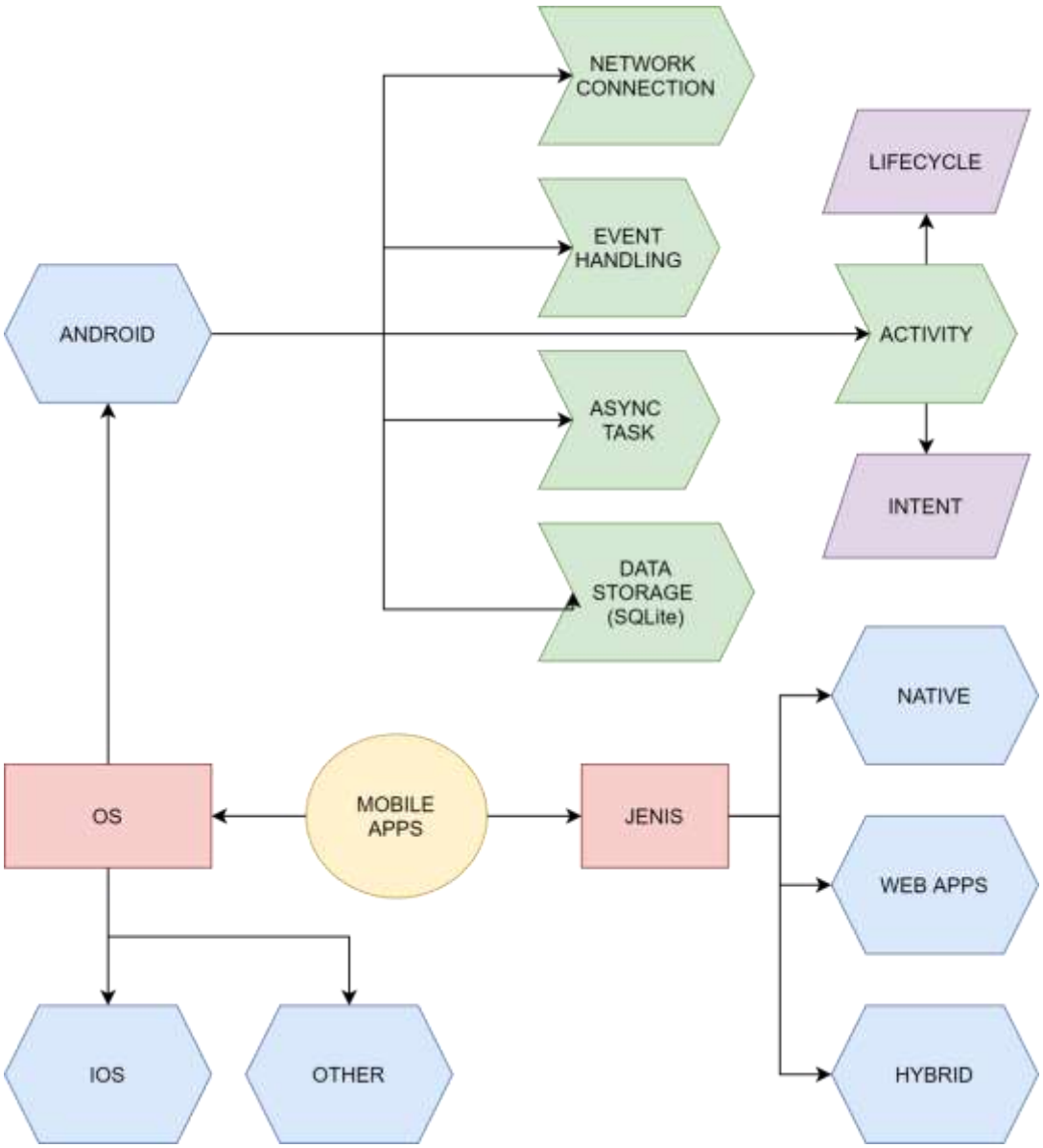
# 5 PEMROGRAMAN PLATFORM KHUSUS

MINDMAP MATA KULIAH PEMROGRAMAN PLATFORM KHUSUS PASCA UTS



# RANGKUMAN PEMROGRAMAN PLATFORM KHUSUS

## PENGEMBANGAN APLIKASI PERANGKAT BERGERAK (*MOBILE*)

### A. MOBILE APP

*Mobile App* adalah suatu software atau perangkat lunak yang berjalan di sebuah *smartphone*, *tablet*, atau perangkat sejenisnya. *Mobile app* memilki beberapa karakteristik, yaitu:

- Fitur yang dimiliki aplikasi bergerak pada umumnya lebih spesifik, instan, dan terbatas.
- Dapat diakses dengan cara-cara yang sederhana/ simpel (*one touch, sliding, non scrollable, pinch*, dan lain sebagainya).
- Tampilan berukuran kecil dan minimal

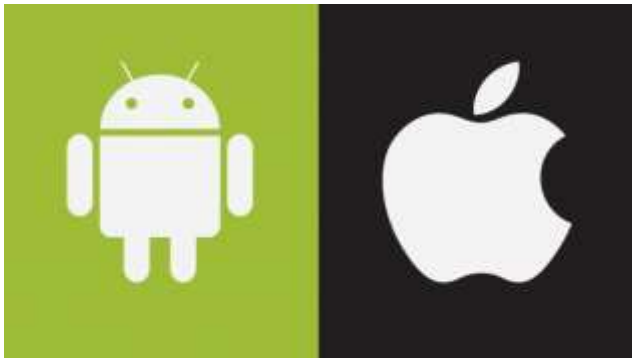
Namun, pada saat ini karakteristik tersebut sudah kurang relevan lagi karena banyak *mobile app* yang sudah tidak mengikuti karakteristik tersebut. Jenis *mobile app* ada tiga yaitu:

1. *Native App* adalah aplikasi yang dibuat, dikompilasi dan diinstall khusus untuk platform tertentu. Native app memilki beberapa kelebihan dan kekurangan nya yaitu:
   a. Bisa berjalan secara offline.
   b. Dapat didistribusikan melalui Google Play, App Store, Windows Store.
   c. Dapat menggunakan API dari perangkat mobile yang digunakan, seperti sensor, kontak, dan sebagainya.
   d. Membutuhkan waktu dan biaya yang lebih dalam membuatnya.
2. *Web Based App*
   a. *Web based app* adalah aplikasi mobile yang dibangun dengan teknologi pembuatan Responsive Web menggunakan HTML, CSS, dan JavaScript. Web based app bersifat cross platform sehingga memungkinkan untuk digunakan di platform lain. Beberapa kelebihan dan kekurangan nya yaitu:
   b. Memungkinkan untuk membuat single version app.
   c. Membutuhkan koneksi internet
   d. Sulit untuk menggunakan API dari perangkat mobile tersebut.
3. *Hybrid* yaitu aplikasi yang menggabungkan antara *Native* dan *Web app*.


### B. ANDROID

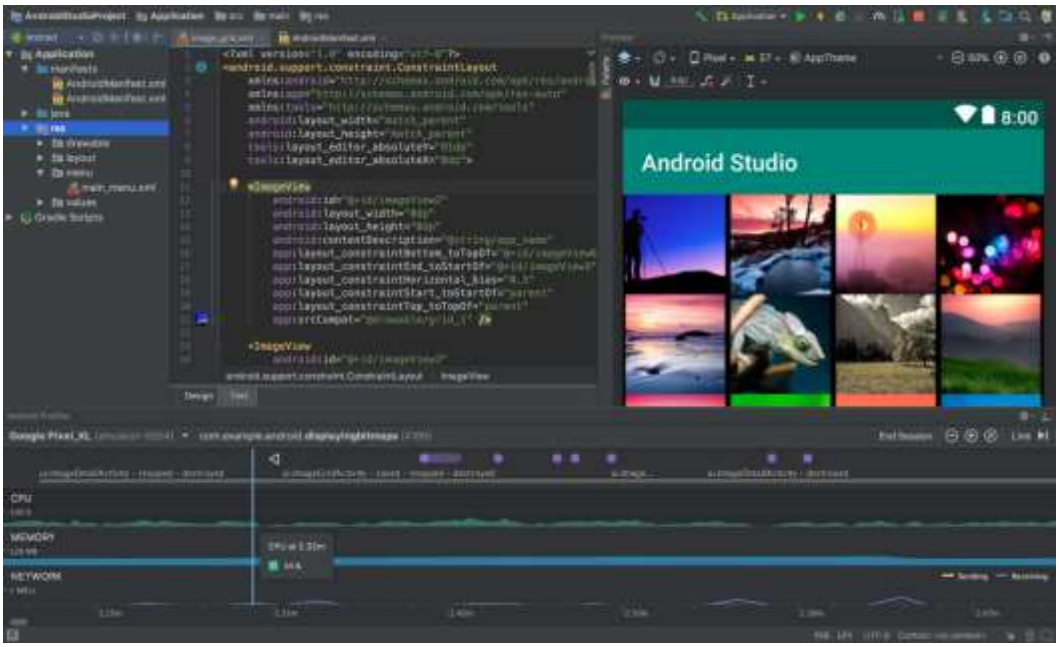
Pada saat ini, *operating system* (OS) yang banyak digunakan adalah iOS dan Android. iOS ada alah OS yang digunakan pada *smartphone* yang dikeluarkan oleh Apple inc, sedangkan Android dikeluarkan oleh Google dan digunakan pada banyak *smartphone* di seluruh dunia. Berdasarkan data yang dikeluarkan oleh Statista, pada bulan September 2020 pengguna Android

di dunia mencapai 72% dari total pengguna *smartphone* dan iOS 27%, dan sisanya menggunakan OS lainnya. Pada tahun 2020/2021, yang akan dipelajari pada perkuliahan semester ini adalah Android.



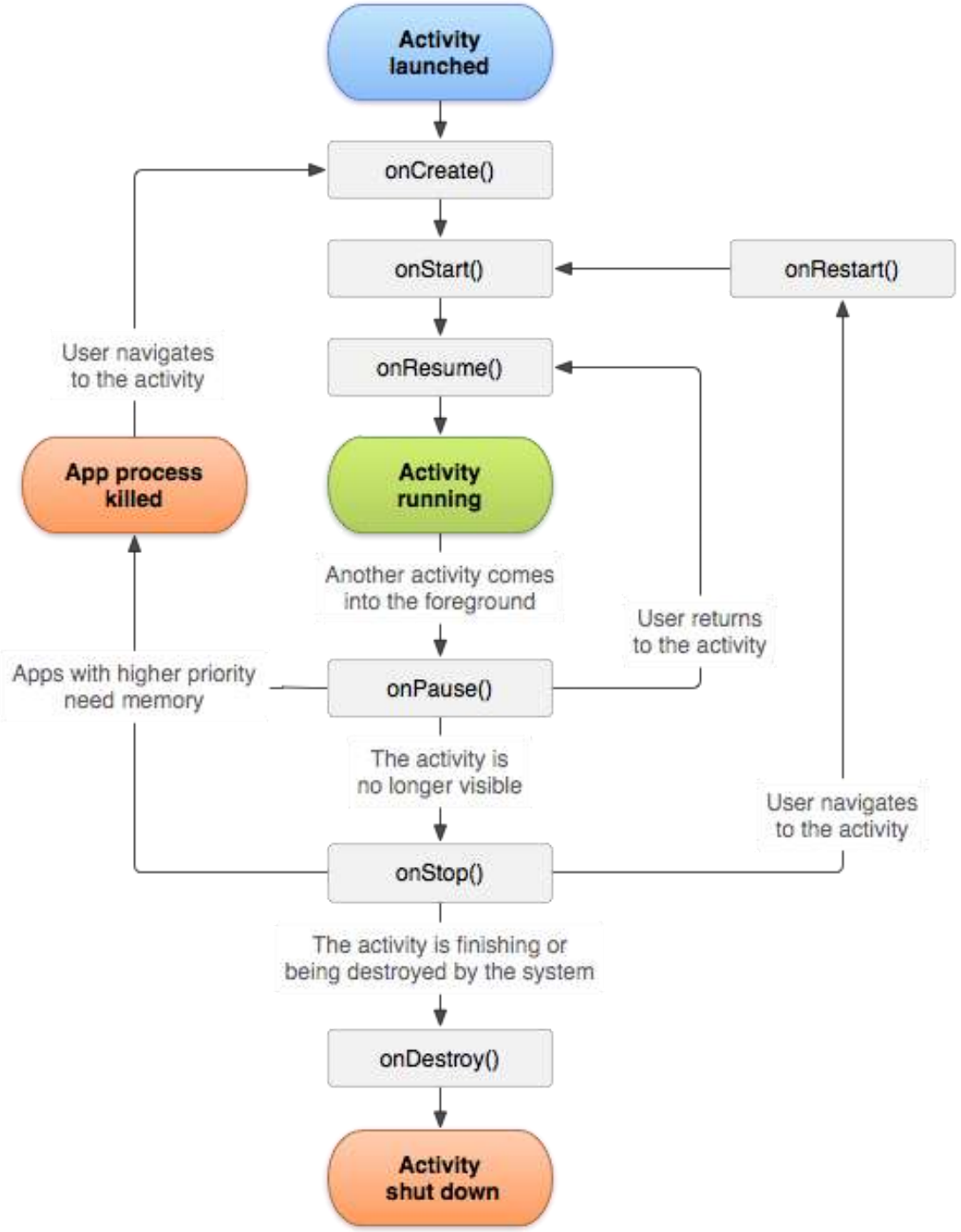
Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware dan aplikasi. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007. Android adalah sistem operasi yang *open source* di bawah Lisensi Apache 2.0. Dengan demikian, Android dapat dimodifikasi secara bebas dan didistribusikan oleh *developer.*

Untuk mengembangkan aplikasi Android, *tools* yang paling umum digunakan adalah Android Studio. Android Studio adalah *tools* resmi yang dikeluarkan oleh Google untuk mengembangkan aplikasi Android. Untuk informasi lebih lanjut dapat diperoleh dari website resmi nya di https://developer.android.com/studio.




## ANDROID ACTIVITY

Di dalam aplikasi Android memiliki banyak *activity* yang berjalan. *Activity* adalah komponen yang dapat dilihat oleh user dan dapat berinteraksi. *Activity System* diatur sebagai *activity stack*. Ketika suatu *activity* dimulai, biasanya akan ditaruh di atas dari *stack* dan *activity* sebelumnya akan berada dibawah *activity* yang sekarang. Dan bisa saja terdapat beberapa activity yang muncul di layar. *Life Cycle* dari *activity* adalah sebagai berikut.



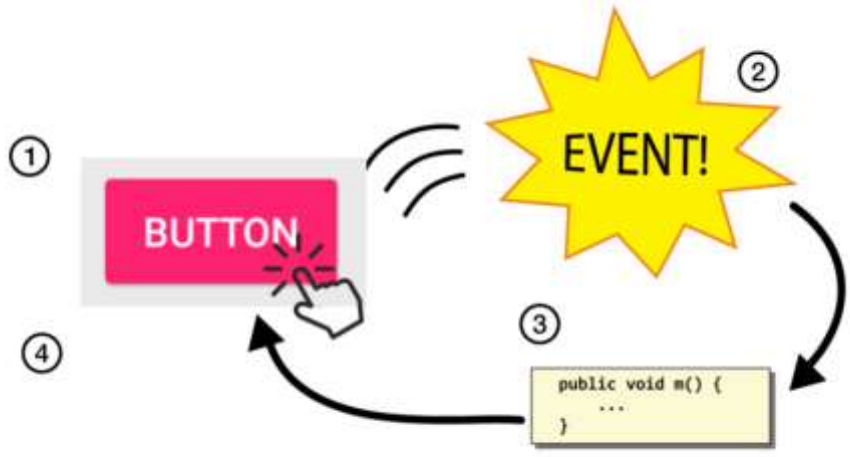

Penjelasan mengenai setiap method yang ada pada Activity Class adalah sebagai berikut:

- *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, *method* ini biasanya dieksekusi setelah onPause().
- *onPause()* akan dipanggil saat ada *Activity* lain yang terbuka.
- *onStop()* adalah kondisi saat *Activity* tidak ditampilkan di layar (misalnya pada saat pengguna menekan tombol *Home*).
- *onRestart()* adalah kondisi saat *Activity* kembali dibuka oleh pengguna.
- *onDestroy()* adalah kondisi saat *Activity* dihilangkan dari *memory*.

Penjelasan lebih lengkap dapat anda baca di dokumentasi Androidnya langsung di https://developer.android.com/guide/components/activities/activity-lifecycle.

## EVENT HANDLING

*Event* adalah suatu aksi yang muncul pada saat *user* berinteraksi seperti *click, typing, scrolling* dengan *widgets*. *Widget* yang dapat berinteraksi dengan user adalah *Button, Checkbox, DatePicker, ImageView*, dan lain sebagainya. Pada prosesnya, ketika *widget* menerima *input* dari *user*, maka akan dipanggil fungsi yang menangani event tersebut. Fungsi tersebut akan melakukan suatu aksi yang bisa saja ditampilkan kembali maupun tidak.




Event Listener adalah suatu *interface* di dalam View class yang mengandung satu *callback method*. *Method* tersebut akan dipanggil apabila *View* tersebut mendapatkan suatu interaksi pengguna melalui *User Interface*. Beberapa *callback method* yang dapat digunakan adalah *onClick(), onLongClick(), onFocusChange(), onKey(), onTouch(), dan onCreateContextMenu()*. Event handler akan dipanggil apabila terdapat suatu event tertentu yang sudah di register sebelumnya.

## INTENT

Aplikasi Android dapat terdiri dari satu atau lebih *activity.* Biasanya setiap activity independen dengan activity lainnya, tetapi bisa berinteraksi dengan bertukar data dan *actions*. Biasanya ada 1 activity dasar yang dibuka pertama kali (main) yang akan berinteraksi dengan user pertama kali. Untuk berpindah dari satu activity ke activity lainnya dapat menggunakan intent. Activity berinteraksi dengan activity lainnya secara *asynchronous*.

Intent adalah sebuah *class* dalam pemrograman android yang berfungsi untuk perpindahan antar *activity* dan dapat digunakan untuk komunikasi dengan *activity* lain. Ada informasi utama di dalam sebuah intent yaitu, data dan action. Action adalah aksi yang dijalankan pada suatu intent, seperti ACTION_VIEW, ACTION_EDIT, ACTION_MAIN, dan lain-lain. Data adalah data yang akan digunakan di dalam sebuah intent. Selain itu, ada beberapa atribut lainnya yang dapat ditambahkan di dalam sebuah intent yaitu *category, type, component,* dan *extras.* Ada dua bentuk intent yang digunakan yaitu :

- Implicit Intent berfungsi untuk memanggil fungsi activity yang sudah ada di fungsi internal android seperti Dial Number, Open Browser dan lainnya.
- Explicit Intent berfungsi untuk mengaktifkan komponen-komponen dalam satu aplikasi yang sama. Misalnya seperti : Berpindah Activity.

Extra adalah metode dari intent untuk mengirimkan tambahan data dari satu *activity* ke *activity* yang dituju. Data yang dikirimkan pun tipenya dapat bermacam-macam, bisa berbentuk String, angka (*integer, float, double*), *ArrayList, boolean, array, character*, dan sebagainya. Namun tidak semua tipe data support untuk dikirim antar *activity*, tipe data yang berukuran besar seperti gambar (image bitmap) atau file, tidak dapat di-passing antar activity melalui metode ini. Intent dapat dipanggil atau dijalankan dengan method *startActivity(intent), sendBroadcast(intent), startService(intent),* atau *bindService(intent)*. Penjelasan lebih lanjut dapat dilihat pada dokumentasi android nya di https://developer.android.com/reference/android/content/Intent. Untuk lebih jelas penggunaan intent dapat dilihat pada *source code* praktikum modul angkatan 60.

## LIST VIEW DAN RECYCLERVIEW

Karena pada perangkat mobile, ukuran layar, dan resource seperti RAM, CPU terbatas, maka untuk menampilkan data yang banyak harus diatur sehingga mudah untuk dilihat dan tidak boros resource. Oleh karena itu, untuk menampilkan data yang cukup banyak, cara paling efektif adalah menggunakan ListView. ListView adalah suatu view yang menampilkan data list secara vertical dan bisa di Scroll. ListView menggunakan adapter untuk berhubungan dengan data. Ada dua macam ListView, yaitu StaticList dan DynamicList. Namun *ListView* masih memilki banyak kekurangan, sehingga dikembangkan lagi *RecyclerView*.

RecyclerView adalah sebuah komponen tampilan (widget) yang lebih canggih ketimbang pendahulunya ListView dan bersifat lebih fleksibel. RecyclerView memiliki kemampuan untuk menampilkan data secara efisien dalam jumlah yang besar. Terlebih jika anda memiliki koleksi data yang tiap elemennya mampu berubah-ubah sewaktu dijalankan (runtime) karena interaksi pengguna atau karena adanya pengaruh dari jaringan internet. Didalam RecyclerView terdapat beberapa komponen penting, yaitu:

- Data : Data yang ditampilkan dalam RecyclerView
- RecyclerView : View untuk menampilkan list data
- RowView : Layout untuk menampilkan 1 row data
- Layout manager : Bagian yang menatur UI pada view
- Adapter : Komponen yang menghubungkan data dengan RecyclerView
- View holder : Untuk menampilkan 1 buah item.

Penjelasan lebih lanjut dapat dilihat pada dokumentasi android nya di https://developer.android.com/guide/topics/ui/layout/recyclerview. Untuk lebih jelas penggunaan dan implementasi *recycleview* dapat dilihat pada *source code* praktikum modul angkatan 60.

## ASYNC TASK DAN NETWORK CONNECTION

### A. ASYNC TASK

*AsyncTask* adalah suatu task yang berjalan secara *asynchronous* atau yang berjalan di background. AsyncTask diperlukan agar program berjalan dengan lancar dengan membagi proses yang bisa dilakukan di background pada beberapa thread sehingga tidak mengganggu performa proses lainnya yang berjalan di thread lain. Gunakan kelas AsyncTask untuk mengimplementasikan tugas asinkron yang berjalan lama di Worker Thread. Worker Thread

adalah Thread yang bukan Thread UI/Main Thread. AsyncTask memungkinkan anda menjalankan pada background dan mempublikasikan hasil di Thread UI tanpa memanipulasi thread. Contoh proses yang bisa dilakukan pada asynctask adalah proses yang berhubungan dengan jaringan, perhitungan yang banyak atau butuh waktu yang lama, download atau upload, memproses gambar, video, dan music, dan memuat data.

*Asynchronous task* didefinisikan dengan proses yang berjalan di background thread dan hasilnya di tampilkan di UI thread. *Asynchronous task* didefinisikan dengan 3 tipe generic yaitu *params, progress,* dan *result,* penjelasannya sebagai berikut:

- Params adalah type dari parameter yang akan dikirimkan dalam pengeksekusian suatu task.
- Progress adalah tipe dari progress pada saat pemrosesan di background.
- Result adalah tipe dari hasil perhitungan didalam background.

Selain itu juga, ada 4 langkah yaitu, onPreExecute, doInBackground, onProgressUpdate dan onPostExecute.

- onPreExecute() dipanggil di UI thread sebelum suatu task di jalankan.
- doInBackground(Params, ...) akan dipanggil secepatnya setelah method onPreExecute dan digunakan untuk menjalankan proses di background.
- onProgressUpdate(Progress, ...) akan dipanggil di UI thread setelah memanggil publishProgress(Progress, ...) dan biasanya akan menampilkan progress dari proses yang dijalankan di background.
- onPostExecute(Result) akan dipanggil di UI thread setelah proses di background selesai.

Perlu diperhatikan bahwa *AsyncTask* sudah mengalami *deprecated* pada API level 30 dan dianjurkan menggunakan *java.util.concurrent* atau *Kotlin concurrency utilities*. Penjelasan lebih lanjut dapat dilihat pada dokumentasi resmi nya yang bisa anda baca di https://developer.android.com/reference/android/os/AsyncTask. Untuk lebih jelas penggunaan dan implementasi *AsyncTask* dapat dilihat pada *source code* praktikum modul angkatan 60.

## B. NETWORK CONNECTION

Pada kebanyakan aplikasi android, biasanya menggunakan protokol HTTP untuk mengirim dan menerima data. Untuk melakukan koneksi dalam aplikasi android ada beberapa step yang dilakukan yaitu:

- Membuat *permission* di Android Manifest untuk menggunakan internet dan akses jaringan.
- Melakukan pengecekan jaringan.
- Membuat *Worker Thread*
- Membuat implementasi *background task*
  - Membuat URI
  - Membuat koneksi HTTP
  - *Connect* dan GET data
- Memproses hasil

Untuk lebih jelas penggunaan dan implementasi Network Connection dengan membuat beberapa case untuk koneksi ke suatu *server* dan dikombinasikan dengan materi sebelum-sebelumnya dapat dilihat pada *source code* praktikum modul angkatan 60 yang akan selalu di update.

Data storage in android dapat digunakan untuk menyimpan data tanpa koneksi internet (offline) di local android. Android sudah menyediakan embedded database yaitu SQLite. Tipe data yang dapat disimpan di dalam SQLite adalah null, integer, real, text, blob. API SQLite di android berada pada android.database.sqlite. Beberapa karakteristik dari SQLite sebagai berikut :

- Mirip library
- Ukuran kecil
- Bisa CRUD (Create, Read, Update, Delete)
- Bisa mengelola sendiri
- Tidak ada server khusus untuk DBMS (Database Management System)
- ACID Compliant
- Akan mengunci database apabila terjadi proses insert
- Query bisa dilakukan secara parallel

Meskipun penggunaan SQLite cukup *powerful*, tetapi ada beberapa hal yang dapat membutuhkan waktu dan tenaga yang lebih saat menggunakannya pada kondisi seperti apabila terjadi perubahan *database scheme* maka harus merubah semua query yang ada, dan juga dalam penggunaannya harus menggunakan banyak *boilerplate* untuk merubah dari *SQL query* menjadi *data object*. Untuk lebih jelas penggunaan dan implementasi SQLite dapat dilihat pada *source code* praktikum modul angkatan 60 yang akan selalu di update.

Catatan :

Materi dibuat hanya untuk me-*review* materi mata kuliah Pemrograman Platform Khusus Pasca UTS. Karena mata kuliah ini berisi praktikum maka berikut adalah link repository untuk project-project untuk memudahkan dalam mempelajari mata kuliah ini. Link mantap jiwa : https://github.com/modul60stis/ppk source code akan di-*update* secara berkala dan pastikan membaca file readme.md nya.

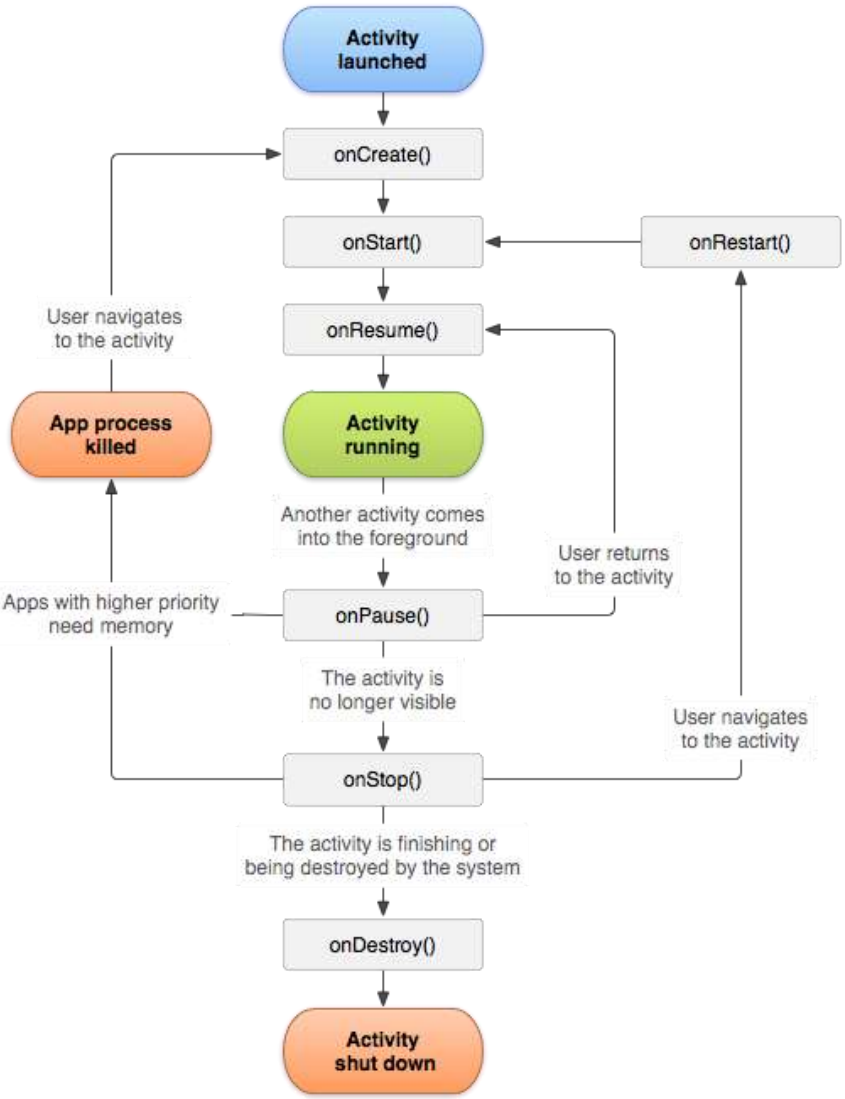
## UJIAN AKHIR SEMESTER GENAP TAHUN AKADEMIK 2019/2020

==========

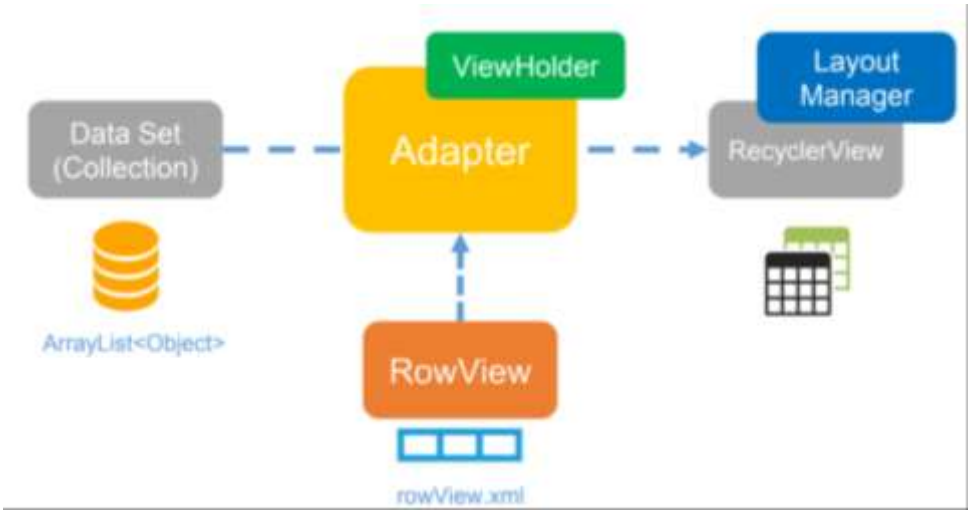
| | |
|---|---|
| MATA KULIAH | : Pemrograman Platform Khusus |
| TINGKAT | : III KOMPUTASI STATISTIK D4 PEMINATAN S |
| DOSEN | : Takdir |
| HARI/TANGGAL | : Senin, 16 Desember 2019 |
| WAKTU | : 90 Menit |
| SIFAT UJIAN | : Tertulis, *close book,* tanpa peralatan elektronik |

==========

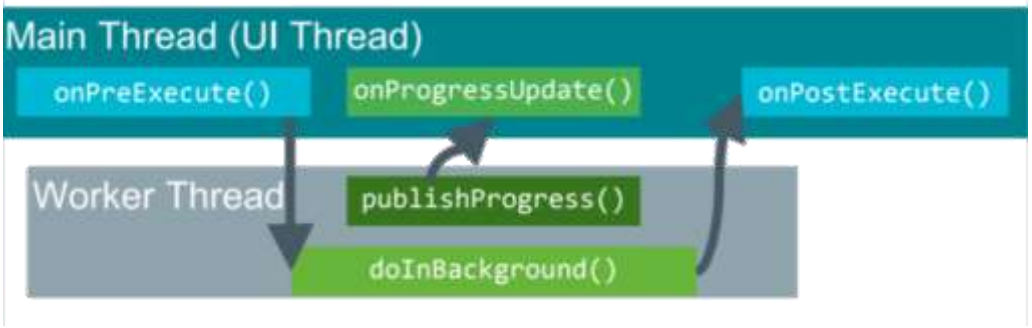
1. Jelaskan perbedaan antara *Native Application*, *Web Application,* dan *Hybrid Application* pada *mobile apps*. Paparkan kriteria dalam memilih salah satu atau beberapa jenis aplikasi *mobile apps* tersebut!
2. Jelaskan apa yang anda pahami mengenai *Android Activity Lifecycle* seperti pada bagan berikut:

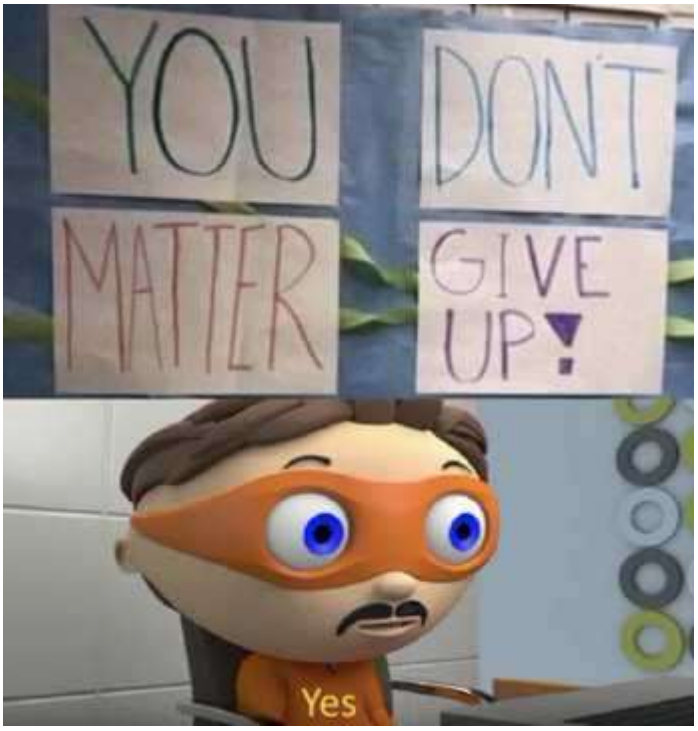


3. Jelaskan yang anda pahami mengenai implementasi *RecycleView* seperti yang ditunjukkan pada gambar berikut. Pastikan setiap komponen pada setiap bagian terjelaskan.'




4. Berikut adalah gambaran proses *AsyncTask.* Berikan penjelasan proses yang terjadi yang dilengkapi proses penerapannya. Anda bisa menganalogikan pada satu contoh aplikasi atau proses untuk menjelaskan.



# PEMBAHASAN UAS GENAP 2019/2020

# PEMROGRAMAN PLATFORM KHUSUS

1. Perbedaan antara *Native Application*, *Web Application,* dan *Hybrid Application* pada *mobile apps* adalah sebagai berikut :
   1. *Native App* adalah aplikasi yang dibuat, dikompilasi dan diinstall khusus untuk platform tertentu. Native app memilki beberapa kelebihan dan kekurangan nya yaitu:
      - Bisa berjalan secara offline.
      - Dapat didistribusikan melalui Google Play, App Store, Windows Store.
      - Dapat menggunakan API dari perangkat mobile yang digunakan, seperti sensor, kontak, dan sebagainya.
      - Membutuhkan waktu dan biaya yang lebih dalam membuatnya.
      - Gunakan *native app* jika aplikasi yang dibuat membutuhkan penggunaan resource yang banyak dan harus digunakan secara efektif. Dengan kata lain, jika aplikasi yang dibuat membutuhkan manajemen resource, seperti ram, processor, dll, yang lebih baik untuk mengejar performa yang lebih optimal. Native app juga harus digunakan jika akan menggunakan API yang hanya bisa diakses menggunakan native code nya.
   2. *Web based app* adalah aplikasi mobile yang dibangun dengan teknologi pembuatan Responsive Web menggunakan HTML, CSS, dan JavaScript. Web based app bersifat cross platform sehingga memungkinkan untuk digunakan di platform lain. Beberapa kelebihan dan kekurangan nya yaitu:
      - Memungkinkan untuk membuat single version app.
      - Membutuhkan koneksi internet
      - Sulit untuk menggunakan API dari perangkat mobile tersebut.
      - Gunakan *Web Based App* jika aplikasi yang digunakan adalah aplikasi yang hanya bisa di akses melalui browser, tidak perlu diubah, dan cross platform. Web based app juga biasa digunakan jika aplikasi yang dibuat tidak membutuhkan banyak API dari mobile devices dan akan dikembangkan secara cepat untuk berbagai macam platform. Jangan gunakan web based app, jika aplikasi yang digunakan membutuhkan performa yang optimal, karena performa yang dihasilkan tidak lebih baik dari native app.
   3. *Hybrid* yaitu aplikasi yang menggabungkan antara *Native* dan *Web app*.

2. Penjelasan mengenai setiap method yang ada pada Activity Class adalah sebagai berikut:
   1. *onCreate()* adalah kondisi awal saat *Activity* baru diciptakan, biasanya dilakukan inisialisasi pada tahapan ini.
   2. *onStart()* adalah saat *Activity* dimulai.
   3. *onResume()* adalah saat *Activity* dibuka kembali, *method* ini biasanya dieksekusi setelah onPause().
   4. *onPause()* akan dipanggil saat ada *Activity* lain yang terbuka dan activity yang sekarang masih tetap terbuka tetapi tidak berada di paling atas dalam *activity stack*.
   5. *onStop()* adalah kondisi saat *Activity* tidak ditampilkan di layar (misalnya pada saat pengguna menekan tombol *Home*) atau berpindah ke *activity* lainnya.
   6. *onRestart()* adalah kondisi saat *Activity* kembali dibuka oleh pengguna.
   7. *onDestroy()* adalah kondisi saat *Activity* dihilangkan dari *memory* atau dimatikan oleh sistem atau pengguna.
3. Didalam RecyclerView terdapat beberapa komponen penting, yaitu:

1. Data : Data yang ditampilkan dalam RecyclerView, data berupa dataset yang memiliki schema tertentu.
2. RowView : Layout (XML) untuk menampilkan 1 row data yang berasal dari dataset untuk ditampilkan di RecycleView.
3. View holder : Objek yang digunakan oleh adapter untuk menampilkan 1 view untuk 1 buah item.
4. Adapter : Komponen yang menghubungkan data dengan RecyclerView dan yang mengatur proses *creating, updating, adding, deleting item* pada saat data berubah.
5. RecyclerView : View untuk menampilkan list data dari Dataset yang ada
6. Layout manager : Bagian yang mengatur UI pada view dan yang mengatur *item views* di dalam RecyclerView.

4. Proses yang terjadi pada *AsyncTask* adalah sebagai berikut :
   1. onPreExecute() dipanggil di UI thread sebelum suatu task di jalankan. Misalnya pada proses download, adalah proses initialisasi sebelum download dimulai.
   2. doInBackground(Params, ...) akan dipanggil secepatnya setelah method onPreExecute dan digunakan untuk menjalankan proses di background. Misalnya pada proses download adalah pada saat proses download berjalan
   3. onProgressUpdate(Progress, ...) akan dipanggil di UI thread setelah memanggil publishProgress(Progress, ...) dan biasanya akan menampilkan progress dari proses yang dijalankan di background. Misalnya pada proses download adalah pada saat menampilkan progres download yang terjadi.
   4. onPostExecute(Result) akan dipanggil di UI thread setelah proses di background selesai. Misalnya pada proses download pada saat download selesai dan menampilkan notifikasi bahwa download telah berhasil.

Catatan: