



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut


Nama	:	AL BYAN AGUNG SHAFIQRI
Nim	:	2241720080
Kelas	:	TI 2H

LEMBAR JAWABAN JOBSHEET-4

Soal No	Jawaban																								
1	<pre>protected \$table = 'm_user'; protected \$primaryKey = 'user_id'; protected \$fillable = ['level_id', 'username', 'nama', 'password']; }</pre>																								
	<pre>public function index(){ \$data =['level_id' => 2, 'username' => 'manager-dua', 'nama' => 'Manager 2', 'password' => Hash::make('12345')]; UserModel::create(\$data); \$user = UserModel::all(); return view('user', ['data' => \$user]); }</pre>																								
	<h3>Data User</h3> <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>4</td><td>customer-1</td><td>Pelanggan Pertama</td><td>4</td></tr><tr><td>5</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	4	customer-1	Pelanggan Pertama	4	5	manager-dua	Manager 2	2
ID	Username	Nama	ID Level Pengguna																						
1	admin	Administrator	1																						
2	manager	Manager	2																						
3	staff	Staff/Kasir	3																						
4	customer-1	Pelanggan Pertama	4																						
5	manager-dua	Manager 2	2																						
	<pre>protected \$fillable = ['level_id', 'username', 'nama'];</pre>																								
	<div><div>Illuminate \Database \QueryException</div><div>SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value</div><div>INSERT INTO `m_user` (`level_id`, `username`, `nama`, `updated_at`, `created_at`) VALUES (2, manager-tiga, Manager 3, 2024-03-18 13:03:58, 2024-03-18 13:03:58)</div></div>																								
	Terjadi Error karna field “password” tidak fillable																								



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

	<div>test: fillabel field</div> <div> iannstronaut pushed 1 commit • 1ec11e1...fac31e8 • 1 minute ago</div>									
2	<pre>\$user = UserModel::find(1); return view('user', ['data' => \$user]);</pre>									
	<pre></tr> <tr> <td>{{ \$data->user_id}}</td> <td>{{ \$data->username}}</td> <td>{{ \$data->nama}}</td> <td>{{ \$data->level_id}}</td> </tr> </table></pre>									
	<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> <p>Query akan mencari data dengan id = 1 dan menampilkanya kedalam view ‘user’</p>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1	
ID	Username	Nama	ID Level Pengguna							
1	admin	Administrator	1							
	<pre>\$user = UserModel::where('level_id', 1)->first(); return view('user', ['data' => \$user]);</pre>									
	<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> <p>Fungsi where akan mencari data sesuai dengan parameter yang disini dinyatakan dengan ‘level_id’ = 1 dan dilanjutkan dengan fungsi first() yang akan mengambil data hanya data yang pertama</p>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1	
ID	Username	Nama	ID Level Pengguna							
1	admin	Administrator	1							
	<pre>\$user = UserModel::firstWhere('level_id', 1); return view('user', ['data' => \$user]);</pre>									



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

firstWhere sama seperti sebelumnya tetapi memiliki sintaks yang lebih singkat

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

Data User

ID	Username	Nama	ID Level Pengguna
	admin	Administrator	


FindOr akan mencari data sesuai dengan parameter, jika data tidak ada maka akan mengembalikan abort 404

```
= UserModel::findOr(20,['username', 'nama'],  
abort(404);
```

404 | NOT FOUND

Dikarnakan findOr tidak ditemukan maka mengembalikan abort 404 seperti gambar diatas

test: find, findWhere, where, findOr , dan first

 iannstronaut committed now



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

2.2	<pre>\$user = UserModel::findOrFail(1); return view('user', ['data' => \$user]);</pre>								
	<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> <p>findOrFail(): method untuk mencari data user berdasarkan id. Method ini akan menghasilkan error jika data user dengan id yang ditentukan tidak ditemukan.</p>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level Pengguna						
1	admin	Administrator	1						
	<pre>\$user = UserModel::where('username', 'manager9')->firstOrFail(); return view('user', ['data' => \$user]);</pre>								
	<div><div>WhatsApp</div><div>Not Found</div><div>localhost:8000/user</div><div>404 NOT FOUND</div></div> <p>Sama seperti findOrFail() , firstOrFail akan mengambil data, jika data tidak ditemukan maka akan mengembalikan error,hanya saja firstOrFailed hanya mengambil data pertama</p>								
	<pre>test: findOrFail & firstOrFail iannstronaut committed now</pre>								
2.3	<pre>\$user = UserModel::where('level_id', 2)->count(); dd(\$user); return view('user', ['data' => \$user]);</pre>								



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

```
2 // app\Http\Controllers\UserController.php:36
```

user_id	level_id	username	nama	password
1	1	admin	Administrator	\$2y\$12\$9lmWA2DR7Y08RIVDyMXvr.G3G8uyBlyYegK6dR9883X...
2	2	manager	Manager	\$2y\$12\$RH9Yv6ENZfID7p3s6C9LN.wFejucAyKK2vcglOcf.7Y...
3	3	staff	Staff/Kasir	\$2y\$12\$WytmAYZ090oCFNcrQl8ZkusXAaJtHIYRKNgIDOpGui...
4	4	customer-1	Pelanggan Pertama	\$2y\$12\$JuR0HGapijN.U6yHfDZRC.zYTLOS3Srbhbxhr7HcML...
5	2	manager-dua	Manager 2	\$2y\$12\$6Vbka.7Ebc7fKnpXeuT20edIS0/LQxbEKKGpeAizD2y...

Fungsi count akan menjumlah data yang query where dapat sesuai parameter

```
$user = UserModel::where('level_id', 2)->count();  
// dd($user);  
return view('user', ['data' => $user]);
```

```
<th>Jumlah Pengguna</th>  
{!-- <th>Username</th>  
<th>Nama</th>  
<th>ID Level Pengguna</th> --}}  
</tr>  
<tr>  
<td>{{ $data }}</td>  
{!-- <td>{{ $data->username }}</td>  
<td>{{ $data->nama }}</td>  
<td>{{ $data->level_id }}</td> --}}  
</tr>
```

Data User

Jumlah Pengguna

2

test: count



iannstronaut committed now



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

2.4

```
$user = UserModel::firstOrCreate([  
    'username' => 'manager',  
    'nama' => 'Manager'  
]);  
return view('user', ['data' => $user]);
```

```
<tr>  
    <td>{{ $data->user_id}}</td>  
    <td>{{ $data->username}}</td>  
    <td>{{ $data->nama}}</td>  
    <td>{{ $data->level_id}}</td>  
</tr>  
</table>
```

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

Fungsi firstOrCreate adalah untuk mencari data pertama sesuai dengan parameter yang diberikan, jika data tidak ada maka data akan dibuat dan dimasukkan kedalam database

```
$user = UserModel::firstOrCreate([  
    'username' => 'manager22',  
    'nama' => 'Manager Dua Dua',  
    'password' => Hash::make('12345'),  
    'level_id' => 2  
]);  
return view('user', ['data' => $user]);
```

Data User

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

user_id	level_id	username	nama
1	1	admin	Administrator
2	2	manager	Manager
3	3	staff	Staff/Kasir
4	4	customer-1	Pelanggan Pertama
5	2	manager-dua	Manager 2
6	2	manager22	Manager Dua Dua

Data tidak ditemukan sebelumnya sehingga data dibuat dan dimasukkan kedalam database

```
$user = UserModel::firstOrCreate([  
    'username' => 'manager',  
    'nama' => 'Manager'  
]);
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

```
// ??-> $user ?? $user = UserModel::firstOrCreate([    'username' => 'manager33',    'nama' => 'Manager Tiga Tiga',    'password' => Hash::make('12345'),    'level_id' => 2]);
```

Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2


```
$user = UserModel::firstOrCreate([    'username' => 'manager33',    'nama' => 'Manager Tiga Tiga',    'password' => Hash::make('12345'),    'level_id' => 2]); $user->save(); return view('user', ['data' => $user]);
```

Data User

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

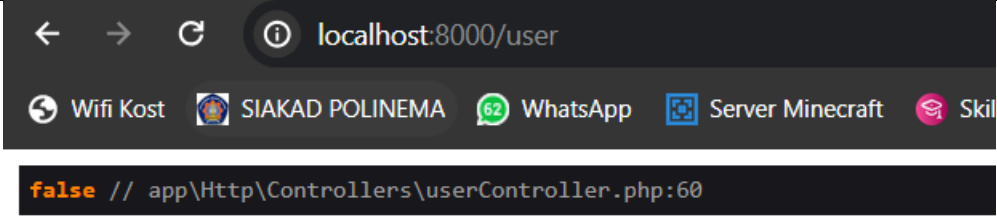
firstOrCreate([: Method ini digunakan untuk mencari data user berdasarkan field yang ditentukan. Jika data user tidak ditemukan, maka method ini akan membuat instance baru dari class UserModel dengan nilai default. Dan sintaks \$user->save() membuat data yang instance dimasukkan kedalam database

test: firstOr Create & firstOrCreate

 iannstronaut committed now



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

2.5	<pre> \$user = UserModel::firstOrCreate(['username' => 'manager355', 'nama' => 'Manager55', 'password' => Hash::make('12345'), 'level_id' => 2]); \$user->username = 'manager 56'; \$user->isDirty(); \$user->isDirty('username'); \$user->isDirty('nama'); \$user->isDirty(['nama', 'username']); \$user->isClean(); \$user->isClean('username'); \$user->isClean('nama'); \$user->isClean(['nama', 'username']); \$user->save(); \$user->isDirty(); \$user->isClean(); dd(\$user->isDirty()); // return view('user', ['data' => \$user]); </pre>
	 <p>Hasil false dikarenakan sejak data di save(); data tidak terjadi modifikasi sehingga data user menjadi clean</p>
	<pre> // // dd(\$user); \$user = UserModel::firstOrCreate(['username' => 'manager11', 'nama' => 'Manager11', 'password' => Hash::make('12345'), 'level_id' => 2]); \$user->username = 'manager 12'; \$user->save(); \$user->wasChange(); \$user->wasChange('username'); \$user->wasChange(['username', 'level_id']); \$user->wasChange('nama'); \$user->wasChange(['nama', 'username']); dd(\$user->wasChange(['nama', 'username'])); </pre>



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

2.6	<pre>public function index(){ \$user = UserModel::all(); return view('user', ['data' => \$user]); }</pre>
	<pre><body> <h1>Data User</h1> + Tambah User <table border="1" cellpadding="2" cellspacing="0"> <tr> <th>ID</th> <th>Username</th> <th>Nama</th> <th>ID Level Pengguna</th> <th>Aksi</th> </tr> @foreach (\$data as \$d) <tr> <td>{{ \$d->user_id }}</td> <td>{{ \$d->username }}</td> <td>{{ \$d->nama }}</td> <td>{{ \$d->level_id }}</td> <td>user_id }}">Ubah user_id }}">Hapus</td> </tr> @endforeach </table> </body></pre>
	<pre><body> <h1>Form Tambah Data User</h1> <form method="POST" action="/user/tambah_simpan"> {{ csrf_field() }} <label>Username</label> <input type="text" name="username" placeholder="Masukkan Username">
 <label>Nama</label> <input type="text" name="nama" placeholder="Masukkan Nama">
 <label>Password</label> <input type="password" name="password" placeholder="Masukkan Password">
 <label>Level ID</label> <input type="number" name="level_id" placeholder="Masukkan ID Level">

 <input type="submit" value="Simpan" class="btn btn-success"> </form> </body></pre>
	<pre>Route::get('/user/tambah', [UserController::class, 'tambah']);</pre>



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

```
public function tambah(){
    return view('user_tambah');
}
```

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
4	customer-1	Pelanggan Pertama	4	Ubah Hapus
5	manager-dua	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager 56	Manager55	2	Ubah Hapus
11	manager 12	Manager11	2	Ubah Hapus

Form Tambah Data User

Username

Masukkan Username

Nama

Masukkan Nama

Password

Masukan Password

Level ID

Masukkan ID Level

Simpan

Saat + Tambah User diklik akan diarahkan ke halaman user/tambah yang berisi form penambahan data

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

```
public function tambah_simpan(Request $req){
    UserModel::create([
        'username' => $req->username,
        'nama' => $req->nama,
        'password' => Hash::make($req->password),
        'level_id' => $req->level_id,
    ]);
    return redirect('/user');
}
```

Form Tambah Data User

Username

JDoe

Nama

John Doe

Password

.....

Level ID

1

Simpan



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

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
4	customer-1	Pelanggan Pertama	4	Ubah Hapus
5	manager-dua	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager 56	Manager55	2	Ubah Hapus
11	manager 12	Manager11	2	Ubah Hapus
12	JDoe	John Doe	1	Ubah Hapus

Data berhasil ditambahkan

```
<body>
<h1>Form Edit Data User</h1>
<a href="/user">Kembali</a>
<br><br>

<form method="post" action="/user/ubah_simpan/{{ $data->user_id }}">

    {{ csrf_field() }}
    {{ method_field("PUT") }}

    <label>Username</label>
    <input type="text" name="username" placeholder="Masukkan Username" value="{{ $data->username }}">
    <br>
    <label>Nama</label>
    <input type="text" name="nama" placeholder="Masukkan Nama" value="{{ $data->nama }}">
    <br>
    <label>Password</label>
    <input type="password" name="password" placeholder="Masukan Password" value="{{ $data->password }}">
    <br>
    <label>level ID</label>
    <input type="number" name="level_id" placeholder="Masukkan ID Level" value="{{ $data->level_id }}">
    <br><br>
    <input type="submit" value="Ubah" class="btn btn-success">
</form>
</body>
```

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

```
public function ubah($id){
    $user = UserModel::find($id);
    return view('user_ubah', ['data' => $user]);
}
```

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

```
public function ubah_simpan($id, Request $req){
    $user = UserModel::find($id);

    $user->username = $req->username;
    $user->nama = $req->nama;
    $user->password = Hash::make('$req->username');
    $user->level_id = $req->level_id;

    $user->save();

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



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

Form Tambah Data User

[Kembali](#)

Username
Nama
Password
Level ID

Form Tambah Data User

[Kembali](#)

Username
Nama
Password
Level ID

12	JDoe	Jane Doe	1	Ubah Hapus
----	------	----------	---	--

Data Berhasil Diupdate

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

```
public function hapus($id){  
    $user = UserModel::find($id);  
    $user->delete();  
  
    return redirect('/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
4	customer-1	Pelanggan Pertama	4	Ubah Hapus
5	manager-dua	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager 56	Manager55	2	Ubah Hapus
11	manager 12	Manager11	2	Ubah Hapus
12	JDoe	Jane Doe	1	Ubah Hapus



Jurusan Teknologi Informasi Politeknik Negeri Malang
Jobsheet Minggu-4: Model & ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut

	<div><h2>Data User</h2><p>+ Tambah User</p><table><thead><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th><th>Aksi</th></tr></thead><tbody><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>4</td><td>customer-1</td><td>Pelanggan Pertama</td><td>4</td><td>Ubah Hapus</td></tr><tr><td>5</td><td>manager-dua</td><td>Manager 2</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>6</td><td>manager22</td><td>Manager Dua Dua</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>7</td><td>manager33</td><td>Manager Tiga Tiga</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>8</td><td>manager 56</td><td>Manager55</td><td>2</td><td>Ubah Hapus</td></tr><tr><td>11</td><td>manager 12</td><td>Manager11</td><td>2</td><td>Ubah Hapus</td></tr></tbody></table><h2>Data Berhasil Dihapus</h2></div>	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	4	customer-1	Pelanggan Pertama	4	Ubah Hapus	5	manager-dua	Manager 2	2	Ubah Hapus	6	manager22	Manager Dua Dua	2	Ubah Hapus	7	manager33	Manager Tiga Tiga	2	Ubah Hapus	8	manager 56	Manager55	2	Ubah Hapus	11	manager 12	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																																															
4	customer-1	Pelanggan Pertama	4	Ubah Hapus																																															
5	manager-dua	Manager 2	2	Ubah Hapus																																															
6	manager22	Manager Dua Dua	2	Ubah Hapus																																															
7	manager33	Manager Tiga Tiga	2	Ubah Hapus																																															
8	manager 56	Manager55	2	Ubah Hapus																																															
11	manager 12	Manager11	2	Ubah Hapus																																															
	<div><div>feat: add CRUD</div><div> iannstronaut pushed 1 commit • 2a53074...c7fb4c2 • 14 seconds ago</div></div>																																																		
	<pre>public function level(): BelongsTo{ return \$this->belongsTo(LevelModel::class); }</pre>																																																		
	<pre>public function index(){ \$user = UserModel::with('level')->get(); dd(\$user); }</pre>																																																		
	<pre>Illuminate\Database\Eloquent\Collection {#325 ▼ // app\Http\Controllers\UserController.php:21 #items: array:9 [▼ 0 => App\Models\UserModel {#332 ▶} 1 => App\Models\UserModel {#333 ▶} 2 => App\Models\UserModel {#334 ▶} 3 => App\Models\UserModel {#335 ▶} 4 => App\Models\UserModel {#336 ▶} 5 => App\Models\UserModel {#337 ▶} 6 => App\Models\UserModel {#338 ▶} 7 => App\Models\UserModel {#339 ▶} 8 => App\Models\UserModel {#340 ▶}] #escapeWhenCastingToString: false }</pre>																																																		
	<div><div>feat: add Relations UserModel</div><div> iannstronaut pushed 1 commit • c7fb4c2...b160f95 • 3 seconds ago</div></div>																																																		