

JOBSHEET 4

LAPORAN TUGAS

**Diajukan untuk Memenuhi Tugas
Pemograman Web Lanjut**

Jurusan:

TEKNOLOGI INFORMASI

Program Studi:

TEKNIK INFORMATIKA

Disusun oleh

1. Afrizal Dwi Septian (2241720122) / 01 / 2H



**POLITEKNIK NEGERI MALANGTEKNIK
INFORMATIKA**

2023

Mata Kuliah : Pemrograman Web Lanjut (PWL)

Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis

Semester : 4 (empat) / 6 (enam)

Pertemuan ke- : 1 (satu)

JOBSHEET 04 MODEL dan ELOQUENT ORM

A. Praktikum 1 – \$fillable

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

```
app > Models > UserModel.php > UserModel
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class UserModel extends Model
9  {
10     use HasFactory;
11
12     protected $table = 'm_user';
13     protected $primaryKey = 'user_id';
14
15     protected $fillable = ['level_id', 'username', 'nama'];
16 }
```

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

```
//JS 4
$data = [
    'level_id' => 2,
    'username' => 'manager_dua',
    'nama' => 'Manager 2',
    'password' => Hash::make('12345'),
];
UserModel::create($data);

//coba akses userModel
$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	3
7	customer-1	Pelanggan Pertama	4
8	manager_dua	Manager 2	2

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

```
protected $fillable = ['level_id', 'username', 'nama'];
```
5. Ubah kembali file controller UserController.php seperti pada gambar di bawah hanya bagian array pada \$data

```
Illuminate\Database\QueryException PHP 8.3.3 10.47.0
SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value
INSERT INTO `m_user` (`level_id`, `username`, `nama`, `updated_at`, `created_at`) VALUES (2, mo
```

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

```
Illuminate\Database\QueryException PHP 8.3.3 10.47.0
SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value
INSERT INTO `m_user` (`level_id`, `username`, `nama`, `updated_at`, `created_at`) VALUES (2, mo
```

7. Laporkan hasil Praktikum-1 ini dan commit perubahan pada git.
Jawab: Screenshoot ada disetiap langkah

B. Praktikum 2.1 – Retrieving Single Models

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

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

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

```
1 <!DOCTYPE html>
2 <head>
3     <title>Data User</title>
4 </head>
5 <body>
6     <h1>Data User</h1>
7     <table border="1" cellpadding="2" cellspacing="0">
8         <tr>
9             <th>ID</th>
10            <th>Username</th>
11            <th>Nama</th>
12            <th>ID Level Pengguna</th>
13        </tr>
14        <tr>
15            <td>{{ $data->user_id }}</td>
16            <td>{{ $data->username }}</td>
17            <td>{{ $data->nama }}</td>
18            <td>{{ $data->level_id }}</td>
19        </tr>
20    </table>
21 </body>
22 </html>
```

3. Simpan kode program Langkah 1 dan 2. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan

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

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

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

5. Simpan kode program Langkah 4. Kemudian jalankan link <http://localhost:8000/user> 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

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

7. Simpan kode program Langkah 6. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan

Data User			
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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::findor(1, ['username', 'nama'], function () {
            abort(code: 404);
        });
        return view('user', data: ['data' => $user]);
    }
}
```

9. Simpan kode program Langkah 8. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan

Data User

ID	Username	Nama	ID Level Pengguna
	admin	Administrator	

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

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

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

- Simpan kode program Langkah 10. Kemudian jalankan link <http://localhost:8000/user> pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan

404 | NOT FOUND

- Laporkan hasil Praktikum-2.1 ini dan commit perubahan pada git.
Jawab: Screenshoot ada di masing masing langkah

C. Praktikum 2.2 – No found Exception

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

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

- 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

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

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

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



- Laporkan hasil Praktikum-2.2 ini dan commit perubahan pada git.
Jawab: Screenshoot ada disetiap Langkah

D. Praktikum 2.3 – Retrieving Aggregates

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::where('level_id', 2)->count();
        dd(vars[0]: $user);
        return view('user', data: ['data' => $user]);
    }
}
```

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

```
3 // app\Http\Controllers\UserController.php:15
```

Menampilkan output 3 karena where mencari level_id dengan nilai 2 dan count untuk menghitung data dari where yang dimana terdapat 3 data.

- 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

```

1 <!DOCTYPE html>
2 <head>
3   <title>Data User</title>
4 </head>
5 <body>
6   <h1>Data User</h1>
7   <table border="1" cellpadding="2" cellspacing="0">
8     <tr>
9       <td>Jumlah Pengguna</td>
10    </tr>
11    <tr>
12      <td>{{ $data }}</td>
13    </tr>
14  </table>
15 </body>
16 </html>

```

Data User	
Jumlah Pengguna	3

- Laporkan hasil Praktikum-2.3 ini dan commit perubahan pada git.
Jawab: Screenshoot ada disetiap langkah

E. Praktikum 2.4 – Retrieving or Creating Models

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

```

class UserController extends Controller
{
    public function index()
    {
        // $user = UserModel::where('level_id', 2)->count();
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager',
                'nama' => 'Manager',
            ],
            []
        );
        return view('user', ['data' => $user]);
    }
}

```

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

```

<body>
<h1>Data User</h1>
<table border="1" cellpadding="2" cellspacing="0">
  <tr>
    <th>ID</th>
    <th>Username</th>
    <th>Nama</th>
    <th>ID Level Pengguna</th>
  </tr>
  <tr>
    <td>{{ $data->user_id }}</td>
    <td>{{ $data->username }}</td>
    <td>{{ $data->nama }}</td>
    <td>{{ $data->level_id }}</td>
  </tr>
</table>
</body>

```

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

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

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

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

- Simpan kode program Langkah 4. Kemudian jalankan pada browser dan amati apa yang terjadi dan cek juga pada phpMyAdmin pada tabel m_user serta beri penjelasan dalam laporan

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

Data User

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

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

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

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

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

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


```

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

```

9. Simpan kode program Langkah 8. Kemudian jalankan pada browser dan amati apa yang terjadi dan cek juga pada phpMyAdmin pada tabel m_user serta beri penjelasan dalam laporan

Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

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

```

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

```

11. Simpan kode program Langkah 9. Kemudian jalankan pada browser dan amati apa yang terjadi dan cek juga pada phpMyAdmin pada tabel m_user serta beri penjelasan dalam laporan

Data User

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

12. Laporkan hasil Praktikum-2.4 ini dan commit perubahan pada git.
Jawab: Screenshoot ada disetiap langkah

F. Praktikum 2.5 – Attribute Changes

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::create(
            [
                'username' => 'manager55',
                'nama' => 'Manager55',
                'password' => Hash::make( value: '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(); // false
        $user->isClean('username'); // false
        $user->isClean('nama'); // true
        $user->isClean(['nama', 'username']); // false

        $user->save();

        $user->isDirty(); // false
        $user->isClean(); // true
        dd( vars[0]: $user->isDirty());
    }
}
```

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

```
false // app\Http\Controllers\UserController.php:37
```

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

```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::create(
            [
                'username' => 'manager11',
                'nama' => 'Manager11',
                'password' => Hash::make( value: '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( vars[0]: $user->wasChanged(['nama', 'username'])); // true
    }
}
```

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

```
true // app\Http\Controllers\UserController.php:30
```

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

Jawab: Screenshoot ada disetiap Langkah

G. Praktikum 2.6 – CRUD

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

```
<!DOCTYPE html>
<html>
<head>
<title>Data User</title>
</head>
<body>
<h1>Data User</h1>
<a href="user/tambah">+ Tambah User</a>
<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

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

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

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	3	Ubah Hapus
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
8	manager_dua	Manager 2	2	Ubah Hapus
9	manager_tiga	Manager 3	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
13	manager12	Manager11	2	Ubah Hapus

- 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

```
<!DOCTYPE html>
<html>
<head>
  <title>Form Tambah Data User</title>
</head>
<body>
  <h1>Form Tambah Data User</h1>
  <form method="post" action="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>
```

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

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

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

```

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

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

```

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

Form Tambah Data User

Username

Nama

Password

Level ID

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

```
Route::post('uri: '/user/tambah_simpan', action: [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 di bawah method tambah seperti gambar di bawah ini

```

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

    return redirect(to: '/user');
}

```

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

Form Tambah Data User

Username

Nama

Password

Level ID

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	3	Ubah Hapus
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
8	manager_dua	Manager 2	2	Ubah Hapus
9	manager_tiga	Manager 3	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
13	manager12	Manager11	2	Ubah Hapus
14	afrizalseptian	Afrizal Dwi Septian	1	Ubah Hapus

- 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

```
<!DOCTYPE html>
<html>
<head>
  <title>Form Ubah Data User</title>
</head>
<body>
  <h1>Form Ubah Data User</h1>
  <a href="..">Kembali</a>
  <br><br>

  <form method="post" action="../ubah_simpan/{{ $data->user_id }}">
    {{ csrf_field() }}
    {{ method_field('PUT') }}

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

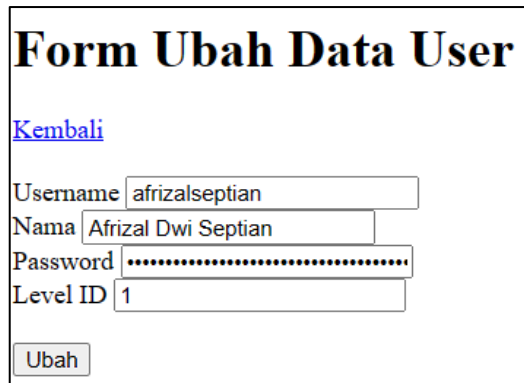
- Tambahkan script pada routes dengan nama file web.php. Tambahkan seperti gambar di bawah ini

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

- 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

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



Form Ubah Data User

[Kembali](#)

Username

Nama

Password

Level ID

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

```
Route::put('/user/ubah_simpan/{id}', action: [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

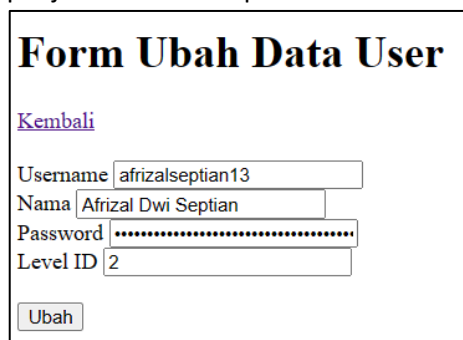
```
public function ubah_simpan($id, Request $request)
{
    $user = UserModel::find($id);

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

    $user->save();

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

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



Form Ubah Data User

[Kembali](#)

Username

Nama

Password

Level ID

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	3	Ubah Hapus
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
8	manager_dua	Manager 2	2	Ubah Hapus
9	manager_tiga	Manager 3	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
13	manager12	Manager11	2	Ubah Hapus
14	afrizalseptian13	Afrizal Dwi Septian	2	Ubah Hapus

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

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

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

```
public function hapus($id)
{
    $user = UserModel::find($id);
    $user->delete();

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

20. Simpan kode program Langkah 18 dan 19. Kemudian jalankan pada browser dan klik tombol hapus, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan

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	3	Ubah Hapus
7	customer-1	Pelanggan Pertama	4	Ubah Hapus
8	manager_dua	Manager 2	2	Ubah Hapus
9	manager_tiga	Manager 3	2	Ubah Hapus
10	manager22	Manager Dua Dua	2	Ubah Hapus
11	manager33	Manager Tiga Tiga	2	Ubah Hapus
12	manager56	Manager55	2	Ubah Hapus
13	manager12	Manager11	2	Ubah Hapus

21. Laporkan hasil Praktikum-2.6 ini dan commit perubahan pada git.
Jawab : Screenshoot ada di masing masing langkah

H. Praktikum 2.7 – Relationship

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

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

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

```
public function index()
{
    $user = UserModel::with(relations: 'level')->get();
    dd(vars[0]: $user);
}
```

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

```
Illuminate\Database\Eloquent\Collection {#320 ▼ // app\Http\Controllers\UserController.php:17
  #items: array:10 [▶]
  #escapeWhenCastingToString: false
}
```

4. Laporkan hasil Praktikum-2.7 ini dan commit perubahan pada git.
Jawab: Screenshoot ada disetiap Langkah