

Nama : Candra Dinata

Nim : 2311104061

Kelas : SE0702

OUTPUT :

```
Microsoft Visual Studio Debug Console
===== Program Kode Buah =====
Masukkan nama buah: Apel
Kode buah Apel: A00

C:\Users\Candra Dinata\source\repos\JURNALMODUL4_2311104061\JURNALMODUL4_2311104061\bin\Debug\net8.0\JURNALMODUL4_2311104061.exe (process 29288) 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 . . .
```

```
Microsoft Visual Studio Debug Console
===== Program Posisi Karakter Game =====
Posisi awal: Berdiri
Beralih ke posisi Terbang
Posisi take off
Beralih ke posisi Jongkok
Posisi landing
Beralih ke posisi Berdiri
Beralih ke posisi Tengkurap
Posisi istirahat

C:\Users\Candra Dinata\source\repos\JURNALMODUL4_2311104061\JURNALMODUL4_2311104061\bin\Debug\net8.0\JURNALMODUL4_2311104061.exe (process 2868) 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 . . .
```

KODE :

```
1  using System;
2  using System.Collections.Generic;
3
4  class KodeBuah
5  {
6      private static Dictionary<string, string> kodeBuahDict = new Dictionary<string, string>
7      {
8          { "Apel", "A00" }, { "Aprikot", "B00" }, { "Alpukat", "C00" },
9          { "Pisang", "D00" }, { "Paprika", "E00" }, { "Kurma", "K00" },
10         { "Durian", "L00" }, { "Anggur", "M00" }, { "Melon", "N00" },
11         { "Semangka", "O00" }
12     };
13
14     public static string GetKodeBuah(string namaBuah)
15     {
16         return kodeBuahDict.ContainsKey(namaBuah) ? kodeBuahDict[namaBuah] : "Kode tidak ditemukan";
17     }
18 }
19
20 class Program
21 {
22     static void Main()
23     {
24         Console.WriteLine("==== Program Kode Buah =====");
25         Console.Write("Masukkan nama buah: ");
26         string inputBuah = Console.ReadLine();
27         Console.WriteLine($"Kode buah {inputBuah}: {KodeBuah.GetKodeBuah(inputBuah)}");
28     }
29 }
30
```

Kode di atas menerapkan teknik **table-driven** untuk mencari kode buah berdasarkan nama buah yang dimasukkan oleh pengguna. Kelas KodeBuah menyimpan pasangan nama buah dan kode buah dalam **Dictionary<string, string>**, sehingga pencarian kode buah dapat dilakukan secara efisien. Metode GetKodeBuah akan mengembalikan kode buah jika nama buah yang dimasukkan ada dalam kamus, atau menampilkan pesan "Kode tidak ditemukan" jika tidak ada. Di dalam Program.cs, program meminta pengguna memasukkan nama buah melalui Console.ReadLine(), lalu menggunakan KodeBuah.GetKodeBuah(inputBuah) untuk mendapatkan dan menampilkan kode buahnya. Program ini menggunakan konsep pemetaan (mapping) dengan struktur data **Dictionary** untuk menyelesaikan pencarian kode secara cepat dan terstruktur.

```

1  using System;
2
3  class PosisiKarakterGame
4  {
5      public enum State { Berdiri, Jongkok, Tengkurap, Terbang }
6      private State posisiSaatIni;
7
8      public PosisiKarakterGame()
9      {
10         posisiSaatIni = State.Berdiri;
11         Console.WriteLine("Posisi awal: Berdiri");
12     }
13
14     public void UbahPosisi(State posisiBaru)
15     {
16         Console.WriteLine($"Beralih ke posisi {posisiBaru}");
17
18         if (posisiSaatIni == State.Terbang && posisiBaru == State.Jongkok)
19             Console.WriteLine("Posisi landing");
20
21         if (posisiSaatIni == State.Berdiri && posisiBaru == State.Terbang)
22             Console.WriteLine("Posisi take off");
23
24         if (posisiSaatIni == State.Berdiri && posisiBaru == State.Tengkurap)
25             Console.WriteLine("Posisi istirahat");
26
27         if (posisiSaatIni == State.Tengkurap && posisiBaru == State.Berdiri)
28             Console.WriteLine("Posisi standby");
29
30         posisiSaatIni = posisiBaru;
31     }
32 }
33 class Program
34 {
35     static void Main()
36     {
37         Console.WriteLine("==== Program Posisi Karakter Game =====");
38         PosisiKarakterGame karakter = new PosisiKarakterGame();
39         karakter.UbahPosisi(PosisiKarakterGame.State.Terbang);
40         karakter.UbahPosisi(PosisiKarakterGame.State.Jongkok);
41         karakter.UbahPosisi(PosisiKarakterGame.State.Berdiri);
42         karakter.UbahPosisi(PosisiKarakterGame.State.Tengkurap);
43     }
44 }
45

```

Kode di atas menerapkan **state-based construction** untuk mengelola perubahan posisi karakter dalam sebuah game. Kelas `PosisiKarakterGame` memiliki **enum State** yang merepresentasikan empat kemungkinan posisi karakter: **Berdiri**, **Jongkok**, **Tengkurap**, dan **Terbang**. Saat objek `PosisiKarakterGame` dibuat, karakter berada dalam keadaan awal **Berdiri**. Metode `UbahPosisi(State posisiBaru)` memungkinkan perubahan posisi dengan menampilkan pesan sesuai transisi tertentu,

seperti "Posisi take off" saat berpindah dari **Berdiri** ke **Terbang**, atau "Posisi landing" saat berpindah dari **Terbang** ke **Jongkok**. Pada Program.cs, program membuat objek PosisiKarakterGame dan mengubah posisinya secara bertahap dengan memanggil UbahPosisi(), yang mencetak transisi ke layar. Implementasi ini meniru mekanisme perubahan state dalam game berdasarkan kondisi tertentu.