

## LAPORAN PRAKTIKUM TP 4

2311104072 – Jauhar Fajar Zuhair

---

### Bagian I: Interface & Kelas KodePos

Kode Inti:

```
public interface IKodePos
{
    int? GetKodePos(string kelurahan);
}

public class KodePos : IKodePos
{
    private readonly Dictionary<string, int> kodePosTable;

    public KodePos()
    {
        kodePosTable = new Dictionary<string,
int>(StringComparer.OrdinalIgnoreCase)
        {
            { "Batununggal", 40266 },
            { "A. Kujangsari", 40287 },
            { "Mengger", 40267 },
            { "Wates", 40256 },
            { "Cijaura", 40287 },
            { "Jatisari", 40286 },
            { "Margasari", 40286 },
            { "Sekejati", 40286 },
            { "Kebonwaru", 40272 },
            { "Maleer", 40274 },
            { "Samoja", 40273 }
        };
    }

    public int? GetKodePos(string kelurahan)
    {
        if (kodePosTable.TryGetValue(kelurahan, out int kodePos))
        {
            return kodePos;
        }
        return null;
    }
}
```

Penjelasan Singkat:

- **IKodePos:** Interface dengan metode `GetKodePos` yang mengembalikan `int?`.
- **KodePos:** Implementasi interface dengan `Dictionary` untuk menyimpan pasangan kelurahan–kode pos. Metode `GetKodePos` mengambil kode pos sesuai input kelurahan.

---

## Bagian II: Enum dan Kelas DoorMachine

### Kode Inti:

```
public enum DoorState
{
    Terkunci,
    Terbuka
}
```

```
public class DoorMachine
{
    private DoorState _currentState;
    public DoorMachine()
    {
        _currentState = DoorState.Terkunci;
        Console.WriteLine("Pintu terkunci");
    }
    public void BukaPintu()
    {
        if (_currentState == DoorState.Terkunci)
        {
            _currentState = DoorState.Terbuka;
            Console.WriteLine("Pintu tidak terkunci");
        }
        else
        {
            Console.WriteLine("Pintu sudah terbuka");
        }
    }
    public void KunciPintu()
    {
        if (_currentState == DoorState.Terbuka)
        {
            _currentState = DoorState.Terkunci;
            Console.WriteLine("Pintu terkunci");
        }
        else
        {
            Console.WriteLine("Pintu sudah terkunci");
        }
    }
}
```

### Penjelasan Singkat:

- **DoorState:** Enum untuk menandai kondisi pintu (Terkunci/Terbuka).
- **DoorMachine:** Kelas yang mengatur logika buka/tutup pintu.
  - **Konstruktor:** Mengawali pintu dalam keadaan terkunci.
  - **BukaPintu():** Mengubah status pintu dari terkunci menjadi terbuka.
  - **KunciPintu():** Mengubah status pintu dari terbuka menjadi terkunci.

---

## Bagian III: Main Program – Penggunaan KodePos

### Kode Inti:

```
IKodePos kodePosService = new KodePos();
Console.WriteLine("Masukkan nama kelurahan: ");
string kelurahanInput = Console.ReadLine();

int? kodePos = kodePosService.GetKodePos(kelurahanInput);
if (kodePos.HasValue)
{
    Console.WriteLine($"Kode pos untuk {kelurahanInput} adalah: {kodePos.Value}");
}
else
{
    Console.WriteLine("Kelurahan tidak ditemukan.");
}
```

### Penjelasan Singkat:

- **kodePosService:** Membuat instance dari `KodePos` yang mengimplementasikan `IKodePos`.
- **Input Kelurahan:** Meminta user memasukkan nama kelurahan.
- **Pencarian Kode Pos:** Memanggil `GetKodePos` dan menampilkan hasil jika ditemukan.

---

## Bagian IV: Main Program – Penggunaan DoorMachine

### Kode Inti:

```
DoorMachine doorMachine = new DoorMachine();

Console.WriteLine("Tekan ENTER untuk membuka pintu...");
Console.ReadLine();
doorMachine.BukaPintu();

Console.WriteLine("Tekan ENTER untuk mengunci pintu...");
Console.ReadLine();
doorMachine.KunciPintu();
```

### Penjelasan Singkat:

- **Inisialisasi:** Membuat objek `DoorMachine` (pintu awalnya terkunci).
- **Buka Pintu:** Menunggu input ENTER, lalu memanggil `BukaPintu()`.
- **Kunci Pintu:** Menunggu input ENTER, lalu memanggil `KunciPintu()`.