ß

Implementasi admob & upload aplikasi play store

School of SMK Negeri 7 Samarinda

August 2019

Android Developer

Fundamental Course

Belajar membuat aplikasi android pemula







**Dasar – Dasar Developer Android – Konsep**

**Belajar mengembangkan aplikasi android**

**Refrensi Konsep**

*Dikembangkan oleh Tim Pelatihan Developer Google*



**Bagian 1 : Pengantar Android**

**Apa yang dimaksud dengan Android?**

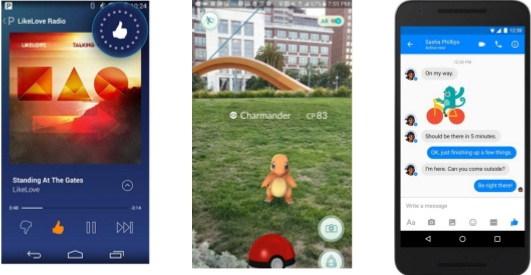
Android adalah sistem operasi dan platform pemrograman yang dikembangkan oleh Google untuk ponsel cerdas dan perangkat seluler lainnya (seperti tablet). Android bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. Android menyertakan kit development perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna Android. Android juga menyediakan pasar untuk mendistribusikan aplikasi. Secara keseluruhan, Android menyatakan ekosistem untuk aplikasi seluler.

## 

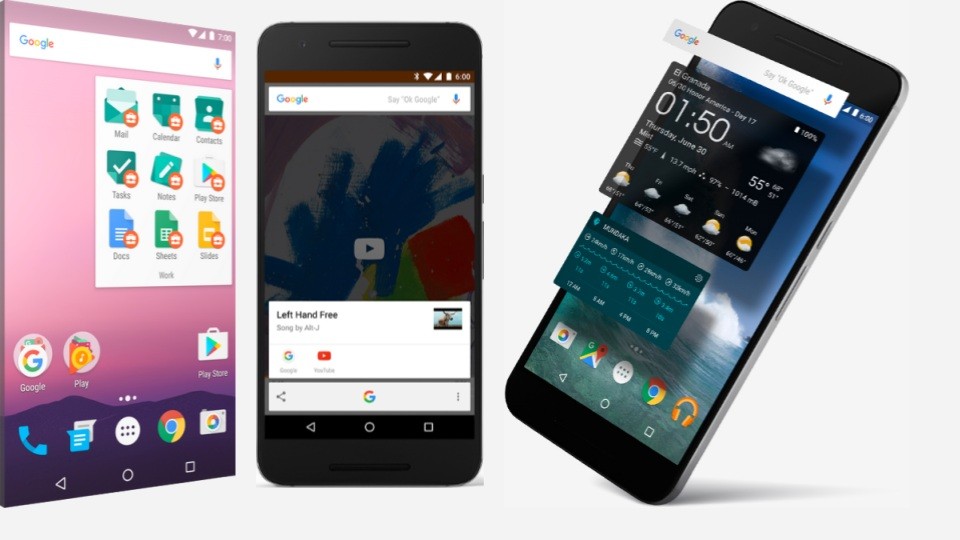
## Mengapa mengembangkan aplikasi untuk Android?

Aplikasi dikembangkan untuk berbagai alasan: menjawab kebutuhan bisnis, membangun layanan baru, membuat bisnis baru, dan menyediakan game serta jenis materi lainnya untuk pengguna. Developer memilih untuk mengembangkan bagi Android agar bisa menjangkau sebagian besar pengguna perangkat seluler.

### Platform paling populer untuk aplikasi seluler



**Pengalaman terbaik untuk pengguna aplikasi**

Android menyediakan antarmuka pengguna (UI) layar sentuh untuk berinteraksi dengan aplikasi. Antarmuka pengguna Android sebagian besar berdasarkan pada manipulasi langsung, menggunakan isyarat sentuhan seperti menggesek, mengetuk, dan mencubit untuk memanipulasi objek di layar. Selain keyboard, ada keyboard virtual yang bisa disesuaikan untuk masukan teks. Android juga bisa mendukung pengontrol game dan keyboard fisik berukuran penuh yang dihubungkan dengan Bluetooth atau USB.

Layar utama Android bisa berisi sejumlah laman *ikon aplikasi*, yang akan meluncurkan aplikasi terkait, dan *widget*, dengan menampilkan materi langsung yang diperbarui secara otomatis seperti cuaca, kotak masuk email pengguna, atau ticker berita.

Android juga bisa memutar materi multimedia seperti musik, animasi, dan video. Gambar di atas menampilkan ikon aplikasi pada layar utama (kiri), musik yang diputar (tengah), dan widget yang ditampilkan (kanan). Sepanjang bagian atas layar terdapat bilah status, yang menampilkan informasi tentang perangkat dan konektivitasnya. Layar utama Android bisa terdiri dari sejumlah laman, yang bisa digesek mundur dan maju oleh pengguna.

Android didesain untuk menyediakan respons cepat terhadap masukan pengguna. Selain antarmuka sentuh yang berubah- ubah, kemampuan getaran perangkat Android bisa menyediakan umpan balik sentuhan. Perangkat keras internal seperti akselerometer, giroskop, dan sensor kedekatan, digunakan oleh banyak aplikasi untuk merespons tindakan pengguna Android didesain untuk menyediakan respons cepat terhadap masukan pengguna. Selain antarmuka sentuh yang berubah- ubah, kemampuan getaran perangkat Android bisa menyediakan umpan balik sentuhan. Perangkat keras internal seperti akselerometer, giroskop, dan sensor kedekatan, digunakan oleh banyak aplikasi untuk merespons tindakan pengguna tambahan. Sensor tersebut bisa mendeteksi rotasi layar dari potret ke lanskap untuk tampilan yang lebih lebar atau sensor bisa memungkinkan pengguna untuk menyetir kendaraan virtual dengan memutar perangkat seolah-olah setir mobil.

Platform Android, berdasarkan pada kernel Linux, terutama didesain untuk perangkat seluler layar sentuh seperti ponsel cerdas dan tablet. Karena perangkat Android biasanya bertenaga baterai, Android didesain untuk mengelola proses guna menjaga konsumsi daya tetap minimum, sehingga menyediakan penggunaan baterai lebih lama.

# **Bagian 2 : Hasilkan uang dari aplikasi Anda**

Membuat aplikasi dan melihatnya berjalan adalah hal menarik. Namun bagaimana cara menghasilkan uang dari aplikasi Anda?

Pertama, Anda perlu membuat aplikasi yang bekerja dengan baik, cukup cepat, tidak mogok, dan berguna atau menghibur. Aplikasi harus menarik sehingga pengguna tidak hanya ingin memasang dan menggunakannya, melainkan ingin terus menggunakannya.

Dengan anggapan aplikasi Anda sudah sempurna dan menyediakan fitur berguna, menghibur, atau menarik, ada beragam cara menghasilkan uang dari aplikasi tersebut.

### Cara menghasilkan uang

Model monetisasi:

* Model Premium - pengguna membayar untuk mengunduh aplikasi.
* Model Freemium:
* gratis mengunduh aplikasi.
* pengguna membayar untuk peningkatan versi atau pembelian dalam aplikasi.
* Berlangganan - pengguna membayar biaya berkala untuk aplikasi.
* Periklanan - aplikasi gratis namun menampilkan iklan.

### Aplikasi premium

Pengguna membayar di depan untuk mengunduh aplikasi premium. Untuk aplikasi yang menyediakan fungsiolitas yang diinginkan kepada sedikit pengguna yang sangat ditargetkan, menyertakan harga ke aplikasi bisa memberikan sumber pendapatan. Ketahuilah bahwa sebagian pengguna akan menolak mengunduh aplikasi jika harus membayar, atau jika tidak bisa mencobanya terlebih dahulu secara gratis. Jika pengguna bisa menemukan aplikasi lainnya yang serupa dan tersedia gratis atau lebih rendah, mereka mungkin lebih memilih untuk mengunduh dan mencoba tersebut.

### Aplikasi freemium

Aplikasi "freemium" adalah kompromi antara aplikasi yang sepenuhnya gratis dengan aplikasi yang mengenakan biaya pemasangan. Aplikasi tersebut tersedia untuk pemasangan gratis, baik dengan fungsionalitas terbatas maupun selama durasi terbatas. Sasaran Anda untuk aplikasi freemium seharusnya untuk meyakinkan pengguna terhadap nilai aplikasi tersebut, sehingga setelah menggunakannya untuk sementara waktu, mereka akan bersedia membayar agar tetap bisa menggunakannya atau untuk meningkatkan versi demi mendapatkan fitur lainnya. Pernahkah Anda mengunduh aplikasi gratis, kemudian membayar untuk meningkatkan versi demi fungsionalitas? Apa yang Anda sukai pada aplikasi tersebut sehingga bersedia membayarnya?

Cara lain untuk menghasilkan uang dari aplikasi freemium adalah menyediakan pembelian dalam aplikasi. Untuk game, aplikasi Anda mungkin menawarkan level materi baru dalam game, atau item baru untuk membuatnya jadi lebih menyenangkan. Pikirkan tentang game seluler yang pernah Anda mainkan. Manakah yang menawarkan pembelian dalam aplikasi dan untuk apa? Pernahkah Anda membuat pembelian dalam aplikasi di aplikasi seluler?

### Berlangganan

Pada model berlangganan, pengguna membayar biaya berkala untuk menggunakan aplikasi. Ini sangat mirip dengan model premium, hanya saja pengguna membayar pada siklus penagihan teratur, bukan sekali saja pada waktu pemasangan. Anda bisa mempersiapkan langganan agar pengguna membayar setiap bulan atau setiap tahun. Jika Anda menyediakan aplikasi secara berlangganan, pertimbangkan penawaran pembaruan materi teratur atau beberapa layanan lainnya yang menjamin pembayaran berulang.

Jika Anda memutuskan untuk menawarkan aplikasi dengan model berlangganan, sebaiknya biarkan pengguna mencobanya secara gratis. Banyak pengguna akan menolak memasang aplikasi yang membuat mereka harus membayar sebelum mencoba untuk melihat apakah aplikasi itu sesuai dengan kebutuhan.

### Periklanan

Salah satu strategi monetisasi umum adalah menyediakan aplikasi gratis, namun menjalankan iklan di dalamnya. Pengguna bisa menggunakan aplikasi Anda sebanyak yang mereka suka, namun sesekali aplikasi akan menampilkan iklan.

Jika aplikasi Anda menampilkan iklan, selalu pertimbangkan pengguna. Jika aplikasi Anda menampilkan begitu banyak iklan yang mengganggu pengguna, mereka mungkin berhenti menggunakannya atau mencopot pemasangannya.

Mudah sekali menambahkan iklan ke aplikasi. Cara terbaik untuk memasukkan iklan ke dalam aplikasi Anda adalah menggunakan AdMob.

# **AdMob**

Google menyediakan alat bagi pengiklan untuk membuat iklan dan mendefinisikan kriteria target iklan mereka. Untuk contoh kriteria target, iklan dapat ditargetkan untuk ditampilkan kepada orang-orang di lokasi tertentu. Google memiliki banyak persediaan iklan untuk ditampilkan di situs web dan aplikasi seluler. Anda bisa menampilkan iklan dari persediaan ini di aplikasi dengan menggunakan AdMob (singkatan dari Ads on Mobile).

Untuk menampilkan iklan di aplikasi, tambahkan AdView di layout aktivitas dan tulis sedikit kode boilerplate untuk memuat iklan. Bila pengguna menjalankan aplikasi Anda dan membuka aktivitas, iklan itu akan muncul di AdView. Anda tidak perlu mempersoalkan pencarian iklan yang akan ditampilkan karena Google yang akan menanganinya.

### Cara iklan membantu Anda menghasilkan uang

Google membayar Anda bila pengguna mengeklik iklan di aplikasi Anda. Jumlah persis yang dibayarkan kepada Anda bergantung pada iklan yang ditampilkan, dan seberapa banyak pengiklan bersedia membayar untuk iklan mereka. Jumlah total yang dibayarkan oleh pengiklan dibagi antara Google dan penerbit situs web atau aplikasi tempat iklan itu muncul.

### Jangan mengeklik iklan di aplikasi Anda sendiri

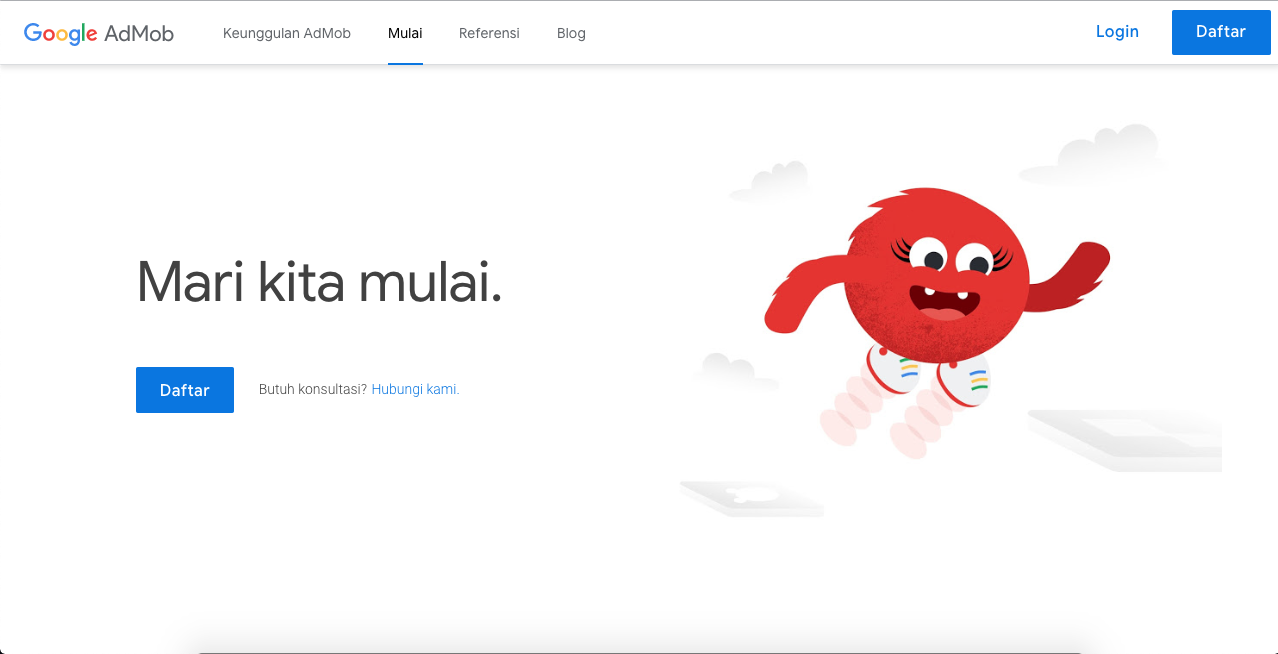
Google memiliki kebijakan yang mencegah penerbit situs web dan penerbit aplikasi mengeklik iklan di situs web dan aplikasi mereka sendiri. Pengiklan membayar bila orang mengklik iklan mereka, sehingga tidak adil bila Anda menampilkan iklan di aplikasi sendiri, kemudian mengekliknya, dan menyebabkan pengiklan membayar Anda karena mengklik iklan di aplikasi sendiri.

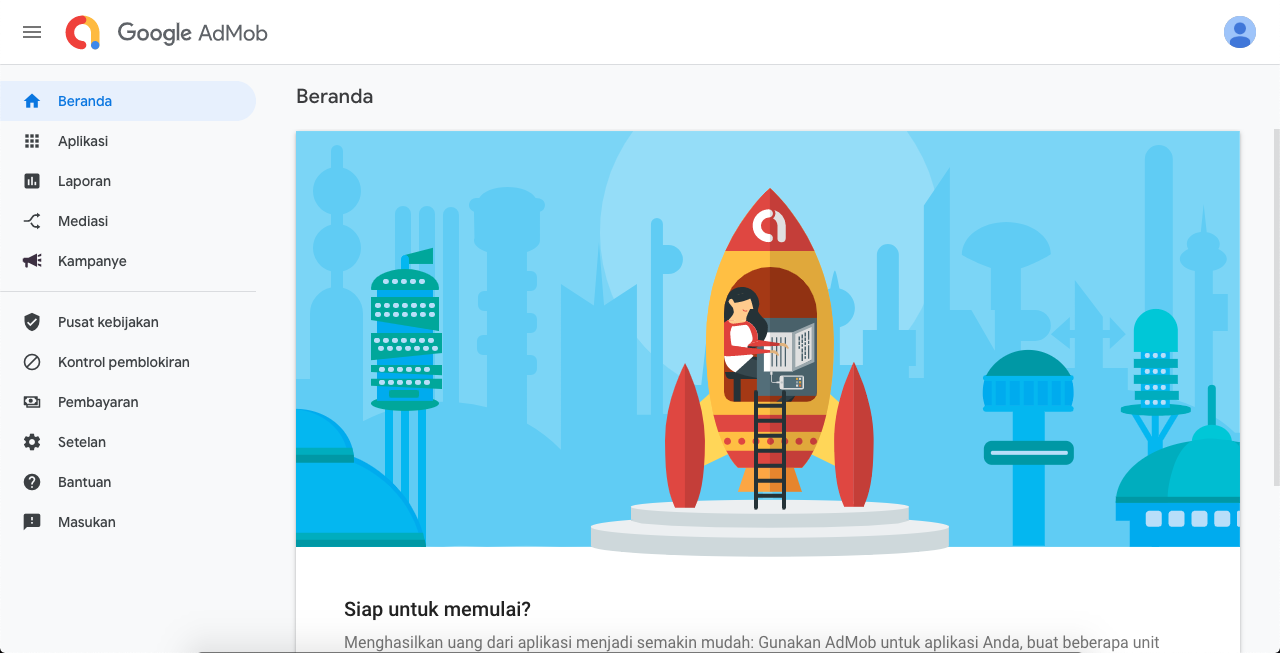
Baca selengkapnya tentang kebijakan AdMob di [pusat bantuan AdMob](https://support.google.com/admob/answer/2753860?hl=en&amp;ref_topic=2745287).

**Buat akun AdMob**

Sebelum bisa bereksperimen dengan menjalankan iklan di aplikasi, Anda perlu mengaktifkan AdMob untuk aplikasi itu. Untuk mengaktifkan AdMob, gunakan langkah-langkah ini:

1. Open link web **Admob** [**https://admob.google.com/home/**](https://admob.google.com/home/) untuk mendaftarkan email, Pilih **Mulai,** lalu pilih **daftar.**





1. Ikuti wizard pendaftaran untuk membuat akun AdMob dan menambahkan aplikasi ke AdMob.

Untuk menampilkan iklan di aplikasi, Anda memerlukan ID aplikasi AdMob dan ID unit iklan. Anda bisa mendapatkan keduanya di konsol AdMob.

## Implemantasikan AdMob di aplikasi Anda

Untuk menampilkan iklan di aplikasi:

1. Tambahkan AdView ke layout untuk aktivitas yang akan menampilkan iklan.
2. Lakukan inisialisasi iklan AdMob di peluncur aplikasi, dengan memanggil MobileAds.initialize() di metode onCreate() aktivitas utama Anda.
3. Perbarui onCreate() aktivitas itu untuk memuat iklan ke dalam AdView .

### Bersiaplah menjalankan pengujian iklan

Selagi mengembangkan dan menguji aplikasi, Anda bisa menampilkan dan menguji iklan untuk memastikan aplikasi telah dipersiapkan dengan benar untuk menampilkan iklan. Saat menguji iklan Anda perlu:

* ID (EMEI) perangkat Anda untuk menjalankan pengujian iklan. Untuk mendapatkan ID perangkat:
* Buka **Settings** > **About phone** > **status> IMEI.**
* Panggil nomor **\*#06#**.
* [ID aplikasi AdMob](https://support.google.com/admob/answer/6232340) Anda. Dapatkan di konsol AdMob.
* [ID unit iklan](https://support.google.com/admob/answer/3052638). Dapatkan di konsol AdMob.

### Tambahkan AdView untuk menampilkan iklan

### Format penerapan iklan

### AdMob menawarkan sejumlah format iklan yang berbeda, sehingga Anda dapat memilih salah satu yang paling sesuai dengan pengalaman pengguna aplikasi Anda.

Banner

Iklan spanduk adalah iklan gambar atau teks segi empat yang menempati tempat dalam tata letak aplikasi. Mereka tetap di layar saat pengguna berinteraksi dengan aplikasi, dan dapat menyegarkan secara otomatis setelah periode waktu tertentu. Jika Anda baru mengenal periklanan seluler, itu adalah tempat yang bagus untuk memulai.

### Interstitial

### Pengantara adalah iklan layar penuh yang menutupi antarmuka aplikasi hingga ditutup oleh pengguna. Mereka paling baik digunakan pada jeda alami dalam alur eksekusi aplikasi, seperti antara level game atau hanya setelah tugas selesai.

### Rewarded

### Iklan video yang dihargai adalah iklan video layar penuh yang pengguna dapat tonton secara keseluruhan dengan imbalan imbalan dalam aplikasi.

### Native

### Native adalah format iklan berbasis komponen yang memberi Anda kebebasan untuk menyesuaikan cara aset seperti berita utama dan ajakan bertindak disajikan dalam aplikasi Anda. Dengan memilih font, warna, dan detail lainnya untuk Anda sendiri, Anda dapat membuat presentasi iklan yang alami dan tidak mengganggu yang dapat menambah pengalaman pengguna yang kaya.

### Example : Penerapakan jenis jenis ikan admob

### 

### 

### Implementasi Iklan

Import Mobile Ads SDK

allprojects {  
    repositories {  
        google()  
        jcenter()  
    }  
}

dependencies {  
    implementation fileTree(dir: 'libs', include: ['\*.jar'])  
    implementation 'androidx.appcompat:appcompat:1.0.2'  
    **implementation 'com.google.android.gms:play-services-ads:18.1.1'**  
}

Update AndroidManifest.xml

<manifest>  
    <application>  
        **<!-- Sample AdMob App ID: ca-app-pub-3940256099942544~3347511713 -->  
        <meta-data  
            android:name="com.google.android.gms.ads.APPLICATION\_ID"  
            android:value="YOUR\_ADMOB\_APP\_ID"/>**  
    </application>  
</manifest>

**Tambahkan AdView untuk menampilkan iklan**

Di file layout aktivitas yang akan memunculkan iklan, tambahkan AdView :

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:tools="http://schemas.android.com/tools"  
        android:layout\_height="match\_parent"  
        android:layout\_width="match\_parent"  
        android:paddingLeft="@dimen/activity\_horizontal\_margin"  
        android:paddingRight="@dimen/activity\_horizontal\_margin"  
        android:paddingTop="@dimen/activity\_vertical\_margin"  
        android:paddingBottom="@dimen/activity\_vertical\_margin"  
        tools:context=".MainActivity">  
  
        <TextView android:text="Belajar Admob"  
            android:layout\_width="wrap\_content"  
            android:layout\_height="wrap\_content" />  
  
        **<com.google.android.gms.ads.AdView  
            xmlns:ads="http://schemas.android.com/apk/res-auto"  
            android:id="@+id/adView"  
            android:layout\_width="wrap\_content"  
            android:layout\_height="wrap\_content"  
            android:layout\_centerHorizontal="true"  
            android:layout\_alignParentBottom="true"  
            ads:adSize="BANNER"  
            ads:adUnitId="ID\_UNIT\_YOU\_ADMOB">  
        </com.google.android.gms.ads.AdView>**  
  
</RelativeLayout>

AdView adView = new AdView(this);  
mAdView = findViewById(R.id.*adView*);  
AdRequest adRequest = new AdRequest.Builder()

.addTestDevice("#########")

.build();  
mAdView.loadAd(adRequest);  
mAdView.setAdListener(new AdListener() {  
 @Override  
 public void onAdFailedToLoad(int errorCode) {  
 Toast.*makeText*(getApplicationContext(),"Ad Failed to load",Toast.*LENGTH\_SHORT*).show();  
 // Code to be executed when an ad request fails.  
 }  
});

**Sample ad units**

|  |  |
| --- | --- |
| Ad format | Sample ad unit ID |
| [Banner](https://developers.google.com/admob/android/banner#add_adview) | ca-app-pub-3940256099942544/6300978111 |
| [Interstitial](https://developers.google.com/admob/android/interstitial#create_an_interstitial_ad_object) | ca-app-pub-3940256099942544/1033173712 |
| [Interstitial Video](https://developers.google.com/admob/android/interstitial#create_an_interstitial_ad_object) | ca-app-pub-3940256099942544/8691691433 |
| [Rewarded Video](https://developers.google.com/admob/android/rewarded-video#request_rewarded_video_ad) | ca-app-pub-3940256099942544/5224354917 |
| [Native Advanced](https://developers.google.com/admob/android/native-advanced#build_an_adloader) | ca-app-pub-3940256099942544/2247696110 |
| [Native Advanced Video](https://developers.google.com/admob/android/native-advanced#build_an_adloader) | ca-app-pub-3940256099942544/1044960115 |

### Interstitial

private InterstitialAd mInterstitialAd;

mInterstitialAd = new InterstitialAd(this);  
mInterstitialAd.setAdUnitId("ca-app-pub-3940256099942544/8691691433");  
mInterstitialAd.loadAd(new AdRequest.Builder()

.addTestDevice("#########")

.build());  
  
Button btnInterstitia = findViewById(R.id.*btnInterstitial*);  
btnInterstitia.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mInterstitialAd.isLoaded()) {  
 mInterstitialAd.show();  
 } else {  
 Log.*d*("TAG", "The interstitial wasn't loaded yet.");  
 }  
 }  
});

### Rewarded

private RewardedVideoAd mRewardedVideoAd;

mRewardedVideoAd = MobileAds.*getRewardedVideoAdInstance*(this);  
mRewardedVideoAd.setRewardedVideoAdListener(this);  
  
mRewardedVideoAd.loadAd("ca-app-pub-9336656000959615/9093284059",  
 new AdRequest.Builder()

.addTestDevice("#########")

.build());  
  
Button btnReward = findViewById(R.id.*btnReward*);  
btnReward.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (mRewardedVideoAd.isLoaded()) {  
 mRewardedVideoAd.show();  
 }  
 }  
});

Refrensi Pembelajaran Admob:

* <https://developers.google.com/admob/android/quick-start>

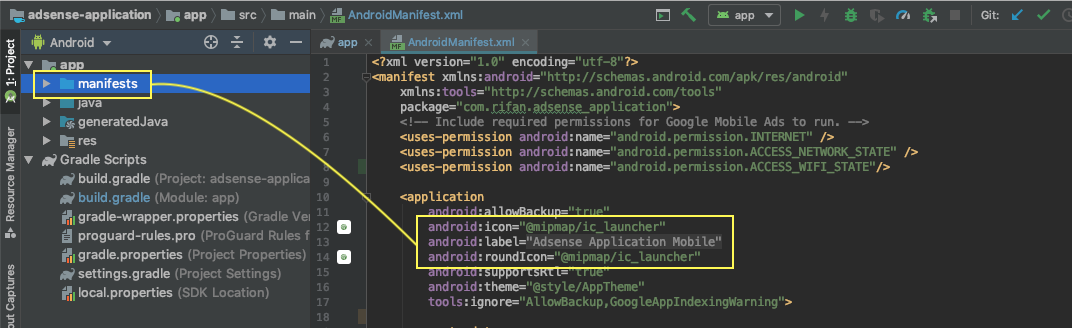
**Bagian 3 : Upload Aplikasi Ke Google Play Store**

Beberapa tahapan dan berkas yang di siapkan dalam upload aplikasi yang telah kita buat ke google play store.

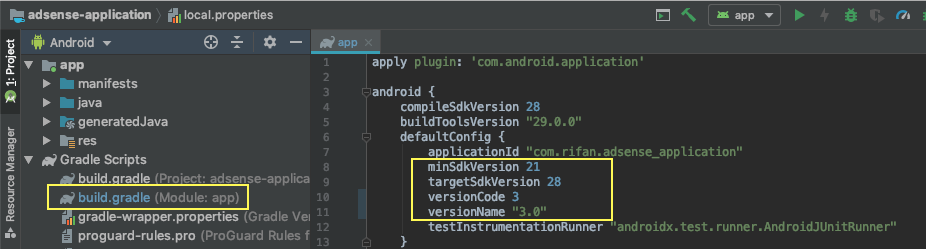
**Build Release App**

Tahap ini adalah tahap awal sebelum aplikasi di upload ke play store.

* Periksa semua Script tidak terdapat **ERROR.**
* Pastikan nama & icon aplikasi sudah sesuai dengan yang kita inginkan.



* Version & Target Android



## Android Studio

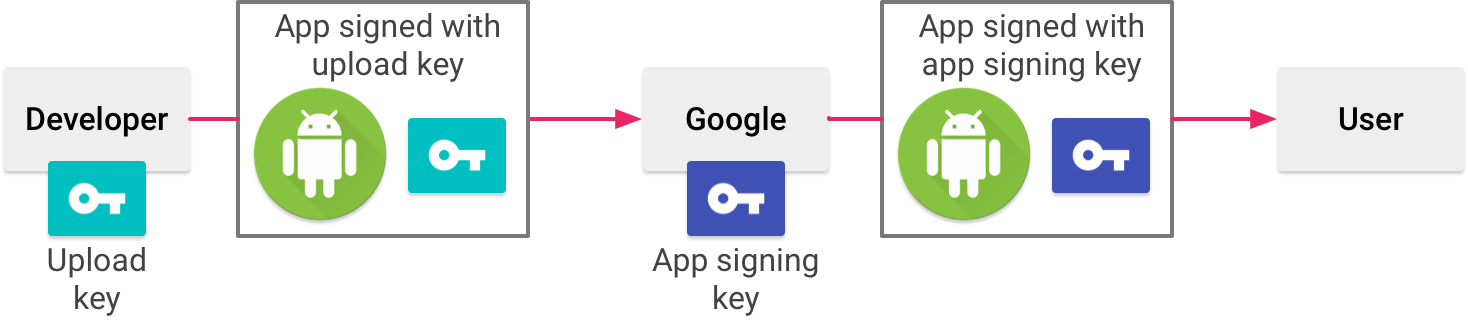
## Buat APK bertandatangan untuk rilis

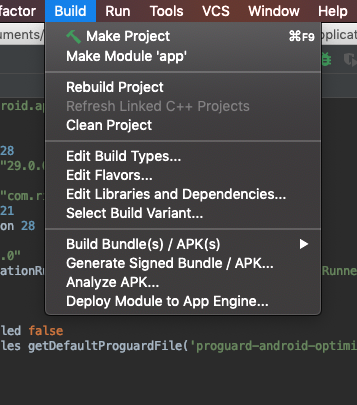
Bila aplikasi siap diunggah ke Google Play, Anda harus membuat dan menandatangani APK aplikasi. Android Studio memiliki alat untuk menghasilkan APK dan menandatanganinya dengan sertifikat digital.

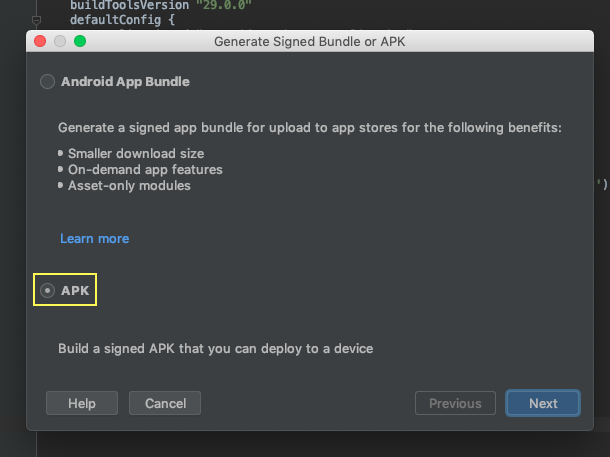
Bila menandatangani aplikasi, Android Studio akan membuat sertifikat publik dan kunci privat. Android Studio melampirkan sertifikat publik ke APK. Anda harus menyimpan kunci privat dalam keystore secara aman

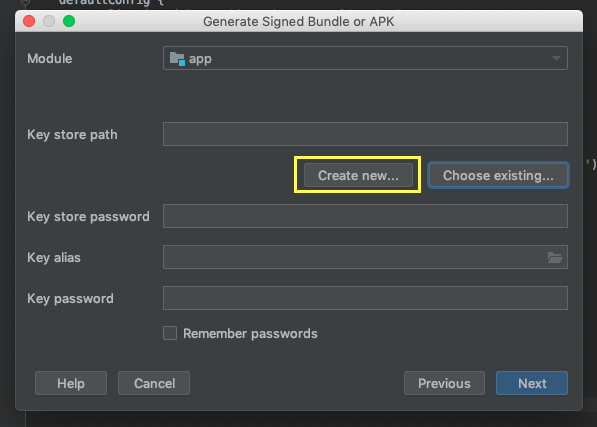
Sertifikat kunci publik berfungsi sebagai "sidik jari" yang secara unik mengaitkan APK dengan Anda dan kunci privat yang bersangkutan. Hal ini membantu Android memastikan bahwa pembaruan mendatang untuk APK adalah otentik dan berasal dari Anda, penulis aslinya.

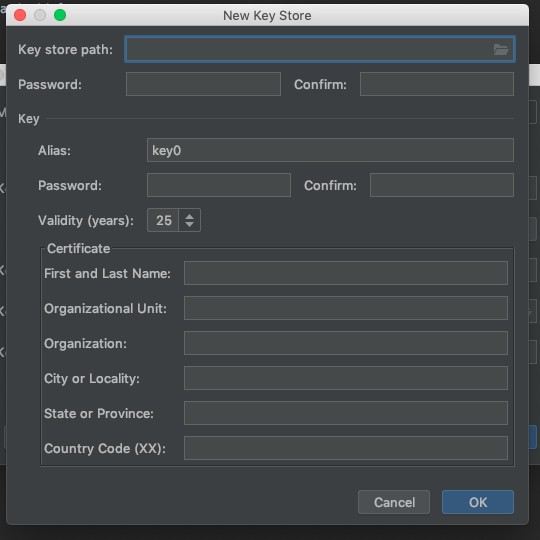
Untuk informasi tentang sertifikat digital, menyimpan kunci privat, dan menghasilkan APK bertandatangan digital, lihat panduan Tanda Tangani Aplikasi Anda.

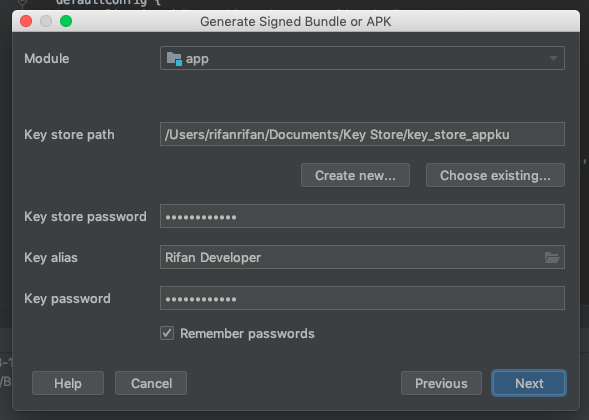


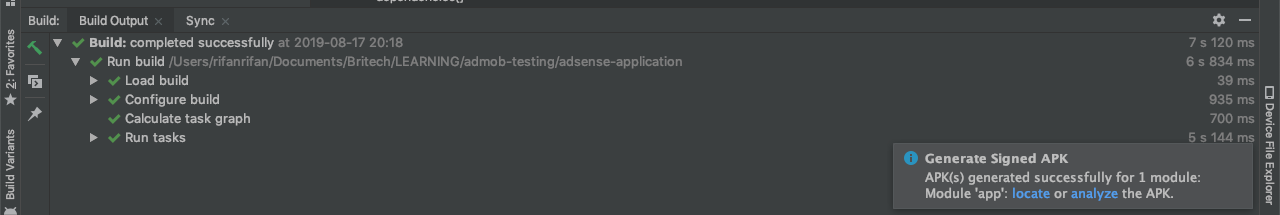












**MIT App Inventor**

**Keystore pada App Inventor**

Berbeda dengan aplikasi yang di buat dengan **Android Studio** di mana keystore dapat di buat secara langsung pada software, sedangkan untuk Keystore pada **MIT App Inventor,** keystoredi berikan setelah anda bulild aplikasi anda.

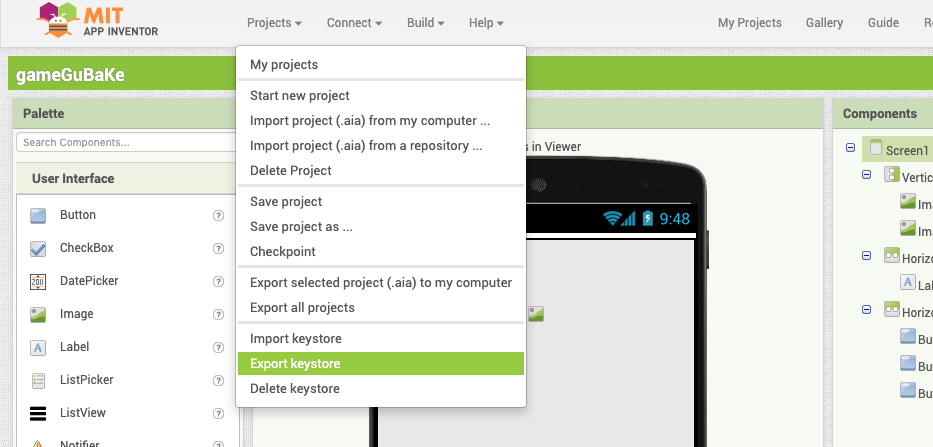


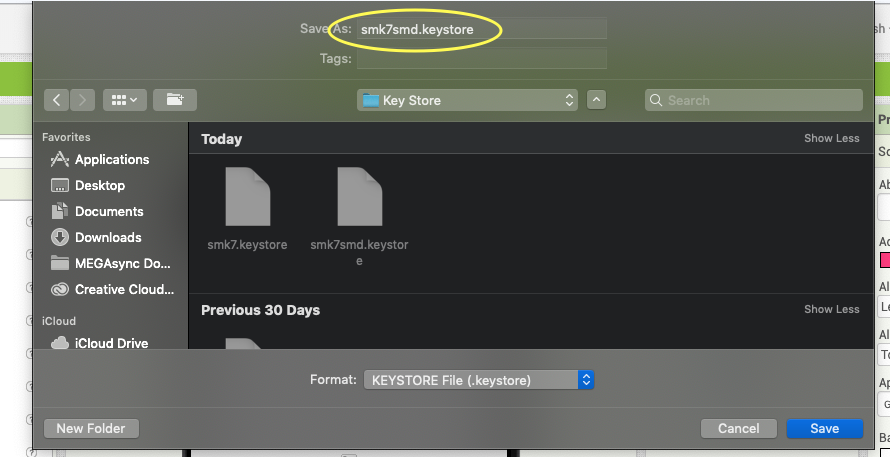
Sehingga keystore yang anda perlukan sudah tersedia untuk aplikasi yang anda buat. Sehingga keystore tidak perlu dibuat secara manual. Pastikan keystore yang dimiliki harus di simpan dengan baik, di karenakan pada saat anda mengupdate aplikasi yang ada di play store anda membutuhkan keystore yang digunakan sebelumnya.

**Backup Keystore**

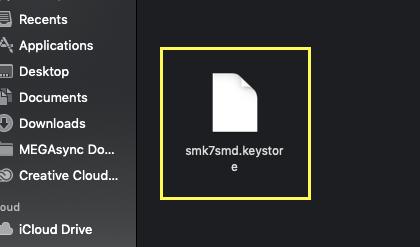
Setelah anda membuild aplikasi anda maka anda akan mendapatkan keystore, anda dapat menyimpan key store tersebut dengan cara:

* Klik Project pada menu toolbar
* Pilih export keystore
* Lalu simpan di dalam computer anda
* Simpan dengan format file .keystore





Maka kita akan mendapat kan satu back file keystore dari aplikasi yang kita buat



## Bagian 4 : Publikasikan aplikasi Anda!

Bila telah menguji aplikasi, membersihkannya, mengurangi ukurannya, dan menghasilkan APK, Anda siap mempublikasikannya ke Google Play.

Setelah mengunggah aplikasi ke Google Play, Anda bisa menjalankan pengujian alfa dan beta sebelum merilisnya ke publik. Menjalankan pengujian alfa dan beta memungkinkan Anda berbagi aplikasi dengan pengguna sungguhan, dan mendapatkan masukan dari mereka. Masukan ini tidak muncul sebagai ulasan di Google Play.

Jalankan pengujian alfa sewaktu Anda mengembangkan aplikasi. Gunakan pengujian alfa untuk versi eksperimen awal  dari aplikasi yang mungkin berisi fungsionalitas tidak lengkap atau tidak stabil. Menjalankan pengujian alfa juga merupakan cara yang baik untuk berbagi aplikasi Anda dengan teman dan keluarga.

Jalankan pengujian beta dengan pengguna sungguhan dalam jumlah terbatas, untuk melakukan pengujian akhir sebelum aplikasi dipublikasikan.

Setelah aplikasi dipublikasikan, pengguna bisa memberikan ulasan. Jadi, pastikan Anda menguji aplikasi secara menyeluruh sebelum dimasukkan ke Google Play untuk diunduh siapa saja.

Untuk informasi selengkapnya mengenai pengujian alfa dan beta, lihat:

* Panduan developer: developer.android.com/distribute/engage/beta.html
* Pusat bantuan: support.google.com/googleplay/android-develope[r/answ](https://www.google.com/)er/3131213

## Buat akun di Google Play Developer Console

Bila ingin menjalankan pengujian alfa dan beta, atau mempublikasikan aplikasi ke publik di Google Play, Anda perlu mengunggah APK di Google Play Developer Console.

Buka konsol di [**play.google.com/apps/publish/**](http://play.google.com/apps/publish/)

Untuk memulai, dapatkan akun developer Google Play. Anda perlu membayar untuk akun tersebut. Langkah-langkah tingkat tinggi adalah:

1. Buka [**play.google.com/apps/publish/**](http://play.google.com/apps/publish/)
2. Terima perjanjian.
3. Bayar biaya pendaftaran.
4. Masukkan detail, seperti nama, alamat, situs web, telepon, dan preferensi email Anda.

Bila telah mempersiapkan akun, Anda bisa mengunggah APK. Dalam antarmuka Google Play Developer Console, pilih:

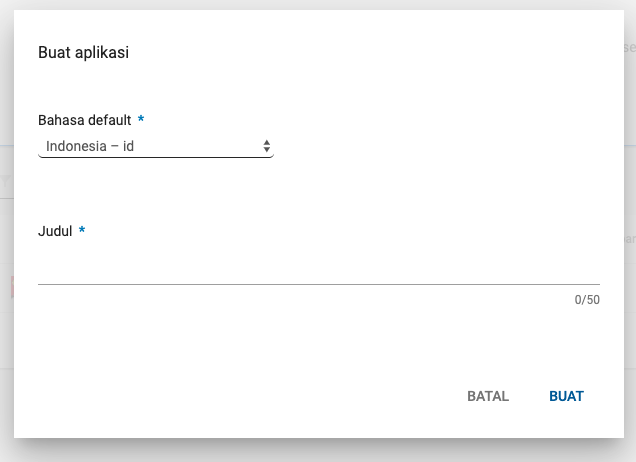
* Production
* Beta Testing
* Alpha Testing

Kemudian Anda bisa menjelajah ke APK yang akan diunggah, atau menyeret dan melepasnya ke konsol.

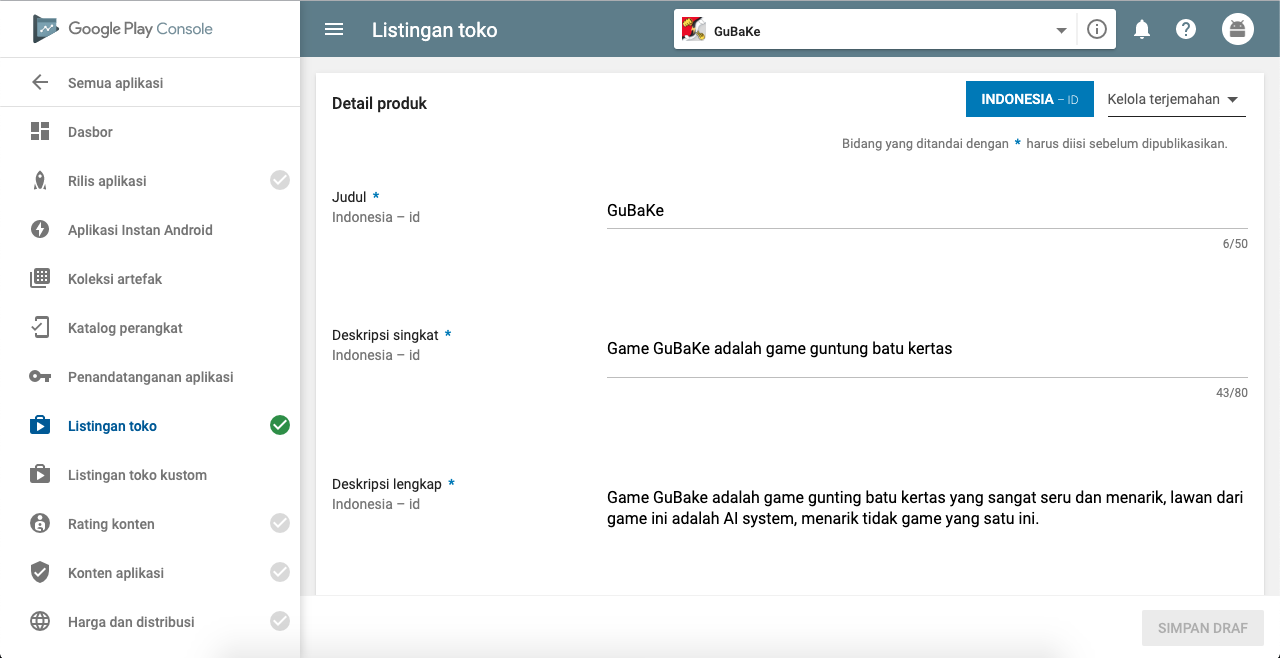
**Tahap melengkapi berkas aplikasi**

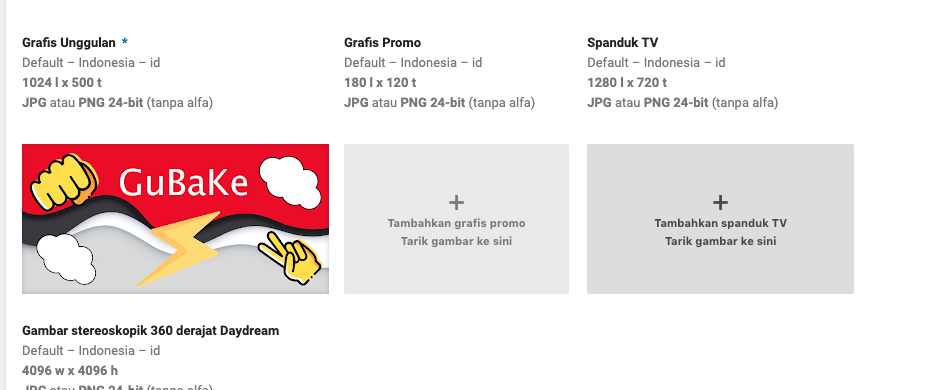
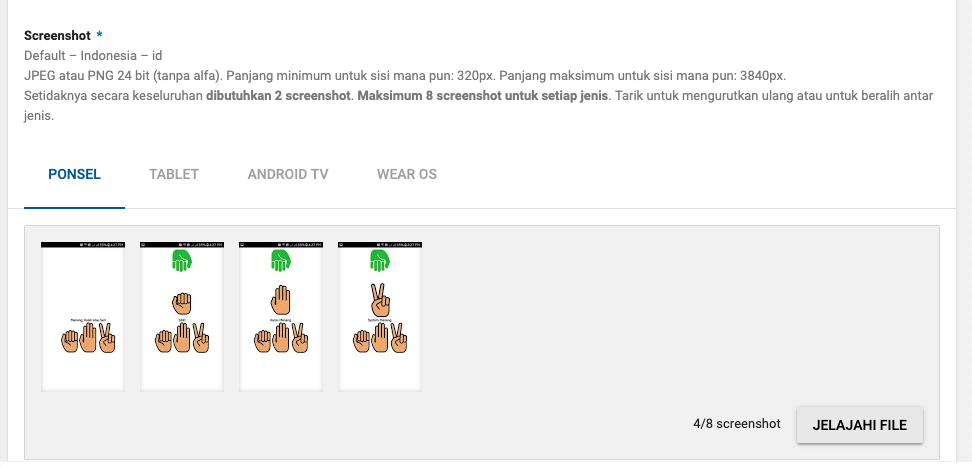
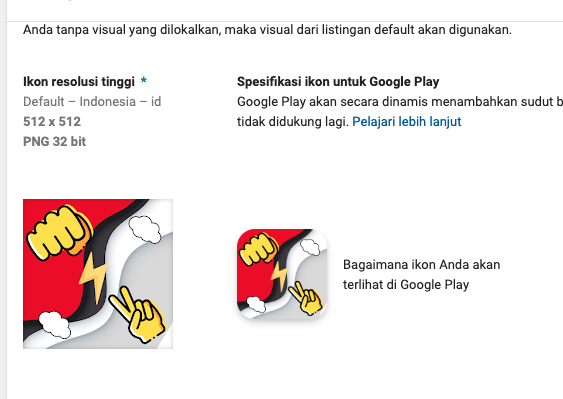
Ada beberapa tahapan dalam mempublish apliksi yang ke play store, diantaranya adalah membuat gambar icon aplikasi, Banner , screenshot tampilan aplikasi dan lainnya.

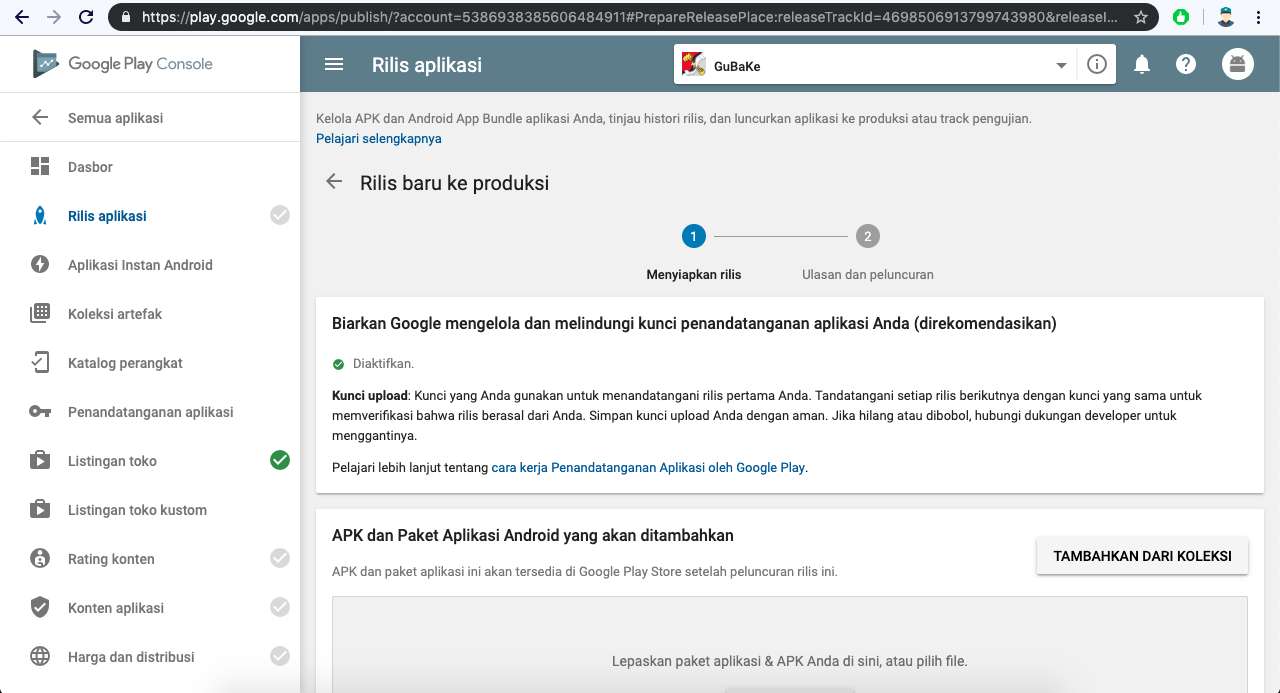
**Tahap – tahap upload**

****

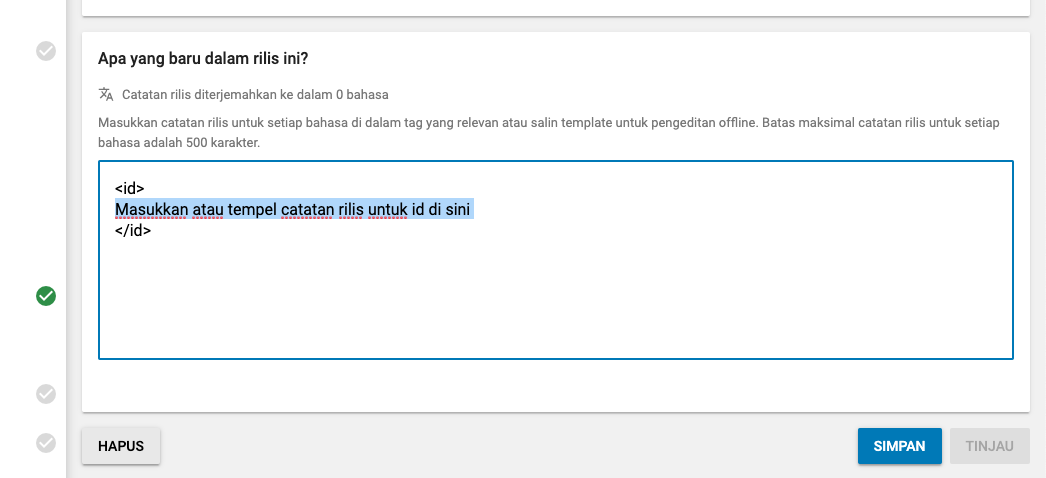
**Memberikan Judul atau nama aplikasi.**

****

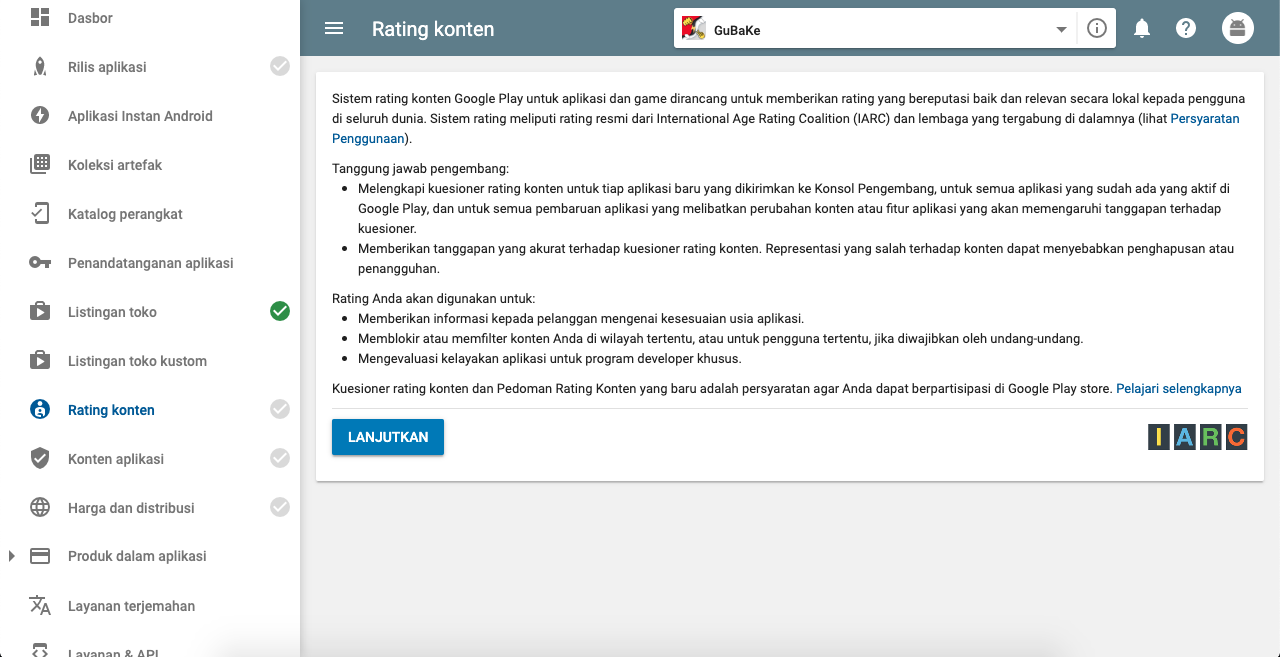
****

****

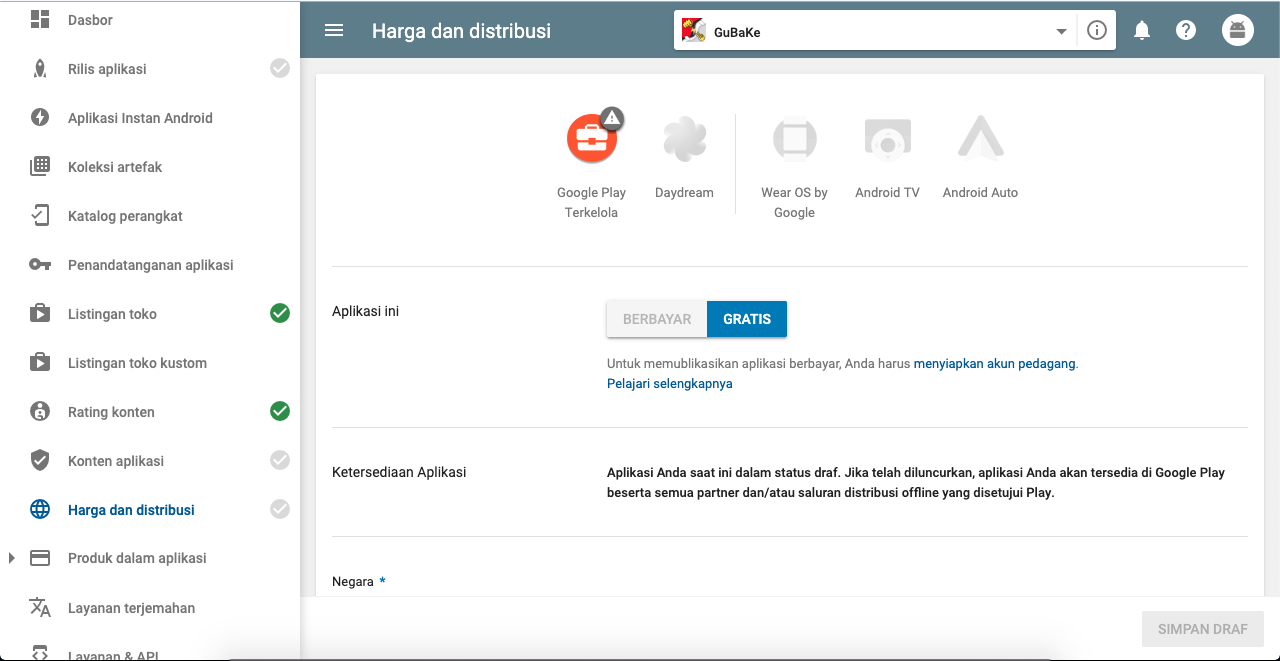
**Upload aplikasi yang sudah di build sebelum**

****

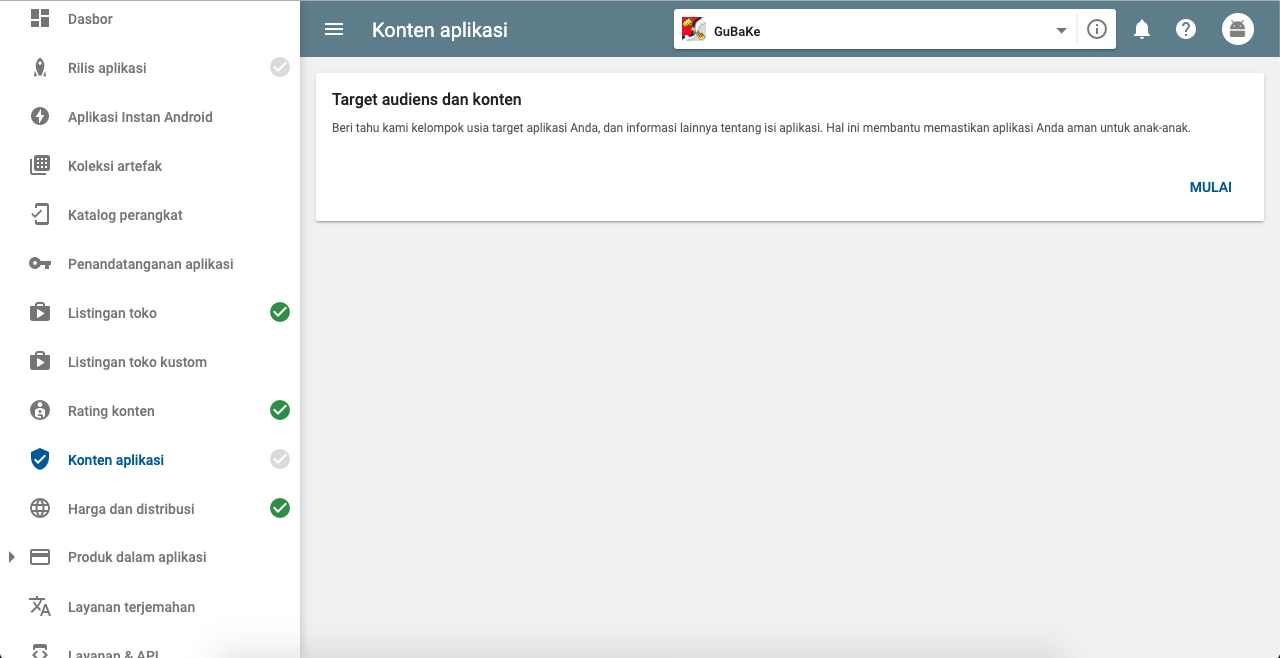
**Menambahkan deskripsi tentang aplikasi yang di buat**

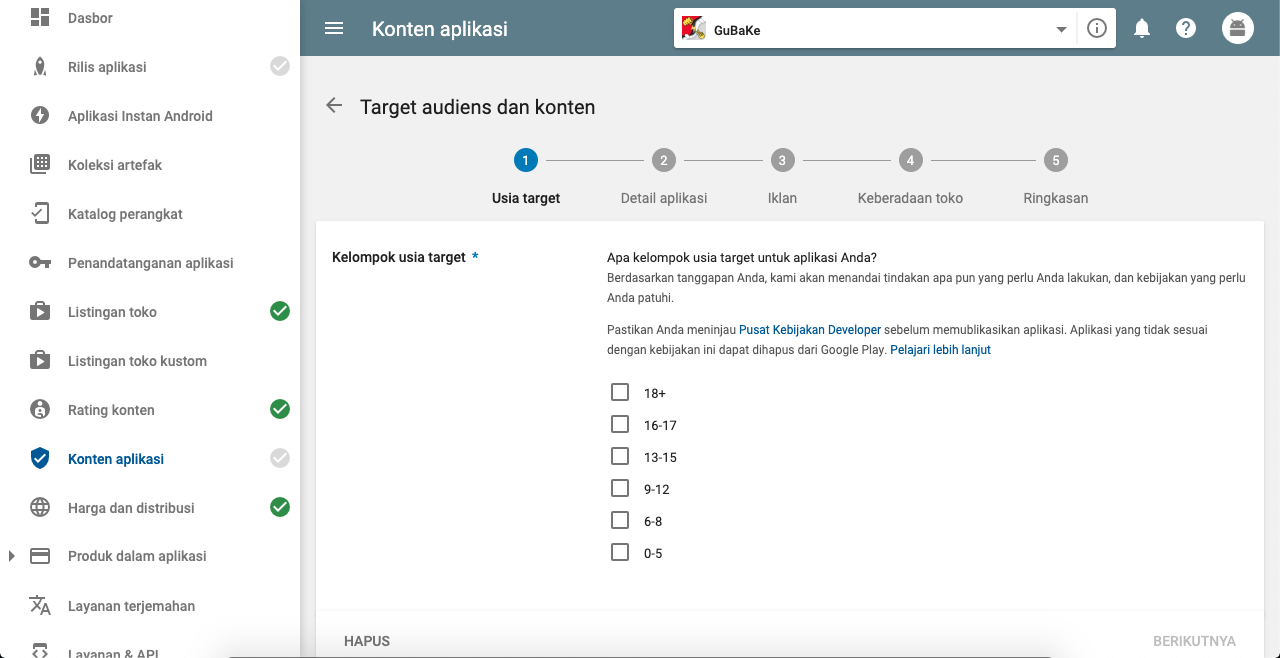
****

**Setelah anda menhupload apk, anda dapat melanjutkan tahap memasang rating pada aplikasi**

****

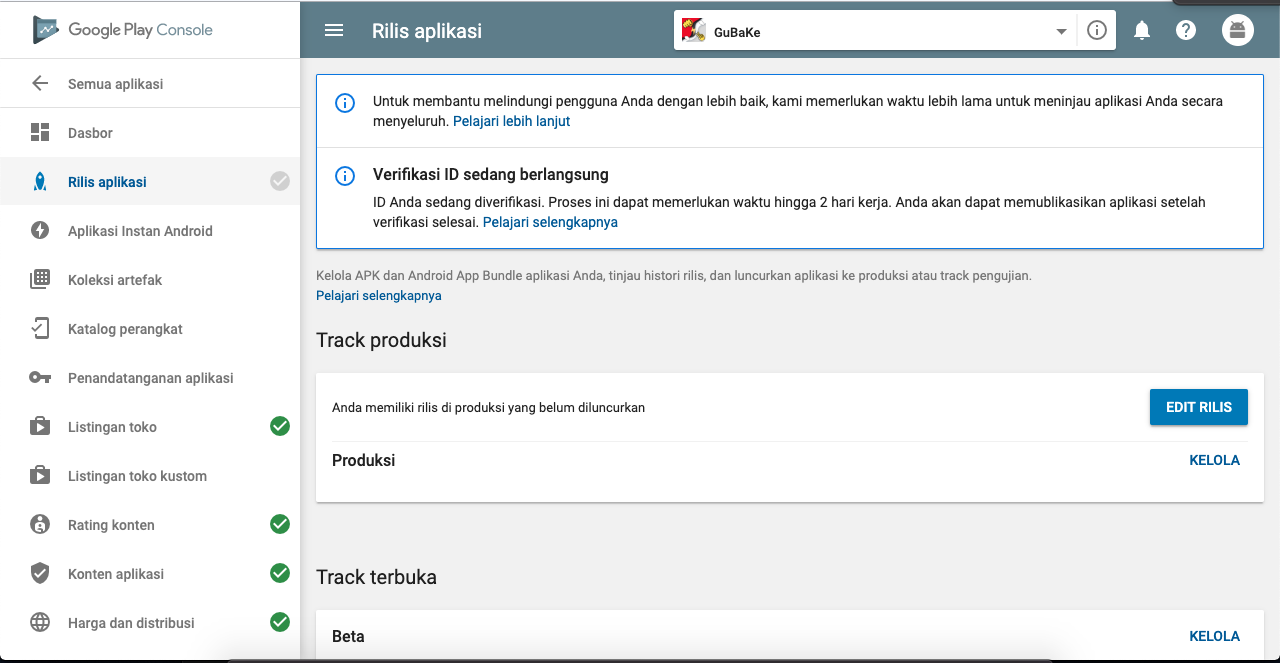
**Menerapkan harga & distribusi**

****

****

**Tahap target audiens atau pengguna**

Setelelah semua tahapan di atas terpenuhi semua maka lanjut kembali ke bagian **Rilis Aplikasi** untuk melakukan tahap **Publish Aplikasi.**



## Kirim aplikasi Anda untuk dipublikasikan

Bila mengunggah aplikasi untuk produksi, Google akan memeriksa aplikasi Anda. Google menjalankan pemeriksaan otomatis dan manual pada aplikasi Anda.

Jika aplikasi ditolak, perbaiki masalah dan coba lagi!

**Penagguhan Aplikasi**

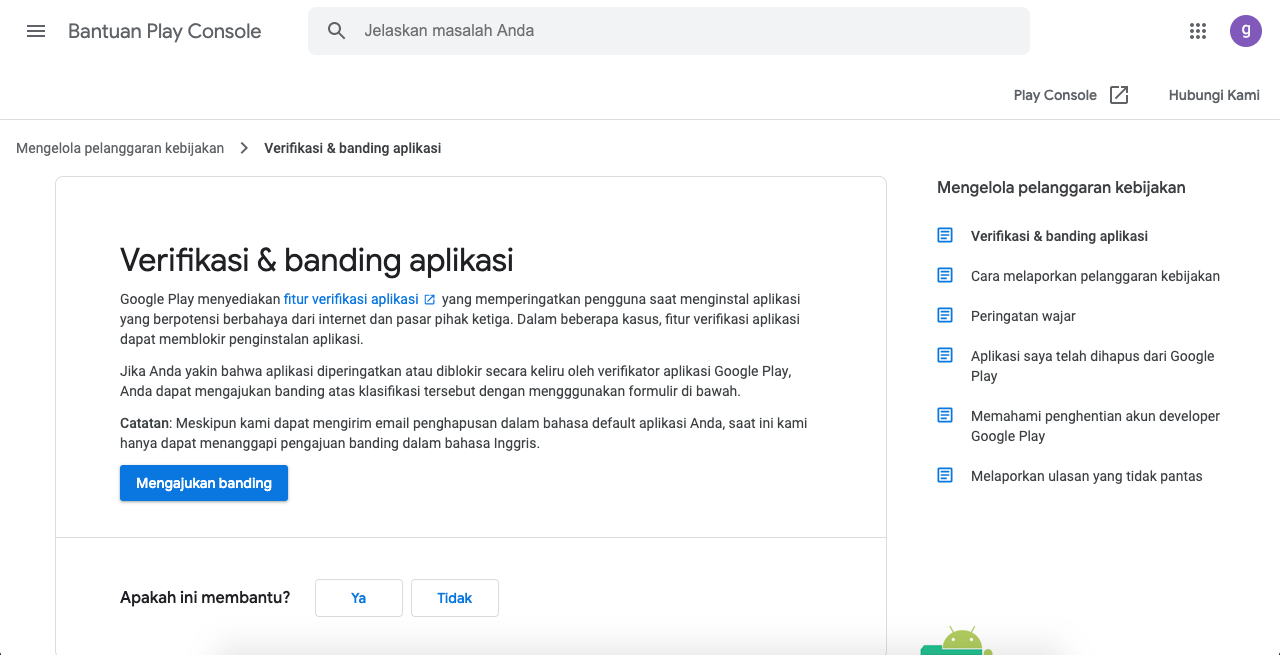
**Faktor – factor aplikasi di tangguhkan**

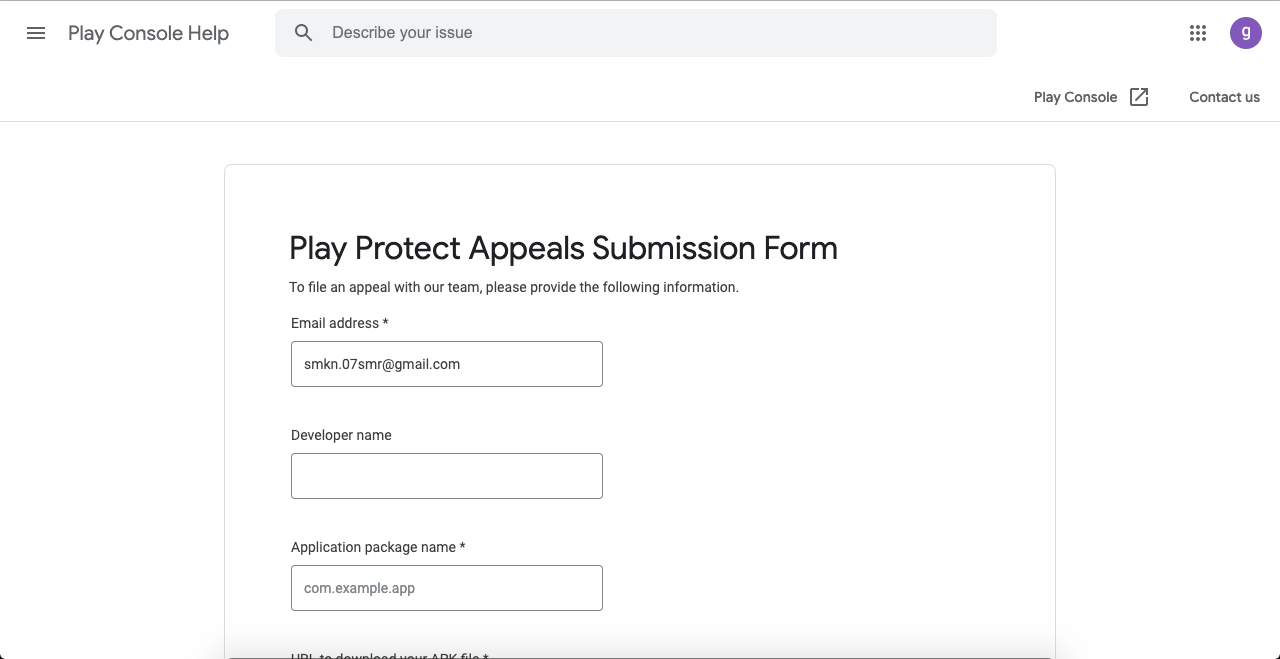
Ada beberapa factor yang menyebabkan aplikasi yang anda upload ke play stor di tangguhkan atau di tolak

* Aplikasi mengandung unsur plagiat seperti nama dan icon
* Menggunakan data instansi pemerintahan
* Menggunakan data pribabadi sebagai pengumpulan data seperti data KTP, SIM, dan Passport dengan akun yang tidak sesuai dengan instansinya.
* Verifikasi data yang kurang lengkap.

Dalam tahap ini kita dapat mengajukan banding terhadap google dengan mengirimkan surat banding kepada google untuk untuk menangguhkan penolakkan upload aplikasi kita dengan menjelaskan tetang aplikasi yang kita buat. Verifikasi banding bias memakan waktu kurang lebih 1 minggu sejak pengajuan banding dikirim.

<https://support.google.com/googleplay/android-developer/answer/2992033?hl=id>



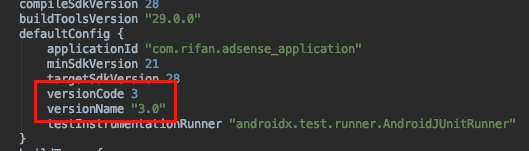




**Pembaruan Aplikasi**

Setiap aplikasi yang dibuat dan telah di upload ke play store pastinya akan ada pembaruan aplikasi, pihak pengembang dapat mengirim pembaruan aplikasi ke playstore dengan mengikuti syarat-syrat yang telah di tetapkan oleh pihak google diantaranya yaitu;

* Menyertakan keystore aplikasi yang di upload pertamakali
* Mengubah version Code pada aplikasi yang akan di upload sebagai pembaruan
* Perubahan version Name pada aplikasi

Setelah anda melakukan perubahan data tersebut maka anda build aplikasi yang akan di upload ke play store.

**Upload Pembaruan Aplikasi**

Untuk tahap upload apk pembaruan sama dengan cara yang telah di jelaskan di atas sebelumnya. Jika ada pembaruan mengenai nama aplikasi screanshoot aplakasi, deskripsi aplikasi dan icon aplikasi anda dapat melakukannya.

## Rangkuman akhir

Anda sudah mencapai bagian akhir kursus ini. Kami berharap Anda menikmati perjalanan dan siap untuk membangun aplikasi Android sendiri. Kami menantikan aplikasi Anda di Google Play Store!

## Ketahui selengkapnya

### Mempersiapkan aplikasi Anda

* Mempersiapkan untuk rilis developer.android.com/studio/publish/preparing.html
* Daftar periksa peluncuran developer.android.com/distribute/tools/launch-checklist.html
* Daftar periksa kualitas aplikasi inti developer.android.com/distribute/essentials/quality/core.html
* Menangani data pengguna play.google.com/about/privacy-security/user-data/
* Filter aplikasi [developer.android.com/google/play/filters.html](https://developer.android.com/google/play/filters.html)
* API level min, maks, dan target [developer.android.com/guide/topics/manifest/uses-sdk-element.html](https://developer.android.com/guide/topics/manifest/uses-sdk-element.html)
* Ragam produk [developer.android.com/studio/build/index.html](https://developer.android.com/studio/build/index.html)
* Tetapkan versi aplikasi Anda [developer.android.com/studio/publish/versioning.html](https://developer.android.com/studio/publish/versioning.html)
* Filter Google Play [developer.android.com/google/play/filters.html](https://developer.android.com/google/play/filters.html)
* Kurangi ukuran aplikasi [developer.android.com/topic/performance/reduce-apk-size.htm](https://developer.android.com/topic/performance/reduce-apk-size.html) Tandatangani aplikasi Anda [developer.android.com/studio/publish/app-signing.html](https://developer.android.com/studio/publish/app-signing.html)

### Google Play Developer Console

* Masuklah ke konsol: [play.google.com/apps/publish/](https://play.google.com/apps/publish/)
* Panduan developer: [developer.android.com/distribute/googleplay/developer-console.html](https://developer.android.com/distribute/googleplay/developer-console.html)
* Pusat bantuan: [support.google.com/googleplay/android-developer/#topic=3450769](https://support.google.com/googleplay/android-developer/#topic%3D3450769)
* Mulai mempublikasikan [developer.android.com/distribute/googleplay/start.html](https://developer.android.com/distribute/googleplay/start.html)
* Firebase Test Lab: [firebase.google.com/docs/test-lab/](https://firebase.google.com/docs/test-lab/)

### Pengujian Alfa dan Beta

* Panduan developer: [developer.android.com/distribute/engage/beta.html](https://developer.android.com/distribute/engage/beta.html)
* Pusat bantuan: [support.google.com/googleplay/android-developer/answer/3131213](https://support.google.com/googleplay/android-developer/answer/3131213)