



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

<http://www.polinema.ac.id>

LAPORAN PRAKTIKUM

PEMROGRAMAN WEB LANJUT



RIZQI REZA DANUARTA

2241720057

D4-TEKNIK INFORMATIKA

KELAS 2C



F. ELOQUENT ORM

1. Kita buat file model untuk tabel m_user dengan mengetikkan perintah

```
php artisan make:model UserModel
```

2. Setelah berhasil generate model, terdapat 2 file pada folder model yaitu file User.php bawaan dari laravel dan file UserModel.php yang telah kita buat. Kali ini kita akan menggunakan file UserModel.php

```
namespace App\Models;

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

3 references | 0 implementations
class UserModel extends Model
{
    use HasFactory;
    0 references
    protected $table = 'm_user';
    0 references
    protected $primaryKey = 'user_id';
}
```

3. Kita modifikasi route web.php untuk mencoba routing ke controller UserController

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

4. Sekarang, kita buat file controller UserController dan memodifikasinya seperti berikut



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

<http://www.polinema.ac.id>

```
namespace App\Http\Controllers;

use App\Models\UserModel;
use Illuminate\Support\Facades\Hash;
use Illuminate\Http\Request;

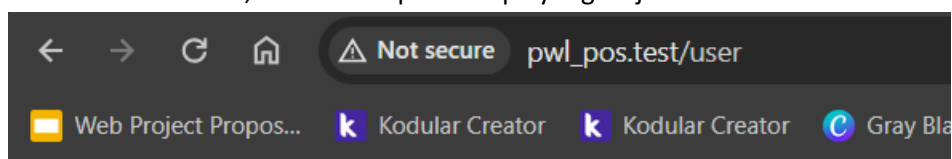
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        $user = UserModel::all();
        return view('user', ['data' => $user]);
    }
}
```

5. Kemudian kita buat view user.blade.php



```
resources > views > user.blade.php > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Data User</title>
8 </head>
9 <body>
10     <h1>Data User</h1>
11     <table border="1" cellpadding="2" cellspacing="0">
12         <tr>
13             <th>ID</th>
14             <th>Username</th>
15             <th>Nama</th>
16             <th>ID Level Pengguna</th>
17         </tr>
18         @foreach ($data as $d)
19             <tr>
20                 <td>{{ $d->user_id }}</td>
21                 <td>{{ $d->username }}</td>
22                 <td>{{ $d->nama }}</td>
23                 <td>{{ $d->level_id }}</td>
24             </tr>
25         @endforeach
26     </table>
27 </body>
28 </html>
```

6. Jalankan di browser, catat dan laporkan apa yang terjadi



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3

7. Setelah itu, kita modifikasi lagi file UserController

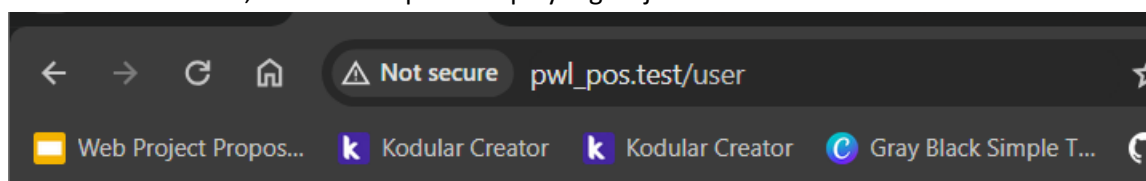


```
use App\Models\UserModel;
use Illuminate\Support\Facades\Hash;
use Illuminate\Http\Request;

2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        // $user = UserModel::all();
        // return view('user', ['data' => $user]);
        $data = [
            'username' => 'cutomer-1',
            'nama' => 'Pelanggan',
            'password' => Hash::make('12345'),
            'level_id' => 4
        ];
        UserModel::insert($data);

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

8. Jalankan di browser, amati dan laporkan apa yang terjadi



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
12	cutomer-1	Pelanggan	6



9. Kita modifikasi lagi file UserController menjadi seperti berikut

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        // $user = UserModel::all();
        // return view('user', ['data' => $user]);

        // $data = [
        //     'username' => 'customer-1',
        //     'nama' => 'Pelanggan',
        //     'password' => Hash::make('12345'),
        //     'level_id' => 4
        // ];
        // UserModel::insert($data);

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

        $data = [
            'nama' => 'Pelanggan Pertama',
        ];
        UserModel::where('username', 'customer-1')->update($data);

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

10. Jalankan di browser, amati dan laporkan apa yang terjadi
11. Jika sudah, laporkan hasil Praktikum-6 ini dan commit perubahan pada git



ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
12	cutomer-1	Pelanggan	6

A. PROPERTI \$fillable DAN \$guarded

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

```
3 references | 0 implementations
class UserModel extends Model
{
    use HasFactory;
    0 references
    protected $table = 'm_user';
    0 references
    protected $primaryKey = 'user_id';
    /*
    *
    *
    * @var array
    */
    0 references
    protected $fillable = [
        'level_id',
        'username',
        'nama',
        'password'
    ];
}
```



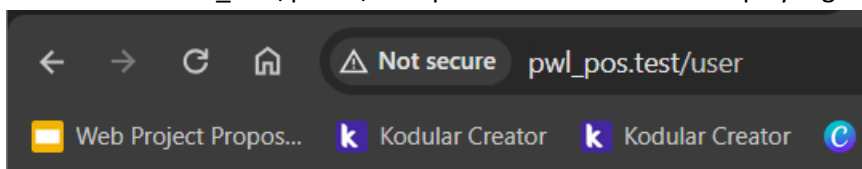
2. Buka file controller dengan nama UserController.php dan ubah script untuk menambahkan data baru seperti gambar di bawah ini

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        // $user = UserModel::all();
        // return view('user', ['data' => $user]);

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

        $user = UserModel::all();
        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



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
12	cutomer-1	Pelanggan	6
15	manager_dua	Manager 2	2

4. Ubah file model UserModel.php seperti pada gambar di bawah ini pada bagian \$fillable



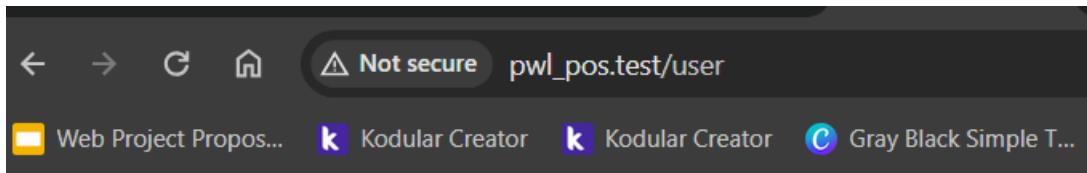
```
0 references
protected $fillable = [
    'level_id',
    'username',
    'nama'
];
```

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

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

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

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



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
12	cutomer-1	Pelanggan	6
15	manager_dua	Manager 2	2
16	manager_tiga	Manager 3	2



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

B. RETREIVING SINGLE MODELS

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

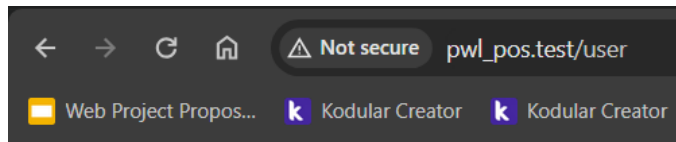
```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        $user = UserModel::find(1);
        return view('user', ['data' => $user]);
    }
}
```

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

```
resources > views > user.blade.php > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width,
6     <meta http-equiv="X-UA-Compatible" content="ie=edge
7     <title>Data User</title>
8 </head>
9 <body>
10     <h1>Data User</h1>
11     <table border="1" cellpadding="2" cellspacing="0">
12         <tr>
13             <th>ID</th>
14             <th>Username</th>
15             <th>Nama</th>
16             <th>ID Level Pengguna</th>
17         </tr>
18         @foreach ($data as $d)
19             <tr>
20                 <td>{{ $data->user_id }}</td>
21                 <td>{{ $data->username }}</td>
22                 <td>{{ $data->nama }}</td>
23                 <td>{{ $data->level_id }}</td>
24             </tr>
25         @endforeach
26     </table>
27 </body>
28 </html>
```



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



Data User

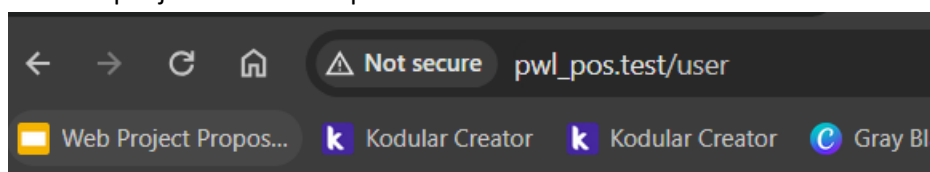
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1
1	admin	Administrator	1

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

```
1 reference | 0 overrides
public function index()
{

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

5. 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



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

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {

        $user = UserModel::firstwhere('level_id', 1);
        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

Hasilnya sama dimana kode program diatas diigunkan untuk mencari data user dengan id 1

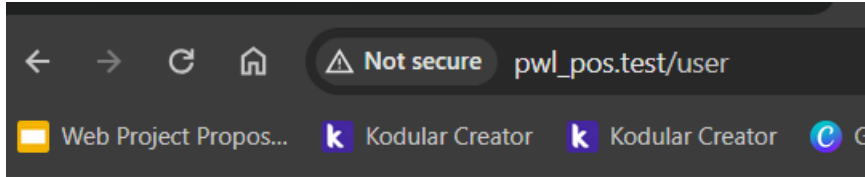
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

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

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {

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

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



Data User

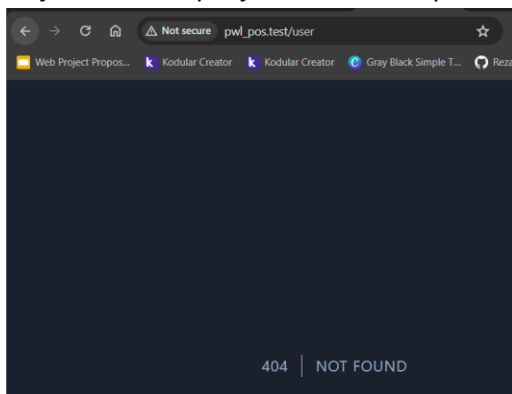
ID	Username	Nama	ID Level Pengguna
	admin	Administrator	

findOr() mencari pengguna dengan user_id 1. Jika tidak ditemukan, fungsi callback abort(404) akan dipanggil dan menghasilkan respons HTTP 404. Jika ditemukan, variabel \$user akan berisi data pengguna tersebut dan akan dikirim ke view 'user'.

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

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    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





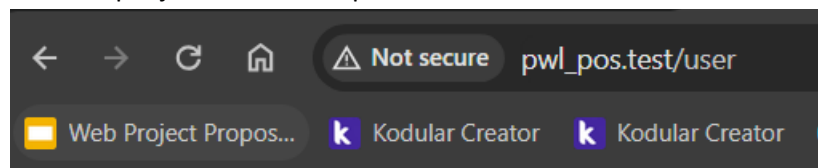
Karena di data saya tidak terdapat id dengan nomer 20 maka halaman secara otomatis akan menampilkan HTTP 404 Not Found

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

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {

        $user = UserModel::findOrFail(1);
        return view('user', ['data' => $user]);
    }
}
```

- 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

Hasil dari findOrFail adalah metode bawwan yang di sediakan oleh laravel untuk mencari entri dalam tabel berdasarkan kunci utama id jika entri tidak di temukan akan menghasilkan pengecualian ModelNotFoundException

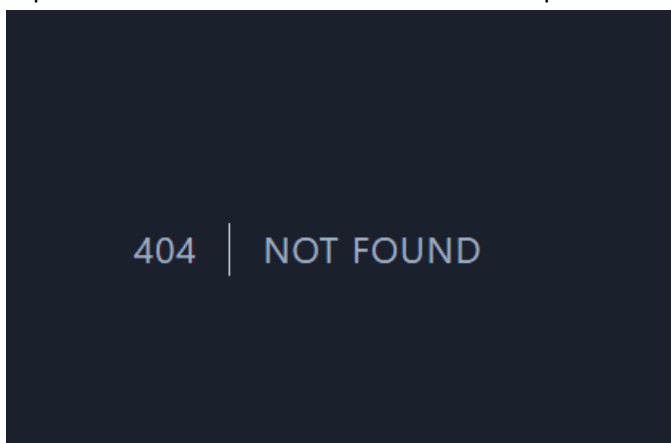
- Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini



```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {

        $user = UserModel::where('username', 'manager9')->firstOrFail();
        return view('user', ['data' => $user]);
    }
}
```

15. Simpan kode program Langkah 3. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan
16. Laporkan hasil Praktikum-2.2 ini dan commit perubahan pada git.



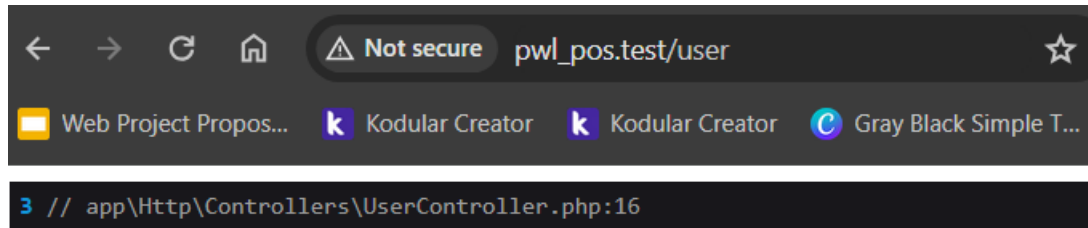
Hasil not found dikarenakan pada kode program diatas ingin mencari username dengan nama manager9 karena di data saya atau di database tidak ada username dengan nama manager9 maka hasil akan not found yang akan ditampilkan

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

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {

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

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

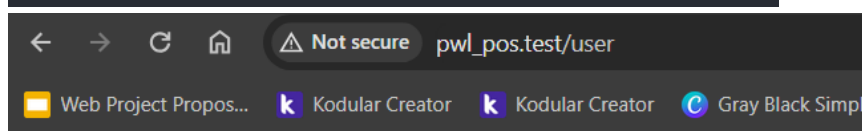


UserModel::where('level_id', 2)->count() digunakan untuk menghitung jumlah pengguna yang memiliki level_id dengan nilai 2.

dd(\$user) digunakan untuk men-debug nilai dari variabel \$user sebelum halaman dirender. Fungsi dd() akan menampilkan nilai variabel dan menghentikan eksekusi lebih lanjut dari skrip, sehingga halaman tidak akan di-render sebelum Anda melihat hasil debug.

19. 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

```
<body>
  <h1>Jumlah Pengguna dengan Level ID 2</h1>
  <table border="1" cellpadding="5" cellspacing="0">
    <tr>
      <th>No</th>
      <th>Jumlah Pengguna</th>
    </tr>
    <tr>
      <td>1</td>
      <td>{{ $data }}</td>
    </tr>
  </table>
</body>
</html>
```



Jumlah Pengguna dengan Level ID 2

No	Jumlah Pengguna
1	3



Karena di kode program saya menghitung berapa jumlah id dengan level id 2 maka yang tertampil seperti output diatas

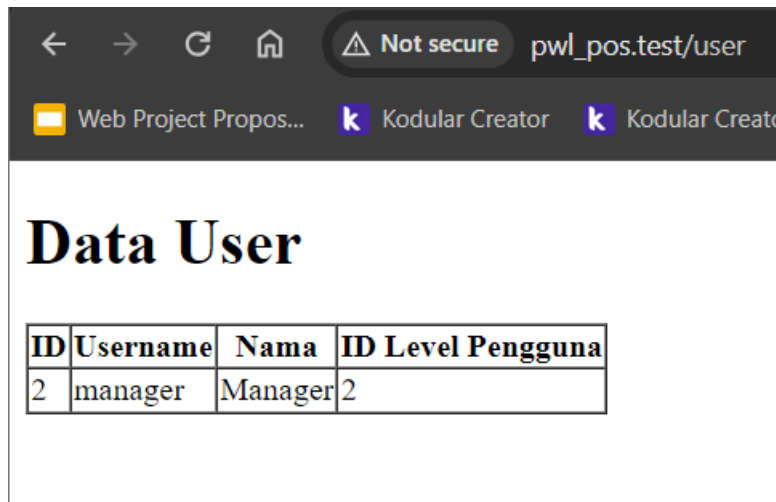
20. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager',
                'nama' => 'Manager',
            ]
        );
        return view('user', ['data' => $user]);
    }
}
```

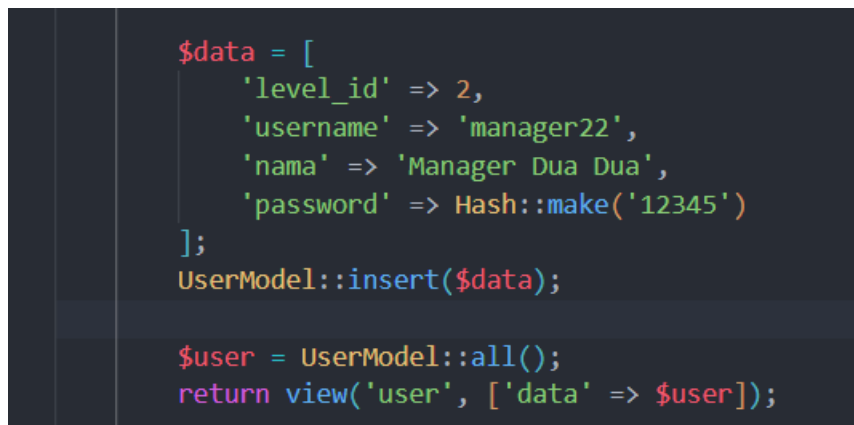
21. Ubah kembali file view dengan nama user.blade.php dan ubah script seperti gambar di bawah ini

```
<meta http-equiv="X-UA-Compatible" content="ie=
<title>Data User</title>
</head>
<body>
<h1>Data User</h1>
<table border="1" cellpadding="2" cellspacing="
    <tr>
        <th>ID</th>
        <th>Username</th>
        <th>Nama</th>
        <th>ID Level Pengguna</th>
    </tr>
    {{-- @foreach ($data as $d)
    <tr>
        <td>{{ $data->user_id }}</td>
        <td>{{ $data->username }}</td>
        <td>{{ $data->nama }}</td>
        <td>{{ $data->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>
</body>
</html>
```

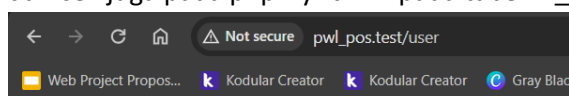
22. Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan



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



24. 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



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
12	cutomer-1	Pelanggan	6
15	manager_dua	Manager 2	2
16	manager_tiga	Manager 3	2
19	manager22	Manager Dua Dua	2

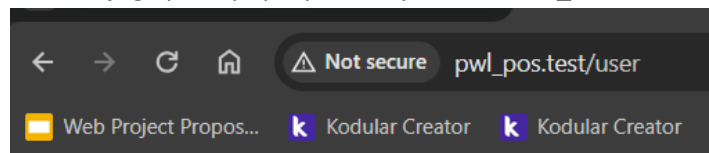


Maka data akan di tambahkan dengan username manager22 dengan anama Manager Dua Dua

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

```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager',
                'nama' => 'Manager',
            ]
        );
        return view('user', ['data' => $user]);
    }
}
```

26. 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



Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

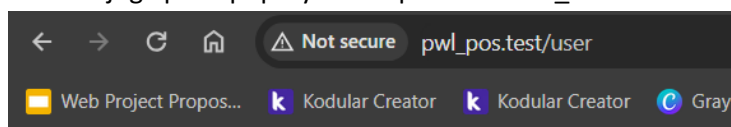
Hasil dari output diatas mengapa seperti itu dikarenakan pada usercontroller untuk menemukan entri dalam tabel UserModel dengan nama pengguna ('username') 'manager' dan nama 'Manager'. Jika tidak ada entri yang ditemukan dengan kriteria tersebut, maka akan dibuat entri baru dengan menggunakan data tersebut.

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



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

28. 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



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
12	cutomer-1	Pelanggan	6
15	manager_dua	Manager 2	2
16	manager_tiga	Manager 3	2
19	manager22	Manager Dua Dua	2
20	manager33	Manager Tiga Tiga	2

Menambahkan data dengan username manager dengan nama Manager Tiga Tiga

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



```
2 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        $user = UserModel::create([
            'username' => 'manager55',
            'nama' => 'Manager55',
            'password' => Hash::make('12345'),
            'level_id' => 2,
        ]);

        $user->username = 'manager56';

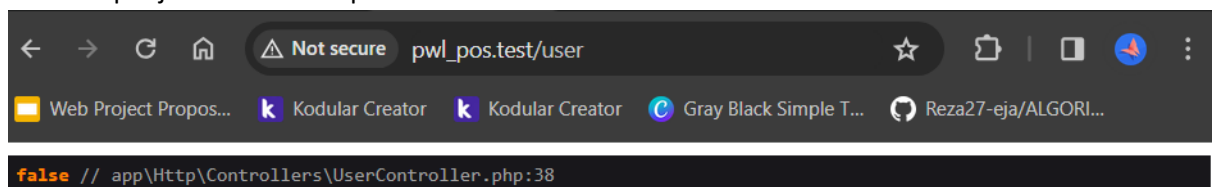
        $user->isDirty();//true
        $user->isDirty('username');//true
        $user->isDirty('nama');//false
        $user->isDirty(['nama', 'username']);//true

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

        $user->save();

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

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



31. Simpan kode program Langkah 3. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan
32. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini



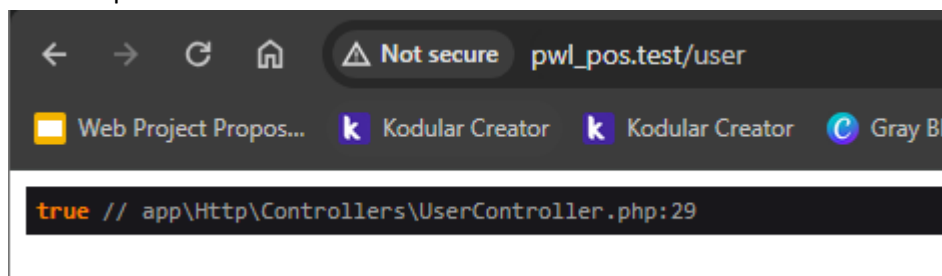
```
1 reference | 0 overrides
public function index()
{
    $user = UserModel::create([
        'level_id' => 2,
        'username' => 'manager11',
        'nama' => 'Manager1',
        'password' => Hash::make('12345')
    ]);

    $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
}
```

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

Hasil output



34. Laporkan hasil Praktikum-2.5 ini dan commit perubahan pada git.
35. Buka file view pada user.blade.php dan buat scripnya menjadi seperti di bawah ini

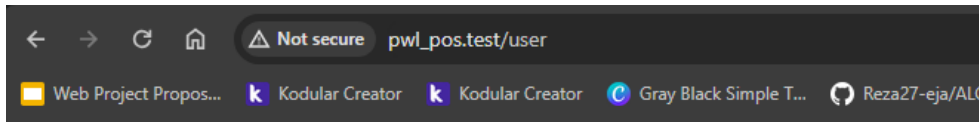


```
resources > views > user.blade.php > @html > @body > @table > @tr > @td
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <title>Data User</title>
8 </head>
9 <body>
10 <h1>Data User</h1>
11 <table border="1" cellpadding="2" cellspacing="0">
12 <tr>
13 <th>ID</th>
14 <th>Username</th>
15 <th>Nama</th>
16 <th>ID Level Pengguna</th>
17 </tr>
18 @foreach ($data as $d)
19 <tr>
20 <td>{{ $d->user_id }}</td>
21 <td>{{ $d->username }}</td>
22 <td>{{ $d->nama }}</td>
23 <td>{{ $d->level_id }}</td>
24 <td>
25 <a href="/user/ubah/{{ $d->user_id }}">Ubah</a>
26 <a href="/user/hapus/{{ $d->user_id }}">Hapus</a>
27 </td>
28 </tr>
29 @endforeach
30 </table>
```

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

```
app > Http > Controllers > UserController.php > PHP Intelephense > UserController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\User;
6 use App\Models\UserModel;
7 use Illuminate\Support\Facades\Hash;
8 use Illuminate\Http\Request;
9
10 2 references | 0 implementations
11 class UserController extends Controller
12 {
13     1 reference | 0 overrides
14     public function index()
15     {
16         $user = UserModel::all();
17         return view('user', ['data' => $user]);
18     }
19 }
```

37. 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	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
12	cutomer-1	Pelanggan	6	Ubah Hapus
15	manager_dua	Manager 2	2	Ubah Hapus
16	manager_tiga	Manager 3	2	Ubah Hapus
19	manager22	Manager Dua Dua	2	Ubah Hapus
20	manager33	Manager Tiga Tiga	2	Ubah Hapus
21	manager56	Manager55	2	Ubah Hapus
22	manager55	Manager55	2	Ubah Hapus
25	manager12	Manager1	2	Ubah Hapus

Hasil outputnya maka akan seperti diatas Dimana dalam view untuk bagian user di tambahkan sebuah button yang digunakan untuk mengedit atau menghapus

38. 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

```
resources > views > user_tambah.blade.php > html
1  <?php
2  <?php
3  <?php
4  <?php
5  <?php
6  <?php
7  <?php
8  <?php
9  <?php
10 <?php
11 <?php
12 <?php
13 <?php
14 <?php
15 <?php
16 <?php
17 <?php
18 <?php
19 <?php
20 <?php
21 <?php
22 <?php
23 <?php
24 <?php
25 <?php
26 <?php
27 <?php
28 <?php
29 <?php
30 <?php
```




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

```
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']);
```

40. 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

```
1 reference | 0 overrides
public function tambah()
{
    return view('user_tambah');
}
```

41. 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

← → ↻ 🏠 Not secure pwl_pos.test/user/tambah

Web Project Propos... Kodular Creator Kodular Creator Gray Black Simple T... Reza27-eja/ALC

Form Tambah Data User

Username

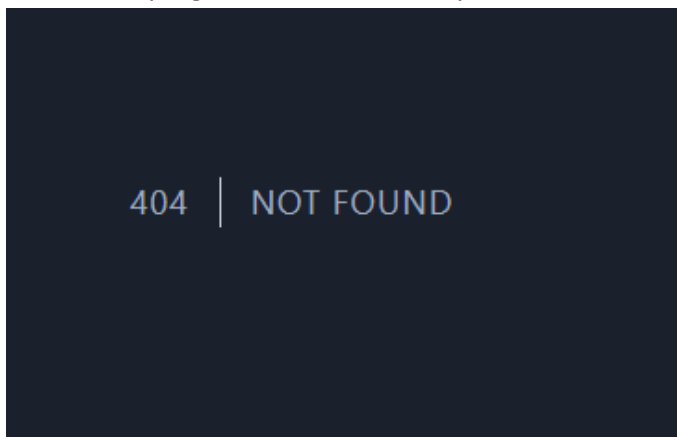
Nama

Password

Level ID



Maka yang terjadi jika kita sudah mengisi data yang tertera pada form maka kita bisa klik simpan namun maka data tidak bisa di proses karena button simpan masih belum diarahkan ke rute yang tersedia maka hasilnya akan not found



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

```
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::get('/user/tambah_simpan', [UserController::class, 'tambah_simpan'])
```

43. 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

```
1 reference | 0 overrides
public function tambah_simpan(Request $request)
{
    UserModel::create([
        'username' => $request->username,
        'nama' => $request->nama,
        'password' => Hash::make('$request->password'),
        'level_id' => $request->level_id
    ]);
}
```

44. 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



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

<http://www.polinema.ac.id>

← → ↺ 🏠 Not secure pwl_pos.test/user

Web Project Propos... Kodular Creator Kodular Creator Gray Black Simple T...

Data User

ID	Username	Nama	ID Level Pengguna	
1	admin	Administrator	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
12	cutomer-1	Pelanggan	6	Ubah Hapus
15	manager_dua	Manager 2	2	Ubah Hapus
16	manager_tiga	Manager 3	2	Ubah Hapus
19	manager22	Manager Dua Dua	2	Ubah Hapus
20	manager33	Manager Tiga Tiga	2	Ubah Hapus
21	manager56	Manager55	2	Ubah Hapus
22	manager55	Manager55	2	Ubah Hapus
25	manager12	Manager1	2	Ubah Hapus
26	manager25	tenza	2	Ubah Hapus
27	manager31	reza	2	Ubah Hapus

☐ Edit Copy Delete

252 manager12 Manager1 \$2y\$12\$Af6wZ3fS2xaC5sXLWS72suBI9hwklav3bmzkxz0Qp8... 2024-03-07 13:02:14 2024-03-07 13:02:14

☐ Edit Copy Delete

262 manager25 tenza \$2y\$12\$zcSVvEjA4z/STCihgdOgLe8Kc DPI2a6M7fBRg6bS0B... 2024-03-07 14:26:55 2024-03-07 14:26:55

Hasil dari kode program diatas dimana kita membuat form tambah data user pada view user_tambah dan pada form tersebut menggunakan method post yang akan digunakan untuk mengirimkan data lalu jika sudah mengisi form tersebut maka akan mengarah ke route yang sesuai rutenya lalu akan dikirimkan data yang sudah kita inputkan ke database maka hasil outputnya akan seperti gambar diatas

- Langkah berikutnya membuat update atau ubah data user dengan cara bikin file baru pada view dengan nama user_ubah.blade.php dan buat scripturnya menjadi seperti di bawah ini



```
</head>
<body>
  <h1>Form Tambah Data User</h1>
  <form method="POST" action="/user/tambah_simpan">

    {{ csrf_field() }}
    {{ method_field('input')}}

    <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>
```

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

```
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']);
```

47. 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

```
1 reference | 0 overrides
public function ubah($id)
{
    $user = UserModel::find($id);
    return view('user_ubah', ['data' => $user]);
}
```

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



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

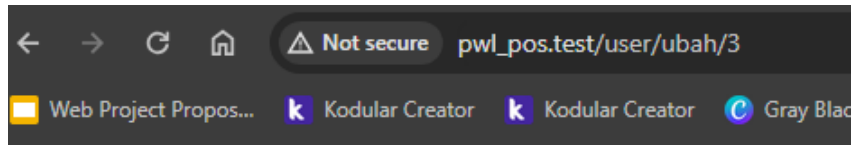
POLITEKNIK NEGERI MALANG

JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141

Telp. (0341) 404424 – 404425, Fax (0341) 404420

<http://www.polinema.ac.id>



Form Ubah Data User

Username

Nama

Password

Level ID

Hasil erro karena pada router masih belum dikoneksikan atau di arahkan ke ubah_simpan

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

```
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']);
```

50. 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 in

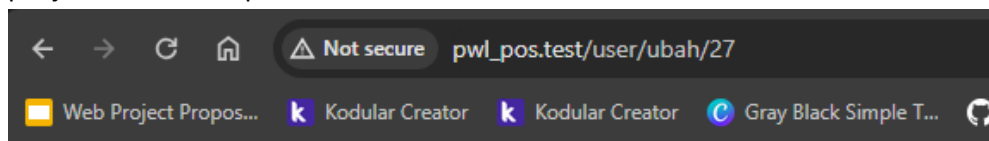


```
1 reference | 0 overrides
public function ubah_simpan($id, Request $request)
{
    $user = UserModel::find($id);

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

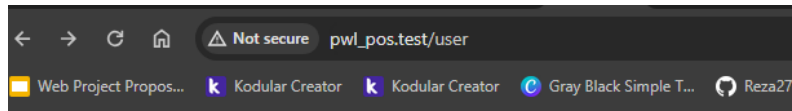
    $user->save();
    return redirect('/user');
}
```

51. 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



Form Ubah Data User

Username	<input type="text" value="manager55"/>
Nama	<input type="text" value="rizqi reza danuarta"/>
Password	<input type="password" value="....."/>
Level ID	<input type="text" value="12345"/>
<input type="button" value="Simpan"/>	



Data User

ID	Username	Nama	ID Level Pengguna	
1	admin	Administrator	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
12	customer-1	Pelanggan	6	Ubah Hapus
15	manager_dua	Manager 2	2	Ubah Hapus
16	manager_tiga	Manager 3	2	Ubah Hapus
19	manager22	Manager Dua Dua	2	Ubah Hapus
20	manager33	Manager Tiga Tiga	2	Ubah Hapus
21	manager56	Manager55	2	Ubah Hapus
22	manager55	Manager55	2	Ubah Hapus
25	manager12	Manager1	2	Ubah Hapus
26	manager25	tenza	2	Ubah Hapus
27	manager31	reza danuarta	2	Ubah Hapus
28	manager2222	test	2	Ubah Hapus

Maka data user yang sesuai id yang ingin kita ubah maka akan keubah karena didalam kode program diatas menggunakan parameter id yang Dimana kita ingin mengubah data dengan id user berapa maka akan keubah sesuai id yang kita tentukan

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

```
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']);
```

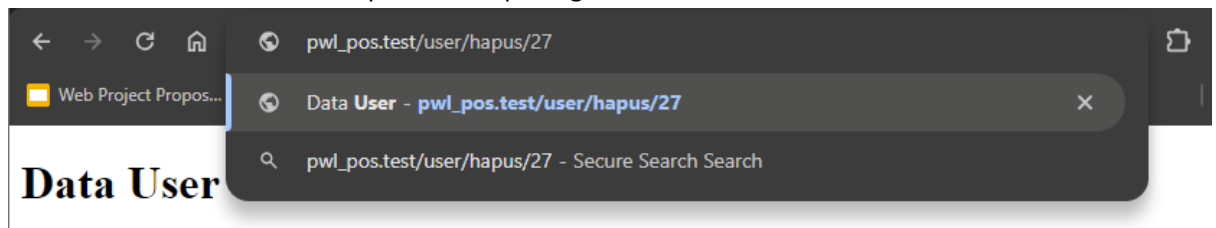
53. Tambahkan script pada controller dengan nama file UserController.php. Tambahkan script dalam class dan buat method baru dengan nama hapus dan diletakan di bawah method ubah_simpan seperti gambar di bawah ini



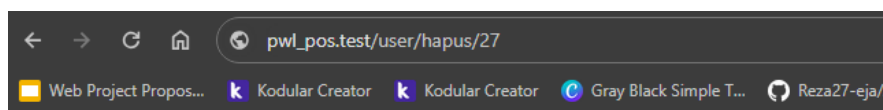
```
1 reference | 0 overrides
public function hapus($id)
{
    $user = UserModel::find($id);
    $user->delete();

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

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



Jika kita mengetikkan routenya seperti gambar diatas maka id yang kita ketikkan pada rute tersebut secara otomatis pada table dan database akan terhapus



Data User

ID	Username	Nama	ID Level Pengguna	
1	admin	Administrator	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
12	cutomer-1	Pelanggan	6	Ubah Hapus
15	manager_dua	Manager 2	2	Ubah Hapus
16	manager_tiga	Manager 3	2	Ubah Hapus
19	manager22	Manager Dua Dua	2	Ubah Hapus
20	manager33	Manager Tiga Tiga	2	Ubah Hapus
21	manager56	Manager55	2	Ubah Hapus
22	manager55	Manager55	2	Ubah Hapus
25	manager12	Manager1	2	Ubah Hapus
26	manager25	tenza	2	Ubah Hapus



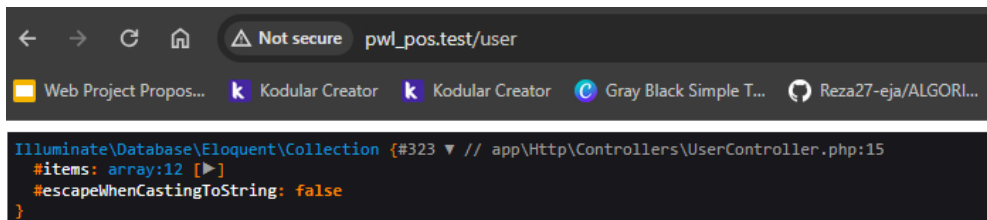
55. Buka file model pada UserModel.php dan tambahkan scripnya menjadi seperti di bawah ini

```
0 references | 0 overrides
public function level(): BelongsTo
{
    return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
}
```

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

```
1 reference | 0 overrides
public function index()
{
    $user = UserModel::with('level')->get();
    dd($user);
}
```

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



Tujuan dari kode program diatas digunakan untuk mengambil data pengguna beserta level mereka dari basis data setelah data berhasil di ambil maka hasilnya akan ditampilkan menggunakan fungsi dd() yang membantu dalam debugging dan memeriksa hasil dari operasi database yang dilakukan

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



```
7 references | 0 implementations
class UserController extends Controller
{
    1 reference | 0 overrides
    public function index()
    {
        $user = UserModel::with('level')->get();
        return view('user', ['data' => $user]);
    }
}
```

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

```
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Data User</title>
</head>
<body>
    <h1>Data User</h1>
    <table border="1" cellpadding="2" cellspacing="0">
        <tr>
            <td>ID</td>
            <td>Username</td>
            <td>Nama</td>
            <td>ID Level Pengguna</td>
            <td>Kode Level</td>
            <td>Nama Level</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>{{ $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>
</body>
```

60. Simpan kode program Langkah 4 dan 5. Kemudian jalankan link pada browser, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan 7. Laporkan hasil Praktikum-2.7 ini dan commit perubahan pada git.

Tambahan kode program diatas digunakan lebih memeperjelaskan hasil outputnya yang Dimana di kode program sebelumnya menggunakan fungsi dd() namun sekarang di return di view user dan terdapat tambahan kolom Kode Level Nama Level maka hasil outputnya akan seperti berikut ini



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

<http://www.polinema.ac.id>

← → ↻ 🏠 ⚠ Not secure pwl_pos.test/user

📄 Web Project Propos... 📄 Kodular Creator 📄 Kodular Creator 🌐 Gray Black Simple T... 🔄 Reza27-eja/ALGORI...

Data User

ID	Username	Nama	ID Level Pengguna	Kode Level	Nama Level	Aksi
1	admin	Administrator	1	ADM	Adminstrator	Ubah Hapus
2	manager	Manager	2	MNG	Manager	Ubah Hapus
3	staff	Staff/Kasir	3	STF	Staff/Kasir	Ubah Hapus
12	cutomer-1	Pelanggan	6	CUS	Pelanggan	Ubah Hapus
15	manager_dua	Manager 2	2	MNG	Manager	Ubah Hapus
16	manager_tiga	Manager 3	2	MNG	Manager	Ubah Hapus
19	manager22	Manager Dua Dua	2	MNG	Manager	Ubah Hapus
20	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah Hapus
21	manager56	Manager55	2	MNG	Manager	Ubah Hapus
22	manager55	Manager55	2	MNG	Manager	Ubah Hapus
25	manager12	Manager1	2	MNG	Manager	Ubah Hapus
26	manager25	tenza	2	MNG	Manager	Ubah Hapus

Berikut ini adalah link repository github untuk pertemuan 4