Mengirim Email Menggunakan Gmail SMTP
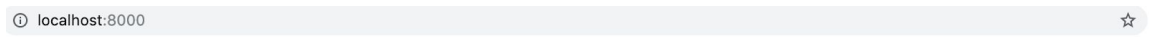
Tujuan

Bagimana supaya aplikasi bisa mengirim email?
Pengiriman email ini berguna sekalai untuk melakukan konfirmasi, misalkan setelah melakukan pendaftaran user akan menerima email untuk melakukan konfirmasi pendaftaran. Contoh lain adalh ketika kita melakukan 'Lupa Password', maka aplikasi akan mengirim email untuk me-reset password kita.
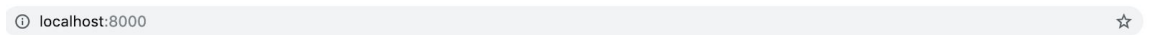
Kita akan membuat aplikasi yang bisa mngirim email.
1. User akan memasukan alamat email nya

2. Aplikasi melakukan validasi email required

3. Kalau user memasukan email yang valid, maka aplikasi akan mengirim email
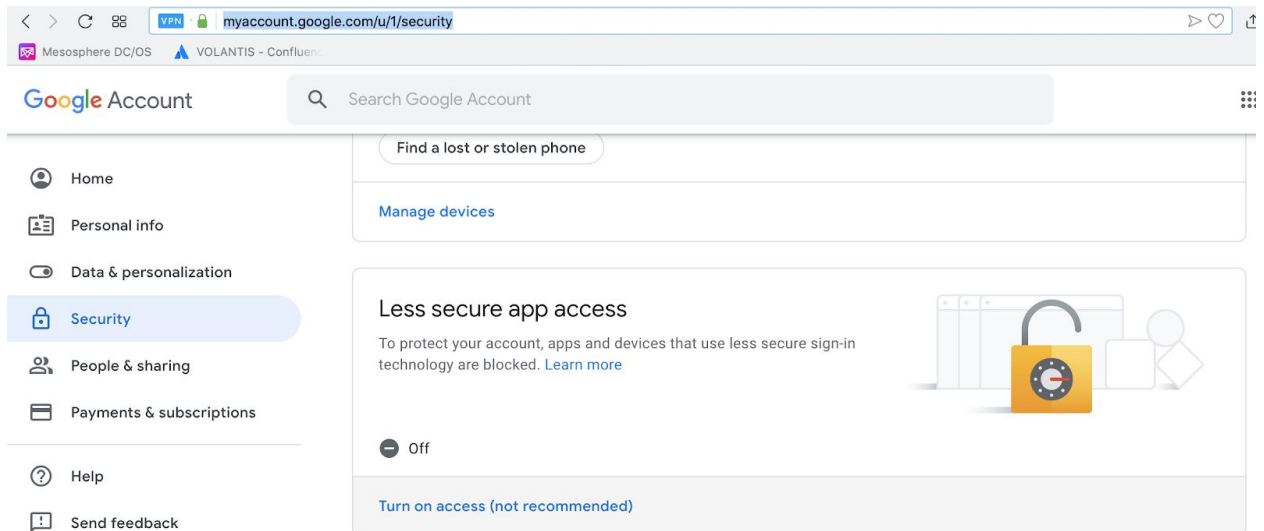

4. Finish

Setting Gmail

Supaya kita bisa mengirim email menggunakan Gmail melalui aplikasi laravel, maka kita perlu melakukan settingan pada akun google kita.Ikuti langkah-langah dibawah ini.

1. https://myaccount.google.com/security#connectedapps
2. **Sign-in & security -> Connected apps & sites -> Allow less secure apps** settings.

3. "Allow less secure apps" **ON**.
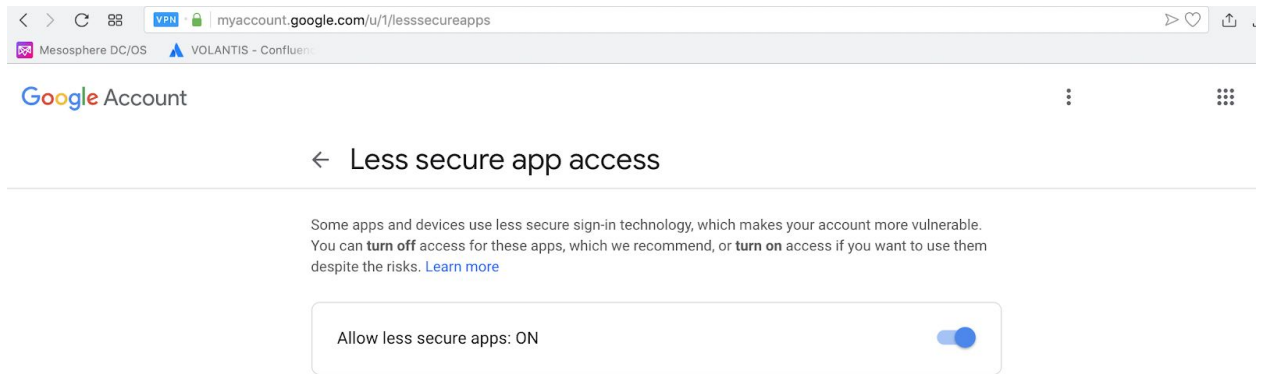
4. Finish

Unlock Gmail Captcha

Tahap ini adalah lanjutan dari setting Gmail.

1. https://www.google.com/accounts/DisplayUnlockCaptcha
2. Klik Continue
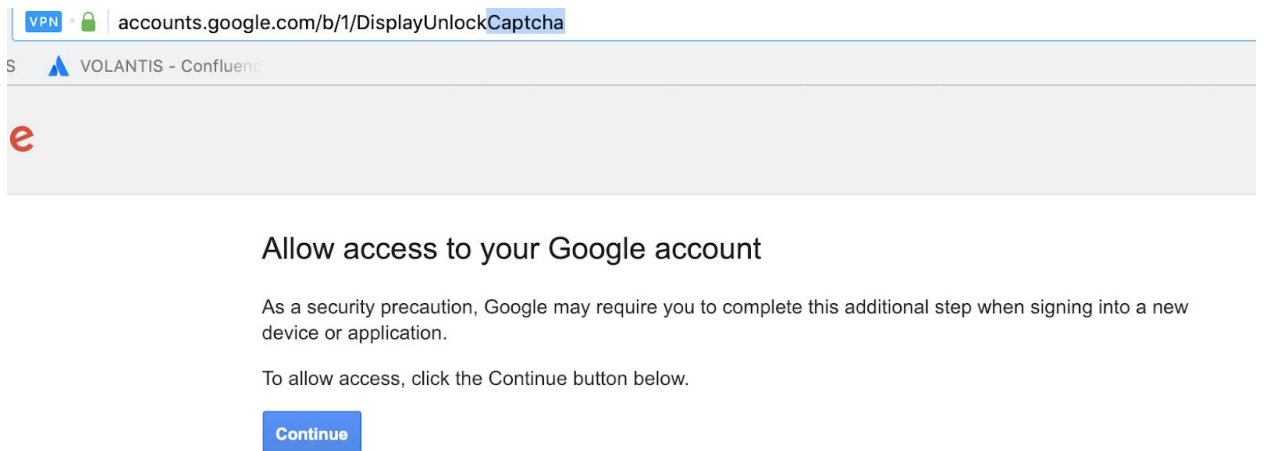
3. Konfirmasi bahwa akun gmail kita telah di unlcok/enable
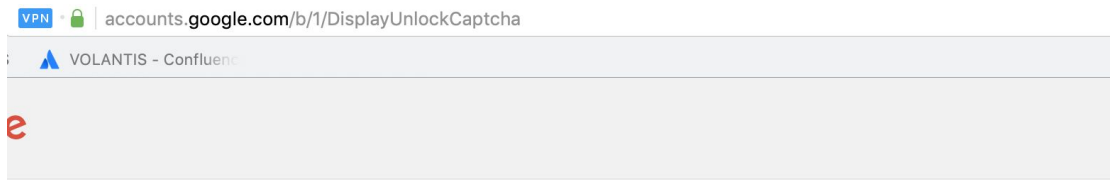
4. Finish

Setting Basic Laravel App

1. Install laravel
   composer global require laravel/installer

2. Membuat project laravel
   composer create-project --prefer-dist laravel/laravel laravel_mailer

3. Generate artisan key
   php artisan key:generate

4. Setting Databse pada file .env
   DB_CONNECTION=mysql
   DB_HOST=127.0.0.1
   DB_PORT=3306
   DB_DATABASE=homestead
   DB_USERNAME=homestead
   DB_PASSWORD=secret

5. Edit file resources/views/welcome.php. SIlahkan anda berkreasi untuk meuat
   halaman web yang lebih bagus, yang perlu diperhatikan ada pada:
   - Line 18 - 26
   - Line 28 - 38
   - Line 40 - 47

```html
1   <!doctype html>
2   <html lang="{{ app()->getLocale() }}">
3       <head>
4           <meta charset="utf-8">
5           <meta http-equiv="X-UA-Compatible" content="IE=edge">
6           <meta name="viewport" content="width=device-width, initial-scale=1">
7           <title>Laravel</title>
8           <!-- Bootstrap core CSS -->
9           <link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
10          <!-- Fonts -->
11          <link href="https://fonts.googleapis.com/css?family=Raleway:100,600" rel="stylesheet" type="text/css">
12      </head>
13      <body>
14          <div class="container" style="margin-top: 100px; width: 600px;">
15              <h2 class="text-center">Please Type Your Email Address</h2>
16              <h5 class="text-center">Demo App: How To Send Email With Gmail</h5><br><br>
17
18              <!-- Code to show Success Message -->
19              @if(session('success'))
20                  <div class="alert alert-success alert-dismissible fade show" role="alert">
21                      {{session('success')}}
22                      <button type="button" class="close" data-dismiss="alert" aria-label="Close">
23                      <span aria-hidden="true">&times;</span>
24                      </button>
25                  </div>
26              @endif
27
28              <!-- Code to show Error Message -->
29              @if(count($errors)>0)
30                  @foreach($errors->all() as $error)
31                      <div class="alert alert-danger alert-dismissible fade show" role="alert">
32                          {{$error}}
33                          <button type="button" class="close" data-dismiss="alert" aria-label="Close">
34                          <span aria-hidden="true">&times;</span>
35                          </button>
36                      </div>
37                  @endforeach
38              @endif
39
40              <!-- FOrm to input email address -->
41              <form action="hello/send-welcome-email" method="POST">
42                  {{ csrf_field() }}
43                  <div class="form-group">
44                      <input type="email" class="form-control text-center" placeholder="Provide an email" name="email"><br>
45                      <button type="submit" class="btn btn-primary btn-md" style="margin-left: 0px;margin: auto; display: block;
                        "t>SEND</button>
46                  </div>
47              </form>
48          </div>
49      <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
50      <!-- Bootstrap core JavaScript -->
51      <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0/js/bootstrap.min.js"></
        script>
52      </body>
53  </html>
54
```

6. Finish

Add Mailer Functionality

1. Edit mailer setting pada file .env
   MAIL_DRIVER=smtp
   MAIL_HOST=smtp.gmail.com
   MAIL_PORT=587
   MAIL_USERNAME=YOUR_GMAIL_EMAIL
   MAIL_PASSWORD=YOUR_GMAIL_PASSWORD
   MAIL_ENCRYPTION=tls
2. Edit file config/mail.php pada line 58-61 (number line nya bisa berbeda).

   'from' => [
        'address' => env('MAIL_FROM_ADDRESS',
   PLEASE_EDIT_ME@example.com'),
        'name' => env('MAIL_FROM_NAME', 'PLEASE_EDIT_ME'),
      ],

3. Buat class mailer di app/Mailer/HelloMailer.php

```php
<?php

namespace App\Mailer;

use Illuminate\Bus\Queueable;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;
use Illuminate\Contracts\Queue\ShouldQueue;

class HelloMailer extends Mailable
{
    use Queueable, SerializesModels;

    /**
     * Create a new message instance.
     *
     * @return void
     */
    public function __construct()
    {
        //
    }

    /**
     * Build the message.
     *
     * @return $this
     */
    public function build()
    {
        # this code will use template in resource/views/mailer/hello_mailer.blade.php
        # this code will send email with subject 'Hello Mailer'
        return $this->view('mailer/hello_mailer')->subject('Hello Mailer');
    }
}
```

4. Buat view untuk template email di resources/views/mailer/hello_mailer.blade.php.
   Silahkan anda berkreasi untuk membuat template email yang lebih bagus.

```html
<!DOCTYPE html>
<html>
<head>
  <title>Hello Mailer</title>
</head>
<body>
  <p>This email sent by laravel application with gmail SMPTP.</p>
</body>
</html>
```

5. Tambahkan routing untuk menerima input email dari user

```php
<?php

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', function () {
    return view('welcome');
});

// Create routing for send email
Route::post('hello/send-welcome-email', 'HelloController@sendWelcomeEmail');
```

6. Buat controller baru pada app/Http/Controllers/HelloController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Input;
use Illuminate\Support\Facades\Mail; // Laravel mailer library
use App\Mailer\HelloMailer; // Our mailer

class HelloController extends Controller
{
    public function sendWelcomeEmail(Request $request)
    {
        // Validate email address
        $this->validate($request, [
            'email' => 'required'
        ]);

        // This code will trigger HelloMailer class to send email
        Mail::to($request->input('email'))->send(new HelloMailer());

        // Redirect and send message about status sending email request
        return redirect()->back()->with('success', 'Email has been sent successfully. Please check your email.'
        );
    }
}
```

7. Silahkan anda jalankan aplikasi anda, dan test untuk mengirim email

Reference

1. https://laravel.com/docs/5.8/mail
2. https://learninglaravel.net/learn-to-send-emails-using-gmail-and-sendgrid-in-laravel-5
3. https://1netwiki.com/wiki/45
4. https://learninglaravel.net/learn-to-send-emails-using-gmail-and-sendgrid-in-laravel-5