

Nama : Misyhel Oktavia Br Nababan

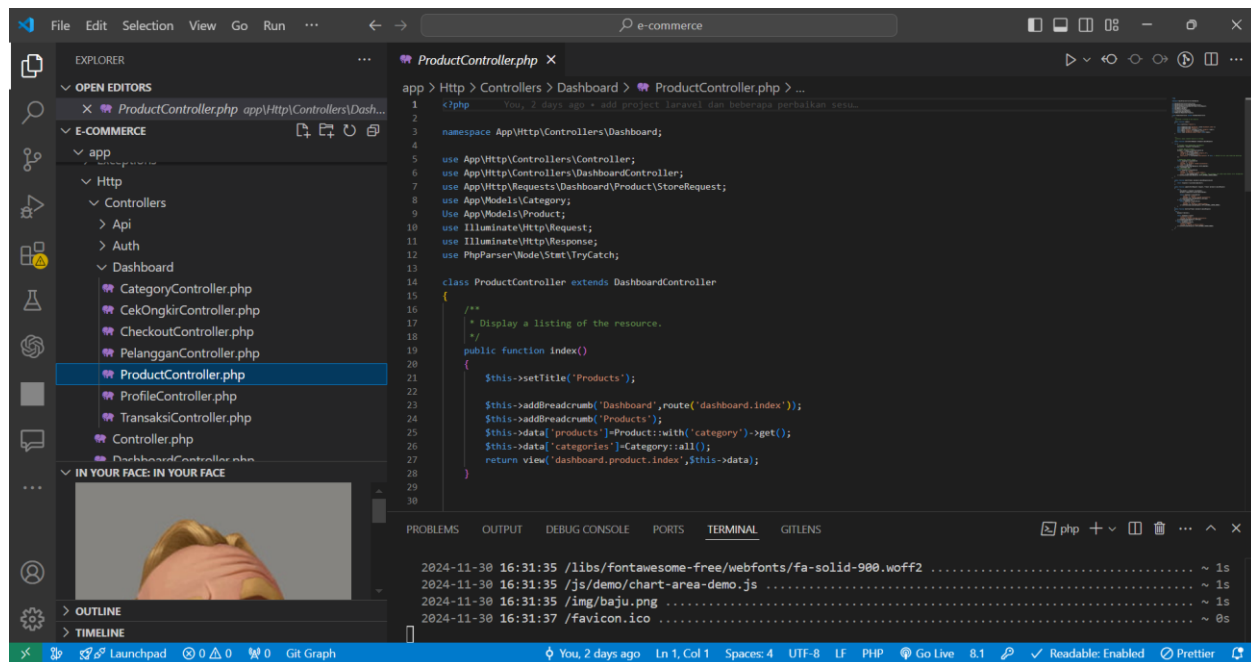
Kelas : PW 1

Kelompok : 5

1. Membuat Halaman Pengelolaan Produk

Pertama dimulai dari product controller sebagai berikut :

Karna fungsi nya laravel akan akses terlebih dahulu controller yg di request user dan lalu laravel akan mengarahkan user ke routes nya



The screenshot shows the Visual Studio Code editor with a project named 'e-commerce'. The Explorer sidebar on the left shows the file structure: 'app' > 'Http' > 'Controllers' > 'Dashboard' > 'ProductController.php'. The 'ProductController.php' file is open in the editor, showing the following code:

```
1 <?php You, 2 days ago • add project laravel dan beberapa perbaikan sesu...
2
3 namespace App\Http\Controllers\Dashboard;
4
5 use App\Http\Controllers\Controller;
6 use App\Http\Controllers\DashboardController;
7 use App\Http\Requests\Dashboard\ProductStoreRequest;
8 use App\Models\Category;
9 use App\Models\Product;
10 use Illuminate\Http\Request;
11 use Illuminate\Http\Response;
12 use PhpParser\Node\Stmt\TryCatch;
13
14 class ProductController extends DashboardController
15 {
16     /**
17      * Display a listing of the resource.
18      */
19     public function index()
20     {
21         $this->setTitle('Products');
22
23         $this->addBreadcrumb('Dashboard', route('dashboard.index'));
24         $this->addBreadcrumb('Products');
25         $this->data['products'] = Product::with('category')->get();
26         $this->data['categories'] = Category::all();
27         return view('dashboard.product.index', $this->data);
28     }
29
30 }
```

The bottom of the image shows the 'TERMINAL' panel with the following output:

```
2024-11-30 16:31:35 /libs/fontawesome-free/webfonts/fa-solid-900.woff2 ..... ~ 1s
2024-11-30 16:31:35 /js/demo/chart-area-demo.js ..... ~ 1s
2024-11-30 16:31:35 /img/baju.png ..... ~ 1s
2024-11-30 16:31:37 /favicon.ico ..... ~ 0s
```

The status bar at the bottom indicates the file is 'PHP', 'Go Live', '8.1', 'Readable: Enabled', and 'Prettier'.

2. Lalu membuat routes web.php sebagai berikut :

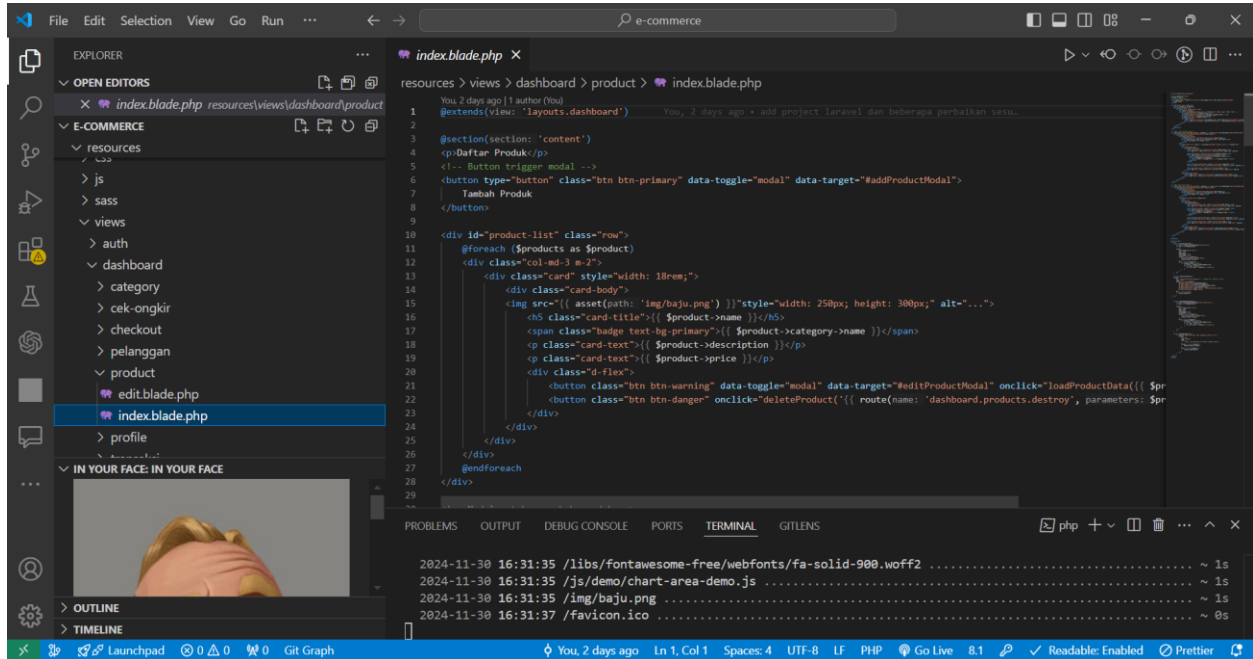
Setelah itu routes akan mengarahkan user ke halaman view yang telah kita siapkan sebagai tampilan user sebagai pengelolaan data produk

```
routes > web.php
31 Route::prefix('dashboard')->middleware(['auth', 'profile.check'])->name('dashboard.')->group(function () {
47     Route::resource('products', ProductController::class)->names('products');
48
49     Route::redirect('test', 'products');
50 });
51
52 Route::resource('dashboard/profile', ProfileController::class)->only(['index', 'update']);
53
54 Route::get('home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
55
56 // Route group dengan prefix 'admin'
57 Route::prefix('admin')->group(function () {
58     Route::get('kontakkami', function () {
59         return view('dashboard.kontakkami');
60     })->name('kontakkami');
61 });
62
63 Route::get('parse-data/{nama_lengkap}/{email}/{jenis_kelamin}', [ParsingDataController::class, 'parseData'])
64     ->name('parse.data');
65
66 Route::get('kategori', function () {
67     return view('kategori');
68 })->name('kategori');
69
70 Route::get('transaksi', function () {
71     return view('transaksi');
72 });
73
74
75
```

2024-11-30 16:31:35 /libs/fontawesome-free/webfonts/fa-solid-900.woff2 ~ 1s
2024-11-30 16:31:35 /js/demo/chart-area-demo.js ~ 1s
2024-11-30 16:31:35 /img/baju.png ~ 1s
2024-11-30 16:31:37 /favicon.ico ~ 0s

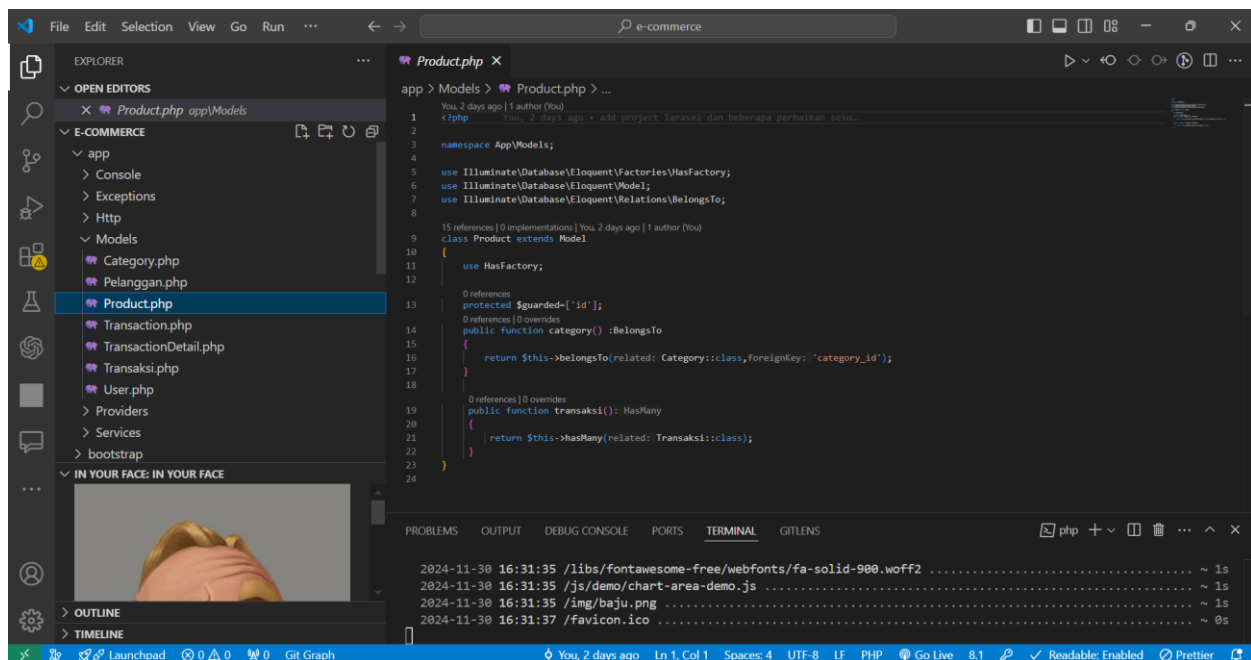
3. Setelah itu kita buat view dari tampilan untuk pengelolaan produk sebagai berikut :

Dibawah ini adalah screenshoot kode pembuatan view untuk pengelolaan produk dan selanjutnya akan dibuatkan kode untuk validasi ketika dilakukan crud produk yaitu pada bagian model dari produk tsb



```
resources > views > dashboard > product > index.blade.php
1  You 2 days ago | 1 author (You)
2  @extends('view: layouts.dashboard')
3
4  @section('content')
5  <p>Daftar Produk</p>
6  <!-- Button trigger modal -->
7  <button type="button" class="btn btn-primary" data-toggle="modal" data-target="#addProductModal">
8  Tambah Produk
9  </button>
10
11 <div id="product-list" class="row">
12   @foreach ($products as $product)
13   <div class="col-md-3">
14     <div class="card" style="width: 18rem;">
15       <div class="card-body">
16         
17         <h5 class="card-title">{{ $product->name }}</h5>
18         <p class="card-text">{{ $product->category->name }}</p>
19         <p class="card-text">{{ $product->description }}</p>
20         <p class="card-text">{{ $product->price }}</p>
21         <div class="d-flex">
22           <button class="btn btn-warning" data-toggle="modal" data-target="#editProductModal" onclick="loadProductData('{{ $pr
23           <button class="btn btn-danger" onclick="deleteProduct('{{ route('name: 'dashboard.products.destroy', parameters: $pr
24         </div>
25       </div>
26     </div>
27   @endforeach
28 </div>
29
```

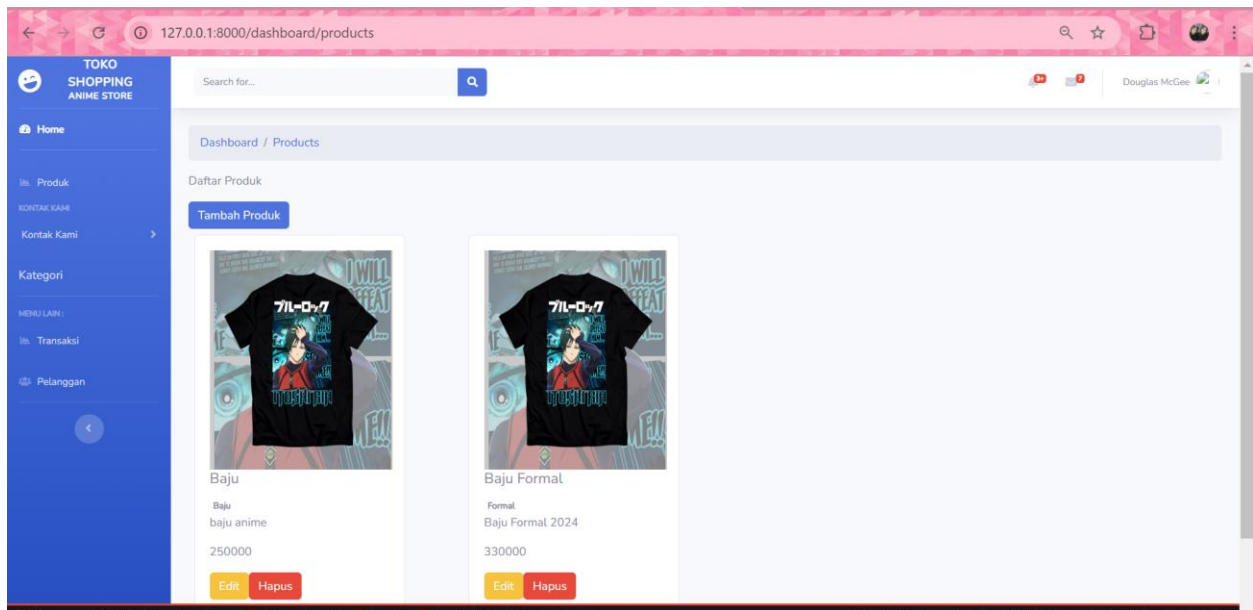
4. Pada bagian model produk dibawah ini berfungsi sebagai validasi yg dilakukan untuk melakukan pengelolaan crud pada produk



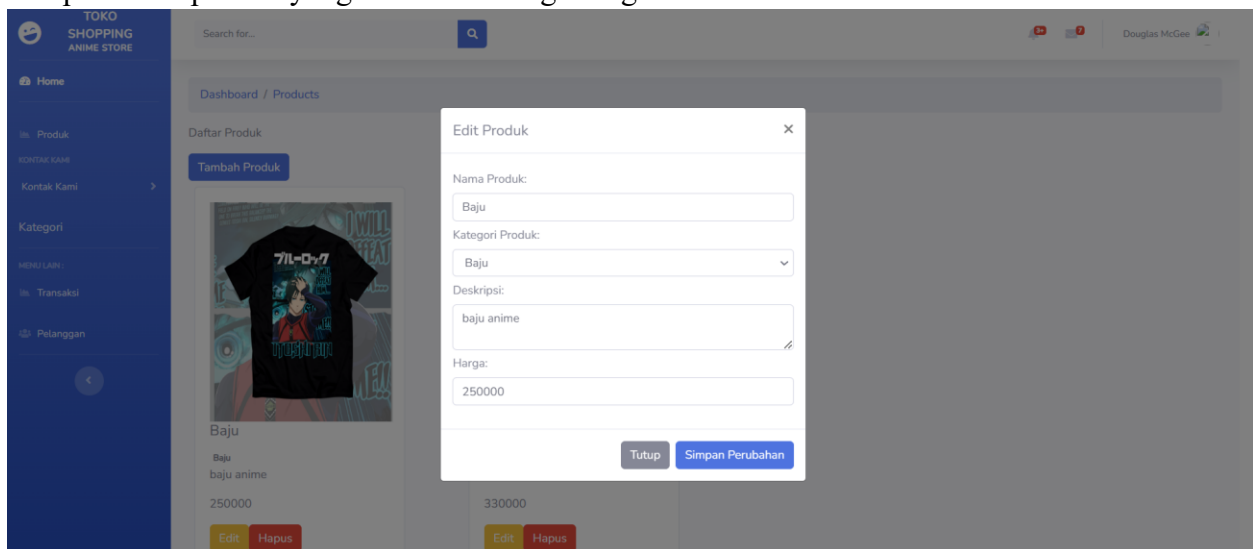
```
app > Models > Product.php > ...
1  You 2 days ago | 1 author (You)
2  <?php
3
4  namespace App\Models;
5
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Database\Eloquent\Model;
8  use Illuminate\Database\Eloquent\Relations\BelongsTo;
9
10 class Product extends Model
11 {
12     use HasFactory;
13
14     0 references | 0 guarded-['id'];
15     0 references | 0 overrides
16     public function category(): BelongsTo
17     {
18         return $this->belongsTo(related: Category::class, foreignKey: 'category_id');
19     }
20
21     0 references | 0 overrides
22     public function transaksi(): HasMany
23     {
24         return $this->hasMany(related: Transaksi::class);
25     }
26 }
```

5. Tampilan crud produk dan edit produk yang telah dibuat sebagai berikut :

Tampilan menampilkan produk setelah ditambah produk



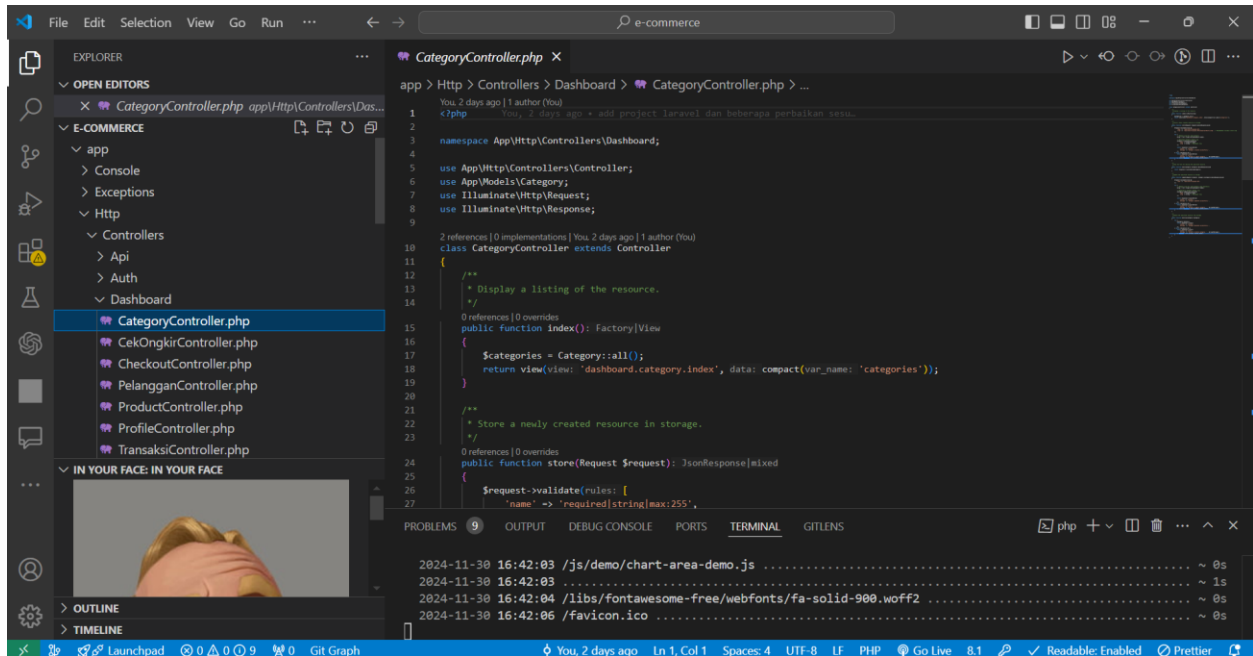
Tampilan edit produk yang telah dirancang sebagai berikut :



6. Membuat Halaman Pengelolaan Kategori

Pertama dimulai dari category controller sebagai berikut :

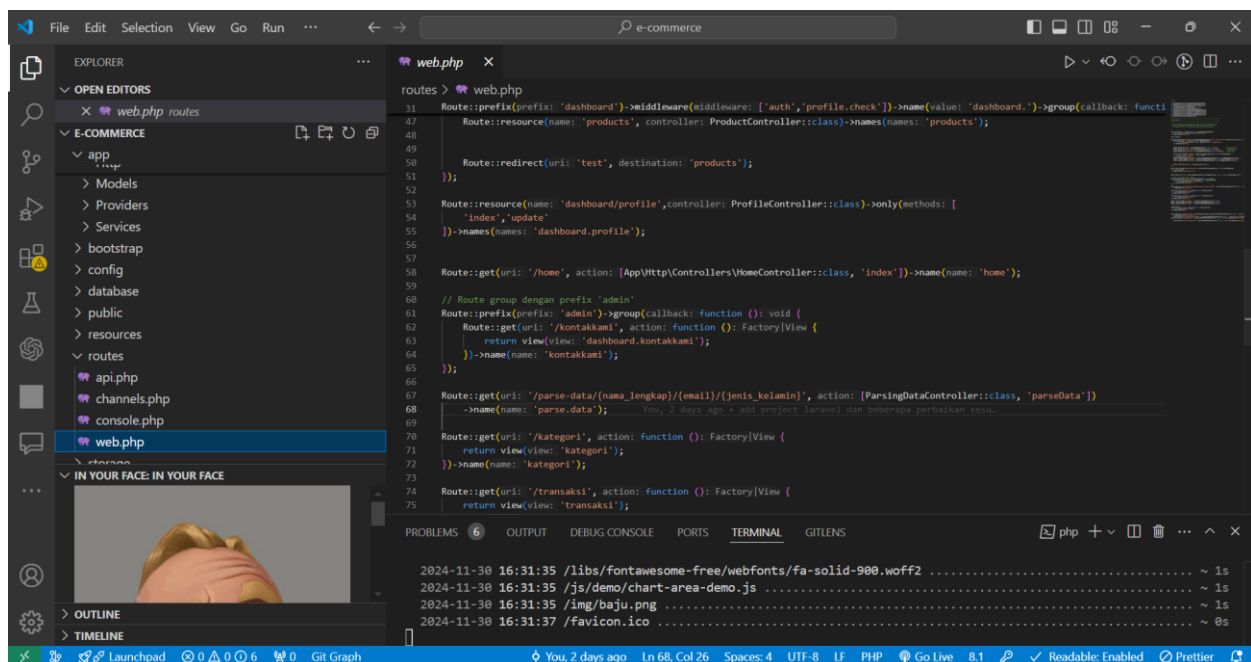
Karna fungsi nya laravel akan akses terlebih dahulu controller yg di request user dan lalu laravel akan mengarahkan user ke routes nya



```
CategoryController.php
app > Http > Controllers > Dashboard > CategoryController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Dashboard;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Category;
7 use Illuminate\Http\Request;
8 use Illuminate\Http\Response;
9
10
11 class CategoryController extends Controller
12 {
13     /**
14      * Display a listing of the resource.
15      */
16     public function index(): Factory\View
17     {
18         $categories = Category::all();
19         return view('dashboard.category.index', data: compact('categories'));
20     }
21
22     /**
23      * Store a newly created resource in storage.
24      */
25     public function store(Request $request): JsonResponse|mixed
26     {
27         $request->validate(rules: [
28             'name' => 'required|string|max:255',
29         ]);
30     }
31 }
```

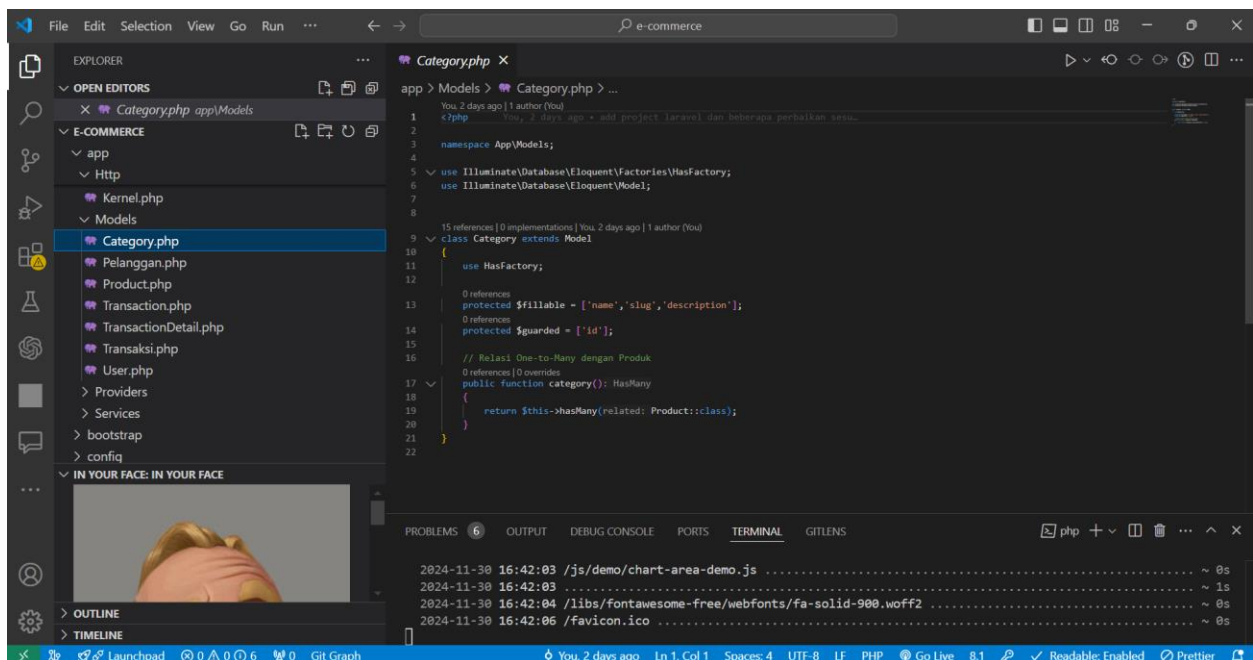
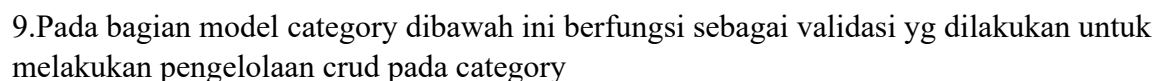
7. Lalu membuat routes web.php sebagai berikut :

Setelah itu routes akan mengarahkan user ke halaman view yang telah kita siapkan sebagai tampilan user sebagai pengelolaan data category



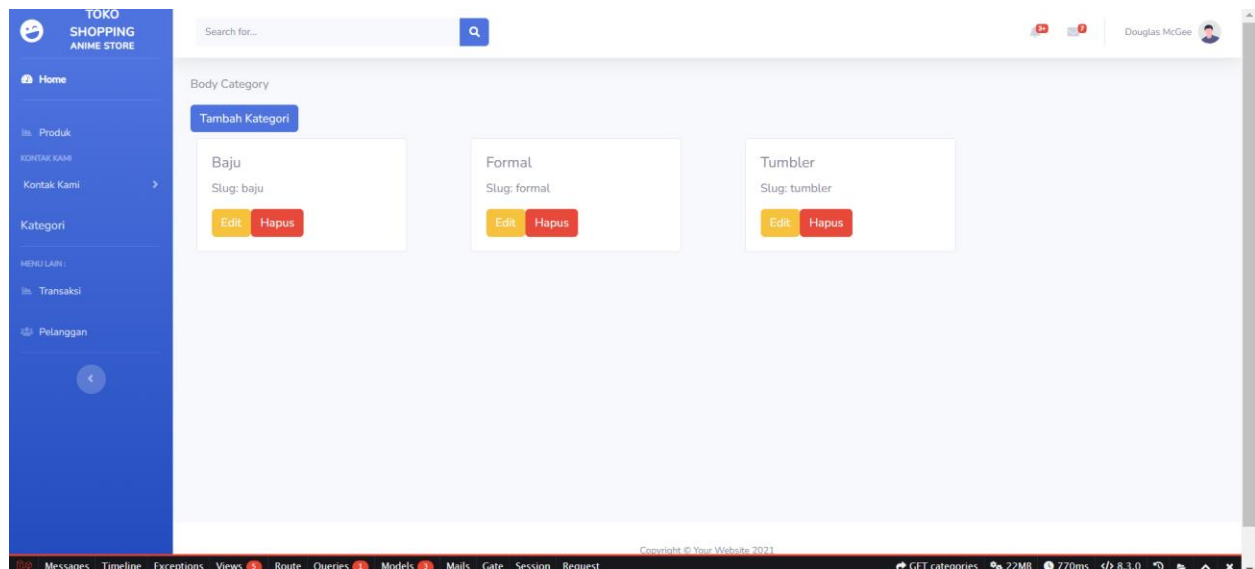
```
routes > web.php
31 Route::prefix('dashboard')->middleware(middleware: ['auth','profile.check'])->name(value: 'dashboard')->group(callback: function () {
32     Route::resource(name: 'products', controller: ProductController::class)->names(names: 'products');
33
34     Route::redirect(uri: 'test', destination: 'products');
35 });
36
37 Route::resource(name: 'dashboard/profile', controller: ProfileController::class)->only(methods: [
38     'index','update'
39 ])->names(names: 'dashboard-profile');
40
41 Route::get(uri: '/home', action: [App\Http\Controllers\HomeController::class, 'index'])->name(name: 'home');
42
43 // Route group dengan prefix 'admin'
44 Route::prefix(prefix: 'admin')->group(callback: function () { void {
45     Route::get(uri: '/kontakami', action: function () { Factory\View {
46         return view('dashboard.kontakami');
47     } }->name(name: 'kontakami');
48 });
49
50 Route::get(uri: '/parse-data/{nama_lengkap}/{email}/{jenis_kelamin}', action: [ParsingDataController::class, 'parseData'])
51     ->name(name: 'parse.data');
52
53 Route::get(uri: '/kategori', action: function () { Factory\View {
54     return view('kategori');
55 } }->name(name: 'kategori');
56
57 Route::get(uri: '/transaksi', action: function () { Factory\View {
58     return view('transaksi');
59 } });
60 }
```

Dibawah ini adalah screenshoot kode pembuatan view untuk pengelolaan category dan selanjutnya akan dibuatkan kode untuk validasi ketika dilakukan crud category yaitu pada bagian model dari category tsb



10. Tampilan crud category dan edit category yang telah dibuat sebagai berikut :

Tampilan menampilkan category setelah ditambah category



Tampilan edit produk yang telah dirancang sebagai berikut :

