

Jurnal Modul 4  
2311104059  
Zhafir Zaidan Avail

Hasil Run:

```
=== PROGRAM KODE BUAH ===
Masukkan nama buah: Apel
Kode buah dari Apel: A00

=== PROGRAM POSISI KARAKTER GAME ===

Simulasi tombolS:
Posisi sekarang: Jongkok

Simulasi tombolW:
Posisi sekarang: Berdiri

Simulasi tombolS lagi:
Posisi sekarang: Jongkok

Simulasi tombolW lagi:
Posisi sekarang: Berdiri

=== Aksi Tambahan Sesuai NIM ===
Saat TombolS ditekan: tombol arah bawah ditekan
Saat TombolW ditekan: tombol arah atas ditekan

C:\Users\zhafi\source\repos\Jurnal_3\Jurnal_3\bin\Debug\net8.0\Jurnal_3.exe (process 13660) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```

Source Code:

➤ kodeBuah.cs

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

namespace Jurnal_4
{
    public class KodeBuah
    {
        private Dictionary<string, string> tabelKodeBuah = new
Dictionary<string, string>()
        {
            {"Apel", "A00"},
            {"Aprikot", "B00"},
            {"Alpukat", "C00"},
            {"Pisang", "D00"},
            {"Paprika", "E00"},
            {"Kurma", "K00"},
            {"Durian", "L00"},
            {"Anggur", "M00"},
            {"Melon", "N00"},
            {"Semangka", "O00"}
        };

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

Penjelasan:

Class **KodeBuah** merupakan class yang berfungsi untuk menyimpan dan mengelola data kode buah menggunakan teknik **table driven**. Di dalam class ini terdapat sebuah

**dictionary** yang berisi pasangan data nama buah dan kode buah, seperti "Apel" dengan kode "A00" dan seterusnya. Class ini memiliki sebuah method bernama `getKodeBuah()` yang menerima input berupa nama buah, kemudian mengembalikan kode buah yang sesuai berdasarkan data dalam dictionary. Jika nama buah tidak ditemukan, maka akan mengembalikan pesan "Kode tidak ditemukan". Dengan menggunakan class ini, pengguna cukup memanggil method tersebut untuk mendapatkan kode buah tanpa harus menggunakan banyak percabangan (if/else atau switch), sehingga lebih praktis dan efisien.

➤ **PosisiKarakterGame.cs**

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

namespace Jurnal_3
{
    public class PosisiKarakterGame
    {
        enum Posisi { Berdiri, Jongkok, Tengkurap, Terbang }
        private Posisi posisiSekarang = Posisi.Berdiri;

        // Method untuk tombol W
        public void tombolW()
        {
            switch (posisiSekarang)
            {
                case Posisi.Jongkok:
                    posisiSekarang = Posisi.Berdiri;
                    break;
                case Posisi.Berdiri:
                    posisiSekarang = Posisi.Terbang;
                    break;
                case Posisi.Tengkurap:
                    posisiSekarang = Posisi.Jongkok;
                    break;
                case Posisi.Terbang:
                    posisiSekarang = Posisi.Berdiri;
                    break;
            }
            Console.WriteLine("Posisi sekarang: " + posisiSekarang);
        }

        // Method untuk tombol S
        public void tombolS()
        {
            switch (posisiSekarang)
            {
                case Posisi.Berdiri:
                    posisiSekarang = Posisi.Jongkok;
                    break;
                case Posisi.Jongkok:
                    posisiSekarang = Posisi.Tengkurap;
                    break;
            }
            Console.WriteLine("Posisi sekarang: " + posisiSekarang);
        }

        // Tambahan aksi sesuai NIM % 3
        public void aksiCustom(int nim)
        {
            if (nim % 3 == 0)
            {
                Console.WriteLine("Saat TombolS ditekan: tombol arah bawah ditekan");
            }
        }
    }
}
```

```
        Console.WriteLine("Saat TombolW ditekan: tombol arah atas  
ditekan");  
    }  
    else if (nim % 3 == 1)  
    {  
        Console.WriteLine("State ke Berdiri: posisi standby");  
        Console.WriteLine("State ke Tengkurap: posisi istirahat");  
    }  
    else if (nim % 3 == 2)  
    {  
        Console.WriteLine("State Terbang ke Jongkok: posisi landing");  
        Console.WriteLine("State Berdiri ke Terbang: posisi take  
off");  
    }  
}  
  
}
```

Penjelasan:

Class **PosisiKarakterGame** merupakan class yang dirancang untuk mensimulasikan pergerakan atau posisi karakter dalam game menggunakan **state-based construction**. Class ini menggunakan **enumerasi (enum)** untuk mendefinisikan posisi karakter, yaitu **Berdiri, Jongkok, Tengkurap, dan Terbang**. Class ini memiliki dua method utama yaitu **tombolW()** dan **tombolS()** yang masing-masing berfungsi untuk mengubah posisi karakter berdasarkan tombol yang ditekan (W untuk naik posisi, S untuk turun posisi). Selain itu, class ini juga dilengkapi dengan method **aksiCustom()** yang menampilkan aksi tambahan sesuai hasil pembagian NIM dengan 3, sehingga outputnya bisa menyesuaikan karakteristik mahasiswa. Dengan class ini, alur perubahan posisi karakter menjadi lebih terstruktur dan mudah dipahami karena berbasis state.

➤ Program.cs

```
using Jurnal_3;

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("=== PROGRAM KODE BUAH ===");
        KodeBuah kodeBuah = new KodeBuah();
        Console.Write("Masukkan nama buah: ");
        string namaBuah = Console.ReadLine();
        Console.WriteLine($"Kode          buah          dari          {namaBuah}:");
        {kodeBuah.getKodeBuah(namaBuah)}");

        Console.WriteLine("\n=== PROGRAM POSISI KARAKTER GAME ===");
        PosisiKarakterGame karakter = new PosisiKarakterGame();

        Console.WriteLine("\nSimulasi tombolS:");
        karakter.tombolS(); // Coba tekan S

        Console.WriteLine("\nSimulasi tombolW:");
        karakter.tombolW(); // Coba tekan W

        Console.WriteLine("\nSimulasi tombolS lagi:");
        karakter.tombolS(); // Tekan S lagi

        Console.WriteLine("\nSimulasi tombolW lagi:");
        karakter.tombolW(); // Tekan W lagi

        Console.WriteLine("\n=== Aksi Tambahan Sesuai NIM ===");
        int nim = 12345678; // GANTI dengan NIM kamu
        karakter.aksiCustom(nim);
    }
}
```

Penjelasan:

Class Main atau Program merupakan class utama yang berfungsi untuk menjalankan seluruh alur program. Di dalam class ini, dilakukan pemanggilan method dari class KodeBuah untuk menampilkan kode buah berdasarkan nama buah yang dimasukkan. Setelah itu, program juga meminta input NIM dari saya yaitu 2311104059 yang kemudian digunakan untuk menginisialisasi class PosisiKarakterGame, sehingga karakter dapat disimulasikan bergerak sesuai tombol yang ditekan. Class ini juga memanggil beberapa method seperti tombolS() dan tombolW() untuk menampilkan transisi posisi karakter dari satu state ke state lain, serta menampilkan output tambahan berdasarkan hasil mod NIM. Dengan kata lain, class Main menjadi pengontrol alur utama program yang mengintegrasikan fungsi dari semua class lain yang dibuat.