

PKL-02

Membuat Tabel Product

1. Membuat Model dan Migration Product

Silahkan jalankan perintah berikut didalam terminal/CMD:

```
Php artisan make:model Product -m
```

2. Mengatur struktur tabel Product

Selanjutnya silahkan buka file *migration* yang berada folder

`database/migrations/xxxx_xx_xx_create_products_table.php`, lalu ubah method `up()` menjadi seperti berikut:

```
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();
    });
}
```

3. Menambahkan mass Assignment

Setelah kita bias membuat table **product** menggunakan *migration*, langkah berikutnya adalah menyipkan **Model Product** agar bias menerima data secara *mass assignment*.

Mass Assignment di Laravel digunakan untuk mengisi banyak kolom sekaligus dalam satu perintah, misalnya saat menyimpan data dari *form*. Agar hal ini aman, Laravel mengharuskan kita mendefinisikan field apa saja yang boleh diisi melalui property `$fillable` pada Model.

Silahkan anda buka file Model Product yang berada di lokasi berikut:

```
App/Models/Product.php
```

Kemudian ubah kodenya menjadi seperti berikut ini:

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
```

```
class Product extends Model
{
    /**
     * fillable
     *
     * @var array
     */
    protected $fillable = [
        'image',
        'title',
        'description',
        'price',
        'stock',
    ];
}
```

4. Menjalankan migration

Setelah kita selesai membuat struktur table **Products** dan **Mess Assignment** pada **Model Product** , sekarang saatnya menjalankan migration agar table tersebut benar-benar dibuat didatabase.

Silahkan jalankan perintah berikut ini didalam :

```
php artisan optimize:clear
php artisan migrate:fresh
```

Pada tutorial berikutnya, kita akan mulai belajar bagaimana membuat CRUD Product di Laravel 13.

CRUD Product

Silahkan jalankan perintah berikut ini didalam :

```
php artisan make:resource ProductResource
```

Silahkan buka **app/Http/Resources/ProductResource.php**

```
public function toArray(Request $request): array
{
    return [
        'id' => $this->id,
        'image' => $this->image,
        'title' => $this->title,
        'description' => $this->description,
        'price' => (float) $this->price,
        'stock' => (int) $this->stock,
        'created_at' => $this->created_at->format('Y-m-d H:i:s'),
    ];
}
```

Silahkan jalankan perintah berikut ini didalam :

```
php artisan make:request StoreProductRequest
php artisan make:request UpdateProductRequest
```

Silahkan buka **app/Http/Requests/StoreProductRequest.php**

```
<?php

namespace App\Http\Resources;

use Illuminate\Http\Request;
use Illuminate\Http\Resources\Json\JsonResource;

class ProductResource extends JsonResource
{
    /**
     * Transform the resource into an array.
     *
     * @return array<string, mixed>
     */
}
```

```

public function toArray(Request $request): array
{
    return [
        'id' => $this->id,
        'image' => $this->image,
        'title' => $this->title,
        'description' => $this->description,
        'price' => (float) $this->price,
        'stock' => (int) $this->stock,
        'created_at' => $this->created_at->format('Y-m-d H:i:s'),
    ];
}
}

```

Silahkan jalankan perintah berikut ini didalam :

```
php artisan make:controller Api/ProductController --api
```

Silahkan buka **app/Http/Controllers/Api/ProductController.php**

```

<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Product;
use App\Http\Resources\ProductResource;
use App\Http\Requests\StoreProductRequest;

class ProductController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        return ProductResource::collection(Product::paginate(10));
    }
}

```

```

// POST /api/products
public function store(StoreProductRequest $request)
{
    $product = Product::create($request->validated());
    return new ProductResource($product);
}

// GET /api/products/{product}
public function show(Product $product)
{
    return new ProductResource($product);
}

// PUT/PATCH /api/products/{product}
public function update(UpdateProductRequest $request, Product $product)
{
    $product->update($request->validated());
    return new ProductResource($product);
}

// DELETE /api/products/{product}
public function destroy(Product $product)
{
    $product->delete();
    return response()->json([
        'message' => 'Product deleted successfully.'
    ], Response::HTTP_OK);
}
}

```

Silahkan buka **routes/api.php**

```

<?php

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\Api\AuthController;
use App\Http\Controllers\Api\ProductController;

Route::middleware('auth:sanctum')->group(function () {
    Route::get('/user', [AuthController::class, 'getUser']);
    Route::post('/logout', [AuthController::class, 'logout']);
    Route::apiResource('products', ProductController::class);
});

```

```
});
```

```
Route::post('register', [AuthController::class, 'register']);
```

```
Route::post('login', [AuthController::class, 'login']);
```