Mobile Programming

Pertemuan 5

Menu

- Life Cycle Activity
- Proses dan Siklus Hidup Aplikasi

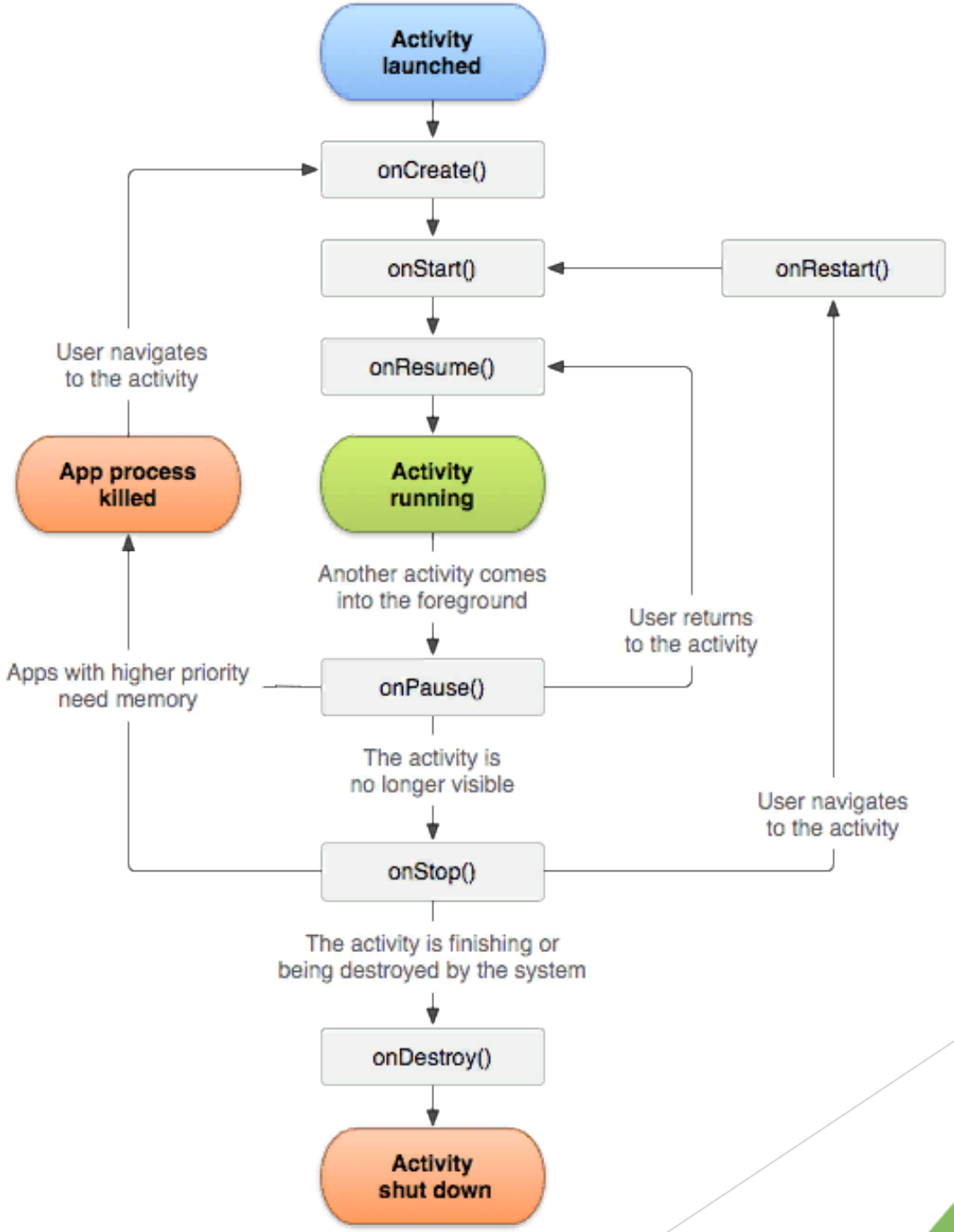
Mengapa Ada Siklus Aktivitas?

- Aplikasi Crash jika pengguna menerima panggilan telepon atau beralih ke aplikasi lain saat menggunakan aplikasi Anda.
- Mengkonsumsi sumber daya sistem ketika pengguna tidak secara aktif menggunakannya.
- Kehilangan progress pengguna jika mereka meninggalkan aplikasi Menghancurkan atau kehilangan progress pengguna ketika layar berganti orientasi

Konsep Siklus Aktivitas

- Untuk menavigasi transisi antara tahap siklus hidup aktivitas, kelas Activity menyediakan enam set inti callbacks:
  - onCreate (),
  - onStart (),
  - onResume (),
  - onPause (),
  - onStop (), dan
  - onDestroy ()

Ilustrasi


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Penjelasan

- Ketika pengguna mulai meninggalkan aktivitas, sistem memanggil metode untuk membongkar aktivitas.
- Dalam beberapa kasus, pembongkaran ini hanya sebagian; aktivitas masih berada dalam memori, dan masih dapat kembali ke latar depan.
- Jika pengguna kembali ke aktivitas itu, aktivitas dilanjutkan dari tempat pengguna tadi pergi. Dengan beberapa pengecualian, aplikasi dibatasi untuk memulai aktivitas saat berjalan di latar belakang.

Siklus Callbacks

- onCreate (),
- onStart (),
- onResume (),
- onPause (),
- onStop (),
- onDestroy ()

onCreate()

- Panggilan balik ini, akan dieksekusi saat sistem pertama kali membuat aktivitas.
- Pada pembuatan aktivitas, aktivitas memasuki kondisi ***Created***.
- Dalam metode onCreate (), Anda melakukan startup aplikasi dasar yang harus terjadi sekali selama seumur hidup aktivitas.

Contoh Kode

```
override fun onCreate(savedInstanceState: Bundle?) {
    // call the super class onCreate to complete the creation of activity like
    // the view hierarchy
    super.onCreate(savedInstanceState)
```

onStart()

- Ketika aktivitas memasuki kondisi ***Started***, sistem memanggil callbacks ini.
- Panggilan onStart () membuat aktivitas terlihat oleh pengguna, saat aplikasi mempersiapkan aktivitas untuk memasuki latar depan dan menjadi interaktif.
- Misalnya, metode ini adalah tempat aplikasi menginisialisasi kode yang mengelola UI.

onResume()

- Ketika aktivitas memasuki keadaan ***Resumed***, ia datang ke latar depan, dan kemudian sistem memanggil callbacks onResume ().
- Aplikasi akan melakukan ini
ketika pengguna kembali dari aktivitas lain, atau layar perangkat mati.

Contoh Kode

```
@OnLifecycleEvent(Lifecycle.Event.ON_RESUME)
fun initializeCamera() {
    if (camera == null) {
        getCamera()
    }
}
```

onPause()

- Sistem memanggil metode ini sebagai indikasi pertama bahwa pengguna meninggalkan aktivitas (meskipun itu tidak selalu berarti aktivitas sedang dihancurkan);
- Ini menunjukkan bahwa aktivitas tidak lagi di latar depan (meskipun mungkin masih terlihat jika pengguna berada dalam mode multi-jendela).

Contoh Kode

```
@OnLifecycleEvent(Lifecycle.Event.ON_PAUSE)
fun releaseCamera() {
    camera?.release()
    camera = null
}
```

onStop()

- Ketika aktivitas Anda tidak lagi terlihat oleh pengguna, aplikasi telah memasuki status **Stopped**, dan sistem memanggil callback onStop().
- Sistem juga dapat memanggil onStop () ketika aktivitas telah selesai berjalan, dan akan dihentikan.

Contoh Kode

```
override fun onStop() {
    // call the superclass method first
    super.onStop()

    // save the note's current draft, because the activity is stopping
    // and we want to be sure the current note progress isn't lost.
    val values = ContentValues().apply {
        put(NotePad.Notes.COLUMN_NAME_NOTE, getCurrentNoteText())
        put(NotePad.Notes.COLUMN_NAME_TITLE, getCurrentNoteTitle())
    }
```

onDestroy()

- onDestroy() dipanggil sebelum aktivitas dihancurkan. Sistem memanggil callbacks ini karena:

1. aktivitas sedang selesai, atau
2. sistem sementara menghancurkan aktivitas karena perubahan konfigurasi

Status Aktivitas dan Pembersihan Memori

- Sistem membunuh proses ketika perlu membebaskan RAM;
- Kemungkinan sistem membunuh proses tertentu tergantung pada kondisi proses pada saat itu.

Cont'd

Likelihood of being killed	Process state	Activity state
Least	Foreground (having or about to get focus)	Created Started Resumed
More	Background (lost focus)	Paused
Most	Background (not visible)	Stopped
	Empty	Destroyed

Proses dan Siklus Hidup Aplikasi

- Dalam kebanyakan kasus, setiap aplikasi Android berjalan dalam proses Linux.
- Proses ini dibuat ketika aplikasi dijalankan, dan akan tetap berjalan sampai tidak diperlukan lagi

Cont'd

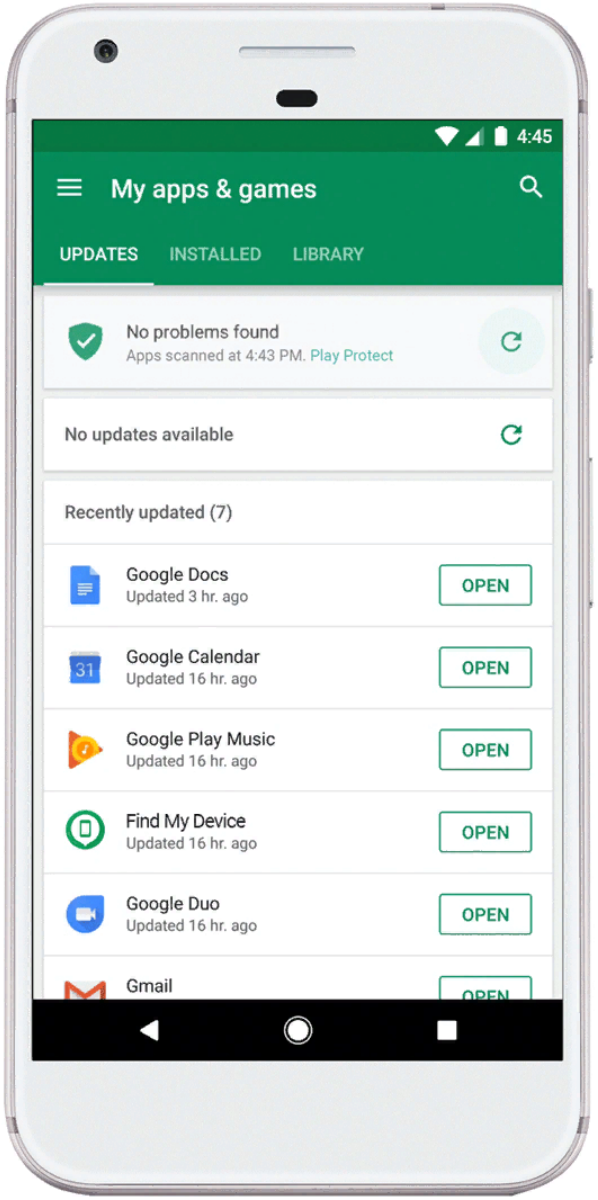
- Penting bahwa pengembang aplikasi memahami bagaimana komponen aplikasi yang berbeda (khususnya Kegiatan, Layanan, dan BroadcastReceiver) memengaruhi umur proses aplikasi.
- Tidak menggunakan komponen-komponen ini dengan benar dapat mengakibatkan sistem membunuh proses aplikasi

Cont'd

- Untuk menentukan proses mana yang harus dimatikan ketika kehabisan memori, Android menempatkan setiap proses ke dalam "hierarki kepentingan"

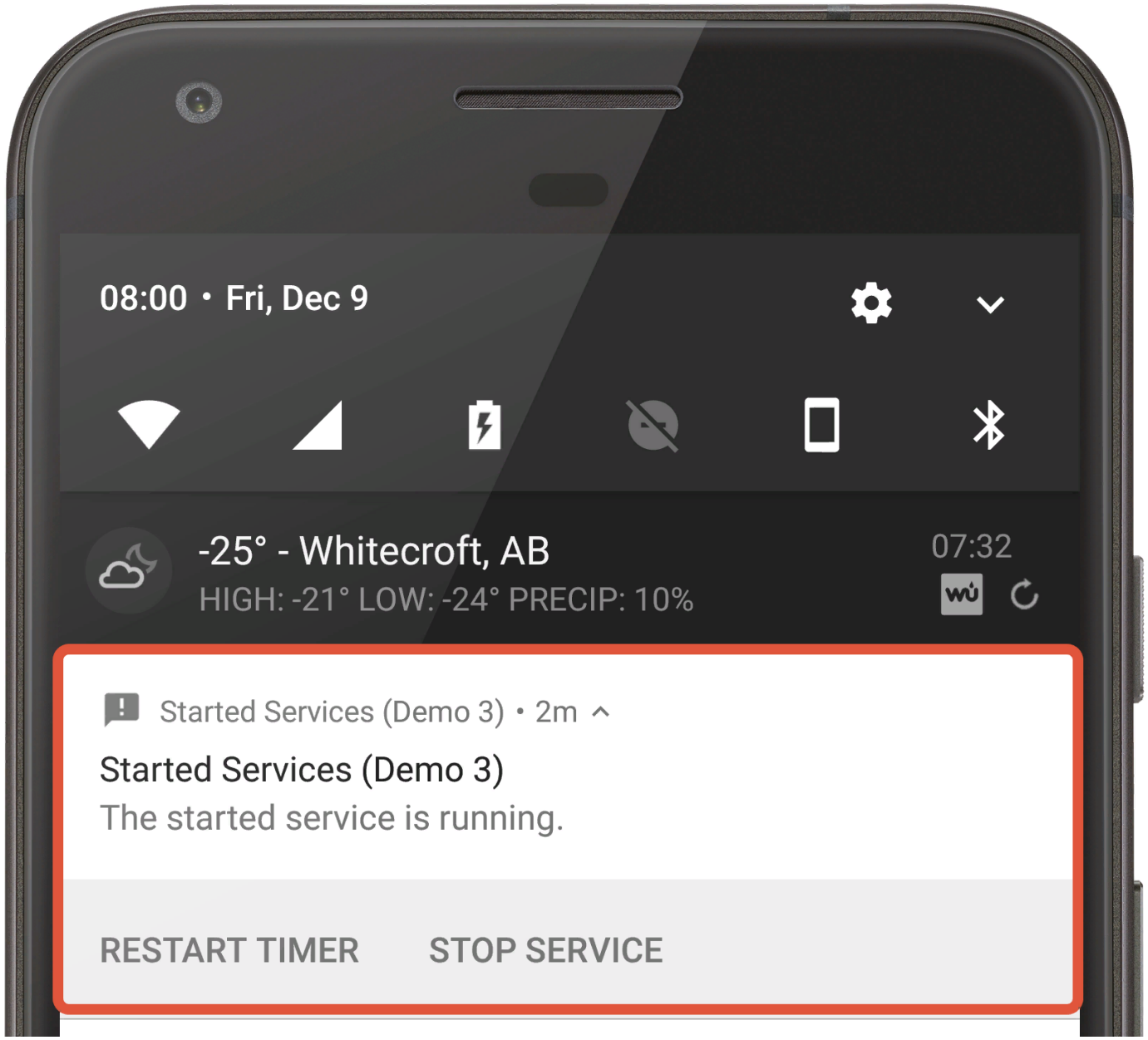
Jenis Proses - Foreground

- Proses foreground adalah proses yang sedang dilakukan pengguna, di dalamnya terdapat komponen:
  - Activity yang sedang digunakan oleh pengguna
  - BroadcastReceiver yang sedang berjalan
  - Dan Service yang sedang mengeksekusi kode di salah satu callbacks


4:45
My apps & games
UPDATES
INSTALLED
LIBRARY
No problems found
Apps scanned at 4:43 PM. Play Protect
No updates available
Recently updated (7)
Google Docs
Updated 3 hr. ago
OPEN
Google Calendar
Updated 16 hr. ago
OPEN
Google Play Music
Updated 16 hr. ago
OPEN
Find My Device
Updated 16 hr. ago
OPEN
Google Duo
Updated 16 hr. ago
OPEN
Gmail
OPEN

Jenis Proses – Visible Process

- Visible Process adalah proses yang apabila dimatikan paksa dapat menyebabkan dampak negatif yang nyata pada pengalaman pengguna. Contohnya:
- Activity yang berjalan yang terlihat di layar pengguna, namun dalam keadaan ter-Pause
- Service yang berjalan sebagai service foreground


08:00 • Fri, Dec 9
-25° - Whitecroft, AB
HIGH: -21° LOW: -24° PRECIP: 10%
07:32
Started Services (Demo 3) • 2m
Started Services (Demo 3)
The started service is running.
RESTART TIMER
STOP SERVICE

Jenis Proses

- Proses layanan adalah proses memegang Layanan yang telah dimulai dengan metode startService().
- Proses cache adalah proses yang saat ini tidak diperlukan, sehingga sistem bebas untuk membunuhnya ketika memori dibutuhkan di tempat lain.