**Laporan Akhir Pengujian Perangkat Lunak(Software Resto kita, Pemesanan Obat, my.unpam.ac.id)**

****

Nama : Tubagus Ganendra Krisna Priya

NIM : 2239099057

Dibuat untuk memenuhi tugas Mata Kuliah PKPL kelas B

**Fakultas Teknik Industri Prodi Informatika**

**Universitas Ahmad Dahlan Yogyakarta**

* **Laporan Review Kontrak Proyek TI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Kriteria Review | Bukti ScreenShcoot | Hasil Review | Keterangan |
| 1. | Spesifikasi Sistem |  | * Pasal 4.2 Bag B   Aplikasi Andorid dengan Minimal Spesifikasi OS 4.0.   * Pasal 1 Bag A   Fungsi Check-in dan Order. | Minimal spesifikasi yang dikembangkan pada Aplikasi RestoKita adalah sebuah software yang dapat dijalankan dengan minimal OS Android 4.0. Fitur -fitur yang nantinya akan terdapat pada aplikasi adalah fitur Check-in restaurant dan Order |
| 2. | Time Line |  | * Pasal 5   Pengerjaan dilaksanakan pada 1 Desember 2014 dan selesai pada 1 Juni 2015   * Pasal 1 Bagian C   Sprint Planning diadakan setiap tanggal 1 dan 15 setiap bulan selama waktu pengerjaan | Waktu yang diberikan oleh **Pihak Kedua** **(client)** kepada **Pihak Pertama** **(Pengembang)** adalah selama 6 bulan yakni dari tanggal 1 Desember 2014 - 1 Juni 201. Kemudian akan diadakan sprint planning setiap 2 bulan sekalli yakni pada tangal 1 dan 15 setiap bulannya. |
| 3. | Biaya |  | * Pasal 3 Bag B   Jumlah Nominal yang disepakati dalam pengadaan, pembuatan, pengembangan perangkat lunak yang telah disepakati adalah sebesar Rp 120.000.000,00 (seratus dua puluh juta rupiah) dan akan dibayarkan oleh PIHAK KEDUA secara bertahap di setiap akhir Sprint Retrospective   * Pasal 7 Bag H   Anggaran tiap Sprint adalah sebesar Rp.20.000.000 (dua puluh juta rupiah). | Nominal yang telah ditetapkan dan akan dibayarkan oleh **Pihak Kedua (client)** kepada **Pihak Pertama (Pengembang)** adalah Rp.120.000.000.  Pembayaran akan dilakukan sebanyak 6 tahap. Pembayaran akan dilakukan setiap akhir Sprint Retrospective yang dilaksanakan setiap tanggal 1 dan 15 setiap bulannya selama masa pengerjaan yang telah ditentukan. Setiap akhir Sprint Retrospective **Pihak Kedua (Client)** akan membayarkan sebesar Rp 20.000.000 kepada **Pihak Pertama (Pengembang)**. |

* **Manual Testing Scenario Software Restoku - Kasir Warung Makan**

|  |  |
| --- | --- |
| Project Name : Restoku – Kasir Warung Makan | Reviewed By : Tubagus Ganendra |
| Module Name : User | Reviewed Date :04/11/2022 |
| Author : PT. Restoku Andalan Indonesia | |

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario ID | Module | Requirement | Test Scenario |
| TS01 | User | R1 Restoku Halaman Login | Check Launching Application |
| TS02 | User | R1 Restoku Halaman Login | Check Capture Form |
| TS03 | User | R1 Restoku Halaman Login | Check User Login |
| TS04 | User | R2 Restoku Halaman Utama User | Check Navigasi menu dan home page |
| TS05 | User | R3 Insert | Check Menambah Menu |
| TS06 | User | R3 Delete | Check Menghapus Menu |

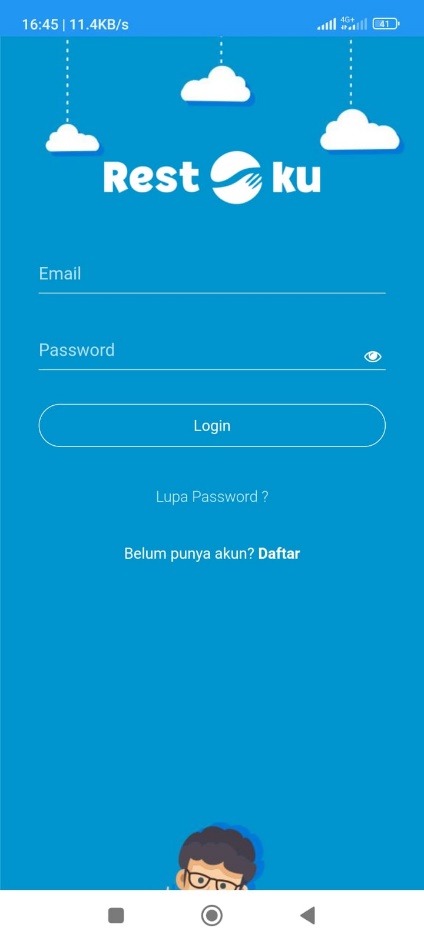
* **Laporan Manual Testing Scenario Software Restoku - Kasir Warung Makan**

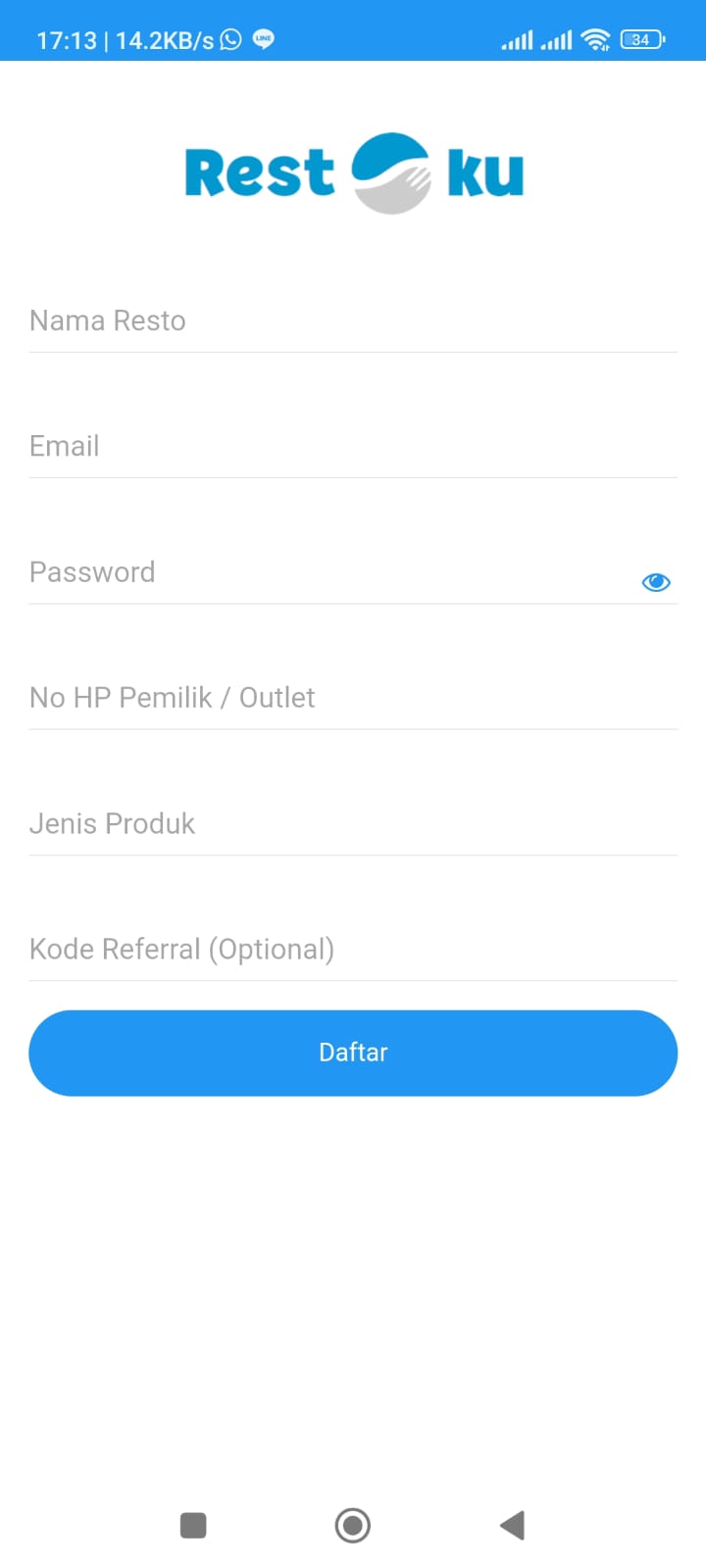
|  |  |
| --- | --- |
| Project Name : Restoku – Kasir Warung Makan | Reviewed By : Tubagus Ganendra |
| Module Name : User | Reviewed Date :04/11/2022 |
| Author : PT. Restoku Andalan Indonesia | |

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario ID | Module | Requirement | Test Scenario |
| TS01 | User | R1 Restoku Halaman Login | Check Launching Application |
| TS02 | User | R1 Restoku Halaman Login | Check Capture Form |
| TS03 | User | R1 Restoku Halaman Login | Check User Login |
| TS04 | User | R2 Restoku Halaman Utama User | Check Navigasi menu dan home page |
| TS05 | User | R3 Insert | Check Menambah Menu |
| TS06 | User | R3 Delete | Check Menghapus Menu |

**Manual Testing**

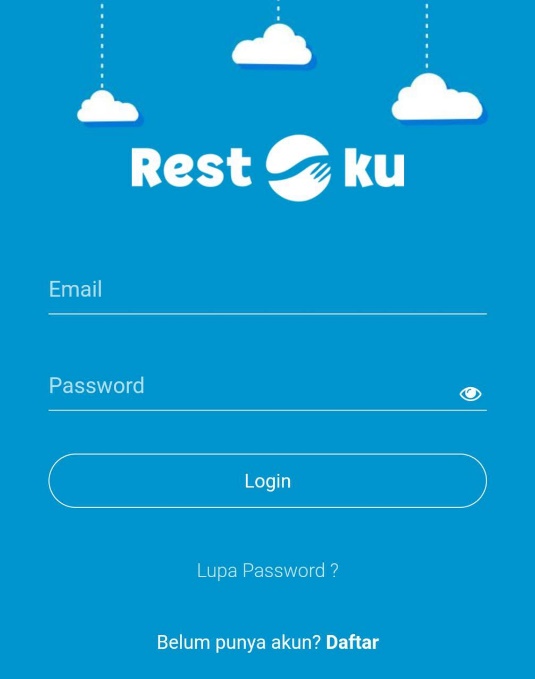
TS01 = Check Launching Aplikasi



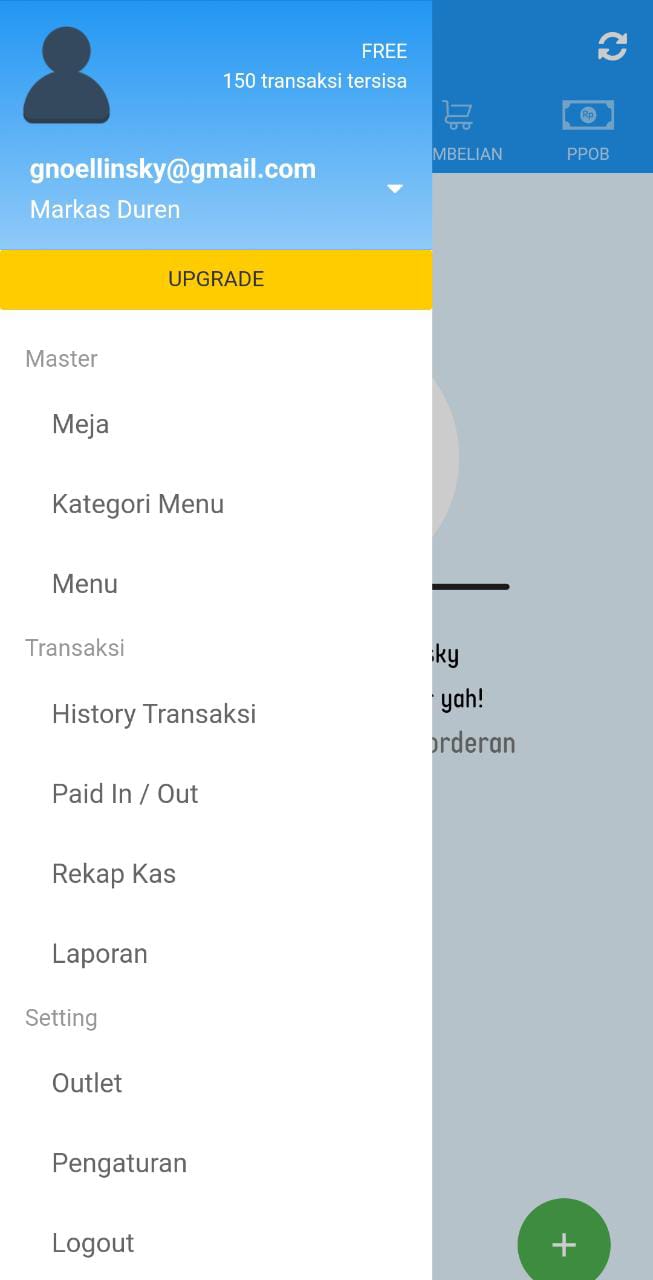
TS02 = Check Capture Form

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 |
| Nama Resto | Yes | Yes | Yes | Yes | Yes | Yes | No | No | No | No | No | No |
| Email | Yes | No | Yes | Yes | Yes | Yes | Yes | No | No | No | No | No |
| Password | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | No | No | No | No |
| No HP | Yes | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | No | No | No |
| Jenis Produk | Yes | Yes | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | No | No |
| Kode Referral | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | No |
| Result |  | Error | Error | Error | Error |  | Error | Erorr | Error | Error | Error | Error |

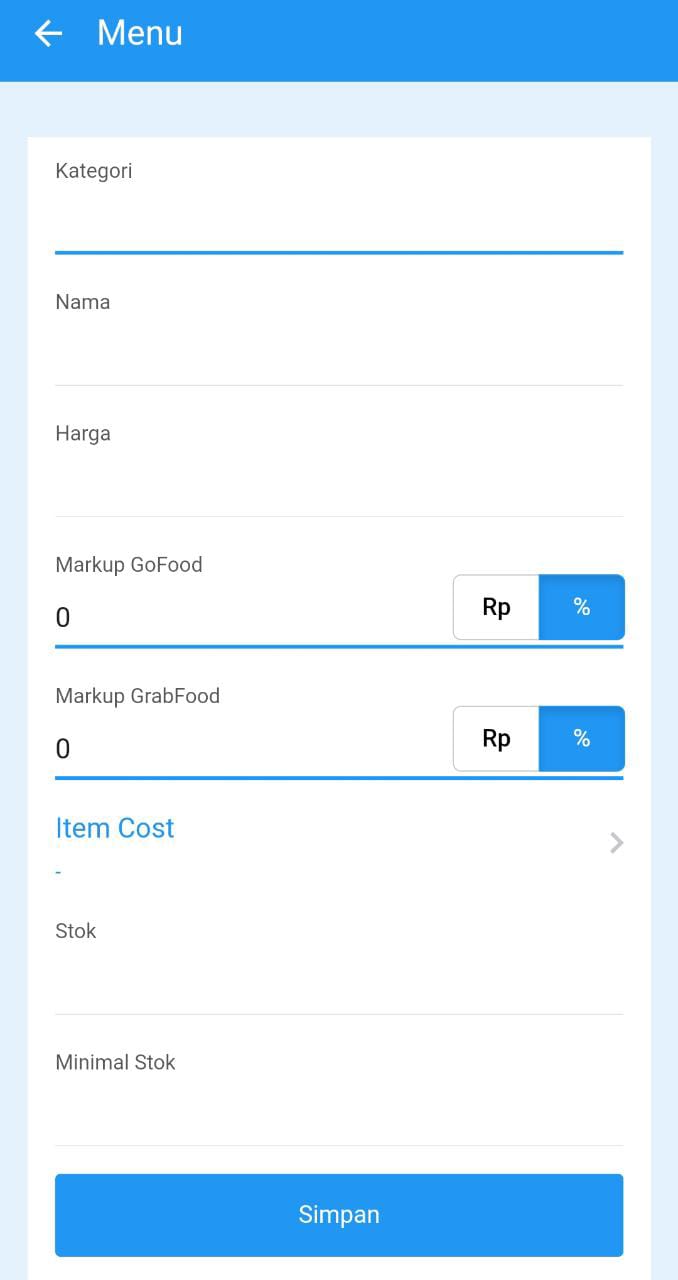
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Inputs | C1 | C2 | C3 | C4 | C5 | C6 |
| Email | Yes | Yes | Yes | No | Yes | No |
| Password | Yes | No | Yes | Yes | No | No |
| Login | Yes | Yes | No | Yes | No | No |
| Result |  | Error | Error | Error | Error | Error |

TS03 = Check User Login

TS04 = Check Navigasi menu dan home page

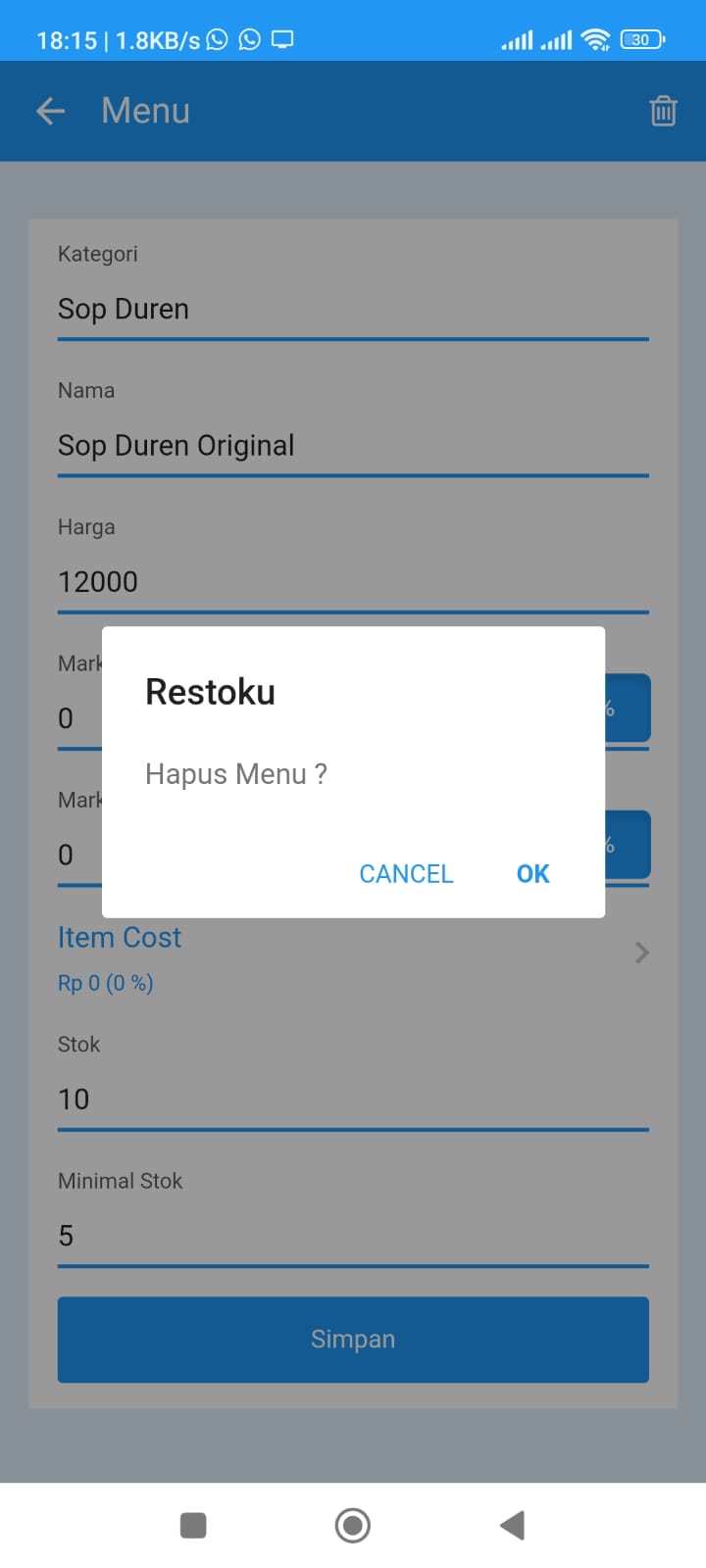


TS05 = Check Insert Menu



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Inputs | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 |
| Kategori | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | No | No |
| Nama | Yes | No | Yes | Yes | Yes | Yes | Yes | Yes | No | No |
| Harga | Yes | Yes | No | Yes | Yes | Yes | Yes | Yes | No | No |
| Markup Gofood | Yes | Yes | Yes | No | Yes | Yes | Yes | No | Yes | No |
| Markup Grabfood | Yes | Yes | Yes | Yes | No | Yes | Yes | No | Yes | No |
| Stock | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes | No | No |
| Minimal Stock | Yes | Yes | Yes | Yes | Yes | Yes | No | Yes | No | No |
| Results |  | Error | Error |  |  | Error | Error |  | Error | Error |

TS06 = Check Delete Menu

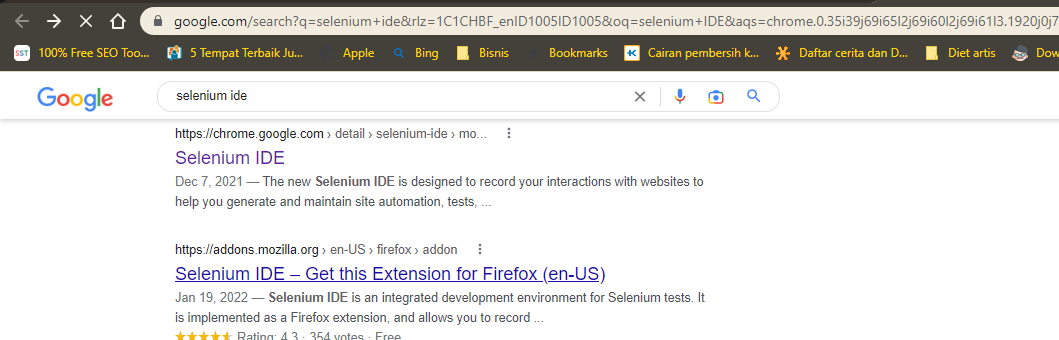


Kesimpulan :

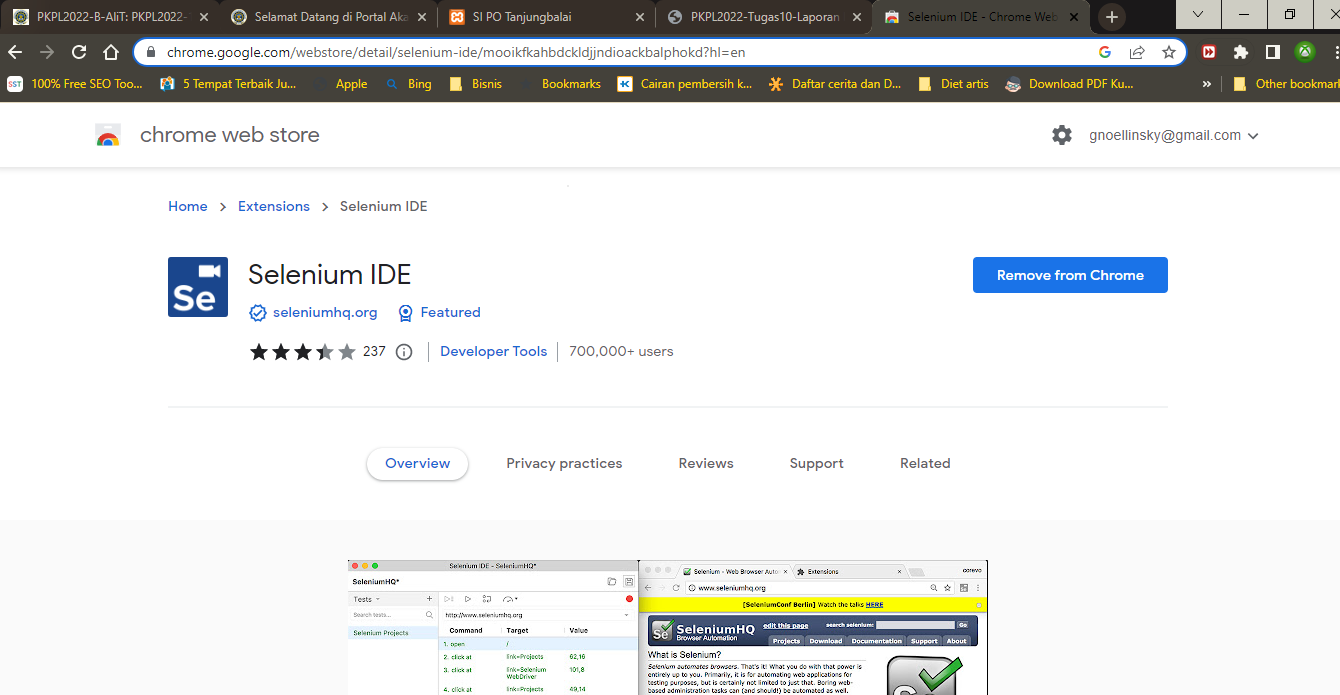
Pada TS02 Capture Form dapat berhasil bila semua form telah terisi. Pada TS03 apabila E-mail dan Password benar dan klik tombol login maka login akan berhasil. Kemudian pada TS05 untuk insert menu dapat dilakukan apabila semua form kecuali form Markup Gofoof dan Markup Grabfood telah terisi. Lalu pada TS06 untuk dapat mendelete menu maka dilakukan langkah-langkah klik menu makanan lalu klik delete, kemudian klik OK dan menu akan terhapus.

* **Pengujian Otomatis Aplikasi Pemesanan Obat dengan Selenium IDE**

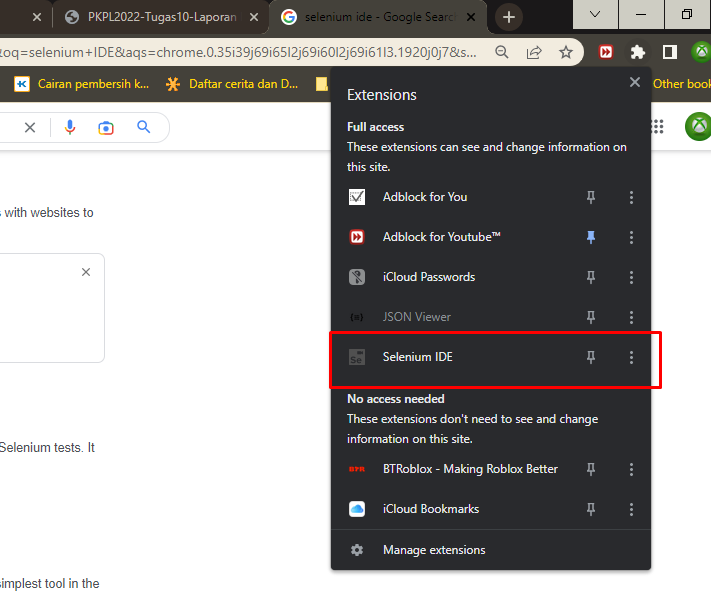
Langkah pertama yaitu buka Chrome, kemudian ketikan selenium IDE pada kolom pencarian.



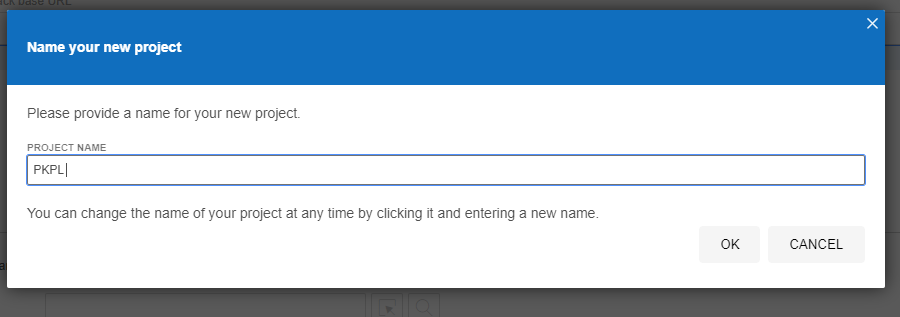
Kemudian klik pada chrome web store, dan install. Setelah itu klik add to chrome extension.



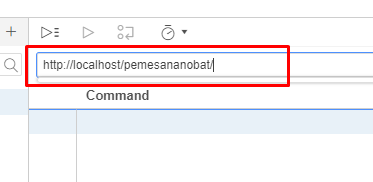
Setelah terinstal, klik menu extensions dan pilih Selenium IDE



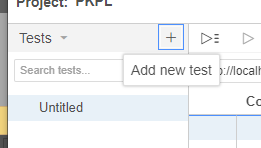
Setekah itu, tab Selenium IDE akan terbuka. Untuk memulai proyek ,pertama-tama klik create new project, kemudian berikan nama untuk proyek yang akan dikerjakan.



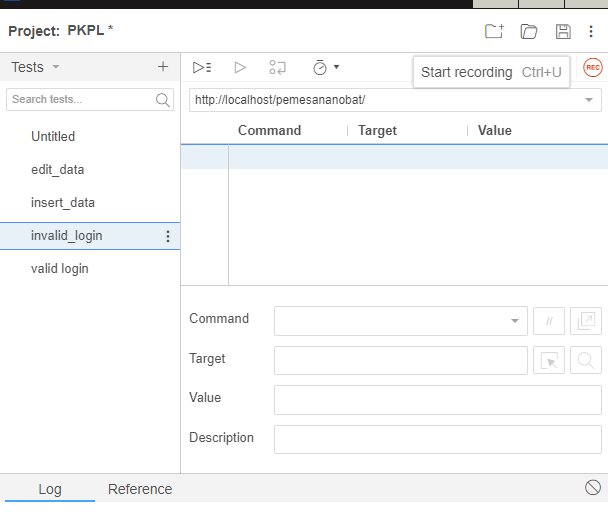
Masukan link URL untuk aplikasi yang akan diuji coba.



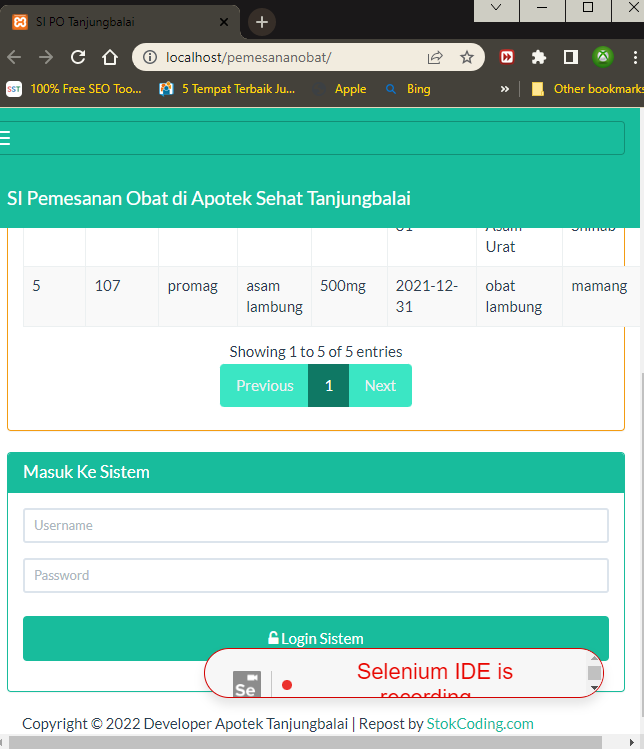
Klik ikon “+” untuk membuat test baru dan berikan nama untuk pengujian yang akan dilakukan.



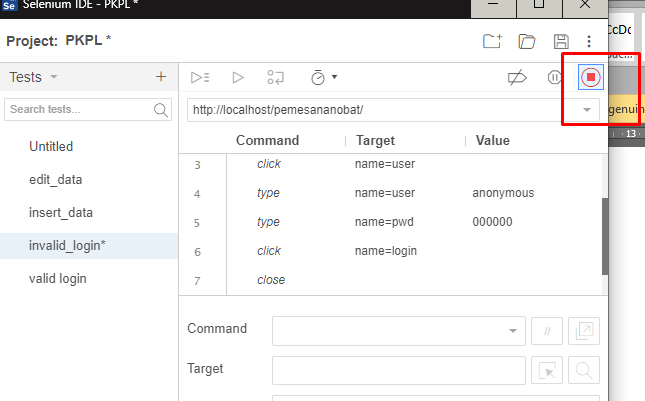
Untuk memulai pengujian, klik menu Record pada nama pengujian yang dipilih.



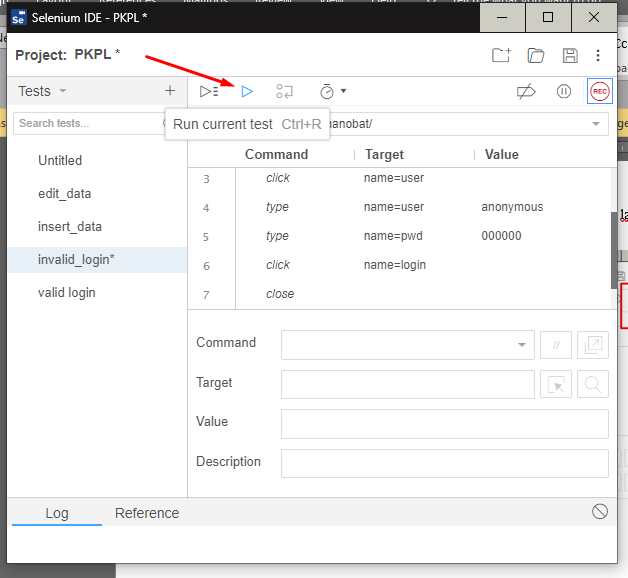
Kemudian nantinya selenium IDE akan mebuka tab baru yang menuju kepada web yang akan diuji. Setelah itu lakukan pengujian sesuai dengan fitur yang akan di uji cobakan.



Jika sudah selesai, kik tombol close, dan hentikan recording. Setelah itu, langkah pada pengujian akan ditampilkan.



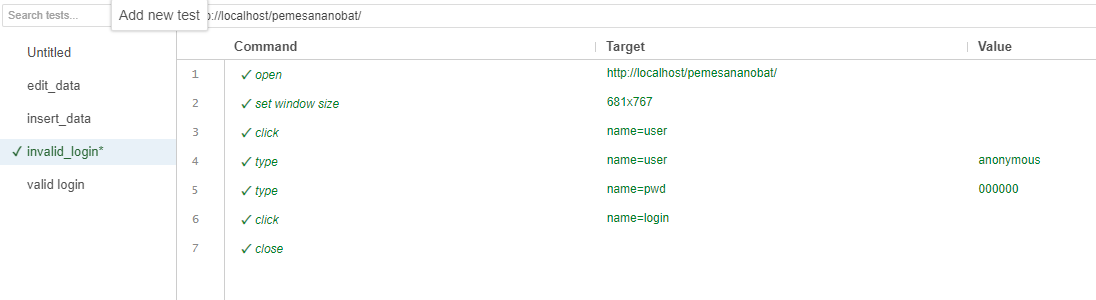
Untuk melakukan uji otomatis, klik tombol run test.



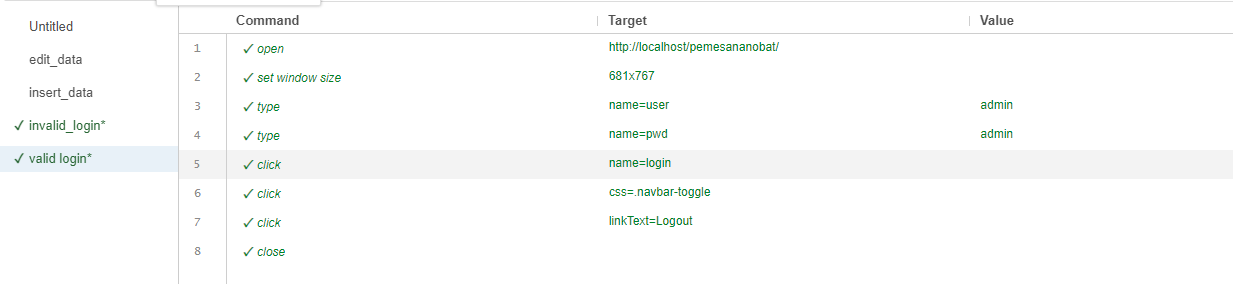
Kemudian Selenium IDE akan menjalankan test secara otomatis dengan membuka tab baru yang merupakan website yang akan di ujicoba. Setelah uji coba selesai hasil pengujian akan ditampilkan.

Berikut adalah hasil pengujian yang telah dilakukan kepada aplikasi Pemesanan Obat. :

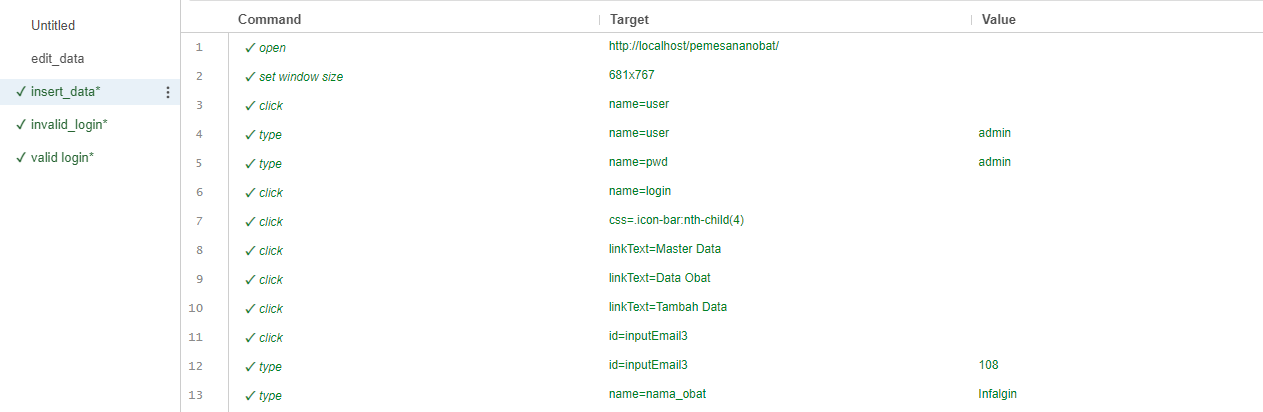
* **Invalid Login**

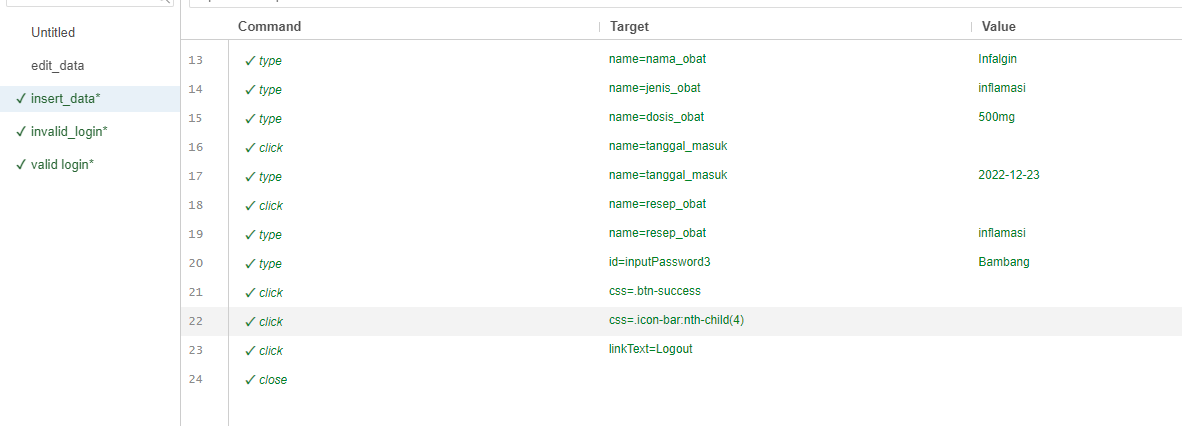


* **Valid Login**

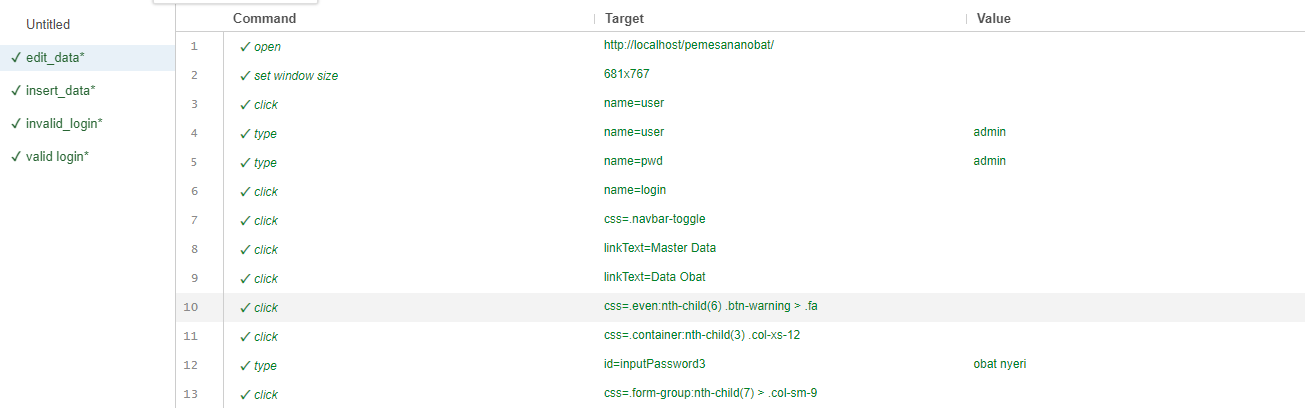


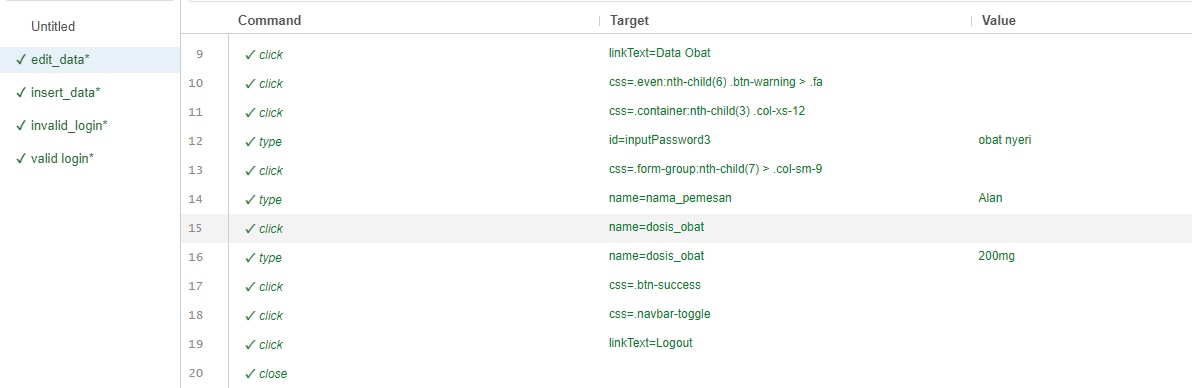
* **Insert Data**





* **Edit Data**

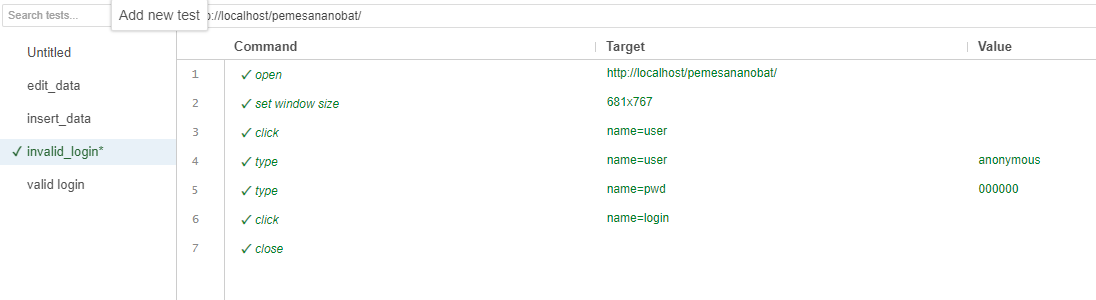




* **Laporan Pengujian Otomatis Aplikasi Pemesanan Obat dengan Selenium IDE**

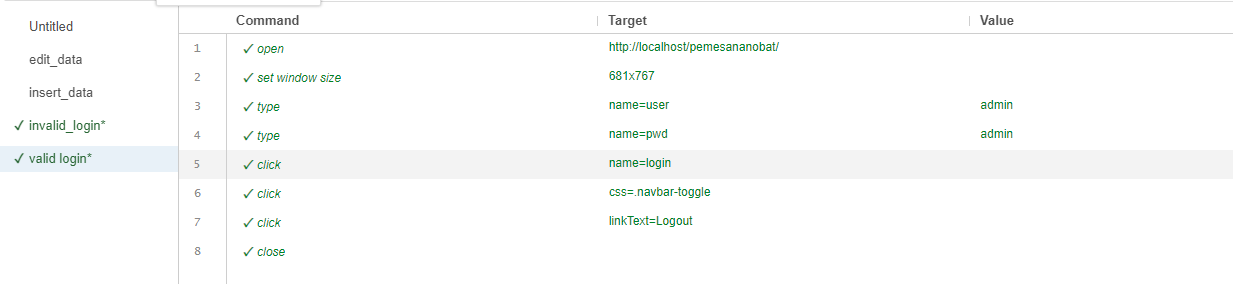
Ada 4 fitur yang diujikan dalam pengujian otomatis, dengan hasil pengujian yang telah dilakukan kepada aplikasi Pemesanan Obat. :

* **Invalid Login**



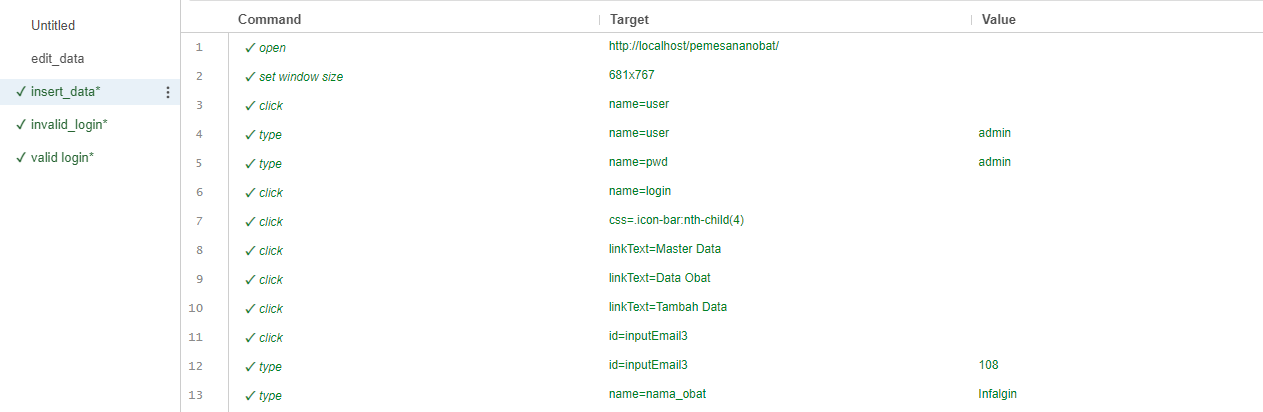
Pada pengujian invalid login, terdapat 7 langkah pengujian. Tujuan dari pengujian ini adalah untuk mengetahui apabila ada kesalahan dari username dan password yang diinputkan pada aplikasi apakah user tetap dapat mengakses aplikasi atau tidak dengan hasil uji yang diharapkan adalah *user tidak akan dapat mengakses aplikasi*. Pada langkah pengujian ke-4 dan ke-5 adalah langkah ketika memasukan username dan password yang salah dan hasilnya adalah sukses dan tidak ada kejanggalan pada system.

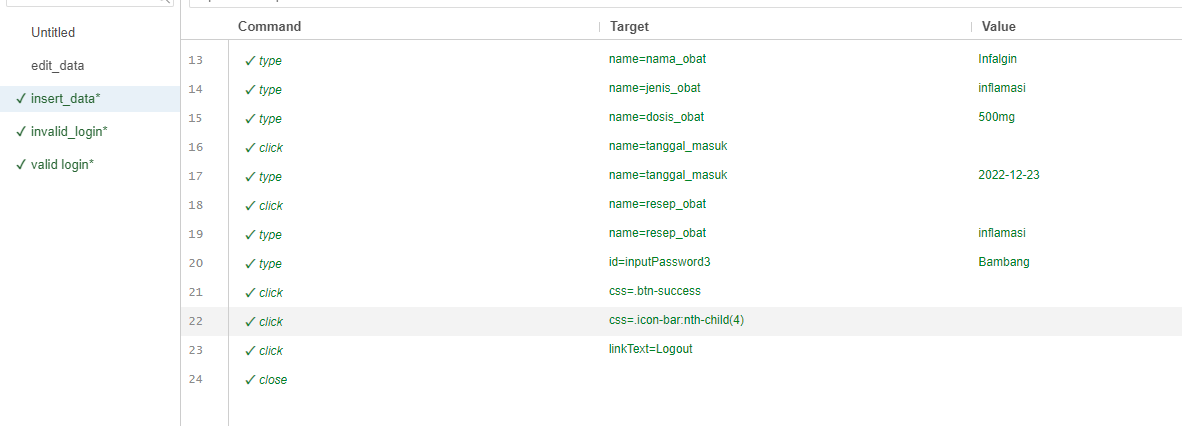
* **Valid Login**



Kemudian pengujian yang kedua adalah pengujian valid login yang memiliki 8 langkah pengujian. Tujuan dari pengujian ini adalah untuk mengetahui bila username dan password yang diinputkan pada aplikasi adalah username dan password yang terdaftar, apakah user akan dapat mengakses data pada aplikasi atau tidak dengan hasil yang diharapkan adalah *user dapat mengakses data pada aplikasi*. Pada langkah pengujian ke-3 dan ke-4 adalah langkah ketika memasukan username dan password yang terdaftar dan hasilnya adalah user dapat mengakses data dan tidak ada kejanggalan pada system. Kemudian pada langkah ke-7 adalah pengujian logout dan user berhasil logout.

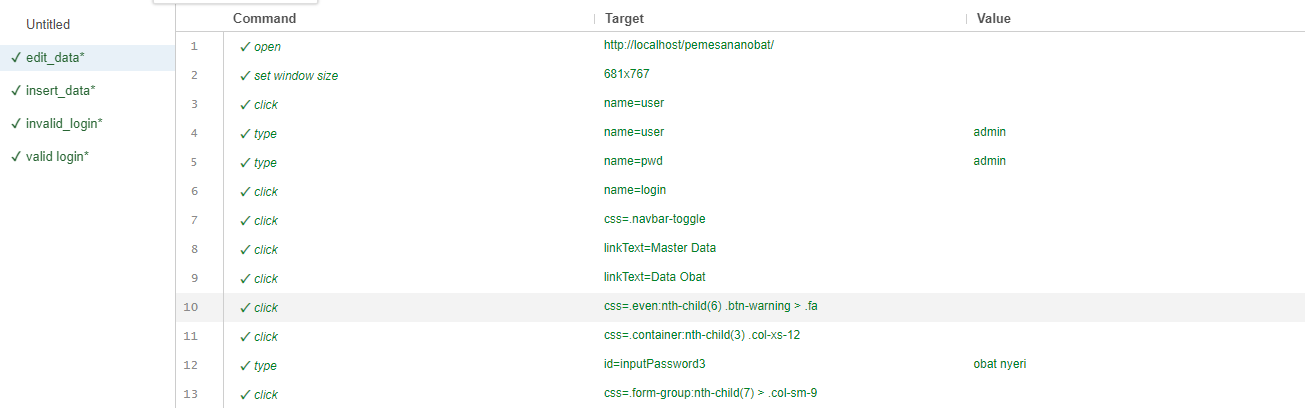
* **Insert Data**

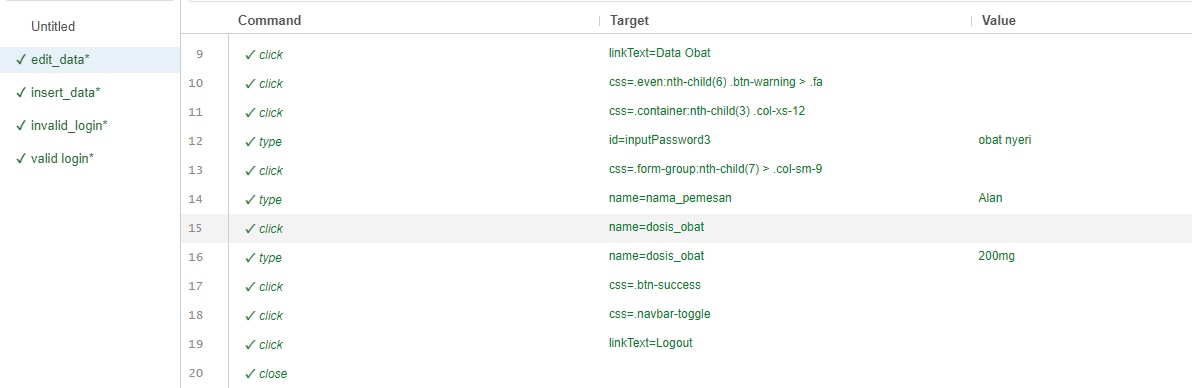




Pada pengujian ke-3, yang menjadi acuan uji coba adalah fitur menambah data pada aplikasi pemesanan obat. Tujuan dari uji coba insert\_data adalah untuk mengetahui apakah fitur insert data yang ada pada aplikasi berjalan dengan baik dengan hasil yang diharapkan adalah *Data dapat ditambahkan pada aplikasi*. Pengujian ke-12 sampai ke-20 adalah langkah pengujian untuk insert data, dengan hasil pengujiannya adalah sukses dan data berhasil ditambahkan.

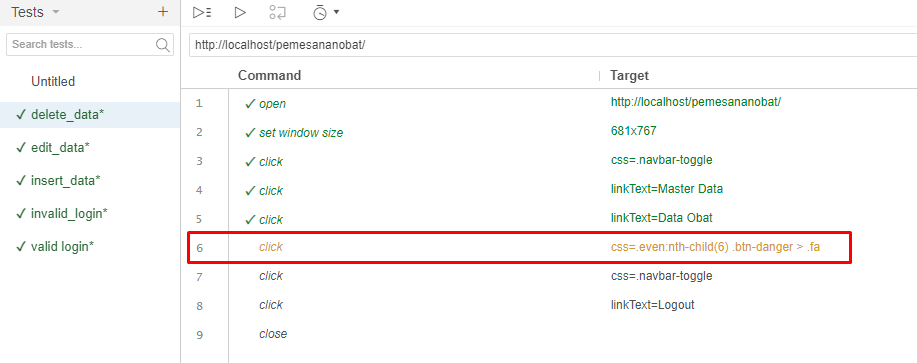
* **Edit Data**





Pada pengujian ke-4, yang menjadi acuan uji coba adalah fitur mengedit data pada aplikasi pemesanan obat. Tujuan dari uji coba edit\_data adalah untuk mengetahui apakah fitur editdata yang ada pada aplikasi berjalan dengan baik dengan hasil yang diharapkan adalah *Data dapat diedit atau diupdate pada aplikasi*. Langkah pengujian ke-12 sampai ke-16 adalah langkah pengujian untuk edit data, dengan hasil pengujiannya adalah sukses dan data berhasil diubah.

* Kejanggalan pada pengujian
* Delete data



Pada pengujian delete data, uji coba memiliki 9 langkah. Sebelum melakukan pengujian otomatis, langkah-langkah pengujian akan ditentukan terlebih dahulu setelah itu baru pengujian otomatis dapat dilakukan. Pada langkah ke-6 data yang akan dihapus tidak dapat ditemukan karena data sudah terlebih dahulu terhapus pada penentuan langkah-langkah pengujian otomatis. Hal ini membuat pengujian otomatis tidak dapat dilanjutkan, dan langkah pengujian terjebak pada langkah ke-6. Menurut pendapat saya, pengujian otomatis dengan Selenium IDE tidak cocok untuk menguji fitur menghapus data.

* **Pengujian Non-Fungsional Stress dengan JMeter pada aplikasi my.unpam.ac.id**

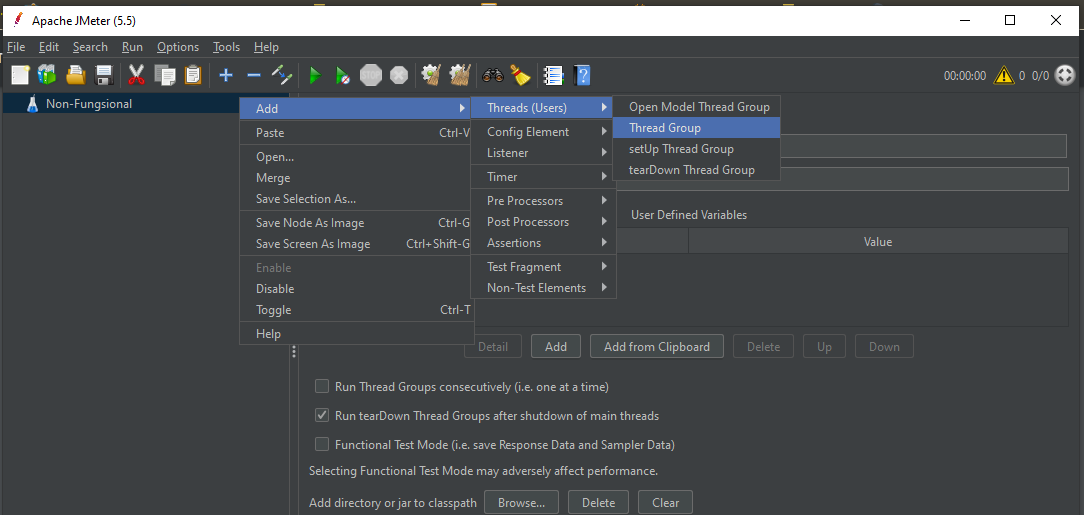
Metode pengujian yang akan dilakukan adalah pengujian Non-Fungsional Stress test dengan menggunakan aplikasi Apache JMeter. Tujuan pengujian Stress Test adalah untuk memeriksa kemampuan dari aplikasi yang akan ditest dalam menerima traffic dari luar, dengan tujuan menguji aplikasi/website apakah akan terjadi down atau tidak saat banyak user yang mengakses aplikasi, website tersebut.

* **Rangkaian Pengujian**

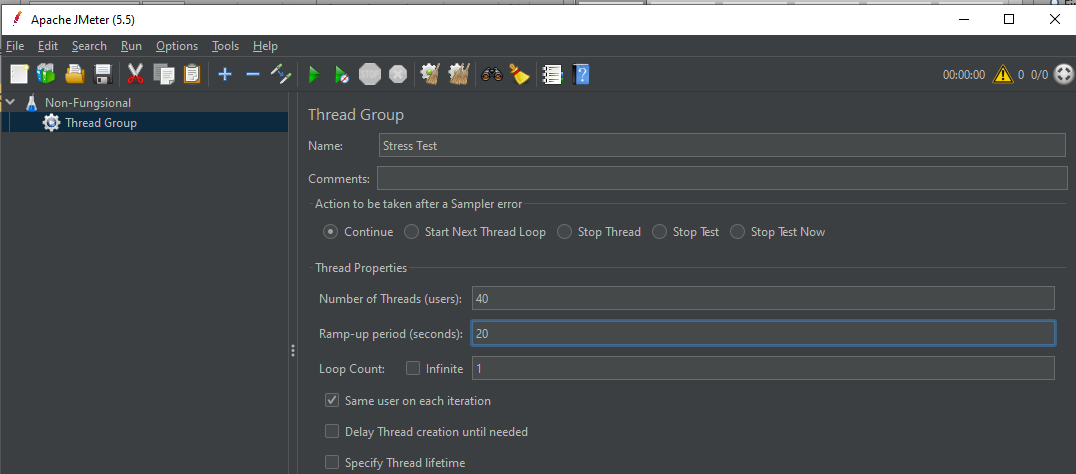
Pengujian yang akan dilakukan adalah :

* Threads Test :
* Homepage
* Login
* Particular Element Test
* **Tahapan Pengujian**

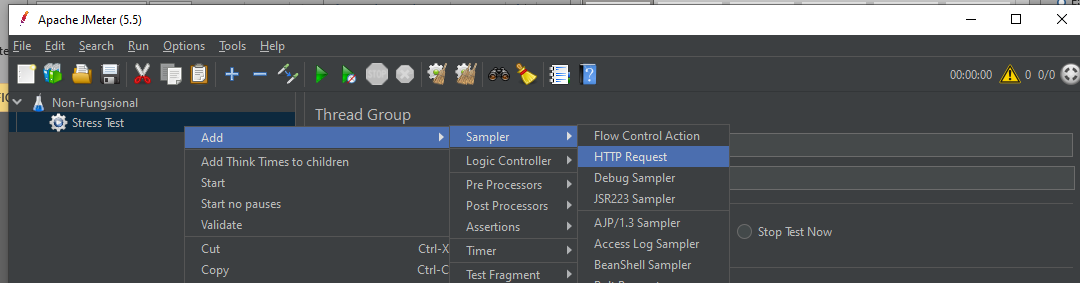
Langkah pertama, buka aplikasi Apache JMeter, kemudian klik Kanan dan pilih Add => Threads(User) => Thread Group.



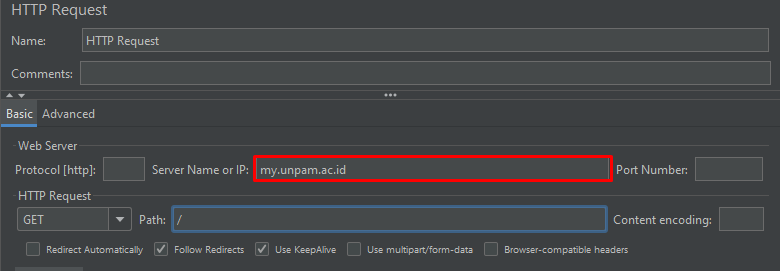
Pada Menu Threads nama pengujian saya isi dengan “Stress Test”. Kemudian untuk Number of User saya isi sebanyak 40, lama periode sebanyak 20 dan loop sebanyak 1.



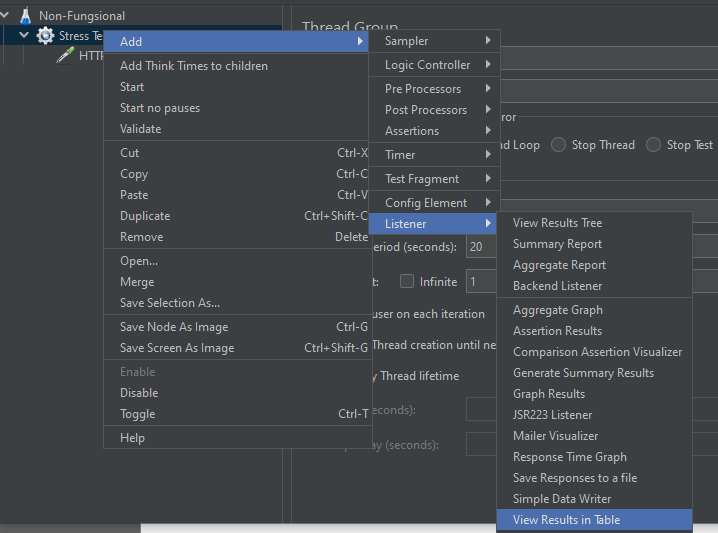
Kemudian, karena aplikasi yang akan diujikan berbasis web maka klik kanan pada Stress Test, klik Add => Sampler=> HTTP Request.



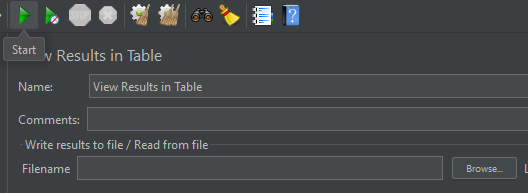
Pada HTTP Request , masukan server address dari aplikasi yang akan diujikan. Pada bagian Path karena pengujian pertama yang akan dilakukan adalah Homepage test maka isi dengan “/”



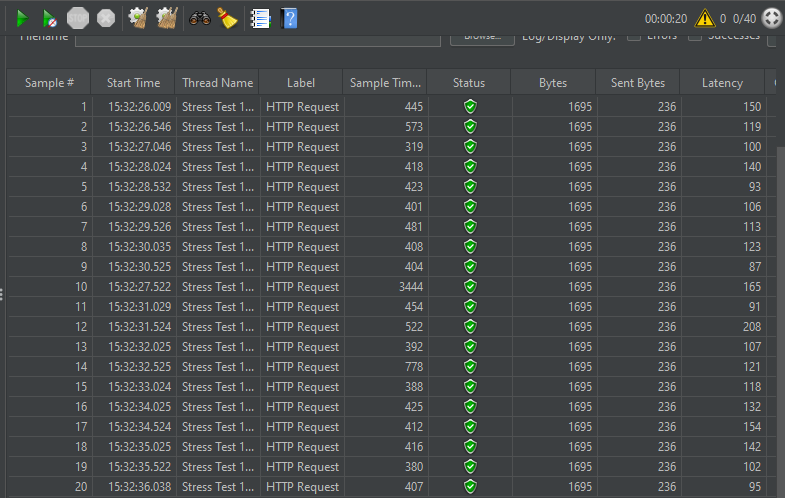
Untuk melihat hasil pengujian yang akan dilakukan, klik kanan pada Stress Test kemudian pilih Add=>Listener=>View Result in Table. Maka nanti Hasil pengujian akan ditampilkan dalam bentuk table.

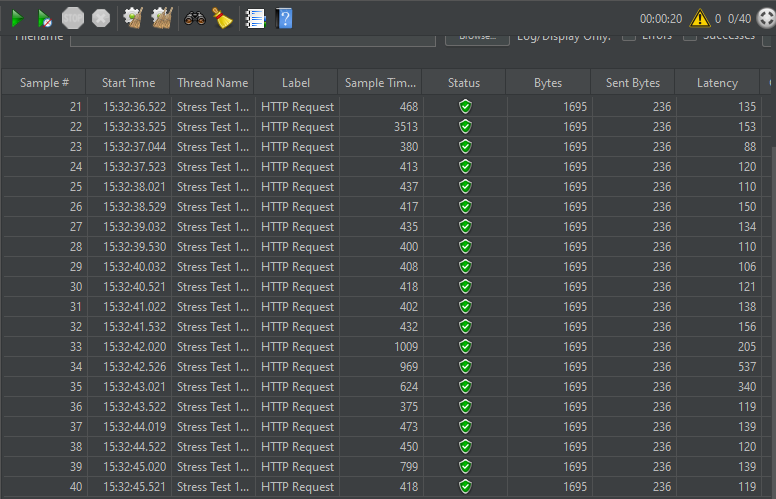


Untuk melakukan pengujian klik ikon start, namun sebelum melakukan pengujian alangkah baiknya menyimpan hasil pengujian terlebih dahulu.

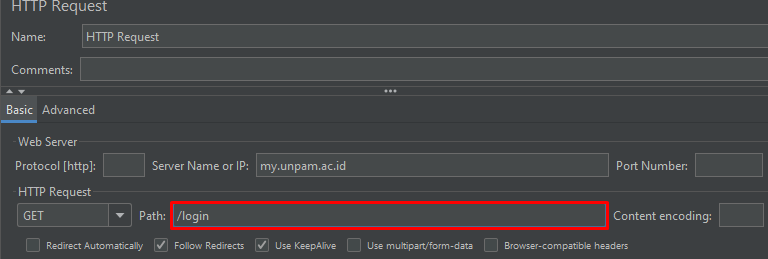


Berikut adalah hasil pengujian pada Homepage

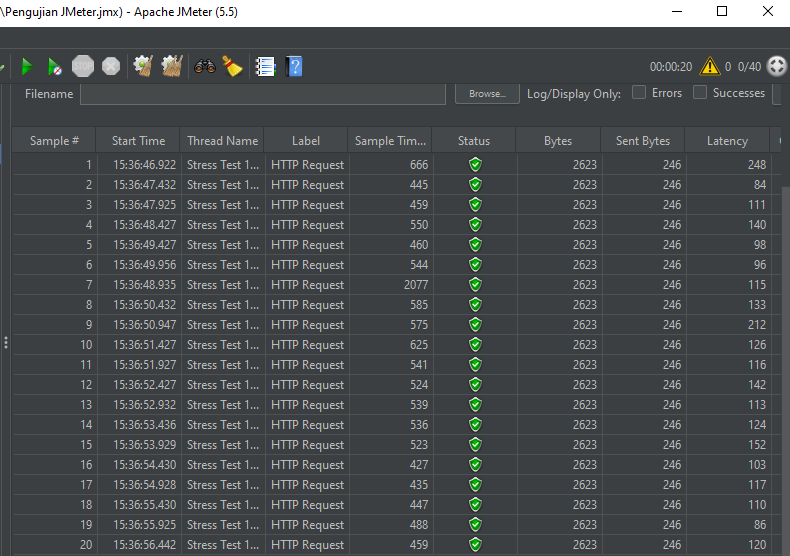


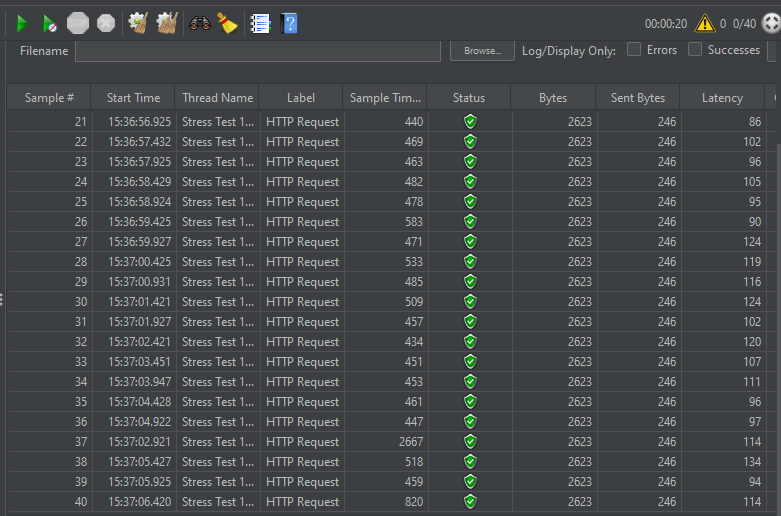


Sebanyak 40 user virtual yang diujikan dengan lama periode 20 detik dan 1 pengulangan hasilnya adalah semua user berhasil dan tidak ada yang gagal.



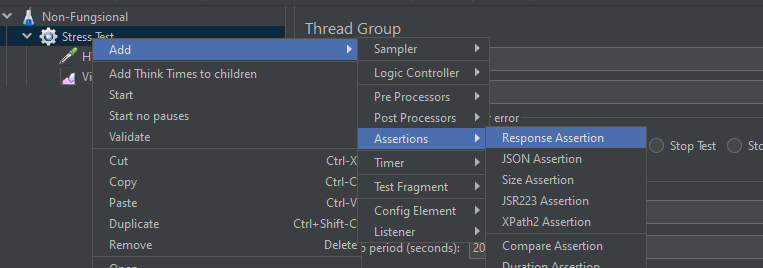
Selanjutnya adalah pengujian pada Login page dengan pengaturan yang sama dengan homepage yakni sebanyak 40 user virtual dengan panjang periode 20 detik. Hasilnya :



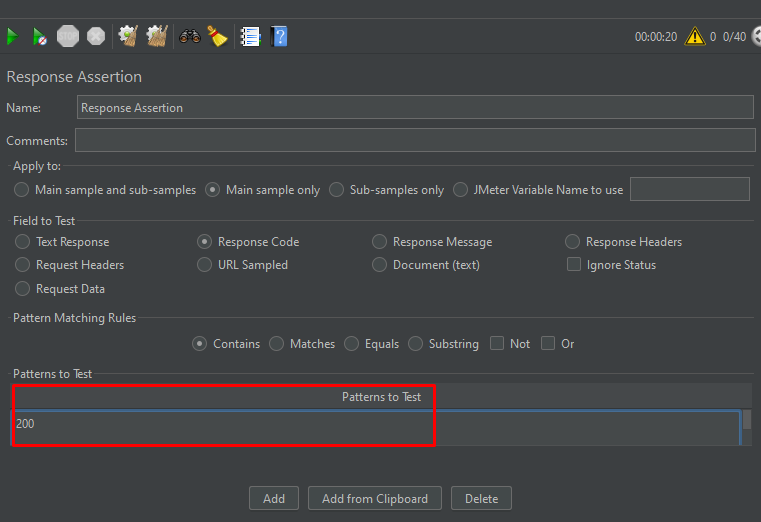


Particular Element Test

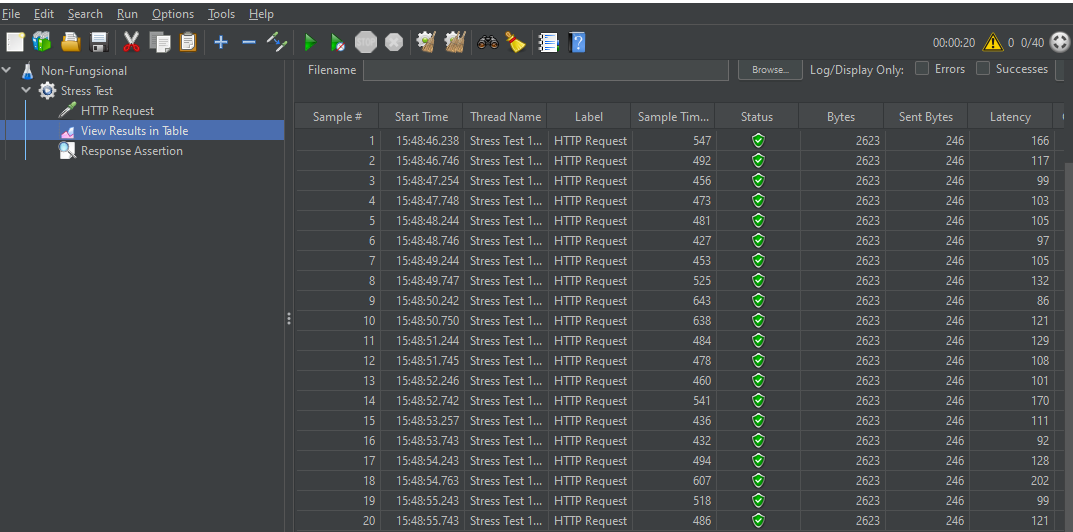
Untuk melakukan pengujian Particular Element, klik kanan pada Stress Test, kemudian klik Add=>Assertions=>Response Assertion.



Lalu pada pengaturan Response Assertion lakukan seperti diawah ini. Untuk Pattern to Test sebelumnya klik Add, dan masukan 200 yang merupakan kode default dari “HTTP”.



Untuk melihat hasil pengujian klik menu “View Result in Table”.



Berikut adalah hasil pengujian Particular Element dengan seluruh User virtual berhasil tanpa mengalami kegagalan. Namun apabila Pattern to Test diganti dengan kode 201 maka hasilnya akan gagal seperti berikut :

