

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN WEB LANJUT**  
**MODEL dan ELOQUENT ORM**



**Disusun Oleh :**

Qusnul Diah Mawanti

2341760035

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2025**



Jurusan Teknologi Informasi – Politeknik Negeri  
Malang

Jobsheet 4 : Model dan Eloquent ORM

Mata Kuliah : Pemrograman Web Lanjut

Kelas : 2A – SIB

Nama : Qusnul Diah Mawanti

No absen : 25

NIM : 2341760035

Maret 2025

## **PERSIAPAN**

Sebelumnya kita sudah membahas mengenai Migration, Seeder, DB Façade, Query Builder, dan sedikit tentang Eloquent ORM yang ada di Laravel. Sebelum kita masuk pada pembuatan aplikasi berbasis website, alangkah baiknya kita perlu menyiapkan Basis data sebagai tempat menyimpan data-data pada aplikasi kita nanti. Selain itu, umumnya kita perlu menyiapkan juga data awal yang kita gunakan sebelum membuat aplikasi, seperti data user administrator, data pengaturan sistem, dll. Dalam pertemuan kali ini kita akan memahami tentang bagaimana cara menampilkan data, mengubah data, dan menghapus data menggunakan teknik Eloquent.

Sesuai dengan **Studi Kasus PWL.pdf**.

Jadi project Laravel 10 kita masih sama dengan menggunakan repositori **PWL\_POS**.

*Project PWL\_POS* akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

```
C:\laragon\www>composer create-project "laravel/laravel:^10.0" PWL_POS
Creating a "laravel/laravel:^10.0" project at "./PWL_POS"
Installing laravel/laravel (v10.3.3)
- Installing laravel/laravel (v10.3.3): Extracting archive
Created project in C:\laragon\www\PWL_POS
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 111 installs, 0 updates, 0 removals
- Locking brick/math (0.12.3)
```

## DASAR TEORI

### A. PROPERTI `$fillable` DAN `$guarded`

Variable `$fillable` berguna untuk mendaftarkan atribut (nama kolom) yang bisa kita isi ketika melakukan insert atau update ke database. Sebelumnya kita sudah memahami menambahkan record baru ke database. Untuk langkah menambahkan Variable `$fillable` bisa dengan menambahkan script seperti di bawah ini pada file model

#### 1. Praktikum 1: `$fillable`

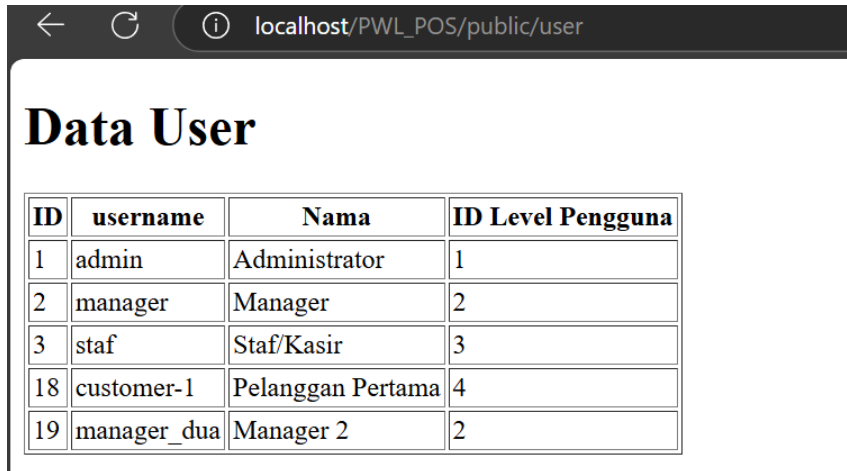
- Buka file model dengan nama `UserModel.php` dan tambahkan `$fillable` seperti gambar di bawah ini.

```
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'; // Mendefinisikan nama tabel digunakan oleh model ini
13     protected $primaryKey = 'user_id'; // Mendefinisikan primary key dari tabel yang digunakan
14     protected $fillable = ['level_id', 'username', 'nama', 'password'];
15 }
```

- Buka file controller dengan nama `UserController.php` dan ubah script untuk menambahkan data baru seperti gambar di bawah ini.

```
app > Http > Controllers > UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         // tambah data user dengan Eloquent Model
14         $data = [
15             'level_id' => 2,
16             'username' => 'manager_dua',
17             'nama' => 'Manager 2',
18             'password' => Hash::make('12345'),
19         ];
20         UserModel::create($data); // tambahkan data ke tabel m_user
21
22         // coba akses model UserModel
23         $user = UserModel::all(); // ambil semua data dari tabel m_user
24         return view('user', ['data' => $user]);
25     }
26 }
```

- c. Simpan kode program Langkah 1 dan 2, dan jalankan perintah web server. Kemudian jalankan link [localhostPWL\\_POS/public/user](http://localhost/PWL_POS/public/user) pada browser dan amati apa yang terjadi.



ID	username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staf	Staf/Kasir	3
18	customer-1	Pelanggan Pertama	4
19	manager_dua	Manager 2	2

**\$fillable** Mendefinisikan atribut yang dapat diisi secara massal.

- d. Ubah file model **UserModel.php** seperti pada gambar di bawah ini pada bagian **\$fillable**

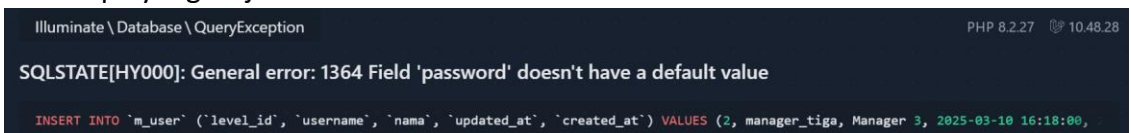
```
protected $fillable = ['level_id', 'username', 'nama'];
```

- e. Ubah kembali file controller **UserController.php** seperti pada gambar di bawah hanya bagian array pada **\$data**

```
public function index()
{
    // tambah data user dengan Eloquent Model
    $data = [
        'level_id' => 2,
        'username' => 'manager_tiga',
        'nama' => 'Manager 3',
        'password' => Hash::make('12345'),
    ];
    UserModel::create($data); // tambahkan data ke tabel m_user

    // coba akses model UserModel
    $user = UserModel::all(); // ambil semua data dari tabel m_user
    return view('user', ['data' => $user]);
}
```

- f. Simpan kode program Langkah 4 dan 5. Kemudian jalankan pada browser dan amati apa yang terjadi.



```
Illuminate\Database\QueryException
SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value
INSERT INTO `m_user` (`level_id`, `username`, `nama`, `updated_at`, `created_at`) VALUES (2, manager_tiga, Manager 3, 2025-03-10 16:18:00,
```

Kolom **password** tidak terdaftar dalam **\$fillable** di **UserModel.php**, sehingga tidak bisa dimasukkan secara mass assignment. Eloquent hanya mengizinkan **mass assignment** pada kolom yang ada di **\$fillable**. Jika kolom tidak dimasukkan, Laravel akan mengabaikan data tersebut saat **create()** dijalankan.

```
protected $fillable = ['level_id', 'username', 'nama', 'password'];
```

# Data User

ID	username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staf	Staf/Kasir	3
18	customer-1	Pelanggan Pertama	4
23	manager_dua	Manager 2	2
26	manager_tiga	Manager 3	2

g. Laporkan hasil Praktikum-1 ini dan *commit* perubahan pada *git*.

## B. RETRIEVING SINGLE MODELS

Selain mengambil semua rekaman yang cocok dengan kueri tertentu, Anda juga dapat mengambil rekaman tunggal menggunakan metode `find`, `first`, atau `firstWhere`. Daripada mengembalikan kumpulan model, metode ini mengembalikan satu contoh model dan dilakukan pada controller:

### 1. Praktikum 2.1: Retrieving Single Models

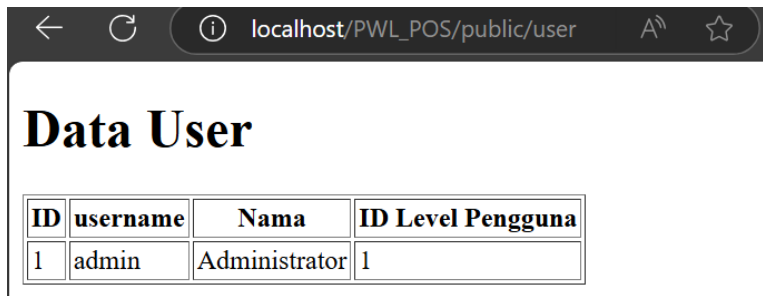
- Buka file controller dengan nama `UserController.php` dan ubah script seperti gambar di bawah ini.

```
$user = UserModel::find(1);  
return view('user', ['data' => $user]);
```

- Buka file view dengan nama `user.blade.php` dan ubah *script* seperti gambar di bawah ini.

```
resources > views > user.blade.php > ...  
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4 <title>Data User</title>  
5 </head>  
6 <body>  
7 <h1>Data User</h1>  
8 <table border="1" cellpadding="2" cellspacing="0">  
9 <tr>  
10 <th>ID</th>  
11 <th>username</th>  
12 <th>Nama</th>  
13 <th>ID Level Pengguna</th>  
14 </tr>  
15 <tr>  
16 <td>{{ $data->user_id }}</td>  
17 <td>{{ $data->username }}</td>  
18 <td>{{ $data->nama }}</td>  
19 <td>{{ $data->level_id }}</td>  
20 </tr>  
21 </table>  
22 </body>  
23 </html>
```

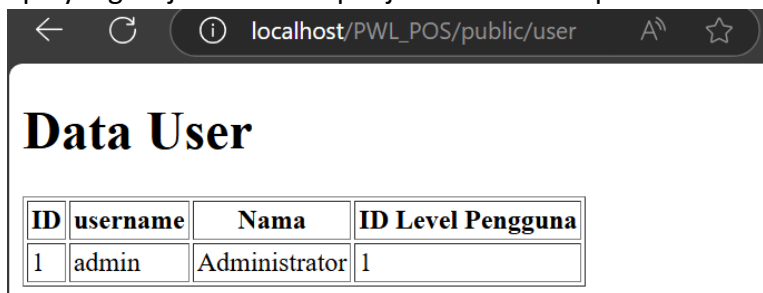
- Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



- d. Ubah file controller dengan nama [UserController.php](#) dan ubah *script* seperti gambar di bawah ini.

```
$user = UserModel::where('level_id', 1)->first();
return view('user', ['data' => $user]);
```

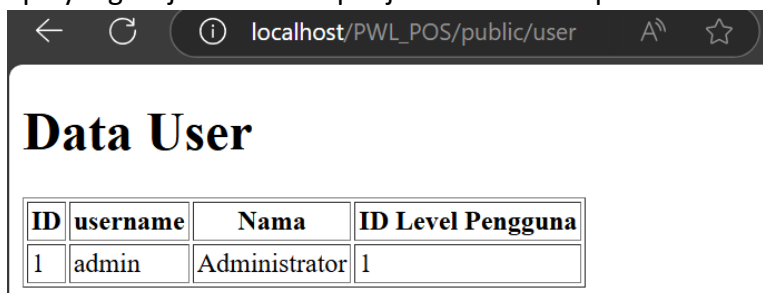
- e. Simpan kode program Langkah 4. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan.



- f. Ubah file controller dengan nama [UserController.php](#) dan ubah *script* seperti gambar di bawah ini.

```
$user = UserModel::firstwhere('level_id', 1);
return view('user', ['data' => $user]);
```

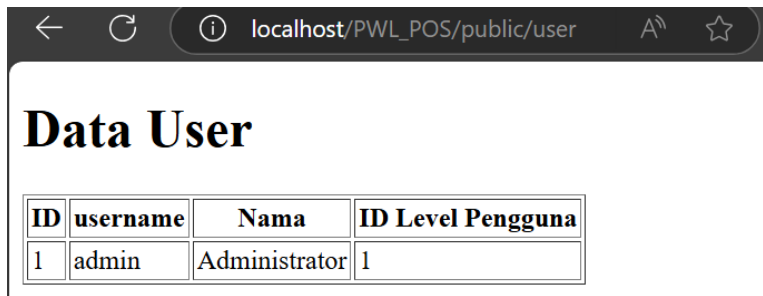
- g. Simpan kode program Langkah 6. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



- h. Ubah file controller dengan nama [UserController.php](#) dan ubah *script* seperti gambar di bawah ini.

```
//Metode findOr
$user = UserModel::findOr(1, ['username', 'nama'], function () {
    abort(404);
});
return view('user', ['data' => $user]);
```

- i. Simpan kode program Langkah 8. Kemudian pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



- j. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini.

```
//Metode findOr
$user = UserModel::findOr(20, ['username', 'nama'], function () {
    abort(404);
});
return view('user', ['data' => $user]);
```

- k. Simpan kode program Langkah 10. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



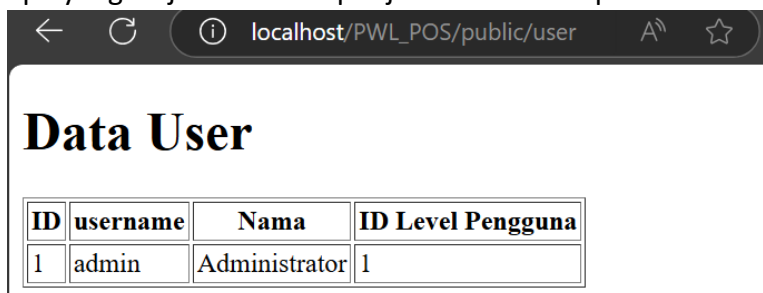
- l. Laporkan hasil Praktikum-2.1 ini dan *commit* perubahan pada *git*.

## 2. Praktikum 2.2: NOT FOUND EXCEPTION

- a. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini.

```
// Metode findOrFail
$user = UserModel::findOrFail(1);
return view('user', ['data' => $user]);
```

- b. Simpan kode program Langkah 1. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



- c. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini.

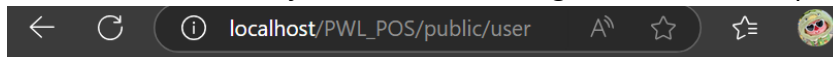
```
// Metode firstOrFail
$user = UserModel::where('username', 'manager9')->firstOrFail();
return view('user', ['data' => $user]);
```

- d. Simpan kode program Langkah 3. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



**NOT FOUND** terjadi karena username manager9 tidak ditemukan pada database sehingga data yang diminta tidak muncul.

Data akan ditemukan jika username manager9 ditambahkan pada database.



## Data User

ID	username	Nama	ID Level Pengguna
29	manager9	Manager 9	4

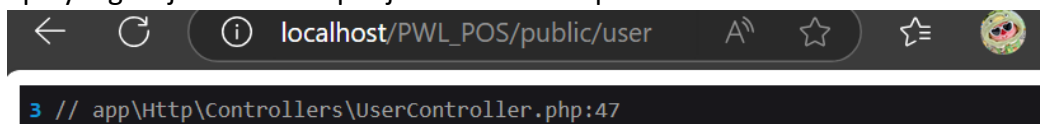
- e. Laporkan hasil Praktikum-2.2 ini dan *commit* perubahan pada *git*.

### 3. Praktikum 2.3: RETREIVING AGGREGATES

- a. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini.

```
$user = UserModel::where('level_id', 2)->count();  
dd($user);  
return view('user', ['data' => $user]);
```

- b. Simpan kode program Langkah 1. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



`dd($user);` akan menghentikan eksekusi sebelumnya, sehingga `return view` tidak akan dijalankan.

- c. Buat agar jumlah *script* pada langkah 1 bisa tampil pada halaman *browser*, sebagai contoh bisa lihat gambar di bawah ini dan ubah *script* pada file *view* supaya bisa muncul datanya.

#### UserController.php

```
// Jumlah pengguna  
$jumlahPengguna = UserModel::where('level_id', 2)->count();  
return view('user', ['jumlahPengguna' => $jumlahPengguna]);
```

#### user.blade.php



```
resources > views > user.blade.php > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Data User</title>
5      </head>
6      <body>
7          <h1>Data User</h1>
8          <table border="1" cellpadding="2" cellspacing="0">
9              <tr>
10                 <th>Jumlah Pengguna</th>
11             </tr>
12             <tr>
13                 <td>{{ $jumlahPengguna }}</td>
14             </tr>
15         </table>
16     </body>
17 </html>
```

## Data User

Jumlah Pengguna
3

d. Laporkan hasil Praktikum-2.3 ini dan *commit* perubahan pada *git*.

### 4. Praktikum 2.4: RETREIVING OR CREATING MODELS

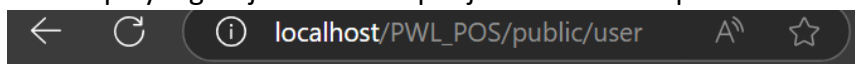
a. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini.

```
// Metode firstOrCreate
$user = UserModel::firstOrCreate(
    [
        'username' => 'manager',
        'nama' => 'Manager',
    ],
);
return view('user', ['data' => $user]);
```

b. Ubah kembali file *view* dengan nama `user.blade.php` dan ubah *script* seperti gambar di bawah ini.

```
resources > views > user.blade.php > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Data User</title>
5 </head>
6 <body>
7 <h1>Data User</h1>
8 <table border="1" cellpadding="2" cellspacing="0">
9 <tr>
10 <th>ID</th>
11 <th>username</th>
12 <th>Nama</th>
13 <th>ID Level Pengguna</th>
14 </tr>
15 <tr>
16 <td>{{ $data->user_id }}</td>
17 <td>{{ $data->username }}</td>
18 <td>{{ $data->nama }}</td>
19 <td>{{ $data->level_id }}</td>
20 </tr>
21 </table>
22 </body>
23 </html>
```

- c. Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



## Data User

ID	username	Nama	ID Level Pengguna
2	manager	Manager	2

- d. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini.

```
$user = UserModel::firstOrCreate(
[
    'username' => 'manager22',
    'nama' => 'Manager Dua Dua',
    'password' => Hash::make('12345'),
    'level_id' => 2
],
);
return view('user', ['data' => $user]);
```

- e. Simpan kode program Langkah 4. Kemudian jalankan pada *browser*

## Data User

ID	username	Nama	ID Level Pengguna
30	manager22	Manager Dua Dua	2

dan amati apa yang terjadi dan cek juga pada *phpMyAdmin* pada tabel [m\\_user](#) serta beri penjelasan dalam laporan.

user_id	level_id	username	nama	password
1	1	admin	Administrator	\$2y\$12\$njFEs2WS04WuYQoKLPnFguW1hFTOz9o.TZeyfdFTI.0...
2	2	manager	Manager	\$2y\$12\$Knb1OTxo6ufl4KH9P.oOCodAvildBxiUCICyxWxtS/P...
3	3	staf	Staf/Kasir	\$2y\$12\$XryQK9sK4uY7mmO0TSsDQeuCXU5PjnYk7xnrczMFCBQ..
18	4	customer-1	Pelanggan Pertama	\$2y\$12\$mVWFqznzmhZqU3Jwvgc0d.jbV4K0Y6GLGXd5u3zslzs...
23	2	manager_dua	Manager 2	\$2y\$12\$FCn.TcUt.SnFKZ1kFuoBROj60GyMWK65Q3S7Q9M3OyX...
26	2	manager_tiga	Manager 3	\$2y\$12\$wtfR6t6c.rmlxQ8vW66hvOilLU67c/1A1Rub31PiJfT...
29	2	manager9	Manager 9	\$2y\$12\$IIHb2K1YxHYvxbLwQakIdeSMxPHbmDgVT9H3IRVydW9...
30	2	manager22	Manager Dua Dua	\$2y\$12\$af5DIbRju7XWl6FdCs2LOShcY/C7EZe9vk7QbGbLYh...

- f. Ubah file controller dengan nama [UserController.php](#) dan ubah *script* seperti gambar di bawah ini.

```
// Metode firstOrCreate
$user = UserModel::firstOrCreate(
    [
        'username' => 'manager',
        'nama' => 'Manager',
    ],
);
return view('user', ['data' => $user]);
```

- g. Simpan kode program Langkah 6. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.

## Data User

ID	username	Nama	ID Level Pengguna
2	manager	Manager	2

- h. Ubah file controller dengan nama [UserController.php](#) dan ubah *script* seperti gambar di bawah ini.

```

$user = UserModel::firstOrCreate(
    [
        'username' => 'manager33',
        'nama' => 'Manager Tiga Tiga',
        'password' => Hash::make('12345'),
        'level_id' => 2
    ],
);
return view('user', ['data' => $user]);

```

- i. Simpan kode program Langkah 8. Kemudian jalankan pada *browser*

## Data User

ID	username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

dan amati apa yang terjadi dan cek juga pada *phpMyAdmin* pada tabel *m\_user* serta beri penjelasan dalam laporan.

user_id	level_id	username	nama	password
1	1	admin	Administrator	\$2y\$12\$njFEs2WS04WuYQoKLPnFguW1hFTOz9o.TZeydFTI.0...
2	2	manager	Manager	\$2y\$12\$Knb1OTxo6ufI4KH9P.oOCOdAvildBxiUCICyxWXS/P...
3	3	staf	Staf/Kasir	\$2y\$12\$XryQK9sK4uY7mmO0TSsDQeuCXU5PjnYk7xnrczMFCBQ...
18	4	customer-1	Pelanggan Pertama	\$2y\$12\$mVWFqznzmhZqU3Jwvgc0d.jbV4K0Y6GLGXd5u3zslzs...
23	2	manager_dua	Manager 2	\$2y\$12\$FCn.TcUt.SnFKZ1kFuoBROj60GyMWK65Q3S7Q9M3OyX...
26	2	manager_tiga	Manager 3	\$2y\$12\$wtfR6t6c.rmlxQ8vW66hvOilU67c/1A1Rub31PiJfT...
29	2	manager9	Manager 9	\$2y\$12\$IIHb2K1YxHYvxbLwQakIdeSMxPHbmDgVT9H3IRVydW9...
30	2	manager22	Manager Dua Dua	\$2y\$12\$af5DIbRjzu7XWl6FdCs2LOShcY/C7EZe9vk7QbGbLYh...

- j. Ubah file controller dengan nama *UserController.php* dan ubah *script* seperti gambar di bawah ini.

```

$user = UserModel::firstOrCreate(
    [
        'username' => 'manager33',
        'nama' => 'Manager Tiga Tiga',
        'password' => Hash::make('12345'),
        'level_id' => 2
    ],
);
$user->save();
return view('user', ['data' => $user]);

```

- k. Simpan kode program Langkah 9. Kemudian jalankan pada *browser*

## Data User

ID	username	Nama	ID Level Pengguna
31	manager33	Manager Tiga Tiga	2

dan amati apa yang terjadi dan cek juga pada *phpMyAdmin* pada tabel *m\_user* serta beri penjelasan dalam laporan.

user_id	level_id	username	nama	password
1	1	admin	Administrator	\$2y\$12\$njFEs2WS04WuYQoKLPnFguW1hFTOz9o.TZeyfdFTI.0...
2	2	manager	Manager	\$2y\$12\$Knb1OTxo6ufl4KH9P.oOCOdAvildBxiUCICyxWXtS/P...
3	3	staf	Staf/Kasir	\$2y\$12\$XryQK9sK4uY7mm00TSsDQeuCXU5PjnYk7xnrczMFCBQ...
18	4	customer-1	Pelanggan Pertama	\$2y\$12\$mVWFqznzmhZqU3Jwvgc0d.jbV4K0Y6GLGXd5u3zslzs...
23	2	manager_dua	Manager 2	\$2y\$12\$FCn.TcUt.SnFKZ1kFuoBROj60GyMWK65Q3S7Q9M3OyX...
26	2	manager_tiga	Manager 3	\$2y\$12\$wtfR6t6c.rmlxQ8vW66hvOilU67c/1A1Rub31PiJfT...
29	2	manager9	Manager 9	\$2y\$12\$IIHb2K1YxHYvxbLwQakIdeSMxPHbmDgVT9H3IRVydW9...
30	2	manager22	Manager Dua Dua	\$2y\$12\$af5DlbRjzu7XWl6FdCs2LOShcY/C7EZe9vk7QbGbLYh...
31	2	manager33	Manager Tiga Tiga	\$2y\$12\$bVRFUV2Huk18Ts8h93ugOT4dNDOeAEqs1GQhnrxWdu...

- Laporkan hasil Praktikum-2.4 ini dan *commit* perubahan pada *git*.

## 5. Praktikum 2.5: ATRIBUT CHANGES

- Ubah file controller dengan nama *UserController.php* dan ubah *script* seperti gambar di bawah ini.

```
// Metode isDirty
$user = UserModel::create([
    'username' => 'manager55',
    'nama' => 'Manager55',
    'password' => Hash::make('12345'),
    'level_id' => 2
]);
$user->username = 'manager56';

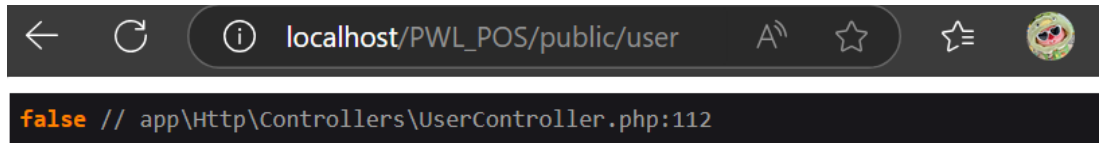
$user->isDirty(); //true
$user->isDirty('username'); //true
$user->isDirty('nama'); //false
$user->isDirty(['nama', 'username']); //true

$user->isClean(); //false
$user->isClean('username'); //false
$user->isClean('nama'); //true
$user->isClean(['nama', 'username']); //false

$user->save();

$user->isDirty(); //false
$user->isClean(); //true
dd($user->isDirty());
```

- b. Simpan kode program Langkah 1. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



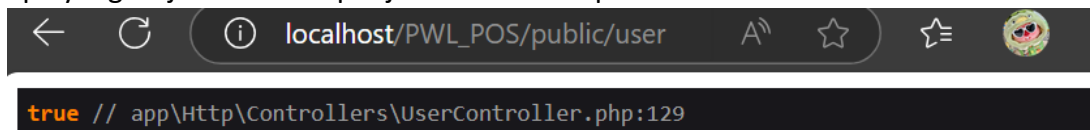
- c. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini.

```
// Metode wasChanged
$user = UserModel::create([
    'username' => 'manager11',
    'nama' => 'Manager11',
    'password' => Hash::make('12345'),
    'level_id' => 2
]);

$user->username = 'manager12';
$user->save();

$user->wasChanged(); //true
$user->wasChanged('username'); //true
$user->wasChanged(['username', 'level_id']); //true
$user->wasChanged('nama'); //false
dd($user->wasChanged(['nama', 'username'])); //true
```

- d. Simpan kode program Langkah 3. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



- e. Laporkan hasil Praktikum-2.5 ini dan *commit* perubahan pada *git*.

## 6. Praktikum 2.6: *CREATE, READ, UPDATE, DELETE (CRUD)*

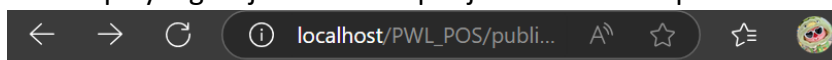
- a. Buka file view pada `user.blade.php` dan buat scripnya menjadi seperti di bawah ini.

```
resources > views > user.blade.php > table > tr > td
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Data User</title>
5 </head>
6 <body>
7 <h1>Data User</h1>
8 <a href="/user/tambah">+ Tambah User</a>
9 <table border="1" cellpadding="2" cellspacing="0">
10 <tr>
11 <td>ID</td>
12 <td>Username</td>
13 <td>Nama</td>
14 <td>ID Level Pengguna</td>
15 <td>Aksi</td>
16 </tr>
17 @foreach ($data as $d)
18 <tr>
19 <td>{{ $d->user_id }}</td>
20 <td>{{ $d->username }}</td>
21 <td>{{ $d->nama }}</td>
22 <td>{{ $d->level_id }}</td>
23 <td><a href="/user/ubah/{{ $d->user_id }}">Ubah</a> |
24 <a href="/user/hapus/{{ $d->user_id }}">Hapus</a>
25 </td>
26 </tr>
27 @endforeach
28 </table>
29 </body>
30 </html>
```

- b. Buka file controller pada `UserController.php` dan buat scriptnya untuk *read* menjadi seperti di bawah ini.

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

- c. Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan.



## Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	<a href="#">Ubah</a>   <a href="#">Hapus</a>
2	manager	Manager	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
3	staf	Staf/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
23	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
26	manager_tiga	Manager 3	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
29	manager9	Manager 9	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
30	manager22	Manager Dua Dua	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
31	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
32	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
33	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>

- d. Langkah berikutnya membuat *create* atau tambah data user dengan cara bikin file baru pada view dengan nama `user_tambah.blade.php` dan buat scriptnya menjadi seperti di bawah ini.

```
resources > views > user_tambah.blade.php > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Data User</title>
5  </head>
6  <body>
7  |   <h1>Form Tambah Data User</h1>
8  |   <form method="post" action="/user/tambah_simpan">
9  |       {{ csrf_field() }}
10 |
11 |       <label>Username</label>
12 |       <input type="text" name="username" placeholder="Masukkan Username">
13 |       <br>
14 |       <label>Nama</label>
15 |       <input type="text" name="nama" placeholder="Masukkan Nama">
16 |       <br>
17 |       <label>Password</label>
18 |       <input type="password" name="password" placeholder="Masukkan Password">
19 |       <br>
20 |       <label>Level ID</label>
21 |       <input type="number" name="level_id" placeholder="Masukkan ID Level">
22 |       <br><br>
23 |       <input type="submit" class="btn btn-success" value="Simpan">
24 |
25 |   </body>
26 </html>
```

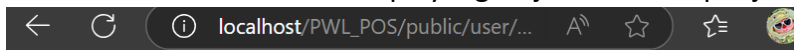
- e. Tambahkan *script* pada routes dengan nama file `web.php`. Tambahkan seperti gambar di bawah ini.

```
Route::get('/user/tambah', [UserController::class, 'tambah']);
```

- f. Tambahkan *script* pada controller dengan nama file `UserController.php`. Tambahkan *script* dalam class dan buat method baru dengan nama `tambah` dan diletakkan di bawah method `index` seperti gambar di bawah ini.

```
public function tambah()
{
    return view('user_tambah');
}
```

- g. Simpan kode program Langkah 4 s/d 6. Kemudian jalankan pada *browser* dan klik link “+ Tambah User” amati apa yang terjadi dan beri penjelasan dalam laporan.



## Form Tambah Data User

Username	<input type="text" value="Masukkan Username"/>
Nama	<input type="text" value="Masukkan Nama"/>
Password	<input type="password" value="Masukkan Password"/>
Level ID	<input type="text" value="Masukkan ID Level"/>
<input type="button" value="Simpan"/>	



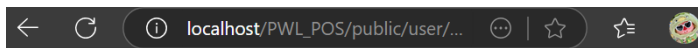
- h. Tambahkan *script* pada routes dengan nama file *web.php*. Tambahkan seperti gambar di bawah ini.

```
Route::get('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);
```

- i. Tambahkan script pada controller dengan nama file *UserController.php*. Tambahkan *script* dalam class dan buat method baru dengan nama **tambah\_simpan** dan diletakan di bawah method tambah seperti gambar di bawah ini.

```
public function tambah_simpan(Request $request)
{
    UserModel::create([
        'username' => $request->username,
        'nama' => $request->nama,
        'password' => Hash::make($request->password),
        'level_id' => $request->level_id
    ]);
    return redirect('/user');
}
```

- j. Simpan kode program Langkah 8 dan 9. Kemudian jalankan link <localhost:8000/user/tambah> atau [localhost/PWL\\_POS/public/user/tambah](localhost/PWL_POS/public/user/tambah) pada *browser*



## Form Tambah Data User

Username

Nama

Password

Level ID

dan input formnya dan simpan, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan.

Data baru berhasil ditambahkan.

## Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	<a href="#">Ubah</a>   <a href="#">Hapus</a>
2	manager	Manager	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
3	staf	Staf/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
23	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
26	manager_tiga	Manager 3	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
29	manager9	Manager 9	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
30	manager22	Manager Dua Dua	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
31	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
32	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
33	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
34	customer-2	Pelanggan ke dua	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>

- k. Langkah berikutnya membuat *update* atau ubah data user dengan cara bikin file baru pada view dengan nama `user_ubah.blade.php` dan buat scriptnya menjadi seperti di bawah ini.

```
resources > views > user_ubah.blade.php > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Data User</title>
5 </head>
6 <body>
7 <h1>Form Ubah Data User</h1>
8 <a href="/user">Kembali</a>
9 <br><br>
10
11 <form method="post" action="/user/ubah_simpan/{{ $data->user_id }}">
12
13     {{ csrf_field() }}
14     {{ method_field('PUT') }}
15
16     <label>Username</label>
17     <input type="text" name="username" placeholder="Masukkan Username" value="{{ $data->username }}">
18     <br>
19     <label>Nama</label>
20     <input type="text" name="nama" placeholder="Masukkan Nama" value="{{ $data->nama }}">
21     <br>
22     <label>Password</label>
23     <input type="password" name="password" placeholder="Masukkan Password" value="{{ $data->password }}">
24     <br>
25     <label>Level ID</label>
26     <input type="number" name="level_id" placeholder="Masukkan ID Level" value="{{ $data->level_id }}">
27     <br><br>
28     <input type="submit" class="btn btn-success" value="Ubah">
29 </body>
30 </html>
```

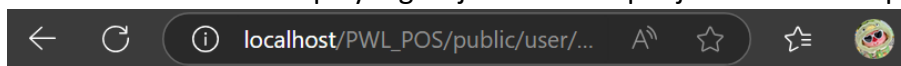
- l. Tambahkan *script* pada routes dengan nama file `web.php`. Tambahkan seperti gambar di bawah ini.

```
Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);
```

- m. Tambahkan *script* pada controller dengan nama file `UserController.php`. Tambahkan *script* dalam class dan buat method baru dengan nama `ubah` dan diletakan di bawah method `tambah_simpan` seperti gambar di bawah ini.

```
public function ubah($id)
{
    $user = UserModel::find($id);
    return view('user_ubah', ['data' => $user]);
}
```

- n. Simpan kode program Langkah 11 sd 13. Kemudian jalankan pada browser dan klik link “Ubah” amati apa yang terjadi dan beri penjelasan dalam laporan.



## Form Ubah Data User

[Kembali](#)

Username

Nama

Password

Level ID

- o. Tambahkan *script* pada *routes* dengan nama file `web.php`. Tambahkan seperti gambar di bawah ini.

```
Route::get('/user/ubah_simpan/{id}', [UserController::class, 'ubah_simpan']);
```

- p. Tambahkan *script* pada controller dengan nama file `UserController.php`. Tambahkan *script* dalam class dan buat method baru dengan nama **ubah\_simpan** dan diletakan di bawah method ubah seperti gambar di bawah ini.

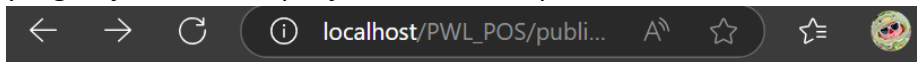
```
public function ubah_simpan($id, Request $request)
{
    $user = UserModel::find($id);

    $user->username = $request->username;
    $user->nama = $request->nama;
    $user->password = Hash::make('$request->password');
    $user->level_id = $request->level_id;

    $user->save();

    return redirect('/user');
}
```

- q. Simpan kode program Langkah 15 dan 16. Kemudian jalankan link `localhost:8000/user/ubah/1` atau `localhost/PWL_POS/public/user/ubah/1` pada *browser* dan ubah input formnya dan klik tombol ubah, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan.



## Form Ubah Data User

[Kembali](#)

Username   
Nama   
Password   
Level ID

Data berhasil diubah

## Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	<a href="#">Ubah</a>   <a href="#">Hapus</a>
2	manager	Manager	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
3	staf	Staf/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
23	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
26	manager_tiga	Manager 3	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
29	manager9	Manager 9	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
30	manager22	Manager Dua Dua	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
31	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
32	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
33	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
34	customer-3	Pelanggan ke tiga	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>

- r. Berikut untuk langkah *delete* . Tambahkan script pada routes dengan nama file [web.php](#). Tambahkan seperti gambar di bawah ini.

```
Route::get('user/hapus/{id}', [UserController::class, 'hapus']);
```

- s. Tambahkan *script* pada controller dengan nama file [UserController.php](#).  
 Tambahkan *script* dalam class dan buat method baru dengan nama hapus dan diletakkan di bawah method ubah\_simpan seperti gambar di bawah ini.

```
public function hapus($id)
{
    $user = UserModel::find($id);
    $user->delete();

    return redirect('/user');
}
```

- t. Simpan kode program Langkah 18 dan 19. Kemudian jalankan pada *browser* dan klik tombol hapus, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan.

Data customer-3 berhasil dihapus.

## Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	<a href="#">Ubah</a>   <a href="#">Hapus</a>
2	manager	Manager	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
3	staf	Staf/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
23	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
26	manager_tiga	Manager 3	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
29	manager9	Manager 9	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
30	manager22	Manager Dua Dua	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
31	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
32	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
33	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>

- u. Laporkan hasil Praktikum-2.6 ini dan *commit* perubahan pada *git*.

## 7. Praktikum 2.7: *RELATIONSHIPS*

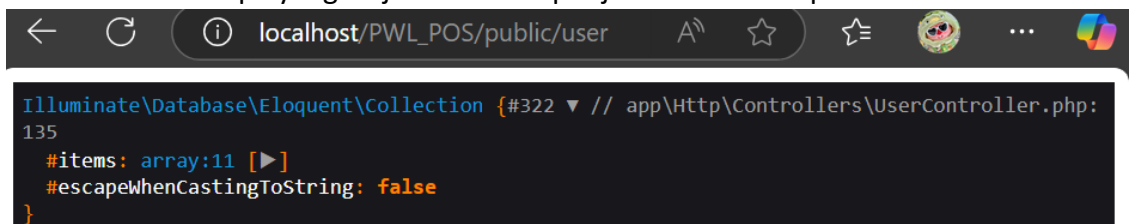
- a. Buka file model pada `UserModel.php` dan tambahkan scripnya menjadi seperti di bawah ini.

```
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  use Illuminate\Database\Eloquent\Relations\BelongsTo;
8
9  class UserModel extends Model
10 {
11     use HasFactory;
12
13     protected $table = 'm_user'; // Mendefinisikan nama tabel digunakan oleh model ini
14     protected $primaryKey = 'user_id'; // Mendefinisikan primary key dari tabel yang digunakan
15     protected $fillable = ['level_id', 'username', 'nama', 'password'];
16
17     public function level(): BelongsTo
18     {
19         return $this->belongsTo(UserModel::class, 'level_id', 'level_id');
20     }
21 }
```

- b. Buka file controller pada `UserController.php` dan ubah method *script* menjadi seperti di bawah ini.

```
$user = UserModel::with('level')->get();
dd($user);
```

- c. Simpan kode program Langkah 2. Kemudian jalankan link pada *browser*, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan.



```
Illuminate\Database\Eloquent\Collection {#322} // app\Http\Controllers\UserController.php:135
#items: array:11 [▶]
#escapeWhenCastingToString: false
}
```

Pesan `Illuminate\Database\Eloquent\Collection {#322}` menunjukkan bahwa hasil query dari `UserModel` menggunakan **Eloquent ORM** berupa **koleksi (Collection) Laravel**, bukan array biasa. Koleksi ini berisi **11 data user**, yang dapat diakses menggunakan metode **Collection Laravel** seperti `first()`, `toArray()`, atau dengan **perulangan (foreach)** untuk menampilkan setiap elemen. Jika ingin melihat isi data secara langsung dalam bentuk array, gunakan `dd($user->toArray())`, sedangkan jika hanya ingin mengambil satu data pertama, gunakan `$user->first()->username`.

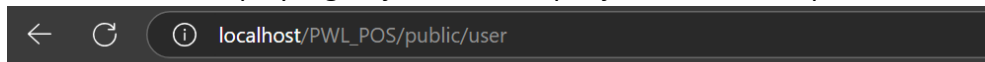
- d. Buka file controller pada `UserController.php` dan ubah method *script* menjadi seperti di bawah ini.

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

- e. 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
2  <html>
6  <body>
33  <h1>Data User</h1>
34  <a href="{{ url('user/tambah') }}">+ Tambah User</a>
35  <table border="1" cellpadding="2" cellspacing="0">
36      <tr>
37          <td>ID</td>
38          <td>Username</td>
39          <td>Nama</td>
40          <td>ID Level Pengguna</td>
41          <td>Kode Level</td>
42          <td>Nama Level</td>
43          <td>Aksi</td>
44      </tr>
45      @foreach ($data as $d)
46          <tr>
47              <td>{{ $d->user_id }}</td>
48              <td>{{ $d->username }}</td>
49              <td>{{ $d->nama }}</td>
50              <td>{{ $d->level_id }}</td>
51              <td>{{ $d->level->level_kode }}</td>
52              <td>{{ $d->level->level_nama }}</td>
53              <td><a href="{{ url('user/ubah/' . $d->user_id) }}">Ubah</a> |
54                  <a href="{{ url('user/hapus/' . $d->user_id) }}">Hapus</a>
55              </td>
56          </tr>
57      @endforeach
58  </table>
59  </body>
```

- f. Simpan kode program Langkah 4 dan 5. Kemudian jalankan link pada *browser*, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan.



## Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Kode Level	Nama Level	Aksi
1	admin	Administrator	1	ADM	Administrator	<a href="#">Ubah</a>   <a href="#">Hapus</a>
2	manager	Manager	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
3	staf	Staf/Kasir	3	STF	Staf/Kasir	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	customer-1	Pelanggan Pertama	4	PLG	Pelanggan	<a href="#">Ubah</a>   <a href="#">Hapus</a>
23	manager_dua	Manager 2	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
26	manager_tiga	Manager 3	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
29	manager9	Manager 9	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
30	manager22	Manager Dua Dua	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
31	manager33	Manager Tiga Tiga	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
32	manager56	Manager55	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
33	manager12	Manager11	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>

- g. Laporkan hasil Praktikum-2.7 ini dan *commit* perubahan pada *git*.