

LAPORAN PRAKTIKUM 4

Mata Kuliah Pemrograman Website Lanjut

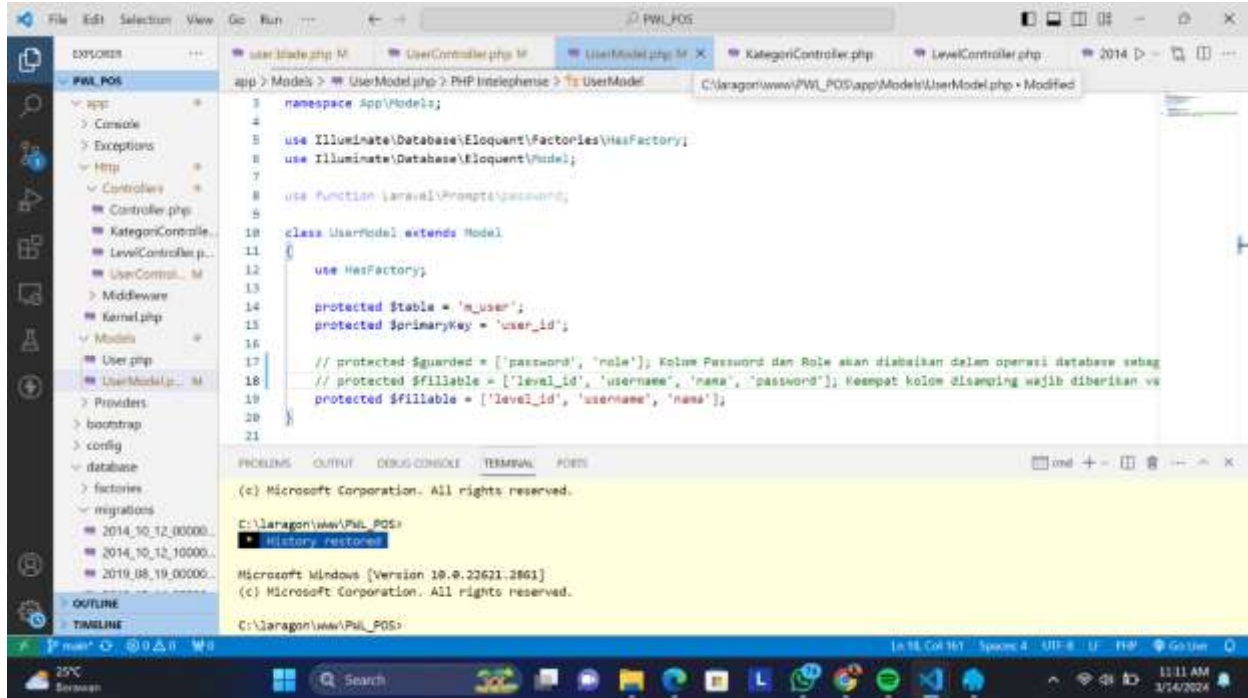


Oleh :

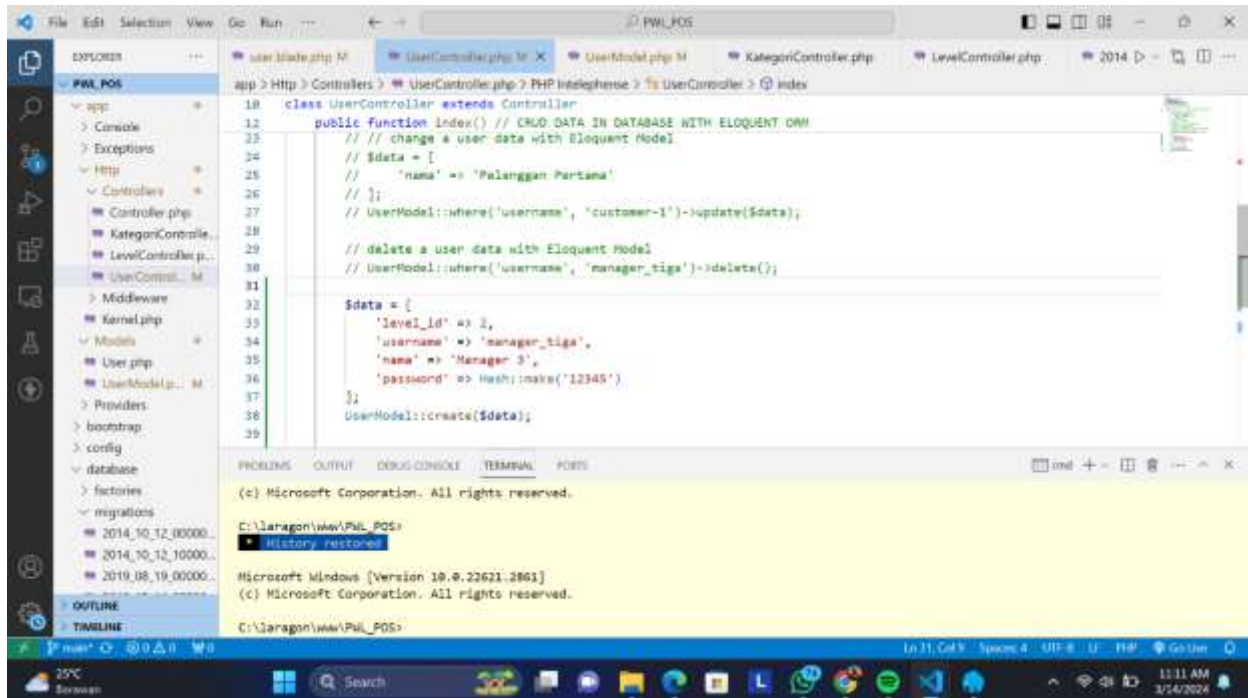
Avicenna Mumtaza
NIM 2241720112

Teknik Informatika 2H

A. Praktikum 1.1 - \$fillable

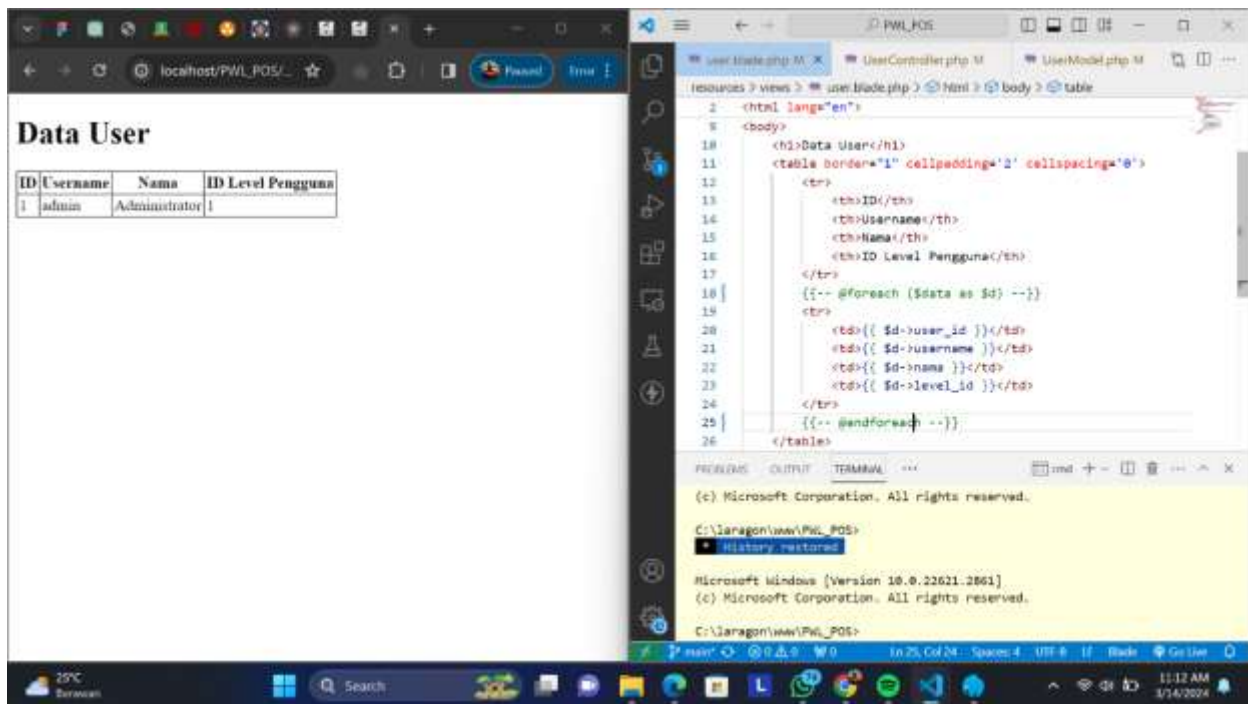
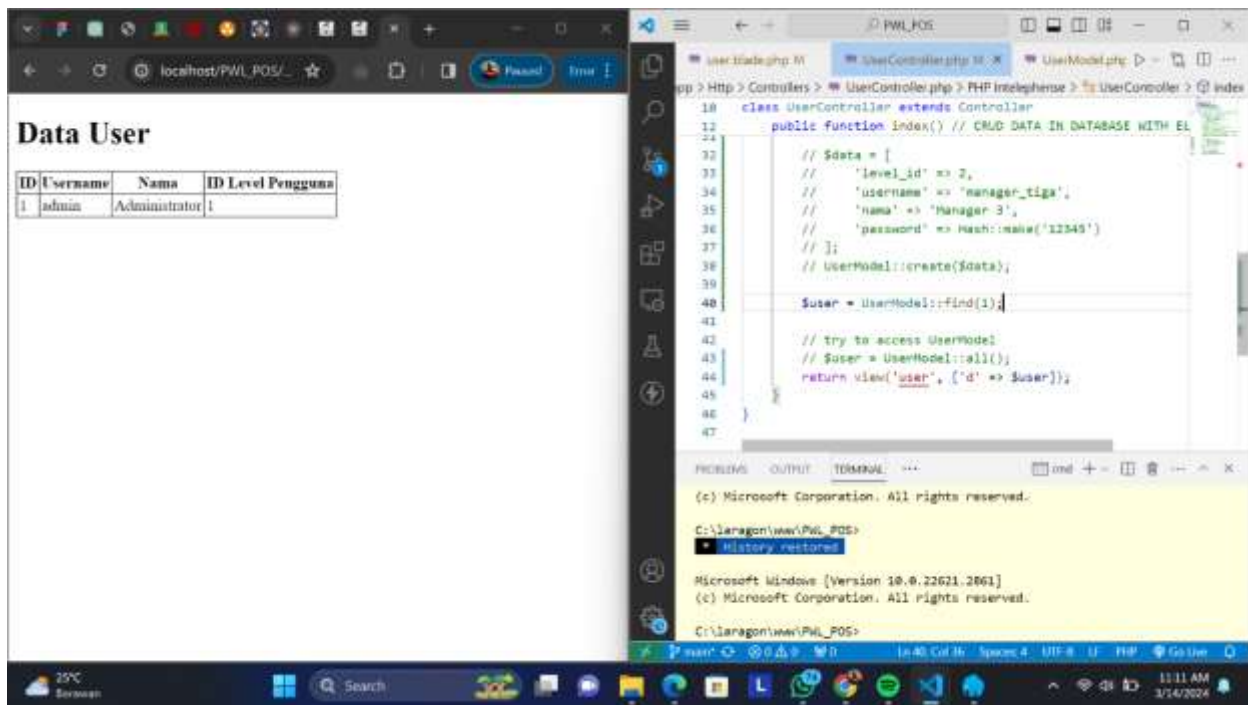


```
1 namespace App\Models;
2
3 use Illuminate\Database\Eloquent\Factories\HasFactory;
4 use Illuminate\Database\Eloquent\Model;
5
6 use function Laravel\Prompts\password;
7
8 class UserModel extends Model
9 {
10     use HasFactory;
11
12     protected $table = 'm_user';
13     protected $primaryKey = 'user_id';
14
15     // protected $guarded = ['password', 'role']; Kolom Password dan Role akan diabaikan dalam operasi database sebag
16     // protected $fillable = ['level_id', 'username', 'nama', 'password']; Keempat kolom di samping wajib diberikan vs
17     protected $fillable = ['level_id', 'username', 'nama'];
18 }
19
20
21
```



```
18 class UserController extends Controller
19 {
20     public function index() // CRUD: DATA IN DATABASE WITH ELOQUENT ORM
21     {
22         // // change a user data with Eloquent Model
23         // $data = [
24         //     'nama' => 'Pelanggan Pertama'
25         // ];
26         // UserModel::where('username', 'customer-1')->update($data);
27
28         // delete a user data with Eloquent Model
29         // UserModel::where('username', 'manager_tiga')->delete();
30
31         $data = [
32             'level_id' => 2,
33             'username' => 'manager_tiga',
34             'nama' => 'Manager 3',
35             'password' => Hash::make('12345')
36         ];
37         UserModel::create($data);
38     }
39 }

```



Mass Assignment adalah cara untuk menyimpan banyak data sekaligus ke dalam database menggunakan array atau objek. Dalam konteks Laravel, ini sering digunakan dalam kombinasi dengan metode create, update, atau fill pada model eloquent. Sebagai contoh, jika memiliki sebuah formulir dengan beberapa bidang input, kita bisa langsung meneruskan seluruh data input tersebut ke model untuk disimpan dalam database dengan mudah menggunakan mass assignment.

B. Praktikum 2.1 - Retrieving Single Models

The screenshot shows a web browser on the left displaying a table titled "Data User". The table has three columns: ID, Username, and ID Level Pengguna. The first row contains the values 1, admin, and Administrator. On the right, a code editor shows the PHP code for the `UserController.php` file. The code defines a `UserController` class that extends `Controller`. It includes a `index` method that queries the database for a user with `level_id = 1` and returns the result as a view named `'user'`. The terminal at the bottom shows the command prompt and the output of the application.

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

```
18 class UserController extends Controller
19 {
20     public function index() // CROD DATA IN DATABASE WITH EL
21     {
22         // $data = [
23         //     'level_id' => 2,
24         //     'username' => 'manager_tiga',
25         //     'nama' => 'Manager 3',
26         //     'password' => Hash::make('12345')
27         // ];
28         // UserModel::create($data);
29
30         $user = UserModel::where('level_id', 1)->first();
31
32         // Try to access UserModel
33         // $user = UserModel::all();
34         return view('user', ['d' => $user]);
35     }
36 }
```

The screenshot shows a web browser on the left displaying the same "Data User" table as in the previous image. On the right, the code editor shows the PHP code for the `user.blade.php` file. The code uses Blade templating to display the user data in a table. It includes a `@foreach` loop that iterates over the `$data` array and displays the user ID, username, and name in the table cells. The terminal at the bottom shows the command prompt and the output of the application.

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

```
18 <h2>Data User</h2>
19 <table border="1" cellpadding="2" cellspacing="0">
20 <tr>
21 <th>ID</th>
22 <th>Username</th>
23 <th>Nama</th>
24 <th>ID Level Pengguna</th>
25 </tr>
26 <tr>
27 <td>{{ $d->user_id }}</td>
28 <td>{{ $d->username }}</td>
29 <td>{{ $d->nama }}</td>
30 <td>{{ $d->level_id }}</td>
31 </tr>
32 </table>
```


localhost/PWL_POS/... **Passwd** **Time**

Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

```
18 class UserController extends Controller
19 {
20     public function index() // CRUD DATA IN DATABASE WITH EL
21     {
22         // $data = [
23         //     'level_id' => 2,
24         //     'username' => 'manager_tiga',
25         //     'nama' => 'Manager 3',
26         //     'password' => Hash::make('12345')
27         // ];
28         // UserModel::create($data);
29
30         $user = UserModel::where('level_id', 1)->first();
31
32         // try to access UserModel
33         // $user = UserModel::all();
34         return view('user', ['d' => $user]);
35     }
36 }
```

PROBLEMS OUTPUT TERMINAL

(c) Microsoft Corporation. All rights reserved.

C:\Jারণon\www\PWL_POS>

Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Jারণon\www\PWL_POS>

localhost/PWL_POS/... **Passwd** **Time**

Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

```
18 class UserController extends Controller
19 {
20     public function index() // CRUD DATA IN DATABASE WITH ELOQUE
21     {
22         // $data = [
23         //     'level_id' => 2,
24         //     'username' => 'manager_tiga',
25         //     'nama' => 'Manager 3',
26         //     'password' => Hash::make('12345')
27         // ];
28         // UserModel::create($data);
29
30         $user = UserModel::firstwhere('level_id', 1);
31
32         // try to access UserModel
33         // $user = UserModel::all();
34         return view('user', ['d' => $user]);
35     }
36 }
```

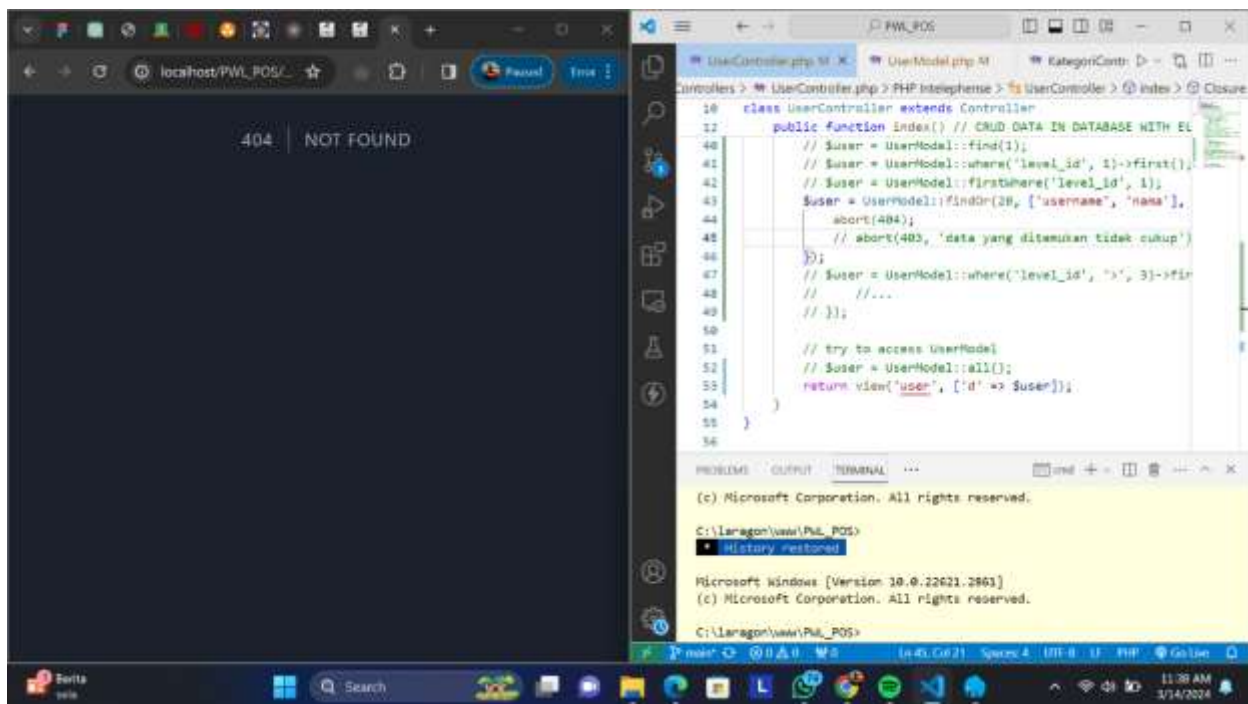
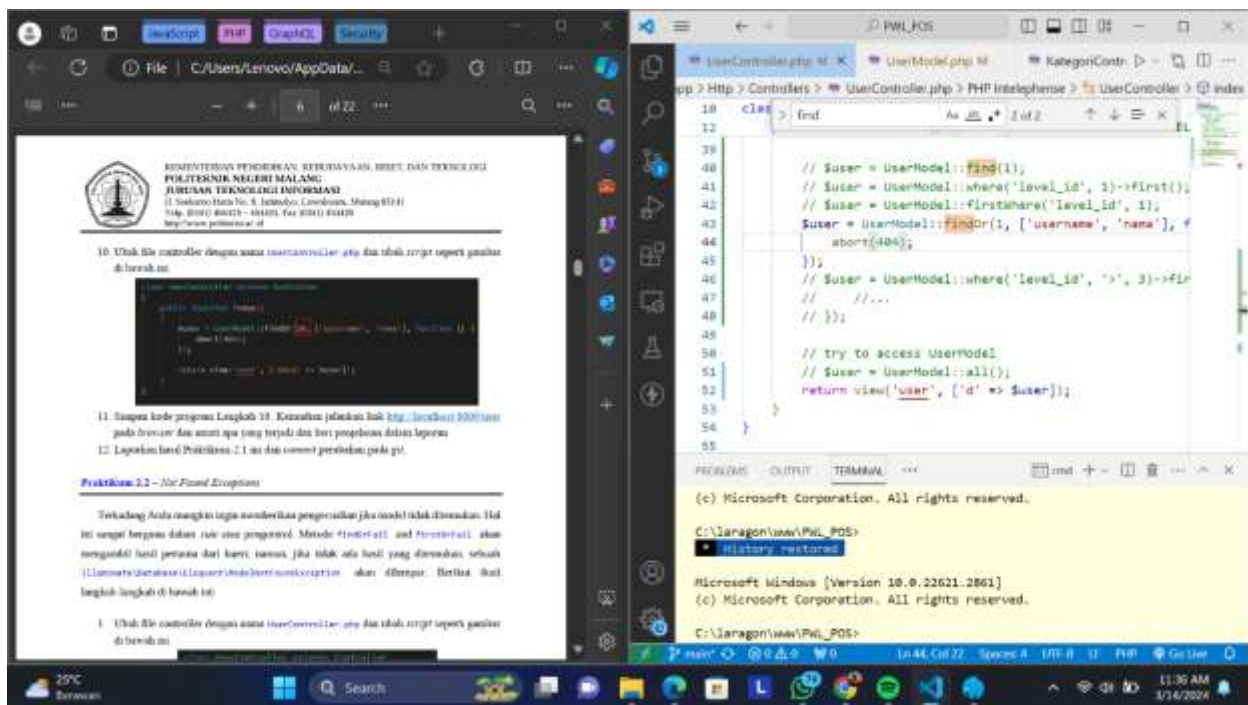
PROBLEMS OUTPUT TERMINAL

(c) Microsoft Corporation. All rights reserved.

C:\Jারণon\www\PWL_POS>

Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Jারণon\www\PWL_POS>



Terlihat browser menghasilkan output error dengan kode 404 yang artinya *not found* hal tersebut dikarenakan jumlah baris yang ada di database tidak mencapai jumlah baris yang diminta, yakni 20.

C. Praktikum 2.2 - Not Found Exceptions

The screenshot shows a web browser at `localhost/PWL_POS/` displaying a table titled "Data User". The table has three columns: "ID", "Username", and "ID Level Pengguna". The first row contains the values "1", "admin", and "Administrator".

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

The code editor on the right shows the `UserController.php` file. The `index` function is defined as follows:

```
10 class UserController extends Controller
11 {
12     public function index() // CRUD DATA IN DATABASE WITH EL
13     {
14         // $user = UserModel::find(1);
15         // $user = UserModel::where('level_id', 1)->first();
16         // $user = UserModel::firstWhere('level_id', 1);
17         // $user = UserModel::findOr(20, ['username', 'nana']
18         // abort(404);
19         // abort(403, 'data yang ditemukan tidak cukup');
20         // }
21         // $user = UserModel::where('level_id', '>', 3)->first();
22         // ...
23         // }
24         // $user = UserModel::findOrFail(1);
25
26         // try to access UserModel
27         // $user = UserModel::all();
28         return view('user', ['d' => $user]);
29     }
30 }
```

The terminal output shows the command `C:\laragon\user\PWL_POS>` and the response `History restored`.

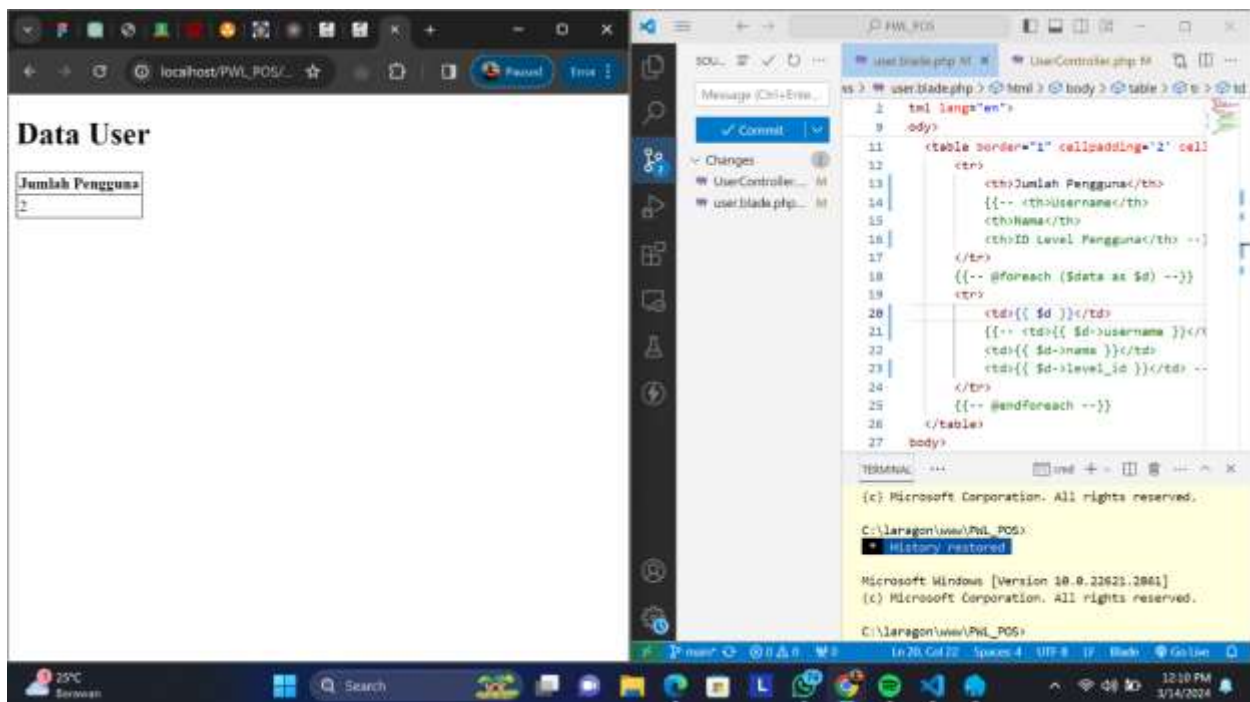
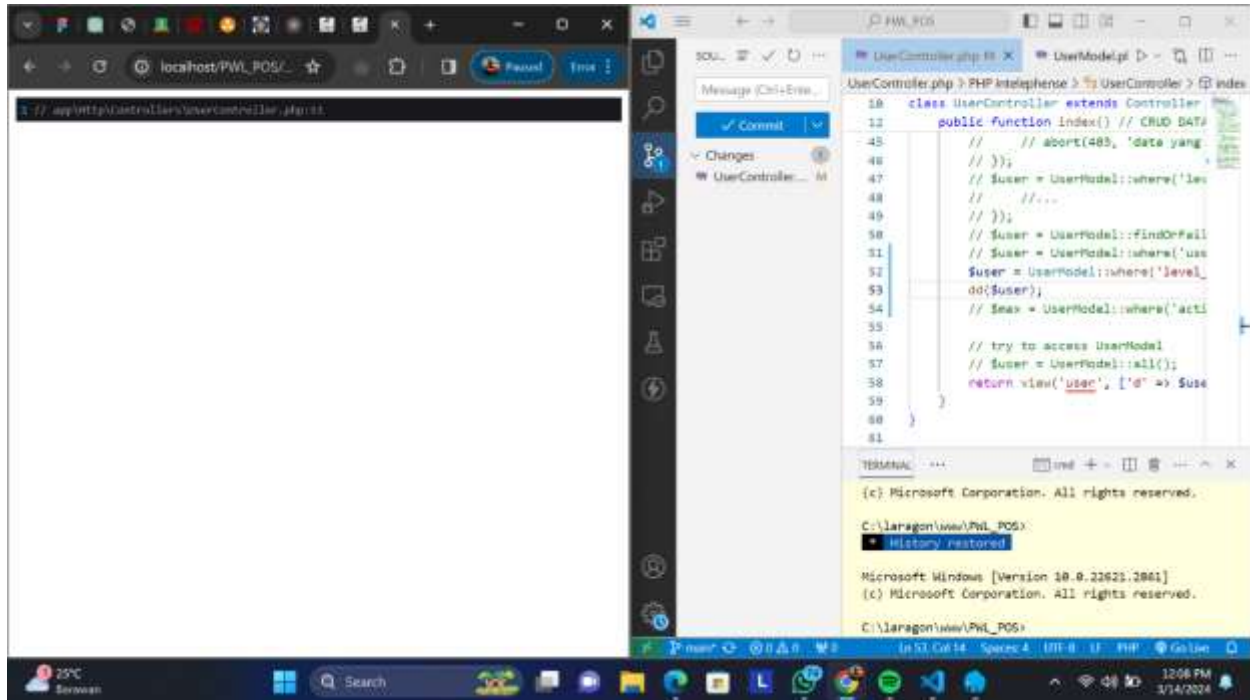
The screenshot shows the same web browser at `localhost/PWL_POS/` displaying a "404 | NOT FOUND" error. The code editor on the right shows the `UserController.php` file. The `index` function is defined as follows:

```
10 class UserController extends Controller
11 {
12     public function index() // CRUD DATA IN DATABASE WITH EL
13     {
14         // $user = UserModel::find(1);
15         // $user = UserModel::where('level_id', 1)->first();
16         // $user = UserModel::firstWhere('level_id', 1);
17         // $user = UserModel::findOr(20, ['username', 'nana']
18         // abort(404);
19         // abort(403, 'data yang ditemukan tidak cukup');
20         // }
21         // $user = UserModel::where('level_id', '>', 3)->first();
22         // ...
23         // }
24         // $user = UserModel::findOrFail(1);
25
26         // try to access UserModel
27         // $user = UserModel::all();
28         return view('user', ['d' => $user]);
29     }
30 }
```

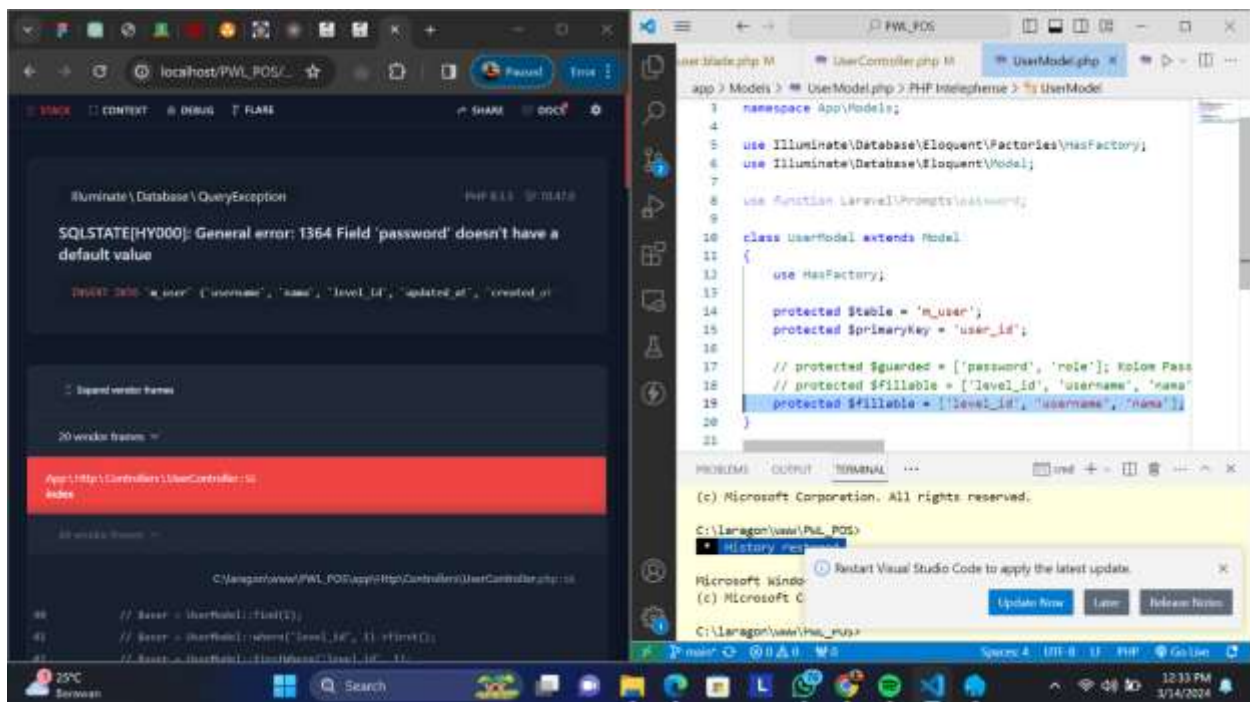
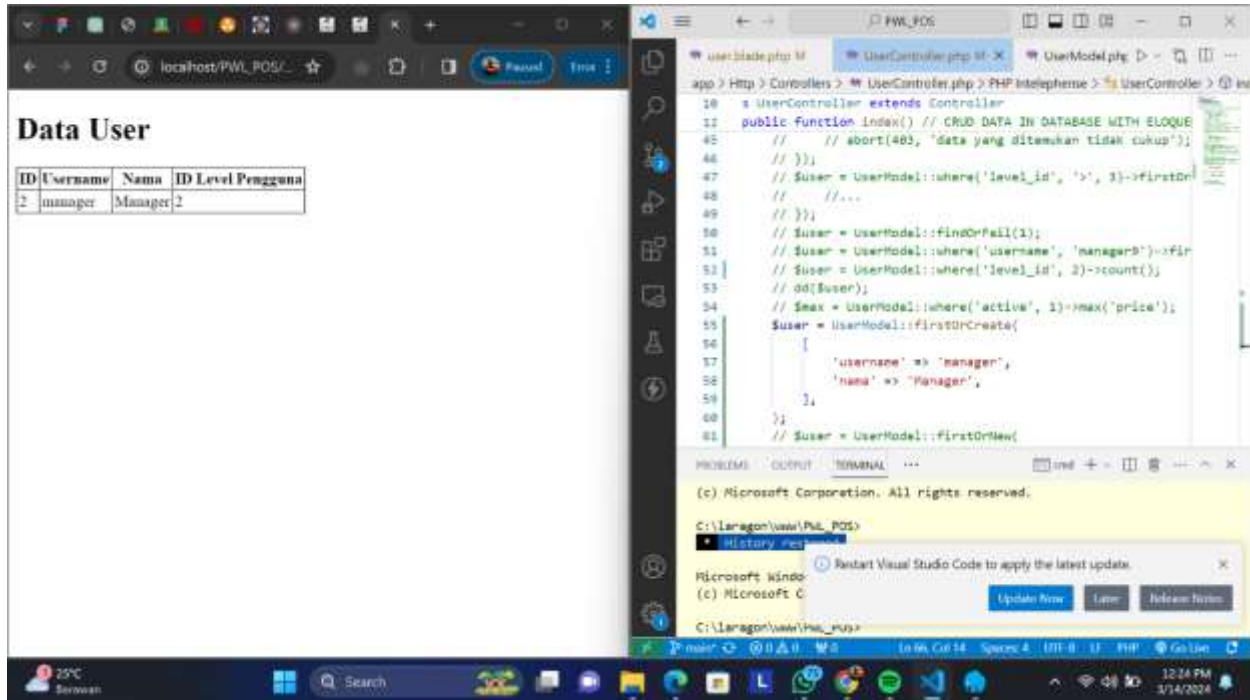
The terminal output shows the command `C:\laragon\user\PWL_POS>` and the response `History restored`.

Terlihat browser menghasilkan output error dengan kode 404 yang artinya *not found* hal tersebut dikarenakan tidak ada satupun baris yang memiliki value 'manager9' pada kolom 'username'.

D. Praktikum 2.3 - Retrieving Aggregates



E. Retrieving or Creating Models



localhost/PWL_POS/

Data User

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

```
10 class UserController extends Controller
11 public function index() // CRUD DATA IN DATABASE WITH EL
46 // }
47 // $user = UserModel::where('level_id', '=', 3)->fir
48 // }
49 // }
50 // $user = UserModel::findOrFail(1);
51 // $user = UserModel::where('username', 'manager9')-
52 // $user = UserModel::where('level_id', 2)->count();
53 // dd($user);
54 // $user = UserModel::where('active', 1)->max('price'
55 $user = UserModel::firstOrCreate(
56 [
57     'username' => 'manager22',
58     'name' => 'Manager Dua Dua',
59     'password' => Hash::make('12345'),
60     'level_id' => 2,
61 ],
62 );
```

PROBLEMS OUTPUT TERMINAL

(c) Microsoft Corporation. All rights reserved.

C:\laragon\www\PWL_POS>

Microsoft Windows
(c) Microsoft C

C:\laragon\www\PWL_POS>

Restart Visual Studio Code to apply the latest update.

Update Now Later Release Notes

25°C Serasan

12:43 PM 3/14/2024

localhost/PWL_POS/

Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

```
55 // $user = UserModel::firstOrCreate(
56 [
57     'username' => 'manager22',
58     'name' => 'Manager Dua Dua',
59     'password' => Hash::make('12345'),
60     'level_id' => 2,
61 ],
62 );
63 $user = UserModel::firstOrCreate(
64 [
65     'username' => 'manager33',
66     'name' => 'Manager Tiga Tiga',
67     'password' => Hash::make('12345'),
68     'level_id' => 2,
69 ],
70 );
71 );
```

PROBLEMS OUTPUT TERMINAL

(c) Microsoft Corporation. All rights reserved.

C:\laragon\www\PWL_POS>

Microsoft Windows
(c) Microsoft C

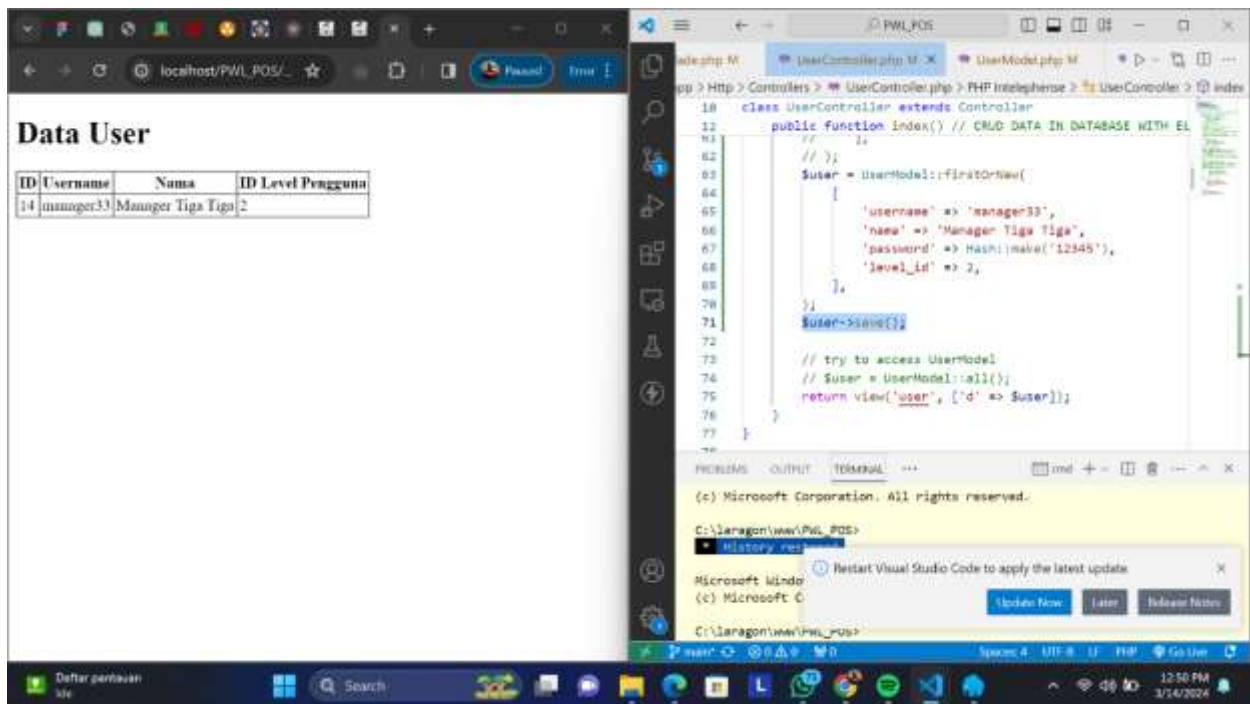
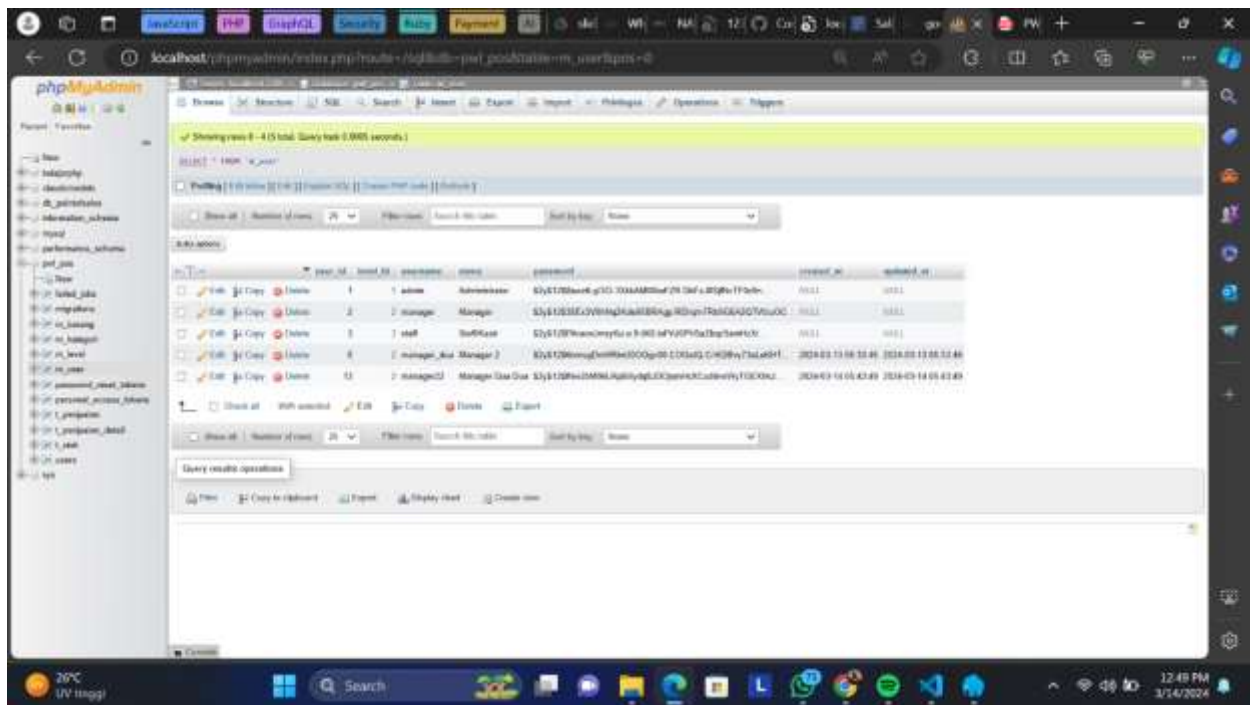
C:\laragon\www\PWL_POS>

Restart Visual Studio Code to apply the latest update.

Update Now Later Release Notes

25°C Serasan

12:46 PM 3/14/2024



F. Praktikum 2.5 - Attributes Changes

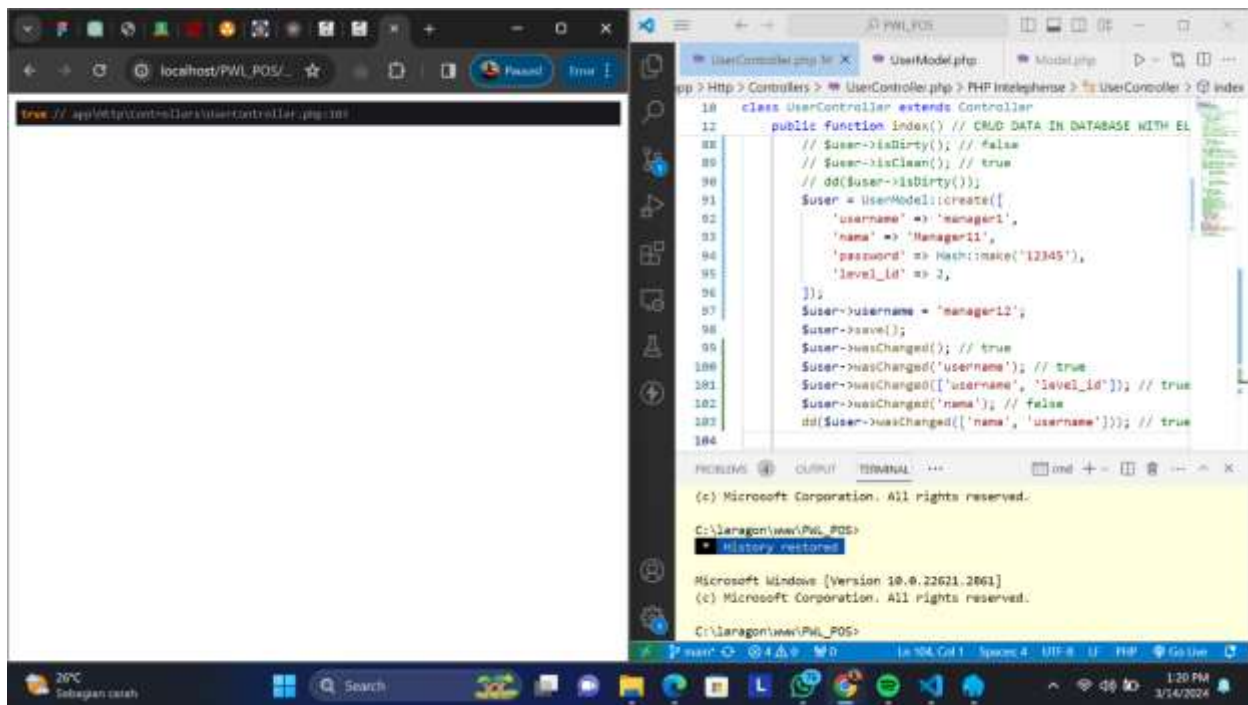
The screenshot shows a web browser at `localhost/PWL_POS/` displaying a table titled "Data User". The table has three columns: "ID", "Username", and "ID Level Pengguna". It contains one row with the values "14", "manager33", and "Maanger Tiga Tiga" (note the typo). To the right, the VS Code editor shows the `UserController.php` file. The `index` method is highlighted, showing a query to fetch data from the database and a call to `$user->save()`. The terminal at the bottom shows the command `php artisan serve` being executed.

ID	Username	ID Level Pengguna
14	manager33	Maanger Tiga Tiga

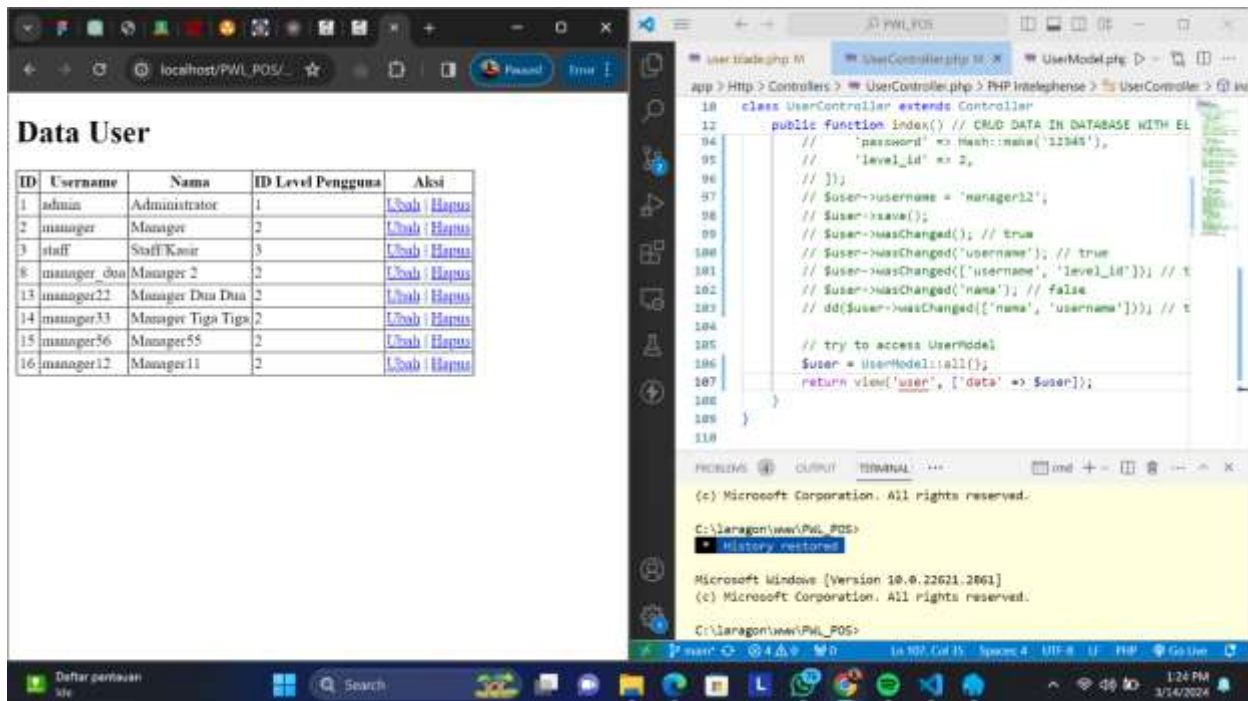
```
18 class UserController extends Controller
19 {
20     public function index() // CRUD DATA IN DATABASE WITH EL
21     {
22         // try to access UserModel
23         // $user = UserModel::all();
24         $user = UserModel::find(14);
25         return view('user', ['d' => $user]);
26     }
27 }
```

The screenshot shows the same web browser, but the URL is now `localhost/PWL_POS/userController.php/14`. The browser displays a "false" message. In VS Code, the `UserController.php` file is open, showing the `update` method. This method updates the user's name and username, saves the changes, and then redirects to the `index` method. The terminal shows the command `php artisan serve` being executed.

```
78 $user->username = 'manager56';
79 $user->isDirty();
80 $user->isDirty('username');
81 $user->isDirty('name');
82 $user->isDirty(['name', 'username']);
83
84 $user->isClean();
85 $user->isClean('username');
86 $user->isClean('name');
87 $user->isClean(['name', 'username']);
88
89 $user->save();
90
91 $user->isDirty();
92 $user->isClean();
93 dd($user->isDirty());
94 }
```

G. Praktikum 2.6 - Create, Read, Update, Delete (CRUD)



H. Praktikum 2.7 - Relationships

