

```
using System;

namespace TugasModul4
{
    public class DoorMachine
    {
        private string state = "Terkunci"; // State awal adalah
        terkunci

        public void Kunci()
        {
```

```
        state = "Terkunci";

        Console.WriteLine("Pintu terkunci");
    }

    public void Buka()
    {
        state = "Terbuka";

        Console.WriteLine("Pintu tidak terkunci");
    }

    public string GetState()
    {
        return state;
    }
}
```

2. Main Program

```
using System;

namespace TugasModul4
{
    class Program
    {
        static void Main(string[] args)
        {
            DoorMachine door = new
            DoorMachine();

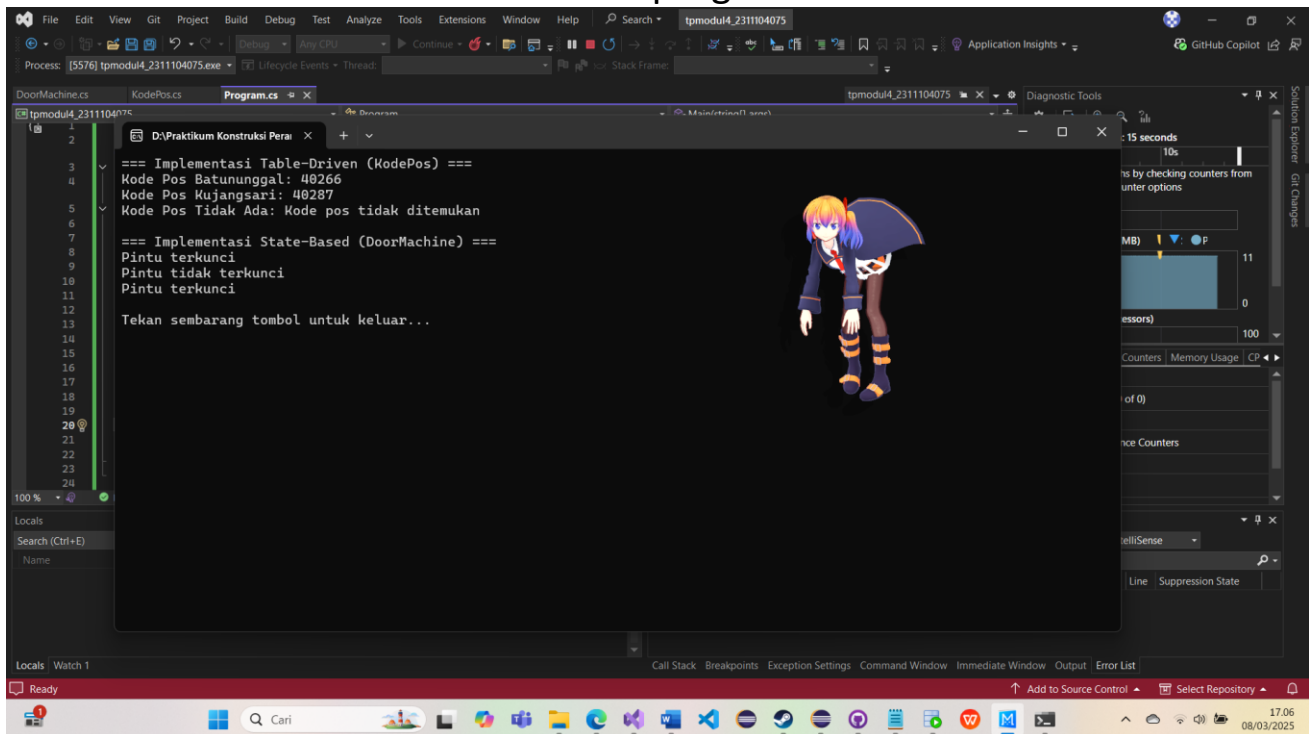
            // State awal: Terkunci
            door.Kunci();

            // Ubah state menjadi Terbuka
            door.Buka();

            // Ubah kembali state menjadi Terkunci
            door.Kunci();
        }
    }
}
```

Output Program

Berikut adalah screenshot hasil dari program:



Penjelasan

Tugas ini bertujuan untuk memahami dua pendekatan pemrograman yang berbeda, yaitu **Table-Driven** dan **State-Based**, yang digunakan dalam pengembangan perangkat lunak.

Table-Driven (KodePos):

Konsep ini menggunakan tabel (dalam bentuk dictionary pada C#) untuk menyimpan data yang berpasangan, seperti nama wilayah dan kode posnya.

Dengan pendekatan ini, kita dapat mencari kode pos berdasarkan nama wilayah secara efisien tanpa menggunakan banyak logika bercabang seperti if-else atau switch-case.

Pada implementasi, tabel ini diakses menggunakan metode `GetKodePos`, yang mengembalikan kode pos sesuai nama wilayah yang diberikan, atau pesan "Kode pos tidak ditemukan" jika wilayah tersebut tidak ada di tabel.

State-Based (DoorMachine):

Pendekatan ini menggunakan state machine sederhana untuk memodelkan status pintu (terkunci atau terbuka).

Setiap perubahan status dilakukan dengan metode Kunci atau Buka, yang mengubah state internal dari objek pintu. Implementasi ini menunjukkan bagaimana sebuah state machine dapat digunakan untuk melacak kondisi sistem dan membatasi transisi antara status tertentu.

Kesimpulan

Table-Driven:

Pendekatan ini sangat cocok untuk kasus yang membutuhkan pengambilan data berdasarkan pasangan kunci-nilai, karena sederhana, mudah dipahami, dan efisien. Dalam tugas ini, pendekatan ini berhasil digunakan untuk memetakan wilayah ke kode pos.

State-Based:

Pendekatan ini sangat bermanfaat untuk memodelkan sistem yang memiliki status tertentu, seperti pintu yang hanya dapat berpindah antara status terkunci dan terbuka. Implementasi ini memberikan pemahaman tentang bagaimana state machine dapat diterapkan untuk mengontrol transisi dalam perangkat lunak.

Gabungan Implementasi:

Kedua pendekatan ini menunjukkan fleksibilitas dalam pengembangan perangkat lunak. Dengan menggabungkan kedua metode ini, kita bisa memahami cara menangani masalah berbeda dengan solusi yang tepat, efisien, dan modular.

