

# **LAPORAN PRAKTIKUM PEMROGRAMAN WEB**

**Judul: Laravel 2**



**Disusun oleh  
Adeline Fellita Marwa (M0521002)**

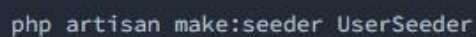
**PROGRAM STUDI INFORMATIKA  
FAKULTAS INFORMASI DAN SAINS DATA  
UNIVERSITAS SEBELAH MARET  
2023**

Pada PPW23-10 ini saya menggunakan project lanjutan dari PPW23-09. pada praktikum kali ini kita akan mengimplementasikan materi laravel lanjutan:

- Factory & Seeder
- Eloquent & Query Builder
- Case Study & Initialization

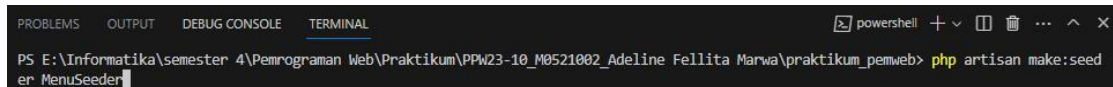
Berikut adalah langkah-langkah yang saya lakukan dalam praktikum ini:

1. Langkah pertama adalah membuat *seeder*, Untuk menghasilkan seeder, jalankan perintah `artisan make:seeder` pada terminal. Semua *seeder* yang dihasilkan oleh *framework* akan ditempatkan di direktori `database/seeder`



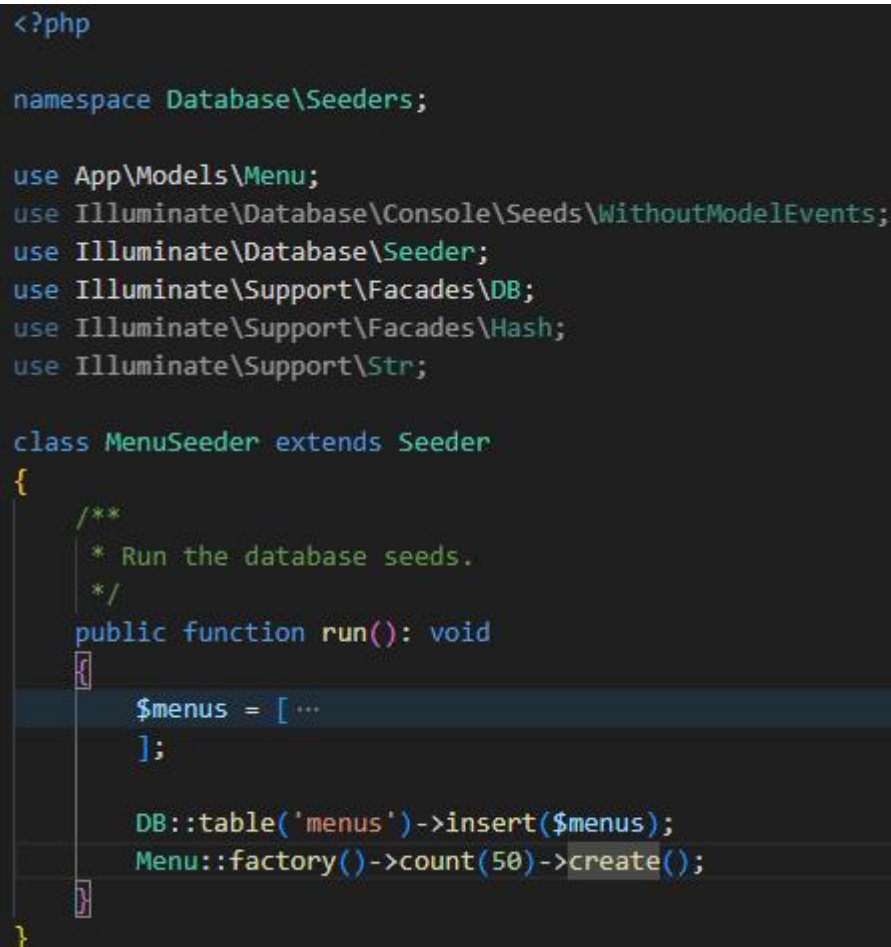
```
php artisan make:seeder UserSeeder
```

Pada praktikum ini saya mengganti nama file `UserSeeder` menjadi `MenuSeeder`



```
PS E:\Informatika\semester 4\Pemrograman Web\Praktikum\PPW23-10_M0521002_Adelina Fellita Marwa\praktikum_pemweb> php artisan make:seeder MenuSeeder
```

2. mari ubah *Class MenuSeeder* default dan tambahkan *insert Statement* database ke `runmetode`:



```
<?php

namespace Database\Seeders;

use App\Models\Menu;
use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Str;

class MenuSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $menus = [ ...
        ];

        DB::table('menus')->insert($menus);
        Menu::factory()->count(50)->create();
    }
}
```

3. Setelah itu kita memanggil Addition Seeder, Di dalam DatabaseSeeder kelas, Anda dapat menggunakan callmetode untuk mengeksekusi kelas seed tambahan.

```
10 class DatabaseSeeder extends Seeder
11 {
12     /**
13      * Seed the application's database.
14      */
15     public function run(): void
16     {
17
18         $this->call([
19             MenuSeeder::class,
20         ]);
21     }
22 }
```

4. Kemudian jalan Seeder dengan perintah artisan `db:seed`, Secara default, `db:seed` perintah menjalankan `Database\Seeders\DatabaseSeeder`, Namun, Anda dapat menggunakan `--class` opsi untuk menentukan kelas seeder tertentu untuk dijalankan secara individual:

```
php artisan db:seed

php artisan db:seed --class=UserSeeder
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL powershell + - x

PS E:\Informatika\semester 4\Pemrograman Web\Praktikum\PPW23-10\_M0521002\_Adelina Fellita Marwa\praktikum\_pemweb> php artisan db:seed

Setelah perintah berhasil dijalankan maka table menus pada database stabucks akan terisi oleh data seeder

Result Grid							
Filter Rows:							
	id	name	price	size	temperature	created_at	updated_at
▶	1	Kopi Hitam	28078	Small	Hot	NULL	NULL
	2	Kopi Susu	46610	Medium	Hot	NULL	NULL
	3	Espresso	37837	Small	Hot	NULL	NULL
	4	Cappuccino	35681	Large	Hot	NULL	NULL
	5	Iced Coffee	42469	Large	Cold	NULL	NULL

5. Untuk langkah berikutnya kita akan mendefinisikan *Factories*. Langkah pertama yang kita lakukan adalah membuat Factory menjalankan perintah artisan di bawah ini pada terminal:

```
php artisan make:factory PostFactory
```

Pada project ini saya mengganti file `PostFactory` dengan `MenuFactory`

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS E:\Informatika\semester 4\Pemrograman Web\Praktikum\PPW23-10_M0521002_Adeline Fellita Marwa\praktikum_pemweb> php artisan make:factory MenuFactory --model=Menu
```

6. mari ubah *Class* `MenuFactory` default dan tambahkan *state* pada function definition:

```
namespace Database\Factories;

use App\Models\Menu;
use Illuminate\Database\Eloquent\Factories\Factory;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Menu>
 */
class MenuFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    protected $model = Menu::class;
    public function definition(): array
    {
        return [
            'name' => fake()->randomElement(['Espresso', 'Cappuccino', 'Latte', 'Mocha', 'Americano', '
            'size' => fake()->randomElement(['Small', 'Medium', 'Large']),
            'temperature' => fake()->randomElement(['Hot', 'Cold']),
            'price' => fake()->randomNumber(5, true),
        ];
    }
}
```

7. Kemudian jalan *Seeder* dengan perintah `artisan db:seed`, Secara default, `db:seed` perintah menjalankan `Database\Seeders\DatabaseSeeder`, Namun, Anda dapat menggunakan `--class` opsi untuk menentukan kelas *seeder* tertentu untuk dijalankan secara individual:

```
php artisan db:seed

php artisan db:seed --class=UserSeeder

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS E:\Informatika\semester 4\Pemrograman Web\Praktikum\PPW23-10_M0521002_Adeline Fellita Marwa\praktikum_pemweb> php artisan db:seed
```

Setelah perintah berhasil dijalankan maka table *menus* pada database *stabucks* akan terisi oleh data *Factory* terdapat 50 data.

Result Grid							
Filter Rows: <input type="text"/>							
	id	name	price	size	temperature	created_at	updated_at
	48	Frappuccino	54668	Large	Cold	2023-07-03 22:40:39	2023-07-03 22:40:39
	49	Turkish Cof...	18929	Large	Hot	2023-07-03 22:40:39	2023-07-03 22:40:39
	50	Long Black	77388	Medium	Cold	2023-07-03 22:40:39	2023-07-03 22:40:39
	51	Vienna Cof...	35452	Medium	Hot	2023-07-03 22:40:39	2023-07-03 22:40:39
	52	Flat White	81890	Medium	Cold	2023-07-03 22:40:39	2023-07-03 22:40:39
	53	Vienna Cof...	26774	Large	Hot	2023-07-03 22:40:39	2023-07-03 22:40:39
	54	Americano	35670	Small	Hot	2023-07-03 22:40:39	2023-07-03 22:40:39
	55	Cortado	82417	Large	Hot	2023-07-03 22:40:39	2023-07-03 22:40:39
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

8. Kemudian langkah selanjutnya adalah menjalankan query database, buka file `MenuController.php` pada direktori `app\Http\Controller`.

```
<?php

namespace App\Http\Controllers;
use Illuminate\Support\Facades\DB;
use App\Models\Menu;
use Illuminate\Http\Request;

class MenuController extends Controller
{
    //
    public function index(){

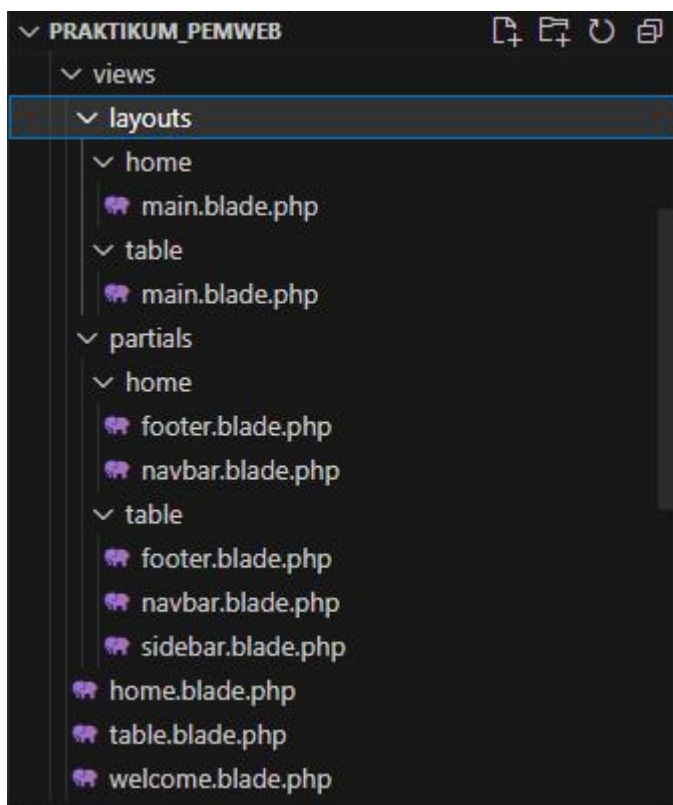
        $menus = DB::table('menus')
            ->select('*')
            ->where('temperature', 'Hot')
            ->where('size', 'small')
            ->get();
        return view('table',[
            'menus' => $menus
        ]);
    }
}
```

Disini ingin menampilkan menu yang temperature nya hot dan sizenya small, maka saya menggunakan query seperti di atas.

9. Jalankan perintah `php artisan serve` Kemudian buka alamat server nya pada browser kemudian tampilan tabel nya akan sesuai dengan query yang saya gunakan.

ID	Name	Size	Temperature	Price
1	Kopi Hitam	Small	Hot	Rp28.078
3	Espresso	Small	Hot	Rp37.837
12	Latte	Small	Hot	Rp52.973
20	Long Black	Small	Hot	Rp62.188
21	Irish Coffee	Small	Hot	Rp70.312
24	Vienna Coffee	Small	Hot	Rp82.397
25	Macchiato	Small	Hot	Rp72.976
28	Americano	Small	Hot	Rp78.542
29	Mocha	Small	Hot	Rp60.738
31	Latte	Small	Hot	Rp23.801

10. Kemudian tahap selanjutnya adalah Case Study & Initialization. Caranya adalah kita membuat direktori baru pada direktori views. Dengan susunan seperti gambar di bawah.

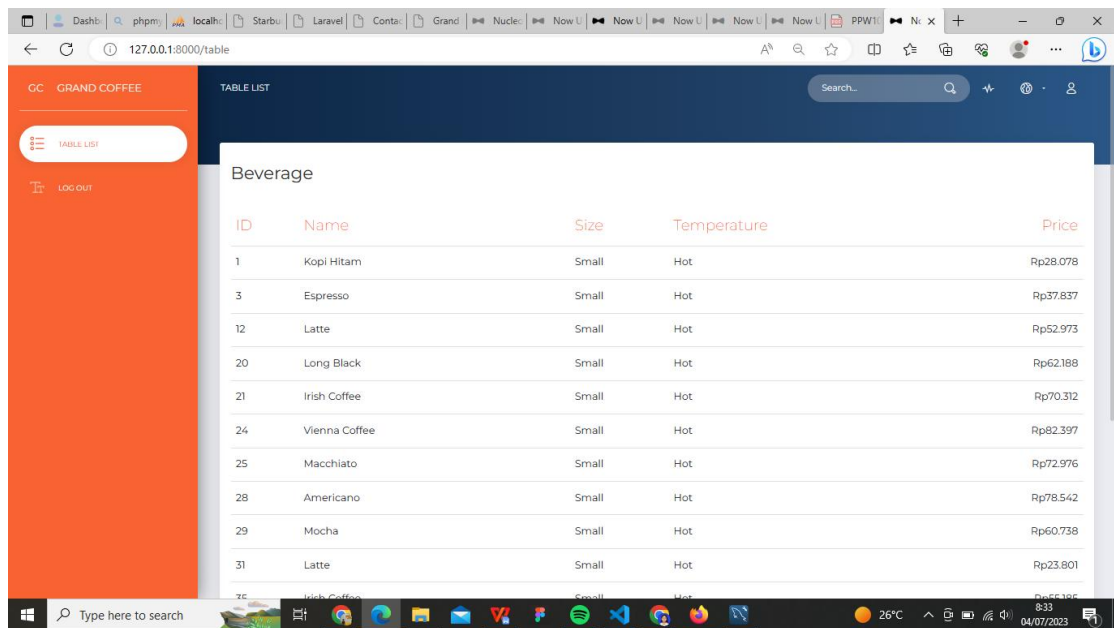


Pada file `home.blade.php` kita pindahkan bagian `footer` dan `navbar` ke file di dalam direktori `partials\home` kemudian kita pindahkan juga bagian `navbar` ke atas dan `footer` ke bawah ke dalam file `main.blade.php` di dalam direktori `layouts\home` kemudian tambahkan `@include('partials.home.navbar')`



diantara batas navbar dan tambahkan juga `@include('partials.home.footer')` diantara batas footer, kemudian tambahkan juga `@yield('container')` diantara batas navbar dan footer. Kemudian tambahkan `@extends('layouts.home.main')` dan `@section('container')` dan `@endsection` pada file `home.blade.php` Lakukan hal yang pada file `table.blade.php`

11. Jika ada kesalahan langkah maka akan terjadi error pada halamannya ketika di-*refresh* jika langkah sudah benar tidak akan terjadi apa-apa pada halaman (tampilan sama).



The screenshot shows a web application interface for 'GRAND COFFEE'. On the left is an orange sidebar with a hamburger menu icon and a 'LOG OUT' button. The main content area has a dark blue header with a search bar and a 'TABLE LIST' title. Below the title is a table with the heading 'Beverage'. The table has five columns: ID, Name, Size, Temperature, and Price. It contains 12 rows of data for various coffee drinks. The browser's address bar shows the URL '127.0.0.1:8000/table'. The Windows taskbar at the bottom shows the date '04/07/2023' and the time '8:33'.

ID	Name	Size	Temperature	Price
1	Kopi Hitam	Small	Hot	Rp28.078
3	Espresso	Small	Hot	Rp37.837
12	Latte	Small	Hot	Rp52.973
20	Long Black	Small	Hot	Rp62.188
21	Irish Coffee	Small	Hot	Rp70.312
24	Vienna Coffee	Small	Hot	Rp82.397
25	Macchiato	Small	Hot	Rp72.976
28	Americano	Small	Hot	Rp78.542
29	Mocha	Small	Hot	Rp60.738
31	Latte	Small	Hot	Rp23.801
32	Irish Coffee	Small	Hot	Rp55.305