

LAPORAN PRAKTIKUM
Tugas Pendahuluan Modul 4
Statebased Table Driven Construction



Disusun Oleh:

Berlian Seva Astryana 2311104067

S1SE-07-02

Dosen :

Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

PURWOKERTO

2024

1. Penjelasan Sourcode

```
public class KodePos
{
    private static Dictionary<string, string> tabelKodePos = new Dictionary<string, string>()
    {
        {"Batununggal", "40266"},
        {"Kujangsari", "40287"},
        {"Mengger", "40267"},
        {"Wates", "40256"},
        {"Cijaura", "40287"},
        {"Jatisari", "40286"},
        {"Margasari", "40286"},
        {"Sekejati", "40286"},
        {"Kebonwaru", "40272"},
        {"Maleer", "40274"},
        {"Samojah", "40273"}
    };
}
```

Potongan kode di atas adalah implementasi teknik table-driven, di mana data kelurahan dan kode pos disimpan dalam sebuah Dictionary. Teknik ini mempermudah pencarian karena cukup menggunakan kunci (nama kelurahan), dan tidak perlu menulis banyak if-else. Penambahan atau penghapusan data juga menjadi lebih sederhana karena hanya mengubah isi tabel.

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

Enum DoorState digunakan untuk merepresentasikan dua kondisi pintu, yaitu Terkunci dan Terbuka. Enum ini membantu program untuk lebih terstruktur dalam menangani status, serta mempermudah implementasi logika kondisi di class utama.

```
public DoorMachine()
{
    stateSekarang = DoorState.Terkunci;
    Console.WriteLine("Pintu terkunci");
}
```

Constructor ini dijalankan saat objek DoorMachine dibuat. Pada awalnya, state pintu di-set ke Terkunci dan langsung mencetak pesan "Pintu terkunci". Ini sesuai instruksi bahwa state awal harus dalam keadaan terkunci.

```
public void GantiState(DoorState newState)
{
    stateSekarang = newState;

    if (stateSekarang == DoorState.Terkunci)
        Console.WriteLine("Pintu terkunci");
    else if (stateSekarang == DoorState.Terbuka)
        Console.WriteLine("Pintu tidak terkunci");
}
```

Method GantiState mengubah status pintu ke kondisi baru (Terkunci atau Terbuka) dan mencetak output yang sesuai dengan state tersebut. Ini adalah implementasi dari state-based construction, di mana setiap perubahan state menghasilkan efek (output) yang berbeda.

```
Console.WriteLine("Kode pos untuk Batununggal: " + KodePos.getKodePos("Batununggal"));
```

Baris ini merupakan pemanggilan method `getKodePos` dari class `KodePos`, untuk menampilkan kode pos dari kelurahan "Batununggal". Ini membuktikan bahwa data dari dictionary dapat diakses langsung dan digunakan di program utama.

```
DoorMachine pintu = new DoorMachine();  
pintu.GantiState(DoorState.Terbuka);  
pintu.GantiState(DoorState.Terkunci);
```

Potongan ini membuat objek `pintu` dan mensimulasikan dua kali perubahan state: dari `Terkunci` ke `Terbuka`, lalu kembali ke `Terkunci`. Setiap perubahan state menampilkan output di console sesuai kondisi, menunjukkan bahwa logika state berjalan dengan benar.

2. Output

```
Kode pos untuk Batununggal: 40266
```

```
Pintu terkunci  
Pintu tidak terkunci  
Pintu terkunci
```