

**LAPORAN**  
**HandsOut(Jobsheet) - Pemrograman Web – Eloqeunt ORM**

**Fendi Zulkarnain**



**VSGA X TSA**  
**POLITEKNIK NEGERI MALANG KAMPUS LUMAJANG**  
*Jl. Lintas Timur, Area Sawah/Kebun, Jogotrunan, Kec. Lumajang, Kabupaten Lumajang,*  
*Jawa Timur 67314*  
**2024**

## Percobaan 1 : Implementasi Eloquent ORM

1. Membuat model UserModel menggunakan perintah sebagai berikut :

```
C:\laragon\www\pwl_pos>php artisan make:model UserModel
```

```
INFO Model [C:\laragon\www\pwl_pos\app\Models\UserModel.php] created successfully.
```

2. Setelah berhasil generate model, terdapat 2 file pada folder model yaitu file User.php bawaan dari laravel dan file UserModel.php yang telah kita buat. Kali ini kita akan menggunakan file UserModel.php
3. Kita buka file UserModel.php dan modifikasi seperti berikut :

```
UserModel.php U x
app > Models > UserModel.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class UserModel extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'm_user';
13     protected $primaryKey = 'user_id';
14 }
15
```

4. Buat controller baru dengan nama UserController

```
C:\laragon\www\pwl_pos>php artisan make:controller UserController
```

```
INFO Controller [C:\laragon\www\pwl_pos\app\Http\Controllers\UserController.php] created successfully.
```

5. Kita modifikasi route web.php untuk mencoba routing ke controller UserController

```
web.php
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\KategoriController;
4  use App\Http\Controllers\LevelController;
5  use App\Http\Controllers\UserController;
6  use Illuminate\Support\Facades\Route;
7
8  Route::get('/', function () {
9      return view('welcome');
10 });
11
12 Route::get('/level', [LevelController::class, 'index']);
13 Route::get('/kategori', [KategoriController::class, 'index']);
14 Route::get('/user', [UserController::class, 'index']);
```

6. Sekarang, kita buat file controller UserController dan memodifikasinya seperti berikut :

```
UserController.php U X
app > Http > Controllers > UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel as ModelsUserModel;
6  use Illuminate\Http\Request;
7
8  class UserController extends Controller
9  {
10     public function index()
11     {
12         // Coba akses model UserModel
13         $user = ModelsUserModel::all(); // Ambil semua data dari tabel m_user
14         return view('user', ['data'=>$user]);
15     }
16 }
```

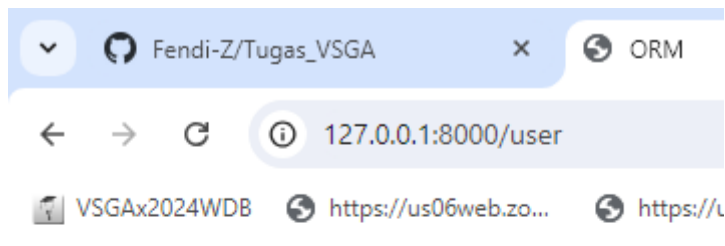
7. Kemudian kita buat view user.blade.php

```
C:\laragon\www\pwl_pos>php artisan make:view user
```

**INFO** View [C:\laragon\www\pwl\_pos\resources\views\user.blade.php] created successfully.

```
user.blade.php U X
resources > views > user.blade.php > html > body > table > tr
1  <!DOCTYPE html>
2  <html Lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>ORM</title>
8  </head>
9  <body>
10     <h1>Data Level Pengguna</h1>
11     <table border="1" cellpadding="2" cellspacing="0">
12         <tr>
13             <th>ID</th>
14             <th>Username</th>
15             <th>Nama</th>
16             <th>ID Level Pengguna</th>
17         </tr>
18         @foreach ($data as $item)
19             <tr>
20                 <td>{{ $item->user_id }}</td>
21                 <td>{{ $item->username }}</td>
22                 <td>{{ $item->nama }}</td>
23                 <td>{{ $item->level_id }}</td>
24             </tr>
25         @endforeach
26     </table>
27 </body>
28 </html>
```

8. Jalankan di browser



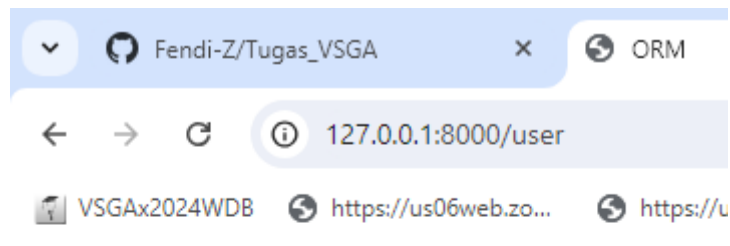
## Data Level Pengguna

ID	Username	Nama	ID Level Pengguna
1	Admin	Administrator	1
2	anager	Manager	2
3	staff	Staff/Kasir	3

9. Setelah itu, kita modifikasi lagi file UserController

```
UserController.php U x
app > Http > Controllers > UserController.php > UserController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel as ModelsUserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         // // Coba akses model UserModel
14         // $user = ModelsUserModel::all(); // Ambil semua data dari tabel m_user
15         // return view('user', ['data'=>$user]);
16
17         // Menambahkan data menggunakan ORM
18         $data = [
19             'username' => 'Dika',
20             'nama' => 'Dika',
21             'password' => Hash::make('1234'),
22             'user_id' => 4,
23             'level_id' => 1,
24         ];
25         ModelsUserModel::insert($data);
26
27         // Akses UserModel
28         $user = ModelsUserModel::all();
29         return view('user', ['data'=>$user]);
30     }
31 }
```

10. Jalankan di browser, amati dan laporkan apa yang terjadi



## Data Level Pengguna

ID	Username	Nama	ID Level Pengguna
1	Admin	Administrator	1
2	anager	Manager	2
3	staff	Staff/Kasir	3
4	Dika	Dika	1

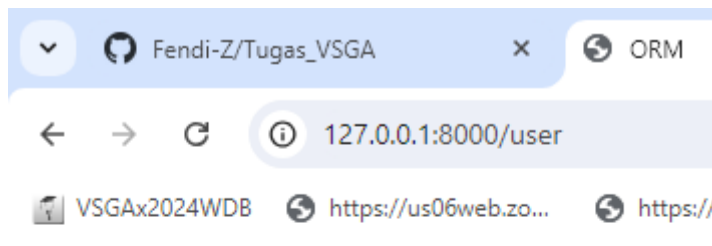
11. Kita modifikasi lagi file UserController (update data ) menjadi seperti berikut

```

UserController.php U X
app > Http > Controllers > UserController.php > UserController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel as ModelsUserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         // // Coba akses model UserModel
14         // $user = ModelsUserModel::all(); // Ambil semua data dari
15         // return view('user', ['data'=>$user]);
16
17         // Menambahkan data menggunakan ORM
18         // $data = [
19         //     'username' => 'Dika',
20         //     'nama' => 'Dika',
21         //     'password' => Hash::make('1234'),
22         //     'user_id' => 4,
23         //     'level_id' => 1,
24         // ];
25         // ModelsUserModel::insert($data);
26
27         $data = [
28             'nama' => "Anton"
29         ];
30         ModelsUserModel::where('username', 'Dika')->update($data);
31
32         // Akses UserModel
33         $user = ModelsUserModel::all();
34         return view('user', ['data'=>$user]);
35     }
36 }

```

12. Jalankan di browser, amati dan laporkan apa yang terjadi



## Data Level Pengguna

ID	Username	Nama	ID Level Pengguna
1	Admin	Administrator	1
2	anager	Manager	2
3	staff	Staff/Kasir	3
4	Dika	Anton	1

13. Hapus data

```
// Delete data
ModelsUserModel::where('username', 'Dika')->delete();

// Akses UserModel
$user = ModelsUserModel::all();
return view('user', ['data'=>$user]);
```

14. Silahkan dicoba pada browser



## Data Level Pengguna

ID	Username	Nama	ID Level Pengguna
1	Admin	Administrator	1
2	Manager	Manager	2
3	Staff	Staff/Kasir	3

### Percobaan 2 : Implementasi Eloquent ORM – 2

1. Membuat table migration dengan nama create\_anggotas\_table

```
C:\laragon\www\pwl_pos>php artisan make:migration create_anggotas_table --create=anggotas
```

```
INFO Migration [C:\laragon\www\pwl_pos\database\migrations\2024_08_15_052156_create_anggotas_table.php] created successfully.
```

2. Setelah membuat migration buka file migration create\_anggota\_table.php, tambahkan code dan setelah itu migrate

```
database > migrations > 2024_08_15_052156_create_anggotas_table.php > class > up > Closure
1  <?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::create('anggotas', function (Blueprint $table) {
15              $table->id();
16              $table->char('nip', 8)->unique();
17              $table->string('nama');
18              $table->dateTime('tanggal_lahir');
19              $table->decimal('nilai', 3, 2)->default(1.00);
20              $table->timestamps();
21          });
22      }
23
24      /**
25       * Reverse the migrations.
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
C:\laragon\www\pwl_pos>php artisan make:migration create_anggotas_table --create=anggotas

INFO Migration [C:\laragon\www\pwl_pos\database\Migrations\2024_08_15_052156_create_anggotas_table.php] created successfully.

C:\laragon\www\pwl_pos>php artisan migrate

INFO Running migrations.

2024_08_15_052156_create_anggotas_table ..... 5,205ms DONE
```

3. Buat controller dengan nama AnggotasController

```
C:\laragon\www\pwl_pos>php artisan make:controller AnggotasController

INFO Controller [C:\laragon\www\pwl_pos\app\Http\Controllers\AnggotasController.php] created successfully.
```

4. Buat beberapa route

```
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\AnggotasController;
4  use App\Http\Controllers\KategoriController;
5  use App\Http\Controllers\LevelController;
6  use App\Http\Controllers\UserController;
7  use Illuminate\Support\Facades\Route;
8
9  Route::get('/', function () {
10      return view('welcome');
11  });
12
13  Route::get('/level', [LevelController::class, 'index']);
14  Route::get('/kategori', [KategoriController::class, 'index']);
15  Route::get('/user', [UserController::class, 'index']);
16
17  Route::get('/cekobject', [AnggotasController::class, 'cekObject'],);
18  Route::get('/insert', [AnggotasController::class, 'insert'],);
19  Route::get('/update', [AnggotasController::class, 'update'],);
20  Route::get('/delete', [AnggotasController::class, 'delete'],);
21  Route::get('/all', [AnggotasController::class, 'all'],);
22  Route::get('/find', [AnggotasController::class, 'find'],);
23  Route::get('/getwhere', [AnggotasController::class, 'getwhere'],);
```

5. Buat model dengan nama Anggotas



```
C:\laragon\www\pw1_pos>php artisan make:model Anggotas
```

INFO Model [C:\laragon\www\pw1\_pos\app\Models\Anggotas.php] created successfully.

6. Buat function untuk melihat data

```
app > Http > Controllers > AnggotasController.php > Anggo
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Anggotas;
6  use Illuminate\Http\Request;
7
8  class AnggotasController extends Controller
9  {
10     //
11     public function cekObject() {
12         $anggota = new Anggotas();
13
14         dump($anggota);
15     }
16 }
```

7. Membuat function untuk input data

```
public function insert() {
    $anggota = new Anggotas;
    $anggota->nip='1234321';
    $anggota->nama='Dika';
    $anggota->tanggal_lahir='2001-12-31';
    $anggota->nilai='3.2';
    $anggota->save();
    echo 'Data berhasil disimpan';
}
```

8. Jalan url pada web browser dan liat pada database perubahannya

Fendi-Z/Tugas\_VSGA 127.0.0.1:8000/insert

127.0.0.1:8000/insert

VSGAx2024WDB https://us06web.zo... https://us06web.zo...

Data berhasil disimpan

	id	nip	nama	tanggal_lahir	nilai	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1234321	Dika	2001-12-31 00:00:00	3.20	2024-08-15 06:45:58	2024-08-15 06:45:58

9. Membuat function untuk Update data

```
public function update() {
    $anggota = Anggotas::find(1);
    $anggota->nama='Dono';
    $anggota->nilai='3.0';
    $anggota->save();
    echo 'Data berhasil diupdate';
}
```

10. Buka browser dan jalan url nya serta cek di DB nya apakah ada perubahan\

Browser tabs: Fendi-Z/Tugas\_VSGA, 127.0.0.1:8000/update

Address bar: 127.0.0.1:8000/update

Page title: VSGAx2024WDB

Page content: Data berhasil diupdate

	id	nip	nama	tanggal_lahir	nilai	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	1234321	Dono	2001-12-31 00:00:00	3.00	2024-08-15 06:45:58	2024-08-15 07:01:47

11. Buat function Hapus data

```
public function delete() {
    $anggota = Anggotas::find(1);
    $anggota->delete();
    echo "Data berhasil dihapus";
}
```

12. Buka browser ketik url functionnya dan cek pada DB

Browser tabs: Fendi-Z/Tugas\_VSGA, 127.0.0.1:8000/delete

Address bar: 127.0.0.1:8000/delete

Page title: VSGAx2024WDB

Page content: Data berhasil dihapus

13. Tambahkan data lebih dari satu

	id	nip	nama	tanggal_lahir	nilai	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	2	1234322	Dia	2001-12-31 00:00:00	3.20	2024-08-15 07:08:24	2024-08-15 07:08:24
<input type="checkbox"/> Edit Copy Delete	3	1234323	Dina	2001-08-31 00:00:00	3.50	2024-08-15 07:08:58	2024-08-15 07:08:58

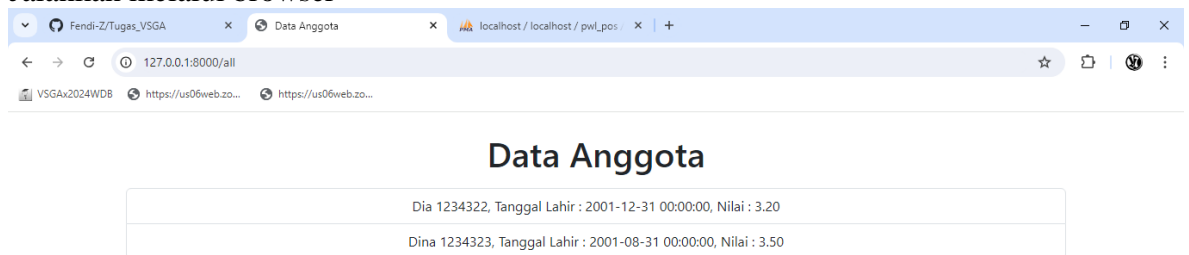
14. Buat function untuk mengirim data ke view

```
public function all() {
    $result = Anggotas::all();
    return view('tampilkan_anggotas', ['anggotas'=>$result]);
}
```

15. Buat tampilan (view)

```
tampilkan_anggotas.blade.php U X
resources > views > @ tamplikan_anggotas.blade.php > html
1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>Data Anggota</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QwTKZypl
```

## 16. Jalankan melalui browser



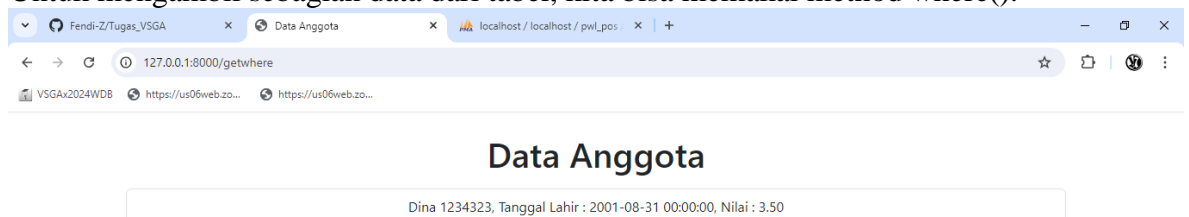
17. Method `find()` bisa dipakai untuk mencari data Model berdasarkan kolom id. Hasil dari method ini langsung berbentuk object, yang sama seperti method `first()`. Berikut contoh penggunaannya:

```
public function find() {
    $result = Anggotas::find(3);
    return view('tampilkan_anggotas', ['anggotas'=>$result]);
}
```

## 18. Jalankan urlnya



19. Untuk mengambil sebagian data dari tabel, kita bisa memakai method `where()`.



### Percobaan 3 : Implementasi Relasi One-to-Many

1. Buat file model dengan nama LevelModel.php menggunakan perintah php artisan make:model LevelModel

```
C:\laragon\www\pwl_pos>php artisan make:model LevelModel
```

```
INFO Model [C:\laragon\www\pwl_pos\app\Models\LevelModel.php] created successfully.
```

2. Buka file model pada LevelModel.php dan tambahkan scripnya menjadi seperti di bawah ini.

```
app > Models > LevelModel.php > LevelModel > users
1  <?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\HasMany;
8
9  class LevelModel extends Model
10 {
11     protected $table = 'm_level';
12     protected $primaryKey = 'level_id';
13
14     public function users():HasMany {
15         return $this->hasMany(UserModel::class);
16     }
17 }
```

3. Pada UserModel.php tambahkan scripnya menjadi seperti di bawah ini.

```
app > Models > UserModel.php > UserModel > level
1  <?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
9  class UserModel extends Model
10 {
11     // use HasFactory;
12
13     protected $table = 'm_user';
14     protected $primaryKey = 'user_id';
15
16     public function level():BelongsTo {
17         return $this->belongsTo(LevelModel::class, 'level_id');
18     }
19 }
```

4. Buka file controller pada UserController.php dan ubah bagian akses UserModel /menampilkan data menjadi seperti di bawah ini.

```
// Akses UserModel
$user = ModelsUserModel::with('level')->get();
return view('user', ['data'=>$user]);
```

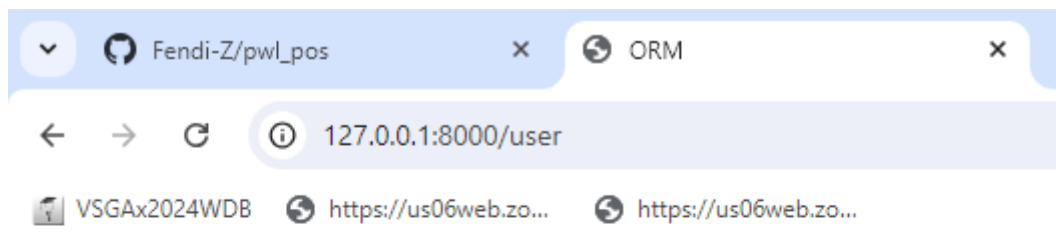
5. Buka file view pada user.blade.php dan ubah script menjadi seperti di bawah ini

```

resources > views > user.blade.php > html > body > table > tr > td
1  <!DOCTYPE html>
2  <html Lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>ORM</title>
8  </head>
9  <body>
10     <h1>Data Level Pengguna</h1>
11     <table border="1" cellpadding="2" cellspacing="0">
12         <tr>
13             <th>ID</th>
14             <th>Username</th>
15             <th>Nama</th>
16             <th>ID Level Pengguna</th>
17             <th>Kode Level</th>
18             <th>Nama Level</th>
19         </tr>
20         @foreach ($data as $item)
21             <tr>
22                 <td>{{ $item->user_id }}</td>
23                 <td>{{ $item->username }}</td>
24                 <td>{{ $item->nama }}</td>
25                 <td>{{ $item->level_id }}</td>
26                 <td>{{ $item->level->level_kode }}</td>
27                 <td>{{ $item->level->level_nama }}</td>
28             </tr>
29         @endforeach
30     </table>
31 </body>
32 </html>

```

6. Simpan kode program. Kemudian jalankan link pada browser.



## Data Level Pengguna

ID	Username	Nama	ID Level Pengguna	Kode Level	Nama Level
1	Admin	Administrator	1	ADM	Administrator
2	Manager	Manager	2	MNG	Managar
3	Staff	Staff/Kasir	3	STF	Staff/Kasir