

PEMROGRAMMAN WEB LANJUT
JOBSHEET 04 - MODEL DAN ELOQUENT ORM



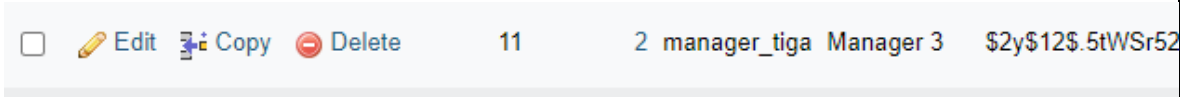
Nama : Pattricya Sonnya Fridayanti

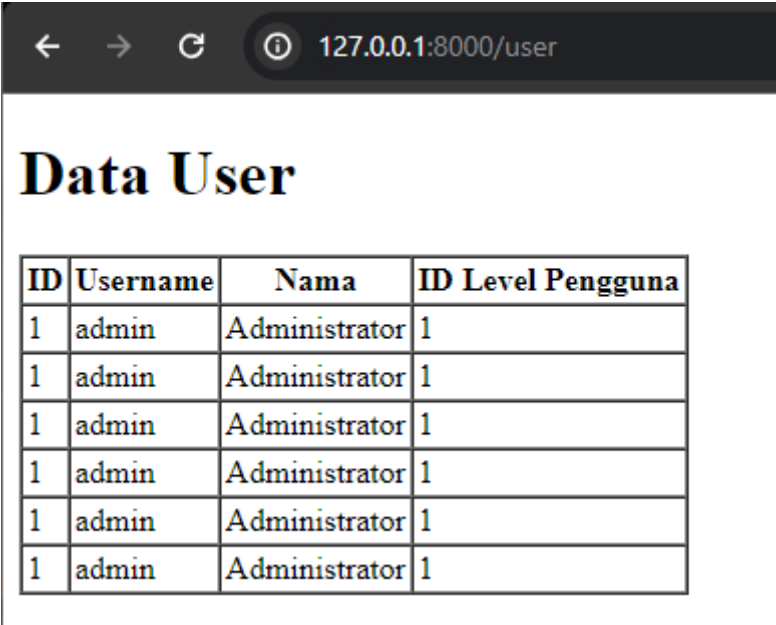
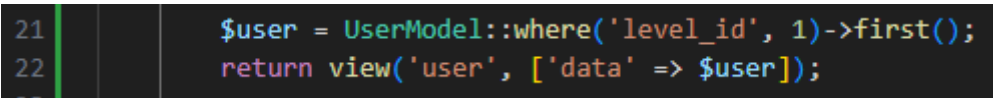
NIM : 2141762060

Kelas : SIB 3C

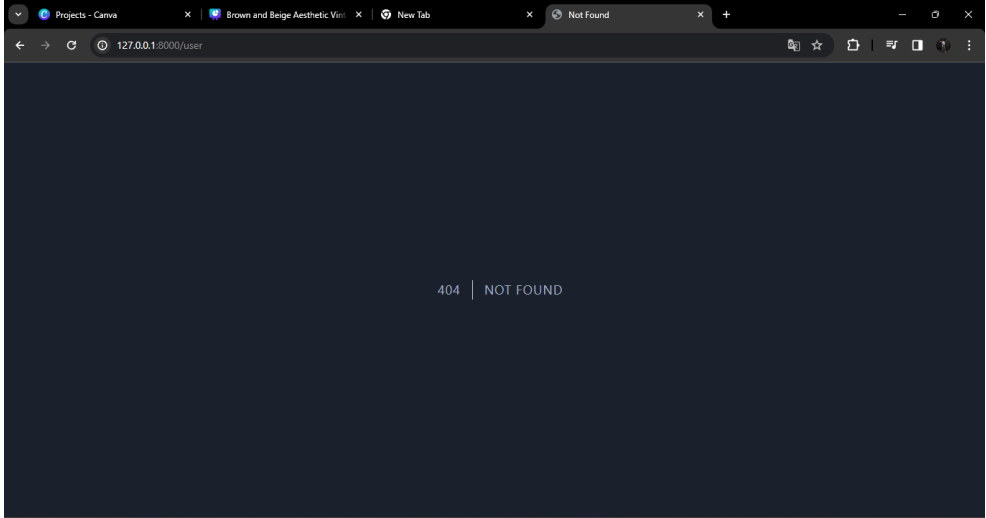
Politeknik Negeri Malnag
Teknologi Informasi
DIV Sistem Informasi Bisnis

No	Keterangan
Praktikum 1 - \$fillable	
1.	<p>Buka file model dengan nama UserModel.php dan tambahkan \$fillable seperti gambar di bawah ini</p> <pre> 10 use HasFactory; 11 12 protected \$table = 'm_user'; 13 protected \$primaryKey = 'user_id'; 14 15 @var array 16 protected \$fillable = ['level_id', 'username', 'nama', 'password']; 17 </pre>
2.	<p>Buka file controller dengan nama UserController.php dan ubah script untuk menambahkan data baru seperti gambar di bawah ini</p> <pre> 9 class UserController extends Controller 10 { 11 public function index(){ 12 13 \$data = [14 'level_id' => 2, 15 'username' => 'manager_dua', 16 'nama' => 'Manager 2', 17 'password' => Hash::make('12345') 18]; 19 UserModel::create(\$data); 20 21 \$user = UserModel::all(); 22 return view('user', ['data' => \$user]); 23 } 24 } </pre>
3.	<p>Simpan kode program Langkah 1 dan 2, dan jalankan perintah web server. Kemudian jalankan link localhostPWL_POS/public/user pada browser dan amati apa yang terjadi</p> 

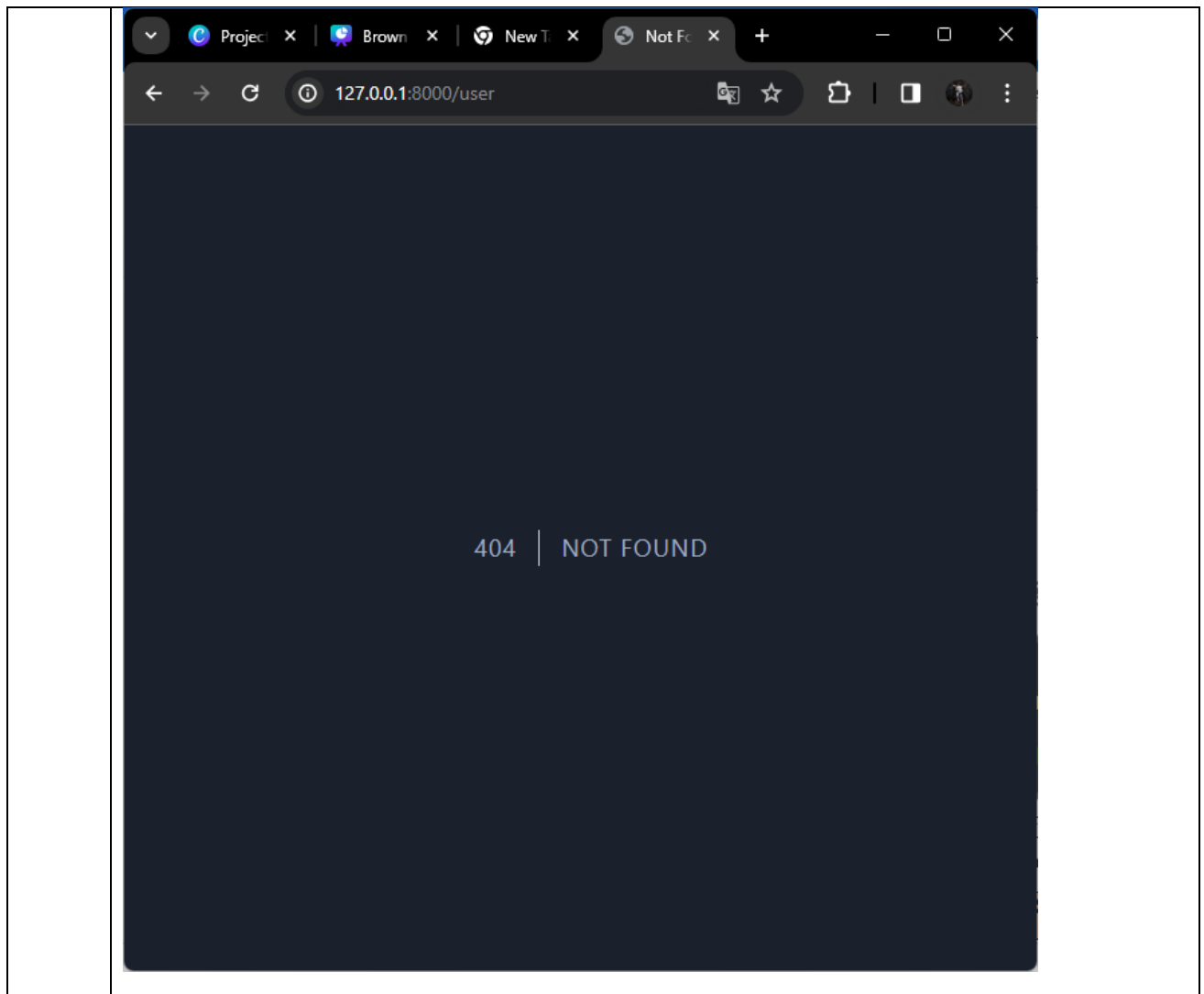
	- Error karena ada @var array
4.	<p>Ubah file model UserModel.php seperti pada gambar di bawah ini pada bagian \$fillable</p> <pre> 8 class UserModel extends Model 9 { 10 use HasFactory; 11 12 protected \$table = 'm_user'; 13 protected \$primaryKey = 'user_id'; 14 15 protected \$fillable = ['level_id', 'username', 'nama']; 16 }</pre>
5.	<p>Ubah kembali file controller UserController.php seperti pada gambar di bawah hanya bagian array pada \$data</p> <pre> 11 public function index(){ 12 13 \$data = [14 'level_id' => 2, 15 'username' => 'manager_tiga', 16 'nama' => 'Manager 3', 17 'password' => Hash::make('12345') 18]; 19 UserModel::create(\$data); 20 21 \$user = UserModel::all(); 22 return view('user', ['data' => \$user]); </pre>
6.	<p>Simpan kode program Langkah 4 dan 5. Kemudian jalankan pada browser dan amati apa yang terjadi</p> 
Praktikum 2.1 – Retrieving Single Models	
1.	<p>Buka file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> <pre> 21 \$user = UserModel::find(1); 22 return view('user', ['data' => \$user]); </pre>
2.	<p>Buka file view dengan nama user.blade.php dan ubah script seperti gambar di bawah ini</p>

	<pre>18 @foreach (\$data as \$d) 19 <tr> 20 <td>{{ \$data->user_id }}</td> 21 <td>{{ \$data->username }}</td> 22 <td>{{ \$data->nama }}</td> 23 <td>{{ \$data->level_id }}</td> 24 </tr> 25 @endforeach</pre>	
3.	<p>Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> <div></div>	
4.	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> <div></div>	
5.	<p>Simpan kode program Langkah 4. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p>	

</

	<pre> 21 \$user = UserModel::findOr(1, ['username', 'nama'], function(){ 22 abort(404); 23 }); 24 return view('user', ['data' => \$user]); </pre>
9.	Simpan kode program Langkah 8. Kemudian pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan
10.	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> <pre> 21 \$user = UserModel::findOr(20, ['username', 'nama'], function(){ 22 abort(404); 23 }); 24 return view('user', ['data' => \$user]); </pre>
11.	<p>Simpan kode program Langkah 10. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> 
Praktikum 2.2 – Not Found Exceptions	
1.	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> <pre> 11 public function index(){ 12 13 \$user = UserModel::findOrFail(1); 14 return view('user', ['data' => \$user]); 15 } </pre>
2.	Simpan kode program Langkah 1. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan

</

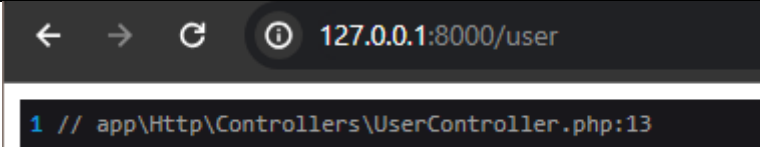
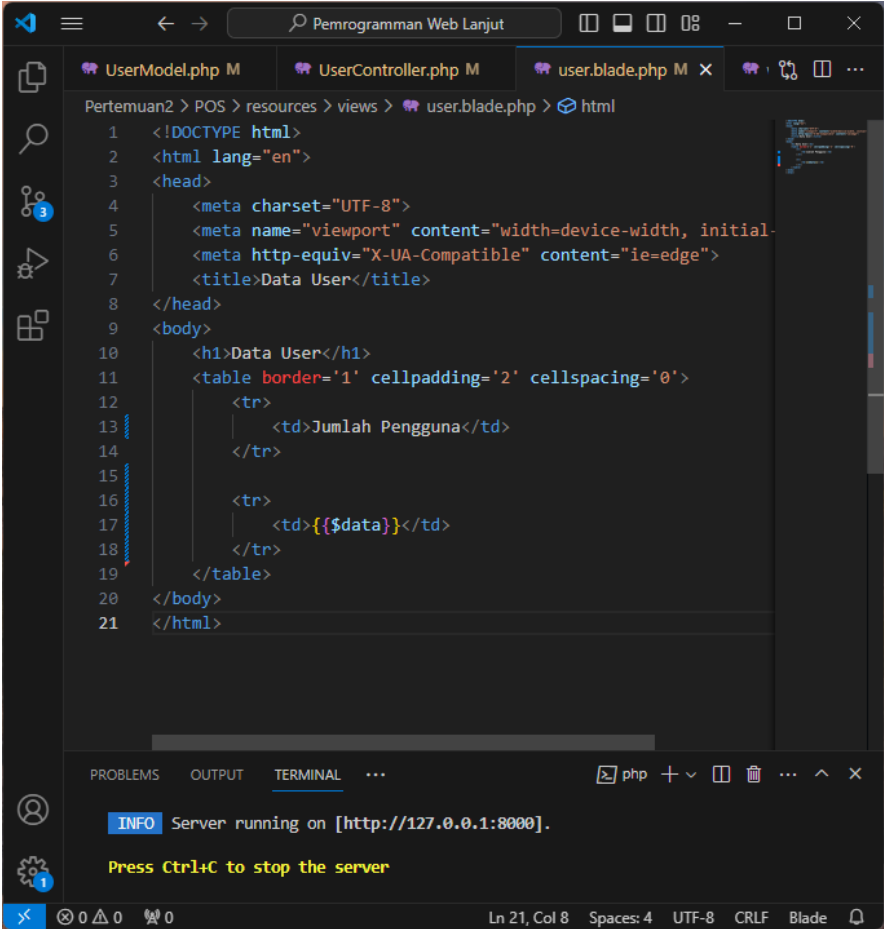
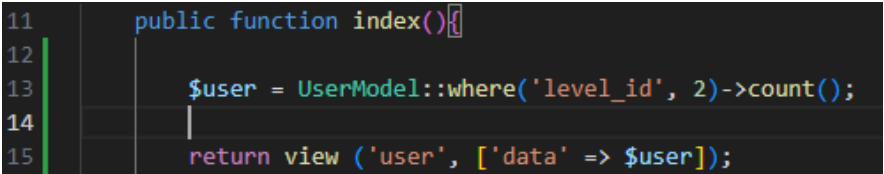


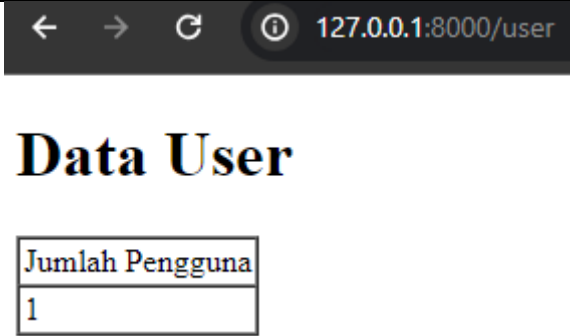
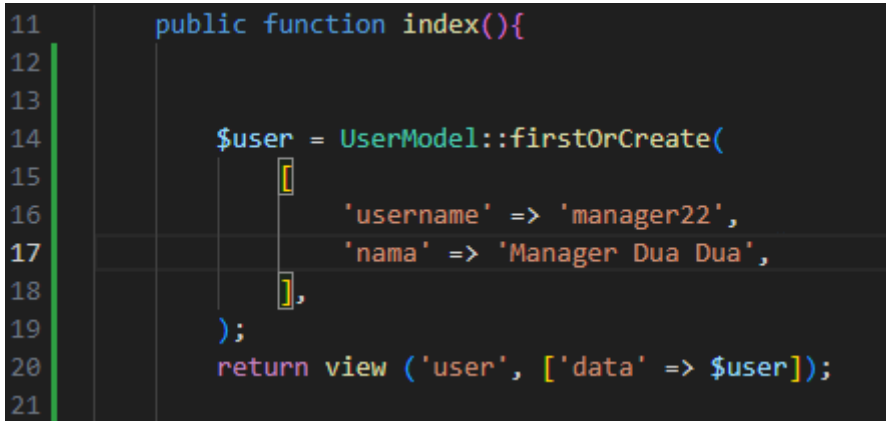
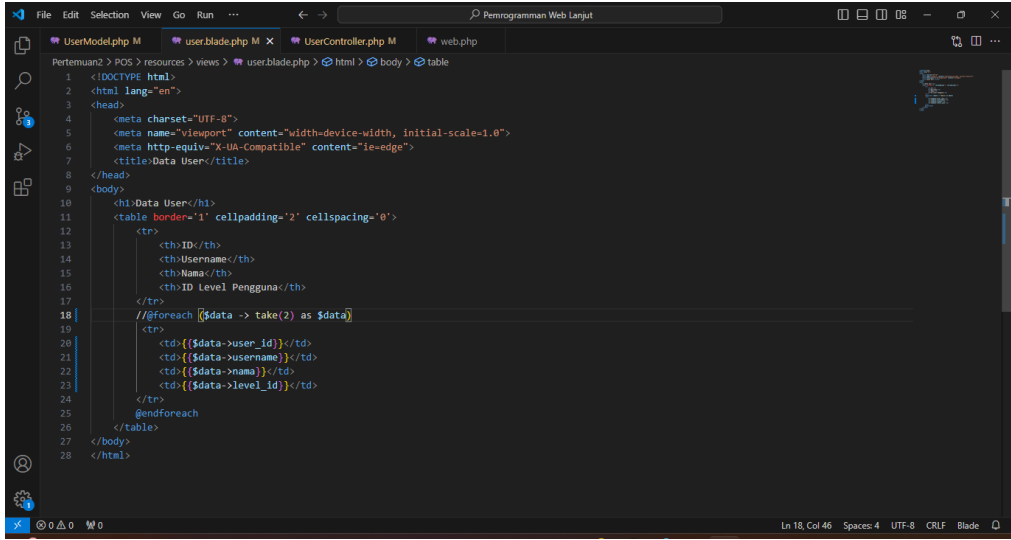
Praktikum 2.3 – Retrieving Aggregates

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

```
9 class UserController extends Controller
10 {
11     public function index(){
12         $user = UserModel::where('level_id', 2)->count();
13         dd($user);
14         return view ('user', ['data' => $user]);
```

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

	
3.	<p>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</p> <ul style="list-style-type: none"> Ubah user.blade.php  <ul style="list-style-type: none"> Isi dari UserController.php  <ul style="list-style-type: none"> Hasil

	
Praktikum 2.4 – Retrieving or Creating Models	
1.	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p>  <pre> 11 public function index(){ 12 13 14 \$user = UserModel::firstOrCreate(15 [16 'username' => 'manager22', 17 'nama' => 'Manager Dua Dua', 18], 19); 20 return view ('user', ['data' => \$user]); 21 </pre>
2.	<p>Ubah kembali file view dengan nama user.blade.php dan ubah script seperti gambar di bawah ini</p>  <pre> 1 <doctype html> 2 <html lang="en"> 3 <head> 4 <meta charset="UTF-8"> 5 <meta name="viewport" content="width=device-width, initial-scale=1.0"> 6 <meta http-equiv="X-UA-Compatible" content="ie=edge"> 7 <title>Data User</title> 8 </head> 9 <body> 10 <h1>Data User</h1> 11 <table border="1" cellpadding="2" cellspacing="0"> 12 <tr> 13 <th>ID</th> 14 <th>Username</th> 15 <th>Nama</th> 16 <th>ID Level Pengguna</th> 17 </tr> 18 @foreach (\$data -> take(2) as \$data) 19 <tr> 20 <td>{{ \$data->user_id }}</td> 21 <td>{{ \$data->username }}</td> 22 <td>{{ \$data->nama }}</td> 23 <td>{{ \$data->level_id }}</td> 24 </tr> 25 @endforeach 26 </table> 27 </body> 28 </html> </pre>
3.	<p>Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p>

← → ↻ ⓘ 127.0.0.1:8000/user

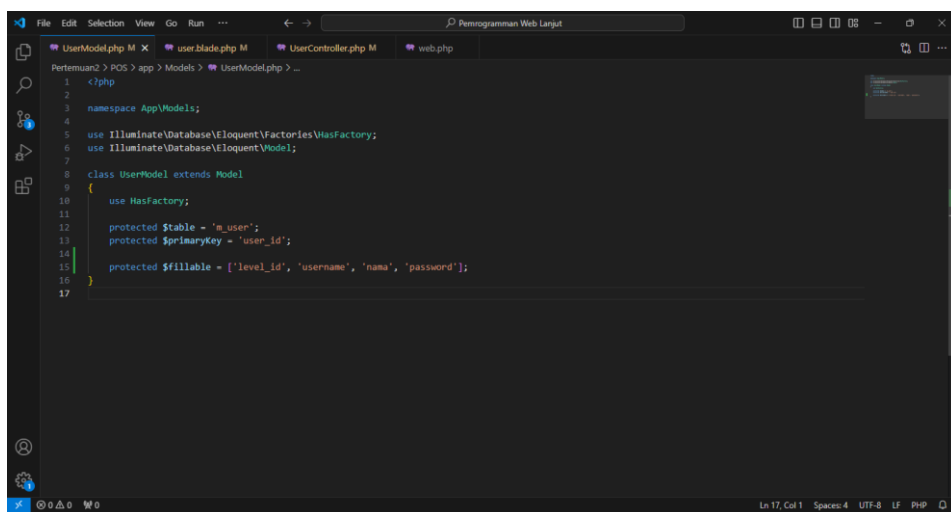
Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

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

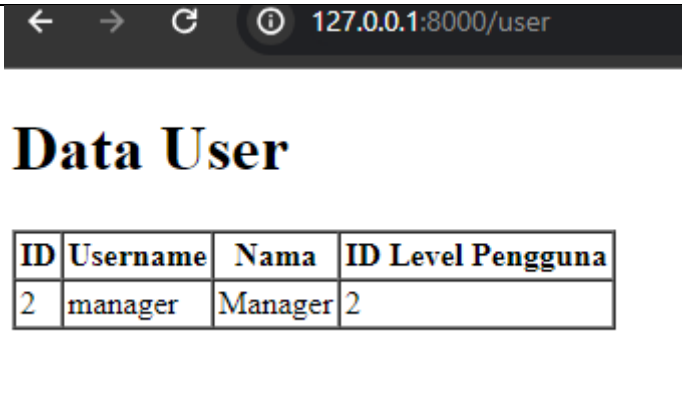
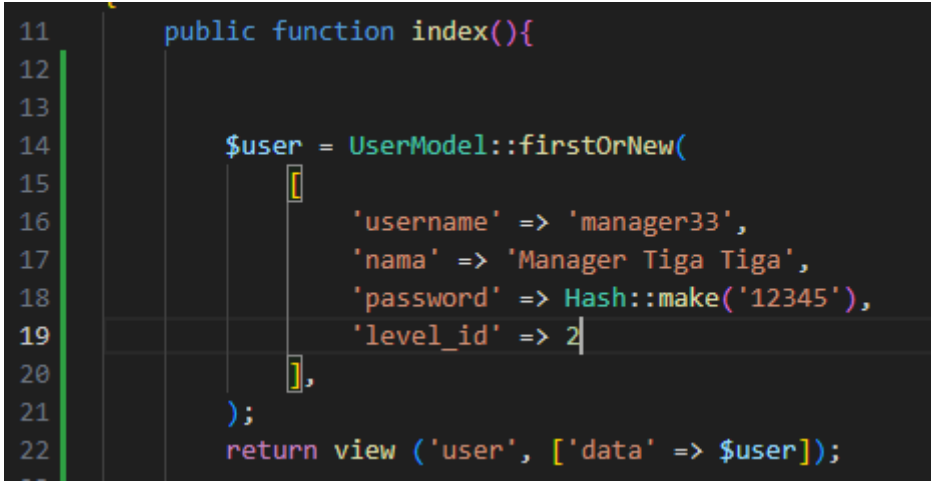
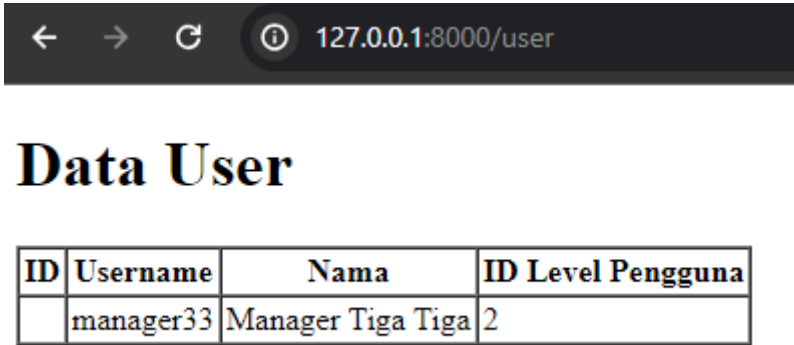
```
11 public function index(){
12
13
14     $user = UserModel::firstOrCreate([
15         [
16             'username' => 'manager22',
17             'nama' => 'Manager Dua Dua',
18             'password' => Hash::make('12345'),
19             'level_id' => 2
20         ],
21     ]);
22     return view('user', ['data' => $user]);
```

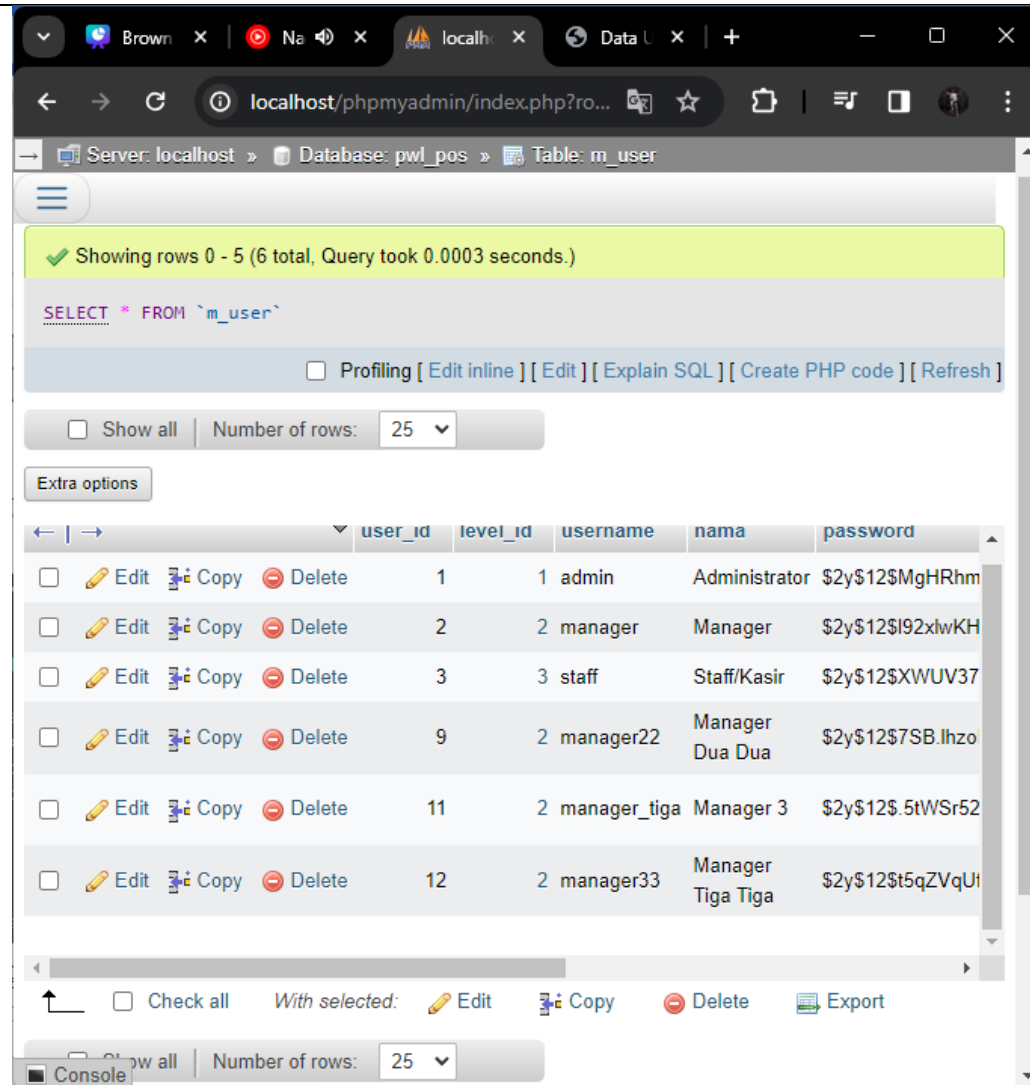
- Ubah UserModel



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class UserModel extends Model
9 {
10     use HasFactory;
11
12     protected $table = 'm_user';
13     protected $primaryKey = 'user_id';
14
15     protected $fillable = ['level_id', 'username', 'nama', 'password'];
16
17 }
```

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

	
8.	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> 
9.	<p>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</p> 



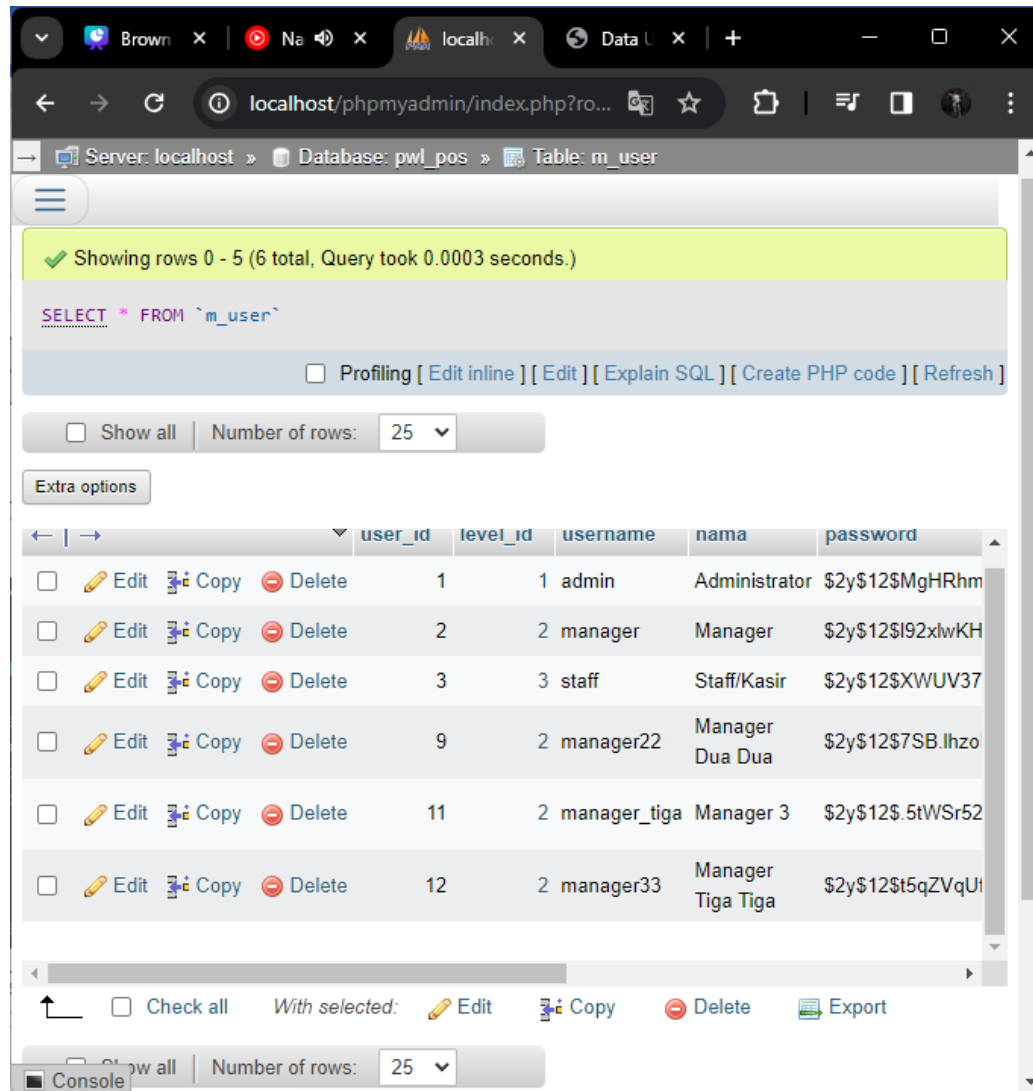
10. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```

11 public function index(){
12
13
14     $user = UserModel::firstOrCreate(
15         [
16             'username' => 'manager33',
17             'nama' => 'Manager Tiga Tiga',
18             'password' => Hash::make('12345'),
19             'level_id' => 2
20         ],
21     );
22     $user->save();
23
24     return view('user', ['data' => $user]);

```

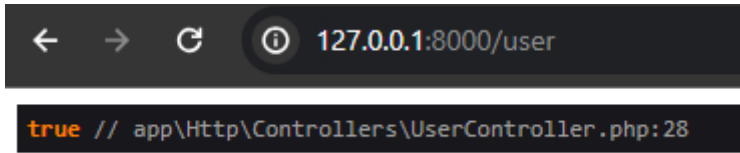
11. Simpan kode program Langkah 9. Kemudian jalankan pada browser dan amati apa yang terjadi dan cek juga pada phpMyAdmin pada tabel m_user serta beri penjelasan dalam laporan



Praktikum 2.5 – Attribute Changes

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

	<pre> 10 class UserController extends Controller 12 public function index(){ 13 \$user = UserModel::create(14 [15 'username' => 'manager55', 16 'nama' => 'Manager55', 17 'password' => Hash::make('12345'), 18 'level_id' => 2, 19] 20); 21 \$user -> username = 'manager56'; 22 23 \$user->isDirty(); //true 24 \$user->isDirty('username'); //true 25 \$user->isDirty('nama'); //false 26 \$user->isDirty(['nama', 'username']); //true 27 28 \$user -> isClean(); //false 29 \$user -> isClean('username'); //false 30 \$user -> isClean('nama'); //true 31 \$user -> isClean(['nama', 'username']); //false 32 33 \$user -> save(); 34 35 \$user->isDirty(); //false 36 \$user -> isClean(); //true 37 dd(\$user->isDirty()); </pre>
2.	<p>Simpan kode program Langkah 1. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> 
3.	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p>

	<pre> 10 class UserController extends Controller 11 { 12 public function index(){ 13 \$user = UserModel::create([14 'username' => 'manager11', 15 'nama' => 'Manager11', 16 'password' => Hash::make('12345'), 17 'level_id' => 2, 18]); 19 20 \$user -> username = 'manager12'; 21 22 \$user -> save(); 23 24 \$user->wasChanged(); // true 25 \$user->wasChanged('username'); // true 26 \$user->wasChanged(['username', 'level_id']); // true 27 \$user->wasChanged('nama'); // false 28 \$user->wasChanged(['nama', 'username']); // true 29 } </pre>
4.	<p>Simpan kode program Langkah 3. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> 
Praktikum 2.6 – Create, Read, Update, Delete (CRUD)	
1.	Buka file view pada user.blade.php dan buat scripnya menjadi seperti di bawah ini

	<pre> 9 <body> 10 <h1>Data User</h1> 11 + Tambah User 12 <table border='1' cellpadding='2' cellspacing='0'> 13 <tr> 14 <td>ID</td> 15 <td>Username</td> 16 <td>Nama</td> 17 <td>ID Level Pengguna</td> 18 <td>Aksi</td> 19 </tr> 20 @foreach (\$data as \$d) 21 <tr> 22 <td>{{ \$d->user_id }}</td> 23 <td>{{ \$d->username }}</td> 24 <td>{{ \$d->nama }}</td> 25 <td>{{ \$d->level_id }}</td> 26 <td>user_id }}">Ubah 27 user_id }}">Hapus</td> 28 </tr> 29 @endforeach 30 </table> 31 </body> </pre>
2.	<p>Buka file controller pada UserController.php dan buat scriptnya untuk read menjadi seperti di bawah ini</p> <pre> 10 class UserController extends Controller 11 { 12 public function index(){ 13 \$user = UserModel::all(); 14 return view('user', ['data' => \$user]); 15 } </pre>
3.	<p>Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p>

Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
9	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager_tiga	Manager 3	2	Ubah Hapus
12	manager33	Manager Tiga Tiga	2	Ubah Hapus
14	manager56	Manager55	2	Ubah Hapus
19	manager12	Manager11	2	Ubah Hapus

4. 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

```

10      <h1>Form Tambah Data User</h1>
11      <form method="post" action="/user/tambah_simpan">
12
13          {{csrf_field()}}
14
15          <label>Username</label>
16          <input type="text" name="username" placeholder="Masukkan Username">
17          <br>
18          <label>Nama</label>
19          <input type="text" name="nama" placeholder="Masukkan Nama">
20          <br>
21          <label>Password</label>
22          <input type="password" name="password" placeholder="Masukkan Password">
23          <br>
24          <label>Level ID</label>
25          <input type="number" name="level_id" placeholder="Masukkan ID Level">
26          <br><br>
27          <input type="submit" class="btn btn-success" value="Simpan">
28      </form>
29  </body>

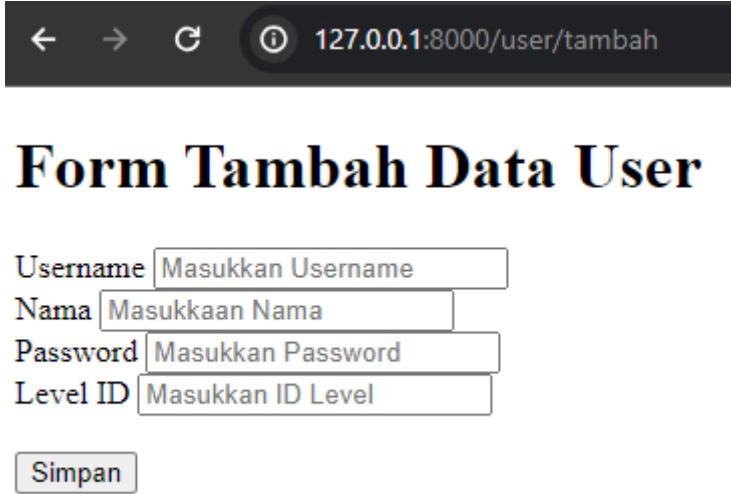
```

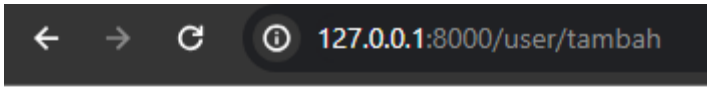
5. Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

```

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

```

6.	<p>Tambahkan script pada controller dengan nama file UserController.php. Tambahkan script dalam class dan buat method baru dengan nama tambah dan diletakan di bawah method index seperti gambar di bawah ini</p> <pre> 10 class UserController extends Controller 11 { 12 public function index(){ 13 \$user = UserModel::all(); 14 return view('user', ['data' => \$user]); 15 } 16 17 public function tambah(){ 18 return view('user_tambah'); 19 } </pre>
7.	<p>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</p>  <p>The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/user/tambah'. Below the browser, there is a form titled 'Form Tambah Data User'. The form contains four input fields: 'Username' with placeholder text 'Masukkan Username', 'Nama' with placeholder text 'Masukkan Nama', 'Password' with placeholder text 'Masukkan Password', and 'Level ID' with placeholder text 'Masukkan ID Level'. At the bottom of the form is a button labeled 'Simpan'.</p>
8.	<p>Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini</p> <pre> 53 Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']); </pre>
9.	<p>Tambahkan script pada controller dengan nama file UserController.php. Tambahkan script dalam class dan buat method baru dengan nama tambah_simpan</p>

	<pre>21 public function tambah_simpan(Request \$request){ 22 UserModel::create([23 'username' => \$request->username, 24 'nama' => \$request->nama, 25 'password' => Hash::make('\$request->password'), 26 'level_id' => \$request->level_id 27]); 28 29 return redirect('/user'); 30 }</pre>
10.	<p>Simpan kode program Langkah 8 dan 9. Kemudian jalankan link localhost:8000/user/tambah atau localhost/PWL_POS/public/user/tambah pada browser dan input formnya dan simpan, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan</p> <div data-bbox="317 801 1027 887"></div> <h2 data-bbox="331 936 1023 994">Form Tambah Data User</h2> <div data-bbox="331 1043 829 1292"><p>Username <input type="text" value="manager_tambah"/></p><p>Nama <input type="text" value="Manager Tambah"/></p><p>Password <input type="password" value="....."/></p><p>Level ID <input type="text" value="2"/></p><p><input type="button" value="Simpan"/></p></div> <p data-bbox="320 1364 574 1402">Setelah klik simpan</p>

← → ↺ 127.0.0.1:8000/user 🔑 🗨️ ☆ 🏠

Data User

[+ Tambah User](#)

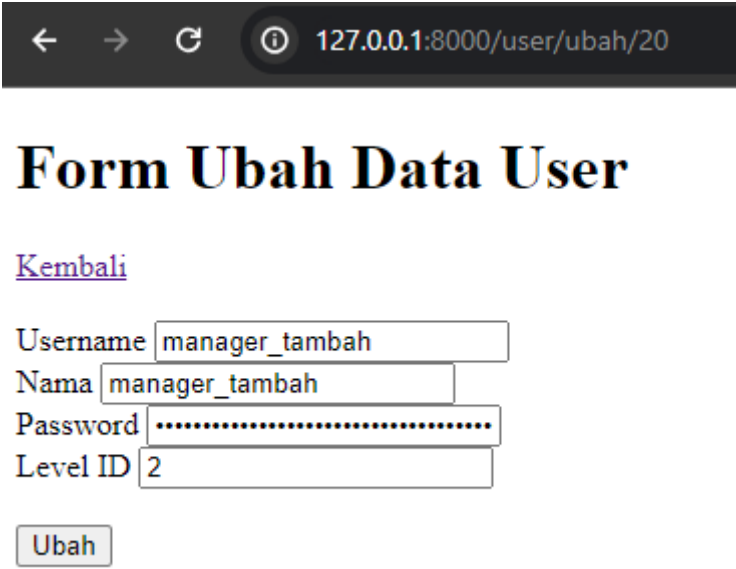
ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
9	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager_tiga	Manager 3	2	Ubah Hapus
12	manager33	Manager Tiga Tiga	2	Ubah Hapus
14	manager56	Manager55	2	Ubah Hapus
19	manager12	Manager11	2	Ubah Hapus
20	manager_tambah	Manager Tambah	2	Ubah Hapus

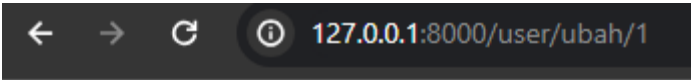
11. Langkah berikutnya membuat update atau ubah data user dengan cara bikin file baru pada view dengan nama user_ubah.blade.php dan buat scriptnya menjadi seperti di bawah ini

```
 9 <body>
10 <h1>Form Ubah Data User</h1>
11 <a href="/user">Kembali</a>
12 <br><br>
13
14 <form method='post' action="/user/ubah_simpan/{{ $data->user_id }}">
15     {{ csrf_field() }}
16     {{ method_field('PUT') }}
17
18     <label>Username</label>
19     <input type="text" name="username" placeholder="Masukkan Username" value="{{ $data->username }}">
20     <br>
21     <label>Nama</label>
22     <input type="text" name="nama" placeholder="Masukkan Nama" value="{{ $data->username }}">
23     <br>
24     <label>Password</label>
25     <input type="password" name="password" placeholder="Masukkan Password" value="{{ $data->password }}">
26     <br>
27     <label>Level ID</label>
28     <input type="number" name="level_id" placeholder="Masukkan ID Level" value="{{ $data->level_id }}">
29     <br><br>
30     <input type="submit" class="btn btn-success" value="Ubah">
31 </form>
32 </body>
```

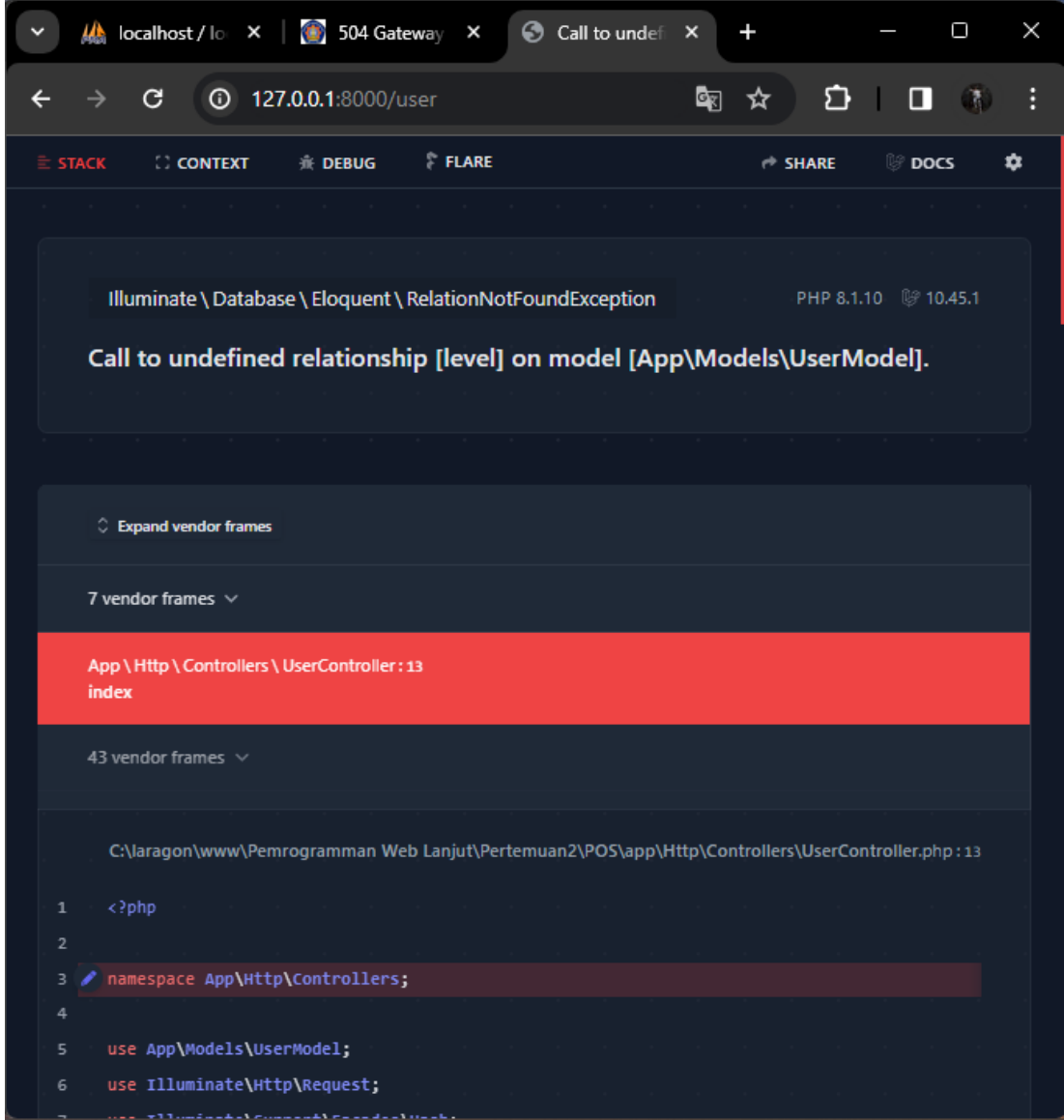
12. Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

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

13.	<p>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</p> <pre data-bbox="319 347 1273 515"> 32 public function ubah(\$id){ 33 \$user=UserModel::find(\$id); 34 return view('user_ubah', ['data' => \$user]); 35 } </pre>
14.	<p>Simpan kode program Langkah 11 sd 13. Kemudian jalankan pada browser dan klik link “Ubah” amati apa yang terjadi dan beri penjelasan dalam laporan</p>  <p>The screenshot shows a web browser address bar with the URL <code>127.0.0.1:8000/user/ubah/20</code>. Below the browser, there is a form titled Form Ubah Data User. The form includes a link Kembali and several input fields: Username (manager_tambah), Nama (manager_tambah), Password (masked with dots), and Level ID (2). At the bottom of the form is a button labeled Ubah.</p>
15.	<p>Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini</p> <pre data-bbox="319 1429 1487 1467"> 57 Route::put('/user/ubah_simpan/{id}', [UserController::class, 'ubah_simpan']); </pre>
16.	<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</p>

	<pre> 37 public function ubah_simpan(\$id, Request \$request){ 38 \$user = UserModel::find(\$id); 39 40 \$user->username = \$request->username; 41 \$user->nama = \$request->nama; 42 \$user->password = Hash::make('\$request -> password'); 43 \$user->level_id = \$request->level_id; 44 45 \$user->save(); 46 47 return redirect('/user'); 48 } </pre>
17.	<p>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</p> <p>Mengubah nama dari admin mnejadi admin ubah</p>  <h2 style="text-align: center;">Form Ubah Data User</h2> <p>Kembali</p> <p>Username <input type="text" value="admin"/></p> <p>Nama <input type="text" value="admin ubah"/></p> <p>Password <input type="password" value="....."/></p> <p>Level ID <input type="text" value="1"/></p> <p><input type="button" value="Ubah"/></p> <p>Setelah klik Ubah</p>

	 <pre> 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Factories\HasFactory; 6 use Illuminate\Database\Eloquent\Model; 7 8 class LevelModel extends Model 9 { 10 use HasFactory; 11 12 protected \$table = 'm_level'; 13 protected \$primaryKey = 'level_id'; 14 15 } </pre> <ul style="list-style-type: none"> Menambahkan script <pre> 13 protected \$table = 'm_user'; 14 protected \$primaryKey = 'user_id'; 15 16 protected \$fillable = ['level_id', 'username', 'nama', 'password']; 17 18 public function level():BelongsTo{ 19 return \$this -> belongsTo(LevelModel::class, 'level_id', 'level_id'); 20 } </pre>
2.	<p>Buka file controller pada UserController.php dan ubah method script menjadi seperti di bawah ini</p> <pre> 12 public function index(){ 13 \$user = UserModel::with('level')->get(); 14 dd(\$user); 15 } </pre>
3.	<p>Simpan kode program Langkah 2. Kemudian jalankan link pada browser, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan</p>

	
4.	<p>Buka file controller pada UserController.php dan ubah method script menjadi seperti di bawah ini</p> <pre data-bbox="319 1494 1192 1644"> 12 public function index(){ 13 \$user = UserModel::with('level')->get(); 14 return view('user', ['data' => \$user]); 15 }</pre>
5.	<p>Buka file view pada user.blade.php dan ubah script menjadi seperti di bawah ini</p>

```

9      <body>
10         <h1>Data User</h1>
11         <a href="/user/tambah"> + Tambah User</a>
12         <table border='1' cellpadding='2' cellspacing='0'>
13             <tr>
14                 <td>ID</td>
15                 <td>Username</td>
16                 <td>Nama</td>
17                 <td>ID Level Pengguna</td>
18                 <td>Kode Level</td>
19                 <td>Nama Level</td>
20                 <td>Aksi</td>
21             </tr>
22             @foreach ($data as $d)
23                 <tr>
24                     <td>{{ $d->user_id }}</td>
25                     <td>{{ $d->username }}</td>
26                     <td>{{ $d->nama }}</td>
27                     <td>{{ $d->level_id }}</td>
28                     <td>{{ $d->level->level_kode }}</td>
29                     <td>{{ $d->level->level_nama }}</td>
30                     <td><a href="/user/ubah/{{ $d->user_id }}">Ubah</a>
31                     <a href="/user/hapus/{{ $d->user_id }}">Hapus</a></td>
32                 </tr>
33             @endforeach
34         </table>
35     </body>

```

6. 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	admin ubah	1	ADM	Administrator	Ubah Hapus
2	manager	Manager	2	MNG	Manager	Ubah Hapus
3	staff	Staff/Kasir	3	STF	Staff/Kasir	Ubah Hapus
9	manager22	Manager Dua Dua	2	MNG	Manager	Ubah Hapus
11	manager_tiga	Manager 3	2	MNG	Manager	Ubah Hapus
12	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah Hapus
14	manager56	Manager55	2	MNG	Manager	Ubah Hapus
19	manager12	Manager11	2	MNG	Manager	Ubah Hapus

