



Activity Lifecycle dan Konep Layout

Activity

Activity berfungsi sebagai titik masuk bagi pengguna seperti window pada aplikasi dekstop. Setiap activity mempunyai layout yang memiliki sebuah antarmuka pengguna (user interface) untuk berinteraksi dengan pengguna.

Bingkai: Activity yang menampilkan layar aplikasi kita.

Foto: Desain tampilan aplikasi yang bisa kita sesuaikan.



Activity

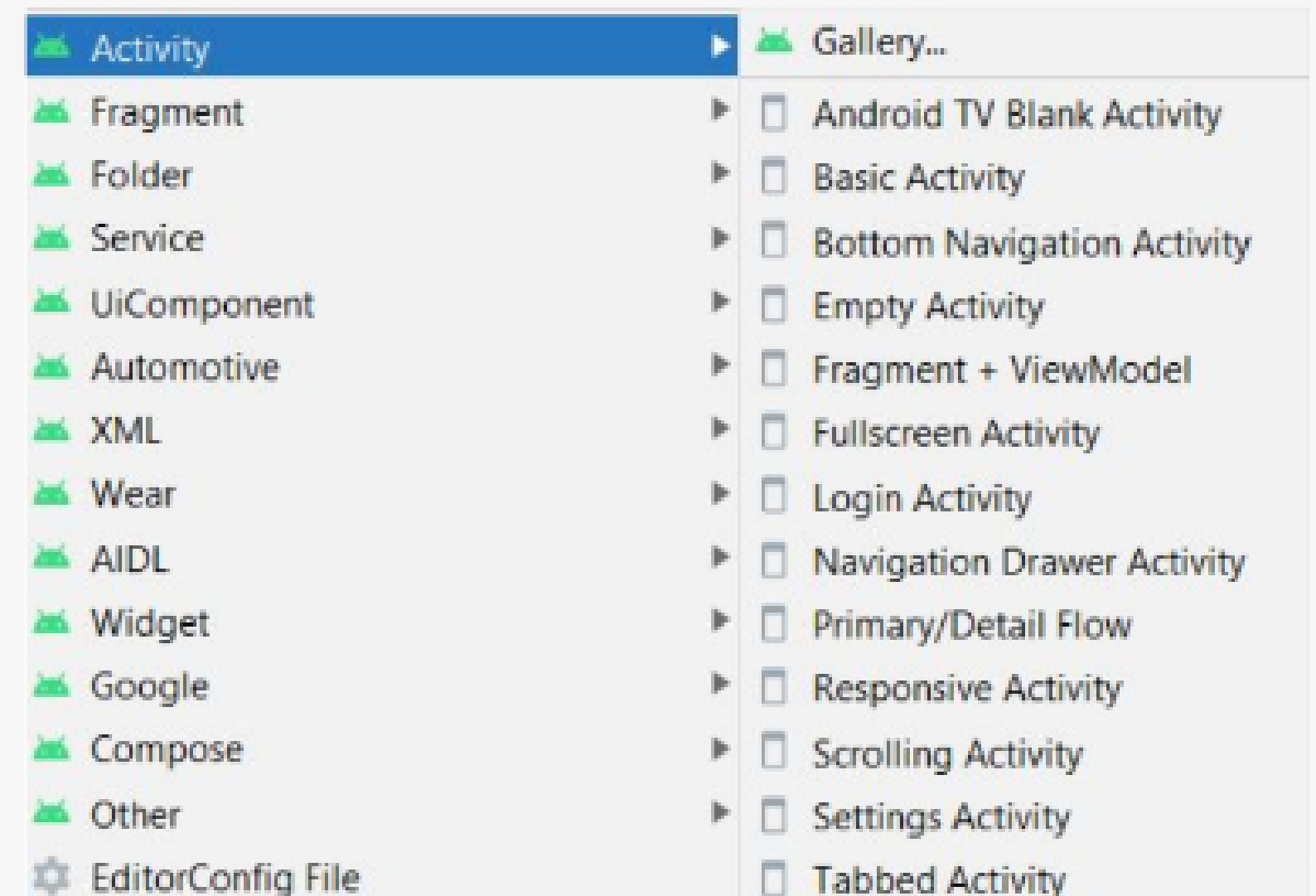
Activity terdaftar pada Android Manifest.

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

Activity

Cara membuat activity :

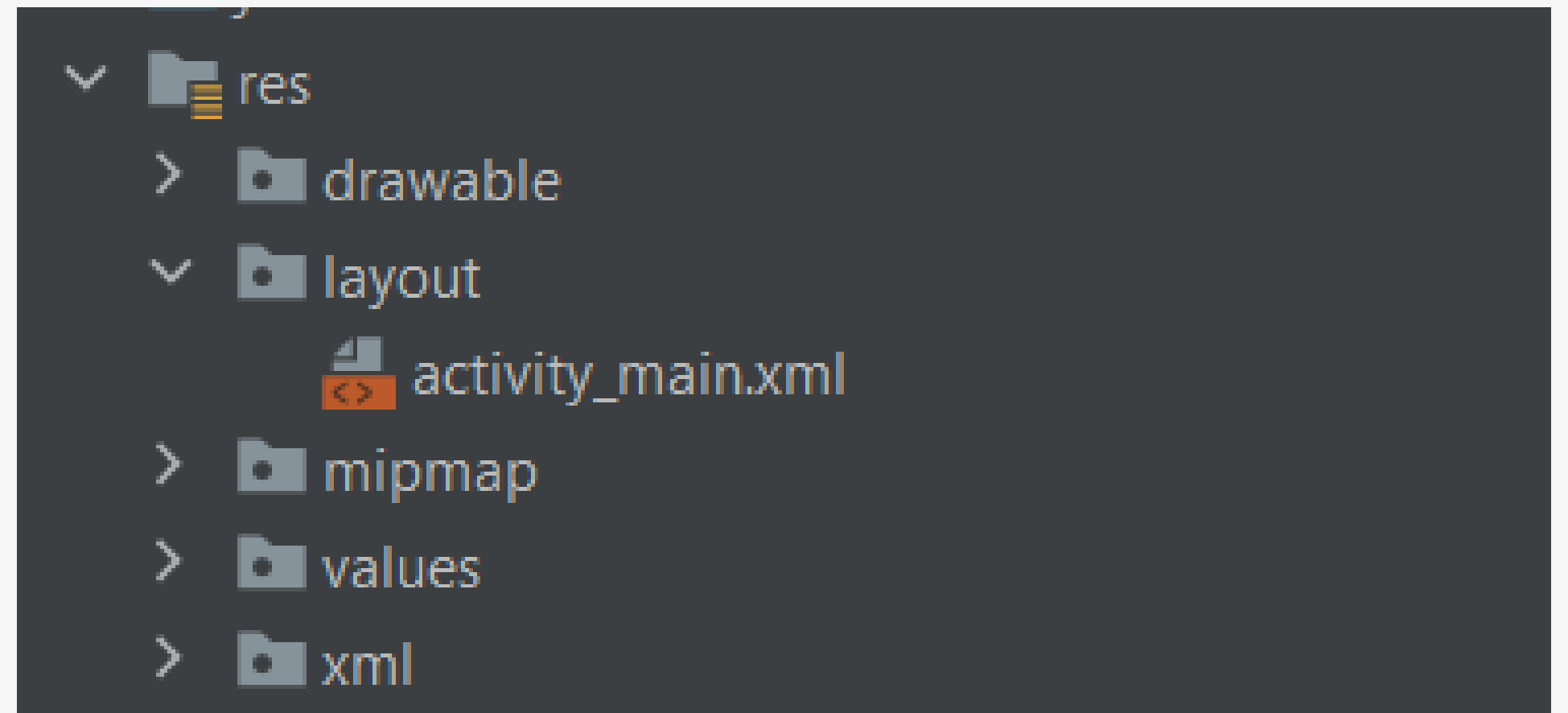
- Klik kanan pada package tempat Activity baru akan dibuat
- Arahkan ke New
- Arahkan ke Activity



Activity

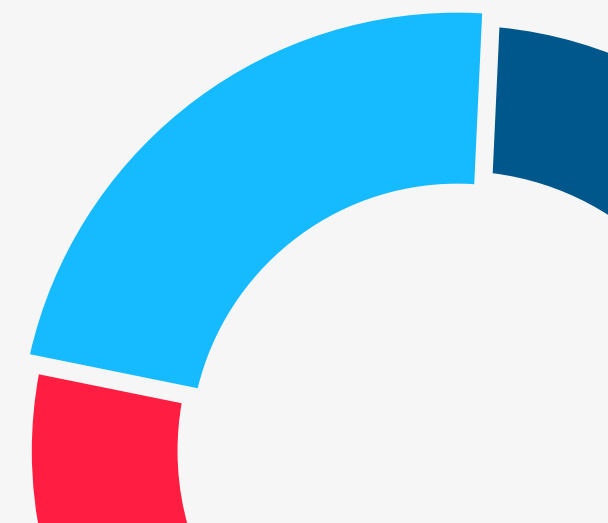
Kita bisa mengedit tampilan dari activity tersebut melalui file bernama activity.xml

Kita bisa mengakses file tersebut pada folder :
Res > Layout > Pilih activity xml





Activity Lifecycle



Activity Lifecycle

Aplikasi juga memiliki lifecycle. Mulai dari activity dibuat sampai activity itu tidak digunakan atau mati.

Contoh lifecycle :

Telur ----- Ulat ----- Kepompong ----- Kupu - Kupu
on Create on Start on Destroy

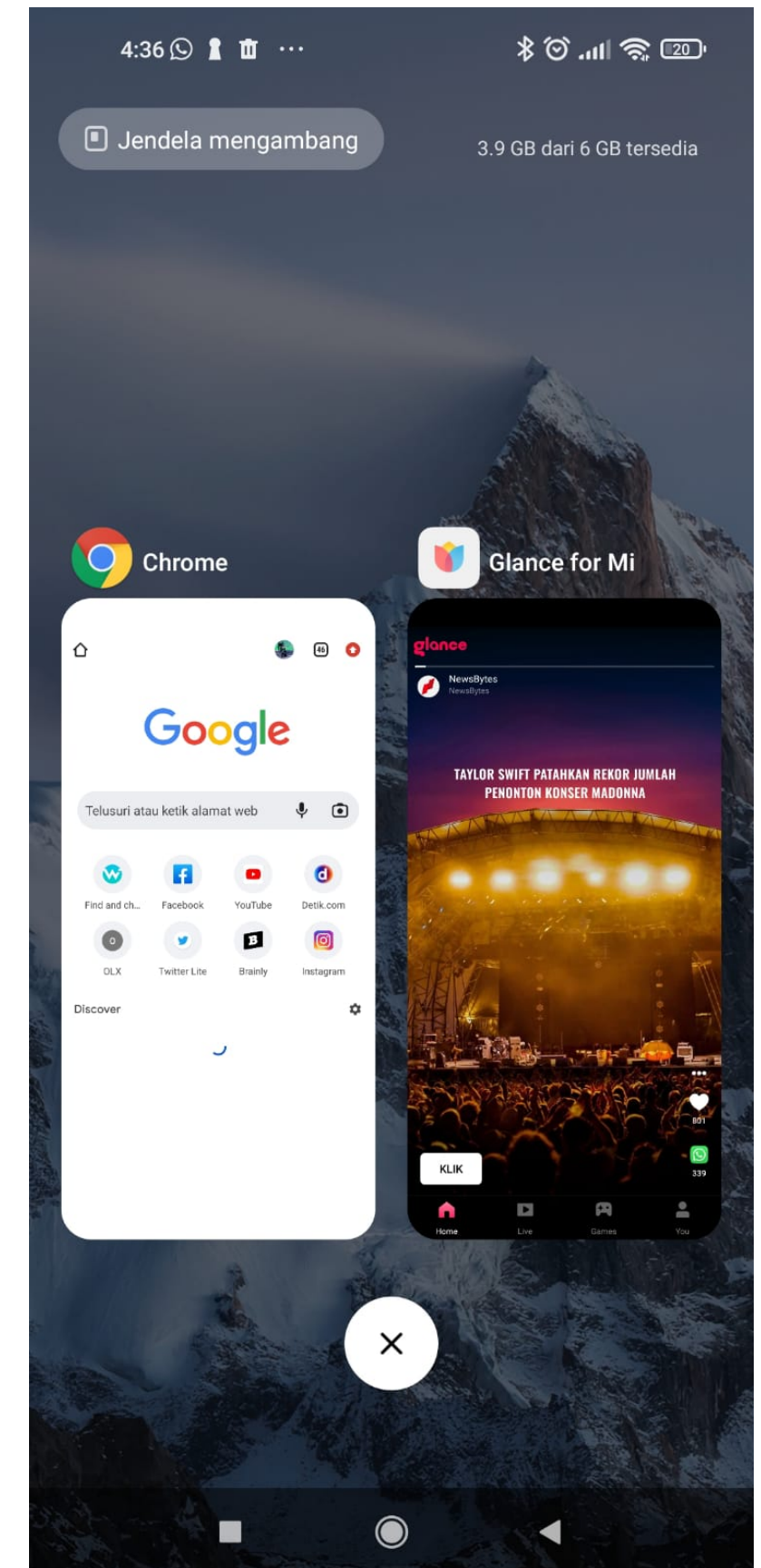
Activity Lifecycle

Kondisi 1

Activity berada pada *foreground* (pada posisi teratas dari tumpukan)

Proses yang Terjadi

- Activity akan aktif atau sedang berjalan.
- Biasanya Activity yang berada pada foreground merupakan Activity yang sedang berinteraksi dengan pengguna.



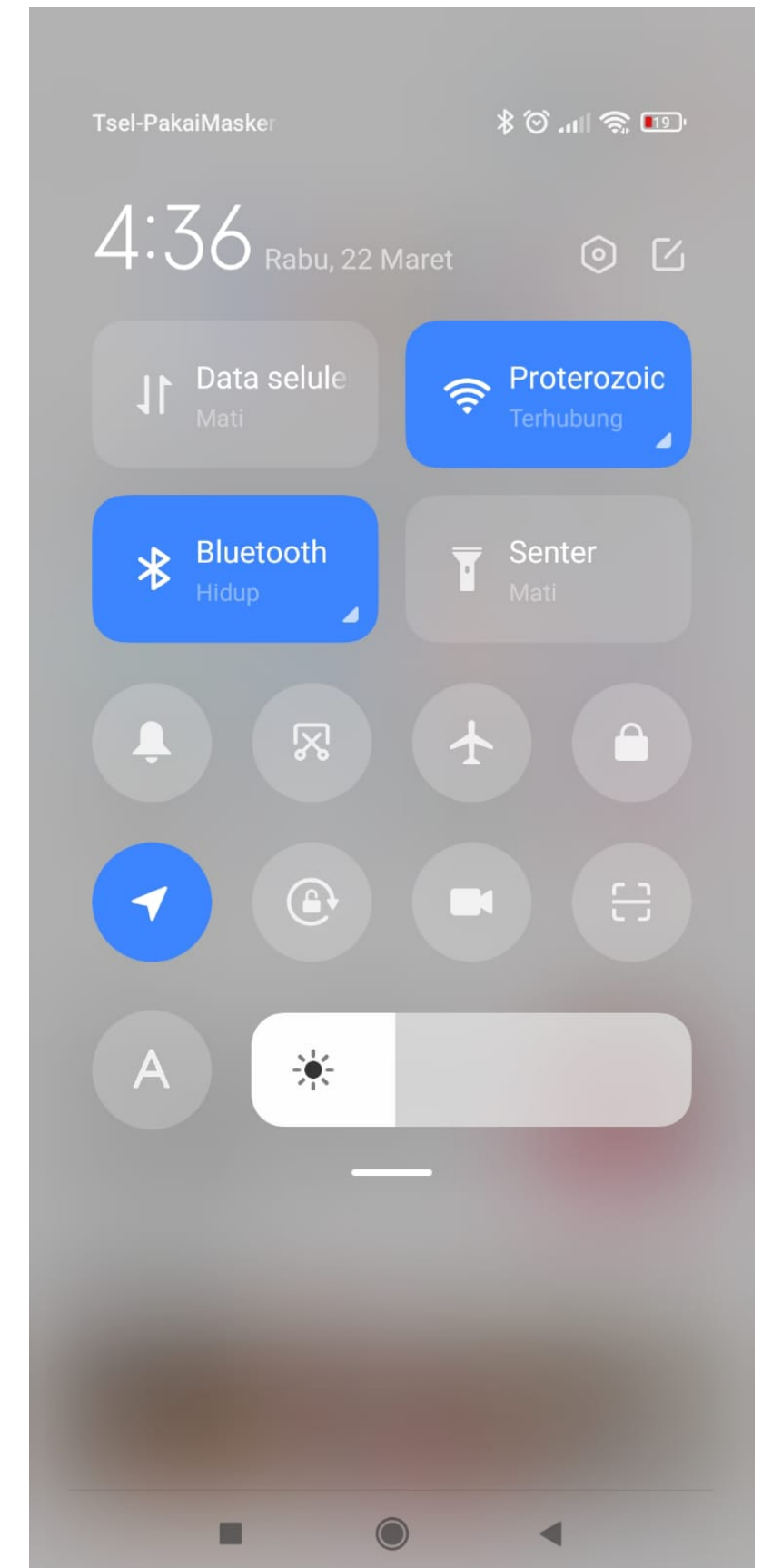
Activity Lifecycle

Kondisi 2

Activity telah kehilangan fokus tetapi masih tampil pada pengguna.

Proses yang Terjadi

- Biasanya Activity tersebut sebelumnya tidak tampil secara utuh atau sebuah Activity transparan memiliki fokus di atas Activity kita.
- Activity lain memiliki posisi lebih tinggi dalam mode multi-window atau Activity itu sendiri tidak fokus dalam mode window saat itu.
- Activity tersebut benar-benar berjalan dan mempertahankan semua informasi dan tetap melekat pada window manager.



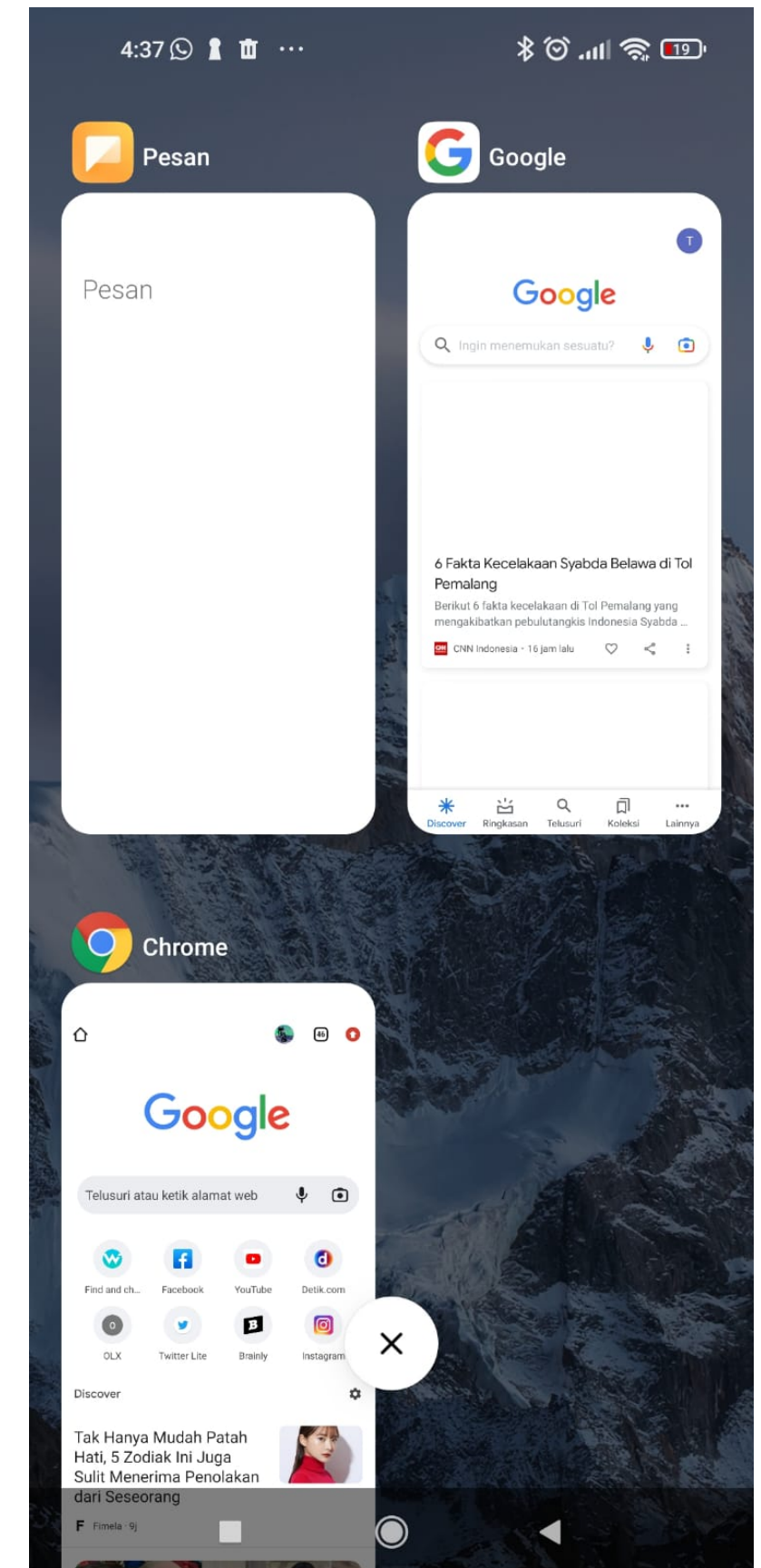
Activity Lifecycle

Kondisi 3

Activity sepenuhnya ditimpa oleh Activity lain.

Proses yang Terjadi

- Activity akan dihentikan atau disembunyikan.
- Activity tersebut masih menyimpan semua informasi kondisi dan anggotanya.
- Namun, tampilannya tidak lagi terlihat oleh pengguna dan akan **dihancurkan otomatis** oleh sistem jika memori diperlukan di tempat lain.



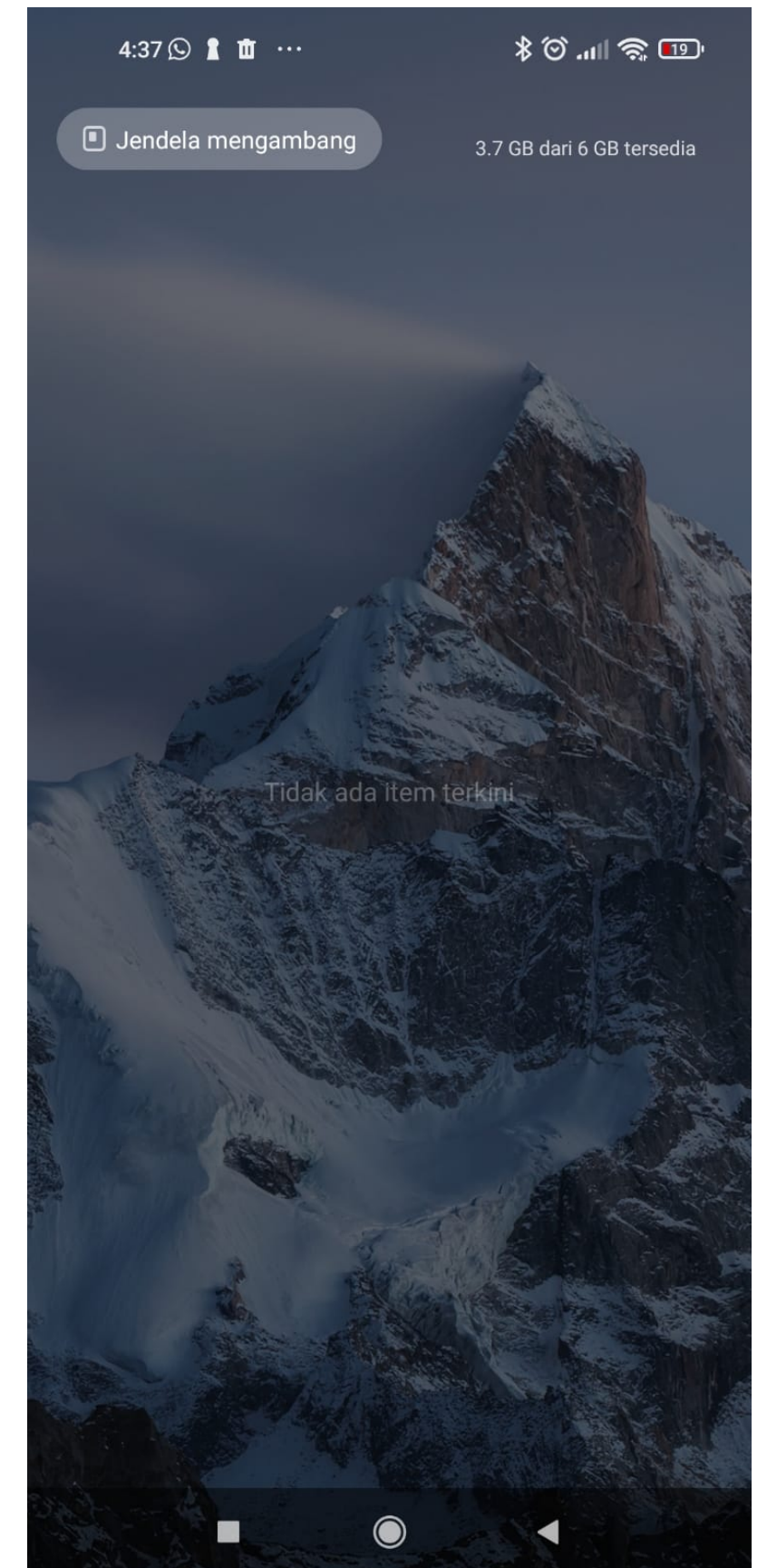
Activity Lifecycle

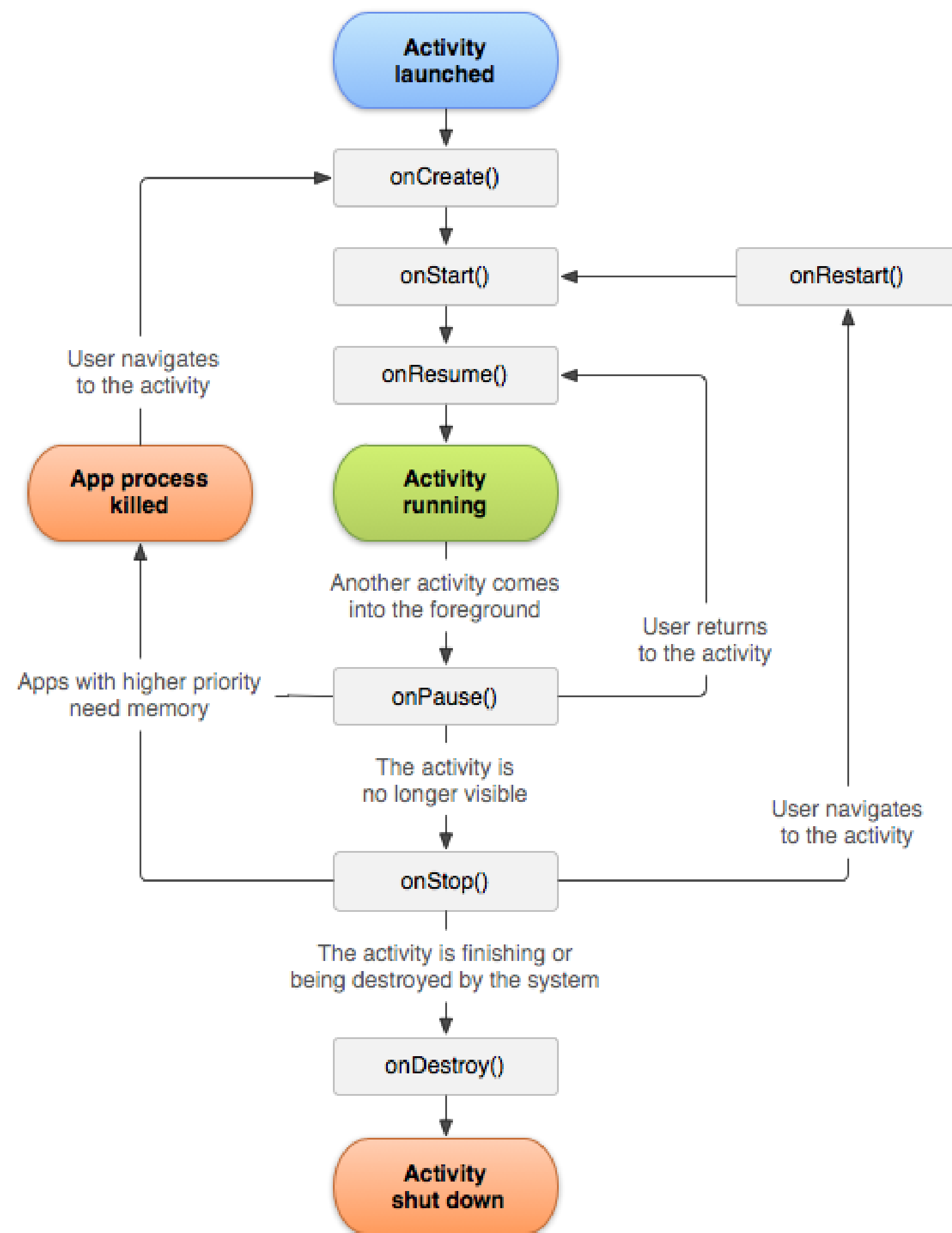
Kondisi 4

Activity dihancurkan atau dimatikan prosesnya oleh sistem

Proses yang Terjadi

- Sistem dapat menghilangkan Activity dari memori dengan meminta untuk menghancurkannya, atau hanya mematikan prosesnya.
- Ketika itu ditampilkan lagi kepada pengguna, itu harus sepenuhnya dimulai kembali dan dikembalikan ke kondisi sebelumnya.





Activity Lifecycle

Dari diagram lifecycle tersebut, ada 3 nested loop pada Android lifecycle :

- Entire Lifetime
- Visible Lifetime
- Foreground Lifetime

Activity Lifecycle

Entire Lifetime

Merupakan masa hidup (lifetime) saat method `onCreate()` dipanggil hingga method `onDestroy()` dipanggil. Sebuah Activity harus menyiapkan semua sources seperti User Interfaces di `onCreate()` dan melepaskan semua resources yang tersisa di `onDestroy()`. Contohnya, jika mempunyai sebuah thread berjalan di background untuk mengunduh data, itu dapat dibuat di `onCreate()` dan kemudian dihentikan di `onDestroy()`

Activity Lifecycle

Visible Lifetime

Merupakan masa hidup sebuah Activity antara panggilan method `onStart()` hingga ke `onStop()`. Pada waktu ini, pengguna dapat melihat Activity yang tampil pada layar dan bisa berinteraksi dengan pengguna tersebut. Kalau Activity lain dipanggil dan berada di foreground, maka `onStop()` bakal dipanggil dalam Lifecycle dan membuat Activity yang saat ini terlihat jadi nggak terlihat lagi.

Selama Activity berada di waktu Visible Lifetime, method `onStart()` dan `onStop()` biasanya akan dipanggil beberapa kali sehingga sebuah Activity berubah antara terlihat dan disembunyikan dari pengguna.

Activity Lifecycle

Foreground lifetime

Merupakan masa hidup Activity yang terjadi antara panggilan method `onResume()` hingga ke `onPause()`. Pada waktu ini, Activity akan terlihat dan sering berinteraksi dengan pengguna. Fun fact-nya, suatu Activity bisa sering melakukan transisi masuk dan keluar dari foreground.

Misalnya gini, `onPause()` dipanggil saat perangkat tidur atau ketika dialog muncul. Karena kondisi ini sering terjadi transisi, code dalam kedua method ini harus cukup ringan untuk menghindari transisi lambat yang membuat pengguna menunggu.

Activity Lifecycle

Activity States

| State | Callbacks | Description |
|-----------|-------------|--------------------------------------|
| Created | onCreate() | Activity sedang diinisialisasi. |
| Started | onStart() | Activity terlihat oleh user. |
| Resumed | onResume() | Activity memiliki fokus input. |
| Paused | onPause() | Activity tidak memiliki fokus input. |
| Stopped | onStop() | Activity tidak lagi terlihat. |
| Destroyed | onDestroy() | Activity dihancurkan. |

Activity Lifecycle

`onCreate()`

- Activity dibuat dan proses inisialisasi lainnya terjadi.
- Kamu harus mengimplementasi method ini.
- Inflate activity UI dan menjalankan logika startup aplikasi lainnya.

Activity Lifecycle

onStart()

- Activity menjadi terlihat pada pengguna.
- Method dipanggil setelah activity memanggil onCreate() atau onRestart() jika activity sebelumnya dihentikan.

Activity Lifecycle

onResume()

- Activity mendapatkan fokus input.
 - Pengguna dapat berinteraksi dengan activity tersebut.
- Activity tetap berada dalam kondisi resumed hingga sistem memicu activity untuk di jeda (tobe paused).

Activity Lifecycle

onPause()

- Activity kehilangan fokusnya (tidak berada di foreground).
- Activity masih terlihat namun pengguna tidak dapat berinteraksi dengannya.
- Kebalikan dari onResume().

Activity Lifecycle

OnStop()

- Activity tidak terlihat lagi pada pengguna.
- Melepaskan semua resources yang tidak dibutuhkan lagi.
- Menyimpan semua data persistent state yang masih diproses sehingga pengguna tidak kehilangan datanya.

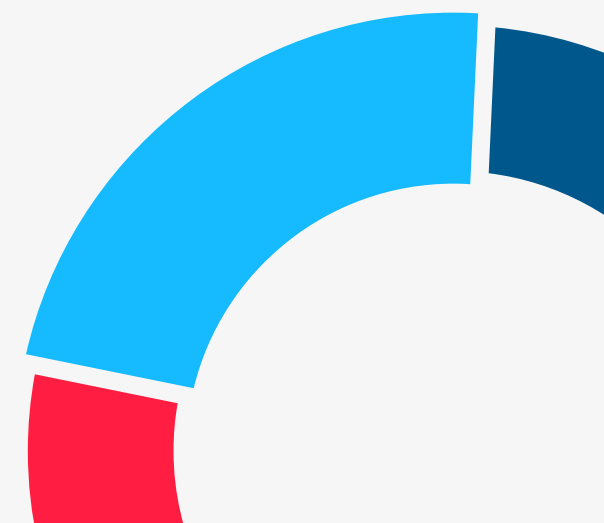
Activity Lifecycle

`onDestroy()`

- Activity akan dihancurkan yang dapat disebabkan oleh:
 - Activity telah selesai atau diberhentikan oleh pengguna.
 - Perubahan konfigurasi.
- Melakukan semua pembersihan terakhir resources.
- Jangan bergantung kepada method ini untuk melakukan penyimpanan data user (lakukan pada method sebelumnya).



Konep Layout



Layout

Jadi, XML itu apa sih?

Android XML adalah format pemrograman dalam bentuk teks yang bisa memberi keleluasaan pada kita dalam mendesain UI layout dan elemen layar lain yang dikandungnya dengan cepat.

Layout

Hierarki Layout

Dalam file xml pada layout harus berisi objek berikut ini :

- Parent / Root
- View
- ViewGroup[

Layout

View

View merupakan komponen yang berada pada aplikasi yang digunakan sebagai class dasar untuk UI di android.

Layout

Root / Parent Layout

Berikut ini adalah beberapa Root atau parent layout yang akan digunakan dalam aplikasi Android:

- LinearLayout
- RelativeLayout
- TableLayout
- FrameLayout
- ConstraintLayout
- WebView
- ListView
- GridView
- ScrollView

Layout

View

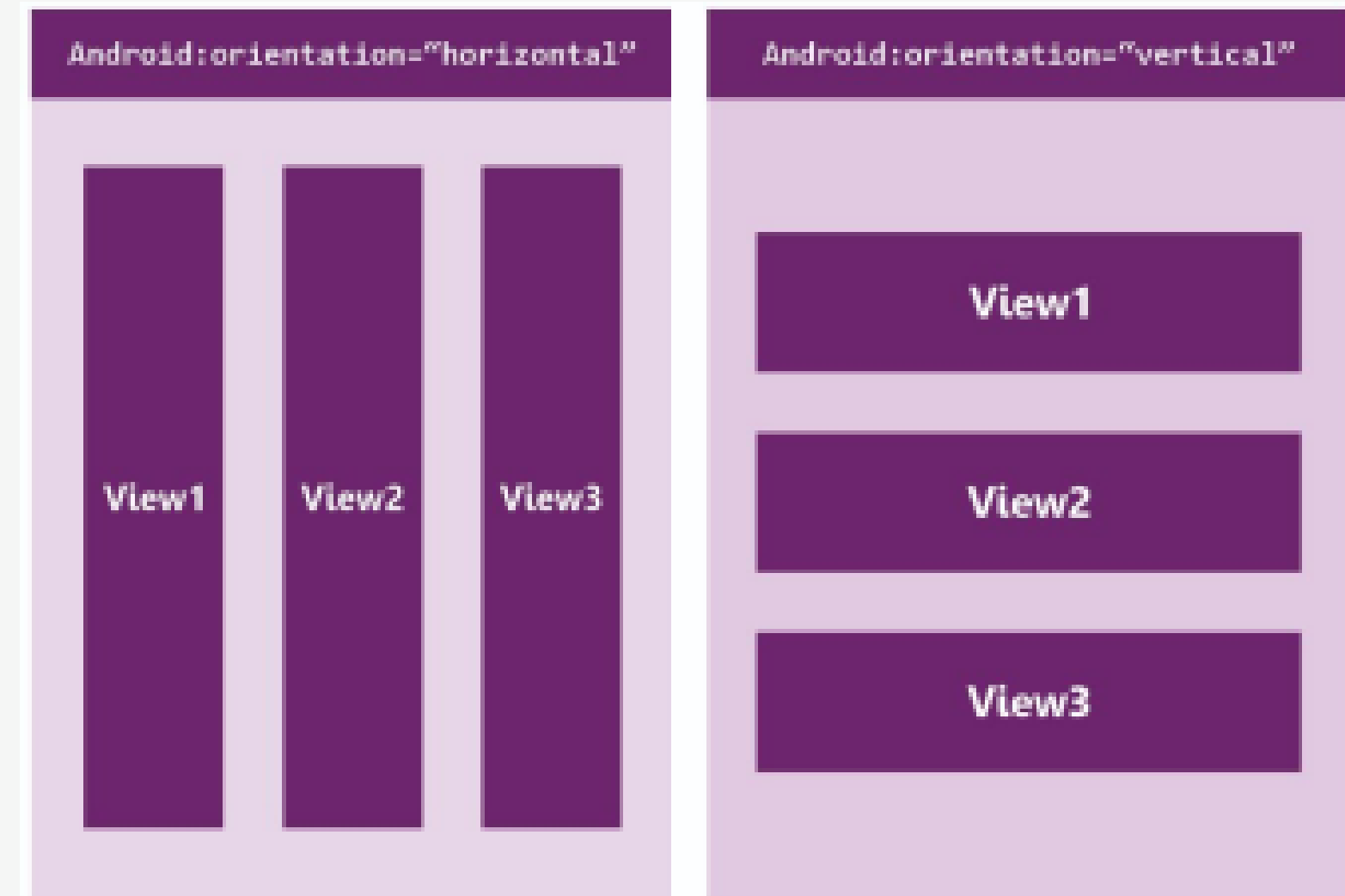
Berikut ini adalah beberapa subclass View umum yang akan digunakan dalam aplikasi Android:

- TextView
- EditText
- Button
- CheckBox
- RadioButton
- ImageButton
- ProgressBar
- Spinner

Layout

LinearLayout

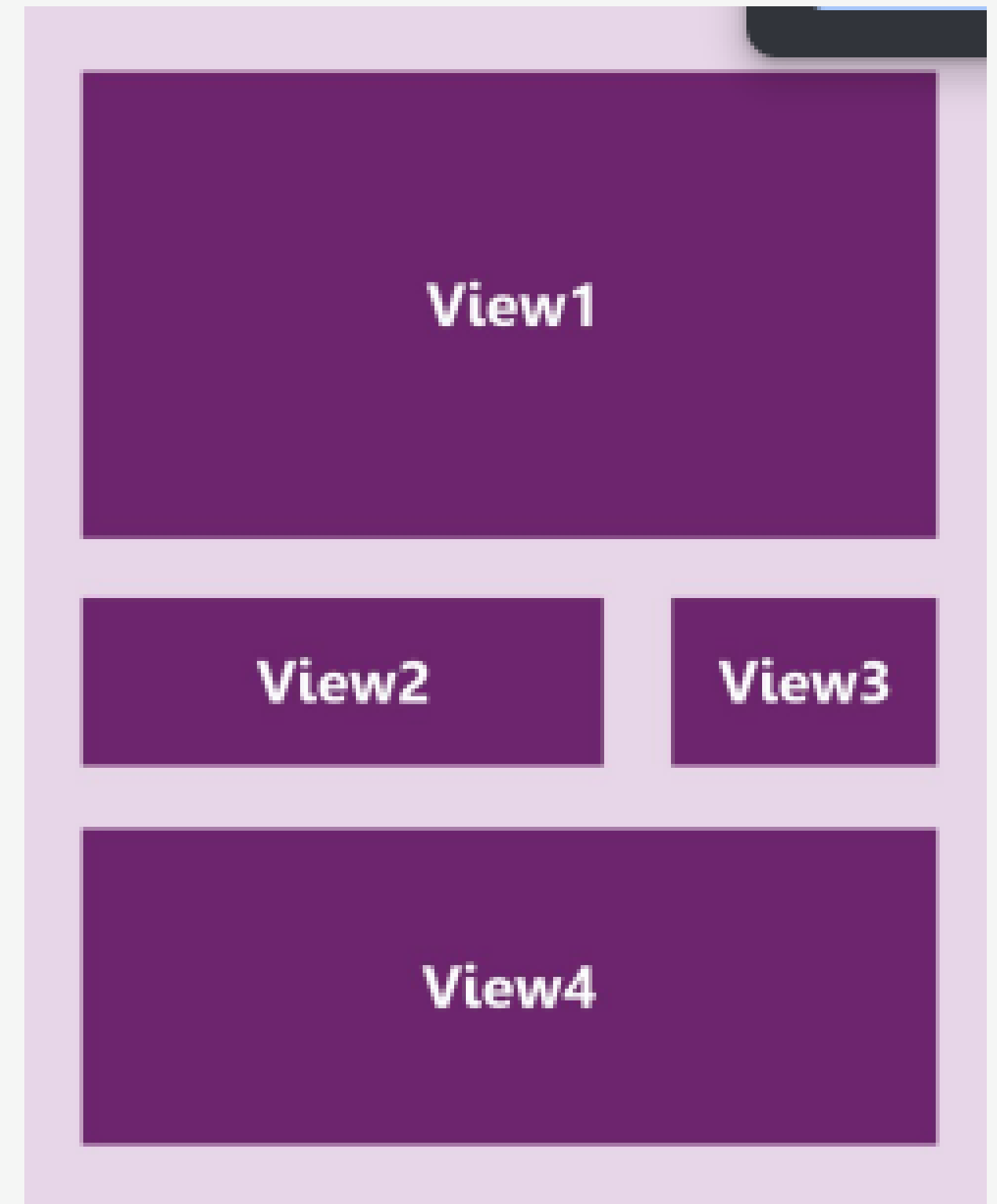
LinearLayout adalah layout yang akan menempatkan komponen-komponen di dalamnya secara berkelompok. Nah, kita dapat memilih apakah layout akan dimuat secara horizontal maupun vertikal. Kita dapat menentukan arah LinearLayout menggunakan attribute `android:orientation`.



Layout

RelativeLayout

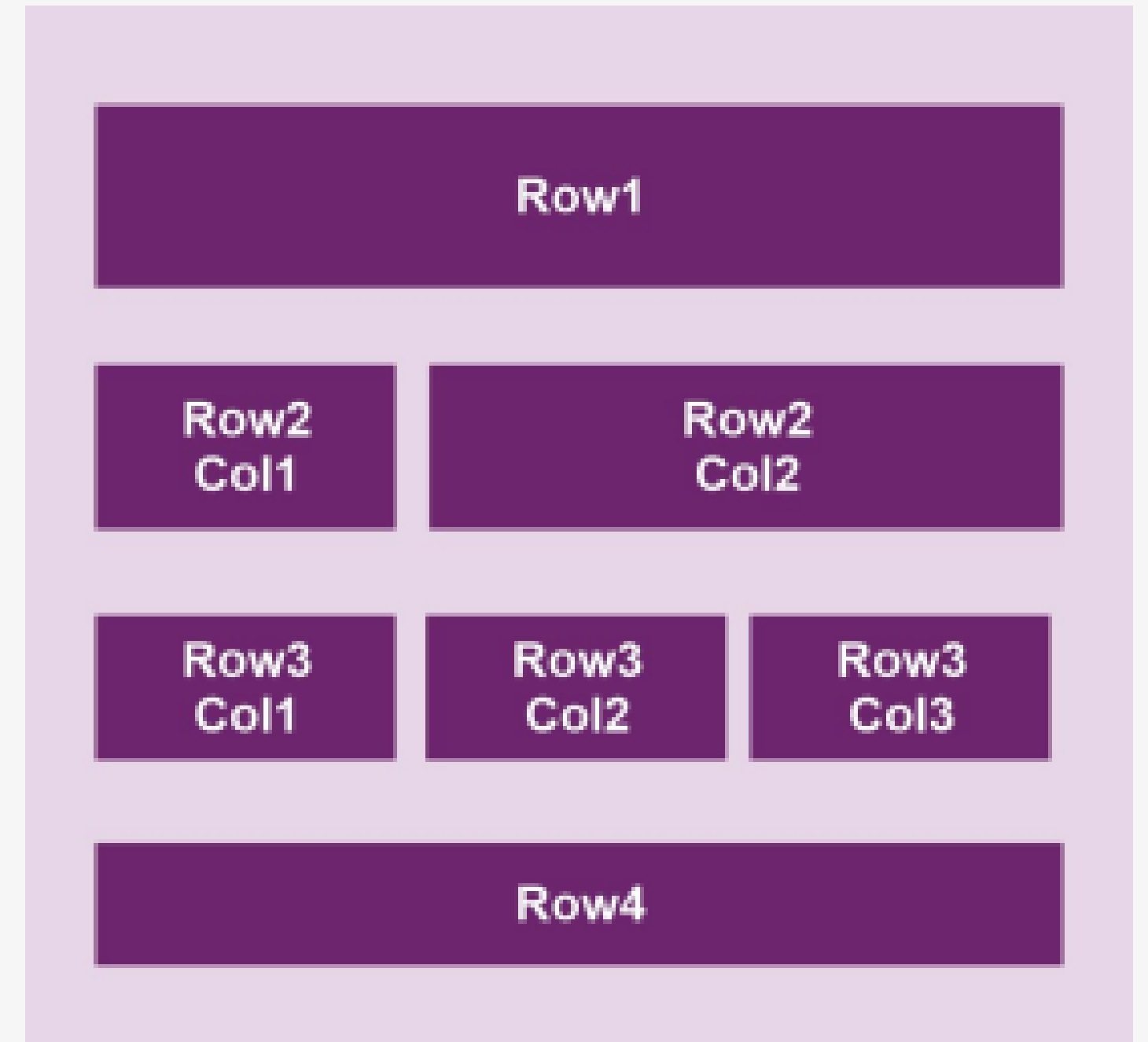
RelativeLayout digunakan untuk menentukan posisi setiap komponen secara relatif terhadap komponen yang lain.



Layout

TableLayout

TableLayout merupakan layout yang digunakan untuk menampilkan komponen dalam baris dan kolom, seperti membuat tabel. Nah, kita bisa menggunakan komponen `<TableRow>` untuk membuat baris di tabel.



Layout

FrameLayout

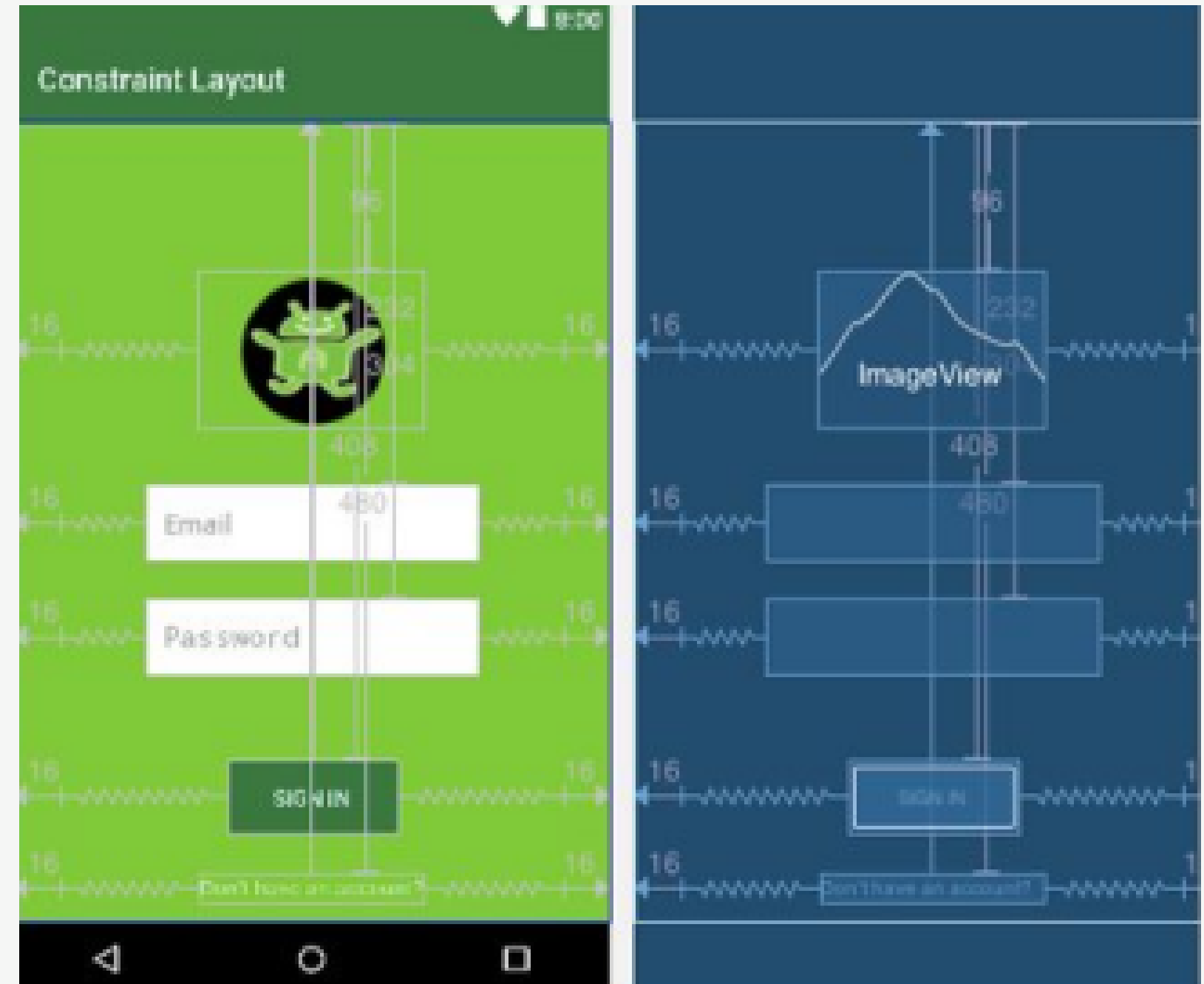
FrameLayout akan membuat komponen yang ada di dalamnya menjadi menumpuk atau saling menutupi satu sama lain. Komponen yang pertama pada layout ini akan menjadi alas bagi komponen-komponen di atasnya.



Layout

ConstraintLayout

ConstraintLayout memungkinkan kita membuat layout yang kompleks dengan hierarki flat (tanpa Nested ViewGroup).



Layout

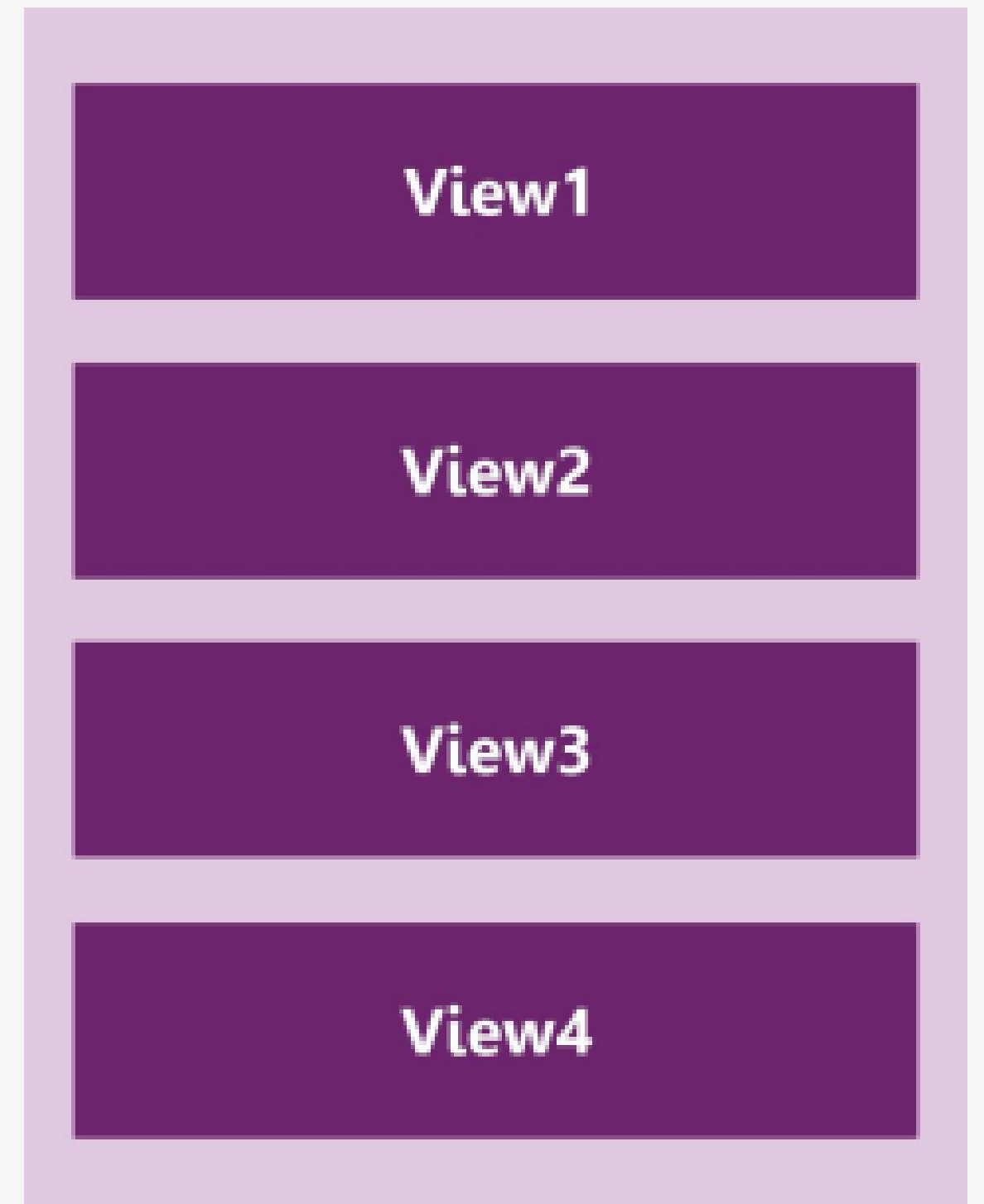
WebView

WebView pada android merupakan pengembangan dari kelas ViewGroup dan digunakan untuk menampilkan konten halaman web HTML statis atau konten halaman web dengan URL di aplikasi android sebagai bagian dari layout activity.

Layout

ListView

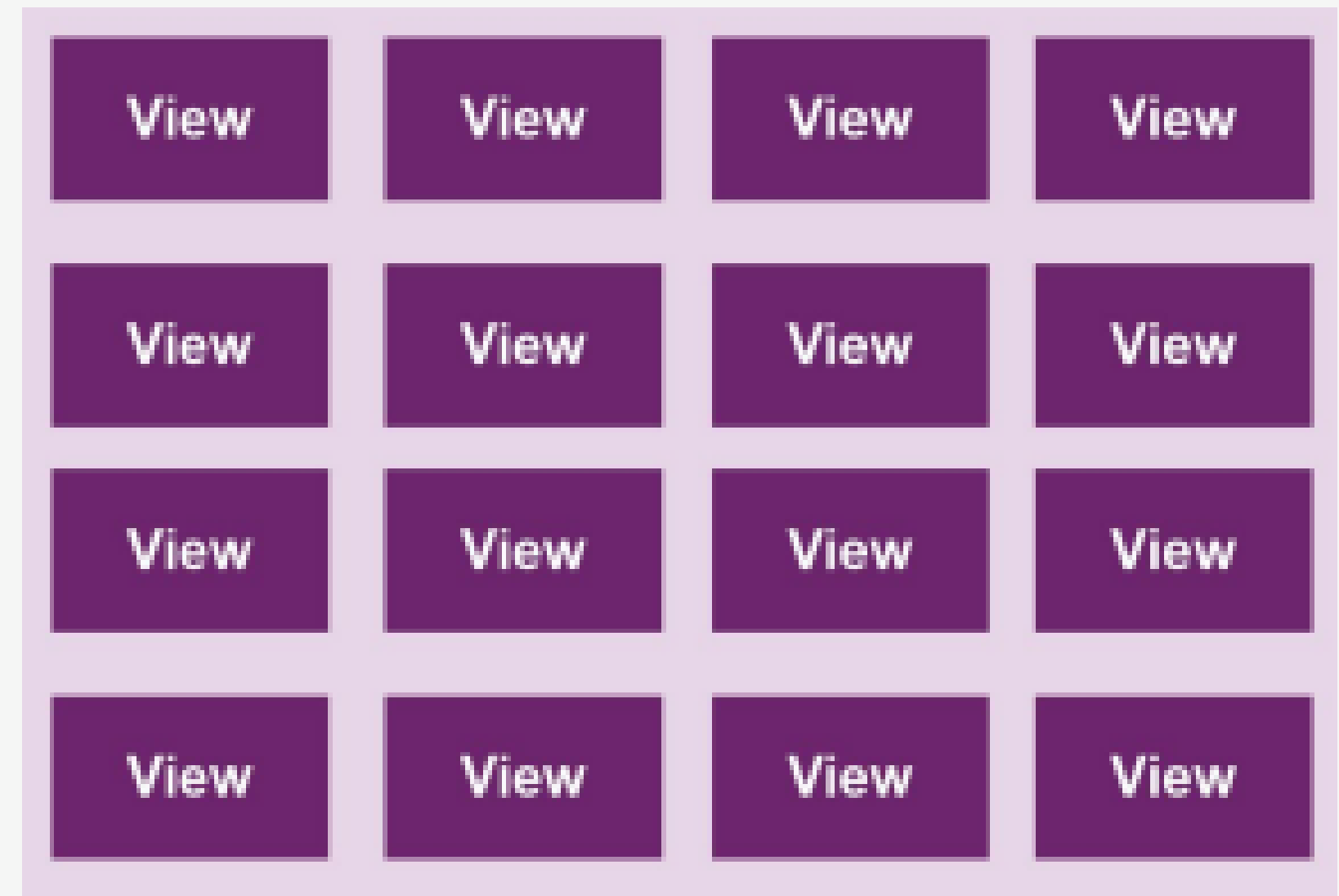
ListView adalah ViewGroup yang berfungsi untuk mengelompokkan beberapa item dan menampilkannya dalam list yang dapat di scroll secara vertikal.



Layout

GridView

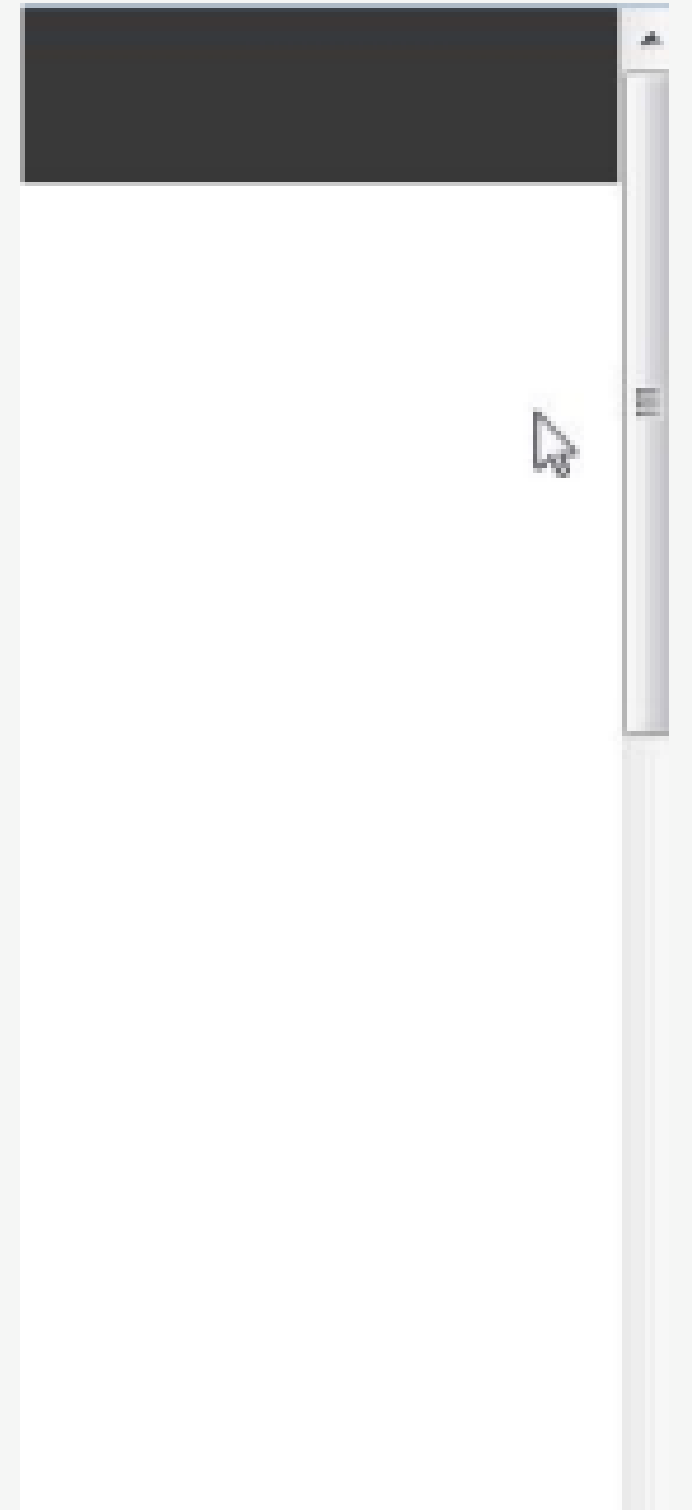
GridView adalah ViewGroup yang digunakan untuk menampilkan item dalam bentuk kotak-kotak dua dimensi (baris dan kolom), kotak yang dapat di scroll dan item grid secara otomatis dimasukkan ke GridView layout menggunakan ListAdapter.



Layout

ScrollView

ScrollView adalah sejenis layout yang berguna untuk menambahkan scroll bar vertikal atau horizontal ke konten yang lebih besar dari ukuran sebenarnya seperti LinearLayout, RelativeLayout, FrameLayout, dll.





Terimakasih

