



Nama : Nimas Septiandini
No : 17
NIM : 2341760087
Kelas : SIB2B

Mata Kuliah : Pemrograman Web Lanjut (PWL) Nimas
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. PROPERTI `$fillable` DAN `$guarded`

1. `$fillable`

Praktikum 1 - `$fillable`:

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'];
}
```

```
class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user'; //mendefinisikan nama tabel yang digunakan
    protected $primaryKey = 'user_id'; //Mendefinisikan primary key dari t

    //JS4 Prak 1: $fillable
    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 }
25
```

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

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

= pada browser dapat dilihat bahwa user dengan username 'manager_dua' telah ditambahkan pada Data User.

4. Ubah file model [UserModel.php](#) seperti pada gambar di bawah ini pada bagian `$fillable`

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

```
app > Models > UserModel.php
9 {
15 protected $fillable = ['level_id', 'username', 'nama'];
```

5. Ubah kembali file controller [UserController.php](#) seperti pada gambar di bawah hanya bagian array pada `$data`

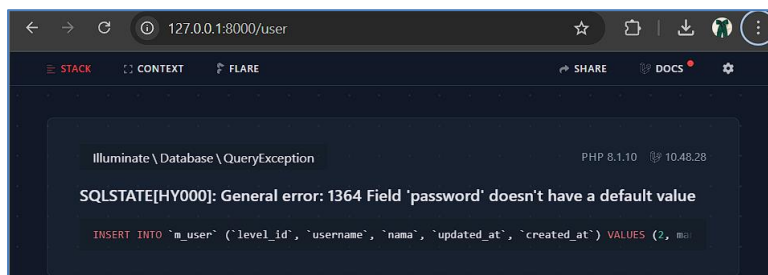
```
public function index()
{
    $data = [
        'level_id' => 2,
        'username' => 'manager_tiga',
        'nama' => 'Manager 3',
        'password' => Hash::make('12345')
    ];
    UserModel::create($data);

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



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

6. Simpan kode program Langkah 4 dan 5. Kemudian jalankan pada *browser* dan amati apa yang terjadi
= Terjadi error saat menampilkan Data User dikarenakan pada saat create data user manager_tiga digunakan password, namun variabel password pada UserModel dihapus.



7. Laporkan hasil Praktikum-1 ini dan *commit* perubahan pada *git*.
GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git

B. RETRIEVING SINGLE MODELS

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

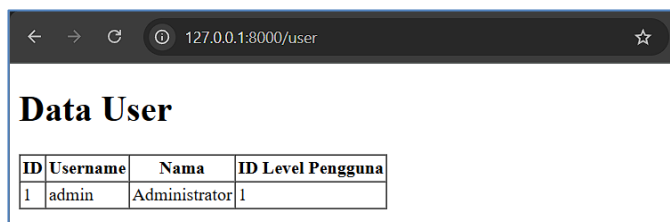
```
app > Http > Controllers > UserController.php  
10 {  
11     public function index(){  
27         $user = UserModel::find(1); //mencari data dengan id 1  
28         return view('user', ['data' => $user]);  
29     }  
30 }
```



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>
<tr>
  <th>ID</th>
  <th>Username</th>
  <th>Nama</th>
  <th>ID Level Pengguna</th>
</tr>
<tr>
  <!--Modifikasi Prak 2.1-->
  <td>{{ $data->user_id }}</td>
  <td>{{ $data->username }}</td>
  <td>{{ $data->nama }}</td>
  <td>{{ $data->level_id }}</td>
</tr>
```

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



ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

- = Pada browser, data user yang ditampilkan hanya user dengan id 1 sesuai modifikasi pada file *UserController* yakni `$user = UserModel::find(1);` yang akan menampilkan user dengan id 1.

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

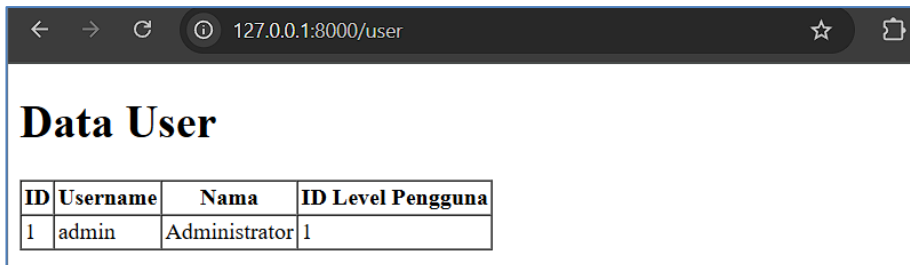
```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::where('level_id', 1)->first();
        return view('user', ['data' => $user]);
    }
}
```



```
$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

= Browser hanya menampilkan Data User dengan id 1. Kode program where yang digunakan memiliki fungsi yang mirip dengan find. Perbedaannya Where akan menampilkan data dengan id 1 yang menjadi default, sedangkan find akan menampilkan semua data dengan id 1 jika data dengan id 1 yang ada berjumlah lebih dari 1.



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

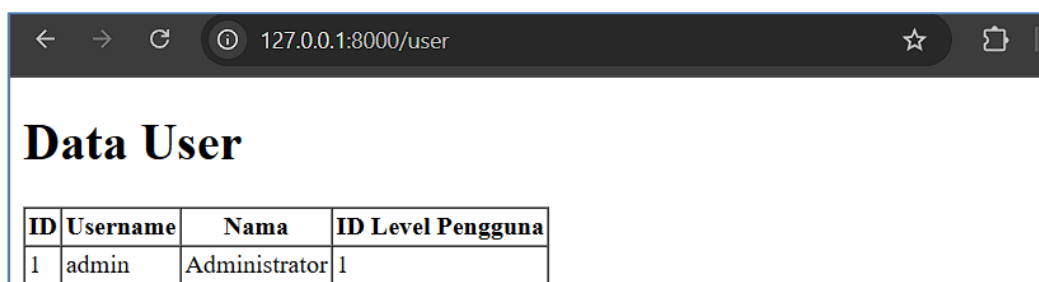
```
class UserController extends Controller  
{  
    public function index()  
    {  
        $user = UserModel::firstWhere('level_id', 1);  
        return view('user', ['data' => $user]);  
    }  
}
```

```
$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

= Hasil yang ditampilkan pada browser tetap sama. Hal ini dikarenakan `firstWhere('level_id', 1)` adalah versi lebih singkat dari `where('level_id', 1)->first()`. `firstWhere()` dapat digunakan jika hanya menggunakan satu kondisi, jika kondisi lebih dari satu bisa menggunakan `where()->first()`.



ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

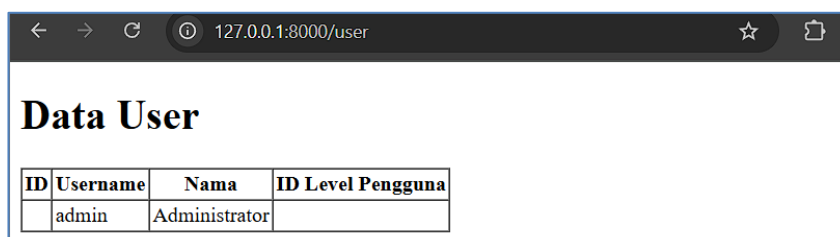
8. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar di bawah ini

```
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

= Kode menggunakan `findOr()` untuk mencari data pengguna dengan ID 1, dan jika tidak ditemukan, akan memanggil `abort(404)`. Karena data pengguna dengan ID 1 ditemukan, halaman menampilkan tabel data pengguna tanpa terjadi error 404.



ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

10. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar



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

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

```
$user = UserModel::findOr(20, ['username', 'nama'], function () {
    abort(404);
});
return view('user', ['data' => $user]);
```

11. Simpan kode program Langkah 10. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan

= data pengguna dengan ID 20 tidak ditemukan sehingga abort 404 akan dipanggil

404 | NOT FOUND

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

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git

Praktikum 2.2 – Not Found Exceptions

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

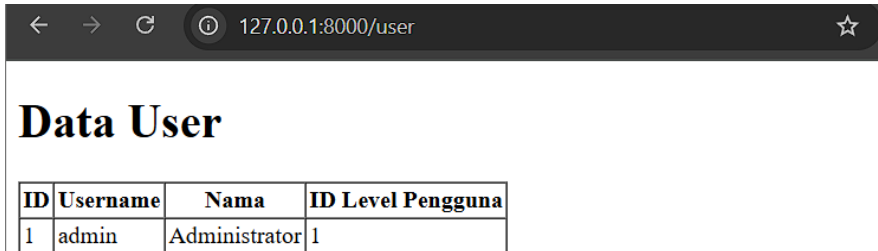
```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::findOrFail(1);
        return view('user', ['data' => $user]);
    }
}
```

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



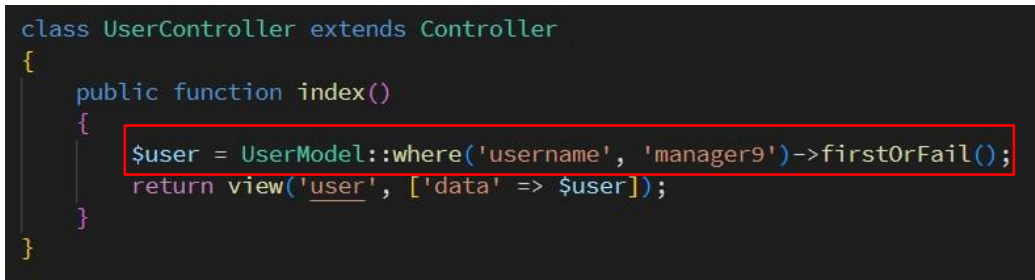
2. Simpan kode program Langkah 1. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan

= Kode menggunakan ``findOrFail(1)``, yang akan mencari data dengan ID 1 dan menampilkan error jika tidak ditemukan. Karena data dengan ID 1 ada, halaman berhasil menampilkan informasi pengguna tanpa error.



ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

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::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

= Kode mencari pengguna dengan ``username` "manager9"` menggunakan ``firstOrFail()``, yang akan menampilkan error 404 jika data tidak ditemukan. Karena "manager9" tidak ada di database, halaman menampilkan error "404 | NOT FOUND".



5. Laporkan hasil Praktikum-2.2 ini dan *commit* perubahan pada *git*.

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



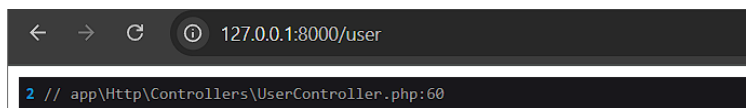
Praktikum 2.3 – Retrieving Aggregates

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

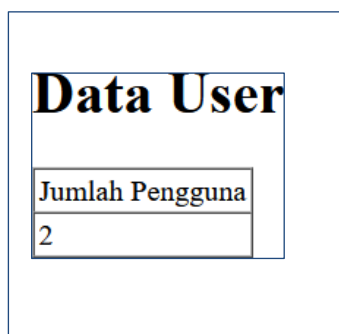
```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::where('level_id', 2)->count();
        dd($user);
        return view('user', ['data' => $user]);
    }
}
```

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

2. Simpan kode program Langkah 1. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan
= Kode menghitung jumlah pengguna dengan `level_id` 2 menggunakan `count()`, lalu hasilnya ditampilkan dengan `dd(\$user)`. Output di browser menunjukkan jumlah pengguna dengan `level_id` 2, yaitu 2.



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





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

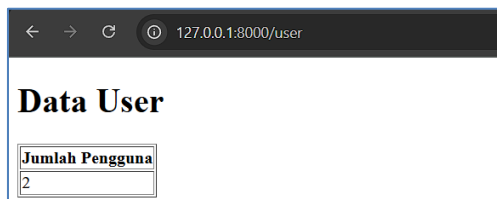
- Modifikasi UserController

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

- Modifikasi user.blade.php

```
resources > views > user.blade.php / ...  
1 <doctype html>  
2 <html>  
3 <head>  
4 <title>Data User</title>  
5 </head>  
6 <body>  
7 <h1>Data User</h1>  
8 <!-- table border="1" cellpadding="2" cellspacing="0" -->  
9 <table border="1" cellpadding="2" cellspacing="0">  
10 <tr>  
11 <th>ID</th>  
12 <th>Username</th>  
13 <th>Nama</th>  
14 <th>ID Level Pengguna</th>  
15 </tr>  
16 <tr>  
17 <!--Modifikasi Prak 2.1-->  
18 <td>{{ $data->user_id }}</td>  
19 <td>{{ $data->username }}</td>  
20 <td>{{ $data->nama }}</td>  
21 <td>{{ $data->level_id }}</td>  
22 </tr>  
23 </table> -->  
24 <table border="1">  
25 <tr>  
26 <th>Jumlah Pengguna</th>  
27 </tr>  
28 <tr>  
29 <td>{{ $data }}</td>  
30 </tr>  
31 </table>  
32 </body>  
33 </html>
```

- Hasil tampilan pada browser



- GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git

Praktikum 2.4 – Retrieving or Creating Models

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]);  
    }  
}
```



```
$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>
```

```
<tr>
    <!--Modifikasi Prak 2.1-->
    <td>{{ $data->user_id }}</td>
    <td>{{ $data->username }}</td>
    <td>{{ $data->nama }}</td>
    <td>{{ $data->level_id }}</td>
</tr>
```

3. Simpan kode program Langkah 1 dan 2. Kemudian jalankan pada *browser* dan amati apa yang terjadi dan beri penjelasan dalam laporan
- = Kode `firstOrCreate()` digunakan untuk mencari atau membuat data baru dengan username "manager" dan nama "Manager". Hasilnya ditampilkan di halaman web dengan format tabel sesuai tampilan di `user.blade.php`.

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2



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

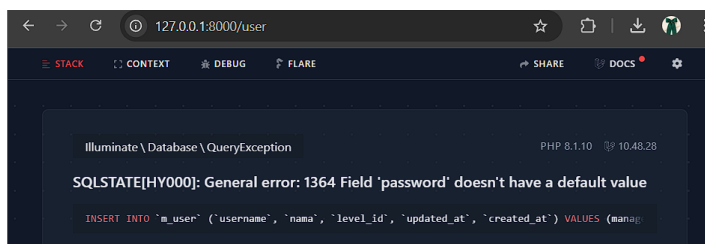
```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager22',
                'nama' => 'Manager Dua Dua',
                'password' => Hash::make('12345'),
                'level_id'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

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

```
$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

= Terjadi error karena variabel password sempat dihapus



- Modifikasi file `UserModel.php` untuk mengembalikan variabel password

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

- Hasil setelah modifikasi

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



	user_id	level_id	username	nama	password	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	admin	Administrator	\$2y\$12\$ShO/Q/FimB.mnqqX52p3Y.rJFnRjYEUHJOnBcKtZx5v...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	manager	Manager	\$2y\$12\$V8Q5JIA8DylakVET7/yZde6O4.0O45cLXdYE38uTmwd...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$3XHggduCoDhcOZ.gHhJ0tes5NKFL2X/1q3H35c79NH...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	2	manager_dua	Manager 2	\$2y\$12\$ScR4ukthrf6LIN12e53Q/LON1gQOS6zDYABQNXHmCUw...	2025-03-09 08:22:29	2025-03-09 08:22:29
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	2	manager22	Manager Dua Dua	\$2y\$12\$yLcGDcWuo2B2IQPwweS3kubsvCs4uYZrpmq4rkPKcMh...	2025-03-09 11:48:12	2025-03-09 11:48:12

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::firstOrNew(
            [
                'username' => 'manager',
                'nama' => 'Manager',
            ],
        );

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

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

= Kode menggunakan `firstOrNew`, yang akan mencari data dengan `username = 'manager'`. Jika tidak ditemukan, objek baru akan dibuat tetapi belum disimpan ke database. Data tetap ditampilkan di view, tetapi tidak akan tersimpan permanen sampai `save()` dipanggil.

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

8. Ubah file controller dengan nama `UserController.php` dan ubah *script* seperti gambar



```
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]);
    }
}
```

```
$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

= Output di browser menampilkan data pengguna dalam bentuk tabel, sesuai dengan data yang diambil dari database menggunakan Eloquent ORM. Data yang ditampilkan mencakup ID, username, nama, dan ID level pengguna.

← → ↻ ⓘ 127.0.0.1:8000/user

Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

	user_id	level_id	username	nama	password	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	admin	Administrator	\$2y\$12\$shO/Q/F.imB.minqqX52p3Y.rJFnRjYEUHJOnBcKtZx5v...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	manager	Manager	\$2y\$12\$V8Q5JIA8DylakVET7/yZde6O4.OO45cLxY38uTmwd...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	staff	Staff/Kasir	\$2y\$12\$3XhGqduCoDhcOZ_gHhJOtes5NkFL2X/1q3H35c79NH...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	2	manager_dua	Manager 2	\$2y\$12\$5cR4ukthF6LIN12e53Q/LON1gQOS6zDYABQNXHmCUw...	2025-03-09 08:22:29	2025-03-09 08:22:29
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	2	manager22	Manager Dua	\$2y\$12\$yLcGdcWuo2B2IQPwweS3kubsvCs4uYZrpmq4kPKcMh...	2025-03-09 11:48:12	2025-03-09 11:48:12

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

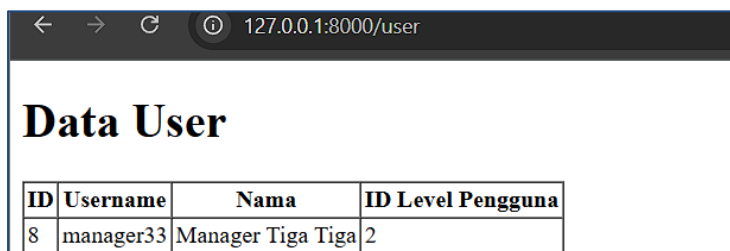


```
class UserController extends Controller
{
    public function index()
    {
        $user = UserModel::firstOrCreate(
            [
                'username' => 'manager33',
                'nama' => 'Manager Tiga Tiga',
                'password' => Hash::make('12345'),
                'level_id'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







```
//Prak 2.4 No 10
$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

= Pada output browser, data pengguna dengan username **manager33** berhasil ditampilkan dengan ID **8**, menunjukkan bahwa data baru telah disimpan ke dalam database karena penggunaan `$user->save()`.



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

		user_id	level_id	username	nama	password	created_at	updated_at
<input type="checkbox"/>	 Edit  Copy  Delete	1	1	admin	Administrator	\$2y\$12\$Sh0/Qf/ImB.mnqX52p3YrJFnRjYEUHJOnBcKtZx5v...	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	2	2	manager	Manager	\$2y\$12\$V8Q5JIA8DylakVET7YzDe6Q4.OO45cLXdY38uTnwd...	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	3	3	staff	Staff/Kasir	\$2y\$12\$3XHgqduCoDhcOZ.gHhJOtes5NkFL2X/Iq3H35c79NH...	NULL	NULL
<input type="checkbox"/>	 Edit  Copy  Delete	6	2	manager_dua	Manager 2	\$2y\$12\$ScR4ukthF6LIN12e53Q/LON1gQOS6zDYABQNXHmCUw...	2025-03-09 08:22:29	2025-03-09 08:22:29
<input type="checkbox"/>	 Edit  Copy  Delete	7	2	manager22	Manager Dua Dua	\$2y\$12\$SyLcGDcWuo2B2IQPwweS3kubsvCs4uYzpmq4rkPKcMh...	2025-03-09 11:48:12	2025-03-09 11:48:12
<input type="checkbox"/>	 Edit  Copy  Delete	8	2	manager33	Manager Tiga Tiga	\$2y\$12\$Rmb9okHZZdd1nB1eipkuoevr1t2JYTr3bGKvWvOI12...	2025-03-09 13:12:34	2025-03-09 13:12:34

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

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



Praktikum 2.5 – Attribute Changes

- 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());
    }
}
```

```
$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']); //false

$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());
```

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

127.0.0.1:8000/user

false // app\Http\Controllers\UserController.php:144

	user_id	level_id	username	nama	password	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	admin	Administrator	\$2y\$12\$H0QIFmB mmoqK52p3YrIFrRy/EUJOnBckz5v...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	manager	Manager	\$2y\$12\$V8Q5WASDyWkVET7Y2e604.0045dX0YE38uTwd...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	staff	StaffKasir	\$2y\$12\$3XHgguC0DheOZ gHqOles5HwFL2XV1q3H35c79HjL...	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	2	manager_dua	Manager 2	\$2y\$12\$RkUthF6L1N12e53QL0N1goO56cDYABONXHmCUw...	2025-03-09 08:22:29	2025-03-09 08:22:29
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	2	manager22	Manager Dua	\$2y\$12\$yL6GDCWuo2B2QpWweS3kubvC4uY2pmq48PKMh...	2025-03-09 11:48:12	2025-03-09 11:48:12
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	2	manager33	Manager Tiga Tiga	\$2y\$12\$Rmb9kH2Zd6f1b1e1gkusev112j7r3GKrVwOH12...	2025-03-09 13:12:34	2025-03-09 13:12:34
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	2	manager56	Manager55	\$2y\$12\$Qz7uWYZTAeId8KqAk06x6mL39uHJ1u605a8ilv...	2025-03-09 13:27:47	2025-03-09 13:27:47



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

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::create([
            'username' => 'manager11',
            'nama' => 'Manager11',
            'password' => Hash::make('12345'),
            'level_id' => 2,
        ]);

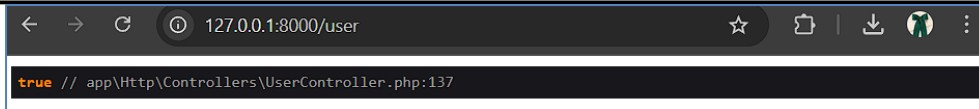
        $user->username = 'manager12';

        $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
    }
}
```

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

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



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

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git

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

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>
```

```
<h1>Data User</h1>
<a href="/user/tambah">+ Tambah User</a>
<table border="1" cellpadding="2" cellspacing="0">
  <tr>
    <th>ID</th>
    <th>Username</th>
    <th>Nama</th>
    <th>ID Level Pengguna</th>
    <th>Aksi</th>
  </tr>
  @foreach ($data as $d)
    <tr>
      <!--Modifikasi Prak 2.1, 2.4, 2.6-->
      <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>
```



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]);
    }
}

$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
= Halaman akan menampilkan seluruh daftar user

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/user". The page title is "Data User". Below the title is a link "+ Tambah User". The main content is a table with 5 columns: ID, Username, Nama, ID Level Pengguna, and Aksi. The table contains 11 rows of user data.

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
6	manager_dua	Manager 2	2	Ubah Hapus
7	manager22	Manager Dua Dua	2	Ubah Hapus
8	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	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>
```

```
<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="Masukkan Username">
    <br>
    <label>Nama</label>
    <input type="text" name="nama" placeholder="Masukkan Nama">
    <br>
    <label>Password</label>
    <input type="password" name="password" placeholder="Masukkan Password">
    <br>
    <label>Level ID</label>
    <input type="number" name="level_id" placeholder="Masukkan ID Level">
    <br><br>
    <input type="submit" class="btn btn-success" value="Simpan">

</form>
```

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

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

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
= Ketika Tambah user ditekan, pengguna akan diarahkan ke form tambah data user

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']);
```

```
Route::get('/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 diletakkan di bawah method *tambah* seperti gambar di bawah ini

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

= Ketika URL */tambah* dijalankan, maka pengguna akan diarahkan menuju form tambah data user yang nantinya akan diarahkan ke *tambah_simpan*.

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
6	manager_dua	Manager 2	2	Ubah Hapus
7	manager22	Manager Dua Dua	2	Ubah Hapus
8	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus
12	manager17	Manager17	2	Ubah Hapus

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>
```

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

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

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

```
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 diletakkan di bawah method `tambah_simpan` seperti gambar di bawah ini

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

```
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
- = Ketika link `Ubah` di klik, maka pengguna akan diarahkan ke halaman form ubah data user

127.0.0.1:8000/user/ubah/12

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}', [UserController::class, 'ubah_simpan']);
```

```
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

= Nama admin berhasil diubah menjadi Adminn. Ketika pengguna mengirim permintaan PUT ke `/user/ubah_simpan/{id}`, sistem akan mencari data pengguna berdasarkan `id`, lalu memperbarui username, nama, password (dengan hashing menggunakan `Hash::make()`), dan level pengguna sesuai input dari request sebelum menyimpannya ke database dan mengarahkan kembali ke halaman daftar pengguna.

Form Ubah Data User

[Kembali](#)

Username

Nama

Password

Level ID

Form Ubah Data User

[Kembali](#)

Username

Nama

Password

Level ID

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Adminn	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']);
```



```
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 diletakkan di bawah method `ubah_simpan` seperti gambar di bawah ini

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

    return redirect('/user');
}

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

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

20. Simpan kode program Langkah 18 dan 19. Kemudian jalankan pada *browser* dan klik tombol `hapus`, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan = Data `manager17` yang sebelumnya ada berhasil dihapus dengan menekan `Hapus`

Data User

+ Tambah User

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Adminn	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
6	manager_dua	Manager 2	2	Ubah Hapus
7	manager22	Manager Dua Dua	2	Ubah Hapus
8	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus
12	manager17	Manager17	2	Ubah Hapus

+ Tambah User

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Adminn	1	Ubah Hapus
2	manager	Manager	2	Ubah Hapus
3	staff	Staff/Kasir	3	Ubah Hapus
6	manager_dua	Manager 2	2	Ubah Hapus
7	manager22	Manager Dua Dua	2	Ubah Hapus
8	manager33	Manager Tiga Tiga	2	Ubah Hapus
10	manager56	Manager55	2	Ubah Hapus
11	manager12	Manager11	2	Ubah Hapus

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



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>

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



Praktikum 2.7 – Relationships

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

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

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

= Output di browser menampilkan hasil `dd($user);` yang merupakan debug dump dari data user yang diambil menggunakan Eloquent dengan relasi `'level'`. Data ditampilkan dalam bentuk `Illuminate\Database\Eloquent\Collection`, menunjukkan daftar objek `'UserModel'` yang diambil dari database.

```
127.0.0.1:8000/user
Illuminate\Database\Eloquent\Collection {#318 // app\Http\Controllers\UserController.php:15}
#items: array:8 [▼]
  0 => App\Models\UserModel {#325 ▶}
  1 => App\Models\UserModel {#326 ▶}
  2 => App\Models\UserModel {#327 ▶}
  3 => App\Models\UserModel {#328 ▶}
  4 => App\Models\UserModel {#329 ▶}
  5 => App\Models\UserModel {#330 ▶}
  6 => App\Models\UserModel {#331 ▶}
  7 => App\Models\UserModel {#332 ▶}
#escapeWhenCastingToString: false
}
```



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]);
}

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

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>
```

```
<table border="1" cellpadding="2" cellspacing="0">
    <tr>
        <th>ID</th>
        <th>Username</th>
        <th>Nama</th>
        <th>ID Level Pengguna</th>
        <th>Kode Level</th>
        <th>Nama Level</th>
        <th>Aksi</th>
    </tr>
    @foreach ($data as $d)
        <tr>
            <!--Modifikasi Prak 2.1, 2.4, 2.6, 2.7-->
            <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>
```

6. Simpan kode program Langkah 4 dan 5. Kemudian jalankan link pada *browser*, kemudian amati apa yang terjadi dan beri penjelasan dalam laporan
= Output pada browser menampilkan tabel berisi data pengguna dari database,



termasuk ID, username, nama, ID level pengguna, kode level, dan nama level, dengan opsi "Ubah" dan "Hapus" untuk setiap entri. Data ini diambil menggunakan Eloquent ORM dengan relasi ke tabel level pengguna.

Data User

[+ Tambah User](#)

ID	Username	Nama	ID Level Pengguna	Kode Level	Nama Level	Aksi
1	admin	Adminn	1	ADM	Administrator	Ubah Hapus
2	manager	Manager	2	MNG	Manager	Ubah Hapus
3	staff	Staff/Kasir	3	STF	Staff/Kasir	Ubah Hapus
6	manager_dua	Manager 2	2	MNG	Manager	Ubah Hapus
7	manager22	Manager Dua Dua	2	MNG	Manager	Ubah Hapus
8	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah Hapus
10	manager56	Manager55	2	MNG	Manager	Ubah Hapus
11	manager12	Manager11	2	MNG	Manager	Ubah Hapus

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

GitHub: https://github.com/septyandini921/PemrogramanWebLanjut_2025.git



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>
