Mobile Programming

Catatan Kuliah #6

Alauddin Maulana Hirzan, M. Kom
0607069401

Activity dan Life Cycle

Activity dan Life Cycle

Apa itu **Activity** #1

Menurut **developers.android.com**:

> *Kelas **Activity** adalah komponen penting dari aplikasi Android, dan cara activity diluncurkan dan disatukan adalah bagian mendasar dari model aplikasi platform. Tidak seperti paradigma pemrograman yang meluncurkan aplikasi dengan metode main(), sistem Android memulai kode dalam instance Activity dengan memanggil metode callback spesifik yang sesuai dengan tahapan tertentu dalam siklus hidupnya.*

Activity dan Life Cycle

Apa itu **Activity** #2

Activity menyediakan jendela tempat aplikasi menggambar UI-nya. Jendela ini biasanya memenuhi layar, tetapi mungkin lebih kecil dari layar dan mengapung di atas jendela lain.

Umumnya, satu aktivitas mengimplementasikan satu layar dalam aplikasi. Misalnya, salah satu aktivitas aplikasi dapat mengimplementasikan layar Preferensi, sementara aktivitas lain mengimplementasikan layar Pilih Foto.

Activity dan Life Cycle

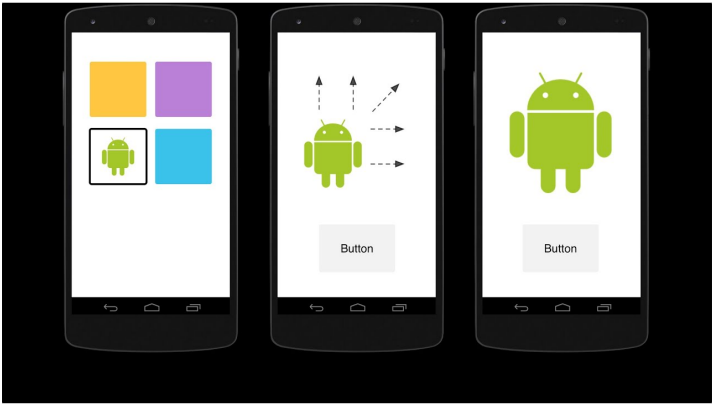
Apa itu **Activity** #3

Sebagian besar aplikasi berisi beberapa layar, yang berarti terdiri dari beberapa aktivitas. Biasanya, satu aktivitas dalam aplikasi ditetapkan sebagai aktivitas utama, yang merupakan layar pertama yang muncul saat pengguna meluncurkan aplikasi. Setiap aktivitas kemudian dapat memulai aktivitas lain untuk melakukan tindakan yang berbeda.

Sebagai contoh, aktivitas utama dalam aplikasi email sederhana dapat menyediakan layar yang menampilkan kotak masuk email. Dari sana, aktivitas utama dapat meluncurkan aktivitas lain yang menyediakan layar untuk tugas-tugas seperti menulis email dan membuka email individual.

Activity dan Life Cycle

Apa itu **Activity** #4



Activity dan Life Cycle

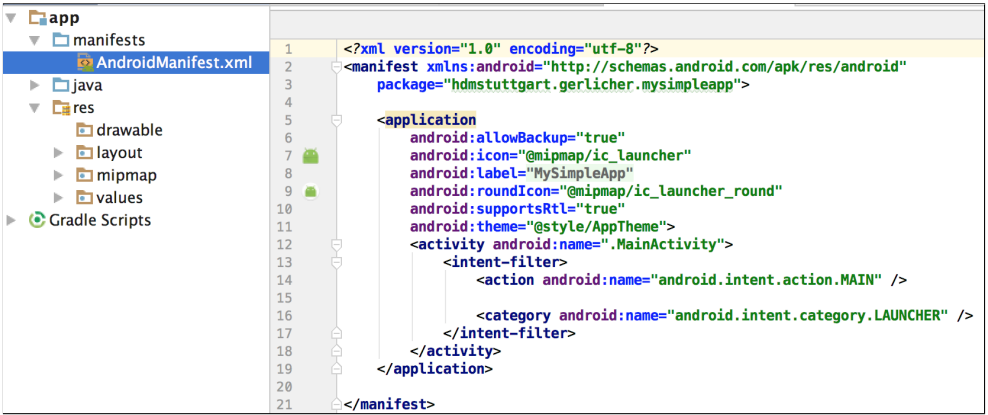
Registrasi Activity dengan Manifest #1

Untuk bisa menggunakan Activity, programmer diharuskan untuk melakukan registrasi Activity yang dibuat ke dalam file **AndroidManifest.xml**. Jika **Activity** tidak ditambahkan ke dalam **AndroidManifest.xml**, Android Studio tidak dapat mengenali **Activity** yang akan dibuat

Informasi
Android Studio akan secara otomatis melakukan registrasi ini apabila menggunakan cara tertentu.

Activity dan Life Cycle

Registrasi Activity dengan Manifest #2



Activity dan Life Cycle

Activity Life Cycle #1

Dalam penggunaan **Activity**, Android tidak serta merta melakukan eksekusi antarmuka. Android harus mengikuti sebuah siklus yang mengatur bagaimana Activity diaktifkan.

Sederhananya, aplikasi mengalami 4 siklus:

- Jika aktivitas berada di latar depan layar, berarti **aktif atau sedang berjalan.**
- Jika aktivitas telah kehilangan fokus tetapi masih ditampilkan kepada pengguna, aktivitas tersebut **terlihat**
- Jika aktivitas benar-benar dikaburkan oleh aktivitas lain, maka **dihentikan atau disembunyikan**
- Sistem dapat menghapus aktivitas dari memori dengan menghentikan prosesnya, sehingga aktivitas tersebut **hancur.**

Activity dan Life Cycle

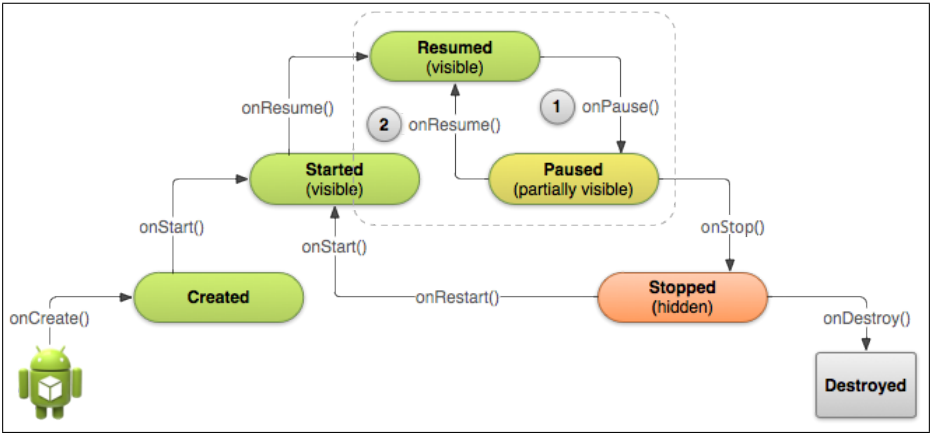
Activity Life Cycle #2

Secara menyeleuruh ada beberapa siklus yang ada. Berikut ini adalah siklus daripada **Activity**

- onCreate();
- onStart();
- onRestart();
- onResume();
- onPause();
- onStop();
- onDestroy();

Activity dan Life Cycle

Activity Life Cycle #3



Activity dan Life Cycle

Activity Life Cycle #4

Ada tiga loop utama yang mungkin menarik bagi pemrogram untuk memantau dalam aktivitas:

- **Seluruh masa hidup aktivitas** terjadi antara panggilan pertama ke **onCreate(Bundle)** sampai dengan panggilan terakhir ke **onDestroy()**. Sebuah activity akan melakukan semua penyiapan state "global" di onCreate(), dan melepaskan semua sumber daya yang tersisa di onDestroy().

Activity dan Life Cycle

Activity Life Cycle #5

- **Masa aktif aktivitas**/terlihat yang terlihat terjadi antara panggilan ke **onStart()** hingga panggilan yang sesuai ke **onStop()**. Selama waktu ini, pengguna dapat melihat aktivitas di layar, meskipun mungkin tidak berada di latar depan dan berinteraksi dengan pengguna. Di antara kedua metode ini, pengguna dapat mengelola sumber daya yang diperlukan untuk menampilkan aktivitas kepada pengguna
- **Masa hidup latar depan** aktivitas terjadi antara pemanggilan **onResume()** hingga pemanggilan **onPause()**. Selama waktu ini, aktivitas terlihat, aktif, dan berinteraksi dengan pengguna. Sebuah aktivitas dapat sering berada di antara kondisi dilanjutkan dan dijeda

Activity dan Life Cycle

Activity Life Cycle - onCreate() #1

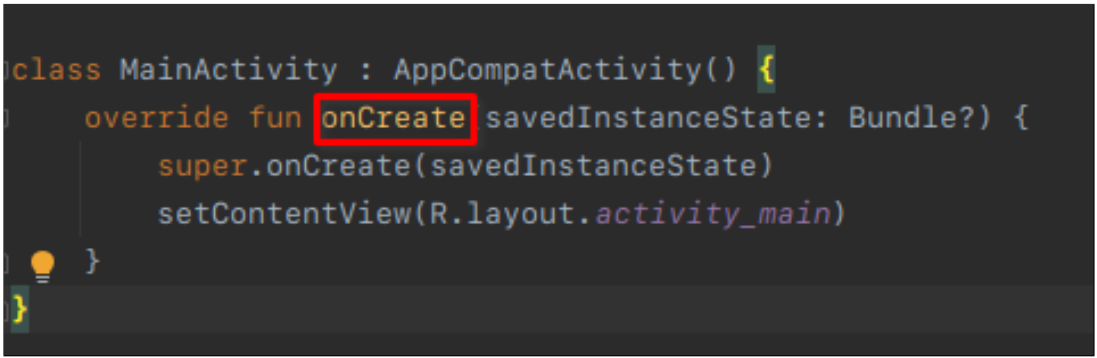
Tahap ini terjadi ketika aplikasi baru pertama kali dibuka oleh pengguna. Pada tahap ini, semua elemen utama aplikasi, seperti layout dan komponen, dibuat dan diinisialisasi.

Dipanggil ketika aktivitas pertama kali dibuat. Di sinilah programmer harus melakukan semua penyiapan statis normal: membuat tampilan, mengikat data ke daftar, dll. Metode ini juga menyediakan Bundle yang berisi status aktivitas yang sebelumnya dibekukan, jika ada. Selalu diikuti dengan onStart().

Informasi
Panggilan **onCreate** juga dapat digunakan untuk mengukur penggunaan memori suatu aplikasi. Sehingga aplikasi bisa mengetahui apakah ada aplikasi terlalu berat atau tidak

Activity dan Life Cycle

Activity Life Cycle - onCreate() #2



Activity dan Life Cycle

Activity Life Cycle - onStart() dan onRestart() #1

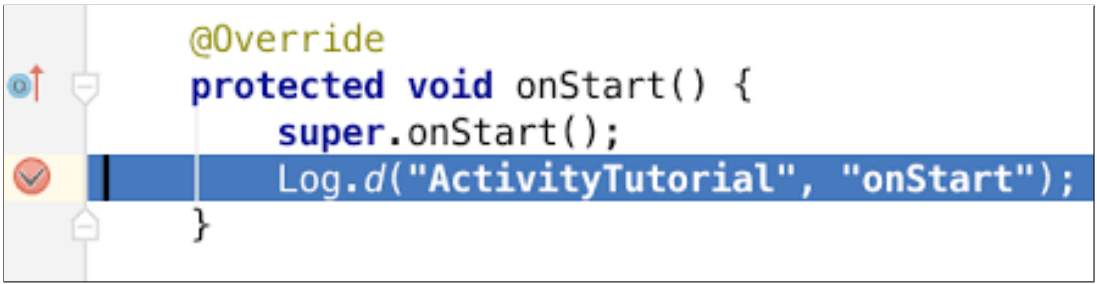
Tahap ini terjadi ketika aplikasi mulai ditampilkan pada layar ponsel pengguna. Pada tahap ini, aplikasi siap untuk berinteraksi dengan pengguna.

Dipanggil ketika aktivitas terlihat oleh pengguna. Diikuti dengan onResume() jika aktivitas muncul di latar depan, atau onStop() jika aktivitas tersebut disembunyikan.

Informasi
Karena **onCreate** hanya melakukan **loading** aplikasi ke memori, sehingga untuk bisa menjalankan aplikasi perlu melakukan panggilan **onStart**

Activity dan Life Cycle

Activity Life Cycle - onStart() dan onRestart() #2



Activity dan Life Cycle

Activity Life Cycle - onResume() dan onPause() #1

onResume() : Tahap ini terjadi ketika pengguna memulai aplikasi setelah aplikasi dihentikan atau tertutup secara sementara. Pada tahap ini, aplikasi kembali aktif dan siap untuk berinteraksi dengan pengguna.

onPause() : Tahap ini terjadi ketika pengguna meninggalkan aplikasi atau ketika aplikasi tidak lagi berada di latar depan ponsel pengguna. Pada tahap ini, aplikasi mempersiapkan diri untuk dihentikan sementara atau dimasukkan ke dalam status background.

Activity dan Life Cycle

Activity Life Cycle - onResume() dan onPause() #2

```
override fun onPause() {
    super.onPause()
    // This method is called when the activity is about to be paused
    // Put your onPause logic here
}

override fun onResume() {
    super.onResume()
    // This method is called when the activity is resumed after being pa
    // Put your onResume logic here
}
```

Activity dan Life Cycle

Activity Life Cycle - onStop() #1

Tahap ini terjadi ketika pengguna meninggalkan aplikasi dan aplikasi tidak lagi ditampilkan pada layar ponsel pengguna. Pada tahap ini, aplikasi tidak lagi aktif dan siap untuk dihentikan.

Dipanggil ketika aktivitas tidak lagi terlihat oleh pengguna. Hal ini dapat terjadi karena aktivitas baru sedang dimulai di atas, aktivitas yang sudah ada dibawa ke depan aktivitas ini, atau aktivitas ini sedang dihancurkan.

Informasi
Aplikasi yang sudah dihentikan **Stopped** harus dimulai ulang dengan menggunakan **onRestart()**

Activity dan Life Cycle

Activity Life Cycle - onStop() #2

```
override fun onStop() {
    super.onStop()
    // This method is called when the activity is stopped
    // Notify the user here
    Toast.makeText(this, "Activity stopped", Toast.LENGTH_SHORT).show()
}
```

Activity dan Life Cycle

Activity Life Cycle - onDestroy() #1

Tahap ini terjadi ketika aplikasi dihentikan secara permanen oleh pengguna atau oleh sistem operasi. Pada tahap ini, semua sumber daya yang digunakan oleh aplikasi dibebaskan dan dihapus dari memori.

Panggilan terakhir yang diterima pengguna sebelum aktivitas dihancurkan. Hal ini dapat terjadi karena aktivitas sedang selesai atau sistem menghancurkan sementara contoh aktivitas ini untuk menghemat ruang.

Activity dan Life Cycle

Activity Life Cycle - onDestroy() #2

Apa yang dilakukan app ketika **onDestroy()** dipanggil:

- Membersihkan sumber daya
- Menyimpan data
- Membatalkan pendaftaran penerima dan pendengar siaran
- Melepaskan referensi memori

Activity dan Life Cycle

Pentingnya **Lifecycle** #1

Sistem Android berusaha mempertahankan proses aplikasi selama mungkin, tetapi pada akhirnya perlu menghapus proses lama ketika memori hampir habis. Seperti yang dijelaskan di Siklus Hidup Aktivitas, keputusan tentang proses mana yang akan dihapus terkait erat dengan kondisi interaksi pengguna dengan proses tersebut.

Sistem akan membunuh proses yang kurang penting (yang terakhir) sebelum membunuh proses yang lebih penting (yang pertama).

Activity dan Life Cycle

Pentingnya **Lifecycle** #2

Empat indikator ini yang menjadi pertimbangan untuk mematikan Activity:

- Aktivitas latar depan dianggap paling penting.
- Aktivitas yang terlihat dianggap sangat penting dan tidak akan dimatikan kecuali jika diperlukan untuk menjaga aktivitas latar depan tetap berjalan.
- Aktivitas latar belakang tidak lagi penting, sehingga sistem dapat dengan aman mematikan prosesnya untuk mendapatkan kembali memori untuk proses latar depan atau proses yang terlihat.
- Proses kosong adalah proses yang tidak memiliki aktivitas atau komponen aplikasi lain

Activity dan Life Cycle

Activity Proses Panjang

Kadang-kadang sebuah Aktivitas mungkin perlu melakukan operasi jangka panjang yang ada secara independen dari siklus hidup aktivitas itu sendiri. Contohnya adalah aplikasi kamera yang memungkinkan pengguna mengunggah gambar ke situs web.

Pengunggahan mungkin membutuhkan waktu yang lama, dan aplikasi harus mengizinkan pengguna untuk meninggalkan aplikasi ketika sedang berjalan. Untuk mencapai hal ini, Activity harus memulai sebuah Service di mana unggahan berlangsung. Hal ini memungkinkan sistem untuk memprioritaskan proses dengan benar


THANK YOU