Unit Test

AuthViewModelTest

1. Ketika login gagal
    a. Memastikan fungsi login() pada AuthUseCase mengembalikan nilai Resource.Error()
    b. Memastikan fungsi login() pada AuthViewModel melakukan observe terhadap data dari Resource<LoginResponse>
    c. Memverifikasi fungsi login() pada AuthUseCase dipanggil

2. Ketika login berhasil
    a. Memastikan fungsi login() pada AuthUseCase mengembalikan nilai Resource.Success dan membawa nilai response yang sesuai
    b. Memastikan fungsi login() pada AuthViewModel melakukan observe terhadap data dari Resource<LoginResponse>
    c. Memverifikasi fungsi login() pada AuthUseCase dipanggil

3. Ketika register gagal
    a. Memastikan fungsi createUser() pada AuthUseCase mengembalikan nilai Resource.Error()
    b. Memastikan fungsi createUser() pada AuthViewModel melakukan observe terhadap data dari Resource<RegisterResponse>
    c. Memverifikasi fungsi createUser() pada AuthUseCase dipanggil

4. Ketika register berhasil
    a. Memastikan fungsi createUser() pada AuthUseCase mengembalikan nilai Resource.Success() dan membawa nilai response yang sesuai

b. Memastikan fungsi createUser() pada AuthViewModel melakukan observe terhadap data dari Resource<RegisterResponse>
c. Memverifikasi fungsi createUser() pada AuthUseCase dipanggil

HomeViewModelTest

1. Ketika berhasil mendapatkan data *stories*
    a. Memastikan fungsi getAllStories() pada StoryUseCase mengembalikan nilai PagingData<Story>
    b. Memastikan fungsi getAllStories() pada HomeViewModel melakukan observe terhadap data PagingData<Story>
    c. Memverifikasi fungsi getAllStory pada StoryUseCase dipanggil

MapsViewModelTest

1. Ketika berhasil mendapatkan data *stories* yang memiliki lokasi
    a. Memastikan fungsi getAllStoriesWithLocation() pada MapsUseCase mengembalikan nilai Resource<StoriesResponse>
    b. Memastikan fungsi getAllStoriesWithLocation() pada MapsViewModel melakukan observe terhadap data Resource<StoriesResponse>
    c. Memverifikasi fungsi getAllStoriesWithLocation() pada MapsUseCase dipanggil

PostStoryViewModelTest

1. Memastikan fungsi uploadStory() pada StoryUseCase mengembalikan nilai Resource<PostStoryResponse>
2. Memastikan variabel isSuccess pada PostStoryViewModel melakukan observe terhadap data PostStoryResponse jika fungsi uploadStory() pada PostStoryViewModel berhasil
3. Memverifikasi fungsi uploadStory() pada StoryUseCase dipanggil

Note: Pada *project* ini *repository* tidak dilakukan *testing* dikarenakan pada *repository* memakai *abstract class* tidak langsung berinteraksi terhadap ApiService. Namun pada *project* ini dilakukan *testing* pada *class Remote* yang bertugas berinteraksi langsung dengan *interface* ApiService.

AuthRemoteTest

1. Ketika login berhasil
    a. Memastikan fungsi login() pada ApiService mengembalikan response yang diharapkan
    b. Memastikan nilai ApiResponse.Success pada fungsi login() membawa data response yang tidak null, tidak error dan sesuai yang diharapkan.

2. Ketika register berhasil
    a. Memastikan fungsi create() pada ApiService mengembalikan response yang diharapkan

b. Memastikan nilai ApiResponse.Success pada fungsi createUser() membawa data response yang tidak null, tidak error dan sesuai yang diharapkan.

StoryRemoteTest

1. Ketika mendapatkan *stories* berhasil
    a. Memastikan fungsi getAllStories() pada ApiService mengembalikan data response yang sesuai.
    b. Memastikan data response tidak null
    c. Memastikan data response sesuai yang diharapkan

2. Ketika mendapatkan *stories with location* berhasil
    a. Memastikan fungsi getAllStories() pada ApiService mengembalikan data response yang sesuai
    b. Memastikan nilai ApiResponse.Success pada fungsi getAllStoriesWithLocation() membawa data response yang tidak null, tidak kosong dan sesuai yang diharapkan.

3. Ketika berhasil *upload story*
    a. Memastikan fungsi uploadStory() pada ApiService mengembalikan nilai response yang sesuai
    b. Memastikan nilai ApiResponse.Success pada fungsi uploadStory() membawa data response yang tidak error

Integration Testing

HomeTestActivity (@MediumTest, @LargestTest)

1. Menjalankan Home Activity
    a. Memastikan btnPost(ImageButton) tampil dilayar
    b. Memastikan mapsStories(ImageButton) tampil dilayar
    c. Memastikan btnSettings(ImageButton) tampil dilayar
    d. Memastikan rvStories(RecyclerView) tampil dilayar
    e. Memastikan progressBar hilang setelah rvStories berhasil memuat data
    f. Memastikan rvStories dapat melakukan scrolling

2. Menjalankan Detail Activity
    a. Memastikan saat salah satu item pada rvStories di klik, maka akan berpindah ke halaman detail
    b. Memastikan tvStoryUsername tampil pada halaman detail
    c. Memastikan tvStoryDescription tampil pada halaman detail
    d. Memastikan tvStoryDate tampil pada halaman detail
    e. Memastikan ivStoryImage tampil pada halaman detail
    f. Memastikan btnBack dapat berfungsi saat di klik

3. Menjalankan Maps Activity
    a. Memastikan button mapsStories berfungsi saat di klik dan dapat berpindah ke halaman maps
    b. Memastikan layout map tampil pada halaman map

4. Menjalankan PostStory Activity
    a. Memastikan btnPost berfungsi saat di klik dan dapat berpindah ke halaman post story

b. Memastikan photoLayout(ConstraintLayout) tampil dilayar
c. Memastikan ivAddStory(ImageView) tampil dilayar
d. Memastikan pickFormGalery(TextView) tampil dilayar
e. Memastikan takePhoto(TextView) tampil dilayar
f. Memastikan tilDescription(TextInputLayout) tampil dilayar
g. Memastikan btnUpload(Button) tampil dilayar
h. Memastikan btnArrowBack(ImageButton) tampil dilayar dan berfungsi saat di klik