

STMIK Kaputama



Pertemuan 2

PENGANTAR TEKNOLOGI ANDROID

Kristina Annatasia Br Sitepu, M.Kom





PEMBAHASAN

Pengantar Android

Komponen User Interface

Konsep Activity

Siklus Activity



MENGENAL ANDROID STUDIO

Android Studio adalah Integrated Development Environment (IDE) resmi untuk pengembangan aplikasi Android.

Berbasis editor kode dan alat developer yang andal dari IntelliJ IDEA.



MENGENAL ANDROID STUDIO

Android Studio menawarkan lebih banyak fitur yang mampu meningkatkan produktivitas Anda saat membuild aplikasi Android, seperti:

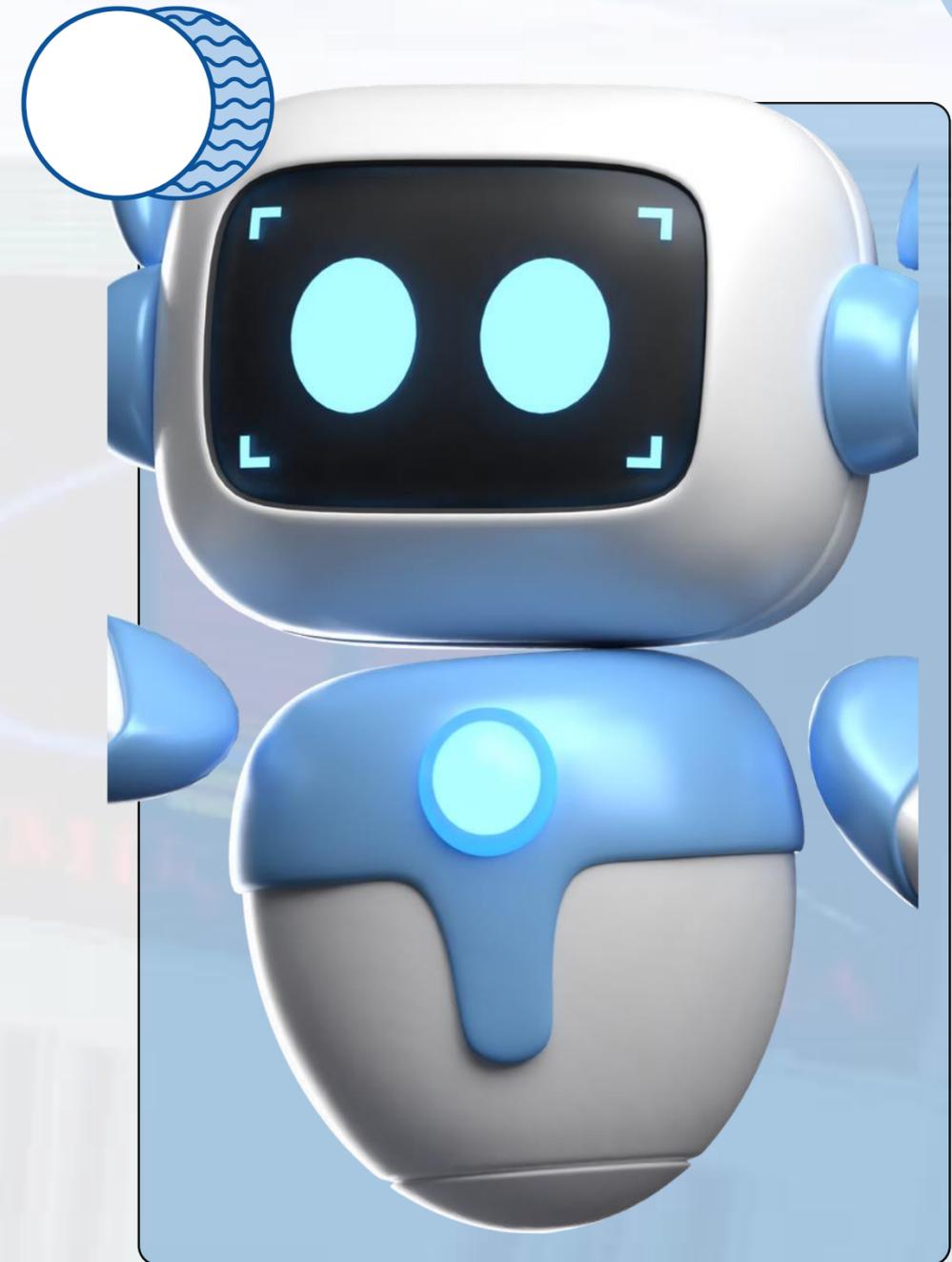
1. Sistem build berbasis Gradle yang fleksibel.
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android.
4. Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi.

MENGENAL ANDROID STUDIO

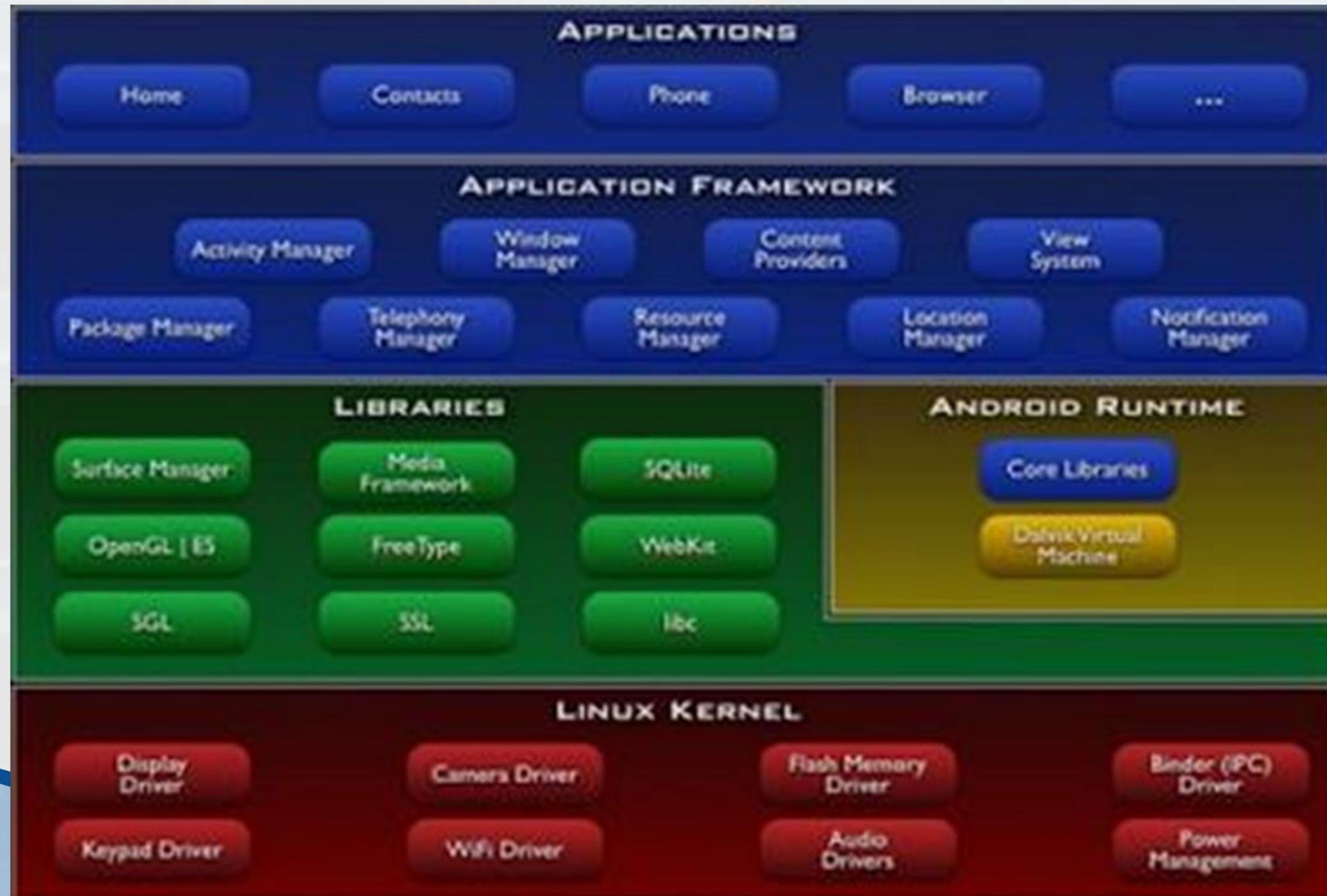
5. Template Kode dan Integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel.
6. Framework dan alat pengujian yang lengkap.
7. Alat Lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine.

ARSITEKTUR PLATFORM ANDROID

Android dibangun dengan menggunakan asas object oriented, dimana elemen-elemen penyusun sistem operasinya berupa objek yang dapat kita gunakan kembali/reusable.



GAMBAR Arsitektur Platform Android



1. LINUX KERNEL



- Basis dari Sistem Android: Android dibangun di atas kernel Linux, yang menyediakan berbagai fitur dasar seperti manajemen proses, manajemen memori, sistem file, driver perangkat keras, keamanan, dan komunikasi antar proses (IPC).
- Driver Perangkat Keras: Kernel ini berfungsi sebagai antarmuka antara perangkat keras dan software, menyediakan driver untuk berbagai perangkat keras seperti layar, kamera, Wi-Fi, Bluetooth, dan lain-lain.
- Keamanan dan Memori: Kernel juga bertanggung jawab atas keamanan inti, seperti manajemen akses dan memori, serta pengaturan proses.

2. LIBRARY



- Native C/C++ Libraries: Di atas kernel Linux, Android menggunakan sejumlah pustaka C/C++ yang mengelola berbagai fungsi sistem. Contoh pustaka penting di sini termasuk OpenGL untuk rendering grafis 2D/3D, WebKit untuk rendering halaman web, dan SQLite untuk basis data relasional.
- Android Runtime: Bagian dari pustaka ini adalah Android Runtime (ART), yang menggantikan Dalvik Virtual Machine di versi Android terbaru. ART bertanggung jawab untuk mengeksekusi kode aplikasi. ART bekerja dengan menggunakan teknik kompilasi AOT (Ahead of Time) untuk meningkatkan performa aplikasi dan efisiensi memori.

3. APPLICATION FRAMEWORK

API dan Layanan untuk Aplikasi: Lapisan ini menyediakan berbagai framework dan layanan dasar yang digunakan oleh aplikasi Android. Contoh komponen dalam framework ini meliputi:

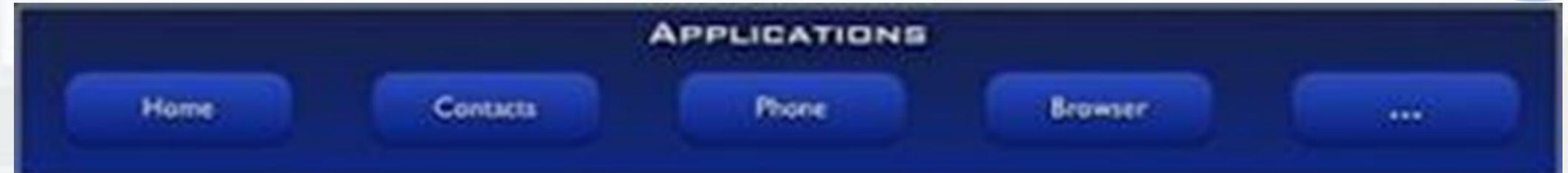
- Activity Manager: Mengelola siklus hidup aplikasi.
- Resource Manager: Mengelola sumber daya non-kode, seperti layout XML, string, dan gambar.
- Notification Manager: Mengelola notifikasi yang dapat dilihat pengguna.
- Content Providers: Memungkinkan aplikasi berbagi data antar aplikasi.
- Location Manager: Menyediakan layanan berbasis lokasi, seperti GPS.

3. APPLICATION FRAMEWORK

Lapisan ini memungkinkan pengembang aplikasi untuk memanfaatkan layanan inti Android tanpa harus berinteraksi langsung dengan sistem operasi yang lebih rendah.



4. APPLICATION



- Aplikasi Pengguna: Di lapisan paling atas dari arsitektur, terdapat aplikasi yang digunakan oleh pengguna. Contohnya termasuk aplikasi perpesanan, pemutar media, dan browser. Aplikasi-aplikasi ini dibangun menggunakan bahasa pemrograman Java, Kotlin, atau sekarang juga Jetpack Compose, dan memanfaatkan API dari Application Framework di bawahnya.
- Aplikasi Pihak Ketiga: Pengembang juga bisa membuat aplikasi mereka sendiri yang berjalan di atas framework ini dan diinstal dari Google Play Store.

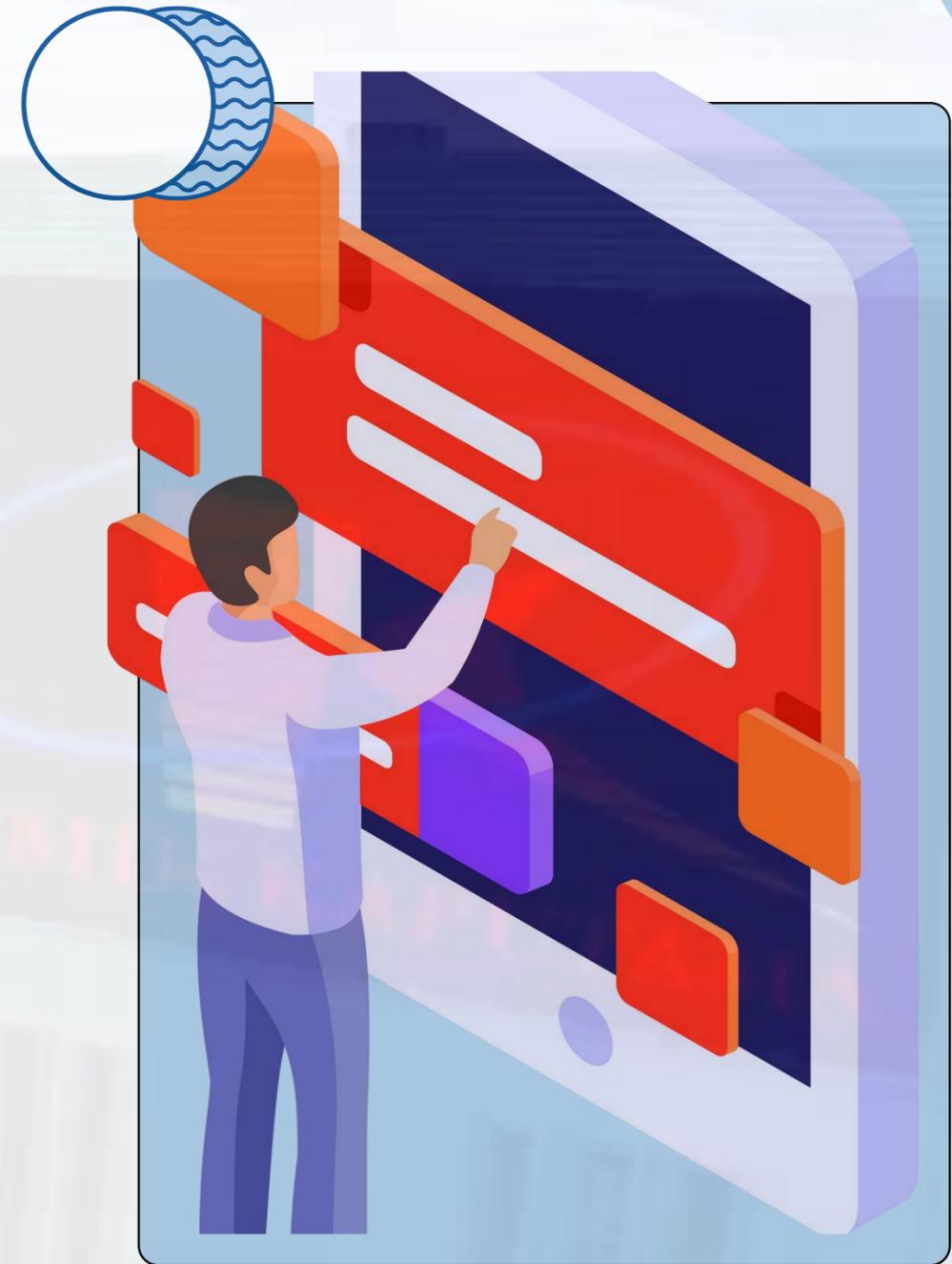
5.KOMPONEN KUNCI TAMBAHAN

- ART (Android Runtime): Sebelumnya Dalvik, kini ART menjalankan aplikasi dengan lebih efisien dengan kompilasi Ahead-Of-Time (AOT) yang meningkatkan performa aplikasi dibandingkan Dalvik yang menggunakan JIT (Just-In-Time) compilation.
- HAL (Hardware Abstraction Layer): Komponen yang mengizinkan Application Framework untuk berinteraksi dengan perangkat keras tanpa harus langsung mengakses kernel. HAL menyediakan antarmuka standar yang bisa digunakan oleh hardware.

KOMPONEN USER INTERFACE

Pengertian Komponen UI

Komponen UI adalah elemen-elemen yang digunakan untuk membangun tampilan antarmuka pengguna di aplikasi Android.



KOMPONEN USER INTERFACE

1. Views

- View adalah kelas dasar untuk komponen UI yang tampil di layar.
- Setiap elemen grafis, seperti tombol atau teks, adalah subclass dari View.
- View bisa berupa elemen visual sederhana atau kompleks seperti tombol, kotak teks, atau gambar.
- Contoh View umum : TextView, ImageView, Button, EditText.

KOMPONEN USER INTERFACE

2. ViewGroup

- ViewGroup adalah kelas yang menampung dan mengatur kumpulan View.
- ViewGroup bertindak sebagai wadah atau kontainer bagi komponen UI lainnya, baik itu View sederhana maupun ViewGroup lainnya.
- ViewGroup digunakan untuk mengatur tata letak (layout) dan interaksi antara berbagai elemen di layar.
- Contoh ViewGroup yang umum: LinearLayout, RelativeLayout, ConstraintLayout, FrameLayout.

KOMPONEN USER INTERFACE

3. Layouts

- Layout adalah jenis khusus dari ViewGroup yang digunakan untuk mengatur tata letak komponen UI dalam aplikasi.
- Berikut beberapa jenis layout yang umum: LinearLayout, RelativeLayout, ConstraintLayout, GridLayout, FrameLayout

KOMPONEN USER INTERFACE

4. Widgets

- Widgets adalah komponen UI yang lebih spesifik dan dapat langsung digunakan oleh pengguna untuk berinteraksi dengan aplikasi.
- Widgets ini merupakan subclass dari View.
- Contoh : Button, CheckBox, RadioButton, Spinner, SeekBar, ProgressBar, Switch.

KOMPONEN USER INTERFACE

5. Menu

- Menu adalah elemen UI yang memungkinkan pengguna berinteraksi dengan berbagai opsi.
- Android mendukung beberapa jenis menu: Options Menu, Contextual Menu, Popup Menu.

KOMPONEN USER INTERFACE

6. Dialog

- Dialog adalah jendela kecil yang muncul di atas UI aplikasi saat ini untuk meminta masukan pengguna atau memberikan informasi.
- Ada beberapa jenis dialog: AlertDialog, DatePickerDialog, TimePickerDialog.

KOMPONEN USER INTERFACE

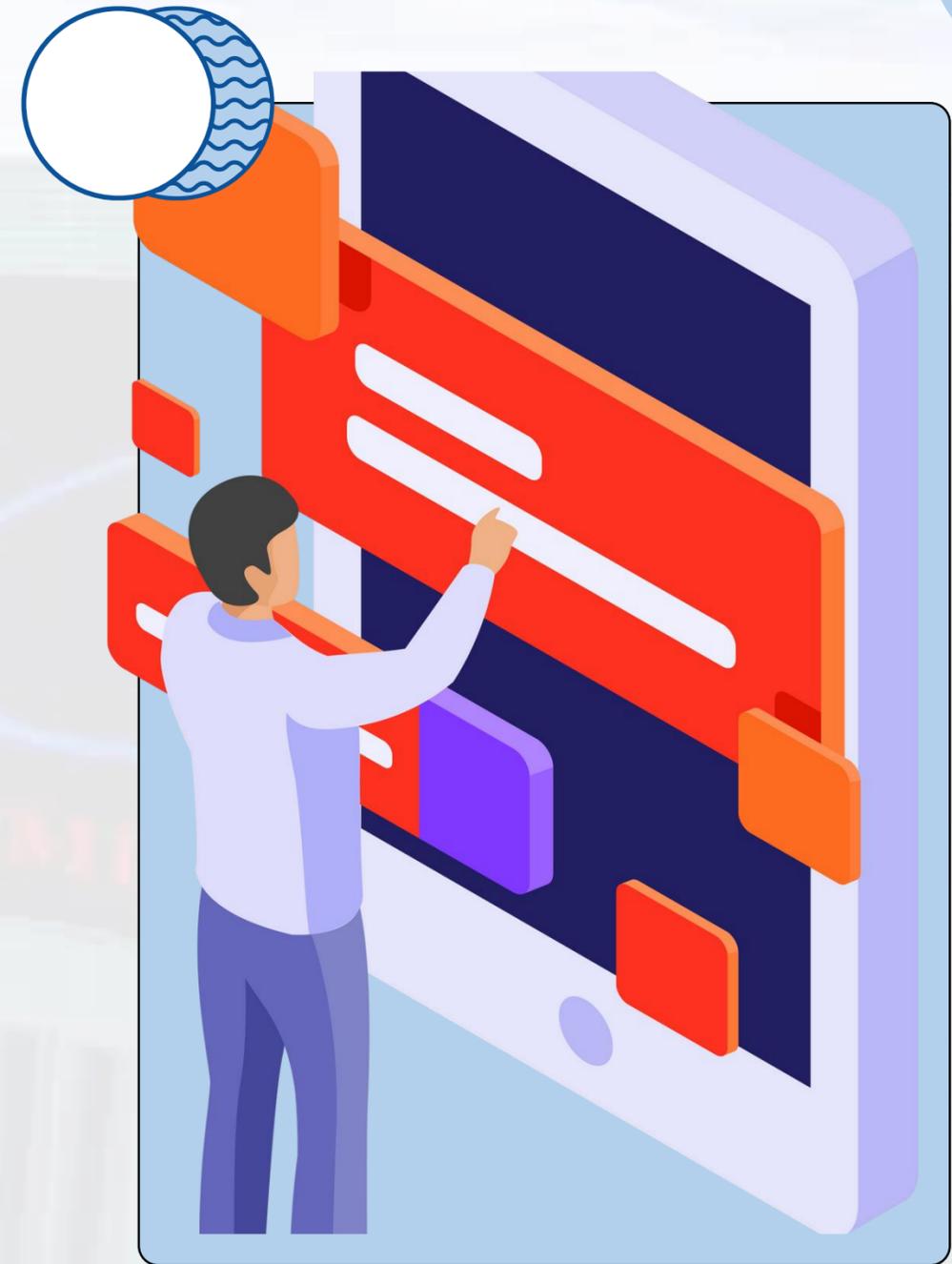
7. RecyclerView

- RecyclerView adalah komponen UI yang lebih canggih dibandingkan ListView. RecyclerView digunakan untuk menampilkan daftar panjang atau grid data dalam format yang efisien. RecyclerView mendaur ulang item yang tidak lagi terlihat di layar untuk menghemat memori dan meningkatkan performa.
- RecyclerView memerlukan Adapter dan ViewHolder untuk menentukan bagaimana data ditampilkan dan dikelola.

PACKAGE, PROJECT STRUCTURE, ANDROIDMANIFEST.XML

Pengertian Komponen UI

Komponen UI adalah elemen-elemen yang digunakan untuk membangun tampilan antarmuka pengguna di aplikasi Android.



PACKAGE

Pengertian Package

Package di Android adalah sebuah cara untuk mengelompokkan dan mengorganisir kode-kode program dalam bentuk modul atau direktori. Ini membantu dalam menjaga struktur kode yang bersih, terorganisir, dan dapat dikelola dengan baik. Setiap package mengandung satu atau lebih file kelas (Java/Kotlin) yang terkait secara logis.



PACKAGE

Hierarki Package

Package di Android mengikuti struktur hierarkis yang mirip dengan direktori pada sistem file. Setiap package biasanya ditulis dalam format terbalik dari nama domain (misalnya, `com.example.myapp`) untuk memastikan bahwa package memiliki nama yang unik di seluruh aplikasi.



TUJUAN DAN MANFAAT PENGGUNAAN PACKAGE

Organisasi Kode

Membantu mengelompokkan kode yang memiliki fungsionalitas serupa, misalnya `com.example.myapp.ui` untuk elemen UI, `com.example.myapp.data` untuk manajemen data, dan sebagainya.

Modularitas

Membantu dalam pembuatan aplikasi yang modular, dimana setiap package dapat dikerjakan, diuji, dan diperbarui secara independen.

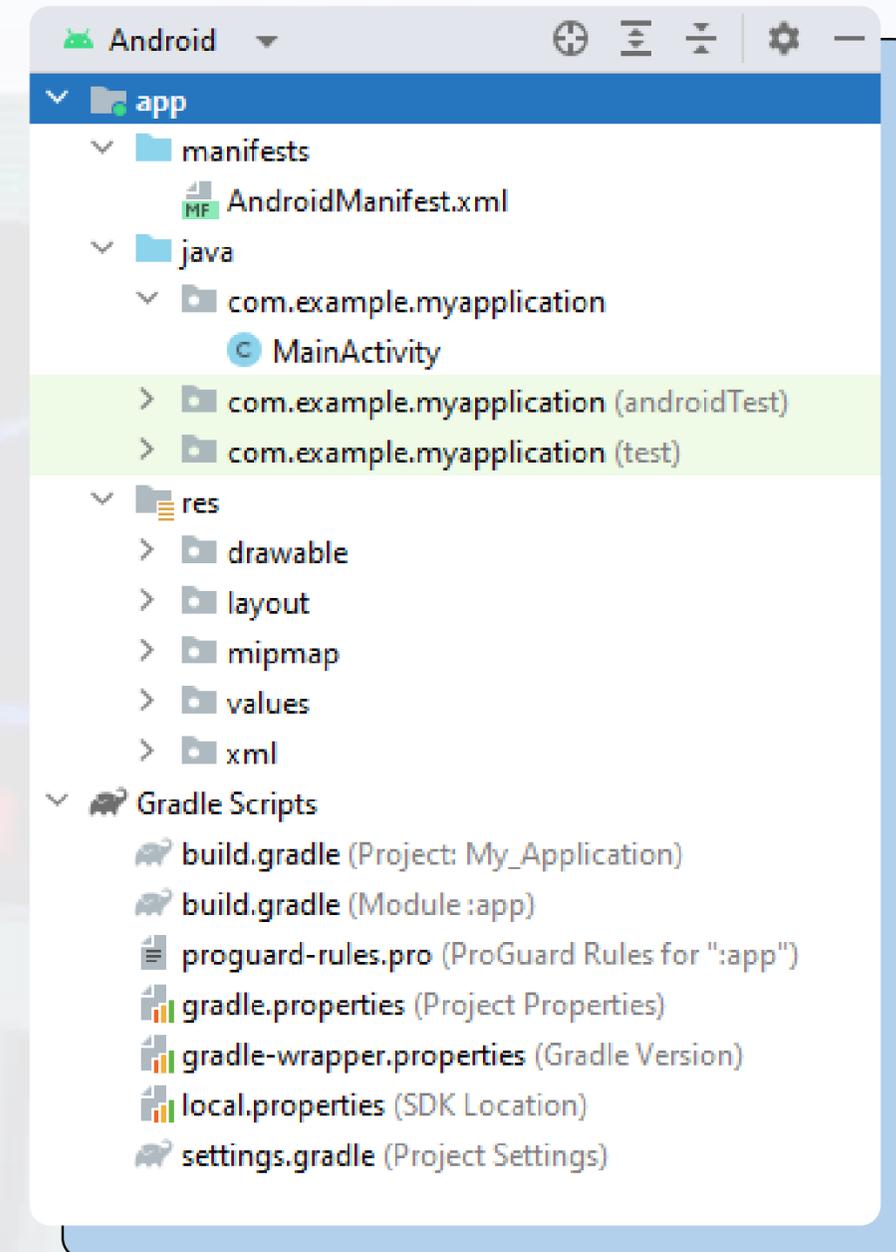
Pemeliharaan Kode

Memudahkan pengelolaan dan pemeliharaan kode dalam proyek yang besar dengan membaginya menjadi bagian-bagian yang lebih kecil dan lebih spesifik

PROJECT STRUCTURE

Pendahuluan

Struktur proyek Android terdiri dari berbagai direktori dan file yang mendukung pengembangan aplikasi. Setiap bagian memiliki fungsi tertentu untuk mengatur kode sumber, sumber daya (resources), konfigurasi, dan pengaturan build. Di Android Studio, struktur proyek terbagi menjadi beberapa direktori utama yang meliputi kode Java/Kotlin, resource, file manifest, dan file build.



DIREKTORI UTAMA DALAM PROYEK ANDROID

Organisasi Kode

Membantu mengelompokkan kode yang memiliki fungsionalitas serupa, misalnya `com.example.myapp.ui` untuk elemen UI, `com.example.myapp.data` untuk manajemen data, dan sebagainya.

Modularitas

Membantu dalam pembuatan aplikasi yang modular, dimana setiap package dapat dikerjakan, diuji, dan diperbarui secara independen.

Pemeliharaan Kode

Memudahkan pengelolaan dan pemeliharaan kode dalam proyek yang besar dengan membaginya menjadi bagian-bagian yang lebih kecil dan lebih spesifik

ANDROIDMANIFEST.XML

Fungsi AndroidManifest.xml:

File konfigurasi yang berisi informasi penting tentang aplikasi seperti activity, permissions, dan metadata aplikasi.

Elemen-Elemen Penting:

- `<application>`: Elemen root yang mendeklarasikan keseluruhan aplikasi.
- `<activity>`: Mendeklarasikan aktivitas di dalam aplikasi.
- `<uses-permission>`: Mendeklarasikan izin yang diperlukan oleh aplikasi.

KONSEP ACTIVITY

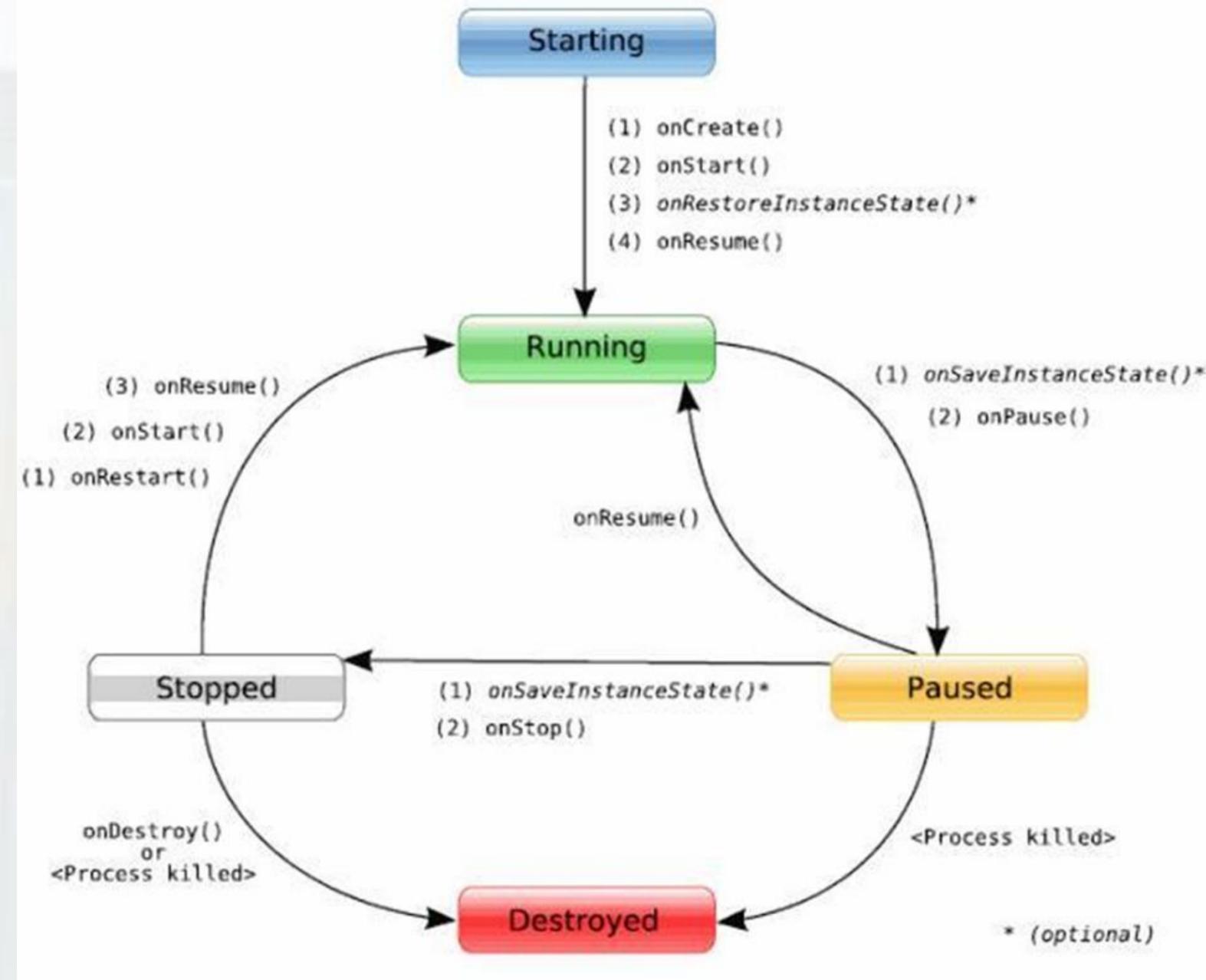
Activity adalah komponen utama dalam aplikasi Android yang bertindak sebagai layar atau halaman di aplikasi.

Mekanisme Activity:

- Lifecycle : Bagaimana sebuah activity diciptakan, dijalankan, di-pause, dihentikan, dan dihancurkan.
- Intent : Mekanisme untuk berpindah antar activity atau mengirim data antar activity.

SIKLUS HIDUP ACTIVITY

- onCreate(): Dipanggil saat activity pertama kali dibuat.
- onStart(): Dipanggil saat activity akan ditampilkan ke pengguna.
- onResume(): Dipanggil saat activity mulai berinteraksi dengan pengguna.
- onPause(): Dipanggil saat aktivitas akan di-pause (misalnya saat berpindah ke activity lain).
- onStop(): Dipanggil saat aktivitas tidak lagi terlihat oleh pengguna.
- onDestroy(): Dipanggil saat aktivitas akan dihancurkan.





se k i a n

TERIMA KASIH



Oleh : Kristina Annatasia Br Sitepu, M.Kom

