



Jurusan Teknologi Informasi Politeknik Negeri Malang

## Jobsheet-11 : RESTFUL API

Mata Kuliah Praktikum Pemrograman Web Lanjut (PWL)

Maret 2024

Nama : Elvaretta Salsabilla

NIM : 2241720116

No/Kelas : 14/TI-2H

### JOB SHEET 11

### RESTFUL API 2

#### 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

```
PS C:\laragon\www\PWL_POS> php artisan make:migration add_image_to_m_user_table
```

```
INFO Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_05_15_054505_add_image_to_m_user_table.php] created successfully.
```

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

```
<?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:

```
PS C:\laragon\www\PWL_POS> php artisan migrate
```

```
INFO Running migrations.
```

```
2024_05_15_054505_add_image_to_m_user_table ..... 228ms DONE
```



Jurusan Teknologi Informasi Politeknik Negeri Malang

## Jobsheet-11 : RESTFUL API

Mata Kuliah Praktikum Pemrograman Web Lanjut (PWL)

Maret 2024

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

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\casts\Attribute;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Tymon\JWTAuth\Contracts\JWTSubject;

41 references | 0 implementations
class UserModel extends Authenticatable implements JWTSubject
{
    0 references | 0 overrides
    public function getJWTIdentifier(){
        return $this->getKey();
    }
    0 references | 0 overrides
    public function getJWTCustomClaims(){
        return [];
    }
    use HasFactory;

    0 references
    protected $table = 'm_user'; // mendefinisikan nama tabel yang digunakan oleh model ini
    0 references
    public $timestamps = false;
    0 references
    protected $primaryKey = 'user_id'; // mendefinisikan primary key dari tabel yang digunakan

    0 references
    protected $fillable = [
        'username',
        'nama',
        'password',
        'level_id',
        'image'
    ];

    0 references | 0 overrides
    public function level(): BelongsTo
    {
        return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
    }

    0 references | 0 overrides
    protected function image(): Attribute
    {
        return Attribute::make(
            get: fn ($image) => url('/storage/posts/' . $image),
        );
    }
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang

## Jobsheet-11 : RESTFUL API

Mata Kuliah Praktikum Pemrograman Web Lanjut (PWL)

Maret 2024

6. Lakukan modifikasi pada Controllers/Api/RegisterController

```
1 reference | 0 implementations
class RegisterController extends Controller
{
    0 references | 0 overrides
    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);
    }
}
```

7. Anda dapat menambahkan detail untuk spesifikasi image pada validator

```
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|image|mimes:jpeg,png,jpg,gif,svg|max:2048'
    ]);
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang

## Jobsheet-11 : RESTFUL API

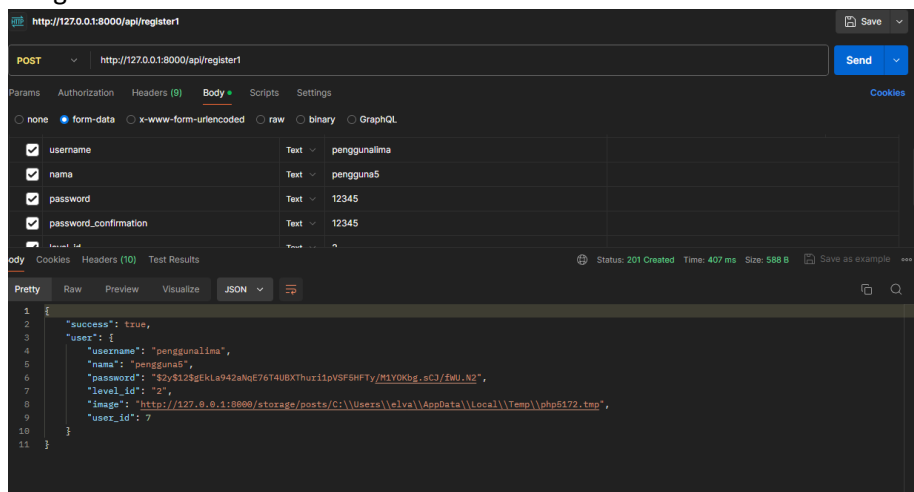
Mata Kuliah Praktikum Pemrograman Web Lanjut (PWL)

Maret 2024

8. Ubah atau tambahkan register1 pada routes/api.php

```
Route::post('/register1', App\Http\Controllers\Api\RegisterController::class)->name('register1');
```

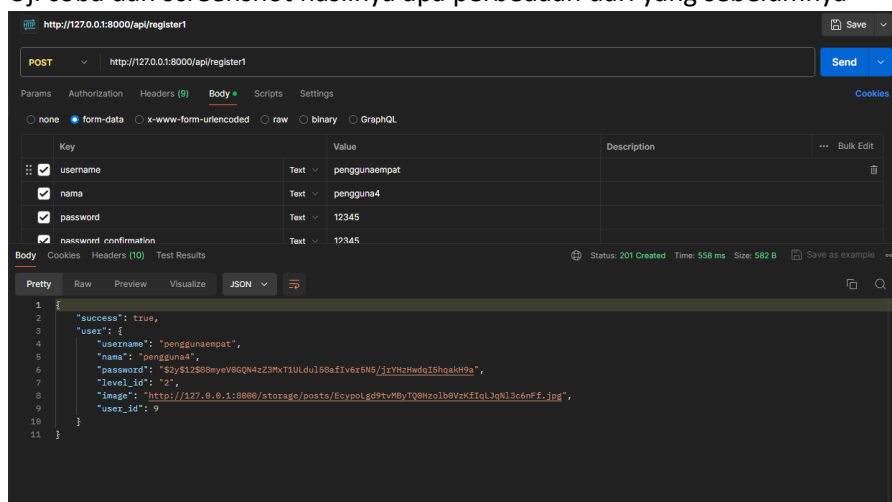
9. Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar <http://127.0.0.1:8000/api/register1> dengan method POST dan klik send



10. Pada Controllers/Api/RegisterController bagian create user ganti dengan

```
// Create user
$user = UserModel::create([
    'username' => $request->username,
    'nama' => $request->nama,
    'password' => bcrypt($request->password),
    'level_id' => $request->level_id,
    'image' => $request->image->hashName(),
]);
```

11. Uji coba dan screenshot hasilnya apa perbedaan dari yang sebelumnya



Hasil perbedaan



Jurusan Teknologi Informasi Politeknik Negeri Malang

## Jobsheet-11 : RESTFUL API

Mata Kuliah Praktikum Pemrograman Web Lanjut (PWL)

Maret 2024

			9	2	penggunaempat	pengguna4	\$2y\$12\$88myeV0GQN4zZ3MxT1ULdu58afv6r5N5jrYHzHwd...	NULL	NULL	EcypoLgd9tvMByTQ0Hzolb0VzKflqLJqNI3c6nFf.jpg
			10	2	penggunalima	pengguna5	\$2y\$12\$88bldYIAyEk0y95oNlpgem42HQOW8aaqtAZXhnDprB...	NULL	NULL	C:\Users\elva\AppData\LocalTemp\php165E.tmp

Sebelum menggunakan `hasName()` direktori image yang disimpan akan terlihat dalam database, setelah menggunakan `hasName` dalam database hanya akan menampilkan nama image yang diunggah.

### 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.

#### Barang

##### 1. Migrations barang image

```
<?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');
        });
    }
};
```

##### 2. Jalankan migration

```
PS C:\laragon\www\PWL_POS> php artisan migrate
```

**INFO** Running migrations.

2024\_05\_15\_065314\_add\_image\_to\_m\_barang\_table ..... 20ms **DONE**

##### 3. Modifikasi barangModel



Jurusan Teknologi Informasi Politeknik Negeri Malang

## Jobsheet-11 : RESTFUL API

Mata Kuliah Praktikum Pemrograman Web Lanjut (PWL)

Maret 2024

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\casts\Attribute;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

28 references | 0 implementations
class BarangModel extends Model
{
    0 references
    protected $table = 'm_barang';
    0 references
    protected $primaryKey = 'barang_id';

    0 references
    protected $fillable =
    [
        'barang_id',
        'kategori_id',
        'barang_kode',
        'barang_nama',
        'harga_beli',
        'harga_jual',
        'image' //tambahan
    ];
    0 references | 0 overrides
    public function kategori(): BelongsTo
    {
        return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
    }

    0 references | 0 overrides
    protected function image(): Attribute
    {
        return Attribute::make(
            get: fn ($image) => url('/storage/posts/' . $image),
        );
    }
}
```

### 4. Modifikasi barang controller

```
<?php

namespace App\Http\Controllers\Api;

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

6 references | 0 implementations
class BarangController extends Controller
{
    1 reference | 0 overrides
    public function index(){
        return BarangModel::all();
    }

    1 reference | 0 overrides
    public function store(Request $request){

        $request->validate([
            'barang_kode' => 'required|string|min:3|unique:m_barang,barang_kode',
            'barang_nama' => 'required|string|max:100',
            'kategori_id' => 'required|integer',
            'harga_beli' => 'required|integer',
            'harga_jual' => 'required|integer',
            'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048'
        ]);

        $barang = BarangModel::create([
            'barang_kode' => $request->barang_kode,
            'barang_nama' => $request->barang_nama,
            'kategori_id' => $request->kategori_id,
            'harga_beli' => $request->harga_beli,
            'harga_jual' => $request->harga_jual,
            'image' => $request->image->hashName()
        ]);
        return response()->json($barang, 201);
    }

    1 reference | 0 overrides
    public function show(BarangModel $barang){
        return BarangModel::find($barang);
    }
}
```



Jurusan Teknologi Informasi Politeknik Negeri Malang

## Jobsheet-11 : RESTFUL API

Mata Kuliah Praktikum Pemrograman Web Lanjut (PWL)

Maret 2024

```
1 reference | 0 overrides
public function update(Request $request, BarangModel $barang){

    $request->validate([
        'barang_kode' => 'required|string|min:3|unique:m_barang,barang_kode',
        'barang_nama' => 'required|string|max:100',
        'kategori_id' => 'required|integer',
        'harga_beli' => 'required|integer',
        'harga_jual' => 'required|integer',
        'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048'
    ]);

    $barang->update([
        'barang_kode' => $request->barang_kode,
        'barang_nama' => $request->barang_nama,
        'kategori_id' => $request->kategori_id,
        'harga_beli' => $request->harga_beli,
        'harga_jual' => $request->harga_jual,
        'image' => $request->image->hashName()
    ]);
    return BarangModel::find($barang->barang_id);
}

1 reference | 0 overrides
public function destroy(BarangModel $barang){
    $barang->delete();
    return response()->json([
        'success' => true,
        'message' => 'Data Terhapus'
    ]);
}
```

### 5. Input gambar pada m\_barang dengan POST

Postman interface showing a POST request to `http://127.0.0.1:8000/api/items`. The request body is a JSON object with the following data:

```
{
  "barang_kode": "FND",
  "barang_nama": "Fondation",
  "kategori_id": "2",
  "harga_beli": "50000",
  "harga_jual": "55000",
  "image": "http://127.0.0.1:8000/storage/posts/a0ezTyuRDnyMrijOOD5H17mANzWMfia1UB5sjmH9.jpg",
  "created_at": "2024-05-15T07:14:37.000000Z",
  "updated_at": "2024-05-15T07:14:37.000000Z",
  "barang_id": 11
}
```

The response is a 201 Created status with the following JSON data:

```
{
  "barang_id": 11,
  "kategori_id": 2,
  "barang_kode": "FND",
  "barang_nama": "Fondation",
  "harga_beli": "50000",
  "harga_jual": "55000",
  "created_at": "2024-05-15T07:14:37.000000Z",
  "updated_at": "2024-05-15T07:14:37.000000Z",
  "image": "http://127.0.0.1:8000/storage/posts/a0ezTyuRDnyMrijOOD5H17mANzWMfia1UB5sjmH9.jpg"
}
```

### 6. Menampilkan hasil input dengan GET

Postman interface showing a GET request to `http://127.0.0.1:8000/api/items/11`. The response is a 200 OK status with the following JSON data:

```
{
  "barang_id": 11,
  "kategori_id": 2,
  "barang_kode": "FND",
  "barang_nama": "Fondation",
  "harga_beli": "50000",
  "harga_jual": "55000",
  "created_at": "2024-05-15T07:14:37.000000Z",
  "updated_at": "2024-05-15T07:14:37.000000Z",
  "image": "http://127.0.0.1:8000/storage/posts/a0ezTyuRDnyMrijOOD5H17mANzWMfia1UB5sjmH9.jpg"
}
```

## Transaksi

### 1. Tambah transaksiController pada folder API



Jurusan Teknologi Informasi Politeknik Negeri Malang

## Jobsheet-11 : RESTFUL API

Mata Kuliah Praktikum Pemrograman Web Lanjut (PWL)

Maret 2024

```
<?php
namespace App\Http\Controllers\Api;

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

2 references | 0 implementations
class TransaksiController extends Controller
{
    1 reference | 0 overrides
    public function show($id)
    {
        $penjualan = PenjualanModel::with('detail.barang')->findOrFail($id);

        $response = [
            'ID' => $penjualan->penjualan_id,
            'Pembeli' => $penjualan->pembeli,
            'Kode Penjualan' => $penjualan->penjualan_kode,
            'Tanggal Penjualan' => $penjualan->penjualan_tanggal,
            'Barang yang dibeli' => [],
            'Total Belanja' => 0,
        ];

        foreach ($penjualan->detail as $item) {
            $barang = $item->barang;
            $subtotal = $item->harga * $item->jumlah;
            $response['Barang yang dibeli'][] = [
                'Barang' => $barang->barang_nama,
                'Harga' => $item->harga,
                'Jumlah' => $item->jumlah,
                'Subtotal' => $subtotal,
                'Image' => url('storage/posts/' . $barang->image),
            ];
            $response['Total Belanja'] += $subtotal;
        }

        return response()->json($response, 200);
    }
}
```

2. Tambahkan route pada web/api

```
Route::get('/transaksi/{penjualan}', [TransaksiController::class, 'show']);
```

3. Tampilan GET transaksi

