

NIM : 2241720095
Nama : Ana Bellatus Mustaqfiro
Kelas : D4 TI - 2H
No. Urut : 04
Mata Kuliah : Pemrograman Web Lanjut Jobsheet 4 (Eloquent ORM)

Praktikum 1 - \$fillable

A. \$fillable

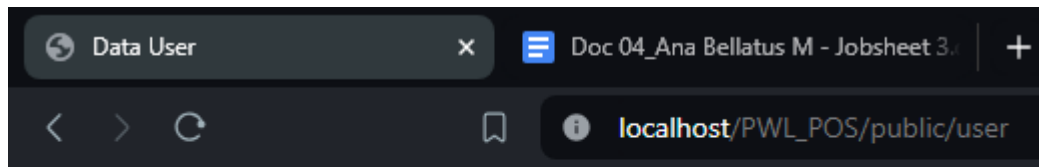
1. Buka file model dengan nama UserModel.php dan tambahkan \$fillable seperti gambar di bawah ini

```
UserModel.php M X
app > Models > UserModel.php > UserModel > $fillable
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 }
```

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

```
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 }
```

3. 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



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
6	customer-1	Pelanggan Pertama	4
9	manager_dua	Manager 2	2

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

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

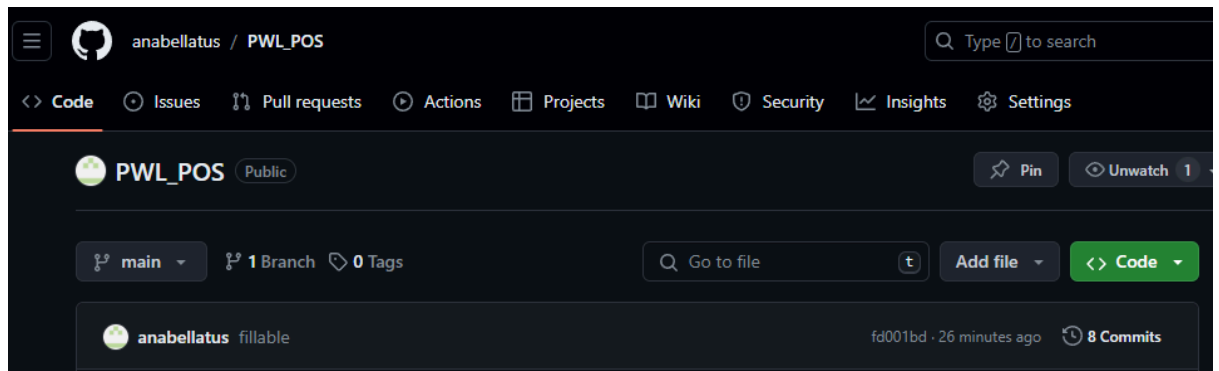
5. Ubah kembali file controller UserController.php seperti pada gambar di bawah hanya bagian array pada \$data

```
9 class UserController extends Controller
10 {
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     }
20 }
```

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



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



B. \$guarded

Kebalikan dari \$fillable adalah \$guarded. Semua kolom yang kita tambahkan ke \$guarded akan diabaikan oleh Eloquent ketika kita melakukan insert/update. Secara default \$guarded isinya array(""), yang berarti semua atribut tidak bisa diset melalui mass assignment (jabarkan istilah ini).

Praktikum 2.1 Retrieving Single Models

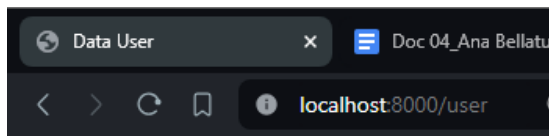
1. Buka file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         $user = UserModel::find(1);
14         return view('user', ['data' => $user]);
15     }
16 }
```

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

```
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>
```

3. Simpan kode program Langkah 1 dan 2. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



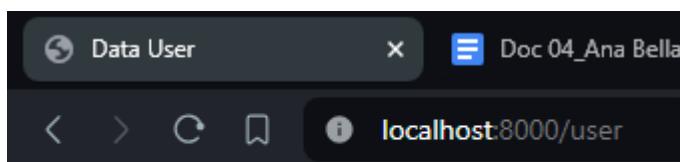
Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

4. 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     {
13         $user = UserModel::where('level_id', 1)->first();
14         return view('user', ['data' => $user]);
15     }
}
```

5. Simpan kode program Langkah 4. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



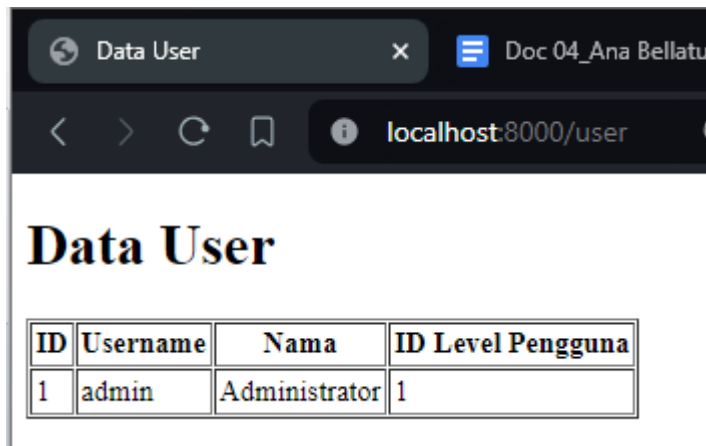
Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

6. 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     {
13         $user = UserModel::firstWhere('level_id', 1);
14         return view('user', ['data' => $user]);
15     }
}
```

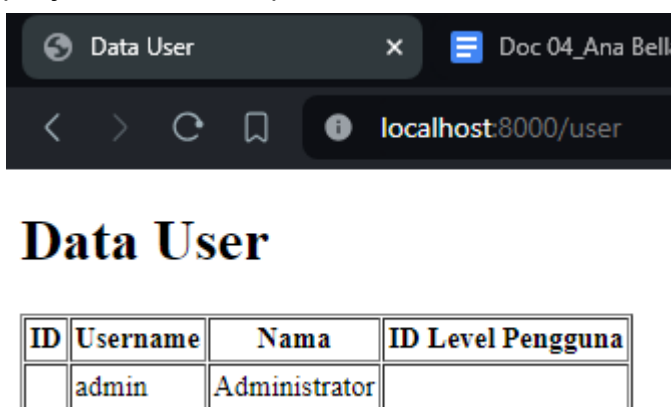
7. Simpan kode program Langkah 6. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



8. 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     {
13         $user = UserModel::findor(1, ['username', 'nama'], function(){
14             abort(404);
15         });
16         return view('user', ['data' => $user]);
17     }
18 }
```

9. Simpan kode program Langkah 8. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



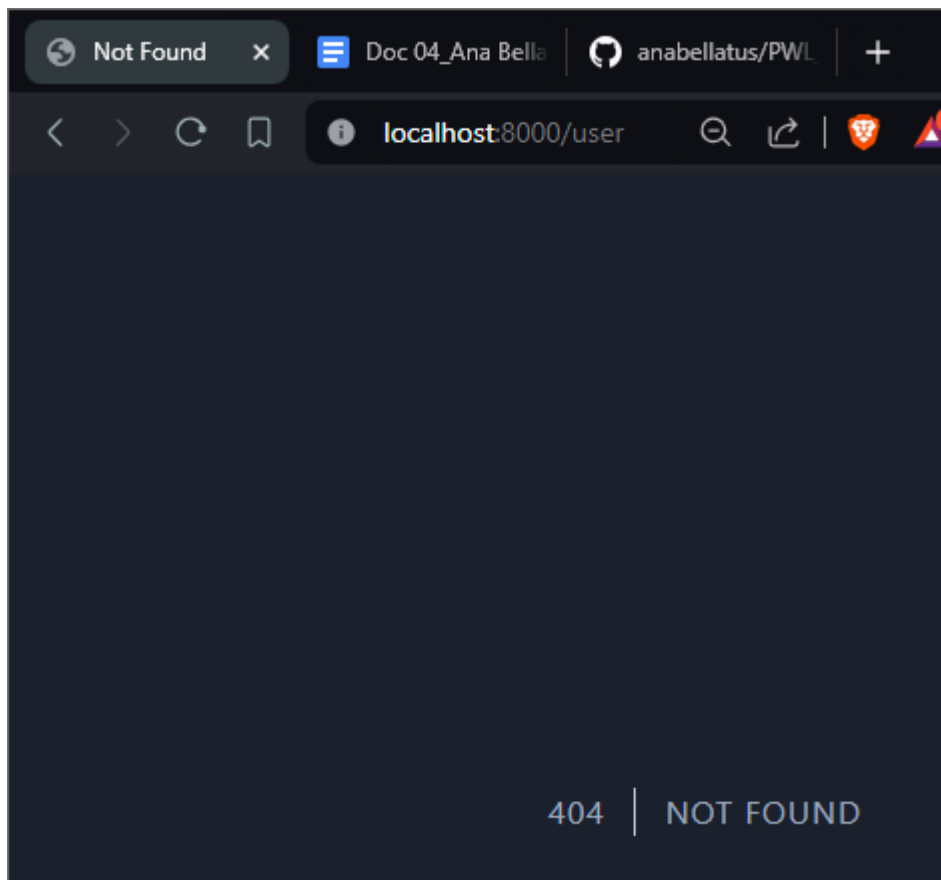
10. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```

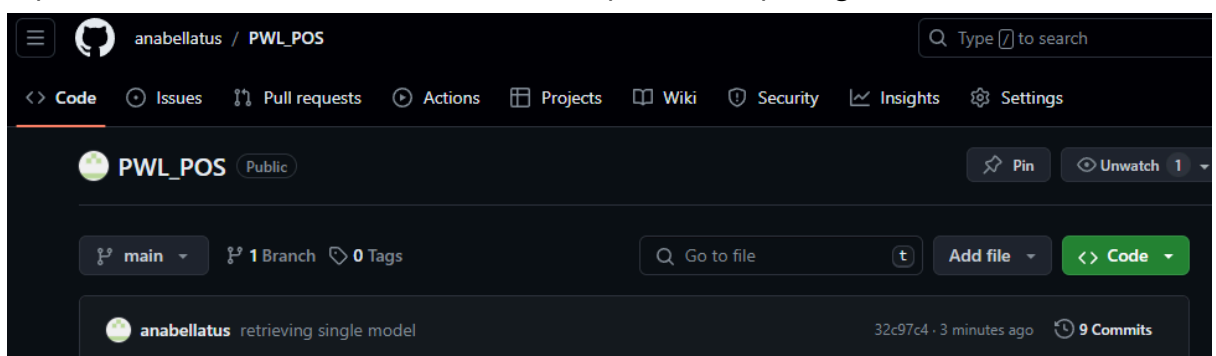
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         $user = UserModel::findor(20, ['username', 'nama'], function(){
14             abort(404);
15         });
16         return view('user', ['data' => $user]);
17     }

```

11. Simpan kode program Langkah 10. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



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

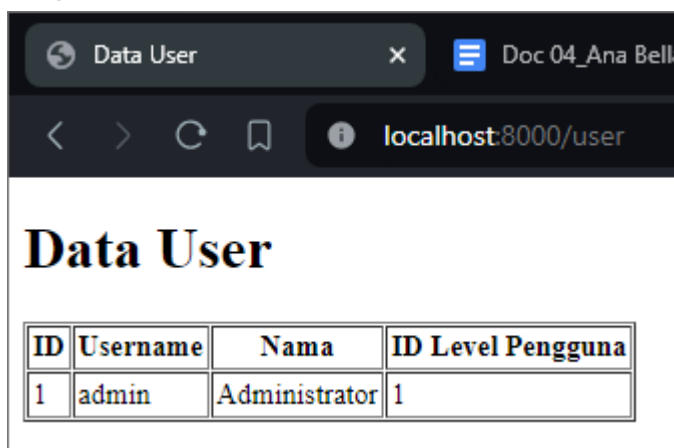


Praktikum 2.2 - Not Found Exceptions

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     {
13         $user = UserModel::findOrFail(1);
14         return view('user', ['data' => $user]);
15     }
16 }
```

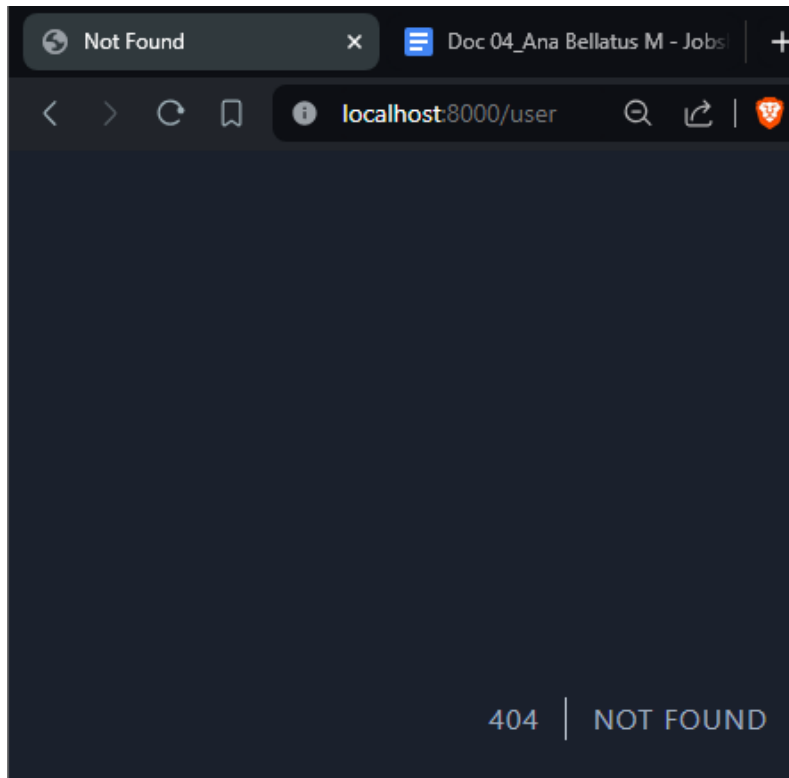
2. Simpan kode program Langkah 1. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



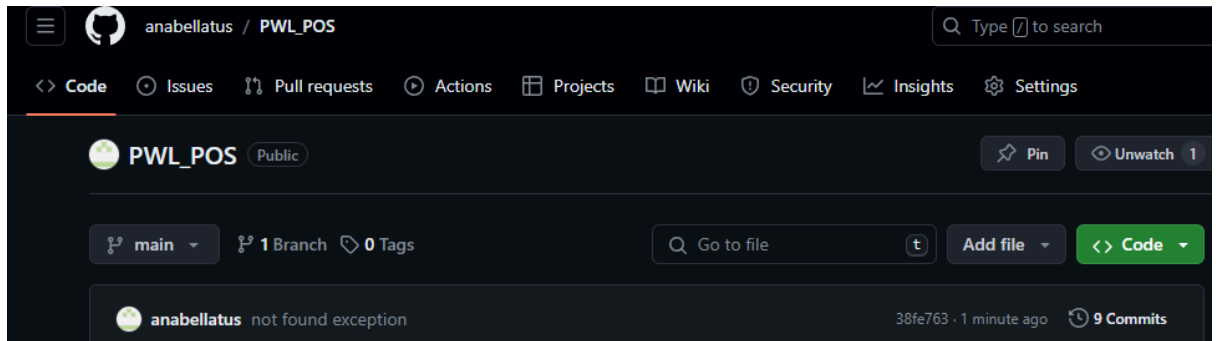
3. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         $user = UserModel::where('username', 'manager9')->firstOrFail();
14         return view('user', ['data' => $user]);
15     }
16 }
```

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



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

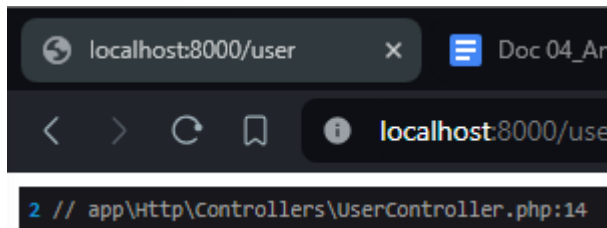


Praktikum 2.3 - Retrieving Aggregates

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     {
13         $user = UserModel::where('level_id', 2)->count();
14         dd($user);
15         return view('user', ['data' => $user]);
16     }
17 }
```

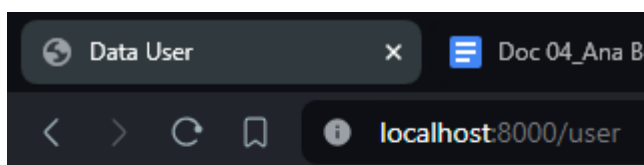

2. Simpan kode program Langkah 1. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



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

```
9 class UserController extends Controller
10 {
11     public function index()
12     {
13         $user = UserModel::where('level_id', 2)->count();
14         return view('user', ['data' => $user]);
15     }
16 }
```

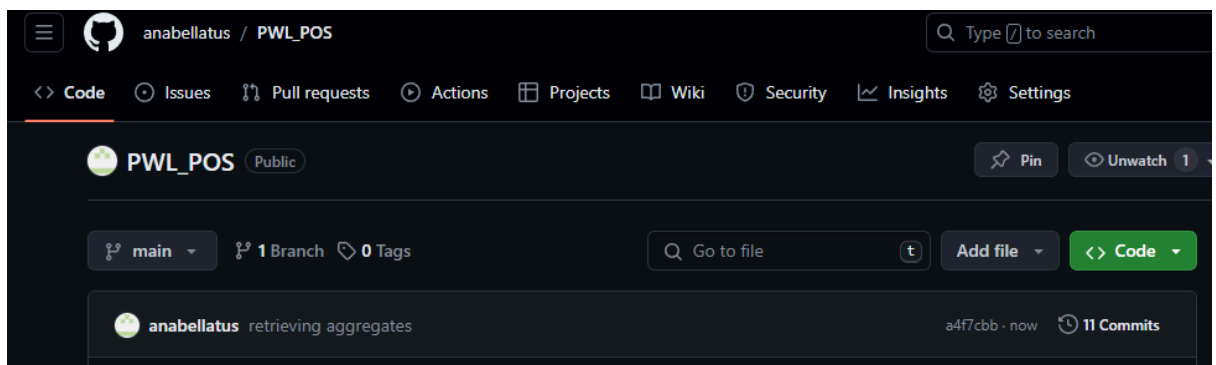
```
6 <body>
7     <h1>Data User</h1>
8     <table border="1" cellpadding="2" cellspacing="0">
9         <tr>
10             <th>Jumlah Pengguna</th>
11         </tr>
12         <tr>
13             <td>{{ $data }}</td>
14         </tr>
15     </table>
16 </body>
```



Data User

Jumlah Pengguna
2

4. Laporan hasil praktikum 2.3 ini dan commit perubahan pada git



Praktikum 2.4 - Retrieving or Creating Models

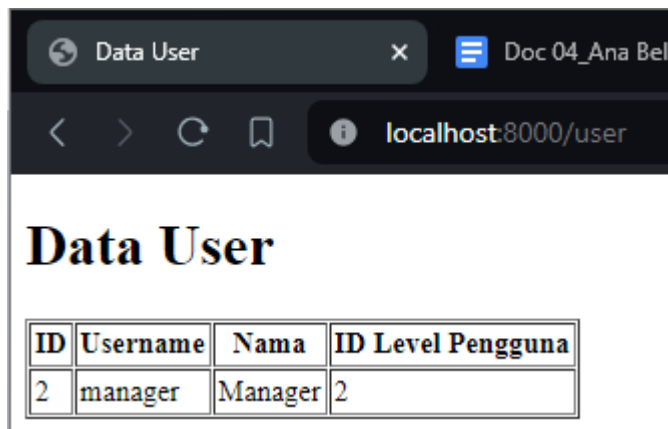
1. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         $user = UserModel::firstOrCreate(
14             [
15                 'username' => 'manager',
16                 'nama' => 'Manager',
17             ],
18         );
19         return view('user', ['data' => $user]);
20     }
```

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

```
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>
```

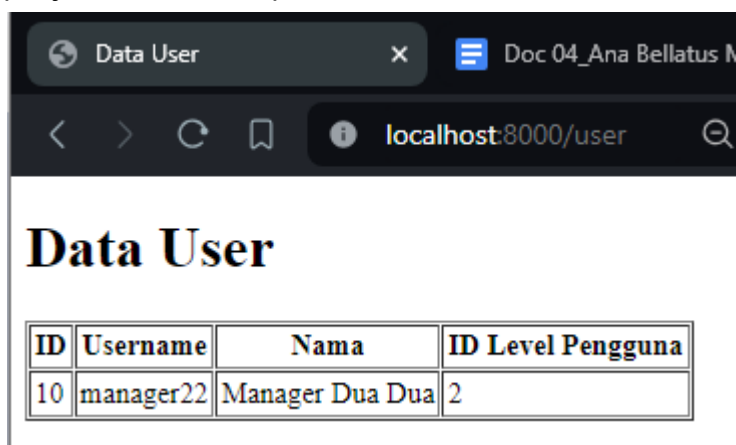
3. Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



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

```
11 class UserController extends Controller
12 {
13     public function index()
14     {
15         $user = UserModel::firstOrCreate(
16             [
17                 'username' => 'manager22',
18                 'nama' => 'Manager Dua Dua',
19                 'password' => Hash::make('12345'),
20                 'level_id' => 2,
21             ],
22         );
23         return view('user', ['data' => $user]);
24     }
25 }
```

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



user_id	level_id	username	nama	password	created_at	updated_at
1	1	admin	Administrator	\$2y\$12\$G0ZleXmDBBddPP35GNZpe2RjLcxss3iwtUY3DSA/M/...	NULL	NULL
2	2	manager	Manager	\$2y\$12\$qYFZLkc3OVbMDMOen63p0.1Rs0uzlQN1KgZB6C.NsXt...	NULL	NULL
3	3	staff	Staff/Kasir	\$2y\$12\$87W6e6ZhT1WaulXGzRSujuEnd8WG44OCX0o9evRGkD3...	NULL	NULL
6	4	customer-1	Pelanggan Pertama	\$2y\$12\$NnusYn8g42kg98jHH04C3e4SGAvDCPCzQ8Z5tL4yJ0E...	NULL	2024-03-13 01:22:18
9	2	manager_dua	Manager 2	\$2y\$12\$B8GZULsKi7XtDSOip6YfI/OroSvYyhNKbuEexjoypJ3...	2024-03-19 16:24:58	2024-03-19 16:24:58
10	2	manager22	Manager Dua Dua	\$2y\$12\$14Fzde0B45Gfcg2EuGMNYOmNbq8/0F.xlxK8MMXSig8...	2024-03-19 21:51:15	2024-03-19 21:51:15

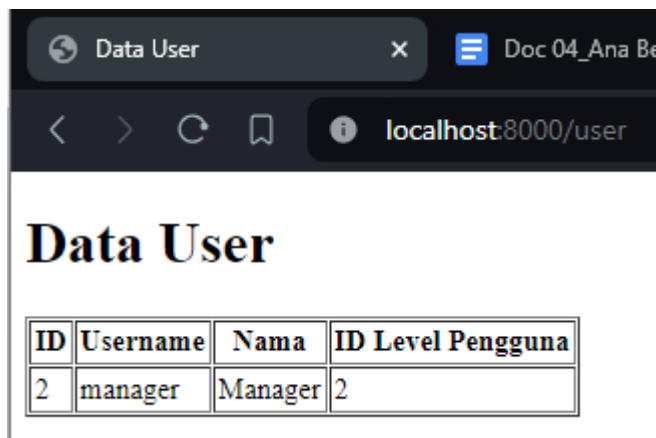
- Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```

11 class UserController extends Controller
12 {
13     public function index()
14     {
15         $user = UserModel::firstOrNew(
16             [
17                 'username' => 'manager',
18                 'nama' => 'Manager',
19             ],
20         );
21         return view('user', ['data' => $user]);
22     }

```

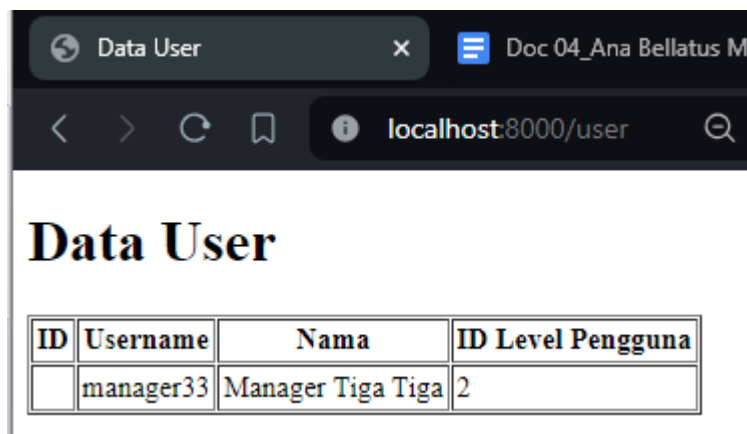
- Simpan kode program Langkah 6. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



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

```
11 class UserController extends Controller
12 {
13     public function index()
14     {
15         $user = UserModel::firstOrCreate(
16             [
17                 'username' => 'manager33',
18                 'nama' => 'Manager Tiga Tiga',
19                 'password' => Hash::make('12345'),
20                 'level_id' => 2,
21             ],
22         );
23         return view('user', ['data' => $user]);
24     }
25 }
```

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



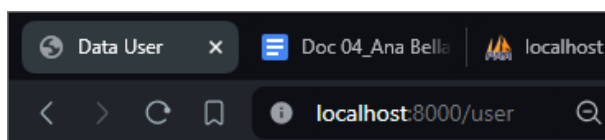
ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

user_id	level_id	username	nama	password	created_at	updated_at
1	1	admin	Administrator	\$2y\$12\$6G0ZleXmDBBddPP35GNZpe2RjLcxss3iwtUY3DSA/M...	NULL	NULL
2	2	manager	Manager	\$2y\$12\$qYFZLkc3OVBmDMOen63p0.tRs0uzlQN1KgZB6C.NsXt...	NULL	NULL
3	3	staff	Staff/Kasir	\$2y\$12\$87W6e6ZhT1WaulXGzRSJuEnd8WG44OCXOo9evRGkD3...	NULL	NULL
6	4	customer-1	Pelanggan Pertama	\$2y\$12\$NnusYn8g42kg98jHH04C3e4SGAvDCPCzQ8Z5tL4yJ0E...	NULL	2024-03-13 01:22:18
9	2	manager_dua	Manager 2	\$2y\$12\$B8GZULsKI7XtDSOip6Y/IOrSvYyhNKbuEexjoypJ3...	2024-03-19 16:24:58	2024-03-19 16:24:58
10	2	manager22	Manager Dua Dua	\$2y\$12\$14Fzde0B45Gfcg2EuGMNYOmNbq8/0F.xlxK8MMXSiG8...	2024-03-19 21:51:15	2024-03-19 21:51:15

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

```
11 class UserController extends Controller
12 {
13     public function index()
14     {
15         $user = UserModel::firstOrCreate(
16             [
17                 'username' => 'manager33',
18                 'nama' => 'Manager Tiga Tiga',
19                 'password' => Hash::make('12345'),
20                 'level_id' => 2,
21             ],
22         );
23         $user->save();
24
25         return view('user', ['data' => $user]);
26     }
27 }
```

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

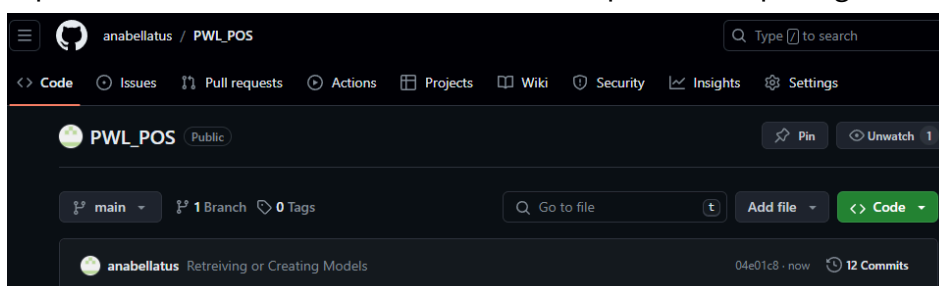


Data User

ID	Username	Nama	ID Level Pengguna
11	manager33	Manager Tiga Tiga	2

user_id	level_id	username	nama	password	created_at	updated_at
1	1	admin	Administrator	\$2y\$12\$G0ZleXmDBBddPP35GNZpe2RjLcxss3iwtUY3DSA/W...	NULL	NULL
2	2	manager	Manager	\$2y\$12\$qYFZLkc3OVBMdMOen63p0.tRs0uzIQN1KgZB6C.NsXt...	NULL	NULL
3	3	staff	Staff/Kasir	\$2y\$12\$87W6e6ZhT1WaulXGzRSujuEnd8WG44OCXOo9evRGkD3...	NULL	NULL
6	4	customer-1	Pelanggan Pertama	\$2y\$12\$NnusYn8g42kg98jHH04C3e4SGAvDCPCzQ8Z5tL4yJ0E...	NULL	2024-03-13 01:22:18
9	2	manager_dua	Manager 2	\$2y\$12\$B8GZULsK17XtDSOip6Y/OroSvYyhNKbuEexjoypJ3...	2024-03-19 16:24:58	2024-03-19 16:24:58
10	2	manager22	Manager Dua Dua	\$2y\$12\$t4Fzde0B45Gfcg2EuGMNYOmNbq8/0F.xlxK8MMXSiG8...	2024-03-19 21:51:15	2024-03-19 21:51:15
11	2	manager33	Manager Tiga Tiga	\$2y\$12\$iklJ4CgciGogH9j/BESK0mWFLemISVrhLAO/PDre/s...	2024-03-19 21:57:47	2024-03-19 21:57:47

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

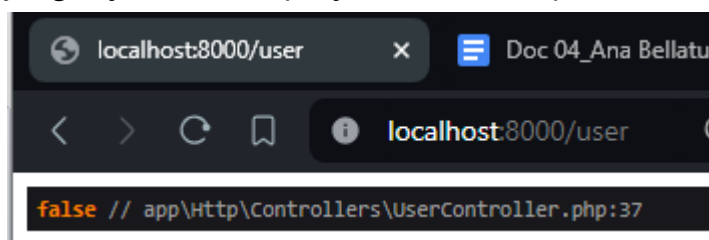


Praktikum 2.5 Attribute Changes

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

```
11  class UserController extends Controller
12  {
13      public function index()
14      {
15          $user = UserModel::create([
16              'username' => 'manager55',
17              'nama' => 'Manager55',
18              'password' => Hash::make('12345'),
19              'level_id' => 2,
20          ]);
21
22          $user->username = 'manager56';
23
24          $user->isDirty();
25          $user->isDirty('username');
26          $user->isDirty('nama');
27          $user->isDirty(['nama', 'username']);
28
29          $user->isClean();
30          $user->isClean('username');
31          $user->isClean('nama');
32          $user->isClean(['nama', 'username']);
33
34          $user->save();
35          $user->isDirty();
36          $user->isClean();
37          dd($user->isDirty());
38      }
```

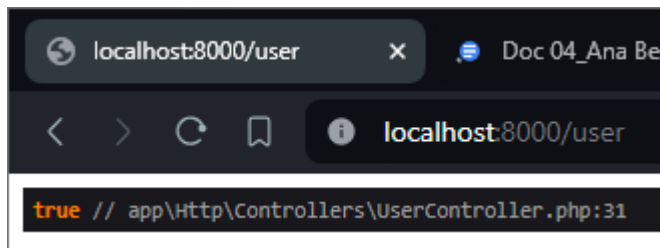
2. Simpan kode program Langkah 1. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



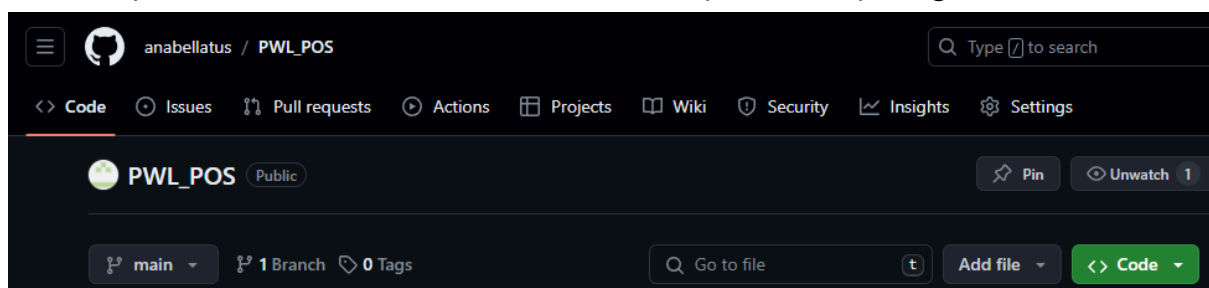
- Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
11 class UserController extends Controller
12 {
13     public function index()
14     {
15         $user = UserModel::create([
16             'username' => 'manager11',
17             'nama' => 'Manager11',
18             'password' => Hash::make('12345'),
19             'level_id' => 2,
20         ]);
21
22         $user->username = 'manager12';
23
24         $user->save();
25
26         $user->wasChanged();
27         $user->wasChanged('username');
28         $user->wasChanged(['username', 'password']);
29         $user->wasChanged('nama');
30         $user->isClean();
31         dd($user->wasChanged(['nama', 'username']));
32     }
```

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



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



Praktikum 2.6 – Create, Read, Update, Delete (CRUD)

1. Buka file view pada user.blade.php dan buat scripnya menjadi seperti dibawah ini

```
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                 <th>ID</th>
12                 <th>Username</th>
13                 <th>Nama</th>
14                 <th>ID Level Pengguna</th>
15                 <th>Aksi</th>
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></td>
25                 </tr>
26             @endforeach
27         </table>
```

2. Buka file controller pada UserController.php dan buat scriptnya untuk read menjadi seperti di bawah ini

```
11 class UserController extends Controller
12 {
13     public function index()
14     {
15         $user = UserModel::all();
16         return view('user', ['data' => $user]);
17     }
18 }
```

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	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
6	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager_dua	Manager 2	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
16	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

```
resources > views > user_tambah.blade.php > html
1  <html>
2      <head>
3          <title>Tambah Data</title>
4      </head>
5      <body>
6          <h1>Form Tambah Data User</h1>
7          <form method="post" action="/user/tambah_simpan">
8              {{ csrf_field() }}
9
10             <label>Username</label>
11             <input type="text" 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><br>
22             <input type="submit" class="btn btn-success" value="Simpan">
23         </form>
24     </body>
25 </html>
```

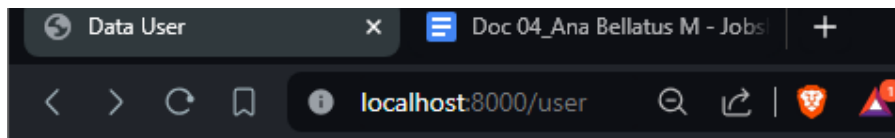
5. Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

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

6. 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

```
11 class UserController extends Controller
12 {
13     public function index()
14     {
15         $user = UserModel::all();
16         return view('user', ['data' => $user]);
17     }
18
19     public function tambah()
20     {
21         return view('user_tambah');
22     }
23 }
```

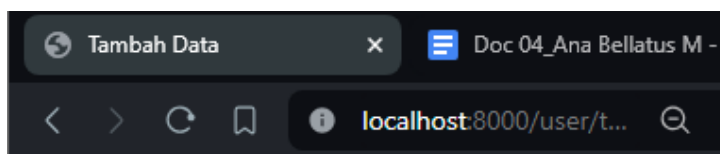
7. Simpan kode program Langkah 4 s/d 6. Kemudian jalankan pada browser dan klik link “+ Tambah User” amati apa yang terjadi dan beri penjelasan dalam laporan



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
6	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager_dua	Manager 2	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus



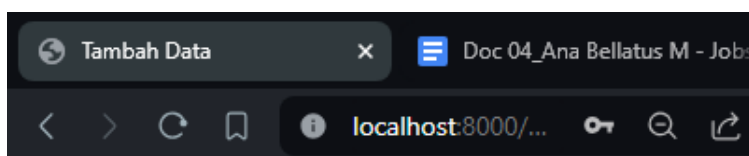
Form Tambah Data User

Username

Nama

Password

Level ID



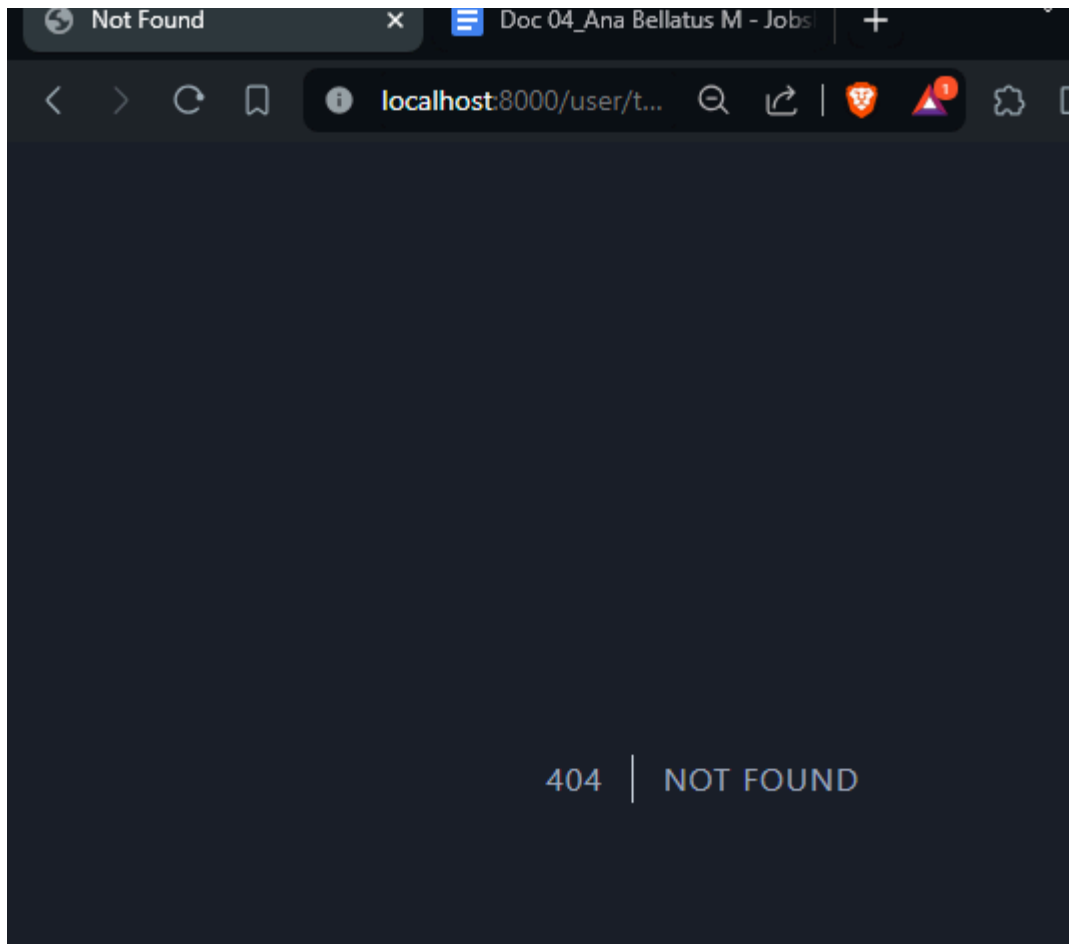
Form Tambah Data User

Username

Nama

Password

Level ID



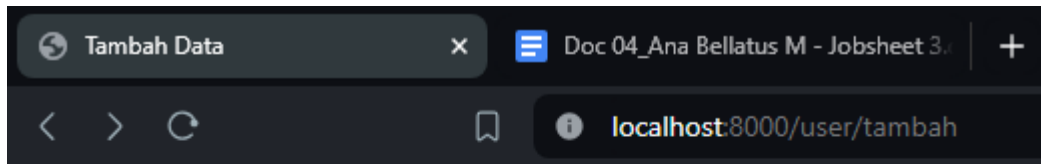
8. Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

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

9. 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

```
11 class UserController extends Controller
24     public function tambah_simpan(Request $request)
25     {
26         UserModel::create([
27             'username' => $request->username,
28             'nama' => $request->nama,
29             'password' => Hash::make('$request->password'
30             'level_id' => $request->level_id
31         ]);
32
33         return redirect('/user');
34     }
```

10. 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



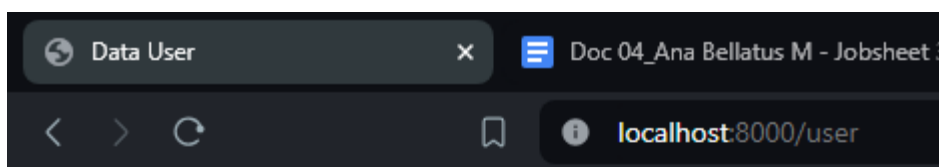
Form Tambah Data User

Username

Nama

Password

Level ID



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
6	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager_dua	Manager 2	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus
17	staff1	Staff Satu	3	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

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

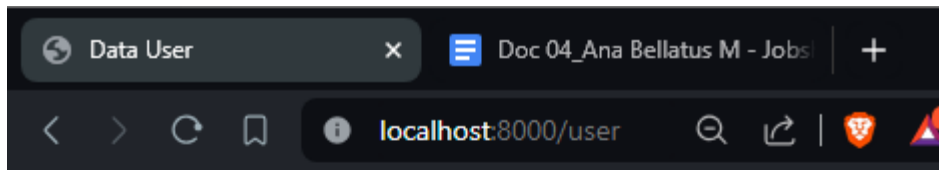
12. Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

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

13. 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

```
11 class UserController extends Controller
36
37     public function ubah($id)
38     {
39         $user = UserModel::find($id);
40         return view('user_ubah', ['data' => $user]);
41     }
```

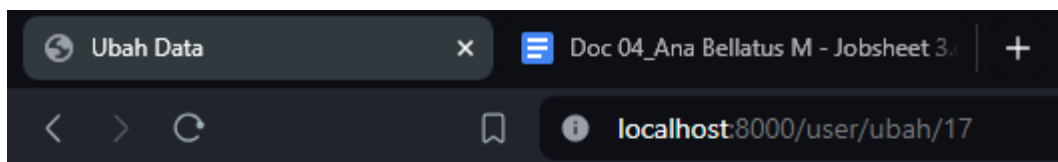
14. . Simpan kode program Langkah 11 sd 13. Kemudian jalankan pada browser dan klik link “Ubah” amati apa yang terjadi dan beri penjelasan dalam laporan



Data User

[+ Tambah User](#)

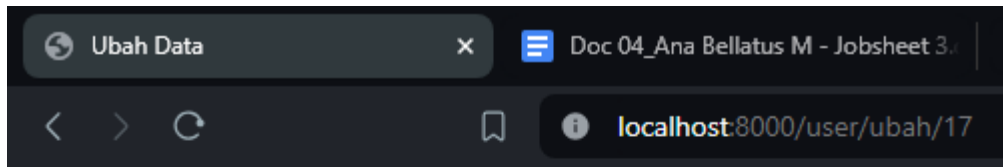
ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff Kasir	3	Ubah Hapus
6	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager_dua	Manager 2	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus
17	staff1	Staff Satu	3	Ubah Hapus



Form Ubah Data User

[Kembali](#)

Username	<input type="text" value="staff1"/>
Nama	<input type="text" value="staff1"/>
Password	<input type="password" value="....."/>
Level ID	<input type="text" value="3"/>
<input type="button" value="Ubah"/>	



Form Ubah Data User

[Kembali](#)

Username	<input type="text" value="staff2"/>
Nama	<input type="text" value="staff2"/>
Password	<input type="password" value="....."/>
Level ID	<input type="text" value="3"/>
<input type="button" value="Ubah"/>	

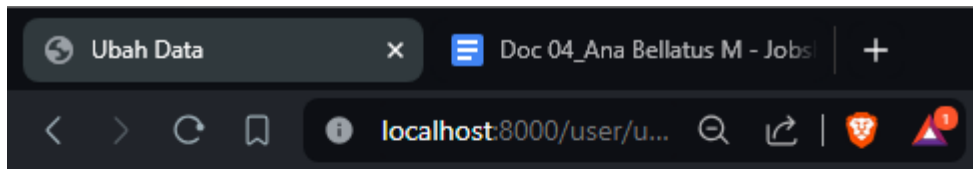
15. Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

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

16. 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 in

```
app > Http > Controllers > UserController.php > ...
11 class UserController extends Controller
42     public function ubah_simpan($id, Request $request){
43         $user = UserModel::find($id);
44
45         $user->username = $request->username;
46         $user->nama = $request->nama;
47         $user->password = Hash::make('$request->password');
48         $user->level_id = $request->level_id;
49         $user->save();
50
51         return redirect('/user');
52     }
```

17. 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

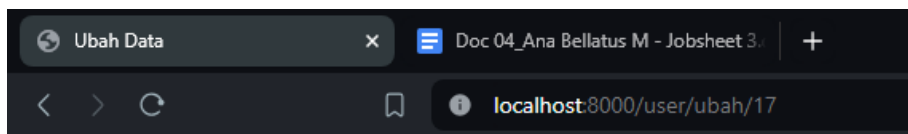
[Kembali](#)

Username

Nama

Password

Level ID



Form Ubah Data User

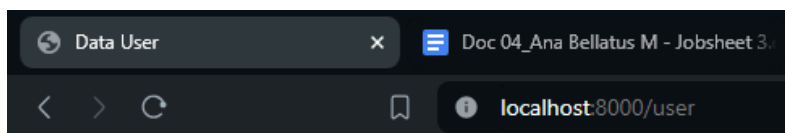
[Kembali](#)

Username

Nama

Password

Level ID



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
6	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager_dua	Manager 2	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus
17	staff2	Staff Dua	3	Ubah Hapus

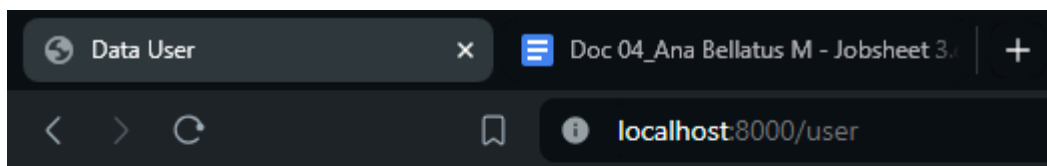
18. Berikut untuk langkah delete . Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

```
30 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 diletakan di bawah method ubah_simpan seperti gambar di bawah ini

```
11 class UserController extends Controller
54     public function hapus($id)
55     {
56         $user = UserModel::find($id);
57         $user->delete();
58
59         return redirect('/user');
60     }
```

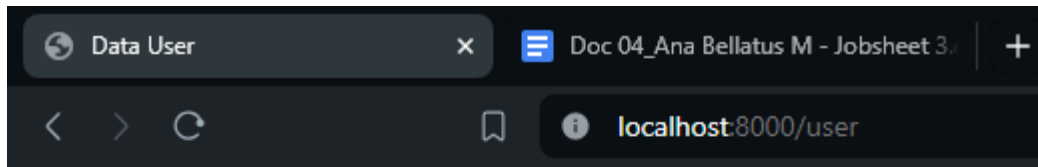
20. Simpan kode program Langkah 18 dan 19. Kemudian jalankan pada browser dan klik tombol hapus, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan



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
6	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager_dua	Manager 2	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus
17	staff2	Staff Dua	3	Ubah Hapus

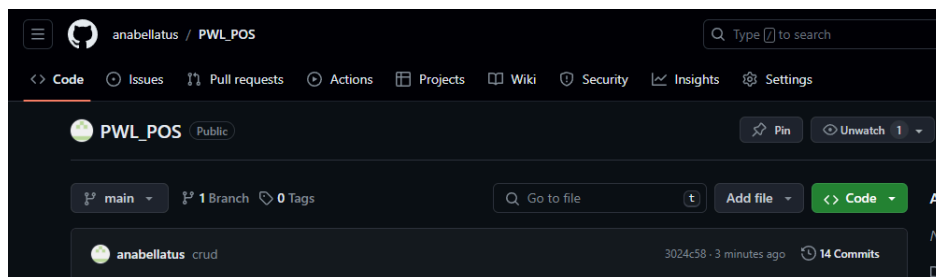


Data User

[+ 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
6	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager_dua	Manager 2	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus

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



Praktikum 2.7 - Relationships

1. Buka file model pada UserModel.php dan tambahkan scriptnya menjadi seperti di bawah ini

```

9  class UserModel extends Model
10 {
11     use HasFactory;
12
13     protected $table = 'm_user';
14     protected $primaryKey = 'user_id';
15
16     protected $fillable = ['level_id', 'username', 'password', 'nama'];
17
18     public function level(): BelongsTo
19     {
20         return $this->belongsTo(LevelModel::class);
21     }

```

2. Buka file controller pada UserController.php dan ubah method script menjadi seperti di bawah ini

```
11 class UserController extends Controller
12 {
13     public function index()
14     {
15         $user = UserModel::with('level')->get();
16         dd($user);
17         // return view('user', ['data' => $user]);
18     }
}
```

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

```
Illuminate\Database\Eloquent\Collection {#319 ▼ // app\Http\Controllers\UserController.php:16
  #items: array:9 [▼
    0 => App\Models\UserModel {#326 ▶}
    1 => App\Models\UserModel {#327 ▶}
    2 => App\Models\UserModel {#328 ▶}
    3 => App\Models\UserModel {#329 ▶}
    4 => App\Models\UserModel {#330 ▶}
    5 => App\Models\UserModel {#331 ▶}
    6 => App\Models\UserModel {#332 ▶}
    7 => App\Models\UserModel {#333 ▶}
    8 => App\Models\UserModel {#334 ▶}
  ]
  #escapeWhenCastingToString: false
}
```

- 4.

```
Illuminate\Database\Eloquent\Collection {#319 ▼ // app\Http\Controllers\UserController.php:16
  #items: array:9 [▼
    0 => App\Models\UserModel {#326 ▶}
    1 => App\Models\UserModel {#327 ▶}
    2 => App\Models\UserModel {#328 ▶}
    3 => App\Models\UserModel {#329 ▶}
    4 => App\Models\UserModel {#330 ▼}
    #connection: "mysql"
    #table: "m_user"
    #primaryKey: "user_id"
    #keyType: "int"
    +incrementing: true
    #with: []
    #withCount: []
    +preventsLazyLoading: false
    #perPage: 15
    +exists: true
    +wasRecentlyCreated: false
    #escapeWhenCastingToString: false
    #attributes: array:7 [▶]
    #original: array:7 [▶]
    #changes: []
    #casts: []
    #classCastCache: []
    #attributeCastCache: []
    #dateFormat: null
    #appends: []
    #dispatchesEvents: []
    #observables: []
    #relations: array:1 [▶]
    #touches: []
    +timestamps: true
    +usesUniqueIds: false
    #hidden: []
    #visible: []
    #fillable: array:4 [▶]
    #guarded: array:1 [▶]
  ]
  5 => App\Models\UserModel {#331 ▶}
  6 => App\Models\UserModel {#332 ▶}
  7 => App\Models\UserModel {#333 ▶}
  8 => App\Models\UserModel {#334 ▶}
  #escapeWhenCastingToString: false
}
```

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

