* Test get trending film from api and return success
  + Menginisiasi kelas **TrendingPagingSource**
  + Memanipulasi data ketika pemanggilan data trendingFilm di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan data yang dihasilkan kelas **paging source** sesuai dengan yang diharapkan
* Test get popular movies from api and return success
  + Menginisiasi kelas **MoviePagingSource**
  + Memanipulasi data ketika pemanggilan data popular movies di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan data yang dihasilkan kelas **paging source** sesuai dengan yang diharapkan
* Test get popular series from api and return success
  + Menginisiasi kelas **SeriesPagingSource**
  + Memanipulasi data ketika pemanggilan data popular series di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan data yang dihasilkan kelas **paging source** sesuai dengan yang diharapkan
* Test get trending film and return no data
  + Menginisiasi kelas **TrendingPagingSource**
  + Memanipulasi data ketika pemanggilan data trendingFilm di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test get popular movies and return no data
  + Menginisiasi kelas **MoviesPagingSource**
  + Memanipulasi data ketika pemanggilan data popular movies di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test get popular series and return no data
  + Menginisiasi kelas **SeriesPagingSource**
  + Memanipulasi data ketika pemanggilan data popular series di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test get trending film and return request error
  + Menginisiasi kelas **TrendingPagingSource**
  + Memanipulasi data ketika pemanggilan data trending film di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test get popular movies and return request error
  + Menginisiasi kelas **MoviesPagingSource**
  + Memanipulasi data ketika pemanggilan popular movies film di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test get popular series and return request error
  + Menginisiasi kelas **SeriesPagingSource**
  + Memanipulasi data ketika pemanggilan data popular series di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test get trending films and return connection error
  + Menginisiasi kelas **TrendingPagingSource**
  + Memanipulasi data ketika pemanggilan data trending film di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test get popular movies and return connection error
  + Menginisiasi kelas **MoviesPagingSource**
  + Memanipulasi data ketika pemanggilan data popular movies di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test get popular series and return connection error
  + Menginisiasi kelas **SeriesPagingSource**
  + Memanipulasi data ketika pemanggilan data popular series di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Test search film by query from api and return success
  + Menginisiasi kelas **SearchPagingSource**
  + Memanipulasi data ketika pemanggilan data trending film di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan data yang dihasilkan kelas **paging source** sesuai dengan yang diharapkan
* Test search film by query from api and return no data
  + Menginisiasi kelas **SearchPagingSource**
  + Memanipulasi data ketika pemanggilan data trendingFilm di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Search film by query from api and return request error
  + Menginisiasi kelas **SearchPagingSource**
  + Memanipulasi data ketika pemanggilan data trending film di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
* Search film by query from api and return connection error
  + Menginisiasi kelas **SearchPagingSource**
  + Memanipulasi data ketika pemanggilan data trending film di kelas **remote data source**
  + Memuat data melalui **paging source**
  + Memastikan metode pada kelas **data source** terpanggil 1 kali
  + Memastikan status response data sama dengan error
  + Memastikan konten error sesuai dengan yang diharapkan
  + Test get movies detail from api and return success
* Memanipulasi data ketika pemanggilan data movie di kelas **repository**
* Memastikan metode di kelas **repository** terpanggil
* Memastikan status response **success**
* Memastikan data response tidak null
* Memastikan data **detail** tidak null
* Memastikan data **title** tidak null
* Memastikan data **title** sesuai dengan yang diharapkan
* Memastikan data **genre** tidak null
* Memastikan jumlah data **genre** sesuai dengan yang diharapkan
* Memastikan data **poster** tidak null
* Memastikan data **poster** sesuai dengan yang diharapkan
* Memastikan data **rating** tidak null
* Memastikan data **rating** sesuai dengan yang diharapkan
* Memastikan data **releaseDate** tidak null
* Memastikan data **releaseDate** sesuai dengan yang diharapkan
* Memastikan data **synopsis** tidak null
* Memastikan data **synopsis** sesuai dengan yang diharapkan
* Memastikan data **type** tidak null
* Memastikan data **type** sesuai dengan yang diharapkan
  + Test get series detail from api and return success
* Memanipulasi data ketika pemanggilan data movie di kelas **repository**
* Memastikan metode di kelas **repository** terpanggil
* Memastikan status response **success**
* Memastikan data response tidak null
* Memastikan data **detail** tidak null
* Memastikan data **title** tidak null
* Memastikan data **title** sesuai dengan yang diharapkan
* Memastikan data **genre** tidak null
* Memastikan jumlah data **genre** sesuai dengan yang diharapkan
* Memastikan data **poster** tidak null
* Memastikan data **poster** sesuai dengan yang diharapkan
* Memastikan data **rating** tidak null
* Memastikan data **rating** sesuai dengan yang diharapkan
  + Memastikan data **releaseDate** tidak null
  + Memastikan data **releaseDate** sesuai dengan yang diharapkan
  + Memastikan data **synopsis** tidak null
  + Memastikan data **synopsis** sesuai dengan yang diharapkan
  + Memastikan data **type** tidak null
  + Memastikan data **type** sesuai dengan yang diharapkan
  + Test get detail movies from api and return no data
  + Memanipulasi data ketika pemanggilan data trendingFilm di kelas **repository**
  + Memastikan metode di kelas **repository** terpanggil
  + Memastikan status response **no data**
  + Test get detail series from api and return no data
  + Memanipulasi data ketika pemanggilan data movies di kelas **repository**
  + Memastikan metode di kelas **repository** terpanggil
  + Memastikan status response **no data**
  + Test get detail movies from api and return request error
  + Memanipulasi data ketika pemanggilan data trendingFilm di kelas **repository**
  + Memastikan metode di kelas **repository** terpanggil
  + Memastikan status response **error**
  + Memastikan data **message** tidak null
  + Memastikan data **message** sesuai dengan yang diharapkan
  + Test get detail series from api and return request error
  + Memanipulasi data ketika pemanggilan data movies di kelas **repository**
  + Memastikan metode di kelas **repository** terpanggil
  + Memastikan status response **error**
  + Memastikan data **message** tidak null
  + Memastikan data **message** sesuai dengan yang diharapkan
  + Test get detail movies from api and return connection error
  + Memanipulasi data ketika pemanggilan data trendingFilm di kelas **repository**
  + Memastikan metode di kelas **repository** terpanggil
  + Memastikan status response **error**
  + Memastikan data **message** tidak null
  + Memastikan data **message** sesuai dengan yang diharapkan
  + Test get detail series from api and return connection error
  + Memanipulasi data ketika pemanggilan data movies di kelas **repository**
  + Memastikan metode di kelas **repository** terpanggil
  + Memastikan status response **error**
  + Memastikan data **message** tidak null
  + Memastikan data **message** sesuai dengan yang diharapkan