

“Lanjutan Merancang Tampilan Android Apps”

Ikrima Amanda



Multi State View

Deskripsi

- Sebuah konsep dimana tampilan aplikasi berubah sesuai dengan berbagai kondisi atau status yang terjadi.
- Dalam pengembangan aplikasi, terdapat beberapa keadaan yang mungkin terjadi, seperti tampilan konten yang sedang dimuat (loading), tampilan failed, atau tampilan data kosong (empty).
- Berguna dalam meningkatkan pengalaman pengguna dengan memberikan umpan balik visual yang tepat ketika keadaan berubah.

Contoh Keadaan dalam Multi-State View

- **Loading state:** Ketika data sedang dimuat, multi-state view menampilkan indikator loading atau animasi yang sesuai untuk memberi tahu pengguna bahwa proses sedang berlangsung.
- **Error state:** Jika terjadi kesalahan saat memuat data atau berinteraksi dengan server, multi-state view menampilkan pesan kesalahan yang relevan dan memberikan opsi untuk mencoba kembali.
- **Empty state:** Jika tidak ada data yang tersedia atau hasil pencarian kosong, multi-state view menampilkan pesan atau icon yang sesuai untuk memberi tahu pengguna tentang kondisi tersebut.
- **Content state:** Ini adalah keadaan dimana data telah berhasil dimuat dan tampilan utama aplikasi ditampilkan kepada pengguna.

Langkah Penggunaan Multi-State View

- Tambahkan dependensi sesuai dokumentasi
- Buat default layout ketika view loading, error, dan empty
- Buat sebuah file, contoh beri nama helper
- Di dalam file helper buat function untuk mengatur tampilan ketika multi-state view digunakan
- Buka file xml dari activity/fragment, tambahkan multi-state view dan atur sesuai kebutuhan
- Buka file kotlin dari activity/fragment dan atur multi-state view sesuai kebutuhan

Shimmer Effect

Deskripsi

- Efek visual yang memberikan tampilan animasi berkedip seperti kilauan atau cahaya yang bergerak secara bergelombang pada area yang ditentukan di layar.
- Shimmer effect biasanya digunakan saat konten sedang dimuat atau ketika aplikasi sedang mengambil data dari server.
- Tujuan => memberikan umpan balik visual kepada pengguna bahwa ada sesuatu yang sedang terjadi dan konten akan segera dimuat.
- Efek ini juga membantu mengisi ruang kosong dengan animasi yang menarik dan dapat membuat pengguna merasa ada interaksi dengan aplikasi walaupun konten belum tampil di halaman aplikasi.

Langkah Penggunaan Shimmer Effect

- Tambahkan dependensi sesuai dengan dokumentasi
- Buat item untuk shimmer effect sesuai kebutuhan
- Tambahkan ShimmerFrameLayout pada file XML activity/fragment
- Konfigurasi ShimmerFrameLayout pada file activity/fragment
- Aktifkan Shimmer Effect
- Atur visibilitas shimmer effect

Shape

Deskripsi

- Merujuk pada tampilan atau bentuk visual yang diterapkan pada elemen UI, seperti tombol, latar belakang, atau kotak teks.
- Shape membantu developer untuk mengubah tampilan dan gaya elemen UI dengan berbagai bentuk, warna, dan efek.
- Shape diimplementasikan menggunakan XML Drawable di Android, yang mendefinisikan berbagai properti dan atribut untuk mengatur tampilan elemen UI.

Atribut penting dalam membuat Shape

- **shape** => menentukan jenis bentuk yang ingin diterapkan pada elemen UI. Beberapa nilai yang umum digunakan adalah "rectangle" (persegi panjang), "oval" (elips), "line" (garis), atau "ring" (lingkaran).
- **corners** => untuk mengatur sudut-sudut elemen UI. Dapat menentukan sudut-sudut yang berbeda untuk menciptakan tampilan sudut tumpul, sudut runcing, atau sudut bulat.
- **gradient** => menerapkan efek gradien warna pada shape. Titik awal dan akhir gradient dapat ditentukan, serta warna-warna yang ingin digunakan dalam gradien.
- **solid** => mengatur warna latar belakang shape.
- **stroke** => menambahkan garis tepi (stroke) pada shape. Lebar dan warna stroke dapat diatur sesuai kebutuhan.
- **size** => mengatur dimensi shape, seperti lebar dan tinggi. Dapat menggunakan nilai absolut (misalnya, "100dp") atau relatif (misalnya, "50%").

Referensi

- <https://github.com/Kennyc1012/MultiStateView>
- <https://github.com/facebook/shimmer-android>
- <https://facebook.github.io/shimmer-android/>

Tugas Akhir

Aplikasi Penyimpanan Data Buku

- Terdapat **splash screen aplikasi**
- Terdapat **shimmer** ketika proses mengambil data dari database (menggantikan progress bar)
- Terdapat **tampilan list buku** (Cover buku, Judul buku, Nama Penulis) => diambil dari **Firestore Realtime DB**
- Jika list buku masih kosong, tampilkan view yang sesuai
- Terdapat floating button untuk **menambahkan data buku yang baru** (Cover buku dapat diinputkan berupa pemilihan foto pada galeri atau input URL)
- Data yang diinputkan => Cover buku, judul, nama penulis, tahun terbit, dan kategori buku (Diinputkan ke dalam **Firestore Realtime DB**)

Tugas Akhir

Aplikasi Penyimpanan Data Buku

- Item buku pada list buku dapat diklik dan berpindah ke halaman baru dengan detail data dari buku yang diklik/dipilih sebelumnya
- **Detail data buku mencakup** => Cover buku, judul, nama penulis, tahun terbit, dan kategori (Data detail buku diambil dari Firebase Realtime DB)
- Pada halaman detail buku terdapat button untuk **mengedit data buku** dan **hapus data buku (show pop untuk konfirmasi menghapus data)**
- **Total ada 5 tampilan** => splash screen, list buku, input data buku baru, detail buku, dan edit data buku.
- Dikumpulkan dalam bentuk Link Google Drive (**Berisi link repo GitHub dari Tugas Day 6 dan APK**)
- Deadline pengumpulan tugas => **12 Juli 2023 10.00 WIB**
- **Link pengumpulan tugas** => <https://forms.gle/1W2Sku2JJH51K2Dj6>

Stuck?

talk with ur best friend...
chatgpt, google, stack over flow, medium

The Most Important Thing...

<https://www.instagram.com/reel/CtoupCfADAv/?igshid=Y2lzZGU1MTFhOQ%3D%3D>

“Practice doesn’t make perfect. Perfect practice makes perfect”

- Vince Lombardi

Terima kasih!

telegram : @ikrimaamandaa

LinkedIn :

<https://www.linkedin.com/in/ikrimaamanda/>

