**LAPORAN TUGAS BESAR CLO 2**

**CUI AUTO VENDING Using C# with Visual Studio 2022**



Dibuat Oleh:

EVI FITRIYA 1201222005

**SOFTWARE ENGINEERING**

**TELKOM UNIVERSITY**

**2025**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc198242389)

[DESKRIPSI SINGKAT 3](#_Toc198242390)

[DAFTAR ANGGOTA KELOMPOK 3](#_Toc198242391)

[GITHUB 4](#_Toc198242392)

[IMLEMENTASI DESIGN BY KONTRAK 5](#_Toc198242393)

[HASIL UNIT TESTING 7](#_Toc198242394)

[HASIL PERFOMANCE TESTING 11](#_Toc198242395)

# DESKRIPSI SINGKAT

Auto Vending adalah aplikasi smart vending machine berbasis self-checkout yang memungkinkan pengguna melakukan pemesanan dan pembayaran produk secara mandiri tanpa bantuan operator. Aplikasi ini dikembangkan menggunakan C# dalam Visual Studio 2022 dan dirancang untuk diintegrasikan langsung dengan perangkat vending fisik. Dengan antarmuka yang intuitif dan pengalaman transaksi yang cepat, Auto Vending bertujuan untuk meningkatkan efisiensi layanan penjualan otomatis, serta memberikan fleksibilitas tinggi dalam pengelolaan produk dan konfigurasi sistem.

Fitur-fitur yang tersedia:

* Perubahan Status
* Pemilihan Produk
* Ganti Bahasa
* Manajemen Produk
* Jam operasional
* Convert Mata Uang
* Checkout
* Transaksi

# DAFTAR ANGGOTA KELOMPOK

1. Zidan Irfan Zaky 1201220003
2. Farhan Nugraha Sasongko Putra 1201220449
3. Radinka Putra Rahadian 1201220020
4. Giovan Deo Pratama 1201220450
5. Evi Fitriya 1201222005

Dengan Pembagian Tugas sesuai arahan dimana 1 teknik konstruksi dipegang oleh maximal 2 orang:



# GITHUB

Repositori ini menggunakan sistem version control GitHub untuk mendokumentasikan proses konstruksi proyek secara kolaboratif. Terdapat *branch* pribadi yang digunakan untuk pengembangan fitur atau modul secara terpisah, sebelum digabungkan ke *branch* utama (*main*) melalui proses *commit* dan *merge*. Pendekatan ini memastikan riwayat perubahan tercatat dengan jelas serta mempermudah kolaborasi dan pengelolaan versi.

Berikut Link Github Kami:

<https://github.com/ZidZaky/KonsPL-PEOPLEEEE.git>

Dengan bukti:  
Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

# IMLEMENTASI DESIGN BY KONTRAK

Sebuah gambar berisi teks, cuplikan layar, software, tampilan

Konten yang dihasilkan AI mungkin salah.Sebuah gambar berisi teks, cuplikan layar, software, Font

Konten yang dihasilkan AI mungkin salah.

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Dari semua fungsi yang saya miliki, hanya fungsi ini yg menerima parameter, meskipun ada fungsi ShowProductBuy() dengan parameter inputan tapi variable ini berasal dari Order(), jadi cukup di cek di fungsi Order() saja.

Sebuah gambar berisi cuplikan layar, teks, Font, garis

Konten yang dihasilkan AI mungkin salah.

Ada juga fungsi Payyment() yang menerima parameter products tapi ini juga sama-sama berasal dari Order jadi saya rasa ini hanya perlu di validasi sekali di Order() saja.

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Konten yang dihasilkan AI mungkin salah.

Berikut implementasi kodenya:

Sebuah gambar berisi teks, cuplikan layar, Font, garis

Konten yang dihasilkan AI mungkin salah.

Dimana fungsi string.IsNullOrWhiteSpace() jika inputan dari berisi null/””/” “/”\t”/”\n”\dll akan bernilai true, dan agar debugnya bekerja harus dalam mode false maka dari itu kita beri ! sebagai tanda not, kemudian ketika nilainya false akan memunculkan “Inputan tidak boleh kosong”

Lalu inputan!=”0” mengecek apakah isinya “0” jika ya maka akan muncul Inputan tidak boleh 0 di debug.

berikut buktinya:

Sebuah gambar berisi teks, cuplikan layar, Font

Konten yang dihasilkan AI mungkin salah.

Dan akan menghentikan program lalu memberitahu kita kode mana yang terhenti:

Sebuah gambar berisi teks, cuplikan layar, Font, garis

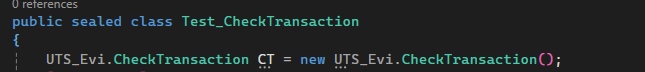
Konten yang dihasilkan AI mungkin salah.

# HASIL UNIT TESTING

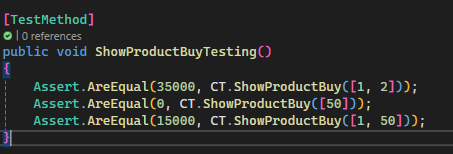
Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Terdapat 2 Class yang saya test yakni statusVending dan CheckTransaction, adapun untuk CheckTransaction sendiri sebagai berikut:



Yang paling pertama kita persiapkan variable yang mewakili class CheckTransaction(),



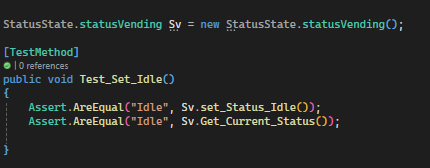
showProductTesting, adalah mengecek apakah hitungan total transaksinya sesuai dengan yang diminta, kita gunakan Assert.AreEqual untuk mengecek ekspektasi dan realitanya sama atau belum.

Selanjutnya kita testing function Product apakah hasil returnnya sama dengan data produk yg sudah di siapkan.

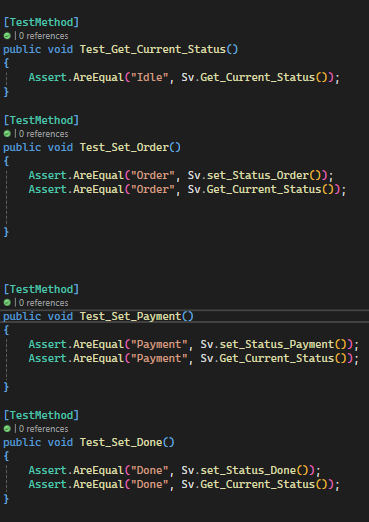


Dengan sama sama menggunakan Assert.AreEqual tp harus dilooping terlebih dahulu agar terbaca sama antara yang diharapkan dengan kenyataannya.

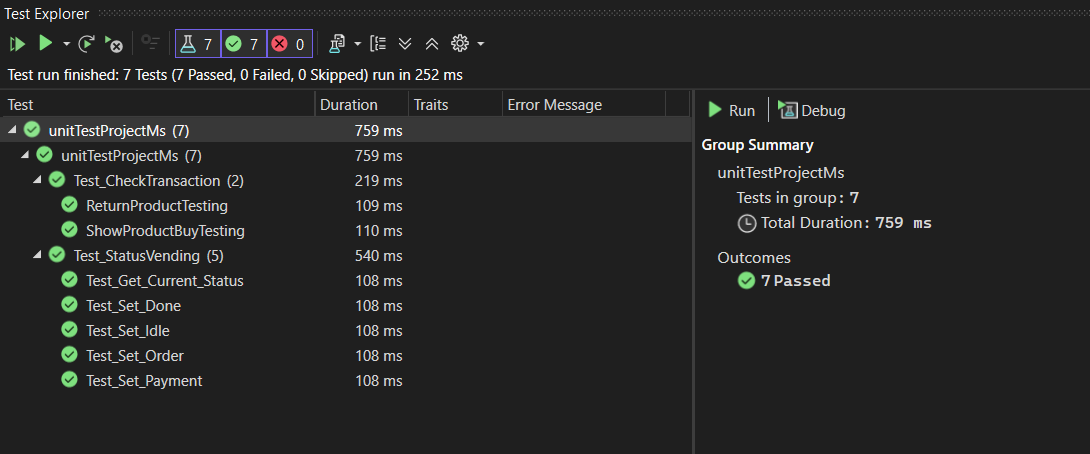
Lalu di classTest status Vending



Kita cek apakah berhasil menset status ke idle menggunakan sama sama AreEqual

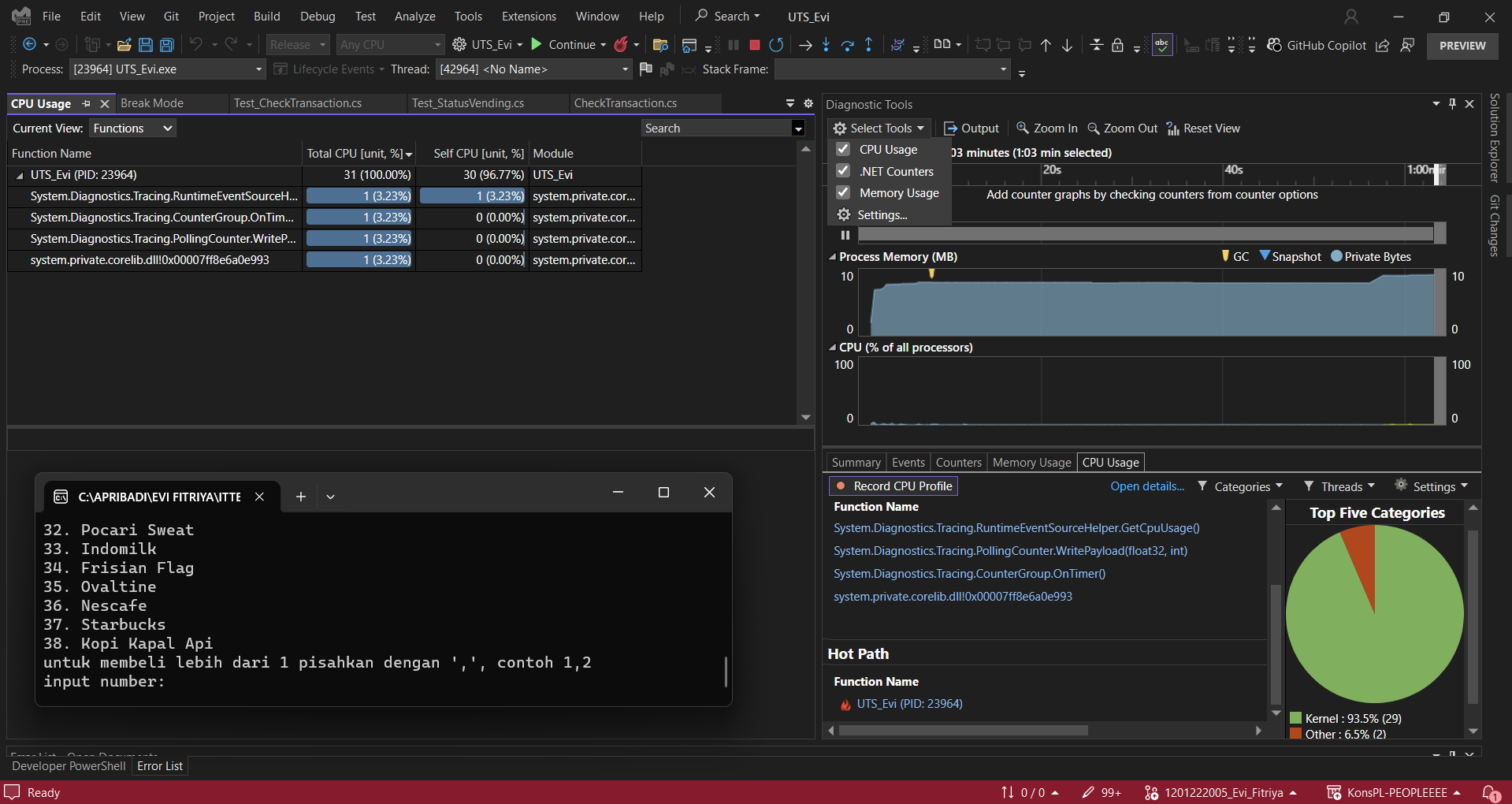


Dan semuanya juga sama.

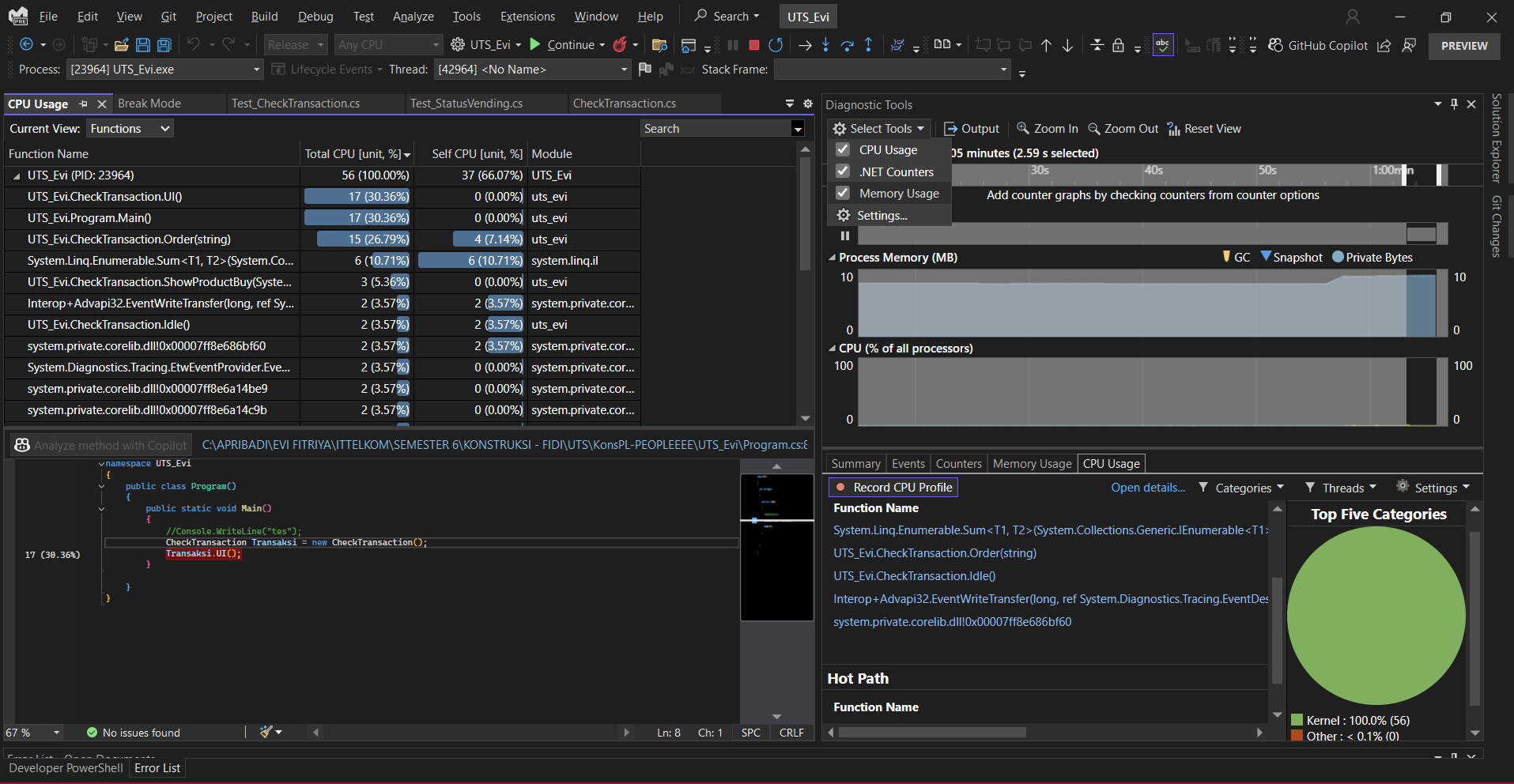
Berikut hasil unit testnya:

Menunjukkan bahwa semua test berhasil atau passed.

# HASIL PERFOMANCE TESTING

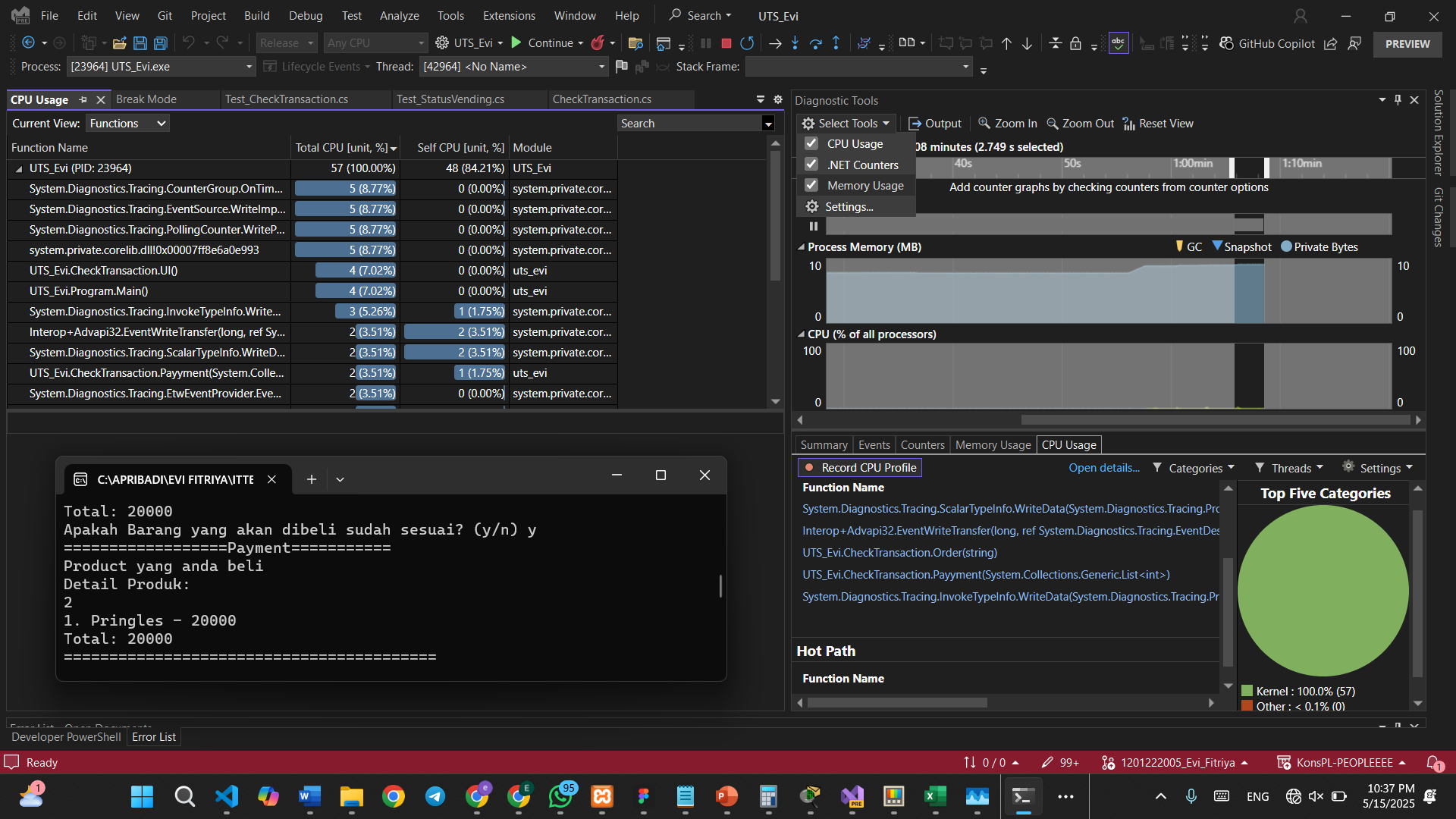


Berdasarkan data diatas ketika baru pertama menjalankan aplikasi, akan menggunakan kernel sebesar 93.5% dan other nya 6.5%, lalu fungsi dengan konsumsi CPU terbanyak berada pada UTS\_Evi

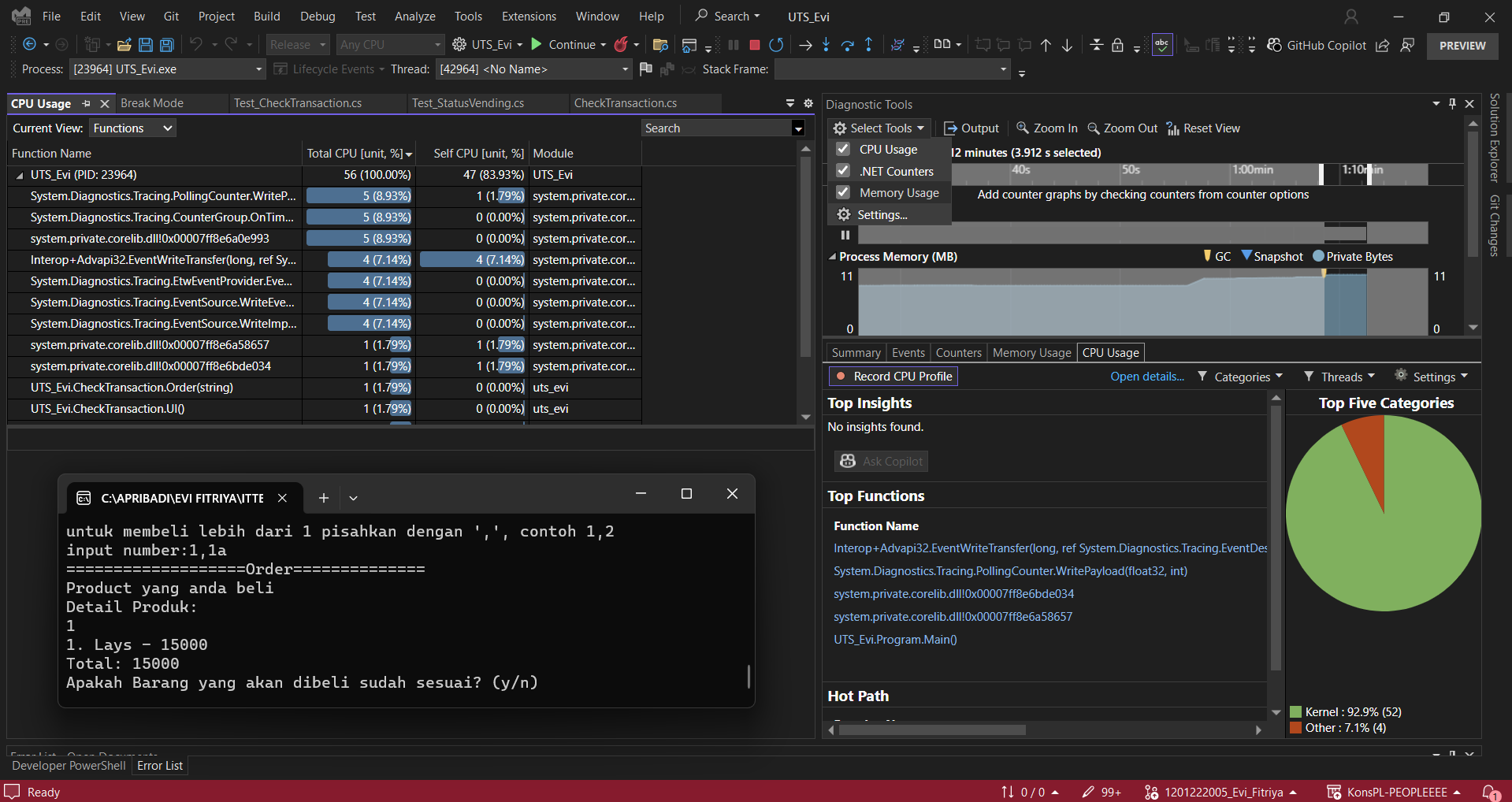


Kemudian disini fungsi UI() menggunakan 30.36% dari total CPU lalu Main() milik class program juga di 30% lebih, kemudian di Order() di 26.79%.

Lalu Ketika run aplikasi di inputan selannjutnya ini menggunakan kernel sebanyak 100%



Pada inputan ini atau perjalanan selanjutnya, tidak ada yang benar-benar menggunakan CPU banyak, tetpi rata namun masih 57, dan kernel digunakan sebanyak 100%



Lalu ini kembali lagi ke pertama aplikasi di mulai atau kondisi idle, jadi kernel digunakan sekitar 90% lebih seperti sebelumnya.