Di Submission ini, saya mengikuti saran serta memberi tambahan, yakni

* Menerapkan tampilan aplikasi yang sesuai standar:
  + Tampilan aplikasi memiliki *width*, *height*, *margin*, dan *padding*yang sesuai.
  + Pemilihan warna yang sesuai tema aplikasi.
  + Tidak ada komponen yang saling bertumpuk.
  + Penggunaan komponen yang sesuai dengan fungsinya.  
    Contoh : Komponen ImageView yang dijadikan sebagai button navigasi.
* Menuliskan kode dengan *bersih*.
  + Bersihkan comment dan kode yang tidak digunakan.
  + Indentasi yang sesuai.
  + Menghapus import yang tidak digunakan.
* Terdapat improvisasi fitur seperti (*pilih satu*):
  + Splash screen yang sesuai dengan tema aplikasi.
  + Share.
  + dsb.
* Menambahkan test case untuk skenario pengujian berikut:
  + Data gagal dimuat.
* Menerapkan ViewModel dengan baik dan benar.
* Tambahan: Menggunakan API dari TheMovies Database sebagai *remote data source*

**Skenario Uji Unit (Unit Test) dan Uji Instrumen (Instrumented Test):**

**Unit Testing**

1. TrendingMoviesViewModelTest
   1. Memuat **Trending Movies Daily**
      1. Memastikan **API dengan endpoint Trending Movies Daily** berhasil di akses
      2. Memastikan **Response Body** yang dimuat dari **API endpoint Trending Movies Daily** tidak *null*
      3. Memastikan isi dari **key ‘result’** pada **API endpoint Trending Movies Daily** tidak nol
   2. Memuat **Trending Movies Weekly**
      1. Memastikan **API dengan endpoint Trending Movies Weekly** berhasil di akses
      2. Memastikan **Response Body** yang dimuat dari **API endpoint Trending Movies Weekly** tidak *null*
      3. Memastikan isi dari key **‘result’** pada **API endpoint Trending Movies Weekly** tidak kosong
   3. Memuat **Trending Movies Daily ViewModel**
      1. Memastikan **API dengan endpoint Trending Movies Daily** berhasil di akses
      2. Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Movies Daily** sama (waktu tunggu 10 detik)
   4. Memuat **Trending Movies Weekly ViewModel**
      1. Memastikan **API dengan endpoint Trending Movies Weekly** berhasil di akses
      2. Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Movies Weekly** sama (waktu tunggu 10 detik)
2. TrendingTelevisionShowViewModelTest
   1. Memuat **Trending Television Show Daily**
      1. Memastikan **API dengan endpoint Trending Television Show Daily** berhasil di akses
      2. Memastikan **Response Body** yang dimuat dari **API endpoint Trending Television Show Daily** tidak *null*
      3. Memastikan isi dari **key ‘result’** pada **API endpoint Trending Television Show Daily** tidak nol
   2. Memuat **Trending Television Show Weekly**
      1. Memastikan **API dengan endpoint Trending Television Show Weekly** berhasil di akses
      2. Memastikan **Response Body** yang dimuat dari **API endpoint Trending Television Show Weekly** tidak *null*
      3. Memastikan isi dari key **‘result’** pada **API endpoint Trending Television Show Weekly** tidak kosong
   3. Memuat **Trending Television Show Daily ViewModel**
      1. Memastikan **API dengan endpoint Trending Television Show Daily** berhasil di akses
      2. Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Television Show Daily** sama (waktu tunggu 10 detik)
   4. Memuat **Trending Television Show Weekly ViewModel**
      1. Memastikan **API dengan endpoint Trending Television Show Weekly** berhasil di akses
      2. Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Television Show Weekly** sama (waktu tunggu 10 detik)
3. DetailMoviesViewModelTest
   1. Memuat Detail Movies API
      1. Memastikan **API dengan endpoint Detail Movies** berhasil di akses
      2. Memastikan hasil keluaran **API** tidak *null*
      3. Memastikan hasil keluaran API dengan key **‘originalName’** sesuai dengan harapan
      4. Memastikan hasil keluaran API dengan key **‘releaseDate’** sesuai dengan harapan
   2. Memuat Detail Movies ViewModel
      1. Memastikan **API dengan endpoint Detail Movies** berhasil di akses
      2. Memastikan hasil keluaran **ViewModel** dengan **API endpoint Detail Movies** sama (waktu tunggu 10 detik)
   3. Memuat Detail Movies API Error
      1. Memastikan **API dengan endpoint Detail Movies** berhasil di akses
      2. Memastikan hasil keluaran ViewModel **error** saat Movie ID diisi nilai nol
4. DetailTelevisionShowViewModelTest
   1. Memuat Detail Television Show API
      1. Memastikan **API dengan endpoint Detail Television Show** berhasil di akses
      2. Memastikan hasil keluaran **API** tidak *null*
      3. Memastikan hasil keluaran API dengan key **‘originalName’** sesuai dengan harapan
      4. Memastikan hasil keluaran API dengan key **‘firstAirDate’** sesuai dengan harapan
   2. Memuat Detail Television Show ViewModel
      1. Memastikan **API dengan endpoint Detail Television Show** berhasil di akses
      2. Memastikan hasil keluaran **ViewModel** dengan **API endpoint Detail Movies** sama (waktu tunggu 10 detik)
   3. Memuat Detail Television Show API Error
      1. Memastikan **API dengan endpoint Detail Television Show** berhasil di akses
      2. Memastikan hasil keluaran ViewModel **error** saat Movie ID diisi nilai nol
5. ConverterTest
   1. Memastikan **Converter List Movie Genre** ke String memiliki keluaran yang diharapkan
   2. Memastikan **Converter List Television Show Genre** ke String memiliki keluaran yang diharapkan
   3. Memastikan **Converter List Movie Genre** menghasilkan nilai String kosong apabila tidak ada item pada list
   4. Memastikan **Converter List Television Show Genre** menghasilkan nilai String kosong apabila tidak ada item pada list
   5. Memastikan perhitungan durasi **163 Menit** ke **Jam, Menit** menghasilkan **2h 43m**

**Instrumented Test**

1. SplashActivityTest
   1. Memastikan **MainActivity** muncul setelah **Splash Screen** muncul selama 3 detik.
2. MainActivityTest
   1. Memastikan **Fragment Movies** tampil
      1. Memberi tindakan klik pada item Movies di BottomNavigationBar
      2. Memastikan Fragment Movies tampil pada layar
      3. Memastikan judul pada ActionBar sesuai dengan yang diharapkan
   2. Memastikan **Fragment Television Show** tampil
      1. Memberi tindakan klik pada **item Television Show** di BottomNavigationBar
      2. Memastikan **Fragment Television Show** tampil pada layar
      3. Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
   3. Memastikan **Fragment Bookmark** tampil
      1. Memberi tindakan klik pada **item Bookmark** di BottomNavigationBar
      2. Memastikan **Fragment Bookmark** tampil pada layar
      3. Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
   4. Memastikan **Fragment About** tampil
      1. Memberi tindakan klik pada **item About** di BottomNavigationBar
      2. Memastikan **Fragment About** tampil pada layar
      3. Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
   5. Memastikan daftar **Movie Trending harian** tampil
      1. Memberi tindakan klik pada item **Movies** di BottomNavigationBar
      2. Memastikan data yang tampil berjumlah **20 item**
      3. Memberi tindakan klik pada **item pertama** pada daftar **Trending Movies Daily**
      4. Memastikan **image\_movie\_poster** tampil
      5. Memastikan **text\_movie\_title** tampil dan tidak kosong
      6. Memastikan **text\_released\_duration** tampil dan tidak kosong
      7. Memastikan **text\_genres** tampil dan tidak kosong
   6. Memastikan daftar **Movie Trending mingguan** tampil
      1. Memberi tindakan klik pada item **Movies** di BottomNavigationBar
      2. Memastikan data yang tampil berjumlah **20 item**
      3. Memberi tindakan klik pada **item pertama** pada daftar **Trending Movies Weekly**
      4. Memastikan **image\_movie\_poster** tampil
      5. Memastikan **text\_movie\_title** tampil dan tidak kosong
      6. Memastikan **text\_released\_duration** tampil dan tidak kosong
      7. Memastikan **text\_genres** tampil dan tidak kosong
   7. Memastikan daftar **Television Show Trending harian** tampil
      1. Memberi tindakan klik pada item **Television Show** di BottomNavigationBar
      2. Memastikan data yang tampil berjumlah **20 item**
      3. Memberi tindakan klik pada **item pertama** pada daftar **Trending Television Show Daily**
      4. Memastikan **image\_tv\_poster** tampil
      5. Memastikan **text\_tv\_title** tampil dan tidak kosong
      6. Memastikan **text\_first\_air\_duration** tampil dan tidak kosong
      7. Memastikan **text\_genres** tampil dan tidak kosong
   8. Memastikan daftar **Television Show Trending mingguan** tampil
      1. Memberi tindakan klik pada item **Television Show** di BottomNavigationBar
      2. Memastikan data yang tampil berjumlah **20 item**
      3. Memberi tindakan klik pada **item pertama** pada daftar **Trending Television Show Daily**
      4. Memastikan **image\_tv\_poster** tampil
      5. Memastikan **text\_tv\_title** tampil dan tidak kosong
      6. Memastikan **text\_first\_air\_duration** tampil dan tidak kosong
      7. Memastikan **text\_genres** tampil dan tidak kosong

Di Submission ini, saya mengikuti beberapa saran, yakni

* Menerapkan tampilan aplikasi yang sesuai standar:
  + Tampilan aplikasi memiliki *width*, *height*, *margin*, dan *padding*yang sesuai.
  + Pemilihan warna yang sesuai tema aplikasi.
  + Tidak ada komponen yang saling bertumpuk.
  + Penggunaan komponen yang sesuai dengan fungsinya.  
    Contoh : Komponen ImageView yang dijadikan sebagai button navigasi.
* Menuliskan kode dengan *bersih*.
  + Bersihkan comment dan kode yang tidak digunakan.
  + Indentasi yang sesuai.
  + Menghapus import yang tidak digunakan.
* Menambahkan test case untuk skenario pengujian berikut:
  + Data gagal dimuat.
* Menggunakan data **API Movie Catalogue** dan **indikator loading** ketika memuat data.
* Menggunakan library pihak ketiga seperti **Retrofit**, Fast Android Networking, dsb.
* Menggunakan Dependency Injection Library seperti Dagger, Koin, dsb. (Menggunakan **Dagger Hilt**)

Namun sebagai catatan, saya menggunakan **Double Bang Operator (!!)** pada binding agar **menghindari adanya kebocoran memory** (Leaked Memory)

**Skenario Uji Unit (Unit Test) dan Uji Instrumen (Instrumented Test):**

**Unit Testing**

* TrendingMoviesViewModelTest
  + Memuat **Trending Movies Daily**
    - Memastikan **API dengan endpoint Trending Movies Daily** berhasil di akses
    - Memastikan **Response Body** yang dimuat dari **API endpoint Trending Movies Daily** tidak *null*
    - Memastikan isi dari **key ‘result’** pada **API endpoint Trending Movies Daily** tidak nol
  + Memuat **Trending Movies Weekly**
    - Memastikan **API dengan endpoint Trending Movies Weekly** berhasil di akses
    - Memastikan **Response Body** yang dimuat dari **API endpoint Trending Movies Weekly** tidak *null*
    - Memastikan isi dari key **‘result’** pada **API endpoint Trending Movies Weekly** tidak kosong
  + Memuat **Trending Movies Daily ViewModel**
    - Memastikan **API dengan endpoint Trending Movies Daily** berhasil di akses
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Movies Daily** sama (waktu tunggu 10 detik)
  + Memuat **Trending Movies Weekly ViewModel**
    - Memastikan **API dengan endpoint Trending Movies Weekly** berhasil di akses
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Movies Weekly** sama (waktu tunggu 10 detik)
* TrendingTelevisionShowViewModelTest
  + Memuat **Trending Television Show Daily**
    - Memastikan **API dengan endpoint Trending Television Show Daily** berhasil di akses
    - Memastikan **Response Body** yang dimuat dari **API endpoint Trending Television Show Daily** tidak *null*
    - Memastikan isi dari **key ‘result’** pada **API endpoint Trending Television Show Daily** tidak nol
  + Memuat **Trending Television Show Weekly**
    - Memastikan **API dengan endpoint Trending Television Show Weekly** berhasil di akses
    - Memastikan **Response Body** yang dimuat dari **API endpoint Trending Television Show Weekly** tidak *null*
    - Memastikan isi dari key **‘result’** pada **API endpoint Trending Television Show Weekly** tidak kosong
  + Memuat **Trending Television Show Daily ViewModel**
    - Memastikan **API dengan endpoint Trending Television Show Daily** berhasil di akses
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Television Show Daily** sama (waktu tunggu 10 detik)
  + Memuat **Trending Television Show Weekly ViewModel**
    - Memastikan **API dengan endpoint Trending Television Show Weekly** berhasil di akses
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Television Show Weekly** sama (waktu tunggu 10 detik)
* DetailMoviesViewModelTest
  + Memuat Detail Movies API
    - Memastikan **API dengan endpoint Detail Movies** berhasil di akses
    - Memastikan hasil keluaran **API** tidak *null*
    - Memastikan hasil keluaran API dengan key **‘originalName’** sesuai dengan harapan
    - Memastikan hasil keluaran API dengan key **‘releaseDate’** sesuai dengan harapan
  + Memuat Detail Movies ViewModel
    - Memastikan **API dengan endpoint Detail Movies** berhasil di akses
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Detail Movies** sama (waktu tunggu 10 detik)
  + Memuat Detail Movies API Error
    - Memastikan **API dengan endpoint Detail Movies** berhasil di akses
    - Memastikan hasil keluaran ViewModel **error** saat Movie ID diisi nilai nol
* DetailTelevisionShowViewModelTest
  + Memuat Detail Television Show API
    - Memastikan **API dengan endpoint Detail Television Show** berhasil di akses
    - Memastikan hasil keluaran **API** tidak *null*
    - Memastikan hasil keluaran API dengan key **‘originalName’** sesuai dengan harapan
    - Memastikan hasil keluaran API dengan key **‘firstAirDate’** sesuai dengan harapan
  + Memuat Detail Television Show ViewModel
    - Memastikan **API dengan endpoint Detail Television Show** berhasil di akses
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Detail Movies** sama (waktu tunggu 10 detik)
  + Memuat Detail Television Show API Error
    - Memastikan **API dengan endpoint Detail Television Show** berhasil di akses
    - Memastikan hasil keluaran ViewModel **error** saat Movie ID diisi nilai nol
* MoviesRepositoryTest
  + Memuat Trending Movies Daily
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan MoviesRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari MoviesRepository tidak null
  + Memuat Trending Movies Weekly
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan MoviesRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari MoviesRepository tidak null
  + Memuat Movie Details
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan MoviesRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari MoviesRepository tidak null
    - Memverifikasi keluaran observer dari ViewModel benar dan memiliki keluaran bebas ber-class MovieDetailsResponse
* TelevisionShowRepositoryTest
  + Memuat Trending Television Show Daily
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan TelevisionShowRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari TelevisionShowRepository tidak null
  + Memuat Trending Television Show Weekly
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan TelevisionShowRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari TelevisionShowRepository tidak null
  + Memuat Movie Details
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan TelevisionShowRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari TelevisionShowRepository tidak null
    - Memverifikasi keluaran observer dari ViewModel benar dan memiliki keluaran bebas ber-class TelevisionDetailsResponse
* ConverterTest
  + Memastikan **Converter List Movie Genre** ke String memiliki keluaran yang diharapkan
  + Memastikan **Converter List Television Show Genre** ke String memiliki keluaran yang diharapkan
  + Memastikan **Converter List Movie Genre** menghasilkan nilai String kosong apabila tidak ada item pada list
  + Memastikan **Converter List Television Show Genre** menghasilkan nilai String kosong apabila tidak ada item pada list
  + Memastikan perhitungan durasi **163 Menit** ke **Jam, Menit** menghasilkan **2h 43m**

**Instrumented Test**

* SplashActivityTest
  + Memastikan **MainActivity** muncul setelah **Splash Screen** muncul selama 3 detik.
* MainActivityTest
  + Memastikan **Fragment Movies** tampil
    - Memberi tindakan klik pada item Movies di BottomNavigationBar
    - Memastikan Fragment Movies tampil pada layar
    - Memastikan judul pada ActionBar sesuai dengan yang diharapkan
  + Memastikan **Fragment Television Show** tampil
    - Memberi tindakan klik pada **item Television Show** di BottomNavigationBar
    - Memastikan **Fragment Television Show** tampil pada layar
    - Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
  + Memastikan **Fragment Bookmark** tampil
    - Memberi tindakan klik pada **item Bookmark** di BottomNavigationBar
    - Memastikan **Fragment Bookmark** tampil pada layar
    - Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
  + Memastikan **Fragment About** tampil
    - Memberi tindakan klik pada **item About** di BottomNavigationBar
    - Memastikan **Fragment About** tampil pada layar
    - Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
  + Memastikan daftar **Movie Trending harian** tampil
    - Memberi tindakan klik pada item **Movies** di BottomNavigationBar
    - Memastikan data yang tampil berjumlah **20 item**
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Movies Daily**
    - Memastikan **image\_movie\_poster** tampil
    - Memastikan **text\_movie\_title** tampil dan tidak kosong
    - Memastikan **text\_released\_duration** tampil dan tidak kosong
    - Memastikan **text\_genres** tampil dan tidak kosong
  + Memastikan daftar **Movie Trending mingguan** tampil
    - Memberi tindakan klik pada item **Movies** di BottomNavigationBar
    - Memastikan data yang tampil berjumlah **20 item**
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Movies Weekly**
    - Memastikan **image\_movie\_poster** tampil
    - Memastikan **text\_movie\_title** tampil dan tidak kosong
    - Memastikan **text\_released\_duration** tampil dan tidak kosong
    - Memastikan **text\_genres** tampil dan tidak kosong
  + Memastikan daftar **Television Show Trending harian** tampil
    - Memberi tindakan klik pada item **Television Show** di BottomNavigationBar
    - Memastikan data yang tampil berjumlah **20 item**
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Television Show Daily**
    - Memastikan **image\_tv\_poster** tampil
    - Memastikan **text\_tv\_title** tampil dan tidak kosong
    - Memastikan **text\_first\_air\_duration** tampil dan tidak kosong
    - Memastikan **text\_genres** tampil dan tidak kosong
  + Memastikan daftar **Television Show Trending mingguan** tampil
    - Memberi tindakan klik pada item **Television Show** di BottomNavigationBar
    - Memastikan data yang tampil berjumlah **20 item**
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Television Show Daily**
    - Memastikan **image\_tv\_poster** tampil
    - Memastikan **text\_tv\_title** tampil dan tidak kosong
    - Memastikan **text\_first\_air\_duration** tampil dan tidak kosong
    - Memastikan **text\_genres** tampil dan tidak kosong

--------------------------

Di Submission ini, saya mengikuti beberapa saran, yakni

* Menerapkan tampilan aplikasi yang sesuai standar:
  + Tampilan aplikasi memiliki *width*, *height*, *margin*, dan *padding*yang sesuai.
  + Pemilihan warna yang sesuai tema aplikasi.
  + Tidak ada komponen yang saling bertumpuk.
  + Penggunaan komponen yang sesuai dengan fungsinya.  
    Contoh : Komponen ImageView yang dijadikan sebagai button navigasi.
* Menuliskan kode dengan *bersih*.
  + Bersihkan comment dan kode yang tidak digunakan.
  + Indentasi yang sesuai.
  + Menghapus import yang tidak digunakan.
* Menambahkan test case untuk skenario pengujian berikut:
  + Data gagal dimuat.
* Menggunakan data **API Movie Catalogue** dan **indikator loading** ketika memuat data.
* Menggunakan library pihak ketiga seperti **Retrofit**, Fast Android Networking, dsb.
* Menggunakan Dependency Injection Library seperti Dagger, Koin, dsb. (Menggunakan **Dagger Hilt**)
* Menerapkan testing untuk fungsi menambah dan menghapus item favorite. **(Fitur Favorite digantikan dengan Bookmark, konsepnya sama, hanya berbeda nama dan ikon saja)**

Dan menanggapi catatan pada Reviewer Note pada submission selanjutnya, yaitu:

* Penerapan Instrumentation Test belum tepat. Pada halaman detail belum dilakukan pengujian untuk seluruh komponen yang ada di halaman detail :
  + Button bookmark **(fungsi ini sudah dilakukan testing pada fungsi shouldBookmarkMovieWorked() dan shouldBookmarkTelevisionShowWorked() dan ada pada submission sebelumnya)**
  + Progressbar user score **(sudah saya tambahkan saat test Trending Movies dan Trending Television Show, baik itu Daily maupun Weekly)**
  + Textview user score **(sudah saya tambahkan saat test Trending Movies dan Trending Television Show, baik itu Daily maupun Weekly)**
  + dan lain-lain (termasuk action refresh dan share) pada action bar **(sudah ditambahkan. Fungsinya: shouldMovieDetailsCanBeRefreshed() dan shouldTelevisionShowDetailsCanBeRefreshed())**
* Pengunaan double bang (!!) pada binding tidak akan mencegah aplikasi ada dari memory leak, untuk menghindari memory leak pastikan untuk membuat binding menjadi null pada saat fragment di destroy, kamu bisa memanfaatkan onDestroyView() untuk mengembalikan binding menjadi null (sudah Anda terapkan). Hindari penggunaan double bang operator (!!) saat pengecekan null, karena akan memaksa suatu variable menjadi non-null. Dan jika ternyata variable tersebut bernilai null, maka bisa menyebabkan NPE. Periksa kembali semua kode kamu dan jangan biarkan satupun operator tersebut tersisa. **(sudah saya perbaiki pada Activity maupun Fragment)**

Dan sebagai tanggapan tambahan pada catatan pertama, semuanya dilakukan dari **MainActivityTest** agar halaman detail mendapatkan data yang akurat (langsung dari API nya)

**Skenario Uji Unit (Unit Test) dan Uji Instrumen (Instrumented Test):**

**Unit Testing**

* TrendingMoviesViewModelTest
  + Memuat **Trending Movies Daily**
    - Memastikan **API dengan endpoint Trending Movies Daily** berhasil di akses
    - Memastikan **Response Body** yang dimuat dari **API endpoint Trending Movies Daily** tidak *null*
    - Memastikan isi dari **key ‘result’** pada **API endpoint Trending Movies Daily** tidak nol
  + Memuat **Trending Movies Weekly**
    - Memastikan **API dengan endpoint Trending Movies Weekly** berhasil di akses
    - Memastikan **Response Body** yang dimuat dari **API endpoint Trending Movies Weekly** tidak *null*
    - Memastikan isi dari key **‘result’** pada **API endpoint Trending Movies Weekly** tidak kosong
  + Memuat **Trending Movies Daily ViewModel**
    - Mengambil data dari **API endpoint Trending Movies Daily** sebagai data untuk *testing*
    - Memanipulasi kelas **Repository** sebagai obyek tiruan
    - Memanipulasi kelas **Retrofit API** serta **hasil keluarannya** sebagai obyek tiruan
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Movies Daily** sama (waktu tunggu 10 detik)
  + Memuat **Trending Movies Weekly ViewModel**
    - Mengambil data dari **API endpoint Trending Movies Weekly** sebagai data untuk *testing*
    - Memanipulasi kelas **Repository** sebagai obyek tiruan
    - Memanipulasi kelas **Retrofit API** serta **hasil keluarannya** sebagai obyek tiruan
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Movies Weekly** sama (waktu tunggu 10 detik)
* TrendingTelevisionShowViewModelTest
  + Memuat **Trending Television Show Daily**
    - Memastikan **API dengan endpoint Trending Television Show Daily** berhasil di akses
    - Memastikan **Response Body** yang dimuat dari **API endpoint Trending Television Show Daily** tidak *null*
    - Memastikan isi dari **key ‘result’** pada **API endpoint Trending Television Show Daily** tidak nol
  + Memuat **Trending Television Show Weekly**
    - Memastikan **API dengan endpoint Trending Television Show Weekly** berhasil di akses
    - Memastikan **Response Body** yang dimuat dari **API endpoint Trending Television Show Weekly** tidak *null*
    - Memastikan isi dari key **‘result’** pada **API endpoint Trending Television Show Weekly** tidak kosong
  + Memuat **Trending Television Show Daily ViewModel**
    - Mengambil data dari **API dengan endpoint Trending Television Show Daily** sebagai data untuk *testing*
    - Memanipulasi kelas **Repository** sebagai obyek tiruan
    - Memanipulasi kelas **Retrofit API** serta **hasil keluarannya** sebagai obyek tiruan
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Television Show Daily** sama (waktu tunggu 10 detik)
  + Memuat **Trending Television Show Weekly ViewModel**
    - Mengambil data dari **API dengan endpoint Trending Television Show Weekly** sebagai data untuk *testing*
    - Memanipulasi kelas **Repository** sebagai obyek tiruan
    - Memanipulasi kelas **Retrofit API** serta **hasil keluarannya** sebagai obyek tiruan
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Trending Television Show Weekly** sama (waktu tunggu 10 detik)
* DetailMoviesViewModelTest
  + Memuat Detail Movies API
    - Memastikan **API dengan endpoint Detail Movies** berhasil di akses
    - Memastikan hasil keluaran **API** tidak *null*
    - Memastikan hasil keluaran API dengan key **‘originalName’** sesuai dengan harapan
    - Memastikan hasil keluaran API dengan key **‘releaseDate’** sesuai dengan harapan
  + Memuat Detail Movies ViewModel
    - Mengambil data dari **API dengan endpoint Detail Movies** sebagai data untuk *testing*
    - Memanipulasi kelas **Repository** sebagai obyek tiruan
    - Memanipulasi kelas **Retrofit API** serta **hasil keluarannya** sebagai obyek tiruan
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Detail Movies** sama
  + Memuat Detail Movies API Error
    - Mengambil data dari **API dengan endpoint Detail Movies** sebagai data untuk *testing*
    - Memanipulasi kelas **Repository** sebagai obyek tiruan
    - Memanipulasi kelas **Retrofit API** serta **hasil keluarannya** sebagai obyek tiruan
    - Memastikan hasil keluaran ViewModel **error** saat Movie ID diisi nilai nol
* DetailTelevisionShowViewModelTest
  + Memuat Detail Television Show API
    - Memastikan **API dengan endpoint Detail Television Show** berhasil di akses
    - Memastikan hasil keluaran **API** tidak *null*
    - Memastikan hasil keluaran API dengan key **‘originalName’** sesuai dengan harapan
    - Memastikan hasil keluaran API dengan key **‘firstAirDate’** sesuai dengan harapan
  + Memuat Detail Television Show ViewModel
    - Mengambil data dari **API dengan endpoint Detail Television Show** sebagai data untuk *testing*
    - Memanipulasi kelas **Repository** sebagai obyek tiruan
    - Memanipulasi kelas **Retrofit API** serta **hasil keluarannya** sebagai obyek tiruan
    - Memastikan hasil keluaran **ViewModel** dengan **API endpoint Detail Television Show** sama
  + Memuat Detail Television Show API Error
    - Mengambil data dari **API dengan endpoint Detail Television Show** sebagai data untuk *testing*
    - Memanipulasi kelas **Repository** sebagai obyek tiruan
    - Memanipulasi kelas **Retrofit API** serta **hasil keluarannya** sebagai obyek tiruan
    - Memastikan hasil keluaran ViewModel **error** saat Movie ID diisi nilai nol
* BookmarkRepositoryTest
  + Memuat Bookmark Movies
    - Memuat obyek tiruan untuk class LocalDataSource, dan BookmarkRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari BookmarkRepository tidak null
    - Memastikan jumlah data dari respon tidak nol
  + Memuat Bookmark Television Show
* MoviesRepositoryTest
  + Memuat Trending Movies Daily
    - Memuat obyek tiruan untuk class LocalDataSource, dan BookmarkRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari BookmarkRepository tidak null
    - Memastikan jumlah data dari respon tidak nol
  + Memuat Trending Movies Weekly
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan MoviesRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari MoviesRepository tidak null
  + Memuat Movie Details
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan MoviesRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari MoviesRepository tidak null
    - Memverifikasi keluaran observer dari ViewModel benar dan memiliki keluaran bebas ber-class MovieDetailsResponse
* TelevisionShowRepositoryTest
  + Memuat Trending Television Show Daily
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan TelevisionShowRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari TelevisionShowRepository tidak null
  + Memuat Trending Television Show Weekly
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan TelevisionShowRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari TelevisionShowRepository tidak null
  + Memuat Movie Details
    - Memuat obyek tiruan untuk class Call, LocalDataSource, dan TelevisionShowRepository
    - Memverifikasi keluaran dari LocalDataSource
    - Memastikan respon dari TelevisionShowRepository tidak null
    - Memverifikasi keluaran observer dari ViewModel benar dan memiliki keluaran bebas ber-class TelevisionDetailsResponse
* ConverterTest
  + Memastikan **Converter List Movie Genre** ke String memiliki keluaran yang diharapkan
  + Memastikan **Converter List Television Show Genre** ke String memiliki keluaran yang diharapkan
  + Memastikan **Converter List Movie Genre** menghasilkan nilai String kosong apabila tidak ada item pada list
  + Memastikan **Converter List Television Show Genre** menghasilkan nilai String kosong apabila tidak ada item pada list
  + Memastikan perhitungan durasi **163 Menit** ke **Jam, Menit** menghasilkan **2h 43m**

**Instrumented Test**

* SplashActivityTest
  + Memastikan **MainActivity** muncul setelah **Splash Screen** muncul selama 3 detik.
* MainActivityTest
  + Memastikan **Fragment Movies** tampil
    - Memberi tindakan klik pada item Movies di BottomNavigationBar
    - Memastikan Fragment Movies tampil pada layar
    - Memastikan judul pada ActionBar sesuai dengan yang diharapkan
  + Memastikan **Fragment Television Show** tampil
    - Memberi tindakan klik pada **item Television Show** di BottomNavigationBar
    - Memastikan **Fragment Television Show** tampil pada layar
    - Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
  + Memastikan **Fragment Bookmark** tampil
    - Memberi tindakan klik pada **item Bookmark** di BottomNavigationBar
    - Memastikan **Fragment Bookmark** tampil pada layar
    - Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
  + Memastikan **Fragment About** tampil
    - Memberi tindakan klik pada **item About** di BottomNavigationBar
    - Memastikan **Fragment About** tampil pada layar
    - Memastikan **judul pada ActionBar** sesuai dengan yang diharapkan
  + Memastikan daftar **Movie Trending harian** tampil
    - Memberi tindakan klik pada item **Movies** di BottomNavigationBar
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Movies Daily**
    - Memastikan **image\_movie\_poster** tampil
    - Memastikan **text\_movie\_title** tampil dan tidak kosong
    - Memastikan **text\_released\_duration** tampil dan tidak kosong
    - Memastikan **text\_genres** tampil dan tidak kosong
    - Memastikan **pb\_user\_score** tampil
    - Memastikan **text\_user\_score** tampil dan tidak kosong
  + Memastikan daftar **Movie Trending mingguan** tampil
    - Memberi tindakan klik pada item **Movies** di BottomNavigationBar
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Movies Weekly**
    - Memastikan **image\_movie\_poster** tampil
    - Memastikan **text\_movie\_title** tampil dan tidak kosong
    - Memastikan **text\_released\_duration** tampil dan tidak kosong
    - Memastikan **text\_genres** tampil dan tidak kosong
    - Memastikan **pb\_user\_score** tampil
    - Memastikan **text\_user\_score** tampil dan tidak kosong
  + Memastikan daftar **Television Show Trending harian** tampil
    - Memberi tindakan klik pada item **Television Show** di BottomNavigationBar
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Television Show Daily**
    - Memastikan **image\_tv\_poster** tampil
    - Memastikan **text\_tv\_title** tampil dan tidak kosong
    - Memastikan **text\_first\_air\_duration** tampil dan tidak kosong
    - Memastikan **text\_genres** tampil dan tidak kosong
    - Memastikan **pb\_user\_score** tampil
    - Memastikan **text\_user\_score** tampil dan tidak kosong
  + Memastikan daftar **Television Show Trending mingguan** tampil
    - Memberi tindakan klik pada item **Television Show** di BottomNavigationBar
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Television Show Daily**
    - Memastikan **image\_tv\_poster** tampil
    - Memastikan **text\_tv\_title** tampil dan tidak kosong
    - Memastikan **text\_first\_air\_duration** tampil dan tidak kosong
    - Memastikan **text\_genres** tampil dan tidak kosong
    - Memastikan **pb\_user\_score** tampil
    - Memastikan **text\_user\_score** tampil dan tidak kosong
  + Memastikan fitur **Bookmark Movies** bekerja
    - Memberi tindakan klik pada item **Bookmark** di BottomNavigationBar
    - Memberi tindakan klik pada **tab pertama** pada halaman Bookmark
    - Memastikan **anim\_movie\_not\_found** tampil
    - Memastikan **text\_movie\_not\_found** tampil
    - Memberi tindakan klik pada item **Movie** di BottomNavigationBar
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Movies Daily**
    - Memberi tindakan klik pada komponen **button\_bookmark**
    - Memberi tindakan *backward*atau **menekan tombol Back** pada sistem
    - Memberi tindakan klik pada item **Bookmark** di BottomNavigationBar
    - Memberi tindakan klik pada **tab pertama** pada halaman Bookmark
    - **Menunggu 1 detik** untuk memastikan respon dari Paging bekerja semestinya
    - Memberi tindakan klik pada **item pertama** pada daftar **Movies Bookmark**
    - Memberi tindakan klik pada komponen **button\_bookmark**
    - Memberi tindakan *backward*atau **menekan tombol Back** pada sistem
    - Memberi tindakan klik pada **tab pertama** pada halaman Bookmark
    - Memastikan **anim\_movie\_not\_found** tampil
    - Memastikan **text\_movie\_not\_found** tampil
  + Memastikan fitur **Bookmark Television** Show bekerja
    - Memberi tindakan klik pada item **Bookmark** di BottomNavigationBar
    - Memberi tindakan klik pada **tab kedua** pada halaman Bookmark
    - Memastikan **anim\_tv\_not\_found** tampil
    - Memastikan **text\_tv\_not\_found** tampil
    - Memberi tindakan klik pada item **Television Show** di BottomNavigationBar
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Television Show Daily**
    - Memberi tindakan klik pada komponen **button\_bookmark**
    - Memberi tindakan *backward*atau **menekan tombol Back** pada sistem
    - Memberi tindakan klik pada item **Bookmark** di BottomNavigationBar
    - Memberi tindakan klik pada **tab kedua** pada halaman Bookmark
    - **Menunggu 1 detik** untuk memastikan respon dari Paging bekerja semestinya
    - Memberi tindakan klik pada **item pertama** pada daftar **Television Show Bookmark**
    - Memberi tindakan klik pada komponen **button\_bookmark**
    - Memberi tindakan *backward*atau **menekan tombol Back** pada sistem
    - Memberi tindakan klik pada **tab kedua** pada halaman Bookmark
    - Memastikan **anim\_movie\_not\_found** tampil
    - Memastikan **text\_movie\_not\_found** tampil
  + Memastikan informasi **Movies** dapat dibagikan
    - Memberi tindakan klik pada item **Movies** di BottomNavigationBar
    - **Menunggu 1 detik** untuk memastikan respon dari Paging bekerja semestinya
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Movies Daily**
    - Memberi tindakan klik pada **menu\_share** di bagian ActionBar
    - Memastikan dialog **Bagikan** (bawaan sistem) tampil
    - Memberi tindakan **klik tombol Kembali** pada System UI Android
  + Memastikan informasi **Television Show** dapat dibagikan
    - Memberi tindakan klik pada item **Television Show** di BottomNavigationBar
    - **Menunggu 1 detik** untuk memastikan respon dari Paging bekerja semestinya
    - Memberi tindakan klik pada **item pertama** pada daftar **Trending Television Show Daily**
    - Memberi tindakan klik pada **menu\_share** di bagian ActionBar
    - Memastikan dialog **Bagikan** (bawaan sistem) tampil
    - Memberi tindakan **klik tombol Kembali** pada System UI Android
  + Memastikan semua fitur dalam **About** bekerja
    - Memberi tindakan klik pada item **About** di BottomNavigationBar
    - Memberi tindakan klik pada **item pertama** pada daftar **Options**
    - Memastikan bahwa aksi intent **ACTION\_VIEW** berhasil dijalankan dan memiliki data **myFacebook**
    - Memberi tindakan **klik tombol Kembali** pada System UI Android
    - Menunggu hingga aplikasi kembali terdeteksi dengan waktu habis tunggu 1 detik
    - Memberi tindakan klik pada **item kedua** pada daftar **Options**
    - Memastikan bahwa aksi intent **ACTION\_VIEW** berhasil dijalankan dan memiliki data **myGitHub**
    - Memberi tindakan **klik tombol Kembali** pada System UI Android
    - Menunggu hingga aplikasi kembali terdeteksi dengan waktu habis tunggu 1 detik
    - Memberi tindakan klik pada **item ketiga** pada daftar **Options**
    - Memastikan bahwa aksi intent **ACTION\_SENDTO** berhasil dijalankan dan memiliki data **myEmail**
    - Memberi tindakan **klik tombol Kembali** pada System UI Android
    - Memberi tindakan **klik tombol Kembali** pada System UI Android
    - Menunggu hingga aplikasi kembali terdeteksi dengan waktu habis tunggu 1 detik