



# Mobile Programming

## Catatan Kuliah #13

Alauddin Maulana Hirzan, M. Kom

0607069401

The background consists of two large, overlapping geometric shapes. A teal-colored shape is in the upper-left corner, and a light beige shape is in the lower-left corner. The rest of the background is white. The word 'Kompatibilitas' is centered in the white area.

Kompatibilitas



# Kompatibilitas

## Apa itu **Kompatibilitas**?

Kompatibilitas Android mengacu pada kemampuan aplikasi Android untuk berjalan pada berbagai versi sistem operasi Android dan pada berbagai perangkat Android. Android adalah platform sumber terbuka dengan ekosistem perangkat yang beragam, masing-masing dengan spesifikasi perangkat keras, ukuran layar, dan konfigurasi perangkat lunaknya sendiri.

Memastikan kompatibilitas sangat penting bagi pengembang untuk menjangkau basis pengguna yang lebih besar dan memberikan pengalaman yang konsisten di berbagai perangkat.

# Kompatibilitas

Kompatibilitas = Manjangkau







# Kompatibilitas

## Aspek Kompatibilitas

Terdapat beberapa aspek dalam menentukan jangkauan **Kompatibilitas** seperti:

- ▶ Level API SDK Android
- ▶ Ukuran Layar dan Resolusi
- ▶ Perangkat Keras dan Sensor
- ▶ Platform
- ▶ Pustaka Pihak Ketiga



# Kompatibilitas

## Aspek Kompatibilitas - **Level API SDK Android**

Aspek ini merupakan aspek yang paling krusial karena menentukan versi Android yang dapat menggunakan aplikasi tersebut. Secara **default**, Android Studio akan menggunakan versi **tertinggi** dari **SDK** yang terpasang. Namun dalam menentukan versi **terendah** yang dapat dijangkau oleh aplikasi adalah **programmer**.

Sederhananya:

**Android Studio** menggunakan **API SDK Terbaru**, dan **Programmer** menentukan **Versi Terendah** yang dapat dijangkau aplikasi



# Kompatibilitas

## Aspek Kompatibilitas - Level API SDK Android

Sebagai contoh:

- ▶ Seorang programmer membuat aplikasi dengan **SDK 33** (tertinggi saat itu)
- ▶ Di dalam proyeknya, programmer tersebut **membatasi** hanya aplikasi dengan **SDK 28** saja yang bisa menggunakan aplikasi tersebut
- ▶ **Sehingga:** Android dengan **SDK di bawah 28 (<28)** tidak dapat memasang aplikasi tersebut
- ▶ **Solusi:** Upgrade Android



# Kompatibilitas

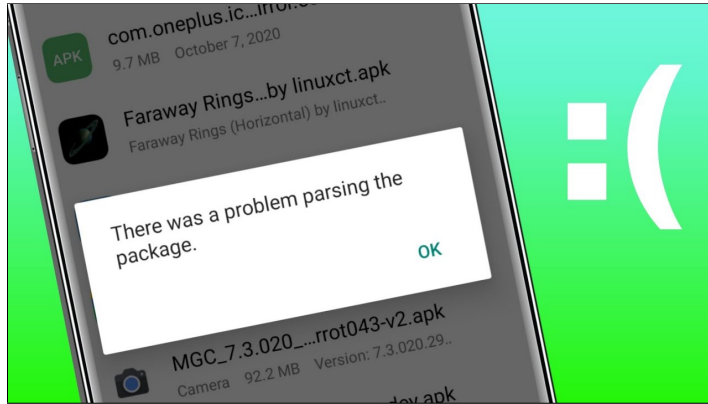
## Aspek Kompatibilitas - **Level API SDK Android**

### Cara Kerja Kompatibilitas API:

Android menyediakan pustaka kompatibilitas, seperti Pustaka Dukungan Android (sekarang digantikan oleh AndroidX) dan pustaka Jetpack, yang menawarkan kompatibilitas ke belakang untuk fitur dan API yang lebih baru pada versi Android yang lebih lama. Pustaka ini memungkinkan pengembang untuk mengakses fungsionalitas yang lebih baru sambil memastikan kompatibilitas dengan perangkat yang menjalankan versi Android yang lebih lama.

# Kompatibilitas

## Error Parsing APK Menjadi Indikasi Tidak Ada Dukungan Versi





# Kompatibilitas

## Aspek Kompatibilitas - **Level API SDK Android**

Dampak negatif dari Kompatibilitas yang terlalu luas berupa:

- ▶ Akses Terbatas ke Fitur Baru
- ▶ Peningkatan Kompleksitas Pengembangan
- ▶ Keterbatasan Kinerja
- ▶ Potensi Bug dan Masalah
- ▶ Peningkatan Tantangan Fragmentasi Perangkat Keras



# Kompatibilitas

## Aspek Kompatibilitas - **Ukuran Layar dan Resolusi**

Tantangan berikutnya adalah ukuran dan resolusi layar. Semakin terpecahnya perangkat menjadi berbagai macam jenis seperti **Smart Watch** dan **Smart TV**, maka kompatibilitas untuk perangkat jenis ini pun perlu dipertimbangkan.

Gawai akan memiliki layar yang semakin padat (DPI tinggi)

Smart Watch memiliki layar yang kecil

Smart TV memiliki layar yang besar dan kepadatan tinggi



# Kompatibilitas

## Aspek Kompatibilitas - Ukuran Layar dan Resolusi

Sehingga dapat pengembangan aplikasi yang tidak sesuai dapat menyebabkan:

- ▶ Layar yang besar dengan kepadatan tinggi dapat mengakibatkan aplikasi terlihat sangat kecil
- ▶ Layar yang kecil dapat mengakibatkan aplikasi biasa terlihat sangat besar

Maka untuk mengatasi hal ini, digunakanlah standar untuk membantu pengembang aplikasi Android lintas layar





# Kompatibilitas

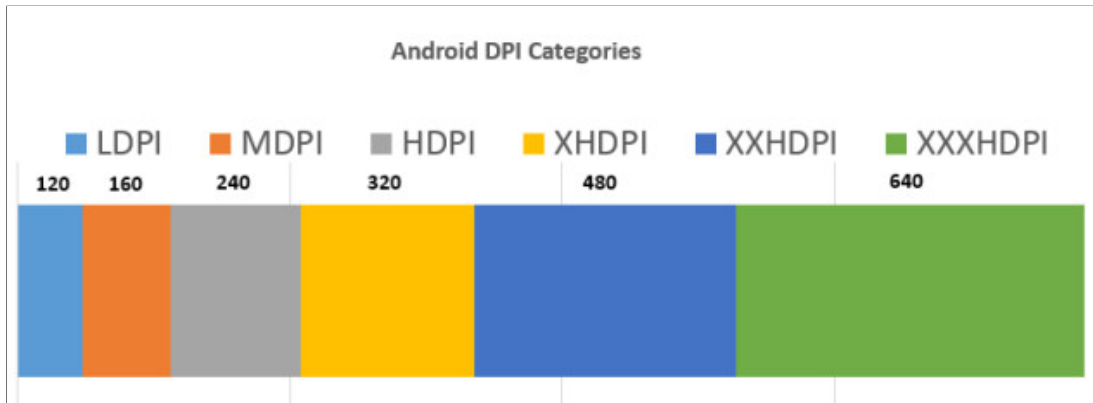
## Aspek Kompatibilitas - Ukuran Layar dan Resolusi

Standar ukuran layar secara internasional:

- ▶ Layar kecil (hingga 4 inci)
  - ▶ LDPI (Low DPI): sekitar 120 DPI
- ▶ Layar sedang (4 hingga 6 inci)
  - ▶ MDPI (Medium DPI): sekitar 160 DPI
  - ▶ HDPI (High DPI): sekitar 240 DPI
- ▶ Layar besar (6 hingga 9 inci):
  - ▶ XHDPI (eXtra High DPI): sekitar 320 DPI
- ▶ XXHDPI (eXtra-eXtra-High DPI): sekitar 480 DPI
- ▶ XXXHDPI (eXtra-eXtra-eXtra-High DPI): sekitar 640 DPI
- ▶ Layar ekstra besar (9 inci ke atas):
  - ▶ Bervariasi secara signifikan berdasarkan jenis perangkat dan penggunaan

# Kompatibilitas

## Aspek Kompatibilitas - Ukuran Layar dan Resolusi





# Kompatibilitas

## Aspek Kompatibilitas - **Perangkat Keras dan Sensor**

Faktor lainnya yang menentukan adalah perangkat keras itu sendiri. Banyak manufaktur Gawai yang membuat berbagai macam perangkat keras dengan berbagai chip teknologinya. Ada yang menggunakan SOC Mediatek, maupun SnapDragon.

Meskipun berbeda SoC nya, programmer tetap wajib mengakomodasi semua perangkat keras yang dirilis oleh manufaktur.



# Kompatibilitas

## Aspek Kompatibilitas - Perangkat Keras dan Sensor

Berkat adanya **Hardware Abstraction Layer** yang ada di dalam arsitektur Android, membantu programmer dalam membuat aplikasi lintas perangkat.

Dengan adanya **HAL** ini, programmer tidak perlu memikirkan jenis perangkat maupun sensornya. Programmer hanya cukup mengetahui API yang disediakan oleh Android untuk mengakses perangkat.

# Kompatibilitas

## Aspek Kompatibilitas - Perangkat Keras dan Sensor





# Kompatibilitas

## Aspek Kompatibilitas - Platform

Semakin majunya teknologi, mendorong banyak perangkat untuk mengadopsi Android ke perangkat mereka. Banyak sekali perangkat yang memasukkan Android ke perangkat mereka seperti:

- ▶ Desktop
- ▶ Smart Watch
- ▶ Smart TV
- ▶ Set Top Box
- ▶ Mobil
- ▶ Smart Refrigerator



# Kompatibilitas

## Aspek Kompatibilitas - Platform

Untuk mengatasi hal ini, Android Studio sudah memberikan pilihan untuk pembuatan proyek Android di awal prosesnya. Programmer dapat memilih apakah user ingin membuat aplikasi Smart Watch, Android, Tablet, maupun TV.

Template dasar ini akan secara otomatis menentukan ukuran layar dari aplikasi yang dibuat.w

# Kompatibilitas

## Aspek Kompatibilitas - Platform







# Kompatibilitas

## Aspek Kompatibilitas - Pihak Ketiga

Aspek ini salah satu aspek yang paling kompleks. Berbeda dengan aplikasi Android murni yang hanya mengandalkan API internal Android, aplikasi yang mengandalkan API pihak ketiga mau tidak mau harus mempertimbangkan kompatibilitas API tersebut.

Jika API Android bisa diatur otomatis, maka API pihak ketiga harus dicek dan diuji terlebih dahulu apakah API yang disediakan dapat digunakan untuk perangkat jenis lama.

Hal ini akan menjadi semakin kompleks ketika aplikasi mengandalkan banyak API luar dan membentuk **dependency tree** yang mendalam.



# Kompatibilitas

## Aspek Kompatibilitas - Pihak Ketiga

```
+--- com.facebook.fresco:fresco:0.9.0
|   +--- com.facebook.fresco:drawee:0.9.0
|       +--- com.android.support:support-v4:21.0.3 -> 23.4.0 (*)
|       \--- com.facebook.fresco:fbcore:0.9.0
+--- com.facebook.fresco:fbcore:0.9.0
|   \--- com.facebook.fresco:imagepipeline:0.9.0
|       +--- com.nineoldandroids:library:2.4.0
|       +--- com.android.support:support-v4:21.0.3 -> 23.4.0 (*)
|       +--- com.parse.bolts:bolts-android:1.1.4
|       +--- com.facebook.fresco:fbcore:0.9.0
|       \--- com.facebook.fresco:imagepipeline-base:0.9.0
|           +--- com.nineoldandroids:library:2.4.0
|           +--- com.android.support:support-v4:21.0.3 -> 23.4.0 (*)
|           +--- com.parse.bolts:bolts-android:1.1.4
|           \--- com.facebook.fresco:fbcore:0.9.0
```



# Kompatibilitas

## Aspek Kompatibilitas - Pihak Ketiga

Meskipun sudah menggunakan teknologi Sistem Build dengan Gradle, tetapi tidak menjamin dependency yang digunakan akan kompatibel dengan API target. Oleh karena itu, penting sekali untuk menguji aplikasi yang dibuat dengan dependency-dependency nya

### Informasi

Ketika dependency sudah berada yang sulit untuk diselesaikan karena konflik dengan dependency lainnya, maka mau tidak mau programmer harus menaikkan versi terakhir yang didukung aplikasi.

