

1. 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:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS> php artisan make:migration add_image_to_m_user_table
[INFO] Migration [D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS\database\migrations\2024_05_15_064803_add_image_to_m_user_table.php] created successfully.
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS>
```

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

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

```
2014_10_12_000000_create_users_table ..... 34ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 5ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 22ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 27ms DONE
2024_03_06_071941_create_m_level_table ..... 20ms DONE
2024_03_06_072602_create_m_kategori_table ..... 28ms DONE
2024_03_06_073036_create_m_user_table ..... 45ms DONE
2024_03_06_073423_create_m_barang_table ..... 41ms DONE
2024_03_06_073654_create_t_penjualan_table ..... 37ms DONE
2024_03_06_073807_create_t_stok_table ..... 82ms DONE
2024_03_06_073834_create_t_penjualan_detail_table ..... 0ms DONE
2024_05_15_064803_add_image_to_m_user_table ..... 2ms DONE
```

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

```

pp > Models > UserModel.php > PHP > App\Models\UserModel
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Database\Eloquent\Relations\BelongsTo;
9 use Tymon\JWTAuth\Contracts\JWTSubject;
10 use Illuminate\Database\Eloquent\Casts\Attribute;
11
12 42 references | 0 implementations
13 class UserModel extends Authenticatable implements JWTSubject
14 {
15     use HasFactory;
16
17     0 references
18     protected $table = 'm_user';
19     0 references
20     protected $primaryKey = 'user_id';
21
22     0 references
23     protected $fillable = ['level_id', 'username', 'nama', 'password'];
24     0 references
25     protected $guarded = [];
26
27     0 references | 0 overrides
28     public function getJWTIdentifier()
29     {
30         return $this->getKey();
31     }
32
33     0 references | 0 overrides
34     public function getJWTCustomClaims()
35     {
36         return [];
37     }
38
39     0 references | 0 overrides
40     public function level(): BelongsTo
41     {
42         return $this->belongsTo(LevelModel::class, 'level_id');
43     }
44
45     0 references | 0 overrides
46     protected function image(): Attribute
47     {
48         return Attribute::make(
49             get: fn($image) => url('/storage/posts/' . $image),
50         );
51     }
52 }

```

5. Lakukan modifikasi pada Controllers/Api/RegisterController

```

2 references | 0 implementations
class RegisterController extends Controller
{
    //
    0 references | 0 overrides
    public function __invoke(Request $request)
    {
        $validator = Validator::make($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);
    }
}

```

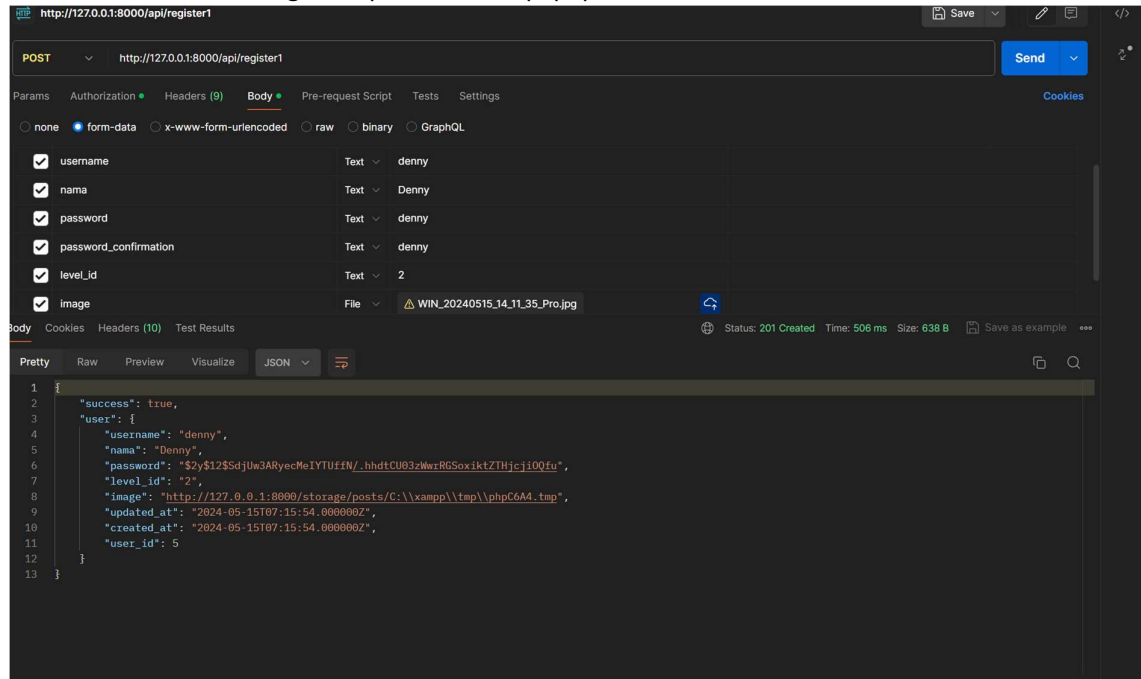
6. Anda dapat menambahkan detail untuk spesifikasi image pada validator

```

{
    $validator = Validator::make($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'
    ]);
}

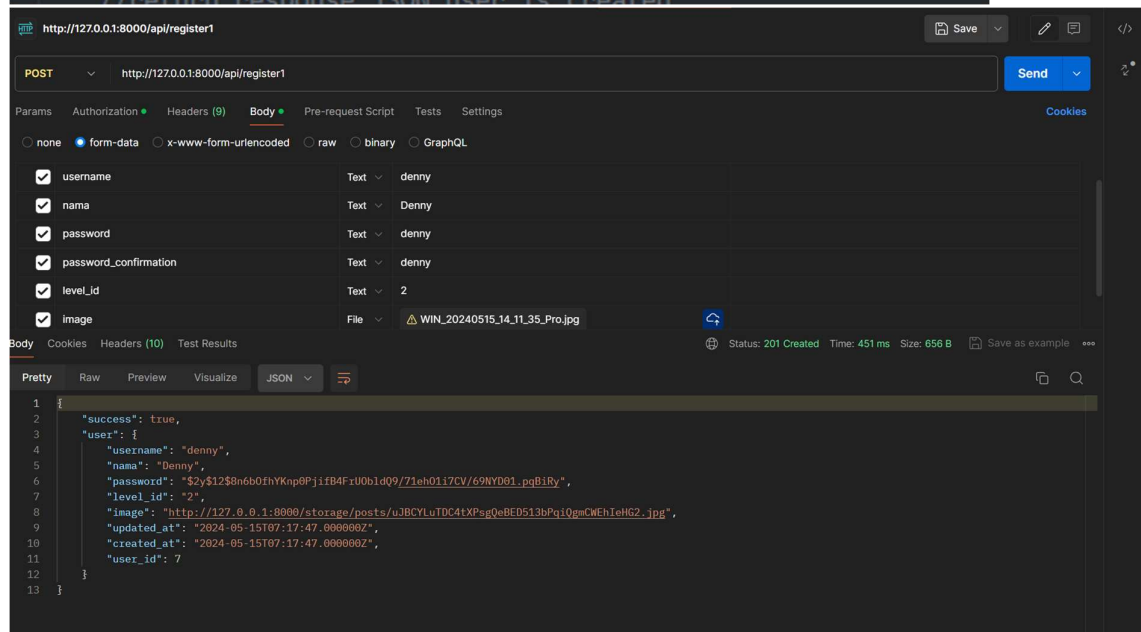
```

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



8. 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(),
]);
//return response JSON user is created
```



Perbedaan terletak pada lokasi path yang dikembalikan.

Tugas

```
0 references
protected $fillable = [
    'kategori_id',
    'barang_kode',
    'barang_nama',
    'harga_beli',
    'harga_jual',
    'image'
];
```

```
16
17 1 reference | 0 overrides
18 public function store(Request $request)
19 {
20
21
22     $level = BarangModel::create([
23         'kategori_id' => $request->kategori_id,
24         'barang_kode' => $request->barang_kode,
25         'image' => $request->image->hashname(),
26         'harga_beli' => $request->harga_beli,
27         'harga_jual' => $request->harga_jual,
28         'barang_nama' => $request->barang_nama,
29
30     ]);
31     return response()->json([
32         $level
33     ], 201);
34 }
35
```

