Laporan Individu Tugas

# 1. Identitas

Nama : Adelia Nafiatul Farida  
NIM : 1201220025  
Fitur yang Dikerjakan:

Modul user Manajemen pesanan (melihat status pesanan, riwayat pesanan).

- Melihat status pesanan

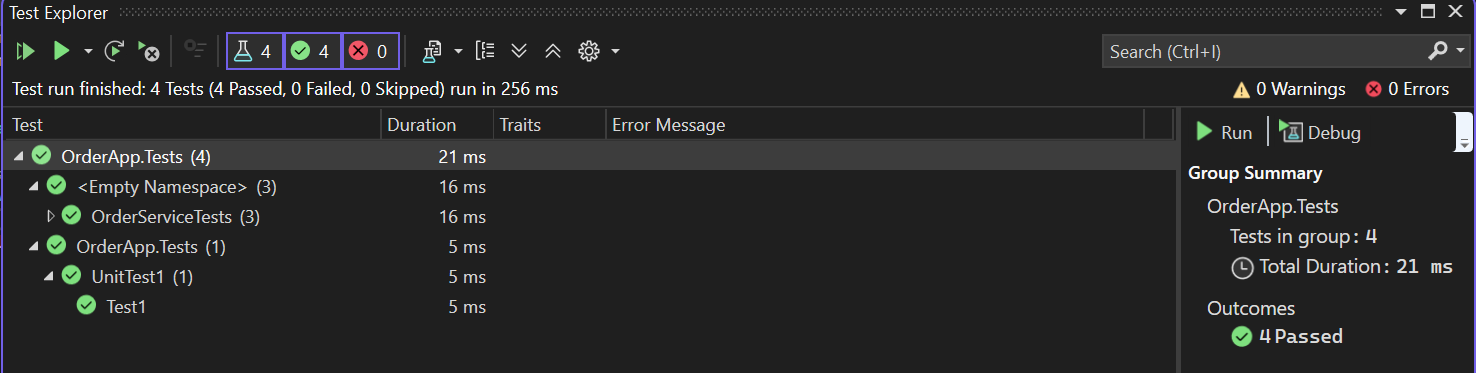
- Riwayat pesanan.

# 2. Teknik Konstruksi yang Digunakan

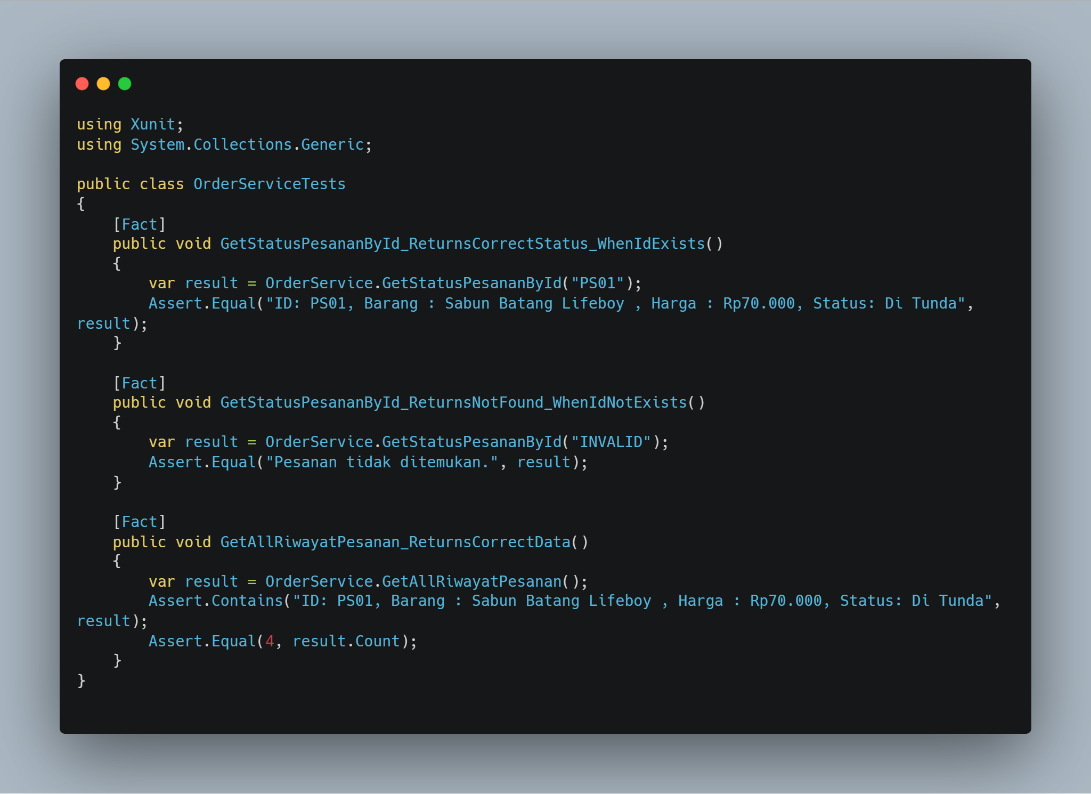
Jelaskan teknik konstruksi (Table Driven dan Parameterization/generics ) yang digunakan.

**Table Driven** adalah teknik yang menggunakan struktur data(biasanya array, dictionary, atau list) untuk menggantikan logika percabangan kompleks (seperti banyak if-else atau switch-case). **Parameterization** atau **Generics** adalah teknik yang memungkinkan pembuatan tipe dataatau fungsi umum yang bisa digunakan dengan berbagai tipe data tanpa mengulang kode.

# 3. Unit Testing

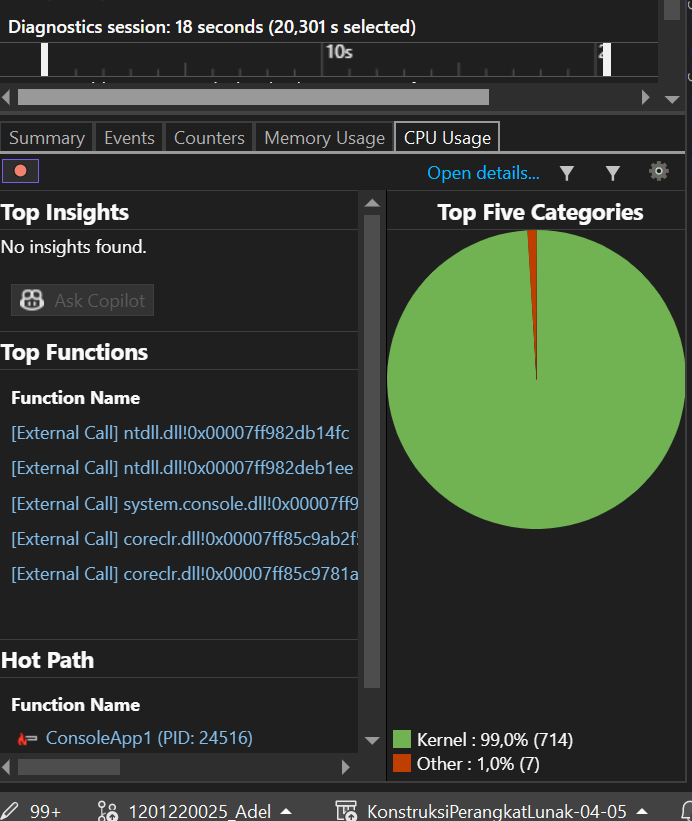
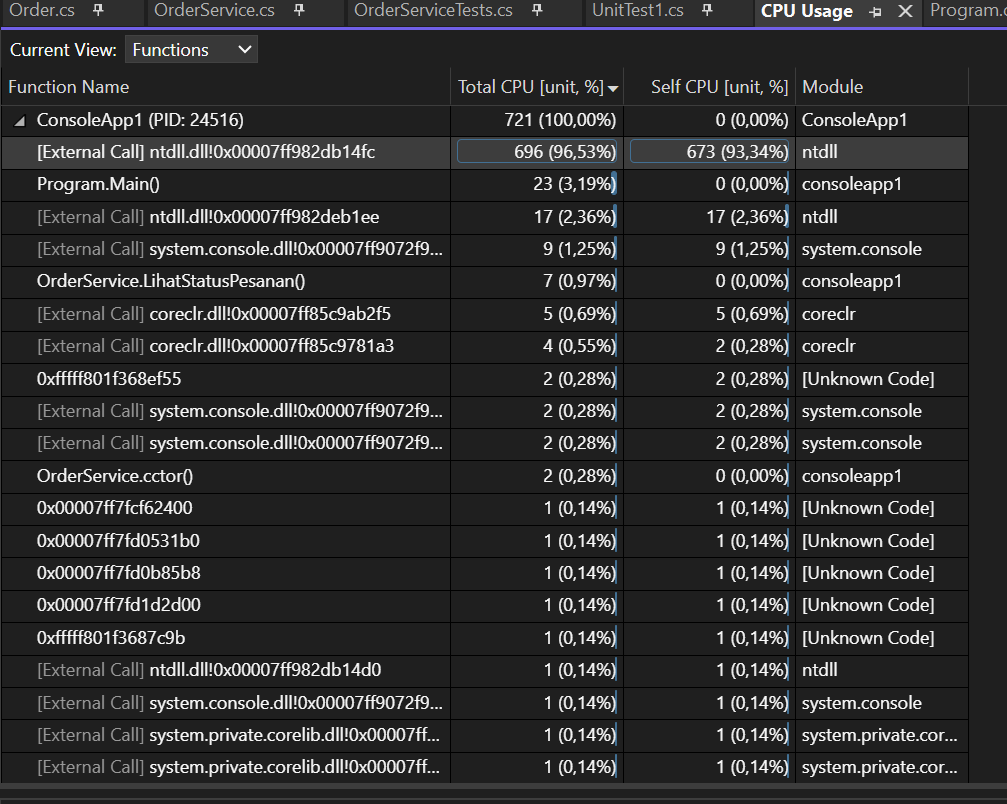


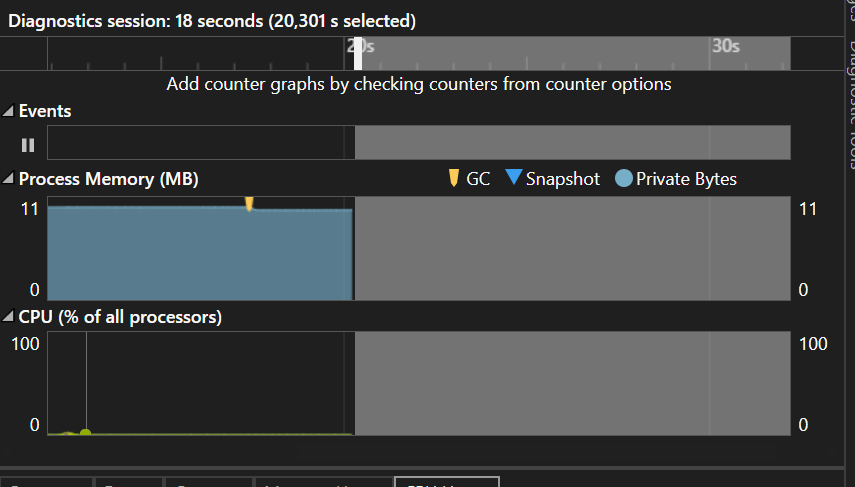
Gambar menunjukkan semua 4 unit test pada proyek berhasil dijalankan tanpa error. Artinya, semua fungsi yang diuji berjalan dengan baik.



Test pertama memastikan bahwa jika dicari ID pesanan yang benar (PS01), hasil yang dikembalikan sesuai dengan data yang diharapkan. Test kedua memeriksa jika ID yang tidak ada (INVALID) dimasukkan, maka sistem akan mengembalikan pesan "Pesanan tidak ditemukan." Test ketiga memeriksa bahwa daftar riwayat pesanan mengandung data tertentu dan berjumlah 4 item. Semua test ini digunakan untuk memastikan bahwa program berjalan dengan benar dan data yang ditampilkan sesuai harapan.

# 4. Performance Testing



Hasil performance testing aplikasi ConsoleApp1 menunjukkan kinerja yang efisien dan ringan. Sebagian besar penggunaan CPU (99%) berada di sisi kernel karena aplikasi menunggu input pengguna. Fungsi Main() dan LihatStatusPesanan() hanya menggunakan sedikit CPU, sementara sisanya berasal dari library sistem. Penggunaan memori stabil di sekitar 11MB tanpa lonjakan atau aktivitas GC berlebih, menandakan manajemen memori yang baik. Secara keseluruhan, aplikasi berjalan optimal, responsif, dan tidak membebani sistem.

# 5. Design by Contract



Kode ini menggunakan prinsip Design by Contract melalui Debug.Assert() untuk memastikan program berjalan sesuai aturan. Precondition memeriksa bahwa input pilihan tidak null sebelum diproses, mencegah error. Postcondition memastikan setelah fungsi LihatRiwayatPesanan() dijalankan, daftar pesanan tidak kosong, menandakan data berhasil diakses. Invariant menjaga daftar pesanan tidak pernah null selama program berjalan, memastikan konsistensi data. Dengan cara ini, program menjadi lebih andal dan mudah mendeteksi kesalahan saat pengembangan.

# 6. Penjelasan Source Code

* Pertama membuat **program.cs** yang berjalan terus – menerus dalam loop while sampai user memilih keluar (“0”) dengan menampilkan menu utama Lihat Status Pesanan, Lihat Riwayat Pesanan, Keluar.



Code diatas menampilkan output dari menu pilihan dan meminta user untuk menginputkan nomor pilihan dari keyboard.

* Selanjutnya adalah penerapan dari Desaign By Contract, dimana mengecek bahwa input pilihan tidak null, Jika null maka program berhenti saat **debug** (tidak saat release build).



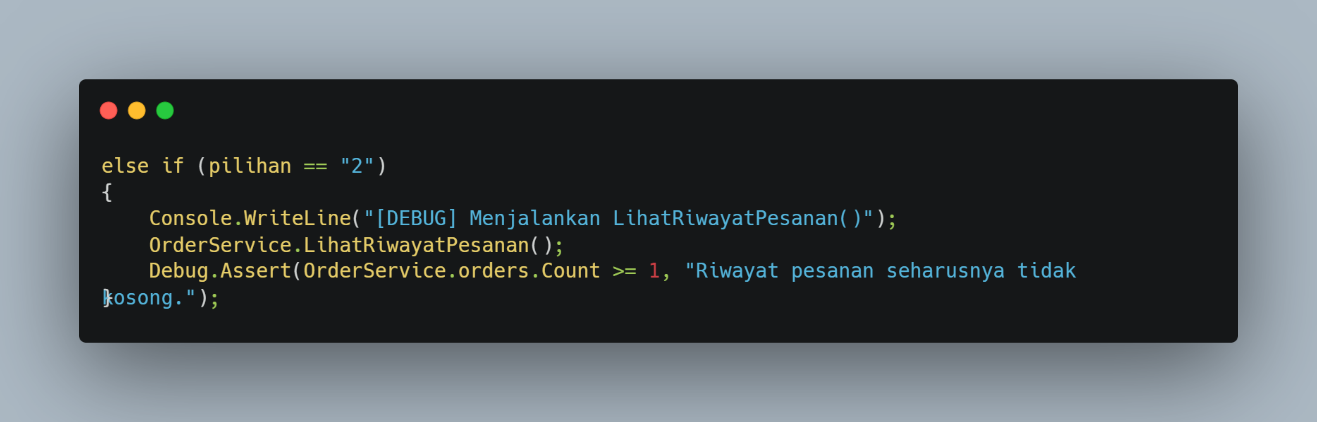
* Jika user mengetik "0", program akan berhenti (break keluar dari while).



* Jika user memilih "1", maka akan dipanggil fungsi OrderService.LihatStatusPesanan() untuk menampilkan status dari suatu pesanan berdasarkan ID.



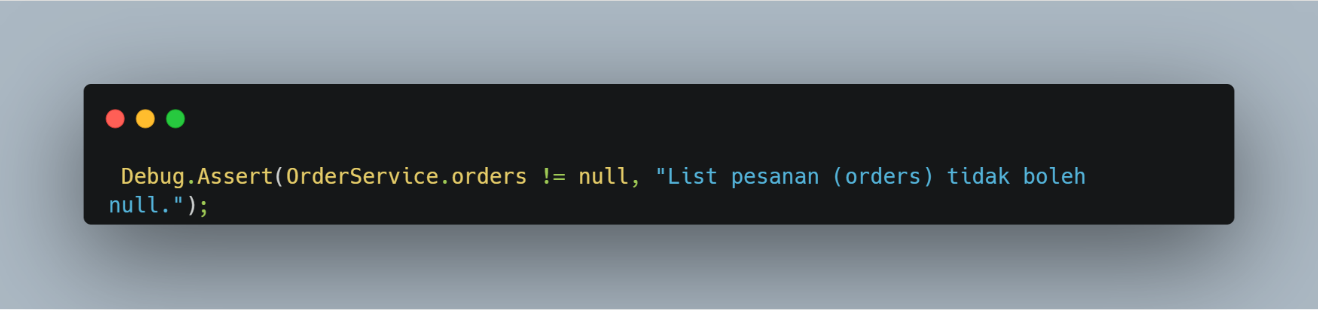
* Program memanggil OrderService.LihatRiwayatPesanan(). Setelah fungsi dijalankan, harus dipastikan bahwa orders tidak kosong (berisi 1 atau lebih data).



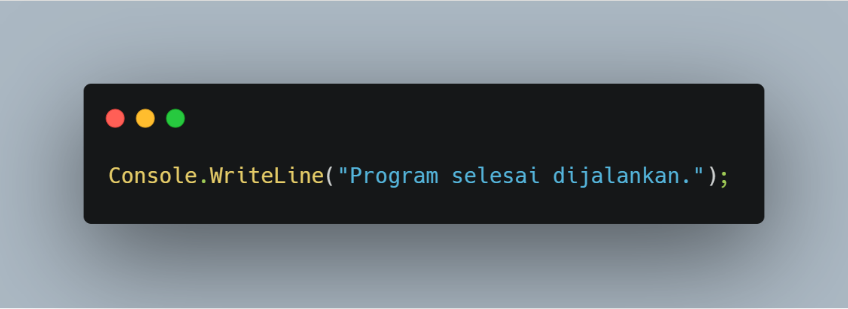
* Jika user memasukkan input selain "0", "1", atau "2", maka dianggap salah dan disuruh ulangi lagi.



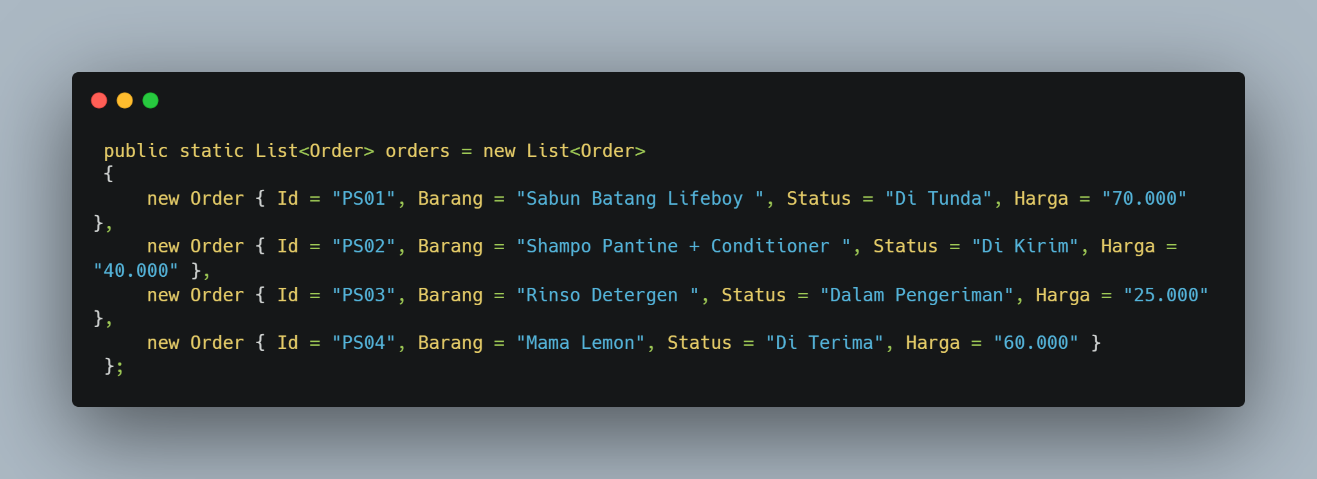
* Mengecek bahwa OrderService.Orders tidak boleh null sepanjang program berjalan.



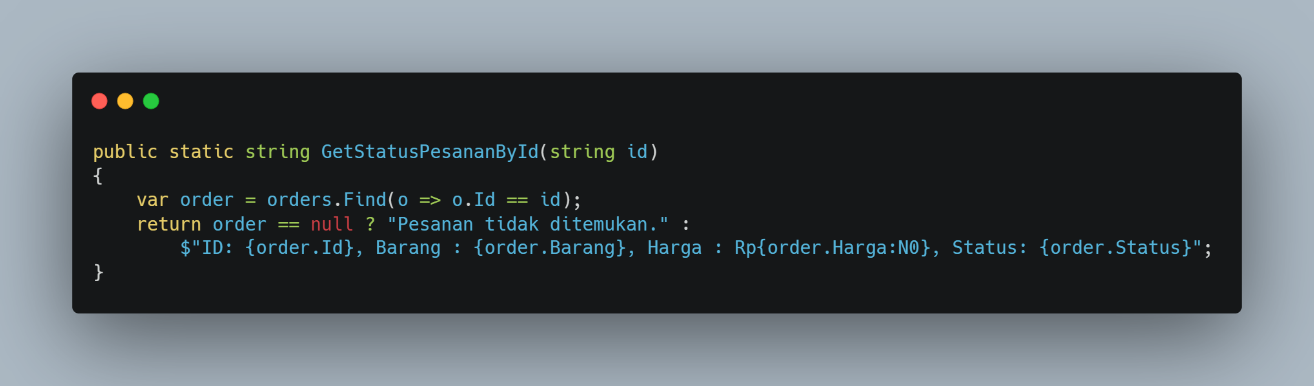
* Pesan ini ditampilkan saat program berhenti dari loop.



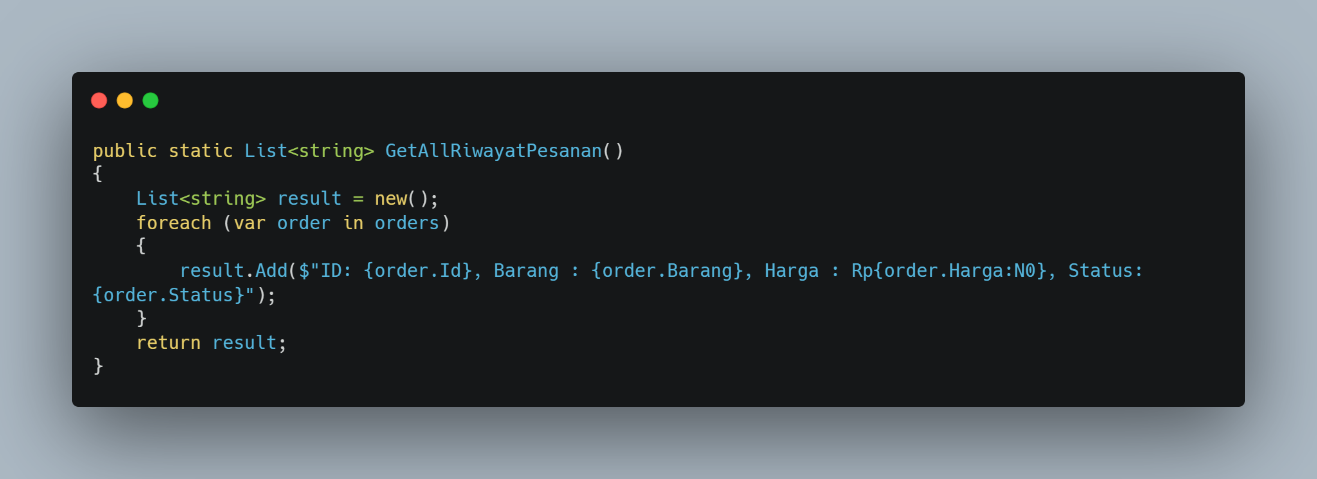
* Pada **OrderService.cs** menyimpan data pesanan dan menampilkan informasi berdasarkan permintaan pengguna, setiap item pesanan memiliki ID, Nama Barang, Harga, dan status.



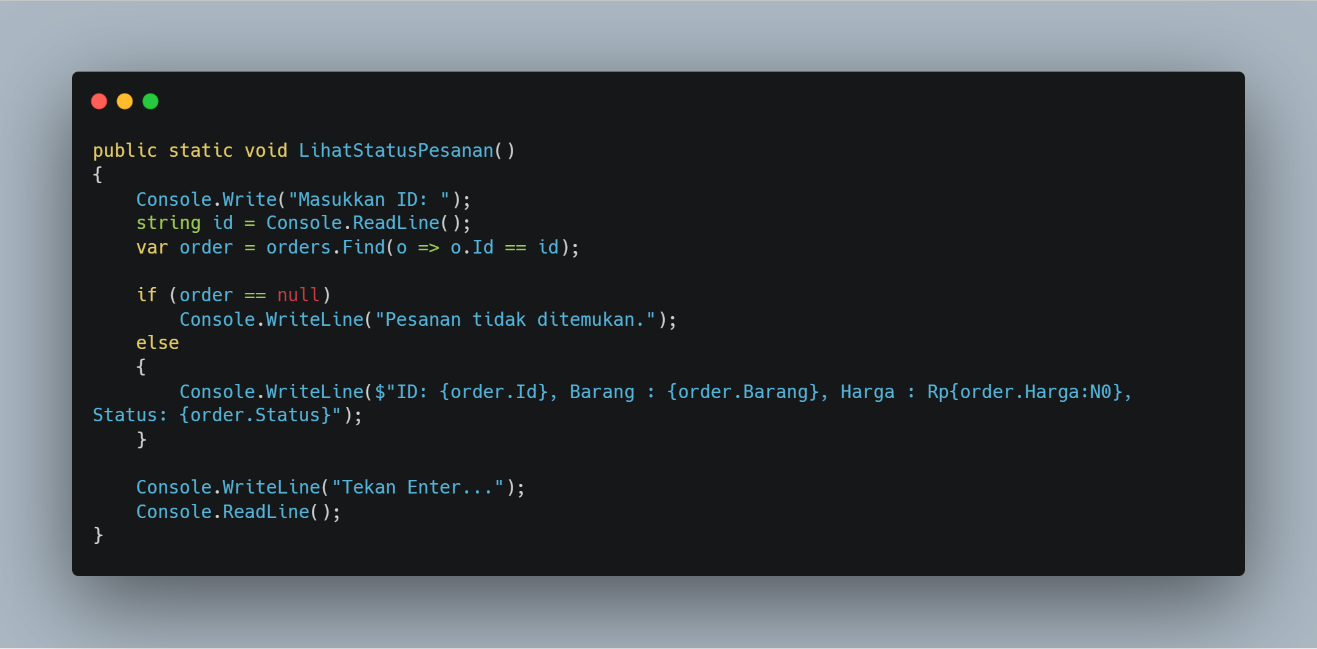
* Fungsi ini menerima ID pesanan sebagai input kemudian mencarinya pada list orders dengan menggunakan find, Jika ketemu, akan mengembalikan informasi ID, Barang, Harga, dan Status dalam bentuk teks. Jika tidak ditemukan, mengembalikan pesan "Pesanan tidak ditemukan.".



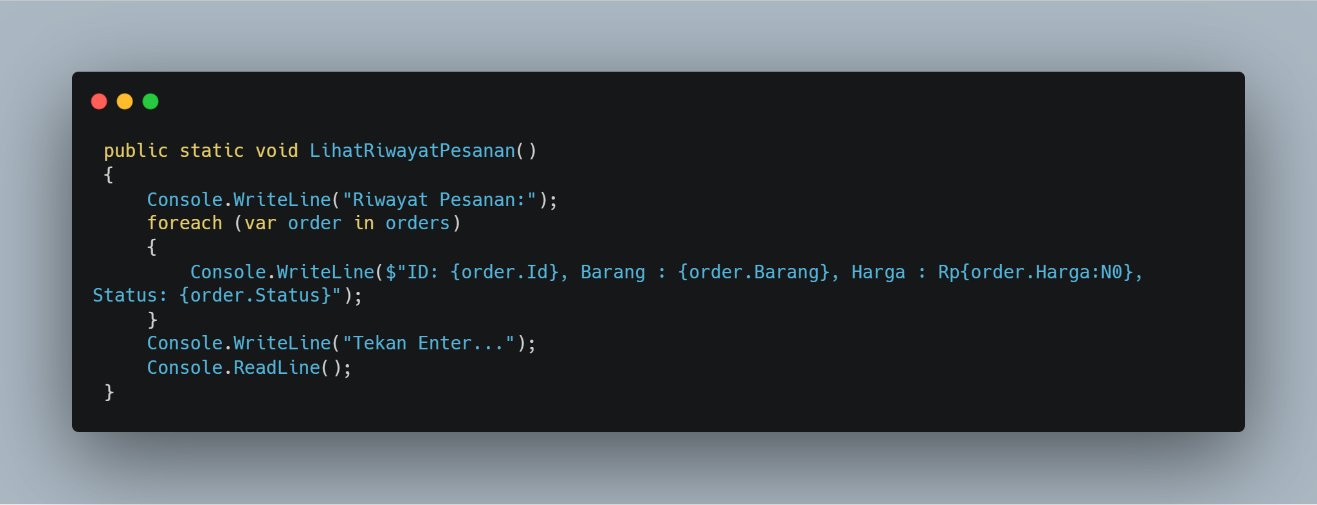
* Mengembalikan daftar semua pesanan dalam bentuk list teks setiap item teks berisi detail: ID, Barang, Harga, dan Status. Fungsinya bisa digunakan untuk unit test atau sistem lain yang ingin mengambil semua data riwayat.



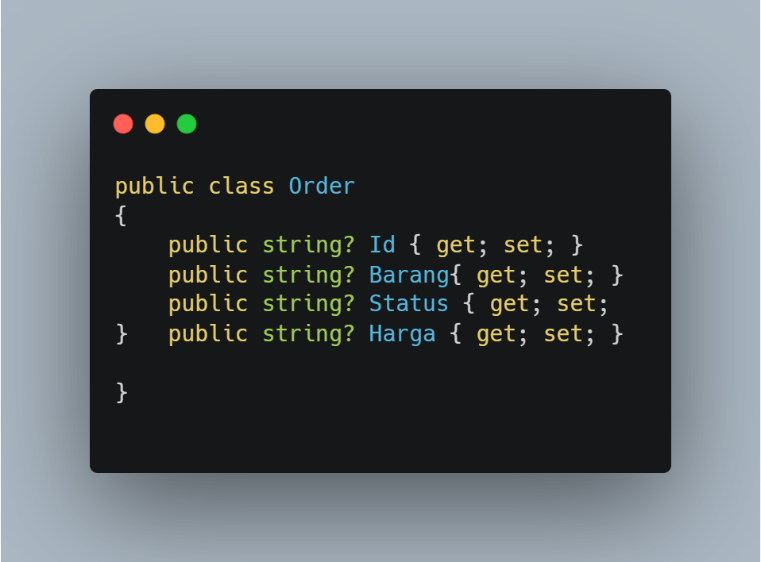
* Fungsi ini digunakan saat pengguna ingin mengecek status pesanan tertentu, program meminta input ID dari konsol, mencari pesanan tersebut, Menampilkan detail jika ditemukan, atau pesan jika tidak ditemukan.



* Menampilkan **semua pesanan yang ada** di layar (riwayat lengkap), Setiap pesanan ditampilkan satu per satu dalam bentuk yang rapi.



* Pada **Order.cs** menampilkan data dari satu pesanan Class Order adalah template data untuk menyimpan informasi pesanan. Isinya ada 4 properti: Id, Barang, Status, dan Harga, semuanya bertipe string dan bisa null (string?). Properti ini bisa dibaca dan diubah. Class ini biasa digunakan untuk membuat daftar pesanan dalam program.



* Pada OrderService.cs Fungsi utamanya adalah menyimpan data pesanan dan menampilkan informasi berdasarkan permintaan pengguna.

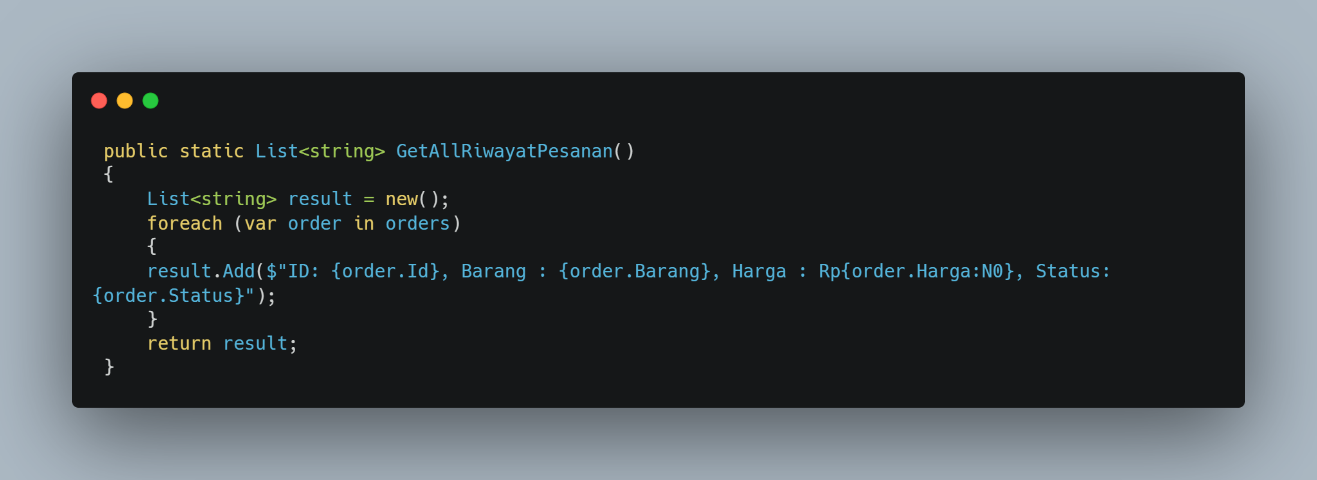


Di sini, data pesanan disimpan dalam list orders, Setiap item pesanan memiliki ID,nama barang, harga, dan status. Data di atas kamu simpan sebagai list — dan kamu menggunakan data ini untuk menjalankan logika, bukan membuat kode bercabang berdasarkan ID atau nama barang.

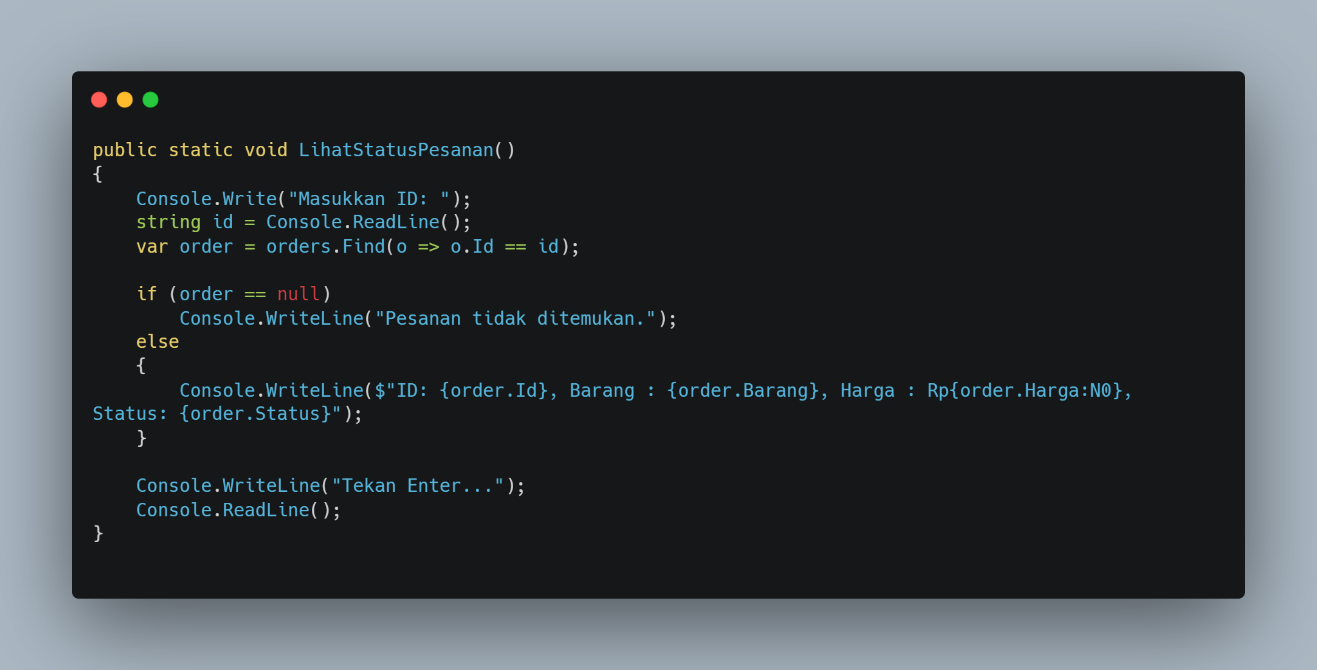
* Mencari pesanan berdasarkan id, Jika ditemukan, akan mengembalikan informasi lengkap pesanan. Jika tidak ditemukan, mengembalikan pesan: "Pesanan tidak ditemukan."



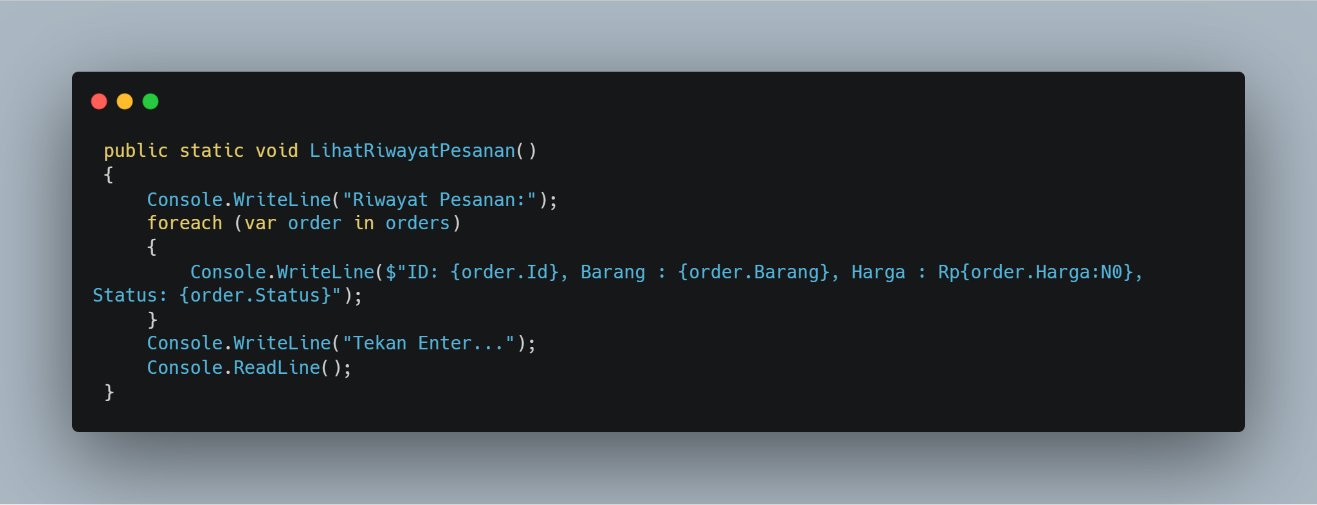
* Mengembalikan semua pesanan dalam bentuk list teks (string), Berguna jika ingin menampilkan semua riwayat dalam bentuk list.



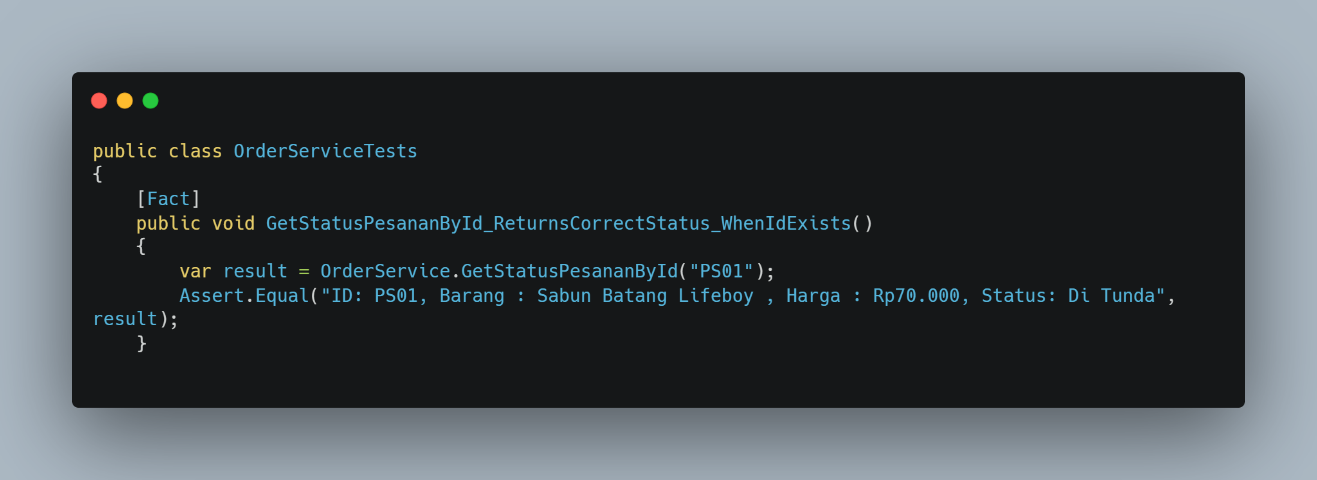
* Meminta user memasukkan ID pesanan, Menampilkan informasi pesanan jika ditemukan.



* Menampilkan semua data pesanan di console, Berguna untuk melihat seluruh riwayat transaksi.

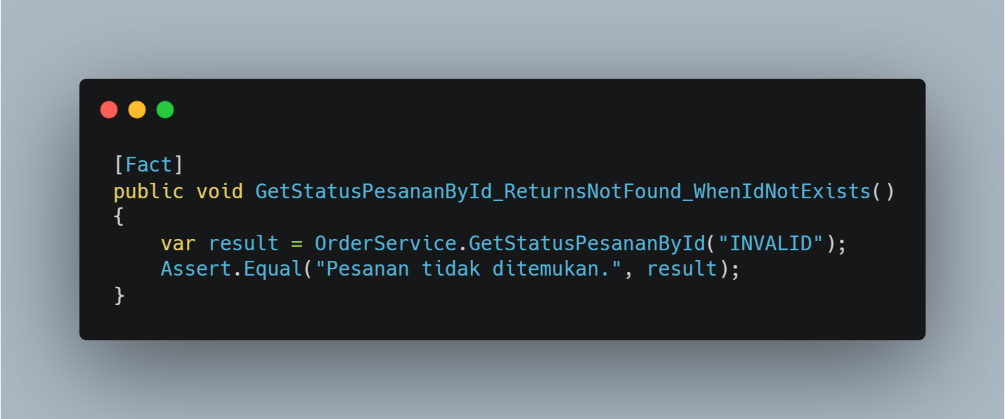


* Unit test yang ditulis menggunakan framework xUnit di C#. Tujuan utamanya adalah untuk menguji fungsi-fungsi di dalam OrderService agar berjalan sesuai harapan.

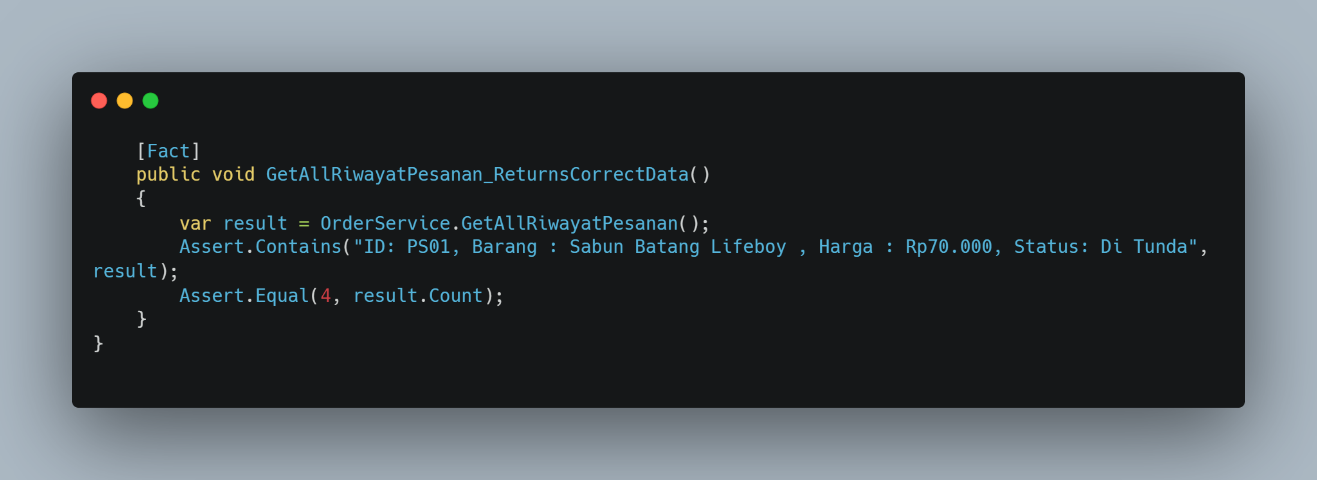


Isinya adalah metode-metode test untuk menguji perilaku dari class OrderService. Fungsi tersebut adalah test case unit (pengujian tunggal), xUnit akan menjalankan setiap [Fact] secara otomatis saat testing.

* Menguji apakah fungsi mengembalikan pesan "Pesanan tidak ditemukan." jika ID tidak ada.



* Menguji fungsi GetAllRiwayatPesanan(). Mengecek apakah Daftar berisi salah satu string yang benar (pakai Assert.Contains) Jumlah data dalam list benar (pakai Assert.Equal(4, result.Count) karena ada 4 pesanan).



# 7. Refleksi Pribadi (Opsional)

Kendala yang saya hadapi mungkin belum terlalu paham sepenuhnya, masih banyak bingung sehingga sering bertanya ke teman sekolmpok.