

Modul : Model di Laravel dengan Eloquent

Laravel dilengkapi dengan sebuah ORM, yaitu Eloquent. ORM adalah kependekan dari Object Oriented Mapping. Penjelasan mudah dari ORM adalah sebuah **teknik** untuk memanipulasi (CRUD) Database secara Object Oriented, sehingga tidak diperlukan lagi bahasa SQL. ORM diimplementasikan dalam bentuk library dan ORM pada Laravel adalah Eloquent. Untuk mengetahui lebih lanjut terkait Eloquent ORM dapat dibaca pada dokumentasi berikut <https://laravel.com/docs/6.x/eloquent>.

Perhatikan aturan main intinya saja:

1. Aturan dasar, Nama Class diawali dengan huruf besar
2. Satu model menghandel satu Tabel
3. Nama model sama dengan nama Tabel, tetapi pada poin ini nanti perlu ada sedikit penyesuaian karena masalah di grammar bahasa inggris

Percobaan:

1. Pastikan konfigurasi Database pada file .env telah anda persiapkan (silahkan buka Modul Query Builder)
2. Kita akan membuat model untuk tabel "user" pada disain Database. Karena didalam directory [nama project]/app sudah ada sebuah file php bernama User.php yang digunakan untuk fungsi lainnya (Authentication) pada Laravel, maka kita akan menggunakan nama lain untuk menghandel tabel user.

user			
user_id	<pi>	Integer	<M>
first_name		Variable characters (50)	
last_name		Variable characters (50)	
email		Variable characters (50)	
phone		Number (12)	
password		Characters (8)	
job_status		Variable characters (15)	
Identifier_1	<pi>		

Ubah panjang data untuk password dari Char(8) menjadi Char(32) / Varchar(32). Password akan disimpan dalam bentuk nilai hash dari MD5 yang berjumlah 32 character. Tambahkan juga atribut *username* dengan tipe data *varchar(50)*.

3. Buat model untuk tabel user dengan nama Pegawai menggunakan artisan
`php artisan make:model Pegawai`
4. Perhatikan aturan main pada model ini:
 - a. Setiap tabel akan berisi banyak isi, contoh tabel pegawai akan berisi banyak informasi pegawai. Sehingga tabel untuk pegawai dalam bahasa inggris akan diberi nama (pada umumnya) pegawai*s*. Ingat, bentuk jamak dalam bahasa inggris selalu ditambah s dibelakangnya.
 - b. Sehingga **terpaksa** kita harus mendefinisikan nama tabel dalam model. Pada kasus kita saat ini, nama tabel berbeda dengan nama Modelnya, yaitu tabel **user**.

Pegawai.php

```
namespace App;

use Illuminate\Database\Eloquent\Model;

class Pegawai extends Model
{
    //mendefinisikan nama table
    protected $table = "user";
}
```

5. Target: Menampilkan data seluruh pegawai ketika membuka halaman data master untuk pegawai. Dari modul-modul sebelumnya, anda sudah memahami bahwa kita perlu melakukan langkah-langkah berikut dengan asumsi Model telah dibuat:
 - a. Buat Route
 - b. Buat Controllernya. Jika Controller sudah ada, tambahkan / sesuaikan fungsinya.
 - c. Buat halaman Viewnya.
6. Route ada pada [nama proyek]/routes/web.php

```
Route::get('/pegawai', 'C_pegawai@index');
```
7. Buat Controller dengan artisan

```
php artisan make:controller C_pegawai
```
8. Panggil model "Pegawai" pada C_pegawai dan siapkan data untuk view

C_pegawai

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Pegawai; //panggil Model Pegawai

class C_pegawai extends Controller
{
    function index()
    {
        //memanggil fungsi dari Eloquent untuk get all data
        $data = Pegawai::all();

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

9. Buat view [nama project]/resources/views/pegawai_page.blade.php

```
<!DOCTYPE html>
<head>
    <title>Halaman Pegawai</title>
</head>
<body>
    <h1>Data Pegawai</h1>
    <br>
    <a href="/pegawai/tambahPegawai"><b>Tambah Data Pegawai</b></a>
    <br>
    <br>
    <table border="1" width="100%">
    <tr>
        <th>Id Pegawai</th>
        <th>Nama Depan</th>
        <th>Nama Belakang</th>
        <th>Username</th>
        <th>Phone</th>
        <th>Email</th>
        <th>Job Status</th>
        <th>action</th>
    </tr>
    @foreach($pegawai as $data)
        <tr>
            <td>{{ $data->user_id }}</td>
            <td>{{ $data->first_name }}</td>
            <td>{{ $data->last_name }}</td>
            <td>{{ $data->username }}</td>
            <td>{{ $data->phone }}</td>
            <td>{{ $data->email }}</td>
            <td>{{ $data->job_status }}</td>
            <td><a href="/pegawai/editPegawai/{{ $data->user_id }}">
                <button> edit </button>
            </a>
            </td>
        </tr>
    @endforeach
    </table>
</body>
</html>
```

10. Buka halaman pegawai dengan url: <http://localhost:8000/pegawai>

Tambah Pegawai

1. Pada halaman ini sebuah informasi baru akan ditambahkan pada tabel user, sehingga perlu didefinisikan nama-nama atribut pada tabel user yang **dapat diisi (Fillable)**. Tambahkan informasi berikut pada model Pegawai.

```
class Pegawai extends Model
{
    //mendefinisikan nama table
    protected $table = "user";

    protected $fillable = ['first_name',
                           'last_name',
                           'username',
                           'email',
                           'phone',
                           'password',
                           'job_status'
                           ];

    public $timestamps = false;
}
```

Secara default, eloquent akan meng-*insert* timestamp dengan atribut bernama 'created_at' dan 'updated_at'. Karena tabel user tidak memiliki kedua atribut tersebut, maka anda harus menonaktifkan timestamp pada model ini.

```
public $timestamps = false;
```

2. Buat route nya

```
Route::get('/pegawai/tambahPegawai', 'C_pegawai@tambahPegawai');
```

3. Buat function tambahPegawai pada C_pegawai

```
function tambahPegawai()
{
    return view('tambahPegawai');
}
```

4. Buat view tambahPegawai.blade.php

```
<!DOCTYPE html>
<head>
    <title>Tambah Data Pegawai</title>
</head>
<body>
    <h1>Tambah Data Pegawai</h1>
    <form action="/pegawai/insertData" method="post">
        <input type = "hidden" name = "_token" value = "<?php echo csrf_token() ?>">
```

```

        Nama Depan : <input type="text" name="f_name"><br>
        Nama Belakang : <input type="text" name="l_name"><br>
        Username : <input type="text" name="username"><br>
        Telp : <input type="text" name="phone"><br>
        email : <input type="email" name="email"><br>
        job status : <input type="text" name="job_status"><br><br>
        Password : <input type="password" name="password" id="pass"><br>
        Confirm Password : <input type="password" name="password" id="copass"
            onkeyup="cekPass()">
        <p id="error" style="color:red"></p><br>
        <input type="submit" value="Simpan">
    </form>

</body>
<script>
    function cekPass()
    {
        var pass = document.getElementById('pass').value;
        var copass = document.getElementById('copass').value;
        var text = document.getElementById('error');
        if(pass != copass)
        {
            text.style.color = 'red';
            text.innerHTML='Password tidak cocok';
        }
        else
        {
            text.style.color = 'green';
            text.innerHTML='OK!!';
        }
    }
</script>

</html>

```

5. Siapkan route untuk POST-nya

```
Route::post('/pegawai/insertData', 'C_pegawai@insertData');
```

6. Buat function insertData pada C_pegawai. Terdapat dua cara untuk melakukannya, yang pertama anda akses langsung Classnya.

```

function insertData(Request $req)
{
    Pegawai::create([
        'first_name' => $req->f_name,
        'last_name' => $req->l_name,

```

```

        'username' => $req->username,
        'email' => $req->email,
        'phone' => $req->phone,
        'job_status' => $req->job_status
    ]);

    return redirect('/pegawai');
}

```

Kedua, anda create sebuah object dari model tersebut:

```

function insertData(Request $req)
{
    $password = md5($req->pass);

    $pegawai = new Pegawai;
    $pegawai->fill([
        'first_name' => $req->f_name,
        'last_name' => $req->l_name,
        'username' => $req->username,
        'email' => $req->email,
        'phone' => $req->phone,
        'password' => $password,
        'job_status' => $req->job_status
    ]);

    $pegawai->save();
    return redirect('/pegawai');
}

```

7. Silahkan anda coba halaman anda! <http://localhost:8000/pegawai/tambahPegawai>

UPDATE

1. Perhatikan view anda untuk menampilkan data pegawai. Terdapat baris seperti berikut

```

<td><a href='/pegawai/editPegawai/{{ $data->user_id }}'>
    <button> edit </button>
</a>
</td>

```

href pada tab <a> diatas telah menunjukan route yang perlu anda buat.

2. Buat route

```

Route::get('/pegawai/editPegawai/{idPegawai}','C_pegawai@editPegawai');

```

3. Buat fungsi editPegawai() pada C_pegawai

```
function editPegawai($idPegawai)
{
    $data = Pegawai::where('user_id', $idPegawai)->get();
    return view('editPegawai', ['pegawai' => $data]);
}
```

4. Buat halaman untuk edit.

```
<!DOCTYPE html>
<head>
    <title>Update Data Pegawai</title>
</head>
<body>
    <h1>Update Data Pegawai</h1>

    <form action="/pegawai/updatePegawai" method="post">
        <input type="hidden" name="_token" value="php echo csrf_token(
) ?&gt;"&gt;
        id Pegawai : &lt;input type="text" name="pegawai_id"
            value="{{ $pegawai[0]-&gt;user_id }}" readonly&gt;&lt;br&gt;
        Nama Depan : &lt;input type="text" name="f_name"
            value="{{ $pegawai[0]-&gt;first_name }}"&gt;&lt;br&gt;
        Nama Belakang : &lt;input type="text" name="l_name"
            value="{{ $pegawai[0]-&gt;last_name }}"&gt;&lt;br&gt;
        Username : &lt;input type="text" name="username"
            value="{{ $pegawai[0]-&gt;username }}"&gt;&lt;br&gt;
        Telp : &lt;input type="text" name="phone" value="{{ $pegawai[0]-
&gt;phone }}"&gt;&lt;br&gt;
        email : &lt;input type="email" name="email" value="{{ $pegawai[0]-
&gt;email }}"&gt;&lt;br&gt;
        job status : &lt;input type="text" name="job_status"
            value="{{ $pegawai[0]-&gt;job_status }}"&gt;&lt;br&gt;
        &lt;input type="submit" value="Update"&gt;
    &lt;/form&gt;

&lt;/body&gt;
&lt;/html&gt;</pre
```

5. Siapkan route untuk POST

```
Route::post('/pegawai/updatePegawai', 'C_pegawai@updatePegawai');
```

6. Buat function ud patePegawai() pada C_pegawai

```
function updatePegawai(Request $req)
{
    $pegawai = Pegawai::where('user_id', $req->pegawai_id)
```

```
->update([
    'first_name' => $req->f_name,
    'last_name' => $req->l_name,
    'username' => $req->username,
    'phone' => $req->phone,
    'email' => $req->email,
    'job_status' => $req->job_status
]);

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

7. Silahkan anda akses url <http://localhost:8000/pegawai> dan click edit pada salah satu baris. Ubah datanya dan simpan. Jika data berhasil terupdate, maka percobaan anda berhasil.

Tugas:

Implementasikan modul pratikum ini pada **dua** halaman Data master anda!