




Mata Kuliah : Pemrograman Web Lanjut (PWL)  
Program Studi : D4 – Sistem Informasi Bisnis  
Semester : 5

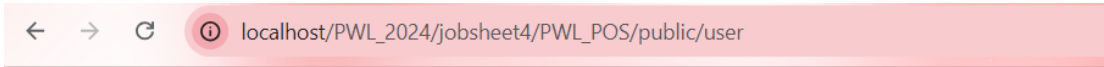
Kelas : SIB  
NIM : 2241760007  
Nama : Anaradi Octa Lavechia  
Jobsheet Ke- : 4 (empat)

## Laporan Jobsheet 04 – MODEL dan ELOQUENT ORM

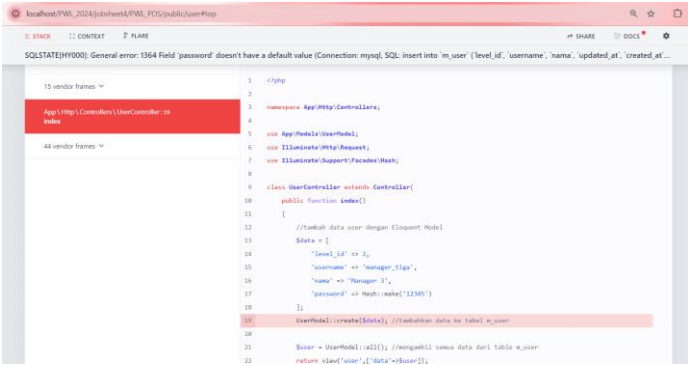
### Praktikum Ke-1- \$fillable

Langkah	Jawaban/Deskripsi
1	<p>Buka file model dengan nama <code>UserModel.php</code> dan tambahkan <code>\$fillable</code> seperti gambar di bawah ini</p> <pre>jobsheet4 &gt; PWL_POS &gt; app &gt; Models &gt;  UserModel.php &gt; ... 1  &lt;?php 2 3  namespace App\Models; 4 5  use Illuminate\Database\Eloquent\Factories\HasFactory; 6  use Illuminate\Database\Eloquent\Model; 7 8  0 references   0 implementations 9  class UserModel extends Model 10 { 11     use HasFactory; 12 13     2 references 14     protected \$table = 'm_user'; //Mendefinisikan nama tabel yang digunakan oleh model ini 15     2 references 16     protected \$primaryKey = 'user_id'; //Mendefinisikan primary key dari table yang digunakan 17 18     /** 19      * The attributes that are mass assignable. 20      * @var array 21      */ 22     protected \$fillable = ['level_id', 'username', 'nama', 'password']; 23 }</pre>
2	<p>Buka file controller dengan nama <code>UserController.php</code> dan ubah <i>script</i> untuk menambahkan data baru seperti gambar di bawah ini</p>



	<pre>jeet4 &gt; PWL_POS &gt; app &gt; Http &gt; Controllers &gt; UserController.php &gt; PHP &gt; UserC 1  &lt;?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     public function index(): Factory View 11     { 12         //tambah data user dengan Eloquent Model 13         \$data = [ 14             'level_id' =&gt; 2, 15             'username' =&gt; 'manager_dua', 16             'nama' =&gt; 'Manager 2', 17             'password' =&gt; Hash::make('12345'), 18         ]; 19         UserModel::create(\$data); //tambahkan data ke table 20 21         \$user = UserModel::all(); //mengambil semua data 22         return view('user',data: ['data'=&gt;\$user]); 23     } 24 } 25</pre>																								
3	<p>Simpan kode program Langkah 1 dan 2, dan jalankan web server. Kemudian jalankan link <a href="localhostPWL_POS/public/user">localhostPWL_POS/public/user</a> pada <i>browser</i> dan amati apa yang terjadi</p> <div></div> <h3>Data User</h3> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr><tr><td>2</td><td>manager</td><td>Manager</td><td>2</td></tr><tr><td>3</td><td>staff</td><td>Staff/Kasir</td><td>3</td></tr><tr><td>14</td><td>customer-1</td><td>Pelanggan Pertama</td><td>4</td></tr><tr><td>15</td><td>manager_dua</td><td>Manager 2</td><td>2</td></tr></table> <p>Berdasarkan pemahaman saya dari hasil praktikum yaitu data yang diinputkan melalui kode di tampilkan di baris terakhir.</p>	ID	Username	Nama	ID Level pengguna	1	admin	Administrator	1	2	manager	Manager	2	3	staff	Staff/Kasir	3	14	customer-1	Pelanggan Pertama	4	15	manager_dua	Manager 2	2
ID	Username	Nama	ID Level pengguna																						
1	admin	Administrator	1																						
2	manager	Manager	2																						
3	staff	Staff/Kasir	3																						
14	customer-1	Pelanggan Pertama	4																						
15	manager_dua	Manager 2	2																						
4	Ubah file model <a href="#">UserModel.php</a> seperti pada gambar di bawah ini pada bagian <a href="#">\$fillable</a>																								



	<pre>bsheet4 &gt; PWL_POS &gt; app &gt; Models &gt; UserModel.php &gt; PHP Intelephense &gt; UserModel &gt; 1  &lt;?php 2 3  namespace App\Models; 4 5  use Illuminate\Database\Eloquent\Factories\HasFactory; 6  use Illuminate\Database\Eloquent\Model; 7 8  9 references   0 implementations 9  class UserModel extends Model 10 { 11     use HasFactory; 12 13     8 references 14     protected \$table = 'm_user'; //Mendefinisikan nama tabel yar 15     8 references 16     protected \$primaryKey = 'user_id'; //Mendefinisikan primary 17 18     /** 19      * The attributes that are mass assignable. 20      * 21      * @var array 22      */ 23     0 references 24     protected \$fillable = ['level_id', 'username', 'nama']; 25 }</pre>
5	<p>Ubah kembali file controller <code>UserController.php</code> seperti pada gambar di bawah hanyabagian array pada <code>\$data</code></p> <pre>jobsheet4 &gt; PWL_POS &gt; app &gt; Http &gt; Controllers &gt; UserController.php &gt; ... 1  &lt;?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  8 references   0 implementations 10 class UserController extends Controller{ 11     3 references   0 overrides 12     public function index(): Factory View 13     { 14         //tambah data user dengan Eloquent Model 15         \$data = [ 16             'level_id' =&gt; 2, 17             'username' =&gt; 'manager_tiga', 18             'nama' =&gt; 'Manager 3', 19             'password' =&gt; Hash::make('12345'), 20         ]; 21         UserModel::create(\$data); //tambahkan data ke tab 22 23         \$user = UserModel::all(); //mengambil semua data 24         return view('user', data: ['data'=&gt;\$user]); 25     } 26 }</pre>
6	<p>Simpan kode program Langkah 4 dan 5. Kemudian jalankan pada <i>browser</i> dan amati apa yang terjadi</p> 



	Berdasarkan pemahaman saya dari hasil praktikum, dikarenakan pada langkah 4 dilakukan penghapusan bagian <i>password</i> dari array <i>\$data</i> . Sehingga tidak ada nilai untuk kolom password yang di-insert, dan karena kolom password diset sebagai NOT NULL tanpa default value, hal ini yang menyebabkan error. Sehingga output yang dihasilkan yaitu <code>SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value</code> yang mana output tersebut menunjukkan bahwa password telah dihapus dari data input sehingga data yang baru diinputkan tidak dapat di <i>insertkan</i> dikarenakan ada satu kolom yang tidak diisi yaitu password.
7	Laporkan hasil Praktikum-1 ini dan <i>commit</i> perubahan pada <i>git</i> .

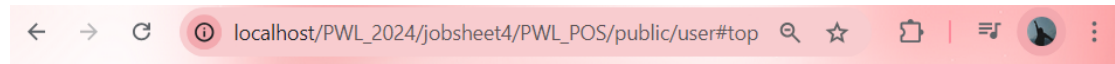
### Praktikum Ke-2.1 – Retrieving Single Models

Langkah	Jawaban/Deskripsi
1	<p>Buka file controller dengan nama <code>UserController.php</code> dan ubah <i>script</i> seperti gambardi bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::find(1); //mengambil semua data dari table m_user         return view(view: 'user',data: ['data'=&gt;\$user]);     } }</pre>
2	<p>Buka file <i>view</i> dengan nama <code>user.blade.php</code> dan ubah <i>script</i> seperti gambar di bawahini</p> <pre>jobsheet4 &gt; PWL_POS &gt; resources &gt; views &gt; user.blade.php &gt; ... 1 &lt;!DOCTYPE html&gt; 2 &lt;html&gt; 3 &lt;head&gt; 4 &lt;title&gt;Data User&lt;/title&gt; 5 &lt;/head&gt; 6 &lt;body&gt; 7 &lt;h1&gt;Data User&lt;/h1&gt; 8 &lt;table border="1" cellpadding="2" cellspacing="0"&gt; 9 &lt;tr&gt; 10 &lt;th&gt;ID&lt;/th&gt; 11 &lt;th&gt;Username&lt;/th&gt; 12 &lt;th&gt;Nama&lt;/th&gt; 13 &lt;th&gt;ID Level pengguna&lt;/th&gt; 14 &lt;/tr&gt; 15 &lt;tr&gt; 16 &lt;td&gt;{{ \$data-&gt;user_id }}&lt;/td&gt; 17 &lt;td&gt;{{ \$data-&gt;username }}&lt;/td&gt; 18 &lt;td&gt;{{ \$data-&gt;nama }}&lt;/td&gt; 19 &lt;td&gt;{{ \$data-&gt;level_id }}&lt;/td&gt; 20 &lt;/tr&gt; 21 &lt;/table&gt; 22 &lt;/body&gt; 23 &lt;/html&gt;</pre>



3

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
1	admin	Administrator	1

Berdasarkan pemahaman saya dari hasil praktikum, dikarenakan pada kode yang diinputkan terdapat kata `find` dengan kode lengkap nya `$user = UserModel::find(1);` yang akan mengambil data dari table `m_user` dengan `ID=1`, maka perintah tersebut menunjukkan pencarian data pada `ID` yang bernilai 1 sehingga output yang di hasilkan *browser* pada hanya satu baris saja dikarenakan yang merupakan id 1 hanya username admin.

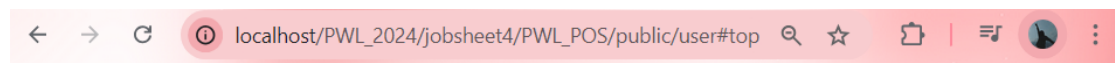
4

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

```
class UserController extends Controller{
    3 references | 0 overrides
    public function index(): Factory|View
    {
        $user = UserModel::where('level_id', 1)->first();
        return view(view: 'user',data: ['data'=>$user]);
    }
}
```

5

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



### Data User

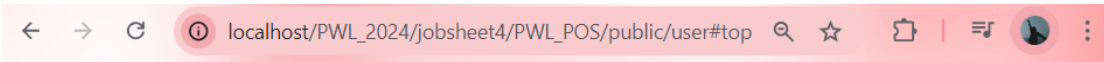
ID	Username	Nama	ID Level pengguna
1	admin	Administrator	1

Berdasarkan pemahaman saya dari hasil praktikum di atas output nya sama dengan praktikum sebelumnya, dikarenakan pada kode yang diinputkan terdapat kata perintah

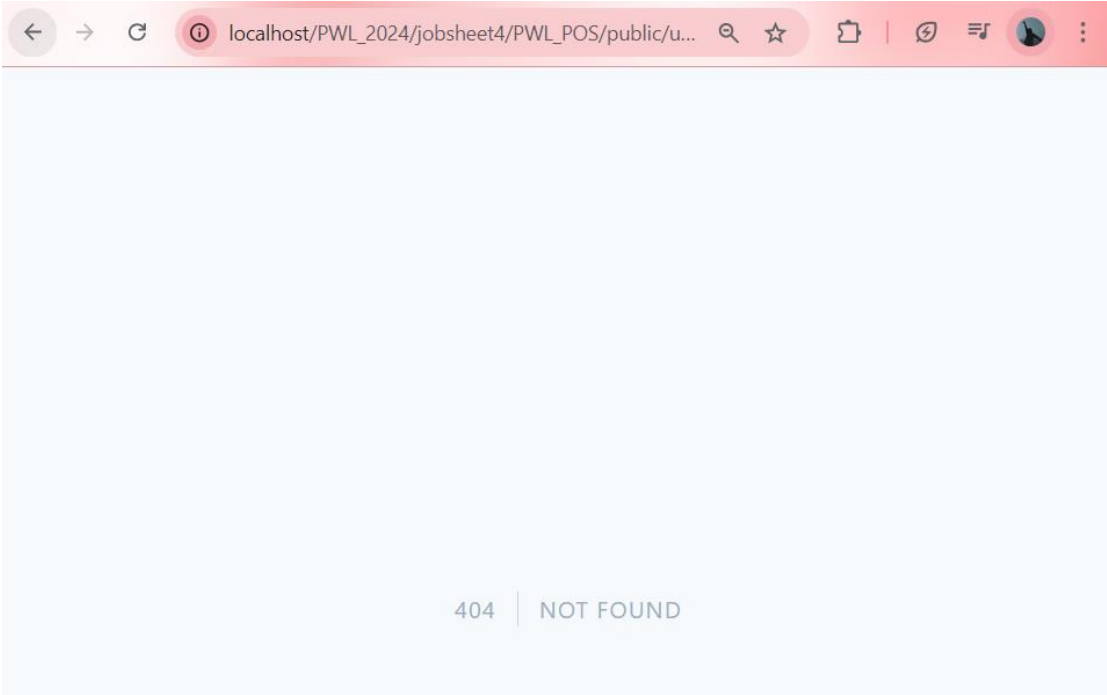


	<p>yang kegunaan nya sama untuk mencari data yaitu perintah <code>where</code> dengan kode lengkap nya <code>\$user = UserModel::where('level_id', 1)-&gt;first();</code> yang akan mengambil data dari table <code>m_user</code> dengan <code>'level_id'</code> bernilai 1, maka perintah tersebut menunjukkan pencarian data pada ID yang bernilai 1 sehingga output yang di hasilkan <i>browser</i> pada hanya satu baris saja dikarenakan yang merupakan id 1 hanya username admin.</p>								
6	<p>Ubah file controller dengan nama <code>UserController.php</code> dan ubah script seperti gambar di bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::firstWhere('level_id', 1);         return view(view: 'user', data: ['data'=&gt;\$user]);     } }</pre>								
7	<p>Simpan kode program Langkah 6. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> <div><div>← → ↻ ⓘ localhost/PWL_2024/jobsheet4/PWL_POS/public/user#top 🔍 ☆ 📁   ≡ 🖱 ⋮</div><div><h3>Data User</h3><table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table><p>Berdasarkan pemahaman saya dari hasil praktikum di atas output nya sama dengan praktikum sebelumnya, dikarenakan pada kode yang diinputkan terdapat kata perintah yang kegunaan nya sama untuk mencari data yaitu perintah <code>where</code> dengan kode lengkap nya <code>\$user = UserModel::firstWhere('level_id', 1);</code> yang akan mengambil data dari table <code>m_user</code> dengan <code>'level_id'</code> bernilai 1, maka perintah tersebut menunjukkan pencarian data pada ID yang bernilai 1 sehingga output yang di hasilkan <i>browser</i> pada hanya satu baris saja dikarenakan yang merupakan id 1 hanya username admin.</p></div></div>	ID	Username	Nama	ID Level pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level pengguna						
1	admin	Administrator	1						



8	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::findOr(1, ['username', 'nama'], function (): never {             abort(code: 404);         });         return view(view: 'user',data: ['data'=&gt;\$user]);     } }</pre>								
9	<p>Simpan kode program Langkah 8. Kemudian pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> <div></div> <p><b>Data User</b></p> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level pengguna</th></tr><tr><td></td><td>admin</td><td>Administrator</td><td></td></tr></table> <p>Berdasarkan pemahaman saya dari hasil praktikum di atas output nya sama dengan praktikum sebelumnya. Namun, pada output ini terdapat sedikit perbedaan dimana pada kolom ID kosong semua dikarenakan inputan perintah dalam controller tidak mengambil kolom ID dari tabel. Hal tersebut ditunjukkan pada kode berikut</p> <pre>\$user = UserModel::findOr(1, ['username', 'nama'], function () {     abort(404); });</pre> <p>Yang mana pada kode tersebut hanya meminta mencari kolom username dan nama sehingga kolom ID tidak diambil dari database.</p>	ID	Username	Nama	ID Level pengguna		admin	Administrator	
ID	Username	Nama	ID Level pengguna						
	admin	Administrator							
10	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p>								



	<pre>class UserController extends Controller{   3 references   0 overrides   public function index(): Factory View   {     \$user = UserModel::findOr(20, ['username', 'nama'], function (): never {       abort(code: 404);     });     return view(view: 'user', data: ['data'=&gt;\$user]);   } }</pre>
11	<p>Simpan kode program Langkah 10. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p>  <p>Berdasarkan hasil praktikum menunjukkan bahwa 404   NOT FOUND yang menunjukkan bahwa data ke 20 tidak dapat ditemukan dikarenakan pada database jumlah data belum mencapai 20.</p>
12	Laporkan hasil Praktikum-2.1 ini dan commit perubahan pada git.

### Praktikum Ke-2.2 – *Not Found Exceptions*

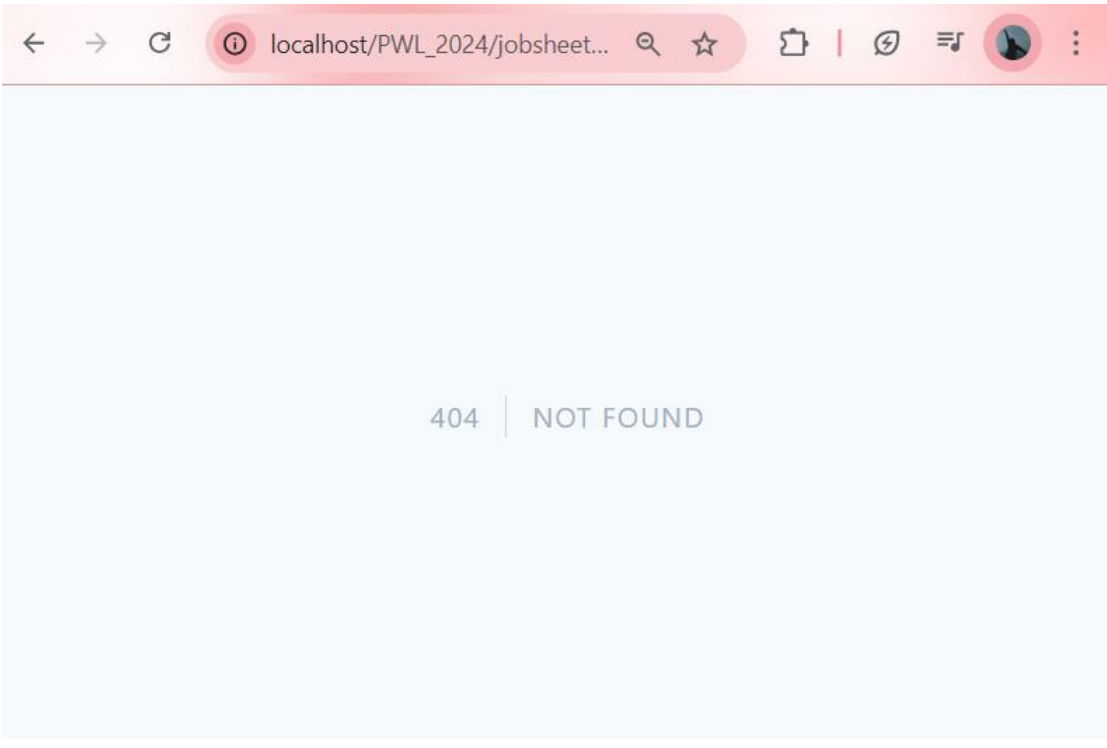
Langkah	Jawaban/Deskripsi
1	Ubah file controller dengan nama <code>UserController.php</code> dan ubah <i>script</i> seperti





	<p>gambardi bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::findOrFail(1);         return view(view: 'user',data: ['data'=&gt;\$user]);     } }</pre>								
2	<p>Simpan kode program Langkah 1. Kemudian jalankan pada <i>browser</i> dan amati apa yangterjadi dan beri penjelasan dalam laporan</p> <div><div><div><div></div><div></div><div></div></div><div>localhost/PWL_2024/jobsheet4/PWL_POS/public/u...</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div> <p><b>Data User</b></p> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table> <p>Berdasarkan pemahaman saya dari hasil praktikum di atas output nya sama dengan praktikum sebelumnya. Dimana perintah yang diinputkan adalah ID 1 yang akan ditampilkan pada view user. <code>UserModel::findOrFail(1)</code> akan mencari data pengguna dengan ID 1 di tabel yang terhubung</p>	ID	Username	Nama	ID Level pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level pengguna						
1	admin	Administrator	1						
3	<p>Ubah file controller dengan nama <code>UserController.php</code> dan ubah <i>script</i> seperti gambardi bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::where('username', 'manager9')-&gt;firstOrFail();         return view(view: 'user',data: ['data'=&gt;\$user]);     } }</pre>								
4	<p>Simpan kode program Langkah 3. Kemudian jalankan pada <i>browser</i> dan amati apa yangterjadi dan beri penjelasan dalam laporan</p>								



	
	Berdasarkan hasil praktikum menunjukkan bahwa 404   NOT FOUND yang menunjukkan bahwa pencarian data berdasarkan username manager9 tidak dapat sehingga output yang dihasilkan adalah error.
5	Laporkan hasil Praktikum-2.2 ini dan <i>commit</i> perubahan pada <i>git</i> .

### Praktikum Ke-2.3 – Retrieving Aggregates

Langkah	Jawaban/Deskripsi
1	<p>Ubah file controller dengan nama <code>UserController.php</code> dan ubah <i>script</i> seperti gambar di bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::where('level_id', 2)-&gt;count();         dd(vars: \$user);          return view(view: 'user', data: ['data'=&gt;\$user]);     } }</pre>



2

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



Berdasarkan pemahaman saya dari hasil praktikum di atas output nya terjadi error dikarenakan view nya belum terdapat modifikasi sehingga output yang dihasilkan tidak sesuai dengan apa yang diinginkan di *controller*.

3

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

- Berikut adalah inputan perubahan pada file `user.blade.php` beserta output yang dihasilkan:

```
DS > resources > views > user.blade.php > html > body > h1 > h1 > table >
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 Pengguha</th>
11             </tr>
12             <tr>
13                 <td>{{ $data }}</td>
14             </tr>
15         </table>
16     </body>
17 </html>
```



	<div><div><div><div><div><div><span></span></div><div><span></span></div></div><div><div><span></span></div><div><span></span></div></div><div><div><span></span></div><div><span></span></div></div></div><div>localhost/PWL_2024/jobsheet4/PWL_POS/public/user#top</div><div><div><div><span></span></div><div><span></span></div><div><span></span></div></div><div><div><span></span></div><div><span></span></div></div></div></div></div><div><h2>Data User</h2><table><tr><td>Jumlah Pengguna</td></tr><tr><td>2</td></tr></table><p>Berdasarkan hasil praktikum menunjukkan bahwa Data User jumlah berguna ada 2 yang ditunjukkan dalam bentuk tabel.</p></div></div>	Jumlah Pengguna	2
Jumlah Pengguna			
2			
4	Laporkan hasil Praktikum-2.3 ini dan <i>commit</i> perubahan pada <i>git</i> .		

#### Praktikum Ke-2.4 – Retrieving or Creating Models

Langkah	Jawaban/Deskripsi
1	<p>Ubah file controller dengan nama <code>UserController.php</code> dan ubah <i>script</i> seperti gambardi bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::firstOrCreate(             [                 'username' =&gt; 'manager',                 'nama' =&gt; 'Manager',             ],         );         return view(view: 'user', data: ['data' =&gt; \$user]);     } }</pre>
2	<p>Ubah kembali file <i>view</i> dengan nama <code>user.blade.php</code> dan ubah <i>script</i> seperti gambar dibawah ini</p>



```
OS > resources > views > user.blade.php > html > body > h1 > h1 > table
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>
```

3

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

localhost/PWL\_2024/jobsheet4/PWL\_POS/public/user#top

## Data User

ID	Username	Nama	ID Level pengguna
2	manager	Manager	2

Berdasarkan hasil praktikum menunjukkan tabel yang memuat tentang Data User manager yang mana sesuai dengan data pengguna yang diinputkan di controller.

4

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

```
class UserController extends Controller{
    3 references | 0 overrides
    public function index(): Factory|View
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager22',
                'nama' => 'Manager Dua Dua',
                'password' => Hash::make('12345'),
                'level_id' => 2
            ],
        );
        return view(view: 'user', data: ['data' => $user]);
    }
}
```



5

Simpan kode program Langkah 4. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan cek juga pada *phpMyAdmin* pada tabel **m\_user** serta beri penjelasan dalam laporan

The screenshot shows a web browser displaying a PHP error message. The error message is: "Illuminate\Database\QueryException SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value". Below the error message, there is a table with columns: user\_id, level\_id, username, nama, and password. The table contains 5 rows of data.

user_id	level_id	username	nama	password
1	1	admin	Administrator	\$2y\$12\$5OkA9UTD2
2	2	manager	Manager	\$2y\$12\$SeFpeouVJE
3	3	staff	Staff/Kasir	\$2y\$12\$t/C5/dCoP4C
14	4	customer-1	Pelanggan Pertama	\$2y\$12\$0a0Wup0KXg
15	2	manager_dua	Manager 2	\$2y\$12\$3MSFk4gJn

Berdasarkan pemahaman saya dari hasil praktikum di atas output nya terjadi error dikarenakan

6

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



```
class UserController extends Controller{
    3 references | 0 overrides
    public function index(): Factory|View
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager',
                'nama' => 'Mahager',
            ],
        );
        return view(view: 'user', data: ['data' => $user]);
    }
}
```

7

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

localhost/PWL\_2024/jobsheet4/PWL\_POS/public/user#top

### Data User

ID	Username	Nama	ID Level pengguna
2	manager	Manager	2

Berdasarkan pemahaman saya dari hasil praktikum di atas output nya berupa tabel yang menampilkan data user yang diinputkan pada file `firstOrCreate` dikarenakan pada database data manager sudah ada maka ditampilkanlah data tersebut.

8

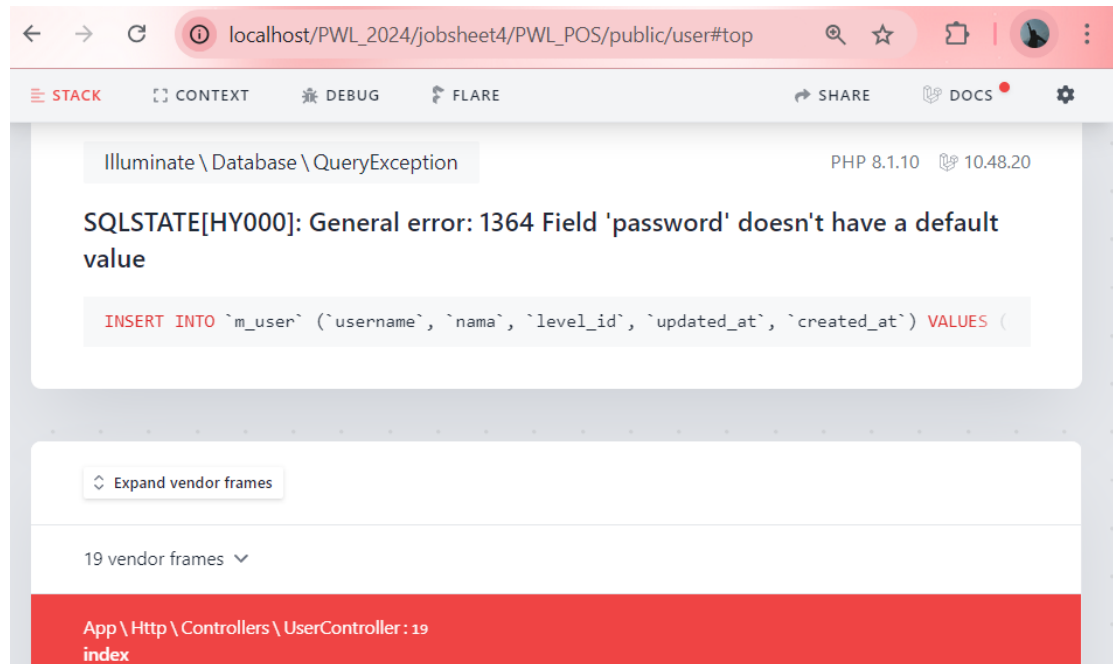
Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini

```
class UserController extends Controller{
    3 references | 0 overrides
    public function index(): Factory|View
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager33',
                'nama' => 'Manager Tiga Tiga',
                'password' => Hash::make('12345'),
                'level_id' => 2
            ],
        );
        return view(view: 'user', data: ['data' => $user]);
    }
}
```



9

Simpan kode program Langkah 8. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan cek juga pada *phpMyAdmin* pada tabel *m\_user* serta beri penjelasan dalam laporan



Berdasarkan pemahaman saya dari hasil praktikum di atas output nya error dikarenakan pada *fillable* tidak ada kolom password sehingga ketika dimasukan inputan seperti diatas tidak dapat diproses ke dalam database.

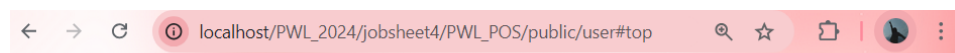
10

Ubah file controller dengan nama *UserController.php* dan ubah *script* seperti ini

```
$user->save();  
return view(view: 'user', data: ['data' => $user]);
```

11

Simpan kode program Langkah 9. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan cek juga pada *phpMyAdmin* pada tabel *m\_user* serta beri penjelasan dalam laporan



### Data User

ID	Username	Nama	ID Level pengguna
16	manager33	Manager Tiga Tiga	2



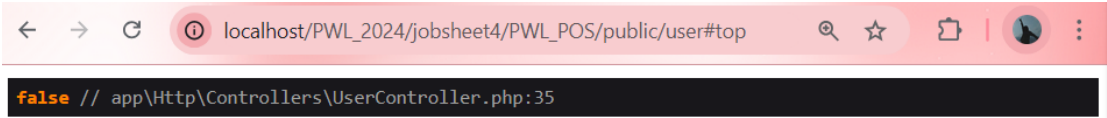

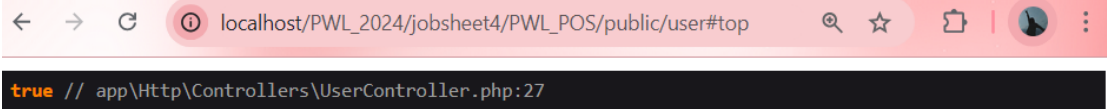


12	Laporkan hasil Praktikum-2.4 ini dan <i>commit</i> perubahan pada <i>git</i> .

### Praktikum Ke-2.5 – *Attribute Changes*

Langkah	Jawaban/Deskripsi
1	<p>Ubah file controller dengan nama <code>UserController.php</code> dan ubah <i>script</i> seperti gambardi bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): never     {         \$user = UserModel::create([             'username' =&gt; 'manager55',             'nama' =&gt; 'Manager55',             'password' =&gt; Hash::make('12345'),             'level_id' =&gt; 2         ]);          \$user-&gt;username = 'manager56';          \$user-&gt;isDirty(); //true         \$user-&gt;isDirty('username'); //true         \$user-&gt;isDirty('nama'); //false         \$user-&gt;isDirty(['nama', 'username']); //true          \$user-&gt;isClean(); //false         \$user-&gt;isClean('username'); //false         \$user-&gt;isClean('nama'); //true         \$user-&gt;isClean(['nama', 'username']); //false          \$user-&gt;save();          \$user-&gt;isDirty(); //false         \$user-&gt;isClean(); //true         dd(vars: \$user-&gt;isDirty());     } }</pre>



2	<p>Simpan kode program Langkah 1. Kemudian jalankan pada <i>browser</i> dan amati apa yang terjadi dan beri penjelasan dalam laporan</p>  <p>Berdasarkan pemahaman saya dari hasil praktikum di atas output nya false dikarenakan setelah perintah <code>\$user-&gt;save()</code> dipanggil, semua perubahan telah disimpan, sehingga model tidak lagi dianggap "dirty," dan <code>isDirty()</code> mengembalikan false.</p>
3	<p>Ubah file controller dengan nama <code>UserController.php</code> dan ubah <i>script</i> seperti gambardi bawah ini</p>  <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): never     {         \$user = UserModel::create([             'username' =&gt; 'manager11',             'nama' =&gt; 'Manager11',             'password' =&gt; Hash::make('12345'),             'level_id' =&gt; 2         ]);          \$user-&gt;username = 'manager12';          \$user-&gt;save();          \$user-&gt;wasChanged(); //true         \$user-&gt;wasChanged('username'); //true         \$user-&gt;wasChanged(['username', 'level_id']); //true         \$user-&gt;wasChanged('nama'); //false         dd(vars: \$user-&gt;wasChanged(['nama', 'username'])); //tr     } }</pre>
4	<p>Simpan kode program Langkah 3. Kemudian jalankan pada <i>browser</i> dan amati apa yang terjadi dan beri penjelasan dalam laporan</p>  <p>Berdasarkan pemahaman saya dari hasil praktikum di atas output nya true dikarenakan pada baris <code>`dd(\$user-&gt;wasChanged(['nama',</code></p>



	'username']]));` terjadi metode `wasChanged()` yang memeriksa apakah ada perubahan pada salah satu atau lebih dari kolom yang disebutkan. Meskipun kolom `nama` tidak berubah, kolom `username` diubah dari `manager11` menjadi `manager12`. Karena ada perubahan pada kolom `username`, maka hasilnya `true`, karena cukup satu kolom yang berubah dari yang diperiksa (`nama` atau `username`).
5	Laporkan hasil Praktikum-2.5 ini dan <i>commit</i> perubahan pada <i>git</i> .

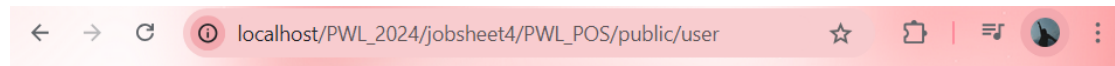
### Praktikum Ke-2.6 – *Create, Read, Update, Delete (CRUD)*

Langkah	Jawaban/Deskripsi
1	<p>Buka file <i>view</i> pada <i>user.blade.php</i> dan buat scripnya menjadi seperti di bawah ini</p> <pre>jobsheet4 &gt; PWL_POS &gt; resources &gt; views &gt; user.blade.php &gt; html &gt; body &gt; table &gt; tr &gt; th 1  &lt;!DOCTYPE html&gt; 2  &lt;html&gt; 3      &lt;head&gt; 4          &lt;title&gt;Data User&lt;/title&gt; 5      &lt;/head&gt; 6      &lt;body&gt; 7          &lt;h1&gt;Data User&lt;/h1&gt; 8          &lt;a href="/user/tambah"&gt;+ Tambah User&lt;/a&gt; 9          &lt;table border="1" cellpadding="2" cellspacing="0"&gt; 10             &lt;tr&gt; 11                 &lt;th&gt;ID&lt;/th&gt; 12                 &lt;th&gt;Username&lt;/th&gt; 13                 &lt;th&gt;Nama&lt;/th&gt; 14                 &lt;th&gt;ID Level pengguna&lt;/th&gt; 15                 &lt;th&gt;Aksi&lt;/th&gt; 16             &lt;/tr&gt; 17             @foreach (\$data as \$d) 18                 &lt;tr&gt; 19                     &lt;td&gt;{{ \$d-&gt;user_id }}&lt;/td&gt; 20                     &lt;td&gt;{{ \$d-&gt;username }}&lt;/td&gt; 21                     &lt;td&gt;{{ \$d-&gt;nama }}&lt;/td&gt; 22                     &lt;td&gt;{{ \$d-&gt;level_id }}&lt;/td&gt; 23                     &lt;td&gt;&lt;a href="/user/ubah/{{ \$d-&gt;user_id }}"&gt;Ubah&lt;/a&gt;   &lt;a href="/user/hapus/{{ \$d-&gt;user_id }}"&gt;Hapus&lt;/a&gt;&lt;/td&gt; 24                 &lt;/tr&gt; 25             @endforeach 26         &lt;/table&gt; 27     &lt;/body&gt; 28 &lt;/html&gt;</pre>
2	<p>Buka file controller pada <i>UserController.php</i> dan buat scriptnya untuk <i>read</i> menjadi seperti di bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::all();         return view(view: 'user', data: ['data' =&gt; \$user]);     } }</pre>



3

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	staff	Staff/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
14	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
15	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
16	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
17	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>

Berdasarkan pemahaman saya dari hasil praktikum di atas submenu +Tambah User telah muncul di atas tabel data user dan ketika diklik terdapat output NOT FOUND.

4

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

```
jobsheet4 > PWL_POS > resources > views > user_tambah.blade.php > html > body > form
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Data Tambah User</title>
5  </head>
6  <body>
7  |   <h1>Form Tambah Data User</h1>
8  |   <form method="post" action="/PWL_2024/jobsheet4/PWL_POS/public/user/tambah_simpan">
9  |       {{ csrf_field() }}
10 |       <label>Username</label>
11 |       <input type="text" id="username" name="username" placeholder="Masukkan Username">
12 |       <br>
13 |       <label>Nama</label>
14 |       <input type="text" name="nama" placeholder="Masukkan Nama">
15 |       <br>
16 |       <label>Password</label>
17 |       <input type="password" name="password" placeholder="Masukkan Password">
18 |       <br>
19 |       <label>Level ID</label>
20 |       <input type="number" name="level_id" placeholder="Masukkan ID Level">
21 |       <br>
22 |       <input type="submit" class="btn btn-success" value="Simpan">
23 |   </form>
24 </body>
25 </html>
```

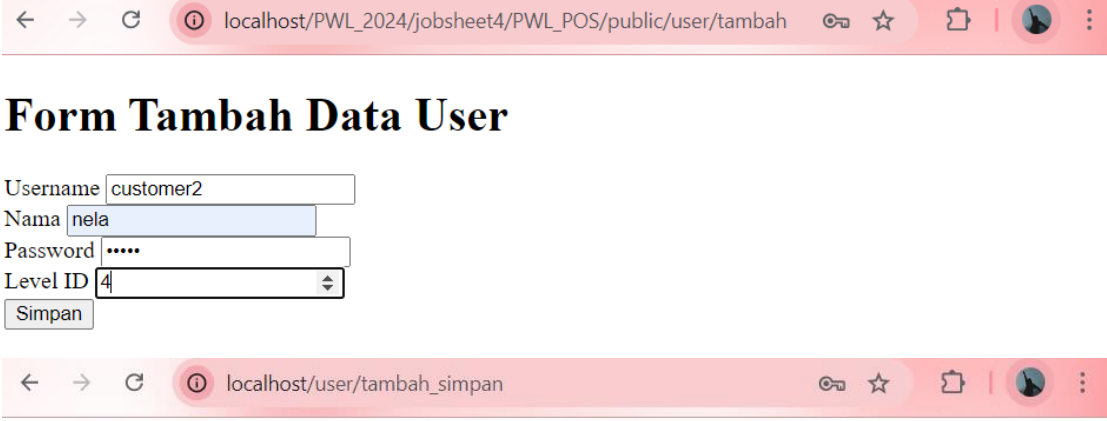

5

Tambahkan *script* pada *routes* dengan nama file `web.php`. Tambahkan seperti gambar



	<p>dibawah ini</p> <pre>jobsheet4 &gt; PWL_POS &gt; routes &gt; web.php &gt; ... 1  &lt;?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 (): Factory View{ 9      return view(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']); 15 Route::get('/user/tambah', [UserController::class, 'tambah']); 16 17 ?&gt;</pre>
6	<p>Tambahkan <i>script</i> pada controller dengan nama file <i>UserController.php</i>. Tambahkan <i>script</i> dalam class dan buat method baru dengan nama tambah dan diletakan di bawah method index seperti gambar di bawah ini</p> <pre>class UserController extends Controller{     3 references   0 overrides     public function index(): Factory View     {         \$user = UserModel::all();         return view(view: 'user', data: ['data' =&gt; \$user]);     }      1 reference   0 overrides     public function tambah(): Factory View     {         return view(view: 'user_tambah');     } }</pre>
7	<p>Simpan kode program Langkah 4 s/d 6. Kemudian jalankan pada <i>browser</i> dan klik link “+ <b>Tambah User</b>” amati apa yang terjadi dan beri penjelasan dalam laporan</p>



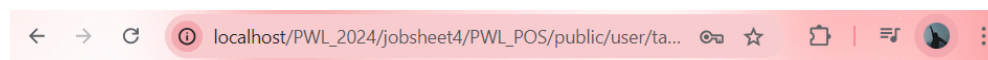
	 <p><b>Form Tambah Data User</b></p> <p>Username <input type="text" value="customer2"/></p> <p>Nama <input type="text" value="nela"/></p> <p>Password <input type="password" value="....."/></p> <p>Level ID <input type="text" value="4"/></p> <p><input type="button" value="Simpan"/></p> <p><b>Not Found</b></p> <p>The requested URL was not found on this server.</p> <p>Berdasarkan pemahaman saya dari hasil praktikum di atas menu Tambah User telah muncul ketika submenu +Tambah User diklik namun setelah penginputan data dan di klik button simpan terdapat output NOT FOUND yang dikarenakan belum adanya a href ke halaman berikutnya setelah penyimpanan data.</p>
8	<p>Tambahkan <i>script</i> pada <i>routes</i> dengan nama file <a href="#">web.php</a>. Tambahkan seperti gambar dibawah ini</p>  <pre>jobsheet4 &gt; PWL_POS &gt; routes &gt; web.php &gt; ... 1  &lt;?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 () { Factory View{ 9      return view(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']); 15 Route::get('/user/tambah', [UserController::class, 'tambah']); 16 17 Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']); 18 ?&gt;</pre>
9	<p>Tambahkan <i>script</i> pada controller dengan nama file <i>UserController.php</i>. Tambahkan <i>script</i> dalam class dan buat method baru dengan nama tambah_simpan dan diletakan dibawah method tambah seperti gambar di bawah ini</p>



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

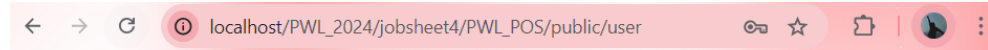
10

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



## Form Tambah Data User

Username   
Nama   
Password   
Level ID



## 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	staff	Staff/Kasir	3	<a href="#">Ubah</a> <a href="#">Hapus</a>
14	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a> <a href="#">Hapus</a>
15	manager_dua	Manager 2	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
16	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
17	manager56	Manager55	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
18	manager12	Manager11	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
19	customer2	Pelanggan	4	<a href="#">Ubah</a> <a href="#">Hapus</a>

Berdasarkan pemahaman saya dari hasil praktikum di atas menu Tambah User setelah penginputan data dan di klik button simpan terdapat output kembali ke menu data user dalam tabel yang merupakan menu sebelumnya dan menunjukkan apakah data yang diinputkan sebelumnya sudah berhasil diinputkan atau belum dengan dilihat pada tabel data user apakah data tersebut sudah ada atau belum.





	<div><div>← T →</div><table><tr><th></th><th></th><th></th><th>user_id</th><th>level_id</th><th>username</th><th>nama</th><th>password</th></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>1</td><td>1</td><td>admin</td><td>Administrator</td><td>\$2y\$12\$5OkA9UTD28mmOSbKTX</td></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>2</td><td>2</td><td>manager</td><td>Manager</td><td>\$2y\$12\$SeFpeouVjEX4zgpawMeS</td></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>3</td><td>3</td><td>staff</td><td>Staff/Kasir</td><td>\$2y\$12\$t/C5/dCoP4CYr7B/Jkmnuu</td></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>14</td><td>4</td><td>customer-1</td><td>Pelanggan Pertama</td><td>\$2y\$12\$0a0Wup0Kxg1BTi5VrRP19</td></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>15</td><td>2</td><td>manager_dua</td><td>Manager 2</td><td>\$2y\$12\$3MSFk4gJnho2o675OJF4</td></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>16</td><td>2</td><td>manager33</td><td>Manager Tiga Tiga</td><td>\$2y\$12\$69Xn5NUUguNTz2aI9gJgI</td></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>17</td><td>2</td><td>manager56</td><td>Manager55</td><td>\$2y\$12\$xbNHPHrm09jdTxQE40gny</td></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>18</td><td>2</td><td>manager12</td><td>Manager11</td><td>\$2y\$12\$1gJLzJ1ISFdimyp3OSpV4</td></tr><tr><td><input type="checkbox"/></td><td> Edit</td><td> Copy</td><td> Delete</td><td>19</td><td>4</td><td>customer2</td><td>Pelanggan</td><td>\$2y\$12\$vivqINZcV/e0J58atghkdusf</td></tr></table></div>				user_id	level_id	username	nama	password	<input type="checkbox"/>	Edit	Copy	Delete	1	1	admin	Administrator	\$2y\$12\$5OkA9UTD28mmOSbKTX	<input type="checkbox"/>	Edit	Copy	Delete	2	2	manager	Manager	\$2y\$12\$SeFpeouVjEX4zgpawMeS	<input type="checkbox"/>	Edit	Copy	Delete	3	3	staff	Staff/Kasir	\$2y\$12\$t/C5/dCoP4CYr7B/Jkmnuu	<input type="checkbox"/>	Edit	Copy	Delete	14	4	customer-1	Pelanggan Pertama	\$2y\$12\$0a0Wup0Kxg1BTi5VrRP19	<input type="checkbox"/>	Edit	Copy	Delete	15	2	manager_dua	Manager 2	\$2y\$12\$3MSFk4gJnho2o675OJF4	<input type="checkbox"/>	Edit	Copy	Delete	16	2	manager33	Manager Tiga Tiga	\$2y\$12\$69Xn5NUUguNTz2aI9gJgI	<input type="checkbox"/>	Edit	Copy	Delete	17	2	manager56	Manager55	\$2y\$12\$xbNHPHrm09jdTxQE40gny	<input type="checkbox"/>	Edit	Copy	Delete	18	2	manager12	Manager11	\$2y\$12\$1gJLzJ1ISFdimyp3OSpV4	<input type="checkbox"/>	Edit	Copy	Delete	19	4	customer2	Pelanggan	\$2y\$12\$vivqINZcV/e0J58atghkdusf
			user_id	level_id	username	nama	password																																																																																			
<input type="checkbox"/>	Edit	Copy	Delete	1	1	admin	Administrator	\$2y\$12\$5OkA9UTD28mmOSbKTX																																																																																		
<input type="checkbox"/>	Edit	Copy	Delete	2	2	manager	Manager	\$2y\$12\$SeFpeouVjEX4zgpawMeS																																																																																		
<input type="checkbox"/>	Edit	Copy	Delete	3	3	staff	Staff/Kasir	\$2y\$12\$t/C5/dCoP4CYr7B/Jkmnuu																																																																																		
<input type="checkbox"/>	Edit	Copy	Delete	14	4	customer-1	Pelanggan Pertama	\$2y\$12\$0a0Wup0Kxg1BTi5VrRP19																																																																																		
<input type="checkbox"/>	Edit	Copy	Delete	15	2	manager_dua	Manager 2	\$2y\$12\$3MSFk4gJnho2o675OJF4																																																																																		
<input type="checkbox"/>	Edit	Copy	Delete	16	2	manager33	Manager Tiga Tiga	\$2y\$12\$69Xn5NUUguNTz2aI9gJgI																																																																																		
<input type="checkbox"/>	Edit	Copy	Delete	17	2	manager56	Manager55	\$2y\$12\$xbNHPHrm09jdTxQE40gny																																																																																		
<input type="checkbox"/>	Edit	Copy	Delete	18	2	manager12	Manager11	\$2y\$12\$1gJLzJ1ISFdimyp3OSpV4																																																																																		
<input type="checkbox"/>	Edit	Copy	Delete	19	4	customer2	Pelanggan	\$2y\$12\$vivqINZcV/e0J58atghkdusf																																																																																		
11	<p>Langkah berikutnya membuat <i>update</i> atau ubah data user dengan cara bikin file baru pada <i>view</i> dengan nama <code>user_ubah.blade.php</code> dan buat scriptnya menjadi seperti di bawah ini</p> <pre>jobsheet4 &gt; PWL_POS &gt; resources &gt; views &gt; user_ubah.blade.php &gt; html &gt; body &gt; form &gt; input 1  &lt;!DOCTYPE html&gt; 2  &lt;html&gt; 3    &lt;head&gt; 4      &lt;title&gt;Data Ubah User&lt;/title&gt; 5    &lt;/head&gt; 6    &lt;body&gt; 7      &lt;h1&gt;Form Ubah Data User&lt;/h1&gt; 8      &lt;a href="/user"&gt;Kembali&lt;/a&gt; 9      &lt;br&gt;&lt;br&gt; 10 11     &lt;form method="post" action="/PWL_2024/jobsheet4/PWL_POS/public/user/ubah_simpan/{{ \$data-&gt;user_id }}"&gt; 12 13       {{ csrf_field() }} 14       {{ method_field('PUT') }} 15 16       &lt;label&gt;Username&lt;/label&gt; 17       &lt;input type="text" name="username" placeholder="Masukan Username" value="{{ \$data-&gt;username }}"&gt; 18       &lt;br&gt; 19       &lt;label&gt;Nama&lt;/label&gt; 20       &lt;input type="text" name="nama" placeholder="Masukan Nama" value="{{ \$data-&gt;username }}"&gt; 21       &lt;br&gt; 22       &lt;label&gt;Password&lt;/label&gt; 23       &lt;input type="password" name="password" placeholder="Masukan Password" value="{{ \$data-&gt;password }}"&gt; 24       &lt;br&gt; 25       &lt;label&gt;Level ID&lt;/label&gt; 26       &lt;input type="number" name="level_id" placeholder="Masukan ID Level" value="{{ \$data-&gt;level_id }}"&gt; 27       &lt;br&gt;&lt;br&gt; 28       &lt;input type="submit" class="btn btn-success" value="Ubah"&gt; 29 30     &lt;/form&gt; 31 32 &lt;/body&gt; &lt;/html&gt;</pre>																																																																																									
12	<p>Tambahkan <i>script</i> pada <i>routes</i> dengan nama file <code>web.php</code>. Tambahkan seperti gambar dibawah ini</p>																																																																																									





```
jobsheet4 > PWL_POS > 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 () { Factory|View{
9      return view(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']);
15 Route::get('/user/tambah', [UserController::class, 'tambah']);
16
17 Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);
18
19 Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);
20
21
```

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

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

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


- Berikut adalah data user sebelum perubahan pada menu ‘Ubah’ yang menampilkan tabel data user

← → ↺ ⓘ localhost/PWL\_2024/jobsheet4/PWL\_POS/public/user ☆

## Data User

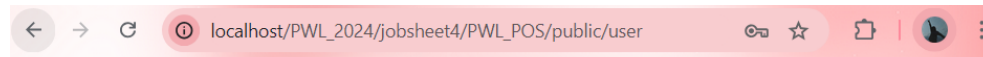
ID	Username	Nama	ID Level pengguna	Aksi
1	admin	Administrator	1	<a href="#">Ubah</a> <a href="#">Hapus</a>
2	manager	Manager	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
3	staff	Staff/Kasir	3	<a href="#">Ubah</a> <a href="#">Hapus</a>
14	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a> <a href="#">Hapus</a>
15	manager_dua	Manager 2	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
16	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
17	manager56	Manager55	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
18	manager12	Manager11	2	<a href="#">Ubah</a> <a href="#">Hapus</a>



	<p>- Tampilan form ubah menu setelah menu 'Ubah' diklik yang mana ketika mau merubah data user dapat dilakukan dengan mengklik button 'Ubah', sedangkan jika tidak jadi mengubah data user dapat mengklik 'Kembali'.</p> 
15	<p>Tambahkan <i>script</i> pada <i>routes</i> dengan nama file <code>web.php</code>. Tambahkan seperti gambar dibawah ini</p> <pre>Route::get('/user/ubah/{id}', [UserController::class, 'ubah']); Route::put('/user/ubah_simpan/{id}', [UserController::class, 'ubah_simpan']);</pre>
16	<p>Tambahkan <i>script</i> pada controller dengan nama file <code>UserController.php</code>. Tambahkan <i>script</i> dalam class dan buat method baru dengan nama <code>ubah_simpan</code> dan diletakan di bawah method <code>ubah</code> seperti gambar di bawah ini</p> <pre>public function ubah_simpan(\$id, Request \$request): Redirector RedirectResponse {     \$user = UserModel::find(\$id);      \$user-&gt;username = \$request-&gt;username;     \$user-&gt;nama = \$request-&gt;nama;     \$user-&gt;password = Hash::make('\$request-&gt;password');     \$user-&gt;level_id = \$request-&gt;level_id;      \$user-&gt;save();      return redirect(to: '/user'); }</pre>
17	<p>Simpan kode program Langkah 15 dan 16. Kemudian jalankan link <code>localhost:8000/user/ubah/1</code> atau <code>localhost/PWL_POS/public/user/ubah/1</code> pada <i>browser</i> dan ubah input formnya dan klik tombol ubah, kemudian amati apa yang terjadinya beri penjelasan dalam laporan</p>



Berdasarkan pemahaman saya dari hasil praktikum di bawah ini setelah menyimpan perubahan data nanti akan dikembalikan ke halaman tabel data user untuk mengetahui perubahan data user telah berhasil atau belum. Dan pada tabel user ini dapat diketahui data mana yang mendapat perubahan dengan melihat perubahan warna pada menu 'Ubah'.



## 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	staff	Staff/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
14	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
15	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
16	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
17	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
19	customer2	Pelanggan2	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>

18

Berikut untuk langkah *delete*. Tambahkan *script* pada *routes* dengan nama file *web.php*. Tambahkan seperti gambar di bawah ini

```
Route::get('/level', [LevelController::class, 'index']);
Route::get('/kategori', [KategoriController::class, 'index']);
Route::get('/user', [UserController::class, 'index']);

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

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

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

19

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): Redirector|RedirectResponse
{
    $user = UserModel::find($id);
    $user->delete();

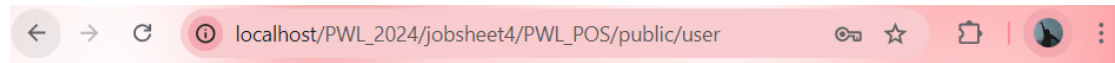
    return redirect(to: '/user');
}
```



20

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

- Berikut tabel data user sebelum penghapusan data user `customer2` di baris terakhir

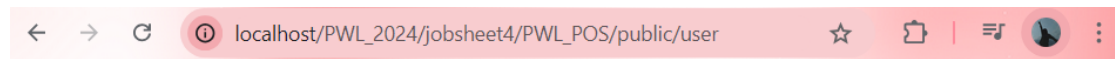


## Data User

[+ 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	staff	Staff/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
14	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
15	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
16	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
17	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
19	customer2	Pelanggan2	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>

- Berikut tabel data user setelah penghapusan data user `customer2` yang sudah menghilang dari tabel data user yang menunjukkan bahwa data `customer2` telah 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	staff	Staff/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
14	customer-1	Pelanggan Pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
15	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
16	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
17	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
18	manager12	Manager11	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>

21

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

## Praktikum Ke-2.7 – Relationship

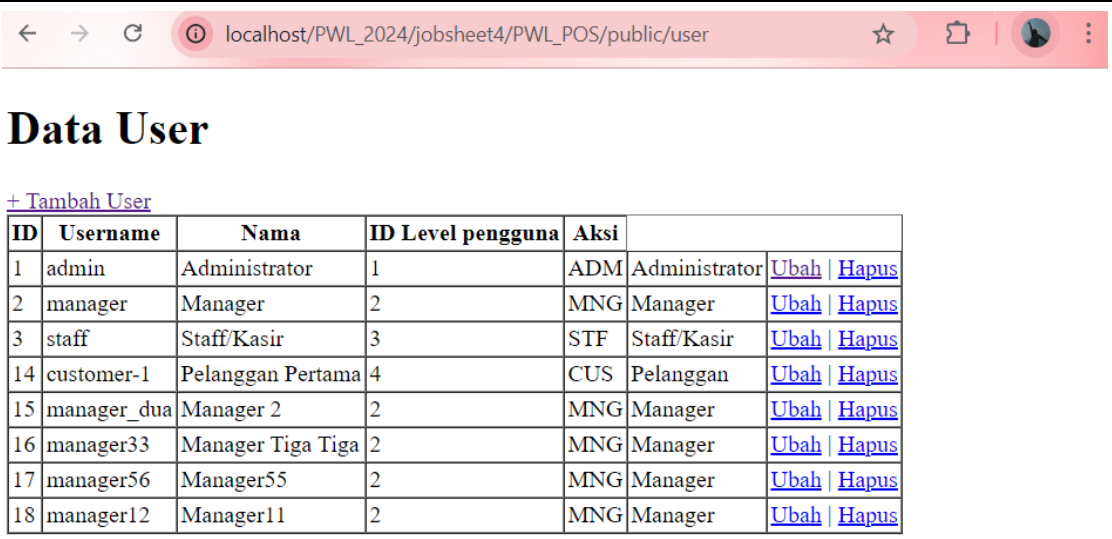


Langkah	Jawaban/Deskripsi
1	<p>Buka file model pada <code>UserModel.php</code> dan tambahkan scripnya menjadi seperti di bawahini</p> <pre>public function level(): BelongsTo {     return \$this-&gt;belongsTo(LevelModel::class, 'level_id', 'level_id'); }</pre>
2	<p>Buka file controller pada <code>UserController.php</code> dan ubah method <code>script</code> menjadi seperti di bawah ini</p> <pre>public function index(): never {     \$user = UserModel::with('level')-&gt;get();     dd(vars: \$user); }</pre>
3	<p>Simpan kode program Langkah 2. Kemudian jalankan link pada <i>browser</i>, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan</p> <pre>jobsheet4 &gt; PWL_POS &gt; app &gt; Models &gt; LevelModel.php &gt; ... 1  &lt;?php 2 3  namespace App\Models; 4 5  use Illuminate\Database\Eloquent\Factories\HasFactory; 6  use Illuminate\Database\Eloquent\Model; 7 8  1 reference   0 implementations 9  class LevelModel extends Model 10 { 11     use HasFactory; 12 13     8 references 14     protected \$table = 'm_level'; // Tabel yang digunakan 15     8 references 16     protected \$primaryKey = 'level_id'; // Primary key 17 18     // Sesuaikan fillable dengan kolom yang ada di tabel m_level 19     0 references 20     protected \$fillable = ['level_name']; // Contoh kolom yang ada di tabel m_level 21 }</pre> <p>localhost/PWL_2024/jobsheet4/PWL_POS/public/user</p> <pre>Illuminate\Database\Eloquent\Collection {#318 ▼ // app\Http\Controllers\UserController.php:63   #items: array:8 [▶]   #escapeWhenCastingToString: false }</pre>



	<p>localhost/PWL_2024/jobsheet4/PWL_POS/public/user</p> <pre>Illuminate\Database\Eloquent\Collection {#318 ▼ // app\Http\Controllers\UserController.php:63   #items: array:8 [▼     0 =&gt; App\Models\UserModel {#325 ▶}     1 =&gt; App\Models\UserModel {#326 ▶}     2 =&gt; App\Models\UserModel {#327 ▶}     3 =&gt; App\Models\UserModel {#328 ▶}     4 =&gt; App\Models\UserModel {#329 ▶}     5 =&gt; App\Models\UserModel {#330 ▶}     6 =&gt; App\Models\UserModel {#331 ▶}     7 =&gt; App\Models\UserModel {#332 ▶}   ]   #escapeWhenCastingToString: false }</pre> <p>Berdasarkan pemahaman saya dari hasil praktikum di atas output nya false dikarenakan kode inputan pada UserController berhasil diproses, tetapi tidak mengembalikan data apapun. Ini berarti bahwa tabel m_user kosong atau data di dalamnya tidak memiliki relasi yang sesuai dengan tabel m_level melalui kolom level_id. Oleh karena itu outputnya berupa collection kosong</p>
4	<p>Buka file controller pada <code>UserController.php</code> dan ubah method <code>script</code> menjadi seperti di bawah ini</p> <pre>public function index(): Factory View {     \$user = UserModel::with('level')-&gt;get();     return view('user', data: ['data' =&gt; \$user]); }</pre>
5	<p>Buka file view pada <code>user.blade.php</code> dan ubah <code>script</code> menjadi seperti di bawah ini</p> <pre>jobsheet4 &gt; PWL_POS &gt; resources &gt; views &gt; user.blade.php &gt; html &gt; body &gt; table &gt; tr &gt; id 1 &lt;!DOCTYPE html&gt; 2 &lt;html&gt; 3 &lt;head&gt; 4 &lt;title&gt;Data User&lt;/title&gt; 5 &lt;/head&gt; 6 &lt;body&gt; 7 &lt;h1&gt;Data User&lt;/h1&gt; 8 &lt;a href="/PWL_2024/jobsheet4/PWL_POS/public/user/tambah"&gt;+ Tambah User&lt;/a&gt; 9 &lt;table border="1" cellpadding="2" cellspacing="0"&gt; 10 &lt;tr&gt; 11 &lt;th&gt;ID&lt;/th&gt; 12 &lt;th&gt;Username&lt;/th&gt; 13 &lt;th&gt;Nama&lt;/th&gt; 14 &lt;th&gt;ID Level pengguna&lt;/th&gt; 15 &lt;th&gt;Aksi&lt;/th&gt; 16 &lt;/tr&gt; 17 @foreach (\$data as \$d) 18 &lt;tr&gt; 19 &lt;td&gt;{{ \$d-&gt;user_id }}&lt;/td&gt; 20 &lt;td&gt;{{ \$d-&gt;username }}&lt;/td&gt; 21 &lt;td&gt;{{ \$d-&gt;nama }}&lt;/td&gt; 22 &lt;td&gt;{{ \$d-&gt;level_id }}&lt;/td&gt; 23 &lt;td&gt;{{ \$d-&gt;level-&gt;level_kode }}&lt;/td&gt; 24 &lt;td&gt;{{ \$d-&gt;level-&gt;level_nama }}&lt;/td&gt; 25 &lt;td&gt;&lt;a href="/PWL_2024/jobsheet4/PWL_POS/public/user/ubah/{{ \$d-&gt;user_id }}"&gt;Ubah&lt;/a&gt;   &lt;a href="/PWL_2024/jobsheet4/PWL_POS/public/user/hapus/{{ \$d-&gt;user_id }}"&gt;Hapus&lt;/a&gt;&lt;/td&gt; 26 &lt;/tr&gt; 27 @endforeach 28 &lt;/table&gt; 29 &lt;/body&gt; 30 &lt;/html&gt;</pre>
6	<p>Simpan kode program Langkah 4 dan 5. Kemudian jalankan link pada <i>browser</i>, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan</p>



					
	<p>Berdasarkan pemahaman saya dari hasil praktikum di atas Menampilkan tabel data user yang kurang lebih sama dengan output pada praktikum sebelumnya, namun bedanya dari inputan di atas tabel data user lebih detail dengan aksi dan tambahan kolom keterangan level_kode serta level_nama per usernya.</p>				
7	Laporkan hasil Praktikum-2.7 ini dan <i>commit</i> perubahan pada <i>git</i> .				