

LAPORAN JOBSHEET 4  
PEMROGRAMAN WEB LANJUT



Disusun Oleh :  
RASENDRIYA DAFA SETIADI  
NIM. 2341720125  
TI-2E / 25

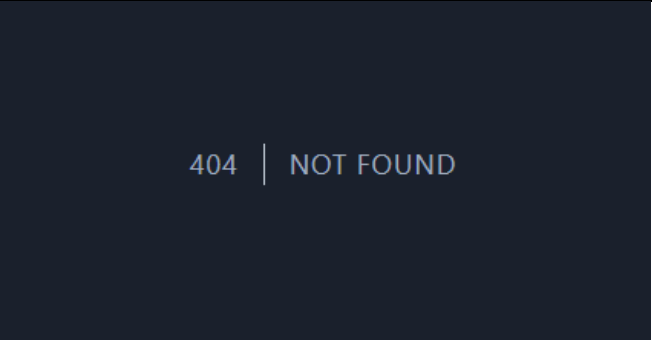
D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG

2024/2025

1.	Fillable																									
a.	<pre>class UserModel extends Model {     use HasFactory;     protected \$table = 'm_user';     protected \$primaryKey = 'user_id';      protected \$fillable = ['level_id', 'username', 'nama', 'password']; }</pre>																									
b.	<pre>class UserController extends Controller {     1 reference   0 overrides   Tabnine   Edit   Test   Explain   Document     public function index(): Factory View{         \$data = [             'level_id' =&gt; 2,             'username' =&gt; 'manager_dua',             'nama' =&gt; 'Manager 2',             'password' =&gt; Hash::make('12345'),             'email' =&gt; 'manager@dua.com',         ];         UserModel::create(\$data);          \$user = UserModel::all();         return view('user', data: ['data' =&gt; \$user]);     } }</pre>																									
c.	<h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr><tr><td>2</td><td>manager</td><td>Manager</td><td>2</td></tr><tr><td>3</td><td>staff</td><td>Staff/Kasir</td><td>3</td></tr><tr><td>4</td><td>customer-1</td><td>pelanggan pertama</td><td>4</td></tr><tr><td>5</td><td>manager_dua</td><td>Manager 2</td><td>2</td></tr></table>		ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1	2	manager	Manager	2	3	staff	Staff/Kasir	3	4	customer-1	pelanggan pertama	4	5	manager_dua	Manager 2	2
ID	Username	Nama	ID Level Pengguna																							
1	admin	Administrator	1																							
2	manager	Manager	2																							
3	staff	Staff/Kasir	3																							
4	customer-1	pelanggan pertama	4																							
5	manager_dua	Manager 2	2																							
d.	<pre>protected \$fillable = ['level_id', 'username', 'nama'];</pre>																									
e.	<pre>public function index(): Factory View{     \$data = [         'level_id' =&gt; 2,         'username' =&gt; 'manager_tiga',         'nama' =&gt; 'Manager 3',         'password' =&gt; Hash::make('12345'),         'email' =&gt; 'manager@dua.com',     ];     UserModel::create(\$data); }</pre>																									
f.	<div>Illuminate \ Database \ QueryException</div> <div>PHP 8.1.10 10:48:28</div> <div>SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value</div> <div>INSERT INTO `m_user` (`level_id`, `username`, `nama`, `updated_at`, `created_at`) VALUES (2, ma</div> <div>Hasil akan error karena kita telah menghapus fillable password pada model UserModel.php.</div>																									

2.1	Retrieving Single Models									
a.	<pre>\$user = UserModel::find(1); return view(view: 'user', data: ['data' =&gt; \$user]); }</pre>									
b.	<pre>&lt;tr&gt;     &lt;th&gt;ID&lt;/th&gt;     &lt;th&gt;Username&lt;/th&gt;     &lt;th&gt;Nama&lt;/th&gt;     &lt;th&gt;ID Level Pengguna&lt;/th&gt; &lt;/tr&gt;</pre> <p>Ctrl+L to chat, Ctrl+K to generate</p> <pre>&lt;tr&gt;     &lt;td&gt;{{ \$data-&gt;user_id }}&lt;/td&gt;     &lt;td&gt;{{ \$data-&gt;username }}&lt;/td&gt;     &lt;td&gt;{{ \$data-&gt;nama }}&lt;/td&gt;     &lt;td&gt;{{ \$data-&gt;level_id }}&lt;/td&gt; &lt;/tr&gt;</pre>									
c.	<h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table> <p>Hasil yang ditampilkan akan mengeluarkan satu data pada tabel user saja karena pada controller kita menggunakan find(1) dari yang sebelumnya menggunakan all().</p>		ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level Pengguna							
1	admin	Administrator	1							
d.	<pre>\$user = UserModel::where('level_id', 1)-&gt;first(); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>									
e.	<h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table> <p>Hasil yang ditampilkan sama, perbedaannya adalah jika pada find(1) mencari berdasarkan primary key dari tabel, pada percobaan ini mencari berdasarkan level_id dan mengambil record pertama yang ditemukan.</p>		ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level Pengguna							
1	admin	Administrator	1							
f.	<pre>\$user = UserModel::firstWhere('level_id', 1); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>									
g.	<h2>Data User</h2> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table> <p>Hasil yang ditampilkan tetap sama, yang berbeda hanya pada penulisan ORM yang lebih simple dan lebih mudah diketik.</p>		ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1
ID	Username	Nama	ID Level Pengguna							
1	admin	Administrator	1							

h.	<pre>\$user = UserModel::findOr(1,['username','nama'], function(): never{     abort(code: 404); }); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>									
i.	<div><h2>Data User</h2><table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td></td><td>admin</td><td>Administrator</td><td></td></tr></table></div> <p>Yang ditampilkan pada saat website di load maka hanya akan muncul username dan nama saja karena pada controller kita hanya menampilkan username dan nama pada findOr.</p>	ID	Username	Nama	ID Level Pengguna		admin	Administrator		
ID	Username	Nama	ID Level Pengguna							
	admin	Administrator								
j.	<pre>\$user = UserModel::findOr(20,['username','nama'], function(): never{     abort(code: 404); }); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>									
k.	<div><div>404   NOT FOUND</div></div> <p>Hasil akan muncul pesan 404 karena pada controller kita meminta record ke 20 sedangkan tidak ada record dengan id 20.</p>									
2.2	Not Found Exceptions									
a.	<pre>\$user = UserModel::findOrFail(1); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>									
b.	<div><h2>Data User</h2><table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td></tr></table></div> <p>Hasil yang ditampilkan adalah record pertama dari tabel user, fungsi dari kode diatas sama seperti dengan percobaan pada saat menggunakan findOr diikuti dengan abort(404); namun lebih ringkas.</p>	ID	Username	Nama	ID Level Pengguna	1	admin	Administrator	1	
ID	Username	Nama	ID Level Pengguna							
1	admin	Administrator	1							
c.	<pre>\$user = UserModel::where('username', 'manager9')-&gt;findOrFail(); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>									

d.	 <p>Hasil yang ditampilkan adalah 404 karena kita mencari username 'manager9'. Sedangkan pada tabel tidak ada data dengan username tersebut.</p>
2.3	Retreiving Aggregates
a.	<pre>\$user = UserModel::where('level_id', 2)-&gt;count(); dd(vars: \$user); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>
b.	<pre>2 // app\Http\Controllers\UserController.php:25</pre> <p>Hasil yang ditampilkan adalah hasil count dengan menambahkan dd yang berfungsi sebagai debugging pada kode.</p>
c.	<pre>&lt;body&gt; &lt;h1&gt;Data User&lt;/h1&gt; &lt;table border="1" cellpadding="2" cellspacing="0"&gt;   &lt;tr&gt;     &lt;th&gt;ID&lt;/th&gt;     &lt;th&gt;Username&lt;/th&gt;     &lt;th&gt;Nama&lt;/th&gt;     &lt;th&gt;Jumlah Pengguna&lt;/th&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;{{ \$data-&gt;user_id }}&lt;/td&gt;     &lt;td&gt;{{ \$data-&gt;username }}&lt;/td&gt;     &lt;td&gt;{{ \$data-&gt;nama }}&lt;/td&gt;     &lt;td&gt;{{ \$data-&gt;count }}&lt;/td&gt;   &lt;/tr&gt; &lt;/table&gt;  \$user = UserModel::where('level_id', 2)-&gt;count(); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>
2.4	Retreiving or Creating Models
a.	<pre>\$user = UserModel::firstOrCreate(     [         'username' =&gt; 'manager',         'nama' =&gt; 'Manager',     ],     [] ); return view(view: 'user', data: ['data' =&gt; \$user]);</pre>

b.

```

<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>

```

c.

## Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

Hasil kode yang ditampilkan adalah data manager karena sudah ada di database, jika tidak ada maka akan membuat data baru.

d.

```

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

```

e.

## Data User

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

☐  Edit  Copy  Delete 6 2 manager22 Manager Dua Dua \$2y\$12\$4b77soCPTGjwzhM

Hasil yang ditampilkan adalah data manager22 karena jika data tidak ditemukan, firstOrCreate akan membuat data baru di database.

f.

## Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

Hasil yang ditampilkan adalah data manager karena data sudah ada pada database.

g.

```

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

```

h.

Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

Fungsi dari findornew hampir sama dengan findorcreate hanya berbeda findornew data tidak disimpan pada database.

j.

```

$user = UserModel::firstOrCreate(
    [
        'username' => 'manager33',
        'nama' => 'Manager Tiga Tiga',
        'password' => Hash::make(12345),
        'level_id' => 2
    ],
);
$user->save();

```

k.

Data User

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

Jika ditambahkan save maka data akan tersimpan pada database dan mendapatkan id.

2.5

Attribute Changes

a.

```

$user = UserModel::create([
    'username' => 'manager55',
    'nama' => 'Manager55',
    'password' => Hash::make(12345),
    'level_id' => 2,
]);

$user->username = 'manager56';
$user->isDirty();
$user->isDirty('username');
$user->isDirty('nama');
$user->isDirty(['nama', 'username']);

$user->isClean();
$user->isClean('username');
$user->isClean('nama');
$user->isClean(['nama', 'username']);
$user->save();

$user->isDirty();
$user->isClean();
dd(vars: $user->isDirty());
return view('user', data: ['data' => $user]);

```

b.	<pre>false // app\Http\Controllers\UserController.php:56</pre>	
	Hasil dari kode adalah penggunaan isdirty dan isclean adalah cara melacak perubahan data pada Laravel, isdirty bernilai true jika ada perubahan dan sebaliknya pada isclean.	
c.	<pre> \$user = UserModel::create([     'username' =&gt; 'manager11',     'nama' =&gt; 'Manager11',     'password' =&gt; Hash::make(12345),     'level_id' =&gt; 2, ]);  \$user-&gt;username = 'manager12'; \$user-&gt;save(); \$user-&gt;wasChanged(); \$user-&gt;wasChanged('username'); \$user-&gt;wasChanged(['username', 'level_id']); \$user-&gt;wasChanged('nama'); \$user-&gt;wasChanged(['nama', 'username']); dd(vars: \$user-&gt;wasChanged(['nama', 'username'])); </pre>	
d.	<pre>false // app\Http\Controllers\UserController.php:56</pre>	
	Method wasChanged() sangat berguna untuk melacak perubahan yang terjadi pada model setelah disimpan ke database. Ini sering digunakan untuk logging atau validasi perubahan data.	
2.6	Create, Read, Update, Delete (CRUD)	
a.	<pre> &lt;html lang="en"&gt; &lt;head&gt;   &lt;meta charset="UTF-8"&gt;   &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;   &lt;title&gt;Data User&lt;/title&gt; &lt;/head&gt; &lt;body&gt;   &lt;h1&gt;Data User&lt;/h1&gt;   &lt;a href="/user/tambah"&gt;+ Tambah user&lt;/a&gt;   &lt;table border="1" cellpadding="2" cellspacing="0"&gt;     &lt;tr&gt;       &lt;th&gt;ID&lt;/th&gt;       &lt;th&gt;Username&lt;/th&gt;       &lt;th&gt;Nama&lt;/th&gt;       &lt;th&gt;ID Level Pengguna&lt;/th&gt;       &lt;th&gt;Aksi&lt;/th&gt;     &lt;/tr&gt;     @foreach (\$data as \$d)       &lt;tr&gt;         &lt;td&gt;{{ \$d-&gt;user_id }}&lt;/td&gt;         &lt;td&gt;{{ \$d-&gt;username }}&lt;/td&gt;         &lt;td&gt;{{ \$d-&gt;nama }}&lt;/td&gt;         &lt;td&gt;{{ \$d-&gt;level_id }}&lt;/td&gt;         &lt;td&gt;&lt;a href="/user/ubah/{{ \$d-&gt;user_id }}"&gt;Ubah&lt;/a&gt;   &lt;a href="/user/       &lt;/tr&gt;     @endforeach   &lt;/body&gt; &lt;/html&gt; </pre>	
b.	<pre> \$user = UserModel::all(); return view(view: 'user', data: ['data' =&gt; \$user]); </pre>	



c.

## Data User

ID	Username	Nama	ID Level Pengguna	
1	admin	Administrator	1	<a href="#">Ubah</a> <a href="#">Hapus</a>
2	manager	Manager	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
3	staff	Staff/Kasir	3	<a href="#">Ubah</a> <a href="#">Hapus</a>
4	customer-1	pelanggan pertama	4	<a href="#">Ubah</a> <a href="#">Hapus</a>
5	manager_dua	Manager 2	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
6	manager22	Manager Dua Dua	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
7	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a> <a href="#">Hapus</a>
8	manager56	Manager55	2	<a href="#">Ubah</a> <a href="#">Hapus</a>

Hasil kode adalah list semua user dengan tambahan aksi ubah dan hapus

d.

```
<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>

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

e.

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

f.

```
public function tambah(): Factory|View{
    return view(view: 'user_tambah');
}
```

g.

```
Symfony\Component\HttpFoundation\Exception\
MethodNotAllowedHttpException                                PHP 8.1.10  10.48.28

The POST method is not supported for route user/tambah_simpan. Supported
methods: GET, HEAD.
```

Hasil error karena pada route form diarahkan pada tambah\_simpan sedangkan pada route belum ditambahkan.

h.

```
Route::post('/user/tambah_simpan', [UserController::class, 'tambah_simpan']);
```

i.

```
public function tambah_simpan(Request $request): RedirectResponse{
    UserModel::create([
        'username' => $request->username,
        'nama' => $request->nama,
        'password' => Hash::make($request->password),
        'level_id' => $request->level_id,
    ]);
    return redirect(to: '/user');
}
```

j.

## Data User

[+ Tambah user](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	<a href="#">Ubah</a>   <a href="#">Hapus</a>
2	manager	Manager	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
3	staff	Staff/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>
4	customer-1	pelanggan pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>
5	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
6	manager22	Manager Dua Dua	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
7	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
8	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
9	admint	admint	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>

Setelah menambahkan route dan controller tambah\_simpan maka data berhasil disimpan.

k.

```
<body>
  <h1>Form Ubah Data User</h1>
  <br><br>

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

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

l.

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

m.

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

n.

404 | NOT FOUND

Hasil akan not found karena belum menambahkan route untuk mengubah datanya.

o.	<pre>public function ubah_simpan(Request \$request,\$id): RedirectResponse {     \$user = UserModel::find(\$id);     \$user-&gt;username = \$request-&gt;username;     \$user-&gt;nama = \$request-&gt;nama;     \$user-&gt;password = Hash::make(\$request-&gt;password);     \$user-&gt;level_id = \$request-&gt;level_id;     \$user-&gt;save();     return redirect(to: '/user'); }</pre>																																																	
p.	9	admins	admin	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																													
Setelah ditambahkan route dan function maka databerhasil diubah																																																		
q.	<pre>public function hapus(\$id): RedirectResponse {     \$user = UserModel::find(\$id);     \$user-&gt;delete();     return redirect(to: '/user'); }</pre>																																																	
r.	<h2>Data User</h2> <p>+ Tambah user</p> <table><tr><th>ID</th><th>Username</th><th>Nama</th><th>ID Level Pengguna</th><th>Aksi</th></tr><tr><td>1</td><td>admin</td><td>Administrator</td><td>1</td><td><a href="#">Ubah</a>   <a href="#">Hapus</a></td></tr><tr><td>2</td><td>manager</td><td>Manager</td><td>2</td><td><a href="#">Ubah</a>   <a href="#">Hapus</a></td></tr><tr><td>3</td><td>staff</td><td>Staff/Kasir</td><td>3</td><td><a href="#">Ubah</a>   <a href="#">Hapus</a></td></tr><tr><td>4</td><td>customer-1</td><td>pelanggan pertama</td><td>4</td><td><a href="#">Ubah</a>   <a href="#">Hapus</a></td></tr><tr><td>5</td><td>manager_dua</td><td>Manager 2</td><td>2</td><td><a href="#">Ubah</a>   <a href="#">Hapus</a></td></tr><tr><td>6</td><td>manager22</td><td>Manager Dua Dua</td><td>2</td><td><a href="#">Ubah</a>   <a href="#">Hapus</a></td></tr><tr><td>7</td><td>manager33</td><td>Manager Tiga Tiga</td><td>2</td><td><a href="#">Ubah</a>   <a href="#">Hapus</a></td></tr><tr><td>8</td><td>manager56</td><td>Manager55</td><td>2</td><td><a href="#">Ubah</a>   <a href="#">Hapus</a></td></tr></table>					ID	Username	Nama	ID Level Pengguna	Aksi	1	admin	Administrator	1	<a href="#">Ubah</a>   <a href="#">Hapus</a>	2	manager	Manager	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>	3	staff	Staff/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>	4	customer-1	pelanggan pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>	5	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>	6	manager22	Manager Dua Dua	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>	7	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>	8	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>
ID	Username	Nama	ID Level Pengguna	Aksi																																														
1	admin	Administrator	1	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																														
2	manager	Manager	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																														
3	staff	Staff/Kasir	3	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																														
4	customer-1	pelanggan pertama	4	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																														
5	manager_dua	Manager 2	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																														
6	manager22	Manager Dua Dua	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																														
7	manager33	Manager Tiga Tiga	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																														
8	manager56	Manager55	2	<a href="#">Ubah</a>   <a href="#">Hapus</a>																																														
2.7	Relationships																																																	
a.	<pre>public function level(): BelongsTo{     return \$this-&gt;belongsTo(LevelModel::class, 'level_id', 'level_id'); }</pre>																																																	
b.	<pre>public function index(): Factory View{     \$user = UserModel::with('level')-&gt;get();     dd(vars: \$user); }</pre>																																																	
c.	<pre>Illuminate\Database\Eloquent\Collection {#318 ▼ // app\Http\Controllers\UserController.php:13   #items: array:8 [▶]   #escapeWhenCastingToString: false }</pre> <p>Hasil dari kode adalah menampilkan debug yang berisi data user beserta data level</p>																																																	
d.	<pre>public function index(): Factory View{     \$user = UserModel::with('level')-&gt;get();     return view(view: 'user', data: ['data' =&gt; \$user]); }</pre>																																																	

e.

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

f.

## Data User

[+ Tambah user](#)

ID	Username	Nama	ID Level Pengguna	Kode Level	Nama Level	Aksi
1	admin	Administrator	1	ADM	Administrator	<a href="#">Ubah</a>   <a href="#">Hapus</a>
2	manager	Manager	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
3	staff	Staff/Kasir	3	STF	Staff/Kasir	<a href="#">Ubah</a>   <a href="#">Hapus</a>
4	customer-1	pelanggan pertama	4	cus	Customer	<a href="#">Ubah</a>   <a href="#">Hapus</a>
5	manager_dua	Manager 2	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
6	manager22	Manager Dua Dua	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
7	manager33	Manager Tiga Tiga	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>
8	manager56	Manager55	2	MNG	Manager	<a href="#">Ubah</a>   <a href="#">Hapus</a>

Hasil pada kode adalah melakukan relasi menggunakan belongsTo pada level dan user yang membuat nama dan kode level bisa ditampilkan pada tabel user.