Unit test

1. LocalMoviesRepositoryTest
   - Memastikan jumlah data **movies** yang didapatkan tidak kosong
     - Mendapatkan data dari *local source*
     - Memastikan bahwa jumlah data **movies** sama dengan **19**
   - Memastikan data **detail movie** yang didapatkan dengan parameter ***id*** tidak ***null***
     - Memastikan data detail dari *local source* tidak bernilai ***null***
2. LocalShowsRepositoryTest
   - Memastikan jumlah data **shows** yang didapatkan tidak kosong
     - Mendapatkan data dari *local source*
     - Memastikan bahwa jumlah data **shows** sama dengan **20**
   - Memastikan data **detail show** yang didapatkan dengan parameter ***id*** tidak ***null***
     - Memastikan data detail dari *local source* tidak bernilai ***null***
3. MoviesViewModelTest
   - Memastikan jumlah data **movies** yang didapatkan dari repository tidak kosong
     - Membuat skenario pemanggilan repository yang menghasilkan kumpulan data kosong
     - Mendapatkan kumpulan data dari repository
     - Memastikan bahwa fungsi repository telah terpanggil dari dalam *ViewModel*
     - Memastikan bahwa jumlah data yang didapatkan dari pemanggilan repository tidak sama dengan data yang dipanggil langsung
   - Memastikan data **movies** yang berbentuk **entity** berhasil di-*map* dalam bentuk **model**
     - Membuat skenario pemanggilan repository yang menghasilkan kumpulan data berbentuk **entity**
     - Mendapatkan kumpulan data dari repository
     - Memastikan bahwa fungsi repository telah terpanggil dari dalam *ViewModel*
     - Memastikan bahwa kumpulan data yang didapatkan dari pemanggilan repository berbentuk **model**
   - Memastikan data **detail movie** yang berbentuk **entity** berhasil di-*map* dalam bentuk **model**
     - Membuat skenario pemanggilan repository yang menghasilkan detail data berbentuk **entity**

- Mendapatkan detail data dari repository
- Memastikan bahwa fungsi repository telah terpanggil dari dalam *ViewModel*
- Memastikan bahwa detiail data yang didapatkan dari pemanggilan repository berbentuk **model**

- Memastikan jika data **detail movie** yang didapatkan dari repository adalah ***null*****,** maka akan megembalikan data kosong
  - Membuat skenario pemanggilan repository yang menghasilkan detail data ***null***
  - Mendapatkan detail data dari repository
  - Memastikan bahwa fungsi repository telah terpanggil dari dalam *ViewModel*
  - Memastikan bahwa detail data yang didapatkan dari pemanggilan repository berisi data kosong
  - Memastikan bahwa detiail data yang didapatkan dari pemanggilan repository tidak berisi data yang vallid

4. ShowsViewModelTest
   - Memastikan jumlah data **shows** yang didapatkan dari repository tidak kosong
     - Membuat skenario pemanggilan repository yang menghasilkan kumpulan data kosong
     - Mendapatkan kumpulan data dari repository
     - Memastikan bahwa fungsi repository telah terpanggil dari dalam *ViewModel*
     - Memastikan bahwa jumlah data yang didapatkan dari pemanggilan repository tidak sama dengan data yang dipanggil langsung
   - Memastikan data **shows** yang berbentuk **entity** berhasil di-*map* dalam bentuk **model**
     - Membuat skenario pemanggilan repository yang menghasilkan kumpulan data berbentuk **entity**
     - Mendapatkan kumpulan data dari repository
     - Memastikan bahwa fungsi repository telah terpanggil dari dalam *ViewModel*
     - Memastikan bahwa kumpulan data yang didapatkan dari pemanggilan repository berbentuk **model**
   - Memastikan data **detail show** yang berbentuk **entity** berhasil di-*map* dalam bentuk **model**
     - Membuat skenario pemanggilan repository yang menghasilkan detail data berbentuk **entity**

- Mendapatkan detail data dari repository
- Memastikan bahwa fungsi repository telah terpanggil dari dalam *ViewModel*
- Memastikan bahwa detiail data yang didapatkan dari pemanggilan repository berbentuk **model**

- Memastikan data **detail show** yang didapatkan dari repsitory tidak kosong
  - Membuat skenario pemanggilan repository yang menghasilkan detail data ***null***
  - Mendapatkan detail data dari repository
  - Memastikan bahwa fungsi repository telah terpanggil dari dalam *ViewModel*
  - Memastikan bahwa detail data yang didapatkan dari pemanggilan repository berisi data kosong
  - Memastikan bahwa detiail data yang didapatkan dari pemanggilan repository tidak berisi data yang vallid

Instrumentation Test

1. SplashFragmentTest
   - Mengetes fungsionalitas navigasi dari ***SplashFragment*** yang berpindah ke ***MainFragment*** setelah di-*delay* selama 1750ms
     - Mendapatkan ***NavController***
     - Meluncurkan ***FragmentScenario*** untuk ***SplashFragment***
     - Memasang ***NavGraph*** pada ***NavController***
     - Memasang ***NavController*** pada ***View***
     - Men-*delay* selama 1750ms
     - Memastikan bahwa ***id*** dari destinasi sekarang, sama seperti ***id*** destinasi tujuan
2. MainFragmentTest
   - Mengetes fungsionalitas ***TabLayout*** menu & ***ViewPager2***
     - Mendapatkan ***NavController***
     - Meluncurkan ***FragmentScenario*** untuk ***MainFragment***
     - Memasang ***NavGraph*** pada ***NavController***
     - Memasang ***NavController*** pada ***View***
     - Memastikan komponen ***view_pager*** sudah tampil
     - Mengklik komponen dengan teks ***Shows*** untuk berpindah halaman
     - Memastikan komponen ***shows_fragment*** sudah tampil
     - Melakukan aksi *swipeRight()* unttuk berpindah halaman

- Memastikan komponen ***movies_fragment*** sudah tampil
- Melakukan aksi *swipeLeft()* unttuk berpindah halaman
- Memastikan komponen ***shows_fragment*** sudah tampil
- Mengklik komponen dengan teks ***Movies*** untuk berpindah halaman
- Memastikan komponen ***movies_fragment*** sudah tampil

3. MoviesFragmentTest
   - Mengetes fungsionalitas dari ***ViewPager2*** yang digunakan untuk tampilan ***Carousel***
     - Meluncurkan ***FragmentScenario*** untuk ***MoviesFragment***
     - Memastikan komponen ***movies_fragment*** sudah tampil
     - Melakukan aksi *swipeLeft()* pada komponen ***movies_carousel*** sebanyak 4 kali
     - Melakukan aksi *swipeRight()* pada komponen ***movies_carousel*** sebanyak 4 kali
   - Mengetes fungsionalitas dari ***RecyclerView***
     - Meluncurkan ***FragmentScenario*** untuk ***MoviesFragment***
     - Memastikan komponen ***movies_fragment*** sudah tampil
     - Melakukan aksi *scroll* pada komponen ***rv_popular_movies*** sampai posisi terakhir
     - Melakukan aksi *scroll* pada komponen ***rv_popular_movies*** sampai posisi paling awal
   - Memastikan data yang dipilih dari item ***Carousel*** menampilkan data yang sesuai di ***DetailItemFragment***
     - Mendapatkan ***NavController***
     - Meluncurkan ***FragmentScenario*** untuk ***MoviesFragment***
     - Memasang ***NavGraph*** pada ***NavController***
     - Memasang ***NavController*** pada ***View***
     - Memastikan komponen ***movies_carousel*** sudah tampil
     - Melakukan aksi *swipeLeft()* pada komponen ***movies_carousel*** sebanyak 3 kali, kemudian diakhiri dengan aksi *click*
     - Meluncurkan ***FragmentScenario*** untuk ***DetailItemFragment*** dengan data *bundle*
     - Memastikan komponen ***tv_title*** sudah tampil
     - Memastikan data yang ditampilkan oleh ***tv_title*** sesuai dengan data yang dikirimkan
   - Memastikan data yang dipilih dari item list menampilkan data yang sesuai di ***DetailitemFragment***

- Mendapatkan ***NavController***
- Meluncurkan ***FragmentScenario*** untuk ***MoviesFragment***
- Memasang ***NavGraph*** pada ***NavController***
- Memasang ***NavController*** pada ***View***
- Memastikan komponen ***rv_popular_movies*** sudah tampil
- Melakukan aksi *scroll* pada komponen ***rv_popular_movies*** sampaaai posisi tertentu, kemudian diakhiri dengan aksi *click*
- Meluncurkan ***FragmentScenario*** untuk ***DetailItemFragment*** dengan data *bundle*
- Memastikan komponen ***tv_title*** sudah tampil
- Memastikan data yang ditampilkan oleh ***tv_title*** sesuai dengan data yang dikirimkan

4. MoviesFragmentTest
   - Mengetes fungsionalitas dari ***ViewPager2*** yang digunakan untuk tampilan ***Carousel***
     - Meluncurkan ***FragmentScenario*** untuk ***ShowsFragment***
     - Memastikan komponen ***shows_fragment*** sudah tampil
     - Melakukan aksi *swipeLeft()* pada komponen ***shows_fragment*** sebanyak 4 kali
     - Melakukan aksi *swipeRight()* pada komponen ***shows_fragment*** sebanyak 4 kali
   - Mengetes fungsionalitas dari ***RecyclerView***
     - Meluncurkan ***FragmentScenario*** untuk ***ShowsFragment***
     - Memastikan komponen ***shows_fragment*** sudah tampil
     - Melakukan aksi *scroll* pada komponen ***rv_popular_shows*** sampai posisi terakhir
     - Melakukan aksi *scroll* pada komponen ***rv_popular_shows*** sampai posisi paling awal
   - Memastikan data yang dipilih dari item ***Carousel*** menampilkan data yang sesuai di ***DetailItemFragment***
     - Mendapatkan ***NavController***
     - Meluncurkan ***FragmentScenario*** untuk ***ShowsFragment***
     - Memasang ***NavGraph*** pada ***NavController***
     - Memasang ***NavController*** pada ***View***
     - Memastikan komponen ***shows_carousel*** sudah tampil
     - Melakukan aksi *swipeLeft()* pada komponen ***shows_carousel*** sebanyak 3 kali, kemudian diakhiri dengan aksi *click*
     - Meluncurkan ***FragmentScenario*** untuk ***DetailItemFragment*** dengan data *bundle*

- Memastikan komponen ***tv_title*** sudah tampil
- Memastikan data yang ditampilkan oleh ***tv_title*** sesuai dengan data yang dikirimkan

- Memastikan data yang dipilih dari item list menampilkan data yang sesuai di ***DetailitemFragment***
  - Mendapatkan ***NavController***
  - Meluncurkan ***FragmentScenario*** untuk ***ShowsFragment***
  - Memasang ***NavGraph*** pada ***NavController***
  - Memasang ***NavController*** pada ***View***
  - Memastikan komponen ***rv_popular_shows*** sudah tampil
  - Melakukan aksi *scroll* pada komponen ***rv_popular_shows*** sampaaai posisi tertentu, kemudian diakhiri dengan aksi *click*
  - Meluncurkan ***FragmentScenario*** untuk ***DetailItemFragment*** dengan data *bundle*
  - Memastikan komponen ***tv_title*** sudah tampil
  - Memastikan data yang ditampilkan oleh ***tv_title*** sesuai dengan data yang dikirimkan

5. DetailItemFragmentTest
   - Memastikan back button bekerja sebagaimana mestinya
     - Mendapatkan ***NavController***
     - Meluncurkan ***FragmentScenario*** untuk ***ShowsFragment***
     - Memasang ***NavGraph*** pada ***NavController***
     - Memasang ***NavController*** pada ***View***
     - Memastikan komponen ***back_button*** sudah tampil
     - Melakukan aksi klik pada ***back_button***
   - Memastikan Data yang ditampilkan sudah sesuai
     - Meluncurkan ***FragmentScenario*** untuk ***DetailItemFragment*** dengan data *bundle*
     - Memastikan data yang ditampilkan oleh tiap komponen View sudah benar atau berhasil ditampilkan