

TP MODUL 14



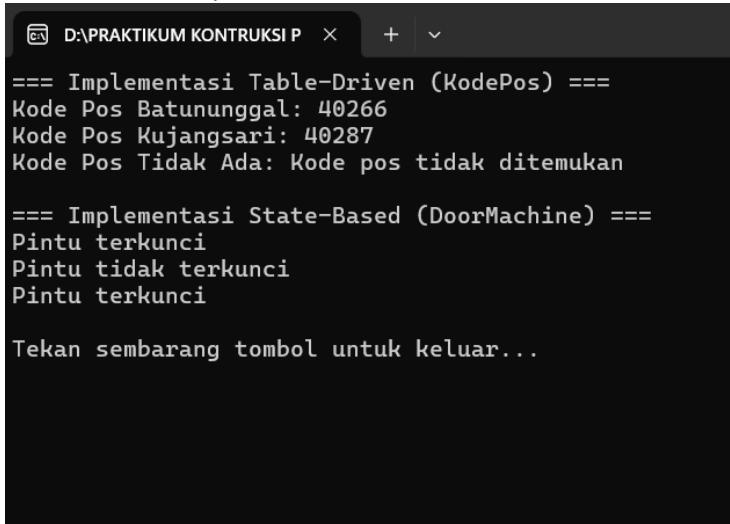
Disusun Oleh :
Nama : Ganes Gemi Putra
Kelas : SE-07-02
NIM : (2311104075)

Dosen : YUDHA ISLAMI SULISTYA

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2025

Link Github : https://github.com/GANES2/KPL_GANESGEMIPUTRA_2311104075_SE-07-02/tree/main

I. Screenshot Output



```

D:\PRAKTIKUM KONTRUKSI P + v
==== Implementasi Table-Driven (KodePos) ====
Kode Pos Batununggal: 40266
Kode Pos Kujangsari: 40287
Kode Pos Tidak Ada: Kode pos tidak ditemukan

==== Implementasi State-Based (DoorMachine) ====
Pintu terkunci
Pintu tidak terkunci
Pintu terkunci

Tekan sembarang tombol untuk keluar...

```

II. Jawaban Instruksi TP

1. MEMBUAT PROJECT MODUL

Jawab:

Saya menggunakan folder dari TP Modul 4 karena memiliki struktur program yang sederhana dan mudah untuk direfactor. Folder telah disalin dan dinamai ulang menjadi tpmodul14_2311104075, lalu dibuka di Visual Studio.

2. REFACTORING DENGAN STANDAR CODE

A. Naming Convention

Elemen	Contoh	Keterangan
Variable lokal	kelurahan, kodePos	camelCase sesuai standar C#
Property/Attribute	dataKelurahan	camelCase (non-public), PascalCase jika public
Method/Function/Procedure	GetKodePos(), Kunci()	PascalCase sesuai konvensi method .NET

B. White space dan Indentation

Aspek	Status	Penjelasan
Indentasi konsisten	✓	4 spasi per blok
Jarak antar blok kode	✓	Ada baris kosong antar method/class
Baris kosong berlebih	✗	Tidak ditemukan

C. Variable/Attribute Declarations

Elemen	Status	Penjelasan
Dictionary	✓	Dideklarasikan di awal method
Variabel lokal	✓	Diinisialisasi sebelum digunakan

D. Comments

Bagian Kode	Status	Penjelasan
Komentar per class/method	✓	Menjelaskan tujuan fungsi
Komentar tidak berlebihan	✓	Hanya bagian penting saja
Format komentar	✓	Menggunakan // satu baris

III. Penjelasan Kode Implementasi

❖ **KodePos.cs**

Kelas KodePos menggunakan dictionary Dictionary<string, string> untuk menyimpan pasangan nama kelurahan dan kode pos. Fungsi GetKodePos() mencari nilai berdasarkan nama kelurahan yang dimasukkan pengguna.

```
public string GetKodePos(string kelurahan)
{
    if (data.ContainsKey(kelurahan))
        return data[kelurahan];
    else
        return "Kode Pos Tidak Ditemukan";
}
```

❖ **DoorMachine.cs**

Class ini menerapkan Finite State Machine (FSM) dengan dua state: TERKUNCI dan TERBUKA. Method seperti Buka() dan Kunci() digunakan untuk mengganti state pintu.

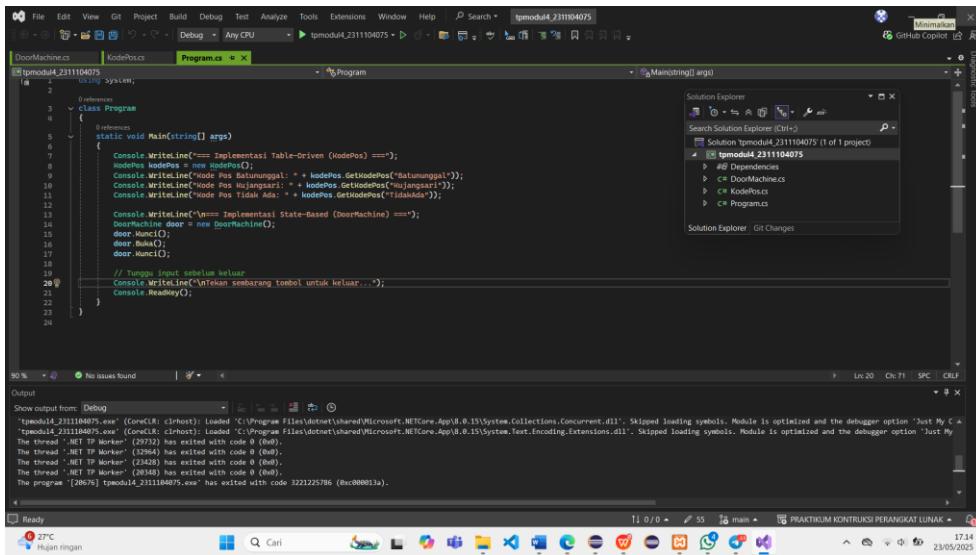
```
public void Buka()
{
    if (state == "TERKUNCI")
    {
        state = "TERBUKA";
        Console.WriteLine("Pintu berhasil dibuka.");
    }
    else
    {
        Console.WriteLine("Pintu sudah terbuka.");
    }
}
```

❖ **Program.cs**

Merupakan titik masuk utama aplikasi. Menjalankan dua fitur utama secara berurutan: table-driven (kode pos) dan state-based (pintu).

```
static void Main(string[] args)
{
    Console.WriteLine("== Implementasi Table-Driven (KodePos) ==");
    KodePos kodePos = new KodePos();
    Console.WriteLine("Kode Pos Batununggal: " +
kodePos.GetKodePos("Batununggal"));

    Console.WriteLine("\n== Implementasi State-Based (DoorMachine) ==");
    DoorMachine door = new DoorMachine();
    door.Kunci();
    door.Buka();
}
```



```

1 //using system;
2 
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         Console.WriteLine("== Implementasi Table-Driven (KodePos) ==");
8         KodePos kodePos = new KodePos();
9         Console.WriteLine("Kode Pos Batumunggal: " + kodePos.GetKodePos("batumunggal"));
10        Console.WriteLine("Kode Pos MuJanggari: " + kodePos.GetKodePos("muJanggari"));
11        Console.WriteLine("Kode Pos Tidak Ada: " + kodePos.GetKodePos("tidakada"));
12 
13        Console.WriteLine("\n== Implementasi State-Based (DoorMachine) ==");
14        DoorMachine door = new DoorMachine();
15        door.Buka();
16        door.Buka();
17        door.Hunci();
18        door.Hunci();
19 
20        // Tunggu input sebelum keluar
21        Console.WriteLine("\nTekan sembarang tombol untuk keluar...");
22    }
23 }

```

No issues found | Output

Show output from: Debug

```

tmodul4_2311104075.exe (CoreCLR; clrhost) Loaded 'C:\Windows\assembly\Microsoft\NETCore.App\8.0.1\System.Collections.Compatibility.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
tmodul4_2311104075.exe (CoreCLR; clrhost) Loaded 'C:\Windows\assembly\shared\Microsoft\NETCore.App\8.0.1\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is optimized and the debugger option 'Just My Code' is enabled.
The thread ".NET TPL Worker" (29732) has exited with code 0 (0).
The thread ".NET TPL Worker" (37964) has exited with code 0 (0).
The thread ".NET TPL Worker" (37965) has exited with code 0 (0).
The thread ".NET TPL Worker" (37966) has exited with code 0 (0).
The program [28676] tmodul4_2311104075.exe has exited with code 3221225786 (0xc000013a).

```

Program ini merupakan titik awal dari aplikasi. Method Main() digunakan untuk menjalankan dua bagian utama dari program, yaitu **table-driven model** dan **state-based model**.

- **Bagian pertama** menampilkan fitur pencarian kode pos berdasarkan nama kelurahan. Program membuat objek dari class KodePos, kemudian memanggil method GetKodePos() dengan input berbeda. Output menunjukkan kode pos yang sesuai atau pesan jika tidak ditemukan.
- **Bagian kedua** menampilkan simulasi buka dan kunci pintu menggunakan konsep **finite state machine (FSM)**. Objek DoorMachine dibuat dan diberikan beberapa perintah: mengunci pintu, membuka, dan mengunci kembali. Transisi state ditangani secara internal dan status perubahan ditampilkan ke pengguna.
- Di akhir eksekusi, program akan menampilkan instruksi untuk menekan sembarang tombol agar jendela tidak langsung tertutup, memungkinkan pengguna membaca hasil program terlebih dahulu.

Program ini berjalan di atas platform .NET, menggunakan bahasa C#, dan telah ditulis dengan memperhatikan standar coding yang baik.

