

A. Source Code

- **KodePos.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace no4
{
    class KodePos
    {
        // Dictionary untuk menyimpan daftar kode pos berdasarkan
        // kelurahan
        private static Dictionary<string, string> kodePosTable = new
        Dictionary<string, string>
        {
            { "Batununggal", "40266" },
            { "Kujangsari", "40287" },
            { "Mengger", "40267" },
            { "Wates", "40256" },
            { "Cijaura", "40287" },
            { "Jatisari", "40286" },
            { "Margasari", "40286" },
            { "Sekejati", "40286" },
            { "Kebonwaru", "40272" },
            { "Maleer", "40274" },
            { "Samoja", "40273" }
        };

        // Method untuk mendapatkan kode pos berdasarkan kelurahan
        public static string getCodePos(string? kelurahan)
        {
            // Jika kelurahan null atau kosong, kembalikan pesan error
            if (string.IsNullOrEmpty(kelurahan))
            {
                return "Nama kelurahan tidak boleh kosong.";
            }

            // Cek apakah kelurahan ada dalam Dictionary
            if (kodePosTable.ContainsKey(kelurahan))
```

```

        {
            return kodePosTable[kelurahan]; // Kembalikan kode pos jika
ditemukan
        }
        else
        {
            return "Kode Pos tidak ditemukan."; // Jika tidak ditemukan,
berikan pesan ini
        }
    }
}

```

Keterangan:

Kode di atas adalah implementasi class KodePos dalam bahasa C# yang berfungsi untuk mencari kode pos berdasarkan nama kelurahan menggunakan Dictionary. Dictionary kodePosTable menyimpan pasangan nama kelurahan sebagai key dan kode pos sebagai value. Method getCodePos(string? kelurahan) digunakan untuk mengambil kode pos dari kelurahan yang dimasukkan. Jika input kosong (null atau ""), program akan mengembalikan pesan "Nama kelurahan tidak boleh kosong.". Jika kelurahan ditemukan dalam Dictionary, program akan mengembalikan kode pos yang sesuai, sedangkan jika tidak ditemukan, program akan menampilkan "Kode Pos tidak ditemukan.". Struktur ini memastikan program tetap robust dan mudah digunakan untuk pencarian kode pos secara dinamis.

- **DoorMachine.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace no4
{
    using System;

    class DoorMachine
    {
        public enum State { Terkunci, Terbuka }

        private State currentState;

        public DoorMachine()

```

```

        {
            currentState = State.Terkunci;
        }

        public void BukaPintu()
        {
            if (currentState == State.Terkunci)
            {
                currentState = State.Terbuka;
                Console.WriteLine("Pintu terbuka");
            }
        }

        public void KunciPintu()
        {
            if (currentState == State.Terbuka)
            {
                currentState = State.Terkunci;
                Console.WriteLine("Pintu terkunci");
            }
        }
    }
}

```

Keterangan:

Kode di atas merupakan implementasi class DoorMachine dalam C# yang menerapkan state-based construction untuk mensimulasikan mekanisme buka dan kunci pintu. Class ini memiliki enum State dengan dua keadaan: Terkunci dan Terbuka, serta atribut currentState untuk menyimpan kondisi pintu saat ini. Saat objek DoorMachine dibuat, pintu dalam keadaan terkunci secara default. Method BukaPintu() digunakan untuk membuka pintu jika dalam keadaan terkunci, sedangkan KunciPintu() digunakan untuk mengunci pintu jika dalam keadaan terbuka. Setiap perubahan state akan menghasilkan output ke console, memastikan pengguna dapat melihat status pintu secara real-time

- Program.cs

```

using no4;
using System;

class Program
{
    static void Main()
    {
        DoorMachine pintu = new DoorMachine();
        pintu.BukaPintu();
    }
}

```

```

        pintu.KunciPintu();
    }
}

```

Keterangan:

Kode di atas adalah program utama yang menggunakan class DoorMachine dari namespace no4 untuk mensimulasikan mekanisme buka dan kunci pintu. Program pertama-tama membuat objek pintu dari class DoorMachine, yang secara default berada dalam keadaan terkunci. Kemudian, method BukaPintu() dipanggil untuk mengubah status pintu menjadi terbuka, diikuti dengan pemanggilan KunciPintu() untuk mengembalikan statusnya menjadi terkunci. Output pada console akan menunjukkan perubahan status pintu secara berurutan, memastikan bahwa mekanisme state-based construction bekerja dengan benar.

B. Hasil Run

- Table driven

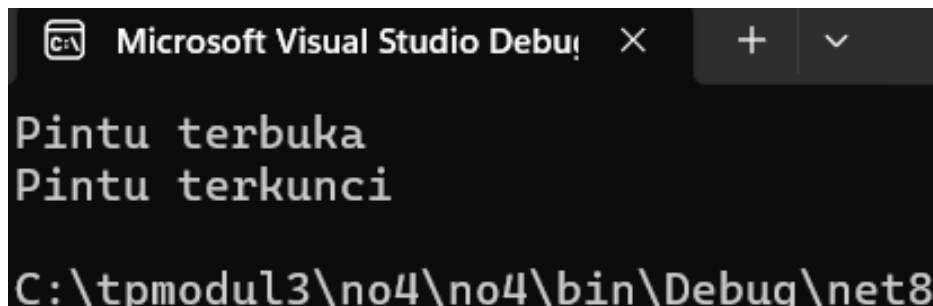


```

Masukkan nama kelurahan: Wates
Kode pos Wates: 40256

```

- Stase Based



```

Pintu terbuka
Pintu terkunci
C:\tpmodul3\no4\no4\bin\Debug\net8

```

C. Screenshot Proyek

- Kodepos.cs

```
1  // Kodepos.cs
2
3  // Dictionary untuk menyimpan daftar kode pos berdasarkan kabupaten
4  private static Dictionary<string, string> kodePosTable = new Dictionary<string, string>();
5
6  // Inisialisasi data
7  static void Main()
8  {
9      // Data Kabupaten
10     List<string> kabupaten = new List<string> {
11         "Kabupaten", "Kodepos",
12         "Kabupaten", "Kodepos",
13         "Kabupaten", "Kodepos",
14         "Kabupaten", "Kodepos",
15         "Kabupaten", "Kodepos",
16         "Kabupaten", "Kodepos",
17         "Kabupaten", "Kodepos",
18         "Kabupaten", "Kodepos",
19         "Kabupaten", "Kodepos",
20         "Kabupaten", "Kodepos",
21         "Kabupaten", "Kodepos",
22         "Kabupaten", "Kodepos",
23         "Kabupaten", "Kodepos",
24         "Kabupaten", "Kodepos",
25         "Kabupaten", "Kodepos",
26         "Kabupaten", "Kodepos",
27         "Kabupaten", "Kodepos",
28         "Kabupaten", "Kodepos",
29         "Kabupaten", "Kodepos",
30         "Kabupaten", "Kodepos",
31         "Kabupaten", "Kodepos",
32         "Kabupaten", "Kodepos",
33         "Kabupaten", "Kodepos",
34         "Kabupaten", "Kodepos",
35         "Kabupaten", "Kodepos",
36         "Kabupaten", "Kodepos",
37         "Kabupaten", "Kodepos",
38         "Kabupaten", "Kodepos",
39         "Kabupaten", "Kodepos",
40         "Kabupaten", "Kodepos",
41         "Kabupaten", "Kodepos",
42         "Kabupaten", "Kodepos",
43         "Kabupaten", "Kodepos",
44         "Kabupaten", "Kodepos",
45         "Kabupaten", "Kodepos",
46         "Kabupaten", "Kodepos",
47         "Kabupaten", "Kodepos",
48         "Kabupaten", "Kodepos",
49         "Kabupaten", "Kodepos",
50         "Kabupaten", "Kodepos",
51         "Kabupaten", "Kodepos",
52         "Kabupaten", "Kodepos",
53         "Kabupaten", "Kodepos",
54         "Kabupaten", "Kodepos",
55         "Kabupaten", "Kodepos",
56         "Kabupaten", "Kodepos",
57         "Kabupaten", "Kodepos",
58         "Kabupaten", "Kodepos",
59         "Kabupaten", "Kodepos",
60         "Kabupaten", "Kodepos",
61         "Kabupaten", "Kodepos",
62         "Kabupaten", "Kodepos",
63         "Kabupaten", "Kodepos",
64         "Kabupaten", "Kodepos",
65         "Kabupaten", "Kodepos",
66         "Kabupaten", "Kodepos",
67         "Kabupaten", "Kodepos",
68         "Kabupaten", "Kodepos",
69         "Kabupaten", "Kodepos",
70         "Kabupaten", "Kodepos",
71         "Kabupaten", "Kodepos",
72         "Kabupaten", "Kodepos",
73         "Kabupaten", "Kodepos",
74         "Kabupaten", "Kodepos",
75         "Kabupaten", "Kodepos",
76         "Kabupaten", "Kodepos",
77         "Kabupaten", "Kodepos",
78         "Kabupaten", "Kodepos",
79         "Kabupaten", "Kodepos",
80         "Kabupaten", "Kodepos",
81         "Kabupaten", "Kodepos",
82         "Kabupaten", "Kodepos",
83         "Kabupaten", "Kodepos",
84         "Kabupaten", "Kodepos",
85         "Kabupaten", "Kodepos",
86         "Kabupaten", "Kodepos",
87         "Kabupaten", "Kodepos",
88         "Kabupaten", "Kodepos",
89         "Kabupaten", "Kodepos",
90         "Kabupaten", "Kodepos",
91         "Kabupaten", "Kodepos",
92         "Kabupaten", "Kodepos",
93         "Kabupaten", "Kodepos",
94         "Kabupaten", "Kodepos",
95         "Kabupaten", "Kodepos",
96         "Kabupaten", "Kodepos",
97         "Kabupaten", "Kodepos",
98         "Kabupaten", "Kodepos",
99         "Kabupaten", "Kodepos",
100    };
101
102    // Method untuk mendapatkan kode pos berdasarkan kabupaten
103    public static string GetKodepos(string kabupaten)
104    {
105        // Jika kabupaten null atau kosong, kembalikan pesan error
106        if (string.IsNullOrEmpty(kabupaten))
107        {
108            return "Nama kabupaten tidak boleh kosong.";
109        }
110
111        // Cari apakah kabupaten ada dalam Dictionary
112        if (kodePosTable.ContainsKey(kabupaten))
113        {
114            return kodePosTable[kabupaten]; // kembalikan kode pos jika ditemukan
115        }
116        else
117        {
118            return "Kode pos tidak ditemukan."; // jika tidak ditemukan, kembalikan pesan ini
119        }
120    }
121
122    }
```

- DoorMachine.cs

```
1  // DoorMachine.cs
2
3  // DoorMachine
4  {
5      // 0 references
6      public enum State { Tertutup, Terbuka }
7
8      // 1 reference
9      private State currentState;
10
11      // 1 reference
12      public DoorMachine()
13      {
14          currentState = State.Tertutup;
15      }
16
17      // 1 reference
18      public void BukaPintu()
19      {
20          if (currentState == State.Tertutup)
21          {
22              currentState = State.Terbuka;
23              Console.WriteLine("Pintu terbuka");
24          }
25      }
26
27      // 1 reference
28      public void KunciPintu()
29      {
30          if (currentState == State.Terbuka)
31          {
32              currentState = State.Tertutup;
33              Console.WriteLine("Pintu terkunci");
34          }
35      }
36
37      }
38
39  }
```

- Program.cs

```
1  // Program.cs
2
3  // class Program
4  {
5      // 0 references
6      static void Main()
7      {
8          DoorMachine pintu = new DoorMachine();
9          pintu.BukaPintu();
10         pintu.KunciPintu();
11     }
12 }
```