

Nota Pelajar 19: Integrasi Gerbang Pembayaran Bahagian 1

Objektif

- Faham cara kerja sistem pembayaran moden (Stripe).
- Setup Stripe Sandbox environment.
- Integrasi Laravel Cashier dengan Stripe API.
- Jalankan pembayaran test dalam projek Laravel.
- Faham konsep webhook & callback.

Apa Itu Stripe?

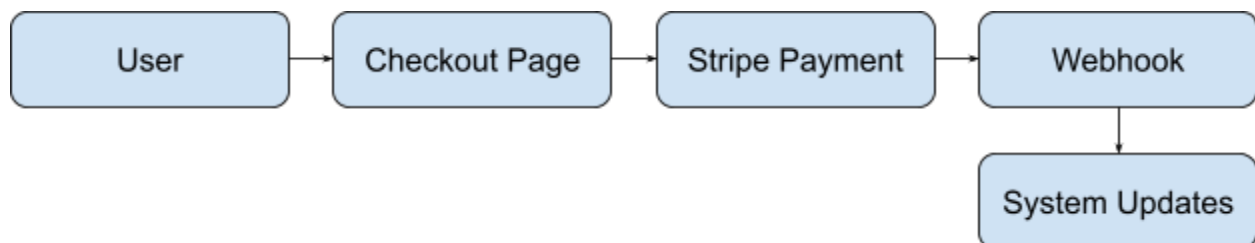
Stripe ialah **gerbang pembayaran (payment gateway)** yang membolehkan sistem menerima bayaran secara online menggunakan kad kredit/debit, FPX, e-wallet, dan lain-lain.

Kelebihan Stripe:

- Mudah diintegrasikan dengan Laravel
- Ada Dashboard realtime untuk pantau transaksi
- Sokong banyak kaedah pembayaran
- Ada **test mode (sandbox)** untuk developer

Bagaimana Stripe Berfungsi

Flow asas pembayaran:



Isitilah Penting:

- API Key → pengenalan aplikasi kamu.
- Session ID → data checkout sementara.
- Webhook → notifikasi bila bayaran berjaya/gagal.

Apa Itu Laravel Cashier?

Laravel Cashier ialah **package rasmi Laravel** untuk mengurus pembayaran dengan Stripe (atau Paddle).

Ia bantu kita handle perkara rumit macam:

- Langganan (subscriptions)
- Satu kali bayaran (one-time payment)
- Invois & Resit
- Webhook event (auto update status bila bayar berjaya)

Langkah 1: Setup Akaun Stripe

- 1 Pergi ke <https://dashboard.stripe.com/register>
- 2 Daftar akaun baru (guna email dev)
- 3 Masuk ke dashboard → *Developer Mode (Test Mode)*
- 4 Salin **Publishable Key** dan **Secret Key**

Contoh:

STRIPE_KEY=pk_test_12345...

STRIPE_SECRET=sk_test_67890...

Langkah 2: Pasang Laravel Cashier

Dalam projek Laravel:

```
composer require laravel/cashier
```

Setup model `User`:

```
<?php

use Laravel\Cashier\Billable;

Class User extends Model
{
    use Billable;
}
```

Jalankan migration:

```
php artisan migrate
```

Cashier akan cipta table **subscriptions**, **charges**, dll.

Langkah 3: Tambah Stripe Keys dalam .env

Buka **.env** file dan masukkan:

```
STRIPE_KEY=pk_test_XXXXX
STRIPE_SECRET=sk_test_XXXXX
STRIPE_WEBHOOK_SECRET=
```

Buka **config/services.php**:

```
'stripe' => [
    'secret' => env('STRIPE_SECRET'),
],
```

Langkah 4: Buat Akaun Stripe Test Customer

Stripe bagi kita **kad palsu (test card)** untuk simulate transaksi.



TAMING TECH SDN. BHD.

321A, Tingkat 1, Lorong Selangor, Taman Melawati,

53100 Kuala Lumpur.

Tel: +60 13-446 4601 email: hq@tamingtech.my

Contoh kad test:

Card Number: 4242 4242 4242 4242

Exp: 12/34

CVC: 123

Langkah 5: Setup Stripe CLI (Optional tapi Power)

Stripe CLI ni penting untuk **listen webhook event secara local**.

Install Stripe CLI (homebrew):

```
brew install stripe/stripe-cli/stripe
```

Install Stripe CLI (windows)

1. Download the latest windows zip file from GitHub
2. Unzip the `stripe\_X.X.X\_windows\_x86\_64.zip` file.
3. Add the path to the unzipped `stripe.exe` file to your `Path` environment variable.

Login:

```
stripe login
```

Connect ke projek:

```
stripe listen --forward-to http://localhost:8000/stripe/webhook
```

Langkah 6: Buat Payment Test Basic

Cuba buat route test untuk pastikan connection berfungsi

```
Route::get('/checkout', function (Request $request) {
    $stripePriceId = 'price_deluxe_album';
    $quantity = 1;

    return $request->user()->checkout([$stripePriceId => $quantity], [
        'success_url' => route('checkout-success') .
        'session_id={CHECKOUT_SESSION_ID}',
        'cancel_url' => route('checkout-cancel'),
    ]);
});

Route::view('/checkout/success',
'checkout.success')->name('checkout-success');

Route::view('/checkout/cancel',
'checkout.cancel')->name('checkout-cancel');
```

Kalau response keluar JSON dari Stripe → setup dah berjaya

Urus Webhook & Callback

1. Buat route untuk webhook:

```
Route::post('/stripe/webhook', [StripeWebhookController::class, 'handle']);
```

2. Log event dalam controller:

```
Public function handle(Request $request) {
    Log::info($request->all());
}
```

Bila payment berjaya, Stripe akan ping endpoint ini.

Langkah 7: Pantau Dashboard Stripe

- Pergi ke **Stripe Dashboard** → **Payments**
- Tengok transaction yang baru dibuat
- Lihat status (Succeeded / Pending)



TAMING TECH SDN. BHD.

321A, Tingkat 1, Lorong Selangor, Taman Melawati,

53100 Kuala Lumpur.

Tel: +60 13-446 4601 email: hq@tamingtech.my

Tips Developer

Stripe Key Best Practices

- Jangan kongsi `sk_test` key
- Simpan semua key dalam `.env`
- Untuk production → guna `pk_live` & `sk_live`

Webhook Tip

- Gunakan `stripe listen` semasa testing
- Simpan `WEBHOOK_SECRET` dalam `.env`
- Validate event signature sebelum update database