* **Activity**: Activity merupakan komponen utama dalam pengembangan aplikasi Android yang digunakan untuk menampilkan antarmuka pengguna (UI) dan mengelola interaksi pengguna. Setiap layar dalam aplikasi Android direpresentasikan oleh satu Activity, mirip dengan jendela di aplikasi desktop. Activity dapat berisi tata letak UI, widget, dan logika bisnis yang diperlukan untuk melayani kebutuhan pengguna.
* **Fragment**: Fragment adalah bagian modular dari antarmuka pengguna (UI) dalam aplikasi Android yang dapat digunakan kembali. Dibandingkan dengan Activity yang merepresentasikan satu layar, Fragment memungkinkan pembangunan UI yang lebih fleksibel dan modular. Fragment dapat digunakan untuk menampilkan bagian-bagian kecil dari UI atau untuk membuat UI yang lebih kompleks yang dapat digunakan kembali di berbagai layar.
* **View**: View merupakan elemen dasar dari antarmuka pengguna dalam aplikasi Android. Ini mencakup berbagai jenis komponen seperti TextView (teks), Button (tombol), ImageView (gambar), dan lainnya. View digunakan untuk membangun tata letak UI dan menampilkan data kepada pengguna. Dalam Android Studio, komponen UI direpresentasikan oleh View yang dapat ditempatkan dan disesuaikan menggunakan Editor Layout.
* **Intent**: Intent adalah objek yang digunakan untuk memfasilitasi komunikasi antara berbagai komponen aplikasi Android, seperti Activity, Service, atau BroadcastReceiver. Intent membawa informasi tentang tindakan yang ingin dilakukan pengguna, seperti membuka Activity baru, memulai layanan (Service), atau mengirimkan siaran (broadcast). Intent digunakan untuk memulai komponen-komponen ini dan dapat membawa data tambahan (extra) bersama dengan mereka.
* **Service**: Service adalah komponen aplikasi Android yang berjalan di latar belakang dan tidak memiliki antarmuka pengguna (UI) yang terlihat. Service digunakan untuk menjalankan operasi jangka panjang atau berkelanjutan, seperti memutar musik, mengunduh data, atau melakukan sinkronisasi. Service dapat dijalankan secara independen dari Activity dan terus berjalan bahkan setelah Activity ditutup.
* **Recycler View**: Recycler View adalah komponen UI dalam Android yang digunakan untuk menampilkan daftar data yang besar secara efisien. Recycler View mendaur ulang elemen tampilan (View) yang tidak terlihat oleh pengguna untuk menghemat memori dan meningkatkan kinerja aplikasi. Ini sangat berguna untuk menampilkan daftar data yang dinamis, seperti daftar kontak, daftar artikel, atau hasil pencarian.
* **Jetpack Compose**: Jetpack Compose adalah toolkit UI modern untuk membangun antarmuka pengguna (UI) aplikasi Android. Menggunakan paradigma pemrograman deklaratif, Jetpack Compose memungkinkan pengembang untuk membuat UI dengan lebih mudah, cepat, dan responsif. Ini merupakan alternatif yang inovatif dan efisien untuk pembuatan UI dibandingkan dengan pendekatan tradisional menggunakan XML layout dan View.

1. Apa perbedaan fragment dan Activity? Dan ketika apa fragment itu dibutuhkan?
2. **Activity**:
   * Activity adalah komponen utama dalam aplikasi Android yang merepresentasikan satu layar atau jendela dalam aplikasi.
   * Setiap layar atau bagian utama dari UI dalam aplikasi Android biasanya diimplementasikan sebagai Activity.
   * Activity memiliki siklus hidup (lifecycle) sendiri yang berhubungan dengan aktivitas pengguna, seperti pembuatan (creation), memulai (starting), menghentikan (stopping), dan menghancurkan (destroying) Activity.
   * Activity dapat berdiri sendiri atau memulai Activity lain, dan dapat menerima respons dari Activity lain melalui metode onActivityResult().
3. **Fragment**:
   * Fragment adalah bagian modular dari antarmuka pengguna (UI) dalam aplikasi Android yang dapat digunakan kembali.
   * Fragment biasanya digunakan untuk memecah layar Activity menjadi bagian-bagian yang lebih kecil dan mandiri, sehingga memungkinkan pembangunan UI yang lebih fleksibel dan modular.
   * Fragment memiliki siklus hidup sendiri yang terkait dengan siklus hidup Activity tempat ia ditempatkan. Fragment dapat ditempatkan, ditambahkan, dihapus, dan diganti di dalam Activity saat runtime.
   * Fragment dapat berinteraksi satu sama lain dan dengan Activity induknya. Mereka juga dapat berbagi data dengan Activity atau Fragment lain dalam hierarki UI.

Fragment biasanya dibutuhkan dalam beberapa situasi, termasuk:

* **Pembangunan UI yang fleksibel**: Ketika Anda ingin membagi layar Activity menjadi bagian-bagian yang dapat digunakan kembali atau mengganti tampilan tanpa mengganti seluruh layar, Fragment sangat berguna. Misalnya, jika Anda memiliki layar Activity yang memiliki beberapa bagian yang dapat diakses melalui tab atau navigasi drawer, Fragment dapat digunakan untuk menampilkan konten dari setiap tab atau item navigasi.
* **Penanganan konfigurasi yang berubah**: Fragment berguna dalam menangani perubahan konfigurasi, seperti rotasi perangkat, di mana Activity dapat dihancurkan dan dibuat kembali. Dalam kasus ini, Fragment dapat tetap bertahan dan melekat pada Activity, sehingga menghindari kehilangan data atau status UI.
* **Meningkatkan modularitas dan fleksibilitas**: Fragment memungkinkan pembangunan UI yang lebih modular, di mana setiap bagian dapat dikembangkan, diuji, dan dipelihara secara terpisah. Ini dapat mempercepat pengembangan aplikasi dan memungkinkan penggunaan kembali komponen UI di berbagai bagian aplikasi.

1. Apa yang kalian ketahui tentang Main Thread pada Android?

Pada Android, **Main Thread** atau **UI Thread** adalah thread khusus yang bertanggung jawab untuk:

* **Menangani update UI (User Interface):** Ini termasuk hal-hal seperti menggambar elemen di layar, memperbarui teks, dan menangani interaksi pengguna seperti klik dan sentuhan.
* **Menjalankan event loop:** Main Thread terus-menerus memproses event yang diterima dari sistem Android, seperti event sentuhan, event keyboard, dan event lainnya.

Main Thread sangat penting karena ini adalah satu-satunya thread yang dapat langsung memperbarui UI aplikasi. Menjalankan tugas berat atau memakan waktu lama di Main Thread dapat **memblokir UI**, yang berarti aplikasi menjadi tidak responsif dan tampak macet bagi pengguna.

1. Bottom of Form