

# Mobile Programming

- Pertemuan 13

## Apa itu Service?

- Service bukanlah proses terpisah. Objek Service berjalan dalam proses yang sama seperti bagian dari aplikasi .
- Service bukanlah thread dari sebuah aplikasi

# Fitur Service

- Fasilitas untuk aplikasi untuk memberi tahu sistem tentang sesuatu yang ingin dilakukan di latar belakang, meminta sistem untuk menjadwalkan pekerjaan untuk service, untuk dijalankan sampai service atau pengguna secara eksplisit menghentikannya.

## Cont'd

- Suatu fasilitas bagi suatu aplikasi untuk memaparkan sebagian fungsionalitasnya ke aplikasi lain, yang memungkinkan koneksi lama dibuat ke layanan untuk berinteraksi dengannya.

# Service Lifecycle

- Jika seseorang memanggil **Context.startService()** maka sistem akan mengambil service
- Service akan terus berjalan sampai **Context.stopService()** atau **stopSelf()** dipanggil

# Cont'd

- Untuk service yang dimulai, ada dua mode operasi tambahan tambahan yang dapat mereka dijalankan, tergantung pada nilai `onStartCommand ()`:
- `START_STICKY` digunakan untuk service yang secara eksplisit dimulai dan dihentikan sesuai kebutuhan



## Cont'd

- `START_NOT_STICKY` atau `START_REDELIVER_INTENT` digunakan untuk service yang seharusnya tetap berjalan saat memproses perintah yang dikirim kepadanya.

## Cont'd

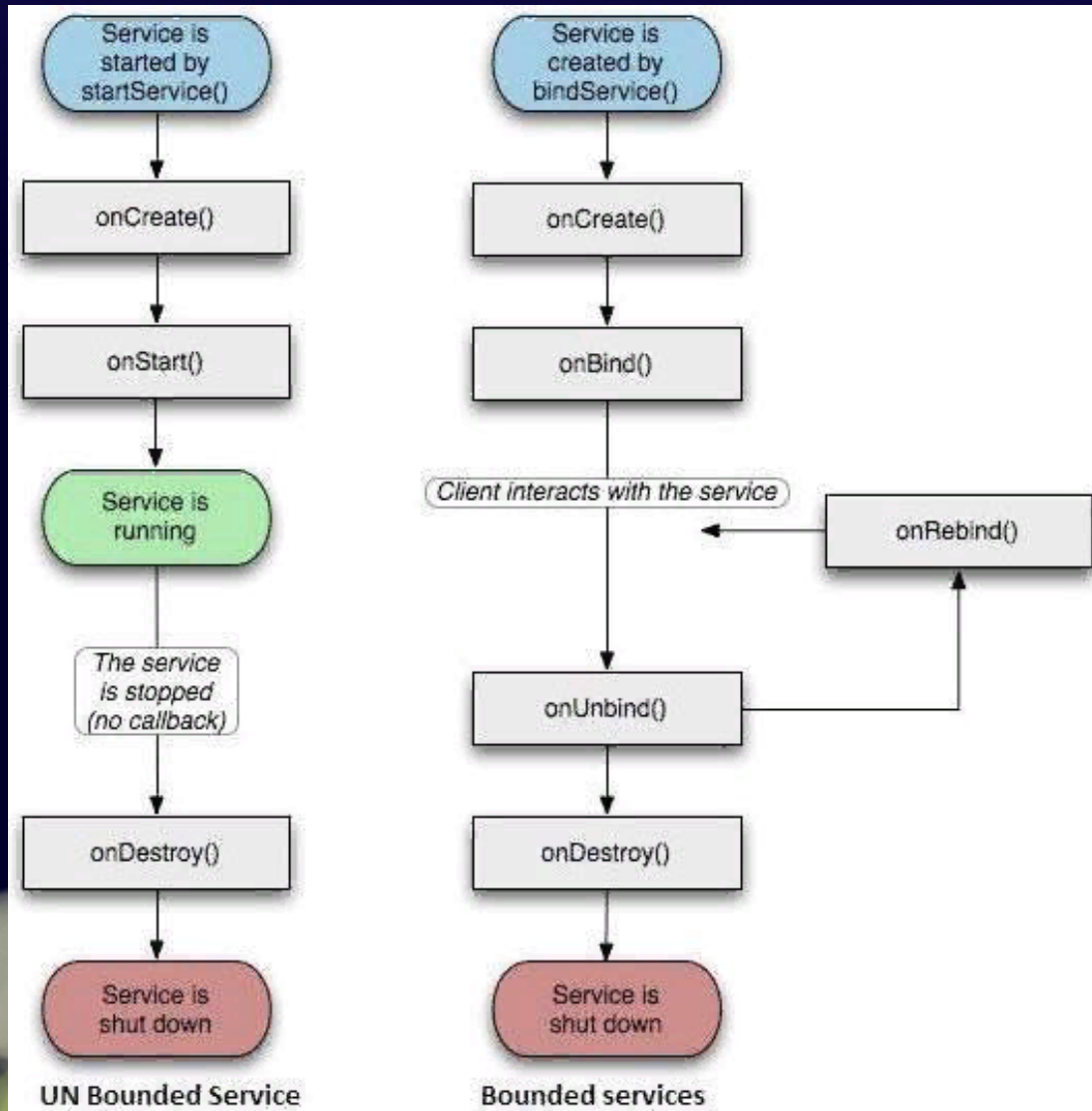
- Klien juga dapat menggunakan **Context.bindService()** untuk mendapatkan koneksi persisten ke layanan. Ini juga menciptakan layanan jika belum berjalan



## Cont'd

- Metode **onDestroy()** service dipanggil dan service dihentikan secara efektif. Semua pembersihan (menghentikan thread, membatalkan registrasi penerima) harus selesai setelah kembali dari onDestroy ().

# Ilustrasi



# Izin

- Akses global ke service dapat diberlakukan ketika dinyatakan dalam tag `<service>` manifestnya.
- Dengan melakukannya, aplikasi lain perlu mendeklarasikan elemen `<uses-permission>` yang sesuai dalam manifest mereka sendiri untuk dapat memulai, menghentikan, atau mengikat ke layanan.

# Process Life Cycle

- Sistem Android akan mencoba untuk menjaga proses hosting service selama service telah dimulai atau klien terikat padanya.
- Saat kehabisan memori dan perlu mematikan proses yang ada, prioritas proses hosting service akan menjadi lebih tinggi

# Contoh Service Lokal

- Salah satu penggunaan Service yang paling umum adalah komponen sekunder yang berjalan berdampingan dengan suatu aplikasi.
- Semua komponen .apk berjalan dalam proses yang sama kecuali dinyatakan secara eksplisit, jadi ini adalah situasi yang khas.

# Bound Service

- Server dalam antarmuka klien-server. Ini memungkinkan komponen (seperti aktivitas) untuk mengikat service, mengirim permintaan, menerima tanggapan, dan melakukan komunikasi antarproses (IPC).
- Biasanya hanya hidup saat ia melayani komponen aplikasi lain dan tidak berjalan di latar belakang tanpa batas.



## Cont'd

- Service Bound adalah implementasi kelas Service yang memungkinkan aplikasi lain untuk mengikatnya dan berinteraksi dengannya.
- Untuk memberikan pengikatan pada service, harus menerapkan metode panggilan balik onBind ().

# Membuat Bound Service

- Saat membuat service yang mengikat, pengembang harus menyediakan **IBinder** yang menyediakan antarmuka pemrograman yang dapat digunakan klien untuk berinteraksi dengan service

# Cont'd

- Menggunakan kelas Binder
- Jika service bersifat pribadi untuk aplikasi dan berjalan dalam proses yang sama dengan klien, lalu menggunakan kelas Binder kemudian mengembalikan nilai dari `onBind()`

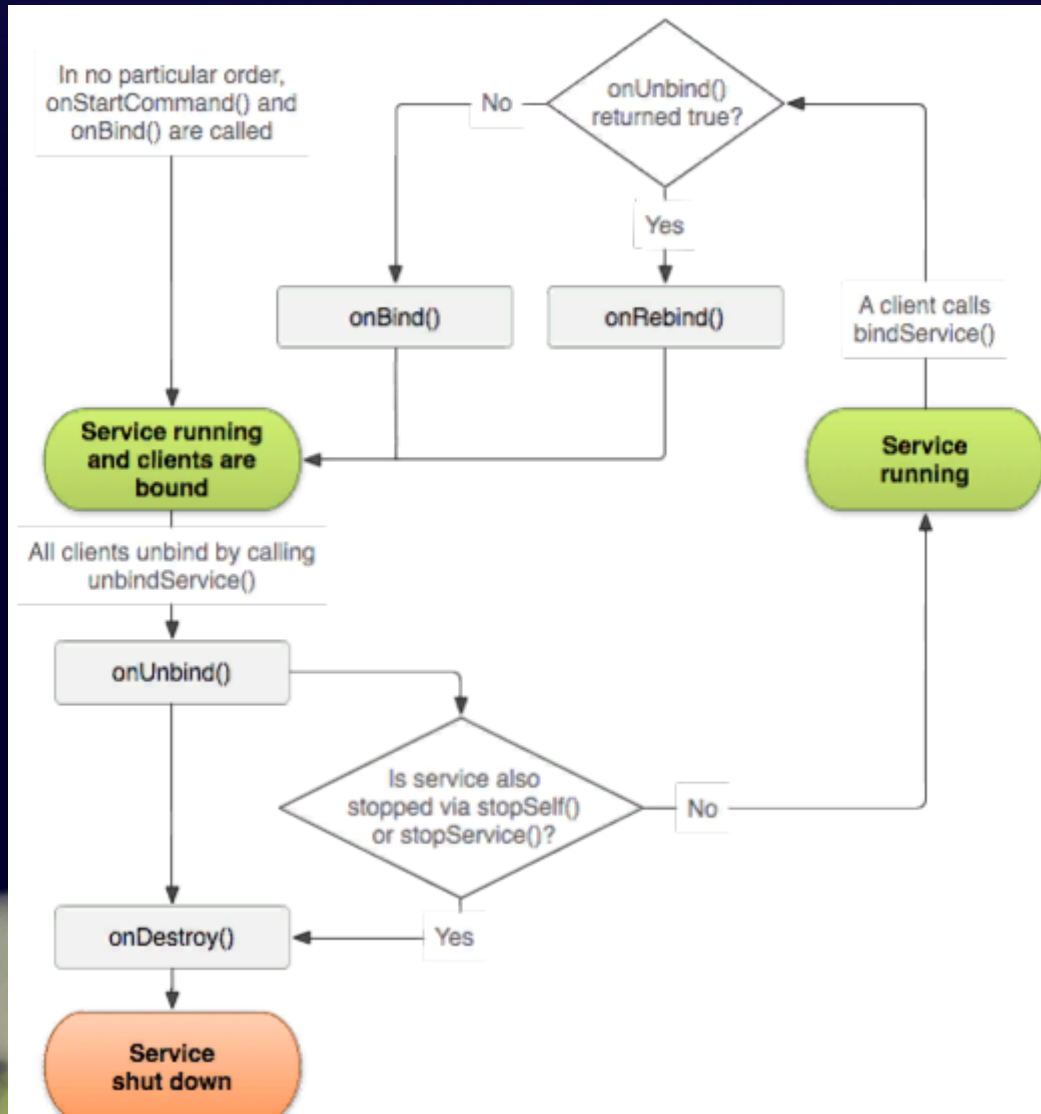
# Cont'd

- Menggunakan Messenger
- Jika membutuhkan antarmuka untuk bekerja di berbagai proses, pengembang dapat membuat antarmuka untuk service dengan Messenger

# Cont'd

- Menggunakan AIDL
- Android Interface Definition Language (AIDL) mendekomposisi objek menjadi primitif yang dapat dipahami oleh sistem operasi dan mengaturnya di seluruh proses untuk melakukan IPC.

# Service Bind Life Cycle





## Cont'd

- Ketika sebuah service tidak terikat dari semua klien, sistem Android menghancurkannya
- Pengembang tidak harus mengelola siklus hidup service, jika itu murni bound service — sistem Android akan mengaturnya

## Cont'd

- Selain itu, jika service dimulai dan menerima pengikatan, maka ketika sistem memanggil metode `onUnbind ()`, pengembang dapat mengembalikan nilai `true` jika pengembang ingin menerima panggilan ke `onRebind ()` disaat berikutnya