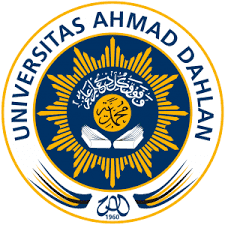
**Laporan Tugas Akhir Pengujian Perangkat Lunak Menggunakan Metode Black Box, White Box, dengan Tools Selenium IDE dan JMeter**



Nama : Sahda Salsabil

NIM : 2000018089

Kelas : B

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS AHMAD DAHLAN**

1. **Aplikasi Pendeteksi Jerawat**

**Pengujian Kualitas Manual Black Box Pada Aplikasi Pendetekasi Jerawat**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Skenario Pengujian | Test Case | Hasil yang diharapkan | Hasil Test Case | Kesimpulan |
| 1 | Memasukan citra wajah berjerawat pada fitur Tambah Gambar. |  | Sistem akan muncul windows file input untuk memasukkan data citra yang akan digunakan untuk memasukkan dalam system. |  | Sesuai dengan yang diinginkan pengguna. |
| 2 | Menentukan citra wajah berjerawat pada fitur Tambah Gambar untuk diidentifikasi. |  | Sistem akan menampilkann data yang telah dipilih pengguna. |  | Sesuai dengan yang diinginkan pengguna. |
| 3 | Melakukan deteksi citra pada fitur Deteksi Jerawat. |  | Sistem akan memunculkan jumlah jerawat dan jenis jerawat pada citra wajah berjerawat. Warna putih untuk jenis jerawat “whitehead” komedo,merah untuk jenis jerawat “papul”, ungu untuk jenis jerawat “pustul” dan kuning untuk jerawat “nodul”. |  | Sesuai dengan yang diinginkan pengguna. |
| 4. | Mencari produk yang digunakan dengan fitur saran produk. |  | Sistem akan memunculkan halaman website AVO untuk saran produk yang akan digunakan pengguna. |  | Sesuai dengan yang diinginkan pengguna. |

1. **Aplikasi DEBT Manager**

**Pengujian Kualitas Manual Pada Aplikasi Debt Manager**

1. **Diagram Path**
2. Listing Program

ContainerButtonProgress(

widget: ProgressButtonCustom(

colorIdle: Variables.primaryColor,

colorSuccess: Variables.colorGreen,

idleText: 'Pinjam',

idleIcon: Icon(Icons.send\_outlined, color: Colors.white),

failText: 'Pinjam Success'

failIcon: Icon(Icons.send\_outlined, color: Colors.white),

successText: 'Pinjam Success',

successIcon:

Icon(Icons.send\_outlined, color: Colors.white),

onPressed: () async {

FocusScope.of(context).unfocus();

setState(() => loading = ButtonState.loading);

if (\_formKey.currentState.validate()){

pin = await showPinDialog(

context,

Variables.primaryColor,

Variables.colorGreen,

(val) => {},

pin,

'Enter Your PIN');

pinFromDB =

await DatabaseService(uid: user.uid).getPin();

if (pin = pinFromDB){

tambahUtang = widget.utang.utang + int.parse(utang);

await DatabaseService(uid: user.uid).tambahUtangOrPiutang(

widget.utang.uid,

tambahUtang,

deskripsi,

'utang',

DateTime.now().microsecondsSinceEpoch,

int.parse(utang),

'tambah');

setState(() => loading = ButtonState.success);

await Future.delayed(Duration(seconds: 1));

Navigator.of(context).pushAndRemoveUntil(

MaterialPageRoute(

builder: (context) => BottomNavigation()),

(Route<dynamic> route) => false);

} else {

setState(() => loading = ButtonState.fail);

await Future.delayed(Duration(seconds: 1));

setState(() => loading = ButtonState.fail);

}

} else {

setState(() => loading = ButtonState.fail);

await Future.delayed(Duration(seconds: 1));

setState(() => loading = ButtonState.fail);

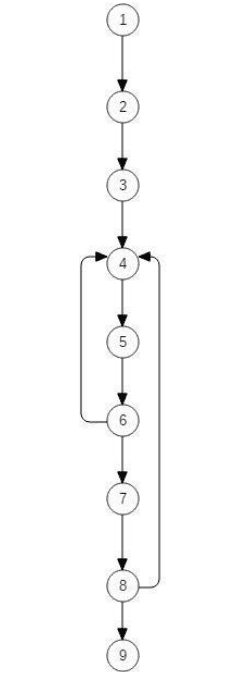
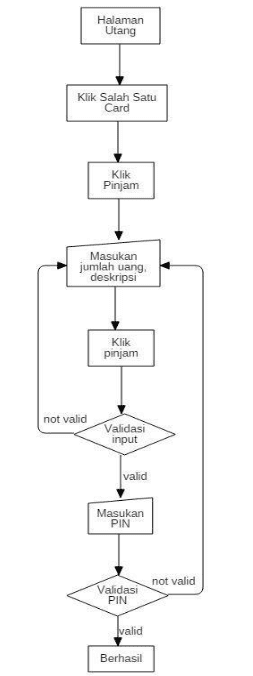
}

},

stateButton: loading,

),

1. Diagram Path



Untuk jalur yang digunakan :

* Jalur path 1 = 1-2-3-4-5-6-7

(1)Berarti halaman awal menampilkan halaman buat utang , (2)kemudian memasukan nama , deskripsi , jutuh tempo, setelah itu(3) klik button pinjam (4)lalu validasi input (valid) , (5)masukan pin(6) setelah itu validasi pin(valid),(7) kemudian berhasil.

* Jalur path 2 = 1-2-3-4-2-3-4-5-6-7

(1)Masuk kehalaman buat utang,(2) lalu masukan nama , deskripsi , jumlah jatuh tempo ,(3) lalu klik button pinjam , (4)validasi input (not valid) ,(2) lalu masukan nama , deskripsi , jumlah jatuh tempo ,(3) lalu klik button pinjam ,(4) validasi input (valid) , (5) masukan pin , (6) validasi pin (valid) ,(7) berhasil.

* Jalur path 3 = 1-2-3-4-5-6-2-3-4-5-6-7

(1)Berarti halaman awal menampilkan halaman buat utang , (2)kemudian memasukan nama , deskripsi , jutuh tempo, setelah itu(3) klik button pinjam (4)lalu validasi input (valid) , (5)masukan pin(6) setelah itu validasi pin (not valid),(2) kemudian memasukan nama , deskripsi , jutuh tempo, setelah itu(3) klik button pinjam (4)lalu validasi input (valid) , (5)masukan pin(6) setelah itu validasi pin (valid), (7) berhasil.

1. **Hasil Perhitungan Kompleksitas Cyclomatic**

V(G) = (E-N)+2

V(G) = jumlah region

E = jumlah edge yang ditentukan dengan gambar panah

N = jumlah simpul grafik (node) dengan gambar lingkaran

V(G) = (8-7)+2

V(G) = 3

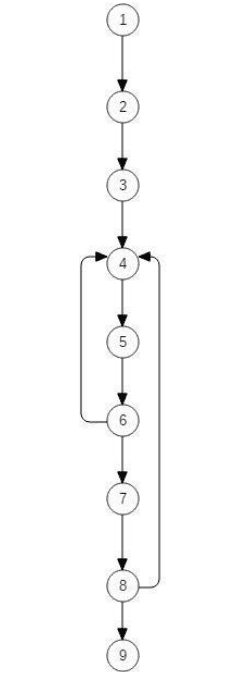
1. **Tabel Jalur Bebas Sejumlah Kompleksitas Cyclomatic**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 |  | 1 |  |  |  |  |  |
| 2 |  |  | 1 |  |  |  |  |
| 3 |  |  |  | 1 |  |  |  |
| 4 |  |  |  |  | 1 |  |  |
| 5 |  |  |  |  |  | 1 |  |
| 6 |  |  |  |  |  |  | 1 |
| 7 |  |  |  |  |  | 1 |  |

|  |  |
| --- | --- |
| Path | 1 |
| Jalur | 1-2-3-4-5-6-7 |
| Skenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid)  7. Berhasil |
| Hasil Pengujian | Berhasil |
| Path | 2 |
| Jalur | 1-2-3-4-2-3-4-5-6-7 |
| Skenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi,jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (not valid)  2. Masukkan nama,deskripsi,jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid)  7. Berhasil |
| Hasil Pengujian | Berhasil |
| Path | 3 |
| Jalur | 1-2-3-4-5-6-2-3-4-5-6-7 |
| Scenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (not valid)  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid) |

1. **Skenario uji dengan menggunakan Teknik pengujian BVA**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Kegiatan** | **Hasil yang diharapkan** | **Hasil** |
| 1 | User memilih halaman buat utang dan memasukan nama, deskripsi,jumlah uang,jatuh tempo, kemudian user memasukan PIN yang valid | Berhasil menambahkan utang dan data berhasil ditambahkan ke database | Sistem menampilkan pesan sukses menambahkan utang dan Kembali ke halaman utama |
| 2 | User memilih halaman buat utang dan tidak memasukan semua kolom input lalu user mengklik tombol pinjam. Lalu user mamasukan Kembali data input,setelah itu user memasukkan PIN yang valid. | Sebelum data berhasil ditambahkan ke database,user mendapatkan pesan error karena tidak mengisi semua kolom input. Kemudian setelah berhasil input dan memasukkan PIN maka data berhasil ditambahkan ke database | Sistem menampilkan pesan error terlebih dahulu karena semua kolom input tidak diisi,baru menampilkan pesan sukses setelah berhasil semua validasi. |
| 3 | User memilih halaman buat utang dan memasukkan semua data input lalu user menglik button pinjam namun salah memasukkan PIN setelah itu user mengulangi mengklik tombol pinjam dan memasukkan PIN yang benar | Muncul pesan error Ketika salah memasukkan PIN, dan berhasil Ketika user sudah memasukkan PIN yang benar | Sistem menampilkan pesan error Ketika user salah memasukkan PIN dan berhasil memasukkan data ke database Ketika user sudah berhasill memasujjan PIN yang benar. |

1. **Hasil Uji White Box**

V(G) = (E-N)+2

V(G) = jumlah region

E = jumlah edge yang ditentukan dengan gambar panah

N = jumlah simpul grafik (node) dengan gambar lingkaran

V(G) = (8-7)+2

V(G) = 3

Jadi jalur path yang didapat ada 3 yaitu :

Jalur 1 = 1-2-3-4-5-6—7

Jalur 2 = 1-2-3-4-2-3-4-5-6-7

Jalur 3 = 1-2-3-4-5-6-2-3-4-5-6-7

|  |  |
| --- | --- |
| Path | 1 |
| Jalur | 1-2-3-4-5-6-7 |
| Skenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid)  7. Berhasil |
| Hasil Pengujian | Berhasil |
| Path | 2 |
| Jalur | 1-2-3-4-2-3-4-5-6-7 |
| Skenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi,jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (not valid)  2. Masukkan nama,deskripsi,jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid)  7. Berhasil |
| Hasil Pengujian | Berhasil |
| Path | 3 |
| Jalur | 1-2-3-4-5-6-2-3-4-5-6-7 |
| Scenario | 1. Halaman buat utang  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (not valid)  2. Masukkan nama,deskripsi, jumlah uang, jatuh tempo.  3. Klik button pinjam  4. Validasi input (valid)  5. Masukan PIN  6. Validasi PIN (valid) |

1. **Hasil Uji Black Box**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Kegiatan** | **Hasil yang diharapkan** | **Hasil** | **Keterangan** |
| 1 | User memilih halaman buat utang dan memasukan nama, deskripsi,jumlah uang,jatuh tempo, kemudian user memasukan PIN yang valid | Berhasil menambahkan utang dan data berhasil ditambahkan ke database | Sistem menampilkan pesan sukses menambahkan utang dan Kembali ke halaman utama | Valid |
| 2 | User memilih halaman buat utang dan tidak memasukan semua kolom input lalu user mengklik tombol pinjam. Lalu user mamasukan Kembali data input,setelah itu user memasukkan PIN yang valid. | Sebelum data berhasil ditambahkan ke database,user mendapatkan pesan error karena tidak mengisi semua kolom input. Kemudian setelah berhasil input dan memasukkan PIN maka data berhasil ditambahkan ke database | Sistem menampilkan pesan error terlebih dahulu karena semua kolom input tidak diisi,baru menampilkan pesan sukses setelah berhasil semua validasi. | Valid |
| 3 | User memilih halaman buat utang dan memasukkan semua data input lalu user menglik button pinjam namun salah memasukkan PIN setelah itu user mengulangi mengklik tombol pinjam dan memasukkan PIN yang benar | Muncul pesan error Ketika salah memasukkan PIN, dan berhasil Ketika user sudah memasukkan PIN yang benar | Sistem menampilkan pesan error Ketika user salah memasukkan PIN dan berhasil memasukkan data ke database Ketika user sudah berhasill memasujjan PIN yang benar. | Valid |

1. **Kesimpulan**

Dapat disimpulkan tujuan aplikasi Debt Manager dilakukan pengujian White-box adalah untuk menguji suatu aplikasi atau software dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak jadi dengan memeriksa komponen perangkat lunak apakah berjalan semestinya dengan melihat internal kode (source code) dari perangkat lunak tersebut.Berdasarkan hasil pengujian “Debt Manager” dapat disimpulkan bahwa aplikasi ini telah lolos tanpa adanya error untuk fungsionalitas utamanya, hal itu terlihat pada pengujian bagian buat utang , bayar utang , dan tambah pinjaman semuanya dinyatakan valid dengan kompleksitas yang rendah karena hanya memiliki 2-3 jalur yang ada pada setiap fungsi.

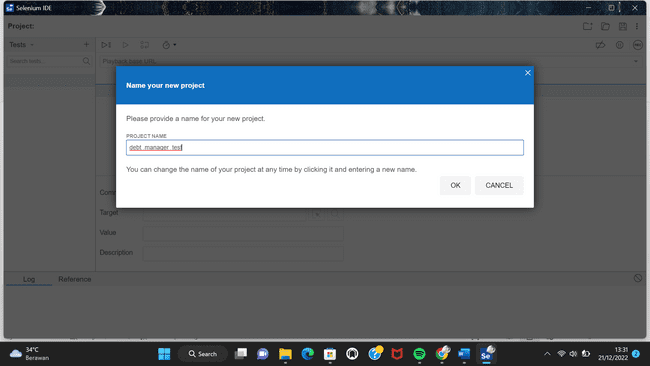
1. **Review**

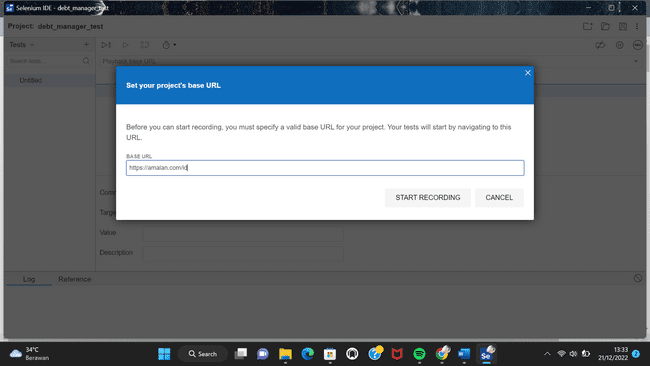
Berdasarkan hasil pengujian yang dilakukan dengan menggunakan white box dan black box. Sistem “Debt Manager” menerapkan metode white-box testing agar hasil pengujiain yang telah didapatkan akan dilakukan evaluasi Kembali,sehingga mampu menyimpulkan sebuah perangkat lunak dapat berjalan sesuai dengan kebutuhan yang diharapkan. Pada pengujian path fungsi buat utang setelah diketahui jalur independennya , dilakukan dengan perbandingan dengan menggunakan hubungan antara cyclomatic complexity dan resiko. Dan didapatkan fungsi buat utang ini memiliki resiko yang rendah dengan tingkat prosedur yang sederhana karena memiliki jalur independent berjumlah 3. Pada table test case bayar utang dibuat berdasarkan jalur independent yang telah didapatkan hasil valid untuk ketiga pengujian tidak meemukan error.

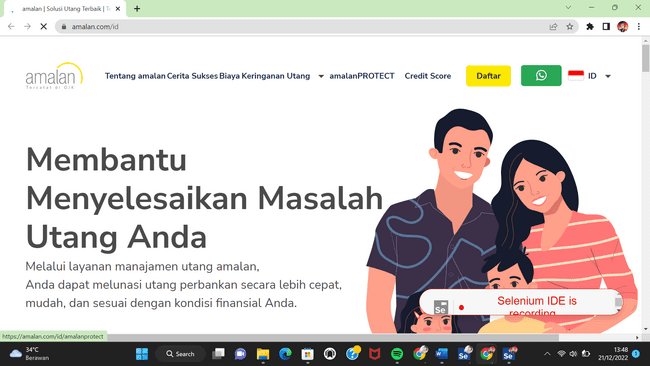
1. **Rekomendasi**

Pada aplikasi Debt Manager di rekomendasikan bagi pengguna yang membutuhkan pengelolaan, yang dapat mudah digunakan untuk mengelola hutang Anda dan hutang orang lain (kredit Anda).

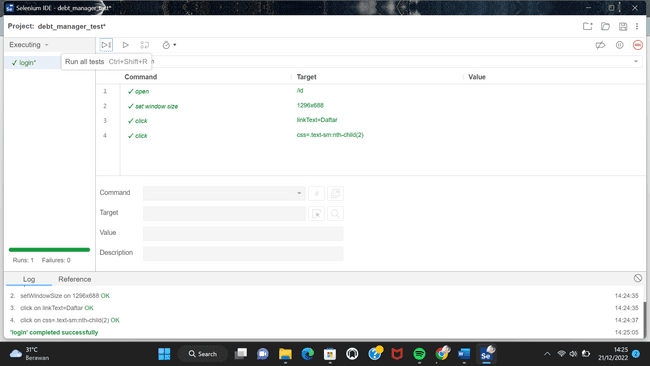
**Proses Pengujian Web Aplikasi Debt Manager Menggunakan Selenium IDE**

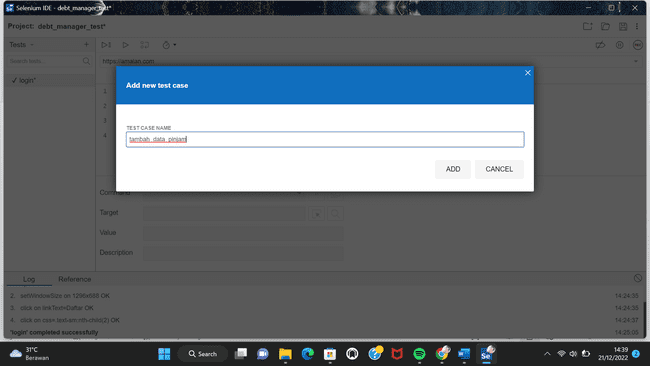
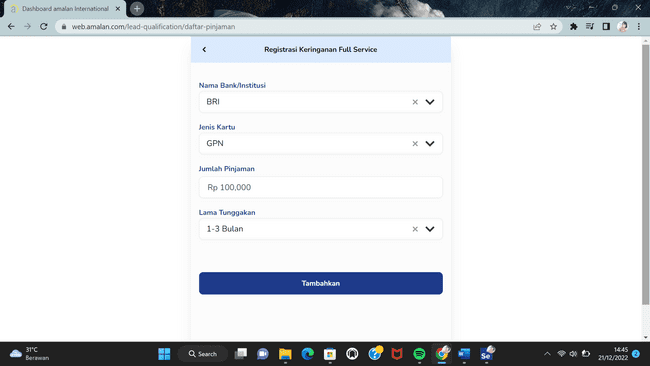
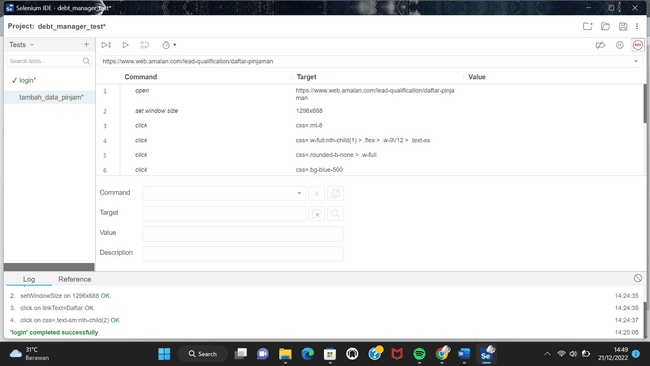
* Proses Pengujian
* Untuk proses pengujian lagkah pertama pilih “Record a new test in a new project” setelah itu isi “project name” lalu klik ok.  
  
* Kemudian isi link url aplikasi atau web yang akan diuji, kemudian start recording.

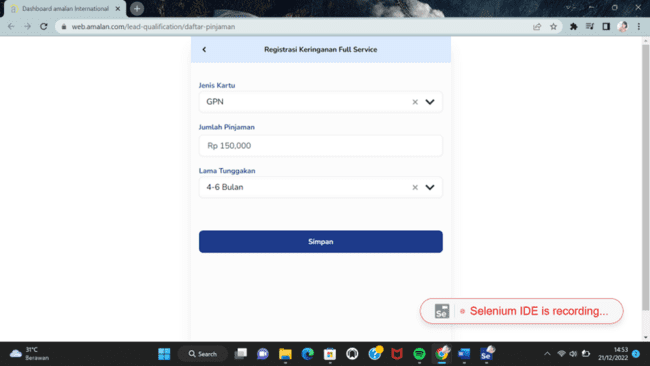


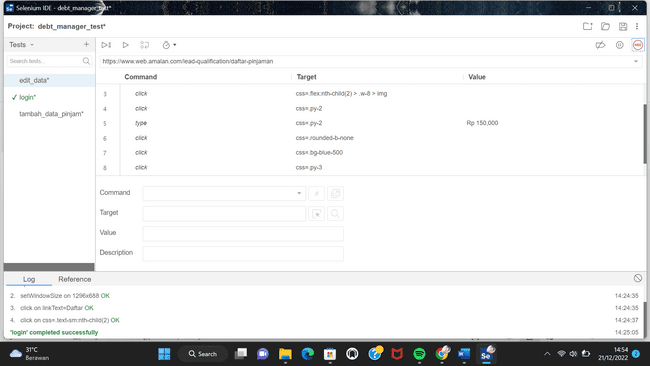
* Setelah itu, akan memunculkan halaman seperti berikut  
  
* Setelah klik daftar kemudian masukkan nomor hp , dan akan menghasilkan record seperti berikut

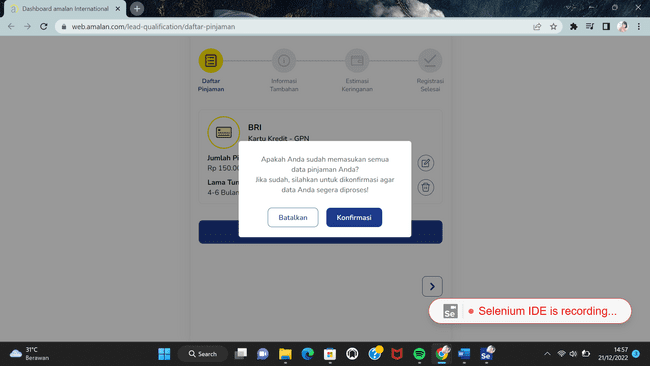
**Login**

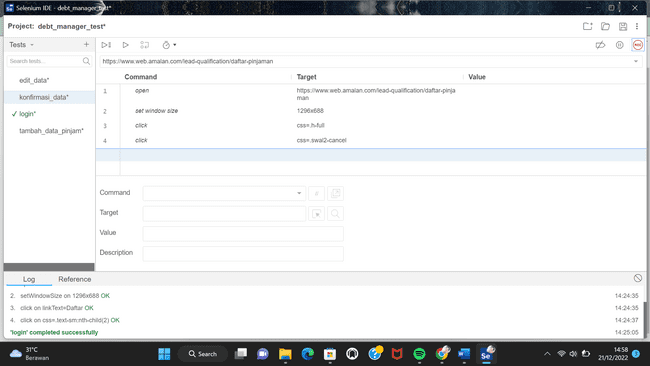


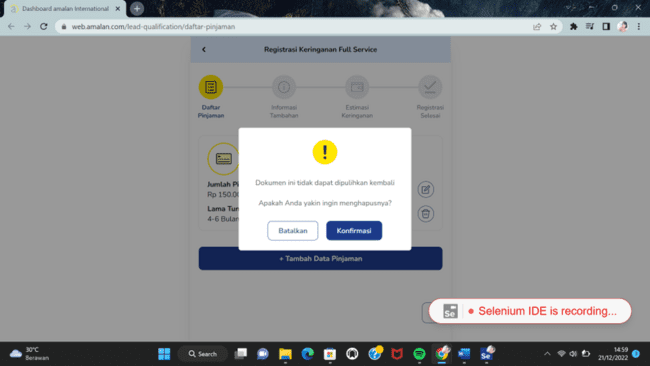
* Klik ikon “+” untuk membuat test baru dan berikan nama untuk pengujian yang akan dilakukan,setelah itu klik “ADD”
* Kemudian isi pada registrasi keringanan full service pada web seperti berikut  
  
* Setelah itu Kembali ke Selenium IDE , lalu running, dan akan menghasilkan seperti berikut:  
  
* Ulangi Langkah sebelumnya , dan akan menghasilkan seperti berikut :
* Edit Data

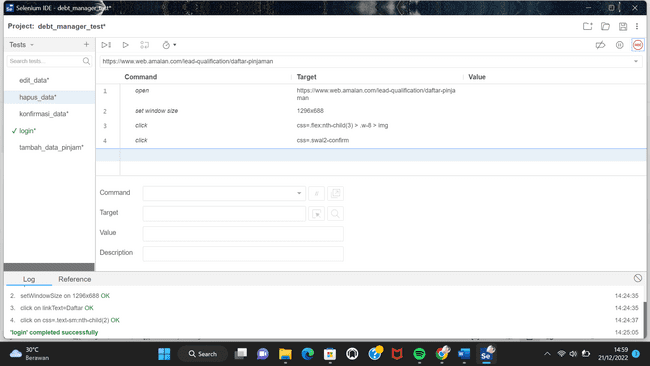


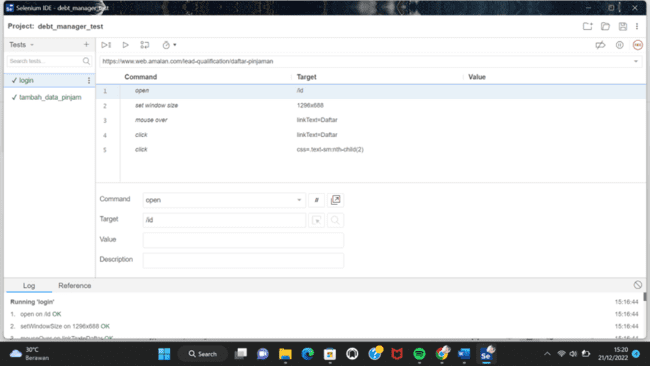
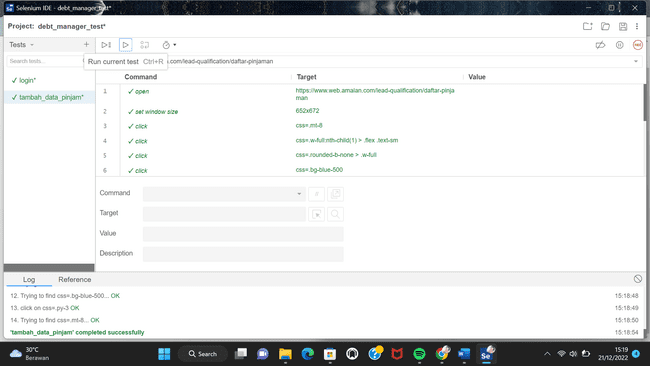
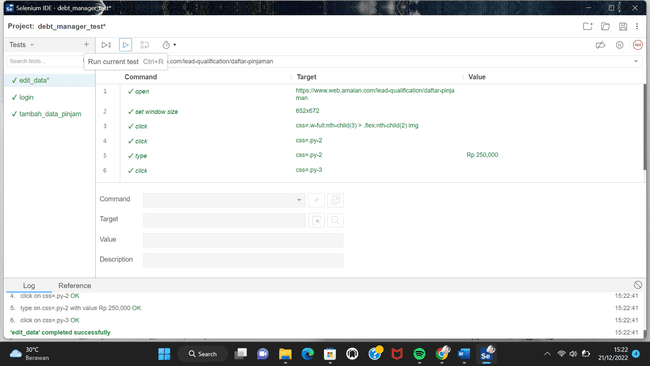
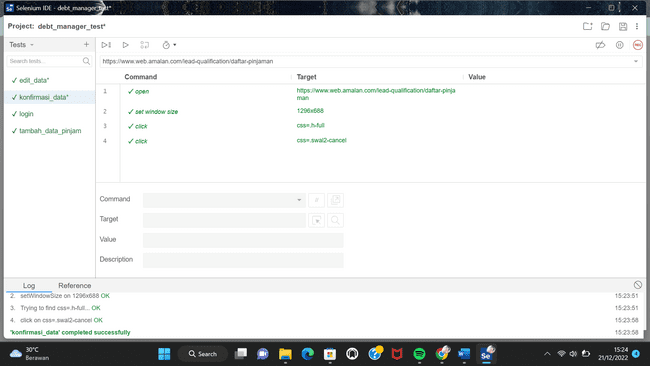


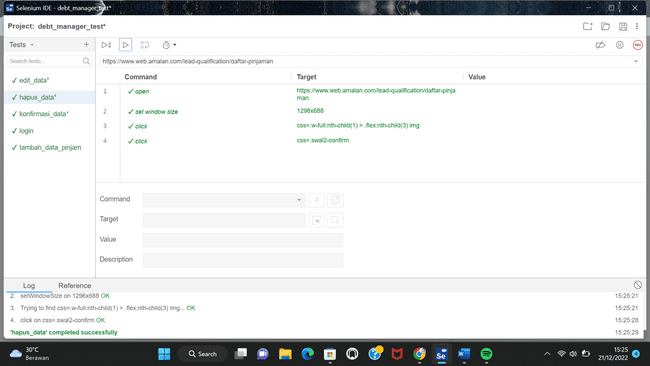
* Konfirmasi  
  



* Hapus Data  
  

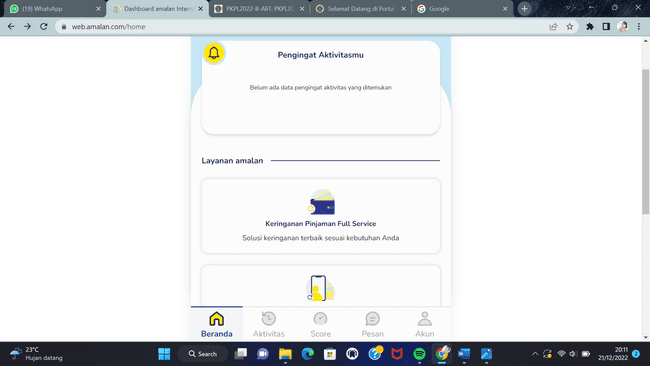


* **Login**  
  Pada hasil uji login , terdapat 5 langkah pengujian. Pengujian ini dilakukan untuk mengetahui apabila terdapat kesalahan saat input email dan password atau nomor handphone , apakah user dapat melakukan login pada aplikasi.  
  
* **Tambah Data Pinjaman**  
  Pada uji tambah data pinjam (Kartu Kredit), memiliki 6 langkah pengujian , user menginputkan nama bank, jenis kartu, jumlah pinjaman, dan lama tunggakan.  
  
* **Edit Data**  
  Pada pengujian Edit Data bertujuan untuk mengetahui fitur edit data dapat berfungsi dengan baik , pada pengujian ke 5 terdapat value Rp 250.000 berhasil diedit dari data yang sebelumnya ,yang artinya fitur edit data dapat berfungsi dengan seharusnya.  
  
* **Konfirmasi Data**Pada uji konfirmasi data terdapt 4 langkah , pada Langkah 3 memunculkan 2 pilihan yaitu “konfirmasi” dan “batal”  
  
* **Hapus Data**

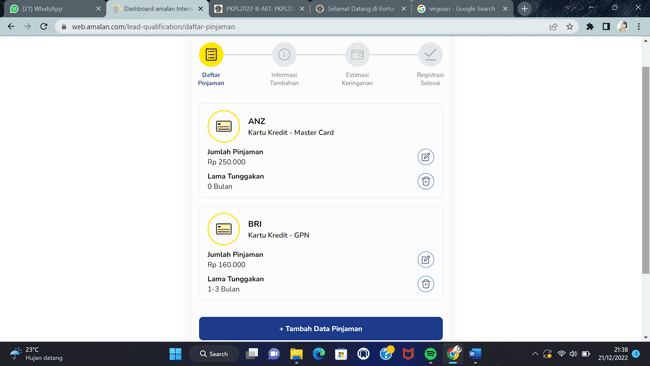
Pada uji hapus data terdapat 4 langkah, hal ini dilakukan pengujian untuk mengetahui apakah fitur hapus data ini sesuai dengan yang diharapkan , hal ini terbukti bahwa hasil uji sukses , fitur hapus data ini berjalan dengan baik, terbukti dengan berhasilnya data dihapus.  
  


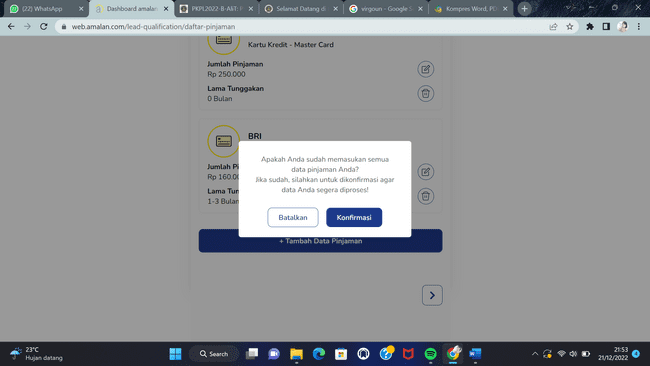
**Review**

Dari hasil yang saya coba pada aplikasi “Debt Manager” atau “Amalan” ini cukup simple dan menarik , seperti menu bar :

* Beranda  
  

Namun pada kasus uji coba ini , difokuskan untuk menguji “Daftar Pinjam” , dengan mengklik “+tambah data pinjaman” user dapat memilih akan meminjam kartu kredit atau yang lain. Kemudian setelah selesai mengisi data pinjaman user mengkonfirmasi untuk mengajukan pinjaman.

Pada aplikasi ini juga menyediakan fitur yang mudah untuk dipahami , namun menurut saya masih kurangnya informasi pada aplikasi ini.  


**Rekomendasi Perbaikan**

Pada bagian “Tambah Data Pinjaman” , terdapat “Daftar Pinjaman” , “Informasi Tambahan” , “Estimasi Keringanan” , “Registrasi Selesai”. Agar bisa melihat “Informasi Tambahan” , “Estimasi Keringanan” pada aplikasi ini harus menyetujui registrasi terlebih dahulu agar bisa melihat fitur selanjutnya. Jadi menurut saya pribadi alangkah baiknya jika registrasi dilakukan diakhir , sehingga user dapat mengetahui informasi secara keseluruhan terlebih dahulu.

**Kesimpulan pada pengujian menggunakan Selenium IDE :**

Pada pengujian menggunakan Selenium IDE ini tidak memerlukan source code untuk menguji suatu web atau aplikasi , dan pengugunaanya cukup mudah hanya perlu memasukan html atau web situs. Sudah dibuktikan dengan menggunakan Selenium IDE aplikasi Debt Manager tidak ada error yang berarti semua aplikasi sudah berjalan dengan fungsinya.

**Pengujian Aplikasi Debt Manager dengan menggunakan JMeter**  
1. Spesifikasi program yang ditest (fitur dan kebutuhan fungsional dan Non fungsional)

* Login

Kebutuhan Fungsional :

* System dapat melakukan input email
* System dapat melakukan verifikasi email yang masuk
* System dapat mengelola akun
* System dapat melakukan input password

Kebutuhan Non Fungsional :

* System menampilkan halaman login yang dapat dipahami
* System memastikan data yang digunakan harus terlindung dari akses yang tidak berwenang.
* Pinjaman Uang

Kebutuhan Fungsinonal :

* Sistem dapat memilih bank yang dituju
* System dapat menginputkan jumlah pinjaman uang yang akan dipinjam
* System dapat melakukan input no rekening user
* System dapat menyimpan hasil inputan oleh user

Kebutuhan Non Fungsional :

* System memiliki design tampilan yang mudah untuk dipahami oleh user.
* System dapat menampilkan hasil inputan user

2. Metode Pengujian

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

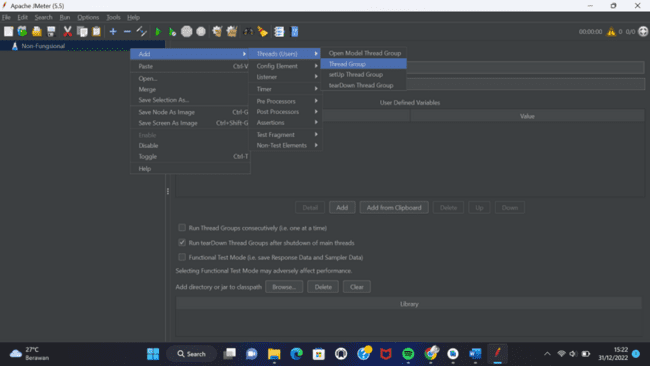
3. Rangkaian Pengujian

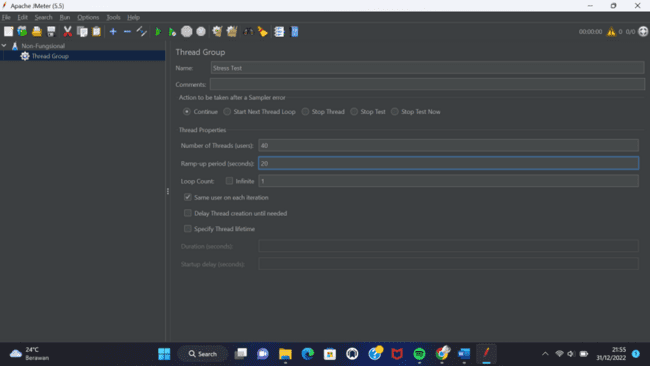
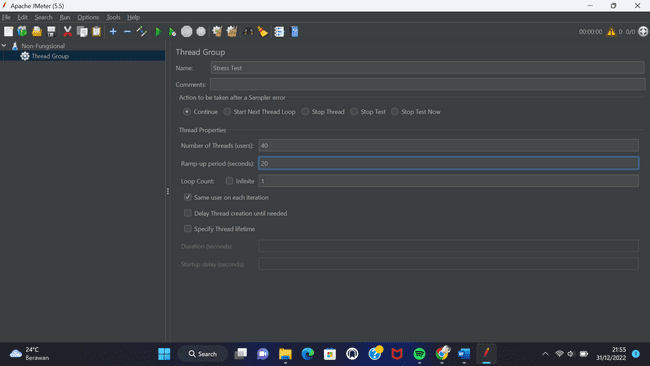
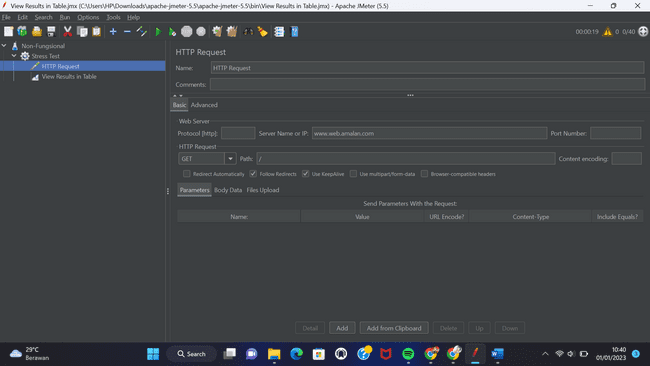
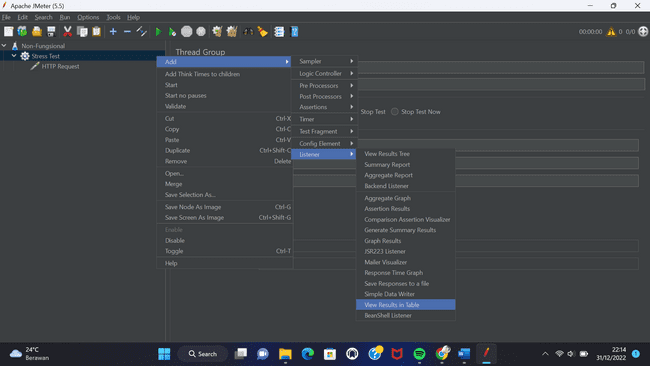
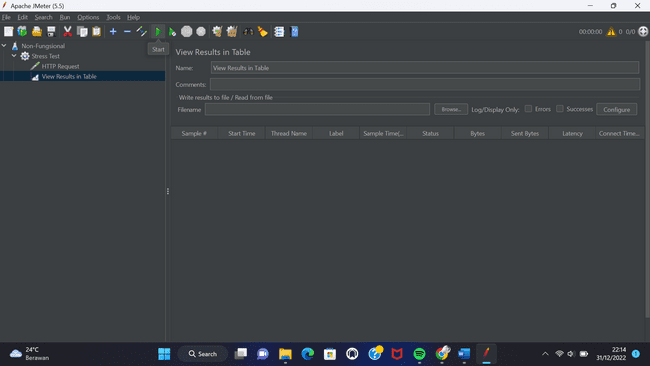
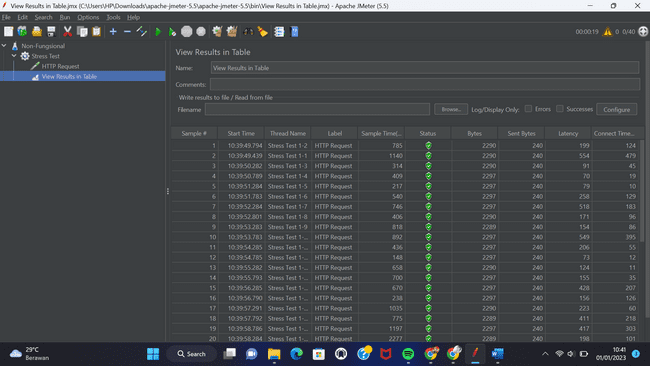
- Home

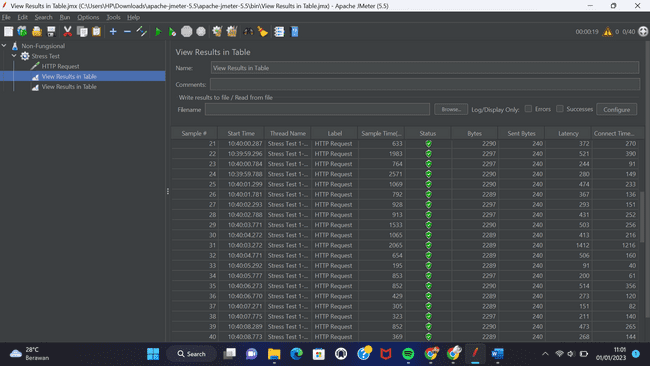
- Login

4. Tahap Pengujian

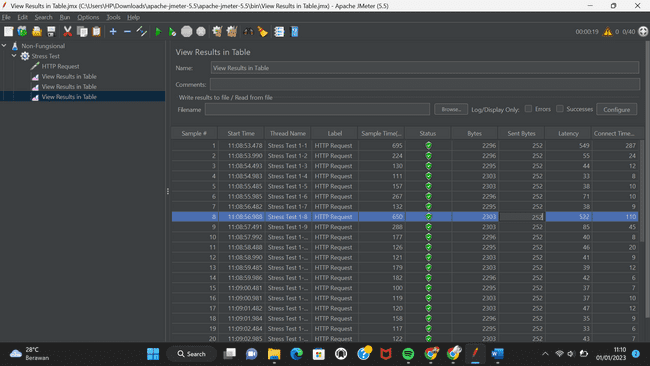
* Langkah pertama, buka aplikasi Apache JMeter, lalu klik Kanan dan pilih Add ==> Threads (User) => Thread Grup.

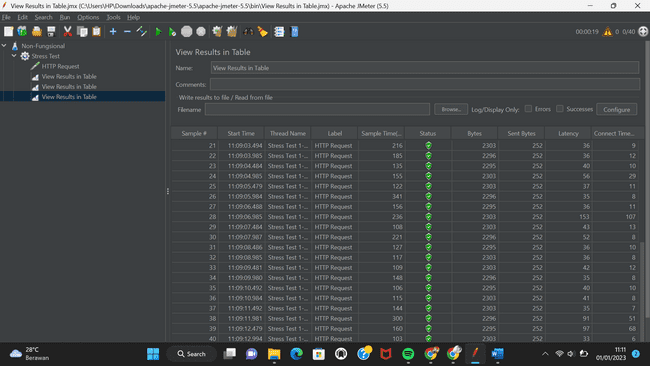


* Pada Menu Threads nama pengujian diisi dengan “Stress Test”. Lalu untuk Number of User saya isi sebanyak 40, lama periode sebanyak 20 dan loop sebanyak 1.  
  
* Pada Menu Threads nama pengujian diisi dengan “Stress Test”. Lalu untuk Number of User saya isi sebanyak 40, lama periode sebanyak 20 dan loop sebanyak 1.  
  
* Kemudian , aplikasi yang akan diuji berbasis web android klik kanan pada Stress Test , klik Add=> Sampler => HTTP Request. Pada HTTP Request , kemudian masukan server address dari aplikasi yang akan diujikan. Pada bagian Path karena pengujian pertama yang akan dilakukan adalah Home test maka isi dengan “/”.  
  
* Untuk melihat hasil pengujian yang akan dilakukan , klik kanan pata Stress Test kemudian pilih Add=> Listener => View Result in Table. Maka hasil pengujian akan ditampilkan dalam bentung table.  
  
* Untuk melakukan pengujian klik ikon start, namun sebelum melakukan pengujian alangkah baiknya menyimpan hasil pengujian terlebih dahulu.  
  
* Berikut adalah hasil pengujian pada Homepage.  
  



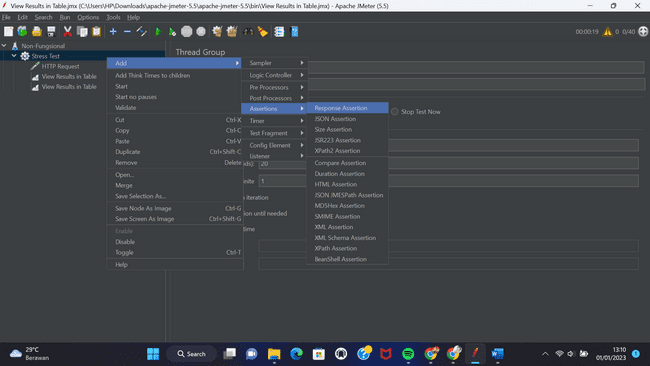
Sebanyak 40 user virtual yang diujikan dengan lama periode 20 detik dan 1 pengulangan hasilnya adalah semua user berhasil dan tidak ada yang gagal.

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

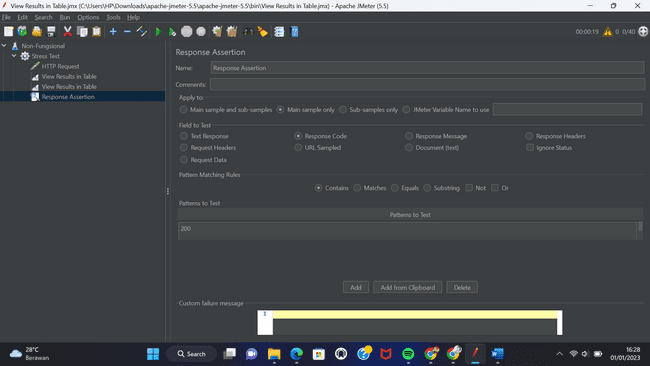


* Particular Element Test

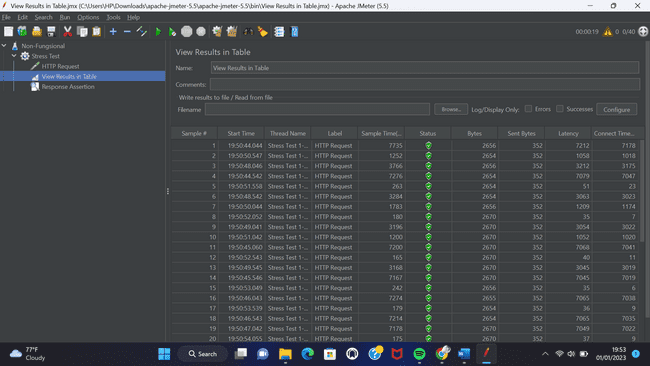
Untuk melakukan pengujian Particular Element, klik kanan pada Stress Test, kemudian klik Add=>Assertions=>Response Assertion.



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”.



**Kesimpulan**

Dapat disimpulkan tujuan aplikasi Debt Manager dilakukan pengujian White-box adalah untuk menguji suatu aplikasi atau software dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang dibuat ada yang salah atau tidak jadi dengan memeriksa komponen perangkat lunak apakah berjalan semestinya dengan melihat internal kode (source code) dari perangkat lunak tersebut.Berdasarkan hasil pengujian “Debt Manager” dapat disimpulkan bahwa aplikasi ini telah lolos tanpa adanya error untuk fungsionalitas utamanya, hal itu terlihat pada pengujian bagian buat utang , bayar utang , dan tambah pinjaman semuanya dinyatakan valid dengan kompleksitas yang rendah karena hanya memiliki 2-3 jalur yang ada pada setiap fungsi.

**Review**

Berdasarkan hasil pengujian yang dilakukan dengan menggunakan white box dan black box. Sistem “Debt Manager” menerapkan metode white-box testing agar hasil pengujiain yang telah didapatkan akan dilakukan evaluasi Kembali,sehingga mampu menyimpulkan sebuah perangkat lunak dapat berjalan sesuai dengan kebutuhan yang diharapkan. Pada pengujian path fungsi buat utang setelah diketahui jalur independennya , dilakukan dengan perbandingan dengan menggunakan hubungan antara cyclomatic complexity dan resiko. Dan didapatkan fungsi buat utang ini memiliki resiko yang rendah dengan tingkat prosedur yang sederhana karena memiliki jalur independent berjumlah 3. Pada table test case bayar utang dibuat berdasarkan jalur independent yang telah didapatkan hasil valid untuk ketiga pengujian tidak meemukan error.

**Rekomendasi**

Pada aplikasi Debt Manager di rekomendasikan bagi pengguna yang membutuhkan pengelolaan, yang dapat mudah digunakan untuk mengelola hutang Anda dan hutang orang lain (kredit Anda).