

Nama: Ahmad Al - Farizi

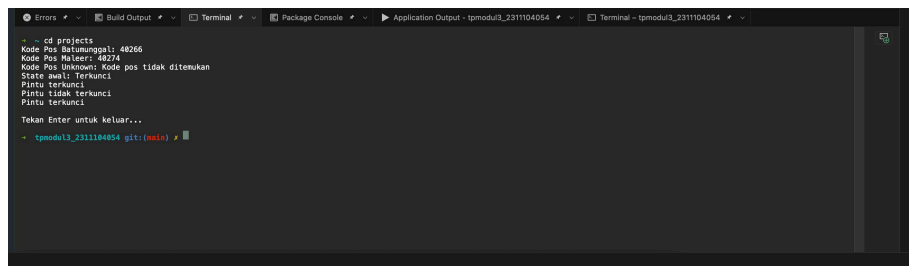
NIM: 2311104054

Kelas: SE-07-02

## TP 4

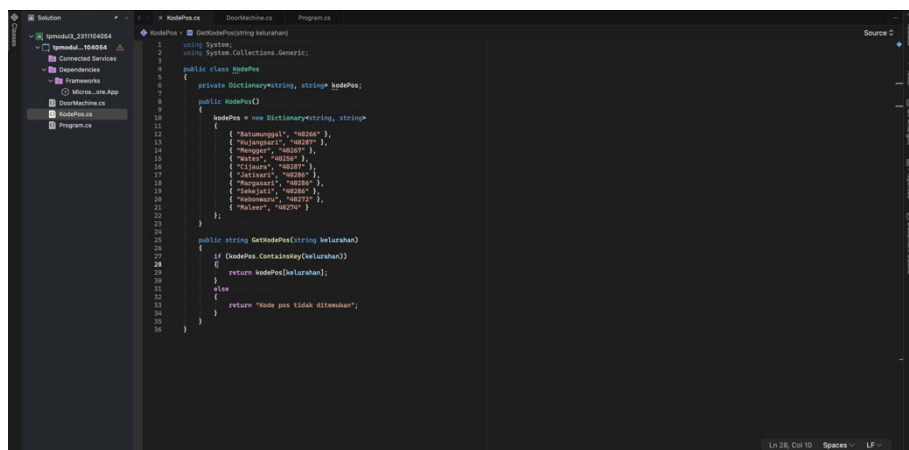
i. <https://github.com/alxfarizi/tpmodul3.git>

ii.



```
cd projects
Kode Pos Batununggal: 48266
Kode Pos Maleer: 48274
Kode Pos Unknown: Kode pos tidak ditemukan
State awal: terkunci
Pintu terkunci
Pintu tidak terkunci
Pintu terkunci
Tekan Enter untuk keluar...
tpmodul3_2311104054 git:(main) %
```

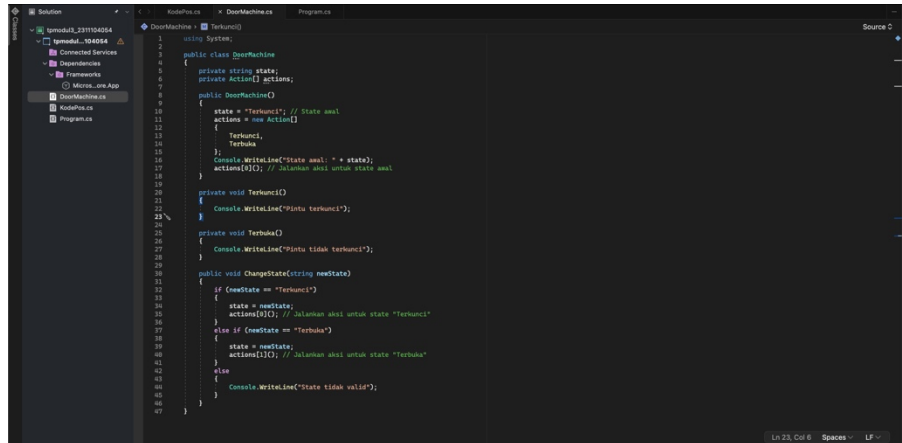
iii. Kelas KodePos bertugas untuk mengelola data kode pos berdasarkan nama kelurahan. Di dalam kelas ini, data kode pos disimpan dalam sebuah Dictionary yang memetakan nama kelurahan ke kode posnya. Kelas ini memiliki sebuah method bernama GetKodePos yang menerima nama kelurahan sebagai parameter dan mengembalikan kode pos yang sesuai. Jika nama kelurahan tidak ditemukan dalam Dictionary, method akan mengembalikan pesan "Kode pos tidak ditemukan". Kelas ini digunakan untuk memudahkan pencarian kode pos berdasarkan input pengguna.



```
Solution
  tpmodul3_2311104054
    tpmodul3_2311104054
      Dependencies
      Frameworks
      tpmodul3_2311104054
        KodePos.cs
        Program.cs

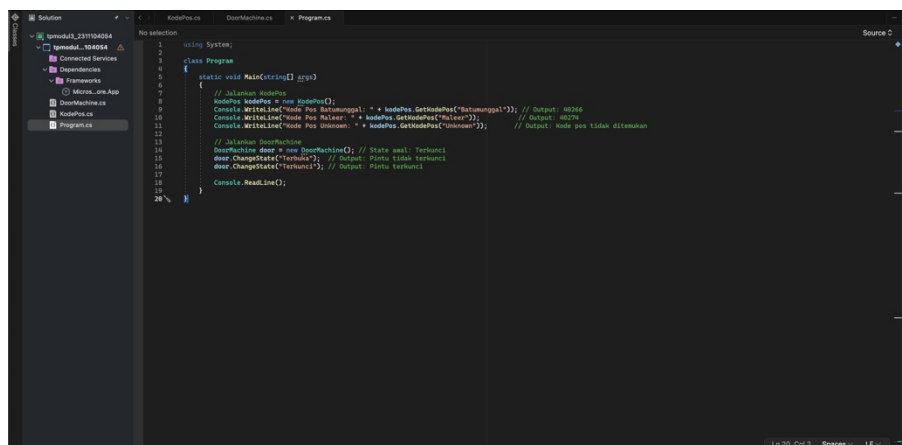
KodePos.cs
1 using System;
2 using System.Collections.Generic;
3
4 public class KodePos
5 {
6     private Dictionary<string, string> kodePos;
7
8     public KodePos()
9     {
10         kodePos = new Dictionary<string, string>
11         {
12             { "Batununggal", "48266" },
13             { "Mojangari", "48287" },
14             { "Mongger", "48287" },
15             { "Mendes", "48288" },
16             { "Cijaur", "48287" },
17             { "Sulitair", "48288" },
18             { "Margasari", "48288" },
19             { "Senejati", "48288" },
20             { "Seneasari", "48272" },
21             { "Maleer", "48274" }
22         };
23     }
24
25     public string GetKodePos(string kelurahan)
26     {
27         if (kodePos.ContainsKey(kelurahan))
28         {
29             return kodePos[kelurahan];
30         }
31         else
32         {
33             return "Kode pos tidak ditemukan";
34         }
35     }
36 }
```

Kelas DoorMachine mensimulasikan sebuah mesin pintu yang memiliki dua state, yaitu "Terkunci" dan "Terbuka". Kelas ini menggunakan teknik state-based construction untuk mengelola perubahan state pintu. Saat objek DoorMachine dibuat, state awalnya adalah "Terkunci", dan pesan "Pintu terkunci" akan ditampilkan ke console. Kelas ini memiliki method ChangeState yang digunakan untuk mengubah state pintu. Jika state diubah ke "Terbuka", pesan "Pintu tidak terkunci" akan ditampilkan, dan jika diubah kembali ke "Terkunci", pesan "Pintu terkunci" akan ditampilkan. Kelas ini membantu mensimulasikan perilaku pintu berdasarkan perubahan state.



```
1 using System;
2
3 public class DoorMachine
4 {
5     private string state;
6     private Action[] actions;
7
8     public DoorMachine()
9     {
10         state = "Terkunci"; // State awal
11         actions = new Action[]
12         {
13             Terkunci,
14             Terbuka
15         };
16         Console.WriteLine("State awal: " + state);
17         actions[0](); // Jalankan aksi untuk state awal
18     }
19
20     private void Terkunci()
21     {
22         Console.WriteLine("Pintu terkunci");
23     }
24
25     private void Terbuka()
26     {
27         Console.WriteLine("Pintu tidak terkunci");
28     }
29
30     public void ChangeState(string newState)
31     {
32         if (newState == "Terkunci")
33         {
34             state = newState;
35             actions[0](); // Jalankan aksi untuk state "Terkunci"
36         }
37         else if (newState == "Terbuka")
38         {
39             state = newState;
40             actions[1](); // Jalankan aksi untuk state "Terbuka"
41         }
42         else
43         {
44             Console.WriteLine("State tidak valid");
45         }
46     }
47 }
```

Kelas Program merupakan entry point dari aplikasi dan bertugas untuk menjalankan logika utama program. Di dalam kelas ini, method Main adalah titik awal eksekusi program. Pada method Main, objek dari kelas KodePos dan DoorMachine dibuat. Objek KodePos digunakan untuk menampilkan kode pos berdasarkan nama kelurahan, sementara objek DoorMachine digunakan untuk mensimulasikan perubahan state pintu. Setelah semua logika dijalankan, program menggunakan Console.ReadLine() untuk menahan console agar tidak langsung tertutup, memungkinkan pengguna melihat hasil output sebelum menutup aplikasi. Kelas ini bertindak sebagai pengendali utama yang mengintegrasikan fungsi dari KodePos dan DoorMachine.



```
1 using System;
2
3 class Program
4 {
5     static void Main(string[] args)
6     {
7         // Jalankan KodePos
8         KodePos kodePos = new KodePos();
9         Console.WriteLine("Kode Pos Batununggal: " + kodePos.GetKodePos("Batununggal")); // Output: 08266
10        Console.WriteLine("Kode Pos Rajer: " + kodePos.GetKodePos("Rajer")); // Output: 08270
11        Console.WriteLine("Kode Pos lainnya: " + kodePos.GetKodePos("Cincong")); // Output: kode pos tidak ditemukan
12
13        // Jalankan DoorMachine
14        DoorMachine door = new DoorMachine(); // State awal: Terkunci
15        door.ChangeState("Terbuka"); // Output: Pintu tidak terkunci
16        door.ChangeState("Terkunci"); // Output: Pintu terkunci
17        Console.ReadLine();
18    }
19 }
```

Hubungan antar kelas:

Kelas KodePos dan DoorMachine adalah dua komponen terpisah yang masing-masing memiliki tanggung jawab spesifik. KodePos fokus pada pengelolaan data kode pos, sedangkan DoorMachine fokus pada simulasi state pintu. Kedua kelas ini diintegrasikan dalam kelas Program, yang bertindak sebagai pengendali utama. Di dalam Program, objek dari kedua kelas tersebut dibuat dan method-method mereka dipanggil untuk menampilkan output ke console. Dengan struktur ini, program menjadi modular dan mudah dikembangkan lebih lanjut.