

Laporan Individu Tugas Besar

1. Identitas

Nama: Kevin Tumanggor

NIM: 1201220044

Modul admin Manajemen pesanan

Fitur yang Dikerjakan:

- melihat daftar pesanan
- mengubah status pesanan

2. Teknik Konstruksi yang Digunakan

Jelaskan teknik konstruksi (Unit Testing / Performance Testing / Design by Contract) yang digunakan.

Teknik Kontruksi	Penjelasan
Unit Testing	Menguji bagian terkecil dari kode (biasanya method/fungsi) secara terpisah untuk memastikan logika berjalan sesuai harapan.
Automata	<p>Model matematika untuk sistem yang memiliki <i>state</i> dan <i>transisi antar state</i>. Digunakan untuk logika status. Menjamin transisi status order berjalan sesuai urutan logis (valid).</p> <pre>public static Dictionary<OrderStatus, List<OrderStatus>> ValidTransitions = new() { { OrderStatus.Pending, new List<OrderStatus> { OrderStatus.Diproses } }, { OrderStatus.Diproses, new List<OrderStatus> { OrderStatus.Bikirim } }, { OrderStatus.Bikirim, new List<OrderStatus> { OrderStatus.Selesai } }, { OrderStatus.Selesai, new List<OrderStatus> { } } }; public static bool IsValidTransition(OrderStatus current, OrderStatus next) { return ValidTransitions.ContainsKey(current) && ValidTransitions[current].Contains(next); }</pre>
API (Application Programming Interface)	Kumpulan endpoint yang digunakan client (frontend atau aplikasi lain) untuk berinteraksi dengan server.

3. Unit Testing

Tuliskan kode unit testing yang telah dibuat beserta penjelasan singkat.

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using OrderManagementAPI.Services;
using OrderManagementAPI.Models;

namespace OrderManagementTests
{
    [TestClass]
    public class OrderServiceTests
    {
        private OrderService _service;

        [TestInitialize]
        public void Setup()
        {
            _service = new OrderService();
        }

        [TestMethod]
        public void GetAll_ShouldReturnData()
        {
            var result = _service.GetAll();
            Assert.IsTrue(result.Count > 0);
        }

        [TestMethod]
        public void UpdateStatus_ValidTransition_ShouldSucceed()
        {
            var result = _service.UpdateStatus(1, OrderStatus.Diproses);
            Assert.IsTrue(result);
        }
    }
}
```

Penjelasan :

```
[TestMethod]
public void GetAll_ShouldReturnData()
{
    var result = _service.GetAll();
    Assert.IsTrue(result.Count > 0);
}
```

Tujuan: Memastikan bahwa method `GetAll()` mengembalikan data order yang tersedia (jumlahnya lebih dari 0).

Artinya: Data dummy dalam `OrderService` benar-benar tersedia dan bisa diambil.

```
[TestMethod]
public void UpdateStatus_ValidTransition_ShouldSucceed()
{
    var result = _service.UpdateStatus(1, OrderStatus.Diproses);
    Assert.IsTrue(result);
}
```

Tujuan: Memastikan bahwa transisi status yang valid dari Pending ke Diproses bisa dilakukan.

Artinya: Metode UpdateStatus() bekerja sesuai aturan transisi yang sudah didefinisikan di OrderStatusTransition.

4. Performance Testing

Tuliskan hasil dan penjelasan pengujian performa

Test	Duration	Tr
OrderManagementTests (2)	148 ms	
OrderManagementTests (2)	148 ms	
OrderServiceTests (2)	148 ms	
GetAll_ShouldReturnData	73 ms	
UpdateStatus_ValidTransition_S...	75 ms	

1. GetAll_ShouldReturnData

Fungsi yang diuji: OrderService.GetAll() yang memiliki tujuan memastikan sistem bisa mengambil semua data pesanan dari list. Hasilnya Tes berhasil, artinya sistem berhasil mengembalikan data pesanan yang tersedia.

2. UpdateStatus_ValidTransition_ShouldUpdateStatus

Fungsi yang diuji: OrderService.UpdateStatus() yang memiliki tujuan memastikan perubahan status dari Pending ke Diproses bisa dilakukan (transisi valid). Hasilnya Tes lulus, artinya logika automata berjalan dengan benar, dan status berhasil diperbarui.

5. Design by Contract

Jelaskan implementasi design by contract yang digunakan dalam fitur Anda.

Teknik Kontruksi	Penjelasan
Design by Contract	<p>Design by Contract adalah pendekatan yang memastikan setiap fungsi memiliki aturan yang harus dipenuhi sebelum dan sesudah dijalankan. Dalam proyekmu, konsep ini terlihat pada metode UpdateStatus dan IsValidTransition, di mana status order hanya boleh berubah jika memenuhi aturan transisi yang valid.</p> <pre>public static Dictionary<OrderStatus, List<OrderStatus>> ValidTransitions = new() { { OrderStatus.Pending, new List<OrderStatus> { OrderStatus.Diproses } }, { OrderStatus.Diproses, new List<OrderStatus> { OrderStatus.Dikirim } }, { OrderStatus.Dikirim, new List<OrderStatus> { OrderStatus.Selesai } }, { OrderStatus.Selesai, new List<OrderStatus>() } }; public static bool IsValidTransition(OrderStatus current, OrderStatus next) { return ValidTransitions.ContainsKey(current) && ValidTransitions[current].Contains(next); }</pre>