**Mobile Final Project Documentation**

Nama : Immanuel Joseph

NIM : 2301852215

Role : Junior Programmer

**My Application :**

* Navigation Graph

Sesuai dengan ketentuan project, saya telah membuat aplikasi yang serupa dengan instruksi pada soal dan contoh aplikasi Kepo itu sendiri. Navigation Graph yang saya gunakan adalah yang terdapat pada soal, sehingga aplikasi saya memiliki alur activity yang identik dengan yang ditentukan pada soal. Saya akan tampilkan navigation graph tesrsebut di bawah ini.

Saya juga telah mengikuti revisi yang diinstruksikan bahwa, kotak Search Todo (Activity setelah Search User) diganti dengan User Detail.

* UI

User Interface dari aplikasi saya, juga telah saya buat secantik mungkin, mengikuti gambaran pada soal dan contoh aplikasi Kepo itu sendiri. Saya berusaha memanfaatkan bantuan layout bawaan android seperti Constraint Layout, Relative Layout, ataupun Coordinator Layout dalam menetukan size item, position, dan margin. Sehingga dapat menyesuaikan ukuran layar device. Namun, terdapat beberapa item yang perlu diatur secara manual, sehingga menjadi kurang dinamis dalam menyesuaikan ukuran layar device. Untuk menentukan warna, saya memilih komposisi warna yang sesuai dengan warna-warna yang digunakan contoh aplikasi.

* Logic

Karena saya menerapkan konsep MVVM, maka secara garis besar seharusnya tidak terdapat banyak perbedaan dalam mengembangkan aplikasi Kepo tersebut. Hampir diseluruh activity yang terdapat pada aplikasi saya menggunakan ViewModel. Untuk activity yang perlu melakukan request ke api, maka yang akan terjadi pada logic aplikasi saya adalah sebagai berikut : Activity (requesting api and observing response) 🡪 ViewModel 🡪 Repository (passing request) 🡪 ApiClient (accessing api interface) 🡪 ApiInterface (returns retrofit call method model) 🡪 ApiClient (executing retrofit call method and processing response). Kemudian, Activity akan mendapatkan notifikasi segera setelah response berhasil diproses dan valuenya berhasil disimpan ke dalam LiveData yang telah diobserve Activity. Proses memperoleh LiveData yang terdapat pada ApiClient yang kemudian diobserve pada Activity adalah : ApiClient (return LiveData) 🡪 Repository (return LiveData) 🡪 ViewModel (return LiveData) 🡪 Activity (observe LiveData and performs action if there any changes). Setiap proses pada request api tersebut menggunakan singleton pattern dan response yang dikembalikan dalam bentuk LiveData, sementara variabel yang diobserve merupakan Single Live Event (implements MutableLiveData). Single Live Event digunakan untuk menghindari konflik observer. Lalu, seluruh request api tersebut dilakukan di background, dengan menggunakan ScheduledExecutorService (Multi Thread). Terakhir, untuk validasi dan logic-logic lainnya, saya buat sesuai dengan petunjuk pada soal dan contoh aplikasi.

* Behavior

Behavior aplikasi saya juga telah saya sesuaikan dengan contoh aplikasi Kepo. Misalnya, ketika user enter search activity, maka keyboard dan focus pada edit text tidak langsung muncul, namun menunggu action dari user (melakukan click pada edit text). Contoh berikutnya, setelah user meng-click search button, maka edit text akan kembali kosong, dan text result akan muncul. Dan contoh-contoh lainnya.

**Obstacles During Development :**

Saya pribadi tidak menemukan kesulitan yang cukup berarti dalam mengembangkan aplikasi Kepo tersebut. Walaupun demikian, menurut saya untuk menerapkan best practices pada aplikasi yang sedang kita kembangkan bukan merupakan hal mudah. Hal tersebut tidak terlepas dari fakta bahwa untuk menerapkan best practices, dibutuhkan pengalaman yang cukup. Namun, untuk saya justru hal tersebut merupakan sebuah tantangan yang harus saya lalui, karena itu saya tidak menganggapnya sebagai sebuah kesulitan. Justru melalui trial and error, saya banyak belajar dan dapat melatih kemampuan saya.

**Tech & Libraries :**

Berikut daftar Tech & Libraries yang saya gunakan dalam mengembangkan aplikasi Kepo :

* Layouting (XML)
* Networking (Retrofit) 🡪 karena sudah menggunakan Retrofit, maka saya tidak menggunakan Volley
* RecyclerView
* Shared Preference
* Data Binding
* MVVM Architecture
* DiffUtil

**Device :**

Device yang saya gunakan untuk melakukan testing adalah :

* Oppo F1 Plus, android 5.1, 1080x1920px
* Samung Galaxy J7 Prime, android 8.1, 1080x1920px

**Disclaimer :**

Tidak semua screen dapat menghandle orientation change.