



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

Nama	:	Candra Ahmad Dani
Nim	:	2341720187
Kelas	:	TI-2A

Praktikum 1

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

```
class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user';
    protected $primaryKey = 'user_id';
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    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

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\UserModel;
7  use Illuminate\Support\Facades\Hash;
8
9  class UserController extends Controller
10 {
11     public function index()
12     {
13         $data = [
14             'level_id' => 2,
15             'username' => 'manager_dua',
16             'nama' => 'Manager 2',
17             'password' => Hash::make('12345')
18         ];
19         UserModel::create($data);
20
21         $user = UserModel::all();
22         return view('user', ['data' => $user]);
23     }
24 }
```

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



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

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

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

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

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

Jawab: Akan terjadi error karena kolom password tidak termasuk dalam properti \$fillable.

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

Praktikum 2.1

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

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



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

2. Buka 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>
<td>ID</td>
<td>Username</td>
<td>Nama</td>
<td>ID Level Pengguna</td>
</tr>
<tr>
<td>{{ $data->user_id }}</td>
<td>{{ $data->username }}</td>
<td>{{ $data->nama }}</td>
<td>{{ $data->level_id }}</td>
</tr>
</table>
</body>
```

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

Jawab: Hasilnya akan muncul seperti dibawah karena percobaan diatas mengambil data user dengan primary key bernilai 1 lalu ditampilkan di user.blade.php.

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' => $user]);
    }
}
```

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



Jawab: Hasilnya sama hanya berbeda di pada percobaan ini mencari Dimana level_id = 1.

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' => $user]);
    }
}
```

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

Jawab: Hasilnya sama hanya ditulis lebih ringkas dengan firstWhere().

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(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

Jawab: Pada percobaan ini kode program akan mencari user_id=1, tapi hanya mengambil kolom username dan nama. Jika tidak ditemukan, akan menampilkan halaman 404 Not



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

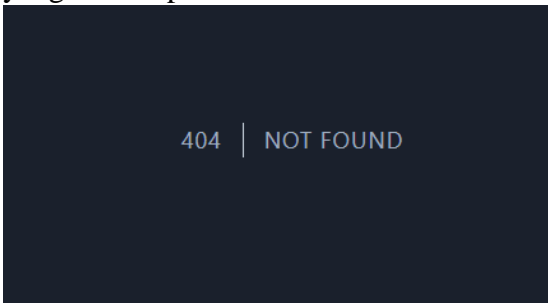
Found.

Data User

ID	Username	Nama	ID Level Pengguna
	admin	Administrator	

10. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini
11. Simpan kode program Langkah 10. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan

Jawab: Hasilnya akan muncul 404 not found, karena menjalankan fungsi abort(404), yang menampilkan halaman 404 Not Found, Karena user_id = 20 tidak ditemukan.



12. Laporkan hasil Praktikum-2.1 ini dan commit perubahan pada git.

Praktikum 2.2

1. Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini
2. Simpan kode program Langkah 1. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan

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

Jawab: Pada percobaan ini findOrFail(1) mencari user dengan user_id = 1, dan jika tidak ditemukan maka akan menampilkan 404 not found, kodenya lebih simpel karena



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

langsung menangani error.

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' => $user]);
    }
}
```

4. Simpan kode program Langkah 3. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan
Jawab: Pada percobaan ini mencari username='manager' dan akan ditampilkan pada ke user.blade.php, dan jika tidak ditemukan maka akan menampilkan 404 not found.

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

- Laporkan hasil Praktikum-2.2 ini dan commit perubahan pada git.
- Ubah file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini
- Simpan kode program Langkah 1. Kemudian jalankan pada browser dan amati apa yang terjadi dan beri penjelasan dalam laporan
Jawab:



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

Jawab: Percobaan ini me

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

3. Buat agar jumlah script pada langkah 1 bisa tampil pada halaman browser, sebagai contoh bisa lihat gambar di bawah ini dan ubah script pada file view supaya bisa muncul datanya

Data User

Jumlah Pengguna
2

Jawab: Dengan tidak mengeksekusi dd(\$user). Dan menambahkan kode pada user.blade.php untuk menampilkan jumlah pengguna.

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

<tr>
    <th>Jumlah Pengguna</th>
</tr>
<tr>
    <td>{{ $data }}</td>
</tr>
```

Data User

Jumlah Pengguna
2

Praktikum 2.4



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

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

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

```
<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>
    </tr>
    <tr>
        <td>{{ $data->user_id }}</td>
        <td>{{ $data->username }}</td>
        <td>{{ $data->nama }}</td>
        <td>{{ $data->level_id }}</td>
    </tr>
</table>
</body>
```

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

Jawab: Pada percobaan ini mencari user dengan username = 'manager' dan nama = 'Manager' dalam tabel m_user. Jika ditemukan, data user tersebut dikembalikan. Jika tidak ditemukan, maka akan membuat user baru dengan data yang diberikan

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

5. 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

Jawab: username = 'manager22', nama = 'Manager Dua Dua', password, dan level_id = 2 dalam tabel m_user. Jika ditemukan, data tersebut dikembalikan. Jika tidak ditemukan, maka akan membuat user baru dengan data yang diberikan

6	2	manager22	Manager Dua Dua	\$2y\$12\$Z2X1wnBtTQfRI5zdatGwuvZ4IRDLeW4AH1BxlgakUU...	2025-03-07 16:11:18	2025-03-07 16:11:18
---	---	-----------	-----------------	---	---------------------	---------------------

Data User

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

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

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

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



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

Jawab: Kode ini mencari user dengan username = 'manager' dan nama = 'Manager'; jika tidak ditemukan, maka akan membuat instance baru tanpa menyimpannya ke database.

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

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

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

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

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

Jawab: Pada percobaan ini membuat instance baru tanpa menyimpannya ke database, karena username = 'manager33' tidak ada di database.

Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

Jawab: Pada percobaan ini akan menyimpan user baru ke database jika belum ada, mirip dengan firstOrCreate.

Data User

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

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

Praktikum 2.5



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

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

        $user->username = 'manager56';

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

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

        $user->save();

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

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

Jawab: Pada percobaan ini membuat user baru, mengubah username, lalu mengecek apakah ada perubahan dengan isDirty() dan isClean(). Setelah save(), perubahan tersimpan, sehingga isDirty() menjadi false dan isClean() menjadi true.

```
false // app\Http\Controllers\UserController.php:74
```

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



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

        $user->username = 'manager12';

        $user->save();

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

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

Jawab: Kode ini membuat user, mengubah username, lalu menyimpannya. Setelah save(), wasChanged() mengonfirmasi perubahan pada username (true), sementara nama tetap tidak berubah (false).

```
true // app\Http\Controllers\UserController.php:85
```

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

Praktikum 2.6

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

```
<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>
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

Jawab: Percobaan ini menampilkan semua data user dan menampilkannya dalam tabel dengan kolom ID, username, nama, level pengguna, serta tombol ubah dan hapus berdasarkan user_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/Kasir	3	Ubah Hapus
4	customer-1	Pelanggan	4	Ubah Hapus
5	manager_2	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager56	Manager55	2	Ubah Hapus
9	manager12	Manager11	2	Ubah Hapus

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

```
<body>
<h1>Form Tambah Data User</h1>
<form method="post" action="/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>
```



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

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

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

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

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

7. Simpan kode program Langkah 4 s/d 6. Kemudian jalankan pada browser dan klik link “+ Tambah User” amati apa yang terjadi dan beri penjelasan dalam laporan
Jawab: Percobaan ini membuat method tambah() untuk menampilkan halaman user_tambah.blade.php untuk menambahkan user.

Form Tambah Data User

Username	<input type="text" value="Masukan Username"/>
Nama	<input type="text" value="Masukan Nama"/>
Password	<input type="text" value="Masukan Password"/>
Level ID	<input type="text" value="Masukan ID Level"/>
<input type="button" value="Simpan"/>	

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

```
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 di



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

bawah method tambah seperti gambar di bawah ini

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

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

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

Jawab: Percobaan memasukkan data user.

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/Kasir	3	Ubah Hapus
4	customer-1	Pelanggan	4	Ubah Hapus
5	manager_2	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager56	Manager55	2	Ubah Hapus
9	manager12	Manager11	2	Ubah Hapus
10	yayaya	Saya Setuju	1	Ubah Hapus



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

```
<body>
<h1>Form Ubah Data User</h1>
<a href="/user">Kembali</a>
<br><br>

<form method="post" action="/user/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->username }}">
    <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>
```

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

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

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

```
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

Jawab: Membuat tampilan form ubah user

Form Ubah Data User

[Kembali](#)

Username	<input type="text" value="jadi"/>
Nama	<input type="text" value="Gini lee"/>
Password	<input type="password" value="....."/>
Level ID	<input type="text" value="1"/>
<input type="button" value="Ubah"/>	



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

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

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

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

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

    $user->save();

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

17. Simpan kode program Langkah 15 dan 16. Kemudian jalankan link localhost:8000/user/ubah/1 atau localhost/PWL_POS/public/user/ubah/1 pada browser dan ubah input formnya dan klik tombol ubah, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan
- Jawab: Mengedit data 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
4	customer-1	Pelanggan	4	Ubah Hapus
5	manager_2	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager56	Manager55	2	Ubah Hapus
9	manager12	Manager11	2	Ubah Hapus
10	yayaya	Saya Setuju	1	Ubah Hapus



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

Setelah edit

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
4	customer-1	Pelanggan	4	Ubah Hapus
5	manager_2	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager56	Manager55	2	Ubah Hapus
9	manager12	Manager11	2	Ubah Hapus
10	jadi	Gini lee	1	Ubah Hapus

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

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

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

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
4	customer-1	Pelanggan	4	Ubah Hapus
5	manager_2	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager56	Manager55	2	Ubah Hapus
9	manager12	Manager11	2	Ubah Hapus
10	jadi	Gini lee	1	Ubah Hapus



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

Setelah dihapus

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
4	customer-1	Pelanggan	4	Ubah Hapus
5	manager_2	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager56	Manager55	2	Ubah Hapus
9	manager12	Manager11	2	Ubah Hapus

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

Praktikum 2.7



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

```
class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user';
    protected $primaryKey = 'user_id';
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = ['level_id', 'username', 'nama', 'password'];

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

2. Buka file controller pada UserController.php dan ubah method script menjadi seperti di bawah ini
3. Simpan kode program Langkah 2. Kemudian jalankan link pada browser, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan
Jawab: Menghubungkan UserModel ke LevelModel lewat 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/Kasir	3	Ubah Hapus
4	customer-1	Pelanggan	4	Ubah Hapus
5	manager_2	Manager 2	2	Ubah Hapus
6	manager22	Manager Dua Dua	2	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	Ubah Hapus
8	manager56	Manager55	2	Ubah Hapus
9	manager12	Manager11	2	Ubah Hapus

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

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



Jurusan Teknologi Informasi Politeknik Negeri Malang
JOBSHEET 04 MODEL dan ELOQUENT ORM
Mata Kuliah Pemrograman Web Lanjut (PWL)

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

```
<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>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>
```

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

Jawab: Menambahkan kolom Kode Level dan Nama Level dari tabel m_level. Data ini dihubungkan ke tabel m_user lewat level_id.

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
4	customer-1	Pelanggan	4	CUS	Pelanggan	Ubah Hapus
5	manager_2	Manager 2	2	MNG	Manager	Ubah Hapus
6	manager22	Manager Dua Dua	2	MNG	Manager	Ubah Hapus
7	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah Hapus
8	manager56	Manager55	2	MNG	Manager	Ubah Hapus
9	manager12	Manager11	2	MNG	Manager	Ubah Hapus

7. Laporkan hasil Praktikum-2.7 ini dan commit perubahan pada git.