Nama : M.Putra Fauzan Fatah  
NIM : 1227050075  
Kelas : B

**Rangkuman materi tentang App ArchitectureAssignment**

1. Android App Architecture:

* Arsitektur aplikasi Android yang baik membantu menghindari "short-term hacks" dan memudahkan pengembangan jangka panjang.
* Prinsip "separation of concerns" diterapkan untuk memisahkan logika bisnis dari UI.
* Android Jetpack menyediakan komponen arsitektur yang membantu merancang aplikasi  
  yang kuat, dapat diuji, dan mudah dipelihara.

2. View Model :

* ViewModel adalah komponen yang menyiapkan data untuk UI.
* Tidak boleh mereferensikan activity, fragment, atau view dalam hirarki view.
* Bertahan selama siklus hidup activity atau fragment, bahkan saat terjadi perubahan konfigurasi.
* Memungkinkan data bertahan saat terjadi perubahan konfigurasi seperti rotasi layar.

3. Data Binding:

* Menghubungkan UI components dalam layout XML langsung ke sumber data.
* Mengurangi kode boilerplate yang biasanya diperlukan untuk menghubungkan data ke UI.
* Bekerja dengan baik bersama ViewModel untuk membuat UI yang responsif.

4. Live Data :

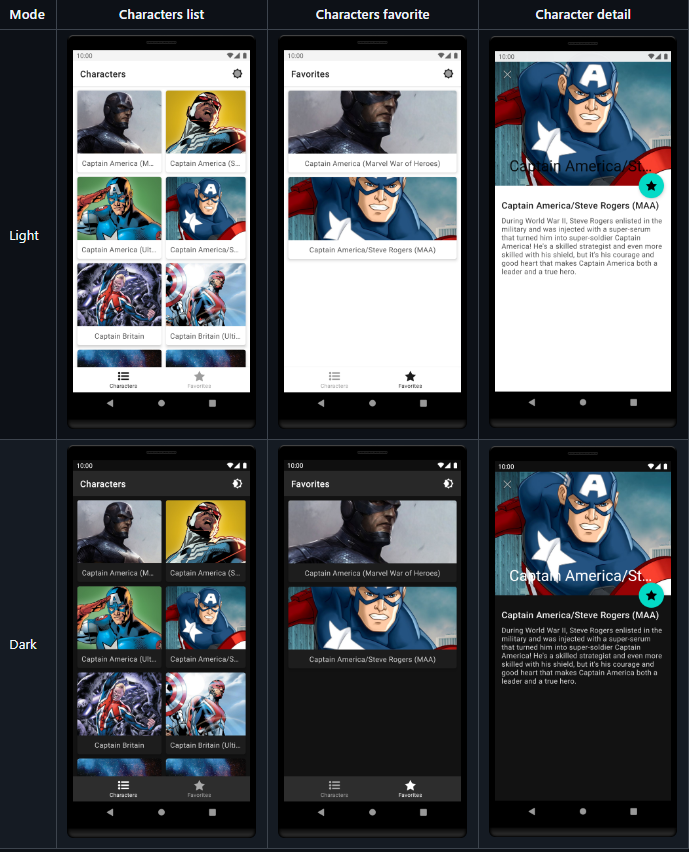
* Kelas pemegang data yang sadar siklus hidup dan dapat diobservasi.
* Sering digunakan oleh ViewModel untuk menyimpan data.
* Observer (biasanya activity atau fragment) dapat ditambahkan atau dihapus.
* Memastikan UI selalu sesuai dengan data terbaru.

5. Transform Live Data

* Memungkinkan manipulasi LiveData menjadi objek LiveData baru.
* Menggunakan metode seperti map() dan switchMap() untuk mentransformasi data.
* Berguna untuk mengubah atau memfilter data sebelum ditampilkan di UI.

**Contoh Program**

Program dibawah saya mengambil contoh dari github https://github.com/vmadalin/android-modular-architecture



**Gambaran Umum Proyek**

Tujuan

Proyek ini berfungsi sebagai aplikasi contoh yang menekankan pada:

* Modularitas: Membagi aplikasi menjadi unit-unit fungsional yang terpisah.
* Skalabilitas: Memudahkan pertumbuhan dan adaptasi seiring waktu.
* Pemeliharaan: Memastikan aplikasi dapat dengan mudah diperbarui dan dimodifikasi.
* Pengujian: Memfasilitasi proses pengujian yang efisien.

Fitur Utama

* Menggunakan arsitektur single-activity dengan komponen Navigasi untuk mengelola fragmen.
* Menerapkan pola desain MVVM (Model-View-ViewModel) untuk memisahkan pengembangan UI dari logika bisnis.
* Mematuhi prinsip S.O.L.I.D, yang mendorong desain perangkat lunak yang dapat dipahami dan fleksibel.
* Mendukung dark theme, meningkatkan pengalaman pengguna dengan mengurangi penggunaan daya dan meningkatkan visibilitas dalam kondisi pencahayaan rendah.

Aspek Teknis

Persiapan Lingkungan

Untuk membangun aplikasi ini, pengguna perlu:

* Android Studio versi 4.1.0 atau lebih baru.
* Kunci API Marvel yang dikonfigurasi dalam file local.properties.

Alat Kualitas Kode

Proyek ini mengintegrasikan beberapa alat analisis statis:

| **Alat** | **File Konfigurasi** | **Perintah Cek** | **Perintah Perbaikan** |
| --- | --- | --- | --- |
| Detekt | /.detekt | ./gradlew detekt | - |
| Ktlint | - | ./gradlew ktlint | ./gradlew ktlintFormat |
| Spotless | /.spotless | ./gradlew spotlessCheck | ./gradlew spotlessApply |
| Lint | /.lint | ./gradlew lint | - |

Alat-alat ini diintegrasikan ke dalam hook pre-commit untuk memastikan kualitas kode sebelum perubahan dikomit.

Komponen Arsitektur

Aplikasi ini memanfaatkan berbagai komponen arsitektur Android, termasuk:

* **ViewModel**: Mengelola data terkait UI secara sadar terhadap siklus hidup.
* **LiveData**: Mengamati perubahan data sambil menghormati siklus hidup komponen.
* **Room**: Menyediakan lapisan abstraksi di atas SQLite untuk akses database yang kuat.

Struktur Modular

Proyek ini dibagi menjadi beberapa modul:

1. **Modul Aplikasi (:app)**: Modul utama aplikasi yang bertanggung jawab untuk membuat bundel aplikasi.
2. **Modul Inti (:core)**: Menangani permintaan jaringan dan akses database.
3. **Modul Fitur (:features)**: Modul dinamis yang dapat diunduh secara independen.
4. **Modul Umum (:commons)**: Mengandung sumber daya bersama di antara modul fitur.
5. **Modul Perpustakaan (:libraries)**: Menampung kode utilitas yang digunakan di berbagai modul.