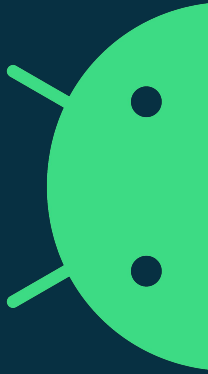


Lab.

Pemrograman Mobile



Learning Objectives

- Activity
- Events



Pert. 3

Activity

Activity

Komponen yang menampilkan *user interface* ke layar pengguna yang berisi informasi dan interaksi

Eg. dashboard Jenius, UI Chat Whatsapp.

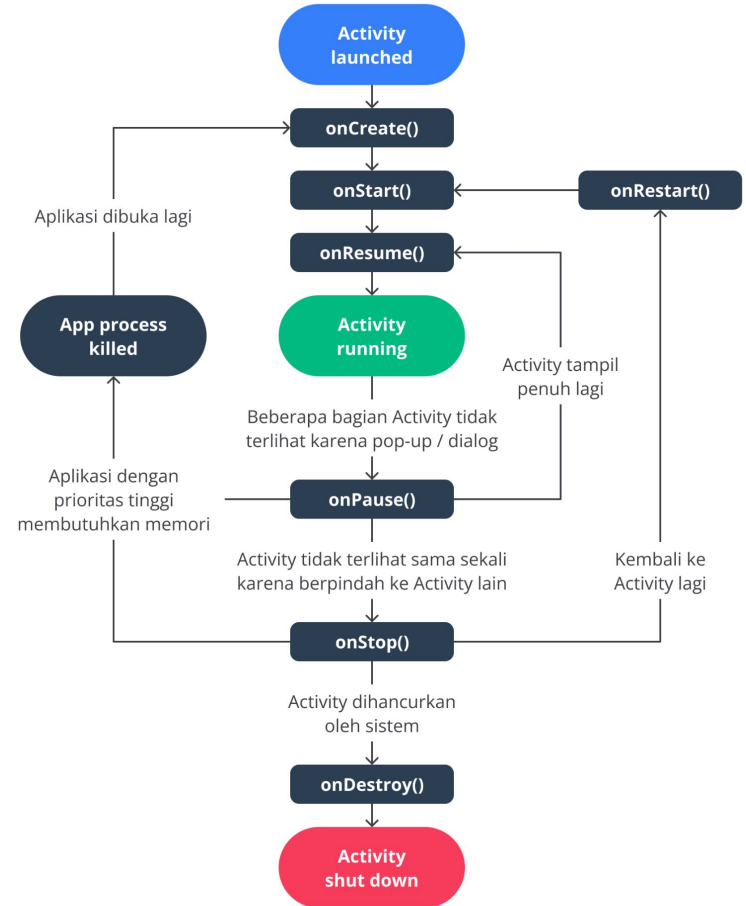
Related files :

MainActivity.kt

Extensi dari superclass activity, mengelola code, dan menampilkan layout

activity_main.xml

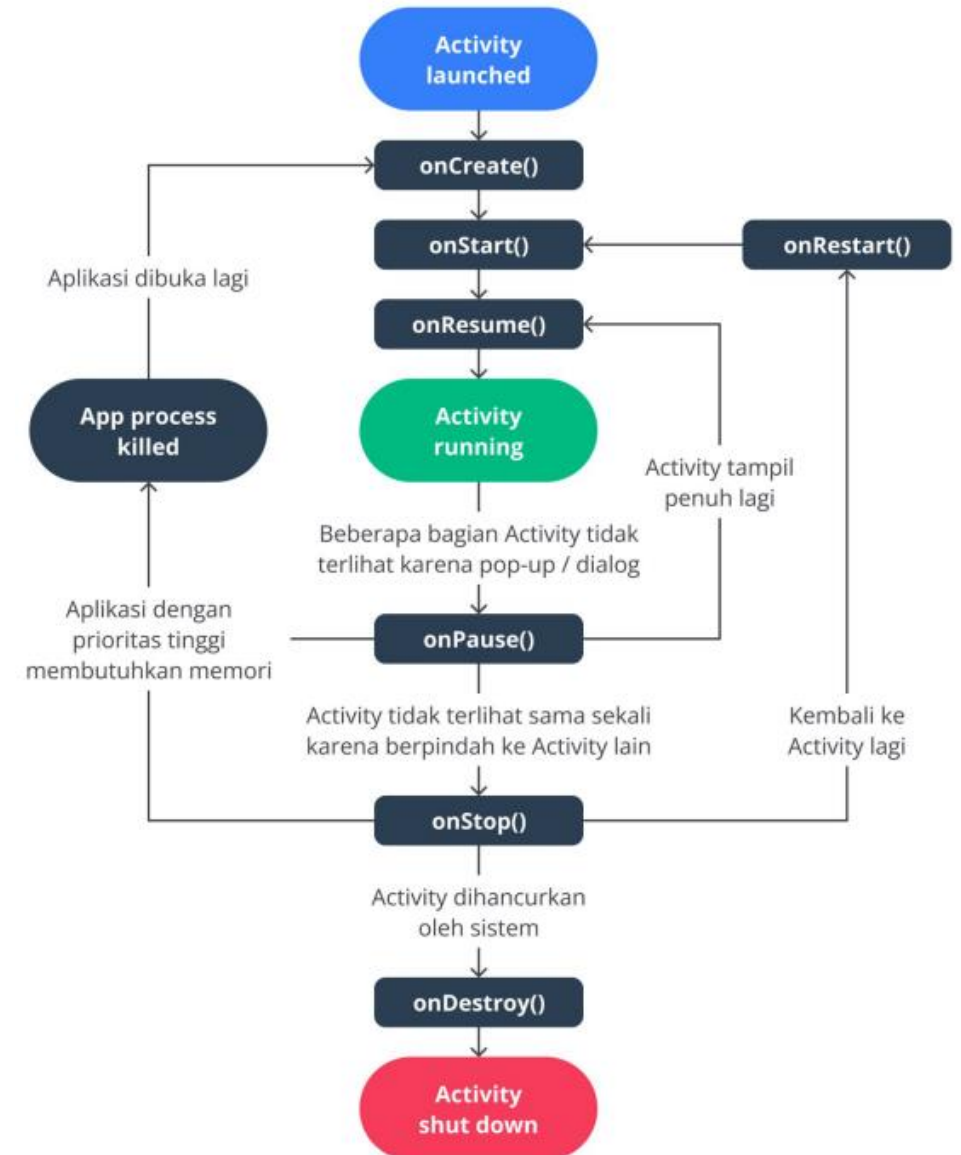
File layout yang akan ditampilkan activity



State onCreate yang otomatis tergenerate di activity main

```
import ...  
  
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

- **onCreate()** adalah kondisi awal saat Activity baru diciptakan, biasanya dilakukan inisialisasi pada tahapan ini.
- **onStart()** adalah saat Activity dimulai
- **onResume()** adalah saat Activity dibuka kembali, biasanya dieksekusi setelah onPause()
- **onPause()** akan dipanggil saat ada Activity lain yang terbuka.
- **onStop()** adalah kondisi saat Activity tidak ditampilkan dilayar (biasanya saat pengguna menekan tombol Home).
- **onRestart()** adalah kondisi saat Activity kembali dibuka oleh pengguna.
- **onDestroy()** adalah kondisi saat Activity dihancurkan pada memori.



Lets see what happens when [activity](#) going through deferents state

- **When open the app**
onCreate() → onStart() → onResume()
- **When home button pressed**
onPaused() → onStop()
- **After pressed home button when again open app from recent task list or clicked on app icon**
onRestart() → onStart() → onResume()
- **When open app another app from notification bar or open settings**
onPaused() → onStop()
- **Back button pressed from another app or settings then used can see our app**
onRestart() → onStart() → onResume()
- **When any dialog open on screen**
onPause()

- **After dismiss the dialog or back button from dialog**
onResume()
- **Any phone is ringing and user in the app**
onPause() → onResume()
- **When user pressed phone's answer button**
onPause()
- **After call end**
onResume()
- **When phone screen off**
onPaused() → onStop()
- **When screen is turned back on**
onRestart() → onStart() → onResume()

Extra Points!

Apa Sifat *Activity Life Cycle*?

**Kenapa *Activity Life Cycle* Penting
diketahui oleh Android Developer?**

Saving Activity State

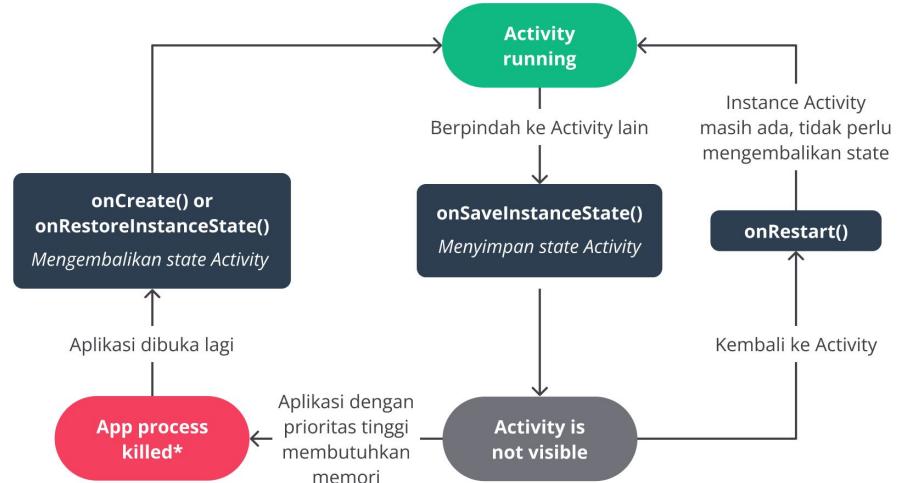
Terjadi saat *pause* dan *resume*, disimpan di memori device.

Ketika sistem menghancurkan activity untuk keperluan memori e.g. memori habis, obyek activity dihancurkan.

Ketika activity ingin ditampilkan kembali diperlukan proses re-*create* activity yang dihancurkan tadi.

onSaveInstanceState()

memiliki bundle yang dapat digunakan untuk menyimpan informasi. Ketika activity di-*restart*, bundle akan diberikan kepada metode onCreate dan onRestoreInstanceState. Bundle tersebut akan dimanfaatkan untuk mengembalikan kembali perubahan yang telah terjadi sebelumnya.



*Activity dihancurkan, namun state dari onSaveInstanceState tersimpan

Extra Points!

**Berikan contoh keadaan
penghancuran activity yang
mentrigger *OnDestroy()***

Pert. 3

Event

Event

Event adalah cara yang berguna untuk mengumpulkan data tentang interaksi pengguna dengan komponen app yang dibangun

- **Event Listeners** – Event listener adalah interface di class View yang berisi callback. Metode ini akan dipanggil oleh framework saat interface yang telah diassign listener dipicu oleh interaksi pengguna dengan item di UI.

- **Event Listeners Registration** – Event Registration adalah proses dimana Event Handler didaftarkan dengan Event Listener sehingga handler dipanggil saat Event Listener menjalankan event.
- **Event Handlers** – Ketika sebuah event terjadi dan kita telah mendaftarkan event listener untuk event tersebut, event listener memanggil Event Handler, yang merupakan metode yang benar-benar menangani event tersebut.

Make new project > Go to your Layout > Change to Linear Layout

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="16dp"
android:orientation="vertical">
...
...
</LinearLayout>
```

We will talk about layout in the next meeting

Adding Textviews and Edittext forms

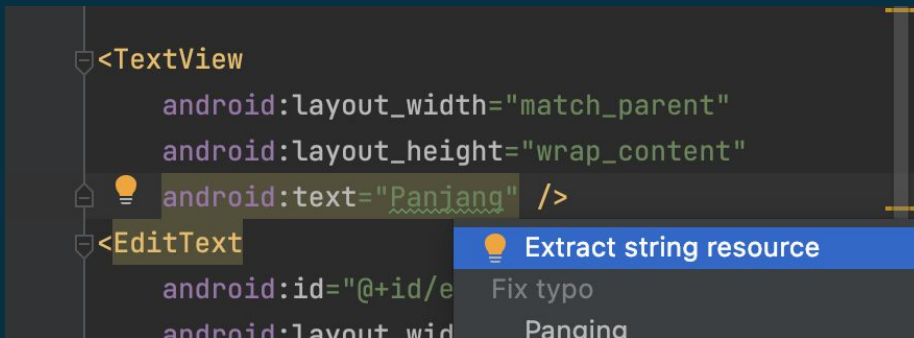
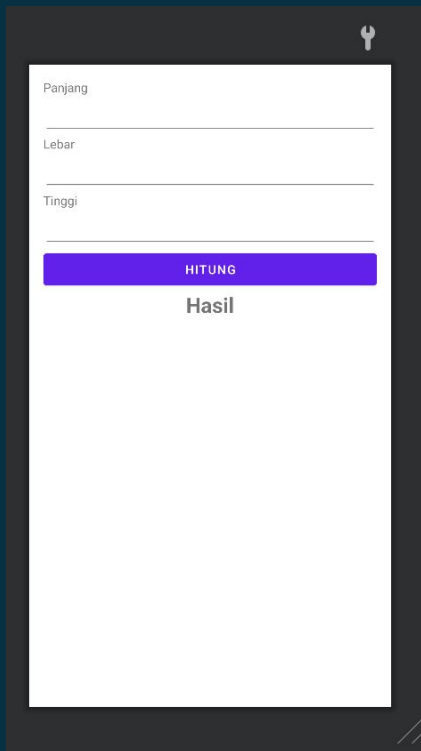
```
...
<TextView
    android:layout width="match parent"
    android:layout height="wrap_content"
    android:text="Panjang" />

<EditText
    android:id="@+id/edt_length"
    android:layout width="match parent"
    android:layout height="wrap content"
    android:inputType="numberDecimal"
    android:lines="1" />

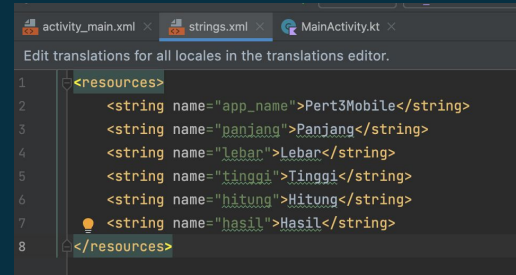
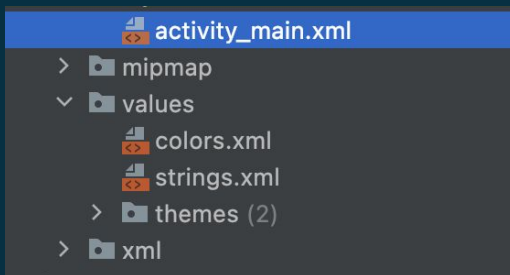
...
```

Adding Textviews for result and Button

```
...  
    <Button  
        android:id="@+id/btn_calculate"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="Hitung" />  
    <TextView  
        android:id="@+id/tv_result"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:gravity="center"  
        android:text="Hasil"  
        android:textSize="24sp"  
        android:textStyle="bold" />  
</LinearLayout>
```

Extract your strings to value resource



Until it's look like this

Adding Logics : define variables

```
...  
private lateinit var edtWidth: EditText  
private lateinit var edtHeight: EditText  
private lateinit var edtLength: EditText  
private lateinit var btnCalculate: Button  
private lateinit var tvResult: TextView  
...
```

Extra Points!

Apa beda *var* dan *val* dalam pemrograman android kotlin?

Adding Logics : define variables

```
class MainActivity : AppCompatActivity() {  
    ...  
    private lateinit var edtWidth: EditText  
    private lateinit var edtHeight: EditText  
    private lateinit var edtLength: EditText  
    private lateinit var btnCalculate: Button  
    private lateinit var tvResult: TextView  
    ...  
}
```

Adding Logics : assigning variables to View values

```
...
setContentView(R.layout.activity_main)

edtWidth = findViewById(R.id.edt_width)
edtHeight = findViewById(R.id.edt_height)
edtLength = findViewById(R.id.edt_length)
btnCalculate = findViewById(R.id.btn_calculate)
tvResult = findViewById(R.id.tv_result)

//listener registration
btnCalculate.setOnClickListener(this)
}
```

```
btnCalculate.setOnClickListener(this)
```



Create extension function 'Button?.setOnClickListener'



Let 'MainActivity' implement interface 'View.OnClickListener'



Add names in comment to call arguments



**Don't forget to
implement the
interface**

Adding Logics : adding event handler

```
...  
override fun onClick(p0: View) {  
    if (p0.id == R.id.btn_calculate) {  
        val inputLength = edtLength.text.toString().trim()  
        val inputWidth = edtWidth.text.toString().trim()  
        val inputHeight = edtHeight.text.toString().trim()  
        val volume = inputLength.toDouble() * inputWidth.toDouble() *  
inputHeight.toDouble()  
        tvResult.text = volume.toString()  
    }  
}
```

Bonus : simple data validation

```
//validation data
var isEmptyFields = false
if (inputLength.isEmpty()) {
    isEmptyFields = true
    edtLength.error = "Field is empty"
}
if (inputWidth.isEmpty()) {
    isEmptyFields = true
    edtWidth.error = "Field ini tidak boleh kosong"
}
...
```

Saving Activity State : Declaring object constant(s), assign state

```
//adding state result parameter saver
companion object {
    private const val STATE_RESULT = "state_result"
}
...
...
//saving state when activity destroyed e.g. flip screen
override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)
    outState.putString(STATE_RESULT, tvResult.text.toString())
}
```

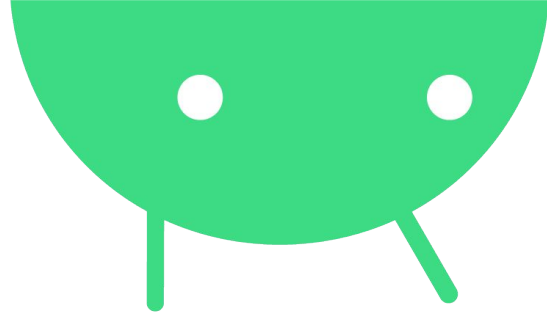

Saving Activity State : access constant(s) if state is not null

```
//assigning state if not null
if (savedInstanceState != null) {
    val result = savedInstanceState.getString(STATE_RESULT)
    tvResult.text = result
}
```

Next Learning Objectives

- Explicit Intent
- Implicit Intent
- Layout(s)





Kirim pertanyaan terkait materi via email :

Subjek : [Lab Mobile C#] <<pertanyaan>>

Email : rasyidhafiz@students.usu.ac.id

Any Question?