

Study Independent

Android Development (Day 7)

Ringkasan

Android Architecture Component

- a. Data Binding
- b. Lifecycles
- c. ViewModel
- d. LiveData
- e. Navigation
- f. Paging
- g. WorkManager
- h. Room

Pembahasan

Data Binding

Library Data Binding adalah support library yang memungkinkan untuk mengikat komponen UI dalam tata letak ke sumber data di aplikasi menggunakan format deklaratif XML. Data Binding memungkinkan untuk berkomunikasi dengan mudah di seluruh tampilan dan sumber data. Pola ini penting untuk banyak desain Android, termasuk model view ViewModel (MVVM), yang saat ini merupakan salah satu pola arsitektur Android yang paling umum. Android Studio mendukung banyak fitur pengeditan untuk kode data binding. Misalnya, Android Studio mendukung fitur berikut untuk ekspresi data binding:

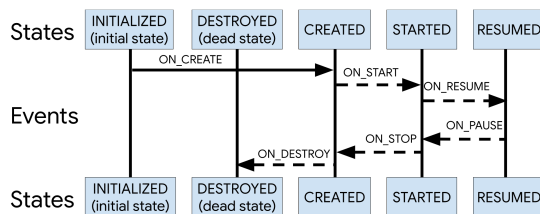
- a. Penyorotan sintaks
- b. Penandaan error sintaks bahasa ekspresi
- c. Pelengkapan kode XML

Lifecycles

Lifecycle adalah class yang menyimpan informasi tentang status siklus proses suatu komponen (seperti aktivitas atau fragmen) dan memungkinkan objek lain untuk mengamati status ini.

LifecycleOwner adalah antarmuka metode tunggal yang menunjukkan bahwa class memiliki Lifecycle, seperti Activity dan Fragment. Lifecycle menggunakan dua enumerasi utama untuk melacak status siklus proses komponen retribusinya:

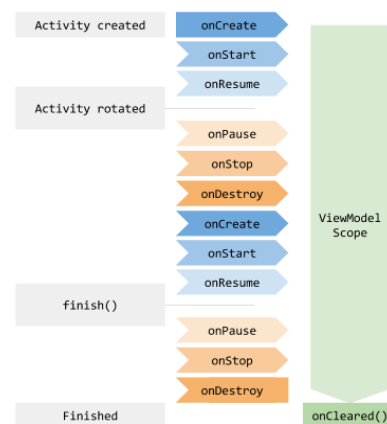
- Event : Peristiwa siklus proses yang dikirim dari framework dan class Lifecycle. Peristiwa ini memetakan ke peristiwa callback dalam aktivitas dan fragment.
- State : Status saat ini dari komponen yang dilacak oleh objek Lifecycle.
- Gambar Diagram Proses



ViewModel

Framework Android mengelola siklus proses pengontrol UI, seperti aktivitas dan fragment. Class ViewModel didesain untuk menyimpan dan mengelola data terkait UI dengan cara yang berbasis siklus proses. ViewModel otomatis disimpan pada saat perubahan konfigurasi sehingga data yang disimpan segera tersedia untuk instance aktivitas atau fragmen berikutnya.

- Siklus Hidup ViewModel
 - Activity Created -> onCreate, onStart, onResume
 - Activity Rotated -> onPause, onStop, onDestroy <-> onCreate, onStart, onResume
 - finish() -> onPause, onStop, onDestroy
 - Finished



- b. Kemudahan yang diberikan ViewModel
 - Berbagi data antar fragment
 - Mengganti Loader dengan ViewModel
 - Menggunakan coroutine dengan ViewModel

LiveData

LiveData adalah class penyimpanan data observable. Tidak seperti observable pada umumnya, LiveData berbasis siklus proses, yang berarti observable ini mengikuti siklus proses komponen aplikasi lainnya, seperti aktivitas, fragment, atau layanan. Basis ini memastikan LiveData hanya mengupdate pengamat komponen aplikasi yang dalam status siklus proses aktif. Berikut ini beberapa keuntungan menggunakan LiveData:

- a. Memastikan UI cocok dengan status data
- b. Tidak ada kebocoran memori
- c. Penghentian proses tidak menyebabkan error
- d. Tidak ada lagi penanganan siklus proses manual
- e. Data selalu diperbarui
- f. Perubahan konfigurasi yang tepat
- g. Berbagi resource

Navigation

Navigasi adalah interaksi yang memungkinkan pengguna melihat-lihat, masuk, dan keluar dari berbagai konten dalam aplikasi. Komponen Navigasi Android Jetpack membantu untuk menerapkan navigasi.

- a. Komponen Navigasi terdiri dari tiga bagian utama yang dijelaskan di bawah:
 - Grafik navigasi: Sumber daya XML yang berisi semua informasi terkait navigasi.
 - NavHost: Komponen Navigasi berisi implementasi NavHost default, NavHostFragment, yang menampilkan tujuan fragmen.
 - NavController: Objek yang mengelola navigasi aplikasi dalam sebuah NavHost.
- b. Komponen Navigasi memberikan sejumlah manfaat lain, termasuk yang berikut:
 - Menangani transaksi fragmen.
 - Menangani aksi Up dan Back dengan benar secara default.
 - Menyediakan standar resources untuk animasi dan transisi.
 - Menerapkan dan menangani deep link.

- Termasuk pola Navigasi UI, seperti navigation drawer dan bottom navigation.
- Safe Args - plugin Gradle yang memberikan keamanan jenis saat menavigasi dan mengirimkan data antar tujuan.
- Mendukung ViewModel - untuk dapat mengatur ViewModel ke grafik navigasi.

Paging

Library Paging membantu memuat dan menampilkan data sedikit demi sedikit. Memuat bagian data berdasarkan permintaan dapat mengurangi penggunaan bandwidth jaringan dan resource sistem.

- Arsitektur library
 - PagedList
 - Data
 - UI
- Mendukung arsitektur data yang berbeda
 - Network Only : ditayangkan hanya dari server backend.
 - Database Only: disimpan hanya di database dalam perangkat.
 - Network & Database: dikombinasikan dengan sumber lain, menggunakan database dalam perangkat sebagai cache.

WorkManager

WorkManager adalah solusi yang direkomendasikan untuk pekerjaan persisten. Pekerjaan bersifat persisten saat pekerjaan tersebut tetap dijadwalkan walaupun aplikasi dimulai ulang dan di-reboot. Karena sebagian besar pemrosesan latar belakang idealnya dilakukan melalui pekerjaan persisten, WorkManager adalah API utama yang direkomendasikan untuk pemrosesan latar belakang. WorkManager menangani tiga jenis pekerjaan persisten:

- Langsung: Tugas yang harus segera dimulai dan diselesaikan. Mungkin diprioritaskan.
- Berjalan Lama: Tugas yang mungkin berjalan lebih lama, kemungkinan lebih dari 10 menit.
- Dapat ditangguhkan: Tugas terjadwal yang dimulai di lain waktu dan dapat dijalankan secara berkala.

Room

Library persistensi Room menyediakan layer abstraksi atas SQLite untuk memungkinkan akses database yang lancar sambil memanfaatkan kemampuan penuh SQLite.

- a. Room memberikan manfaat berikut
 - Verifikasi waktu kompilasi kueri SQL.
 - Anotasi praktis yang meminimalkan kode boilerplate muncul berulang kali dan rentan error.
 - Jalur migrasi database sederhana.
- b. Terdapat tiga komponen utama dalam Room:
 - Class database yang menyimpan database dan berfungsi sebagai titik akses utama bagi koneksi saat ini ke data persisten aplikasi.
 - Entity data yang menampilkan tabel di database aplikasi.
 - Objek akses data (DAO) yang menyediakan berbagai metode agar aplikasi dapat membuat kueri, mengupdate, memasukkan, dan menghapus data dalam database.

Referensi

Berikut merupakan sumber referensi dari materi yang dibuat:

- a. <https://developer.android.com/topic/libraries/data-binding/start>
- b. <https://developer.android.com/topic/libraries/architecture/lifecycle>
- c. <https://developer.android.com/topic/libraries/architecture/viewmodel>
- d. <https://developer.android.com/topic/libraries/architecture/livedata>
- e. <https://developer.android.com/guide/navigation>
- f. <https://developer.android.com/topic/libraries/architecture/paging>
- g. <https://developer.android.com/topic/libraries/architecture/workmanager>
- h. <https://developer.android.com/training/data-storage/room>

Penilaian

Penilaian akan dilakukan dengan memberikan tugas pada minggu terakhir (Jumat).