**Bab 5 | Background Thread**

Background thread adalah jalur kerja tambahan dalam aplikasi yang berjalan di luar jalur utama (main thread), untuk mengerjakan tugas-tugas berat tanpa mengganggu responsivitas aplikasi.

Di Android, background thread digunakan untuk:

* Menghindari aplikasi nge-lag atau freeze saat melakukan operasi berat seperti:  
  + Mengunduh data dari internet
  + Membaca atau menulis file besar
  + Akses ke database
  + Pemrosesan gambar atau video
* Karena Android mengharuskan semua interaksi dengan tampilan (UI) dilakukan di main thread, dan semua tugas berat harus dipindahkan ke background thread.

**5.1 Thread**

Thread adalah unit terkecil dari eksekusi program.  
Di dalam satu aplikasi, kamu bisa menjalankan beberapa thread sekaligus untuk mengolah tugas-tugas berbeda secara paralel atau bersamaan.

Gampangnya, thread itu seperti jalur kerja.  
 Kalau programmu adalah sebuah pabrik, maka thread adalah pekerja yang menyelesaikan tugas-tugas tertentu.

Contoh analogi sederhana:

* Satu aplikasi (proses) = satu pabrik.
* Banyak thread = banyak pekerja di pabrik, masing-masing mengerjakan bagian tugas tertentu.

### Dalam Android

* Main Thread: thread utama, tugasnya khusus untuk mengurus tampilan (UI).
* Background Thread: thread tambahan, tugasnya menjalankan pekerjaan berat (seperti download data, baca file besar) supaya main thread tidak terganggu.

Android mewajibkan semua perubahan UI hanya boleh dilakukan dari Main Thread.

Contoh pengunaan thread :

**Membuat Thread Manual**

****

Penjelasan:

* new Thread(...).start(); artinya membuat thread baru dan langsung memulai jalan.
* run() adalah apa yang mau dikerjakan di thread itu.
* Thread.sleep(1000); untuk pause 1 detik (1000 milidetik) tiap putaran loop.

**Mengupdate UI dari Background Thread**

****

Penjelasan:

* Operasi berat (2+2, contoh sederhana) dikerjakan di thread latar belakang.
* Setelah selesai, pakai runOnUiThread untuk update tampilan di Main Thread.
* Penting: Kalau tidak kembali ke Main Thread, aplikasi bisa crash saat ubah UI.

**6.2 Handler**

Handler adalah alat bantu di Android untuk:

* Mengirim dan menerima pesan (message) atau tugas (runnable) antara thread,
* Mengelola antrian tugas dalam sebuah thread, biasanya Main Thread atau Background Thread

Dengan Handler, kita bisa:

* Menunda eksekusi kode.
* Menjalankan kode dari background thread ke main thread (misal: update tampilan).
* Membuat sistem antrian tugas (task queue) sederhana.

Contoh Pengunaan Handler :

**Menjalankan sesuatu dengan delay 3 detik**

****

Penjelasan:

* postDelayed: menjalankan sesuatu nanti setelah sekian waktu.
* Looper.getMainLooper(): artinya Handler ini terhubung ke Main Thread.

**Mengirim tugas dari Background ke Main Thread**

****

Penjelasan:

* post(): langsung menjalankan kode di Main Thread, secepat mungkin.
* Cocok untuk update UI setelah background task.

**6.3 Executor**

Executor adalah komponen Java (dan Android) yang berfungsi untuk mengelola eksekusi tugas-tugas (Runnable) di thread secara lebih terstruktur dan efisien.

Kalau pakai Thread manual, kamu harus:

* Bikin thread satu-satu.
* Jalankan satu-satu.
* Kelola sendiri kapan selesai, kapan mati.

Kalau pakai Executor, semua itu dikelola otomatis:

* Bisa membuat banyak thread.
* Bisa atur berapa banyak tugas jalan bersamaan.
* Bisa antri tugas yang belum sempat jalan.

Contoh Pengunaan Executor :



Tipe - tipe pada executor :

| **Tipe** | **Penjelasan** |
| --- | --- |
| newSingleThreadExecutor() | Selalu pakai satu thread. |
| newFixedThreadPool(int n) | Pool dengan jumlah thread tetap (misal 5 thread). |
| newCachedThreadPool() | Dinamis dengan bisa menggunakan thread baru atau hapus thread idle. |

**6.4 Penerapan Background Thread pada aplikasi android :**

****

****

****