

## **JURNAL MODUL 4**



**Nama :**

Dimastian Aji Wibowo (2311104058)

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

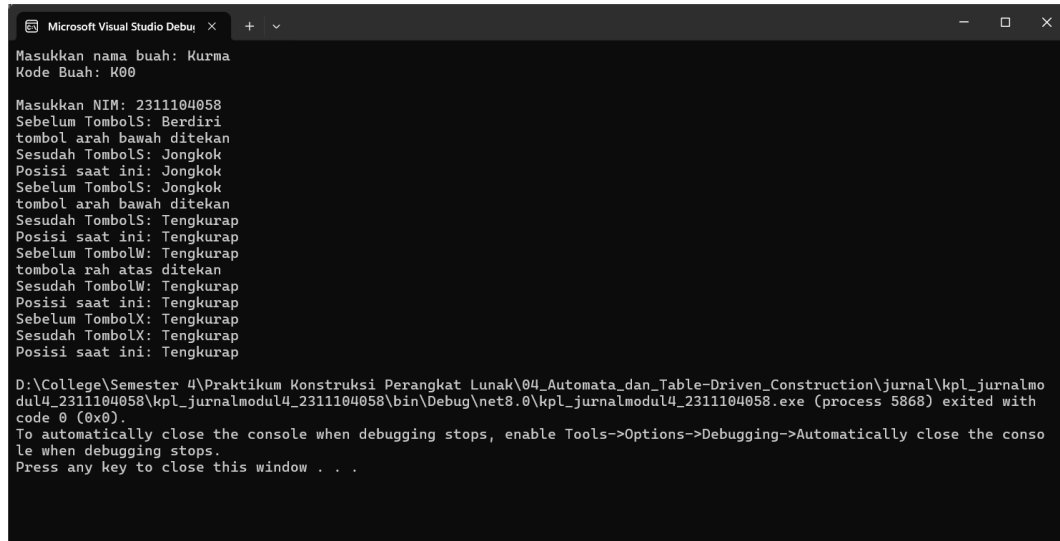
**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## Link Repository Github

[https://github.com/dimastianaji/kpl\\_jurnalmodul4\\_2311104058.git](https://github.com/dimastianaji/kpl_jurnalmodul4_2311104058.git)

## Hasil Run



```
Microsoft Visual Studio Debug
Masukkan nama buah: Kurma
Kode Buah: K00

Masukkan NIM: 2311104058
Sebelum TombolS: Berdiri
tombol arah bawah ditekan
Sesudah TombolS: Jongkok
Posisi saat ini: Jongkok
Sebelum TombolS: Jongkok
tombol arah bawah ditekan
Sesudah TombolS: Tengkurap
Posisi saat ini: Tengkurap
Sebelum TombolW: Tengkurap
tombola rah atas ditekan
Sesudah TombolW: Tengkurap
Posisi saat ini: Tengkurap
Sebelum TombolX: Tengkurap
Sesudah TombolX: Tengkurap
Posisi saat ini: Tengkurap

D:\College\Semester 4\Praktikum Konstruksi Perangkat Lunak\04_Automata_dan_Table-Driven_Construction\jurnal\kpl_jurnalmo
dul4_2311104058\kpl_jurnalmodul4_2311104058\bin\Debug\net8.0\kpl_jurnalmodul4_2311104058.exe (process 5868) exited with
code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

## Penjelasan

Class KodeBuah.cs

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

namespace kpl_jurnalmodul4_2311104058
{
    class KodeBuah
    {
        private Dictionary<string, string> kodeBuah = new
Dictionary<string, string>
        {
            {"Apel", "A00"},
            {"Aprikot", "B00"},
            {"Alpukat", "C00"},
            {"Pisang", "D00"},
            {"Paprika", "E00"},
            {"Blackberry", "F00"},
            {"Kurma", "K00"},
            {"Durian", "L00"},
            {"Anggur", "M00"},
            {"Melon", "N00"},
            {"Semangka", "O00"},
            {"Ceri", "H00"},
            {"Kelapa", "I00"},
            {"Jagung", "J00"}
        };
    }
}
```

```

        public string getKodeBuah(string namaBuah)
        {
            if (kodeBuah.ContainsKey(namaBuah))
            {
                return kodeBuah[namaBuah];
            }
            else
            {
                return "Kode tidak ditemukan";
            }
        }
    }
}

```

Kelas ini memiliki atribut private Dictionary bernama kodeBuah yang berisi pasangan nama buah (key) dan kode buah (value). Method getKodeBuah(string namaBuah) yang berfungsi untuk menerima nama buah sebagai parameter dan memeriksa apakah nama buah tersebut ada dalam Dictionary, jika ada maka mengembalikan kode buah yang sesuai dan jika tidak ada maka mengembalikan string "Kode tidak ditemukan".

Class PosisiKarakterGame.cs

```

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

namespace kp1_jurnalmodul4_2311104058
{
    public enum State
    {
        Berdiri,
        Jongkok,
        Tengkurap,
        Terbang
    }

    public class PosisiKarakterGame
    {
        private State currentState;
        private int nim;

        public PosisiKarakterGame(int nim)
        {
            this.nim = nim;
            currentState = State.Berdiri;
        }

        public void TombolS()
        {

```

```

        Console.WriteLine("Sebelum TombolS: " +
currentState);
        if (nim % 3 == 0)
            Console.WriteLine("tombol arah bawah
ditekan");

        switch (currentState)
        {
            case State.Berdiri:
                currentState = State.Jongkok;
                break;
            case State.Jongkok:
                currentState = State.Tengkurap;
                if (nim % 3 == 1)
                    Console.WriteLine("posisi
istirahat");
                break;
            case State.Terbang:
                currentState = State.Jongkok;
                if (nim % 3 == 2)
                    Console.WriteLine("posisi
landing");
                break;
        }
        Console.WriteLine("Sesudah TombolS: " +
currentState);
    }

    public void TombolW()
    {
        Console.WriteLine("Sebelum TombolW: " +
currentState);
        if (nim % 3 == 0)
            Console.WriteLine("tombola rah atas
ditekan");

        switch (currentState)
        {
            case State.Jongkok:
                currentState = State.Berdiri;
                if (nim % 3 == 1)
                    Console.WriteLine("posisi
standby");
                break;
            case State.Berdiri:
                currentState = State.Terbang;
                if (nim % 3 == 2)
                    Console.WriteLine("posisi take
off");
                break;
        }
        Console.WriteLine("Sesudah TombolW: " +
currentState);
    }

    public void TombolX()
    {
        Console.WriteLine("Sebelum TombolX: " +
currentState);
        if (currentState == State.Jongkok)

```

```

        {
            currentState = State.Terbang;
        }
        Console.WriteLine("Sesudah TombolX: " +
currentState);
    }

    public void PrintState()
    {
        Console.WriteLine("Posisi saat ini: " +
currentState);
    }
}
}

```

Kode tersebut merepresentasikan posisi karakter dalam game menggunakan State Machine dengan enum. Enum ini digunakan untuk mendefinisikan empat status karakter dalam game yaitu Berdiri, Jongkok, Tengkurap, dan Terbang. Class PosisiKarakterGame sendiri mengelola perubahan status karakter berdasarkan tombol yang ditekan dengan atribut currentState untuk menyimpan status karakter saat ini dan nim untuk angka yang mempengaruhi beberapa kondisi dalam perubahan state. Terdapat konstruktor PosisiKarakterGame(int nim) untuk menginisialisasi nim dan mengatur currentState awal ke Berdiri. Terdapat beberapa method yaitu TombolS() untuk menekan tombol "S" (arah bawah), TombolW() untuk menekan tombol "W" (arah atas), TombolX() untuk menekan tombol "X" (khusus naik ke Terbang), dan PrintState() untuk mencetak posisi karakter saat ini.

Class Program.cs

```

using kpl_jurnalmodul4_2311104058;

class Program
{
    static void Main()
    {
        KodeBuah kb = new KodeBuah();
        Console.Write("Masukkan nama buah: ");
        string namaBuah = Console.ReadLine();
        Console.WriteLine("Kode Buah: " +
kb.getKodeBuah(namaBuah));

        Console.WriteLine();

        Console.Write("Masukkan NIM: ");
        long nim = long.Parse(Console.ReadLine());

        PosisiKarakterGame karakter = new
PosisiKarakterGame(nim);
    }
}

```

```
        karakter.TombolS();  
        karakter.PrintState();  
  
        karakter.TombolS();  
        karakter.PrintState();  
  
        karakter.TombolW();  
        karakter.PrintState();  
  
        karakter.TombolX();  
        karakter.PrintState();  
    }  
}
```

Kode di atas adalah program C# yang menguji dua kelas sebelumnya, yaitu KodeBuah dan PosisiKarakterGame. Dengan mengimpor namespace yang berisi kelas KodeBuah dan PosisiKarakterGame. Kelas Program memiliki metode Main() sebagai titik awal eksekusi program diikuti dengan memanggil method – method pada class PosisiKarakterBuah.