

1. Activity adalah salah satu komponen inti dalam pengembangan aplikasi Android. Ini mewakili satu layar dengan antarmuka pengguna yang dapat dijalankan oleh pengguna. Setiap aplikasi Android dapat memiliki satu atau lebih Activity, dan Activity memiliki siklus hidupnya sendiri, yang mencakup berbagai metode seperti `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()`, dan lainnya.

2. Fragment adalah komponen UI yang lebih kecil yang dapat digunakan dalam Activity untuk memecah antarmuka pengguna menjadi bagian yang lebih kecil dan dapat dikelola secara independen. Fragment memiliki siklus hidup mereka sendiri dan dapat digunakan untuk membuat antarmuka yang lebih fleksibel dan modular.

3. Tahapan Siklus Hidup Activity:

- `onCreate()`: Dipanggil saat Activity pertama kali dibuat.
- `onStart()`: Dipanggil saat Activity menjadi terlihat tetapi mungkin belum dapat berinteraksi dengan pengguna.
- `onResume()`: Dipanggil saat Activity menjadi aktif dan berinteraksi dengan pengguna.
- `onPause()`: Dipanggil saat Activity kehilangan fokus tetapi masih terlihat.
- `onStop()`: Dipanggil saat Activity tidak lagi terlihat.
- `onDestroy()`: Dipanggil saat Activity dihancurkan.

4. Kita dapat menyimpan data seperti access token di Android dengan berbagai cara, salah satunya adalah menggunakan SharedPreferences, SQLite Database, atau menyimpannya secara aman menggunakan Android Keystore. SharedPreferences adalah pilihan yang umum digunakan untuk data kecil seperti access token.

5. ANR terjadi ketika aplikasi tidak merespons input pengguna selama jangka waktu tertentu (biasanya 5 detik). Kita dapat mencegah ANR dengan melakukan operasi berat seperti I/O, jaringan, atau dalam thread latar belakang atau menggunakan AsyncTask. Untuk tugas yang lebih panjang atau berat, Kita harus menggunakan thread latar belakang atau layanan.

6. Proses asynchronous adalah proses di mana tugas dijalankan di latar belakang tanpa menghalangi atau memblokir eksekusi tugas lain di thread utama. Ini umumnya digunakan untuk tugas seperti pengambilan data jaringan atau I/O yang dapat memakan waktu.

7. Main thread adalah thread yang digunakan untuk menggambar antarmuka pengguna, dan melakukan tugas berat di dalamnya dapat mengakibatkan aplikasi menjadi tidak responsif. Oleh karena itu, proses yang panjang atau berat harus dilakukan di thread latar belakang.

8. Service adalah komponen yang dapat dijalankan di latar belakang dan tidak memiliki antarmuka pengguna. Mereka digunakan untuk melakukan tugas latar belakang yang berjalan terus-menerus, seperti pemutar musik, pengunduhan file, dll. Kita dapat membuat service yang terus berjalan bahkan setelah aplikasi dipindahkan ke background dengan menggunakan metode `startForegroundService()` dan memberikan notifikasi.

9. Beberapa komponen/library yang dapat digunakan untuk melakukan proses asynchronous di Android termasuk AsyncTask, Thread, Handler, Executor, RxJava, dan Kotlin Coroutines.

10. Dynamic link adalah tautan yang dapat mengarahkan pengguna ke aplikasi Android yang tepat jika diinstal atau ke tautan web jika tidak diinstal. Ini memungkinkan pengembang untuk menyediakan integrasi yang mulus antara aplikasi dan konten web.

11. Kita dapat menjalankan proses secara periodik di Android menggunakan JobScheduler, WorkManager, AlarmManager.

12. Beberapa desain pola yang umum digunakan dalam pengembangan Android termasuk MVC (Model-View-Controller), MVP (Model-View-Presenter), MVVM (Model-View-ViewModel), Singleton, dan Repository Pattern.

13. Alur yang baik untuk mengambil foto, melakukan kompresi gambar, dan menyimpannya ke file melibatkan pengambilan foto menggunakan kamera atau galeri (misalnya dengan Intent), kemudian kompresi gambar dengan mengubah resolusi atau kualitasnya, dan akhirnya menyimpan gambar ke penyimpanan lokal atau cloud sesuai kebutuhan aplikasi Kita. Hal ini dapat melibatkan operasi di thread latar belakang untuk menghindari ANR.