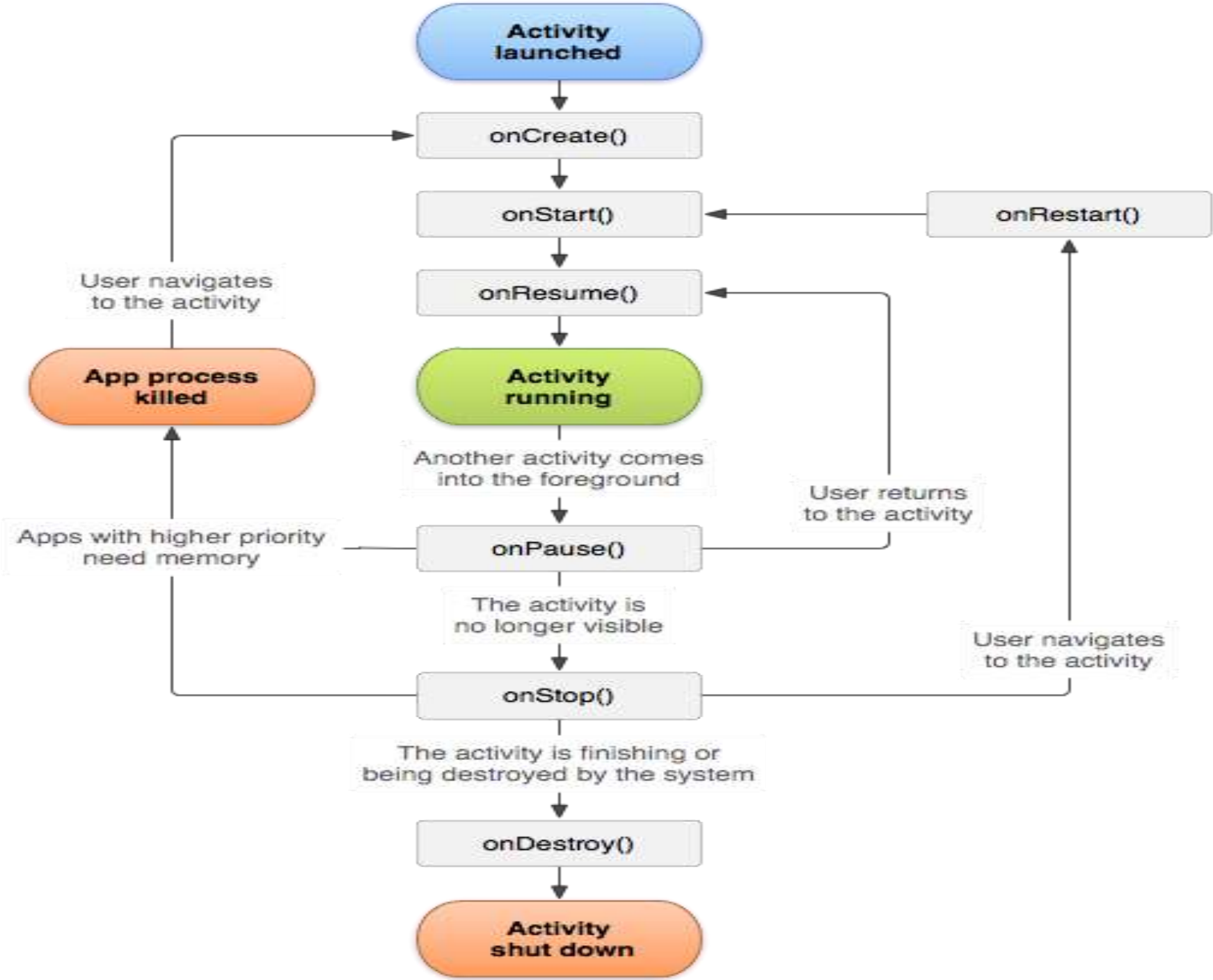
Activity Life Cycle – Siklus Hidup Activity

Guntur Maulana Zamroni
Teknik Informatika
Universitas Ahmad Dahlan

- Aplikasi Android merupakan kumpulan dari beberapa *Activity* yang tergabung secara bebas antara *Activity* satu dengan yang lainnya, yang mana kumpulan tersebut memiliki satu *Activity* utama yang digunakan untuk memanggil *Activity* lainnya secara bertumpuk.
- Setiap *Activity* mendapatkan tampilan/layout sebagai antarmuka dengan user.
- Pendek kata, **di Activity inilah user interface ditampung**

Quiz

Setiap aplikasi Android yang berinteraksi dengan user minimal memiliki satu Activity. (Benar/Salah)


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

method	deskripsi	penggunaan
`onCreate()`	activity dimulai (tapi tidak terlihat ke user)	untuk inisialisasi code, mengatur layout (`setContentView()`), initalize variabel, adapter dan lain-lain
`onStart()`	activity terlihat ke user, tapi belum siap untuk proses interaksi.	untuk memonitor perubahan yang terjadi dalam UI. tapi jarang digunakan.
`onResume()`	activity terlihat dan siap dipakai untuk interaksi dengan user.	untuk animasi, membuka akses eksklusif device seperti kamera dll.
`onPause()`	activity berjalan di *background* dan berhenti berinteraksi ke user. ini bisa terjadi jika ada activity lain yang berjalan diatas activity tersebut.	untuk mengatur ulang/ menyimpan semua proses yang selesai dalam proses `onResume()`.
`onStop()`	activity tidak terlihat ke user	membatalkan semua proses yang ada pada `onStart()`.
`onDestroy()`	hampir sama seperti `onCreate()` namun dalam arti sebaliknya. dipicu dengan `finish()` pada class activity atau sistem mebutuhkan tambahan memori.	pembersihan proses, misal jika ada activity berjalan di *background* maka dibuat berhenti.
`onRestart()`	activity berhenti dan memulai lagi.	jarang di implementasikan.

Pemrograman Activity

Metode yang sering diimplementasikan pada suatu Activity:

1. onCreate(Bundle) adalah dimana kita dapat menginisialisasi activity. Pada bagian ini sering akan memanggil setContentView(int) untuk mendefinisikan layout resources yang mendefinisikan UI, dan menggunakan findViewById(int).
2. onPause adalah dimana kita berurusan dengan pemakai yang meninggalkan activity kita, setiap perubahan harus di commit pada titik ini (Persistence State).

Quiz

```
public class HelloWorldActivity extends _____1{

@_____2
public void _____3(Bundle savedInstanceState) {
_____4.onCreate(savedInstanceState);
setContentView(R.____5.main);
}
```

Pada masing-masing isian kosong dapat diisi dengan keyword:

a. Activity
b. Override
c. Overwrite
d. OnCreate
e. onCreate
f. Initialize
g. Constructor
h. Super
i. Parent
j. Layout
k. id

```java
public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
      System.out.println("onCreate");
    }

    @Override
    protected void onStart() {
      // TODO Auto-generated method stub
      super.onStart();
      System.out.println("onStart");
    }

    @Override
    protected void onRestart() {
      // TODO Auto-generated method stub
      super.onRestart();
      System.out.println("onReStart");
    }

    @Override
    protected void onResume() {
      // TODO Auto-generated method stub
      super.onResume();
      System.out.println("onResume");
    }

    @Override
    protected void onPause() {
      // TODO Auto-generated method stub
      super.onPause();
      System.out.println("onPause");
    }

    @Override
    protected void onStop() {
      // TODO Auto-generated method stub
      super.onStop();
      System.out.println("onStop");
    }

    @Override
    protected void onDestroy() {
      // TODO Auto-generated method stub
      super.onDestroy();
      System.out.println("onDestory");
    }
```

Class MainActivity adalah subclass dari Activity (mewariskan semua atribut dan prilaku dari class Activity)

Menentukan layout yang menjadi UI pada MainActivity.

Directive @Override menyatakan bahwa method onRestart pada parent class Activity akan di-override pada class ini.

Keyword super manyatakan memanggil method pada parent class, dalam hal method onRestart.

Jalankan aplikasi disamping ini, kemudian aktifkan DDMS untuk mengamati Event2 yang terjadi pada suatu activity.

Creating an Activity

- To create an activity, you must create a subclass of Activity (or an existing subclass of it).
- Android menggunakan sistem ***callback*** untuk menerapkan daur hidup Activity. **Callback** adalah fungsi-fungsi yang dipanggil oleh sistem ketika sistem menerima event.

Declaring the activity in the manifest

- You must declare your activity in the manifest file in order for it to be accessible to the system.
- To declare your activity, open your manifest file and add an <activity> element as a child of the <application> element.
- For example:

```
<manifest ... >
  <application ... >
      <activity android:name=".ExampleActivity" />
      ...
  </application ... >
  ...
</manifest >
```

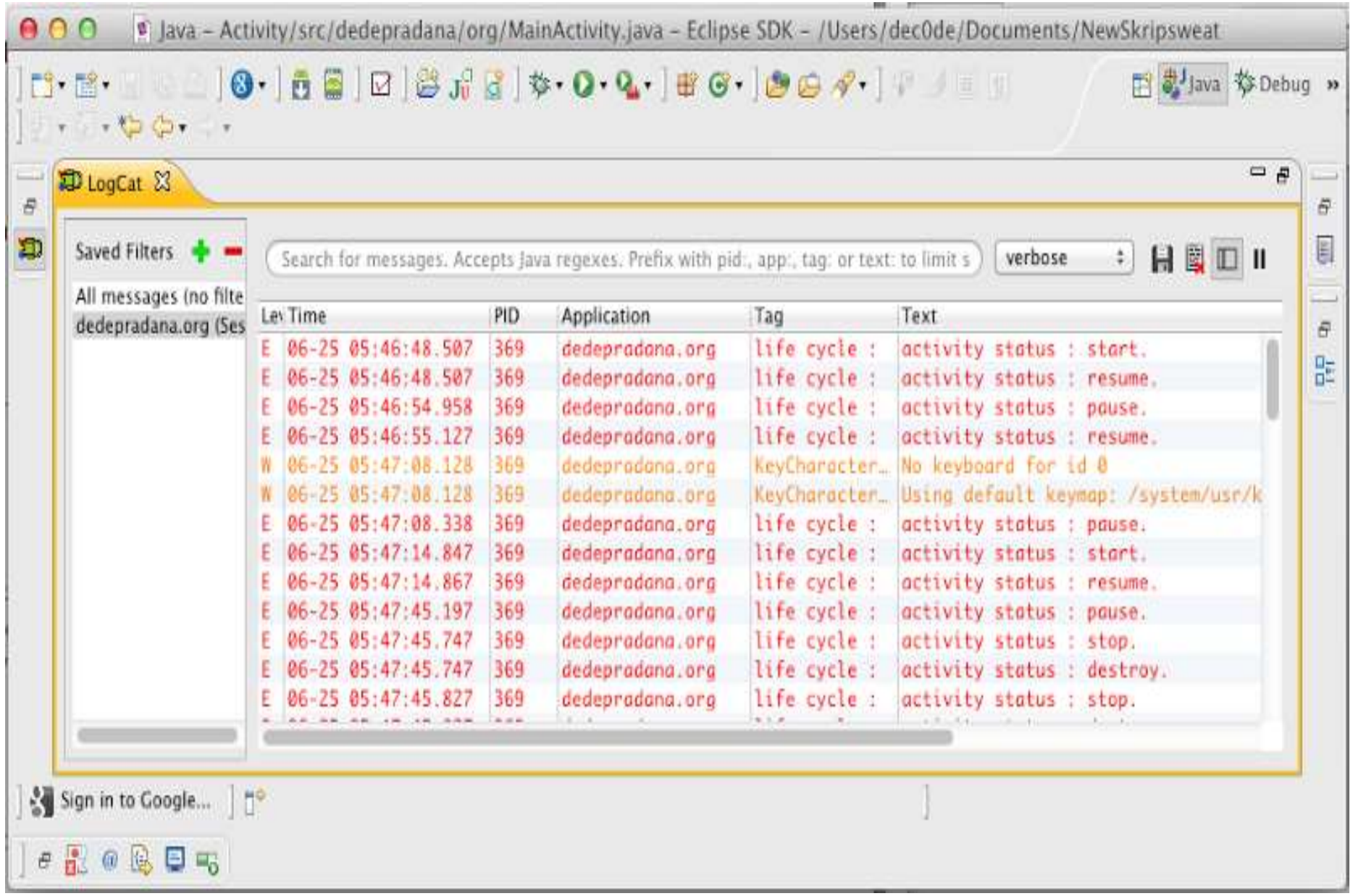
Managing the Activity Lifecycle

- Mengelola Activity Lifecycle dengan menerapkan metode callback sangat penting untuk mengembangkan aplikasi yang kuat dan fleksibel.
  Activity Lifecycle dari suatu kegiatan secara langsung dipengaruhi oleh : hubungannya dengan task/kegiatan lain, task itu sendiri dan back stack.


23:15
8,31K/s
71%
LatihanActivity
BUKA ACTIVITY LAIN
TUTUP ACTIVITY

Implementing the lifecycle callbacks

```java
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

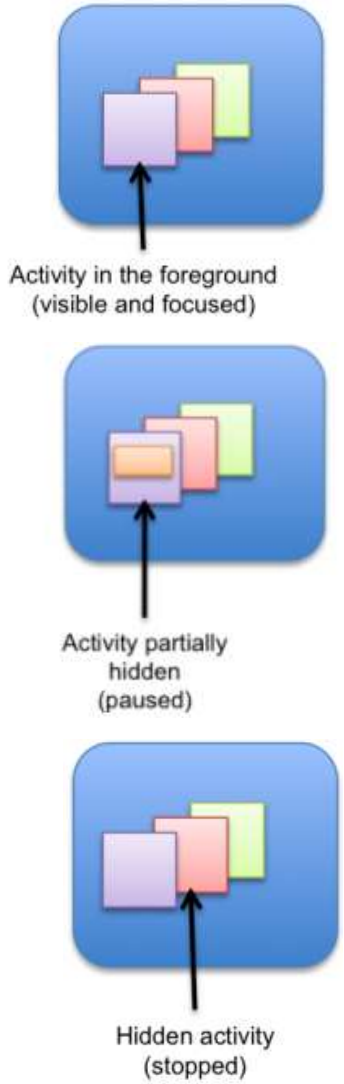

Java – Activity/src/dedepradana/org/MainActivity.java – Eclipse SDK – /Users/dec0de/Documents/NewSkripsweat
Java
Debug
LogCat
Saved Filters
All messages (no filte
dedepradana.org (Ses
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit s
verbose
Lev Time PID Application Tag Text
E 06-25 05:46:48.507 369 dedepradana.org life cycle : activity status : start.
E 06-25 05:46:48.507 369 dedepradana.org life cycle : activity status : resume.
E 06-25 05:46:54.958 369 dedepradana.org life cycle : activity status : pause.
E 06-25 05:46:55.127 369 dedepradana.org life cycle : activity status : resume.
W 06-25 05:47:08.128 369 dedepradana.org KeyCharacter… No keyboard for id 0
W 06-25 05:47:08.128 369 dedepradana.org KeyCharacter… Using default keymap: /system/usr/k
E 06-25 05:47:08.338 369 dedepradana.org life cycle : activity status : pause.
E 06-25 05:47:14.847 369 dedepradana.org life cycle : activity status : start.
E 06-25 05:47:14.867 369 dedepradana.org life cycle : activity status : resume.
E 06-25 05:47:45.197 369 dedepradana.org life cycle : activity status : pause.
E 06-25 05:47:45.747 369 dedepradana.org life cycle : activity status : stop.
E 06-25 05:47:45.747 369 dedepradana.org life cycle : activity status : destroy.
E 06-25 05:47:45.827 369 dedepradana.org life cycle : activity status : stop.
Sign in to Google...

4 State Activity

Status	Deskripsi
Running	Activity terlihat dan dapat berinteraksi dengan pengguna
Paused	Activity masih terlihat, tapi tidak dapat berinteraksi dengan pengguna
Stopped	Activity sudah tidak terlihat tapi masih ada di memory
Killed	Activity sudah tidak terlihat dan dihapus dari sistem karena kebutuhan memory atau method `finish()` dipanggil.

4 State Activity

- Jika suatu Activity berada di foreground dari layar (pada bagian teratas dari Stack), dia adalah aktif dan sedang Running.
- Jika suatu Activity kehilangan fokus tetapi masih Visible (Tidak fullscreen atau Ada activity transparant lain diatasnya), maka dia sedang dalam kondisi pause. Suatu pause Activity masih hidup (tetap menangani state dan informasi serta tetap berada pada Window Manager), tetapi tetap dapat di-Kill oleh sistem jika berada dalam situasi kekurangan memori.
- Jika suatu Activity benar2 tidak kelihatan oleh aplikasi lainnya, dia adalah berada pada kondisi Stop, dia tetap mempertahankan semua state dan informasi tetapi tidak Visible bagi pemakai, dan akan di-Kill oleh sistim jika berada dalam situasi kekurangan memori.
- Jika suatu activity dalam kondisi Pause atau Stop, sistem dapat meng-Kill Activity tersebut dari memori dengan meminta dia untuk selesai, atau dengan membuangnya dari memori, ketika dia ditampilkan kembali ke user, dia harus benar-benar mulai dari awal dan mengembalikan dirinya ke state sebelumnya.



onCreate()

- Fungsi ini dipanggil saat Activity dibuat.
- Biasanya fungsi ini digunakan untuk Mengembalikan Save State, menggambar user interface, Inisialisasi state awal Activity, Menghubungkan UI Element kepada Action Code (seperti OnClick, dll)
- Killable after? No!! bila Activity terus berjalan, sistem akan memanggil fungsi callback **onStart()** dilanjutkan dengan **onResume()**. Setelah itu maka Activity akan berada dalam keadaan *running* dan user dapat berinteraksi dengan Activity yang dibuat.
- Always followed by onStart().

onRestart()

- Called after the activity has been stopped, just prior to it being started again.
- Killable after? No
- Always followed by onStart()

onStart()

- Called just before the activity becomes visible to the user.
- Killable after? No
- Followed by onResume() if the activity comes to the foreground, or onStop() if it becomes hidden.

onResume()

- Called just before the activity starts interacting with the user. At this point the activity is at the top of the activity stack, with user input going to it.
- Killable after? No
- Always followed by onPause().

onPause()

Merupakan event yang dipanggil ketika running Activity beralih ke activity lainnya.

Activity pada State ini dapat saja di-Kill (dalam keadaan kekurangan memori)

Jika suatu activity di-Kill, maka tidak terjadi event onStop dan onDestroy.

- Killable after? Yes
- Followed either by onResume() if the activity returns back to the front, or by onStop() if it becomes invisible to the user.

Quiz

Perhatikan Gambar berikut ini, tunjukan mana Activity yang dalam kondisi Pause.

Activity A

Activity B

onStop()

- Called when the activity is no longer visible to the user. This may happen because it is being destroyed, or because another activity (either an existing one or a new one) has been resumed and is covering it.
- Killable after? Yes
- Followed either by onRestart() if the activity is coming back to interact with the user, or by onDestroy() if this activity is going away.

onDestroy()

- Called before the activity is destroyed.
- This is the final call that the activity will receive.
- It could be called either because the activity is finishing (someone called finish() on it), or because the system is temporarily destroying this instance of the activity to save space.
- Killable after? Yes
- You can distinguish between these two scenarios with the isFinishing() method.

Quiz

Ketika suatu Activity di LAUNCH, tentukan urutan dari pengaktifan Method:

a. onCreate, onStart, onRestart

b. onCreate, onPause, onStart

c. onCreate, onRestart, onStart

d. onCreate, onStart, onResume

e. onCreate, onResume, onStart

Thank you!