

Proses latar belakang android

Background process android

Glosarium

A.

Asynchronous: Merupakan proses yang dijalankan tidak berurutan

ANR (*application not responding*): keadaan dimana sistem android mengeluarkan aplikasi dari stack karena aplikasi tidak melakukan respon selama 5 detik

B.

background process: proses eksekusi yang dilakukan dibalik layar pengguna

C.

Coroutine: metode untuk menangani proses latar belakang pada bahasa pemrograman kotlin

E.

Executors: komponen pada android yang digunakan untuk menangani beberapa thread dalam satu waktu

H.

Handler : komponen pada android yang digunakan untuk berkomunikasi antara ui thread dengan background thread

Secara sederhana, proses latar belakang adalah proses yang berjalan dibalik layar pengguna atau istilah lebih umumnya adalah *background process*, proses latar belakang akan berjalan secara *asynchronous* yang berarti proses yang dijalankan akan tidak berurutan. Contoh dari proses latar belakang pada android adalah ketika pengguna melakukan koneksi dengan server (dalam hal ini bisa mengunduh atau mengunggah sebuah file), proses diatas perlu dilakukan proses latar belakang dengan tujuan untuk mengizinkan pengguna berinteraksi dengan aplikasi sambil menunggu proses latar belakang selesai dilaksanakan.

Namun tanpa penanganan yang tepat, hal ini justru bisa menyebabkan *lagging* atau jeda ketika proses sedang dikerjakan, hal ini akan menimbulkan pengalaman pengguna yang buruk, selain itu, sistem android secara otomatis akan menghitung waktu jeda aplikasi, jika lebih dari 5 detik, maka sistem android akan memunculkan dialog ANR atau *application non responding* dan aplikasi akan dikeluarkan dari *stack*, jika hal ini terjadi pada aplikasi yang telah di rilis kepada pengguna, besar kemungkinan mereka akan melakukan pencopotan instalasi aplikasi, dan ini merupakan dampak paling masif ketika proses latar belakang tidak ditangani dengan baik.

Sebagai developer, tentu kita ingin membuat aplikasi yang berkualitas, selain memiliki tampilan atau *user interface* yang ciamik, aplikasi juga harus dapat menyajikan performa yang optimal ketika digunakan, upaya yang dapat kita lakukan adalah memastikan aplikasi dapat berkomunikasi dengan baik ketika terjadi proses latar belakang, terdapat beberapa metode yang efektif untuk dapat menangani proses latar belakang dan tentunya sesuai dengan pedoman pengembangan dari google, diantaranya adalah kotlin *coroutine*, *Handler* dan *Executors*.

Terdapat beberapa aturan yang dapat dijadikan acuan ketika pembuatan proses latar belakang, aturan itu diantaranya adalah:

1. Membuat *asynchronous task* untuk mencegah aplikasi melakukan blok pada komponen *ui thread*.
2. Hindari pemanggilan komponen *ui* dalam proses latar belakang, hal ini karena komponen *ui* hanya boleh dijalankan di *main thread* atau *ui thread* aplikasi.

Sebelum belajar cara penggunaan komponen yang menangani proses latar belakang, kita perlu tahu apa itu thread, secara singkat merupakan kumpulan perintah yang dapat di eksekusi oleh prosesor, selain itu, thread juga dapat di eksekusi secara bersamaan dengan thread lainnya, dengan memanipulasi kerja thread, kita dapat membuat aplikasi yang dapat menangani proses latar belakang dengan baik.

untuk dapat memahami proses latar belakang pada android, kita perlu untuk mencoba secara langsung dengan sebuah studi kasus, untuk itu, saya telah

membuat video tutorial yang berisi penjelasan dari proses latar belakang hingga implementasi koneksi aplikasi dengan server menggunakan retrofit, video tutorial tersebut dapat diakses pada tautan berikut ini: [\(1015\) BELAJAR BACKGROUND PROCESS & NETWORKING ANDROID - BELAJAR BARENG - YouTube](#)