



# React Native OAuth

Chapter 6 - Topic 5

---

**Selamat datang di Chapter 6 Topic 5 online course  
React Native dari Binar Academy!**





# Heihoooo~

Materi topik 5 tentang Biometric Auth sudah kamu lewati. Congratz! Sekarang saatnya lanjut ke materi topik kelima yaaa. Si Chapter 6 Topic 5 ini, kamu akan belajar tentang **React Native OAuth**.

Let's start!





**Detailnya, kita bakal bahas hal-hal berikut ini:**

- Pengertian OAuth login
- Cara melakukan metode login dengan Google
- Cara melakukan metode social login





Aplikasi dan login ibarat perangko dan lem, nggak bisa dipisahin. Makanya kita akan belajar tentang **OAuth login**.

Lho memang apa kaitannya login sama Oauth login?

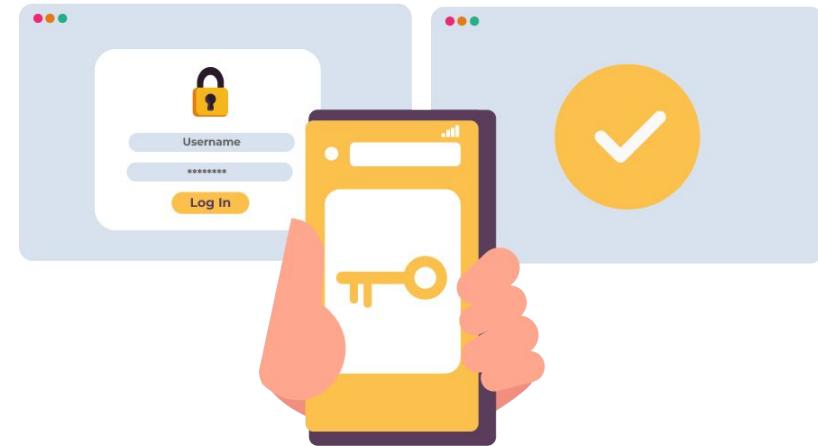




### Mau masuk aplikasi? Login dulu dong~

Yaps, login bukanlah hal yang asing untuk pengguna aplikasi. Hampir di semua aplikasi yang membutuhkan otentikasi menyediakan fitur login di dalamnya.

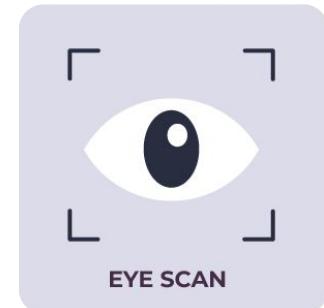
Memang apa sih gunanya? Ya untuk melihat siapa yang login ke dalam aplikasi tersebut dong.





**Ternyata, login nggak sembarang login.  
Ada metodenya, guys. Cari tahu yuk!**

Iya. Sekarang login punya banyak macam metode lho. Salah satunya seperti yang sudah kita pelajari pada chapter sebelumnya, yaitu **Biometrics Auth**.





**Selain Biometric Auth, ada juga metode lainnya nih.**

Kali ini kita nggak akan bahas tentang Biometric Auth lagi ya, guys. Di topik ini kita bakalan mempelajari metode lainnya yaitu **OAuth login**.

Kayak apa ya OAuth login itu?

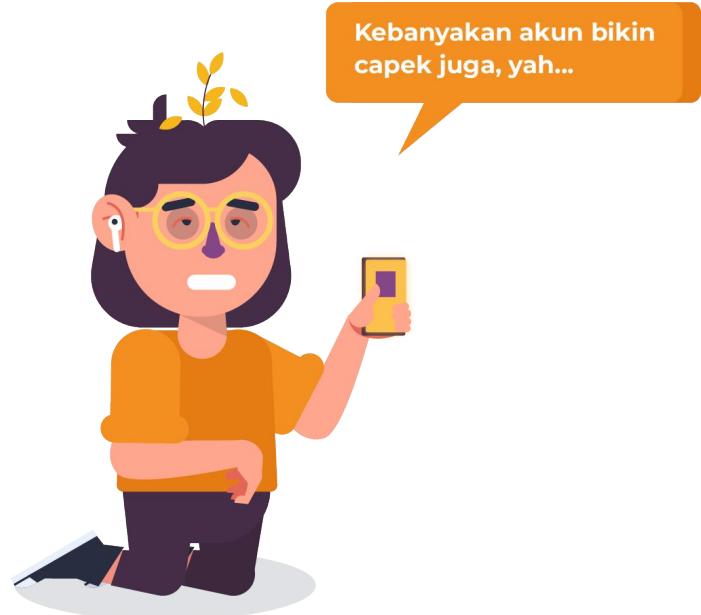




### “Ingat-ingat banyak password, males aku tuhhh!”

Hmm, mungkin sebagian dari kamu ada yang punya banyak social media. Ada yang buat pribadi, kerjaan, hobi, sampai kepoin gebetan, ups!

Itu baru sosmed, belum password e-mail, e-commerce dan aplikasi lainnya yang zaman now kayanya harus banget kamu punya kan.





### Eeh, udah gitu..

Saking banyaknya akun, pas mau login, lupa password. Males banget kan ya kalau harus inget semua password sosmed di kepala yang isinya sudah campur aduk kayak gado-gado ini?

Nah, metode OAutha login, akan memudahkan hidup kalian. Gimana caranya? Kepoin slide selanjutnya deh.





### Kayak gini lho OAuth login tu..

OAuth login menerapkan **SSO (Single Sign On)**, artinya kamu bisa melakukan **login hanya dengan menggunakan satu akun** saja. Jadinya lebih praktis.

Eh tapi.. kayak apa ya contoh OAuth login?

SSO:  
ALL FOR ONE ACCOUNT



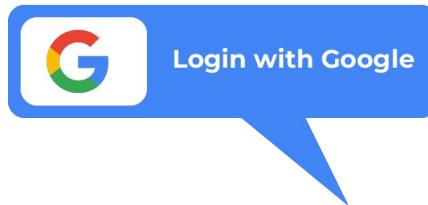
Bukan yang dari fandom  
sebelah, cuy!

Lagian MHA itu  
kali, mbak...



### Yak! Benar!

Salah satu OAuth login bisa kita lakukan dengan **login by Google**.



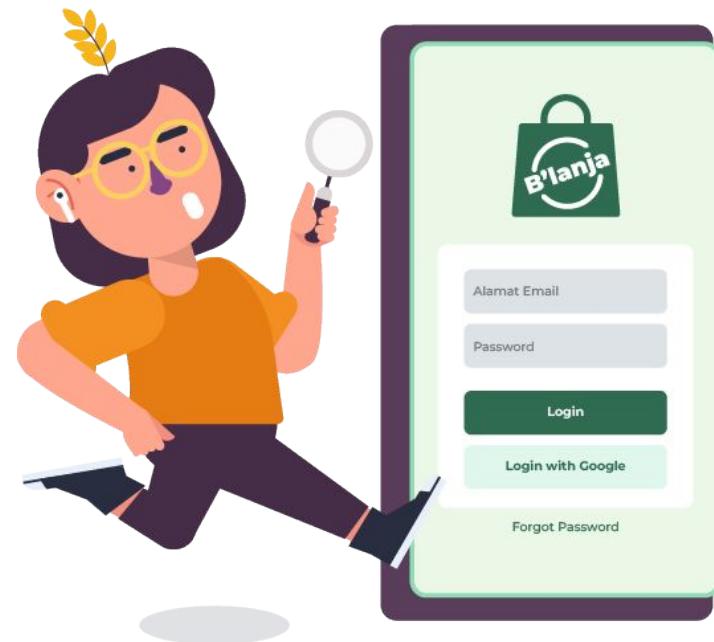


Contoh:

Kamu punya sebuah aplikasi, lalu kamu mau bikin agar proses login di aplikasimu jadi super gampang. Nah, kamu bisa terapin nih login by Google.

**Login by Google ini bisa kamu lakukan terutama untuk smartphone Android.**

Kenapa?





Karena **smartphone Android berasal dari Google**, guys, maka semua orang yang pakai Android pasti punya Google Account (Gmail). Dengan begitu, kamu bisa buat implementasi login by Google.





Login with Google

Oke, nggak usah basa-basi lagi, sekarang saatnya kamu belajar melakukan metode **Login with Google** yuk!





### Langkah pertama,

- Buatlah project baru react-native atau dengan menggunakan project pada chapter ini

Kamu akan menggunakan Firebase untuk melakukan login by Google, jadi pastikan kamu telah melakukan config dan instalasi Firebase.

Silakan cek materi sebelumnya [di sini](#) yaa





Login with Google



Salah satu layanan terbaik dari Firebase adalah Firebase Auth, di mana Firebase menyediakan layanan untuk membuat autentikasi dengan berbagai metode yang ada.





## Langkah kedua,

- Jalankan `npm install @react-native-firebase/auth` , kemudian cd iOS pod install untuk menambah dependency iOS



```
Scanning for project in directory /Users/mikhael/Downloads/pokemonApp
admins-MacBook-Pro-2:pokemonApp 161552.mikhael$ npm install @react-native-firebase/auth
```



Login with Google

### Langkah ketiga,

- Selanjutnya buka dokumentasi [berikut ini](#) untuk melihat langkah-langkah melakukan instalasi yang lebih lengkap dan scroll ke bagian Google:





### 1. Install react-native-google-signin/google-signin

Jalankan npm install @react-native-google-signin/google-signin

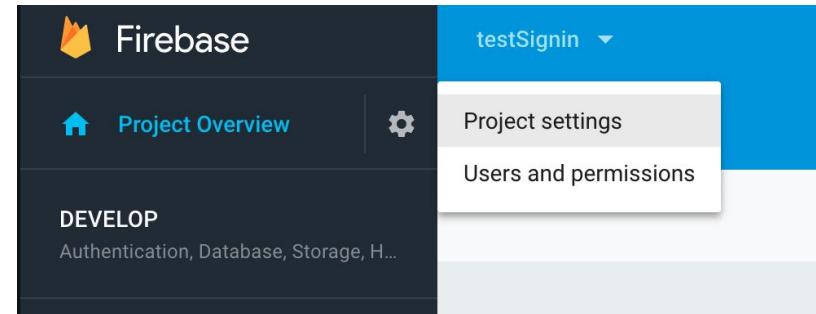


```
[admins-MacBook-Pro-2:pokemonApp 161552.mikhael$ npm install @react-native-google-signin/google-signin
((.....)) : rollbackFailedOptional: verb npm-session 571b8904d0f55d6
```



### 2.1 Google Project Configuration (Android)

- Untuk setup Android melalui Firebase, kamu perlu menambahkan SHA-1 pada project Firebase.
- Di bagian project setting, salin (paste) SHA 1 key, yang bisa kalian dapatkan dengan cd android, kemudian `./gradlew signingReport`.





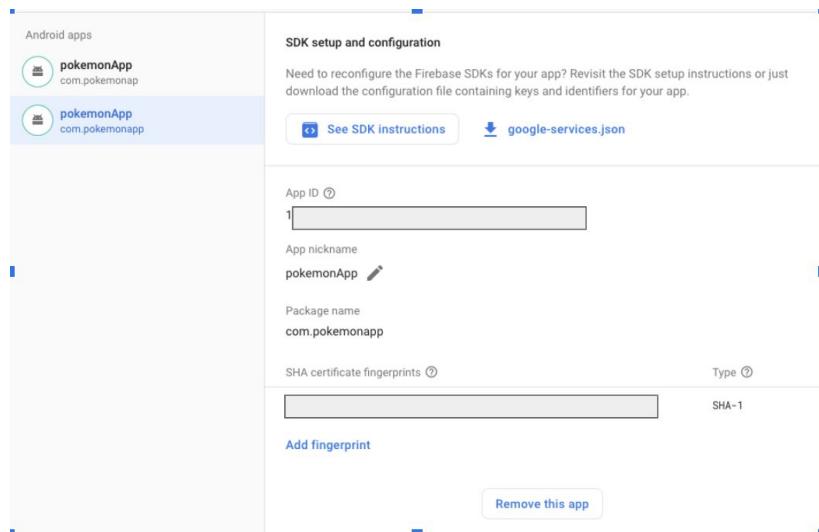
- Setelah menjalankan `./gradlew signingReport` pada langkah sebelumnya, selanjutnya akan muncul SHA-1 seperti gambar di samping.



```
> Task :app:signingReport
Variant: debug
Config: debug
Store: /Users/161552.mikhael/Documents/AlatTempur/pokemonApp/android/app/debug.keystore
Alias: androiddebugkey
MD5: [REDACTED]
SHA1: [REDACTED]
SHA-256: [REDACTED]
:83:32:66:75:91:03:38:9C
Valid until: Wednesday, May 1, 2052
```



- Selanjutnya kamu perlu paste SHA-1 tersebut di file project Firebase. Klik add fingerprint dan tambahkan SHA-1 yang kamu dapatkan [dari sini](#)



The screenshot shows the Firebase Project Overview page. On the left, under 'Android apps', there are two entries: 'pokemonApp' (com.pokemonapp) and another 'pokemonApp' entry (com.pokemonapp). To the right, the 'SDK setup and configuration' section is displayed. It includes a note about reconfiguring Firebase SDKs, a 'See SDK instructions' button, and a download link for 'google-services.json'. Below this, the 'App ID' field contains '1:...', 'App nickname' is set to 'pokemonApp', and the 'Package name' is 'com.pokemonapp'. Under 'SHA certificate fingerprints', there is a text input field containing 'SHA-1' and a 'Type' dropdown set to 'SHA-1'. A blue 'Add fingerprint' button is located below the SHA-1 input field. At the bottom right of the configuration area, there is a 'Remove this app' button.

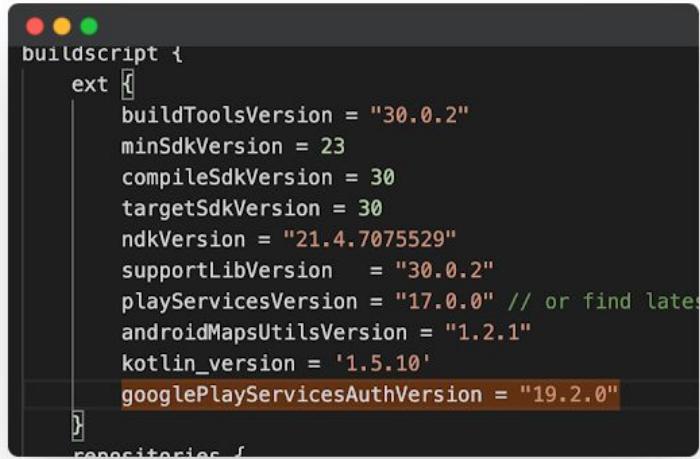


- Jika saat kamu menjalankan `./gradlew signingReport` menemukan error masalah dari Java, coba lakukan manual keytool command pada link [berikut](#) ya!





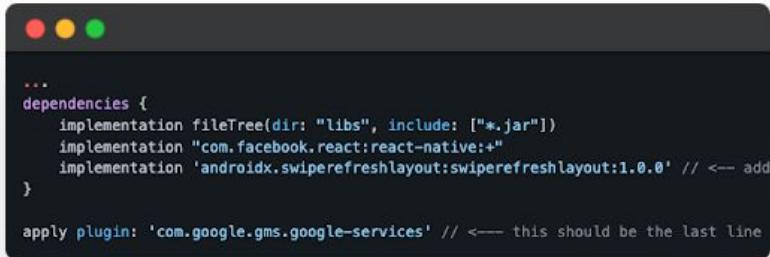
- Selanjutnya update file android/build.gradle-mu dengan gambar di samping. Pastikan paste yang belum ada ya.



```
buildscript {
    ext {
        buildToolsVersion = "30.0.2"
        minSdkVersion = 23
        compileSdkVersion = 30
        targetSdkVersion = 30
        ndkVersion = "21.4.7075529"
        supportLibVersion = "30.0.2"
        playServicesVersion = "17.0.0" // or find latest
        androidMapsUtilsVersion = "1.2.1"
        kotlin_version = '1.5.10'
        googlePlayServicesAuthVersion = "19.2.0"
    }
}
repositories {
```



- Selanjutnya kamu perlu menambahkan baris di samping di file android/app/build.gradle



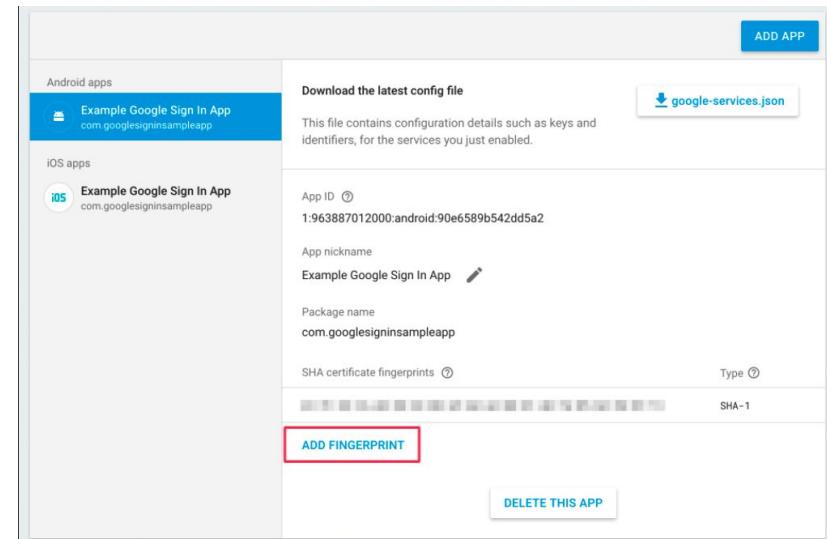
```
...
dependencies {
    implementation fileTree(dir: "libs", include: ["*.jar"])
    implementation "com.facebook.react:react-native:+"
    implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.0.0' // <-- add
}
apply plugin: 'com.google.gms.google-services' // <-- this should be the last line
```



## 2.2 Google Project Configuration

Balik lagi pada ke Firebase karena kamu baru saja menambahkan fingerprint SHA-1. P

Pastikan kamu mendownload ulang file google-service.json, lalu replace yang lama dengan yang baru.



The screenshot shows the 'Google Project Configuration' page in the Firebase console. On the left, there's a sidebar with 'Android apps' and 'iOS apps'. Under 'Android apps', the 'Example Google Sign In App' (package name: com.googleSignInSampleApp) is listed. To the right, there's a section titled 'Download the latest config file' with a download button labeled 'google-services.json'. Below this, there are fields for 'App ID' (1:963887012000:android:90e6589b542dd5a2), 'App nickname' (Example Google Sign In App), 'Package name' (com.googleSignInSampleApp), 'SHA certificate fingerprints' (Type SHA-1), and a 'DELETE THIS APP' button. At the bottom, a red box highlights the 'ADD FINGERPRINT' button.



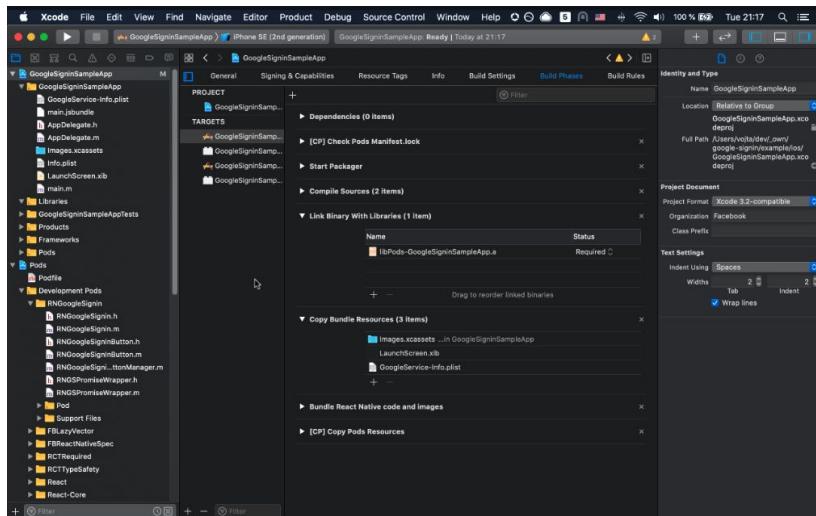
### 2.3. Google Project Configuration (iOS)

- Untuk iOS, tambahkan baris pod 'GoogleSignIn', '~> 6.1.0' di podfile yang bisa kamu temukan di ios/projectName/podfile. Selanjutnya, jalankan pod install.





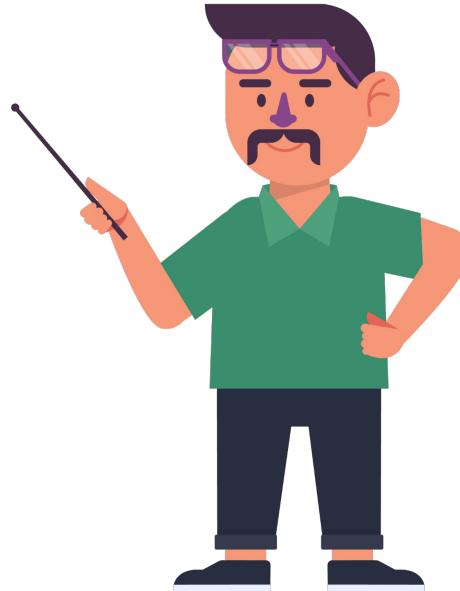
- Kemudian bukalah project ios di xcode. Jika sudah, link dengan RN >=60, sehingga akan menampilkan seperti pada gambar di samping.





- Seperti pada Android, pada iOS kamu juga perlu menambahkan SHA-1.

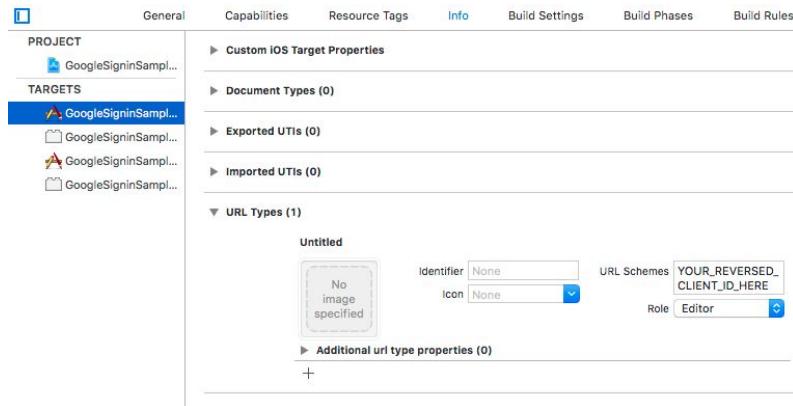
Setelah menambahkan Add iOS Project di Firebase dashboard, download GoogleService-Info.plist di akhir proses.





- Xcode configuration. Selanjutnya kamu perlu menambahkan reserved\_client\_id yang bisa kamu temukan di GoogleService-Info.plist.

Tambahkan reserved\_client\_id pada urltypes di xcode dan isikan url\_schemes.





- Selanjutnya buka file AppDelegate.m, lalu tambahkan #import <RNGoogleSignin/RNGoogleSignin.h>. Setelah itu, tambahkan juga line seperti di atas.

```
#import <FlipperKitReactPlugin/FlipperKitReactPlugin.h>
#import <RNGoogleSignin/RNGoogleSignin.h>
```

```
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url options:(NSDictionary<NSString *,id> *)options {
    return [[FBSDKApplicationDelegate sharedInstance] application:application openURL:url options:options] || [RNGoogleSignIn application:application openURL:url options:options];
}
```



- Selanjutnya, kamu perlu melakukan set enable untuk login pada Firebase.

Dari Firebase dashboard ke bagian authentication, lalu pada provider google set untuk enable.

Provider	Status	
 Google	<input checked="" type="checkbox"/> Enabled	



### 3. Example Code

Selanjutnya kamu akan melakukan contoh sederhana pada login Google.

- Buatlah file login.js pada project React Native-mu, kemudian tambahkan code seperti di samping.
- Import GoogleSignin pada packages, kemudian panggilah .configure() function dengan memberikan webClientId.

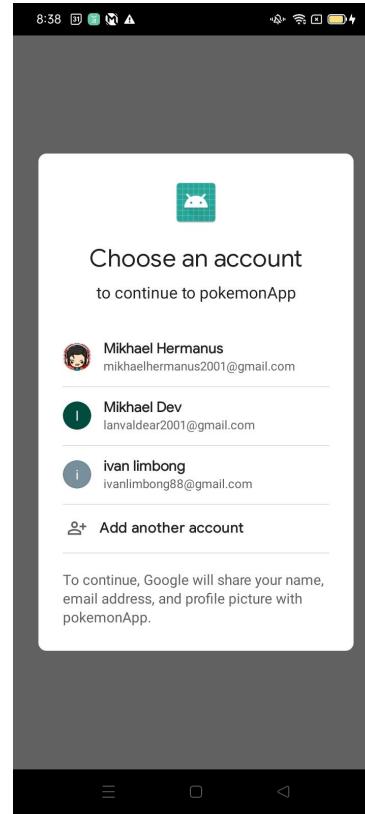


```
import { GoogleSignin } from '@react-native-google-signin/google-signin';

const Login = () => {
  useEffect(() => {
    try {
      GoogleSignin.configure({
        webClientId: 'REDACTED',
      })
    } catch (error) {
      alert(error)
    }
  }, [])
}
```



- Kalau berhasil maka akan menampilkan seperti pada gambar di samping, guys.





- Kalau sudah berhasil, perhatikan respon dari userInfo yang kamu dapat dari Google. Dengan menggunakan informasi dari Google tersebut, kamu bisa menampilkan sesuai kebutuhan informasi.

```
LogIn.js:57
{
  scopes: Array(2),
  serverAuthCode: null,
  idToken: "eyJhbGciOiJSUzI1NiIsImtpZCI6ImYxM2M4Y2EyNjgzNTg2M2...a--v9tmxxBGGU_eQAIJYp5nMAbmlo9y2yYVc01J2FxSguYYcg",
  user: {...} ⓘ
    idToken: "eyJhbGciOiJSUzI1NiIsImtpZCI6ImYxM2M4Y2EyNjgzNTg2M2Y2NzE0MDhmNDE3MzhhN2I0OWU3NDB...",
    scopes: (2) ["https://www.googleapis.com/auth/userinfo.profile", "https://www.googleapis..."],
    serverAuthCode: null
  user: {...}
    email: "mikhaelhermanus2001@gmail.com"
    familyName: "Hermanus"
    givenName: "Mikhael"
    id: "105170787335090012252"
    name: "Mikhael Hermanus"
    photo: "https://lh3.googleusercontent.com/a-/AOh14GiY4d6rVZxp-i2bEug2gcflWEXRy7Srb004E-...",
    __proto__: Object
  __proto__: Object
}
```



### Selesai deh!

Sampai sini, berarti kamu sudah berhasil melakukan login dengan Google. Kalau kamu menghadapi beberapa kendala, silahkan cek dokumentasi untuk debug error message [berikut](#).

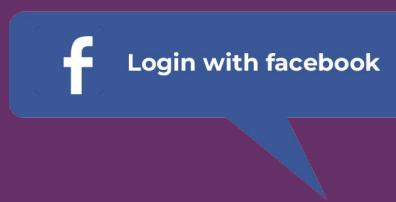
Pastikan SHA 1 yang di-paste untuk Firebase sudah sesuai, dan webclient yang dipanggil di code sama dengan yg terdaftar pada Firebase dashboard ya.





Login with Facebook

Eits, tunggu dulu, ternyata selain dengan Google, OAuth login juga bisa dilakukan dengan Facebook lho. Cuss kita bahas tentang **login with Facebook**.





### Jalan Ninja lainnya: login with Facebook

Yak! Setelah kamu coba membuat SSO login by Google, selanjutnya kamu akan mencoba cara login social lainnya yaitu Facebook.

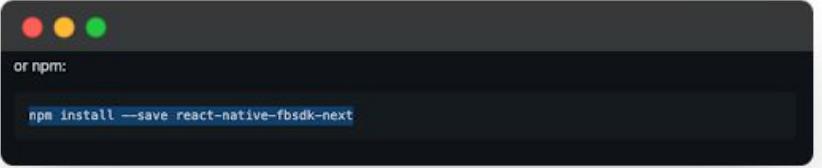
Bagaimana caranya?





### 1. Setting Facebook Developer Account & Keys

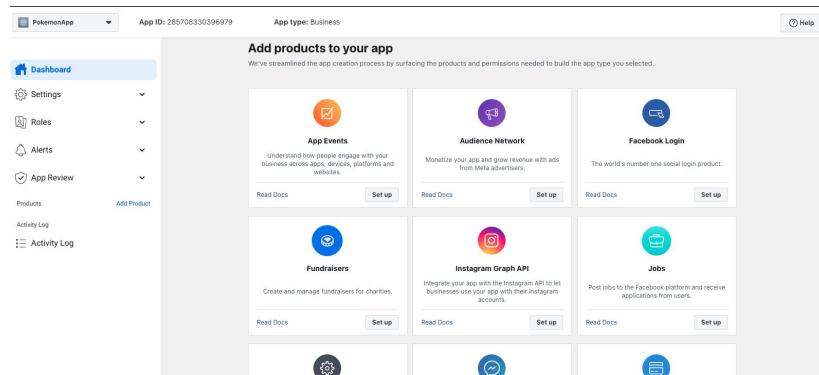
- Install packages react-native-fbsdk-next melalui terminal, kemudian cd ios , pod install untuk installing ios dependencies



```
or npm:  
npm install --save react-native-fbsdk-next
```

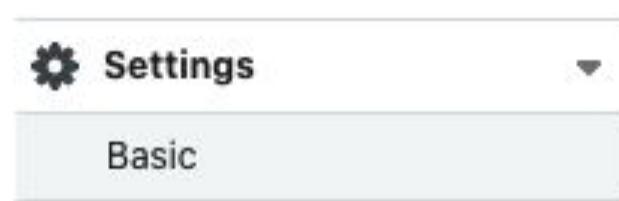


- Buka <https://developers.facebook.com/> kemudian buatlah akun pada web tersebut sampai kamu punya tampilan dashboard seperti di samping.





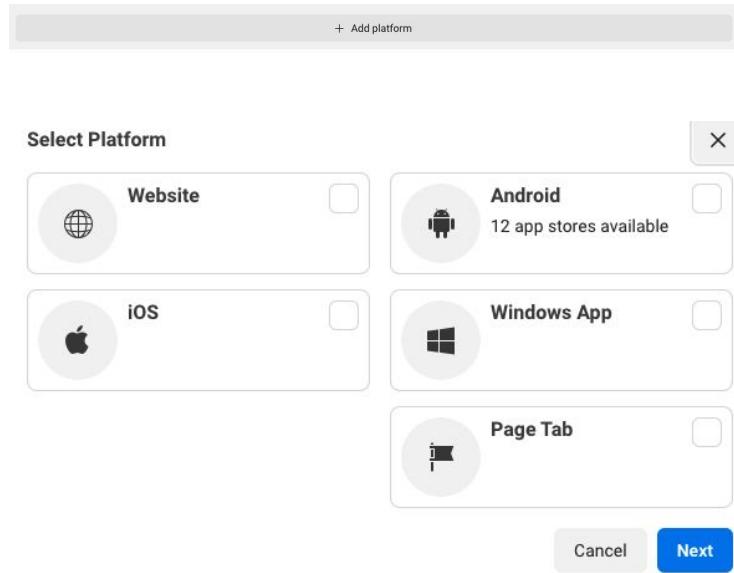
- Setelah itu, kamu perlu mencari app id, yang bisa kamu dapatkan pada menu di samping, pilih settings -> basic.





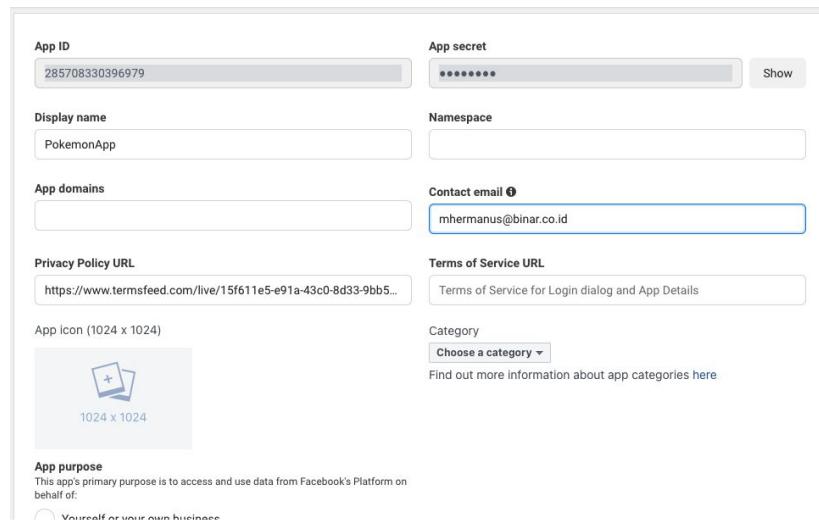
### 2.1 Setup (untuk iOS)

- Pada tampilan basic setting di Facebook developer dashboard akan ada button add platform, klik iOS.  
Lalu isi bundle ID sesuai iOS project-mu. Jika sudah, klik save and change.





- Kalau kamu diminta untuk memasukkan privacy policy URL, buat lah dari termsfeed.com, kemudian paste URL-mu.



The screenshot shows the 'App Review Settings' section of a Facebook developer page for an iOS app. The app's ID is 285708330396979 and its secret is masked as '\*\*\*\*\*'. The display name is 'PokemonApp', and the contact email is mhermanus@binar.co.id (highlighted with a blue border). The app domains field is empty. The privacy policy URL is set to https://www.termsfeed.com/live/15f611e5-e91a-43c0-8d33-9bb5... and the terms of service URL is Terms of Service for Login dialog and App Details. An app icon placeholder is shown, and the category is set to 'Choose a category'. A note indicates that the app's primary purpose is to access and use data from Facebook's Platform on behalf of the user, with the option to choose 'Yourself or your own business'.

App ID  
285708330396979

App secret  
\*\*\*\*\* Show

Display name  
PokemonApp

Namespace

App domains

Contact email ⓘ  
mhermanus@binar.co.id

Privacy Policy URL  
<https://www.termsfeed.com/live/15f611e5-e91a-43c0-8d33-9bb5...>

Terms of Service URL  
Terms of Service for Login dialog and App Details

App icon (1024 x 1024)  
1024 x 1024

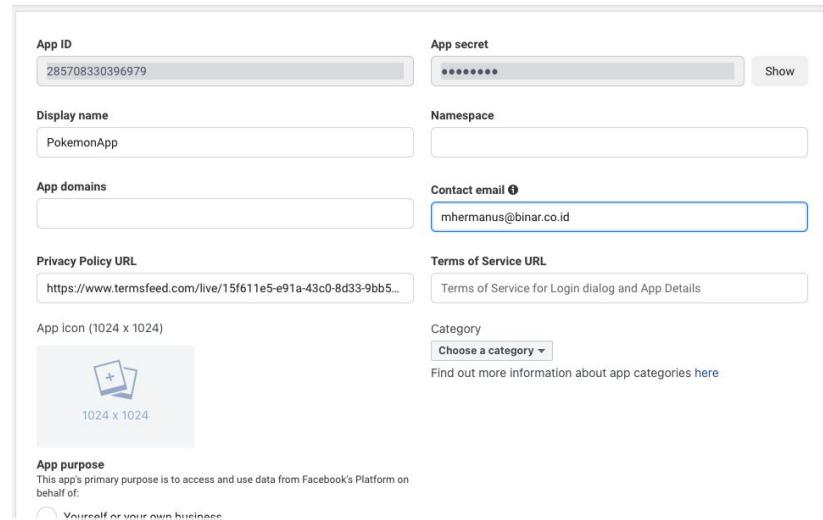
Category  
Choose a category

App purpose  
This app's primary purpose is to access and use data from Facebook's Platform on behalf of:  
 Yourself or your own business



- Selanjutnya, buka project iOS-mu di Xcode.  
Buka AppDelegate.m dan import

```
#import <FBSDKCoreKit/FBSDKCoreKit.h>
```



The screenshot shows the Facebook App Dashboard configuration page. It includes fields for App ID (285708330396979), App secret (redacted), Display name (PokemonApp), App domains (empty), Namespace (empty), Contact email (mhermanus@binar.co.id), Privacy Policy URL (https://www.termsfeed.com/live/15f611e5-e91a-43c0-8d33-9bb5...), Terms of Service URL (Terms of Service for Login dialog and App Details), App icon (1024 x 1024) (a placeholder icon), Category (Choose a category), and App purpose (This app's primary purpose is to access and use data from Facebook's Platform on behalf of: Yourself or your own business).

App ID: 285708330396979

App secret: \*\*\*\*\*

Show

Display name: PokemonApp

Namespace:

App domains:

Contact email: mhermanus@binar.co.id

Privacy Policy URL: https://www.termsfeed.com/live/15f611e5-e91a-43c0-8d33-9bb5...

Terms of Service URL: Terms of Service for Login dialog and App Details

App icon (1024 x 1024):  1024 x 1024

Category: Choose a category

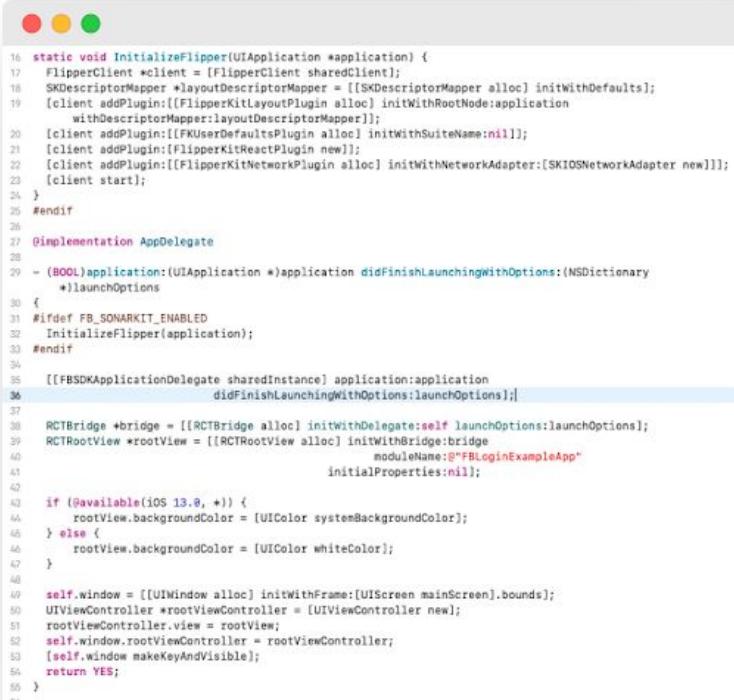
Find out more information about app categories here

App purpose: This app's primary purpose is to access and use data from Facebook's Platform on behalf of:  
 Yourself or your own business



- Pada baris didFinishLaunchingWithOptions tambahkan:

```
[ [FBSDKApplicationDelegate  
sharedInstance]  
application:applatio
```



```
16 static void InitializeFlipper(UIApplication *application) {  
17     FlipperClient *client = [FlipperClient sharedClient];  
18     SKDescriptorMapper *layoutDescriptorMapper = [[SKDescriptorMapper alloc] initWithDefaults];  
19     [client addPlugin:[[FlipperKitLayoutPlugin alloc] initWithRootNode:application  
20         withDescriptorMapper:layoutDescriptorMapper]];  
21     [client addPlugin:[[FKUserDefaultsPlugin alloc] initWithSuiteName:@"nil"]];  
22     [client addPlugin:[[FlipperKitReactPlugin alloc] initWithNetworkAdapter:[SKIOSNetworkAdapter new]]];  
23     [client start];  
24 }  
25 #endif  
26  
27 @implementation AppDelegate  
28  
29 - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary  
30     *)launchOptions  
31 {  
32     #ifdef FB_SONARKIT_ENABLED  
33         InitializeFlipper(application);  
34     #endif  
35     [[FBSDKApplicationDelegate sharedInstance] application:application  
36             didFinishLaunchingWithOptions:launchOptions];  
37  
38     RCTBridge *bridge = [[RCTBridge alloc] initWithDelegate:self launchOptions:launchOptions];  
39     RCTRootView *rootView = [[RCTRootView alloc] initWithBridge:bridge  
40                               moduleName:@"FBLoginExampleApp"  
41                               initialProperties:nil];  
42  
43     if (@available(iOS 13.0, +)) {  
44         rootView.backgroundColor = [UIColor systemBackgroundColor];  
45     } else {  
46         rootView.backgroundColor = [UIColor whiteColor];  
47     }  
48  
49     self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];  
50     UIViewController *rootViewController = [UIViewController new];  
51     rootViewController.view = rootView;  
52     self.window.rootViewController = rootViewController;  
53     [self.window makeKeyAndVisible];  
54     return YES;  
55 }
```

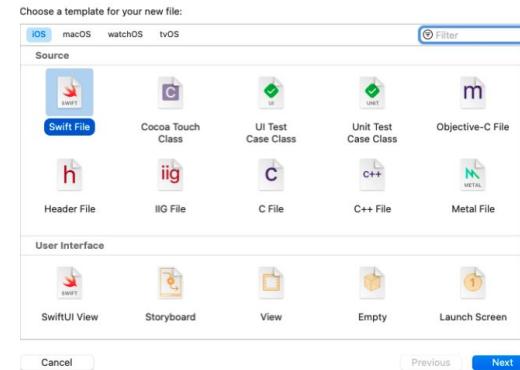
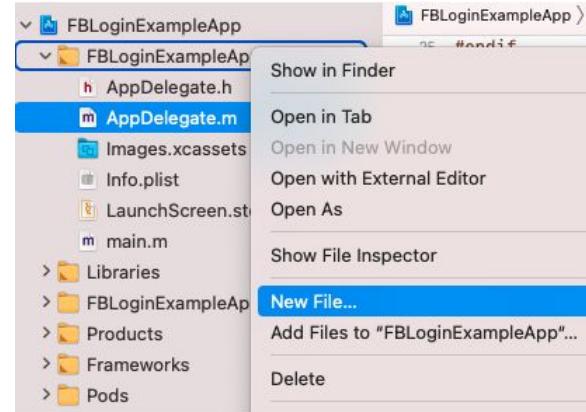


- Kemudian tepat di atas function tambahkan baris codingan seperti line di blok di samping.

```
44     rootView.backgroundColor = [UIColor systemBackgroundColor];
45 } else {
46     rootView.backgroundColor = [UIColor whiteColor];
47 }
48
49 self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
50 UIViewController *rootViewController = [UIViewController new];
51 rootViewController.view = rootView;
52 self.window.rootViewController = rootViewController;
53 [self.window makeKeyAndVisible];
54 return YES;
55 }
56
57 -(BOOL)application:(UIApplication *)app
58 openURL:(NSURL *)url
59 options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> *)options
60 {
61
62     if ([[FBSDKApplicationDelegate sharedInstance] application:app openURL:url options:options]) {
63         return YES;
64     }
65
66     return NO;
67 }
68 }
```

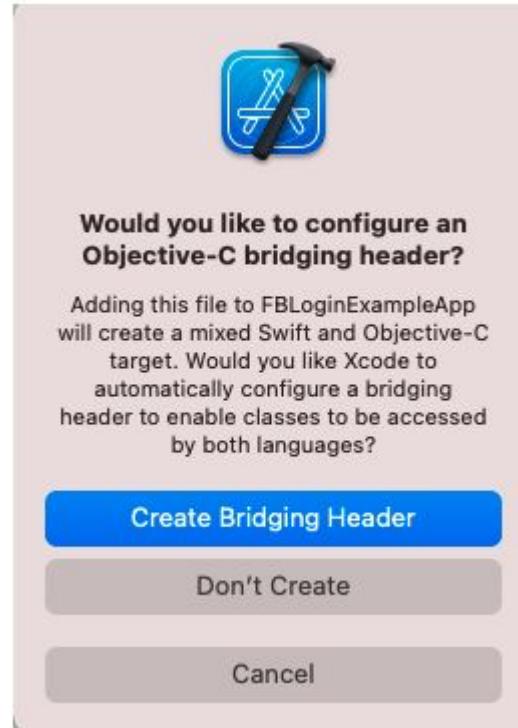


- Lanjut, buatlah Swift file. Setelahnya, klik kanan pada Folder Project, lalu klik New File, pilih Swift file dan klik Next.



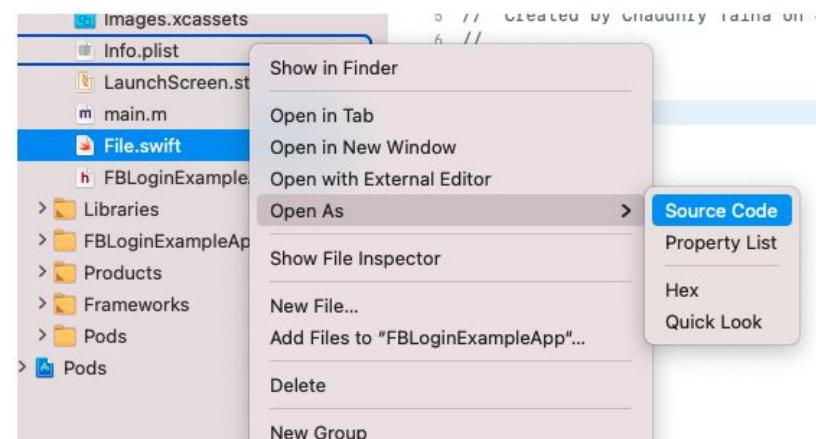


- Pastikan kamu memilih Create Bridging Header kalau muncul pop-up seperti di samping.





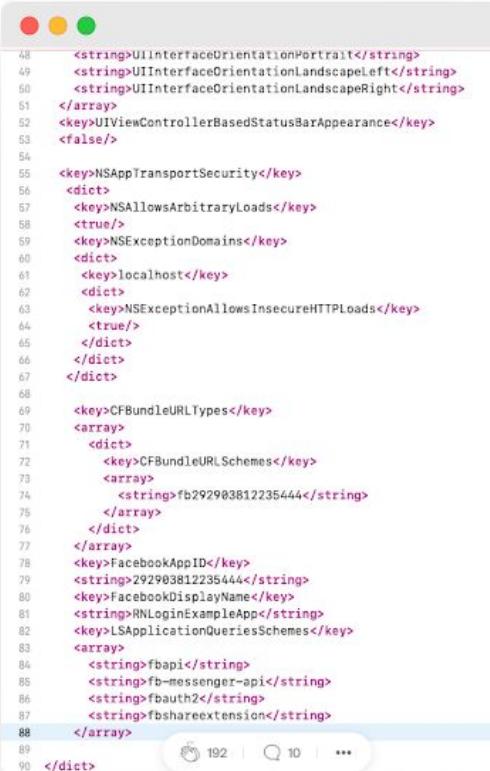
- Klik kanan pada Info.plist kemudian klik open As - Source code





- Tambahkan code berikut sebelum tag terakhir </dict>

Cek link [berikut](#) ini pada step 2 untuk copy code, pastikan untuk mengganti FacookAppID dan FacebookDisplayName pada file tersebut. Kemudian Save dan tutup Xcode



```
48     <string>UIInterfaceOrientationPortrait</string>
49     <string>UIInterfaceOrientationLandscapeLeft</string>
50     <string>UIInterfaceOrientationLandscapeRight</string>
51 </array>
52 <key>UIViewControllerBasedStatusBarAppearance</key>
53     <false/>
54
55 <key>NSAppTransportSecurity</key>
56 <dict>
57     <key>NSAllowsArbitraryLoads</key>
58         <true/>
59     <key>NSExceptionDomains</key>
60         <dict>
61             <key>localhost</key>
62             <dict>
63                 <key>NSEExceptionAllowsInsecureHTTPLoads</key>
64                 <true/>
65             </dict>
66         </dict>
67     </dict>
68
69 <key>CFBundleURLTypes</key>
70 <array>
71     <dict>
72         <key>CFBundleURLSchemes</key>
73         <array>
74             <string>fb292903812235444</string>
75         </array>
76     </dict>
77 </array>
78 <key>FacebookAppID</key>
79     <string>292903812235444</string>
80 <key>FacebookDisplayName</key>
81     <string>RNLoginExampleApp</string>
82 <key>LSApplicationQueriesSchemes</key>
83     <array>
84         <string>fbapi</string>
85         <string>fb-messenger-api</string>
86         <string>fbauth2</string>
87         <string>fbshareextension</string>
88     </array>
89
90 </dict>
```



### Select Platform



[Website]



[iOS]



[Android]



[Windows App]



[Page Tab]



[Xbox]



[PlayStation]

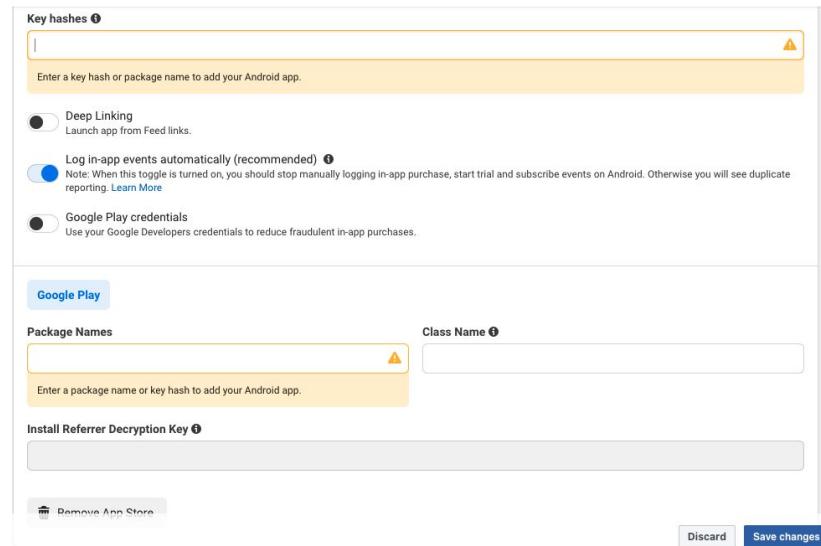
## 2.2 Setup (untuk Android)

- Kembali pada halaman Facebook developer, kemudian tambahkan + add Platform dan pilih Android.

[Cancel](#)



- Buka androidManifest.xml dan copy package name ke bagian mengisi packagename.
- Selanjutnya keyhashes jalankan cd android dan keytool -keystore app/debug.keystore -list -v



The screenshot shows the configuration page for integrating Facebook login into an Android application. It includes fields for entering key hashes, toggles for Deep Linking and automatic log-in app events, and a section for Google Play setup. The Google Play section contains fields for package names and class names, along with a note about manually logging in-app purchases. At the bottom, there are buttons for removing the app from the App Store and saving changes.

Key hashes ⓘ

Enter a key hash or package name to add your Android app.

Deep Linking  
Launch app from Feed links.

Log in-app events automatically (recommended) ⓘ  
Note: When this toggle is turned on, you should stop manually logging in-app purchase, start trial and subscribe events on Android. Otherwise you will see duplicate reporting. [Learn More](#)

Google Play credentials  
Use your Google Developers credentials to reduce fraudulent in-app purchases.

Google Play

Package Names

Class Name ⓘ

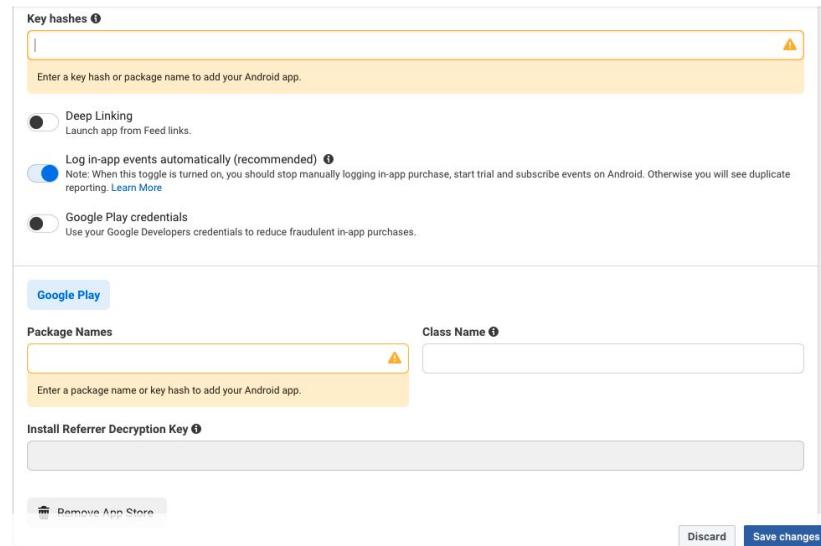
Install Referrer Decryption Key ⓘ

Remove App Store

Discard Save changes



- Jika diminta password, isi dengan “android” atau jika salah bug, debug.keystore di file Android dan cari password.
- Selanjutnya copy SHA-1 untuk kamu paste di Facebook developer.



The screenshot shows the 'Key hashes' section of the Facebook Developer Platform. It includes a text input field for entering a key hash or package name, a note about Deep Linking, a toggle for logging in-app events automatically (which is turned on), and a note about Google Play credentials. Below this is the 'Google Play' section, which contains fields for Package Names and Class Name, both with notes and input fields. At the bottom are buttons for 'Remove App Store', 'Discard', and 'Save changes'.

Key hashes ⓘ

Enter a key hash or package name to add your Android app.

Deep Linking  
Launch app from Feed links.

Log in-app events automatically (recommended) ⓘ  
Note: When this toggle is turned on, you should stop manually logging in-app purchase, start trial and subscribe events on Android. Otherwise you will see duplicate reporting. [Learn More](#)

Google Play credentials  
Use your Google Developers credentials to reduce fraudulent in-app purchases.

Google Play

Package Names

Class Name ⓘ

Enter a package name or key hash to add your Android app.

Install Referrer Decryption Key ⓘ

Remove App Store

Discard Save changes



- Tambahkan line berikut :

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

- dan di atas line <application/> tambahkan

```
<activity  
    android:name="com.facebook.react.devsupport.D  
evSettingsActivity" />
```



```
android > app > src > main > AndroidManifest.xml  
1  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  
2      package="com.fbloginexampleapp"  
3  
4      <uses-permission android:name="android.permission.INTERNET" />  
5  
6      <application  
7          android:name=".MainApplication"  
8          android:label="@string/app_name"  
9          android:icon="@mipmap/ic_launcher"  
10         android:roundIcon="@mipmap/ic_launcher_round"  
11         android:allowBackup="false"  
12         android:theme="@style/AppTheme">  
13             <activity  
14                 android:name=".MainActivity"  
15                 android:label="@string/app_name"  
16                 android:configChanges="keyboard|keyboardHidden|orientation|screenSize|uiMode"  
17                 android:launchMode="singleTask"  
18                 android:windowSoftInputMode="adjustResize">  
19                     <intent-filter>  
20                         <action android:name="android.intent.action.MAIN" />  
21                         <category android:name="android.intent.category.LAUNCHER" />  
22                     </intent-filter>  
23             </activity>  
24             <activity android:name="com.facebook.react.devsupport.DevSettingsActivity" />  
25         </application>  
26     </manifest>  
27 
```



- Selanjutnya tambahkan pada bagian meta data line berikut :

```
<meta-data  
    android:name="com.facebook.sdk.ApplicationId"  
    android:value="@string/facebook_app_id"/>
```



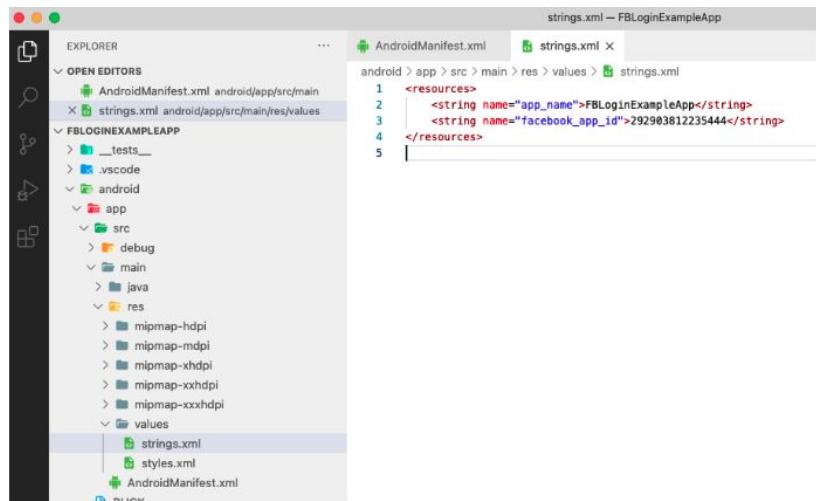
```
1  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  
2      package="com.fbloginexampleapp">  
3  
4      <uses-permission android:name="android.permission.INTERNET" />  
5  
6      <application  
7          android:label="@string/app_name"  
8          android:icon="@mipmap/ic_launcher"  
9          android:roundIcon="@mipmap/ic_launcher_round"  
10         android:allowBackup="false"  
11         android:theme="@style/AppTheme">  
12             <meta-data android:name="com.facebook.sdk.ApplicationId" android:value="@string/facebook_app_id"/>  
13         <activity  
14             android:name=".MainActivity"  
15             android:label="@string/app_name"  
16             android:configChanges="keyboard|keyboardHidden|orientation|screenSize|uiMode"  
17             android:launchMode="singleTask"  
18             android:windowSoftInputMode="adjustResize">  
19                 <intent-filter>  
20                     <action android:name="android.intent.action.MAIN" />  
21                     <category android:name="android.intent.category.LAUNCHER" />  
22                 </intent-filter>  
23             </activity>  
24             <activity android:name="com.facebook.react.devsupport.DevSettingsActivity" />  
25         </application>  
26     </manifest>  
27  
28
```



- Selanjutnya buka file /android/app/src/main/res/values/strings.xml dan tambahkan line berikut:

```
<string  
name="facebook_app_id">292903812235444</string  
>
```

Pastikan ada app\_name dan facebook\_app\_id kamu.



The screenshot shows the Android Studio interface with the project 'FBLOGINEEXAMPLEAPP' open. The left sidebar displays the project structure, including the 'src' folder which contains 'main' and 'java'. The 'main' folder has subfolders for 'res' (containing 'values') and 'AndroidManifest.xml'. The 'values' folder contains 'strings.xml' and 'styles.xml'. The right side shows the code editor for 'strings.xml' with the following content:

```
1 <resources>  
2     <string name="app_name">FBLoginExampleApp</string>  
3     <string name="facebook_app_id">292903812235444</string>  
4 </resources>
```



## 3. Example

- Buatlah code sederhana seperti di samping, dengan text yang dapat diklik bernama login with Facebook.

```
src > component > FacebookLogin.js > LoginFacebook
1  import React from 'react';
2  import { Button, Text, View } from 'react-native';
3  import { LoginManager } from 'react-native-fbsdk-next'
4  function LoginFacebook(props) {
5    return (
6      <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
7        <Text>Facebook Login React Native Example</Text>
8        <Button title="Login with Facebook" onPress={() => {
9          LoginManager.loginWithPermissions(["public_profile", "email"]).then(
10            function (result) {
11              if (result.isCancelled) {
12                alert("Login Cancelled " + JSON.stringify(result));
13              } else {
14                alert("Login success with permissions: " + result.grantedPermissions.toString());
15                alert("Login Success " + result.toString());
16              }
17            },
18            function (error) {
19              alert("Login failed with error: " + error);
20            }
21          )
22        }} />
23      </View>
24    );
25  }
26  export default LoginFacebook;
```

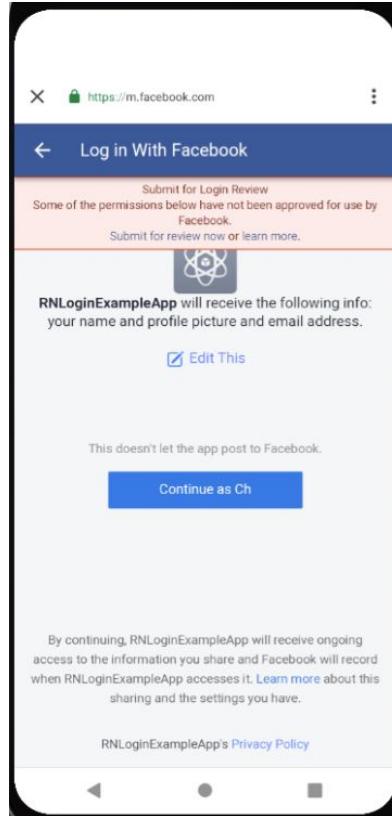


- Jika berjalan sesuai, maka tampilan aplikasi akan seperti gambar di samping. Cobalah untuk mengklik button tersebut.





- Kayak Google, guys, ia akan menampilkan pop-up default dari Facebook.





- Nah, kalau berhasil login maka akan tampil alert yg berisi object data seperti di samping ya.





**Selesai~**

Kamu sudah berhasil membuat login with social media.  
Coba implementasikan pada project-project kamu yaa.

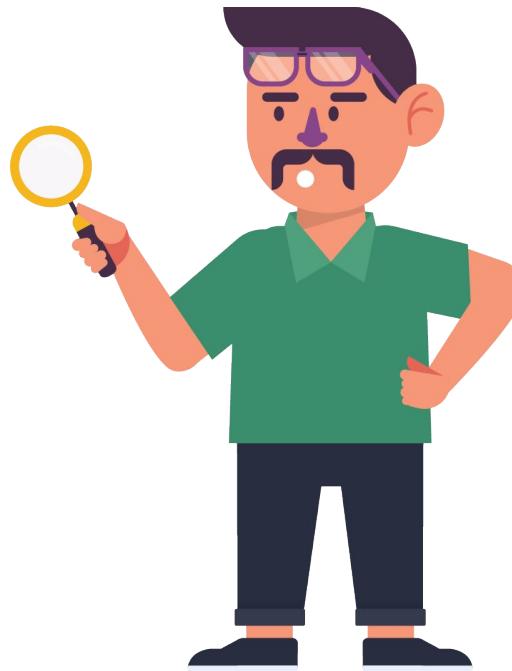




## Sumber-sumber

[official facebook sdk documentation](#)

[example and debugging tutorial](#)



# Terima Kasih!



Chapter ✓

completed