

1. \$fillable dan \$guarded

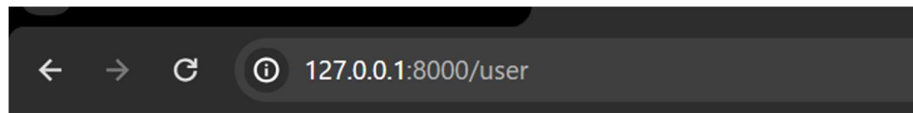
1. Buka file model dengan nama UserModel.php dan tambahkan \$fillable seperti gambar di bawah ini

```
3 references | 0 implementations
8  class UserModel extends Model
9  {
10     use HasFactory;
11
12     0 references
    protected $table = 'm_user';
13     0 references
    protected $primaryKey = 'user_id';
14
15     0 references
    protected $fillable=['level_id','username','nama','password'];
16
17
18 }
```

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

```
22
23     $data = [
24         'level_id' => 2,
25         'nama' => 'manager_dua',
26         'password' => Hash::make('12345')
27     ];
28
29     $user = UserModel::create($data);
30
31
32     $user = UserModel::all();
33     return view('user.user', ['data' => $user]);
34 }
35
36 }
```

3. Simpan kode program Langkah 1 dan 2, dan jalankan perintah web server. Kemudian jalankan link localhostPWL_POS/public/user pada browser dan amati apa yang terjadi



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	administrator	1
2	manager	Manager	2
3	Staff	staff/kasir	3
5	Customer-1	pelanggan pertama	3
6	manager_dua	manager 2	2

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

```
14
15 protected $fillable = ['level_id', 'username', 'nama'];
16
17
18 }
19
```

```
22
23 $data = [
24     'level_id' => 2,
25     'username' => 'manager_tiga',
26     'nama' => 'manager 3',
27     'password' => Hash::make('12345')
28 ];
29
30 $user = UserModel::create($data);
31
```

```
Illuminate \ Database \ QueryException
PHP 8.2.4 10.46.0

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

INSERT INTO `m_user` (`level_id`, `username`, `nama`, `updated_at`, `created_at`) VALUES (2, manager_tiga, manager 3, 2024-03-14 10:14:08, 2024-03-14 10:14:08)
```

2. Guarded

Mass alignment yaitu metode untuk menyimpam model baru dalam satu baris atau pada dasarnya mengatur sekumpulan bidang pada model sekaligus

Atribut yang berada di guarded akan menjadikan atribut tersebut tidak dapat di update atau insert.

B. RETRIEVING SINGLE MODELS

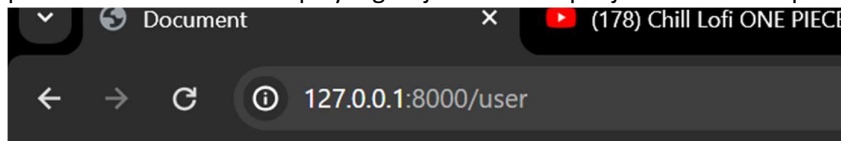
1. Buka file controller dengan nama UserController.php dan ubah script seperti gambar di bawah ini

```
33 // $user = UserModel::all();  
34 $user = UserModel::find(1);  
35 return view('user.user', ['data' => $user]);  
36 }  
37 }  
38
```

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

```
33 // $user = UserModel::all();  
34 $user = UserModel::find(1);  
35 return view('user.user', ['data' => $user]);  
36 }  
37 }  
38
```

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



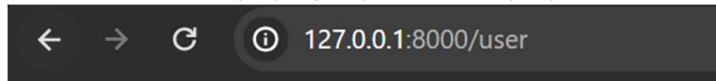
Data User

ID	Username	Nama	ID Level Pengguna
1	admin	administrator	1

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

```
34 $user = UserModel::where('level_id', 1)->first();  
35 return view('user.user', ['data' => $user]);  
36 }  
37 }  
38
```

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



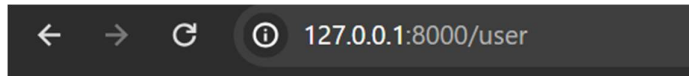
Data User

ID	Username	Nama	ID Level Pengguna
1	admin	administrator	1

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

```
33 // $user = UserModel::all();
34 $user = UserModel::findOr(1, ['username', 'nama'], function () {
35     abort(404);
36 });
37 return view('user.user', ['data' => $user]);
38 }
39 }
40
```

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

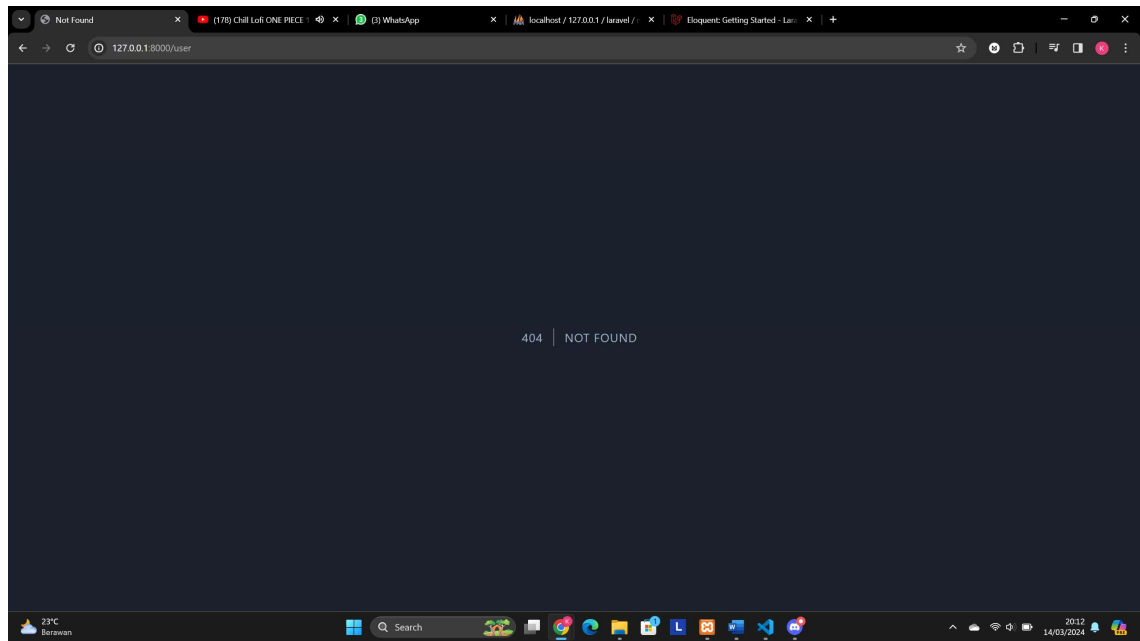


Data User

ID	Username	Nama	ID Level Pengguna
	admin	administrator	

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

```
33 // $user = UserModel::all();
34 $user = UserModel::findOr(20, ['username', 'nama'], function () {
35     abort(404);
36 });
37 return view('user.user', ['data' => $user]);
38 }
39 }
40
```



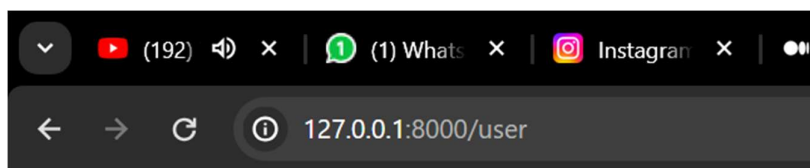
Not Found Exceptions

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

```

37
38     $user = UserModel::findOrFail(1);
39     return view('user.user', ['data' => $user]);
40
41 }
42

```

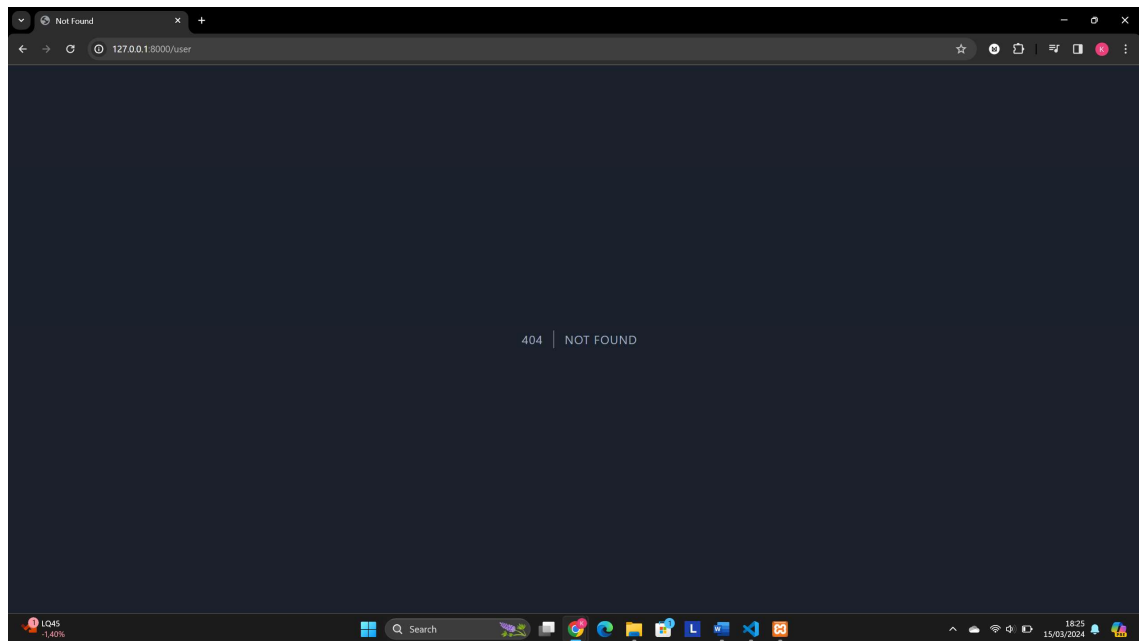


Data User

ID	Username	Nama	ID Level Pengguna
1	admin	administrator	1

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

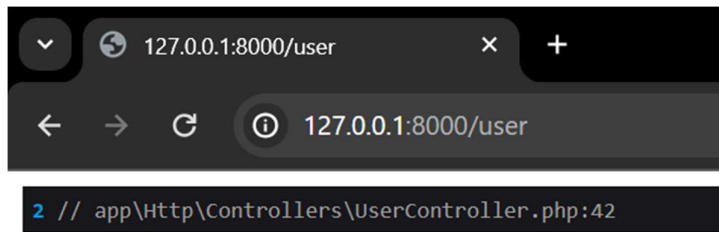
```
38 // $user = UserModel::findOrFail(1);
39 $user = UserModel::where('username', 'manager9')->firstOrFail();
40
41 return view('user.user', ['data' => $user]);
42 }
43
44
```



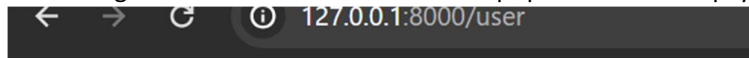
Retrieving Aggregates

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

```
40
41 $user = UserModel::where('level_id', 2)->count();
42 dd($user);
43
44 return view('user.user', ['data' => $user]);
45 }
46
47
```



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



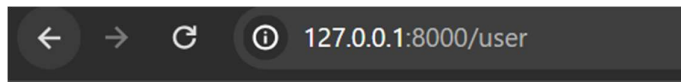
Data User

Jumlah Penggunaa
2

Retreiving or creating models

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

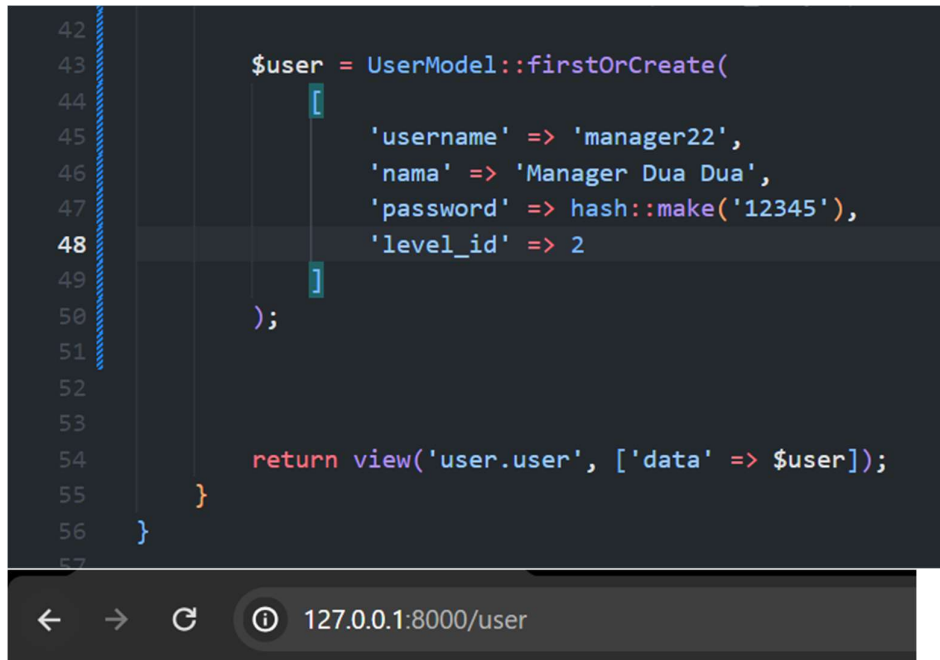
```
41 // $user = UserModel::where('level_id', 2)->count();
42
43 $user = UserModel::firstOrCreate(
44     [
45         'username' => 'manager',
46         'nama' => 'Manager',
47     ]
48 );
49
50
51
52 return view('user.user', ['data' => $user]);
53 }
54 }
55
```



Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

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



Data User

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

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


```
50 // );
51
52 $user= UserModel::firstOrCreate(
53     [
54         'username'=>'manager',
55         'nama'=>'Manager'
56     ]
57 );
58
59 return view('user.user', ['data' => $user]);
60 }
61 }
62
```

← → ↻ ⓘ 127.0.0.1:8000/user

Data User

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

```
51
52 $user = UserModel::firstOrCreate(
53     [
54         'username' => 'manager33',
55         'nama' => 'Manager Tiga Tiga',
56         'password' => hash::make('12345'),
57         'level_id' => 2
58     ]
59 );
60
61 return view('user.user', ['data' => $user]);
62 }
```

← → ↻ ⓘ 127.0.0.1:8000/user

Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

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

```
50 // );
51
52 $user = UserModel::firstOrCreate(
53     [
54         'username' => 'manager33',
55         'nama' => 'Manager Tiga Tiga',
56         'password' => hash::make('12345'),
57         'level_id' => 2
58     ]
59 );
60 $user->save();
61
62 return view('user.user', ['data' => $user]);
```

← → ↻ ⓘ 127.0.0.1:8000/user

Data User

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

Attribute Changes

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

```
61
62     $user = UserModel::create(
63         [
64             'username' => 'manager55',
65             'nama' => 'Manager55',
66             'password' => Hash::make('12345'),
67             'level_id' => 2
68         ]
69     );
70
71     $user->username = 'manager56';
72
73     $user->isDirty();
74     $user->isDirty('username');
75     $user->isDirty('nama');
76     $user->isDirty(['nama', 'username']);
77
78     $user->isClean();
79     $user->isClean('username');
80     $user->isClean('nama');
81     $user->isClean(['nama', 'username']);
82
83     $user->save();
84
85     $user->isDirty();
86     $user->isClean();
87
88     dd($user->isDirty());|
89
90     return view('user.user', ['data' => $user]);
91 }
92
93
```

← → ↻ ⓘ 127.0.0.1:8000/user

false // app\Http\Controllers\UserController.php:88

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

```
61
62     $user = UserModel::create(
63     [
64         'username' => 'manager11',
65         'nama' => 'Manager11',
66         'password' => Hash::make('12345'),
67         'level_id' => 2
68     ]
69 );
70
71 $user->username = 'manager52';
72
73
74 dd($user->wasChanged(['nama', 'username']));
75
```

← → ↻ ⓘ 127.0.0.1:8000/user

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

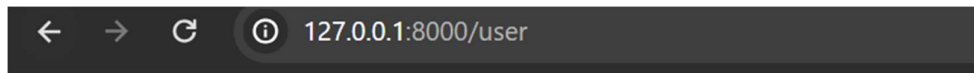
Create, Read, Update, Delete (CRUD)

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

```
2 <html lang="en">
9 <body>
10 <h1>Data User</h1>
11 <table border="1" cellpadding="2" cellspacing="0" >
12 <tr>
13 <th>ID</th>
14 <th>Username</th>
15 <th>Nama</th>
16 <th>ID Level Pengguna</th>
17 <th>Aksi</th>
18 </tr>
19 @foreach ($data as $d)
20
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><a href="/user/ubah">ubah</a> | <a href="/user/hapus/{{ $d->user_id }}">Hapus</a></td>
27 </tr>
28 @endforeach
29
30 </table>
31
32
33
34 </body>
```

2. Buka file controller pada UserController.php dan buat scriptnya untuk read menjadi seperti di bawah ini

```
92 $user = UserModel::all();
93 return view('user.user', ['data' => $user]);
94 }
95
96
```



Data 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
5	Customer-1	pelanggan pertama	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
9	manager56	Manager55	2	ubah Hapus
10	manager11	Manager11	2	ubah Hapus
12	manager13	Manager13	2	ubah Hapus
14	manager25	Manager25	2	ubah Hapus

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

```
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
10     <h1>Form tambah data user</h1>
11
12     <form method="POST" action="/user/tambah_simpan">
13         {{ csrf_field() }}
14         <label for="">username</label>
15         <input type="text" name="username" placeholder="masukkan username" id="">
16         <br>
17         <label for="">nama</label>
18         <input type="text" name="name" placeholder="masukkan name" id="">
19         <br>
20         <label for="">Password</label>
21         <input type="password" name="password" placeholder="masukkan password" id="">
22         <br>
23         <label for="">level ID</label>
24         <input type="number" name="level_id" placeholder="masukkan id level" id="">
25         <br><br>
26         <input type="submit" class="btn btn-success" value="simpan">
27     </form>
28
29 </body>
30 </html>
```

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

```

39
40 Route::get('/user/tambah', [UserController::class, 'tambah']);

```

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

```

1 reference | 0 overrides
96 public function tambah()
97 {
98     return view('user.userTambah');
99 }
100 }
101

```

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

```

1 reference | 0 overrides
101 public function tambah_simpan(Request $request)
102 {
103     UserModel::create(
104         [
105             'username' => $request->username,
106             'nama' => $request->nama,
107             'password' => Hash::make('$request->password'),
108             'level_id' => $request->level_id
109         ]
110     );
111     return redirect('/user');

```

Data User

+ tambah user

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	administrator	1	ubah Hapus
2	manager	Manager	2	ubah Hapus
3	Staff	staff/kasir	3	ubah Hapus
5	Customer-1	pelanggan pertama	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
9	manager56	Manager55	2	ubah Hapus
10	manager11	Manager11	2	ubah Hapus
12	manager13	Manager13	2	ubah Hapus
14	manager25	Manager25	2	ubah Hapus
15	saonoke	krisna	2	ubah Hapus

- Langkah berikutnya membuat update atau ubah data user dengan cara bikin file baru pada view dengan nama user_ubah.blade.php dan buat scriptnya menjadi seperti di bawah ini

```

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 <form method="POST" action="/user/tambah_simpan">
13     {{ csrf_field() }}
14     <label for="">username</label>
15     <input type="text" name="username" placeholder="masukkan username" value="{{ $data->username }}" id="">
16     <br>
17     <label for="">nama</label>
18     <input type="text" name="nama" placeholder="masukkan name" value="{{ $data->name }}" id="">
19     <br>
20     <label for="">Password</label>
21     <input type="password" name="password" placeholder="masukkan password" id="" value="value="{{ $data->password }}">
22     <br>
23     <label for="">level ID</label>
24     <input type="number" name="level_id" placeholder="masukkan id level" id="" value="{{ $data->level_id }}">
25     <br>
26     <input type="submit" class="btn btn-success" value="simpan" >
27 </body>

```

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

```

42
43 Route::get('/user/ubah/{id}', [UserController::class, 'ubah']);
44

```

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


```

14 public function ubah($id)
15 {
16     $user = UserModel::find($id);
17     return view('user.userUbah', ['data' => $user]);
18 }
19 }

```

Form Ubah Data User

[Kembali](#)

username

nama

Password

level ID

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

```

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

```

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

```

120 public function ubah_simpan($id, Request $request)
121 {
122     $user = UserModel::find($id);
123
124     $user->username = $request->username;
125     $user->nama = $request->nama;
126     $user->password = Hash::make($request->password);
127     $user->level_id = $request->level_id;
128
129     $user->save();
130     return redirect('/user');
131 }

```

14	manager23	manager23	2	ubah hapus
15	saonokee	krisna	2	ubah hapus

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


```
5 Route::delete('user/hapus/{id}', [UserController::class, 'hapus']);
6
```

```
132 public function hapus($id)
133 {
134     $user = UserModel::find($id);
135
136     $user->delete();
137     return redirect('/user');
138 }
139 }
```

← → ↺ ⓘ 127.0.0.1:8000/user

Data User

+ [tambah user](#)

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	administrator	1	ubah Hapus
2	manager	Manager	2	ubah Hapus
3	Staff	staff/kasir	3	ubah Hapus
5	Customer-1	pelanggan pertama	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
9	manager56	Manager55	2	ubah Hapus
10	manager11	Manager11	2	ubah Hapus
12	manager13	Manager13	2	ubah Hapus

Relationship

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

```
17 0 references
18     protected $guarded = [];
19
20 0 references | 0 overrides
21     public function level():BelongsTo{
22         return $this->belongsTo(LevelModel::class);
23     }
24 }
```

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

```
0 references | 0 overrides
19 public function level(): BelongsTo
20 {
21     return $this->belongsTo(LevelModel::class, 'level_id');
22 }
23
24 }
25
26 // dd($user->isDirty());
92 $user = UserModel::with('level')->get();
93 dd($user);
94 return view('user.user', ['data' => $user]);
95 }
```

```

#table: "m_user"
#primaryKey: "user_id"
#keyType: "int"
+incrementing: true
#with: []
#withCount: []
+preventsLazyLoading: false
#perPage: 15
+exists: true
+wasRecentlyCreated: false
#escapeWhenCastingToString: false
#attributes: array:7 [▶]
#original: array:7 [▶]
#changes: []
#casts: []
#classCastCache: []
#attributeCastCache: []
#dateFormat: null
#appends: []
#dispatchesEvents: []
#observables: []
#relations: array:1 [▼
  "level" => App\Mode...LevelModel {#298 ▼
    #connection: "mysql"
    #table: "m_level"
    #primaryKey: "level_id"
    #keyType: "int"
    +incrementing: true
    #with: []
    #withCount: []
    +preventsLazyLoading: false
    #perPage: 15
    +exists: true
    +wasRecentlyCreated: false
    #escapeWhenCastingToString: false
    #attributes: array:5 [▼
      "level_id" => 2
      "level_kode" => "MNG"
      "level_nama" => "Manager"
      "created_at" => null
      "updated_at" => null
    ]
    #original: array:5 [▶]
    #changes: []
    #casts: []
    #classCastCache: []
    #attributeCastCache: []
    #dateFormat: null
    #appends: []
    #dispatchesEvents: []
    #observables: []
    #relations: []
    #touches: []
    +timestamps: true
    +usesUniqueIds: false
  }
]

```