

Nama : Ereen Lourenza Natalia Mamahi

Kelas : SIB3C

NIM : 2141762044

Laporan Praktikum Jobsheet 4

MODEL dan ELOQUENT ORM

A. Praktikum 1 - \$fillable

1. UserModel.php

```
class UserModel extends Model
{
    use HasFactory;

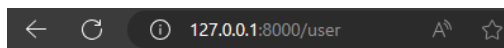
    protected $table = 'm_user'; //mendefinisikan nama tabel yang digunakan oleh model ini
    protected $primaryKey = 'user_id'; //mendefinisikan primary key dari tabel yang digunakan
    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['level_id', 'username', 'nama', 'password'];
}
```

2. UserController.php

```
public function index(){
    //tambah data user dengan Eloquent Model
    $data = [
        'level_id' => 2,
        'username' => 'manager_dua',
        'nama' => 'Manager 2',
        'password' => Hash::make('12345'),
    ];
    UserModel::create($data);

    //coba akses model UserModel
    $user = UserModel::all(); //ambil semua data
    return view('user', ['data' => $user]);
}
```

3. Hasil



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
9	customer-1	Pelanggan Pertama	3
10	manager_dua	Manager 2	2

4. Menghilangkan kolom password pada UserModel.php

```
class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user'; //mendefinisikan nama tabel
    protected $primaryKey = 'user_id'; //mendefinisikan primary key
    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['level_id', 'username', 'nama'];
}
```

5. Tetap memasukkan password pada create data di UserController.php

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

    //coba akses model UserModel
    $user = UserModel::all(); //ambil semua data
    return view('user', ['data' => $user]);
}
```

Terjadi error karena pada UserModel yang disediakan hanya 3 kolom yaitu level_id, username, nama namun pada User Controller ingin memasukkan 4 kolom yaitu level_id, username, nama dan password

```
SQLSTATE[HY000]: General error: 1364 Field 'password' doesn't have a default value
```

```
INSERT INTO `m_user` (`level_id`, `username`, `nama`, `updated_at`, `created_at`) VALUES
```

B. Praktikum 2.1 – Retrieving Single Models

- Menggunakan metode find

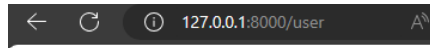
1. UserController.php

```
public function index()
{
    // Ambil model dengan kunci utamanya...
    $user = UserModel::find(1);
    return view('user', ['data' => $user]);
}
```

2. user.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Data User</title>
</head>
<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>
        {{-- @foreach ($data as $d) --}}
        <tr>
            <td>{{ $data->user_id }}</td>
            <td>{{ $data->username }}</td>
            <td>{{ $data->nama }}</td>
            <td>{{ $data->level_id }}</td>
        </tr>
        {{-- @endforeach --}}
    </table>
</body>
</html>
```

3. Hasil



Data User

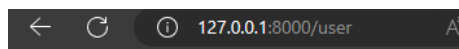
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

- **Metode first()**

1. UserController.php

```
public function index(){  
    // Ambil model dengan kunci utamanya...  
    $user = UserModel::where('level_id', 1)->first();  
    return view('user', ['data' => $user]);  
}
```

2. Hasil



Data User

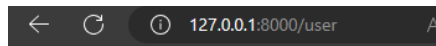
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

- **Metode firstWhere**

1. UserController.php

```
public function index(){  
    // Ambil model dengan kunci utamanya...  
    $user = UserModel::firstWhere('level_id', 1);  
    return view('user', ['data' => $user]);  
}
```

2. Hasil



Data User

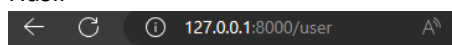
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

- **Metode findOr**

1. UserController.php mencari id 1

```
public function index(){  
    // Ambil model dengan kunci utamanya...  
    $user = UserModel::findOr(1, ['username', 'nama'], function(){  
        abort(404);  
    });  
    return view('user', ['data' => $user]);  
}
```

2. Hasil



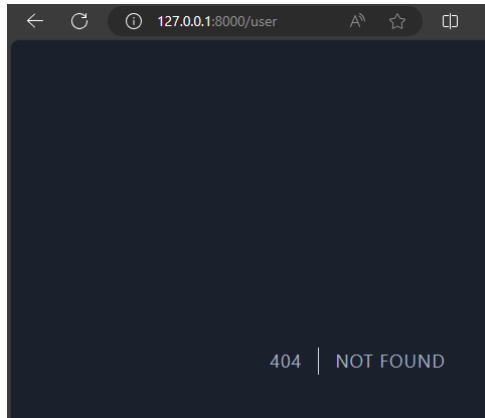
Data User

ID	Username	Nama	ID Level Pengguna
	admin	Administrator	

- UserController.php mencari id 20

```
public function index(){
    // Ambil model dengan kunci utamanya...
    $user = UserModel::findOr(20, ['username','nama'], function(){
        abort(404);
    });
    return view('user', ['data' => $user]);
}
```

- Hasil



- Metode firstOr

- UserController.php mencari id>1

```
public function index(){
    // Ambil model dengan kunci utamanya...
    $user = UserModel::where('level_id','>',1)->firstOr(function(){
        abort(404);
    });
    return view('user', ['data' => $user]);
}
```

- Hasil

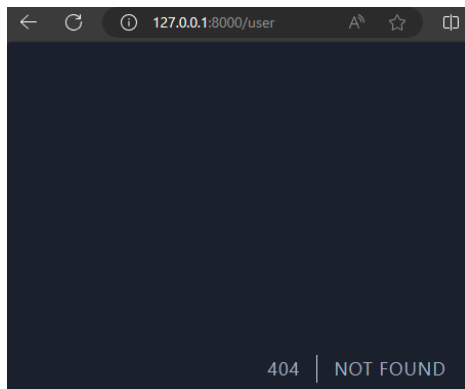
A screenshot of a web browser window with the address bar showing '127.0.0.1:8000/user'. The page title is 'Data User'. Below the title is a table with 4 columns: ID, Username, Nama, and ID Level Pengguna. The table contains one row of data.

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

- UserController.php mencari id>20

```
public function index(){
    // Ambil model dengan kunci utamanya...
    $user = UserModel::where('level_id','>',20)->firstOr(function(){
        abort(404);
    });
    return view('user', ['data' => $user]);
}
```

- Hasil



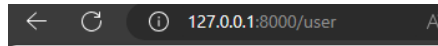
C. Praktikum 2.2 – Not Found Exceptions

- Metode `findOrFail`

1. UserController.php

```
public function index(){
    // Ambil model dengan kunci utamanya...
    $user = UserModel::findOrFail(1);
    return view('user', ['data' => $user]);
}
```

2. Hasil



Data User

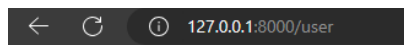
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

- Metode `firstOrFail`

3. UserController.php

```
public function index(){
    // Ambil model dengan kunci utamanya...
    $user = UserModel::where('username', 'manager')->firstOrFail();
    return view('user', ['data' => $user]);
}
```

4. Hasil



Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

Jika hasil ditemukan akan memunculkan data, jika tidak maka akan otomatis fail

D. Praktikum 2.3 – Retrieving Aggregates

1. UserController.php (count)

```
public function index(){
    // Ambil model dengan kunci utamanya...
    $user = UserModel::where('level_id', 2)->count();
    // dd($user);
    return view('user', ['data' => $user]);
}
```

2. user.blade.php

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

3. Hasil



Data User

Jumlah Pengguna
2

E. Praktikum 2.4 – Retrieving or Creating Models

- Metode firstOrCreate

1. UserController.php

```
public function index(){
    $user = UserModel::firstOrCreate(
        [
            'username' => 'manager',
            'nama' => 'Manager'
        ],
    );

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

2. user.blade.php

```
<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>
    {{-- @foreach ($data as $d) --}}
    <tr>
        <td>{{ $data->user_id }}</td>
        <td>{{ $data->username }}</td>
        <td>{{ $data->nama }}</td>
        <td>{{ $data->level_id }}</td>
    </tr>
</table>
```

3. Hasil



Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

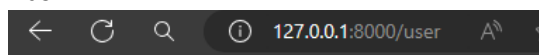
Ditemukan data manager sehingga memunculkan id 2

4. UserController.php

```
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. Hasil



Data User

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

user_id	level_id	username	nama	pass
1	1	admin	Administrator	\$2y\$
2	2	manager	Manager	\$2y\$
3	3	staff	Staff/Kasir	\$2y\$
9	3	customer-1	Pelanggan Pertama	\$2y\$
10	2	manager_dua	Manager 2	\$2y\$
11	2	manager22	Manager Dua Dua	\$2y\$

Saat data tidak ditemukan maka akan membuat data baru dengan id 11

- **Metode firstOrCreate**

1. UserController.php

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

2. Hasil



Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

3. UserController.php

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

4. Hasil

Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

user_id	level_id	username	nama	passw
1	1	admin	Administrator	\$2y\$12
2	2	manager	Manager	\$2y\$12
3	3	staff	Staff/Kasir	\$2y\$12
9	3	customer-1	Pelanggan Pertama	\$2y\$12
10	2	manager_dua	Manager 2	\$2y\$12
11	2	manager22	Manager Dua Dua	\$2y\$12

Tidak ada data baru yang diinsertkan dan hanya disiapkan saja datanya sehingga tidak memiliki id

5. UserController.php

```
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]);
}
```

6. Hasil

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

user_id	level_id	username	nama
1	1	admin	Administrator
2	2	manager	Manager
3	3	staff	Staff/Kasir
9	3	customer-1	Pelanggan Pertama
10	2	manager_dua	Manager 2
11	2	manager22	Manager Dua Dua
12	2	manager33	Manager Tiga Tiga

Terdapat tambahan data jika dilakukan aksi save pada data user sehingga masuk ke database dengan id 12

F. Praktikum 2.5 – Attribute Changes

1. UserController.php

```
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

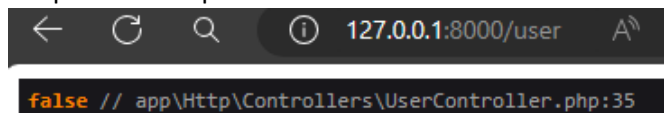
    $user->isClean(); //false
    $user->isClean('username'); //false
    $user->isClean('nama'); //true
    $user->isClean(['nama', 'username']); //false

    $user->save();

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

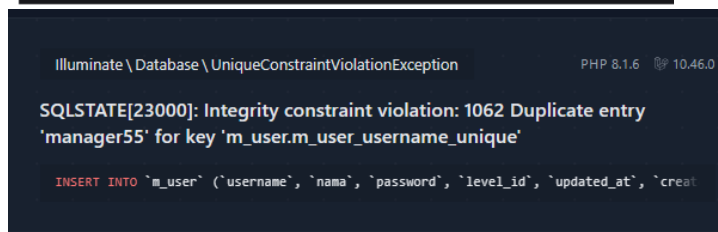
2. Hasil

Output dd merupakan false



← ↻ 🔍 ⓘ 127.0.0.1:8000/user A

false // app\Http\Controllers\UserController.php:35



Illuminate\Database\UniqueConstraintViolationException PHP 8.1.6 10.46.0

SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'manager55' for key 'm_user.m_user_username_unique'

INSERT INTO `m_user` (`username`, `nama`, `password`, `level_id`, `updated_at`, `creat

Jika run kembali maka akan error

user_id	level_id	username	nama	pa
1	1	admin	Administrator	\$2
2	2	manager	Manager	\$2
3	3	staff	Staff/Kasir	\$2
9	3	customer-1	Pelanggan Pertama	\$2
10	2	manager_dua	Manager 2	\$2
11	2	manager22	Manager Dua Dua	\$2
12	2	manager33	Manager Tiga Tiga	\$2
13	2	manager56	Manager55	\$2
14	2	manager55	Manager55	\$2

3. UserController.php

```
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. Hasil

← ↻ 🔍 ⓘ 127.0.0.1:8000/user A

true // app\Http\Controllers\UserController.php:27

20	2	manager12	Manager11
21	2	manager11	Manager11

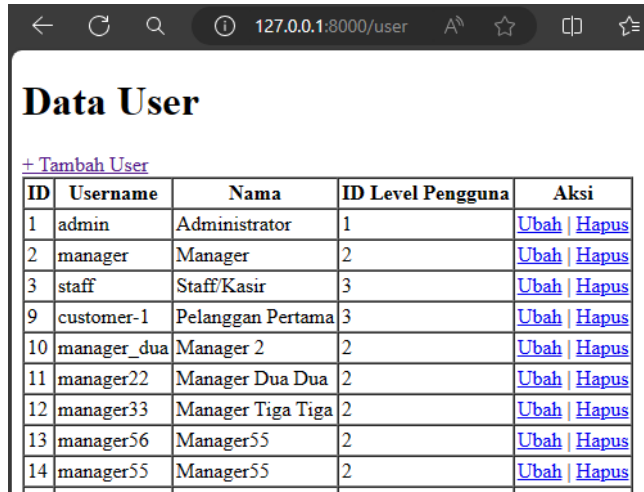
G. Praktikum 2.6 – Create, Read, Update, Delete (CRUD)

- Membuat aksi

1. user.blade.php

```
<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>
            <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
</body>
```

2. UserController.php



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
9	customer-1	Pelanggan Pertama	3	Ubah Hapus
10	manager_dua	Manager 2	2	Ubah Hapus
11	manager22	Manager Dua Dua	2	Ubah Hapus
12	manager33	Manager Tiga Tiga	2	Ubah Hapus
13	manager56	Manager55	2	Ubah Hapus
14	manager55	Manager55	2	Ubah Hapus

- **CREATE/TAMBAH**

1. user_tambah.blade.php

```
<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>
</body>
```

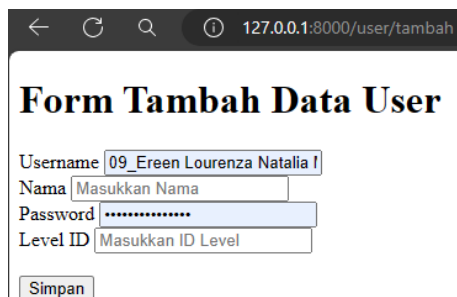
2. Route

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

3. UserController.php

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

4. Hasil



Form Tambah Data User

Username

Nama

Password

Level ID

Membuka url /user/tambah

5. Route tambah_simpan

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

6. UserController.php

```
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');
}
```

7. Input data

← ↻ 🔍 ⓘ 127.0.0.1:8000/user/tambah

Form Tambah Data User

Username

Nama

Password

Level ID

8. Hasil

22	manager112	Manager111	2	Ubah Hapus
23	admin1	Administrator 1	1	Ubah Hapus

- **UPDATE/UBAH**

1. user_ubah.blade.php

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

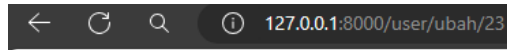
2. Route

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

3. UserController.php

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

4. Hasil



Form Ubah Data User

[Kembali](#)

Username
Nama
Password
Level ID

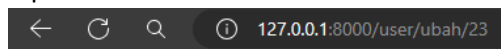
5. Route ubah_simpan

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

6. UserController.php

```
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');  
}
```

7. Input Data



Form Ubah Data User

[Kembali](#)

Username
Nama
Password
Level ID

8. Hasil

23	admin2	Administrator 2	1	Ubah Hapus
----	--------	-----------------	---	--

• DELETE/HAPUS

1. Route hapus

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

2. UserController.php

```
public function hapus($id){  
    $user = UserModel::find($id);  
    $user->delete();  
  
    return redirect('/user');  
}
```

3. Hasil

22	manager112	Manager111	2	Ubah Hapus
----	------------	------------	---	--

H. Praktikum 2.7 – Relationships

1. UserModel.php

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user'; //mendefinisikan nama tabel yang digunakan oleh model ini
    protected $primaryKey = 'user_id'; //mendefinisikan primary key dari tabel yang digunakan
    /**
     * 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. LevelModel.php

```
class LevelModel extends Model
{
    use HasFactory;

    protected $table = 'm_level'; //mendefinisikan nama tabel yang digunakan oleh model ini
    protected $primaryKey = 'level_id'; //mendefinisikan primary key dari tabel yang digunakan
    /**
     * The attributes that are mass assignable
     *
     * @var array
     */
    protected $fillable = ['level_kode', 'level_nama'];
}
```

3. UserController.php

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

4. Hasil

```
Illuminate\Database\Eloquent\Collection {#324 ▼ // app\Http\Controllers\UserController.php:13
  #items: array:14 [▼
    0 => App\Models\UserModel {#331 ▶}
    1 => App\Models\UserModel {#332 ▶}
    2 => App\Models\UserModel {#333 ▶}
    3 => App\Models\UserModel {#334 ▶}
    4 => App\Models\UserModel {#335 ▶}
    5 => App\Models\UserModel {#336 ▶}
    6 => App\Models\UserModel {#337 ▶}
    7 => App\Models\UserModel {#338 ▶}
    8 => App\Models\UserModel {#339 ▶}
    9 => App\Models\UserModel {#340 ▶}
    10 => App\Models\UserModel {#341 ▶}
    11 => App\Models\UserModel {#342 ▶}
    12 => App\Models\UserModel {#343 ▶}
    13 => App\Models\UserModel {#344 ▶}
  ]
  #escapeWhenCastingToString: false
}
```

5. UserController.php

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

6. user.blade.php

```
<th>ID Level Pengguna</th>
<th>Kode Level</th>
<th>Nama Level</th>
<th>Aksi</th>
</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>

```

7. Hasil



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
9	customer-1	Pelanggan Pertama	3	STF	Staff/Kasir	Ubah Hapus
10	manager_dua	Manager 2	2	MNG	Manager	Ubah Hapus
11	manager22	Manager Dua Dua	2	MNG	Manager	Ubah Hapus
12	manager33	Manager Tiga Tiga	2	MNG	Manager	Ubah Hapus
13	manager56	Manager55	2	MNG	Manager	Ubah Hapus
14	manager55	Manager55	2	MNG	Manager	Ubah Hapus
16	manager46	Manager45	2	MNG	Manager	Ubah Hapus
17	manager45	Manager45	2	MNG	Manager	Ubah Hapus
20	manager12	Manager11	2	MNG	Manager	Ubah Hapus
21	manager11	Manager11	2	MNG	Manager	Ubah Hapus
22	manager112	Manager111	2	MNG	Manager	Ubah Hapus