



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Sistem Informasi Bisnis
Semester : 5

Kelas : SIB
NIM : 2241760035
Nama : Fifi Novitasari
Jobsheet Ke- : 4

Laporan Jobsheet

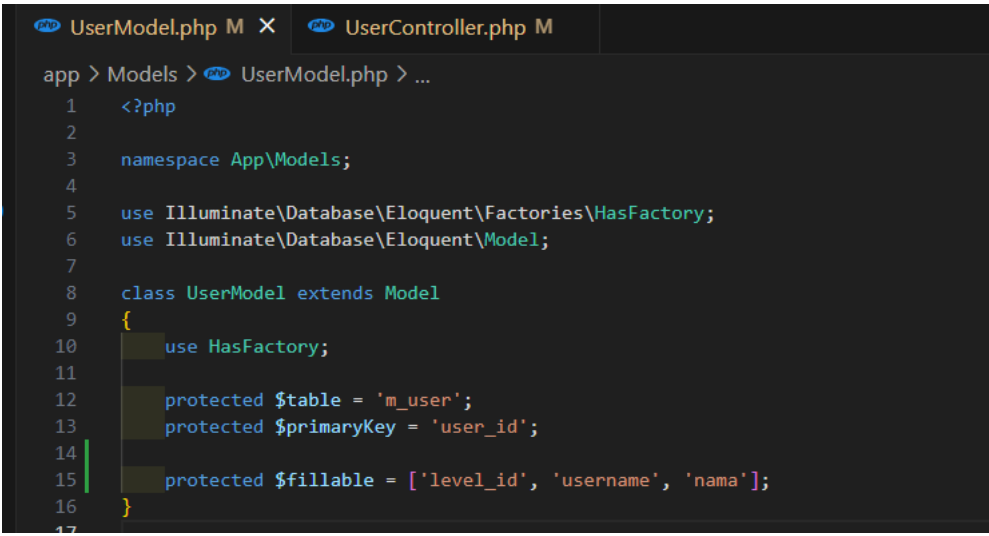
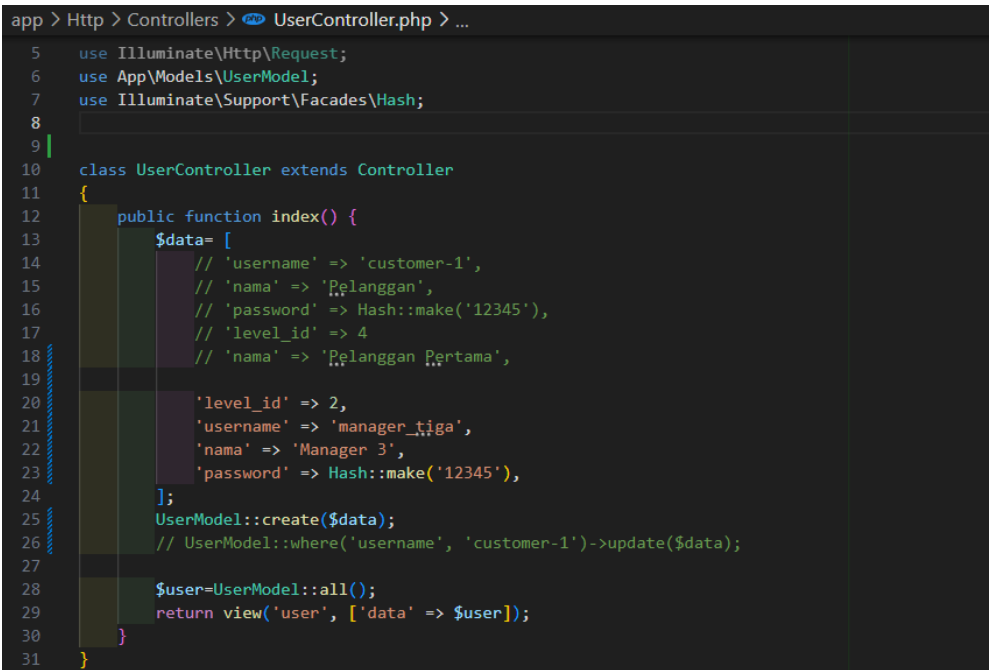
Praktikum Ke-1 \$fillable:

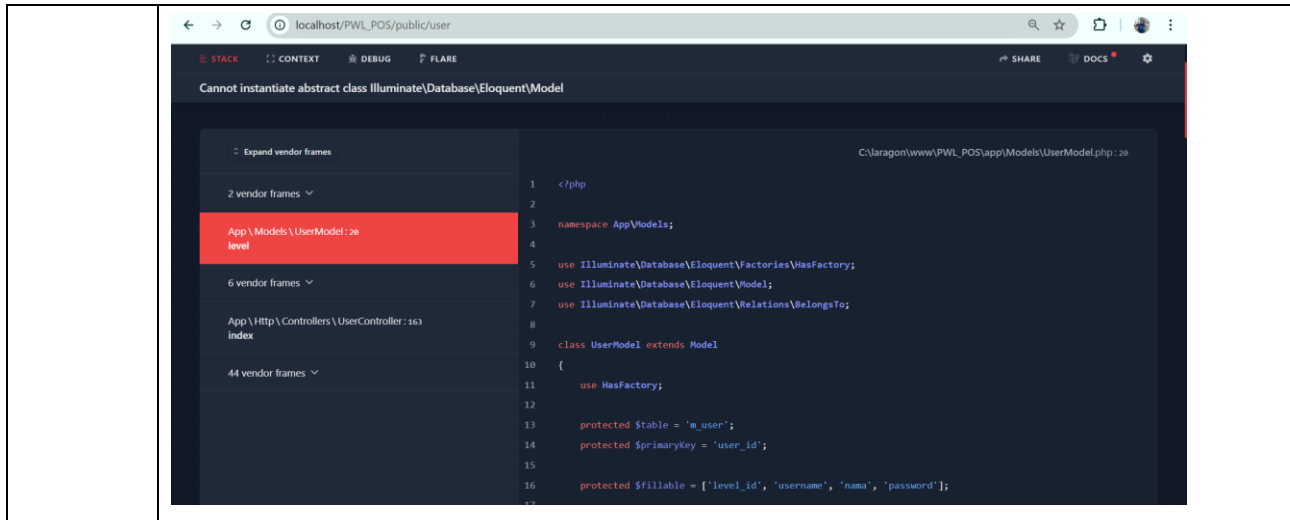
Langkah	Jawaban/Deskripsi
1	<p>Membuka file model dengan nama UserModel.php dan menambahkan \$fillable</p> <pre>app > Models > UserModel.php > UserModel 1 <?php 2 3 namespace App\Models; 4 5 use Illuminate\Database\Eloquent\Factories\HasFactory; 6 use Illuminate\Database\Eloquent\Model; 7 8 class UserModel 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 }</pre>
2	<p>Membuka file controller dengan nama UserController.php dan ubah script untuk menambahkan data baru</p>



	<pre>app > Http > Controllers > UserController.php > ... 5 use Illuminate\Http\Request; 6 use App\Models\UserModel; 7 use Illuminate\Support\Facades\Hash; 8 9 class UserController extends Controller 10 { 11 public function index() { 12 \$data= [13 // 'username' => 'customer-1', 14 // 'nama' => 'Pelanggan', 15 // 'password' => Hash::make('12345'), 16 // 'level_id' => 4 17 // 'nama' => 'Pelanggan Pertama', 18 19 'level_id' => 2, 20 'username' => 'manager_dua', 21 'nama' => 'Manager 2', 22 'password' => Hash::make('12345'), 23]; 24 UserModel::create(\$data); 25 // UserModel::where('username', 'customer-1')->update(\$data); 26 27 \$user=UserModel::all(); 28 return view('user', ['data' => \$user]); 29 } 30 }</pre>																								
3	<p>Menyimpan kode program Langkah 1 dan 2, dan menjalankan web server. localhostPWL_POS/public/user pada browser.</p> <h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr><tr><td>2</td><td>manager</td><td>Manager</td><td>2</td></tr><tr><td>3</td><td>staff</td><td>Staff/Kasir</td><td>3</td></tr><tr><td>9</td><td>Customer-1</td><td>Pelanggan Pertama</td><td>4</td></tr><tr><td>19</td><td>manager_dua</td><td>Manager 2</td><td>2</td></tr></table>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1	2	manager	Manager	2	3	staff	Staff/Kasir	3	9	Customer-1	Pelanggan Pertama	4	19	manager_dua	Manager 2	2
ID	Username	Nama	ID Level Pengguna																						
1	admin	Administrator	1																						
2	manager	Manager	2																						
3	staff	Staff/Kasir	3																						
9	Customer-1	Pelanggan Pertama	4																						
19	manager_dua	Manager 2	2																						
4	Mengubah file model UserModel.php pada bagian \$fillable																								



	 <pre>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', 'nama']; 16 } 17</pre>
5	<p>Mengubah kembali file controller UserController.php hanya bagian array pada \$data</p>  <pre>app > Http > Controllers > UserController.php > ... 5 use Illuminate\Http\Request; 6 use App\Models\UserModel; 7 use Illuminate\Support\Facades\Hash; 8 9 10 class UserController extends Controller 11 { 12 public function index() { 13 \$data= [14 // 'username' => 'customer-1', 15 // 'nama' => 'Pelanggan', 16 // 'password' => Hash::make('12345'), 17 // 'level_id' => 4 18 // 'nama' => 'Pelanggan Pertama', 19 20 'level_id' => 2, 21 'username' => 'manager_tiga', 22 'nama' => 'Manager 3', 23 'password' => Hash::make('12345'), 24]; 25 UserModel::create(\$data); 26 // UserModel::where('username', 'customer-1')->update(\$data); 27 28 \$user=UserModel::all(); 29 return view('user', ['data' => \$user]); 30 } 31 }</pre>
6	<p>Simpan kode program Langkah 4 dan 5. Kemudian menjalankan pada browser. Terjadi error karena password tidak memiliki default</p>



Praktikum Ke-2.1 Retrieving Single Models

Langkah	Jawaban/Deskripsi								
1	Membuka file controller dengan nama UserController.php dan ubah script <pre>27 28 \$user = UserModel::find(1); 29 return view('user', ['data' => \$user]); 30 31 }</pre>								
2	Membuka file view dengan nama user.blade.php dan ubah script <pre>14 15 @foreach (\$data as \$d) 16 <tr> 17 <td>{{ \$data->user_id }}</td> 18 <td>{{ \$data->username }}</td> 19 <td>{{ \$data->nama }}</td> 20 <td>{{ \$data->level_id }}</td> 21 </tr> 22 </pre>								
3	Kemudian menyimpan kode program Langkah 1 & 2 setelah itu menjalankan ke brower localhost/PWL_POS/public/user Data User <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level Pengguna						
1	admin	Administrator	1						
4	Ubah file controller dengan nama UserController.php dan ubah script								



	<pre>app > http > Controllers > UserController.php > ... 5 use Illuminate\Http\Request; 6 use App\Models\UserModel; 7 use Illuminate\Support\Facades\Hash; 8 9 10 class UserController extends Controller 11 { 12 public function index() { 13 // \$data= [14 // // 'username' => 'customer-1', 15 // // 'nama' => 'Pelanggan', 16 // // 'password' => Hash::make('12345'), 17 // // 'level_id' => 4 18 // // 'nama' => 'Pelanggan Pertama', 19 20 // // 'level_id' => 2, 21 // // 'username' => 'manager_tiga', 22 // // 'nama' => 'Manager 3', 23 // // 'password' => Hash::make('12345'), 24 //]; 25 // UserModel::create(\$data); 26 // // UserModel::where('username', 'customer-1')->update(\$data); 27 28 // \$user = UserModel::find(1); 29 30 \$user = UserModel::where('level_id', 1)->first(); 31 return view('user', ['data' => \$user]); 32 } 33 }</pre>								
5	<p>Menyimpan kode program Langkah 4. Kemudian menjalankan pada browser</p> <h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level Pengguna						
1	admin	Administrator	1						
6	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>31 \$user = UserModel::firstwhere('level_id', 1); 32 return view('user', ['data' => \$user]); 33 } 34 }</pre>								
7	<p>Simpan kode program Langkah 6. Kemudian jalankan pada browser</p> <h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level Pengguna						
1	admin	Administrator	1						
8	<p>Ubah file controller dengan nama UserController.php dan ubah script</p>								



	<pre>32 33 \$user = UserModel::findOr(1, ['level_id', 'nama'], function() { 34 abort(404); 35 }); 36 return view('user', ['data' => \$user]); 37 } 38 }</pre>								
9	<p>Simpan kode program Langkah 8. Kemudian pada browser</p> <h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td></td><td>admin</td><td>Administrator</td><td></td></tr></table> <p>Data yang keluar hanya Username dan Nama untuk ID dan ID Level Pengguna tidak kelaur</p>	ID	Username	Nama	ID Level Pengguna		admin	Administrator	
ID	Username	Nama	ID Level Pengguna						
	admin	Administrator							
10	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>32 33 \$user = UserModel::findOr(20, ['level_id', 'nama'], function() { 34 abort(404); 35 }); 36 return view('user', ['data' => \$user]); 37 } 38 }</pre>								
11	<p>Simpan kode program Langkah 10. Kemudian jalankan pada browser</p> <div><p>404 NOT FOUND</p></div> <p>Karena tidak ada hasil yang diinginkan maka kode menjalankan fungsi abort(404)</p>								

Praktikum Ke-2.2 *Not Found Exceptions*

Langkah	Jawaban/Deskripsi
1	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>37 \$user = UserModel::findOrFail(1); 38 return view('user', ['data' => \$user]); 39 } 40 }</pre>
2	<p>Simpan kode program Langkah 1. Kemudian jalankan pada browser</p>



	<h1>Data User</h1> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level Pengguna						
1	admin	Administrator	1						
3	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>37 // \$user = UserModel::findOrFail(1); 38 \$user = UserModel::where('username', 'manager9')->firstOrFail(); 39 return view('user', ['data' => \$user]); 40 } 41 }</pre>								
4	<p>Simpan kode program Langkah 3. Kemudian jalankan pada browser</p> <div><p>404 NOT FOUND</p></div> <p>Tampilan muncul abror(404) karena database tidak ada username Manager9</p>								

Praktikum Ke-2.3 *Retreiving Aggregrates*

Langkah	Jawaban/Deskripsi		
1	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>39 40 \$user = UserModel::where('level_id', 2)->count(); 41 dd(\$user); 42 return view('user', ['data' => \$user]); 43 } 44 }</pre>		
2	<p>Simpan kode program Langkah 1. Kemudian jalankan pada browser</p> <pre>2 // app\Http\Controllers\UserController.php:75</pre>		
3	<p>Buat agar jumlah script pada langkah 1 bisa tampil pada halaman browser</p> <h2>Data User</h2> <table><tr><th>Jumlah Pengguna</th></tr><tr><td>2</td></tr></table>	Jumlah Pengguna	2
Jumlah Pengguna			
2			



Praktikum Ke-2.4 Retrieving or Creating Models

Langkah	Jawaban/Deskripsi								
1	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>43 \$user = UserModel::firstOrCreate(44 [45 'username' => 'manager', 46 'nama' => 'Manager', 47], 48); 49 return view('user', ['data' => \$user]); 50 } 51 }</pre>								
2	<p>Ubah kembali file view dengan nama user.blade.php dan ubah script</p> <pre>resources > views > user.blade.php > html > body > table 1 <!DOCTYPE html> 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>ID</th> 11 <th>Username</th> 12 <th>Nama</th> 13 <th>ID Level Pengguna</th> 14 </tr> 15 @foreach (\$data as \$d) 16 <tr> 17 <td>{{ \$data->user_id }}</td> 18 <td>{{ \$data->username }}</td> 19 <td>{{ \$data->nama }}</td> 20 <td>{{ \$data->level_id }}</td> 21 </tr> 22 </table> 23 </body> 24 </html></pre>								
3	<p>Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada browser</p> <h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>2</td><td>manager</td><td>Manager</td><td>2</td></tr></table> <p>Username manager sudah ada jadi firstOrCreate mengambil data selanjutnya</p>	ID	Username	Nama	ID Level Pengguna	2	manager	Manager	2
ID	Username	Nama	ID Level Pengguna						
2	manager	Manager	2						
4	Ubah file controller dengan nama UserController.php dan ubah								



	<pre>49 50 \$user = UserModel::firstOrCreate(51 [52 'username' => 'manager22', 53 'nama' => 'Manager Dua Dua', 54 'password' => Hash::make('12345'), 55 'level_id' => 2 56], 57); 58 return view('user', ['data' => \$user]); 59 } 60 }</pre>								
5	<p>Simpan kode program Langkah 4. Kemudian jalankan pada browser dan cek juga pada phpMyAdmin pada tabel m_user</p> <h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>21</td><td>manager22</td><td>Manager Dua Dua</td><td>2</td></tr></table> <p>firstOrCreate menambahkan data yang ada saja</p>	ID	Username	Nama	ID Level Pengguna	21	manager22	Manager Dua Dua	2
ID	Username	Nama	ID Level Pengguna						
21	manager22	Manager Dua Dua	2						
4	<p>Ubah file controller dengan nama UserController.php dan ubah</p> <pre>59 60 \$user = UserModel::firstOrCreate(61 [62 'username' => 'manager', 63 'nama' => 'Manager', 64], 65); 66 return view('user', ['data' => \$user]); 67 }</pre>								
5	<p>Simpan kode program Langkah 6. Kemudian jalankan pada browser</p> <h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>2</td><td>manager</td><td>Manager</td><td>2</td></tr></table>	ID	Username	Nama	ID Level Pengguna	2	manager	Manager	2
ID	Username	Nama	ID Level Pengguna						
2	manager	Manager	2						
6	<p>Ubah file controller dengan nama UserController.php dan ubah script</p>								



	<pre>59 \$user = UserModel::firstOrCreate(60 [61 'username' => 'manager33', 62 'nama' => 'Manager Tiga Tiga', 63 'password' => Hash::make('12345'), 64 'level_id' => 2 65], 66); 67 return view('user', ['data' => \$user]); 68 } 69 }</pre>								
7	<p>Simpan kode program Langkah 8. Kemudian jalankan pada browser</p> <h3>Data User</h3> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td></td><td>manager33</td><td>Manager Tiga Tiga</td><td>2</td></tr></table>	ID	Username	Nama	ID Level Pengguna		manager33	Manager Tiga Tiga	2
ID	Username	Nama	ID Level Pengguna						
	manager33	Manager Tiga Tiga	2						
8	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>\$user = UserModel::firstOrCreate(['username' => 'manager33', 'nama' => 'Manager Tiga Tiga', 'password' => Hash::make('12345'), 'level_id' => 2],); \$user->save(); return view('user', ['data' => \$user]); }</pre>								
9	<p>Simpan kode program Langkah 9. Kemudian jalankan pada browser</p> <h3>Data User</h3> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>22</td><td>manager33</td><td>Manager Tiga Tiga</td><td>2</td></tr></table> <p>Ada tambahan pada firstOrCreate yaitu metode save secara manual</p>	ID	Username	Nama	ID Level Pengguna	22	manager33	Manager Tiga Tiga	2
ID	Username	Nama	ID Level Pengguna						
22	manager33	Manager Tiga Tiga	2						

Praktikum Ke-2.5 Attribute Changes

Langkah	Jawaban/Deskripsi
---------	-------------------



1	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>71 \$user = UserModel::create([72 'username' => 'manager55', 73 'nama' => 'Manager55', 74 'password' => Hash::make('12345'), 75 'level_id' => 2, 76]); 77 \$user->username = 'manager56'; 78 79 \$user->isDirty(); //true 80 \$user->isDirty('username'); //true 81 \$user->isDirty('nama'); //false 82 \$user->isDirty(['nama', 'username']); //true 83 84 \$user->isClean(); //false 85 \$user->isClean('username'); //false 86 \$user->isClean('nama'); //true 87 \$user->isClean(['nama', 'username']); //false 88 89 \$user->save(); 90 91 \$user->isDirty(); //false 92 \$user->isClean(); //true 93 dd(\$user->isDirty());</pre>
2	<p>Simpan kode program Langkah 1. Kemudian jalankan pada browser</p> <pre>false // app\Http\Controllers\UserController.php:122</pre>
3	<p>Ubah file controller dengan nama UserController.php dan ubah script</p> <pre>94 95 \$user = UserModel::create([96 'username' => 'manager11', 97 'nama' => 'Manager1', 98 'password' => Hash::make('12345'), 99 'level_id' => 2, 100]); 101 \$user->username = 'manager12'; 102 103 \$user->save(); 104 105 \$user->wasChanged(); //true 106 \$user->wasChanged('username'); //true 107 \$user->wasChanged(['username', 'level_id']); //true 108 \$user->wasChanged('nama'); //false 109 dd(\$user->wasChanged(['nama', 'username'])); //true 110 } 111 }</pre>
4	<p>Simpan kode program Langkah 3. Kemudian jalankan pada browser</p> <pre>true // app\Http\Controllers\UserController.php:138</pre>



Praktikum Ke-2.6 Create, Read, Update, Delete (CRUD)

Langkah	Jawaban/Deskripsi																																																		
1	<p>Buka file view pada user.blade.php dan buat scripnya</p> <pre>resources > views > user.blade.php > html > body > table > tr 1 <!DOCTYPE html> 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 <td>ID</td> 11 <td>Username</td> 12 <td>Nama</td> 13 <td>ID Level Pengguna</td> 14 <td>Aksi</td> 15 </tr> 16 @foreach (\$data as \$d) 17 <tr> 18 <td>{{ \$d->user_id }}</td> 19 <td>{{ \$d->username }}</td> 20 <td>{{ \$d->nama }}</td> 21 <td>{{ \$d->level_id }}</td> 22 <td>user_id }}">Ubah user_id }}">Hapus</td> 23 </tr> 24 </foreach> 25 </table> 26 </body> 27 </html></pre>																																																		
2	<p>Buka file controller pada UserController.php dan buat scripnya</p> <pre>111 \$user = UserModel::all(); 112 return view('user', ['data' => \$user]); 113 } 114 }</pre>																																																		
3	<p>Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada browser</p> <pre>\$user = UserModel::all(); return view('user', ['data' => \$user]); } public function tambah() { return view('user_tambah'); } }</pre>																																																		
4	<p>Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada browser</p> <h2>Data User</h2> <p>+ Tambah User</p> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th><th>Aksi</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td><td>Ubah Hapus</td></tr><tr><td>2</td><td>manager</td><td>Manager</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>3</td><td>staff</td><td>Staff/Kasir</td><td>3</td><td>Ubah Hapus</td></tr><tr><td>9</td><td>Customer-1</td><td>Pelanggan Pertama</td><td>4</td><td>Ubah Hapus</td></tr><tr><td>19</td><td>manager_dua</td><td>Manager 2</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>21</td><td>manager22</td><td>Manager Dua Dua</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>22</td><td>manager33</td><td>Manager Tiga Tiga</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>23</td><td>manager56</td><td>Manager55</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>24</td><td>manager12</td><td>Manager11</td><td>2</td><td>Ubah Hapus</td></tr></table>	ID	Username	Nama	ID Level Pengguna	Aksi	1	admin	Administrator	1	Ubah Hapus	2	manager	Manager	2	Ubah Hapus	3	staff	Staff/Kasir	3	Ubah Hapus	9	Customer-1	Pelanggan Pertama	4	Ubah Hapus	19	manager_dua	Manager 2	2	Ubah Hapus	21	manager22	Manager Dua Dua	2	Ubah Hapus	22	manager33	Manager Tiga Tiga	2	Ubah Hapus	23	manager56	Manager55	2	Ubah Hapus	24	manager12	Manager11	2	Ubah Hapus
ID	Username	Nama	ID Level Pengguna	Aksi																																															
1	admin	Administrator	1	Ubah Hapus																																															
2	manager	Manager	2	Ubah Hapus																																															
3	staff	Staff/Kasir	3	Ubah Hapus																																															
9	Customer-1	Pelanggan Pertama	4	Ubah Hapus																																															
19	manager_dua	Manager 2	2	Ubah Hapus																																															
21	manager22	Manager Dua Dua	2	Ubah Hapus																																															
22	manager33	Manager Tiga Tiga	2	Ubah Hapus																																															
23	manager56	Manager55	2	Ubah Hapus																																															
24	manager12	Manager11	2	Ubah Hapus																																															



5	<p>Langkah berikutnya membuat create atau tambah data user dengan cara bikin file baru pada view dengan nama user_tambah.blade.php dan buat scriptnya</p> <pre>resources > views > user_tambah.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 <meta http-equiv="X-UA-Compatible" content="ie=edge"> 8 <title>Tambah User</title> 9 </head> 10 11 <body> 12 <h1>Form Tambah Data User</h1> 13 <form method="post" action="/user/tambah_simpan"> 14 15 {{ csrf_field() }} 16 17 <label for="">Username</label> 18 <input type="text" name="username" placeholder="Masukan Username"> 19
 20 <label for="">Nama</label> 21 <input type="text" name="nama" placeholder="Masukan Nama"> 22
 23 <label for="">Password</label> 24 <input type="password" name="password" placeholder="Masukan Password"> 25
 26 <label for="">Level ID</label> 27 <input type="number" name="level_id" placeholder="Masukan ID Level"> 28

 29 <input type="submit" class="btn btn-success" value="Simpan"> 30 </form> 31 </body> 32 </html></pre>
6	<p>Tambahkan script pada routes dengan nama file web.php</p> <pre>Route::get('/user/tambah', [UserController::class, 'tambah']);</pre>
7	<p>Tambahkan script pada controller dengan nama file UserController.php. Tambahkan script dalam class dan buat method baru dengan nama tambah</p> <pre>113 } 114 } 115 public function tambah() { 116 return view('user_tambah'); 117 } 118 }</pre>
8	<p>Simpan kode program Langkah 4 s/d 6. Kemudian jalankan pada browser dan klik link “+ Tambah User”</p>



	<div><div>← → ↻ ⓘ localhost/PWL_POS/public/user/tambah</div><h2>Form Tambah Data User</h2><div><div>Username</div><input type="text" value="Masukan Username"/></div><div><div>Nama</div><input type="text" value="Masukan Nama"/></div><div><div>Password</div><input type="password" value="Masukan Password"/></div><div><div>Level ID</div><input type="text" value="Masukan ID Level"/></div><div><div>Simpan</div></div></div>
9	<p>Tambahkan script pada routes dengan nama file web.php</p> <pre>Route::get('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);</pre>
10	<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</p> <pre>118 119 public function tambah_simpan(Request \$request) { 120 UserModel::create([121 'username' => \$request->username, 122 'nama' => \$request->nama, 123 'password' => Hash::make(\$request->password), 124 'level_id' => \$request->level_id 125]); 126 127 return redirect('/user'); 128 } 129 }</pre>
11	<p>Simpan kode program Langkah 8 dan 9. Kemudian jalankan link localhost:8000/user/tambah atau localhost/PWL_POS/public/user/tambah</p> <div><div>← → ↻ ⓘ localhost/PWL_POS/public/user/tambah</div><h2>Form Tambah Data User</h2><div><div>Username</div><input type="text" value="Masukan Username"/></div><div><div>Nama</div><input type="text" value="Masukan Nama"/></div><div><div>Password</div><input type="password" value="Masukan Password"/></div><div><div>Level ID</div><input type="text" value="Masukan ID Level"/></div><div><div>Simpan</div></div></div>
12	<p>Langkah berikutnya membuat update atau ubah data user dengan cara bikin file baru pada view dengan nama user_ubah.blade.php</p>



	<pre>resources > views > user_ubah.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 <meta http-equiv="X-UA-Compatible" content="ie=edge"> 8 <title>Ubah User</title> 9 </head> 10 11 <body> 12 <h1>Form Ubah Data User</h1> 13 Kembali 14

 15 <form method="post" action="/user/ubah_simpan/{{ \$data->user_id }}"> 16 {{ csrf_field() }} 17 {{ method_field('PUT') }} 18 <label for="">Username</label> 19 <input type="text" name="username" placeholder="Masukan Username" value="{{ \$data->username }}"> 20
 21 <label for="">Nama</label> 22 <input type="text" name="nama" placeholder="Masukan Nama" value="{{ \$data->nama }}"> 23
 24 <label for="">Password</label> 25 <input type="password" name="password" placeholder="Masukan Password" value="{{ \$data->password }}"> 26
 27 <label for="">Level ID</label> 28 <input type="number" name="level_id" placeholder="Masukan ID Level" value="{{ \$data->level_id }}"> 29

 30 <input type="submit" class="btn btn-success" value="Ubah"> 31 </form> 32 </body> 33 </html></pre>
13	<p>Tambahkan script pada routes dengan nama file web.php.</p> <pre>Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);</pre>
14	<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</p> <pre>30 public function ubah(\$id) { 31 \$user = UserModel::find(\$id); 32 return view('user_ubah', ['data' => \$user]); 33 }</pre>
15	<p>Tambahkan script pada routes dengan nama file web.php</p> <pre>Route::get('/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</p> <pre>34 public function ubah_simpan(\$id, Request \$request) { 35 \$user = UserModel::find(\$id); 36 \$user->username = \$request->username; 37 \$user->nama = \$request->nama; 38 \$user->password = Hash::make(\$request->password); 39 \$user->level_id = \$request->level_id; 40 \$user->save(); 41 return redirect('/user'); 42 }</pre>



17

Berikut untuk langkah delete . Tambahkan script pada routes dengan nama file web.php.

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

18

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

```

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

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

19

Simpan kode program Langkah 18 dan 19. Kemudian jalankan pada browser dan klik tombol 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
9	Customer-1	Pelanggan Pertama	4	Ubah Hapus
19	manager_dua	Manager 2	2	Ubah Hapus
21	manager22	Manager Dua Dua	2	Ubah Hapus
22	manager33	Manager Tiga Tiga	2	Ubah Hapus
23	manager56	Manager55	2	Ubah Hapus
24	manager12	Manager11	2	Ubah Hapus

Praktikum Ke-2.7 Relationships

Langkah	Jawaban/Deskripsi
1	Buka file model pada UserModel.php dan tambahkan scripnya



	<pre>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 use Illuminate\Database\Eloquent\Relations\BelongsTo; 8 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', 'nama', 'password']; 17 18 public function level(): BelongsTo 19 { 20 return \$this->belongsTo(Model::class, 'level_id', 'level_id'); 21 } 22 } 23</pre>
2	Buka file controller pada UserController.php dan ubah method script <pre>public function index() { \$user = UserModel::with('level')->get(); dd(\$user); }</pre>
3	Simpan kode program Langkah 2. Kemudian jalankan link pada browser <pre>Illuminate\Database\Eloquent\Collection {#319 ▼ // app\Http\Controllers\UserController.php:194 #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 }</pre>
4	Buka file controller pada UserController.php dan ubah method script <pre>public function index() { \$user = UserModel::with('level')->get(); return view('user', ['data' => \$user]); }</pre>
5	Buka file view pada user.blade.php dan ubah script



```
resources > views > user.blade.php > html > body > table
1 <!DOCTYPE html>
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        <td>ID</td>
11        <td>Username</td>
12        <td>Nama</td>
13        <td>ID Level Pengguna</td>
14        <td>Kode Level</td>
15        <td>Nama Level</td>
16        <td>Aksi</td>
17      </tr>
18      @foreach ($data as $d)
19        <tr>
20          <td>{{ $d->user_id }}</td>
21          <td>{{ $d->username }}</td>
22          <td>{{ $d->nama }}</td>
23          <td>{{ $d->level_id }}</td>
24          <td>{{ $d->level->level_kode }}</td>
25          <td>{{ $d->level->level_nama }}</td>
26          <td><a href="/user/ubah/{{ $d->user_id }}">Ubah</a> | <a href="/user/hapus/{{ $d->user_id }}">Hapus</a></td>
27        </tr>
28      </tr>
29    </table>
30  </body>
31 </html>
```

6

Simpan kode program Langkah 4 dan 5. Kemudian jalankan link pada browser

Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Kode Level	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	Satff/Kasir	Ubah Hapus
9	Customer-1	Pelanggan Pertama	4	kode_lv4	Nama Level 4	Ubah Hapus
19	manager_dua	Manager 2	2	MNG	Manager	Ubah Hapus
21	manager22	Manager Dua Dua	2	MNG	Manager	Ubah Hapus
22	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah Hapus
23	manager56	Manager55	2	MNG	Manager	Ubah Hapus
24	manager12	Manager11	2	MNG	Manager	Ubah Hapus

Link Github: