

Nama : Adinda Ivanka Maysanda Putri

Kelas : SIB 2B

NIM : 2341760058

---

## JOBSHEET 11

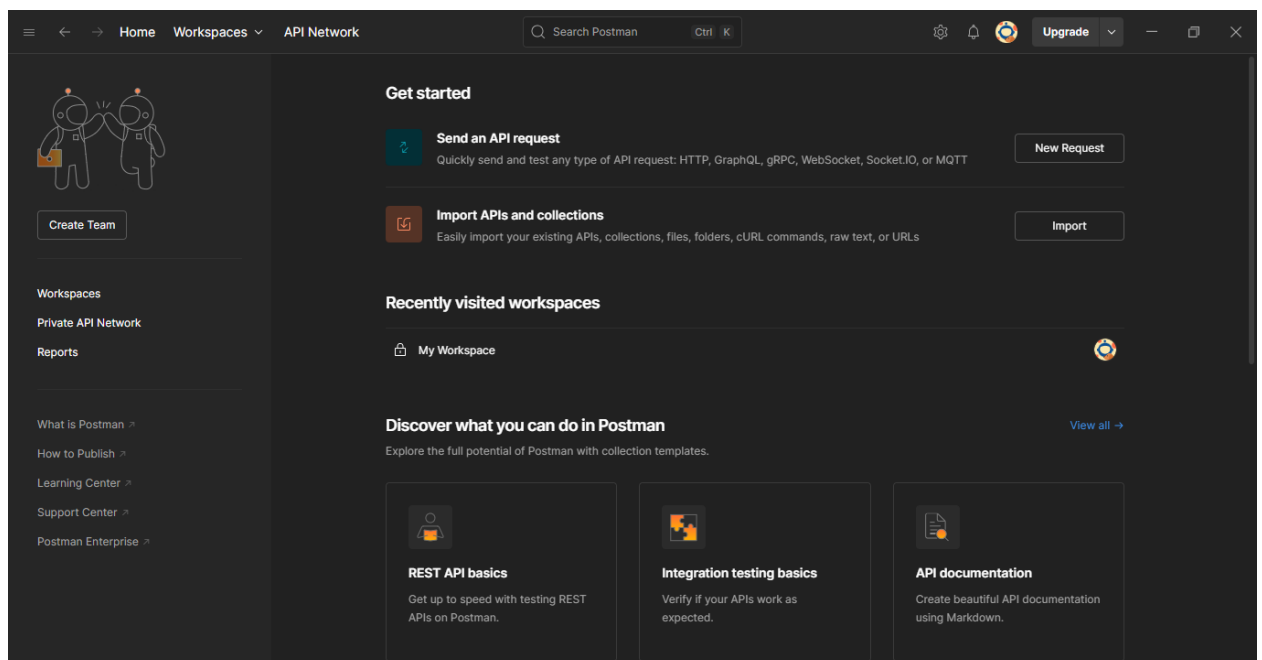
### RESTFUL API 2

---

#### 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 php artisan make:migration add\_image\_to\_m\_user\_table

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 11> php artisan make:migration add_image_to_m_user_table
INFO Migration [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 11\database\Migrations\2025_04_27_144822_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
{
    public function up(): void
    {
        Schema::table('m_barang', function (Blueprint $table) {
            $table->string('image')->nullable();
        });
    }

    public function down(): void
    {
        Schema::table('m_barang', function (Blueprint $table) {
            $table->dropColumn('image');
        });
    }
};

```

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

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 11> php artisan migrate
```

```
INFO Running migrations.
```

```
2025_04_27_144822_add_image_to_m_user_table ..... 802ms DONE
```

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

```

<?php

namespace App\Http\Controllers\Api;

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

```

```

use Illuminate\Support\Facades\Validator;

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',
        ]);

        // If validation 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->hashName(),
        ]);

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

        // Return JSON process insert failed
    }
}

```

```
return response()->json([
    'success' => false,
], 409);
}
}
```

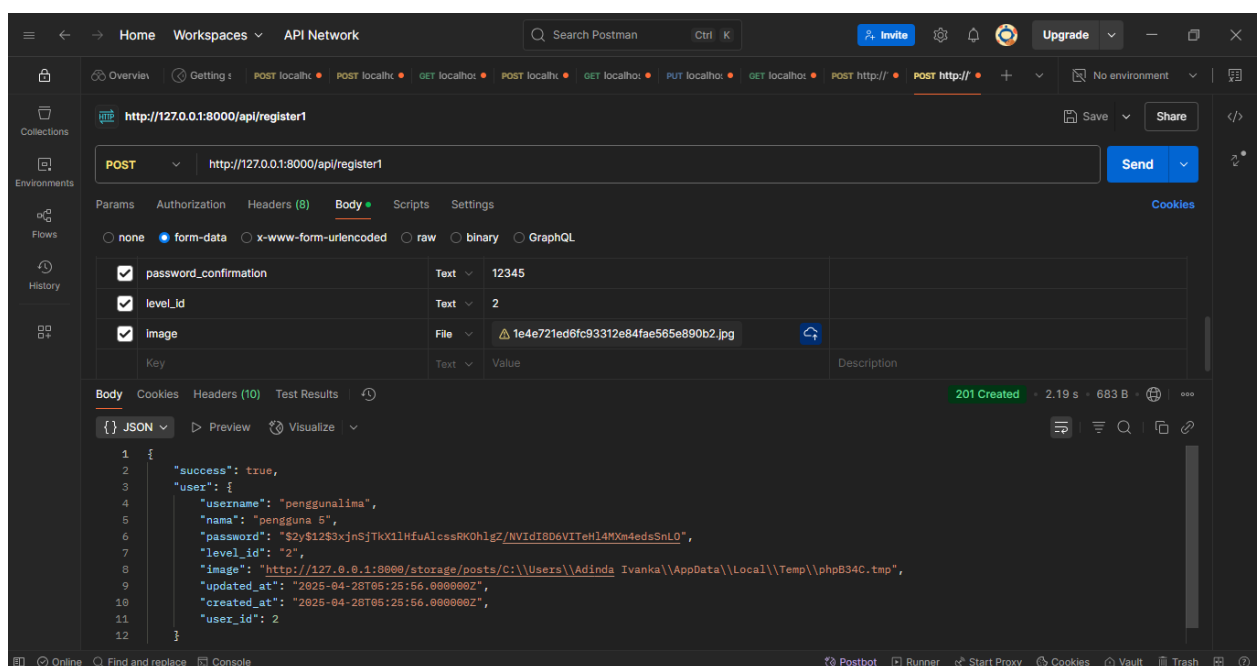
7. Anda dapat menambahkan detail untuk spesifikasi image pada validator

```
'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\RegisterController::class)->name('register1');
```

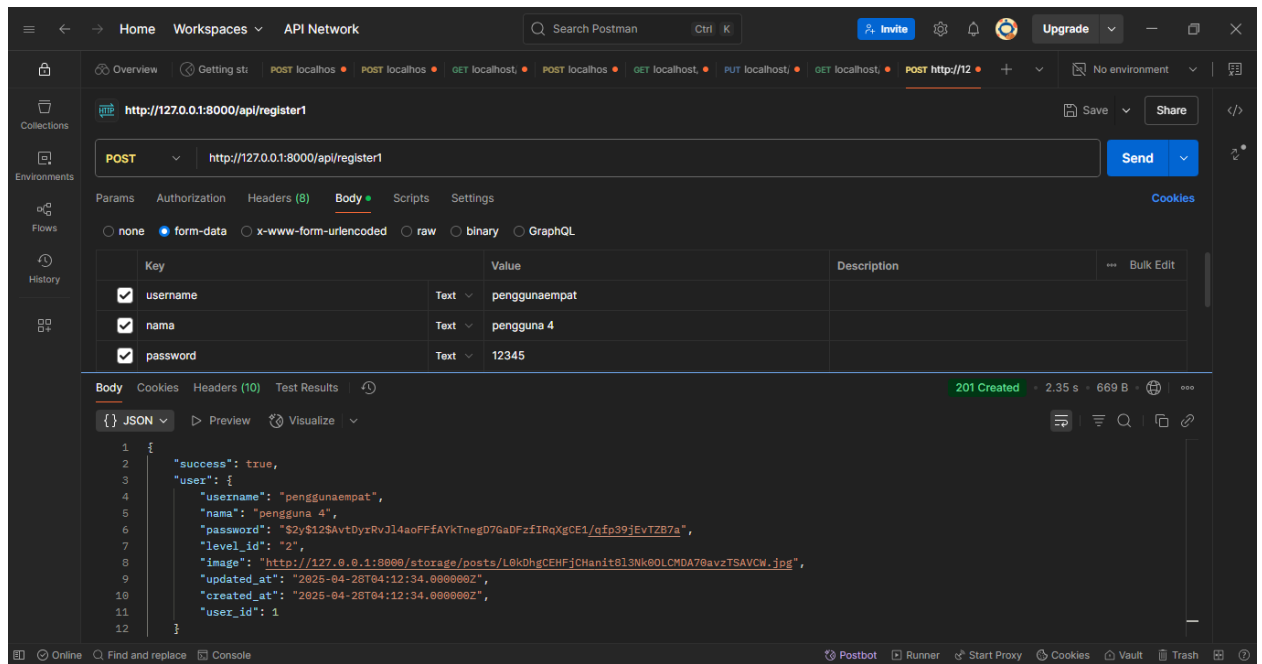
9. Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar <http://127.0.0.1:8000/api/register1> dengan method POST dan klik send



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

```
'image' => $request->image->hashName(),
```

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



## 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.

```
PS C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 11> php artisan make:migration add_image_to_m_barang_table

INFO Migration [C:\laragon\www\Pemrograman-Web-Lanjut\Minggu 11\database\migrations\2025_04_28_053203_add_image_to_m_barang_table.php] created successfully.
```