Build APK

Project Flutter yang telah dibuat dapat kita build menjadi berkas .apk yang dapat berjalan di Android. Build APK adalah suatu proses membungkus aplikasi flutter menjadi format .apk yang nantinya untuk diinstal pada perangkat Android. Berikut tahapan ketika build aplikasi Flutter ke APK.

AndroidManifest.xml

Sebelum mem-build APK, kita akan mengatur berkas android/app/src/main/AndroidManifest.xml. **AndroidManifest.xml** merupakan sebuah berkas yang berisikan informasi mengenai aplikasi Android yang akan di-*build*. Informasi-informasi tersebut berupa nama aplikasi, ikon, *permission*, *screen orientation*, dan lain-lain. Isi dari **AndroidManifest.xml** seperti berikut:

```
1. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2.     package="id.eudeka.samples">
3.     <application
4.         android:name="io.flutter.app.FlutterApplication"
5.         android:label="samples"
6.         android:icon="@mipmap/ic_launcher">
7.         <activity
8.             android:name=".MainActivity"
9.             android:launchMode="singleTop"
10.              android:theme="@style/LaunchTheme"
11.              android:configChanges="orientation|keyboardHidden|keybo
   ard|screenSize|smallestScreenSize|locale|layoutDirection|fontScale|s
   creenLayout|density|uiMode"
12.              android:hardwareAccelerated="true"
13.              android:windowSoftInputMode="adjustResize">
14.              <meta-data
15.                android:name="io.flutter.embedding.android.NormalThem
   e"
16.                android:resource="@style/NormalTheme"
17.                />
18.              <meta-data
19.                android:name="io.flutter.embedding.android.SplashScre
   enDrawable"
20.                android:resource="@drawable/launch_background"
21.                />
22.              <intent-filter>
23.                  <action android:name="android.intent.action.MAIN"/>
24.                  <category android:name="android.intent.category.LAU
   NCHER"/>
25.              </intent-filter>
```

```
        </activity>
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>
</manifest>
```

Setting Nama Aplikasi

Untuk mengatur nama aplikasi, kita cukup mengubah properti android:label yang ada pada file **AndroidManifest.xml** seperti berikut:

```
<application
        android:name="io.flutter.app.FlutterApplication"
        android:label="common_widget"
        android:icon="@mipmap/ic_launcher">
```

menjadi

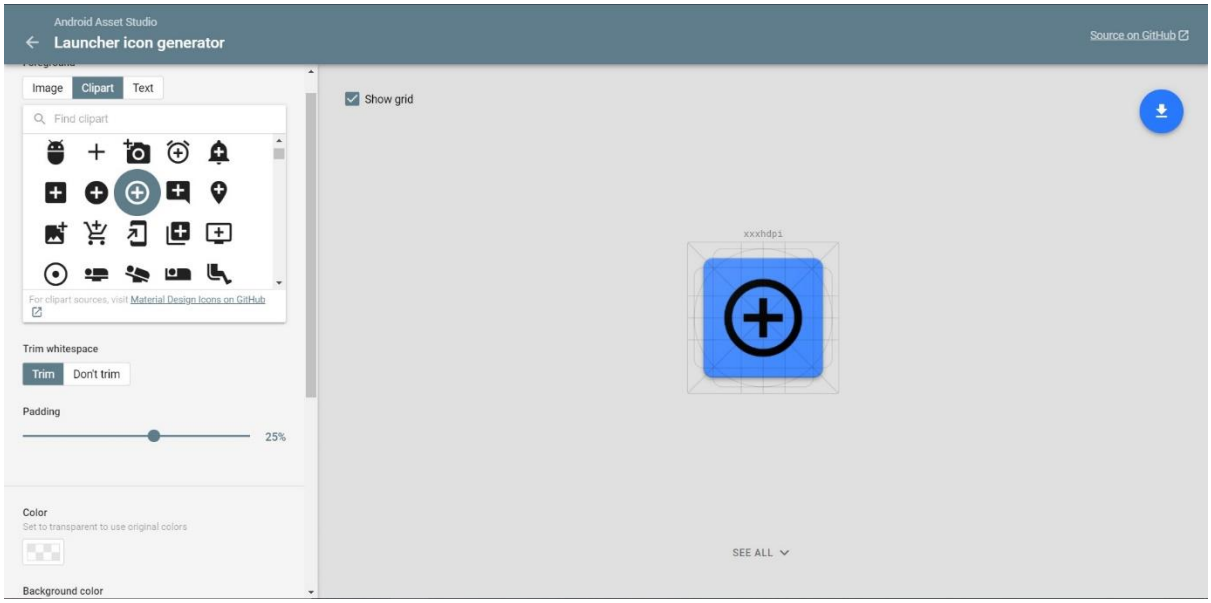
```
<application
        android:name="io.flutter.app.FlutterApplication"
        android:label="Nama Aplikasi"
        android:icon="@mipmap/ic_launcher">
```

Isikan android:label dengan nama aplikasi yang diinginkan. Atau Anda bisa gunakan *library* berikut untuk menghasilkan nama aplikasi dari **pubspec.yaml**.

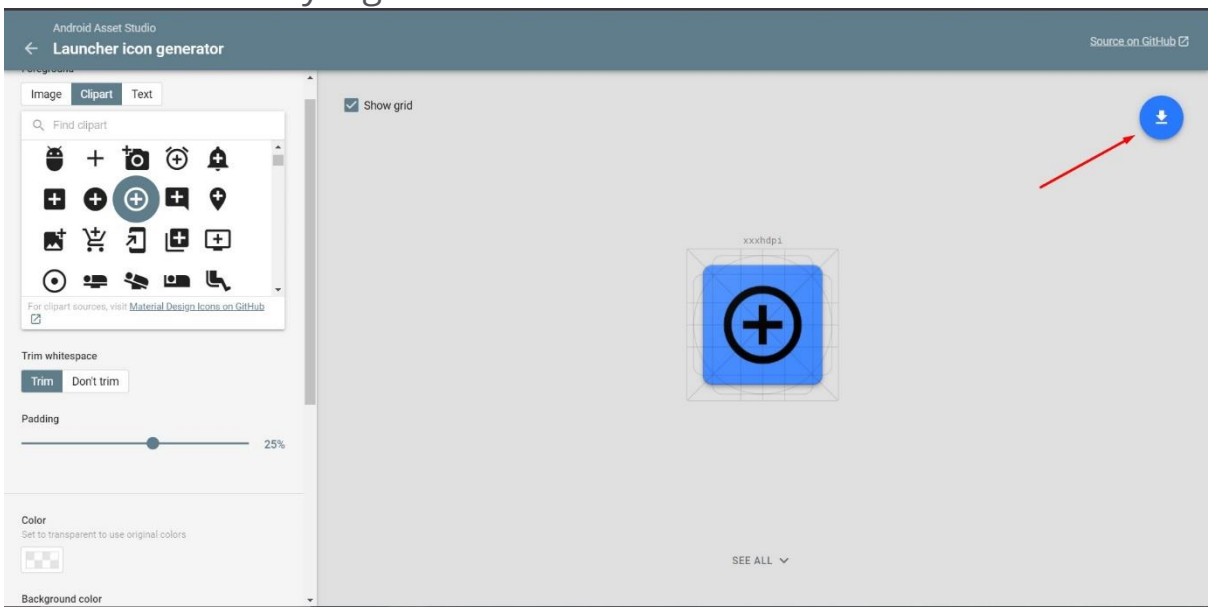
Setting Ikon Aplikasi

Secara *default* ikon aplikasi Flutter kita adalah ikon Flutter. Untuk mengubah icon aplikasi dengan mudah, kita akan mengganti gambar ic_launcher.png yang berada pada folder android/app/src/main/res/ yang terbagi menjadi berbagai *mipmap* (ukuran resolusi ikon).

Hal yang pertama kita lakukan adalah membuat ikon aplikasi dengan Android Asset Studio.



Dengan Android Asset Studio, kita dapat membuat ikon aplikasi dengan mudah dan nantinya akan terbuat dalam berbagai resolusi (*mipmap*). Setelah membuat ikon sesuai dengan keinginan, tekan tombol *download* yang ada di kanan atas.



Setelah mengunduh, *unzip*-lah berkas tersebut dan temukan folder `res/` di dalamnya. Lalu copy folder `res/` ke `android/app/src/main/res/` untuk mengganti `ic_launcher.png` pada setiap *mipmap* dengan ikon aplikasi yang baru. Atau Anda bisa gunakan *library* berikut untuk menghasilkan *icon launcher* dari **pubspec.yaml**.

Setting Perizinan Aplikasi

Ketika aplikasi dalam mode *debug* atau profil, perizinan internet akan secara otomatis ditambahkan. Namun ketika Anda ingin menjalankan atau membuatnya dalam mode rilis, Anda perlu menambahkan semua perizinan yang dibutuhkan pada **AndroidManifest**.

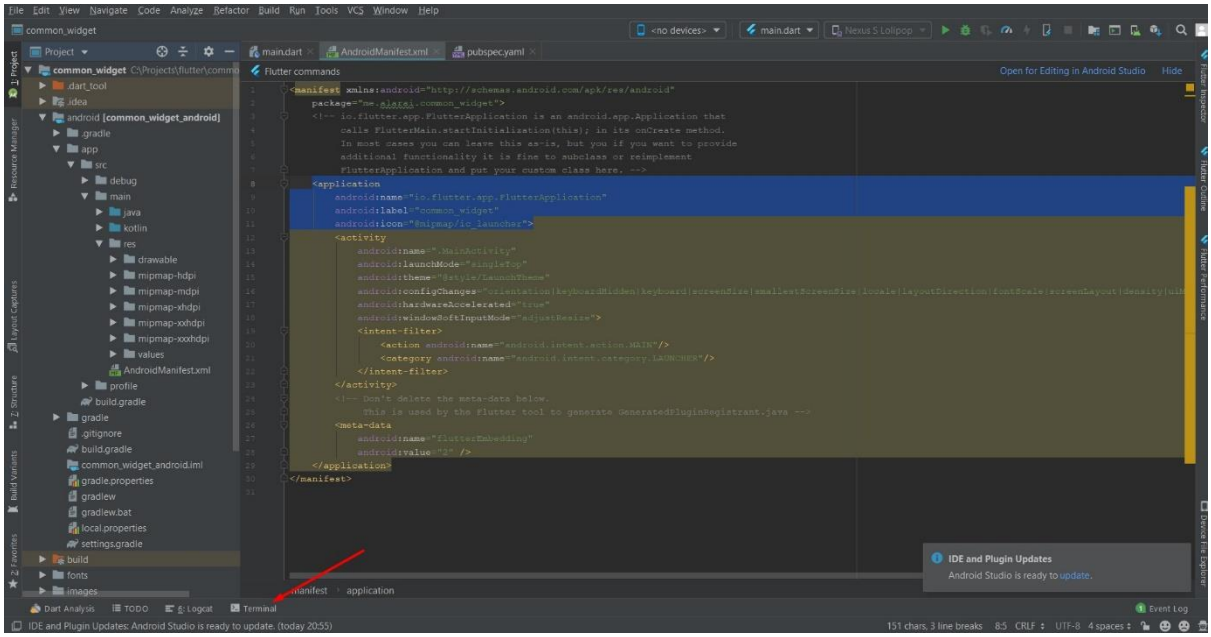
Untuk menambahkan perizinan pada aplikasi Android, Anda bisa menambahkan *tag* **uses-permission** pada **AndroidManifest**, di dalam *tag manifest* dan sejajar *tag application.* Contohnya seperti di bawah ini:

```
1. <uses-permission android:name="android.permission.INTERNET"/>
```

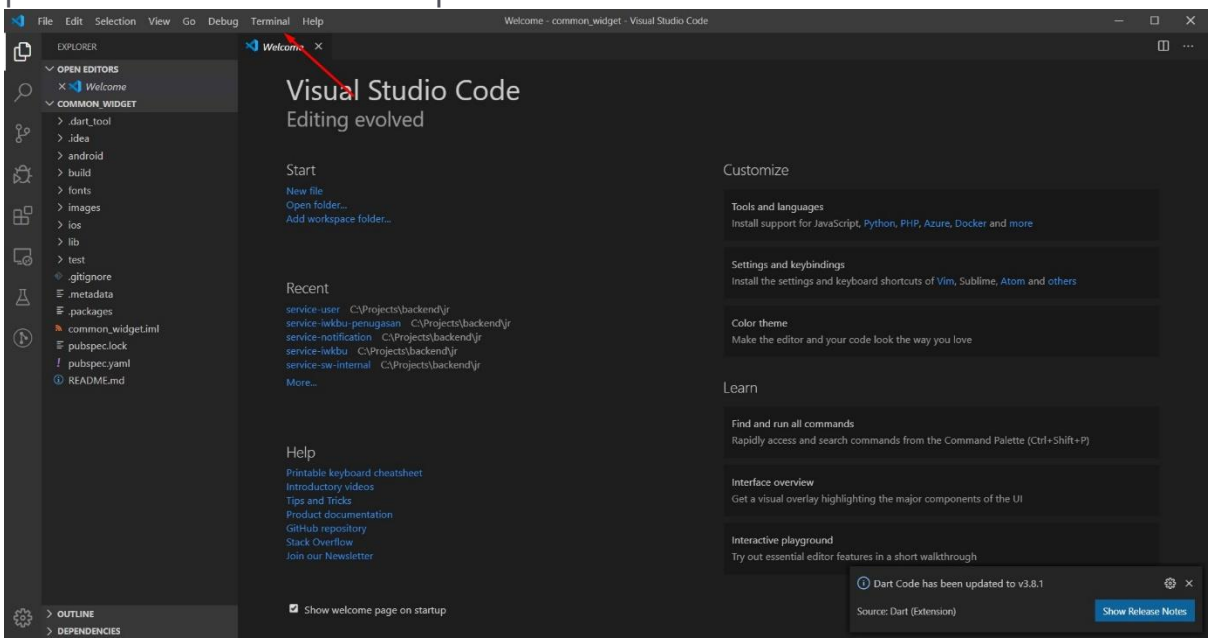
Melakukan Build APK

Setelah kita mengatur nama dan ikon aplikasi, langkah selanjutnya adalah melakukan *build* aplikasi menjadi APK. Sebelumnya terdapat tiga (3) jenis mode aplikasi yang perlu diketahui,
yaitu *debug*, *profile,* dan *release*. APK *debug* umumnya digunakan untuk pengujian dan penggunaan aplikasi secara internal.
Mode *debug* digunakan secara *default* ketika menjalankan aplikasi menggunakan perintah **flutter run**. Sementara untuk bisa dirilis melalui Google Play Store, Anda perlu membuat APK *release*.
Sedangkan mode *profile* sama hal nya dengan *release* hanya saja tetap dapat di-*debug* menggunakan *tools* seperti DevTools dan tidak dapat dijalankan di emulator atau simulator.

Pada kelas ini kita akan mempelajari bagaimana membuat APK *debug*. Caranya ialah menggunakan terminal pada Android Studio. Tekan tombol **Terminal** yang ada pada pojok kiri bawah.



Bila menggunakan Visual Studio Code pilih menu terminal yang ada pada menu kiri atas. Lalu pilih **new terminal**.



Jika terminal telah muncul, tuliskan perintah berikut:

```
1. flutter build apk --debug
```

Tunggu hingga proses *build* berhasil. Setelah berhasil, hasil build yang berupa berkas **apk-debug.apk** akan terletak di folder build/app/outputs/apk/debug/ atau akan muncul direktori tempat tersimpannya berkas ketika proses *build* selesai pada Terminal.

Untuk bisa mem-*build* apk release dan mengunggahnya melalui Google Play Store, Anda memerlukan *signing key*. *Signing key* ini digunakan sebagai tanda tangan supaya aplikasi Anda lebih aman.
Secara *default* Flutter menggunakan *debug key* sebagai *signing key* sehingga Anda sebenarnya bisa membuat apk release dengan menjalankan perintah berikut:

```
1. flutter build apk
```

Namun, tentunya akan lebih baik jika Anda menggunakan signing key milik Anda sendiri. Cara untuk membuat signing key dan membuat apk release dapat Anda baca pada tautan dokumentasi berikut: https://flutter.dev/docs/deployment/android.