Pengembangam Aplikasi Mobile

App Architecture

Nama : Angga Gustian

NIM : 1227050022

Kelas : IF-A

1. Android App Architecture

- **Pentingnya Arsitektur Aplikasi**: Arsitektur aplikasi yang baik sangat penting agar aplikasi dapat di-maintain dengan mudah, scalable, dan efisien. Arsitektur yang buruk menyebabkan "technical debt" yang menambah beban perbaikan di masa depan.

- **Separation of Concerns (Pemilahan Tanggung Jawab)**: Memisahkan tugas UI dari logika bisnis dengan menggunakan komponen arsitektur seperti ViewModel dan LiveData untuk mempermudah kolaborasi dan pengujian kode.

1. View Model

- **Peran ViewModel**: ViewModel bertanggung jawab menyiapkan data untuk UI. Ini memungkinkan data bertahan selama perubahan konfigurasi (misalnya rotasi layar), karena ViewModel hidup selama lifecycle aktivitas atau fragmen masih aktif.

- **Penggunaan dalam Aplikasi**: ViewModel tidak boleh memiliki referensi langsung ke aktivitas atau UI untuk menjaga pemisahan tanggung jawab. Ia digunakan untuk memisahkan logika bisnis dari UI.

1. Data Binding

- **Koneksi langsung antara UI dan ViewModel**: Data binding mempermudah komunikasi antara ViewModel dan tampilan (views) tanpa perantara seperti aktivitas atau fragmen. Ini mengurangi jumlah kode boilerplate dan membuat kode lebih bersih.

- **Deklarasi di XML**: ViewModel dapat dideklarasikan dalam file XML dan dihubungkan langsung dengan elemen UI seperti TextView dan Button.

1. Live Data

- **Observer Pattern**: LiveData adalah data holder yang peka terhadap lifecycle dan dapat diamati oleh UI controller (seperti aktivitas atau fragmen). Ketika data di dalam LiveData berubah, semua observer yang terkait secara otomatis akan diberi tahu dan UI akan diperbarui.

- **MutableLiveData vs LiveData**: MutableLiveData memungkinkan perubahan data dari dalam ViewModel, sedangkan LiveData bersifat read-only bagi observer.

1. Transform Live Data

**Penggunaan Transformasi**: LiveData dapat dimodifikasi menggunakan metode map() atau switchMap() dari kelas Transformations. Transformasi ini memungkinkan perubahan data atau format sebelum dikirim ke observer tanpa mengubah LiveData sumber.

1. Summary

**Kesimpulan**: Dengan mengikuti prinsip arsitektur yang baik, memanfaatkan ViewModel, LiveData, dan data binding, pengembang dapat membuat aplikasi yang lebih efisien, responsif, dan mudah di-maintain. Ini mengurangi kode boilerplate dan membuat UI lebih dinamis dengan data yang selalu up-to-date tanpa kode manual tambahan.