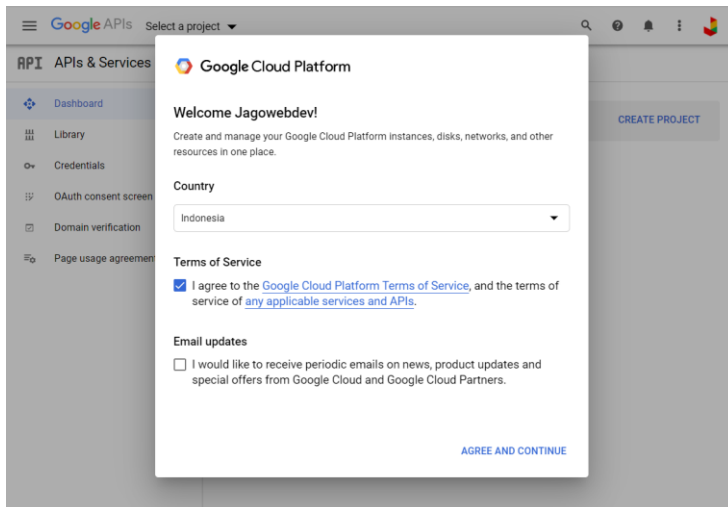


The Gmail Oauth Setup

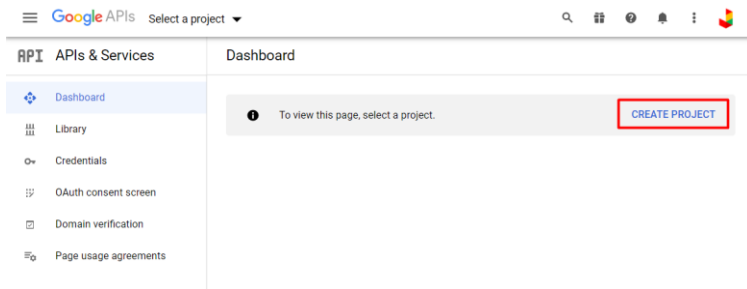
Agar dapat mengirim email menggunakan Gmail Oauth Authentication, kita perlu membuat setidaknya tiga key, yaitu **client_id**, **client_secret**, dan **refresh_token**. Berikut langkah untuk membuat key tersebut.

I. Membuat client_id and client_secret

1. Pertama, kunjungi halaman Google Developer Console melalui url URL: <https://console.developers.google.com/apis/dashboard>. Anda akan diminta konfirmasi penggunaan layanan Google Cloud Platform



Centang bagian I Agree... kemudian klik tombol AGREE AND CONTINUE. Selanjutnya, klik tombol "Create Project"



2. Selanjutnya isikan nama project, kemudian klik tombol **"Create"**

Google APIs Search for APIs and Services

New Project

Warning: You have 12 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)
[MANAGE QUOTAS](#)

Project name *
Jagowebdev Mail ?

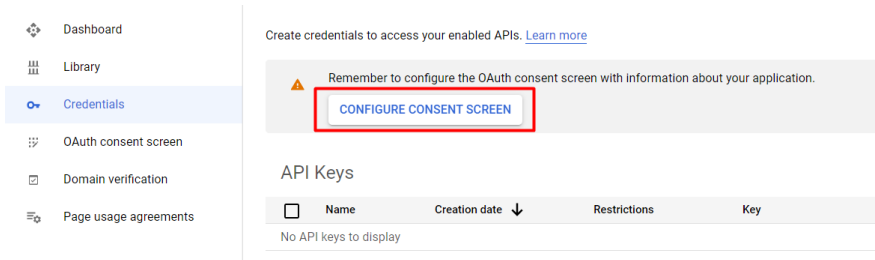
Project ID: jagowebdev-mail. It cannot be changed later. [EDIT](#)

Location *
 No organization [BROWSE](#)

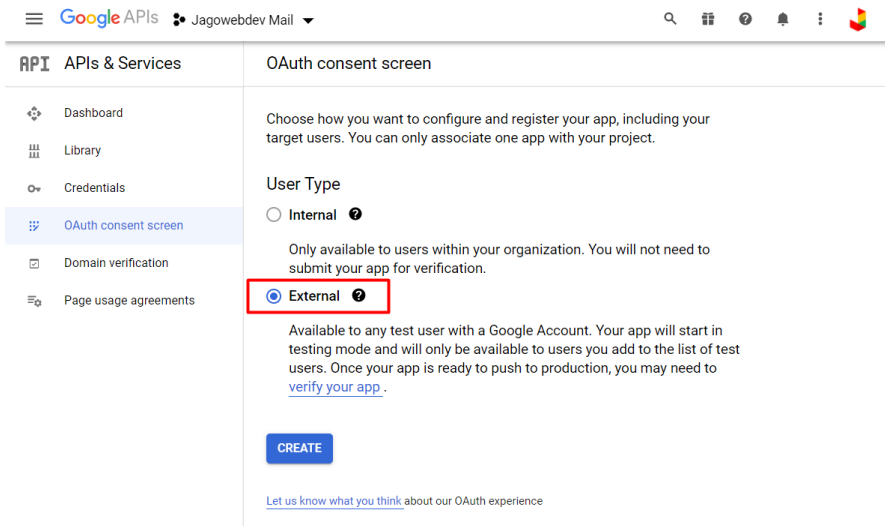
Parent organization or folder

CREATE CANCEL

3. Selanjutnya, pada panel sebelah kiri, klik Credentials, kemudian pada panel sebelah kanan klik CONFIGURE CONSENT SCREEN.



4. Pada halaman OAuth consent screen yang muncul, pilih opsi **External** kemudian klik **CREATE**



5. Selanjutnya, pada halaman Edit app registration, isikan App Name dan User support email

APIs & Services

Dashboard
Library
Credentials
OAuth consent screen
Domain verification
Page usage agreements

Edit app registration

1 OAuth consent screen — 2 Scopes — 3 Test users — 4 Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *

jagowebdev

The name of the app asking for consent

User support email *

jagowebdev@gmail.com

For users to contact you with questions about their consent

App logo

BROWSE

Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 128px by 128px for the best results.

App domain

Learn

How is this info presented to users?
This is the consent screen that users see

Selanjutnya scroll kebawah dan isikan email pada bagian Developer contact information kemudian klik **SAVE AND CONTINUE**

Authorized domains

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorized. [Learn more](#) about the authorized domain limit.

+ ADD DOMAIN

Developer contact information

Email addresses *

jagowebdev@gmail.com

These email addresses are for Google to notify you about any changes to your project.

SAVE AND CONTINUE

CANCEL

Make sure you trust App Name

You may be sharing sensitive info with this site or app. [Learn about how App Name will handle your data by reviewing its terms of service and privacy policies.](#) You can always see or remove access in your [Google Account](#). [Learn about the risks](#)

Cancel

Allow

1. Your app logo (if applicable)
2. Your app name, which will be a dynamic link give users your app's support email address

6. Pada bagian Scopes lewati saja, scroll kebawah dan klik **SAVE AND CONTINUE**

Google APIs

Jagowebdev Mail

Search for APIs and Services

APIs & Services

Dashboard
Library
Credentials
OAuth consent screen
Domain verification
Page usage agreements

Edit app registration

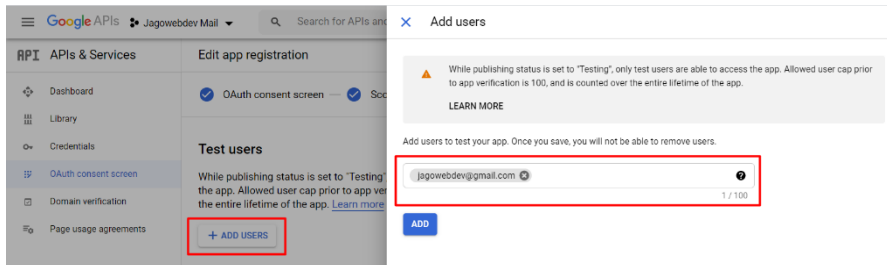
1 OAuth consent screen — 2 Scopes — 3 Test users — 4 Summary

Scopes express the permissions you request users to authorize for your app and allow your project to access specific types of private user data from their Google Account. [Learn more](#)

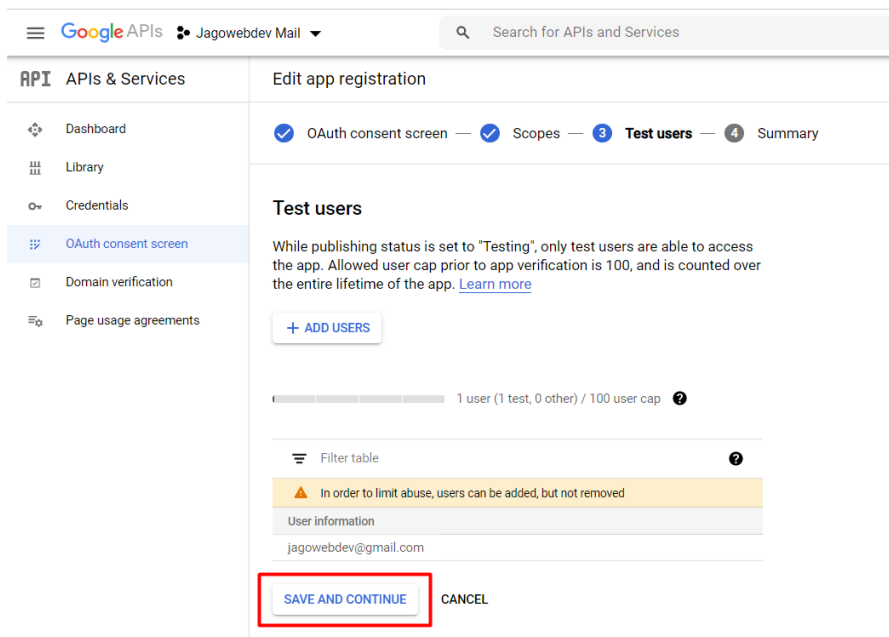
ADD OR REMOVE SCOPES

4 The Gmail Oauth Setup

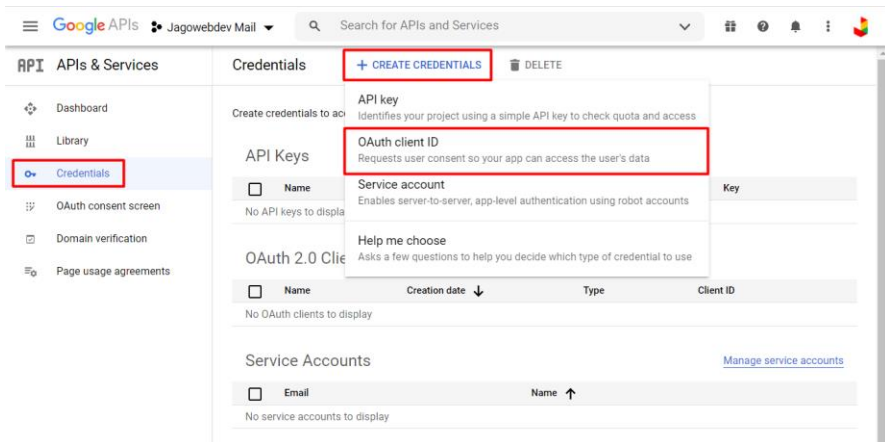
7. Bagian Test users klik tombol + ADD USERS kemudian isikan alamat email yang akan digunakan untuk mengirim email nantinya, setelah selesai klik tombol ADD



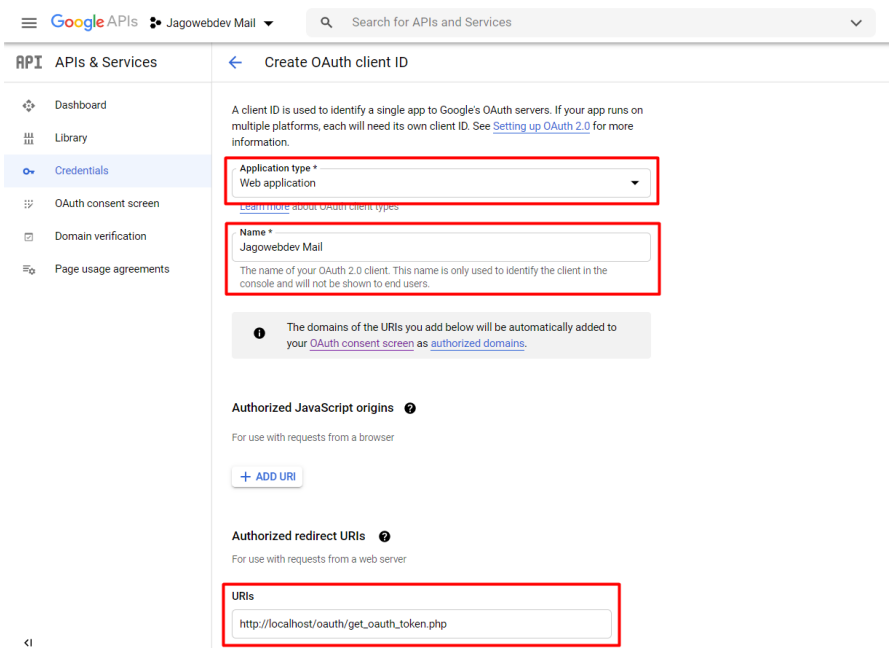
Kembali ke halaman Test users, scroll kebawah dan klik tombol **SAVE AND CONTINUE**



8. Selanjutnya Klik lagi pada menu Credentials yang ada disebelah kiri, kemudian klik menu + Create Credentials yang ada disebelah atas, dan klik menu OAuth client ID

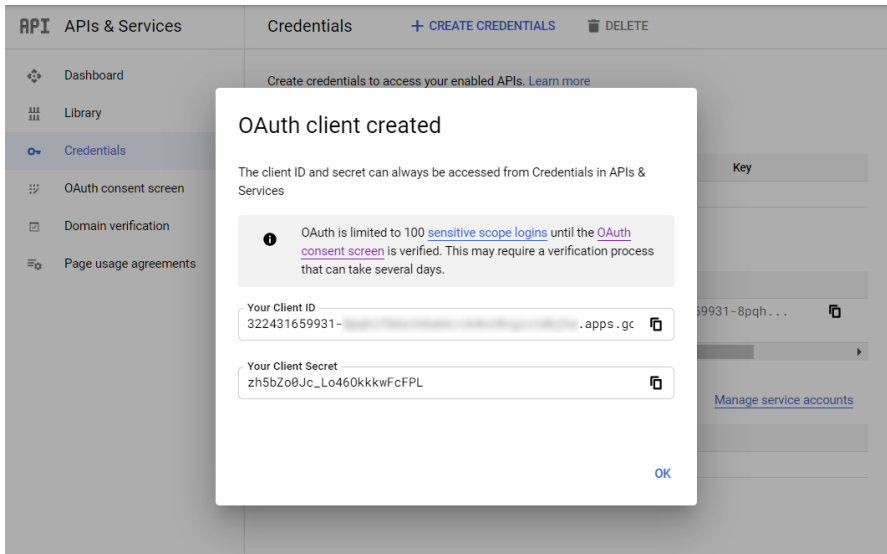


9. Selanjutnya pada menu Create OAuth client ID, bagian Application Type isikan Web application, kemudian bagian name isikan nama OAuth 2.0 Client, selanjutnya pada bagian Authorized redirect URIs yang ada disebelah bawah, isikan alamat URIs, kali ini kita isikan http://localhost/phpmailer/get_oauth_token.php

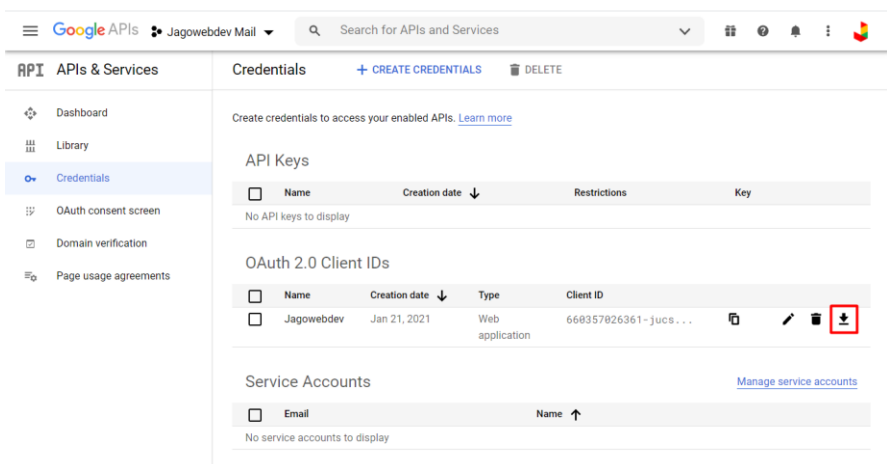


Setelah selesai, klik tombol CREATE yang ada di paling bawah

10. Selanjutnya akan dimunculkan OAuth Client ID dan Client Secret. Anda dapat menyalin nya atau dapat mendownloadnya nanti



Anda juga dapat mendownloadnya di halaman Credentials



II. Membuat refresh_token Key

Untuk membuat refresh token, kita perlu file `get_oauth_token.php` yang disertakan oleh PHPMailer dan library `oauth2-google`. Keduanya ada di file `phpmailer.zip`. Ekstrak file tersebut ke folder `htdocs/phpmailer`. Setelah di ekstrak, struktur file yang kita dapatkan adalah sebagai berikut:

This PC > DATA (D:) > xampp > htdocs > phpmailer >

Name	Type	Size
language	File folder	
src	File folder	
vendor	File folder	
COMMITMENT	File	3 KB
composer.json	JSON File	2 KB
get_oauth_token.php	PHP File	5 KB
LICENSE	File	26 KB
README.md	MD File	15 KB
SECURITY.md	MD File	6 KB
send_email.php	PHP File	2 KB
VERSION	File	1 KB

Pada gambar diatas, library `oauth2-google` ada di folder `vendor`. File `send_email.php` merupakan file php yang saya tambahkan sendiri untuk tes kirim email. Selain folder `vendor` dan file `send_email.php` merupakan file dan folder bawaan PHPMailer

Selanjutnya, untuk membuat refresh token, ikuti langkah berikut:

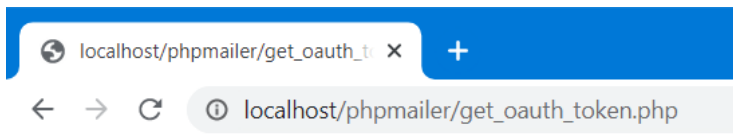
1. Konfigurasi file `get_oauth_token.php`.

Sebelum menjalankan file `get_oauth_token.php`, kita perlu mengatur konfigurasi beberapa parameter yang ada di file tersebut. Buka file tersebut dan cari variabel `$clientId` dan `$clientSecret`. Ganti nilai kedua variabel tersebut dengan nilai `client_id` dan `secret_id` yang kita dapatkan pada proses sebelumnya.

```
$clientId = '98796...';  
$clientSecret = 'j85ZU3yJG_...';
```

2. Jalankan file.

Selanjutnya, buka browser dan jalankan url http://localhost/phpmailer/get_oauth_token.php tampilan yang kita dapatkan adalah sebagai berikut



Select Provider:

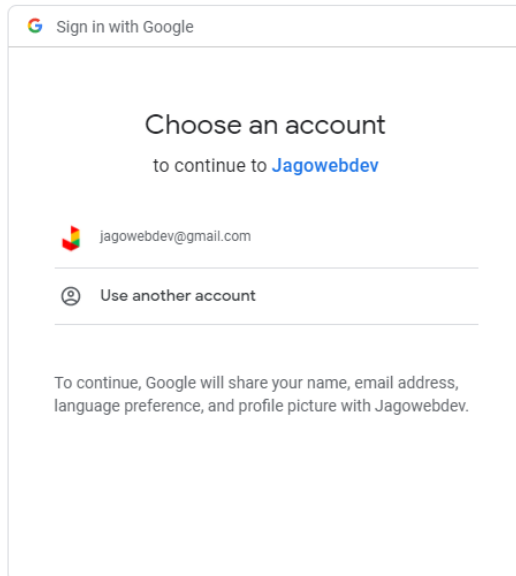
[Google](#) ←

[Yahoo](#)

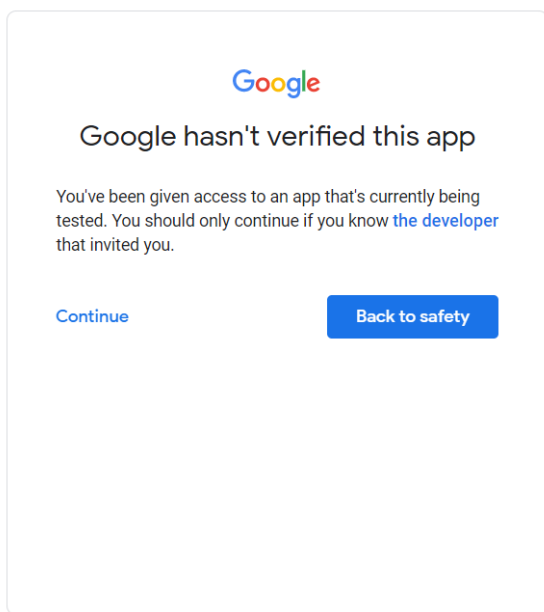
[Microsoft/Outlook/Hotmail/Live/Office365](#)

Klik link Google.

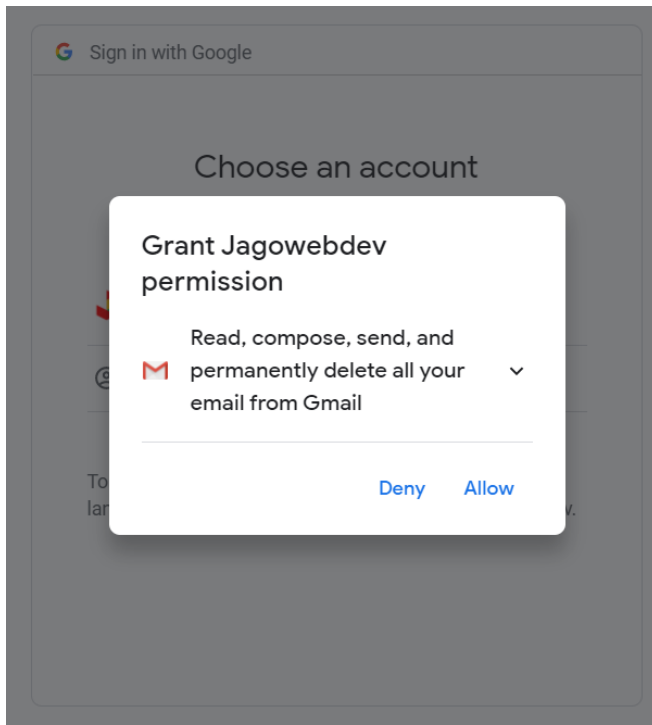
3. Selanjutnya akan terbuka halaman konfirmasi akun google yang ingin dihubungkan.




Selanjutnya muncul peringatan bahwa aplikasi belum diverifikasi oleh Google. Abaikan saja dan klik **Continue**



4. Selanjutnya muncul persetujuan untuk memberikan akses aplikasi ke akun Google kita. Klik Allow



5. Terakhir, terdapat konfirmasi apa saja yang boleh dilakukan oleh aplikasi terhadap akun Google kita. Klik Allow

 Sign in with Google

Confirm your choices

 jagowebdev@gmail.com

You are allowing **Jagowebdev** to:

- ☒ Associate you with your personal info on Google
- ☒ See your personal info, including any personal info you've made publicly available
- ☒ View your email address
- ☒ Read, compose, send, and permanently delete all your email from Gmail

Make sure you trust Jagowebdev

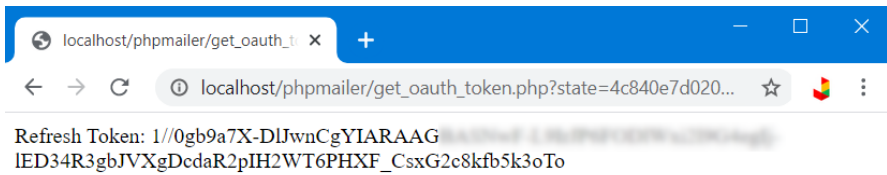
You may be sharing sensitive info with this site or app. Learn about how Jagowebdev will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your [Google Account](#).

[Learn about the risks](#)

[Cancel](#)

[Allow](#)

Selanjutnya akan muncul Refresh Token



Dari gambar diatas, terlihat bahwa refresh tokennya adalah 1//0gb9a7X-DIJwnCgYIARAAG...

Gunakan refresh token ini bersama dengan client_id dan client_secret untuk mengirim email.

III. Tes Kirim Email

Selanjutnya mari kita tes mengirim email dengan client_id, client_secret, dan refresh_token yang telah kita buat.

Buka file send_email.php dan ubah parameter pada script berikut:

```
$provider = new League\OAuth2\Client\Provider\Google([
    'clientId' => '',
    'clientSecret' => ''
]);

$mail->setOAuth( new PHPMailer\PHPMailer\OAuth (
    [
        'provider' => $provider,
        'clientId' => '',
        'clientSecret' => '',
        'refreshToken' => '',
        'userName' => '',
    ]
)
);

$mail->CharSet = 'utf-8';
$mail->setFrom('', 'Tes PHP Mailer');
$mail->addAddress('', 'Nama Tujuan');
$mail->Subject = 'Tes Email';
$mail->msgHTML('Tes kirim email menggunakan PHPMailer dan OAuth2 Gmail');
```

Isikan dengan parameter sebagai berikut:

1. `clientId`, `clientSecret`, dan `refreshToken` sesuai yang telah kita dapatkan tadi.
2. `userName`: alamat email yang telah kita daftarkan pada Test users, pada contoh ini jagowebdev@gmail.com
3. `setFrom`: isikan dengan alamat email yang telah kita daftarkan pada Test users, pada contoh ini jagowebdev@gmail.com
4. `addAddress`: isikan alamat email tujuan.

Setelah selesai, simpan file tersebut dan jalankan pada browser, jika email berhasil dikirim, maka akan muncul pesan Email berhasil dikirim