

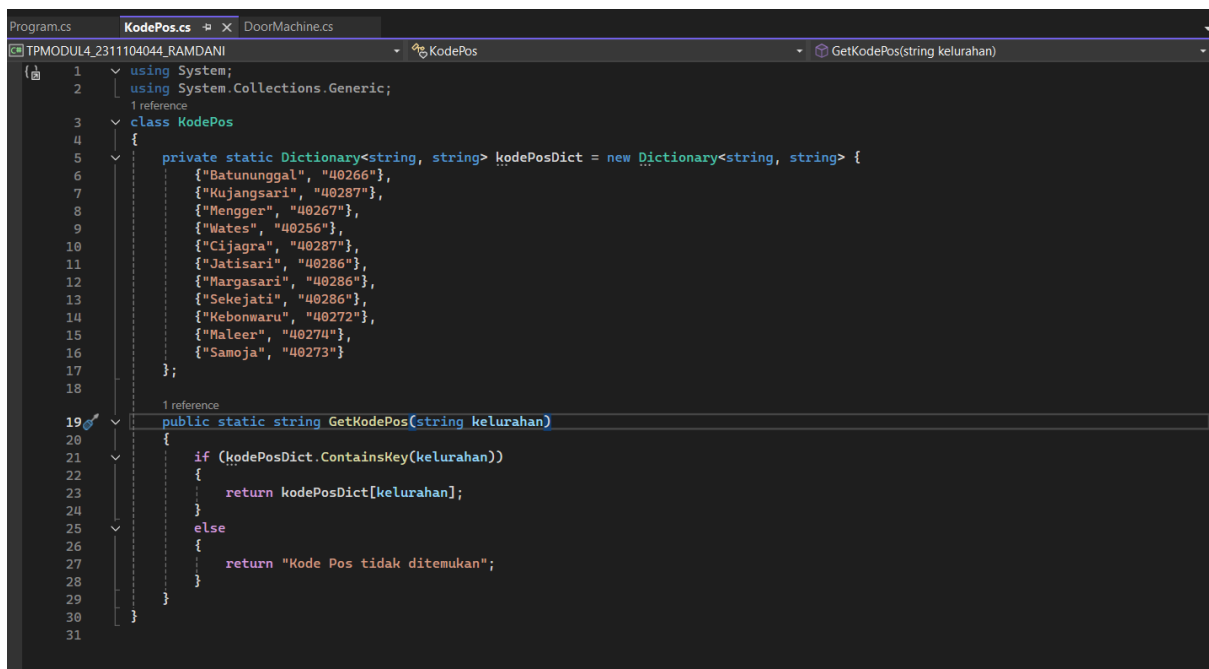
TUGAS PENDAHULUAN 4

Nama: Rifqi Mohamad Ramdani

NIM : 2311104044

Kelas: SE-07-02

Beri nama proyek lalu buat file dan diberi nama class KodePos.cs



```
Program.cs | KodePos.cs | DoorMachine.cs
TPMODUL4_2311104044_RAMDANI | KodePos | GetKodePos(string kelurahan)

1  using System;
2  using System.Collections.Generic;
3  class KodePos
4  {
5      private static Dictionary<string, string> kodePosDict = new Dictionary<string, string> {
6          {"Batununggal", "40266"},
7          {"Kujangsari", "40287"},
8          {"Mengger", "40267"},
9          {"Wates", "40256"},
10         {"Cijagra", "40287"},
11         {"Jatisari", "40286"},
12         {"Margasari", "40286"},
13         {"Sekejati", "40286"},
14         {"Kebonwaru", "40272"},
15         {"Maleer", "40274"},
16         {"Samoja", "40273"}
17     };
18
19     public static string GetKodePos(string kelurahan)
20     {
21         if (kodePosDict.ContainsKey(kelurahan))
22         {
23             return kodePosDict[kelurahan];
24         }
25         else
26         {
27             return "Kode Pos tidak ditemukan";
28         }
29     }
30 }
31
```

Menggunakan Dictionary<string, string> untuk menyimpan daftar kelurahan dan kode posnya.

Method GetKodePos():

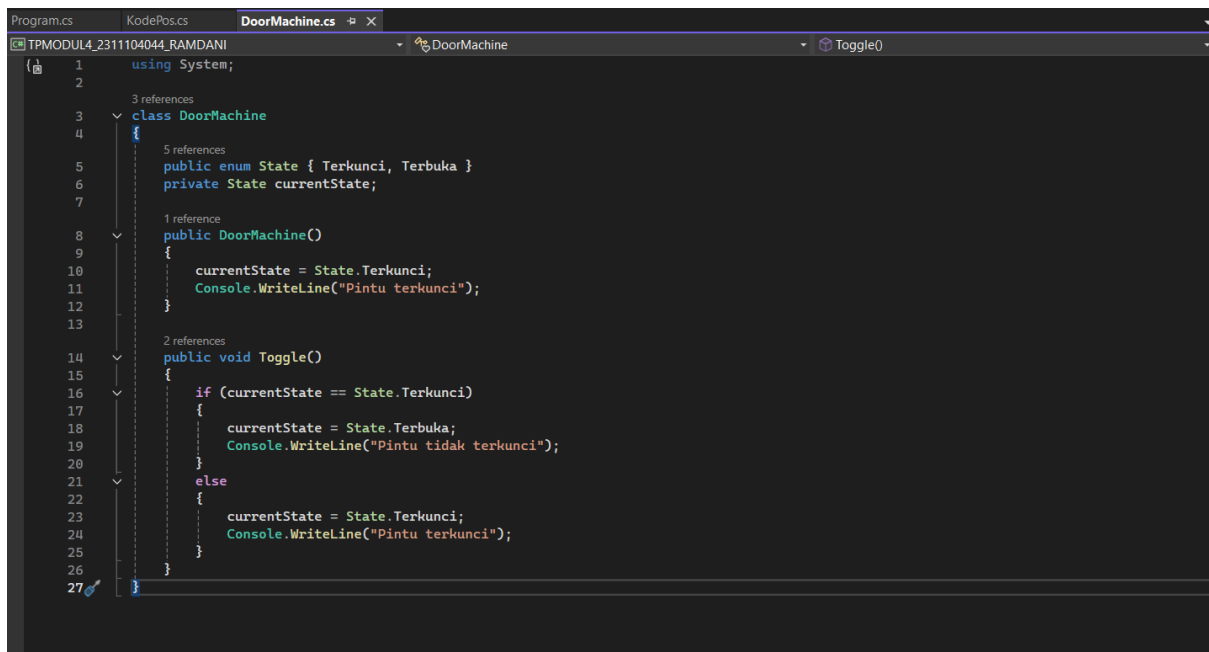
Mencari kode pos berdasarkan input kelurahan.

Jika ditemukan, mengembalikan kode pos.

Jika tidak ditemukan, mengembalikan pesan "Kode Pos tidak ditemukan".

Menggunakan teknik Table-Driven untuk menghindari banyak if-else, sehingga lebih rapi dan efisien.

Lalu membuat file lg dengan nama class DoorMachine.cs



```
1 using System;
2
3 class DoorMachine
4 {
5     public enum State { Terkunci, Terbuka }
6     private State currentState;
7
8     public DoorMachine()
9     {
10         currentState = State.Terkunci;
11         Console.WriteLine("Pintu terkunci");
12     }
13
14     public void Toggle()
15     {
16         if (currentState == State.Terkunci)
17         {
18             currentState = State.Terbuka;
19             Console.WriteLine("Pintu tidak terkunci");
20         }
21         else
22         {
23             currentState = State.Terkunci;
24             Console.WriteLine("Pintu terkunci");
25         }
26     }
27 }
```

Menggunakan enum State { Terkunci, Terbuka } untuk menyimpan dua kondisi pintu:

Terkunci → Default saat pintu pertama kali dibuat.

Terbuka → Jika pintu dibuka.

Konstruktor DoorMachine()

Saat objek pertama kali dibuat, statusnya Terkunci.

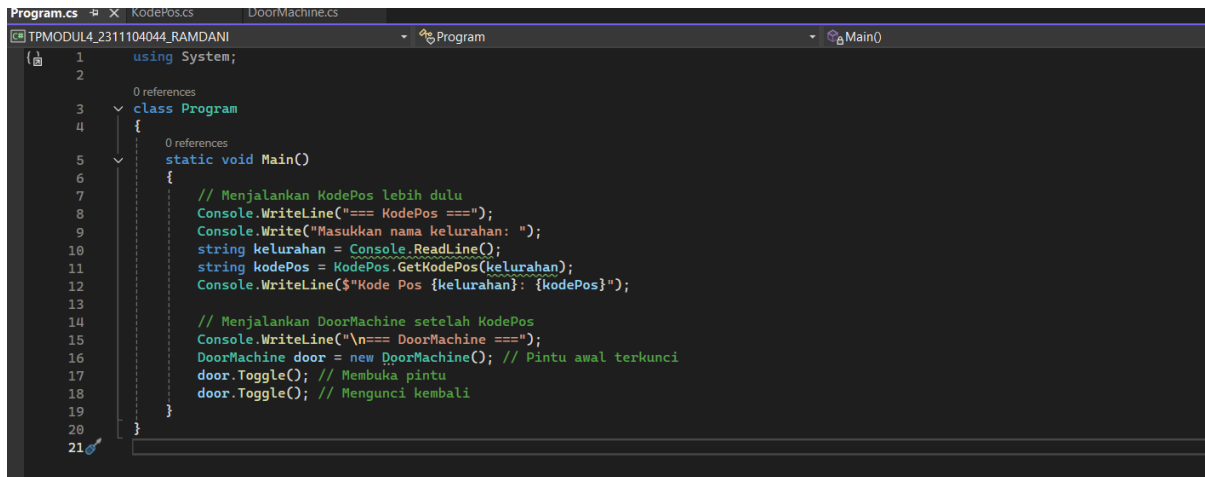
Method Toggle()

Jika Terkunci, maka berubah menjadi Terbuka dan menampilkan "Pintu tidak terkunci".

Jika Terbuka, maka berubah menjadi Terkunci dan menampilkan "Pintu terkunci".

Menggunakan State-Based Construction untuk menangani perubahan status pintu tanpa menggunakan banyak if-else secara manual.

Selanjutnya class Program.cs



```
1 using System;
2
3 class Program
4 {
5     static void Main()
6     {
7         // Menjalankan KodePos lebih dulu
8         Console.WriteLine("=== KodePos ===");
9         Console.Write("Masukkan nama kelurahan: ");
10        string kelurahan = Console.ReadLine();
11        string kodePos = KodePos.GetKodePos(kelurahan);
12        Console.WriteLine($"Kode Pos {kelurahan}: {kodePos}");
13
14        // Menjalankan DoorMachine setelah KodePos
15        Console.WriteLine("\n=== DoorMachine ===");
16        DoorMachine door = new DoorMachine(); // Pintu awal terkunci
17        door.Toggle(); // Membuka pintu
18        door.Toggle(); // Mengunci kembali
19    }
20 }
21
```

Menjalankan fitur KodePos (Table-Driven Method)

Program meminta pengguna memasukkan nama kelurahan.

Menggunakan KodePos.GetKodePos(kelurahan) untuk mencari kode pos.

Menampilkan hasil kode pos ke layar.

Menjalankan fitur DoorMachine (State-Based Construction)

Membuat objek DoorMachine, yang secara default akan menampilkan "Pintu terkunci".

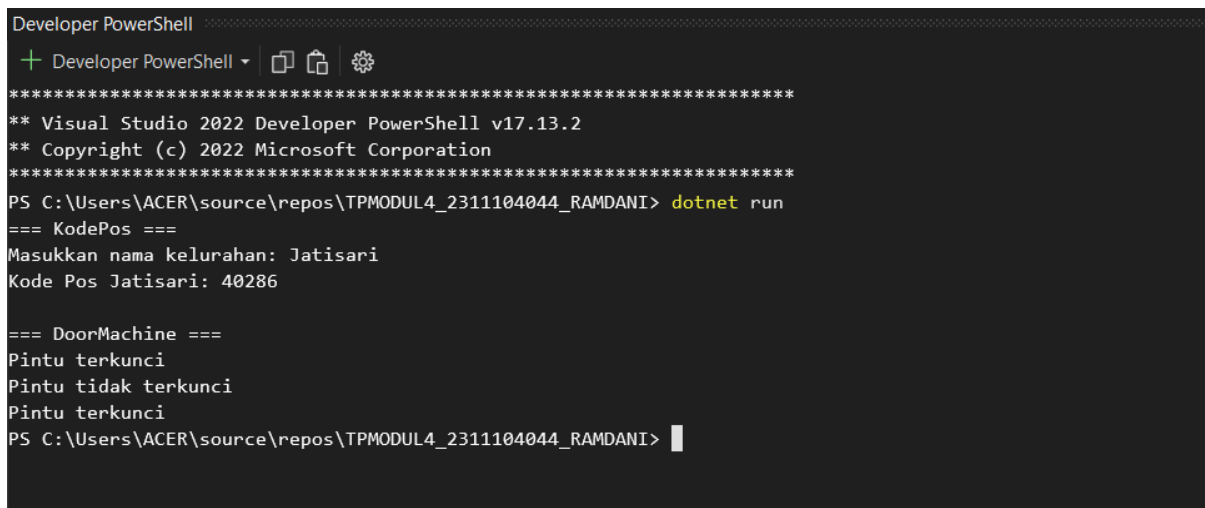
Memanggil Toggle() dua kali:

Pertama kali → Membuka pintu.

Kedua kali → Mengunci pintu kembali.

Menggunakan input dari pengguna (Console.ReadLine()) agar program lebih interaktif.

Maka akan menghasilkan outputnya



```
Developer PowerShell
+ Developer PowerShell v17.13.2
** Visual Studio 2022 Developer PowerShell v17.13.2
** Copyright (c) 2022 Microsoft Corporation
PS C:\Users\ACER\source\repos\TPMODUL4_2311104044_RAMDANI> dotnet run
=== KodePos ===
Masukkan nama kelurahan: Jatisari
Kode Pos Jatisari: 40286

=== DoorMachine ===
Pintu terkunci
Pintu tidak terkunci
Pintu terkunci
PS C:\Users\ACER\source\repos\TPMODUL4_2311104044_RAMDANI>
```

Kesimpulan

KodePos.cs → Menggunakan Table-Driven Method untuk menyimpan dan mencari kode pos.

DoorMachine.cs → Menggunakan State-Based Construction untuk mengelola perubahan status pintu.

Program.cs → Menggabungkan dua fitur di atas dan menerima input dari pengguna.

Program ini lebih efisien karena menggunakan Table-Driven dan State-Based Construction.

Output program dinamis karena bisa menerima input dari pengguna.

Kode lebih mudah dibaca dan dikembangkan di masa depan.