

Modul 13

AdMob

Monetizing your Flutter app using AdMob

Module Overview

Mengenal AdMob

Menambahkan AdMob dalam Aplikasi

Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat memanfaatkan AdMob sebagai salah satu cara menghasilkan uang dari aplikasi mobile.

Menghasilkan uang dari aplikasi adalah tujuan utama bagi banyak pengembang aplikasi. Sebagai pengembang aplikasi, sangat penting bagi anda untuk memahami cara memonetisasi aplikasi yang anda buat secara efektif. Setidaknya ada lima cara untuk memonetisasi aplikasi yang anda bangun, yaitu iklan, pembelian aplikasi, pembelian dalam aplikasi, dan berlangganan.

Dalam pertemuan ini, kita akan membahas bagaimana Anda dapat memonetisasi aplikasi Flutter Anda menggunakan AdMob (iklan untuk seluler). AdMob menyediakan cara bagi Google untuk membantu developer



aplikasi menghasilkan lebih banyak uang dari aplikasi melalui iklan. Setiap iklan tersebut dibuat dan dibayar oleh pengiklan yang ingin mempromosikan produk mereka. Untuk mengintegrasikan layanan AdMob di aplikasi Flutter, Anda akan menggunakan plug-in

google_mobile_ads. Plug-in ini memungkinkan Anda untuk menampilkan iklan banner, interstitial (layar penuh), reward, dan native di dalam aplikasi Flutter anda. Anda akan mencoba untuk membuat aplikasi sederhana yang menampilkan berbagai format iklan saat tombol ditekan.

Setting up AdMob

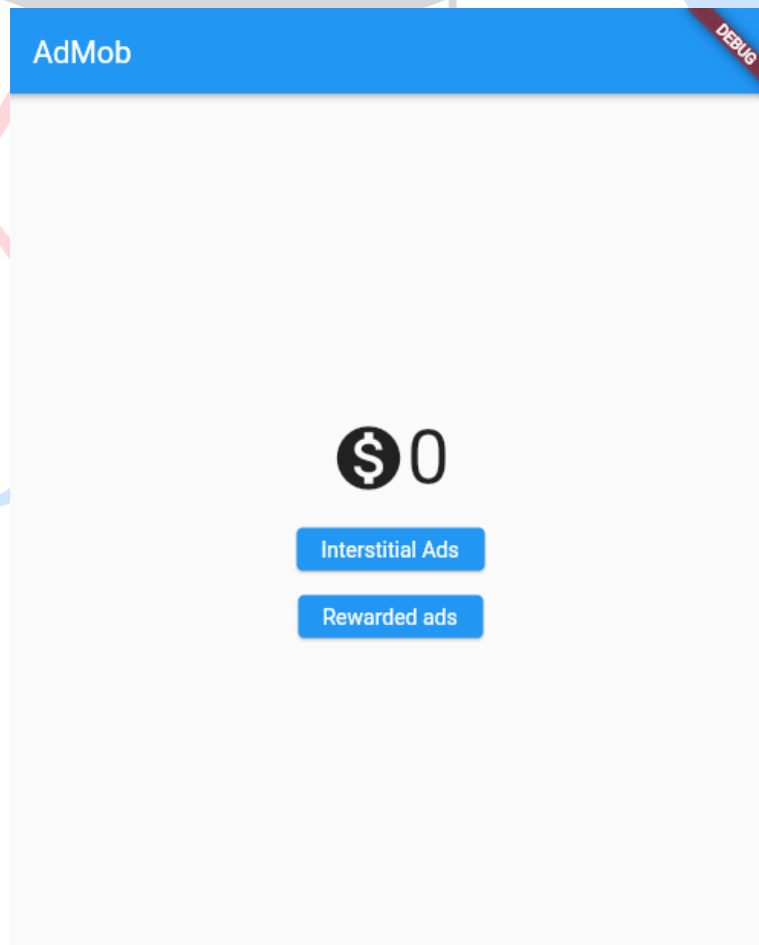
Sebelum Anda mulai mengintegrasikan AdMob di aplikasi Flutter, langkah pertama mengharuskan Anda membuat aplikasi baru di dalam konsol AdMob. Untuk itu, lakukan langkah-langkah berikut ini.

1. rancanglah tampilan sederhana berikut ini.

```
1 import 'package:flutter/material.dart';
2
3 class MainScreen extends StatefulWidget {
4   const MainScreen({Key? key}) : super(key: key);
5
6   @override
7   State<MainScreen> createState() => _MainScreenState();
8 }
9
10 class _MainScreenState extends State<MainScreen> {
11   int coin = 0;
12   @override
13   Widget build(BuildContext context) {
14     return Scaffold(
15       appBar: AppBar(title: const Text("AdMob")),
16       body: Center(
17         child: Column(
18           mainAxisAlignment: MainAxisAlignment.center,
19           children: [
20             Padding(
21               padding: const EdgeInsets.all(8.0),
22               child: Row(
23                 mainAxisAlignment: MainAxisAlignment.center,
24                 children: [
25                   const Icon(
26                     Icons.monetization_on,
27                     size: 50,
28                   ),
29                   Text(
30                     coin.toString(),
31                     style: TextStyle(fontSize: 50),
32                   ),
33                 ],
34               ),
35             ),
```

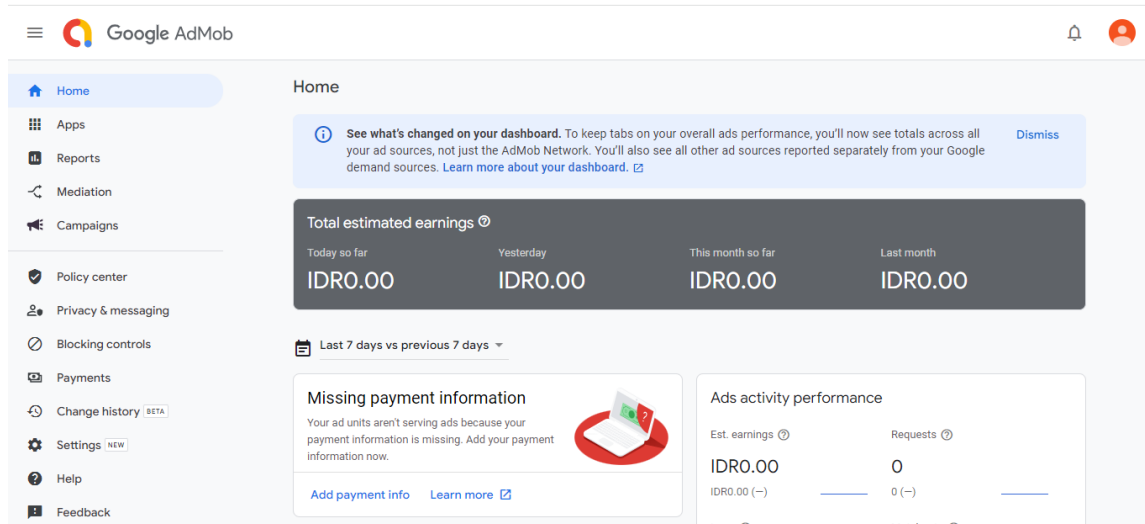
```
36      Padding(  
37        padding: const EdgeInsets.all(8.0),  
38        child: ElevatedButton(  
39          onPressed: () {}, child: Text("Interstitial Ads")),  
40      ),  
41      Padding(  
42        padding: const EdgeInsets.all(8.0),  
43        child:  
44          ElevatedButton(onPressed: () {}, child: Text("Rewarded ads")),  
45      )  
46    ],  
47  )),  
48 );  
49 }  
50 }  
51
```

Jalankan Screen ini pada main.dart anda, sehingga muncul tampilan berikut:

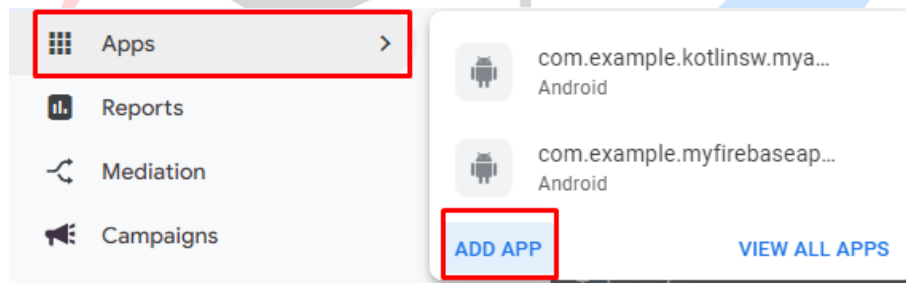


- Daftar dan buka konsol Google admob berikut link berikut

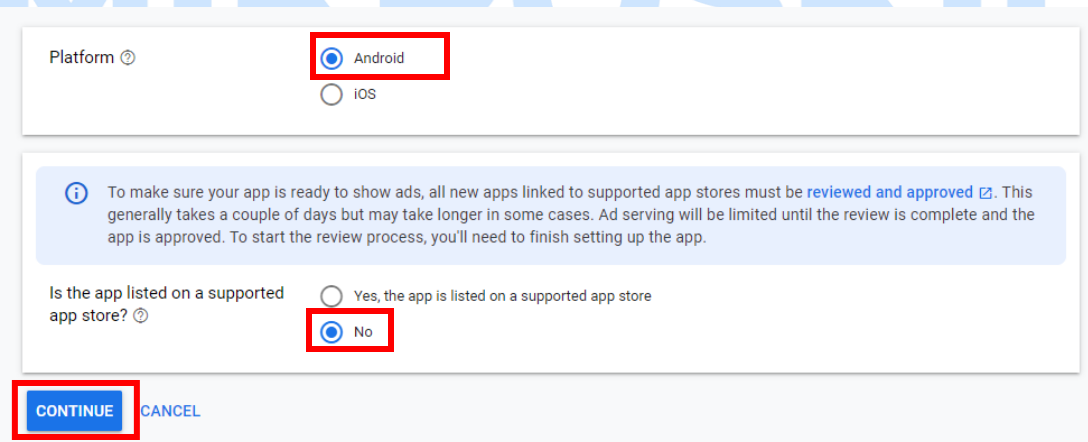
<https://admob.google.com/home/> :



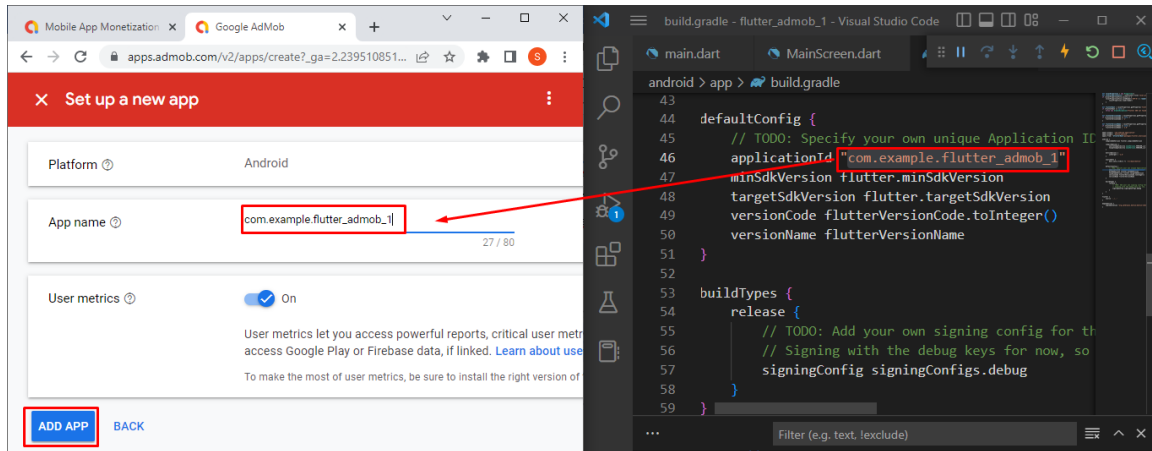
- Daftarkan aplikasi anda pada menu samping **Apps**, dan pilih **Add App**



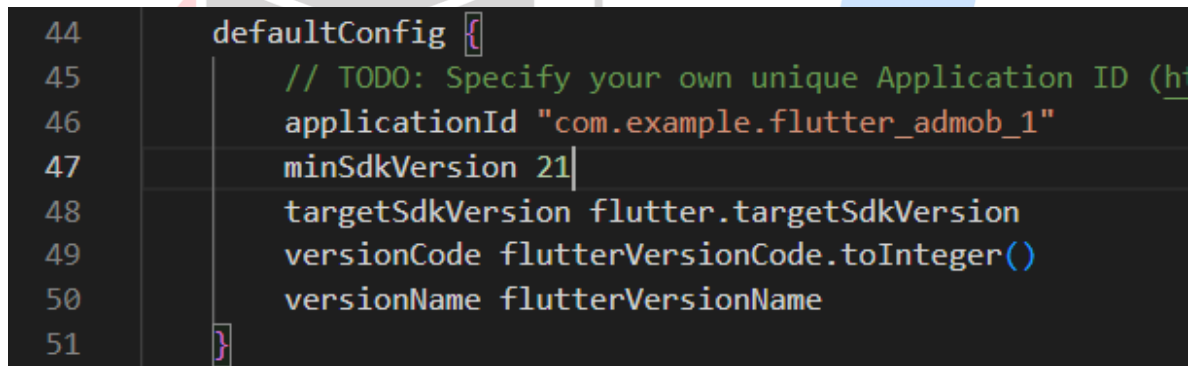
- Pilih Platform aplikasi anda dengan **Android** dan centang **No** pada **Is the app listed on a supported app store?** Yang artinya aplikasi anda belum dipublikasikan pada app store. Dan klik **Continue**



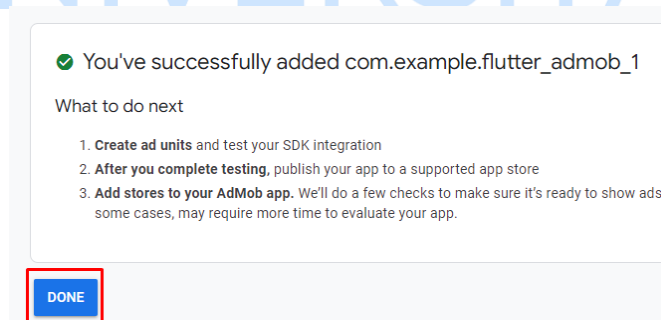
5. Masukkan ID Aplikasi anda (sesuai dengan ID pada **app/build.gradle**), dan klik **ADD APP**



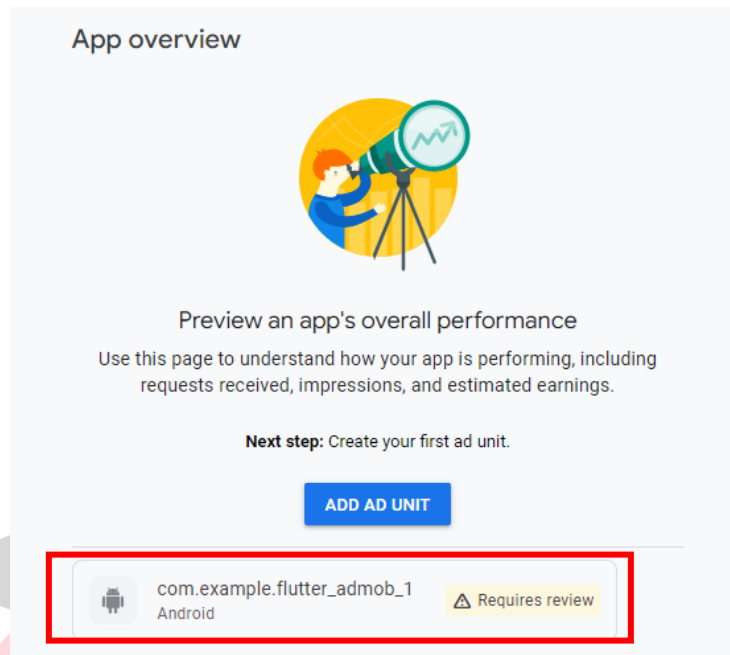
6. Pada **app/build.gradle** ini, pastikan juga **minSdkVeresi** yang digunakan untuk AdMob adalah versi 21



- 7.
8. Dan untuk menyelesaikannya, klik **Done**



9. Setelah itu, aplikasi anda akan muncul pada App Overview pada halaman AdMob.



10. Saat ini, anda masih dalam tahap developer, untuk itu, jangan tambahkan Ad Unit. Anda akan menggunakan Ad testing terlebih dahulu dan memastikan aplikasi anda telah berjalan dengan baik dan dapat dipasarkan sebelum menggunakan Ad Unit. Penggunaan Ad Unit pada proses developer dapat membuat akun anda di **suspend**. Adapun Ad Testing adalah sebagai berikut:

Android app ID/ad unit ID

Item	app ID/ad unit ID
AdMob app ID	ca-app-pub-3940256099942544~3347511713
Banner	ca-app-pub-3940256099942544/6300978111
Interstitial	ca-app-pub-3940256099942544/1033173712
Rewarded	ca-app-pub-3940256099942544/5224354917

11. Masukkan **AdMob app ID** pada file **manifest.xml** pada folder **android\app\src\main** anda, id lainnya akan kita gunakan pada pembahasan **Banner, Interstitial, dan Rewarded**

```

1  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2      package="com.example.flutter_admob_1">
3      <application
4          android:label="flutter_admob_1"
5          android:name="${applicationName}"
6          android:icon="@mipmap/ic_launcher">
7  >
27 </activity>
28 <!-- Don't delete the meta-data below.
30 <meta-data ...
33 <meta-data
34     android:name="com.google.android.gms.ads.APPLICATION_ID"
35     android:value="ca-app-pub-3940256099942544~3347511713"/>
36 </application>
37 </manifest>

```

12. Langkah berikutnya adalah menambahkan dependence **google_mobile_ads** pada **pubspec.yaml** anda.

```

29 dependencies:
30   cupertino_icons: ^1.0.2
31   flutter:
32     sdk: flutter
33   google_mobile_ads: ^2.2.0

```

13. Sekarang, saatnya untuk menginisialisasi plug-in. Buka **main.dart** anda dan tambahkan baris berikut sebelum **runApp** seperti ini:

```

3  import 'package:google_mobile_ads/google_mobile_ads.dart';
4
5  void main() {
6    WidgetsFlutterBinding.ensureInitialized();
7    MobileAds.instance.initialize();
8    runApp(const MyApp());
9  }

```

Pada tahap ini, anda telah berhasil mengatur persiapan dalam menampilkan iklan dari AdMob. Berikutnya kita akan mencoba untuk menampilkan iklan banner, interstitial (layar penuh) dan iklan reward.

Banner

Iklan tipe *banner* merupakan iklan yang akan berada di bagian atas atau bawah layar perangkat. *Banner* tetap akan berada di layar saat pengguna berinteraksi dengan aplikasi,

dan dapat menampilkan iklan baru secara otomatis setelah jangka waktu tertentu. Untuk menggunakan Banner pada aplikasi yang telah kita buat, ikut langkah-langkah berikut:

1. Deklarasikan objek **_bannerAd** sebagai class **BannerAd** dari **google_mobile_ads.dart** dan tambahkan sebuah variabel untuk menentukan apakah banner sudah siap untuk digunakan atau belum yaitu variabel **_isBannerReady**.

```
11 class _MainScreenState extends State<MainScreen> {
12   int coin = 0;
13   late BannerAd _bannerAd;
14   bool _isBannerReady = false;
```

2. Tambahkan fungsi **initState()** untuk memuat banner melalui fungsi/metode **_loadBannerAd()**

```
16 @override
17 void initState() {
18   super.initState();
19   _loadBannerAd();
20 }
21
22 @override
23 Widget build(BuildContext context) {
24   return Scaffold;
```

Quick Fix...

- Create class '_loadBannerAd'
- Create method '_loadBannerAd'**
- Create function '_loadBannerAd'

3. Pada metode **_loadBannerAd()**, yang perlu anda lakukan adalah menginisialisasi **_bannerAd**. Inisialisasi **_bannerAd**, akan menggunakan konstruktor yang terdiri dari 4 parameter, yaitu,

- a. **size** : Menentukan ukuran banner, gunakan **AdSize.banner**,
- b. **adUnitId** : nilai unik yang digunakan untuk menandakan sebuah iklan yang anda daftarkan pada adMob. Pada contoh ini, gunakan ad Unit ID **"ca-app-pub-3940256099942544/6300978111"**, yang merupakan ad unit id khusus Banner untuk proses developer.
- c. **Listener** : digunakan untuk mendengarkan event yang diberikan oleh adMob. Anda dapat menggunakan **BannerAdListener** sebagai event listener anda, yang berisikan

<code>void onAdClicked()</code>	Dijalankan ketika iklan di klik
<code>void onAdClosed()</code>	Dijalankan ketika user kembali/keluar dari iklan setelah di klik

<code>void onAdFailedToLoad (LoadAdError adError)</code>	Dijalankan ketika permintaan Ad gagal
<code>void onAdImpression()</code>	Dijalankan untuk tayangan iklan.
<code>void onAdLoaded()</code>	Dijalankan ketika iklan berhasil didapatkan
<code>void onAdOpened()</code>	Dijalankan saat iklan terbuka dan menutupi layar.

- d. `adRequest` : berisi informasi penargetan (filter iklan) yang digunakan untuk mengambil iklan

untuk itu, isikanlah Inisialisasi `bannerAd` dengan:

```

72 void _loadBannerAd() {
73     _bannerAd = BannerAd(
74         size: AdSize.banner,
75         adUnitId: "ca-app-pub-3940256099942544/6300978111",
76         listener: BannerAdListener(onAdLoaded: (_) {
77             setState(() {
78                 _isBannerReady = true;
79             });
80         }, onAdFailedToLoad: (ad, err) {
81             _isBannerReady = false;
82             ad.dispose();
83         }), // BannerAdListener
84         request: AdRequest()); // BannerAd
85     _bannerAd.load();
86 }

```

4. `_bannerAd.load()` harus digunakan untuk memulai proses permintaan iklan ke **AdMob**.
5. Untuk menampilkan hasil banner pada tampilan widget anda, maka anda harus menggunakan `AdWidget`, dengan parameter `ad`: yang merupakan objek dari **BannerAd**. Untuk menggunakannya, tambahkan widget berikut pada **children terakhir widget column** anda.

```

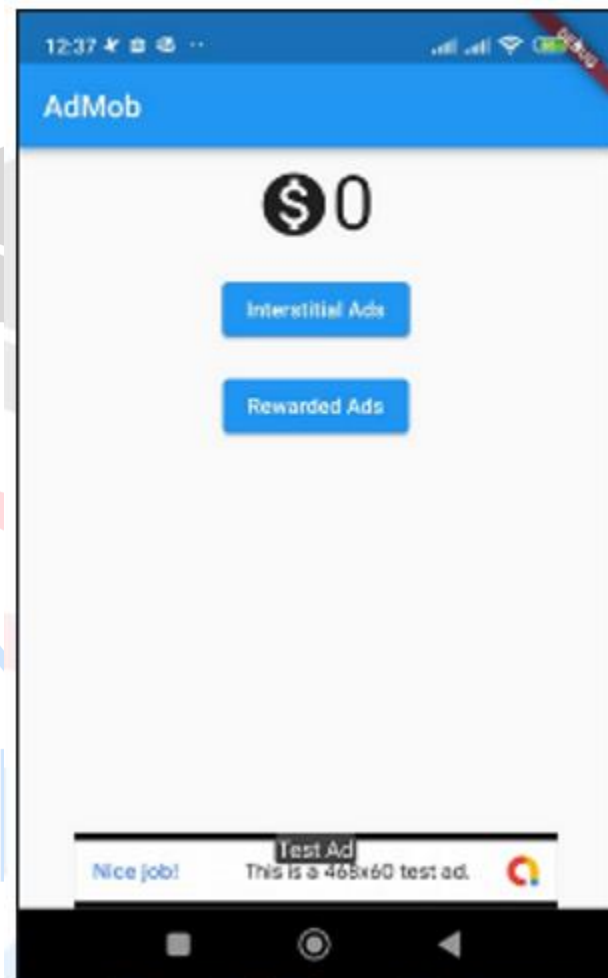
Expanded(
  child: _isBannerReady
    ? Align(
        alignment: Alignment.bottomCenter,
        child: Container(
            width: _bannerAd.size.width.toDouble(),
            height: _bannerAd.size.height.toDouble(),
            child: AdWidget(ad: _bannerAd),

```

```
), // Container  
) // Align  
: Container()), // Expanded
```

6. **_bannerAd.size** digunakan untuk memastikan ukuran penampilan iklan sesuai dengan ukuran iklan banner yang diberikan.

Jalankan aplikasi, maka anda dapat melihat hasil aplikasi anda saat ini adalah sebagai berikut:



Interstitial

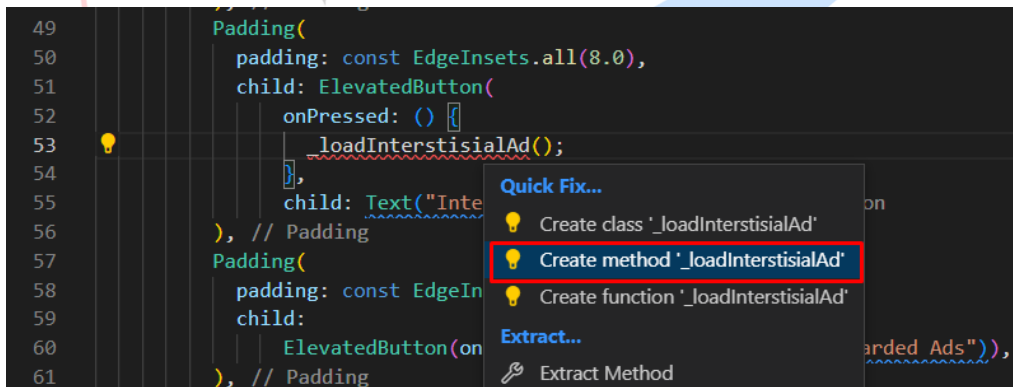
Iklan interstitial merupakan iklan yang ditampilkan pada satu halaman penuh di aplikasi. Ketika menggunakan iklan ini, anda harus memastikan penempatan iklan dilakukan pada

jeda & transisi di antarmuka aplikasi Anda, misalnya setelah menyelesaikan level dalam aplikasi game atau ketika anda membuka aktivitas baru di dalam aplikasi anda. Untuk menggunakan interstitial pada aplikasi yang telah kita buat, ikut langkah-langkah berikut:

1. Deklarasikan objek `_interstitialAd` sebagai class **InterstitialAd** dari **google_mobile_ads.dart** dan tambahkan sebuah Variable untuk menentukan apakah InterstitialAd sudah siap untuk digunakan atau belum yaitu variabel `_isInterstitialReady`.

```
11 class _MainScreenState extends State<MainScreen> {
12   int coin = 0;
13   late BannerAd _bannerAd;
14   bool _isBannerReady = false;
15
16   late InterstitialAd _interstitialAd;
17   bool _isInterstitialReady = false;
```

2. Tambahkan fungsi fungsi/metode `_loadInterstitialAd()` pada properti `onPressed` ElevatedButton "Interstitial Ads"



```
49   Padding(
50     padding: const EdgeInsets.all(8.0),
51     child: ElevatedButton(
52       onPressed: () {
53         _loadInterstitialAd();
54       },
55       child: Text("Interstitial Ads"),
56     ), // Padding
57   Padding(
58     padding: const EdgeInsets.all(8.0),
59     child: ElevatedButton(
60       onPressed: () {
61         _loadInterstitialAd();
62       },
63       child: Text("Interstitial Ads"),
64     ), // Padding
```

3. Pada metode `_loadInterstitialAd()`, yang perlu anda lakukan adalah menginisialisasi **InterstitialAd**. Inisialisasi `InterstitialAd` sedikit berbeda dengan banner, yaitu `InterstitialAd` akan diinisialisasi **adLoadCallback** dari fungsi `InterstitialAd.load()`. Fungsi `InterstitialAd.load()` akan memerlukan 3 parameter, yaitu,
 - a. `adUnitId` : nilai unik yang digunakan untuk menandakan sebuah iklan yang anda daftarkan pada adMob. Pada contoh ini, gunakan ad Unit ID " **ca-app-pub-3940256099942544/1033173712**", yang merupakan ad unit id Interstitial khusus untuk proses developer.

- b. `adRequest` : berisi informasi penargetan (filter iklan) yang digunakan untuk mengambil iklan
- c. `adLoadCallback` : yang berisikan `InterstitialAdLoadCallback()`, yaitu umpan balik yang dipanggil saat iklan Interstitial selesai dimuat.

untuk itu, isikanlah Inisialisasi `_InterstitialAd` dengan cara:

```

97 void _loadInterstitialAd() {
98     InterstitialAd.load(
99         adUnitId: "ca-app-pub-3940256099942544/1033173712",
100         request: AdRequest(),
101         adLoadCallback: InterstitialAdLoadCallback(
102             onAdLoaded: (ad) {
103                 ad.fullScreenContentCallback = FullScreenContentCallback(
104                     onAdDismissedFullScreenContent: (ad) {
105                         print("Close Interstitial Add");
106                     },
107                 ); // FullScreenContentCallback
108                 setState(() {
109                     _isInterstitialReady = true;
110                     _interstitialAd = ad;
111                 });
112             },
113             onAdFailedToLoad: (err) {
114                 _isInterstitialReady = false;
115                 _interstitialAd.dispose();
116             },
117         ), // InterstitialAdLoadCallback
118     );
119 }

```

4. fungsi

```

ad.fullScreenContentCallback = FullScreenContentCallback(
    onAdDismissedFullScreenContent: (ad) {
        print("Close Interstitial Add");
    },
);

```

digunakan untuk menjalankan sebuah perintah ketika iklan ditutup oleh pengguna. Biasanya digunakan untuk melanjutkan navigasi ke layar berikutnya.

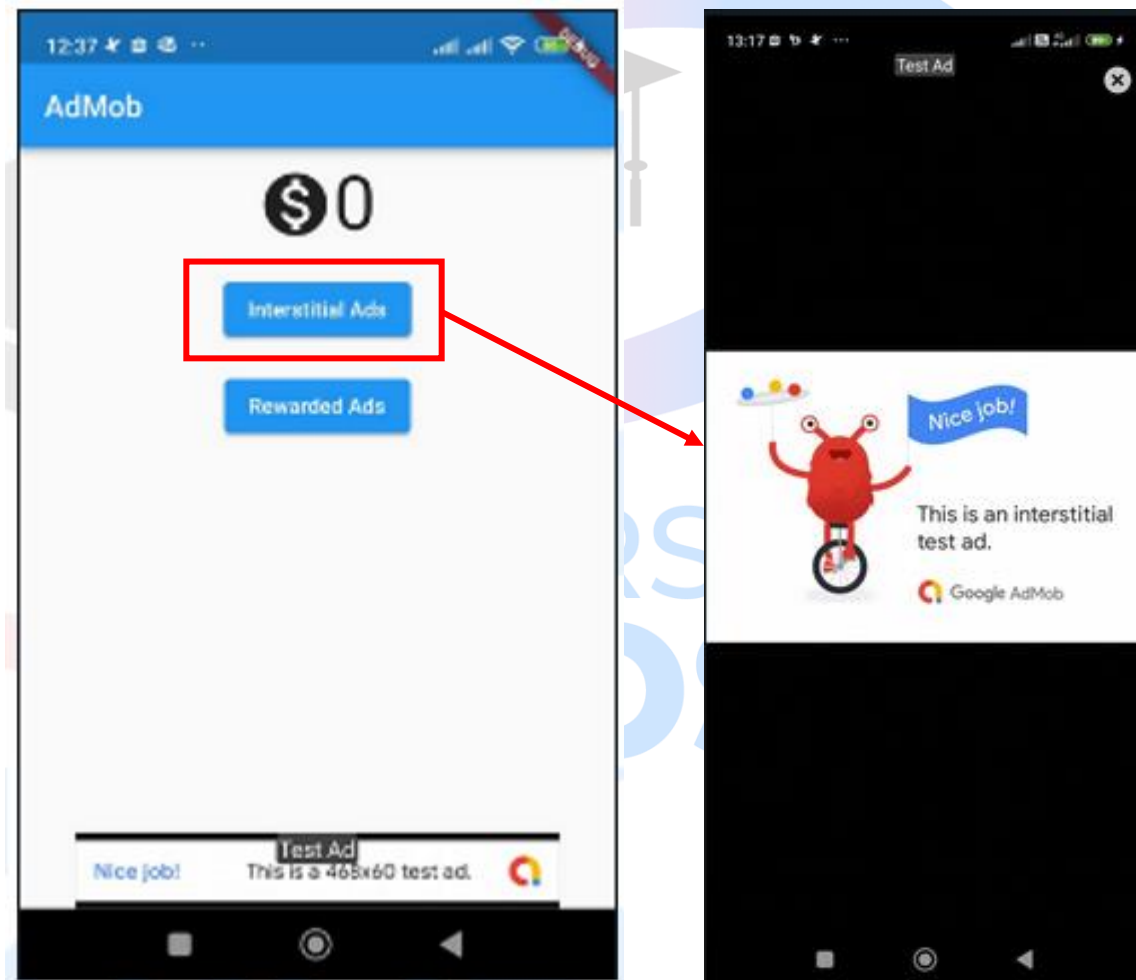
5. Untuk menampilkan `InterstitialAd`, gunakanlah fungsi **`show()`** seperti berikut:

```

49      Padding(
50        padding: const EdgeInsets.all(8.0),
51        child: ElevatedButton(
52          onPressed: () {
53            _loadInterstitialAd();
54            if (_isInterstitialReady) {
55              _interstitialAd.show();
56            }
57          },
58          child: Text("Interstitial Ads"), // ElevatedButton
59        ), // Padding

```

Jalankan aplikasi, maka anda dapat melihat hasil aplikasi anda saat ini ketika menekan tombol **InterstitialAds** akan menampilkan tampilan sebagai berikut:



Reward Ads

Iklan reward merupakan iklan yang banyak digunakan ketika pengguna bermain game atau mengikuti survei, dan menonton iklan video untuk mendapatkan reward dalam aplikasi,

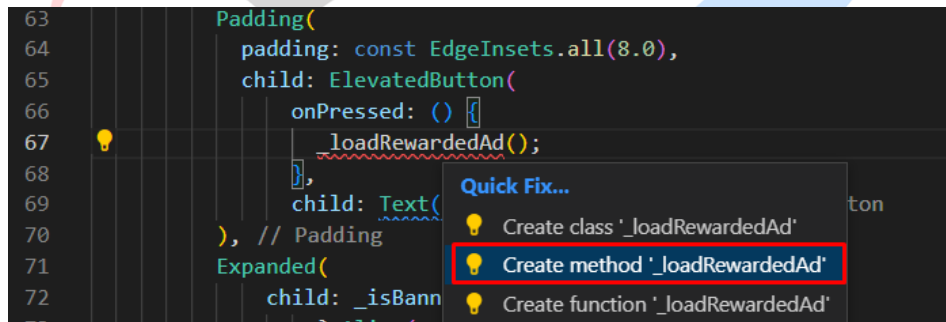
seperti koin, nyawa ekstra, atau extra poin lainnya. Iklan reward akan menampilkan iklan video dan iklan interaktif dengan durasi maksimum untuk memberikan hadiah adalah 30 detik. Jika anda menutup iklan sebelum iklan habis, maka anda dipastikan tidak akan mendapatkan reward.

Untuk menggunakan interstitial pada aplikasi yang telah kita buat, ikut langkah-langkah berikut:

1. Deklarasikan objek `_rewardedAd` sebagai class **RewardedAd** dari **google_mobile_ads.dart** dan tambahkan sebuah variabel untuk menentukan apakah RewardedAd sudah siap untuk digunakan atau belum yaitu variabel `_isRewardedReady`.

```
19 late RewardedAd _rewardedAd;
20 bool _isRewardedReady = false;
```

2. Tambahkan fungsi fungsi/metode `_loadRewardedAd()` pada properti **onPressed** ElevatedButton "Rewarded Ads"



```
63 Padding(
64   padding: const EdgeInsets.all(8.0),
65   child: ElevatedButton(
66     onPressed: () {
67       _loadRewardedAd();
68     },
69     child: Text(
70       ), // Padding
71   Expanded(
72     child: _isBann
73     ),
```

Quick Fix...

- Create class '_loadRewardedAd'
- Create method '_loadRewardedAd'**
- Create function '_loadRewardedAd'

3. Pada metode `_loadRewardedAd()`, yang perlu anda lakukan adalah menginisialisasi `_rewardedAd`. Inisialisasi `_rewardedAd` mirip dengan `_InterstitialAd`, yaitu akan diinisialisasi **rewardedAdLoadCallback** dari fungsi **RewardedAd.load()**. Fungsi **RewardedAd.load()** akan memerlukan 3 parameter, yaitu,

- a. **adUnitId** : nilai unik yang digunakan untuk menandakan sebuah iklan yang anda daftarkan pada adMob. Pada contoh ini, gunakan ad Unit ID **"ca-app-pub-3940256099942544/5224354917"**, yang merupakan ad unit id khusus RewardedAd untuk proses developer.
- b. **adRequest** : berisi informasi penargetan (filter iklan) yang digunakan untuk mengambil iklan

- c. rewardedAdLoadCallback: yang berisikan **RewardedAdLoadCallback()**, yaitu umpan balik yang dipanggil saat iklan Rewarded selesai dimuat.

untuk itu, isikanlah Inisialisasi `_rewardedAd` dengan cara:

```

136 void _loadRewardedAd() {
137     RewardedAd.load(
138         adUnitId: "ca-app-pub-3940256099942544/5224354917",
139         request: AdRequest(),
140         rewardedAdLoadCallback: RewardedAdLoadCallback(),
141         onAdLoaded: (ad) {
142             ad.fullScreenContentCallback = FullScreenContentCallback(
143                 onAdDismissedFullScreenContent: (ad) {
144                     setState(() {
145                         ad.dispose();
146                         _isRewardedReady = false;
147                     });
148                     _loadRewardedAd();
149                 },
150             ); // FullScreenContentCallback
151
152             setState(() {
153                 _isRewardedReady = true;
154                 _rewardedAd = ad;
155             });
156         },
157         onAdFailedToLoad: (err) {
158             _isRewardedReady = false;
159             _rewardedAd.dispose();
160         },
161     ), // RewardedAdLoadCallback
162 };
163 
```

4. Untuk menampilkan RewardedAd dan memberikan reward yang diterima pengguna, gunakanlah fungsi **show()** seperti berikut:

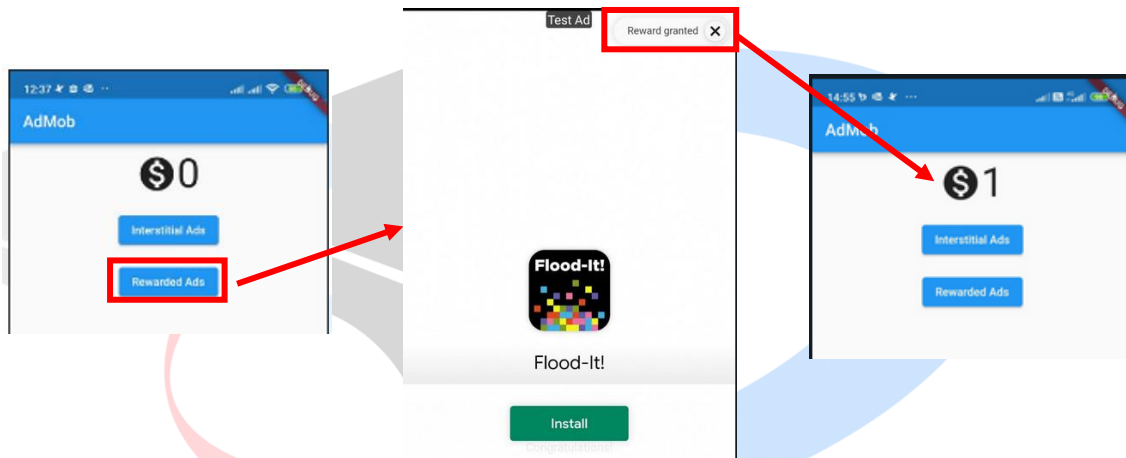
```

64 Padding(
65     padding: const EdgeInsets.all(8.0),
66     child: ElevatedButton(
67         onPressed: () {
68             _loadRewardedAd();
69             if (_isRewardedReady) {
70                 rewardedAd.show(onUserEarnedReward:
71                     (AdWithoutView ad, RewardItem reward) {
72                         setState(() {
73                             coin += 1;
74                         });
75                     });
76             }
77         },
78     ),
79 
```

```
78      child: Text("Rewarded Ads"), // ElevatedButton
79    ), // Padding
```

5. Event **OnUserEarnedReward** merupakan event yang paling penting ketika menggunakan **show()**. Properti ini memastikan pengguna diberi hadiah karena telah selesai menonton video.

Jalankan aplikasi, maka anda dapat melihat hasil aplikasi anda saat ini ketika menekan tombol **InterstitialAds** akan menampilkan tampilan sebagai berikut:



Latihan

1. Ketika membuat iklan, anda juga harus memastikan sebuah fitur pembelian yang dapat menutup iklan. Coba lakukan pembuatan iklan, yang dapat dihilangkan ketika sebuah opsi diberikan kepada user, boleh melalui button, radio button, ataupun switch. Sehingga iklan tersebut tidak akan muncul kembali.
2. Untuk memastikan, Ketika aplikasi di jalankan, anda dapat mengeluarkan iklan yang sesungguhnya, anda harus menambahkan **AD UNIT**. Cari tahu cara untuk menambahkan Ad Unit ke dalam aplikasi yang baru saja anda bangun!