



Mata Kuliah : Pemrograman Web Lanjut (PWL)  
Program Studi : D4 – Sistem Informasi Bisnis  
Semester : 5

Kelas : SIB  
NIM : 2241760007  
Nama : Anaradi Octa Lavechia  
Jobsheet Ke- : 11

## Laporan Jobsheet-11

### RESTFUL API 2

#### Praktikum Ke-1 Implementasi Eloquent Accessor

Langkah	Jawaban/Deskripsi
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, bukaterminal lalu ketikkan <b>php artisan make:migration add_image_to_m_user_table</b>  <pre>C:\laragon\www\POS&gt;php artisan make:migration add_image_to_m_user_table Warning: Module "gd" is already loaded in Unknown on line 0 [INFO] Migration [C:\laragon\www\POS\database\migrations\2024_11_06_070012_add_image_to_m_user_table.php] created successfully. C:\laragon\www\POS&gt;</pre>
3	Buka file migrasi tersebut lalu modifikasi seperti ini lalu simpan:



	<pre>database &gt; migrations &gt; 2024_11_06_070012_add_image_to_m_user_table.php &gt; class &gt; up 1  &lt;?php 2 3  use Illuminate\Database\Migrations\Migration; 4  use Illuminate\Database\Schema\Blueprint; 5  use Illuminate\Support\Facades\Schema; 6 7  return new class extends Migration 8  { 9      /** 10       * Run the migrations. 11       */ 12     public function up(): void 13     { 14         Schema::table('m_user', callback: function (Blueprint \$table): void { 15             \$table-&gt;string(column: 'image'); 16         }); 17     } 18 19     /** 20      * Reverse the migrations. 21      */ 22     public function down(): void 23     { 24         Schema::table('m_user', callback: function (Blueprint \$table): void { 25             \$table-&gt;dropColumn(columns: 'image'); 26         }); 27     } 28 }</pre>	
4	<p>Lakukan jalankan update migrasi dengan cara: <b>php artisan migrate</b></p> <pre>C:\laragon\www\POS&gt;php artisan migrate Warning: Module "gd" is already loaded in Unknown on line 0 [INFO] Running migrations. 2024_11_06_070012_add_image_to_m_user_table ..... 146ms DONE C:\laragon\www\POS&gt;</pre>	
5	<p>Lalu lakukan modifikasi models pada App/Models/UserModel.php</p> <pre>app &gt; Models &gt; M UserModel.php &gt; PHP migration &gt; to UserModel 1  &lt;?php 2 3  namespace App\Models; 4 5  use Illuminate\Database\Eloquent\Factories\HasFactory; 6  use Illuminate\Database\Eloquent\Model; 7  use Illuminate\Database\Eloquent\Relations\BelongsTo; 8  use Illuminate\Foundation\Auth\User as Authenticatable; // implement class Authenticatable 9  use Tymon\JWTAuth\Contracts\JWTSubject; 10 use Illuminate\Database\Eloquent\Casts\Attribute; 11 12 class UserModel extends Authenticatable implements JWTSubject 13 { 14     /** 15      * @var array&lt;int&gt; 16      */ 17     public function getJWTIdentifier(): mixed 18     { 19         return \$this-&gt;getKey(); 20     } 21 22     /** 23      * @var array&lt;int&gt; 24      */ 25     public function getJWTCustomClaims(): array 26     { 27         return []; 28     } 29 30     use HasFactory; 31 32     /** 33      * @var string 34      */ 35     protected \$table = 'm_user'; 36 37     /** 38      * @var string 39      */ 40     protected \$primaryKey = 'user_id'; 41 42     /** 43      * @var array&lt;string&gt; 44      */ 45     protected \$fillable = ['username', 'password', 'nama', 'level_id', 'avatar', 'created_at', 'updated_at']; 46 47     /** 48      * @var array&lt;string&gt; 49      */ 50     protected \$hidden = ['password']; //login di tampilan user select 51 52     /** 53      * @var array&lt;string&gt; 54      */ 55     protected \$casts = ['password' =&gt; 'hashed']; //casting password agar otomatis di hash 56 57     /** 58      * @var array&lt;string&gt; 59      */ 60     public function level(): BelongsTo 61     { 62         return \$this-&gt;belongsTo(relation: LevelModel::class, foreignKey: 'level_id', ownerKey: 'level_id'); 63     } 64 65     /** 66      * @var array&lt;string&gt; 67      */ 68     public function getFullName(): string 69     { 70         return \$this-&gt;level-&gt;level_nama; 71     } 72 73     /** 74      * @var array&lt;string&gt; 75      */ 76     public function hashRole(\$role): bool 77     { 78         return \$this-&gt;level-&gt;level_kode == \$role; 79     } 80 81     /** 82      * @var array&lt;string&gt; 83      */ 84     public function getRole(): mixed 85     { 86         return \$this-&gt;level-&gt;level_kode; 87     } 88 89     /** 90      * @var array&lt;string&gt; 91      */ 92     protected function avatar(): Attribute 93     { 94         return Attribute::make( 95             get: fn (\$avatar): string UriGenerator =&gt; url(path: "/gambar/" . \$avatar), 96         ); 97     } 98 }</pre>	



6

Lakukan modifikasi pada Controllers/Api/RegisterController

```
app > Http > Controllers > Api > RegisterController.php > PHP Intelephense > RegisterController > __invoke
1  <?php
2  namespace App\Http\Controllers\Api;
3  use App\Http\Controllers\Controller;
4  use App\Models\UserModel;
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Validator;
7  class RegisterController extends Controller
8  {
9      public function __invoke(Request $request): JsonResponse|mixed
10     {
11         $validator = Validator::make(data: $request->all(), rules: [
12             'username' => 'required',
13             'nama' => 'required',
14             'password' => 'required|min:5|confirmed',
15             'level_id' => 'required',
16             'avatar' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
17         ]);
18         if ($validator->fails()) {
19             return response()->json(data: $validator->errors(), status: 422);
20         }
21
22         $avatar = $request->avatar;
23
24         $user = UserModel::create(attributes: [
25             'username' => $request->username,
26             'nama' => $request->nama,
27             'password' => bcrypt(value: $request->password),
28             'level_id' => $request->level_id,
29             'avatar' => $avatar->hashName(),
30         ]);
31         if ($user) {
32             return response()->json(data: [
33                 'success' => true,
34                 'user' => $user,
35             ], status: 201);
36         }
37         return response()->json(data: [
38             'success' => false,
39         ], status: 409);
40     }
41 }
```

7

Anda dapat menambahkan detail untuk spesifikasi image pada validator



```
app > Http > Controllers > Api > RegisterController.php > PHP Intelephense > RegisterController > __invoke
1  <?php
2  namespace App\Http\Controllers\Api;
3  use App\Http\Controllers\Controller;
4  use App\Models\UserModel;
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Validator;
7  2 references | 0 implementations
8  class RegisterController extends Controller
9  {
10     0 references | 0 overrides
11     public function __invoke(Request $request): JsonResponse|mixed
12     {
13         $validator = Validator::make(data: $request->all(), rules: [
14             'username' => 'required',
15             'nama' => 'required',
16             'password' => 'required|min:5|confirmed',
17             'level_id' => 'required',
18             'avatar' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
19         ]);
20         if ($validator->fails()) {
21             return response()->json(data: $validator->errors(), status: 422);
22         }
23         $avatar = $request->avatar;
24         $user = UserModel::create(attributes: [
```

Ubah atau tambahkan register1 pada routes/api.php

```
routes > api.php > ...
22 Route::post(uri: '/register', action: App\Http\Controllers\Api\RegisterController::class)->name(name: 'register');
23
24 Route::post(uri: '/login', action: App\Http\Controllers\Api>LoginController::class)->name(name: 'login');
25 Route::middleware(middleware: 'auth:api')->get(uri: '/user', action: function (Request $request): mixed{
26     return $request->user();
27 });
28
29 Route::post(uri: '/logout', action: App\Http\Controllers\Api\LogoutController::class)->name(name: 'logout');
30
31 Route::post(uri: '/register1', action: App\Http\Controllers\Api\RegisterController::class)->name(name: 'register1');
```

8

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

localhost/POS/public/api/kategori?kategori\_nama=Sembako Save

POST localhost/POS/public/api/register1 Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

form-data

username	Supervisor3
nama	Supervisor Tiga
password	123456
password_confirmation	123456
level_id	9
avatar	cing.jpg X

Body 201 Created 1652 ms 660 B Save Response

Pretty Raw Preview Visualize JSON C

```
1 {
2   "success": true,
3   "user": {
4     "username": "Supervisor3",
5     "nama": "Supervisor Tiga",
6     "level_id": "9",
7     "avatar": "http://localhost/POS/public/
8     gambarnuToeFSeNcOUKcMlNKuDs8HmHRYEndIcu7Kqiy8.jpg",
9     "updated_at": "2024-11-06T07:25:06.000000Z",
10    "created_at": "2024-11-06T07:25:06.000000Z",
11    "user_id": 72
12  }
13 }
```



9

Pada Controllers/Api/RegisterController bagian create user ganti dengan

```
app > Http > Controllers > Api > RegisterController.php > PHP Intelephense > RegisterController > __invoke
1  <?php
2  namespace App\Http\Controllers\Api;
3  use App\Http\Controllers\Controller;
4  use App\Models\UserModel;
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Facades\Validator;
7  3 references | 0 implementations
8  class RegisterController extends Controller
9  {
10     0 references | 0 overrides
11     public function __invoke(Request $request): JsonResponse|mixed
12     {
13         $validator = Validator::make(data: $request->all(), rules: [
14             'username' => 'required',
15             'nama' => 'required',
16             'password' => 'required|min:5|confirmed',
17             'level_id' => 'required',
18             'avatar' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
19         ]);
20         if ($validator->fails()) {
21             return response()->json(data: $validator->errors(), status: 422);
22         }
23         $avatar = $request->avatar;
24         $user = UserModel::create(attributes: [
25             'username' => $request->username,
26             'nama' => $request->nama,
27             'password' => bcrypt(value: $request->password),
28             'level_id' => $request->level_id,
29             'avatar' => $avatar->hashName(),
30         ]);
```

10

Uji coba dan screenshot hasilnya apa perbedaan dari yang sebelumnya

[http://localhost/POS/public/api/kategori?kategori\\_nama=Sembako](http://localhost/POS/public/api/kategori?kategori_nama=Sembako) Save

**POST** localhost/POS/public/api/register1 Send

Params Auth Headers (8) **Body** Pre-req. Tests Settings Cookies

form-data

<input checked="" type="checkbox"/>	username	Supervisor3
<input checked="" type="checkbox"/>	nama	Supervisor Tiga
<input checked="" type="checkbox"/>	password	123456
<input checked="" type="checkbox"/>	password_confirmation	123456
<input checked="" type="checkbox"/>	level_id	9
<input checked="" type="checkbox"/>	avatar	cing.jpg <span>×</span>

Body 201 Created 1652 ms 660 B Save Response

Pretty Raw Preview Visualize JSON ↻ C

```
1  {
2    "success": true,
3    "user": {
4      "username": "Supervisor3",
5      "nama": "Supervisor Tiga",
6      "level_id": "9",
7      "avatar": "http://localhost/POS/public/
8      gambairuTomFSenCoUKCmLnKwDs8hwmHHYEndIcU7Kqiy8.jpg",
9      "updated_at": "2024-11-06T07:25:06.000000Z",
10     "created_at": "2024-11-06T07:25:06.000000Z",
11     "user_id": 72
12   }
13 }
```



## 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.

### Soal Praktikum m\_barang

Langkah	Jawaban/Deskripsi
1	<p>Memodifikasi Table m_barang dengan menambahkan column : image, buka terminal lalu ketikkan</p> <p><b>Php artisan make:migration add_image_to_m_user_table</b></p> <pre>C:\laragon\www\POS&gt;Php artisan make:migration add_image_to_m_user_table Warning: Module "gd" is already loaded in Unknown on line 0 INFO Migration [C:\laragon\www\POS\database\Migrations\2024_11_06_072910_add_image_to_m_user_table.php] created successfully. C:\laragon\www\POS&gt;</pre>
2	<p>Buka file migrasi tersebut lalu modifikasi seperti ini lalu simpan</p> <pre>database &gt; migrations &gt; 2024_11_06_072910_add_image_to_m_user_table.php &gt; class &gt; down &gt; Closure 1  &lt;?php 2 3  use Illuminate\Database\Migrations\Migration; 4  use Illuminate\Database\Schema\Blueprint; 5  use Illuminate\Support\Facades\Schema; 6 7  return new class extends Migration 8  { 9      /** 10       * Run the migrations. 11       */ 12       public function up(): void 13       { 14           Schema::table('m_user', callback: function (Blueprint \$table): void { 15               \$table-&gt;string(column: 'image'); 16           }); 17       } 18 19       /** 20        * Reverse the migrations. 21        */ 22       public function down(): void 23       { 24           Schema::table('m_user', callback: function (Blueprint \$table): void { 25               \$table-&gt;dropColumn(columns: 'image'); 26           }); 27       } 28  };</pre>



3	<p>Lakukan jalankan update migrasi dengan cara:</p> <p><b>php artisan migrate</b></p> <pre>C:\laragon\www\POS&gt;php artisan migrate  Warning: Module "gd" is already loaded in Unknown on line 0  [INFO] Running migrations.  2024_11_06_072910_add_image_to_m_user_table ..... 175ms DONE  C:\laragon\www\POS&gt;</pre>
4	<p>Lalu lakukan modifikasi models pada App/Models/BarangModel.php</p> <pre>app &gt; Models &gt; BarangModel.php &gt; PHP Intelephense &gt; BarangModel  1  &lt;?php 2 3  namespace App\Models; 4 5  use Illuminate\Database\Eloquent\Factories\HasFactory; 6  use Illuminate\Database\Eloquent\Model; 7  use Illuminate\Database\Eloquent\Relations\BelongsTo; 8  use Tymon\JWTAuth\Contracts\JWTSubject; 9  use Illuminate\Database\Eloquent\Casts\Attribute; 10 11  30 references   0 implementations 12  class BarangModel extends Model implements JWTSubject 13  { 14      0 references   0 overrides 15      public function getJWTIdentifier(): mixed 16      { 17          return \$this-&gt;getKey(); 18      } 19      0 references   0 overrides 20      public function getJWTCustomClaims(): array 21      { 22          return []; 23      } 24      use HasFactory; 25 26      0 references 27      protected \$table = 'm_barang'; 28 29      0 references 30      protected \$primaryKey = 'barang_id'; 31 32      0 references 33      protected \$fillable = ['barang_id', 'kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual']; 34 35      0 references   0 overrides 36      public function kategori(): BelongsTo{ 37          return \$this-&gt;belongsTo(related: kategorimodel::class, foreignKey: 'kategori_id', ownerKey: 'kategori_id'); 38      } 39  }</pre>
5	<p>Lakukan modifikasi pada Controllers/Api/BarangController</p>



	<pre>app &gt; Http &gt; Controllers &gt; Api &gt; BarangController.php &gt; PHP Intelephense &gt; BarangController &gt; __invoke 1  &lt;?php 2 3  namespace App\Http\Controllers\Api; 4 5  use App\Http\Controllers\Controller; 6  use App\Models\BarangModel; 7  use App\Models\KategoriModel; 8  use Illuminate\Http\Request; 9  use Illuminate\Support\Facades\Validator; 10 11  6 references   0 implementations 12  class BarangController extends Controller 13  { 14      0 references   0 overrides 15      public function __invoke(Request \$request): JsonResponse mixed 16      { 17          \$validator = Validator::make(data: \$request-&gt;all(), rules: [ 18              'kategori_id' =&gt; 'required integer', 19              'barang_kode' =&gt; 'required min:3 max:20 unique:m_barang,barang_kode', 20              'barang_nama' =&gt; 'required string max:100', 21              'harga_beli' =&gt; 'required numeric', 22              'harga_jual' =&gt; 'required numeric', 23              'image' =&gt; 'required image mimes:jpeg,png,jpg,gif,svg max:2048', 24          ]); 25 26          if (\$validator-&gt;fails()) { 27              return response()-&gt;json(data: \$validator-&gt;errors(), status: 422); 28          } 29 30          // Hash nama file dan simpan ke folder public/gambar 31          \$image = \$request-&gt;image; 32          \$hashedName = \$image-&gt;hashName(); 33          \$image-&gt;move(public_path('gambar'), \$hashedName); 34 35          \$user = BarangModel::create(attributes: [ 36              'kategori_id' =&gt; \$request-&gt;kategori_id, 37              'barang_kode' =&gt; \$request-&gt;barang_kode, 38              'barang_nama' =&gt; \$request-&gt;barang_nama, 39              'harga_beli' =&gt; \$request-&gt;harga_beli, 40              'harga_jual' =&gt; \$request-&gt;harga_jual, 41              'image' =&gt; \$hashedName, 42          ]); 43 44          if (\$user) { 45              return response()-&gt;json(data: [ 46                  'success' =&gt; true, 47                  'user' =&gt; \$user, 48              ], status: 201); 49          } 50 51          return response()-&gt;json(data: [ 52              'success' =&gt; false, 53          ], status: 409); 54 55      1 reference   0 overrides 56      public function show(BarangModel \$barang): array BarangModel Collection Model null 57      { 58          return BarangModel::find(id: \$barang); 59      } 60  }</pre>
6	<p>Ubah atau tambahkan barang pada routes/api.php</p> <pre>53 // m_barang 54 Route::get(uri: 'barang', action: [BarangController::class, 'index']); 55 Route::post(uri: 'barang', action: [BarangController::class, 'store']); 56 Route::get(uri: 'barang/{barang}', action: [BarangController::class, 'show']); 57 Route::put(uri: 'barang/{barang}', action: [BarangController::class, 'update']); 58 Route::delete(uri: 'barang/{barang}', action: [BarangController::class, 'destroy']); 59 Route::post(uri: '/barang', action: App\Http\Controllers\Api\BarangController::class-&gt;name(name: 'barang')); 60 Route::get(uri: 'barang/{barang}', action: [BarangController::class, 'show']); 61</pre>





7	<p>Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar</p>  <p>The screenshot shows the Postman interface for a POST request to <code>localhost/POS/public/api/barang</code>. The request is configured with form-data. The body fields are:</p> <table border="1"><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>kategori_id</td><td>4</td></tr><tr><td>barang_kode</td><td>ALF-001</td></tr><tr><td>barang_nama</td><td>Alfa Hello Kitty</td></tr><tr><td>harga_beli</td><td>13000</td></tr><tr><td>harga_jual</td><td>20000</td></tr><tr><td>image</td><td>barang.jpg</td></tr></tbody></table> <p>The response is a JSON object:</p> <pre>{  "success": true,  "user": {    "kategori_id": "4",    "barang_kode": "ALF-001",    "barang_nama": "Alfa Hello Kitty",    "harga_beli": "13000",    "harga_jual": "20000",    "updated_at": "2024-11-06T07:43:45.000000Z",    "created_at": "2024-11-06T07:43:45.000000Z",    "barang_id": 28  }}</pre>	Key	Value	kategori_id	4	barang_kode	ALF-001	barang_nama	Alfa Hello Kitty	harga_beli	13000	harga_jual	20000	image	barang.jpg
Key	Value														
kategori_id	4														
barang_kode	ALF-001														
barang_nama	Alfa Hello Kitty														
harga_beli	13000														
harga_jual	20000														
image	barang.jpg														
8	Menampilkan barang sesuai barang_id yang dipilih dengan method GET														



The screenshot shows a REST client interface. The top bar displays the URL `localhost/POS/public/api/kategori?kategori_nama=Sembako` with a 'Save' button. Below this, a dropdown menu is set to 'GET' and the path `localhost/POS/public/api/barang/28` is entered. A 'Send' button is to the right. Underneath, tabs for 'Params', 'Auth', 'Headers (8)', 'Body', 'Pre-req.', 'Tests', and 'Settings' are visible, along with a 'Cookies' link. The 'Query Params' section contains a table with two columns: 'Key' and 'Value', and a 'Bulk Edit' button. The 'Body' tab is selected, showing a JSON response with the following data: `{ "barang_id": 28, "kategori_id": 4, "barang_kode": "ALF-001", "barang_nama": "Alfa Hello Kitty", "harga_beli": 13000, "harga_jual": 20000, "created_at": "2024-11-06T07:43:45.000000Z", "updated_at": "2024-11-06T07:43:45.000000Z" }`. The status bar at the bottom indicates a 200 OK response, 754 ms latency, and 734 B body size, with a 'Save Response' button.

### Soal Praktikum t\_penjualan

Langkah	Jawaban/Deskripsi
1	<p>Memodifikasi Table m_penjualan dengan menambahkan column : image, buka terminal lalu ketikkan</p> <p><b>Php artisan make:migration add_image_to_t_penjualan_table</b></p> <pre>C:\laragon\www\POS&gt;php artisan make:migration add_image_to_t_penjualan_table Warning: Module "gd" is already loaded in Unknown on line 0 [INFO] Migration [C:\laragon\www\POS\database\Migrations\2024_11_06_074957_add_image_to_t_penjualan_table.php] created successfully. C:\laragon\www\POS&gt;</pre>
2	<p>Buka file migrasi tersebut lalu modifikasi seperti ini lalu simpan</p>



	<pre>database &gt; migrations &gt; 2024_11_06_074957_add_image_to_t_penjualan_table.php &gt; class &gt; down &gt; Closure 1  &lt;?php 2 3  use Illuminate\Database\Migrations\Migration; 4  use Illuminate\Database\Schema\Blueprint; 5  use Illuminate\Support\Facades\Schema; 6 7  return new class extends Migration 8  { 9      /** 10       * Run the migrations. 11       */ 12     public function up(): void 13     { 14         Schema::table('t_penjualan', callback: function (Blueprint \$table): void { 15             \$table-&gt;string(column: 'image'); 16         }); 17     } 18 19     /** 20      * Reverse the migrations. 21      */ 22     public function down(): void 23     { 24         Schema::table('t_penjualan', callback: function (Blueprint \$table): void { 25             \$table-&gt;dropColumn(columns: 'image'); 26         }); 27     } 28 };</pre>
3	<p>Lakukan jalankan update migrasi dengan cara:</p> <p><b>php artisan migrate</b></p> <pre>C:\laragon\www\POS&gt;php artisan migrate Warning: Module "gd" is already loaded in Unknown on line 0 [INFO] Running migrations. 2024_11_06_074957_add_image_to_t_penjualan_table ..... 79ms DONE C:\laragon\www\POS&gt;</pre>
4	<p>Lalu lakukan modifikasi models pada App/Models/BarangModel.php</p>



	<pre>app &gt; Models &gt; PenjualanModel.php &gt; PHP Intelephense &gt; PenjualanModel &gt; image 1  &lt;?php 2  namespace App\Models; 3  use Illuminate\Database\Eloquent\Factories\HasFactory; 4  use Illuminate\Database\Eloquent\Model; 5  use App\Models\PenjualanDetailModel; 6  use Illuminate\Database\Eloquent\Relations\BelongsTo; 7  use Illuminate\Database\Eloquent\Relations\HasMany; 8  use Tymon\JWTAuth\Contracts\JWTSubject; 9  use Illuminate\Database\Eloquent\Casts\Attribute; 10 18 references   0 implementations 11  class PenjualanModel extends Model implements JWTSubject 12  { 13      0 references   0 overrides 14      public function getJWTIdentifier(): mixed 15      { 16          return \$this-&gt;getKey(); 17      } 18      0 references   0 overrides 19      public function getJWTCustomClaims(): array 20      { 21          return []; 22      } 23      use HasFactory; 24      0 references 25      protected \$table = 't_penjualan'; // Mendefinisikan nama tabel yang digunakan oleh model ini 26      0 references 27      protected \$primaryKey = 'penjualan_id'; // Mendefinisikan primary key dari tabel yang digunakan 28      0 references 29      protected \$fillable = ['user_id', 'barang_id', 'pembeli', 'penjualan_kode', 'penjualan_tanggal', 'image']; 30      0 references   0 overrides 31      public function user(): BelongsTo 32      { 33          return \$this-&gt;belongsTo(related: UserModel::class, foreignKey: 'user_id', ownerKey: 'user_id'); 34      } 35      0 references   0 overrides 36      public function detail(): HasMany 37      { 38          return \$this-&gt;hasMany(related: PenjualanDetailModel::class, foreignKey: 'penjualan_id', localKey: 'penjualan_id'); 39      } 40      0 references   0 overrides 41      protected function image(): Attribute 42      { 43          return Attribute::make( 44              get: fn (\$image): string UrlGenerator =&gt; url(path: '/gambar/' . \$image), 45          ); 46      } 47  }</pre>
5	Lakukan modifikasi pada Controllers/Api/PenjualanController



	<pre>app &gt; Http &gt; Controllers &gt; Api &gt; PenjualanController.php &gt; ... 1  &lt;?php 2  namespace App\Http\Controllers\Api; 3  use App\Http\Controllers\Controller; 4  use App\Models\PenjualanModel; 5  use App\Models\UserModel; 6  use Illuminate\Http\Request; 7  use Illuminate\Support\Facades\Validator; 8  class PenjualanController extends Controller 9  { 10     public function __invoke(Request \$request): JsonResponse mixed 11     { 12         \$validator = Validator::make(data: \$request-&gt;all(), rules: [ 13             'user_id' =&gt; 'required integer', 14             'pembeli' =&gt; 'required string min:3 max:100', 15             'penjualan_kode' =&gt; 'required string min:3 unique:t_penjualan,penjualan_kode', 16             'penjualan_tanggal' =&gt; 'required date', 17             'image' =&gt; 'required image mimes:jpeg,png,jpg,gif,svg max:2048', 18         ]); 19         if (\$validator-&gt;fails()) { 20             return response()-&gt;json(data: \$validator-&gt;errors(), status: 422); 21         } 22         // Hash nama file dan simpan ke folder public/gambar 23         \$image = \$request-&gt;image; 24         \$hashedName = \$image-&gt;hashName(); 25         \$image-&gt;move(public_path('gambar'), \$hashedName); 26         \$user = PenjualanModel::create(attributes: [ 27             'user_id' =&gt; \$request-&gt;user_id, 28             'pembeli' =&gt; \$request-&gt;pembeli, 29             'penjualan_kode' =&gt; \$request-&gt;penjualan_kode, 30             'penjualan_tanggal' =&gt; \$request-&gt;penjualan_tanggal, 31             'image' =&gt; \$hashedName, 32         ]); 33         if (\$user) { 34             return response()-&gt;json(data: [ 35                 'success' =&gt; true, 36                 'user' =&gt; \$user, 37             ], status: 201); 38         } 39         return response()-&gt;json(data: [ 40             'success' =&gt; false, 41             ], status: 409); 42     } 43     public function show(PenjualanModel \$penjualan): array Collection Model PenjualanModel null 44     { 45         return PenjualanModel::find(id: \$penjualan); 46     } 47 }</pre>
6	<p>Ubah atau tambahkan barang pada routes/api.php</p> <pre>routes &gt; api.php &gt; ... 63 Route::post(uri: '/penjualan', action: App\Http\Controllers\Api\PenjualanController::class)-&gt;name(name: 'penjualan'); 64 Route::get(uri: 'penjualan/{penjualan}', action: [PenjualanController::class, 'show']); 65</pre>
7	<p>Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar</p>



	<div><div>localhost/POS/public/api/kategori?kategori_nama=Sembako</div><div><div>POST</div><div>localhost/POS/public/api/penjualan</div><div>Send</div></div><div><div>Params</div><div>Auth</div><div>Headers (8)</div><div>Body</div><div>Pre-req.</div><div>Tests</div><div>Settings</div><div>Cookies</div></div><div><div>form-data</div><table><tr><th></th><th>Key</th><th>Value</th><th>...</th><th>Bulk Edit</th></tr><tr><td><input checked="" type="checkbox"/></td><td>user_id</td><td>62</td><td></td><td></td></tr><tr><td><input checked="" type="checkbox"/></td><td>pembeli</td><td>Admin Nela</td><td></td><td></td></tr><tr><td><input checked="" type="checkbox"/></td><td>penjualan_kode</td><td>inih</td><td></td><td></td></tr><tr><td><input checked="" type="checkbox"/></td><td>penjualan_tanggal</td><td>2024-11-06 15:03:00</td><td></td><td></td></tr><tr><td><input checked="" type="checkbox"/></td><td>image</td><td>barang.jpg</td><td></td><td></td></tr></table><div><div>Body</div><div>201 Created 540 ms 705 B</div><div>Save Response</div></div><div><div>Pretty</div><div>Raw</div><div>Preview</div><div>Visualize</div><div>JSON</div><div></div></div><div><pre>2      "success": true, 3      "user": { 4        "user_id": "62", 5        "pembeli": "Admin Nela", 6        "penjualan_kode": "inih", 7        "penjualan_tanggal": "2024-11-06 15:03:00", 8        "image": "http://localhost/POS/public/gambar/ 1GYZ3NHt53Ko0MMFkHp3N1cWMJuSdeDJuNKBvLSq.jpg", 9        "updated_at": "2024-11-06T08:03:24.000000Z", 10       "created_at": "2024-11-06T08:03:24.000000Z",</pre></div></div></div>		Key	Value	...	Bulk Edit	<input checked="" type="checkbox"/>	user_id	62			<input checked="" type="checkbox"/>	pembeli	Admin Nela			<input checked="" type="checkbox"/>	penjualan_kode	inih			<input checked="" type="checkbox"/>	penjualan_tanggal	2024-11-06 15:03:00			<input checked="" type="checkbox"/>	image	barang.jpg		
	Key	Value	...	Bulk Edit																											
<input checked="" type="checkbox"/>	user_id	62																													
<input checked="" type="checkbox"/>	pembeli	Admin Nela																													
<input checked="" type="checkbox"/>	penjualan_kode	inih																													
<input checked="" type="checkbox"/>	penjualan_tanggal	2024-11-06 15:03:00																													
<input checked="" type="checkbox"/>	image	barang.jpg																													
8	Menampilkan barang sesuai barang_id yang dipilih dengan method GET																														



localhost/POS/public/api/kategori?kategori\_nama=Sembako

Save

GETlocalhost/POS/public/api/penjualan/13Send

ParamsAuthHeaders (8)BodyPre-req. TestsSettingsCookies

Body200 OK391 ms676 BSave Response

PrettyRawPreviewVisualizeJSON

```
1  {
2    "penjualan_id": 13,
3    "user_id": 62,
4    "pembeli": "Admin Nela",
5    "penjualan_kode": "inih",
6    "penjualan_tanggal": "2024-11-06 15:03:00",
7    "created_at": "2024-11-06T08:03:24.000000Z",
8    "updated_at": "2024-11-06T08:03:24.000000Z",
9    "image": "http://localhost/POS/public/gambar/
10             1GYZ3NHt53Ko0MMFkHp3N1cWMJuSdeDJuNKBvLSq.jpg"
11  }
12 }
```