



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

Praktikum 1 - \$fillable

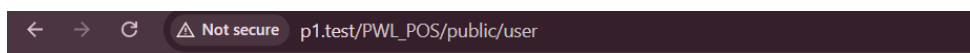
1. Buka file model dengan nama UserModel.php dan tambahkan \$fillable

```
8 class UserModel extends Model
9 {
10     use HasFactory;
11
12     0 references
13     protected $table = 'm_user'; //mendefinisikan nama tabel yang digunakan oleh model ini
14     0 references
15     protected $primaryKey = 'user_id'; //mendefinisikan primary key tabel yang digunakan oleh model ini
16
17     0 references
18     protected $fillable = [
19         'level_id',
20         'username',
21         'nama',
22         'password'
23     ];
24 }
```

2. Buka file controller dengan nama UserController.php dan ubah *script* untuk menambahkan data baru

```
36
37 //-----JS 4 -----
38 $data = [
39     'level_id' => 2,
40     'username' => 'manager-2',
41     'nama' => 'Manager 2',
42     'password' => Hash::make('12345')
43 ];
44 UserModel::create($data);
45 $user = UserModel::all();
46 return view('user', data: ['data' => $user]);
47 }
48
49 }
```

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
8	customer-1	Pelanggan Pertama	4
9	manager-2	Manager 2	2

Menampilkan data beberapa user



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

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

```
0 references
15 protected $fillable = [
16     'level_id',
17     'username',
18     'nama',
19 ];
20 ];
21 }
22
```

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

```
37 //-----JS 4 -----
38 $data = [
39     'level_id' => 2,
40     'username' => 'manager_tiga',
41     'nama' => 'Manager 3',
42     'password' => Hash::make('12345')
43 ];
44 UserModel::create($data);
45 $user = UserModel::all();
46 return view('user', data: ['data' => $user]);
47 }
48 }
49
```

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

```
Illuminate \ Database \ QueryException
PHP 8.1.10 10:48:28

SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value

INSERT INTO `m_user` (`level_id`, `username`, `nama`, `updated_at`, `created_at`) VALUES (2, manager_tiga, Manager 3, 2025-03-08 12:03:10, 2025-03-08 12:03:10)
```

Terjadi eror karena di protected \$fillable tidak ada password saat di insert



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

Praktikum 2.1 – Retrieving Single Models

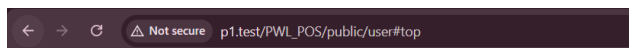
1. Buka file controller dengan nama UserController.php dan ubah *script*

```
48  
49 $user = UserModel::find(1);  
50 return view('user', data: ['data' => $user]);  
51  
52 }  
53
```

2. Buka file *view* dengan nama user.blade.php dan ubah *script*

```
PWL_POS > resources > views > user.blade.php > html > body > table  
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4 <head>  
5 <meta charset="UTF-8">  
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7 <title>Data User</title>  
8 </head>  
9  
10 <body>  
11  
12 <h1>Data User</h1>  
13 <table border="1" cellpadding="2" cellspacing="0">  
14 <tr>  
15 <th>ID</th>  
16 <th>Username</th>  
17 <th>Nama</th>  
18 <th>ID Level Pengguna</th>  
19 </tr>  
20 <tr>  
21 <th>{{ $data->user_id }}</th>  
22 <th>{{ $data->username }}</th>  
23 <th>{{ $data->nama }}</th>  
24 <th>{{ $data->level_id }}</th>  
25 </tr>  
26 </table>  
27 </body>  
28 </html>
```

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

Data yang ditampilkan hanya data idlevel 1

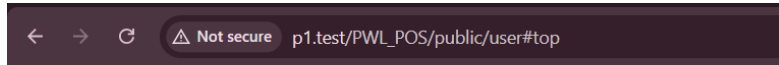
4. Ubah file controller dengan nama UserController.php dan ubah *script* seperti gambar di bawah ini

```
0 $user = UserModel::where('level_id', 1)->first();  
1 return view('user', data: ['data' => $user]);  
2 }
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

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



Data User

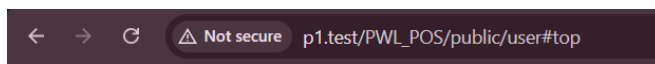
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

Data menampilkan user id 1 jika tidak maka akan not found

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

```
52 $user = UserModel::firstWhere('level_id', 1);  
53 return view(view: 'user', data: ['data' => $user]);  
54 }  
55 }
```

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



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

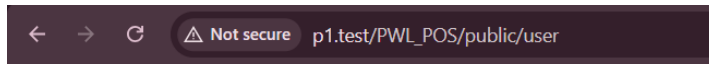
8. Ubah file controller dengan nama UserController.php dan ubah *script* seperti gambar di bawah ini

```
52 $user = UserModel::findOr(1, ['username', 'nama'], function(): never {  
53 abort(code: 404);  
54 });  
55 return view(view: 'user', data: ['data' => $user]);  
56 }  
57 }  
58 }  
59 }
```

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM



Data User

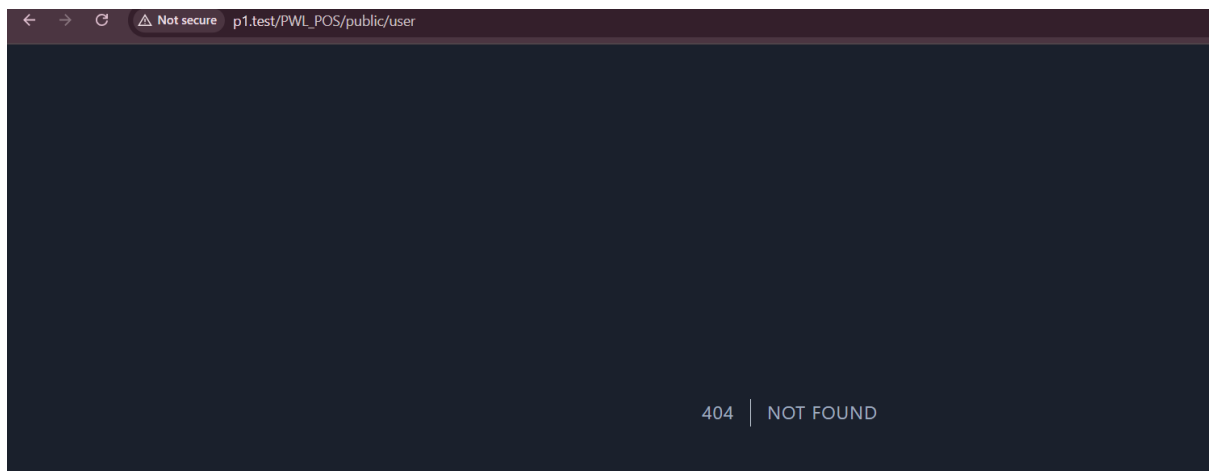
ID	Username	Nama	ID Level Pengguna
	admin	Administrator	

Program tersebut akan mencari id 1 dan akan menampilkan username dan nama

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

```
57     $user = UserModel::findOr(20, ['username', 'nama'], function(): never {  
58         abort(code: 404);  
59     });  
60     return view(view: 'user', data: ['data' => $user]);  
61 }  
62 }  
63 }
```

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



User 20 tidak ditemukan maka akan menampilkan fungsi dari 404



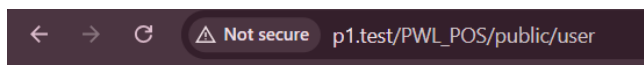
NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

Praktikum 2.2 – Not Found Exceptions

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

```
62 $user = UserModel::findOrFail(1);  
63 return view('user', data: ['data' => $user]);  
64  
65 }  
66
```

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



Data User

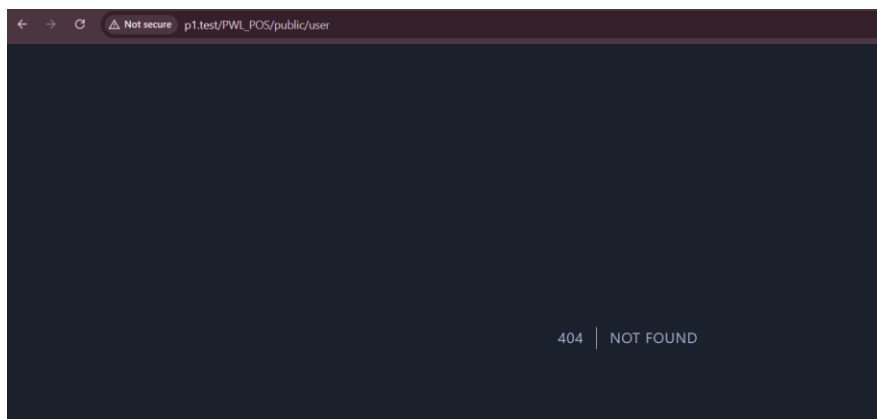
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

Pada browser ditampilkan user id 1, username, nama dan id level pengguna

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

```
64 $user = UserModel::where('username', 'manager9')->firstOrFail();  
65 return view('user', data: ['data' => $user]);  
66  
67 }  
68
```

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



Data manager 9 not found maka akan tampil mengembalikan function findOrFail()



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

Praktikum 2.3 – Retrieving Aggregates

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

```
66 $user = UserModel::where('level_id', 2)->count();  
67 dd(vars: $user);  
68 return view(view: 'user', data: ['data' => $user]);  
69 }  
70 ]  
71
```

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

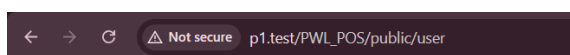


Menghitung record pada level id 2

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

```
65 // $user = UserModel::where('username', 'manager')->first();  
66 $user = UserModel::where('level_id', 2)->count();  
67 // $user = UserModel::add($user);  
68 return view(view: 'user', data: ['data' => $user]);  
69 }  
70 ]  
71
```

```
PWL_POS > resources > views > user.blade.php > html > body > table > p  
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4 <head>  
5 <meta charset="UTF-8">  
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7 <title>Data User</title>  
8 </head>  
9  
10 <body>  
11  
12 <h1>Data User</h1>  
13 <table border="1" cellpadding="2" cellspacing="0">  
14  
15 <p>berjumlah {{ $data }}</p>  
16  
17 </table>  
18 </body>  
19  
20 </html>
```



Data User

berjumlah 2



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

Praktikum 2.4 – Retrieving or Creating Models

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

```
68
69     $user = UserModel::firstOrCreate(
70         [
71             'username' => 'manager',
72             'nama' => 'Manager',
73         ]
74     );
75
76     return view('user', data: ['data' => $user]);
77 }
78 }
79
```

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

```
PWL_POS > resources > views > user.blade.php > html > body > table > tr > th
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Data User</title>
8 </head>
9
10 <body>
11
12     <h1>Data User</h1>
13     <table border="1" cellpadding="2" cellspacing="0">
14         <tr>
15             <th>ID</th>
16             <th>Username</th>
17             <th>Nama</th>
18             <th>ID Level Pengguna</th>
19         </tr>
20         <tr>
21             <th>{{ $data->user_id }}</th>
22             <th>{{ $data->username }}</th>
23             <th>{{ $data->nama }}</th>
24             <th>{{ $data->level_id }}</th>
25         </tr>
26     </table>
27
28 </body>
29 </html>
```

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
2	manager	Manager	2

Data 2 ditemukan dan tampil di web browser, jika tidak maka akan membuat data baru

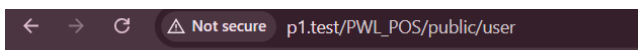


NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

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

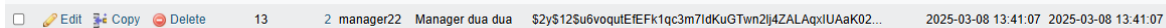
```
69     $user = UserModel::firstOrCreate(  
70         [  
71             'username' => 'manager22',  
72             'nama' => 'Manager dua dua',  
73             'password' => Hash::make('12345'),  
74             'level_id' => 2  
75         ]  
76     );  
77  
78     return view('user', data: ['data' => $user]);  
79 }  
80  
81
```

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



Data User

ID	Username	Nama	ID Level Pengguna
13	manager22	Manager dua dua	2



Data baru dibuat dikarenakan pada data sebelumnya id tidak ditemukan

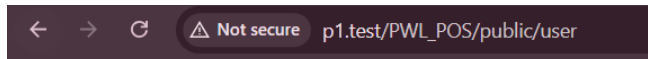
6. Ubah file controller dengan nama UserController.php dan ubah *script* seperti gambar di bawah ini

```
78     $user = UserModel::firstOrCreate(  
79         [  
80             'username' => 'manager',  
81             'nama' => 'Manager'  
82         ]  
83     );  
84  
85     return view('user', data: ['data' => $user]);  
86 }  
87  
88
```

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM



Data User

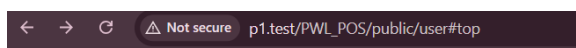
ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

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

```
86     $user = UserModel::firstOrCreate(
87         [
88             'username' => 'manager33',
89             'nama' => 'Manager tiga tiga',
90             'password' => Hash::make('12345'),
91             'level_id' => 2
92         ],
93     );
94
95     return view('view: user', data: ['data' => $user]);
96 }
97
98
```

User 2 ditemukan dikarenakan sudah ada pada data sebelumnya jika tidak ada maka akan dibuat objek baru tetapi tidak langsung disimpan ke dalam database

9. 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
	manager33	Manager tiga tiga	2

Berhasil membuat objek namun tidak langsung disimpan dalam database, jika ingin menyimpan langsung harus menambahkan *user save*

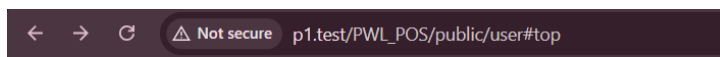
10. Ubah file controller dengan nama UserController.php dan ubah *script* seperti gambar di bawah ini



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

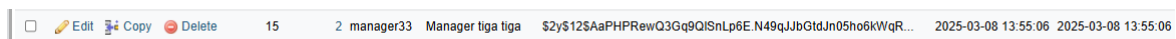
```
85  
86     $user = UserModel::firstOrCreate(  
87         [  
88             'username' => 'manager33',  
89             'nama' => 'Manager tiga tiga',  
90             'password' => Hash::make('12345'),  
91             'level_id' => 2  
92         ],  
93     );  
94     $user->save();  
95  
96     return view('user', data: ['data' => $user]);  
97 }  
98 }  
99
```

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
15	manager33	Manager tiga tiga	2



Data berhasil disimpan dalam database dikarenakan sudah menambahkan kode `$user->save();`



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

Praktikum 2.5 – Attribute Changes

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

```
115 //-----praktikum 2.5 attribute changes
116 $user = UserModel::create(
117     [
118         'username' => 'manager55',
119         'nama' => 'Manager55',
120         'password' => Hash::make('12345'),
121         'level_id' => 2
122     ],
123 );
124
125 $user->username = 'manager56';
126
127 $user->isDirty(); //true
128 $user->isDirty('username'); //true
129 $user->isDirty('nama'); //false
130 $user->isDirty('nama', 'username'); //true
131
132 $user->isClean(); //false
133 $user->isClean('username'); //false
134 $user->isClean('nama'); //true
135 $user->isClean('nama', 'username'); //false
136
137 $user->save();
138
139 $user->isDirty(); //false
140 $user->isClean(); //true
141 dd(vars: $user->isDirty());
142
143
144 //return view('user', ['data' => $user]);
145 }
146
147
```

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

← → ↻ ⚠ Not secure p1.test/PWL_POS/public/user#top

false // app\Http\Controllers\UserController.php:141

Hasil false berarti attribute belum pernah terganti

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

```
145  
146 $user = UserModel::create(  
147     [  
148         'username' => 'manager11',  
149         'nama' => 'Manager11',  
150         'password' => Hash::make('12345'),  
151         'level_id' => 2  
152     ],  
153 );  
154  
155 $user->username = 'manager12';  
156 $user->save();  
157  
158 $user->wasChanged();//true  
159 $user->wasChanged('username'); //false  
160 $user->wasChanged('username', 'level_id'); //true  
161 $user->wasChanged('nama'); //false  
162 dd($user->wasChanged(['nama', 'username'])); //true  
163  
164  
165 }  
166  
167
```

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

← → ↻ Not secure p1.test/PWL_POS/public/user#top
true // app\Http\Controllers\UserController.php:162

Hasil true berarti attribute sudah pernah terganti



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

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

1. Buka file *view* pada *user.blade.php* dan buat scripnya menjadi seperti di bawah ini

```
PWL_POS > resources > views > user.blade.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Data User</title>
8 </head>
9
10 <body>
11
12   <!-- Praktikum 2.6 -->
13
14   <h1>Data User</h1>
15   <a href="user/tambah">+ Tambah User</a>
16   <table border="1" cellpadding="2" cellspacing="0">
17     <tr>
18       <td>ID</td>
19       <td>Username</td>
20       <td>Nama</td>
21       <td>ID Level Pengguna</td>
22       <td>Aksi</td>
23     </tr>
24     @foreach ($data as $d)
25       <tr>
26         <td>{{ $d->user_id }}</td>
27         <td>{{ $d->username }}</td>
28         <td>{{ $d->nama }}</td>
29         <td>{{ $d->level_id }}</td>
30         <td><a href="/user/ubah/ {{ $d->user_id }}">Ubah</a> | <a href="/user/hapus/ {{ $d->user_id }}">Hapus</a></td>
31       </tr>
32     @endforeach
33   </table>
34 </body>
35
36 </html>
```

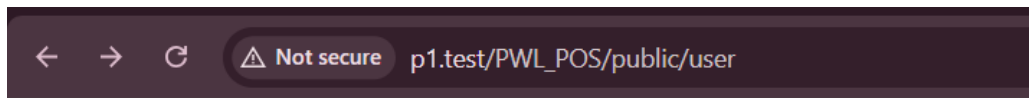
2. Buka file controller pada *UserController.php* dan buat scriptnya untuk *read* menjadi seperti di bawah ini

```
189
190 //praktikum 2.6
191 $user = UserModel::all();
192 return view(view: 'user', data: ['data' => $user]);
193 }
194 }
195
```

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM



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">
8	customer-1	Pelanggan Pertama	4	Ubah Hapus">
9	manager-2	Manager 2	2	Ubah Hapus">
13	manager22	Manager dua dua	2	Ubah Hapus">
15	manager33	Manager tiga tiga	2	Ubah Hapus">
16	manager56	Manager55	2	Ubah Hapus">
17	manager12	Manager11	2	Ubah Hapus">

Menampilkan tabel user ditambah ada crud, namun belum bisa dikarenakan belum dibuat

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

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

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

```
195 |  
196 | //praktikum 2.6  
197 | 1 reference | 0 overrides | Codeium: Refactor | Explain | X  
198 | public function tambah(): Factory|View  
199 | {  
200 |     return view(view: 'user_tambah');  
201 | }  
202 |
```

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

← → ↻ ⚠ Not secure p1.test/PWL_POS/public/user/tambah

Form Tambah Data User

Username

Nama

Password

Level ID

Masuk ke halaman tambah user

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

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

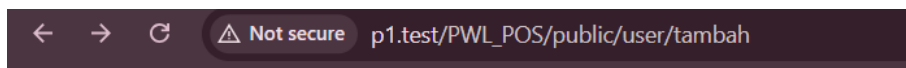
9. Tambahkan script pada controller dengan nama file UserController.php. Tambahkan script dalam class dan buat method baru dengan nama tambah_simpan dan diletakan di bawah method tambah seperti gambar di bawah ini



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

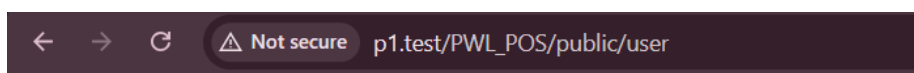
```
201 |  
202 | 1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X  
203 | public function tambah_simpan(Request $request): Redirector|RedirectResponse{  
204 |     UserModel::create([  
205 |         'username' => $request->username,  
206 |         'nama' => $request->nama,  
207 |         'password' => Hash::make($request->password),  
208 |         'level_id' => $request->level_id  
209 |     ]);  
210 |  
211 |     return redirect(to: '/user');  
212 | }  
213 |
```

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
8	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager-2	Manager 2	2	Ubah Hapus
13	manager22	Manager dua dua	2	Ubah Hapus
15	manager33	Manager tiga tiga	2	Ubah Hapus
16	manager56	Manager55	2	Ubah Hapus
17	manager12	Manager11	2	Ubah Hapus
18	dikarriky	DIKA ARIE ARRIFKY	1	Ubah Hapus
22	dika	DIKA ARIE	2	Ubah Hapus



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

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

```
PWL_POS > resources > views > user_ubah.blade.php > html
2 <html lang="en">
8 </head>
9
10 <body>
11 <h1>Form Ubah User</h1>
12 <a href="/user">Kembali</a>
13
14 <br><br>
15 <form action="ubah_simpan/{{ $data->user_id }}" method="POST">
16     {{ csrf_field() }}
17     {{ method_field(method: 'PUT') }}
18
19     <label for="username">Username:</label>
20     <input type="text" name="username" id="username" placeholder="Masukkan Username" value="{{ $data->username }}">
21     <br>
22     <label for="nama">Nama User:</label>
23     <input type="text" name="nama" id="nama" placeholder="Masukkan Nama User" value="{{ $data->nama }}">
24     <br>
25     <label for="password">Password:</label>
26     <input type="password" name="password" id="password" placeholder="Masukkan Password" value="{{ $data->password }}">
27     <br>
28     <label for="level_id">ID Level:</label>
29     <input type="number" name="level_id" id="level_id" placeholder="Masukkan ID Level" value="{{ $data->level_id }}">
30     <br>
31     <input type="submit" value="Ubah" class="btn btn-success">
32 </form>
33 </body>
34
35 </html>
```

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

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

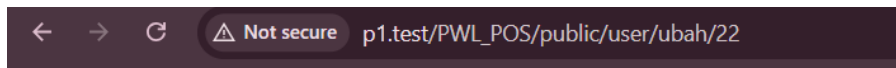
13. Tambahkan *script* pada controller dengan nama file `UserController.php`. Tambahkan *script* dalam class dan buat method baru dengan nama `ubah` dan diletakan di bawah method `tambah_simpan` seperti gambar di bawah ini

```
1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
213 public function ubah($id): Factory|View{
214     $user = UserModel::find($id);
215     return view(view: 'user_ubah', data: ['data' => $user]);
216 }
217
218
```

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM



Form Ubah User

[Kembali](#)

Username:
Nama User:
Password:
ID Level:

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

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

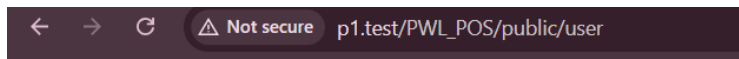
16. Tambahkan *script* pada controller dengan nama file UserController.php. Tambahkan *script* dalam class dan buat method baru dengan nama *ubah_simpan* dan diletakkan di bawah method *ubah* seperti gambar di bawah ini

```
217  
218 1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X  
219 public function ubah_simpan($id, Request $request): Redirector|RedirectResponse{  
220     $user = UserModel::find($id);  
221     $user->username = $request->username;  
222     $user->nama = $request->nama;  
223     $user->password = Hash::make($request->password);  
224     $user->level_id = $request->level_id;  
225     $user->save();  
226     return redirect(to: '/user');  
227 }  
228 }
```

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM



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
8	customer-1	Pelanggan Pertama	4	Ubah Hapus
9	manager-2	Manager 2	2	Ubah Hapus
13	manager22	Manager dua dua	2	Ubah Hapus
15	manager33	Manager tiga tiga	2	Ubah Hapus
16	manager56	Manager55	2	Ubah Hapus
17	manager12	Manager11	2	Ubah Hapus
18	dikarriky	DIKA ARIE ARRIFKY	1	Ubah Hapus
22	dika	DIKA ARIE 2A	2	Ubah Hapus

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

```
31 | 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

```
229 | 1 reference | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X
230 | public function hapus($id): Redirector|RedirectResponse{
231 |     $user = UserModel::find($id);
232 |     $user->delete();
233 |     return redirect(to: '/user');
234 | }
235 |
```

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

Menghapus data username dika



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

← → ↻ ⚠ Not secure p1.test/PWL_POS/public/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 >
8	customer-1	Pelanggan Pertama	4	Ubah Hapus >
9	manager-2	Manager 2	2	Ubah Hapus >
13	manager22	Manager dua dua	2	Ubah Hapus >
15	manager33	Manager tiga tiga	2	Ubah Hapus >
16	manager56	Manager55	2	Ubah Hapus >
17	manager12	Manager11	2	Ubah Hapus >
18	dikarriky	DIKA ARIE ARRIFKY	1	Ubah Hapus >



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

Praktikum 2.7 – Relationships

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

```
23 |  
24 | 0 references | 0 overrides | Codeium: Refactor | Explain | Generate Function Comment | X  
25 | public function level():BelongsTo{  
26 |     return $this->belongsTo(LevelModel::class, 'level_id','level_id');  
27 | }  
28 | }  
29 |
```

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

```
194 | //praktikum 2.7  
195 | $user = UserModel::with('level')->get();  
196 | dd(vars: $user);  
197 | }
```

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

```
11 items - array(10) of App\Models\UserModel (#319)
1 -> App\Models\UserModel (#319)
2 -> App\Models\UserModel (#320)
3 -> App\Models\UserModel (#321)
4 -> App\Models\UserModel (#322)
5 -> App\Models\UserModel (#323)
6 -> App\Models\UserModel (#324)
7 -> App\Models\UserModel (#325)
8 -> App\Models\UserModel (#326)
9 -> App\Models\UserModel (#327)
10 -> App\Models\UserModel (#328)
! Rescaped as CastingToString: false
```

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

```
196 | //dd($user);  
197 | return view(view: 'user', data: ['data' => $user]);  
198 | }
```

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 04 - Eloquent ORM

```
PWL_POS > resources > views > user.blade.php > html > body > table > tr
2 <html lang="en">
10 <body>
14 <div data-user="">
15 <a href="user/tambah">+ Tambah User</a>
16 <table border="1" cellpadding="2" cellspacing="0">
17 <tr>
18 <td>ID</td>
19 <td>Username</td>
20 <td>Nama</td>
21 <td>ID Level Pengguna</td>
22 <td>Level Kode</td>
23 <td>Nama Level</td>
24 <td>Aksi</td>
25 </tr>
26 @foreach ($data as $d)
27 <tr>
28 <td>{{ $d->user_id }}</td>
29 <td>{{ $d->username }}</td>
30 <td>{{ $d->nama }}</td>
31 <td>{{ $d->level_id }}</td>
32 <td>{{ $d->level->level_kode }}</td>
33 <td>{{ $d->level->level_nama }}</td>
34 <td><a href="user/ubah/{{ $d->user_id }}">Ubah</a> | <a href="user/hapus/{{ $d->user_id }}">Hapus</a></td>
35 </tr>
36 @endforeach
37 </table>
38 </body>
39
40 </html>
```

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	Level Kode	Nama Level	Aksi
1	admin	Administrator	1	ADM	Administrator	Ubah Hapus
2	manager	Manager	2	MNG	Manager	Ubah Hapus
3	staff	Staff/Kasir	3	STF	Staff/Kasir	Ubah Hapus
8	customer-1	Pelanggan Pertama	4	CUS	Customer	Ubah Hapus
9	manager-2	Manager 2	2	MNG	Manager	Ubah Hapus
13	manager22	Manager dua dua	2	MNG	Manager	Ubah Hapus
15	manager33	Manager tiga tiga	2	MNG	Manager	Ubah Hapus
16	manager56	Manager55	2	MNG	Manager	Ubah Hapus
17	manager12	Manager11	2	MNG	Manager	Ubah Hapus
18	dikarriky	DIKA ARIE ARRIFKY	1	ADM	Administrator	Ubah Hapus