

LAPORAN PRAKTIKUM
TP MODUL 4
RUNNING MODUL



Nama :

Aulia Jasifa Br Ginting 2311104060

S1SE-07-02

Dosen :

Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

A. Source code dari project yang dibuat

KodePos.cs

```
DoorMachine.cs KodePos.cs* Program.cs
TPMODUL3_2311104060 KodePos postalCodes
1 using System;
2 using System.Collections.Generic;
3
4 // Table-Driven Implementation
5 public class KodePos
6 {
7     // Dictionary to store postal codes
8     private Dictionary<string, string> postalCodes = new Dictionary<string, string>
9     {
10         { "Batununggal", "40266" },
11         { "Kujangsari", "40287" },
12         { "Menger", "40267" },
13         { "Mates", "40256" },
14         { "Cijaura", "40287" },
15         { "Jatisari", "40286" },
16         { "Margasari", "40286" },
17         { "Sekejati", "40286" },
18         { "Hobonmaru", "40272" },
19         { "Malaer", "40274" },
20         { "Samoja", "40273" }
21     };
22
23     // Method to retrieve postal code based on village name
24     public string GetKodePos(string Kelurahan)
25     {
26         if (postalCodes.TryGetValue(Kelurahan, out string kodePos))
27         {
28             return kodePos;
29         }
30         return "Kode pos tidak ditemukan";
31     }
32 }
```

DoorMachine.cs

```
DoorMachine.cs* KodePos.cs* Program.cs
TPMODUL3_2311104060 DoorMachine KunciPintu()
1 public class DoorMachine
2 {
3     // Enum to represent door states
4     private enum DoorState
5     {
6         Terkunci,
7         Terbuka
8     }
9
10     private DoorState currentState = DoorState.Terkunci;
11
12     // Method to simulate opening the door
13     public void BukaPintu()
14     {
15         if (currentState == DoorState.Terkunci)
16         {
17             currentState = DoorState.Terbuka;
18             Console.WriteLine("Pintu tidak terkunci");
19         }
20         else
21         {
22             Console.WriteLine("Pintu sudah terbuka");
23         }
24     }
25
26     // Method to simulate closing the door
27     public void KunciPintu()
28     {
29         if (currentState == DoorState.Terbuka)
30         {
31             currentState = DoorState.Terkunci;
32             Console.WriteLine("Pintu terkunci");
33         }
34         else
35         {
36             Console.WriteLine("Pintu sudah terkunci");
37         }
38     }
39 }
```

Program.cs

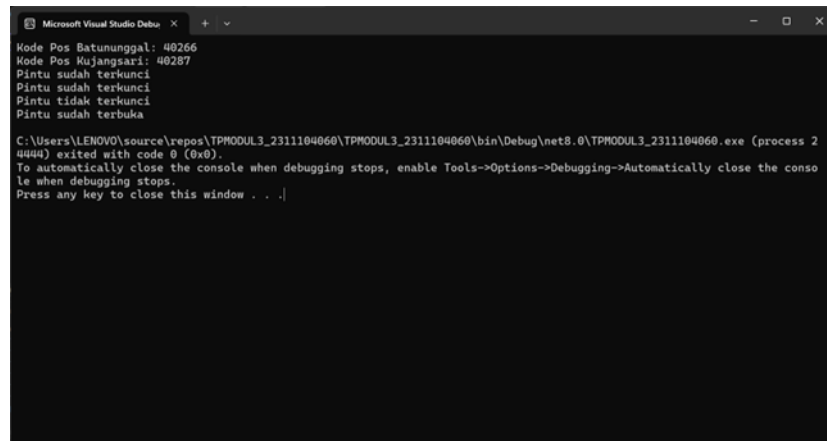
```
DoorMachine.cs KodePos.cs* Program.cs
TPMODUL3_2311104060 Program
1 class Program
2 {
3     static void Main(string[] args)
4     {
5         // Demonstrating KodePos (Table-Driven)
6         KodePos kodePos = new KodePos();
7         Console.WriteLine("Kode Pos Batununggal: " + kodePos.GetKodePos("Batununggal"));
8         Console.WriteLine("Kode Pos Kujangsari: " + kodePos.GetKodePos("Kujangsari"));
9
10        // Demonstrating DoorMachine (State-Based)
11        DoorMachine doorMachine = new DoorMachine();
12        doorMachine.KunciPintu(); // Should print "Pintu terkunci"
13        doorMachine.KunciPintu(); // Should print "Pintu sudah terkunci"
14        doorMachine.BukaPintu(); // Should print "Pintu tidak terkunci"
15        doorMachine.BukaPintu(); // Should print "Pintu sudah terbuka"
16    }
17 }
```

B. File docx/pdf yang berisi:

I. Link github repository

https://github.com/auliajsf06/tpmodul4_2311104060.

II. Screenshot hasil run (hasil console output dari table driven dan state-based)



```
Microsoft Visual Studio Debu
Kode Pos Batununggal: 40266
Kode Pos Kujangsari: 40287
Pintu sudah terkunci
Pintu sudah terkunci
Pintu tidak terkunci
Pintu sudah terbuka

C:\Users\LENOVO\source\repos\TPMODUL3_2311104060\TPMODUL3_2311104060\bin\Debug\net8.0\TPMODUL3_2311104060.exe (process 24444) 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 . . .|
```

III. Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).

a. Kelas KodePos (Implementasi Table-Driven)

Kelas KodePos dirancang untuk menangani pencarian kode pos menggunakan pendekatan table-driven, yang merupakan teknik efisien untuk memetakan data. Di dalam kelas ini, sebuah Dictionary digunakan untuk menyimpan pasangan nama kelurahan dan kode pos. Struktur data dictionary memungkinkan pencarian yang cepat dan akurat dengan kompleksitas waktu $O(1)$, artinya waktu pencarian tetap konstan tidak peduli berapa banyak data yang disimpan. Method `GetKodePos()` berperan sebagai mekanisme utama untuk mengekstrak kode pos, di mana ia menerima nama kelurahan sebagai parameter input. Ketika method dipanggil, ia akan mencoba mengambil kode pos dari dictionary menggunakan metode `TryGetValue()`, yang merupakan pendekatan aman untuk mengakses data. Jika kelurahan ditemukan, method akan mengembalikan kode pos yang sesuai; sebaliknya, jika tidak ditemukan, method akan mengembalikan pesan "Kode pos tidak ditemukan" sebagai indikator bahwa data tidak tersedia.

b. Kelas DoorMachine (Implementasi State-Based)

Kelas DoorMachine mengimplementasikan konsep finite state machine (mesin state terbatas) untuk mensimulasikan perilaku pintu dengan menggunakan enum dan perubahan state. Enum `DoorState` mendefinisikan dua kemungkinan kondisi pintu: `Terkunci` dan `Terbuka`, yang membatasi transisi state hanya pada kondisi yang valid. State awal pintu selalu dimulai dari kondisi terkunci. Method `BukaPintu()` dirancang untuk membuka pintu hanya ketika pintu dalam keadaan terkunci,

mencegah pembukaan pintu yang sudah terbuka. Sebaliknya, method `KunciPintu()` hanya memperbolehkan penguncian pintu yang sedang terbuka. Setiap perubahan state disertai dengan pesan konsol yang memberikan umpan balik tentang aksi yang dilakukan, seperti "Pintu terkunci" atau "Pintu tidak terkunci". Pendekatan state-based ini memastikan bahwa transisi antara state pintu selalu mengikuti aturan logika yang telah ditentukan, mencegah terjadinya kondisi state yang tidak valid.

c. Metode `Main()` dan Demonstrasi

Metode `Main()` berfungsi sebagai titik masuk program dan mendemonstrasikan penggunaan praktis dari kedua kelas yang telah dibuat. Untuk `KodePos`, method ini menunjukkan cara mengambil kode pos untuk beberapa kelurahan, memperlihatkan bagaimana dictionary dapat dengan mudah diakses untuk mendapatkan informasi. Pada bagian `DoorMachine`, method `Main()` mensimulasikan serangkaian interaksi dengan pintu, menampilkan bagaimana state berubah dan pesan yang dihasilkan berdasarkan urutan pemanggilan method `KunciPintu()` dan `BukaPintu()`. Demonstrasi ini tidak hanya menunjukkan fungsionalitas kode, tetapi juga memberikan contoh konkret tentang bagaimana konsep table-driven dan state-based dapat diimplementasikan dalam pemrograman praktis.