KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
https://jti.polinema.ac.id

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

Kelas	:	SIB
NIM	:	2241760026
Nama	:	Siti Faiqoh
Jobsheet Ke-	:	4

Laporan Jobsheet

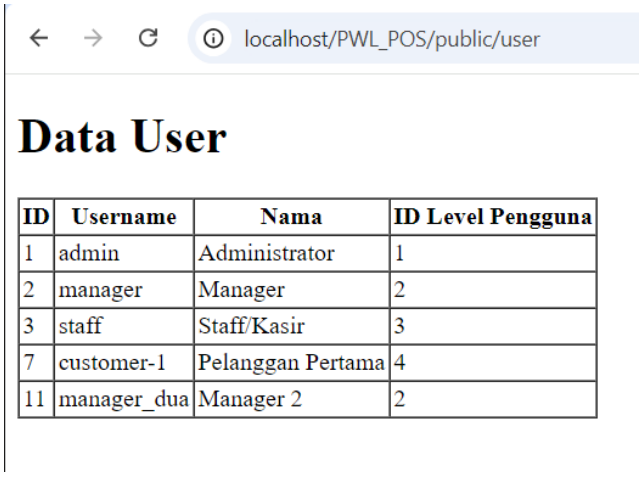
Praktikum 1 - $fillable

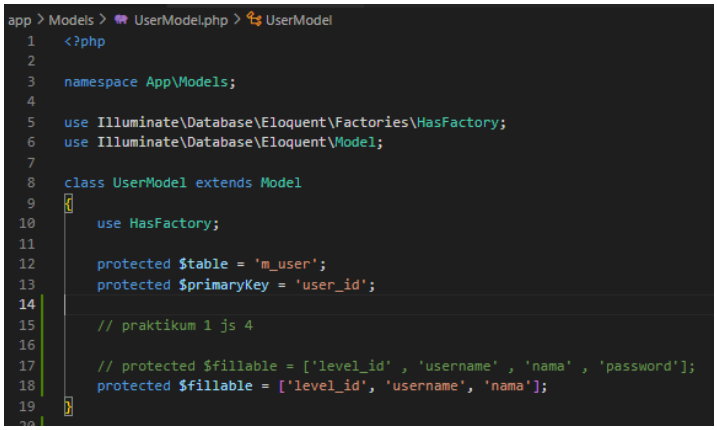
Langkah	**Jawaban/Deskripsi**

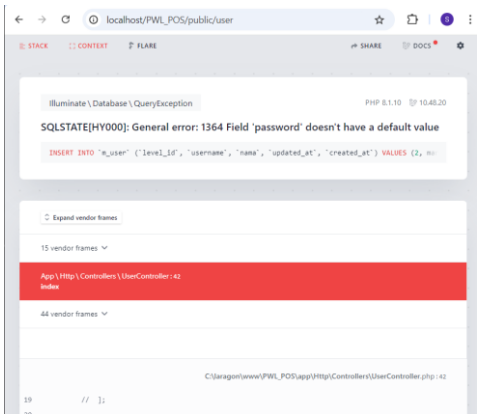
1	Buka file model dengan nama `UserModel.php` dan tambahkan `$fillable` seperti gambardi bawah ini
	app > Models > UserModel.php > ...
	`<?php`
	
	`namespace App\Models;`
	
	`use Illuminate\Database\Eloquent\Factories\HasFactory;`
	`use Illuminate\Database\Eloquent\Model;`
	
	`class UserModel extends Model`
	`{`
	`  use HasFactory;`
	
	`  protected $table = 'm_user';`
	`  protected $primaryKey = 'user_id';`
	
	`  // bagian 1 js 4`
	
	`  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
	app > Http > Controllers > UserController.php > UserController
	`class UserController extends Controller`
	`  public function index()`
	`  // // bagian 1 js 4`
	`  $data = [`
	`  'level_id' => 2,`
	`  'username' => 'manager_dua',`
	`  'nama' => 'Manager 2',`
	`  'password' => Hash::make('12345')`
	`  ];`
	`  UserModel::create($data);`
	
	`  $user = UserModel::all();`
	`  return view('user', ['data' => $user]);`
	`  }`
	`}`

3	Simpan kode program Langkah 1 dan 2, dan jalankan web server. Kemudian jalankan

<table>
<tr><td></td><td>link <code>localhostPWL_POS/public/user</code> pada browser dan amati apa yang terjadi
 

Hasil pengamatan menunjukkan adanya penambahan data karena proses kedua melibatkan pembuatan atau pengisian data baru ke dalam database. Data tersebut kemudian ditampilkan kembali melalui array yang dicreate dan dimasukkan ke database.</td></tr>
<tr><td>4</td><td>Ubah file model <code>UserModel.php</code> seperti pada gambar di bawah ini pada bagian
<code>$fillable</code>
 
</td></tr>
<tr><td>5</td><td>Ubah kembali file controller <code>UserController.php</code> seperti pada gambar di bawah hanya bagian array pada $data</td></tr>
</table>

	app > Http > Controllers > UserController.php > ... 9 class UserController extends Controller 11 public function index() 34 $data = [35 'level_id' => 2, 36 // 'username' => 'manager_dua', 37 // 'nama' => 'Manager 2', 38 'username' => 'manager_tiga', 39 'nama' => 'Manager 3', 40 'password' => Hash::make('12345') 41]; 42 UserModel::create($data); 43 44 $user = UserModel::all(); 45 return view('user', ['data' => $user]); 46 } 47 } 48
6	Simpan kode program Langkah 4 dan 5. Kemudian jalankan pada *browser* dan amati apa yang terjadi  Error SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value, menunjukkan bahwa saya coba memasukkan data ke dalam tabel m_user, tetapi field password tidak diberi nilai, dan di database, field ini mungkin diset sebagai NOT NULL (tidak boleh kosong). Jadi harusnya pada `UserModel.php` terdapat ‘password’.
7	Laporkan hasil Praktikum-1 ini dan *commit* perubahan pada *git*.

Praktikum 2.1 – Retrieving Single Models

Langkah	**Jawaban/Deskripsi**
1	Buka file controller dengan nama `UserController.php` dan ubah *script* seperti gambardi bawah ini

<table>
<tr><td></td><td>

app > Http > Controllers > UserController.php > ...

```
class UserController extends Controller
    public function index()
        // PRAKTIKUM 2.1 JS 4
        $user = UserModel::find(1);
        return view('user', ['data' => $user]);
    }
}
```

</td></tr>
<tr><td>2</td><td>

Buka file *view* dengan nama `user.blade.php` dan ubah *script* seperti gambar di bawahini

resources > views > user.blade.php > html

```
<html>
<body>
    <table border="1" cellpadding="2", cellspacing="0">
        <tr>
            {{-- PRAKTIKUM 2.1 JS 4 --}}
            <td>{{ $data->user_id }}</td>
            <td>{{ $data->username }}</td>
            <td>{{ $data->nama }}</td>
            <td>{{ $data->level_id }}</td>

        </tr>
        @endforeach
    </table>
</body>
</html>
```

</td></tr>
<tr><td>3</td><td>

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

localhost/PWL_POS/public/user

Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1

Penggunaan find() akan menampilkan data berdasarkan user_id. Jika user_id yang dimasukkan adalah 1, maka data dengan user_id 1 yang akan ditampilkan.

</td></tr>
<tr><td>4</td><td>

Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambardi bawah ini

</td></tr>
</table>

<table>
<tr><td></td><td>

app > Http > Controllers > UserController.php > UserController

```php
class UserController extends Controller
    public function index()
        // return view('user', ['data' => $user]);

        // PRAKTIKUM 2.1 JS 4
        // $user = UserModel::find(1);
        $user = UserModel::where('level_id', 1)->first();
        return view('user', ['data' => $user]);
    }
}
```

localhost/PWL_POS/public/user

Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1

</td></tr>
<tr><td>5</td><td>

Simpan kode program Langkah 4. Kemudian jalankan pada *browser* dan amati apa yangterjadi dan beri penjelasan dalam laporan

Hasil pengamatan menunjukkan bahwa program sebelumnya memfilter data berdasarkan kolom level_id dan memasukkan valuenya. Penggunaan first() akan menampilkan baris pertama yang sesuai dengan kriteria yang diberikan.

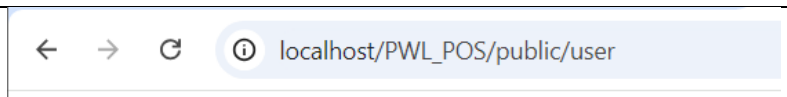
</td></tr>
<tr><td>6</td><td>

Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambardi bawah ini

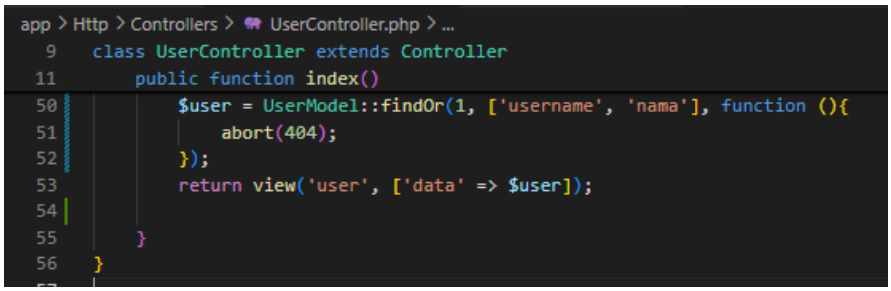
app > Http > Controllers > UserController.php > ...

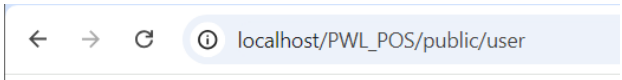
```php
class UserController extends Controller
    public function index()
        // UserModel::create($data);

        // $user = UserModel::all();
        // return view('user', ['data' => $user]);

        // PRAKTIKUM 2.1 JS 4
        // $user = UserModel::find(1);
        // $user = UserModel::where('level_id', 1)->first();
        $user = UserModel::firstWhere('level_id', 1);
        // $user = UserModel::findOr(1, ['username', 'nama'], function (){
        //     abort(404);
        // });
        return view('user', ['data' => $user]);

    }
}
```

</td></tr>
</table>

<table>
<tr><td></td><td>


Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
</td></tr>
<tr><td>7</td><td>Simpan kode program Langkah 6. Kemudian jalankan pada browser dan amati apa yangterjadi dan beri penjelasan dalam laporan
Hasil pengamatan menunjukkan bahwa program mencari baris pertama berdasarkan filter level_id dengan nilai 1. Program kemudian menampilkan hasil dengan level_id 1 dari baris pertama.</td></tr>
<tr><td>8</td><td>Ubah file controller dengan nama <code>UserController.php</code> dan ubah script seperti gambardi bawah ini




Data User

ID	Username	Nama	ID Level Pengguna
	admin	Administrator	
	admin	Administrator	
	admin	Administrator	
	admin	Administrator	
	admin	Administrator	
	admin	Administrator	
</td></tr>
<tr><td>9</td><td>Simpan kode program Langkah 8. Kemudian pada browser dan amati apa yang</td></tr>
</table>

	terjadidan beri penjelasan dalam laporan Program akan mencari data berdasarkan id dengan nilai 1. Jika tidak ditemukan, program akan mencari di kolom username dan nama dengan nilai 1.

10	Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambardi bawah ini
	app > Http > Controllers > UserController.php > ...
	`class UserController extends Controller`
	`public function index()`
	`$user = UserModel::findOr(20, ['username', 'nama'], function (){`
	`abort(404);`
	`});`
	`return view('user', ['data' => $user]);`
	`}`
	`}`
	localhost/PWL_POS/public/user
	404 | NOT FOUND

11	Simpan kode program Langkah 10. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan tampilan pada localhost terdapat 404 |not found dikarenakan id 20 tidak ada / tidak termasuk dalam database user
12	Laporkan hasil Praktikum-2.1 ini dan *commit* perubahan pada *git*.

Praktikum 2.2 – *Not Found Exceptions*

Langkah	**Jawaban/Deskripsi**
1	Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambardi bawah ini

<table>
<tr><td></td><td>

```
app > Http > Controllers > UserController.php > ...
9   class UserController extends Controller
11      public function index()
56          $user = UserModel::findOrFail(1);
57          // $user = UserModel::where('username', 'manager9')-> fistOrFail();
58          return view('user', ['data' => $user]);
59
60      }
61  }
```

</td></tr>
<tr><td>2</td><td>

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

localhost/PWL_POS/public/user

Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1

Dari hasil pengamatan didapatkan bahwa program tersebut mencari id dengan nilai 1

</td></tr>
<tr><td>3</td><td>

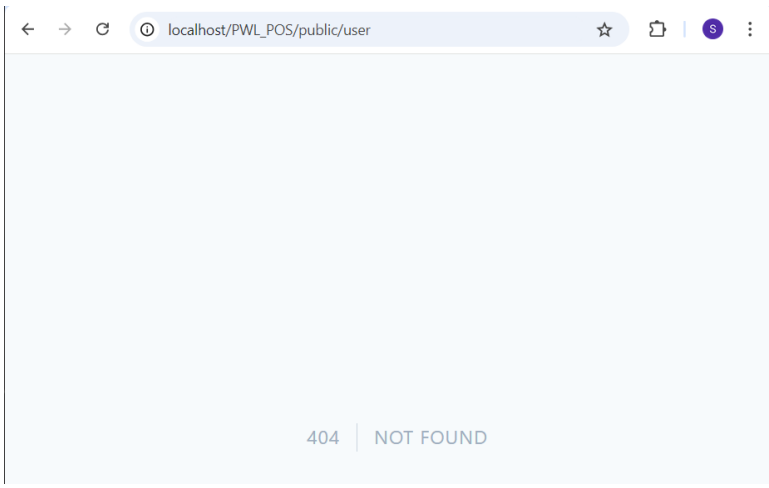
Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini

```
app > Http > Controllers > UserController.php > ...
9   class UserController extends Controller
11      public function index()
57          $user = UserModel::where('username', 'manager9')-> firstOrFail();
58          return view('user', ['data' => $user]);
59
60      }
61  }
```

</td></tr>
<tr><td>4</td><td>

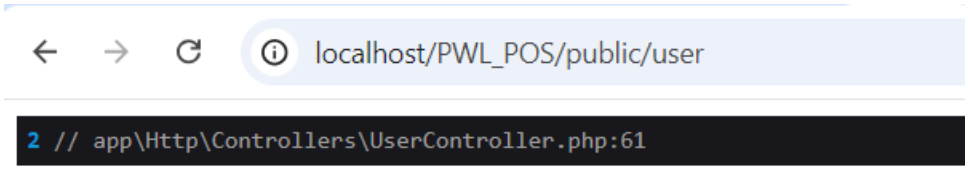
Simpan kode program Langkah 3. Kemudian jalankan pada *browser* dan amati apa yangterjadi dan beri penjelasan dalam laporan

</td></tr>
</table>

	 Program mencari username 'manager9'. Jika ditemukan, akan mengambil baris pertama dengan firstOrFail. Jika tidak ditemukan, akan langsung menampilkan error 404 not found.
5	Laporkan hasil Praktikum-2.2 ini dan *commit* perubahan pada *git*

Praktikum 2.3 – *Retreiving Aggregrates*

Langkah	**Jawaban/Deskripsi**

1	Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini
	app > Http > Controllers > UserController.php > ...
	`9 class UserController extends Controller`
	`11 public function index()`
	`60 $user = UserModel::where('level_id', 2)->count();`
	`61 dd($user);`
	`62 return view('user', ['data' => $user]);`
	`64 }`
	`65 }`
	 
	`2 // app\Http\Controllers\UserController.php:61`

2	Simpan kode program Langkah 1. Kemudian jalankan pada *browser* dan amati apa yangterjadi dan beri penjelasan dalam laporan Program tidak dapat ditampilkan karena penggunaan `dd()` yang menghentikan

	eksekusi, sehingga baris program berikutnya tidak berjalan.

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
	app > Http > Controllers > UserController.php > ...
	`class UserController extends Controller`
	`public function index()`
	`$user = UserModel::where('level_id', 2)->count();`
	`// dd($user);`
	`return view('user', ['data' => $user]);`
	`}`
	`}`
	Dengan cara menghilangkan dd($user) maka tampilan web akan seperti dibawah ini
	localhost/PWL_POS/public/user
	Data User
	Jumlah Pengguna
	2

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

Praktikum 2.4 – *Retreiving or Creating Models*

Langkah	**Jawaban/Deskripsi**

1	Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambardi bawah ini
	app > Http > Controllers > UserController.php > UserController > index
	`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 dibawah ini
	resources > views > user.blade.php > html > body > table > tr > td
	`2 <html>`
	`6 <body>`
	`8 <table border="1" cellpadding="2", cellspacing="0">`
	`39 <th>ID</th>`
	`40 <th>Username</th>`
	`41 <th>Nama</th>`
	`42 <th>ID Level Pengguna</th>`
	`43 </tr>`
	`44 <tr>`
	`45 <td>{{ $data->user_id }}</td>`
	`46 <td>{{ $data->username }}</td>`
	`47 <td>{{ $data->nama }}</td>`
	`48 <td>{{ $data->level_id }}</td>`
	`49 </tr>`
	`50 </table>`
	`51 </body>`
	`52 </html>`

3	Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan localhost/PWL_POS/public/user **Data User** ID | Username | Nama | ID Level Pengguna 2 | manager | Manager | 2 Program akan mencari baris pertama berdasarkan username dan nama yang ditentukan. Jika tidak ditemukan, program akan membuat atau menambahkan data baru dengan username dan nama tersebut.
4	Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini

<table>
<tr><td></td><td>

app > Http > Controllers > UserController.php > ...

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

    }
}
```

Tidak lupa menambahkan usermodel

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

Lalu jalankan web dan cek pada database phpmyadmin

localhost/PWL_POS/public/user

Data User

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

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=pwl_pos&table=m_user

Server: localhost:3306 » Database: pwl_pos » Table: m_user

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Showing rows 0 - 5 (6 total, Query took 0.0021 seconds.)

SELECT * FROM `m_user`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	user_id	level_id	username	nama	password	created_at	updated_at
Edit Copy Delete	1	1	admin	Administrator	$2y$12$KeoQ2dwA.U2KLusYRjXPPuP8JS2dyNxbEGiPtgcjlW2...	NULL	NULL
Edit Copy Delete	2	2	manager	Manager	$2y$12$MeZ/VnTqhDgKVFEDQws4NO9fRwsyZu.B.8W3M5.k93s...	NULL	NULL
Edit Copy Delete	3	3	staff	Staff/Kasir	$2y$12$aQlmmTZr//HstdGh.6MOJO1rvgSVvPQWEVXDsUjFAYj...	NULL	NULL
Edit Copy Delete	7	4	customer-1	Pelanggan Pertama	$2y$12$x/g6rBlD07c638why6m75eX82T0FQCjfjXhKGe6O9YI...	NULL	2024-09-14 09:01:28
Edit Copy Delete	11	2	manager_dua	Manager 2	$2y$12$2jwTkbJiw9dTSjnkw4GvruIWgAHrxIXdxPTFTtrEUod...	2024-09-19 17:25:05	2024-09-19 17:25:05
Edit Copy Delete	12	2	manager22	Manager Dua Dua	$2y$12$il1i3ZJpxk1BsLMwGlfxMOoD7UNJJz/Jr3FSEK/XvIM...	2024-09-20 05:39:56	2024-09-20 05:39:56

Check all | With selected: Edit | Copy | Delete | Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

</td></tr>
<tr><td>5</td><td>Simpan kode program Langkah 4. Kemudian jalankan pada browser dan amati apa yangterjadi dan cek juga pada phpMyAdmin pada tabel `m_user` serta beri penjelasan dalam laporan

Pada database phpmyadmin terdapat tambahan user yaitu manager22</td></tr>
<tr><td>6</td><td>Ubah file controller dengan nama `UserController.php` dan ubah script seperti gambar di bawah ini</td></tr>
</table>

<table>
<tr><td></td><td>

app > Http > Controllers > UserController.php > ...

```php
class UserController extends Controller
    public function index()
        $user = UserModel::firstOrNew(
            [
                // 'username'=> 'manager',
                // 'nama' => 'Manager',
                'username'=>'manager22',
                'nama'=>'Manager Dua Dua',
                // 'username'=>'manager33',
                // 'nama'>'Manager Tiga Tiga',
                // 'password'=> Hash::make('12345'),
                // 'level_id'=> 2
            ],
        );
        // $user->save();
        return view('user', ['data' => $user]);

    }
}
```

localhost/PWL_POS/public/user

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

</td></tr>
<tr><td>7</td><td>

Simpan kode program Langkah 6. Kemudian jalankan pada *browser* dan amati apa yangterjadi dan beri penjelasan dalam laporan

Program mencari berdasarkan kriteria username dan nama. Hasilnya akan ditampilkan di halaman website pada baris pertama. Jika tidak ditemukan, data baru akan dibuat, tetapi tidak disimpan di dalam database.

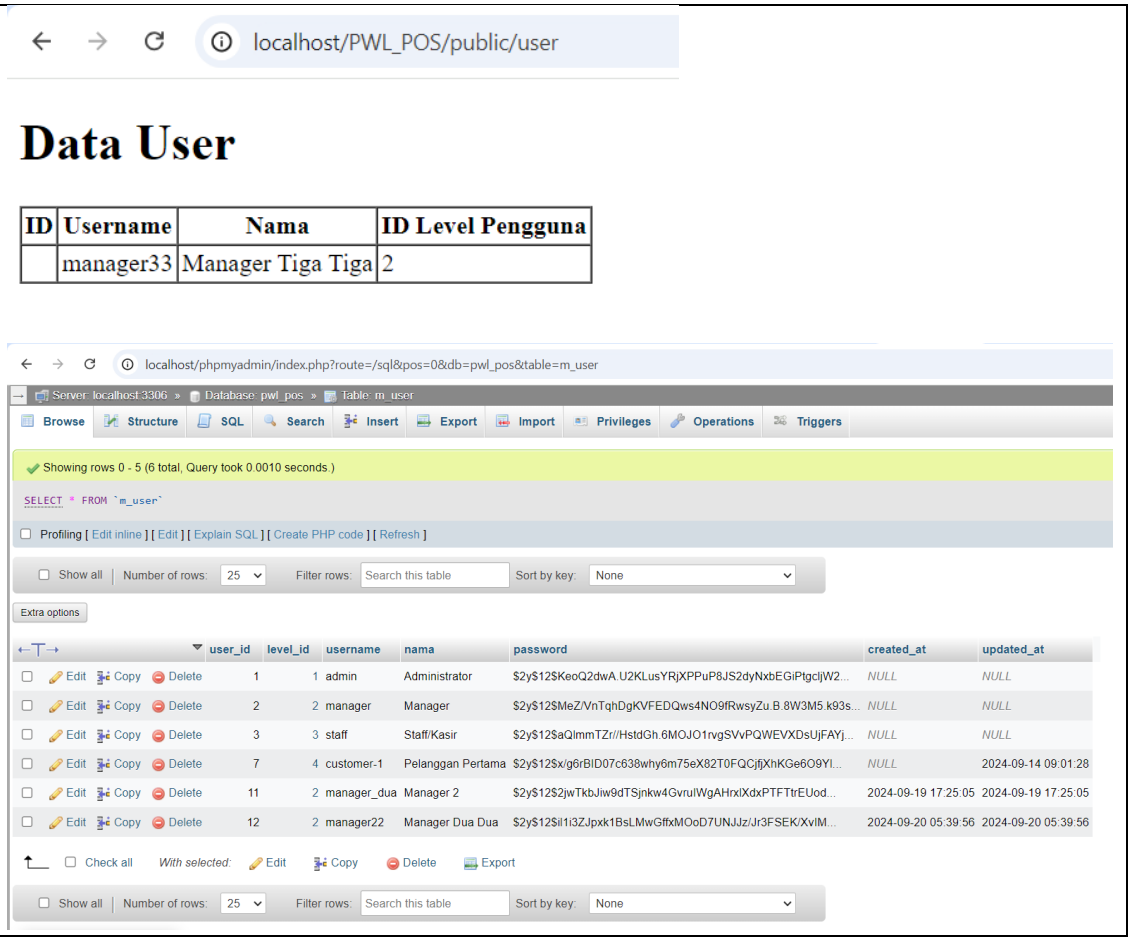
</td></tr>
<tr><td>8</td><td>

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

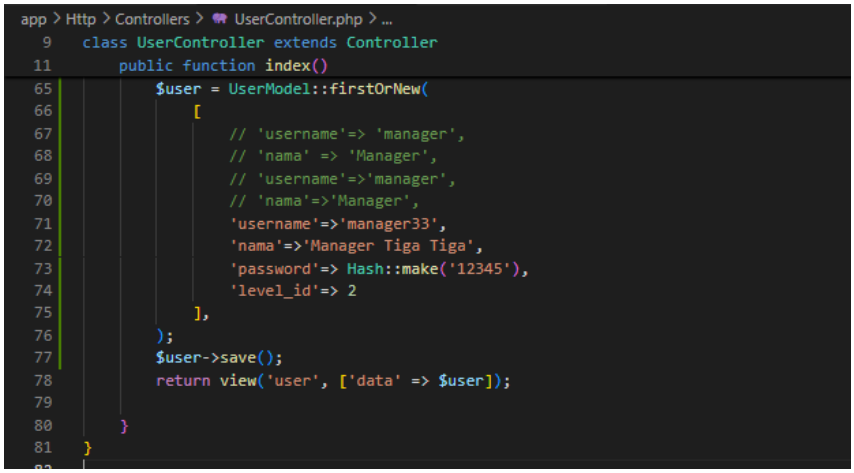
app > Http > Controllers > UserController.php > UserController

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

    }
}
```

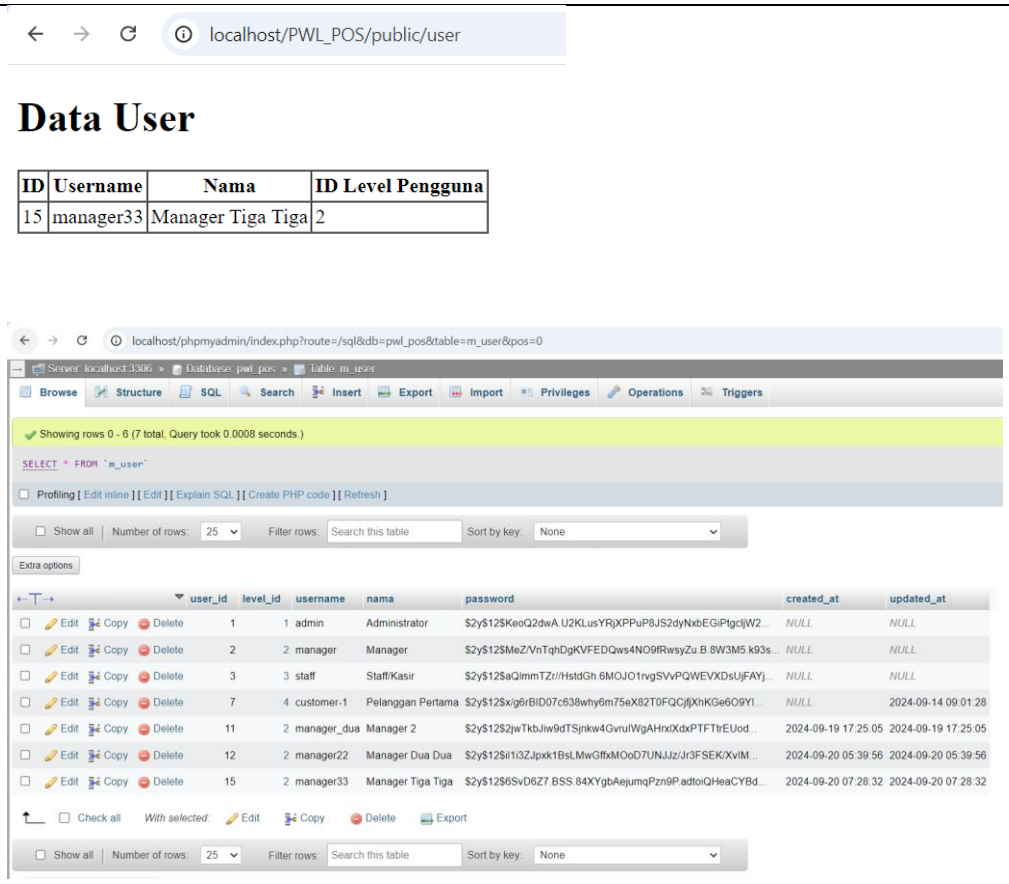
</td></tr>
</table>

<table>
<tr><td></td><td>


</td></tr>
<tr><td>9</td><td>Simpan kode program Langkah 8. Kemudian jalankan pada browser dan amati apa yangterjadi dan cek juga pada phpMyAdmin pada tabel m_user serta beri penjelasan dalam laporan

Tampilan web menampilkan ‘manager33’, tetapi tidak ada perubahan pada database phpmyadmin</td></tr>
<tr><td>10</td><td>Ubah file controller dengan nama <code>UserController.php</code> dan ubah script seperti


</td></tr>
</table>

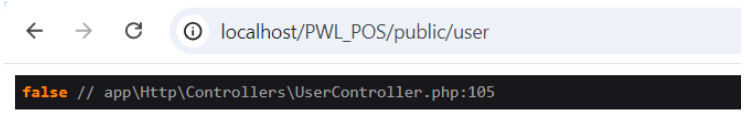
<table>
<tr><td></td><td>


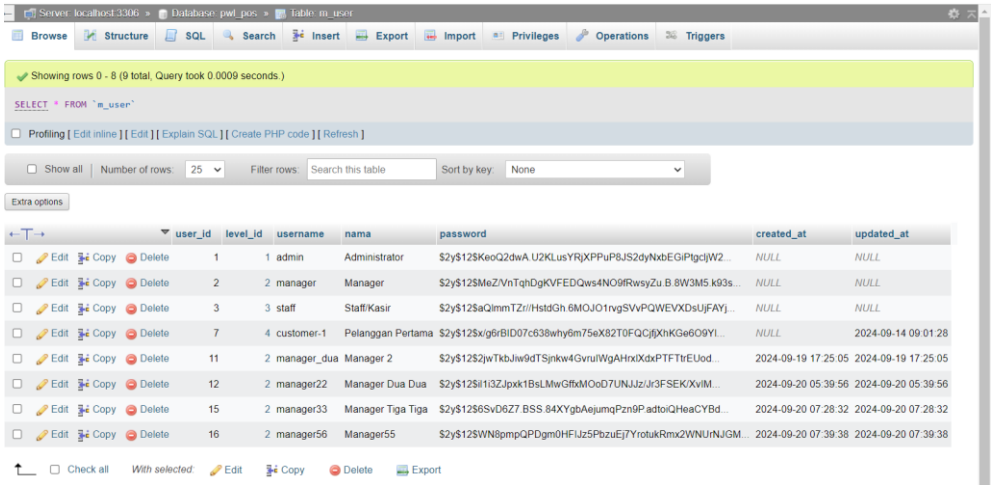
</td></tr>
<tr><td>11</td><td>Simpan kode program Langkah 9. Kemudian jalankan pada browser dan amati apa yangterjadi dan cek juga pada phpMyAdmin pada tabel <code>m_user</code> serta beri penjelasan dalam laporan
Data pada tampilan web ‘manager33’ masuk ke dalam database phpmyadmin karena pada code menambahkan $save->user();</td></tr>
<tr><td>12</td><td>Laporkan hasil Praktikum-2.4 ini dan commit perubahan pada git.</td></tr>
</table>

Praktikum 2.5 – *Attribute Changes*

Langkah	**Jawaban/Deskripsi**
1	Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambardi bawah ini

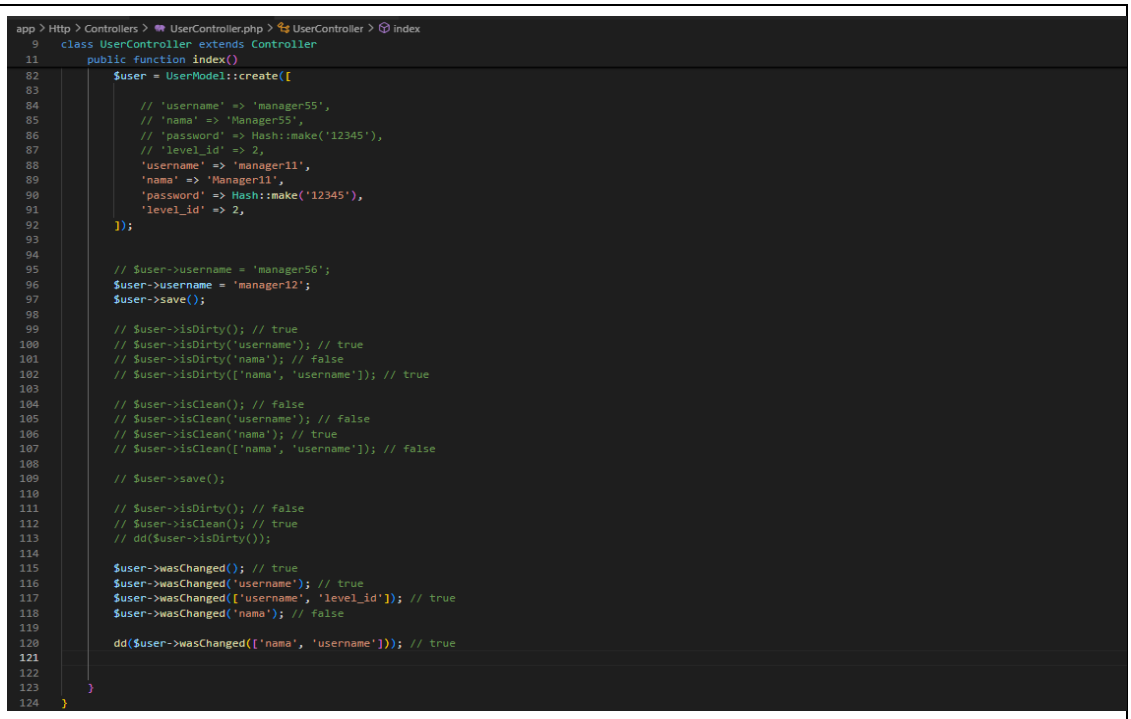
<table>
<tr>
<td></td>
<td>

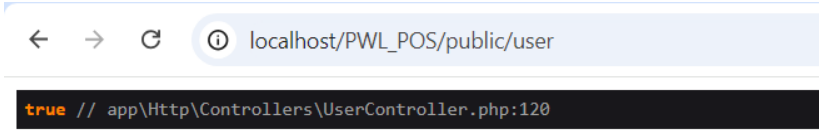


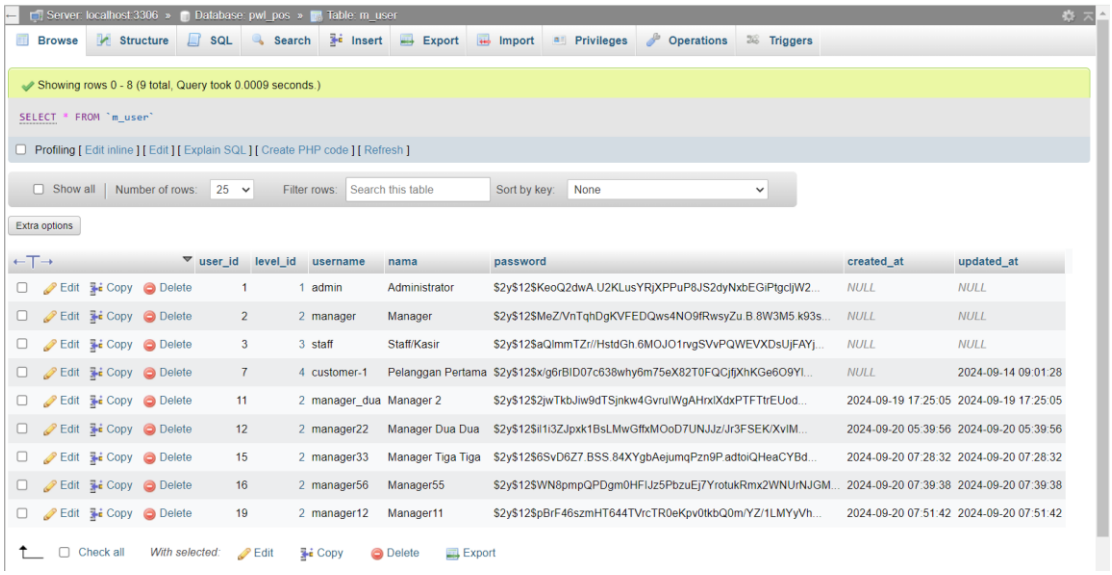


</td>
</tr>
<tr>
<td>2</td>
<td>Simpan kode program Langkah 1. Kemudian jalankan pada browser dan amati apa yangterjadi dan beri penjelasan dalam laporan

Dari hasil pengamatan diatas kenapa hasil nya adalah false dikarenakan setelah perubahan disimpan, tidak ada lagi perubahan baru yang belum disimpan pada objek $user.</td>
</tr>
<tr>
<td>3</td>
<td>Ubah file controller dengan nama <code>UserController.php</code> dan ubah script seperti gambardi bawah ini</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>






</td>
</tr>
<tr>
<td>4</td>
<td>Simpan kode program Langkah 3. Kemudian jalankan pada browser dan amati apa yangterjadi dan beri penjelasan dalam laporan

Hasilnya adalah true karena terjadi perubahan pada atribut nama atau username setelah diinsert ke database. Meskipun perubahan pada kolom username sudah disimpan, wasChanged tetap mendeteksi perubahan sejak pertama kali data diinsert</td>
</tr>
</table>

	ke database.
5	Laporkan hasil Praktikum-2.5 ini dan *commit* perubahan pada *git*.

Praktikum 2.6 – *Create, Read, Update, Delete (CRUD)*

Langkah	**Jawaban/Deskripsi**

1	Buka file *view* pada `user.blade.php` dan buat scritpnya menjadi seperti di bawah ini
	resources > views > user.blade.php > html
	`<html>`
	`<body>`
	`{{-- <table border="1" cellpadding="2", cellspacing="0">`
	`<table border="1" cellpadding="2" cellspacing="0">`
	`<tr>`
	`<td>ID</td>`
	`<td>Username</td>`
	`<td>Nama</td>`
	`<td>ID Level Pengguna</td>`
	`<td>Aksi</td>`
	`</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>`
	`</body>`
	`</html>`

2	Buka file controller pada *UserController.php* dan buat scriptnya untuk *read* menjadi seperti di bawah ini
	app > Http > Controllers > UserController.php > UserController
	`class UserController extends Controller`
	`public function index()`
	`$user = UserModel::all();`
	`return view('user', ['data' => $user]);`
	`}`
	`}`

localhost/PWL_POS/public/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
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
11	manager_dua	Manager 2	2	Ubah Hapus
12	manager22	Manager Dua Dua	2	Ubah Hapus
15	manager33	Manager Tiga Tiga	2	Ubah Hapus
16	manager56	Manager55	2	Ubah Hapus
19	manager12	Manager11	2	Ubah Hapus
20	manager11	Manager11	2	Ubah Hapus

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

Program menampilkan semua data dari database di halaman website, yang selanjutnya dapat digunakan untuk melakukan aksi seperti ubah dan hapus.

4 Langkah berikutnya membuat *create* atau tambah data user dengan cara bikin file barupada *view* dengan nama `user_tambah.blade.php` dan buat scriptnya menjadi seperti di bawah ini

resources > views > user_tambah.blade.php > html

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

        {{ csrf_field() }}

        <label>Username</label>
        <input type="text" name="username" placeholder="Masukan Username">
        <br>

        <label>Nama</label>
        <input type="text" name="nama" placeholder="Masukan Nama">
        <br>

        <label>Password</label>
        <input type="password" name="password" placeholder="Masukan Password">
        <br>

        <label>Level ID</label>
        <input type="number" name="level_id" placeholder="Masukan ID Level">
        <br><br>

        <input type="submit" class="btn btn-success" value="Simpan">
    </form>
</body>
</html>
```

5 Tambahkan *script* pada *routes* dengan nama file `web.php`. Tambahkan seperti gambar dibawah ini

routes > web.php > ...

```php
<?php
use App\Http\Controllers\LevelController;
use App\Http\Controllers\KategoriController;
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;

Route::get('/', function() {
    return view('welcome');
});

Route::get('/level', [LevelController::class, 'index']);
Route::get('/kategori', [KategoriController::class, 'index']);
Route::get('/user', [UserController::class, 'index']);
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

app > Http > Controllers > UserController.php > UserController

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

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

localhost/PWL_POS/public/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
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
11	manager_dua	Manager 2	2	Ubah Hapus
12	manager22	Manager Dua Dua	2	Ubah Hapus
15	manager33	Manager Tiga Tiga	2	Ubah Hapus
16	manager56	Manager55	2	Ubah Hapus
19	manager12	Manager11	2	Ubah Hapus
20	manager11	Manager11	2	Ubah Hapus

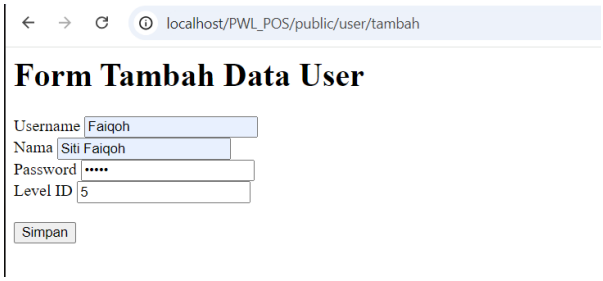
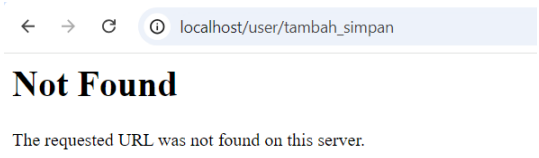
Jika klik tambah user maka akan tampil seperti ini

localhost/user/tambah

Not Found

The requested URL was not found on this server.

Ketika saya ubah alamat web (http://localhost/PWL_POS/public/user/tambah) maka

	akan tampil seperti ini  Saat disimpan tampil seperti ini 
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 Ketika tombol simpan diklik, tidak terjadi apa-apa karena belum ada konfigurasi untuk link tambah_simpan dan belum ada query untuk menyimpan data ke dalam database, sehingga hanya berfungsi sebagai form biasa.

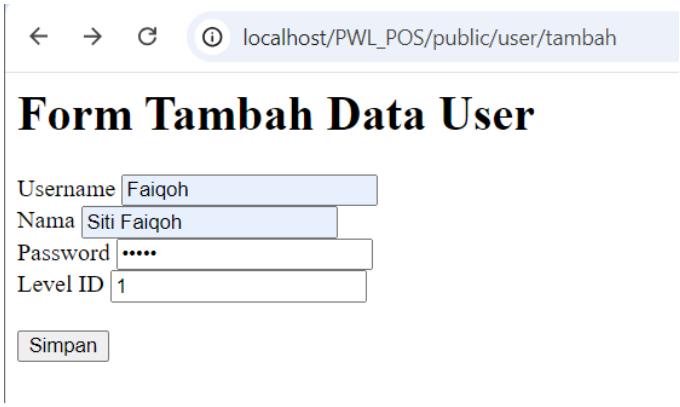
8	Tambahkan *script* pada *routes* dengan nama file `web.php`. Tambahkan seperti gambar dibawah ini
	routes > web.php > ...
	`<?php`
	`use App\Http\Controllers\LevelController;`
	`use App\Http\Controllers\KategoriController;`
	`use App\Http\Controllers\UserController;`
	`use Illuminate\Support\Facades\Route;`
	
	`Route::get('/', function() {`
	`  return view('welcome');`
	`});`
	
	`Route::get('/level', [LevelController::class, 'index']);`
	`Route::get('/kategori', [KategoriController::class, 'index']);`
	`Route::get('/user', [UserController::class, 'index']);`
	`Route::get('/user/tambah', [UserController::class, 'tambah']);`
	`Route::post('/user/tambah_simpan',[UserController::class, 'tambah_simpan']);`

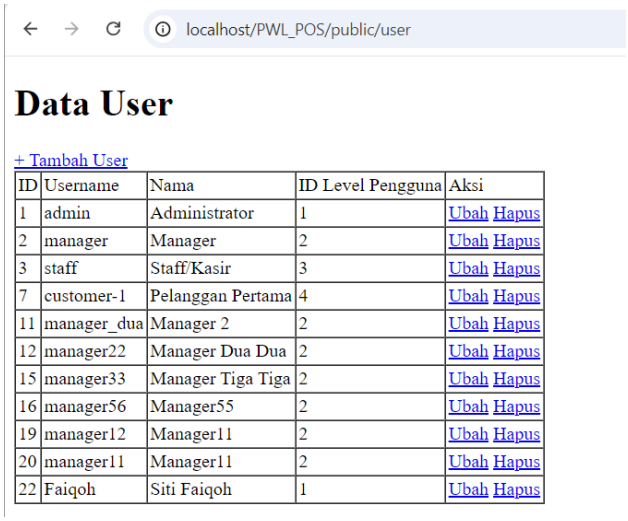
9	Tambahkan *script* pada controller dengan nama file *UserController.php*. Tambahkan *script* dalam class dan buat method baru dengan nama tambah_simpan dan diletakan dibawah method tambah seperti gambar di bawah ini

```
app > Http > Controllers > UserController.php > ...
9   class UserController extends Controller
11      public function index()
122         $user = UserModel::all();
123         return view('user', ['data' => $user]);
124     }
125
126     public function tambah() {
127         return view ('user_tambah');
128     }
129
130     public function tambah_simpan(Request $request) {
131         UserModel::create ([
132             'username' => $request->username,
133             'nama' => $request->nama,
134             'password' => Hash::make($request->password),
135             'level_id' => $request->level_id
136         ]);
137         return redirect('/user');
138     }
139 }
140
```

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





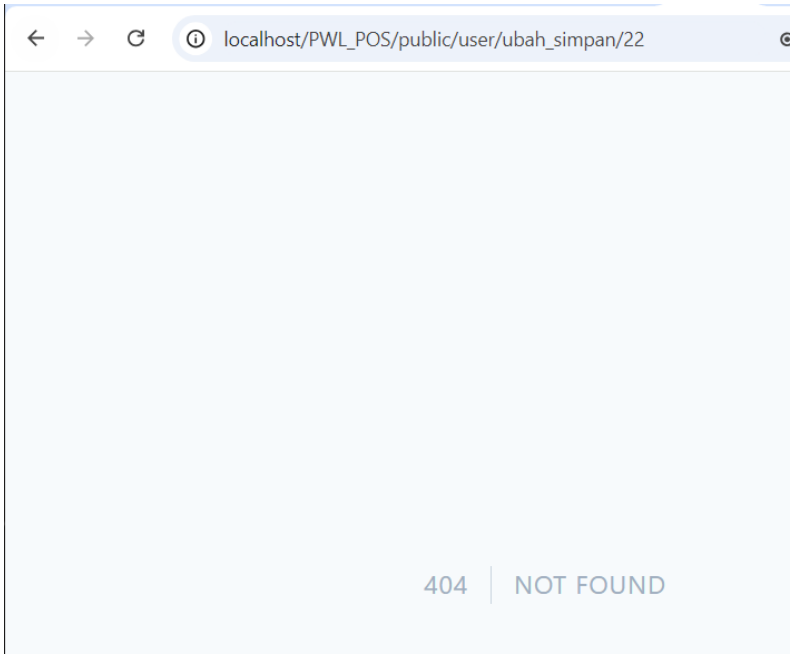
ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
11	manager_dua	Manager 2	2	Ubah Hapus
12	manager22	Manager Dua Dua	2	Ubah Hapus
15	manager33	Manager Tiga Tiga	2	Ubah Hapus
16	manager56	Manager55	2	Ubah Hapus
19	manager12	Manager11	2	Ubah Hapus
20	manager11	Manager11	2	Ubah Hapus
22	Faiqoh	Siti Faiqoh	1	Ubah Hapus

Alur penambahan data dimulai dari form untuk menginput data, yang kemudian dikirim ke controller untuk dimasukkan ke database. Setelah proses berhasil,

	pengguna akan diarahkan kembali dan data ditampilkan, seperti pada gambar di atas.
11	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 resources > views > user_ubah.blade.php > html > body `<html lang="en">` `</head>` `<body>` `<h1>Form Ubah Data User</h1>` `<a href="/user">Kembali</a>` `  ` `<form method="post" action="{{ url('/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 }}">` ` ` `<label>Nama</label>` `<input type="text" name="nama" placeholder="Masukkan Nama" value="{{ $data->nama }}">` ` ` `<label>Password</label>` `<input type="password" name="password" placeholder="Masukkan Password" value="{{ $data->password }}">` ` ` `<label>Level ID</label>` `<input type="number" name="level_id" placeholder="Masukkan ID Level" value="{{ $data->level_id }}">` `  ` `<input type="submit" class="btn btn-success" value="Ubah">` `</form>` `</body>`

12	Tambahkan *script* pada *routes* dengan nama file `web.php`. Tambahkan seperti gambar dibawah ini
	UserController.php M · user_tambah.blade.php U · web.php M · user.blade.php M
	routes > web.php > ...
	`<?php`
	`use App\Http\Controllers\LevelController;`
	`use App\Http\Controllers\KategoriController;`
	`use App\Http\Controllers\UserController;`
	`use Illuminate\Support\Facades\Route;`
	
	`Route::get('/', function() {`
	`  return view('welcome');`
	`});`
	
	`Route::get('/level', [LevelController::class, 'index']);`
	`Route::get('/kategori', [KategoriController::class, 'index']);`
	`Route::get('/user', [UserController::class, 'index']);`
	`Route::get('/user/tambah', [UserController::class, 'tambah']);`
	`Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);`
	`Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);`

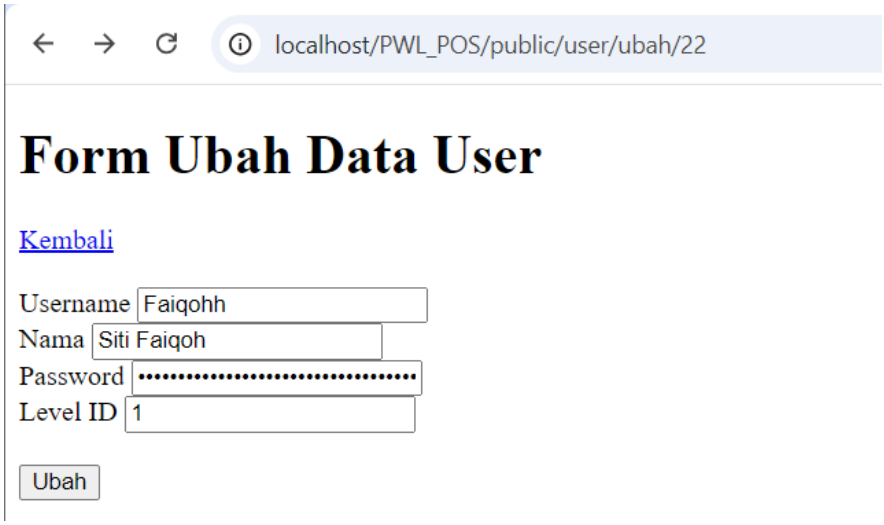
13	Tambahkan *script* pada controller dengan nama file `UserController.php`. Tambahkan *script* dalam class dan buat method baru dengan nama ubah dan diletakan di bawah method tambah_simpan seperti gambar di bawah ini
	app > Http > Controllers > UserController.php > UserController > ubah
	`class UserController extends Controller`
	`public function ubah ($id) {`
	`$user = UserModel::find($id);`
	`return view ('user_ubah', ['data' => $user]);`
	`}`
	`}`

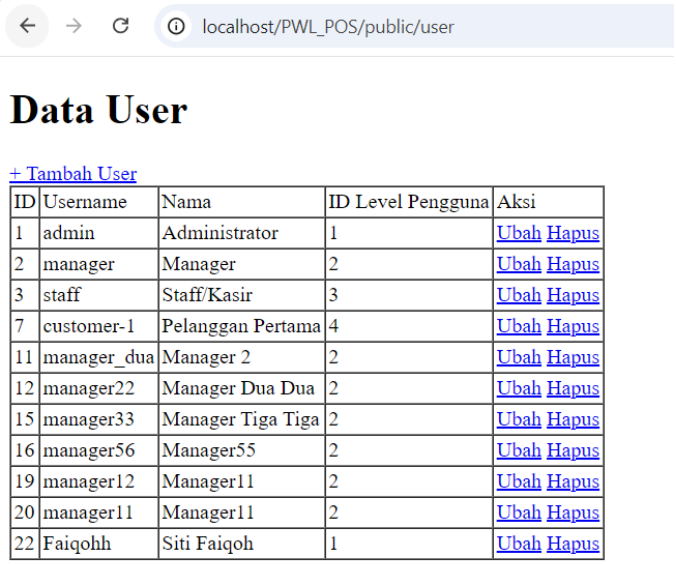
	  
14	Simpan kode program Langkah 11 sd 13. Kemudian jalankan pada *browser* dan klik link "**Ubah**" amati apa yang terjadi dan beri penjelasan dalam laporan http://localhost/PWL_POS/public/user/ubah/22 http://localhost/PWL_POS/public/user/ubah_simpan/22 Ketika klik ubah, perubahan belum bisa dilakukan secara langsung karena belum ada penyimpanan untuk hasil perubahan, dan program hanya menangani input data saja.
15	Tambahkan *script* pada *routes* dengan nama file `web.php`. Tambahkan seperti gambar dibawah ini

	routes > web.php > ...
	`<?php`
	`use App\Http\Controllers\LevelController;`
	`use App\Http\Controllers\KategoriController;`
	`use App\Http\Controllers\UserController;`
	`use Illuminate\Support\Facades\Route;`
	
	`Route::get('/', function() {`
	`  return view('welcome');`
	`});`
	
	`Route::get('/level', [LevelController::class, 'index']);`
	`Route::get('/kategori', [KategoriController::class, 'index']);`
	`Route::get('/user', [UserController::class, 'index']);`
	`Route::get('/user/tambah', [UserController::class, 'tambah']);`
	`Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);`
	`Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);`
	`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 diletakan di bawah method ubah seperti gambar di bawah ini
	app > Http > Controllers > UserController.php > UserController
	`class UserController extends Controller`
	`  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 terjadidan beri penjelasan dalam laporan





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
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
11	manager_dua	Manager 2	2	Ubah Hapus
12	manager22	Manager Dua Dua	2	Ubah Hapus
15	manager33	Manager Tiga Tiga	2	Ubah Hapus
16	manager56	Manager55	2	Ubah Hapus
19	manager12	Manager11	2	Ubah Hapus
20	manager11	Manager11	2	Ubah Hapus
22	Faiqohh	Siti Faiqoh	1	Ubah Hapus

Ketika melakukan perubahan data melalui form, perubahan langsung disimpan ke dalam database dan pengguna diarahkan kembali, sehingga proses update dapat dilakukan.

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

routes > web.php > ...

```php
<?php
use App\Http\Controllers\LevelController;
use App\Http\Controllers\KategoriController;
use App\Http\Controllers\UserController;
use Illuminate\Support\Facades\Route;

Route::get('/', function() {
    return view('welcome');
});

Route::get('/level', [LevelController::class, 'index']);
Route::get('/kategori', [KategoriController::class, 'index']);
Route::get('/user', [UserController::class, 'index']);
Route::get('/user/tambah', [UserController::class, 'tambah']);
Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);
Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);
Route::put('/user/ubah_simpan/{id}', [UserController::class, 'ubah_simpan']);
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

app > Http > Controllers > UserController.php > UserController

```php
class UserController extends Controller
    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 kliktombol hapus, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan

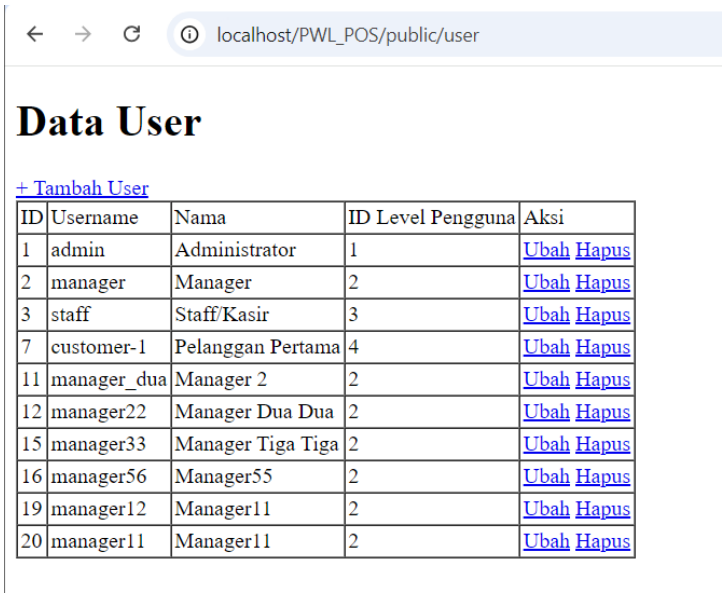
localhost/PWL_POS/public/user

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
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
11	manager_dua	Manager 2	2	Ubah Hapus
12	manager22	Manager Dua Dua	2	Ubah Hapus
15	manager33	Manager Tiga Tiga	2	Ubah Hapus
16	manager56	Manager55	2	Ubah Hapus
19	manager12	Manager11	2	Ubah Hapus
20	manager11	Manager11	2	Ubah Hapus
22	Faiqohh	Siti Faiqoh	1	Ubah Hapus

Lalu ketik http://localhost/PWL_POS/public/user/hapus/22 maka hasilnya

<table>
<tr><td></td><td>


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
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
11	manager_dua	Manager 2	2	Ubah Hapus
12	manager22	Manager Dua Dua	2	Ubah Hapus
15	manager33	Manager Tiga Tiga	2	Ubah Hapus
16	manager56	Manager55	2	Ubah Hapus
19	manager12	Manager11	2	Ubah Hapus
20	manager11	Manager11	2	Ubah Hapus

Penghapusan data dilakukan melalui route yang memanggil controller berdasarkan user_id, mencegah kesalahan penghapusan. Setelah itu, pengguna diarahkan kembali ke halaman awal yang menampilkan semua user.
</td></tr>
<tr><td>21</td><td>Laporkan hasil Praktikum-2.6 ini dan commit perubahan pada git</td></tr>
</table>

Praktikum 2.7 – *Relationships*

Langkah	**Jawaban/Deskripsi**
1	Buka file model pada `UserModel.php` dan tambahkan scritpnya menjadi seperti di bawahini

app > Models > UserModel.php > ...

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\HasOne;

class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user';
    protected $primaryKey = 'user_id';

    // praktikum 1 js 4

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

    // JS 4 PRAKTIKUM 2.7
    protected $fillable = ['level_id' , 'username' , 'nama' , 'password'];
    public function level(): BelongsTo
    {
        return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
    }
}
```

Sebelum membuat itu saya membuat levelmodel.php

app > Models > LevelModel.php > ...

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\HasMany;

class LevelModel extends Model
{
    use HasFactory;

    protected $table = 'm_level';
    protected $primaryKey = 'level_id';

    public function user(): HasMany
    {
        return $this->HasMany(User::class);
    }
}
```

2 Buka file controller pada `UserController.php` dan ubah method *script* menjadi sepertidi bawah ini

<table>
<tr><td></td><td>

app > Http > Controllers > UserController.php > ...

```
9   class UserController extends Controller
165     // JS 4 PRAKTIKUM 2.7
166     public function index() {
167         $user = UserModel::with('level')->get();
168         dd($user);
169     }
170
171     // public function index() {
172     //     $user = UserModel::with('level') -> get();
173     //     return view ('user', ['data' => $user]);
174     // }
175
176 }
```

</td></tr>
<tr><td>3</td><td>

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

localhost/PWL_POS/public/user

```
Illuminate\Database\Eloquent\Collection {#320 ▼ // app\Http\Controllers\UserController.php:168
  #items: array:10 [▶]
  #escapeWhenCastingToString: false
}
```

Hasil di atas menunjukkan jumlah data dari array items dan menjelaskan bahwa escapeCastingToString bernilai false, sehingga karakter tersebut tidak akan di-enkode.

</td></tr>
<tr><td>4</td><td>

Buka file controller pada `UserController.php` dan ubah method *script* menjadi sepertidi bawah ini

app > Http > Controllers > UserController.php > ...

```
9   class UserController extends Controller
165     // JS 4 PRAKTIKUM 2.7
166     // public function index() {
167     //     $user = UserModel::with('level')->get();
168     //     dd($user);
169     // }
170     public function index() {
171         $user = UserModel::with('level') -> get();
172         return view ('user', ['data' => $user]);
173     }
174
175 }
```

</td></tr>
<tr><td>5</td><td>

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

</td></tr>
</table>

resources > views > user.blade.php > html > body > table > table > tr > td

```
2   <html>
6   <body>
8       {{-- <table border="1" cellpadding="2", cellspacing="0">
51      {{-- JS 4 PRAKTIKUM 2.6 --}}
52      <a href="/user/tambah">+ Tambah User</a>
53      <table border="1" cellpadding="2" cellspacing="0">
54          <tr>
55              <td>ID</td>
56              <td>Username</td>
57              <td>Nama</td>
58              <td>ID Level Pengguna</td>
59          {{-- JS 4 PRAKTIKUM 2.7  --}}
60              <td>Kode Level</td>
61              <td>Nama Level</td>
62              <td>Aksi</td>
63          </tr>
64          @foreach ($data as $d)
65          <tr>
66              <td>{{ $d->user_id }}</td>
67              <td>{{ $d->username }}</td>
68              <td>{{ $d->nama }}</td>
69              <td>{{ $d->level_id }}</td>
70              <td> {{$d->level->level_kode}} </td>
71              <td> {{$d->level->level_nama}} </td>
72              <td>
73                  <a href="{{ url('/user/ubah/' . $d->user_id) }}">Ubah</a>
74                  <a href="{{ url('/user/hapus/' . $d->user_id) }}">Hapus</a>
75          </tr>
76          @endforeach
77      </table>
78  </body>
79  </html>
```

localhost/PWL_POS/public/user

Data User

+ Tambah User

ID	Username	Nama	ID Level Pengguna	Kode Level	Nama Level	Aksi
1	admin	Administrator	1	ADM	Administrator	Ubah Hapus
2	manager	Manager	2	MNG	Manager	Ubah Hapus
3	staff	Staff/Kasir	3	STF	Staff/Kasir	Ubah Hapus
7	customer-1	Pelanggan Pertama	4	CST	Customer	Ubah Hapus
11	manager_dua	Manager 2	2	MNG	Manager	Ubah Hapus
12	manager22	Manager Dua Dua	2	MNG	Manager	Ubah Hapus
15	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah Hapus
16	manager56	Manager55	2	MNG	Manager	Ubah Hapus
19	manager12	Manager11	2	MNG	Manager	Ubah Hapus
20	manager11	Manager11	2	MNG	Manager	Ubah Hapus

6	Simpan kode program Langkah 4 dan 5. Kemudian jalankan link pada *browser*, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan Hasil pengamatan menunjukkan bahwa proses join dilakukan untuk menggabungkan dua tabel dalam database dan menampilkannya di halaman website.
7	Laporkan hasil Praktikum-2.7 ini dan *commit* perubahan pada *git*.