Jobsheet 11

Pemrograman Web Lanjut - Minggu 13

DISUSUN OLEH:

NAMA	: Ferdi Riansyah Ramadhani Kusuma
NIM	: 2241720264
JURUSAN	: TEKNOLOGI INFORMASI
KELAS/ ABSEN	: INFORMATIKA - 2H

PROGAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

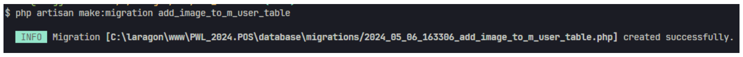
2024

A. ELOQUENT ACCESSOR

Praktikum 1 – Implementasi Eloquent Accessor

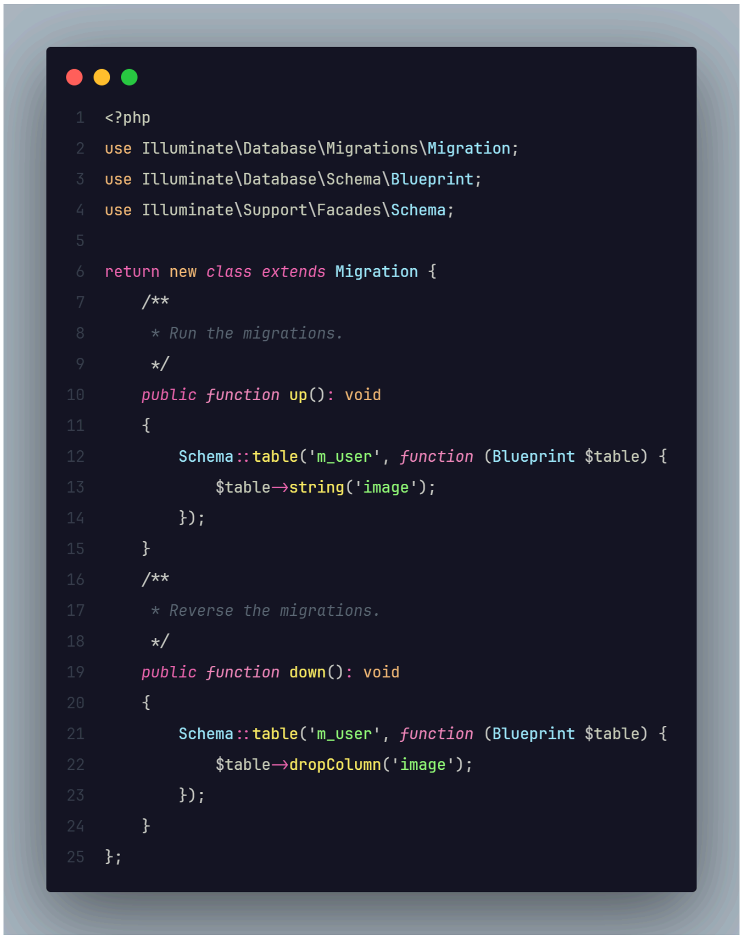
1. Sebelum memulai pastikan REST API, terlebih dahulu pastikan sudah ter instal aplikasi Postman.

2. Kita akan memodifikasi Table m_user dengan menambahkan column : image, buka terminal lalu ketikkan php artisan make:migration add_image_to_m_user_table



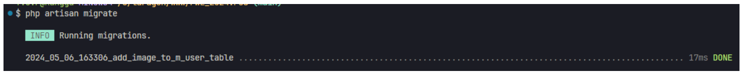
```
$ php artisan make:migration add_image_to_m_user_table

  INFO  Migration [C:\laragon\www\PWL_2024.POS\database\migrations/2024_05_06_163306_add_image_to_m_user_table.php] created successfully.
```

3. Buka file migrasi tersebut lalu modifikasi seperti ini lalu simpan:

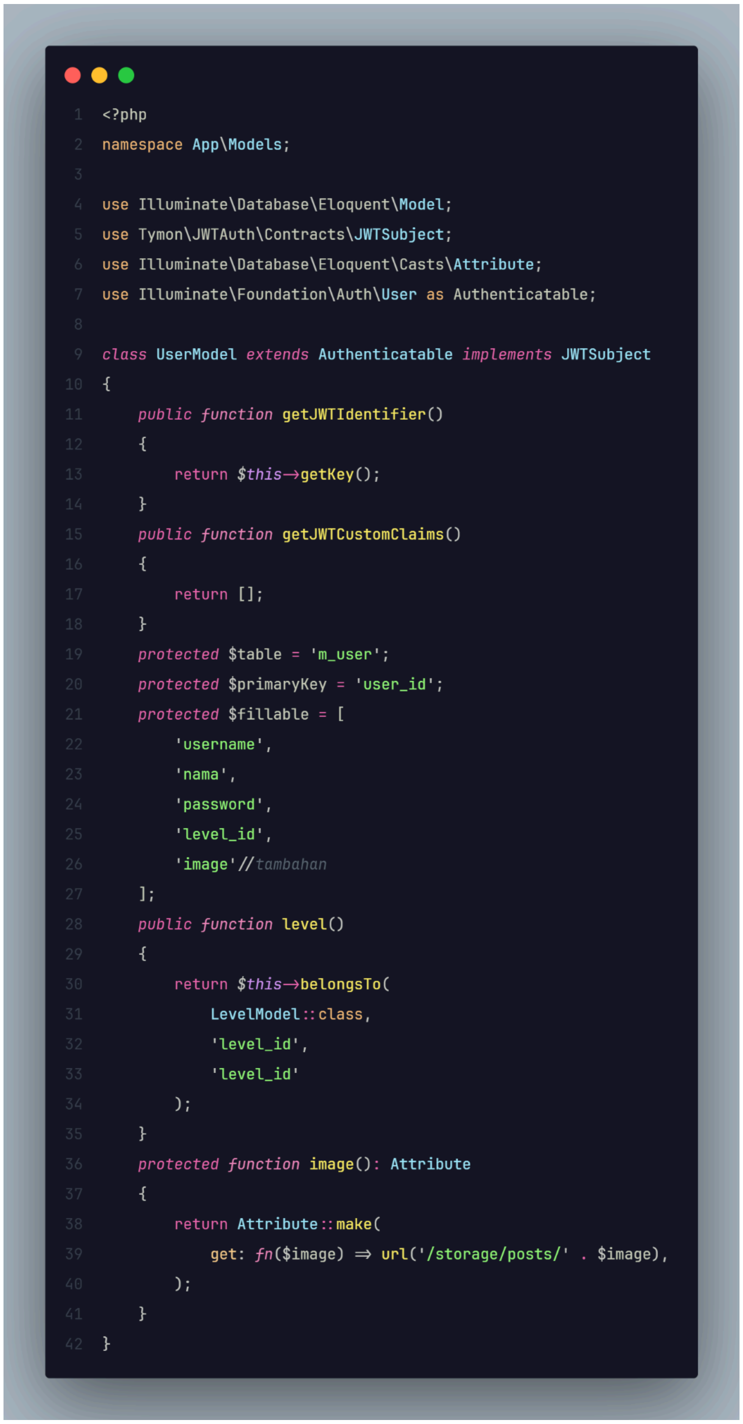


```php
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('m_user', function (Blueprint $table) {
            $table->string('image');
        });
    }
    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('m_user', function (Blueprint $table) {
            $table->dropColumn('image');
        });
    }
};
```

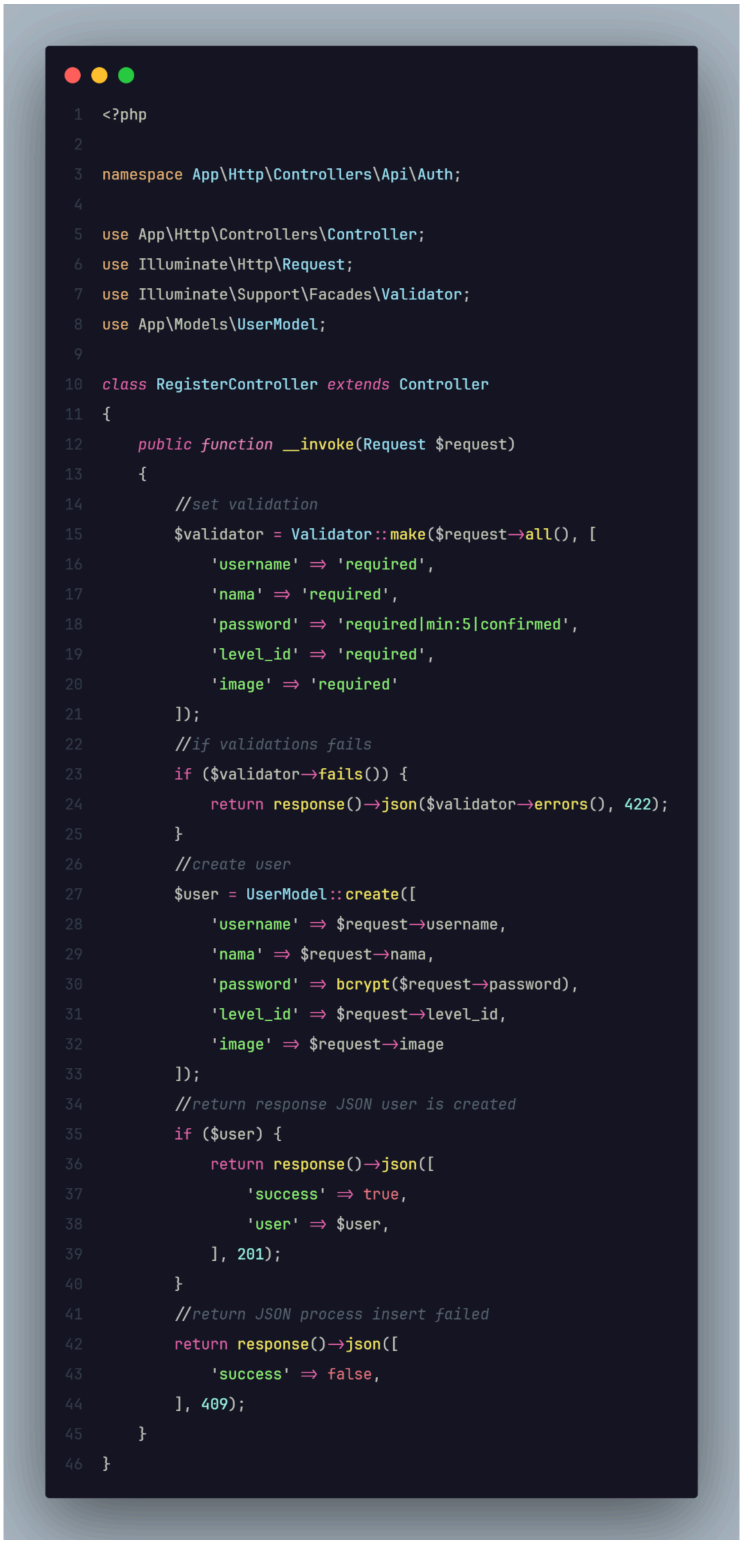
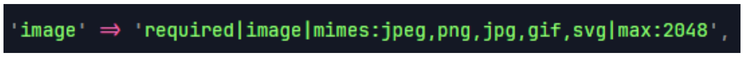
4. Lakukan jalankan update migrasi dengan cara: php artisan migrate



```
$ php artisan migrate

  INFO  Running migrations.

  2024_05_06_163306_add_image_to_m_user_table ................................................................ 17ms DONE
```

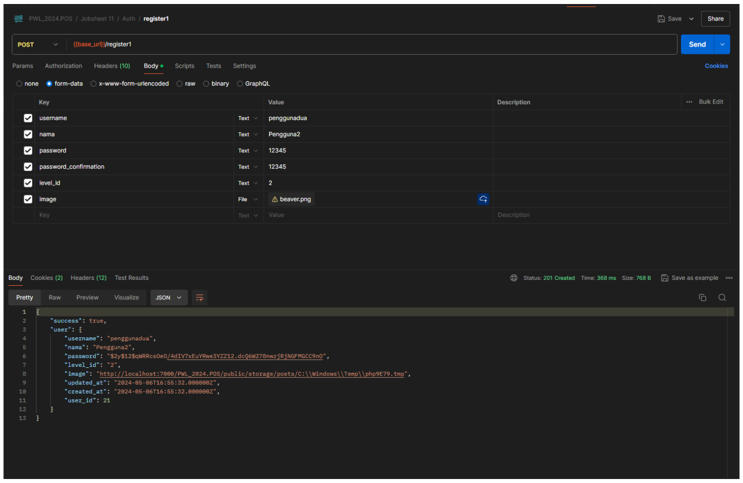
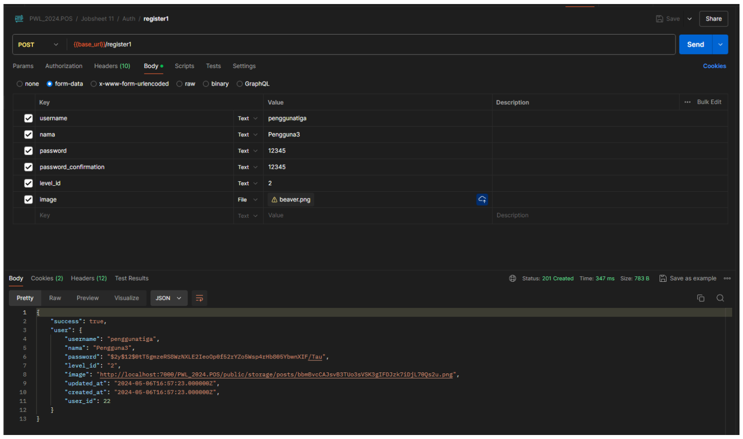
5.	Lalu lakukan modifikasi models pada App/Models/UserModel.php	

```php
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Tymon\JWTAuth\Contracts\JWTSubject;
use Illuminate\Database\Eloquent\Casts\Attribute;
use Illuminate\Foundation\Auth\User as Authenticatable;

class UserModel extends Authenticatable implements JWTSubject
{
    public function getJWTIdentifier()
    {
        return $this->getKey();
    }
    public function getJWTCustomClaims()
    {
        return [];
    }
    protected $table = 'm_user';
    protected $primaryKey = 'user_id';
    protected $fillable = [
        'username',
        'nama',
        'password',
        'level_id',
        'image'//tambahan
    ];
    public function level()
    {
        return $this->belongsTo(
            LevelModel::class,
            'level_id',
            'level_id'
        );
    }
    protected function image(): Attribute
    {
        return Attribute::make(
            get: fn($image) => url('/storage/posts/' . $image),
        );
    }
}
```

6.	Lakukan modifikasi pada Controllers/Api/RegisterController	
7.	Anda dapat menambahkan detail untuk spesifikasi image pada validator	

```php
<?php

namespace App\Http\Controllers\Api\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;
use App\Models\UserModel;

class RegisterController extends Controller
{
    public function __invoke(Request $request)
    {
        //set validation
        $validator = Validator::make($request->all(), [
            'username' => 'required',
            'nama' => 'required',
            'password' => 'required|min:5|confirmed',
            'level_id' => 'required',
            'image' => 'required'
        ]);
        //if validations fails
        if ($validator->fails()) {
            return response()->json($validator->errors(), 422);
        }
        //create user
        $user = UserModel::create([
            'username' => $request->username,
            'nama' => $request->nama,
            'password' => bcrypt($request->password),
            'level_id' => $request->level_id,
            'image' => $request->image
        ]);
        //return response JSON user is created
        if ($user) {
            return response()->json([
                'success' => true,
                'user' => $user,
            ], 201);
        }
        //return JSON process insert failed
        return response()->json([
            'success' => false,
        ], 409);
    }
}
```

```php
'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
```

8.	Ubah atau tambahkan register1 pada routes/api.php	`Route::post('/register1', App\Http\Controllers\Api\Auth\RegisterController::class)->name('register1');`
9.	Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar http://127.0.0.1:8000/api/register1 dengan method POST dan klik send	
10.	Pada Controllers/Api/RegisterController bagian create user ganti dengan	
11.	Uji coba dan screenshot hasilnya apa perbedaan dari yang sebelumnya	Hasil dari image field berubah dari yang sebelumnya adalah raw path dari temp directory menjadi hashname dari image yang telah diupload.

POST {{base_url}}/register1

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value	Description
☑	username	Text	penggunadua	
☑	nama	Text	Pengguna2	
☑	password	Text	12345	
☑	password_confirmation	Text	12345	
☑	level_id	Text	2	
☑	image	File	⚠ beaver.png	
	Key	Text	Value	Description

Status: 201 Created Time: 368 ms Size: 768 B

```
{
    "success": true,
    "user": {
        "username": "penggunadua",
        "nama": "Pengguna2",
        "password": "$2y$12$qWRRcsOeO/4dIV7xEuYRwe3YZZ12.dcQ6W278nwzjRjNGFMGCC9nO",
        "level_id": "2",
        "image": "http://localhost:7000/PWL_2024.POS/public/storage/posts/C:\\Windows\\Temp\\php9E79.tmp",
        "updated_at": "2024-05-06T16:55:32.000000Z",
        "created_at": "2024-05-06T16:55:32.000000Z",
        "user_id": 21
    }
}
```

POST {{base_url}}/register1

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

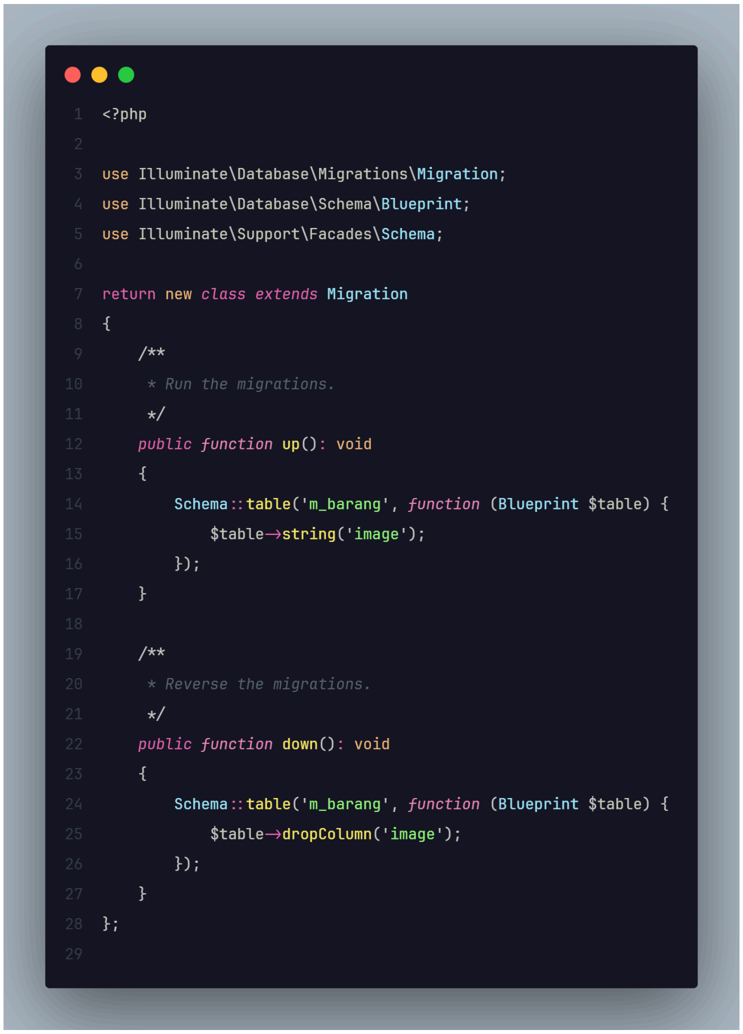
	Key		Value	Description
☑	username	Text	penggunatiga	
☑	nama	Text	Pengguna3	
☑	password	Text	12345	
☑	password_confirmation	Text	12345	
☑	level_id	Text	2	
☑	image	File	⚠ beaver.png	
	Key	Text	Value	Description

Status: 201 Created Time: 347 ms Size: 783 B

```
{
    "success": true,
    "user": {
        "username": "penggunatiga",
        "nama": "Pengguna3",
        "password": "$2y$12$0tT5gmzeRS8WzNXLE2IeoOp0f52rYZo5Wsp4rHb805YbwnXIF/Tau",
        "level_id": "2",
        "image": "http://localhost:7000/PWL_2024.POS/public/storage/posts/bbmBvcCAJsvB3TUo3sVSK3gIFDJzk7iDjL70Qs2u.png",
        "updated_at": "2024-05-06T16:57:23.000000Z",
        "created_at": "2024-05-06T16:57:23.000000Z",
        "user_id": 22
    }
}
```

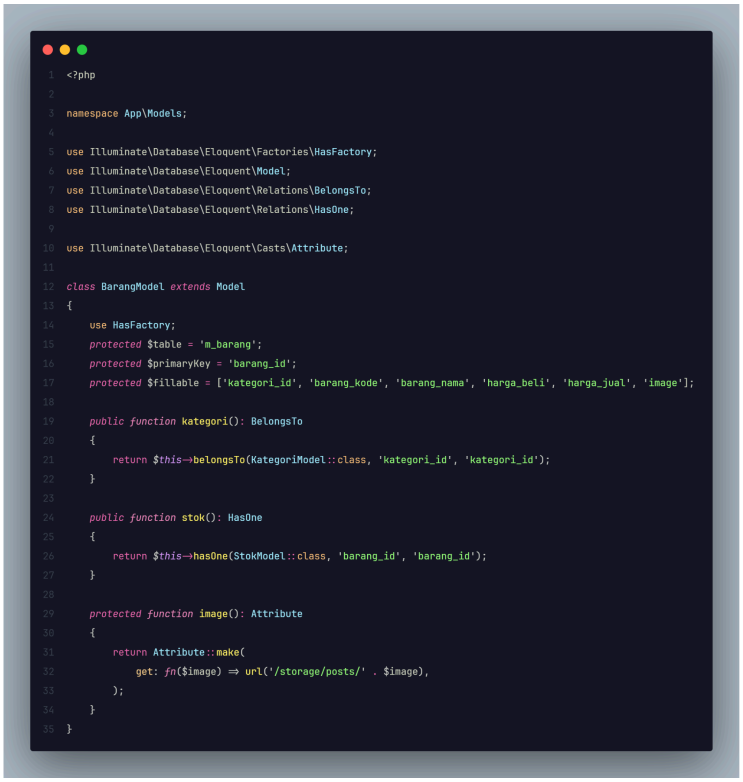
Tugas

Implementasikan API untuk upload file/gambar pada tabel lainnya yaitu tabel m_barang dan gunakan pada transaksi. Uji coba dengan method GET untuk memanggil data yang sudah di inputkan.

1.	Migrasi	

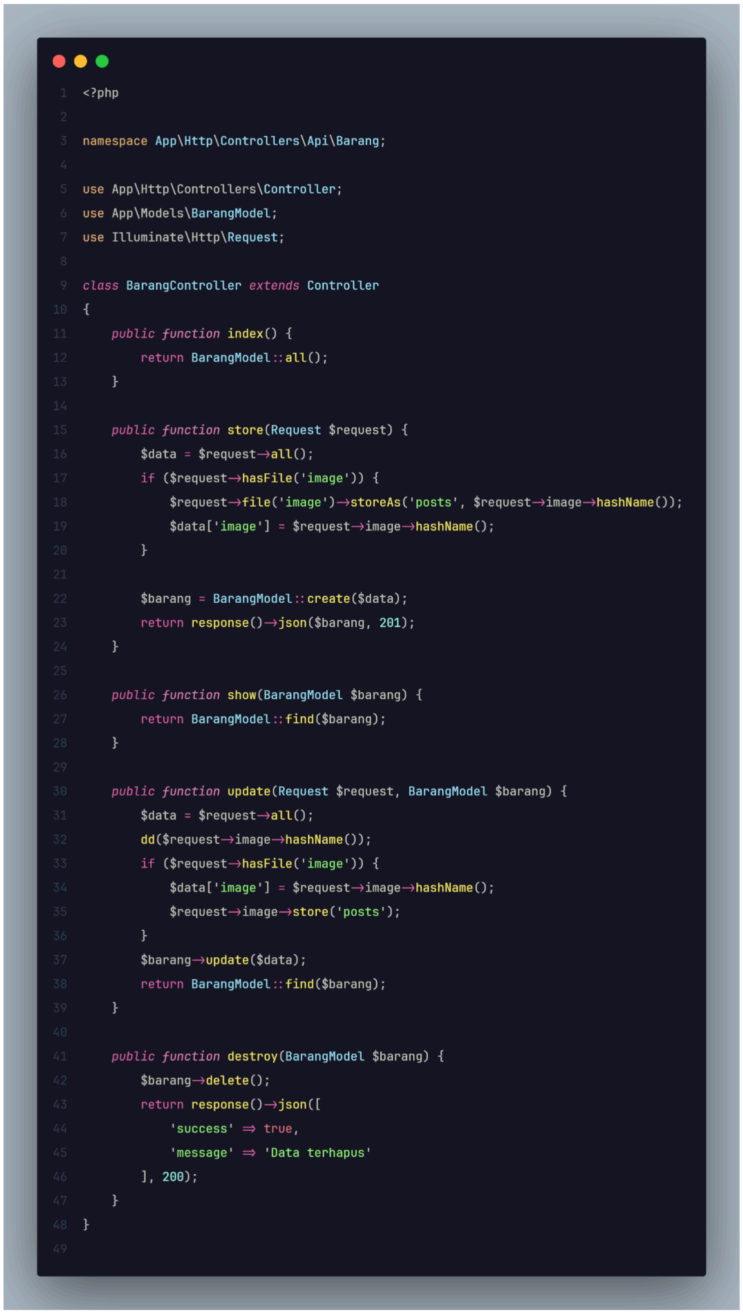
```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('m_barang', function (Blueprint $table) {
            $table->string('image');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('m_barang', function (Blueprint $table) {
            $table->dropColumn('image');
        });
    }
};
```

<table>
<tr>
<td>2.</td>
<td>BarangModel.php</td>
<td>

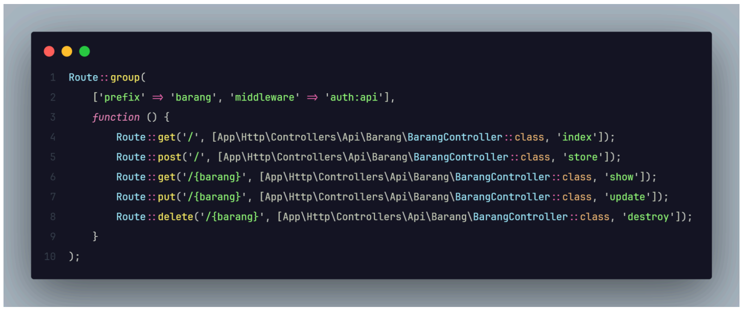


```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\HasOne;

use Illuminate\Database\Eloquent\Casts\Attribute;

class BarangModel extends Model
{
    use HasFactory;
    protected $table = 'm_barang';
    protected $primaryKey = 'barang_id';
    protected $fillable = ['kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual', 'image'];

    public function kategori(): BelongsTo
    {
        return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
    }

    public function stok(): HasOne
    {
        return $this->hasOne(StokModel::class, 'barang_id', 'barang_id');
    }

    protected function image(): Attribute
    {
        return Attribute::make(
            get: fn($image) => url('/storage/posts/' . $image),
        );
    }
}
```

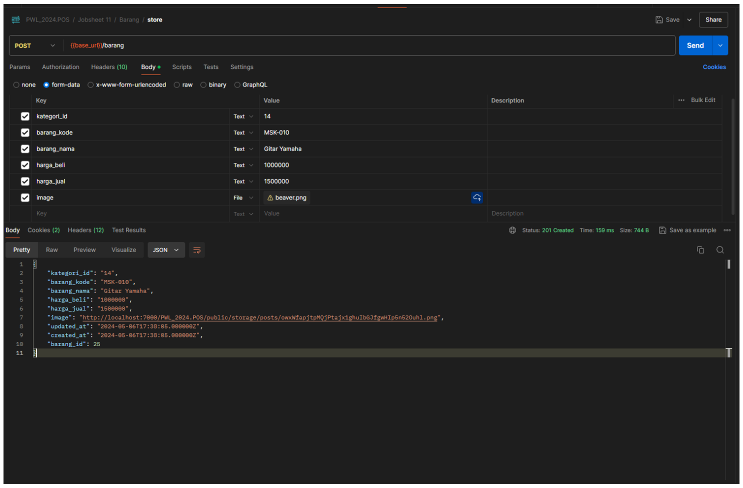
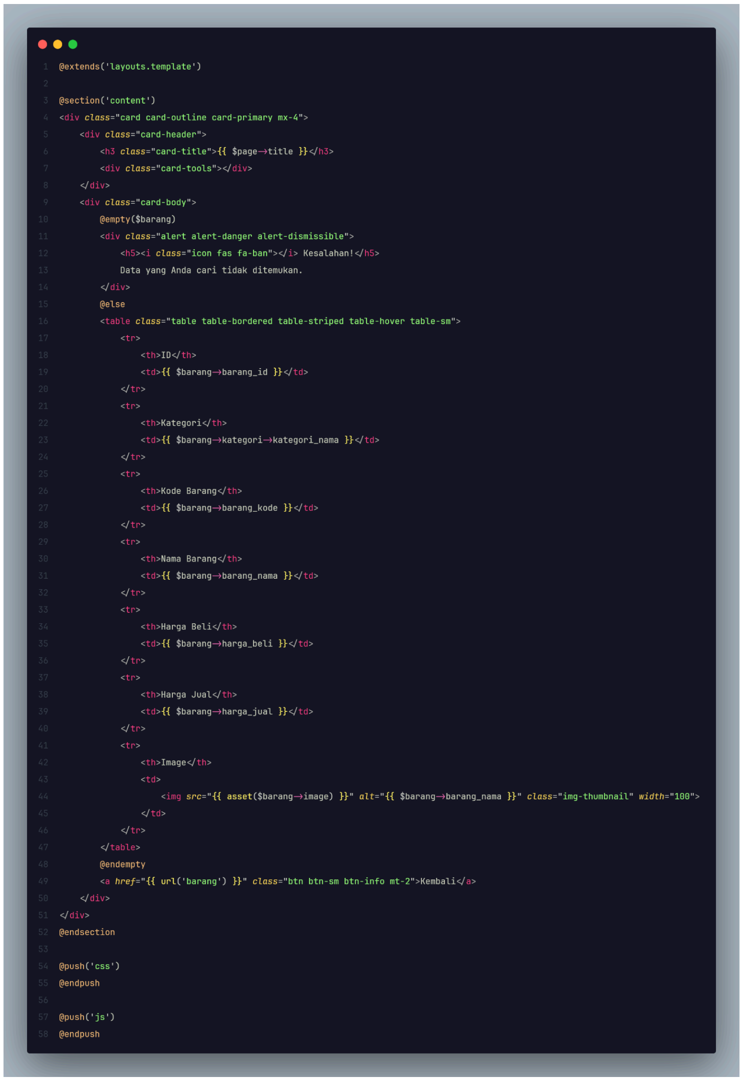
</td>
</tr>
</table>

3.	BarangController.php	

```php
<?php

namespace App\Http\Controllers\Api\Barang;

use App\Http\Controllers\Controller;
use App\Models\BarangModel;
use Illuminate\Http\Request;

class BarangController extends Controller
{
    public function index() {
        return BarangModel::all();
    }

    public function store(Request $request) {
        $data = $request->all();
        if ($request->hasFile('image')) {
            $request->file('image')->storeAs('posts', $request->image->hashName());
            $data['image'] = $request->image->hashName();
        }

        $barang = BarangModel::create($data);
        return response()->json($barang, 201);
    }

    public function show(BarangModel $barang) {
        return BarangModel::find($barang);
    }

    public function update(Request $request, BarangModel $barang) {
        $data = $request->all();
        dd($request->image->hashName());
        if ($request->hasFile('image')) {
            $data['image'] = $request->image->hashName();
            $request->image->store('posts');
        }
        $barang->update($data);
        return BarangModel::find($barang);
    }

    public function destroy(BarangModel $barang) {
        $barang->delete();
        return response()->json([
            'success' => true,
            'message' => 'Data terhapus'
        ], 200);
    }
}

```

4.	API Routes	

```php
Route::group(
    ['prefix' => 'barang', 'middleware' => 'auth:api'],
    function () {
        Route::get('/', [App\Http\Controllers\Api\Barang\BarangController::class, 'index']);
        Route::post('/', [App\Http\Controllers\Api\Barang\BarangController::class, 'store']);
        Route::get('/{barang}', [App\Http\Controllers\Api\Barang\BarangController::class, 'show']);
        Route::put('/{barang}', [App\Http\Controllers\Api\Barang\BarangController::class, 'update']);
        Route::delete('/{barang}', [App\Http\Controllers\Api\Barang\BarangController::class, 'destroy']);
    }
);
```

5.	Tambah Barang Baru dengan Image	
6.	Kode detail barang	

POST {{base_url}}/barang

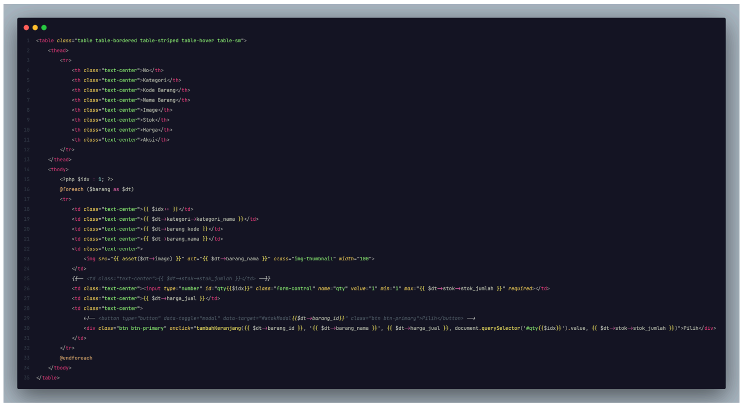
○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value	Description
☑	kategori_id	Text	14	
☑	barang_kode	Text	MSK-010	
☑	barang_nama	Text	Gitar Yamaha	
☑	harga_beli	Text	1000000	
☑	harga_jual	Text	1500000	
☑	image	File	⚠ beaver.png	
	Key	Text	Value	Description

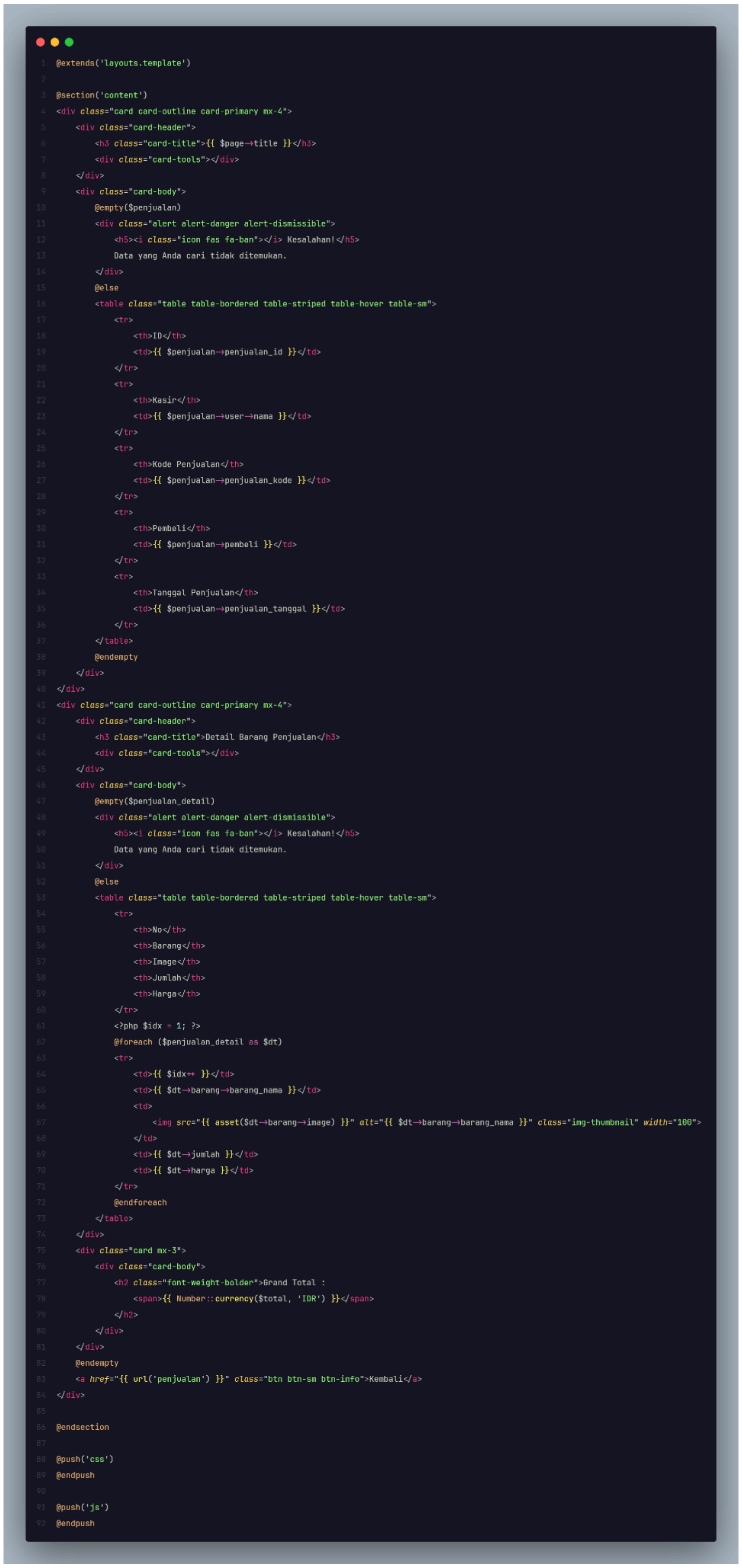
Status: 201 Created Time: 159 ms Size: 744 B

```
{
    "kategori_id": "14",
    "barang_kode": "MSK-010",
    "barang_nama": "Gitar Yamaha",
    "harga_beli": "1000000",
    "harga_jual": "1500000",
    "image": "http://localhost:7000/PWL_2024.POS/public/storage/posts/owxWfapjtpMQjPtajx1ghuIbGJfgwHIp5n520uhl.png",
    "updated_at": "2024-05-06T17:38:05.000000Z",
    "created_at": "2024-05-06T17:38:05.000000Z",
    "barang_id": 25
}
```

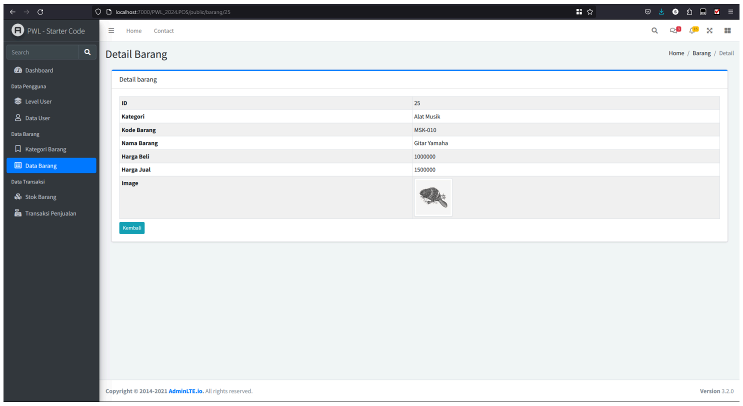
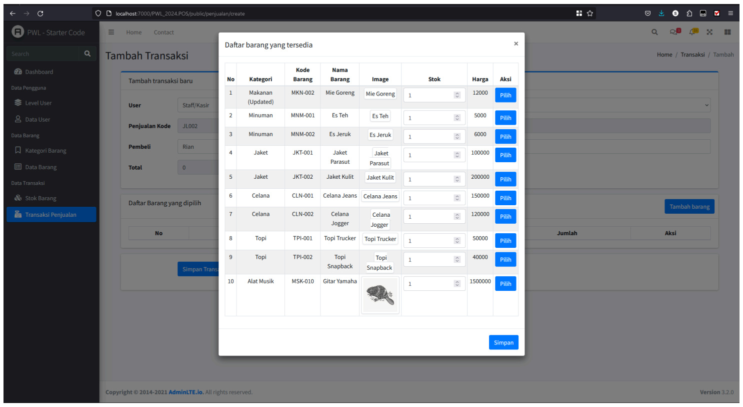
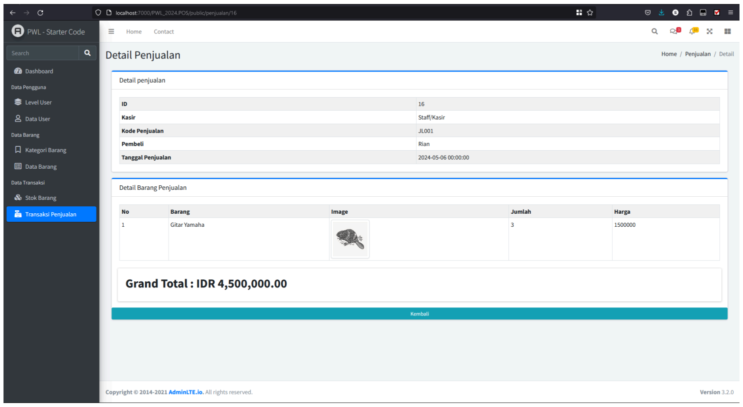
```
@extends('layouts.template')

@section('content')
<div class="card card-outline card-primary mx-4">
    <div class="card-header">
        <h3 class="card-title">{{ $page->title }}</h3>
        <div class="card-tools"></div>
    </div>
    <div class="card-body">
        @empty($barang)
        <div class="alert alert-danger alert-dismissible">
            <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
            Data yang Anda cari tidak ditemukan.
        </div>
        @else
        <table class="table table-bordered table-striped table-hover table-sm">
            <tr>
                <th>ID</th>
                <td>{{ $barang->barang_id }}</td>
            </tr>
            <tr>
                <th>Kategori</th>
                <td>{{ $barang->kategori->kategori_nama }}</td>
            </tr>
            <tr>
                <th>Kode Barang</th>
                <td>{{ $barang->barang_kode }}</td>
            </tr>
            <tr>
                <th>Nama Barang</th>
                <td>{{ $barang->barang_nama }}</td>
            </tr>
            <tr>
                <th>Harga Beli</th>
                <td>{{ $barang->harga_beli }}</td>
            </tr>
            <tr>
                <th>Harga Jual</th>
                <td>{{ $barang->harga_jual }}</td>
            </tr>
            <tr>
                <th>Image</th>
                <td>
                    <img src="{{ asset($barang->image) }}" alt="{{ $barang->barang_nama }}" class="img-thumbnail" width="100">
                </td>
            </tr>
        </table>
        @endempty
        <a href="{{ url('barang') }}" class="btn btn-sm btn-info mt-2">Kembali</a>
    </div>
</div>
@endsection

@push('css')
@endpush

@push('js')
@endpush
```

7.	Kode tabel halaman transaksi	

```
<table class="table table-bordered table-striped table-hover table-sm">
    <thead>
        <tr>
            <th class="text-center">No</th>
            <th class="text-center">Kategori</th>
            <th class="text-center">Kode Barang</th>
            <th class="text-center">Nama Barang</th>
            <th class="text-center">Image</th>
            <th class="text-center">Stok</th>
            <th class="text-center">Harga</th>
            <th class="text-center">Aksi</th>
        </tr>
    </thead>
    <tbody>
        <?php $idx = 1; ?>
        @foreach ($barang as $dt)
        <tr>
            <td class="text-center">{{ $idx++ }}</td>
            <td class="text-center">{{ $dt->kategori->kategori_nama }}</td>
            <td class="text-center">{{ $dt->barang_kode }}</td>
            <td class="text-center">{{ $dt->barang_nama }}</td>
            <td class="text-center">
                <img src="{{ asset($dt->image) }}" alt="{{ $dt->barang_nama }}" class="img-thumbnail" width="100">
            </td>
            {{-- <td class="text-center">{{ $dt->stok->stok_jumlah }}</td> --}}
            <td class="text-center"><input type="number" id="qty{{$idx}}" class="form-control" name="qty" value="1" min="1" max="{{ $dt->stok->stok_jumlah }}" required></td>
            <td class="text-center">{{ $dt->harga_jual }}</td>
            <td class="text-center">
                <!-- <button type="button" data-toggle="modal" data-target="#stokModal{{$dt->barang_id}}" class="btn btn-primary">Pilih</button> -->
                <div class="btn btn-primary" onclick="tambahKeranjang({{ $dt->barang_id }}, '{{ $dt->barang_nama }}', {{ $dt->harga_jual }}, document.querySelector('#qty{{$idx}}').value, {{ $dt->stok->stok_jumlah }})">Pilih</div>
            </td>
        </tr>
        @endforeach
    </tbody>
</table>
```

8.	Kode detail transaksi	

```
@extends('layouts.template')

@section('content')
<div class="card card-outline card-primary mx-4">
    <div class="card-header">
        <h3 class="card-title">{{ $page->title }}</h3>
        <div class="card-tools"></div>
    </div>
    <div class="card-body">
        @empty($penjualan)
        <div class="alert alert-danger alert-dismissible">
            <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
            Data yang Anda cari tidak ditemukan.
        </div>
        @else
        <table class="table table-bordered table-striped table-hover table-sm">
            <tr>
                <th>ID</th>
                <td>{{ $penjualan->penjualan_id }}</td>
            </tr>
            <tr>
                <th>Kasir</th>
                <td>{{ $penjualan->user->nama }}</td>
            </tr>
            <tr>
                <th>Kode Penjualan</th>
                <td>{{ $penjualan->penjualan_kode }}</td>
            </tr>
            <tr>
                <th>Pembeli</th>
                <td>{{ $penjualan->pembeli }}</td>
            </tr>
            <tr>
                <th>Tanggal Penjualan</th>
                <td>{{ $penjualan->penjualan_tanggal }}</td>
            </tr>
        </table>
        @endempty
    </div>
</div>
<div class="card card-outline card-primary mx-4">
    <div class="card-header">
        <h3 class="card-title">Detail Barang Penjualan</h3>
        <div class="card-tools"></div>
    </div>
    <div class="card-body">
        @empty($penjualan_detail)
        <div class="alert alert-danger alert-dismissible">
            <h5><i class="icon fas fa-ban"></i> Kesalahan!</h5>
            Data yang Anda cari tidak ditemukan.
        </div>
        @else
        <table class="table table-bordered table-striped table-hover table-sm">
            <tr>
                <th>No</th>
                <th>Barang</th>
                <th>Image</th>
                <th>Jumlah</th>
                <th>Harga</th>
            </tr>
            <?php $idx = 1; ?>
            @foreach ($penjualan_detail as $dt)
            <tr>
                <td>{{ $idx++ }}</td>
                <td>{{ $dt->barang->barang_nama }}</td>
                <td>
                    <img src="{{ asset($dt->barang->image) }}" alt="{{ $dt->barang->barang_nama }}" class="img-thumbnail" width="100">
                </td>
                <td>{{ $dt->jumlah }}</td>
                <td>{{ $dt->harga }}</td>
            </tr>
            @endforeach
        </table>
    </div>
    <div class="card mx-3">
        <div class="card-body">
            <h2 class="font-weight-bolder">Grand Total :
                <span>{{ Number::currency($total, 'IDR') }}</span>
            </h2>
        </div>
    </div>
    @endempty
    <a href="{{ url('penjualan') }}" class="btn btn-sm btn-info">Kembali</a>
</div>

@endsection

@push('css')
@endpush

@push('js')
@endpush
```

9.	Tampilan pada detail barang	
10.	Tampilan pada transaksi	
11.	Tampilan pada detail transaksi	

localhost:7000/PWL_2024.POS/public/barang/25

PWL - Starter Code

Detail Barang

Home / Barang / Detail

Detail barang

ID	25
Kategori	Alat Musik
Kode Barang	MSK-010
Nama Barang	Gitar Yamaha
Harga Beli	1000000
Harga Jual	1500000
Image	

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0

localhost:7000/PWL_2024.POS/public/penjualan/create

PWL - Starter Code

Tambah Transaksi

Home / Transaksi / Tambah

Tambah transaksi baru

User	Staff/Kasir
Penjualan Kode	JL002
Pembeli	Rian
Total	0

Daftar Barang yang dipilih

No		Jumlah	Aksi

Daftar barang yang tersedia

No	Kategori	Kode Barang	Nama Barang	Image	Stok	Harga	Aksi
1	Makanan (Updated)	MKN-002	Mie Goreng	Mie Goreng	1	12000	Pilih
2	Minuman	MNM-001	Es Teh	Es Teh	1	5000	Pilih
3	Minuman	MNM-002	Es Jeruk	Es Jeruk	1	6000	Pilih
4	Jaket	JKT-001	Jaket Parasut	Jaket Parasut	1	100000	Pilih
5	Jaket	JKT-002	Jaket Kulit	Jaket Kulit	1	200000	Pilih
6	Celana	CLN-001	Celana Jeans	Celana Jeans	1	150000	Pilih
7	Celana	CLN-002	Celana Jogger	Celana Jogger	1	120000	Pilih
8	Topi	TPI-001	Topi Trucker	Topi Trucker	1	50000	Pilih
9	Topi	TPI-002	Topi Snapback	Topi Snapback	1	40000	Pilih
10	Alat Musik	MSK-010	Gitar Yamaha		1	1500000	Pilih

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0

localhost:7000/PWL_2024.POS/public/penjualan/16

PWL - Starter Code

Detail Penjualan

Home / Penjualan / Detail

Detail penjualan

ID	16
Kasir	Staff/Kasir
Kode Penjualan	JL001
Pembeli	Rian
Tanggal Penjualan	2024-05-06 00:00:00

Detail Barang Penjualan

No	Barang	Image	Jumlah	Harga
1	Gitar Yamaha		3	1500000

Grand Total : IDR 4,500,000.00

Copyright © 2014-2021 AdminLTE.io. All rights reserved.

Version 3.2.0