



# Android Development

Study Independent – Kampus Merdeka

Riky Ahmad Fathoni



# Riky Ahmad Fathoni

- Telegram : [@rikyahmad](https://t.me/@rikyahmad)
- LinkedIn : [linkedin.com/in/riky](https://www.linkedin.com/in/riky)
- Facebook : [fb.com/rikyahmadf](https://www.facebook.com/fb.com/rikyahmadf)
- Email : [riky.fathoni@gmail.com](mailto:riky.fathoni@gmail.com)



# Agenda Day 2

- Activity
- Fragment
- Intent dan Filters
- Service dan Broadcast
- Permissions
- Sensor

---

# Activity

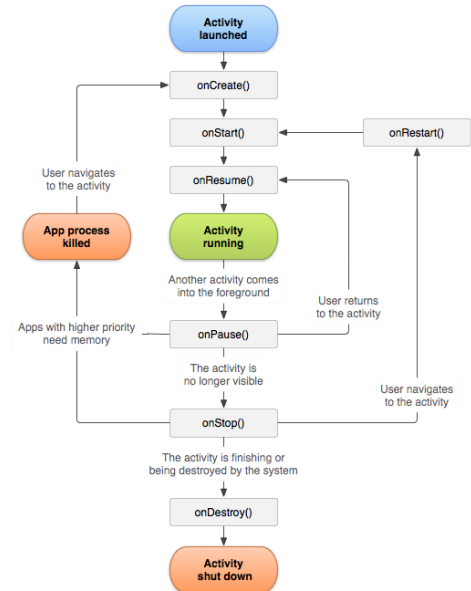


# Activity

- Saat kita membuat aplikasi seperti di pemrograman Java atau Kotlin, main function sebagai function yang akan diluncurkan ketika aplikasi berjalan.
- Di Android tidak seperti itu, Android memiliki fitur yang bernama Activity, dimana nanti object Activity tersebut akan secara otomatis dijalankan oleh Android
- Activity yang dibuat harus dideklarasikan di file **AndroidManifest.xml**
- Sebuah activity memiliki siklus hidup.

# Activity Lifecycle

- Activity tidak hidup abadi selamanya, dia bisa diciptakan dan dihancurkan. Activity memiliki siklus hidup (lifecycle) yang merupakan kondisi yang akan dialami saat diciptakan sampai dihancurkan.
- Ada beberapa kondisi yang akan dialami oleh Activity seperti yang tergambar pada flow chart berikut ini.





# Class Activity

- Untuk membuat Activity, kita perlu membuat class turunan dari Activity
- Saat kita membuat project Android, secara otomatis akan ada sebuah MainActivity yang merupakan class turunan dari AppCompatActivity
- AppCompatActivity merupakan turunan dari class Activity yang memungkinkan kita menggunakan fitur baru Android di versi Android lama, oleh karena itu direkomendasikan menggunakan class AppCompatActivity
- <https://developer.android.com/reference/android/app/Activity>



# Mendaftarkan Activity

- Untuk memberitahu kepada Android, bahwa kita membuat Activity, kita harus mendaftarkannya di Manifest File
- Selain itu, terkadang kita perlu menambahkan intent untuk menambahkan informasi seperti misalnya, menandai sebuah Activity bahwa ini adalah Main Activity, dan menandai bahwa Activity ini harus dijalankan ketika aplikasi Android diluncurkan/dibuka (LAUNCH)



## Kode : Contoh Manifest Activity

```
25 <activity
26     android:name=".activity.MainActivity"
27     android:exported="true" >
28     <intent-filter>
29         <action android:name="android.intent.action.MAIN" />
30
31         <category android:name="android.intent.category.LAUNCHER" />
32     </intent-filter>
33 </activity>
```

## Kode : Activity onCreate()

```
MainActivity.java x
1  package com.staygrateful.app.mbkm2022;
2
3  import ...
7
8  public class MainActivity extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(@Nullable Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14     }
15 }
```

—

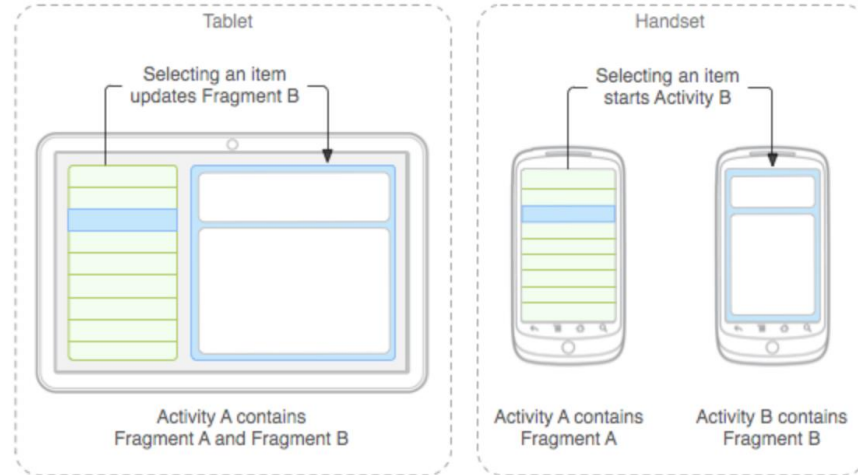
Fragment



# Fragment

- Komponen yang dapat digunakan kembali dari antarmuka pengguna android activity yang digunakan untuk membuat UI yang dinamis dan fleksibel.
- Fragmen memiliki siklus hidup itu sendiri tetapi selalu tertanam dengan aktivitas sehingga siklus hidup fragmen secara langsung dipengaruhi oleh siklus hidup aktivitas.
- Kita bisa menyisipkan fragmen ke dalam layout aktivitas dengan mendeklarasikan fragmen dalam file layout aktivitas, sebagai elemen <fragment>, atau dari kode aplikasi dengan menambahkannya ke ViewGroup yang ada.

# Gambaran Fragment





# Kegunaan Fragment

- Sebelum Fragment diperkenalkan, kita hanya dapat menampilkan satu activity di layar pada satu waktu tertentu sehingga tidak dapat membagi layar.
- Dengan bantuan Fragment, kita dapat membagi layar di berbagai bagian dan mengontrol bagian-bagian yang berbeda secara terpisah.
- Dengan menggunakan Fragment, kita dapat membentuk beberapa Fragment dalam satu activity.
- Fragment memiliki event, layout, dan status mereka sendiri. Hal ini memberikan fleksibilitas dan juga mengatasi limitasi dalam activity di layar pada suatu waktu.



# Kode : Fragment

```
Samplefragment.java
1  package com.staygrateful.app.mbkm2022.fragment;
2
3  import ...
4
12
13  public class SampleFragment extends Fragment {
14
15      public SampleFragment() {
16          // Required empty public constructor
17      }
18
19      @Override
20      public void onCreate(Bundle savedInstanceState) {
21          super.onCreate(savedInstanceState);
22      }
23
24
25      @Override
26      public View onCreateView(LayoutInflater inflater, ViewGroup container,
27                              Bundle savedInstanceState) {
28          // Inflate the layout for this fragment
29          return inflater.inflate(R.layout.fragment_sample, container, attachToRoot: false);
30      }
31  }
```

---

# Intent dan Filter





# Intent

- Intent merupakan mekanisme untuk melakukan sebuah action dan komunikasi antar komponen aplikasi.
- Contoh, Ketika tekan tombol untuk membuka peta, kamera, atau halaman atau activity lainnya. Perpindahan inilah yang dinamakan Intent.
- Intent dapat digunakan pada Activity, Service, Broadcast.



# Intent Filter

- Deklarasi ekspresi dalam file manifes aplikasi yang menetapkan tipe intent yang akan diterima komponen.
- Misalnya, dengan mendeklarasikan filter intent untuk aktivitas, akan memungkinkan aplikasi lain untuk langsung memulai aktivitas dengan intent tertentu.
- Demikian pula, jika tidak mendeklarasikan filter intent untuk suatu aktivitas, maka aktivitas tersebut hanya bisa dimulai dengan intent eksplisit.



# Jenis-Jenis Intent

- Intent Implicit : Berfungsi melakukan perpindahan activity (halaman) menuju ke aplikasi internal smartphone kamu. Contohnya ketika hendak membuka sebuah kamera.
- Intent Explicit : Berfungsi melakukan perpindahan activity (halaman) ke activity (halaman) lainnya dalam satu aplikasi yang sama. Explicit intent bekerja dengan menggunakan nama kelas yang dituju, misal `com.stechoq.activity.DetailActivity`. Umumnya intent ini digunakan untuk mengaktifkan komponen pada satu aplikasi.

# Intent Implicit

```
53 @ public static void shareImplicitIntent(Context context) {  
54     //Perintah Intent Implicit untuk share ke aplikasi lain, misal sosmed  
55     Intent intent = new Intent(Intent.ACTION_SEND);  
56  
57     //Data yang ingin dishare  
58     intent.putExtra(Intent.EXTRA_TEXT, value: "Hallo saya share ke sosial media");  
59     intent.setType("text/plain");  
60  
61     //Menjalankan perintah  
62     context.startActivity(Intent.createChooser(intent, title: "Share to :"));  
63 }
```



# Intent Explicit

```
65 public static void shareExplicitIntent(Activity activity, Class<Activity> activityToOpen) {  
66     //Activity yang dituju  
67     Intent intent = new Intent(activity, activityToOpen);  
68  
69     //Data  
70     intent.putExtra( name: "MY_DATA", value: "Hallo saya share ke sosial media");  
71  
72     //Menjalankan perintah  
73     activity.startActivity(intent);  
74 }
```

---

# Service dan Broadcast



# Service

- Service adalah sebuah komponen aplikasi yang bisa melakukan operasi yang berjalan lama di latar belakang dan tidak menyediakan antarmuka pengguna.
- Misalnya, layanan yang menangani transaksi jaringan, memutar musik, melakukan file I/O, atau berinteraksi dengan penyedia materi dari latar belakang.
  - a. Foreground (latar depan)
  - b. Background (latar belakang)
  - c. Bound (layanan terikat)



# Broadcast

- Broadcast adalah sebuah media untuk mengirim atau menerima pesan siaran dari sistem Android dan aplikasi Android lainnya, mirip dengan pola desain publish-subscribe.
- Misalnya, sistem Android mengirimkan siaran saat terjadi berbagai peristiwa sistem, seperti ketika sistem melakukan booting, sistem mendeteksi perangkat usb atau saat perangkat memulai pengisian daya.
  - a. Send Broadcast (menerima siaran)
  - b. Receive Broadcast (mengirim siaran)



---

# Permissions



# Permissions

- Android secara default tidak membutuhkan izin akses.
- Namun saat aplikasi ingin menggunakan fitur-fitur khusus yang ada di perangkat perlu membutuhkan izin dari pengguna yang secara eksplisit di deklarasikan pada file manifest (misalnya untuk tipe izin runtime seperti, mengakses kamera, mengirim SMS, membaca kontak, dsb.).
- Izin akses runtime dibutuhkan untuk perangkat dengan minimal API 23 atau lebih tinggi.



Sensor



# Sensor

- Sebagian besar perangkat Android memiliki sensor built-in yang mengukur gerakan, orientasi, dan berbagai kondisi lingkungan.
- Sensor ini mampu menyediakan data mentah dengan presisi dan akurasi tinggi, dan berguna jika ingin memantau positioning atau pergerakan tiga dimensi perangkat.
- Misalnya, suatu game mungkin akan melacak pembacaan dari sensor gravitasi perangkat untuk menyimpulkan gerakan dan gestur pengguna yang kompleks, seperti kemiringan, goyangan, rotasi, atau ayunan.
- [https://developer.android.com/guide/topics/sensors/sensors\\_overview](https://developer.android.com/guide/topics/sensors/sensors_overview)



# Kategori Sensor

- Platform Android mendukung tiga kategori sensor yang luas, meliputi:
  - a. Sensor gerak : Sensor ini mengukur gaya akselerasi dan gaya rotasi pada tiga sumbu. Kategori ini mencakup akselerometer, sensor gravitasi, giroskop, dan sensor vektor rotasi.
  - b. Sensor lingkungan: Sensor ini mengukur berbagai parameter lingkungan, seperti tekanan dan suhu udara sekitar, pencahayaan, serta kelembaban. Kategori ini meliputi barometer, fotometer, dan termometer.
  - c. Sensor posisi: Sensor ini mengukur posisi fisik perangkat. Kategori ini meliputi sensor orientasi dan magnetometer.



**Terima kasih**