

Laravel 9

Bab 2 - Membuat dan Menjalankan Migration



TUTORIAL

LARAVEL 9

Membuat & Menjalankan Migration

TUTORIAL LARAVEL 9 #2 : MEMBUAT DAN MENJALANKAN MIGRATION

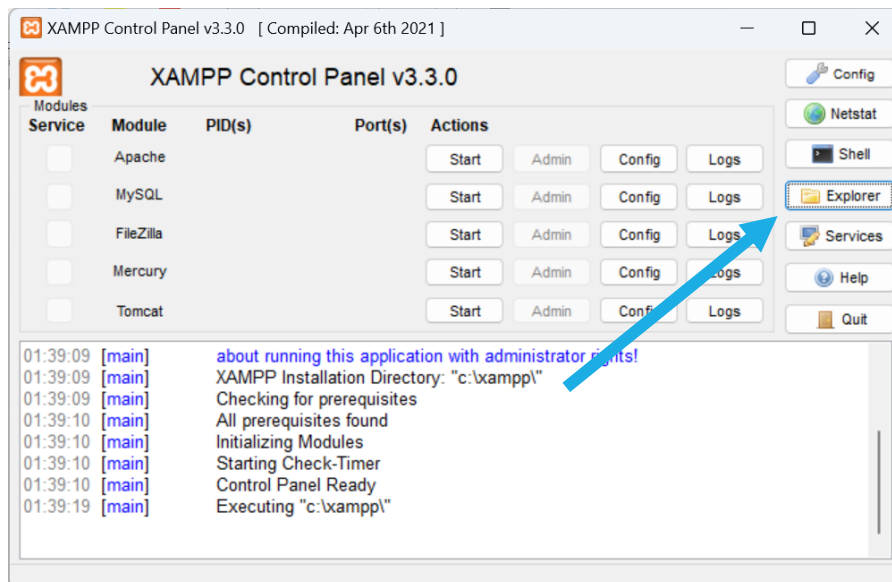
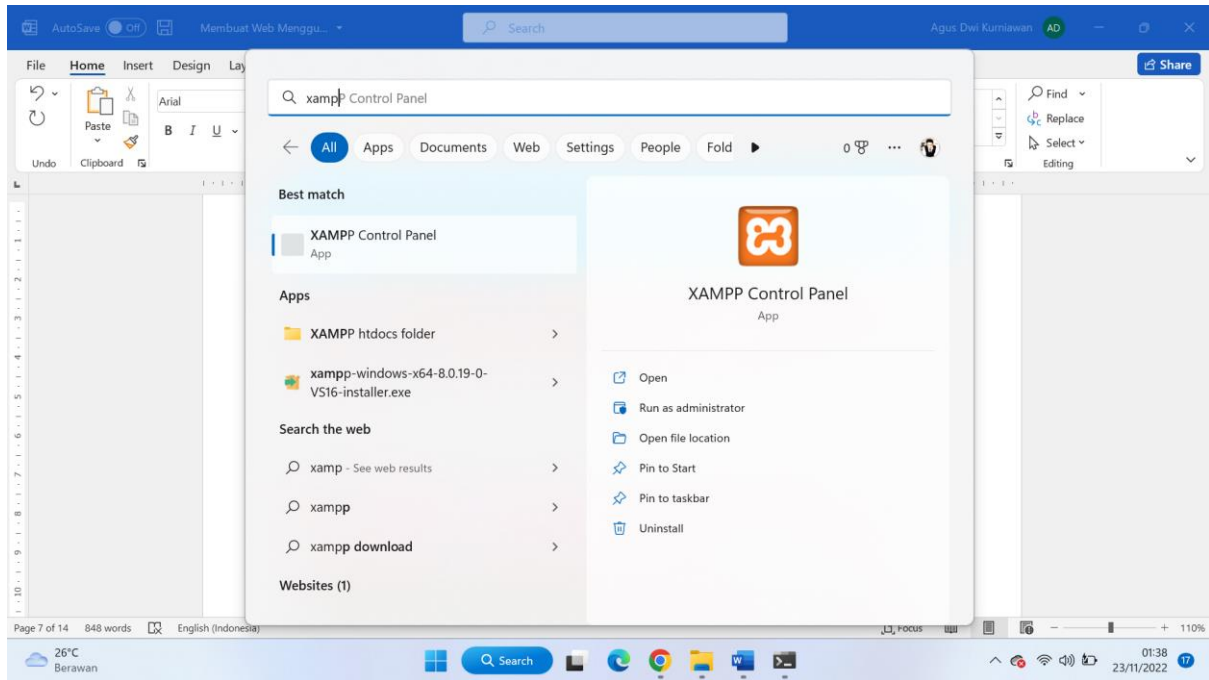
Halo teman-teman semuanya, di artikel sebelumnya kita telah berhasil belajar bagaimana cara melakukan proses instalasi dan menjalankan project baru di Laravel 9. Sekarang, kita akan lanjutkan belajar bagaimana cara membuat migration di Laravel 9.

Migration adalah sebuah version control database, dimana akan membantu team untuk mengubah dan membagikan sebuah schema database dari aplikasi yang dibangun. Jika sebelumnya teman-teman biasanya membuat table-table secara manual di dalam database, maka dengan migration hal itu sudah tidak perlu dilakukan lagi.

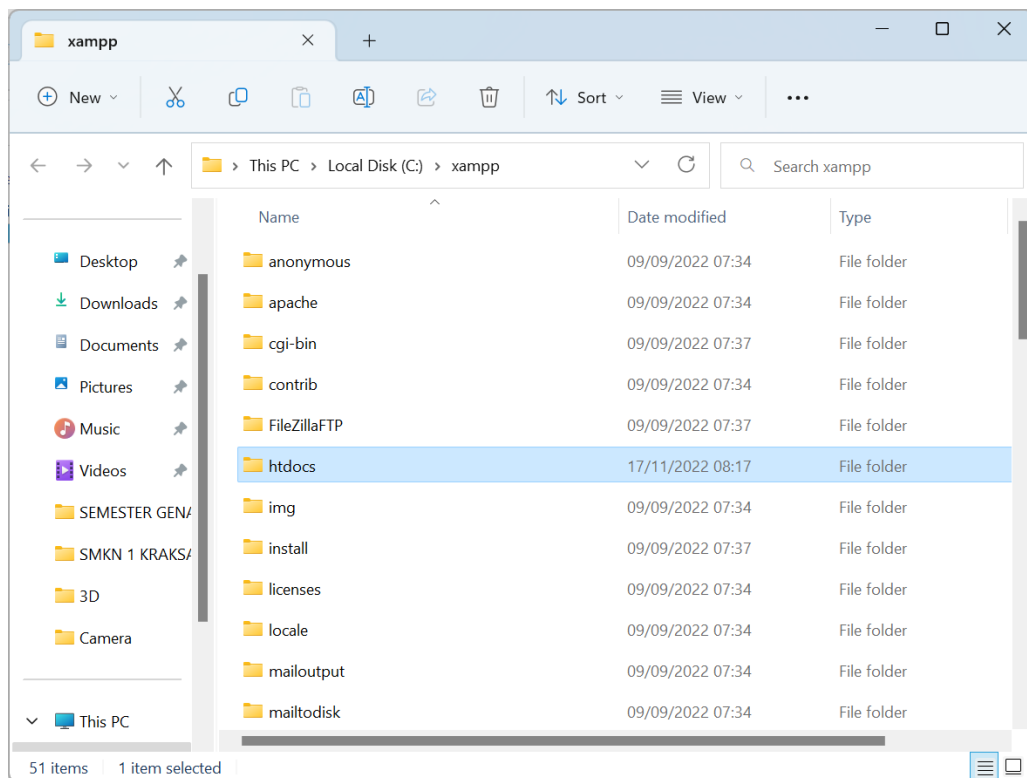
Pada kesempatan kali ini kita akan belajar membuat migration baru, yang mana ini akan kita gunakan untuk studi kasus membuat CRUD di Laravel 9. Langsung saja kita mulai.

LANGKAH 1 – MEMBUKA PROJECT LARAVEL

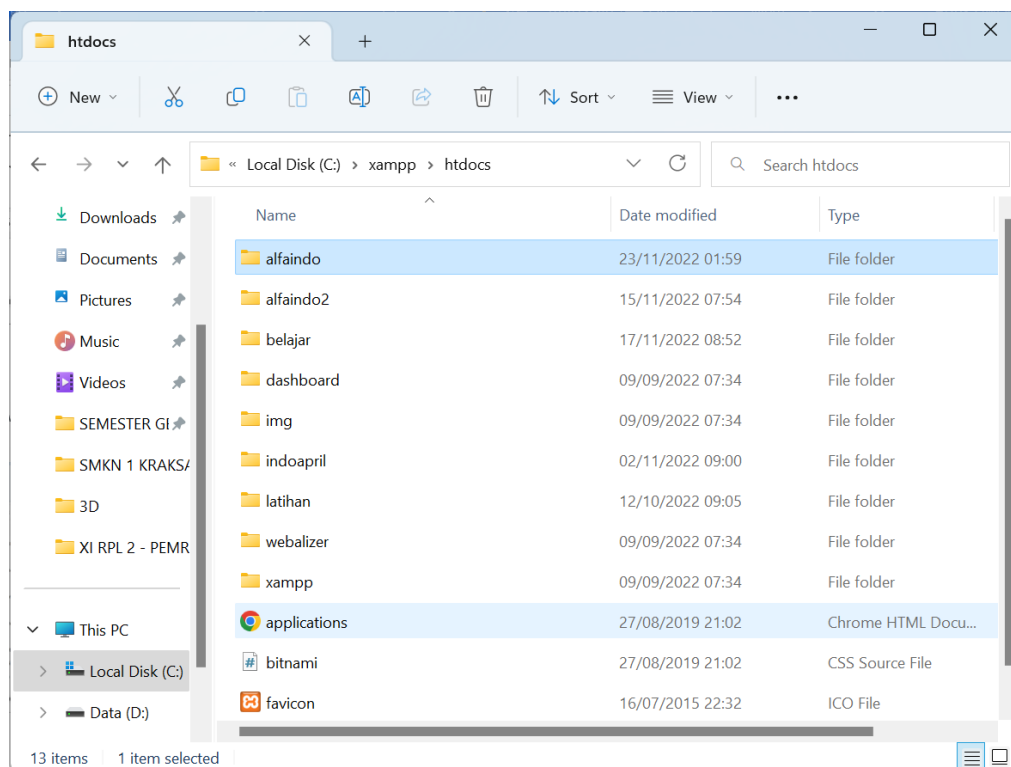
Silahkan masuk di dalam folder htdocs>alfaindo dengan menjalankan perintah change direktori di cmd dengan membuka xampp control panel lalu klik explorer :



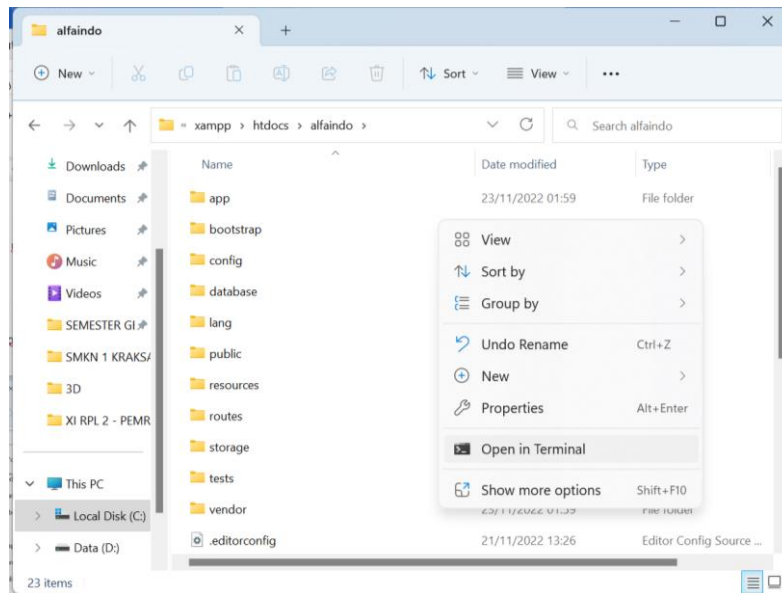
Lalu akan keluar explorer dari xampp kita, selanjutnya silahkan buka folder htdocs



Lalu buka folder alfaindo

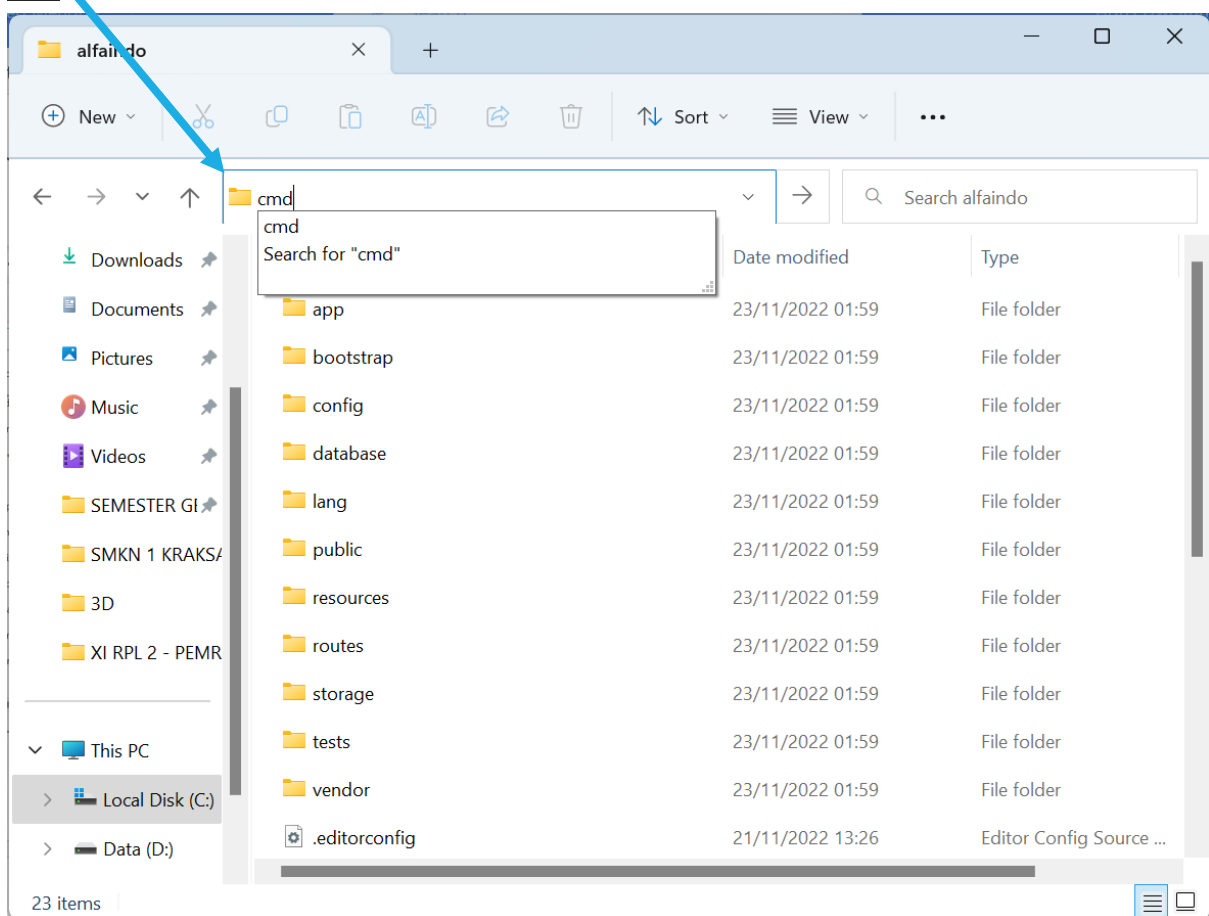


Cara buka di terminal untuk windows 11 silahkan klik kanan > open in terminal :



Untuk windows 10 kebawah bisa klik direktori lalu isi dengan

cmd



Maka akan keluar terminal/cmd dari direktori kita :

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

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

Apabila sudah di direktori maka kita buka isi folder nya di visual studio code dengan perintah :

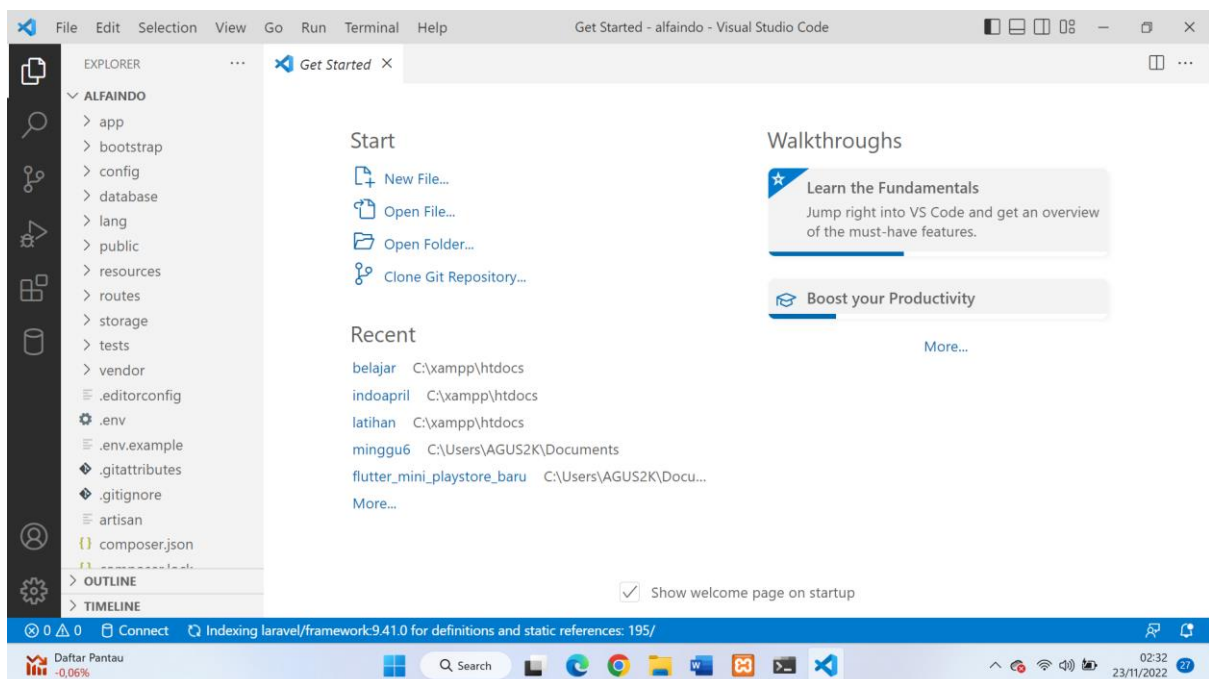
`code .`

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\xampp\htdocs\alfaindo> code .|
```

Tekan Enter, maka project folder akan terbuka di Visual Studio Code .
Seperti gambar dibawah ini:



LANGKAH 2 – KONFIGURASI KONEKSI DATABASE

Karena akan bekerja menggunakan database, maka kita perlu melakukan konfigurasi koneksi database-nya terlebih dahulu. Silahkan buka project kita menggunakan Text Editor, kemudian cari file yang bernama `.env`.

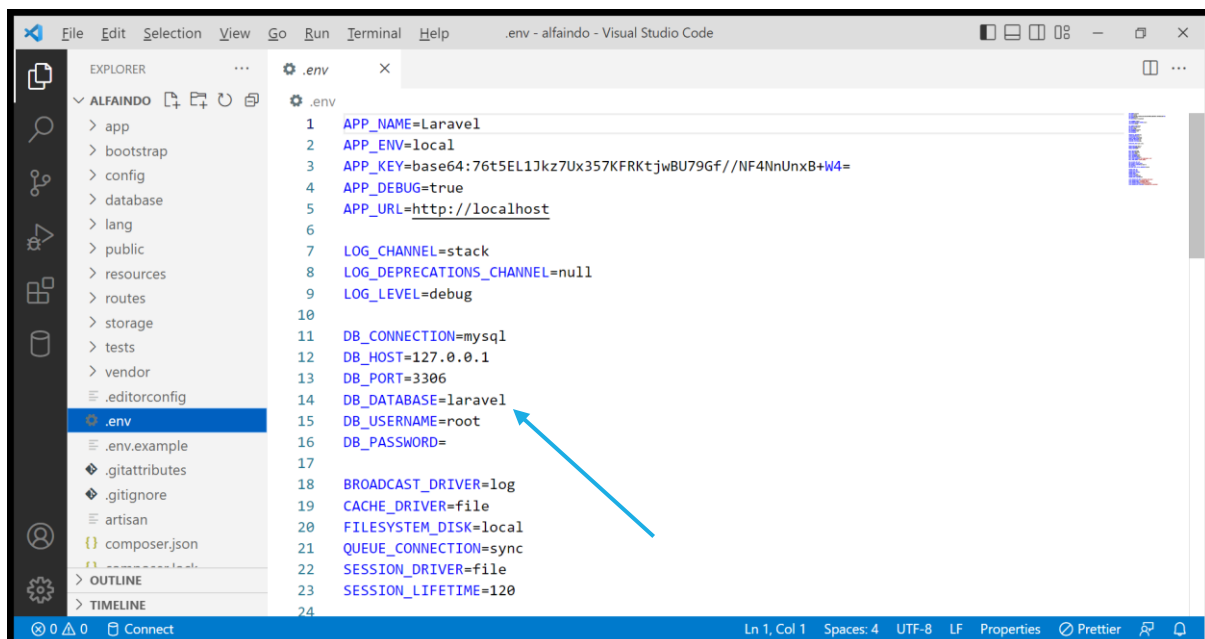
Jika sudah ketemu, silahkan cari kode berikut ini di dalam file `.env`.

```
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

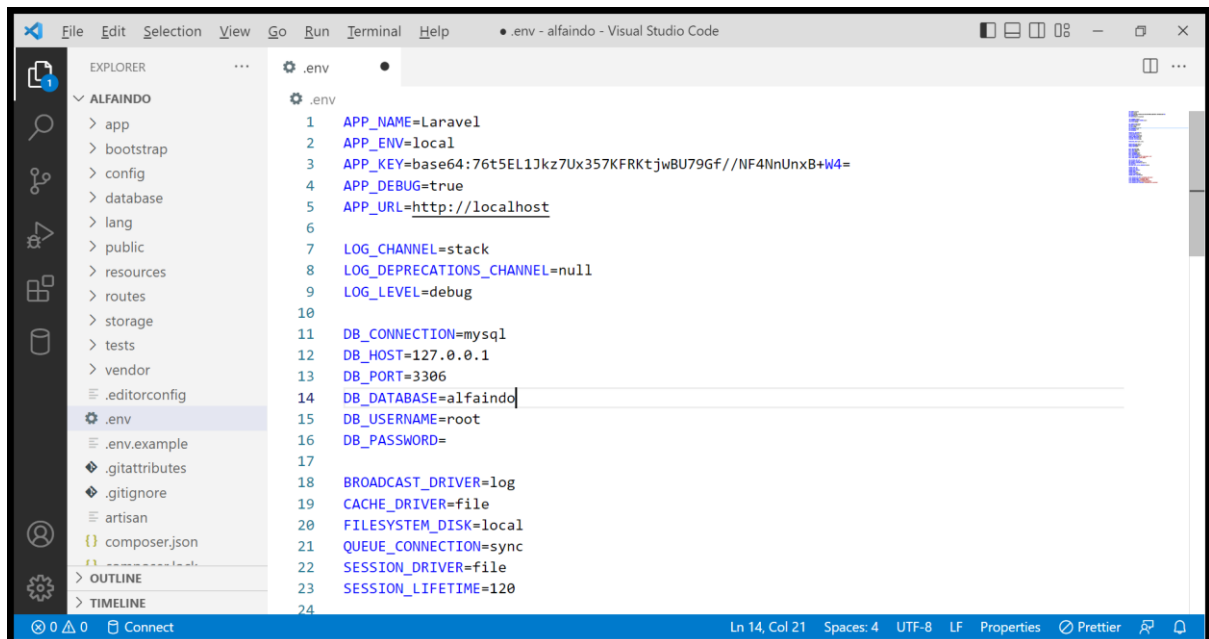
Dan ubahlah menjadi seperti berikut ini :

```
DB_DATABASE=alfaindo
DB_USERNAME=root
DB_PASSWORD=
```

Di atas, kita atur untuk `DB_DATABASE` atau nama database yang akan kita gunakan menjadi `alfaindo`. Kemudian untuk `DB_USERNAME` kita biarkan default, yaitu `root` dan untuk `DB_PASSWORD` silahkan disesuaikan dengan konfigurasi-nya masing-masing. Jika menggunakan XAMPP, maka untuk password-nya dikosongkan saja (by default kosong).



Menjadi seperti gambar dibawah ini :



The image shows a screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named 'ALFAINDO' with various folders and files. The file '.env' is selected and its contents are displayed in the main editor area. The .env file contains configuration for a Laravel application, including database settings, logging, and cache configuration. The status bar at the bottom indicates the current line and column (Ln 14, Col 21) and the file encoding (UTF-8).

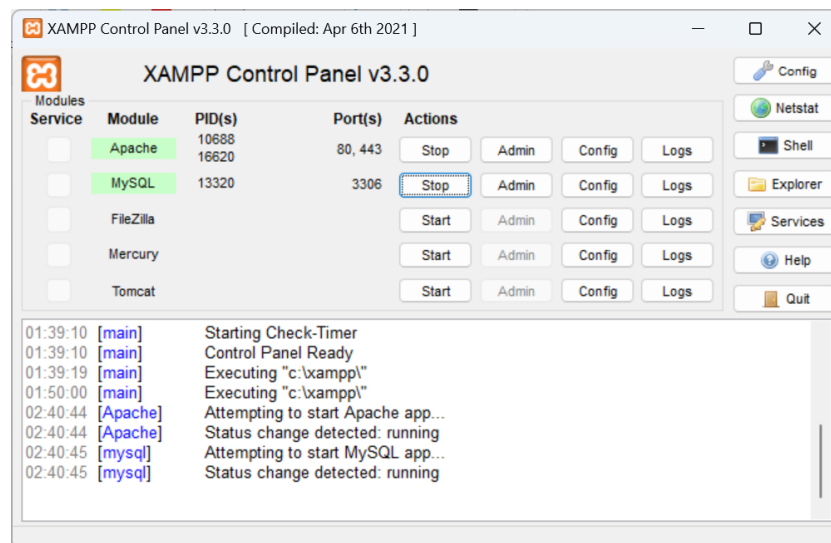
```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:76t5EL1Jkz7Ux357KFRKtjwBU79Gf//NF4NnUnxB+W4=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=alfaindo
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
```

Lalu silahkan **SAVE** file .env nya

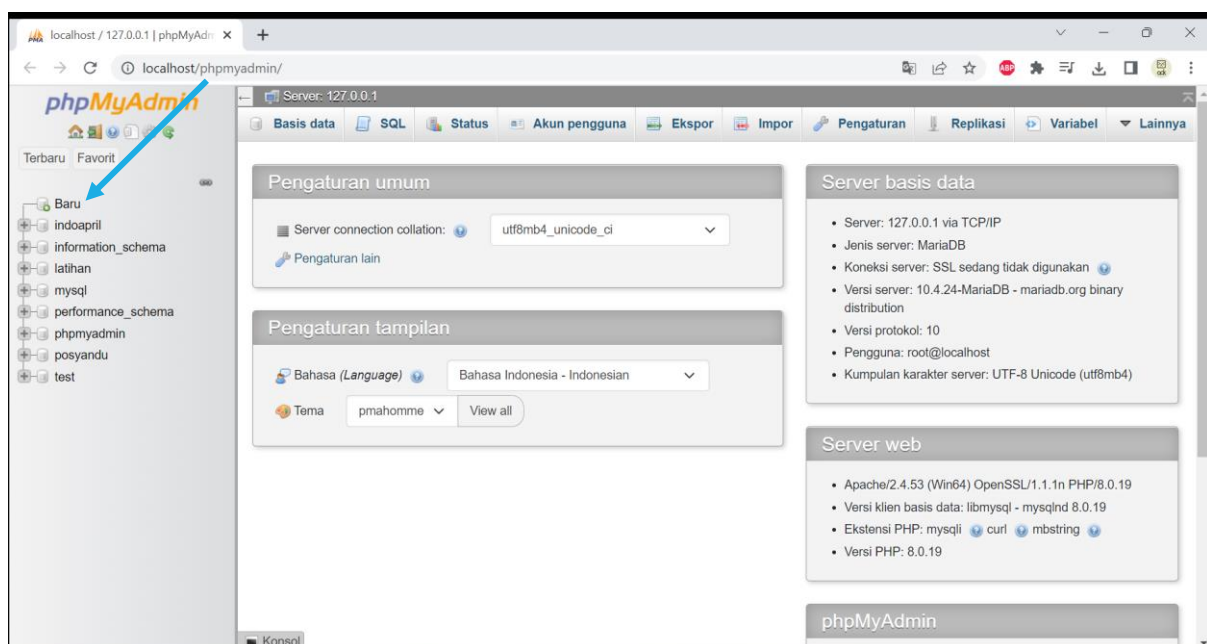
INFORMASI : silahkan restart server Laravel-nya setelah melakukan perubahan di dalam file **.env**.

LANGKAH 3 – KONFIGURASI KONEKSI DATABASE

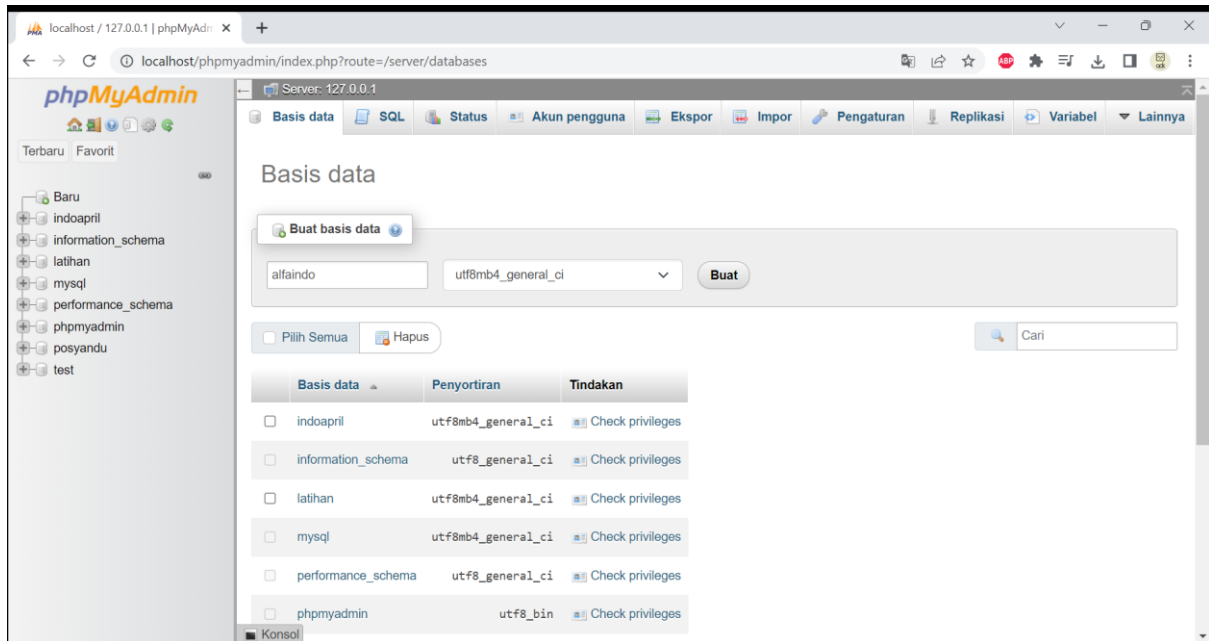
Setelah kita berhasil melakukan konfigurasi koneksi database, sekarang kita lanjutkan untuk membuat database-nya. Sekarang kita lanjutkan untuk membuat database baru di dalam <http://localhost/phpmyadmin> dengan menyalakan xampp dan silahkan buat database baru dengan nama **alfaindo**. Buka xampp control panel lalu nyalakan apache dan mysql nya.



Lalu klik admin di MySQL . Maka akan keluar PHPMyAdmin. Klik Baru



Lalu buat database alfaindo



LANGKAH 4 – MEMBUAT MODEL DAN MIGRATION

Sekarang, kita lanjutkan belajar bagaimana cara membuat model dan migration di Laravel 9. Silahkan jalankan perintah berikut ini di dalam terminal/CMD dan pastikan berada di dalam project Laravel-nya.

```
php artisan make:model Barang -m
```

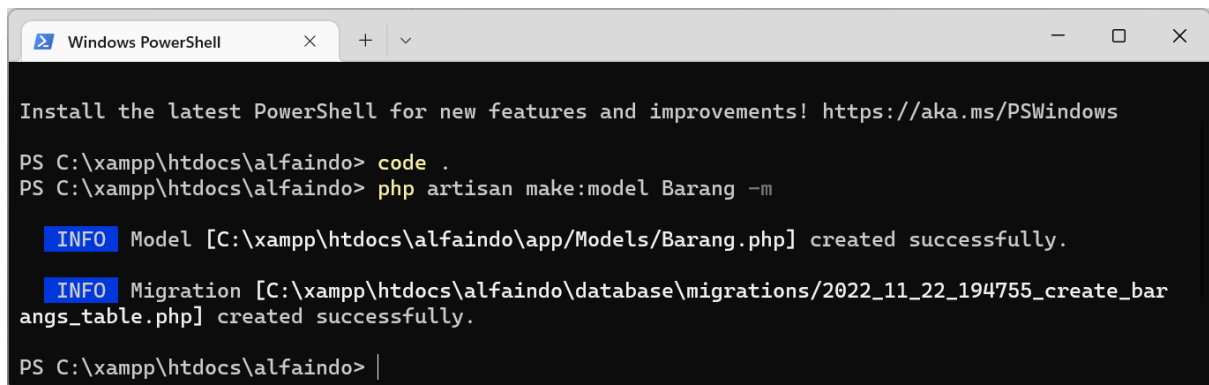


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\xampp\htdocs\alfaindo> code .
PS C:\xampp\htdocs\alfaindo> php artisan make:model Barang -m
```

Perintah di atas, digunakan untuk membuat model baru dengan nama **Post** dan kita tambahkan flag **-m**, yang artinya file migration-nya juga akan ikut dibuat.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\xampp\htdocs\alfaindo> code .
PS C:\xampp\htdocs\alfaindo> php artisan make:model Barang -m

[INFO] Model [C:\xampp\htdocs\alfaindo\app\Models\Barang.php] created successfully.
[INFO] Migration [C:\xampp\htdocs\alfaindo\database\Migrations\2022_11_22_194755_create_barangs_table.php] created successfully.

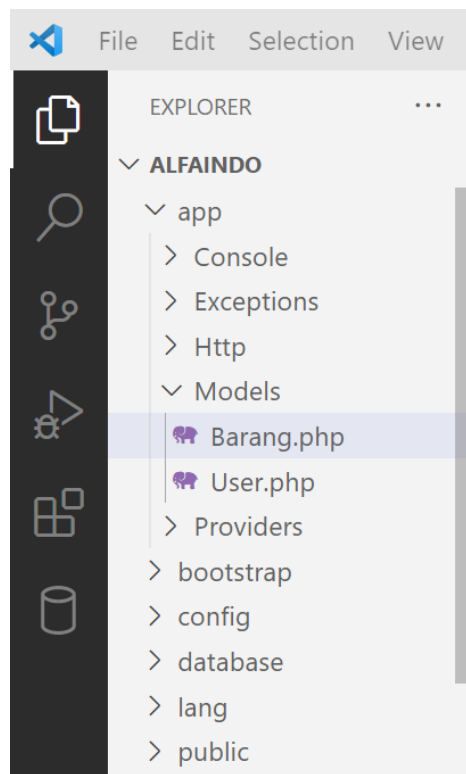
PS C:\xampp\htdocs\alfaindo> |
```

Jika perintah di atas berhasil dijalankan, maka kita akan mendapatkan 2 file baru, yaitu :

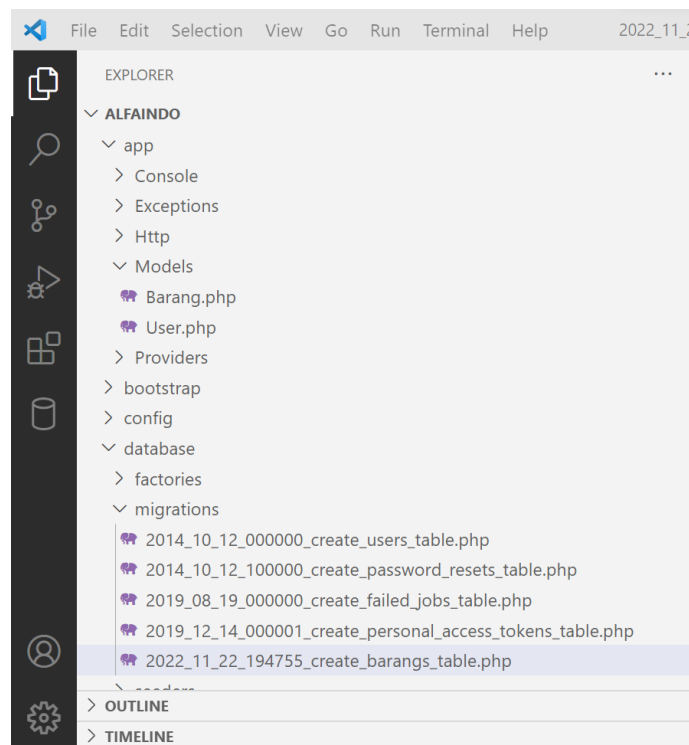
- **app/Models/Post.php**
- **database/migrations/2022_11_22_194755_create_barangs_table.php**

INFORMASI : untuk nama file migration akan random sesuai tanggal pembuatannya.

Tempat Models yang sudah kita buat ada di app\Models\Barang.php



Tempat file migrasi barangs table yang sudah kita dibuat :

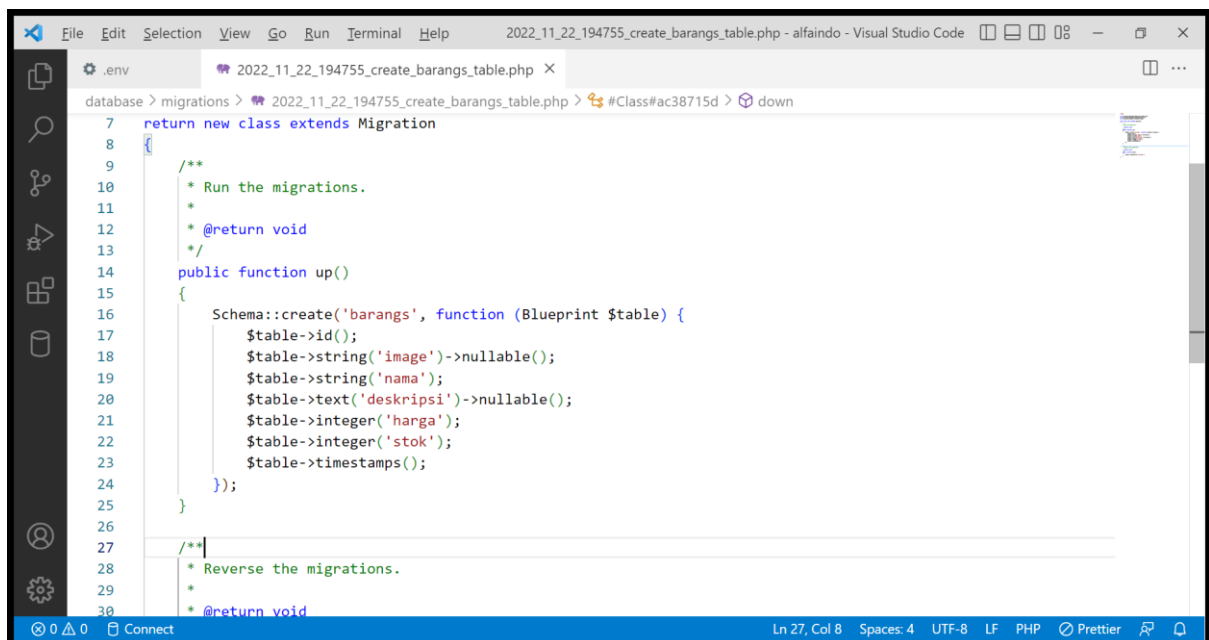


LANGKAH 5 – MENAMBAHKAN FIELD TABLE

Setelah berhasil membuat model dan migration, sekarang kita lanjutkan untuk menambahkan field atau kolom di dalam file migration. Field yang kita tambahkan ini akan digenerate di dalam table `posts` yang ada di database.

Silahkan buka file `database/migrations/2022_11_22_194755_create_barangs_table.php`, kemudian pada bagian `function up`, ubah kode-nya menjadi seperti berikut ini :

```
public function up()
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->string('image')->nullable();
        $table->string('nama');
        $table->text('deskripsi');
        $table->integer('harga');
        $table->string('stok');
        $table->timestamps();
    });
}
```



Dari perubahan kode di atas, kita menambahkan 5 field baru, yaitu :

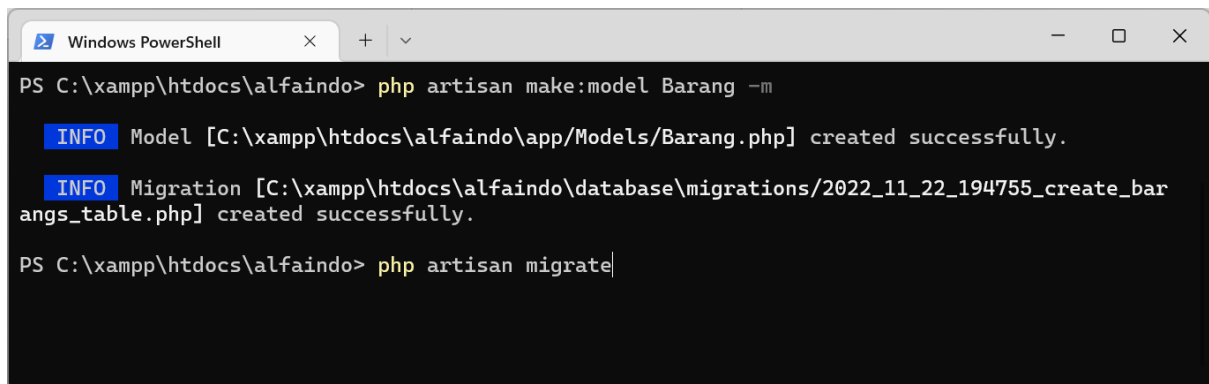
FIELD	TYPE DATA
image	string
nama	string
deskripsi	text
harga	integer
stok	integer

LANGKAH 6 – MENJALANKAN MIGRATION

Sekarang kita akan belajar menjalankan perintah migrate. Perintah ini akan digunakan untuk melakukan proses create table `posts` beserta field yang ada di dalamnya ke dalam database.

Silahkan jalankan perintah berikut ini di dalam terminal/CMD :

```
php artisan migrate
```



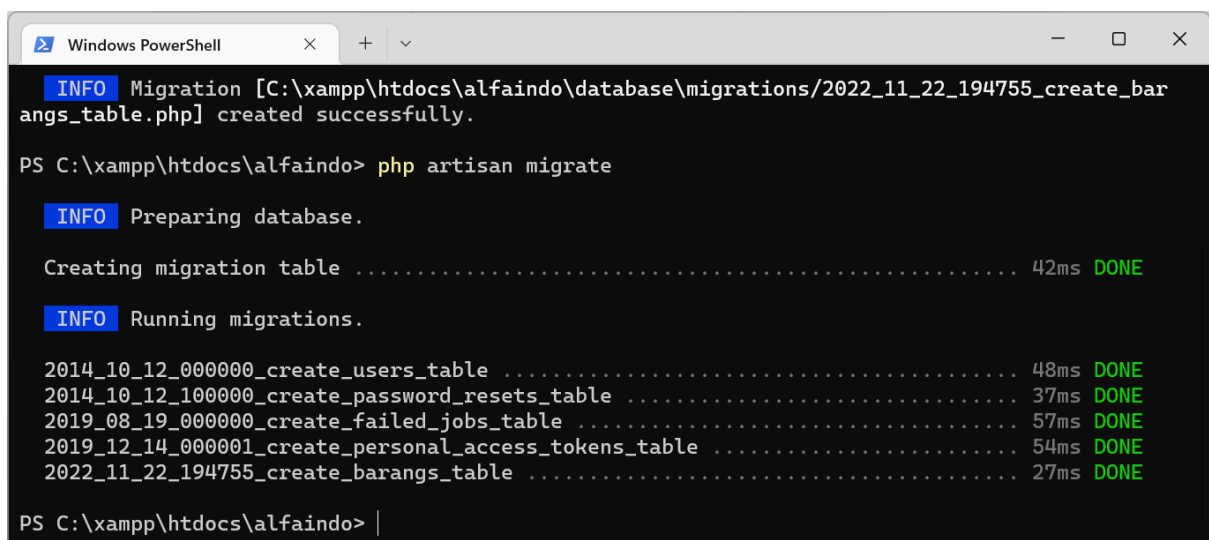
```
Windows PowerShell
PS C:\xampp\htdocs\alfaindo> php artisan make:model Barang -m

[INFO] Model [C:\xampp\htdocs\alfaindo\app\Models\Barang.php] created successfully.

[INFO] Migration [C:\xampp\htdocs\alfaindo\database\Migrations\2022_11_22_194755_create_barangs_table.php] created successfully.

PS C:\xampp\htdocs\alfaindo> php artisan migrate|
```

Jika perintah di atas berhasil dijalankan, maka akan mendapatkan output seperti berikut ini :



```
Windows PowerShell
[INFO] Migration [C:\xampp\htdocs\alfaindo\database\Migrations\2022_11_22_194755_create_barangs_table.php] created successfully.

PS C:\xampp\htdocs\alfaindo> php artisan migrate

[INFO] Preparing database.

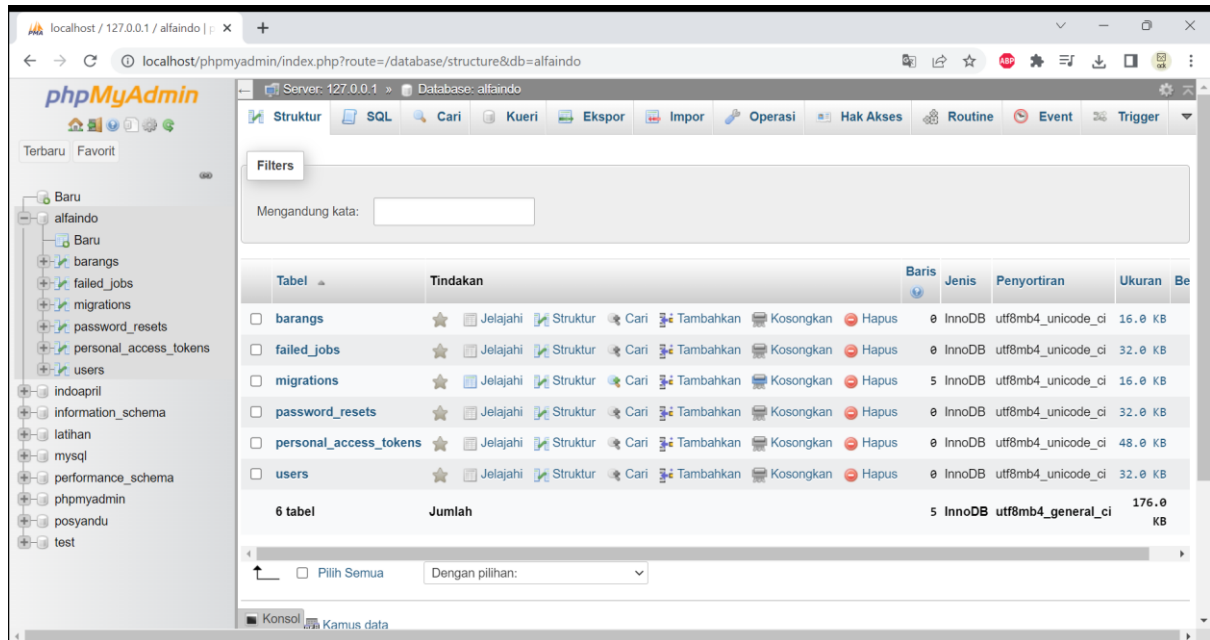
Creating migration table ..... 42ms DONE

[INFO] Running migrations.

2014_10_12_000000_create_users_table ..... 48ms DONE
2014_10_12_100000_create_password_resets_table ..... 37ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 57ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 54ms DONE
2022_11_22_194755_create_barangs_table ..... 27ms DONE

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

Jika berhasil maka teman-teman bisa mencoba cek database dan teman-teman akan menemukan table blogs yang sudah kita buat di dalam file migration.



Kenapa kok jadi barangs ??? Kan kita buat model dengan nama Barang !

Oke, jadi topiknya adalah standar / konvensi penamaan Model dalam Framework Laravel. Pada dasarnya konvensinya adalah nama Model adalah bentuk singular (tunggal) dan diawali dengan huruf kapital sesuai PSR-2 untuk penamaan Class dari nama tabel yang dibuat, contohnya untuk tabel dengan nama “photos” maka penamaan untuk modelnya ialah “Photo”. Jadi itulah kenapa pada saat kita membuat model Barang maka pada saat di migrasi Namanya adalah “barangs”.

LANGKAH 7 – KONFIGURASI MASS ASSIGNMENT

Mass Assignment di Laravel memungkinkan kita untuk mengizinkan sebuah field dari table agar dapat menyimpan sebuah data. Karena table yang kita gunakan nantinya buat menyimpan data, maka kita perlu melakukan konfigurasi Mass Assignment-nya terlebih dahulu.

Silahkan buka file `app/Models/Barang.php`, kemudian ubah kode-nya menjadi seperti berikut ini :

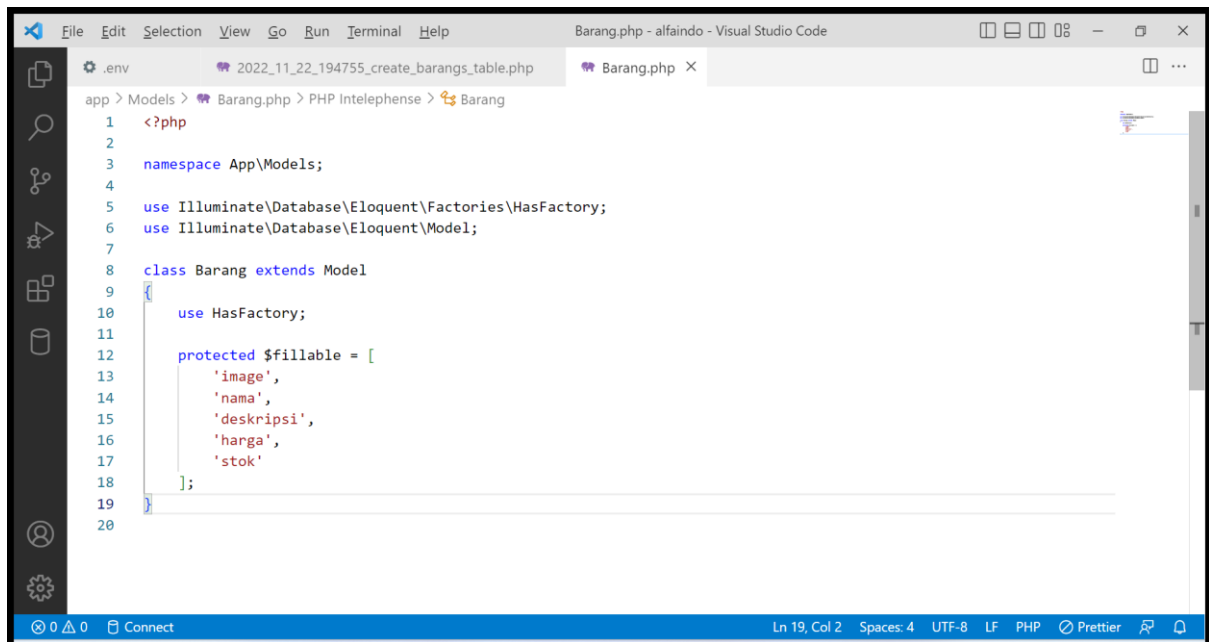
```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Barang extends Model
{
    use HasFactory;

    /**
     * fillable
     *
     * @var array
     */
    protected $fillable = [
        'image',
        'nama',
        'deskripsi',
        'harga',
        'stok',
    ];
}
```



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Barang extends Model
9 {
10     use HasFactory;
11
12     protected $fillable = [
13         'image',
14         'nama',
15         'deskripsi',
16         'harga',
17         'stok'
18     ];
19 }
20
```

Di atas, kita menambahkan properti baru yang bernama **\$fillable**. Properti tersebut yang dinamakan Mass Assignment di Laravel. Dimana di dalamnya kita menambahkan field-field yang diizinkan untuk melakukan manipulasi data.

Sampai disini pembahasan bagaimana cara membuat model dan migration di Laravel 9. Di artikel selanjutnya kita semua akan belajar bagaimana cara menampilkan data dari database di Laravel 9.

Terima Kasih