

Mobile Programming

Pertemuan 06

Alauddin Maulana Hirzan, S.Kom., M.Kom.
NIDN. 0607069401

Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang



- 1 Android Activity
- 2 View dan ViewGroup
- 3 Siklus Hidup Activity
- 4 Android Service
- 5 Siklus Hidup Service

Android Activity

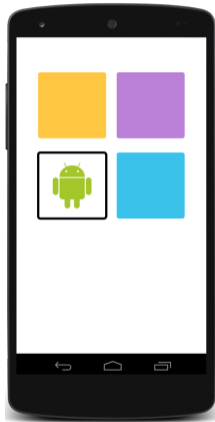
Definisi Activity

Definisi:

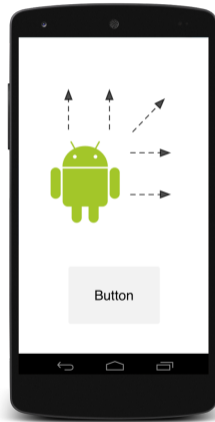
Aktivitas dalam framework Android mewakili satu tugas terfokus yang dapat dilakukan pengguna. Ini berfungsi sebagai titik masuk untuk berinteraksi dengan antarmuka pengguna aplikasi Anda dan biasanya berhubungan dengan satu layar dengan antarmuka pengguna. Aktivitas adalah komponen penting aplikasi Android karena memfasilitasi interaksi pengguna dan navigasi dalam aplikasi.

Android Activity

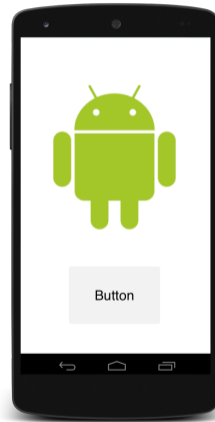
Definisi Activity



Activity 1



Scene Transition Animation
(common element)



Activity 2

Android Activity

Tujuan Activity

Tujuan Activity:

Tujuan utama Aktivitas adalah menyediakan antarmuka pengguna untuk berinteraksi dengan aplikasi Anda. Aktivitas berfungsi sebagai elemen penyusun aplikasi Android, memungkinkan pengembang mengatur antarmuka pengguna dan fungsionalitas ke dalam layar terpisah. Mereka memfasilitasi navigasi pengguna dalam aplikasi dengan memungkinkan pengguna berpindah dari satu layar ke layar lainnya dengan lancar.

Android Activity

Tujuan Activity

Aktivitas juga memainkan peran penting dalam mengelola siklus hidup suatu aplikasi. Mereka melewati berbagai keadaan, seperti pembuatan, memulai, melanjutkan, menjeda, menghentikan, dan menghancurkan, berdasarkan interaksi pengguna dan peristiwa sistem. Manajemen siklus hidup ini penting untuk mengelola sumber daya sistem secara efisien dan memastikan pengalaman pengguna yang lancar.

Android Activity

Anatomi Activity

Activity di Android biasanya terdiri dari beberapa komponen, antara lain:

- 1 **Tata Letak UI:** Tata letak UI mendefinisikan elemen visual yang terdiri dari antarmuka pengguna Aktivitas.
- 2 **Metode Siklus Hidup:** Aktivitas Android mengikuti model siklus hidup yang terdiri dari beberapa metode callback.

- 1 Android Activity
- 2 View dan ViewGroup**
- 3 Siklus Hidup Activity
- 4 Android Service
- 5 Siklus Hidup Service

View dan ViewGroup

Definisi View dan ViewGroup

View:

- Di Android, **View** mewakili satu bagian antarmuka pengguna. Bisa berupa tombol, bidang teks, gambar, atau elemen lain apa pun yang dapat berinteraksi atau dilihat pengguna di layar.
- **View** adalah turunan dari kelas **View** atau salah satu subkelasnya. Mereka bertanggung jawab untuk menggambar diri mereka sendiri dan menangani interaksi pengguna.
- Setiap **View** menempati area persegi panjang di layar dan biasanya diposisikan relatif terhadap wadah induknya.

View dan ViewGroup

Definisi View dan ViewGroup

ViewGroup:

- **ViewGroup** adalah tipe View khusus yang dapat berisi View lain (termasuk ViewGroup lainnya).
- **ViewGroup** juga merupakan subkelas View, dan menyediakan parameter tata letak untuk View turunannya, yang menentukan cara mengatur dan memosisikannya dalam batasannya.
- **ViewGroup** mengelola tata letak dan pengaturan Tampilan turunannya, menentukan posisi, ukuran, dan hubungannya satu sama lain.

View dan ViewGroup

Perbedaan View dan ViewGroup

Perbedaan antara Tampilan dan Grup Tampilan:

1. Struktur:

- **View:** Mewakili satu komponen UI, seperti tombol atau bidang teks.
- **ViewGroup:** Bertindak sebagai wadah untuk beberapa Tampilan (termasuk ViewGroup lainnya), mengatur dan mengaturnya di layar.

2. Hirarki:

- **View:** Bisa menjadi turunan langsung dari ViewGroup atau bisa ada sebagai komponen mandiri.
- **ViewGroup:** Harus digunakan untuk memuat dan mengelola tata letak Tampilan lainnya.

View dan ViewGroup

Perbedaan View dan ViewGroup

3. Tanggung Jawab:

- **View:** Menangani gambar dirinya sendiri di layar dan merespons interaksi pengguna.
- **ViewGroup:** Mengelola tata letak dan pengaturan Tampilan turunannya, menyediakan struktur untuk mengatur UI.

View dan ViewGroup

Komponen Umum View dan ViewGroup

Terdapat beberapa komponen yang sering digunakan seperti:

- 1 **TextView**: Menampilkan teks di layar. Objek ini dapat digunakan untuk menampilkan teks statis atau teks yang diperbarui secara dinamis.
- 2 **Button**: Mewakili tombol yang dapat diklik yang melakukan suatu tindakan saat ditekan.
- 3 **ConstraintLayout**: Memposisikan Objek **View** berdasarkan jarak batas atas, bawah, kiri, kanan terhadap objek lain atau parent.

- 1 Android Activity
- 2 View dan ViewGroup
- 3 Siklus Hidup Activity**
- 4 Android Service
- 5 Siklus Hidup Service

Siklus Hidup Activity

Siklus Hidup Activity

Siklus hidup Aktivitas adalah serangkaian keadaan yang dilalui suatu aktivitas seiring perkembangannya dari dibuat hingga dimusnahkan. Setiap status dalam siklus hidup memiliki metode panggilan balik terkait yang dapat diganti oleh pengembang untuk melakukan tindakan tertentu atau menangani peristiwa.

Siklus Hidup Activity

Tahapan Siklus Hidup Activity

Tahapan-tahapan Siklus:

1. **onCreate()**: Aktivitas sedang dibuat namun belum terlihat oleh pengguna.
2. **onStart()**: Aktivitas terlihat oleh pengguna namun mungkin tidak berada di latar depan, karena aktivitas lain mungkin mengaburkan sebagian aktivitas tersebut.
3. **onResume()**: Aktivitas berada di latar depan dan memiliki fokus. Ini adalah keadaan di mana aktivitas berinteraksi dengan pengguna.
4. **onPause()**: Aktivitas lain berada di latar depan, namun aktivitas ini masih terlihat dan sebagian interaktif.

Siklus Hidup Activity

Tahapan Siklus Hidup Activity

5. **onStop()**: Aktivitas tidak lagi terlihat oleh pengguna. Ini mungkin dimatikan oleh sistem atau dikembalikan ke keadaan awal.
6. **onDestroy()**: Aktivitas dihancurkan baik oleh sistem atau oleh panggilan ke `finish()`.

Siklus Hidup Activity

Penjelasan Siklus Hidup Activity

1. onCreate():

Metode ini dipanggil ketika aktivitas pertama kali dibuat. Hal ini harus digunakan untuk inisialisasi satu kali, seperti membuat tampilan, mengikat data, atau menginisialisasi variabel. Parameter `saveInstanceState` digunakan untuk memulihkan status aktivitas sebelumnya jika aktivitas tersebut dimusnahkan dan dibuat ulang oleh sistem.

2. onStart():

Dipanggil ketika aktivitas mulai terlihat oleh pengguna. Pada titik ini, aktivitas memasuki keadaan dimulai. Metode ini biasanya digunakan untuk menginisialisasi komponen yang diperlukan saat aktivitas terlihat.

Siklus Hidup Activity

Penjelasan Siklus Hidup Activity

3. onResume():

Dipanggil ketika aktivitas hendak berinteraksi dengan pengguna. Metode ini menandai aktivitas berada di latar depan. Hal ini adalah tempat ideal untuk memulai animasi, memperoleh sumber daya eksklusif, atau melakukan tugas apa pun yang harus dilakukan saat aktivitas berada di latar depan.

4. onPause():

Dipanggil ketika aktivitas tidak lagi berada di latar depan dan pengguna tidak lagi berinteraksi dengannya. Metode ini adalah tempat yang baik untuk melakukan perubahan yang belum disimpan atau menjeda proses yang sedang berjalan yang tidak diperlukan saat aktivitas tidak terlihat.

Siklus Hidup Activity

Penjelasan Siklus Hidup Activity

5. `onStop()`:

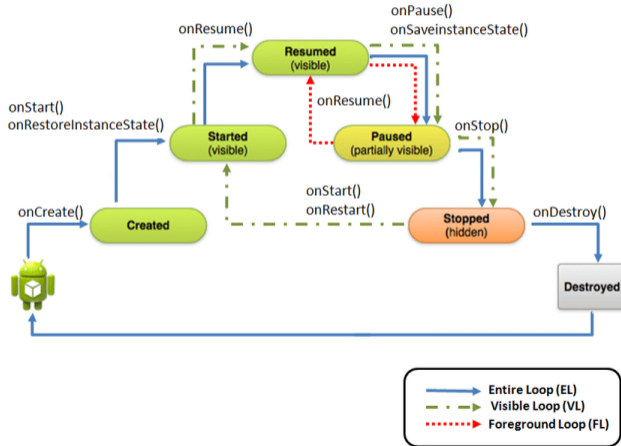
Dipanggil ketika aktivitas tidak lagi terlihat oleh pengguna. Metode ini dipanggil ketika aktivitas memasuki keadaan berhenti, yang mungkin terjadi ketika aktivitas lain muncul di latar depan. Gunakan metode ini untuk melepaskan atau menyesuaikan sumber daya yang tidak diperlukan lagi saat aktivitas tidak terlihat.

6. `onDestroy()`:

Dipanggil sebelum aktivitas dimusnahkan. Metode ini adalah kesempatan terakhir untuk melepaskan sumber daya, membatalkan pendaftaran penerima, atau melakukan pembersihan apa pun yang diperlukan sebelum aktivitas dihapus dari memori.

Siklus Hidup Activity

Ilustrasi Siklus Hidup Activity



- 1 Android Activity
- 2 View dan ViewGroup
- 3 Siklus Hidup Activity
- 4 Android Service**
- 5 Siklus Hidup Service

Android Service

Definisi Service

Definisi:

Layanan / Service di Android adalah komponen yang dirancang untuk melakukan operasi yang berjalan lama di latar belakang tanpa memerlukan interaksi pengguna. Service tidak menyediakan antarmuka pengguna. Sebaliknya, dia berjalan diam-diam di latar belakang, melakukan tugas-tugas seperti memutar musik, menangani transaksi jaringan, atau melakukan operasi I/O file.

Android Service

Jenis Service

1. **Layanan Latar Depan:** Layanan latar depan adalah layanan yang diperhatikan oleh pengguna. Mereka berjalan dengan pemberitahuan terlihat yang memberikan umpan balik berkelanjutan kepada pengguna. Layanan latar depan digunakan untuk tugas-tugas yang dianggap penting bagi pengalaman pengguna
2. **Layanan Latar Belakang:** Layanan latar belakang adalah layanan yang berjalan secara diam-diam di latar belakang tanpa antarmuka pengguna apa pun.

Android Service

Penggunaan Service

Layanan/Service Android digunakan untuk tujuan berikut:

1. **Melakukan Tugas Latar Belakang:** Layanan dapat digunakan untuk melakukan tugas latar belakang yang tidak memerlukan interaksi pengguna, seperti mengunduh file, menyinkronkan data dengan server, atau memproses kumpulan data besar.
2. **Menangani Pemrosesan Jarak Jauh:** Layanan dapat menjalankan tugas untuk proses jarak jauh, memungkinkan komunikasi antara berbagai aplikasi yang berjalan pada perangkat atau pada perangkat berbeda yang terhubung melalui jaringan.

Android Service

Penggunaan Service

- 3. Mengelola Operasi yang Berjalan Lama:** Layanan dapat mengelola operasi yang perlu terus berjalan bahkan ketika UI aplikasi tidak terlihat atau aktif. Misalnya, aplikasi pemutar musik menggunakan layanan untuk terus memutar musik di latar belakang saat pengguna berinteraksi dengan aplikasi lain.
- 4. Menangani Komunikasi Antar-Proses (Inter-Process Communication):** Layanan dapat memfasilitasi komunikasi antara berbagai komponen aplikasi atau antara berbagai aplikasi yang berjalan pada perangkat.

- 1 Android Activity
- 2 View dan ViewGroup
- 3 Siklus Hidup Activity
- 4 Android Service
- 5 Siklus Hidup Service**

Siklus Hidup Service

Siklus Hidup Service

Memiliki kesamaan dengan Siklus Hidup Activity, namun berbeda dalam status keadaannya. Terdapat status baru seperti **onStartCommand()**, **onBind()**, maupun **onUnBind()**.

Sehingga dalam penerapannya memerlukan cara yang berbeda dengan siklus activity

Siklus Hidup Service

Siklus Hidup Service

Status Keadaan Service:

- 1 **Creation:** Layanan dibuat menggunakan konstruktornya atau melalui panggilan ke metode `startService()`.
- 2 **Started:** Layanan dimulai dan dapat berjalan tanpa batas waktu di latar belakang, meskipun komponen yang memulai layanan tersebut dimusnahkan.
- 3 **Bound:** Komponen lain mengikat layanan menggunakan metode `bindService()` untuk berinteraksi dengannya secara langsung.
- 4 **Destroyed:** Layanan dihentikan menggunakan metode `stopService()` atau `stopSelf()`, atau sistem menghancurkannya untuk mendapatkan kembali sumber daya.

Siklus Hidup Service

Metode Siklus Hidup Service

1. **onCreate**: Metode ini dipanggil saat layanan pertama kali dibuat. Biasanya digunakan untuk tugas inisialisasi seperti menyiapkan variabel, menginisialisasi struktur data, atau mengalokasikan sumber daya.
2. **onStartCommand()**: Metode ini dipanggil setiap kali layanan dimulai menggunakan metode `startService()`. Digunakan untuk melakukan pekerjaan sebenarnya dari layanan, seperti menangani permintaan atau menjalankan tugas latar belakang.
3. **onBind()**: Metode ini dipanggil ketika komponen lain (seperti aktivitas) mengikat layanan menggunakan metode `bindService()`. Dia mengembalikan antarmuka `IBinder` yang dapat digunakan klien untuk berkomunikasi dengan layanan.

Siklus Hidup Service

Metode Siklus Hidup Service

- onUnBind():** Metode ini dipanggil ketika semua klien telah terputus dari layanan dengan memanggil `unbindService()`. Metode ini adalah kesempatan untuk membersihkan sumber daya atau melakukan tugas pembersihan lainnya jika diperlukan.
- onDestroy():** Metode ini dipanggil ketika layanan tidak lagi digunakan dan sedang dimusnahkan. Digunakan untuk melepaskan sumber daya apa pun atau melakukan pembersihan akhir sebelum layanan dihentikan.

Terima Kasih