

PEMROGRAMAN WEB LANJUT
JOBSHEET 4 - MODEL DAN ELOQUENT ORM



Disusun oleh:
Stefanus Ageng Budi Utomo (2241720126)

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN AJARAN 2023/2024

Praktikum 1 - \$fillable

1. Buka file model dengan nama UserModel.php dan tambahkan \$fillable seperti gambar di bawah ini

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user'; // Mendefinisiakan nama tabel yang digunakan oleh model ini
    protected $primaryKey = 'user_id'; // Mendefinisikan primary key dari tabel yang digunakan

    protected $fillable = ['level_id', 'username', 'nama', 'password'];
}
```

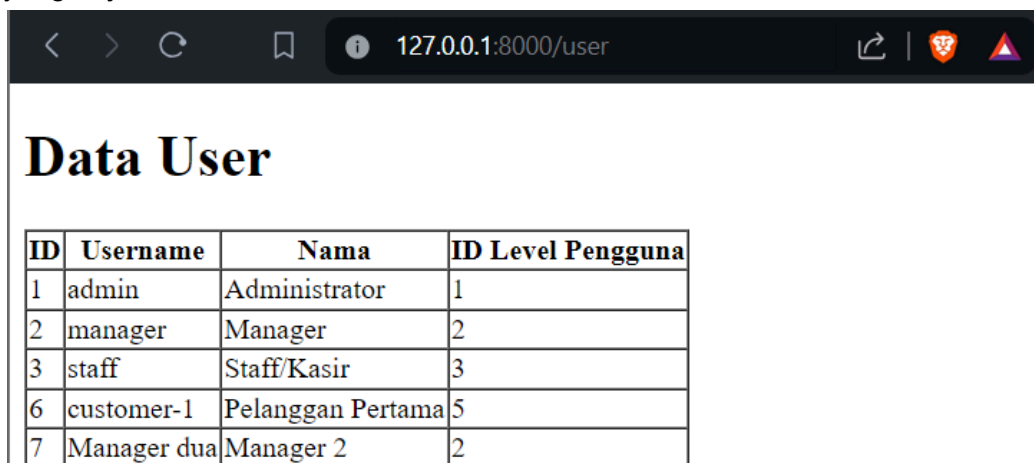
2. Buka file controller dengan nama UserController.php dan ubah script untuk menambahkan data baru seperti gambar di bawah ini

```
$data = [
    'level_id' => 2,
    'username' => 'Manager dua',
    'nama' => 'Manager 2',
    'password' => Hash::make('12345'),
];

UserModel::create($data);

$user = UserModel::all(); // ambil semua data dari tabel m_user
return view('user', ['data' => $user]);
```

3. Simpan kode program Langkah 1 dan 2, dan jalankan perintah web server. Kemudian jalankan link localhostPWL_POS/public/user pada browser dan amati apa yang terjadi



ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
6	customer-1	Pelanggan Pertama	5
7	Manager dua	Manager 2	2

- Data baru 'Manager dua' ditambahkan
4. Ubah file model UserModel.php seperti pada gambar di bawah ini pada bagian \$fillable

```
protected $fillable = ['level_id', 'username', 'nama'];
```

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

```
$data = [  
    'level_id' => 2,  
    'username' => 'Manager tiga',  
    'nama' => 'Manager 3',  
    'password' => Hash::make('12345'),  
];  
UserModel::create($data);
```

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

```
Illuminate \ Database \ QueryException  
PHP 8.3.0 10.47.0  
  
SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value  
  
INSERT INTO  
  `m_user` (  
    `level_id`,  
    `username`,  
    `nama`,  
    `updated_at`,  
    `created_at`  
  )  
VALUES  
  (  
    2,  
    Manager tiga,  
    Manager 3,  
    2024 -03 -13 00: 29: 27,  
    2024 -03 -13 00: 29: 27  
  )
```

- Terjadi error dikarenakan kolom password tidak terdaftar sebagai kolom untuk diisi

7. Laporkan hasil Praktikum-1 ini dan commit perubahan pada git

```
feat: add fillable  
AgengAbi committed now  
3a8a496
```

Praktik 2.1 Retrieving Single Models

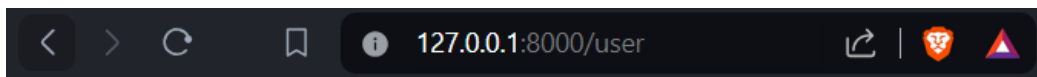
1. Buka file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::find(1);
        return view('user', ['data' => $user]);
    }
}
```

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

```
<table border="1" cellpadding="2" cellspacing="0">
    <tr>
        <th>Username</th>
        <th>Nama</th>
        <th>ID Level Pengguna</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>
        </tr>
    @endforeach --}}
    <tr>
        <td>{{ $data->user_id }}</td>
        <td>{{ $data->username }}</td>
        <td>{{ $data->nama }}</td>
        <td>{{ $data->level_id }}</td>
    </tr>
</table>
```

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



Data User

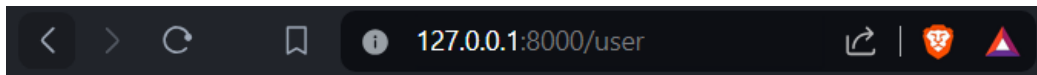
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

- Data yang keluar akan adalah data yang memiliki primary key 1. Jadi, method find() digunakan untuk mencari data dengan primary key yang dilempar pada argument method tersebut.

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::where('level_id', 1)->first();
        return view('user', ['data' => $user]);
    }
}
```

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



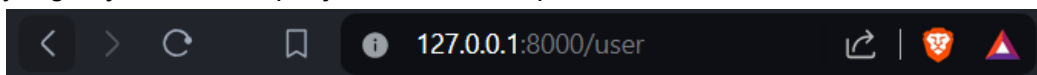
Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

- Hasilnya adalah menampilkan data berdasarkan query builder yaitu data yang memiliki level_id sama dengan 1 dan menampilkan data pertama yang ditemukan, jika tidak ditemukan maka akan bernilai null
- Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
class UserController extends Controller
{
    public function index()
    {
        // $user = UserModel::where('level_id', 1)->first();
        $user = UserModel::firstWhere('level_id', 1);
        return view('user', ['data' => $user]);
    }
}
```

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



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

- Hasilnya masih sama, karena potongan kode tersebut memiliki fungsi yang sama dengan potongan kode sebelumnya. dimana, mencari data dengan

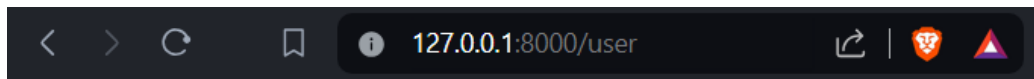
level_id yang sama dengan 1 dan mengembalikan nilai yang pertama kali ditemukan.

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::findOr(1, ['username', 'nama'], function () {
            abort(404);
        });

        return view('user', ['data' => $user]);
    }
}
```

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



Data User

ID	Username	Nama	ID Level Pengguna
	admin	Administrator	

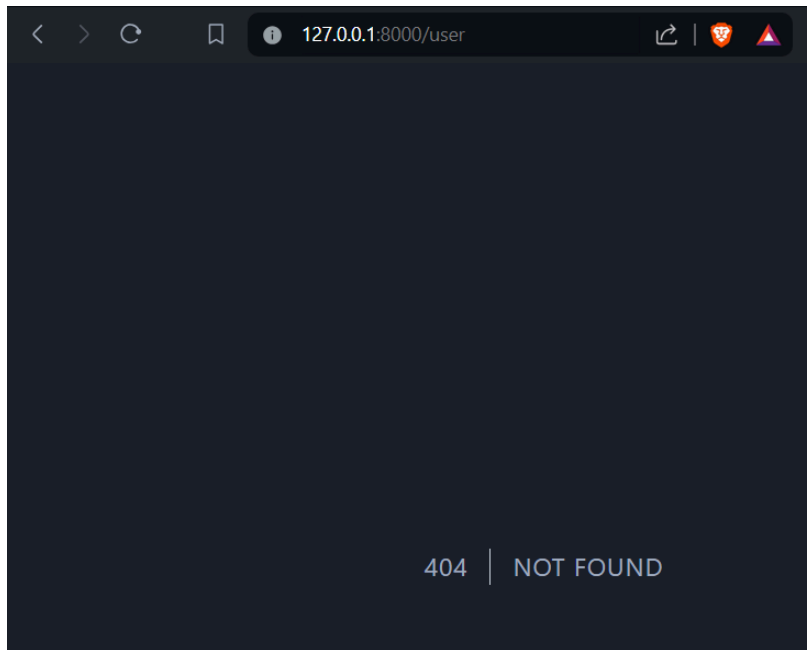
- Data yang tampil hanya sesuai dengan argumen yang dikirim ke parameter method findOr. Dimana yang ditulis hanya username dan nama saja.

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::findOr(20, ['username', 'nama'], function () {
            abort(404);
        });

        return view('user', ['data' => $user]);
    }
}
```

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



- Karena data dengan primary key 20 tidak ditemukan dan tidak valid
12. Laporkan hasil Praktikum-2.1 ini dan commit perubahan pada git.

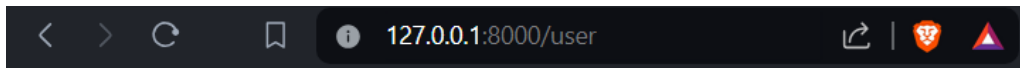


Pratikum 2.2 - Not Found Exceptions

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::findOrFail(1);
        return view('user', ['data' => $user]);
    }
}
```

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



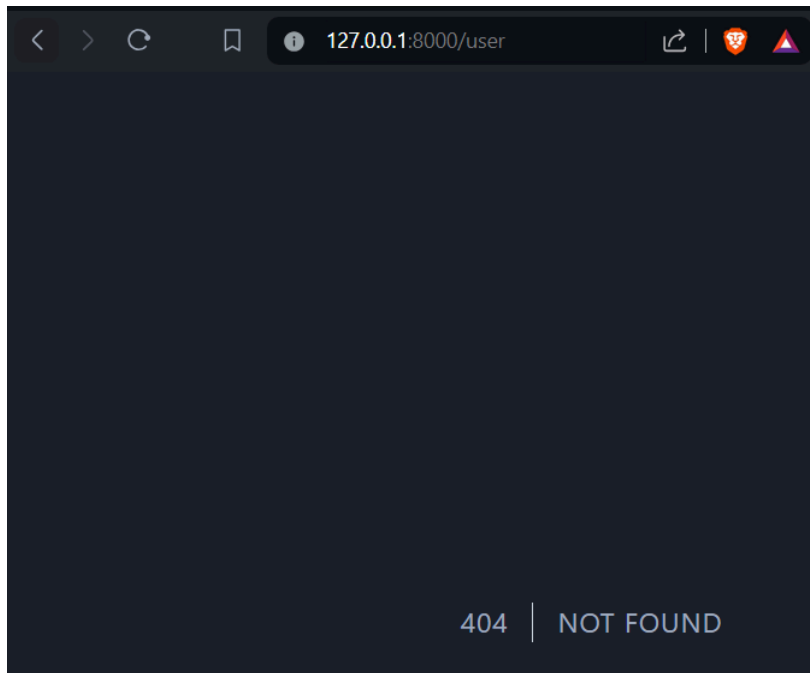
Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

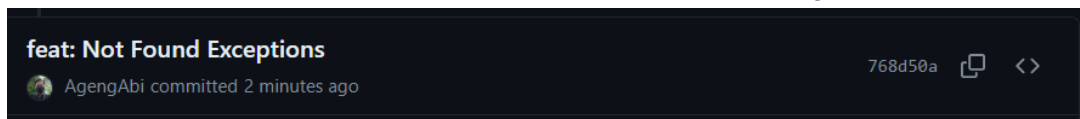
- method `findOrFail(1)` digunakan untuk mencari data dengan primary key yang sama dengan 1. jika data tersebut tidak ditemukan maka akan
3. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::where('username', 'manager9')->firstOrFail();
        return view('user', ['data' => $user]);
    }
}
```

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



- Menghasilkan halaman not found dikarenakan tidak menemukan data yang sesuai dengan query builder `where('username', 'manager9')`. Method `firstOrFail()` untuk mengambil record pertama yang sesuai dengan kondisi. Jika tidak ada record yang sesuai, `firstOrFail()` akan melemparkan exception `ModelNotFoundException`.
5. Laporkan hasil Praktikum-2.2 ini dan commit perubahan pada git.

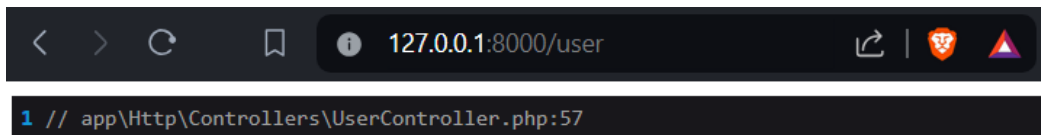


Praktikum 2.3 - Retrieving Aggregates

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::where('level_id', 1)->count();
        dd($user);
        return view('user', ['data' => $user]);
    }
}
```

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



- method dd() dalam PHP, itu akan mencetak nilai dari argumen yang Anda berikan ke layar dan menghentikan eksekusi skrip setelahnya. Jadi, jika melihat output seperti 1 // app\Http\Controllers\UserController.php:57, itu berarti bahwa pada baris ke-57 dari file UserController.php, ada perintah dd(1) yang menyebabkan output 1 ditampilkan dan eksekusi skrip berhenti setelahnya.
3. Buat agar jumlah script pada langkah 1 bisa tampil pada halaman browser, sebagai contoh bisa lihat gambar di bawah ini dan ubah script pada file view supaya bisa muncul datanya

Data User

Jumlah Pengguna
2

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

```
</head>

<body>
  <h1>Data User</h1>
  <table border="1" cellpadding="2" cellspacing="0">
    <tr>
      <th>Jumlah Pengguna</th>
    </tr>
    <tr>
      <td>{{ $data }}</td>
    </tr>
  </table>
</body>

</html>
```

4. Laporkan hasil Praktikum-2.3 ini dan commit perubahan pada git.

feat: Retreiving Aggregates



AgengAbi committed 1 minute ago

59461b1



Praktikum 2.4 - Retrieving or Creating Models

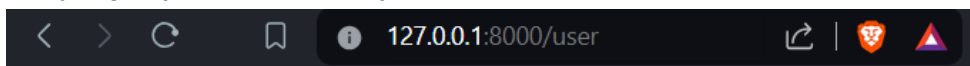
1. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager',
                'nama' => 'Manager',
            ]
        );
        return view('user', ['data' => $user]);
    }
}
```

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

```
<html lang="en">
<body>
<table border="1" cellpadding="2" cellspacing="0">
<tr>
<th>ID</th>
<th>Username</th>
<th>Nama</th>
<th>ID Level Pengguna</th>
</tr>
<tr>
<td>{{ $data->user_id }}</td>
<td>{{ $data->username }}</td>
<td>{{ $data->nama }}</td>
<td>{{ $data->level_id }}</td>
</tr>
</table>
</body>
```

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



Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

- firstOrCreate akan mencari data sesuai dengan argumen yang diberikan, sebagai contoh dari kode diatas, maka model akan mencari username 'manager', dan nama 'Manager' pada tabel m_user dan menampilkannya

melalui view. Namun jika tidak ditemukan maka data baru akan di masukkan ke database sesuai dengan kondisi yang diberikan.

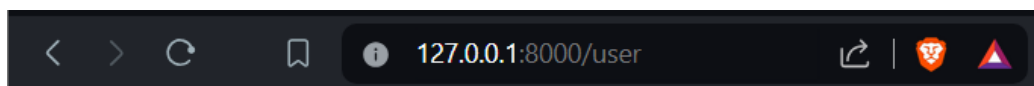
- Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager22',
                'nama' => 'Manager Dua Dua',
                'password' => Hash::make('12345'),
                'level_id' => 2
            ]
        );

        return view('user', ['data' => $user]);
    }
}
```

- 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

				user_id	level_id	username	nama	password
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	1	admin	Administrator	\$2y\$12\$ie9QoXoinIdBWzvlrOGn8ert
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	2	manager	Manager	\$2y\$12\$LyK5atJjNPxSe5pcVTYQzuu
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	3	staff	Staff/Kasir	\$2y\$12\$ZTzxTJuMIEvrRME6Z4A2p.e
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6szSUM3vQW9kj1xlwQ.z
<input type="checkbox"/>	 Edit	 Copy	 Delete	7	2	Manager dua	Manager 2	\$2y\$12\$QH9fATxoO6L0FtwgtW4nd.r



Data User

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

				user_id	level_id	username	nama	password
<input type="checkbox"/>	Edit	Copy	Delete	1	1	admin	Administrator	\$2y\$12\$ie9QoXoinIdBWzvlrOGn8ert
<input type="checkbox"/>	Edit	Copy	Delete	2	2	manager	Manager	\$2y\$12\$LyK5atJjNPxSe5pcVTYQzuu
<input type="checkbox"/>	Edit	Copy	Delete	3	3	staff	Staff/Kasir	\$2y\$12\$ZTzxTJuMIEvrRME6Z4A2p.e
<input type="checkbox"/>	Edit	Copy	Delete	6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6szSUM3vQW9kj1xlwQ.z
<input type="checkbox"/>	Edit	Copy	Delete	7	2	Manager dua	Manager 2	\$2y\$12\$QH9fATxoO6L0FtwgtW4nd.r
<input type="checkbox"/>	Edit	Copy	Delete	8	2	manager22	Manager Dua Dua	\$2y\$12\$wuYns1/XOffUT/3./WHTv

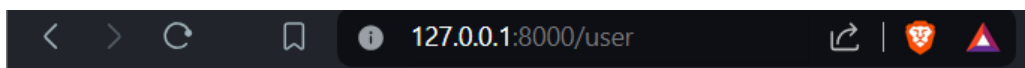
- Dari hasil percobaan tersebut terbukti bahwa method firstOrCreate akan memasukkan data baru sesuai kondisi jika tidak ditemukan.
6. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::firstOrCreate([
            'username' => 'manager',
            'nama' => 'Manager',
        ]);

        return view('user', ['data' => $user]);
    }
}
```

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

	user_id	level_id	username	nama	password
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	1	1	admin	Administrator	\$2y\$12\$ie9QoXoinIdBWzvlrOGn6
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	2	2	manager	Manager	\$2y\$12\$LyK5atJjNPxSe5pcVTYQ
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$ZTzxTJuMIEvrRME6Z4A:
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6szSUM3vQW9kj1xlv:
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	7	2	Manager dua	Manager 2	\$2y\$12\$QH9fATxoO6L0FtwgtW4i
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	8	2	manager22	Manager Dua Dua	\$2y\$12\$wuYns1/XOffUT/3./WHTv



Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

	user_id	level_id	username	nama	password
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	1	1	admin	Administrator	\$2y\$12\$ie9QoXoinIdBWzvlrOGn6
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	2	2	manager	Manager	\$2y\$12\$LyK5atJjNPxSe5pcVTYQ
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$ZTzxTJuMIEvrRME6Z4A:
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6szSUM3vQW9kj1xlv:
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	7	2	Manager dua	Manager 2	\$2y\$12\$QH9fATxoO6L0FtwgtW4i
<input type="checkbox"/> Edit <input type="copy"/> Copy <input type="delete"/> Delete	8	2	manager22	Manager Dua Dua	\$2y\$12\$wuYns1/XOffUT/3./WHTv

- Dari hasil diatas dapat dipastikan bahwa method firstOrCreate akan mencari data sesuai kondisi yang diberikan dan mengembalikannya ke view.

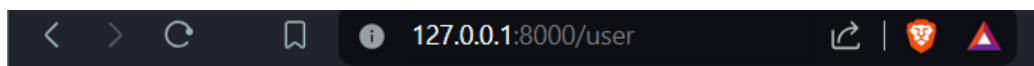
8. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::firstOrCreate([
            'username' => 'manager33',
            'nama' => 'Manager Tiga Tiga',
            'password' => Hash::make('12345'),
            'level_id' => 2
        ]);

        return view('user', ['data' => $user]);
    }
}
```

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

		user_id	level_id	username	nama	password
<input type="checkbox"/>	Edit Copy Delete	1	1	admin	Administrator	\$2y\$12\$ie9QoXoinIdBWzvlrOGn6
<input type="checkbox"/>	Edit Copy Delete	2	2	manager	Manager	\$2y\$12\$LyK5atJjNPxSe5pcVTYQ
<input type="checkbox"/>	Edit Copy Delete	3	3	staff	Staff/Kasir	\$2y\$12\$ZTzxTJuMIEvrRME6Z4A
<input type="checkbox"/>	Edit Copy Delete	6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6szSUM3vQW9kj1xlv
<input type="checkbox"/>	Edit Copy Delete	7	2	Manager dua	Manager 2	\$2y\$12\$QH9fATxoO6L0FtwgtW4i
<input type="checkbox"/>	Edit Copy Delete	8	2	manager22	Manager Dua Dua	\$2y\$12\$wuYns1/XOffUT/3./WHTv



Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

		user_id	level_id	username	nama	password
<input type="checkbox"/>	Edit Copy Delete	1	1	admin	Administrator	\$2y\$12\$ie9QoXoinIdBWzvlrOGn6
<input type="checkbox"/>	Edit Copy Delete	2	2	manager	Manager	\$2y\$12\$LyK5atJjNPxSe5pcVTYQ
<input type="checkbox"/>	Edit Copy Delete	3	3	staff	Staff/Kasir	\$2y\$12\$ZTzxTJuMIEvrRME6Z4A
<input type="checkbox"/>	Edit Copy Delete	6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6szSUM3vQW9kj1xlv
<input type="checkbox"/>	Edit Copy Delete	7	2	Manager dua	Manager 2	\$2y\$12\$QH9fATxoO6L0FtwgtW4i
<input type="checkbox"/>	Edit Copy Delete	8	2	manager22	Manager Dua Dua	\$2y\$12\$wuYns1/XOffUT/3./WHTv

- Dari percobaan method firstOrNew() akan mencari dan menampilkan data sesuai kondisi yang diberikan. Namun, jika tidak maka akan disiapkan modelnya untuk menambahkan data dan data tidak langsung tersimpan ke database.

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::firstOrNew([
            'username' => 'manager33',
            'nama' => 'Manager Tiga Tiga',
            'password' => Hash::make('12345'),
            'level_id' => 2
        ]);
        $user->save();

        return view('user', ['data' => $user]);
    }
}
```

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

	user_id	level_id	username	nama	password
<input type="checkbox"/> Edit Copy Delete	1	1	admin	Administrator	\$2y\$12\$ie9QoXoinIdBWzvlrOGn8
<input type="checkbox"/> Edit Copy Delete	2	2	manager	Manager	\$2y\$12\$LyK5atJjNPxSe5pcVTYQ
<input type="checkbox"/> Edit Copy Delete	3	3	staff	Staff/Kasir	\$2y\$12\$ZTzxTJuMIEvrRME6Z4A
<input type="checkbox"/> Edit Copy Delete	6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6szSUM3vQW9kj1xlw
<input type="checkbox"/> Edit Copy Delete	7	2	Manager dua	Manager 2	\$2y\$12\$QH9fATxoO6L0FtwgtW4i
<input type="checkbox"/> Edit Copy Delete	8	2	manager22	Manager Dua Dua	\$2y\$12\$wuYns1/XOffUT/3./WHTv

127.0.0.1:8000/user			
Data User			
ID	Username	Nama	ID Level Pengguna
9	manager33	Manager Tiga Tiga	2

<div><div><div></div><div></div><div></div></div></div>				user_id	level_id	username	nama	password
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	1	1	admin	Administrator	\$2y\$12\$ie9QoXoinIdBWzvlrOG
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	2	2	manager	Manager	\$2y\$12\$LyK5atJjNPxSe5pcVTY
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	3	3	staff	Staff/Kasir	\$2y\$12\$ZTzxTJuMIEvrRME6Z4
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	6	5	customer-1	Pelanggan Pertama	\$2y\$12\$Le6szSUm3vQW9kj1xI
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	7	2	Manager dua	Manager 2	\$2y\$12\$QH9fATxoO6L0FtwgtW
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	8	2	manager22	Manager Dua Dua	\$2y\$12\$wuYns1/XOfUT/3./WH
<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	<div><div><div></div><div></div><div></div></div></div>	9	2	manager33	Manager Tiga Tiga	\$2y\$12\$xoPuTUnSIILLVeQmPe

- Method `firstOrCreate()` jika tidak menemukan data yang sesuai kondisi maka akan menyiapkan sebuah model yang siap untuk di masukan ke database. ketika kita yakin ingin menyimpannya maka kita perlu menambahkan method `save()`.

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



Praktikum 2.5 - Attribute Changes

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::create([
            'username' => 'manager55',
            'nama' => 'Manager55',
            'password' => Hash::make('12345'),
            'level_id' => 2,
        ]);

        $user->username = 'manager56'; //merubah username

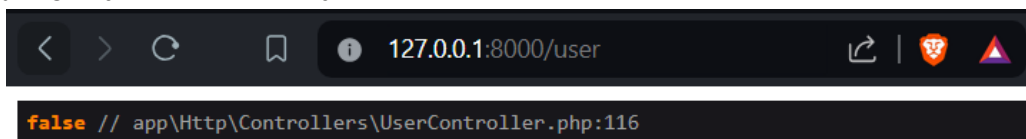
        $user->isDirty(); // true
        $user->isDirty('username'); //true
        $user->isDirty('nama'); //false
        $user->isDirty(['nama', 'username']); //true

        $user->isClean(); //false
        $user->isClean('username'); //true
        $user->isClean('nama'); //true
        $user->isClean(['nama', 'username']);

        $user->save();

        $user->isDirty(); //false
        $user->isClean(); //true
        dd($user->isDirty());
    }
}
```

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



```
false // app\Http\Controllers\UserController.php:116
```

- hasil false didapat pada 'dd(\$user->isDirty)' dikarenakan method isDirty tidak menemukan perubahan setelah model yang dirubah pada baris '\$user->username = "manager56"' sudah disimpan ke database dengan perintah '\$user->save()'.
3. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```

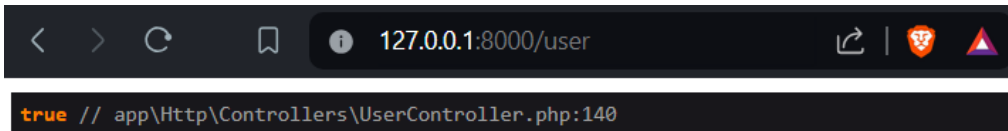
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::create(
            [
                'username' => 'manager11',
                'nama' => 'Manager11',
                'password' => Hash::make('12345'),
                'level_id' => 2,
            ]
        );
        $user->username = 'manager12';

        $user->save();

        $user->wasChanged(); //true
        $user->wasChanged('username'); //true
        $user->wasChanged(['username', 'level_id']); //true
        $user->wasChanged('nama'); //false
        dd($user->wasChanged(['nama', 'username'])); //true
    }
}

```

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



- hasil true tersebut dihasilkan oleh baris 'dd(\$user->wasChanged(['nama', 'username']));' dikarenakan pada kolom atribut 'username' model terdapat perubahan pada baris '\$user->username = "manager12";' meskipun sudah disimpan
5. Laporkan hasil Praktikum-2.5 ini dan commit perubahan pada git.



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

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

```
<html lang="en">
<body>
<h1>Data User</h1>
<a href="/user/tambah">+ Tambah User</a>
<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><a href="/user/ubah/{{ $d->user_id }}">Ubah</a> | <a href="/user/hapus/{{ $d->user_id }}">Hapus</a></td>
    </tr>
  @endforeach
</table>
```

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::all();
        return view('user', ['data' => $user]);
    }
}
```

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

127.0.0.1:8000/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
6	customer-1	Pelanggan Pertama	5	Ubah Hapus
7	Manager dua	Manager 2	2	Ubah Hapus
8	manager22	Manager Dua Dua	2	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus

- Perubahan pada userController mengembalikan semua data pada tabel m_user.
- Perubahan pada user.blade akan menampilkan data yang diterima dari controller lalu menampilkan aksi untuk ubah atau hapus data.

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

```
<html lang="en">
<body>
    <h1>Form Tambah Data User</h1>
    <form method="post" action="/user/tambah_simpan">
        {{ csrf_field() }}

        <label for="username">Username</label>
        <input type="text" name="username" placeholder="Masukan Username">
        <br>
        <label for="nama">Nama</label>
        <input type="text" name="nama" placeholder="Masukan Nama">
        <br>
        <label for="password">Password</label>
        <input type="password" name="password" placeholder="Masukan Password">
        <br>
        <label for="level_id">Level ID</label>
        <input type="number" name="level_id" placeholder="Masukan ID Level">
        <br>
        <input type="submit" value="Simpan" class="btn btn-success">
    </form>
</body>
```

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

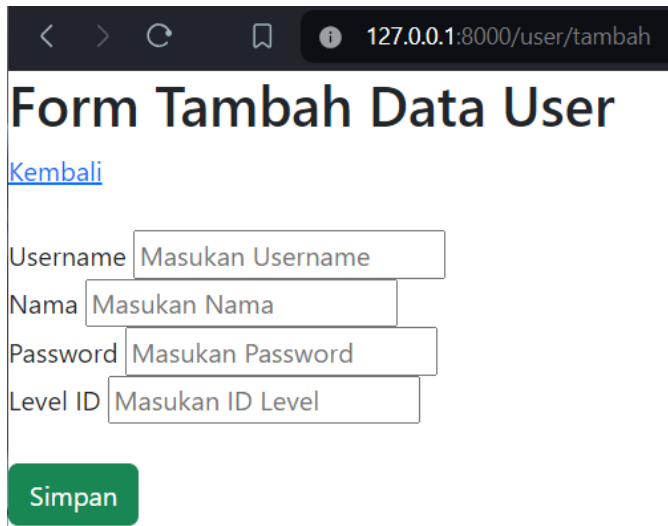
```
Route::get('user/tambah', [UserController::class, 'tambah']);
```

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::all();
        return view('user', ['data' => $user]);
    }

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

7. Simpan kode program Langkah 4 s/d 6. Kemudian jalankan pada browser dan klik link "+ Tambah User" amati apa yang terjadi dan beri penjelasan dalam laporan



- Tampil halaman baru yang terdapat form tambah data user.
8. Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini
9. Tambahkan script pada controller dengan nama file UserController.php. Tambahkan script dalam class dan buat method baru dengan nama tambah_simpan dan diletakkan di bawah method tambah seperti gambar di bawah ini

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

```
public function tambah_simpan(Request $request)
{
    UserModel::create([
        'username' => $request->username,
        'nama' => $request->nama,
        'password' => Hash::make($request->password),
        'level_id' => $request->level_id,
    ]);

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

10. Simpan kode program Langkah 8 dan 9. Kemudian jalankan link localhost:8000/user/tambah atau localhost/PWL_POS/public/user/tambah pada browser dan input formnya dan simpan, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan

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
6	customer-1	Pelanggan Pertama	5	Ubah Hapus
7	Manager dua	Manager 2	2	Ubah Hapus
8	manager22	Manager Dua Dua	2	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus

Form Tambah Data User

[Kembali](#)

Username

Nama

Password

Level ID

Simpan

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
6	customer-1	Pelanggan Pertama	5	Ubah Hapus
7	Manager dua	Manager 2	2	Ubah Hapus
8	manager22	Manager Dua Dua	2	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus
12	manager001	Mana ger	2	Ubah Hapus

- Data berhasil ditambahkan dan halaman diarahkan ke halaman data user

- 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

```

<html lang="en">
<body>
    <br><br>
    <form action="/user/ubah_simpan/{{ $data->user_id }}" method="POST">
        {{ csrf_field() }}
        {{ method_field('PUT') }}

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

```

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

```

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

```

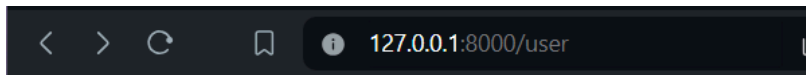
13. Tambahkan script pada controller dengan nama file UserController.php. Tambahkan script dalam class dan buat method baru dengan nama ubah dan diletakan di bawah method tambah_simpan seperti gambar di bawah ini

```

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

```

14. Simpan kode program Langkah 11 sd 13. Kemudian jalankan pada browser dan klik link "Ubah" amati apa yang terjadi dan beri penjelasan dalam laporan



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
6	customer-1	Pelanggan Pertama	5	Ubah Hapus
7	Manager dua	Manager 2	2	Ubah Hapus
8	manager22	Manager Dua Dua	2	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus
12	manager001	Mana ger	2	Ubah Hapus



Form Ubah Data User

[Kembali](#)

Username

Nama

Password

Level ID

[Ubah](#)

- Pada percobaan ini, kita bisa menambahkan data baru dengan menggunakan put pada form lalu meneruskannya ke controller dan controller akan meneruskannya ke database

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

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

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

```

public function ubah_simpan($id, Request $request)
{
    $user = UserModel::find($id);
    $user->username = $request->username;
    $user->nama = $request->nama;
    $user->password = Hash::make($request->password);
    $user->level_id = $request->level_id;
    $user->save();
    return redirect('/user');
}

```

17. Simpan kode program Langkah 15 dan 16. Kemudian jalankan link localhost:8000/user/ubah/1 atau localhost/PWL_POS/public/user/ubah/1 pada browser dan ubah input formnya dan klik tombol ubah, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan

127.0.0.1:8000/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
6	customer-1	Pelanggan Pertama	5	Ubah Hapus
7	Manager dua	Manager 2	2	Ubah Hapus
8	manager22	Manager Dua Dua	2	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus
12	manager001	Mana ger	2	Ubah Hapus

127.0.0.1:8000/user/ubah/1

Form Ubah Data User

[Kembali](#)

Username

Nama

Password

Level ID

Ubah

127.0.0.1:8000/user/ubah/1

Form Ubah Data User

[Kembali](#)

Username

Nama

Password

Level ID

Ubah

127.0.0.1:8000/user

Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	AdministratorSatu	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
6	customer-1	Pelanggan Pertama	5	Ubah Hapus
7	Manager dua	Manager 2	2	Ubah Hapus
8	manager22	Manager Dua Dua	2	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus
12	manager001	Mana ger	2	Ubah Hapus

- Pada percobaan ini, fungsi ubah data dengan method=put. Ditambahkan dengan cara mengklik link ubah pada suatu data lalu mengedit data sesuai keinginan lalu menekan tombol ubah lalu controller akan meneruskannya ke database untuk di update datanya.

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

127.0.0.1:8000/user

Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	AdministratorSatu	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
6	customer-1	Pelanggan Pertama	5	Ubah Hapus
7	Manager dua	Manager 2	2	Ubah Hapus
8	manager22	Manager Dua Dua	2	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus
12	manager001	Mana ger	2	Ubah Hapus

127.0.0.1:8000/user

Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	AdministratorSatu	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
6	customer-1	Pelanggan Pertama	5	Ubah Hapus
7	Manager dua	Manager 2	2	Ubah Hapus
8	manager22	Manager Dua Dua	2	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus

- Pada percobaan ini, ditambahkan fungsi hapus data dengan hanya mengklik link hapus pada tabel data. link hapus akan mengarahkan ke route yang meminta controller untuk menangkap id data lalu menghapusnya dari database.

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

feat: CRUD c44e22a

 AgengAbi committed 1 minute ago  

Praktikum 2.7 - Relationships

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

```
class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user'; // Mendefinisiakan nama tabel yang di
    protected $primaryKey = 'user_id'; // Mendefinisikan primary key dari t


    protected $fillable = ['level_id', 'username', 'nama', 'password'];

    public function level(): BelongsTo
    {
        return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
    }
}
```

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::with('level')->get();
        dd($user);
    }
}
```

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



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/user'. The browser content shows a JSON array of 9 user objects, each with fields like 'id', 'level_id', 'username', 'nama', and 'password'. The array is displayed in a dark-themed editor.

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::with('level')->get();
        // dd($user);
        return view('/user', ['data' => $user]);
    }
}
```

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

```

<body>
<a href="/user/tambah">+ Tambah User</a>
<table border="1" cellpadding="2" cellspacing="0">
  <tr>
    <th>ID</th>
    <th>Username</th>
    <th>Nama</th>
    <th>ID Level Pengguna</th>
    <th>Kode Level</th>
    <th>Nama Level</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>{{ $d->level->level_kode }}</td>
      <td>{{ $d->level->level_nama }}</td>
      <td><a href="/user/ubah/{{ $d->user_id }}">Ubah</a> | <a href="/user/hapus/{{ $d->user_id }}">Hapus</a></td>
    </tr>
  @endforeach
</table>

```

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

127.0.0.1:8000/user

Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Kode Level	Nama Level	Aksi
1	admin	AdministratorSatu	1	ADM	Administrator	Ubah Hapus
2	manager	Manager	2	MNG	Manager	Ubah Hapus
3	staff	Staff/Kasir	3	STF	Staff/Kasir	Ubah Hapus
6	customer-1	Pelanggan Pertama	5	CUS	Pelanggan	Ubah Hapus
7	Manager dua	Manager 2	2	MNG	Manager	Ubah Hapus
8	manager22	Manager Dua Dua	2	MNG	Manager	Ubah Hapus
9	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah Hapus
10	manager56	Manager55	2	MNG	Manager	Ubah Hapus
11	manager12	Manager11	2	MNG	Manager	Ubah Hapus

- Dengan menggunakan relasi model, kita bisa mengambil data dari model yang terhubung untuk kita gunakan. Seperti pada percobaan ini, dimana kita bisa memanggil data kolom 'Kode Level' dan 'Nama Level' yang dimana kolom tersebut dimiliki oleh tabel m_level bukan m_user.
7. Laporkan hasil Praktikum-2.7 ini dan commit perubahan pada git.

feat: Relationships

AgengAbi committed now

cdbbd65