

FRAMEWORK PHP: LARAVEL
MULTIPLE AUTHENTICATION #1
KL PEMROGRAMAN WEB LANJUT

SETTING DATABASE

1. Buat database di phpmyadmin, misal: laravel.
2. Buka file .env untuk setting database.
3. Sesuaikan nama database, username dan password. Misal:

```
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

4. Simpan file .env.

MULTIPLE AUTHENTICATION

1. Laravel sudah memiliki default authentication. Namun hanya satu role yaitu user. Jika ingin membuat lebih dari satu role, maka perlu memodifikasinya. Dalam praktikum ini akan dibuat dua role yaitu user dan admin. Untuk user akan menggunakan default authentication dari laravel.
2. **Buka cmd**, arahkan ke folder project laravel Anda.
3. Ketikkan ***php artisan make:auth***
4. Jika sudah berhasil maka akan muncul file baru di controller, view dan perubahan di routes.
5. Membuat tabel dapat dilakukan secara manual di phpmyadmin atau menggunakan migration.
6. Dalam praktikum ini, akan menggunakan migration tabel bawaan laravel untuk authentication yaitu users dan passwords. Untuk admin, akan dibuat dalam sebuah tabel yang akan didefinisikan sendiri.
7. Buat migration untuk admin dengan mengetikkan ***php artisan make:migration create_admins_table --create=admins***.
8. **Buka folder database/migrations**, maka akan muncul file baru dengan nama ***create_admins_table***. Buka file tersebut.
9. **Tambahkan semua informasi** yang dibutuhkan untuk tabel admin. Misal:

```
public function up()
{
    Schema::create('admins', function (Blueprint $table) {
        $table->increments('id');
        $table->string('name');
        $table->string('email')->unique();
        $table->string('job_title');
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}
```

10. Jalankan semua migration dengan mengetikkan ***php artisan migrate***.
11. Jika muncul error seperti ini:

```
[Illuminate\Database\QueryException]
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too
long; max key length is 767 bytes (SQL: alter table `users` add unique `
users_email_unique`(`email`))
```

```
[PDOException]
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was t
oo long; max key length is 767 bytes
```

12. Buka file **AppServiceProvider.php** di folder **app/Providers**. Tambahkan syntak berikut untuk mengatur panjang default string.

```
use Illuminate\Support\Facades\Schema;

public function boot()
{
    Schema::defaultStringLength(191);
}
```

13. Ketikkan ***php artisan migrate*** lagi. (Catatan: jika masih gagal, coba hapus tabel users di phpmyadmin kemudian ketikkan ***php artisan migrate*** lagi.)
14. Cek ke database, maka akan muncul tabel admins, migrations, password_resets dan users.
15. **Penamaan base database** sebaiknya **jamak**, seperti: admins, users.
16. **Buat model untuk Admin** di folder app. Karena isinya mirip dengan model untuk User, Anda dapat **duplicate User model**. Beri nama file **Admin.php**. Sesuaikan isinya dan simpan.

```

<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;

class Admin extends Authenticatable
{
    use Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password', 'job_title',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];
}

```

17. Atur konfigurasi untuk auth. Buka file **auth.php di folder config**. Di bagian providers, guards dan passwords, tambahkan untuk admin.

```

providers' => [
    'users' => [
        'driver' => 'eloquent',
        'model' => App\User::class,
    ],

    'admins' => [
        'driver' => 'eloquent',
        'model' => App\Admin::class,
    ],

```

```
guards' => [
    'web' => [
        'driver' => 'session',
        'provider' => 'users',
    ],

    'api' => [
        'driver' => 'token',
        'provider' => 'users',
    ],

    'admin' => [
        'driver' => 'session',
        'provider' => 'admins',
    ],

    'admin-api' => [
        'driver' => 'token',
        'provider' => 'admins',
    ],
],
```

```
'passwords' => [
    'users' => [
        'provider' => 'users',
        'table' => 'password_resets',
        'expire' => 60,
    ],

    'admins' => [
        'provider' => 'admins',
        'table' => 'password_resets',
        'expire' => 15,
    ],
],
```

18. **Definisikan variable guard** di Admin model.

```
protected $guard = 'admin';
```

19. **Tambahkan routing untuk admin** di file web.php folder routes.

```
Route::get('/admin', 'AdminController@index')->name('admin.dashboard');
```

20. **Buat AdminController.php** di folder app/Http/Controllers. Karena isinya mirip dengan HomeController.php, Anda dapat **duplicate**. Sesuaikan isinya dan simpan.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AdminController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        return view('admin');
    }
}

```

21. **Buat admin.blade.php** di folder resources/views. Karena isinya mirip dengan home.blade.php, Anda dapat **duplicate**. Sesuaikan isinya dan simpan.

```

@extends('layouts.app')

@section('content')
<div class="container">
    <div class="row">
        <div class="col-md-8 col-md-offset-2">
            <div class="panel panel-default">
                <div class="panel-heading">ADMIN Dashboard</div>

                <div class="panel-body">
                    You are logged in as <strong>ADMIN</strong>!
                </div>
            </div>
        </div>
    </div>
</div>
@endsection

```

22. **Edit** juga untuk **home.blade.php** supaya terlihat bedanya.

```

@extends('layouts.app')

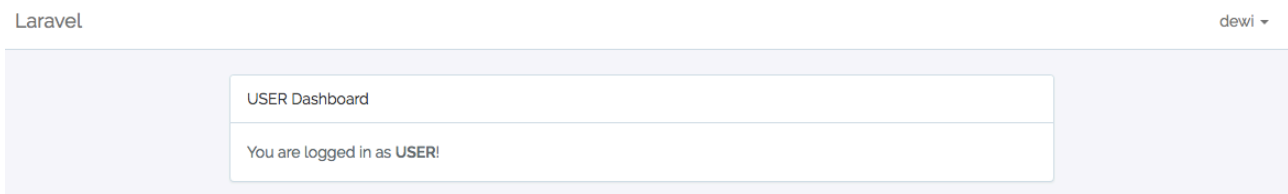
@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-8 col-md-offset-2">
      <div class="panel panel-default">
        <div class="panel-heading">USER Dashboard</div>

        <div class="panel-body">
          You are logged in as <strong>USER</strong>!
        </div>
      </div>
    </div>
  </div>
</div>
@endsection

```

23. Coba jalankan dengan mengetikkan ***php artisan serve***.

24. Coba **register**. Berhasil maka akan menuju ke home seperti berikut.



25. Ubah url <http://127.0.0.1:8000/home> menjadi <http://127.0.0.1:8000/admin>.

26. Hasilnya berhasil menuju ke halaman admin.



27. Namun hal ini tidak sesuai ekpektasi. Seharusnya tidak bisa menuju ke halaman admin.

28. Buka AdminController, edit di bagian `__construct` dengan menambahkan guard admin pada middleware.

```
$this->middleware('auth:admin');
```

29. Refresh dan coba lakukan langkah 25 lagi.

LOGIN IN MULTIPLE USERS

1. Skenario dalam praktikum ini, admin tidak dapat register, sehingga admin ditambahkan melalui manajemen admin.
2. Cek daftar routes dengan mengetikkan ***php artisan route:list***.

3. Buat AdminLoginController di folder `app/Http/Controllers/Auth` dengan mengetikkan **php artisan make:controller Auth/AdminLoginController**.
4. Buat function **__construct**, **showLoginForm** dan **login**.

```
<?php

namespace App\Http\Controllers\Auth;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Auth;

class AdminLoginController extends Controller
{
    public function __construct()
    {
        $this->middleware('guest:admin');
    }

    function showLoginForm()
    {
        return view('auth.admin-login');
    }

    function login(Request $request)
    {
        // Validate the form data
        $this->validate($request, [
            'email' => 'required|email',
            'password' => 'required|min:6',
        ]);

        // Attempt to log the user in
        if (Auth::guard('admin')->attempt(['email' => $request->email, 'password' => $request->password],
            $request->remember)) {
            // If successful, then redirect to their intended location
            return redirect()->intended(route('admin.dashboard'));
        }

        // If unsuccessful, then redirect back to the login with the form data
        redirect()->back()->withInput($request->only('email', 'remember'));
    }
}
```

5. Buat view **admin-login.blade.php** di folder **resources/views/auth**. Duplicate dari file `login.blade.php` saja karena isinya mirip. Sesuaikan isinya dan simpan.

```
<div class="panel-heading">ADMIN Login</div>

<div class="panel-body">
    <form class="form-horizontal" method="POST" action="{{ route('admin.login.submit') }}">
        {{ csrf_field() }}
    </form>
</div>
```

6. Tambahkan **routing untuk login admin** di `web.php` folder `routes`.

```
Route::get('/admin/login', 'Auth\AdminLoginController@showLoginForm')->name('admin.login');
Route::post('/admin/login', 'Auth\AdminLoginController@login')->name('admin.login.submit');
```

7. Karena ada kesamaan routingnya /admin maka bisa **dikelompokkan** menjadi seperti ini.

```
Route::prefix('admin')->group(function() {  
    Route::get('/login', 'Auth\AdminLoginController@showLoginForm')->name('admin.login');  
    Route::post('/login', 'Auth\AdminLoginController@login')->name('admin.login.submit');  
    Route::get('/', 'AdminController@index')->name('admin.dashboard');  
});
```

8. Coba jalankan <http://127.0.0.1:8000/admin/login>

9. Buat dummy data untuk admin dengan tinker. Ketikkan ***php artisan tinker*** dan ketikkan seperti berikut ini.

```
>>> $admin = new App\Admin  
>>> $admin->name = "Anisa"  
>>> $admin->email = "anisa@tes.com"  
>>> $admin->password = Hash::make('12345678')  
>>> $admin->job_title = "Staf"  
>>> $admin->save()
```

10. Coba login sebagai user <http://127.0.0.1:8000/login>. Kemudian coba login sebagai admin <http://127.0.0.1:8000/admin/login>.