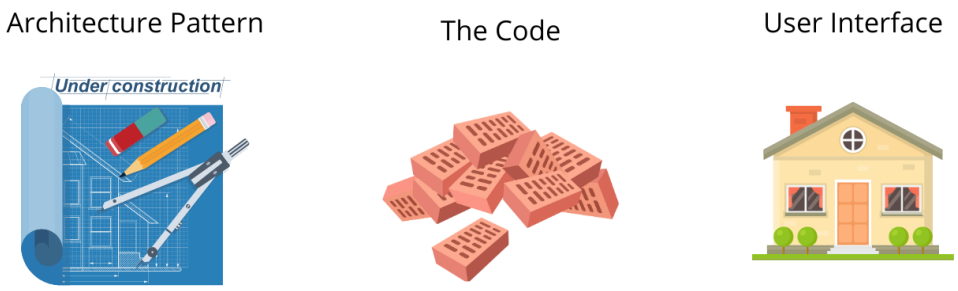
MODUL 5

Room Database Bagian 2 (Update, Delete)

Architecture Pattern merupakan sebuah penjelasan abstrak tentang gimana struktur aplikasi akan dibuat. Kalau kita menggunakan analogi pembangunan sebuah rumah, bayangkan Architecture Pattern adalah cetak biru dari desain rumah yang ingin dibuat, segala macam detail dari rumah yang ingin dibuat seperti struktur, jumlah ruangan, dan lain-lain dibuat di dalam Architecture Pattern. Kalau kita mengambil referensi dari Wikipedia Bahasa Inggris, Architecture Pattern merupakan solusi umum yang bisa dipakai terkait masalah pada arsitektur perangkat lunak. Fungsi dari Architecture Pattern adalah agar kode yang dibuat lebih konsisten karena memiliki pola yang jelas sejak awal, cukup fleksibel untuk pengembangan selanjutnya, dan masih banyak lagi.

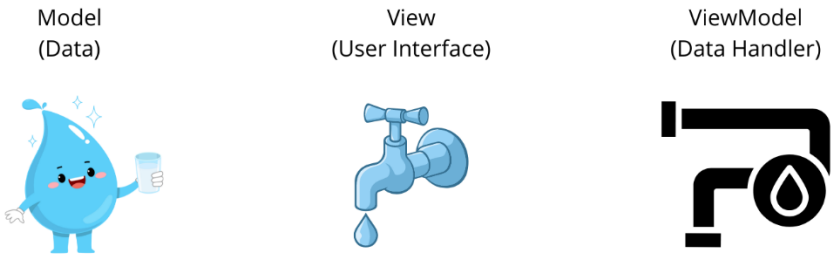


Gambar 1. Architecture Pattern dalam Pembangunan Rumah

Salah satu jenis Architecture Pattern yang cukup terkenal adalah MVVM (Model-View-ViewModel). Kalau kita menggunakan analogi pipa air untuk menjelaskan MVVM, Model diilustrasikan sebagai air.

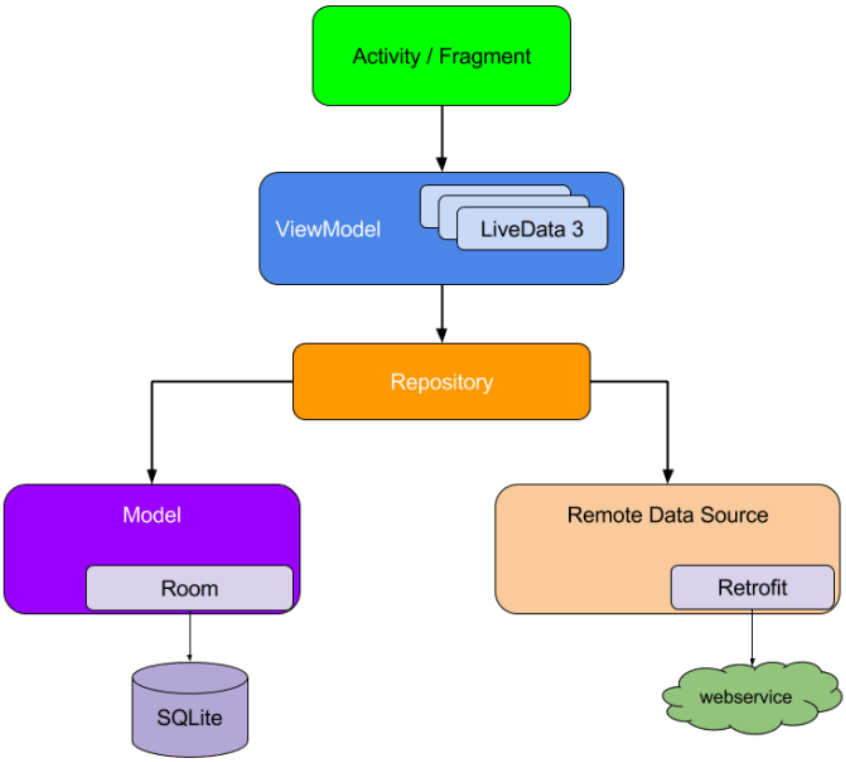
Untuk dapat mengalirkan alir ke kamar mandi sehingga bisa digunakan oleh manusia, kita membutuhkan pipa yang menyalurkan air dari tandon (wadah air) menuju ke kamar mandi, nah pipa tersebut kita sebut sebagai ViewModel.

Lalu agar air dapat digunakan oleh manusia, kita membutuhkan keran yang berfungsi untuk mengalirkan air ke kamar mandi ketika dibutuhkan, nah keran tersebut kita sebut sebagai View (User Interface).



Gambar 2. Analogi MVVM dalam kasus pipa air

Kalo tadi analoginya, lalu gimana penerapannya dalam kasus pembuatan aplikasi? Jadi, konsep MVVM menawarkan Repository sebagai layar abstraksi antara sumber data yang berbeda yang terdapat di aplikasi, lalu ViewModel muncul sebagai jembatan antara Repository dan View (Activity/Fragment), dan View (Activity/Fragment) digunakan untuk menampilkan data yang berasal dari ViewModel kepada user, dan juga bertanggung jawab terhadap aksi yang dilakukan user, lalu aksi tersebut dikirimkan ke ViewModel, yang kemudian akan diteruskan ke Repository sebelum diproses di dalam aplikasi.



Gambar 3. Arsitektur MVVM dalam Aplikasi Android

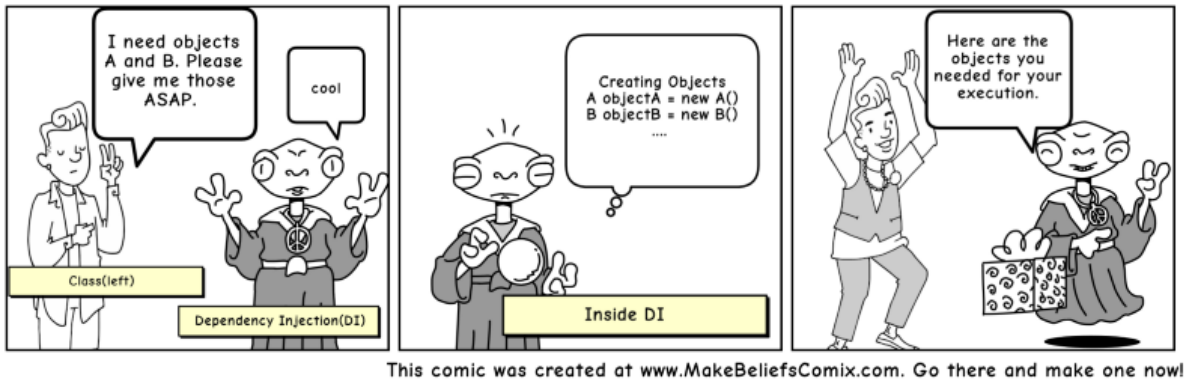
Dependency Injection

Sederhananya, Dependency Injection merupakan sebuah teknik untuk mengatur cara bagaimana suatu objek dibentuk ketika terdapat objek lain yang membutuhkan. Jadi sebelum masuk ke dependency injections, pertama-tama mari kita pahami apa yang dimaksud dengan dependency dalam pemrograman.

Bayangkan kita sedang membuat mobil. Mobil memerlukan mesin untuk berjalan. Tanpa Dependency Injection, mobil harus membuat mesinnya sendiri. Ini tidak ideal karena mobil seharusnya hanya perlu tahu cara menggunakannya, bukan cara membuat mesin.

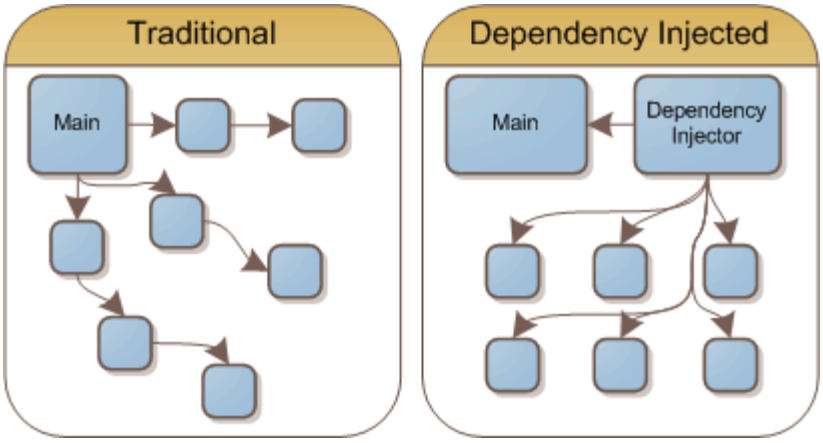
Dengan Dependency Injection, mesin dibuat di tempat lain dan kemudian diberikan (disuntikkan) ke mobil. Sekarang mobil hanya perlu tahu bahwa ia memiliki mesin dan cara menggunakannya. Ini memisahkan mobil dan mesin sehingga kita dapat mengganti mesin tanpa mempengaruhi mobil, atau mengganti mobil tanpa mempengaruhi mesin.

Dalam pemrograman, "mobil" dan "mesin" adalah potongan kode yang berbeda. Dengan Dependency Injection, setiap bagian kode hanya perlu tahu apa yang dibutuhkannya dari bagian kode lain, bukan bagaimana bagian lain tersebut diimplementasikan. Ini membuat sistem lebih fleksibel dan lebih mudah untuk dipelihara dan diuji.

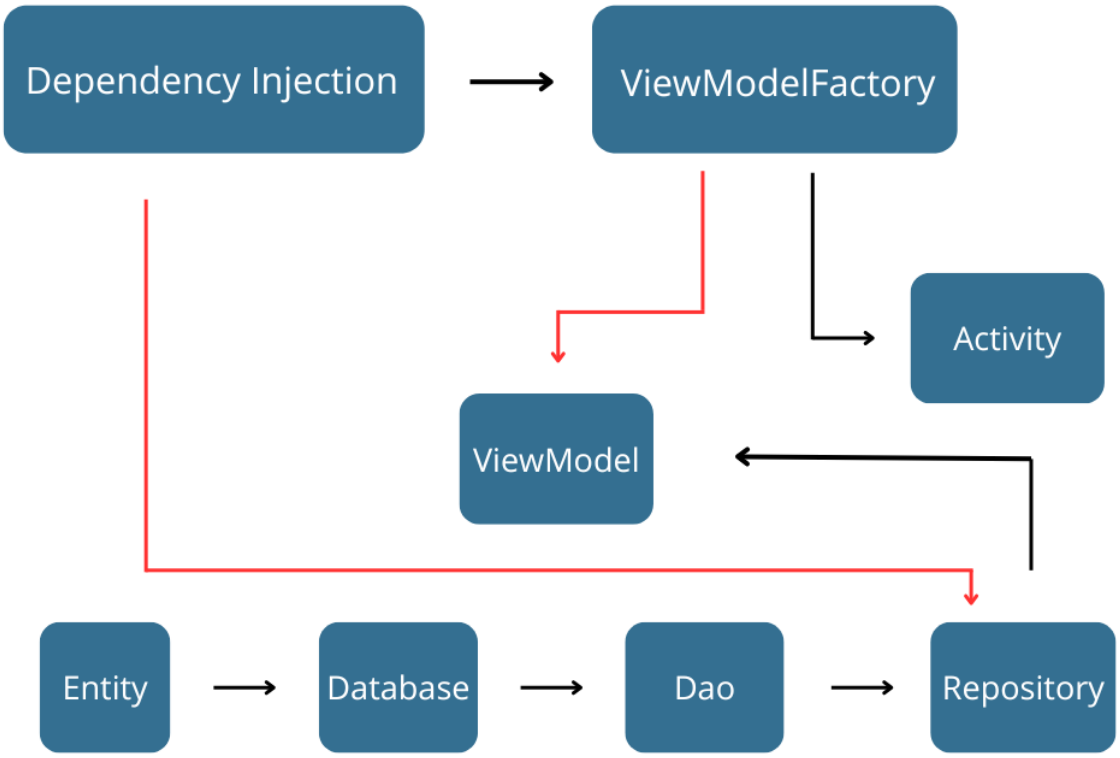


Gambar 4. Ilustrasi Dependency Injection

Terus kalo di atas itu perumpamaan Dependency Injection, di real projectnya cara kerja dependency injection tu kek gmn si? nah silahkan lihat dan pahami 2 gambar di bawah ini.



Gambar 5. Perbandingan Metode Tradisional dan Dependency Injection

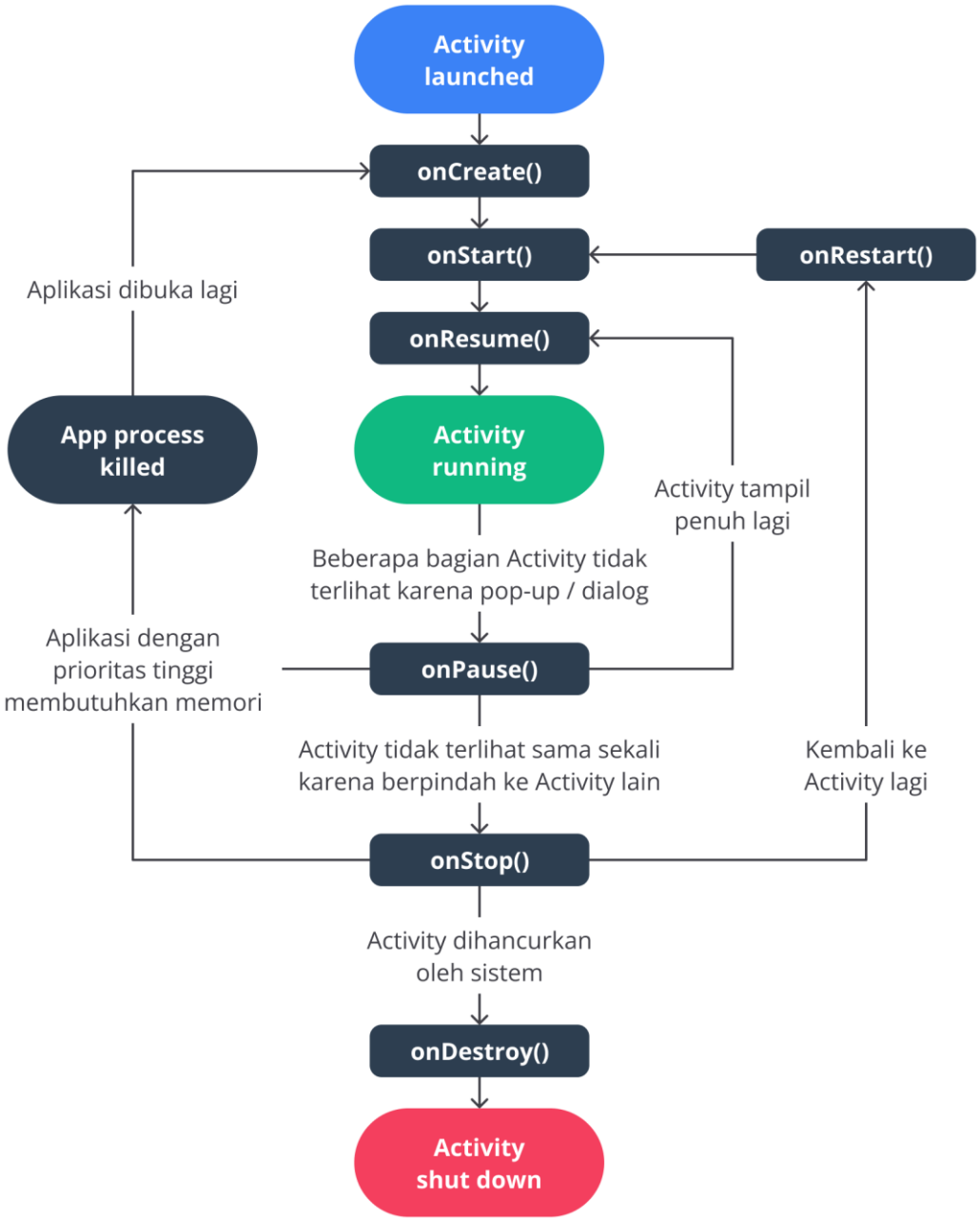


Gambar 6. Diagram Alur Kode Project Modul 4 dan 5

Activity Lifecycle

Activity lifecycle adalah serangkaian state (status) yang dapat dimiliki oleh sebuah activity selama digunakan, mulai dari saat pertama kali dibuat hingga dimusnahkan.

1. onCreate = Ini adalah saat ketika activity pertama kali dibuat. Ini seperti seorang aktor yang baru saja naik panggung, siap untuk tampil. Dalam kode, tahap ini direpresentasikan oleh metode onCreate().
2. onStart = Activity menjadi terlihat oleh pengguna, tetapi mungkin belum berada di latar depan dan interaktif. Ini seperti seorang aktor yang berada di atas panggung, tetapi sorotan belum mengenai mereka. Dalam kode, tahap ini direpresentasikan oleh metode onStart().
3. onResume = Activity berada di latar depan dan pengguna dapat berinteraksi dengannya. Ini seperti seorang aktor yang berada dalam sorotan dan sedang memainkan perannya. Dalam kode, tahap ini direpresentasikan oleh metode onResume().
4. onPause = Activity sebagian tertutup oleh activity lain - activity lain, contohnya seperti tertutup karena pop up/dialog. Activity yang dijeda tidak menerima inputan dari pengguna dan tidak dapat menjalankan kode apa pun. Dalam kode, tahap ini direpresentasikan oleh metode onPause().
5. onStop = Activity sepenuhnya tersembunyi dan tidak terlihat oleh pengguna sama sekali, seperti seorang aktor yang telah meninggalkan panggung sepenuhnya. Dalam kode, tahap ini direpresentasikan oleh metode onStop().
6. onRestart = Activity yang sedang berhenti dipanggil kembali untuk kembali aktif. Hal ini biasanya terjadi ketika pengguna menavigasi kembali ke aktivitas tersebut setelah disembunyikan di balik activity lain. onRestart sama seperti seorang actor yang sudah meninggalkan panggung namun dipanggil kembali untuk tampil. Sebelum mereka bisa naik ke panggung dan mulai tampil (activity dimulai), mereka perlu mempersiapkan diri, mungkin menyegarkan kembali riasan mereka atau melatih dialog mereka. Fase persiapan inilah yang diwakili oleh onRestart().
7. onDestroy = Activity selesai berjalan dan dihapus dari memori. Ini seperti seorang aktor yang telah menyelesaikan perannya dan meninggalkan pertunjukan. Dalam kode, tahap ini direpresentasikan oleh metode onDestroy().



Gambar 7. Activity Lifecycle

Update

Komponen Update digunakan untuk memperbarui data yang terdapat di dalam database. Seperti Insert, proses memperbarui data yang terjadi di dalam Update juga dilakukan secara otomatis, sehingga kita cukup menuliskan data apa yang akan diperbarui di dalam parameter, nantinya kotlin akan membuat query secara otomatis untuk memperbarui data berdasarkan data yang kita tulis di dalam parameter.

Delete

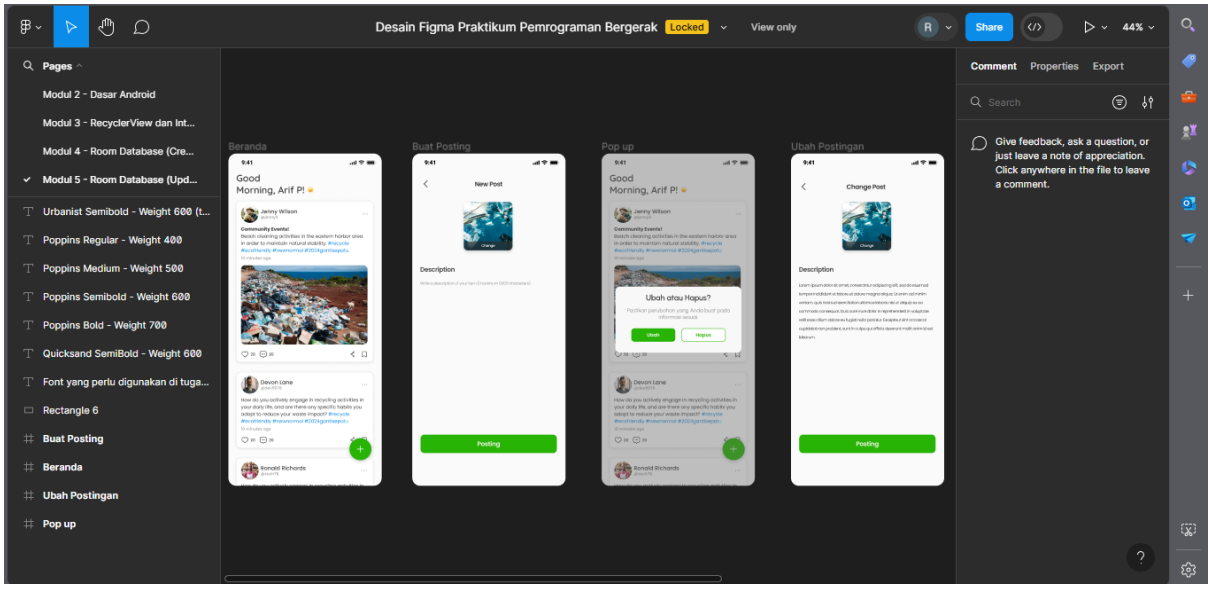
Komponen Delete digunakan untuk menghapus data yang terdapat di dalam database. Seperti Delete, proses menghapus data yang terjadi di dalam Delete juga dilakukan secara otomatis, sehingga kita cukup menuliskan data apa yang akan kita hapus di dalam parameter, nantinya kotlin akan membuat query untuk secara otomatis untuk menghapus data berdasarkan data yang kita tulis di dalam parameter.

Tugas Pendahuluan

1. Jelaskan Architecture Pattern menggunakan Bahasa kalian sendiri? (plis yang ini kalian bkin sendiri, jangan pake ChatGPT, jangan ditulis di soal ya keterangan yang ini)
2. Kenapa kita harus menggunakan Dependency Injection?
3. Andi sedang asyik bermain di ponselnya dan membuka aplikasi Whatsapp untuk mengundang temannya bermain Mobile Legends. Temannya menyetujui undangan tersebut dan mereka berdua pun mulai masuk ke dalam aplikasi Mobile Legends. Andi menunggu di lobby game cukup lama karena temannya belum juga online. Karena itu, Andi kembali mengirim pesan kepada temannya melalui Whatsapp agar segera masuk ke Mobile Legends. Namun, Andi merasa bingung karena aplikasi Whatsapp yang sebelumnya sudah dia buka, kini malah memulai dari awal seolah-olah aplikasi tersebut baru dibuka, padahal sebelumnya dia sedang berbincang dengan temennya melalui chat. Mengapa hal ini bisa terjadi? Dan jelaskan proses hidup (lifecycle) aplikasi Whatsapp dari mulai dibuka hingga ditutup (saat terjadi force closed atau aplikasi memulai dari awal)?

Tugas Praktikum

Lanjutkan tugas sebelumnya di modul 4, lalu buatlah Pop Up untuk Ubah dan Delete Postingan di Modul 5 (cek link berikut ini untuk melihat desainnya)!



Ketentuan dalam pembuatan aplikasi:

1. Menerapkan desain yang tersedia menjadi sebuah aplikasi
2. Menerapkan Room Database dalam pembuatan aplikasi
3. Menerapkan RecyclerView untuk menampilkan data yang tersimpan di dalam Room Database
4. Menerapkan Update dan Delete data dalam program

Alur Aplikasi:

1. Ketika user pertama kali membuka aplikasi, maka akan diarahkan ke Halaman Beranda
2. Halaman berada berisi konten yang berasal dari Room Database yang sudah ditambahkan
3. Ketika titik tiga yang terdapat di dalam konten diklik, maka akan menampilkan Pop Up untuk mengubah atau menghapus item
4. Ketika user memilih tombol hapus, maka konten akan terhapus dan hilang dari beranda
5. Ketika user memilih tombol ubah, maka user akan diarahkan ke Ubah Postingan
6. Ketika user melakukan klik posting diubah postingan, maka konten tersebut akan berubah sesuai dengan data yang dimasukkan oleh user.
7. Ketika item love diklik, maka akan menambah jumlah like dari konten yang diklik
8. User bisa menambahkan konten dengan cara klik Floating Button, lalu masukkan konten yang ingin ditambahkan
9. User bisa menambahkan gambar dari HP mereka
10. Ketika tombol posting diklik, maka akan kembali ke Beranda namun dengan konten baru yang berhasil ditambahkan
11. Ketika tombol back diklik di New Post, maka akan mengembalikan user ke Halaman Beranda

Tips:

1. Buat pop up engga perlu bikin dari awal, bisa pake kode yang udah ada.

***By the way, here are some memes about doing assignments that (maybe) make you smile when many assignments are currently being done (actually, I doubt that you are doing the assignment, but good luck with that).**


TSUNAMI
TUGAS