LAPORAN PRAKTIKUM JOBSHEET 4

MATA KULIAH PEMROGRAMAN WEB LANJUT
MODEL DAN ELOQUENT ORM

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.

Di Susun Oleh :

Louise Nazarossa (2341760117)

SIB-2A/18

PROGRAM STUDI D4 SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

Praktikum 1: Fillable

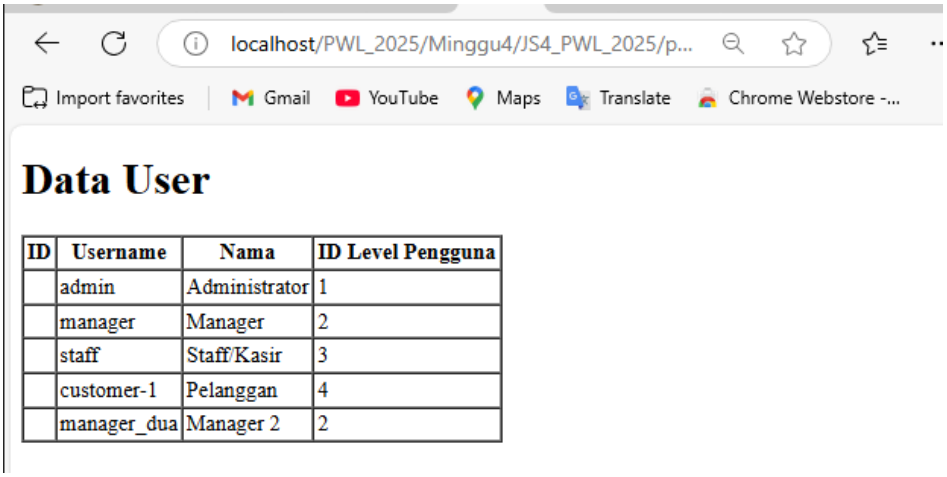
a) Membuka file model dengan nama usermodel.php dan menambahkan $fillable

```php
    /*
    * The atributes that are mass assiggnable.
    *
    *@var array
    */
    protected $fillable = ['level_id', 'username', 'nama', 'password'];
}
```

b) Membuka controller dengan nama UserController.php dan mengubah script untuk menambahkan data baru

```php
    // tambah data user dengan Fillable
    $data = [
        'level_id' => 2,
        'username' => 'manager_dua',
        'nama' => 'Manager 2',
        'password' => Hash::make('12345')
    ];
    UserModel::create($data); //membuat data baru
```

c) Memnjalankan perintah di web server pada browser



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

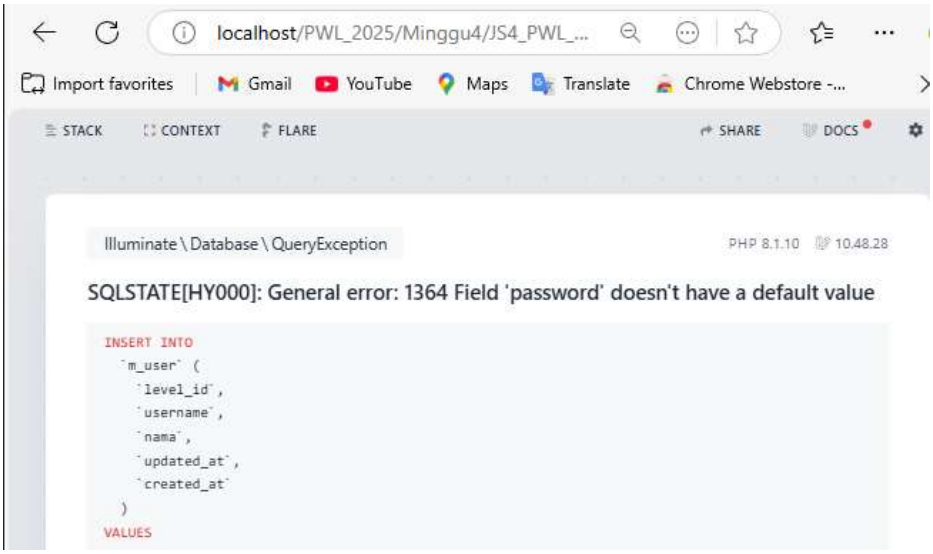
d) Mengubah file model UserModel.php

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

e) Mengubah file controller UserController.php pada bagian array $data

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

f) Menjalankan perintah pada web browser



Terjadi error karena password tidak ada dalam fillable, maka Laravel akan mengabaikan kolom tersebut dan ketika database mencoba menyisipkan aris tanpa kolom password, akan menyebabkan error

Praktikum 2: Retrieving Single Models

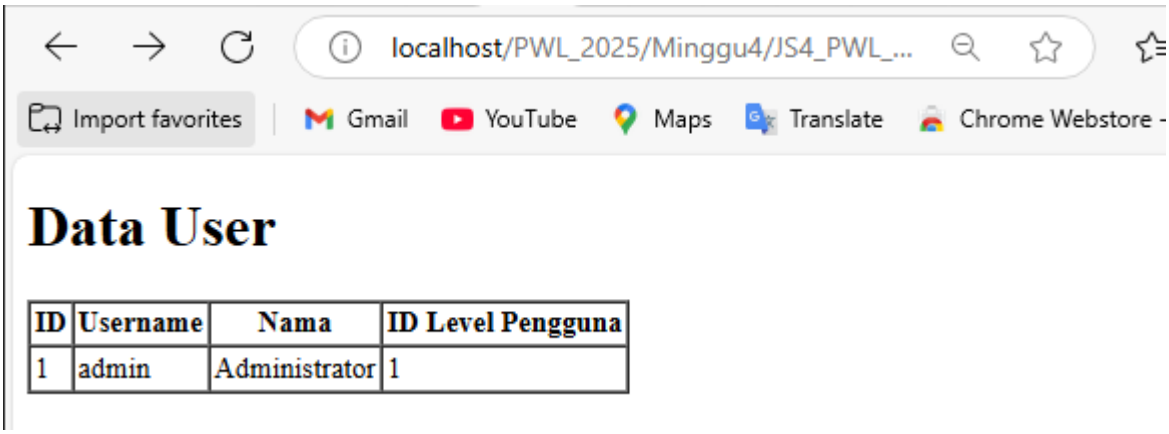
a) Membuka file controller dengan nama UserController.php dan mengubah script

```
11    public function index()
12    {
13        $user = UserModel::find(1);
14        return view('user', ['data' => $user]);
```

b) Mengubah user.blade.php

```
18            <td>{{ $data->id_user}}</td>
19            <td>{{ $data->username}}</td>
20            <td>{{ $data->nama}}</td>
21            <td>{{ $data->level_id}}</td>
22        </tr>
```

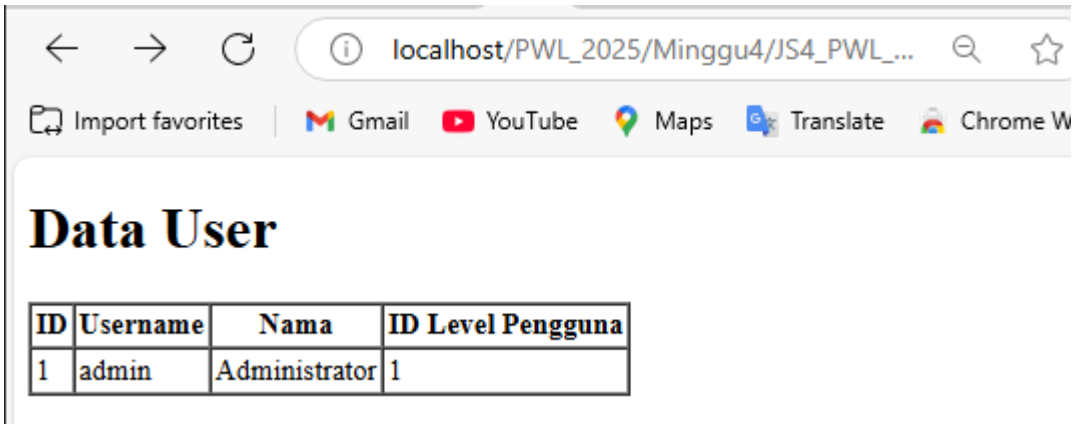
c) Menjalankan pada browser



d) Mengubah file UserController.php

```
13        $user = UserModel::where('level_id', 1)->first();
14        return view('user', ['data' => $user]);
15
```

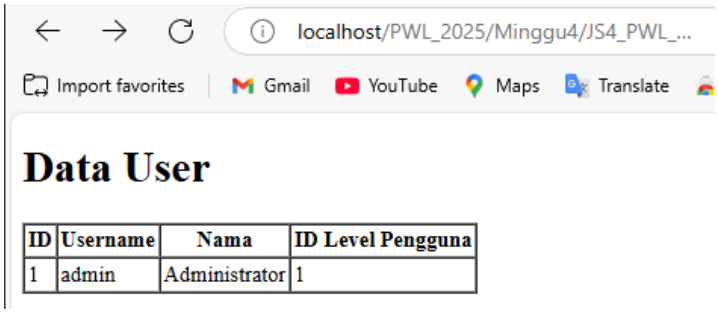
e) Menjalankan pada browser



f) Mengubah file controller dengan nama UserConroller

```
13        $user = UserModel::firstwhere('level_id', 1);
14        return view('user', ['data' => $user]);
15
```

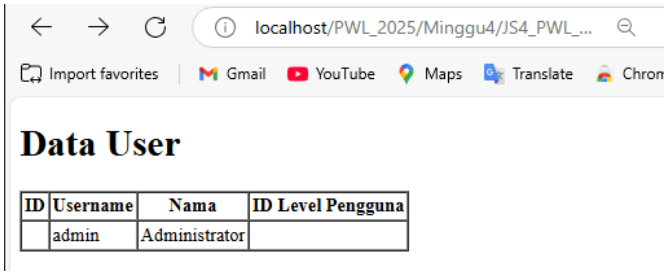
g) Menjalankan pada browser



h) Mengubah file UserController.php

```
13        $user = UserModel::findOr(1, ['username', 'nama'],
14            abort(404);
15        });
16        //$user = UserModel::firstwhere('level_id', 1);
17        return view('user', ['data' => $user]);
```

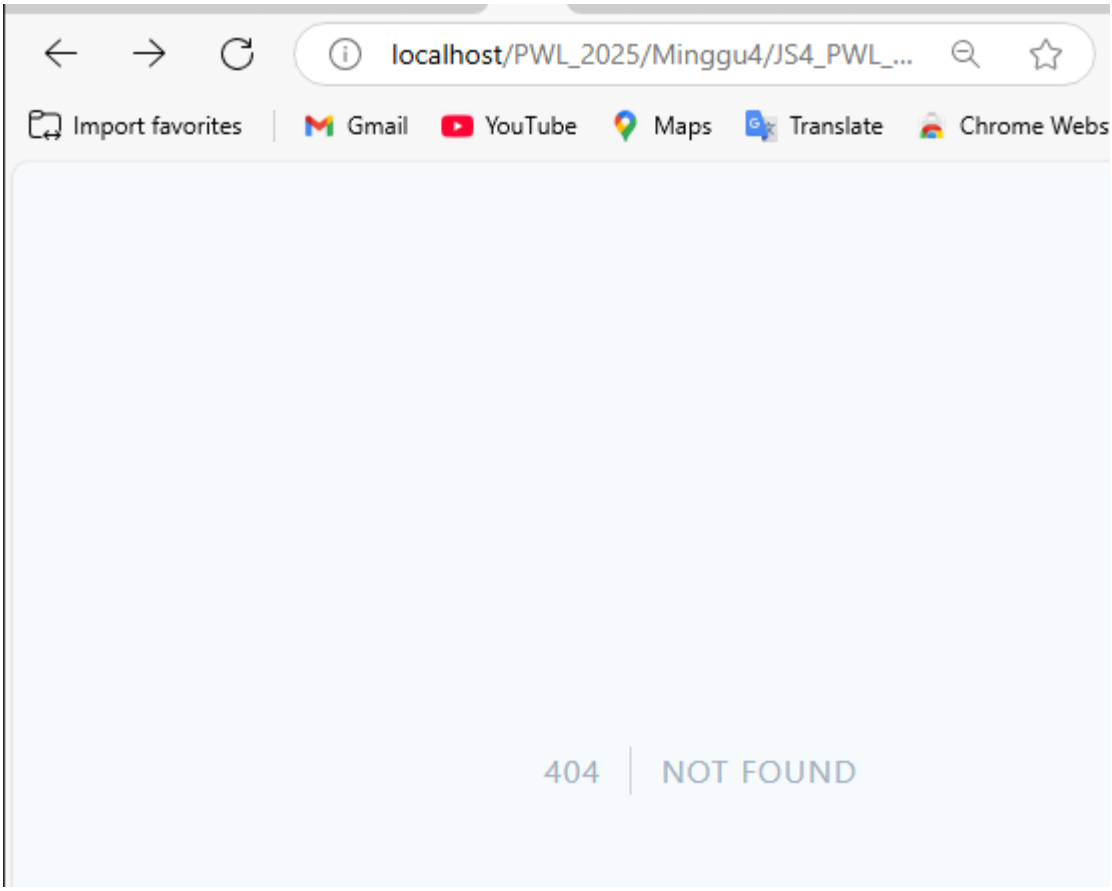
i) Menjalankan pada browser



j) Mengubah file UserController

```
13      $user = UserModel::findOr(20, ['username', 'nama'],
14          abort(404);
15      });
```

k) Menjalankan di browser



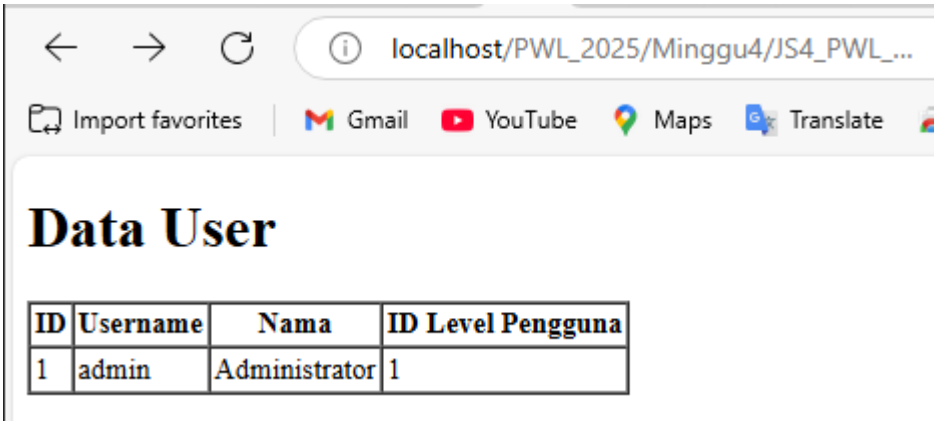
Hal ini dikarenakan id_user dengan id 20 tidak ada

Praktikum 2.2: Not Found Exceptions

a) Mengubah script UserController.php

```
11      public function index()
12      {
13          $user = UserModel::findOrFail(1);
14          return view('user', ['data' => $user]);
15
16      }
17  }
```

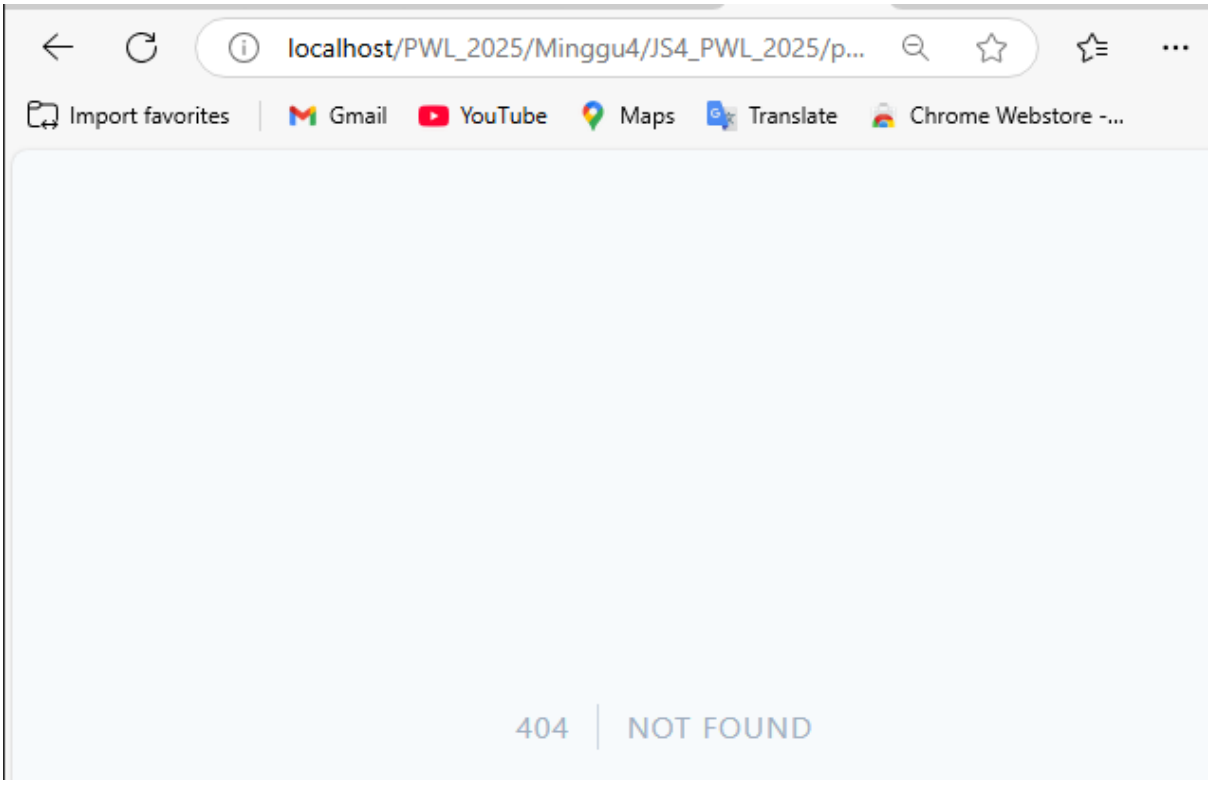
b) Menjalankan pada browser



c) Mengubah script UserController.php

```
11      public function index()
12      {
13          $user = UserModel::where('username', 'manager9')->firstOrFail();
14          return view('user', ['data' => $user]);
15
16      }
```

d) Menjalankan pada browser



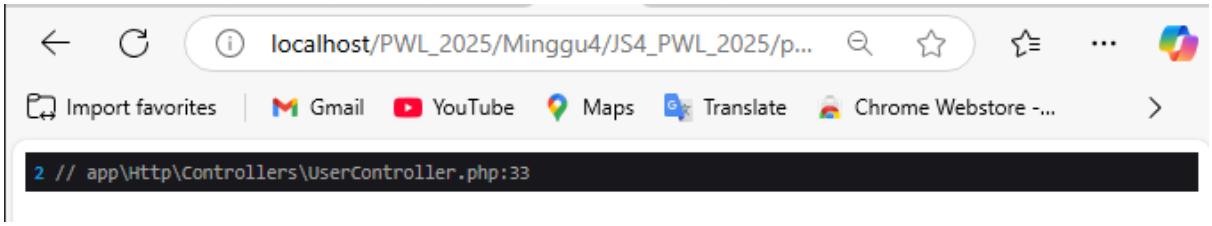
Hal ini dikarenakan data dengan username **manager9** tidak ditemukan dalam tabel m_user, jika ingin menemukan informasi terkait user tersebut, bisa diubah menjadi **manager**

Praktikum 2.3: Retreiving Aggregrates

a) Mengubah file controller dengan nama UserController.php dan mengubah scrip

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

b) Menjalankan pada web browser



Menunjukan bahwa jumlah data dengan level_id 2 adalah 2 dengan lokasi pemanggilan dd($user) berada dibaris 33 pada file UserController.php

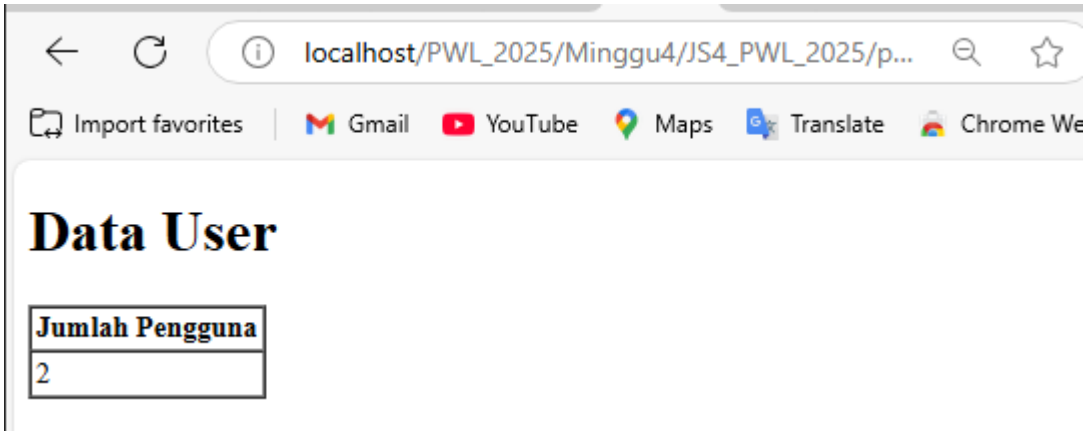
c) Membuat agar jumlah script pada Langkah 1 bisa tampil pada halaman browser

Mengubah script UserController.php dengan menghilangkan dd() agar kode mengembalikan view

```
32        $user = UserModel::where('level_id', 2) ->count();
33        //dd($user); //eksekusi berhenti di dd() dan mengabaikan tampi
34        return view('user', ['data' => $user]);
```

Mengubah script view

```
10        <table border="1" cellpadding="2" cellspacing="0">
11            <tr>
12                <!-- <th>ID</th>
13                <th>Username</th>
14                <th>Nama</th>
15                <th>ID Level Pengguna</th> -->
16                <th>Jumlah Pengguna</th>
17            </tr>
18            <tr>
19                {{-- <td>{{ $data->id_user}}</td>
20                <td>{{ $data->username}}</td>
21                <td>{{ $data->nama}}</td>
22                <td>{{ $data->level_id}}</td> --}}
23                <td>{{$data}}</td>
24            </tr>
25        </table>
26    </body>
```



Praktikum 2.4: Retrieving or Creating Models

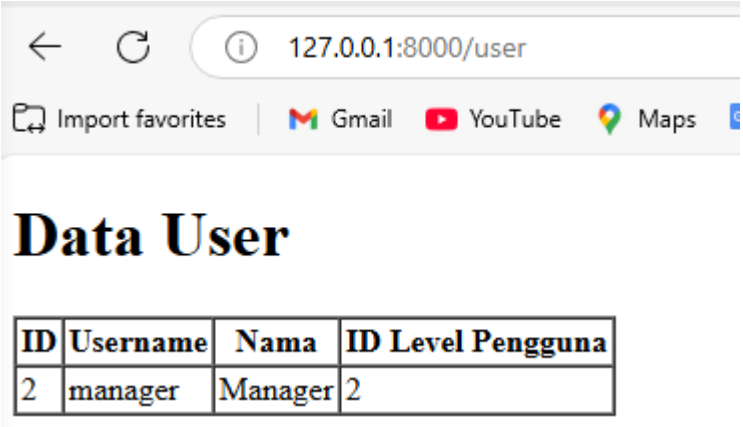
a) Mengubah file controller dengan nama UserController.php dan mengubah script

```
35          $user = UserModel::firstOrCreate(
36              [
37                  'username' => 'manager',
38                  'nama' => ' Manager',
39              ],
40          );
41          return view('user', ['data' => $user]);
```

b) Mengubah file view dengan nama user.blade.php

```
19                  <td>{{ $data->id_user}}</td>
20                  <td>{{ $data->username}}</td>
21                  <td>{{ $data->nama}}</td>
22                  <td>{{ $data->level_id}}</td>
```

c) Menjalankan kode pada browser

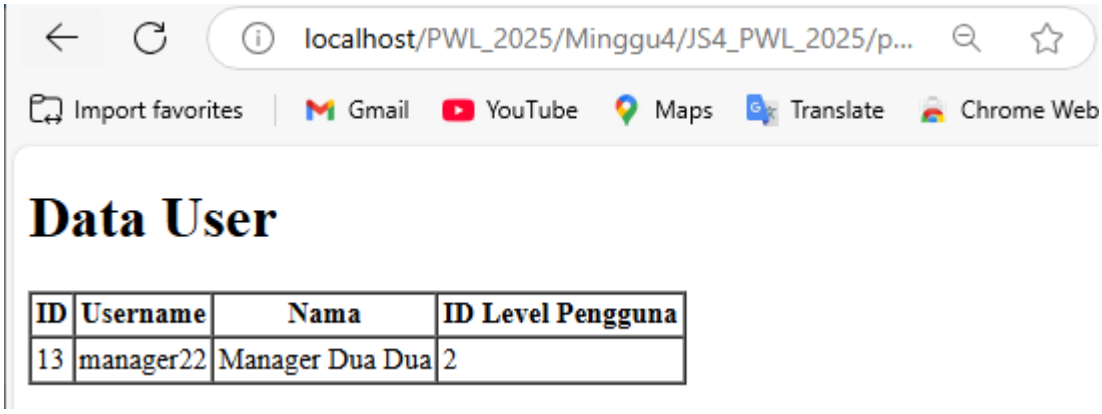


Mencari user dengan username = ‘manager’ dan nama = ‘Manager’ jika ditemukan maka akan mengembalikan data tersebut, jika tidak ditemukan maka akan membuat user baru dengan nama = ‘Manager’ dan nama = ‘Manager’ lalu disimpan pada database. Karena username ‘manager’dan nama’Manager’ telah ada pada database maka nilai akan dikembalikan

d) Mengubah file UserController.php

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

    }
```

e) Menjalankan pada browser dan mengecek pada PphpMyadmin pada tabel m_user



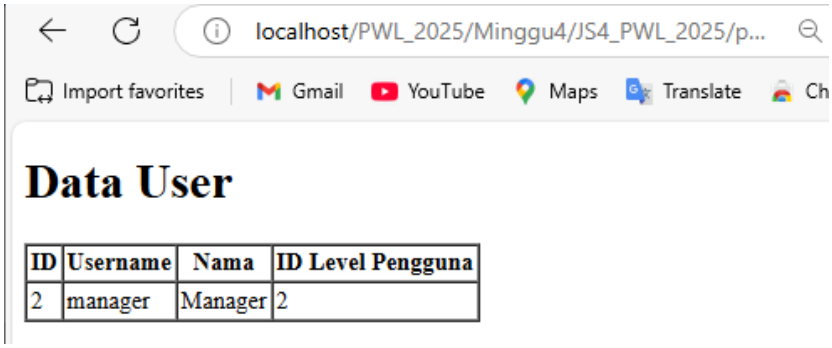
←T→				id_user	level_id	username	nama
	Edit	Copy	Delete	1	1	admin	Administrator
	Edit	Copy	Delete	2	2	manager	Manager
	Edit	Copy	Delete	3	3	staff	Staff/Kasir
	Edit	Copy	Delete	4	4	customer-1	Pelanggan
	Edit	Copy	Delete	12	2	manager_dua	Manager 2
	Edit	Copy	Delete	13	2	manager22	Manager Dua Dua

Data dengan username ‘manager22’ dan nama ‘Manager Dua Dua’ tidak ditemukan pada database tabel m_user, maka Laravel akan membuat data baru dengan username ‘manager22’ dan nama ‘Manager Dua Dua’ pada data base

f) Mengubah file UserController

```php
        $user = UserModel::firstOrNew(
            [
                'username' => 'manager',
                'nama' => 'Manager',

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

g) Menjalankan pada browser

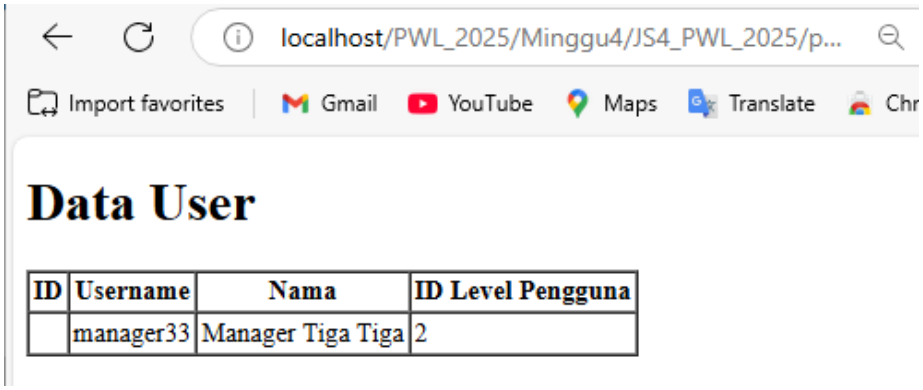


Laravel akan mencari data dengan username = 'manager' dan nama = 'Manager' di dalam tabel m_user. Jika ditemukan, Laravel akan mengembalikan objek model tersebut tanpa menyimpan perubahan baru ke dalam database.

h) Mengubah file UserController

```
43
44          $user = UserModel::firstOrNew(
45              [
46                  'username' => 'manager33',
47                  'nama' => 'Manager Tiga Tiga',
48                  'password' => Hash::make('12345'),
49                  'level_id' => 2
50              ],
51          );
52
```

i) Menjalankan pada browser dan mengecek pada PhpMyAdmin pada m_user



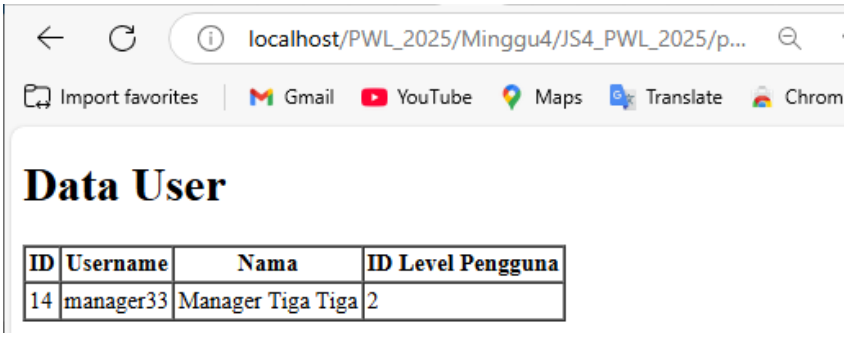
←T→	id_user	level_id	username	nama
Edit Copy Delete	1	1	admin	Administrator
Edit Copy Delete	2	2	manager	Manager
Edit Copy Delete	3	3	staff	Staff/Kasir
Edit Copy Delete	4	4	customer-1	Pelanggan
Edit Copy Delete	12	2	manager_dua	Manager 2
Edit Copy Delete	13	2	manager22	Manager Dua Dua

Laravel akan mencari data pada tabel m_user dengan username ‘manager’ dan nama ‘Manager’. Jika ditemukan maka akan mengembalikan nilai, namun jika tidak ada, Laravel akan membuat instance baru teteapi tidak langsung menyimpan pada databse berbeda dengan firstOrCreate yang langsung menyimpan pada database jika data tidak ditemukan

j) Mengubah file controller UserController.php

```
44        $user = UserModel::firstOrNew(
45            [
46                'username' => 'manager33',
47                'nama' => 'Manager Tiga Tiga',
48                'password' => Hash::make('12345'),
49                'level_id' => 2
50            ],
51        );
52        $user->save();
53        return view('user', ['data' => $user]);
```

k) Menjalankan pada browser dan melihat pada PhpMyAdmin pada tabel m_user



←T→	id_user	level_id	username	nama
Edit Copy Delete	1	1	admin	Administrator
Edit Copy Delete	2	2	manager	Manager
Edit Copy Delete	3	3	staff	Staff/Kasir
Edit Copy Delete	4	4	customer-1	Pelanggan
Edit Copy Delete	12	2	manager_dua	Manager 2
Edit Copy Delete	13	2	manager22	Manager Dua Dua
Edit Copy Delete	14	2	manager33	Manager Tiga Tiga

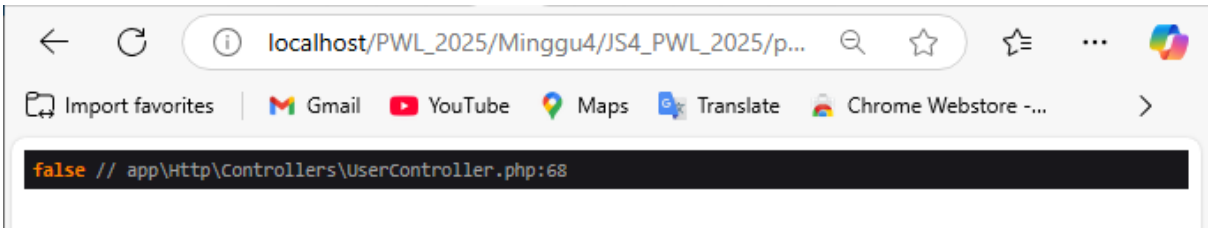
Perbedaan dengan Langkah di sebelumnya dimana Laravel tidak akan menyimpan data ke database saat membuat model baru, ketika ditambahkan kode $user->save(); data akan benar-benar disimpan pada tadabase di tabel m_user

Praktikum 2.5: Attribute Change

a.) Mengubah file controller UserConntroller.php

```php
        );
        $user->username = 'manager56';

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

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

        $user->save();

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

b.) Menjalankan pada crowser



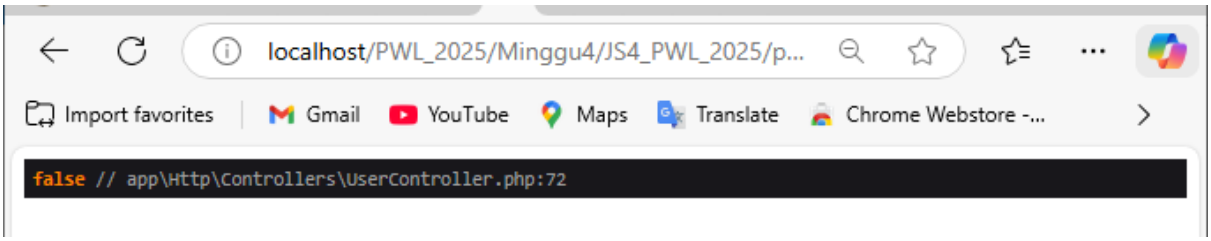
Jika username ‘manager55’ sudah ada pada database maka data akan dikembalikan, jika tidak ada Laravel akan membuat instance baru dengan data yang diisi namum Belum tersimpan pada database

Username kemudian diubah menjadi manager56. Yang membuat isDirty(‘username’) bernilai true karena ada perubahan, seteah pemanggilan save(), perubahan akan tersimpan di database, dan isDirty()menjadi false, menandakan tidak ad perubahan lebih lanjut. Pada akhirnya, dd($user->isDirty()) mencetak false karena Semua perubahan sudah tersimpan.

c.) Mengubah file controller dengan nama UserController.php

```php
        $user = UserModel::firstOrNew(
            [
                '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
    }
}
```

d.) Menjalankan pada browser



Kode ini membuat user baru dengan username 'manager11', lalu mengubahnya menjadi 'manager12' sebelum menyimpannya kembali. wasChanged('username') mengembalikan true karena ada perubahan, sementara wasChanged('nama') mengembalikan false. dd($user->wasChanged(['username', 'level_id'])) mencetak true karena username berubah, meskipun level_id tetap.

Praktikum 2.6: Create, Read, Upload, Delete (CRUD)

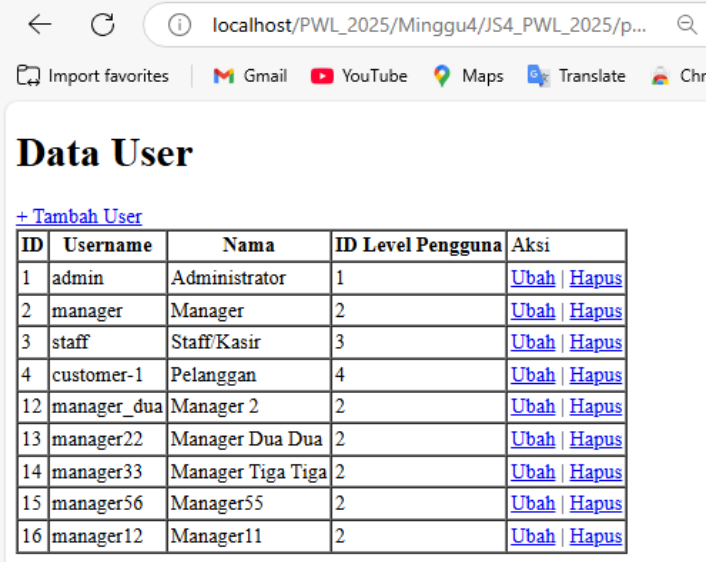
a) Mengubah script pada file user.blade.php

```
12          <th>ID</th>
13          <th>Username</th>
14          <th>Nama</th>
15          <th>ID Level Pengguna</th>
16          <td>Aksi</td>
17      </tr>
18      @foreach ($data as $d)
19
20      <tr>
21          <td>{{ $d->id_user}}</td>
22          <td>{{ $d->username}}</td>
23          <td>{{ $d->nama}}</td>
24          <td>{{ $d->level_id}}</td>
25          <td><a href="/user/ubah{{ $d->user_id}}">Ubah</a> |
26              <a href="/user/hapus{{ $d->user_id}}">Hapus</a></td>
27      </tr>
28      @endforeach
29  </table>
```

b) Mengubah sript pada file UserController.php

```
74          $user = UserModel::all();
75          return view('user', ['data' => $user]);
76
77      }
```

c) Menjalankan kode pada *browser*



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	4	Ubah | Hapus
12	manager_dua	Manager 2	2	Ubah | Hapus
13	manager22	Manager Dua Dua	2	Ubah | Hapus
14	manager33	Manager Tiga Tiga	2	Ubah | Hapus
15	manager56	Manager55	2	Ubah | Hapus
16	manager12	Manager11	2	Ubah | Hapus

Kode UserController.php mengambil semua data dari tabel m_user menggunakan UserModel::all() dan mengirimkannya ke user.blade.php. Di dalam view, data ditampilkan dalam bentuk tabel dengan perulangan @foreach ($data as $d), menampilkan ID, Username, Nama, dan ID Level Pengguna. Selain itu, terdapat tombol Ubah dan Hapus yang mengarah ke /user/ubah/{user_id} dan /user/hapus/{user_id}. Jika database memiliki data, tabel akan terisi sesuai dengan isinya, sedangkan jika kosong, hanya header tabel yang muncul tanpa isi.

d) Membuat create atau tambahan data user dengan cara membuat file baru pada view dengan nama user_tambah.blade.php

```
8   <body>
9       <h1>Form Tambah Data User</h1>
10      <form method="post" action="/user/tambah_simpan">
11
12          {{ csrf-field()}}
13
14          <label>Username</label>
15          <input type="text" name="username" placeholder="Masukan Username">
16          <br>
17          <label>Nama</label>
18          <input type="text" name="nama" placeholder="Masukan Nama">
19          <br>
20          <label>Password</label>
21          <input type="password" name="password" placeholder="Masukan Password">
22          <br>
23          <label>Level ID</label>
24          <input type="number" name="level_id" placeholder="Masukan ID Level">
25          <br><br>
26          <input type="submit" class="btn btn-success" value="Simpan">
27
28      </form>
29  </body>
30  </html>
```

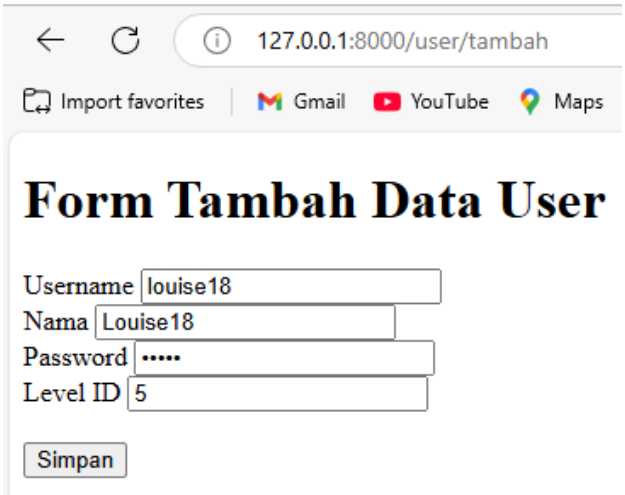
e) Menambahkan script pada file web.php

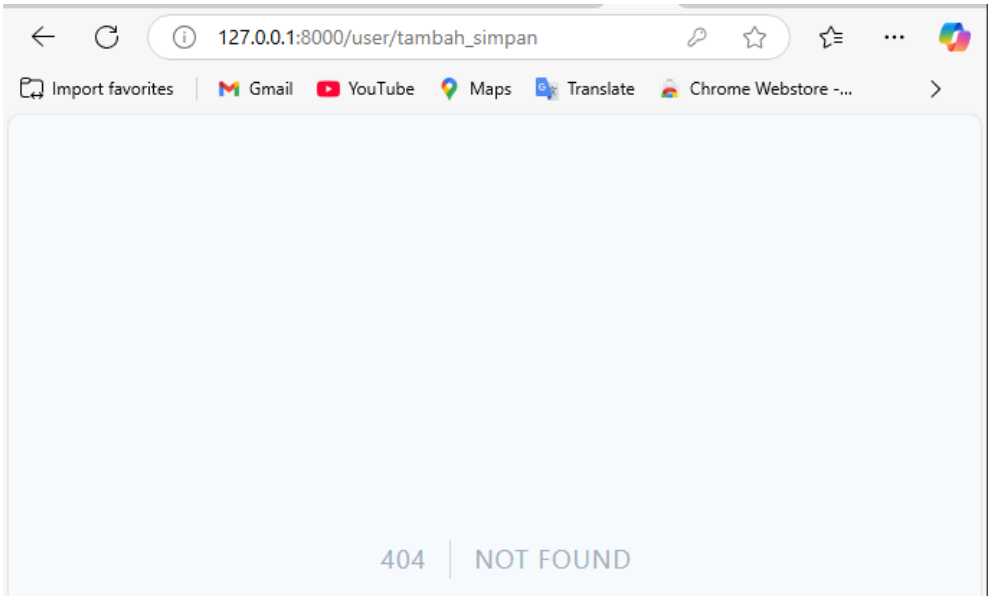
```
15  Route::get('/user/tambah', [UserController::class, 'tambah']);
```

f) Menambahkan script pada controller UserController.php, Serta menambahkan membuat method baru dengan nama tambah dan diletakan dibawah method index

```
76      }
77      public function tambah(){
78          return view('user_tambah');
79      }
```

g) Menjalankan pada browser dan klik +TambahUser





Ketika link +Tambah User di klik maka akan menuju pada /user/tambah namun ketika akan mengklik simpan terjadi not found. Hal ini dikarenakan fungsi /user/tambah_simpan Belum ada pada UserController.php

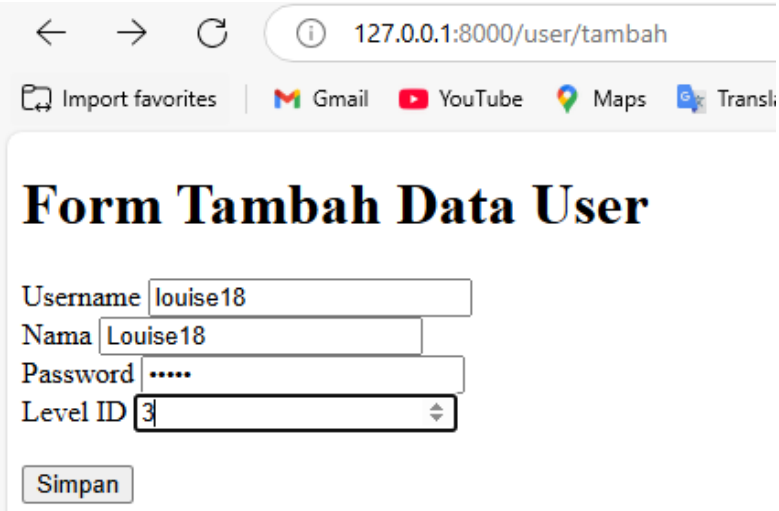
h) Menambahkan script pada routes dengan nama file web.php

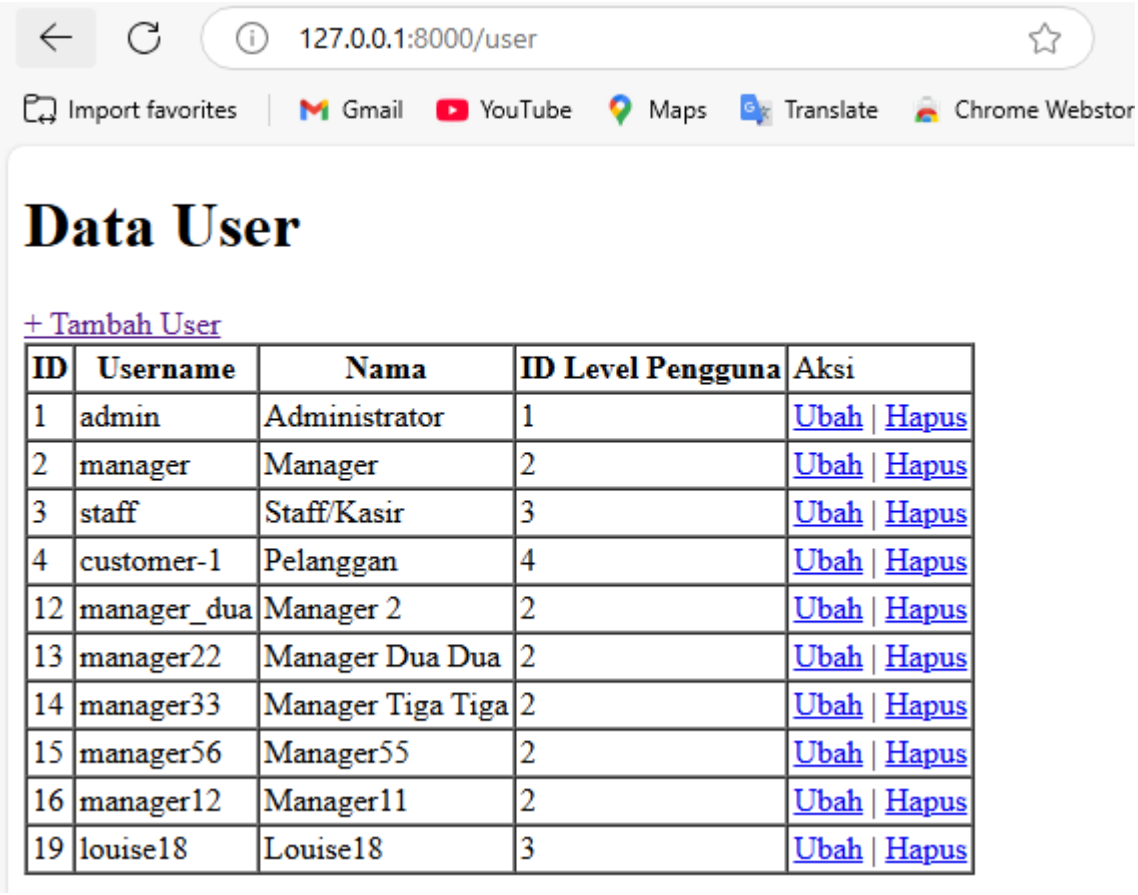
```
16  Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);
```

i) Menambahkan script pada file UserController.php

```
80      public function tambah_simpan(Request $request){
81          UserModel::create([
82              'username' => $request->username,
83              'nama' => $request->nama,
84              'password' => Hash::make('$request->password'),
85              'level_id' => $request->level_id
86          ]);
87          return redirect('/user');
88      }
89  }
```

j) Menjalankan pada browser





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	4	Ubah | Hapus
12	manager_dua	Manager 2	2	Ubah | Hapus
13	manager22	Manager Dua Dua	2	Ubah | Hapus
14	manager33	Manager Tiga Tiga	2	Ubah | Hapus
15	manager56	Manager55	2	Ubah | Hapus
16	manager12	Manager11	2	Ubah | Hapus
19	louise18	Louise18	3	Ubah | Hapus

Setelah penambahan fungsi tambah_simpan pada UserController, data yang akan ditambahkan akan langsung tersimpan di database setelah mengeklik simpan

k) Membuat update atau ubah data user dengan cara membuat file baru pada view dengan nama user_ubah.blade.php

```
8   <body>
9       <h1>Form Ubah Data User</h1>
10      <a href="/user">Kembali</a>
11      <br><br>
12
13      <form method="post" action="/user/ubah_simpan/{{ $data->user_id }}">
14
15          {{ csrf_field() }}
16          {{ method_field('PUT') }}
17
18          <label>Username</label>
19          <input type="text" name="username" placeholder="Masukan Username" value="{{ $data->username }}">
20          <br>
21          <label>Nama</label>
22          <input type="text" name="nama" placeholder="Masukan Nama" value="{{ $data->username }}">
23          <br>
24          <label>Password</label>
25          <input type="password" name="password" placeholder="Masukan Password" value="{{ $data->password }}">
26          <br>
27          <label>Level ID</label>
28          <input type="number" name="level_id" placeholder="Masukan ID Level" value="{{ $data->level_id }}">
29          <br><br>
30          <input type="submit" class="btn btn-success" value="Ubah">
31
32      </form>
33  </body>
34  </html>
```

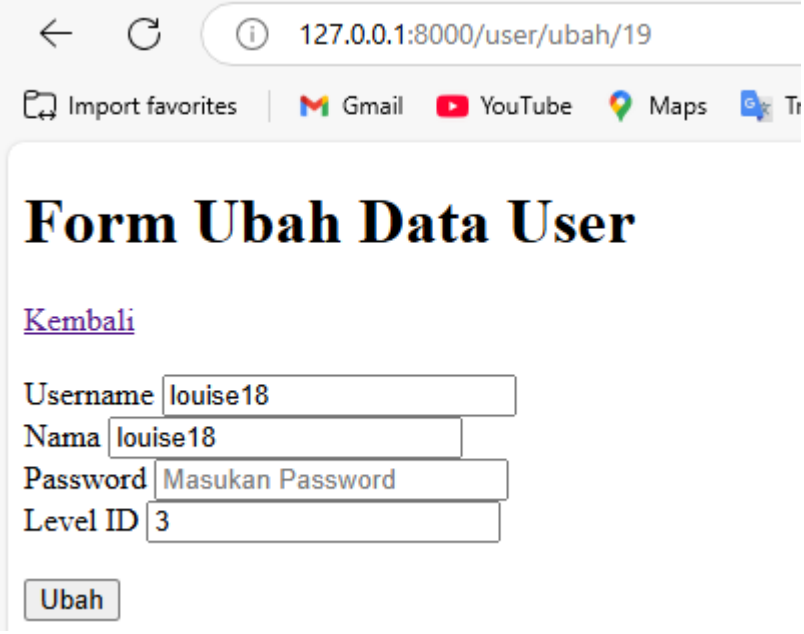
l) Menambahkan script pada routes dengan nama web.php

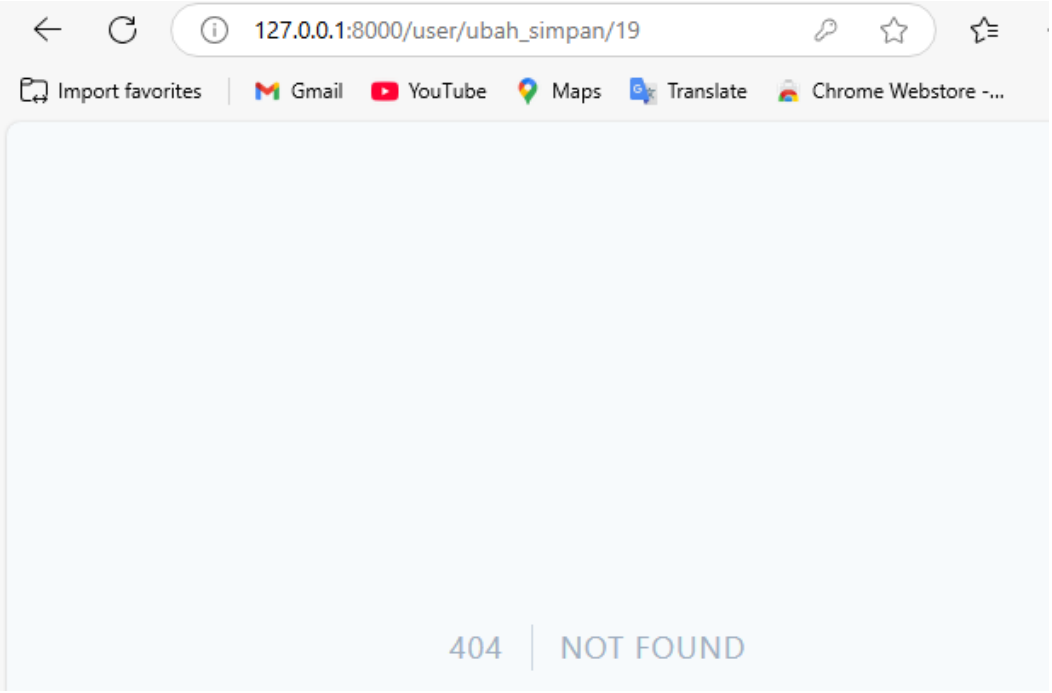
```
Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);
```

m) Menambahkan script pada UserController.php

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

n) Menjalankan di browser dan mengklik link ubah





Akan menuju pada halaman ubah, namun ketika data disimpan menghasilkan not found, hal ini dikarenakan ubah_simpan Belum ada

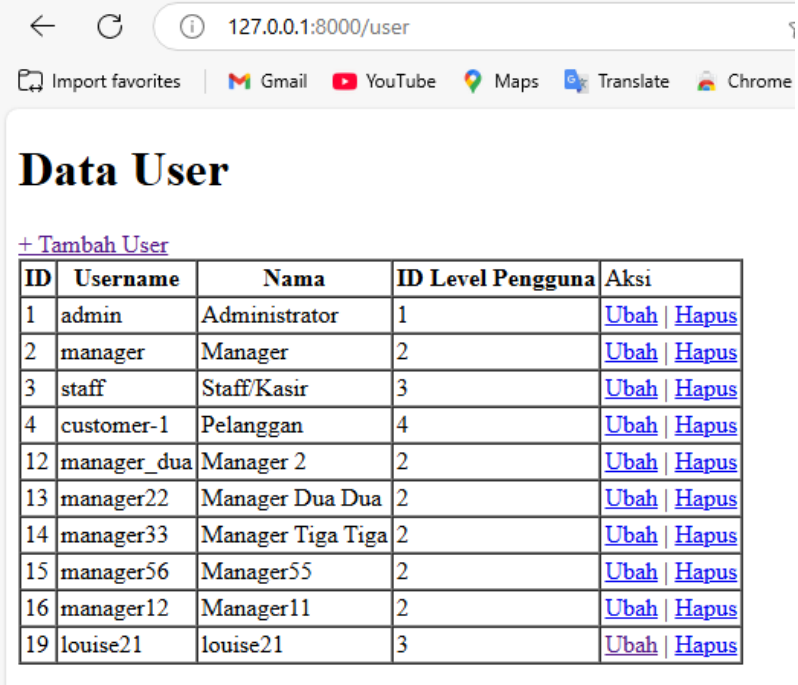
o) Menambahkan script pada file web.php

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

p) Menambahkan script pada UserController.php

```
93      public function ubah_simpan($id, Request $request){
94          $user = UserModel::find($id);
95
96          $user->username = $request->username;
97          $user->nama = $request->nama;
98          $user->password = Hash::make('$request->password');
99          $user->level_id = $request->level_id;
100
101         $user->save();
102         return redirect('/user');
103     }
104 }
```

q) Menjalankan pada web



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	4	Ubah | Hapus
12	manager_dua	Manager 2	2	Ubah | Hapus
13	manager22	Manager Dua Dua	2	Ubah | Hapus
14	manager33	Manager Tiga Tiga	2	Ubah | Hapus
15	manager56	Manager55	2	Ubah | Hapus
16	manager12	Manager11	2	Ubah | Hapus
19	louise21	louise21	3	Ubah | Hapus

Mode ubah berhasil diterapkan, bisa dilihat bahwa username berubah yang awalnya louise18 menjadi louise21

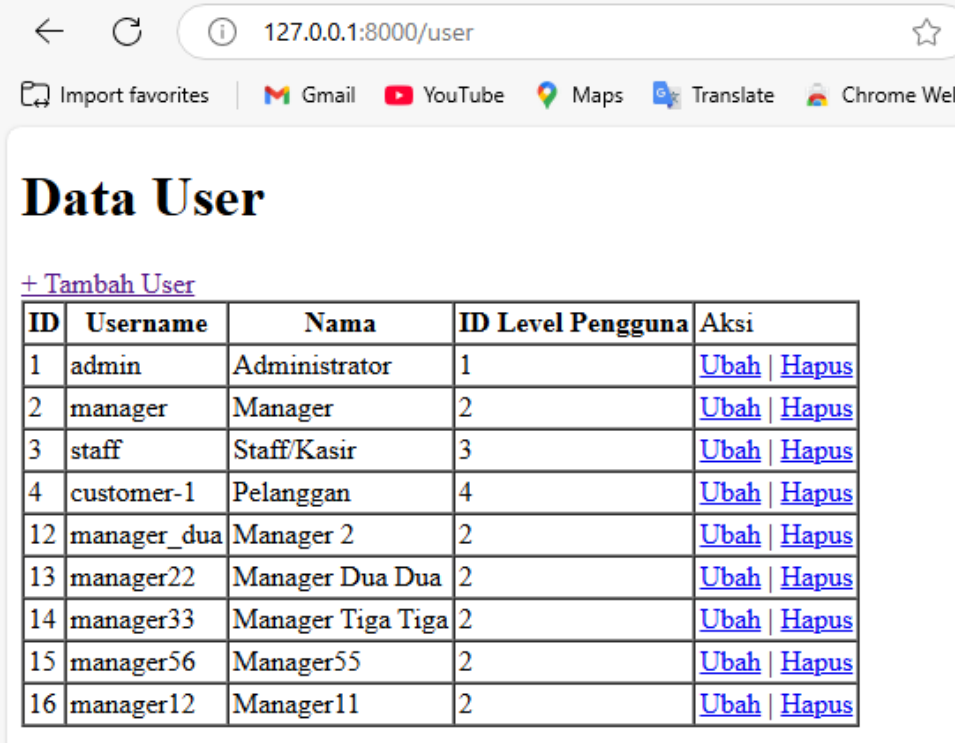
r) Menambahkan Langkah delete di file web.php

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

s) Menambahkan script pada UserController.php

```
104     public function hapus($id){
105         $user = UserModel::find($id);
106         $user->delete();
107
108         return redirect('/user');
109     }
110 }
```

t) Menjalankan pada browser



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	4	Ubah | Hapus
12	manager_dua	Manager 2	2	Ubah | Hapus
13	manager22	Manager Dua Dua	2	Ubah | Hapus
14	manager33	Manager Tiga Tiga	2	Ubah | Hapus
15	manager56	Manager55	2	Ubah | Hapus
16	manager12	Manager11	2	Ubah | Hapus

Ketika mengklik link hapus secara otomatis data akan terhapus

Praktikum 2.7: Relationships

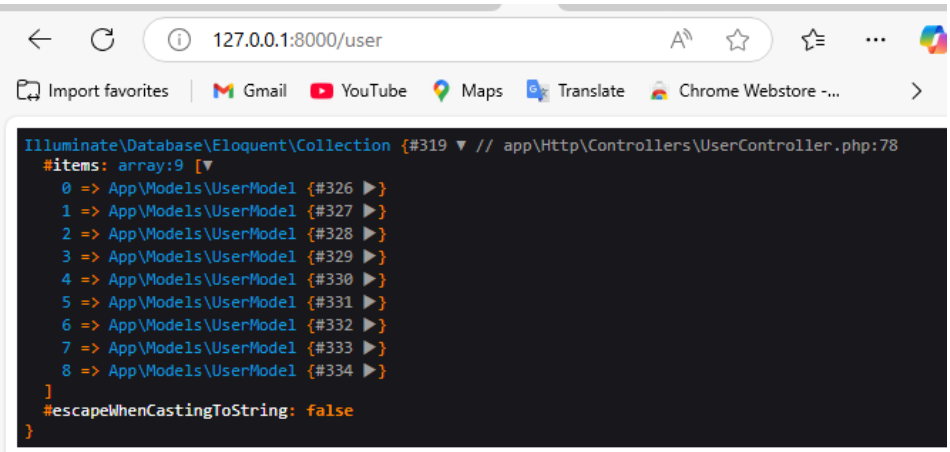
a) Membuka file UserModel.php dan menambahkan script berikut

```
23    public function level(): BelongsTo
24    {
25        return $this->belongsTo(LevelModel::class,'level_id', 'level_id');
26    }
```

b) Menambahkan script pada UserController.php

```
77        $user = UserModel::with('level')->get();
78        dd($user);
79    }
```

c) Menjalankan pada browser



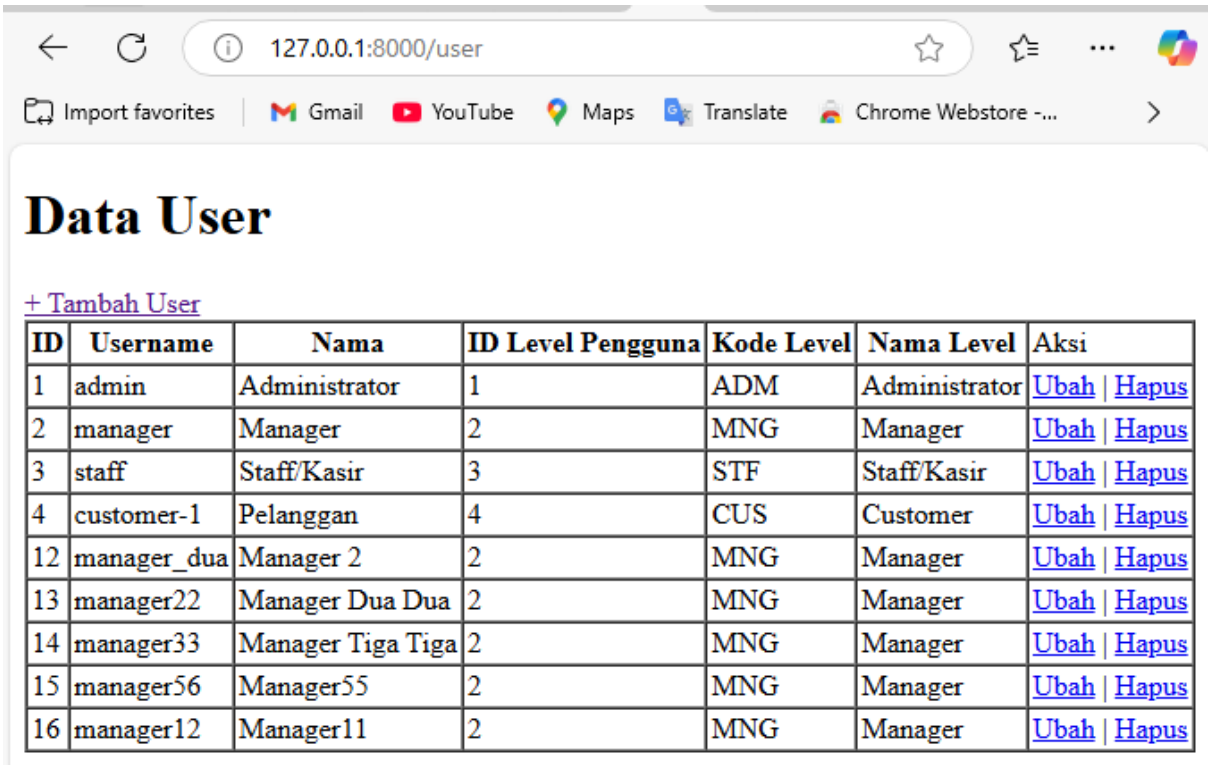
d) Menambahkan script pada UserController.php

```
77        $user = UserModel::with('level')->get();
78        return view('user', ['data' => $user]);
```

e) Menambahkan script pada user.blade.php

```
17                <th>Kode Level</th>
18                <th>Nama Level</th>
19                <td>Aksi</td>
20            </tr>
21            @foreach ($data as $d)
22            <tr>
23                <td>{{ $d->id_user}}</td>
24                <td>{{ $d->username}}</td>
25                <td>{{ $d->nama}}</td>
26                <td>{{ $d->level_id}}</td>
27                <td>{{ $d->level->level_kode}}</td>
28                <td>{{ $d->level->level_nama}}</td>
```

f) Menjalankan pada browser



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	Staff/Kasir	Ubah | Hapus
4	customer-1	Pelanggan	4	CUS	Customer	Ubah | Hapus
12	manager_dua	Manager 2	2	MNG	Manager	Ubah | Hapus
13	manager22	Manager Dua Dua	2	MNG	Manager	Ubah | Hapus
14	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah | Hapus
15	manager56	Manager55	2	MNG	Manager	Ubah | Hapus
16	manager12	Manager11	2	MNG	Manager	Ubah | Hapus