

Jobsheet 04: Model dan Eloquent Orm

Dosen Pengampu Mata Kuliah :

M. Hasyim Ratsanjani, S.Kom., M.Kom.



Dibuat Oleh :

Daffa Yudisa Akbar : TI-2H / 10 : 2241720008

Politeknik Negeri Malang

Jurusan Teknologi Informasi

D-IV Teknik Informatika

2024

A. Praktikum 1 - \$fillable:

No.	Keterangan Langkah dan Gambar
1.	<p>Buka file model dengan nama UserModel.php dan tambahkan \$fillable seperti gambar di bawah ini</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 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', 'name', 'password']; 16 } </pre>
2.	<p>Buka file controller dengan nama UserController.php dan ubah script untuk menambahkan data baru</p>  <pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use App\Models\UserModel; 6 use Illuminate\Http\Request; 7 use Illuminate\Support\Facades\Hash; 8 9 class UserController extends Controller 10 { 11 public function index() 12 { 13 // \$data = [14 // 'username' => 'customer-1', 15 // 'name' => 'Pelanggan', 16 // 'password' => Hash::make('12345'), 17 // 'level_id' => 4 18 //]; 19 // UserModel::insert(\$data); 20 21 // \$data = [22 // 'name' => 'Pelanggan Pertama', 23 //]; 24 // UserModel::where('username', 'customer-1')->update(\$data); 25 26 \$data = [27 'level_id' => 2, 28 'username' => 'manager_dua', 29 'name' => 'Manager 2', 30 'password' => Hash::make('12345') 31]; 32 UserModel::create(\$data); 33 34 \$user = UserModel::all(); 35 return view('user', ['data' => \$user]); 36 } 37 } </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>

localhost / 127.0.0.1 | phpMyAd x Data User x +

localhost/PWL_POS/public/user

Gmail YouTube Siakad Polinema DAFFA YUDISA AKB...

Data User

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

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

```

UserModel.php M x UserController.php M
app > Models > UserModel.php > ...
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', 'name', 'password'];
16     protected $fillable = ['level_id', 'username', 'name'];
17
18 }
19

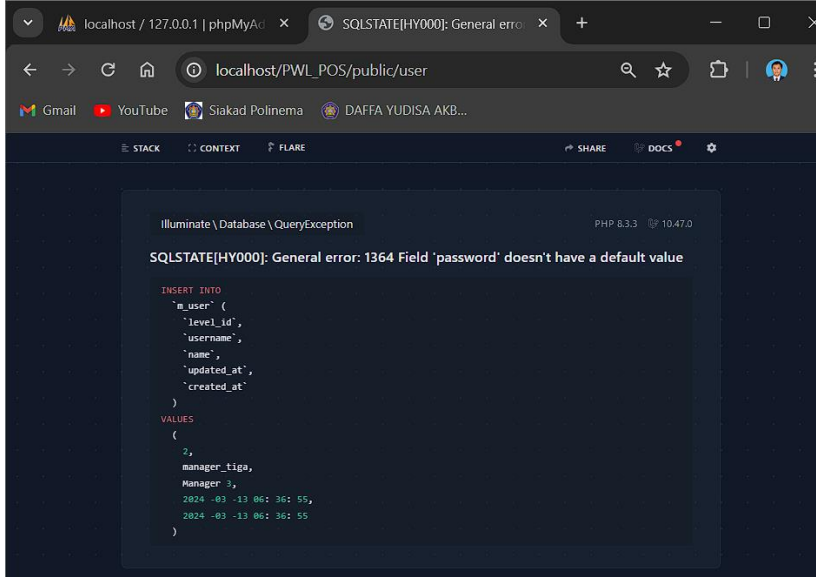
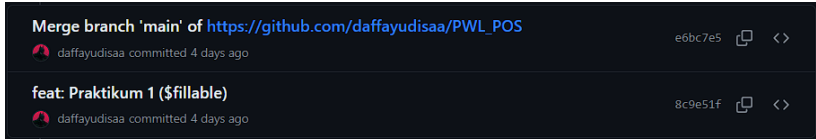
```

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

```


UserModel.php M x UserController.php M x
app > Http > Controllers > UserController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         // $data = [
14         //     'username' => 'customer-1',
15         //     'name' => 'Pelanggan',
16         //     'password' => Hash::make('12345'),
17         //     'level_id' => 4
18         // ];
19         // UserModel::insert($data);
20
21         // $data = [
22         //     'name' => 'Pelanggan Pertama',
23         // ];
24         // UserModel::where('username', 'customer-1')->update($data);
25
26         $data = [
27             'level_id' => 2,
28             'username' => 'manager_tiga',
29             'name' => 'Manager 3',
30             'password' => Hash::make('12345')
31         ];
32         UserModel::create($data);
33
34         $user = UserModel::all();
35         return view('user', ['data' => $user]);
36     }
37 }

```

6.	<p>Simpan kode program Langkah 4 dan 5. Kemudian jalankan pada browser dan amati apa yang terjadi</p> 
7.	<p>Laporkan hasil Praktikum-1 ini dan commit perubahan pada git.</p> 

- 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

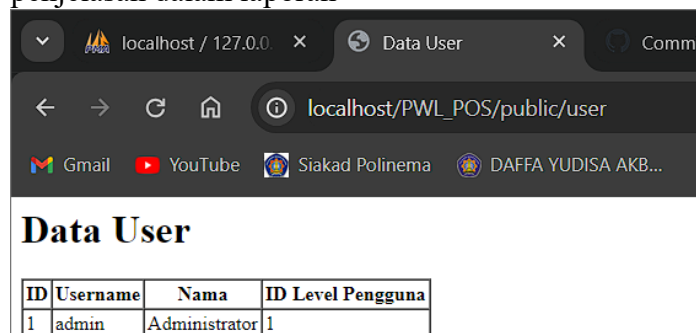
No	Keterangan Langkah dan Gambar
1.	<p>Buka file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> 
2.	<p>Buka file view dengan nama user.blade.php dan ubah script seperti gambar di bawah ini</p>

```

UserModel.php  UserController.php M  user.blade.php M x
resources > views > user.blade.php > html > html
1  <html lang="en">
2  <html>
3  <body>
4  <h1>Data User</h1>
5  <table border="1" cellpadding="2" cellspacing="0">
6  <tr>
7  <th>ID</th>
8  <th>Username</th>
9  <th>Nama</th>
10 <th>ID Level Pengguna</th>
11 </tr>
12 {{-- @foreach($data as $d) --}}
13 <tr>
14 <td>{{ $data->user_id }}</td>
15 <td>{{ $data->username }}</td>
16 <td>{{ $data->name }}</td>
17 <td>{{ $data->level_id }}</td>
18 </tr>
19 {{-- @endforeach --}}
20 </table>
21 </body>
22 </html>

```

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



The screenshot shows a web browser window with the address bar displaying localhost/PWL_POS/public/user. The page title is "Data User". Below the title, there is a table with the following data:

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

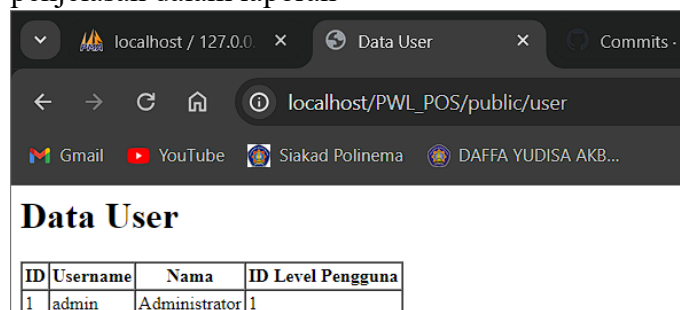
4. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```

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

```

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



The screenshot shows a web browser window with the address bar displaying localhost/PWL_POS/public/user. The page title is "Data User". Below the title, there is a table with the following data:

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

6.

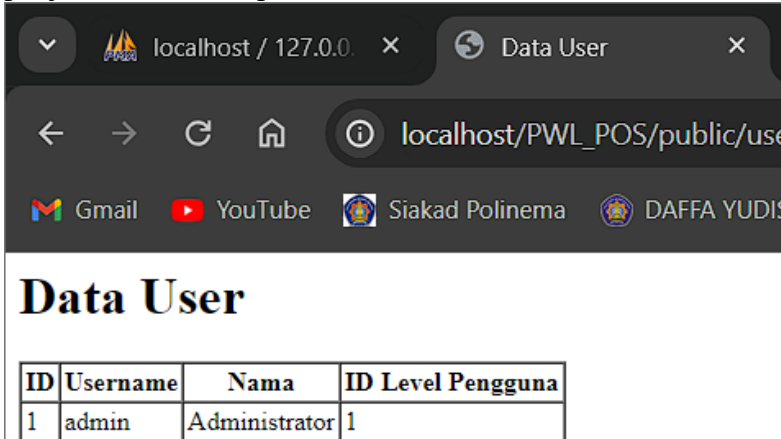
Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```

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

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



ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

8.

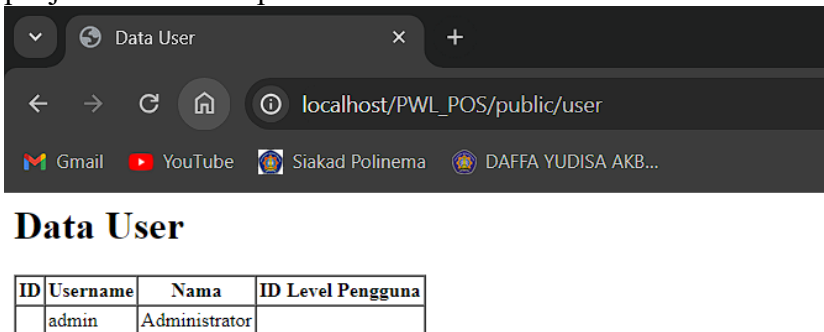
Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```

                public function index()
                {
                    $user = UserModel::findOr(1, ['username', 'name'], function(){
                        abort(404);
                    });
                    return view('user', ['data' => $user]);
                }
            
```

9.

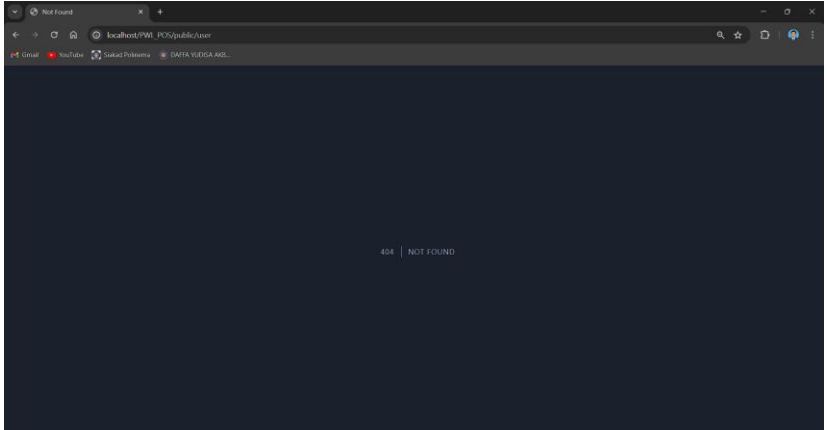
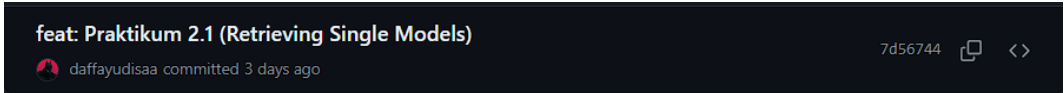
Simpan kode program Langkah 8. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



ID	Username	Nama	ID Level Pengguna
	admin	Administrator	

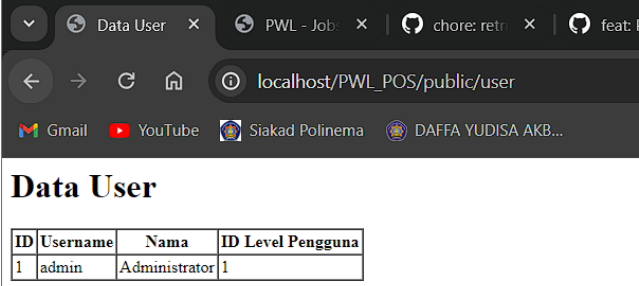
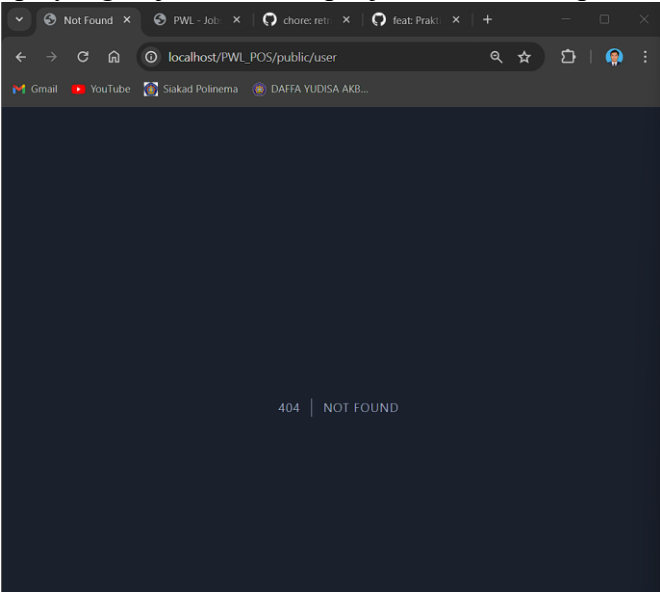

10.

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

	<pre>class UserController extends Controller { public function index() { \$user = UserModel::findOr(20, ['username', 'name'], function(){ abort(404); }); return view('user', ['data' => \$user]); } }</pre>	
11.	<p>Simpan kode program Langkah 10. Kemudian jalankan link http://localhost:8000/user pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> 	
12.	<p>Laporkan hasil Praktikum-2.1 ini dan commit perubahan pada git.</p> 	

C. Praktikum 2.2 - Not Found Exceptions

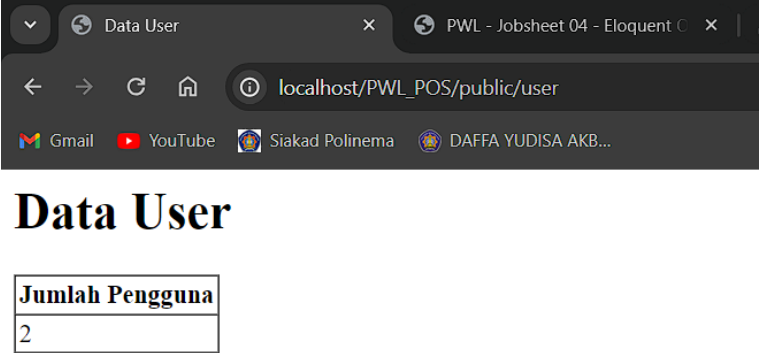

No	Keterangan Langkah dan Gambar
1.	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> <pre>class UserController extends Controller { public function index() { \$user = UserModel::findOrFail(1); return view('user', ['data' => \$user]); } }</pre>
2.	<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>class UserController extends Controller { public function index() { \$user = UserModel::where('username', 'manager9')->firstOrFail(); return view('user', ['data' => \$user]); } }</pre>
4.	<p>Simpan kode program Langkah 3. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> 
5.	<p>Laporkan hasil Praktikum-2.2 ini dan commit perubahan pada git.</p> 

D. Praktikum 2.3 - Retrieving Aggregates

No	Keterangan Langkah dan Gambar
1.	Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

	<pre> class UserController extends Controller { public function index() { \$user = UserModel::where('level_id', 2)→ count(); dd(\$user); return view('user', ['data' ⇒ \$user]); } } </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>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> 

	
4.	<p>Laporkan hasil Praktikum-2.3 ini dan commit perubahan pada git.</p> 

E. Praktikum 2.4 – Retreiving or Creating Models

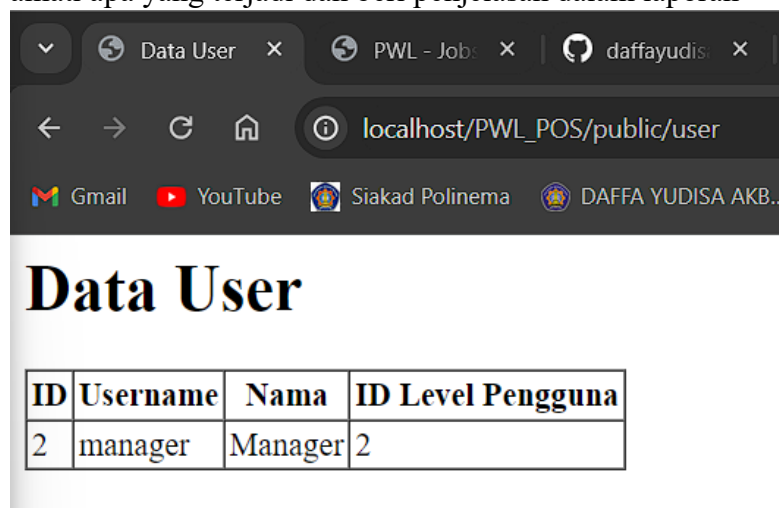
No	Keterangan Langkah dan Gambar
1.	<p>Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini</p> 
2.	<p>Ubah kembali file view dengan nama user.blade.php dan ubah script seperti gambar di bawah ini</p>

```

UserModel.php x UserController.php M user.blade.php M x
resources > views > user.blade.php > html > html
1 <html lang="en">
2 <html>
3   <head>
4     <title>Data User</title>
5   </head>
6   <body>
7     <h1>Data User</h1>
8     <table border="1" cellpadding="2" cellspacing="0">
9       <tr>
10        {{-- <th>Jumlah Pengguna</th> --}}
11        <th>ID</th>
12        <th>Username</th>
13        <th>Nama</th>
14        <th>ID Level Pengguna</th>
15      </tr>
16      {{-- @foreach($data as $d) --}}
17      <tr>
18        {{-- <td>{{ $data }}</td> --}}
19        <td>{{ $data->user_id }}</td>
20        <td>{{ $data->username }}</td>
21        <td>{{ $data->name }}</td>
22        <td>{{ $data->level_id }}</td>
23      </tr>
24      {{-- @endforeach --}}
25    </table>
26  </body>
27 </html>

```

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



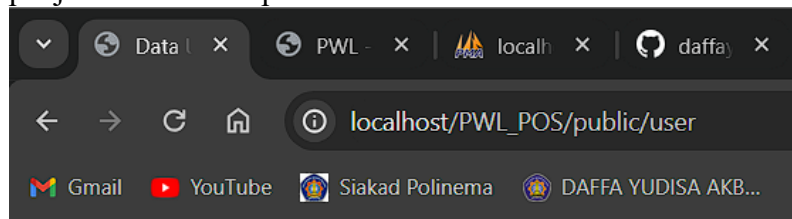
4. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```

UserModel.php | UserController.php M | user.blade.php M
app > Http > Controllers > UserController.php > PHP Intelephense > UserController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Database\Eloquent\ModelNotFoundException;
9
10 class UserController extends Controller
11 {
12     public function index()
13     {
14         $user = UserModel::firstOrCreate(
15             [
16                 'username' => 'manager22',
17                 'name' => 'Manager Dua Dua',
18                 'password' => Hash::make('12345'),
19                 'level_id' => 2
20             ],
21         );
22         return view('user', ['data' => $user]);

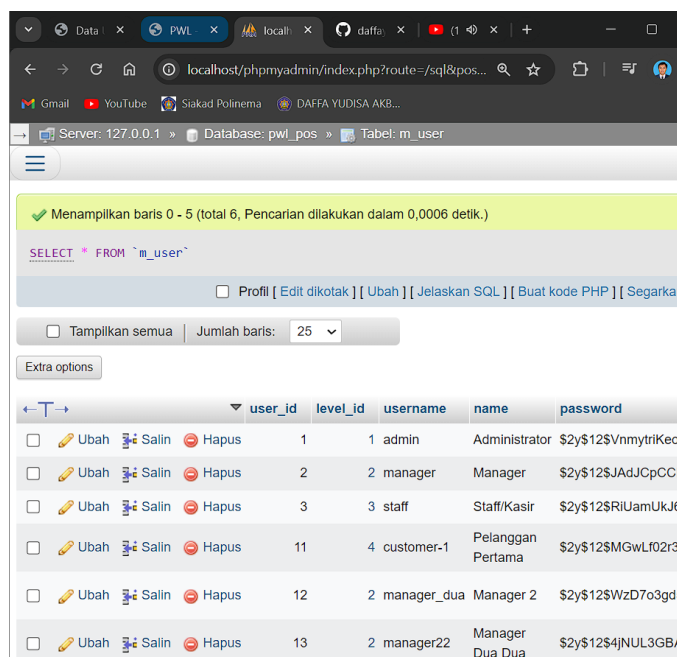
```

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



Data User

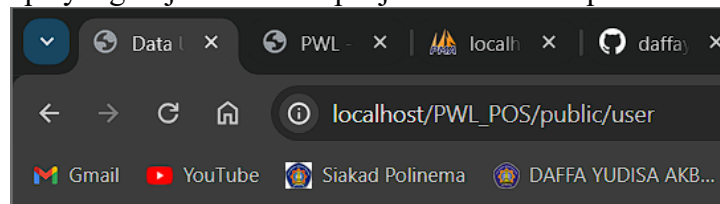
ID	Username	Nama	ID Level Pengguna
13	manager22	Manager Dua Dua	2



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

```
UserModel.php M x UserController.php M x user.blade.php M
app > Http > Controllers > UserController.php > PHP Intelephense > UserController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Database\Eloquent\ModelNotFoundException;
9
10 class UserController extends Controller
11 {
12     public function index()
13     {
14         $user = UserModel::firstOrCreate(
15             [
16                 'username' => 'manager',
17                 'name' => 'manager',
18             ],
19             []
20         );
21         return view('user', ['data' => $user]);
22     }
23 }
```

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



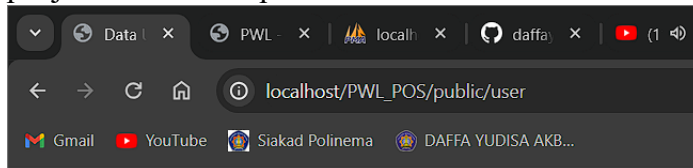
Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

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

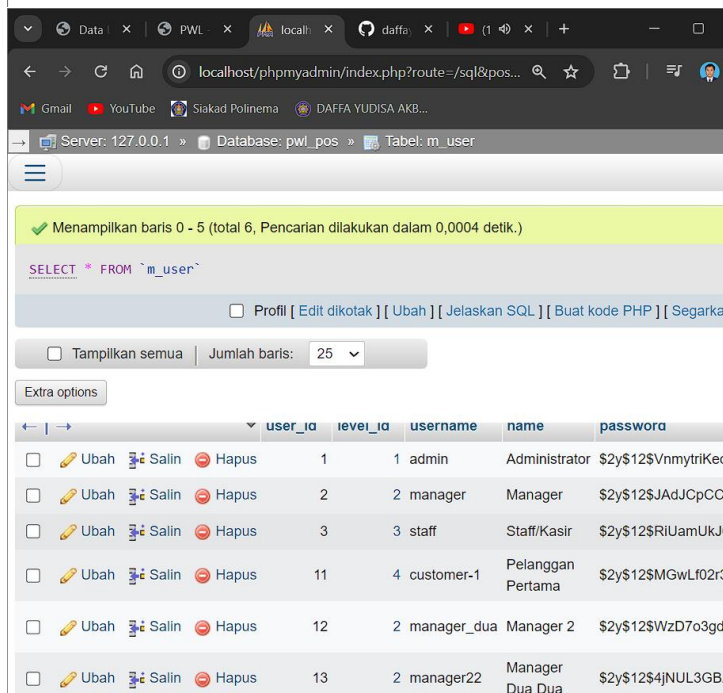
```
UserModel.php M x UserController.php M x user.blade.php M
app > Http > Controllers > UserController.php > PHP Intelephense > UserController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Database\Eloquent\ModelNotFoundException;
9
10 class UserController extends Controller
11 {
12     public function index()
13     {
14         $user = UserModel::firstOrCreate(
15             [
16                 'username' => 'manager33',
17                 'name' => 'Manager Tiga Tiga',
18                 'password' => Hash::make('12345'),
19                 'level_id' => 2
20             ],
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



Data User

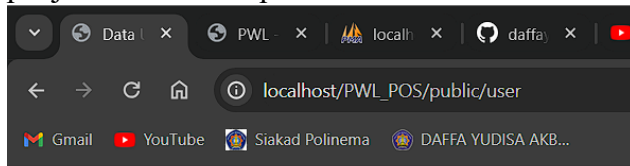
ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2



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

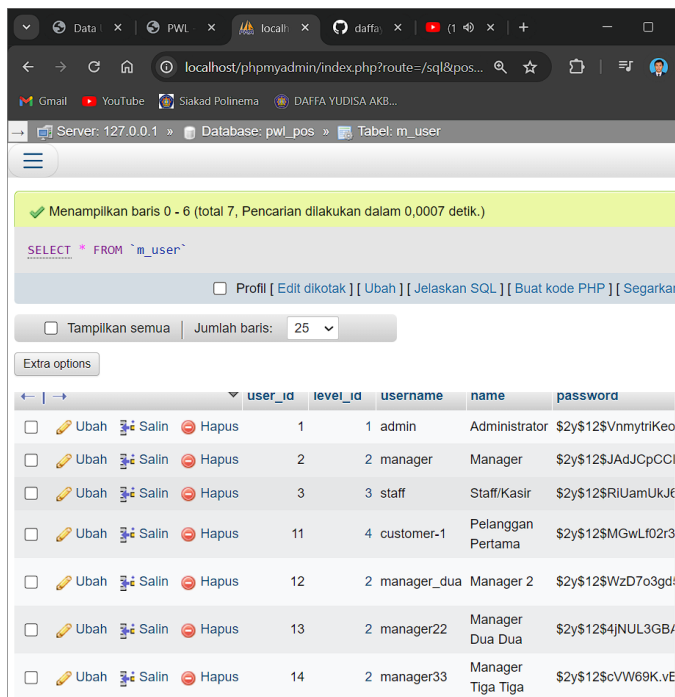
```
UserModel.php M  UserController.php M  user.blade.php M
app > Http > Controllers > UserController.php > PHP Intelephense > UserController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Hash;
8  use Illuminate\Database\Eloquent\ModelNotFoundException;
9
10 class UserController extends Controller
11 {
12     public function index()
13     {
14         $user = UserModel::firstOrCreate(
15             [
16                 'username' => 'manager33',
17                 'name' => 'Manager Tiga Tiga',
18                 'password' => Hash::make('12345'),
19                 'level_id' => 2
20             ],
21         );
22         $user->save();
23
24         return view('user', ['data' => $user]);
25     }
26 }
```

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
14	manager33	Manager Tiga Tiga	2



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

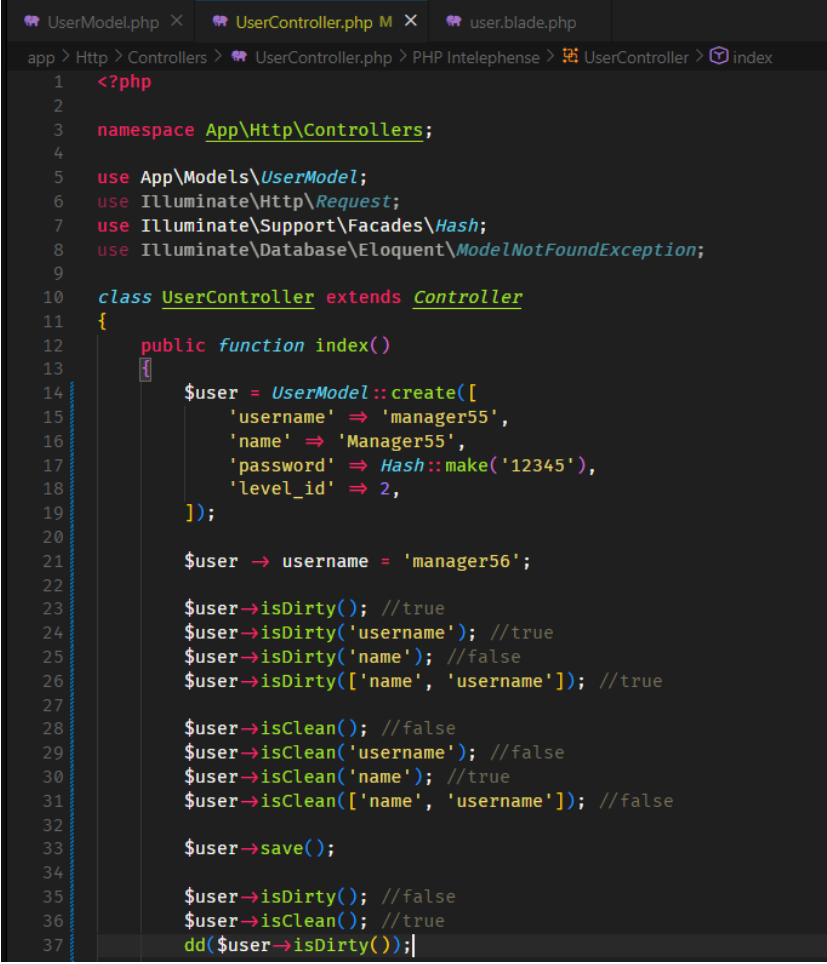
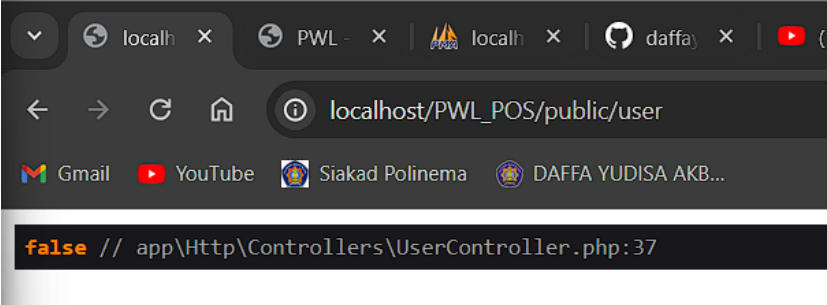
feat: Praktikum 2.4 (Retreiving or Creating Models)

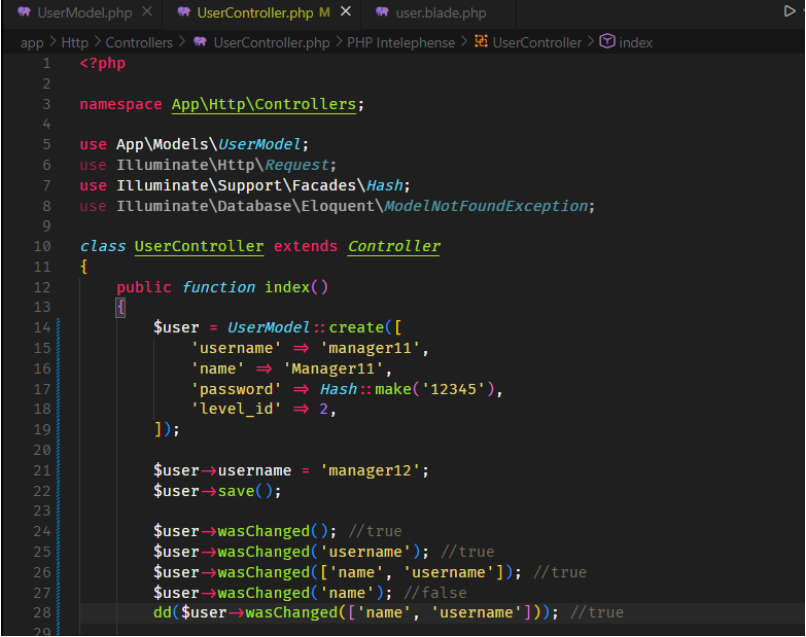
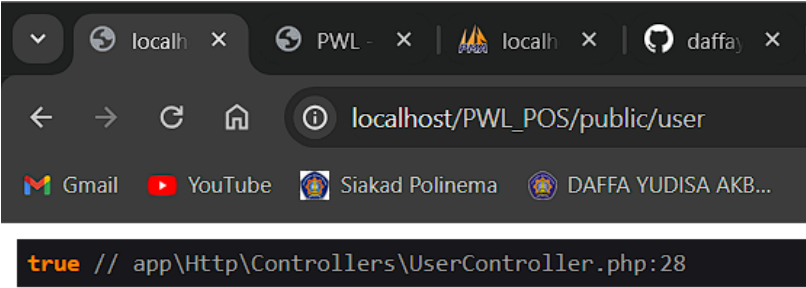

55a58d5

daffayudisaa committed 2 days ago

F. Praktikum 2.5 - Attribute Changes

No	Keterangan Langkah dan Gambar
1.	Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

	 <pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use App\Models\UserModel; 6 use Illuminate\Http\Request; 7 use Illuminate\Support\Facades\Hash; 8 use Illuminate\Database\Eloquent\ModelNotFoundException; 9 10 class UserController extends Controller 11 { 12 public function index() 13 { 14 \$user = UserModel::create([15 'username' => 'manager55', 16 'name' => '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('name'); //false 26 \$user->isDirty(['name', 'username']); //true 27 28 \$user->isClean(); //false 29 \$user->isClean('username'); //false 30 \$user->isClean('name'); //true 31 \$user->isClean(['name', '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> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use App\Models\UserModel; 6 use Illuminate\Http\Request; 7 use Illuminate\Support\Facades\Hash; 8 use Illuminate\Database\Eloquent\ModelNotFoundException; 9 10 class UserController extends Controller 11 { 12 public function index() 13 { 14 \$user = UserModel::create([15 'username' => 'manager11', 16 'name' => 'Manager11', 17 'password' => Hash::make('12345'), 18 'level_id' => 2, 19]); 20 21 \$user->username = 'manager12'; 22 \$user->save(); 23 24 \$user->wasChanged(); //true 25 \$user->wasChanged('username'); //true 26 \$user->wasChanged(['name', 'username']); //true 27 \$user->wasChanged('name'); //false 28 dd(\$user->wasChanged(['name', 'username'])); //true 29 } 30 } </pre>
4.	<p>Simpan kode program Langkah 3. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan</p> 
5.	<p>Laporkan hasil Praktikum-2.5 ini dan commit perubahan pada git.</p> 

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

No	Keterangan Langkah dan Gambar
1.	Buka file view pada user.blade.php dan buat scripnya menjadi seperti di bawah ini

```

resources > views > user.blade.php > html > body > table > tr > td > a
1 <html>
2 <head>
3 <title>Data User</title>
4 </head>
5 <body>
6 <h1>Data User</h1>
7 <a href="{{url ('/user/tambah')}}">+ Tambah User</a>
8 <table border="1" cellpadding="2" cellspacing="0">
9 <tr>
10 <th>Jumlah Pengguna</th> </tr>
11 <td>ID</td>
12 <td>Username</td>
13 <td>Nama</td>
14 <td>ID Level Pengguna</td>
15 <td>Aksi</td>
16 </tr>
17 @foreach($data as $d)
18 <tr>
19 <td>{{ $d->user_id }}</td>
20 <td>{{ $d->username }}</td>
21 <td>{{ $d->name }}</td>
22 <td>{{ $d->level_id }}</td>
23 <td>
24 <a href="{{ url('/user/ubah/' . $d->user_id) }}">Ubah</a> |
25 <a href="{{ url('/user/hapus/' . $d->user_id) }}">Hapus</a></td>
26 </tr>
27 @endforeach
28 </table>
29 </body>
30 </html>

```

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

```

UserModel.php  UserController.php M  user.blade.php M
app > Http > Controllers > UserController.php > PHP Intelephense > UserController > index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\UserModel;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Database\Eloquent\ModelNotFoundException;
9
10 class UserController extends Controller
11 {
12     public function index()
13     {
14
15         $user = UserModel::all();
16
17         return view('user', ['data' => $user]);
18     }
19 }

```

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

Browser view showing the application running on localhost:127.0.0.1. The page title is "Data User". There is a link "+ Tambah User". Below it is a table with columns: ID, Username, Nama, ID Level Pengguna, and Aksi. The table contains 16 rows of user data. Each row has links for "Ubah" (Update) and "Hapus" (Delete) actions.

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
11	customer-1	Pelanggan Pertama	4	Ubah Hapus
12	manager_dua	Manager 2	2	Ubah Hapus
13	manager22	Manager Dua Dua	2	Ubah Hapus
14	manager33	Manager Tiga Tiga	2	Ubah Hapus
15	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

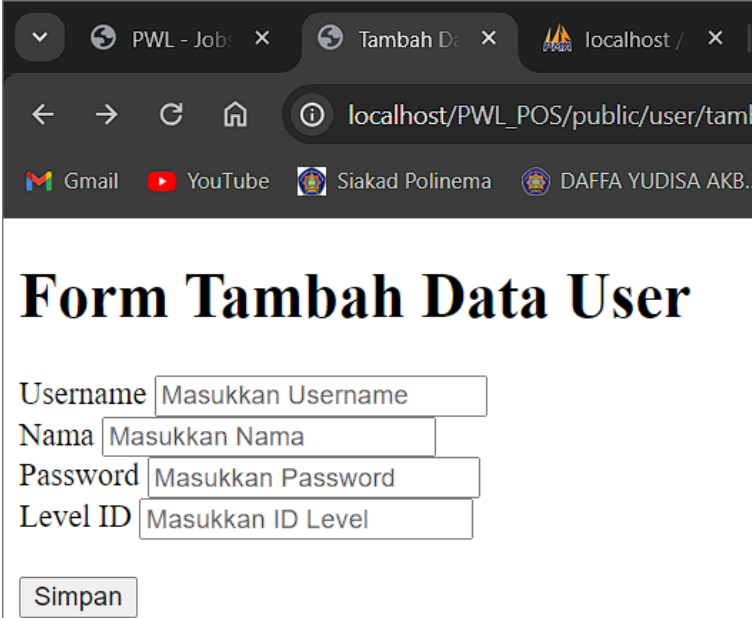
```
UserController.php M user_tambah.blade.php U x web.php M user.blade.php M
resources > views > user_tambah.blade.php > html > head
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Tambah Data User</title>
5 </head>
6 <body>
7 <h1>Form Tambah Data User</h1>
8 <form method="post" action="{{ url('user/tambah_simpan')}}">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

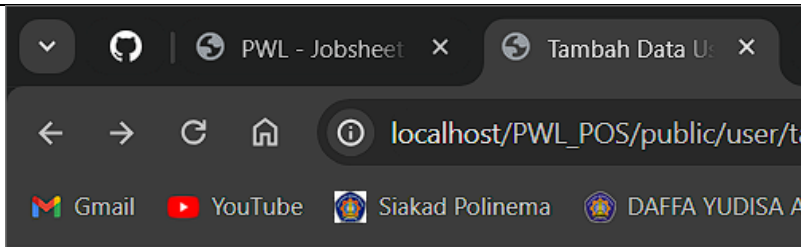
5. Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

```
27
28 Route::get('/user/tambah', [UserController::class, 'tambah']);
29
```

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

```
UserController.php M x user_tambah.blade.php U web.php M
app > Http > Controllers > UserController.php > PHP Intelephense > UserController
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\UserModel;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Database\Eloquent\ModelNotFoundException;
9
10 class UserController extends Controller
11 {
12     public function index()
13     {
14
15         $user = UserModel::all();
16
17         return view('user', ['data' => $user]);
18     }
19
20
21     public function tambah()
22     {
23         return view('user_tambah');
```

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> 
8.	<p>Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini</p> <pre data-bbox="325 981 1366 1037">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 dan diletakan di bawah method tambah seperti gambar di bawah ini</p> <pre data-bbox="325 1200 1177 1563"> public function tambah_simpan(Request \$request) { UserModel::create(['username' => \$request->username, 'name' => \$request->name, 'password' => Hash::make(\$request->password), 'level_id' => \$request->level_id]); return redirect('/user'); } </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>



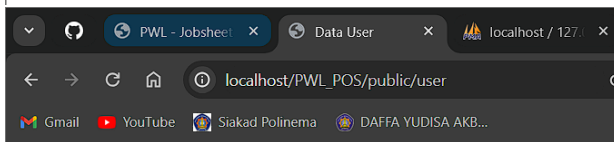
Form Tambah Data User

Username

Nama

Password

Level ID



Data User

[+ Tambah User](#)

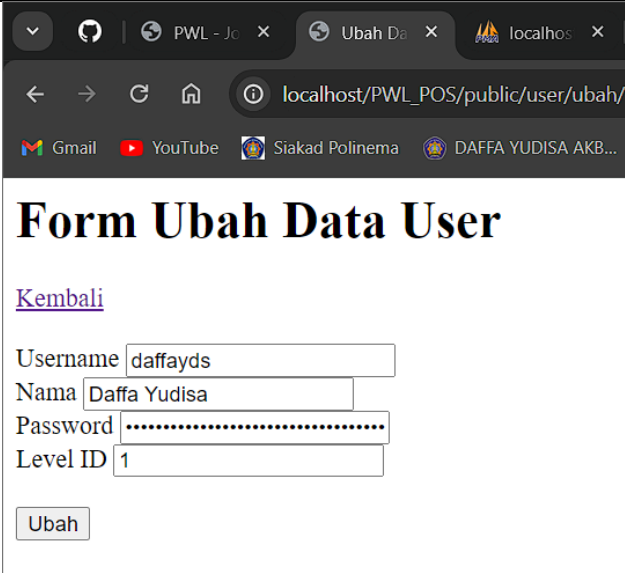
ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
11	customer-1	Pelanggan Pertama	4	Ubah Hapus
12	manager_dua	Manager 2	2	Ubah Hapus
13	manager22	Manager Dua Dua	2	Ubah Hapus
14	manager33	Manager Tiga Tiga	2	Ubah Hapus
15	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus
17	daffayds	Daffa Yudisa	1	Ubah Hapus

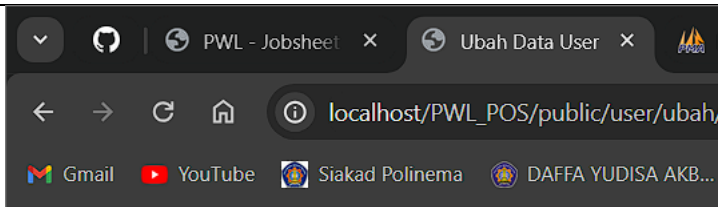
- Langkah berikutnya membuat update atau ubah data user dengan cara bikin file baru pada view dengan nama user_ubah.blade.php dan buat scriptnya menjadi seperti di bawah ini

	<pre> resources > views > user_ubah.blade.php > body > form > input.btn.btn-succes 1 <head> 2 <title>Ubah Data User</title> 3 </head> 4 <body> 5 <h1>Form Ubah Data User</h1> 6 Kembali 7

 8 9 <form method="post" action="{{ url('/user/ubah_simpan/'.\$data->user_id) }}"> 10 11 {{ csrf_field() }} 12 {{ method_field('PUT') }} 13 14 <label>Username</label> 15 <input type="text" name="username" placeholder="Masukkan Username" value="{{ \$data->username }}"> 16
 17 <label>Nama</label> 18 <input type="text" name="name" placeholder="Masukkan Nama" value="{{ \$data->name }}"> 19
 20 <label>Password</label> 21 <input type="password" name="password" placeholder="Masukkan Password" value="{{ \$data->password }}"> 22
 23 <label>Level ID</label> 24 <input type="number" name="level_id" placeholder="Masukkan ID Level" value="{{ \$data->level_id }}"> 25

 26 <input type="submit" class="btn btn-succes" value="Ubah"> 27 28 </form> 29 </body> 30 </html> </pre>
12.	<p>Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini</p> <pre> Route::get('/user/ubah/{id}', [UserController::class, 'ubah']); </pre>
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> public function ubah(\$id) { \$user = UserModel::find(\$id); return view('user_ubah', ['data' => \$user]); } </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>

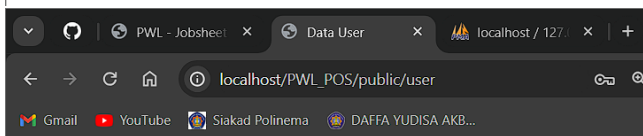
	 <p>Form Ubah Data User</p> <p>Kembali</p> <p>Username <input type="text" value="daffayds"/></p> <p>Nama <input type="text" value="Daffa Yudisa"/></p> <p>Password <input type="password" value="....."/></p> <p>Level ID <input type="text" value="1"/></p> <p><input type="button" value="Ubah"/></p>
15.	<p>Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini</p> <pre>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>public function ubah_simpan(\$id, Request \$request) { \$user = UserModel::find(\$id); \$user->username = \$request->username; \$user->name = \$request->name; \$user->password = Hash::make(\$request->password); \$user->level_id = \$request->level_id; \$user->save(); return redirect('/user'); }</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>



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
11	customer-1	Pelanggan Pertama	4	Ubah Hapus
12	manager_dua	Manager 2	2	Ubah Hapus
13	manager22	Manager Dua Dua	2	Ubah Hapus
14	manager33	Manager Tiga Tiga	2	Ubah Hapus
15	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus
17	daffaydsa	Daffa Yudisa Akbar	2	Ubah Hapus

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

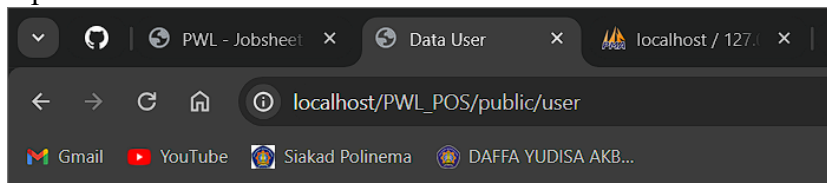
```
Route::get('/user/hapus/{id}', [UserController::class, 'hapus']);
```

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

```
public function hapus($id)
{
    $user = UserModel::find($id);
    $user->delete();

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


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



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
11	customer-1	Pelanggan Pertama	4	Ubah Hapus
12	manager_dua	Manager 2	2	Ubah Hapus
13	manager22	Manager Dua Dua	2	Ubah Hapus
14	manager33	Manager Tiga Tiga	2	Ubah Hapus
15	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus

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

feat: Praktikum 2.6 (- Create, Read, Update, Delete (CRUD))

daffayudisaa committed 10 hours ago

eb1f737



H. Praktikum 2.7 – Relationships

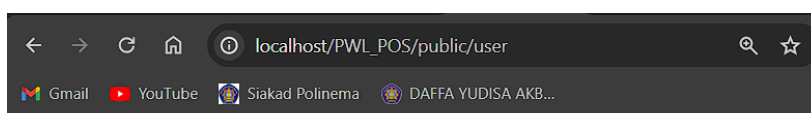
No	Keterangan Langkah dan Gambar
1.	Buka file model pada UserModel.php dan tambahkan scripnya menjadi seperti di bawah ini

	<pre> app > Models > UserModel.php > UserModel 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Factories\HasFactory; 6 use Illuminate\Database\Eloquent\Model; 7 use Illuminate\Database\Eloquent\Relations\BelongsTo; 8 use App\Models\LevelModel; 9 10 11 class UserModel extends Model 12 { 13 use HasFactory; 14 15 protected \$table = 'm_user'; 16 protected \$primaryKey = 'user_id'; 17 18 protected \$fillable = ['level_id', 'username', 'name', 'password']; 19 20 public function level(): BelongsTo 21 { 22 return \$this->belongsTo(LevelModel::class); 23 } 24 25 } 26 </pre>
2.	<p>Buka file controller pada UserController.php dan ubah method script menjadi seperti di bawah ini</p> <p>a. Jika dd(\$user) tidak di comment atau diabaikan</p> <pre> class UserController extends Controller { public function index() { // \$user = UserModel::all(); \$user = UserModel::with('level')->get(); dd(\$user); return view('user', ['data' => \$user]); } } </pre> <p>b. Jika dd(\$user) di comment atau diabaikan</p> <pre> class UserController extends Controller { public function index() { // \$user = UserModel::all(); \$user = UserModel::with('level')->get(); //dd(\$user); return view('user', ['data' => \$user]); } } </pre>
3.	<p>Simpan kode program Langkah 2. Kemudian jalankan link pada browser, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan</p> <p>a. Jika dd(\$user) tidak di comment atau diabaikan pada UserController.php</p>

```
localhost/PWL_POS/public/user

Illuminate\Database\Eloquent\Collection {#322} // app\Http\Controllers\UserController.php:16
#items: array:9 [▼
  0 => App\Models\UserModel {#329 ▶}
  1 => App\Models\UserModel {#330 ▶}
  2 => App\Models\UserModel {#331 ▶}
  3 => App\Models\UserModel {#332 ▶}
  4 => App\Models\UserModel {#333 ▶}
  5 => App\Models\UserModel {#334 ▶}
  6 => App\Models\UserModel {#335 ▶}
  7 => App\Models\UserModel {#336 ▶}
  8 => App\Models\UserModel {#337 ▶}
]
#escapeWhenCastingToString: false
}
```

b. Jika **dd(\$user)** di comment atau diabaikan pada UserController.php




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
11	customer-1	Pelanggan Pertama	4	Ubah Hapus
12	manager_dua	Manager 2	2	Ubah Hapus
13	manager22	Manager Dua Dua	2	Ubah Hapus
14	manager33	Manager Tiga Tiga	2	Ubah Hapus
15	manager56	Manager55	2	Ubah Hapus
16	manager12	Manager11	2	Ubah Hapus

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

feat: Praktikum 2.7 (Relationships)

 daffayudisaa committed 7 hours ago

5f074ca 