

NAMA : IBRAHIM
NIM : G.231.23.0134

Tugas menjelaskan step by steps dalam membuat data bases laravel anda menyalam xampp dan aktifkan apache dan mysql lalu anda bisa membuka cmd lalu mengetik

#composer create-project laravel/laravel myapp

Jika langkah berhasil anda bisa mengetik di cmd berupa

#cd myapp

Lalu jika sudah hingga steps ni anda bisa meinta server pada laravel dengan mengetik

#php artisan serve

Lalu anda akan diberikan server dengan seperti ini

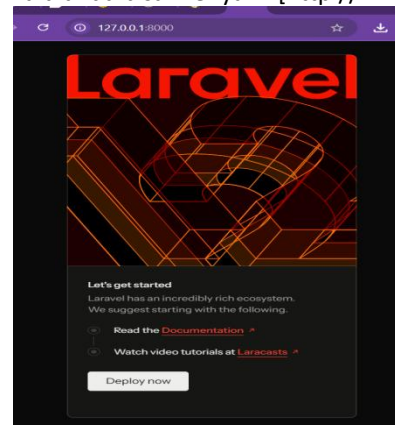
```
C:\xampp\htdocs\baibmm>php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

2025-04-27 12:53:50 / ..... ~ 514.28ms
2025-04-27 12:53:50 /favicon.ico ..... ~ 1s
```

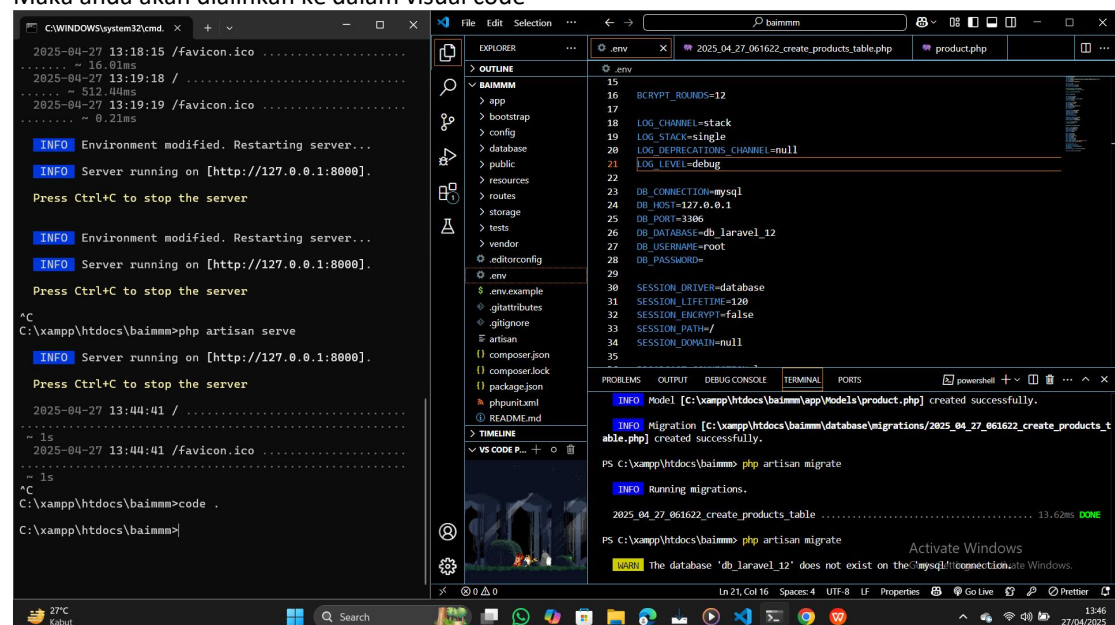
Lalu anda bisa menyalin [http://127.0.0.1:8000] pada website



Tampilan diatas menjelaskan anda berhasil dalam menjankan server pada laravel, langkah selanjutnya adalah anda mengetik pada cmd berupa

#code .

Maka anda akan dialihkan ke dalam visual code

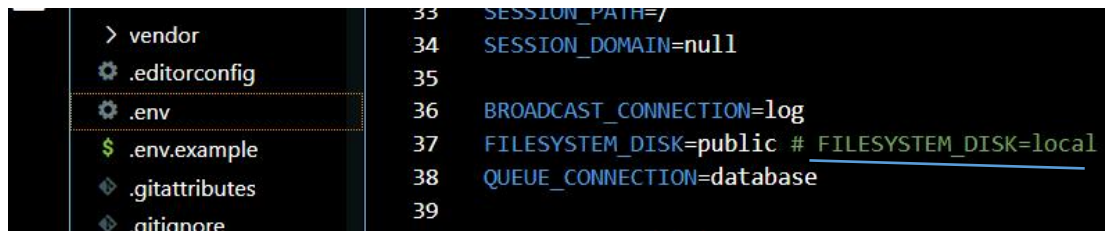


Gambar diatas menjelaskan bahwa anda telah berhasil memasuki visual code dengan cmd. Lalu langkah untuk mebuat data bases adalah

2. Konfigurasi File System

Langkah 1 - Konfigurasi File .env

Pertama, kita akan melakukan konfigurasi di dalam file .env, dimana kita akan mengubah driver yang digunakan oleh Laravel dalam menyimpan file yang diunggah. Silahkan buka file .env, kemudian cari kode berikut ini.



```
33 SESSION_PATH=/
34 SESSION_DOMAIN=null
35
36 BROADCAST_CONNECTION=log
37 FILESYTEM_DISK=public # FILESYSTEM_DISK=local
38 QUEUE_CONNECTION=database
39
```

Anda harus mengubah **FILESYTEM_DISK=local** menjadi **FILESYTEM_DISK=public**

Dari perubahan kode di atas, kita mengubah default driver yang digunakan oleh file system yang semula adalah local menjadi public.

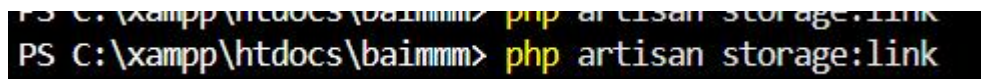
Di atas, jika kita menggunakan driver local, maka file yang diunggah akan masuk ke dalam folder storage/app/private, sehingga kita tidak bisa langsung mengaksesnya. Oleh karena itu, kita mengubahnya ke driver public, agar file yang diunggah disimpan di storage/app/public, sehingga dapat diakses dengan mudah.

Langkah 2 - Menjalankan Storage Link

Laravel hanya bisa mengakses file yang ada di dalam folder public. Sementara itu, file yang diunggah tersimpan di dalam folder storage. Lalu, bagaimana agar file-file tersebut bisa diakses melalui folder public?

Caranya adalah dengan membuat symlink (symbolic link) atau menghubungkan folder storage/app/public ke dalam folder public yang ada di root project.

Laravel sudah menyediakan perintah khusus untuk melakukan proses linking ini, yaitu storage:link. Jadi sekarang silahkan teman-teman jalankan perintah berikut ini di dalam terminal/CMD dan pastikan berada di dalam project Laravel-nya.



```
PS C:\xampp\htdocs\baimmm> php artisan storage:link
PS C:\xampp\htdocs\baimmm> php artisan storage:link
```

3. Membuat Model dan Migration

Apa Itu Model dan Migration?

1. Model Model dalam Laravel adalah representasi dari tabel di database. Dengan model, kita bisa berinteraksi dengan data menggunakan sintaks yang lebih sederhana dibandingkan dengan menulis query SQL secara langsung. Model ini berperan sebagai perantara antara aplikasi dan database.

2. Migration Migration adalah fitur Laravel yang memungkinkan kita untuk mengelola struktur database menggunakan kode. Dengan migration, kita dapat membuat, mengubah, atau menghapus tabel tanpa harus menulis query SQL secara manual. Migration juga membantu dalam menjaga konsistensi struktur database saat bekerja dalam tim.

Langkah 1 - Konfigurasi Koneksi Database

```
21 LOG_LEVEL=debug
22
23 DB_CONNECTION=mysql #DB_CONNECTION=sqlite
24 DB_HOST=127.0.0.1
25 DB_PORT=3306
26 DB_DATABASE=db_laravel_12
27 DB_USERNAME=root
28 DB_PASSWORD=
```

Lalu anda bisa melanjutkan dengan memasukan ke dalam terminal berupa

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\xampp\htdocs\baimmm> php artisan config:clear

INFO Configuration cache cleared successfully.

PS C:\xampp\htdocs\baimmm> 
```

Jika terdapat info seperti yang diatas maka anda berhasil melakukan config

Langkah 2 - membuat model dan migration

Di Laravel, kita bisa membuat Model dan Migration sekaligus dalam satu perintah artisan. Hal ini bertujuan untuk menghemat waktu saat mengembangkan aplikasi. Silakan jalankan perintah berikut di terminal/CMD, pastikan berada di dalam folder project Laravel-nya:

```
PS C:\xampp\htdocs\baimmm> php artisan make:model product -m

ERROR Model already exists.

PS C:\xampp\htdocs\baimmm> 
```

Gambar diatas menjelaskan bahwa saya telah meminta model product sebelumnya jika anda pertama kali maka anda dijelaskan bahwa anda sukses dalam meminta model product

1. app/Models/Product.php
2. database/migrations/2025_02_25_080603_create_products_table.php

Setelah Model dan Migration berhasil dibuat, langkah berikutnya adalah mengedit file Migration untuk mendefinisikan struktur tabel products.

Langkah 3 - Menambahkan field/Kolom di Migration

Sekarang kita akan lanjutkan menambahkan sebuah field-field atau kolom di dalam file migration. Silahkan teman-teman buka file berikut ini database/migrations/, kemudian pada function up ubah kode-nya menjadi seperti berikut ini.

Database/migration/2025/2024_24_04_080603_create_product_table.php

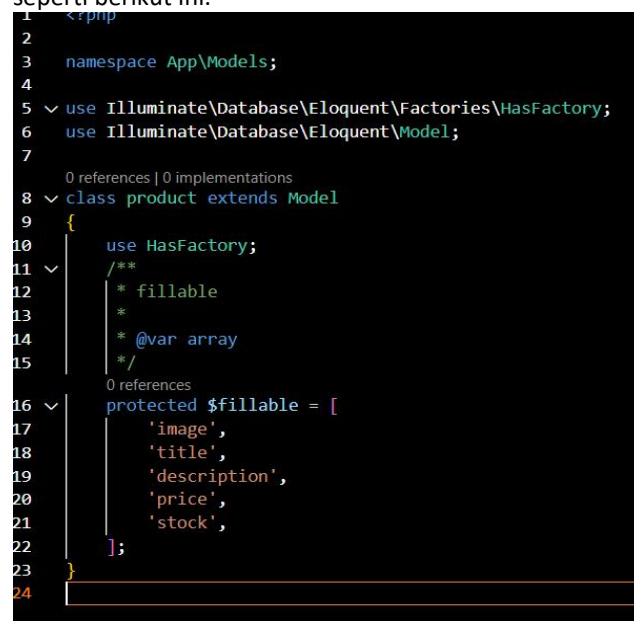
```
public function up(): void
{
    Schema::create('products', function (Blueprint $table) {
        $table->id();
        $table->string('image');
        $table->string('title');
        $table->text('description');
        $table->bigInteger('price');
        $table->integer('stock')->default(0);
        $table->timestamps();
    });
}
```

Langkah 4 - konfigurasi Mass Assignment

Mass Assignment di Laravel adalah fitur yang memungkinkan kita untuk mengisi beberapa kolom database sekaligus menggunakan array data. Ini sering digunakan saat menyimpan data dari sebuah form ke database melalui model Eloquent.

Laravel melindungi Model dari mass assignment secara default untuk mencegah serangan Mass Assignment Vulnerability, di mana pengguna dapat mengisi kolom yang tidak diizinkan, seperti role pengguna atau hak akses.

Silahkan buka file app/Models/Product.php, kemudian ubah semua kode-nya menjadi seperti berikut ini.



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Product extends Model
9 {
10     use HasFactory;
11     /**
12      * fillable
13      *
14      * @var array
15      */
16     protected $fillable = [
17         'image',
18         'title',
19         'description',
20         'price',
21         'stock',
22     ];
23 }
24
```

Dari perubahan kode di atas, kita tambahkan properti \$fillable. Properti ini digunakan untuk menentukan field/kolom mana saja yang boleh diisi.

Langkah 5 - Menjalankan Migration

Setelah berhasil menambahkan field atau kolom di dalam file Migration dan menambahkan Mass Assignment di dalam Model, maka kita akan lanjutkan belajar menjalankan proses migrate.

Dengan menjalankan proses migrate, maka field/kolom yang sudah kita definisikan di dalam file migration akan digenerate ke dalam database.

Silahkan teman-teman jalankan perintah berikut ini di dalam terminal/CMD dan pastikan berada di dalam project Laravel-nya.

```
c:\xampp\htdocs\baimmm>php artisan migrate

INFO Nothing to migrate.
```

```
PS C:\xampp\htdocs\myapps> php artisan migrate

WARN The database 'db_laravel_12' does not exist on the 'mysql' connection.

Would you like to create it? (yes/no) [yes]
> |
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + - - - - -

Would you like to create it? (yes/no) [yes]
> yes

INFO Preparing database.

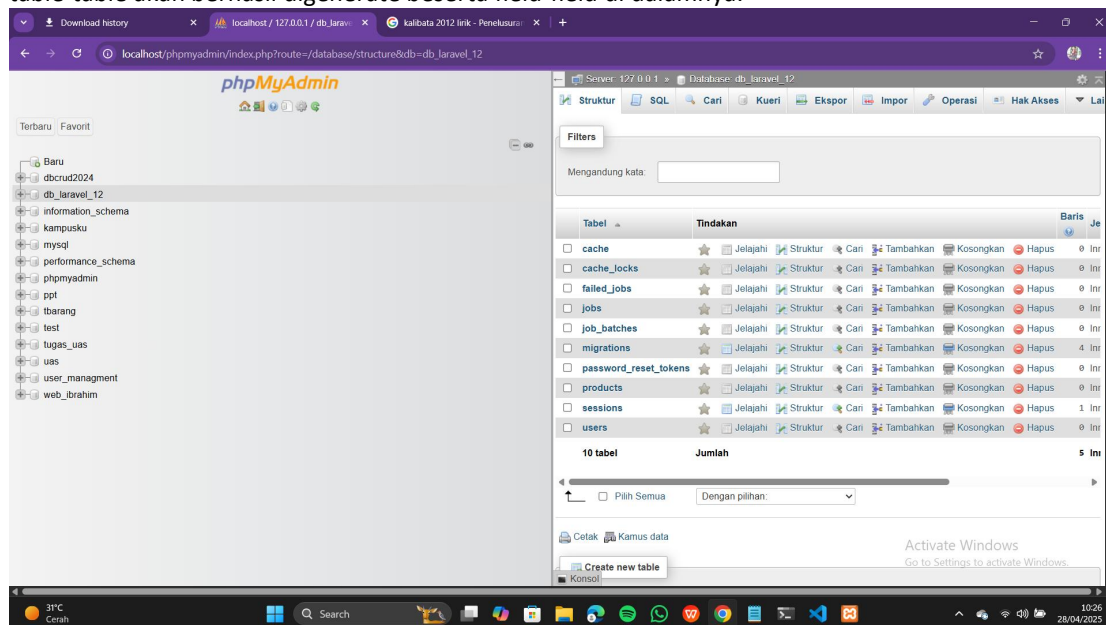
creating migration table ..... 58.53ms DONE

INFO Running migrations.

0001_01_01_000000_create_users_table ..... 110.89ms DONE
0001_01_01_000001_create_cache_table ..... 25.32ms DONE
0001_01_01_000002_create_jobs_table ..... 139.32ms DONE
2025_04_27_042027_create_products_table ..... 13.03ms DONE

PS C:\xampp\htdocs\myapps> |
```

Dan jika teman-teman cek pada PhpMyAdmin di <http://localhost/phpmyadmin>, maka table-table akan berhasil digenerate beserta field-field di dalamnya.



Jika hasilnya berhasil maka hasil yang ditampilkan dala, php myadmin seperti pada gambar diatas