

PEMROGRAMAN WEB LANJUT
JOBSHEET 11 - RESTFUL API 2



Disusun oleh:
Stefanus Ageng Budi Utomo (2241720126)

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN AJARAN 2023/2024

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

```
PS D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS> php artisan make:migration add_image_to_m_user_table
INFO Migration [D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS\database\migrations\2024_05_13_160700_add_image_to_m_user_table.php] created successfully.
```

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

```
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

```
PS D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS> php artisan migrate
INFO Running migrations.
2024_05_13_160700_add_image_to_m_user_table ..... 211ms DONE
```

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

```

class UserModel extends Authenticatable implements JWTSubject
{
    use HasFactory;

    public function getJWTIdentifier()
    {
        return $this->getKey();
    }

    public function getJWTCustomClaims()
    {
        return [];
    }

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

    // protected $fillable = ['user_id', 'level_id', 'username', 'nama', 'password', 'image'];
    protected $fillable = [
        'username',
        'nama',
        'password',
        'level_id',
        'image',
    ];

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

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

```

6. Lakukan modifikasi pada Controllers/Api/RegisterController

```

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

        // 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);
    }
}

```

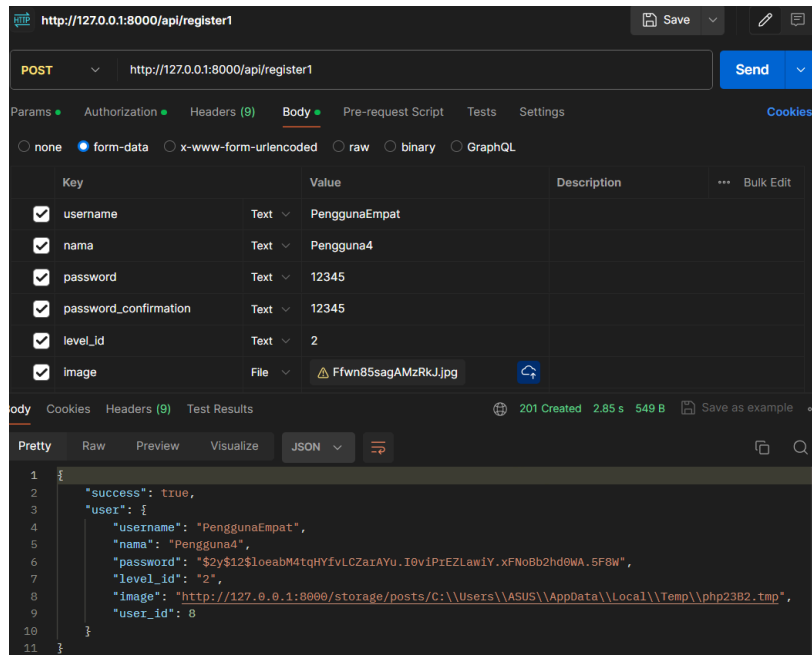
7. Ubah atau tambahkan register1 pada routes/api.php

```

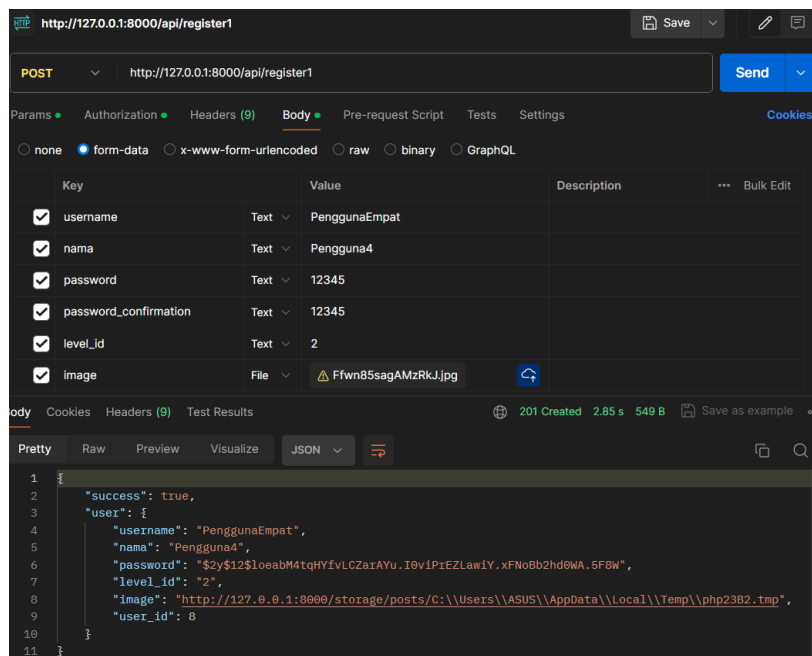
Route::post('/register1', \App\Http\Controllers\Api\RegisterController::class)->name('register1');

```

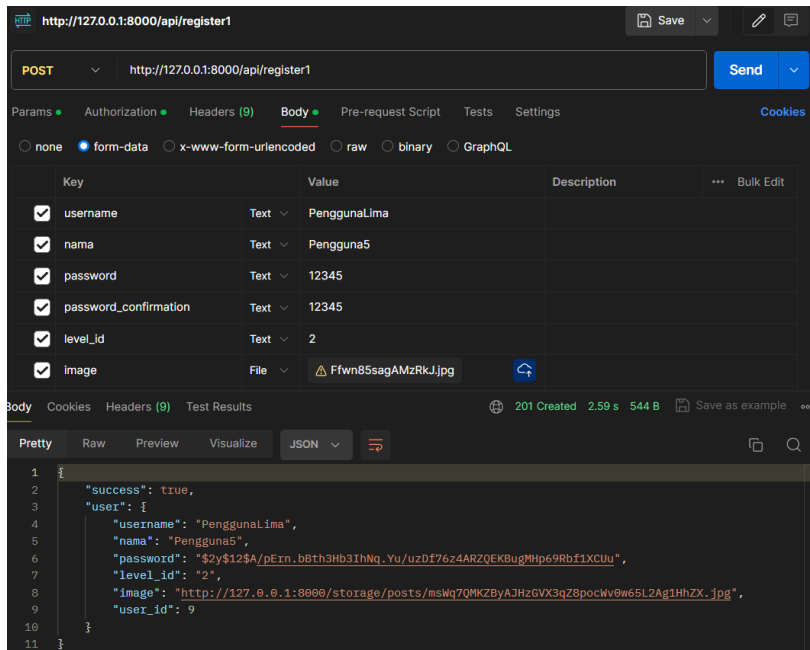
8. Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar <http://127.0.0.1:8000/api/register1> dengan method POST dan klik send



9. Pada Controllers/Api/RegisterController bagian create user ganti dengan `'image' ⇒ $request→image→hashName(),`
10. Uji coba dan screenshot hasilnya apa perbedaan dari yang sebelumnya sebelum



sesudah



- Perbedaananya adalah terletak pada nama file dimana ketika menambahkan ->hashName() maka nama file gambar yang kirim akan di enkripsi.

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.

```
class BarangModel extends Model
{
    protected $table = 'm_barang';
    protected $primaryKey = 'barang_id';
    protected $fillable = ['barang_kode', 'barang_nama', 'kategori_id', 'harga_beli', 'harga_jual', 'image'];
    // use HasFactory:
    public function index(BarangModel $dataTables)
    {
        return $dataTables->render('kategori.index');
    }

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

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

```
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');
        });
    }
};
```

```

public function store(Request $request)
{
    // set validation
    $validator = Validator::make($request->all(), [
        'kategori_id' => 'required|integer',
        'barang_kode' => 'bail|required|string|unique:m_barang,barang_kode',
        'barang_nama' => 'required|string|max:100',
        'harga_beli' => 'required|integer',
        'harga_jual' => 'required|integer',
        'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
    ]);

    // if validations fails
    if ($validator->fails()) {
        return response()->json($validator->errors(), 422);
    };

    // create Barang
    $brg = BarangModel::create([
        'kategori_id' => $request->kategori_id,
        'barang_kode' => $request->barang_kode,
        'barang_nama' => $request->barang_nama,
        'harga_beli' => $request->harga_beli,
        'harga_jual' => $request->harga_jual,
        'image' => $request->image->hashName(),
    ]);

    // return response JSON user is created
    if ($brg) {
        return Response()->json([
            'success' => true,
            'barang' => $brg,
        ], 201);
    }
}

```

http://127.0.0.1:8000/api/barangs

POST http://127.0.0.1:8000/api/barangs Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

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

<input checked="" type="checkbox"/>	kategori_id	Text	1
<input checked="" type="checkbox"/>	barang_kode	Text	BRG002
<input checked="" type="checkbox"/>	barang_nama	Text	Laptop Alienware
<input checked="" type="checkbox"/>	harga_beli	Text	300
<input checked="" type="checkbox"/>	harga_jual	Text	400
<input checked="" type="checkbox"/>	image	File	avg_8_33.png

Body Cookies Headers (9) Test Results 201 Created 1326 ms 611 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "success": true,
3   "barang": {
4     "kategori_id": "1",
5     "barang_kode": "BRG002",
6     "barang_nama": "Laptop Alienware",
7     "harga_beli": "300",
8     "harga_jual": "400",
9     "image": "http://127.0.0.1:8000/storage/posts/z9ED1YZAbHs7V0BH9KnHHfNspmk1e15QoheDhBhq.png",
10    "updated_at": "2024-05-14T13:34:39.000000Z",
11    "created_at": "2024-05-14T13:34:39.000000Z",
12    "barang_id": 13

```