Nama : Putri Norchasana

Kelas(No) : TI-2H (24)

NIM : 2241720036

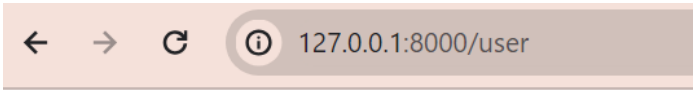
Tugas : Jobsheet 4 ORM

Praktikum 1

1. Fillable

```php
class UserModel extends Model
{
    use HasFactory;

    protected $table = 'm_user';
    protected $primaryKey = 'user_id';
    protected $fillable = ['level_id','username','nama','password'];

}
```

```php
    $data = [
        'level_id'  => 2,
        'username' => 'manager_2',
        'nama' => 'Manager 3',
        'password' => Hash::make('12345'),
    ];

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



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1
2	manager	Manager	2
3	staff	Staff/Kasir	3
4	manager_2	Manager 2	2

```php
protected $table = 'm_user';
protected $primaryKey = 'user_id';
protected $fillable = ['level_id','username','nama'];
```

Penjelasan : Akan terjadi error, karena atribut password tidak didaftarkan, sehingga kolom tersebut tidak bisa kita isi ketika melkukan insert atau update.

2. Guarded

Mass assignment adalah proses di mana sejumlah besar atribut model dalam aplikasi web atau perangkat lunak lainnya dapat diatur secara bersamaan. Dalam konteks Eloquent, yang merupakan bagian dari framework Laravel, mass assignment terjadi ketika kita mencoba menyimpan data ke database dengan memberikan array asosiatif ke model. Eloquent kemudian mencoba untuk mengisi atribut-atribut model dengan nilai-nilai dari array tersebut.

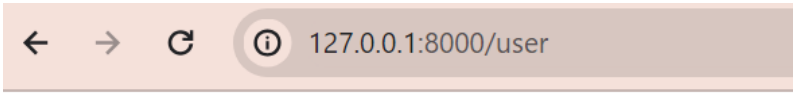
Dalam pengaturan standar Laravel, $guarded adalah array yang berisi daftar atribut yang tidak diizinkan untuk diisi melalui proses mass assignment. Jika sebuah atribut termasuk dalam daftar $guarded, itu berarti bahwa ketika kita mencoba melakukan insert atau update, nilai atribut tersebut akan diabaikan oleh Eloquent. Dengan kata lain, nilai-nilai untuk atribut-atribut yang terdaftar dalam $guarded tidak akan diperbarui atau disimpan ke dalam database ketika menggunakan metode mass assignment.

Praktikum 2.1

Retrieving Single Models

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

```
<table border="1" cellpadding="2" cellspacing="0">

    <tr>
        <th>ID</th>
        <th>Username</th>
        <th>Nama</th>
        <th>ID Level Pengguna</th>
    </tr>      You, 6 days ago • [praktikum 6] Implementasi Eloquent O
    <tr>
        <td>{{$data->user_id}}</td>
        <td>{{$data->username}}</td>
        <td>{{$data->nama}}</td>
        <td>{{$data->level_id}}</td>
    </tr>

</table>
```



Data User

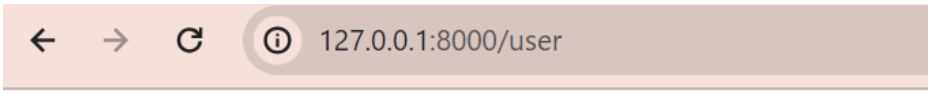
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

```
$user = UserModel::where('level_id',1)->first();
return view('user', ['data' => $user]);
```

Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

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



Data User

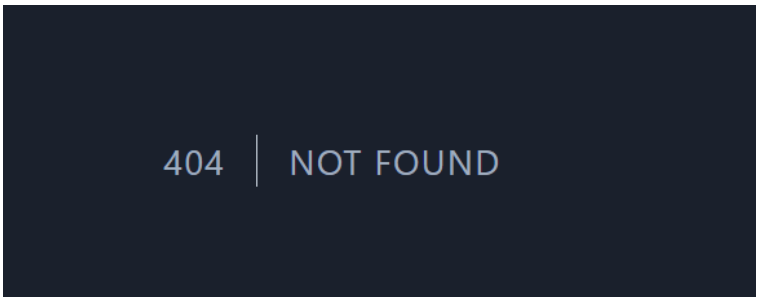
ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

```
$user = UserModel::findOr(1,['username','nama'], function(){
    abort(404);
});
return view('user', ['data' => $user]);
```

Data User

ID	Username	Nama	ID Level Pengguna
	admin	Administrator	

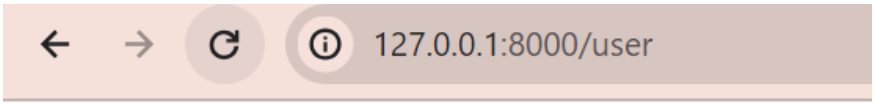
```php
$user = UserModel::findOr(20,['username','nama'], function(){
    abort(404);
});
return view('user', ['data' => $user]);
```



Praktikum 2.2

Not found exceptions

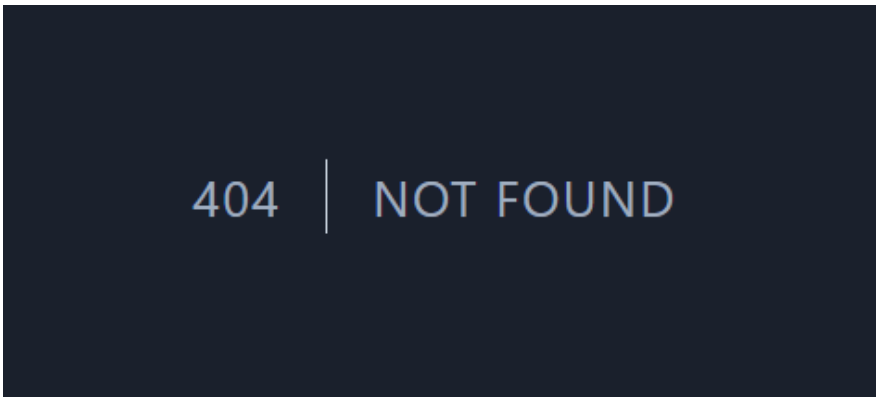
```php
$user = UserModel::findOrFail(1);
return view('user', ['data' => $user]);
```



Data User

ID	Username	Nama	ID Level Pengguna
1	admin	Administrator	1

```php
$user = UserModel::where('username','manager9')->firstOrFail();
return view('user', ['data' => $user]);
```



Praktikum 2.3

Retrieving Aggregates

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

Data User

Jumlah Pengguna
2

Praktikum 2.4

Retreiving or Creating Models

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

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

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

Data User

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

Pertama akan mencari data sesuai dengan data yang ada pada array user, jika data tersebut tidak ada, maka data akan ditambahkan kemudian baru ditampilkan

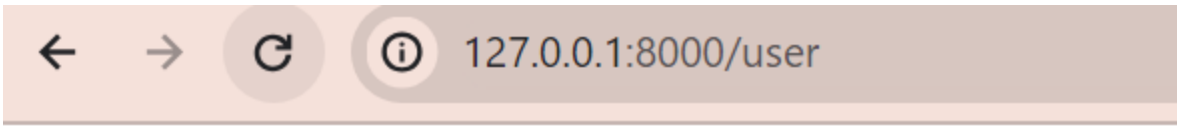
```
$user = UserModel::firstOrCreate(
    [
        'username' => 'manager',
        'nama' => 'Manager',
    //      'password' => Hash::make('12345'),
    //      'level_id' => 2
        ]
);
return view('user', ['data' => $user]);
```

Data User

ID	Username	Nama	ID Level Pengguna
2	manager	Manager	2

Pertama akan mencari data berdasarkan data yang ada pada array user, karena data ditemukan maka akan langsung ditampilkan.

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

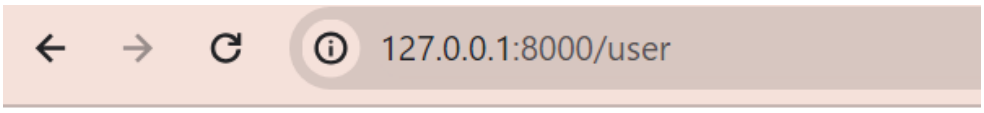


Data User

ID	Username	Nama	ID Level Pengguna
	manager33	Manager Tiga Tiga	2

Ketika menggunakan FirstOrNew maka data hanya ditampilkan dan tidak langsung masuk ke database

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



Data User

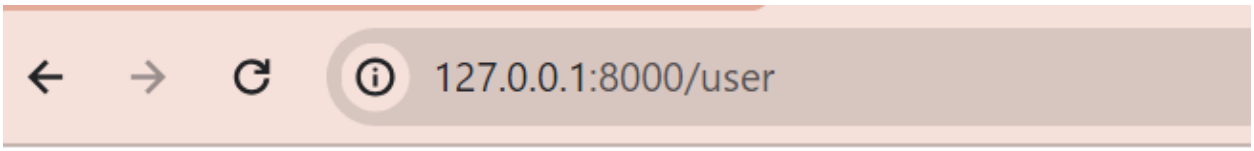
ID	Username	Nama	ID Level Pengguna
10	manager33	Manager Tiga Tiga	2

Jika menggunakan save maka pertama data akan dicari terlebih dahulu, jika data tidak ditemukan maka akan menambahkan data baru pada database.

Praktikum 2.5

Attribute Changes

```php
$user = UserModel::create([
        'username' => 'manager44',
        'nama' => 'Manager44',
        'password' => Hash::make('12345'),
        'level_id' => 2
]);

$user->username = 'manager45';

$user->isDirty(); //t
$user->isDirty('username'); //t
$user->isDirty('nama'); //f
$user->isDirty(['nama','username']); //t

$user->isClean(); //f
$user->isClean('username'); //f
$user->isClean('nama'); //t
$user->isClean(['nama', 'username']); //f

$user->save();
$user->isDirty(); //f
$user->isClean(); //t

dd($user->isDirty());
```




```
false // app\Http\Controllers\UserController.php:84
```

Pertama user bernilai true ketika di cek menggunakan isDirty, karena data belum disimpan pada database, setelah disimpan pada database user akan bernilai false ketika di cek menggunakan isDirty karena memang tidak ada Perubahan data atribut.

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

$user->username='manager12';
$user->save();

$user->wasChanged(); //t
$user->wasChanged('username'); //t
$user->wasChanged(['username','level_id']); //t
$user->wasChanged('nama'); //f
dd($user->wasChanged(['nama','username']));
```


```
true // app\Http\Controllers\UserController.php:101
```

Terdapat perubahan username yaitu dari manger11 menjadi manager12 , jadi ketika di cek menggunakan wasChanged akan bernilai true, dan ketika mengecek lebih dari 1 atribut dimana salah satu atribut terdapat Perubahan maka akan bernilai true, sama halnya seperti konsep and.

Praktikum 2.6

CRUD

```
<body>
    <h1>Data User</h1>
    <a href="/user/tambah"> + Tambah User</a>      You, 1 second ago • Uncommitted changes
    <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>
```

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

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
4	manager_2	Manager 2	2	Ubah | Hapus
6	manager22	Manager Dua Dua	2	Ubah | Hapus
10	manager33	Manager Tiga Tiga	2	Ubah | Hapus
14	manager45	Manager44	2	Ubah | Hapus
15	manager12	Manager11	2	Ubah | Hapus
16	manager11	Manager11	2	Ubah | Hapus

```blade
<body>
    <h1>Form Tambah Data User</h1>
    <form method="post" action="/user/tambah_simpan">
        {{csrf_field()}}
        <label for="Username"></label>
        <input type="text" name="username" placeholder="Masukkan Username">
        <br><br>
        <label for="Nama"></label>
        <input type="text" name="nama" placeholder="Masukkan Nama">
        <br><br>
        <label for="Password"></label>
        <input type="text" name="password" placeholder="Masukkan Password">
        <br><br>
        <label for="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>
```

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

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

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

Form Tambah Data User

manager1

Putri Norchasama

12345

2

Simpan

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
4	manager_2	Manager 2	2	Ubah | Hapus
6	manager22	Manager Dua Dua	2	Ubah | Hapus
10	manager33	Manager Tiga Tiga	2	Ubah | Hapus
14	manager45	Manager44	2	Ubah | Hapus
15	manager12	Manager11	2	Ubah | Hapus
16	manager11	Manager11	2	Ubah | Hapus
18	manager1	Putri Norchasama	1	Ubah | Hapus

Update data

```
<body>
    <h1>Form Ubah Data User</h1>
    <form method="post" action="/user/ubah_simpan">
        {{csrf_field()}}
        {{method_field('PUT')}}

        <label for="Username"></label>
        <input type="text" name="username" placeholder="Masukkan Username" value="{{ $data->username }}">
        <br><br>
        <label for="Nama"></label>
        <input type="text" name="nama" placeholder="Masukkan Nama" value="{{ $data->nama }}">
        <br><br>
        <label for="Password"></label>
        <input type="text" name="password" placeholder="Masukkan Password" value="{{ $data->password }}">
        <br><br>
        <label for="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>
```

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

404 | NOT FOUND

Ketika di klik ubah masih terdapat error 404 karena belum menambahkan function untuk mengubah data dan routing yang mengaturnya

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

```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');

}
```

Form Ubah Data User

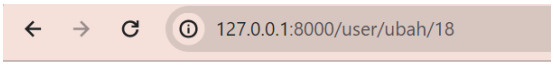
manager1

Putri Norchasama

$2y$12$QvsYvRJ8xo35Bjc

1

Ubah



Form Ubah Data User

manager1

Veradianti

$2y$12$/oGSKES1y7P6TP

1

Ubah

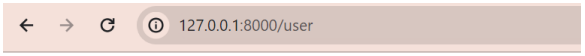
Data User

+ Tambah User

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	Ubah | Hapus
2	manager	Manager	2	Ubah | Hapus
3	staff	Staff/Kasir	3	Ubah | Hapus
4	manager_2	Manager 2	2	Ubah | Hapus
6	manager22	Manager Dua Dua	2	Ubah | Hapus
10	manager33	Manager Tiga Tiga	2	Ubah | Hapus
14	manager45	Manager44	2	Ubah | Hapus
15	manager12	Manager11	2	Ubah | Hapus
16	manager11	Manager11	2	Ubah | Hapus
18	manager1	Veradianti	1	Ubah | Hapus

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

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

    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
4	manager_2	Manager 2	2	Ubah | Hapus
6	manager22	Manager Dua Dua	2	Ubah | Hapus
10	manager33	Manager Tiga Tiga	2	Ubah | Hapus
14	manager45	Manager44	2	Ubah | Hapus
15	manager12	Manager11	2	Ubah | Hapus
16	manager11	Manager11	2	Ubah | Hapus

Menghapus data id 18

Praktikum 2.7

Relationship

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

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

Data User

+ Tambah User

ID	Username	Nama	ID Level Pengguna	Aksi
1	admin	Administrator	1	Ubah | Hapus
2	manager	Manager	2	Ubah | Hapus
3	staff	Staff/Kasir	3	Ubah | Hapus
4	manager_2	Manager 2	2	Ubah | Hapus
6	manager22	Manager Dua Dua	2	Ubah | Hapus
10	manager33	Manager Tiga Tiga	2	Ubah | Hapus
14	manager45	Manager44	2	Ubah | Hapus
15	manager12	Manager11	2	Ubah | Hapus
16	manager11	Manager11	2	Ubah | Hapus