



**TUGAS JURNAL
KONSTRUKSI PERANGKAT LUNAK
MODUL 4**

D
I
S
U
S
U
N

OLEH :
TIURMA GRACE ANGELINA – 2311104042
S1-SE07-02

Dosen :

Yudha Islami Sulistya

**Prodi S1 Rekaya Perangkat Lunak
Direktorat Kampus Purwokerto**

CODE :

KODE DENGAN TEKNIK TABLE DRIVEN

1. KodeBuah.cs

```
using System;
using System.Collections.Generic;

public class KodeBuah
{
    private static Dictionary<string, string> kodeBuah = 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 static string getKodeBuah(string buah)
    {
        return kodeBuah.ContainsKey(buah) ? kodeBuah[buah] : "Kode tidak ditemukan";
    }
}
```

2. Program.cs

```
class Program
{
    static void Main()
    {
        Console.Write("Masukkan nama buah: ");
        string inputBuah = Console.ReadLine();
        Console.WriteLine("Kode untuk {0} adalah: {1}", inputBuah,
        KodeBuah.getKodeBuah(inputBuah));
    }
}
```

Output :

The screenshot shows the Microsoft Visual Studio interface. In the center, there's a code editor window for 'Program.cs' with the following code:

```
1 0 references
2 class Program
3 {
4     0 references
5     static void Main()
6     {
7         Console.WriteLine("Masukkan nama buah: ");
8         string inputBuah = Console.ReadLine();
9         Console.WriteLine("Kode untuk {0} adalah: {1}", inputBuah, KodeBuah.getKodeBuah(inputBuah));
10    }
11 }
```

Below the code editor is a 'Microsoft Visual Studio Debug Console' window showing the output of the program:

```
Masukkan nama buah: Apel
Kode untuk Apel adalah: A00

C:\Users\USER\source\repos\KPL_TiurmaGraceA_2311104042_S1SE-07-02\04_Automata_dan_Table-Driven_Construction\tjmod4_2311104042\bin\Debug\net8.0\tjmod4_2311104042.exe (process 8348) 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 . . .
```

The status bar at the bottom right indicates the date and time: 6:47 PM 3/13/2025.

Penjelasan :

- Inisialisasi Dictionary

```
private static Dictionary<string, string> kodeBuah = new Dictionary<string, string>

{
    {"Apel", "A00"},

    {"Aprikot", "B00"},

    {"Alpukat", "C00"},

    {"Pisang", "D00"},

    {"Paprika", "E00"},

    {"Kurma", "K00"},

    {"Durian", "L00"},

    {"Anggur", "M00"},

    {"Melon", "N00"},

    {"Semangka", "O00"}
```

```
};
```

- Menggunakan struktur data Dictionary untuk menyimpan nama buah sebagai kunci (key) dan kode buah sebagai nilai (value).
- Contoh:
 - Apel → A00
 - Pisang → D00
 - Durian → L00
- Fungsi getKodeBuah()

```
public static string getKodeBuah(string buah)
```

```
{
```

```
    return kodeBuah.ContainsKey(buah) ? kodeBuah[buah] : "Kode tidak ditemukan";  
}
```

- Mencari kode buah berdasarkan input nama buah.
- Jika nama buah ada di Dictionary, maka kode buah akan ditampilkan.
- Jika tidak ditemukan, maka akan muncul pesan "Kode tidak ditemukan".

- Input dari User

```
Console.Write("Masukkan nama buah: ");
```

```
string inputBuah = Console.ReadLine();
```

- Program meminta input nama buah dari user.

- Menampilkan Hasil

```
Console.WriteLine("Kode untuk {0} adalah: {1}", inputBuah,  
KodeBuah.getKodeBuah(inputBuah));
```

- Memanggil fungsi getKodeBuah() untuk mencari kode buah.
- Hasil kode buah akan ditampilkan.

KODE DENGAN TEKNIK STATE-BASED CONSTRUCTION

1. PosisiKarakterGame.cs

```
using System;

public class PosisiKarakterGame
{
    private string state;

    public PosisiKarakterGame()
    {
        state = "Berdiri";
    }

    public void TombolS()
    {
        Console.WriteLine("tombol arah bawah ditekan");

        if (state == "Berdiri")
        {
            state = "Jongkok";
            Console.WriteLine("Posisi jongkok");
        }
        else if (state == "Jongkok")
        {
            state = "Tengkurap";
            Console.WriteLine("Posisi tengkurap");
        }
    }

    public void TombolW()
    {
        Console.WriteLine("tombol arah atas ditekan");

        if (state == "Jongkok")
        {
            state = "Berdiri";
            Console.WriteLine("Posisi berdiri");
        }
        else if (state == "Berdiri")
```

```

    {
        state = "Terbang";
        Console.WriteLine("Posisi terbang");
    }
}
}

```

2. Program1.cs

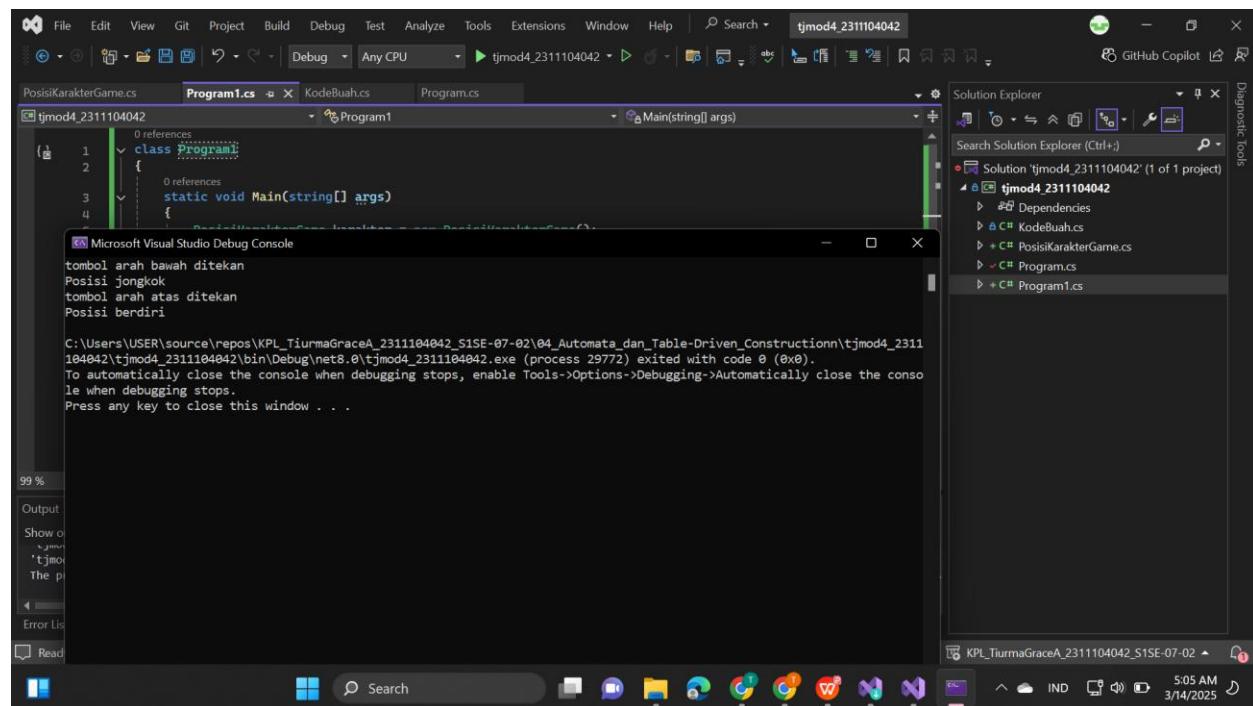
```

class Program1
{
    static void Main(string[] args)
    {
        PosisiKarakterGame karakter = new PosisiKarakterGame();

        karakter.TombolS();
        karakter.TombolW();
    }
}

```

Output :



The screenshot shows the Microsoft Visual Studio interface during a debugging session. The title bar indicates the project name is 'tjmod4_2311104042'. The Solution Explorer on the right shows the solution contains one project named 'tjmod4_2311104042' which includes files 'Dependencies', 'KodeBuh.cs', 'PosisiKarakterGame.cs', 'Program.cs', and 'Program1.cs'. The main code editor window displays the 'Program1.cs' file with its contents. Below the editor is the 'Output' window, which displays the following console logs:

```

tombol arah bawah ditekan
Posisi jongkok
tombol arah atas ditekan
Posisi berdiri

C:\Users\USER\source\repos\KPL_TiurmaGraceA_2311104042_S1SE-07-02\04_Automata_dan_Table-Driven_Construction\tjmod4_2311104042\tjmod4_2311104042\bin\Debug\net8.0\tjmod4_2311104042.exe (process 29772) 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 . . .

```

Penjelasan :

- Inisialisasi State

```
private string state;
```

- Variabel state digunakan untuk menyimpan posisi karakter saat ini.
- State awal adalah "Berdiri".

- Konstruktor PosisiKarakterGame()

```
public PosisiKarakterGame()
```

```
{
```

```
    state = "Berdiri";
```

```
}
```

- Ketika program pertama dijalankan, posisi karakter default-nya adalah "Berdiri".

- Method TombolS()

```
public void TombolS()
```

```
{
```

```
    Console.WriteLine("tombol arah bawah ditekan");
```

```
    if (state == "Berdiri")
```

```
{
```

```
    state = "Jongkok";
```

```
    Console.WriteLine("Posisi jongkok");
```

```
}
```

```
    else if (state == "Jongkok")
```

```
{
```

```
    state = "Tengkurap";
```

```
    Console.WriteLine("Posisi tengkurap");
```

- ```
 }
}

 - Jika posisi karakter "Berdiri", maka ketika tombol S ditekan, berubah menjadi "Jongkok".
 - Jika posisi sudah "Jongkok", maka ketika tombol S ditekan lagi, berubah menjadi "Tengkurap".
```

- **Method TombolW()**

```
public void TombolW()
{
 Console.WriteLine("tombol arah atas ditekan");

 if (state == "Jongkok")
 {
 state = "Berdiri";
 Console.WriteLine("Posisi berdiri");
 }
 else if (state == "Berdiri")
 {
 state = "Terbang";
 Console.WriteLine("Posisi terbang");
 }
}
```

- Jika posisi karakter "Jongkok", maka tombol W ditekan, berubah menjadi "Berdiri".
- Jika posisi "Berdiri", maka tombol W ditekan, berubah menjadi "Terbang".

- **Main Program**

```
static void Main(string[] args)
{
 PosisiKarakterGame karakter = new PosisiKarakterGame();

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

- **Membuat objek karakter.**
- **Tekan tombol S → karakter Jongkok.**
- **Tekan tombol W → karakter kembali Berdiri.**