

## Penjelasan SMTP Mailtrap

### 1. Konfigurasi SMTP di .env

```
51 MAIL_MAILER=smtp
52 MAIL_HOST=sandbox.smtp.mailtrap.io
53 MAIL_PORT=587
54 MAIL_USERNAME=242c86defa1f58
55 MAIL_PASSWORD=c5889c83f52890
56 MAIL_ENCRYPTION=tls
57 MAIL_FROM_ADDRESS="example@gmail.com"
58 MAIL_FROM_NAME="${APP_NAME}"
```

#### Penjelasan:

- **MAIL\_MAILER=smtp**: Menunjukkan bahwa Laravel akan menggunakan protokol **SMTP** untuk pengiriman email.
- **MAIL\_HOST=sandbox.smtp.mailtrap.io**: Alamat server SMTP Mailtrap. Mailtrap digunakan untuk pengujian email dalam lingkungan pengembangan dan memastikan email tidak terkirim ke penerima sebenarnya.
- **MAIL\_PORT=587**: Port yang digunakan untuk koneksi SMTP. Port 587 adalah port standar untuk pengiriman email yang aman menggunakan enkripsi TLS.
- **MAIL\_USERNAME** dan **MAIL\_PASSWORD**: Kredensial yang digunakan untuk otentikasi dengan server SMTP Mailtrap.
- **MAIL\_ENCRYPTION=tls**: Menentukan enkripsi yang digunakan untuk mengamankan koneksi SMTP. TLS (Transport Layer Security) adalah protokol yang umum digunakan.
- **MAIL\_FROM\_ADDRESS**: Alamat email pengirim yang digunakan dalam email.
- **MAIL\_FROM\_NAME="\${APP\_NAME}"**: Nama pengirim yang digunakan dalam email, yang diambil dari konfigurasi aplikasi Laravel (biasanya diatur di file .env).

### 2. Kelas `Mailable` untuk Pengiriman Email

```
3. <?php
4.
5. namespace App\Mail;
6.
7. use Illuminate\Bus\Queueable;
8. use Illuminate\Contracts\Queue\ShouldQueue;
9. use Illuminate\Mail\Mailable;
10. use Illuminate\Mail\Mailables\Content;
11. use Illuminate\Mail\Mailables\Envelope;
12. use Illuminate\Queue\SerializesModels;
13.
14. class TagihanPenggantian extends Mailable
15. {
16.     use Queueable, SerializesModels;
17.
18.     public $tagihan;
19.
20.     /**
21.      * Create a new message instance.
22.      */
23.     public function __construct($tagihan)
24.     {
```

```

25.     $this->tagihan = $tagihan;
26. }
27.
28. public function build()
29. {
30.     return $this->view('emails.tagihan_penggantian')
31.         ->subject('Tagihan Penggantian')
32.         ->with([
33.             'tagihan' => $this->tagihan
34.         ]);
35. }
36.
37. /**
38.  * Get the message envelope.
39.  */
40. public function envelope(): Envelope
41. {
42.     return new Envelope(
43.         subject: 'Tagihan Penggantian',
44.     );
45. }
46.
47. /**
48.  * Get the message content definition.
49.  */
50. public function content(): Content
51. {
52.     return new Content(
53.         view: 'emails.tagihan_penggantian',
54.         with: [
55.             'tagihan' => $this->tagihan
56.         ],
57.     );
58. }
59.
60. /**
61.  * Get the attachments for the message.
62.  *
63.  * @return array<int, \Illuminate\Mail\Mailables\Attachment>
64.  */
65. public function attachments(): array
66. {
67.     return [];
68. }
69. }
70.

```

**Penjelasan:**

- **\$tagihan**: Properti ini menyimpan data tagihan yang akan diteruskan ke email. Data ini dapat berupa objek atau array yang berisi informasi tagihan, seperti ID, nama, jumlah, dan lainnya.
- **\_\_construct(\$tagihan)**: Konstruktor kelas ini menerima parameter **\$tagihan** yang kemudian disimpan dalam properti kelas untuk digunakan dalam konten email.
- **build()**: Method ini bertanggung jawab untuk membangun email. Di dalamnya:
  - **\$this->view('emails.tagihan\_penggantian')**: Menentukan tampilan atau view yang digunakan untuk email. View ini adalah template HTML untuk email yang akan dikirim.
  - **subject('Tagihan Penggantian')**: Menetapkan subjek email.
  - **with(['tagihan' => \$this->tagihan])**: Mengirimkan data **\$tagihan** ke view agar dapat digunakan di dalam template.
- **envelope()**: Metode ini mendefinisikan envelope email, seperti subjek. Ini adalah alternatif cara Laravel menangani subjek di dalam **build()** method.
- **content()**: Mendefinisikan konten email dengan view yang digunakan, serta data yang diteruskan ke view.
- **attachments()**: Method ini mengatur lampiran email. Dalam kasus ini, tidak ada lampiran yang ditambahkan, sehingga mengembalikan array kosong.