Ui Layer

1.RegisterViewModelTest

* Ketika berhasil register user
  + Memastikan data tidak null
  + Memastikan data adalah instance dari NetworkResult.Success
  + Memastikan variabel “message” dari data bernilai “No Error”
  + Memastikan variable “error” dari data bernilai false
* Ketika gagal register user
  + Memastikan data tidak null
  + Memastikan data adalah instance dasri NetworkResult.Error
  + Memastikan variabel “message” dari data bernilai network is error

2.LoginViewModelTest

* Ketika berhasil login user
  + Memastikan data tidak null
  + Memastikan data adalah instance dari NetworkResult.Success
  + Memastikan variabel “message” dari data bernilai “No Error”
  + Memastikan variable “error” dari data bernilai false
* Ketika gagal login user
  + Memastikan data tidak null
  + Memastikan data adalah instance dasri NetworkResult.Error
  + Memastikan variabel “message” dari data bernilai network is error
* Ketika menyimpan user token
  + Menyimpan data ke dalam data store dengan nilai 123
  + Memastikan data tersimpan dengan nilai 123
* Ketika menyimpan login status
  + Menyimpan login status dengan nilai true
  + Memastikan data login status tersimpan dengan nilai true

3.HomeViewModelTest

* Ketika berhasil memuat data story
  + Memastikan data tidak null
  + Memastikan data yang dimasukkan ke AsyncPagingDataDiffer sesuai

4.SharedStoryViewModelTest

* Ketika berhasil mengunggah gambar
  + Memastikan data tidak null
  + Memastikan data adalah instance dari NetworkResult.Success
  + Memastikan variabel message dari data bernilai No Error
  + Memastikan variabel error dari data bernilai false
* Ketika gagal menggungah gambar
  + Memastikan data tidak null
  + Memastikan data adalah instance dasri NetworkResult.Error
  + Memastikan variabel “message” dari data bernilai network is error
* Ketika berhasil menyimpan hasil gambar dari kamera
  + Memastikan data tidak null
  + Memastikan data adalah instance dari ImageModel
  + Memastikan data sesuai

Data Layer

1.StoriesRepositoryTest

* Ketika berhasil mendapatkan data story
  + Memastikan flow yang pertama diexpose oleh repository adalah instance dari NetworkResult.Loading
  + Memastikan flow yang kedua diexpose oleh repository adalah instance dari NetworkResult.Success
  + Memastikan fungsi getStories() dari mock terpanggil
* Ketika gagal mendapatkan data story
  + Memastikan flow yang pertama diexpose oleh repository adalah instance dari NetworkResult.Loading
  + Memastikan flow yang kedua diexpose oleh repository adalah instance dari NetworkResult.Error
  + Memastikan fungsi getStories() dari mock terpanggil
* Ketika berhasil mengunggah gambar
  + Memastikan flow yang pertama diexpose oleh repository adalah instance dari NetworkResult.Loading
  + Memastikan flow yang kedua diexpose oleh repository adalah instance dari NetworkResult.Success
  + Memastikan fungsi uploadStories() dari mock terpanggil
* Ketika gagal menggungah gambar
  + Memastikan flow yang pertama diexpose oleh repository adalah instance dari NetworkResult.Loading
  + Memastikan flow yang kedua diexpose oleh repository adalah instance dari NetworkResult.Error
  + Memastikan fungsi uploadStories() dari mock terpanggil
* Ketika mengambil data story paging
  + Memastikan flow story paging yang diexpose repository tidak null

2.UserRepositoryTest

* Ketika berhasil melakukan login user
  + Memastikan flow yang pertama diexpose oleh repository adalah instance dari NetworkResult.Loading
  + Memastikan flow yang kedua diexpose oleh repository adalah instance dari NetworkResult.Success
  + Memastikan fungsi loginUser() dari mock terpanggil
* Ketika gagal melakukan login user
  + Memastikan flow yang pertama diexpose oleh repository adalah instance dari NetworkResult.Loading
  + Memastikan flow yang kedua diexpose oleh repository adalah instance dari NetworkResult.Error
  + Memastikan fungsi loginUser() dari mock terpanggil
* Ketika berhasil melakukan register user
  + Memastikan flow yang pertama diexpose oleh repository adalah instance dari NetworkResult.Loading
  + Memastikan flow yang kedua diexpose oleh repository adalah instance dari NetworkResult.Success
  + Memastikan fungsi registerUser() dari mock terpanggil
* Ketika gagal melakukan register user
  + Memastikan flow yang pertama diexpose oleh repository adalah instance dari NetworkResult.Loading
  + Memastikan flow yang kedua diexpose oleh repository adalah instance dari NetworkResult.Error
  + Memastikan fungsi registerUser() dari mock terpanggil