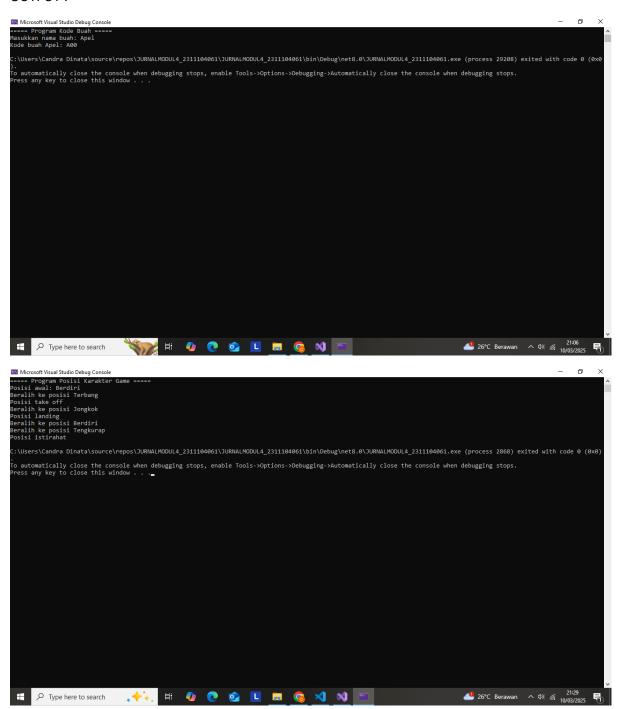
Nama: Candra Dinata

Nim: 2311104061

Kelas: SE0702

## OUTPUT:



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>**, 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.

```
using System;
    class PosisiKarakterGame
        public enum State { Berdiri, Jongkok, Tengkurap, Terbang }
        private State posisiSaatIni;
        public PosisiKarakterGame()
            posisiSaatIni = State.Berdiri;
            Console.WriteLine("Posisi awal: Berdiri");
        public void UbahPosisi(State posisiBaru)
            Console.WriteLine($"Beralih ke posisi {posisiBaru}");
            if (posisiSaatIni == State.Terbang && posisiBaru == State.Jongkok)
                Console.WriteLine("Posisi landing");
            if (posisiSaatIni == State.Berdiri && posisiBaru == State.Terbang)
                Console.WriteLine("Posisi take off");
            if (posisiSaatIni == State.Berdiri && posisiBaru == State.Tengkurap)
                Console.WriteLine("Posisi istirahat");
            if (posisiSaatIni == State.Tengkurap && posisiBaru == State.Berdiri)
                Console.WriteLine("Posisi standby");
            posisiSaatIni = posisiBaru;
33 class Program
        static void Main()
            Console.WriteLine("===== Program Posisi Karakter Game =====");
            PosisiKarakterGame karakter = new PosisiKarakterGame();
            karakter.UbahPosisi(PosisiKarakterGame.State.Terbang);
            karakter.UbahPosisi(PosisiKarakterGame.State.Jongkok);
            karakter.UbahPosisi(PosisiKarakterGame.State.Berdiri);
            karakter.UbahPosisi(PosisiKarakterGame.State.Tengkurap);
```

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.