

# Pertemuan 3

Upload Image

# Sebelum itu ada Revisi nih

- Middleware
- Penambahan redirect saat register + flash message
  - Link referensi
    - <https://laravel.com/docs/10.x/session#flash-data>
    - <https://laravel.com/docs/10.x/responses#redirecting-with-flashed-session-data>
    - <https://getbootstrap.com/docs/5.3/components/alerts/#examples>
- Perubahan Mekanisme login
  - Link referensi
    - <https://laravel.com/docs/10.x/authentication#authenticating-users>
- Penambahan navbar agar lebih memudahkan navigasi
  - Link referensi
    - <https://getbootstrap.com/docs/5.3/components/navbar/#how-it-works>

# Apa itu Middleware?

*“Middleware provide a convenient mechanism for inspecting and filtering HTTP requests entering your application. For example, Laravel includes a middleware that verifies the user of your application is authenticated. If the user is not authenticated, the middleware will redirect the user to your application's login screen. However, if the user is authenticated, the middleware will allow the request to proceed further into the application.*

*Additional middleware can be written to perform a variety of tasks besides authentication. For example, a logging middleware might log all incoming requests to your application. There are several middleware included in the Laravel framework, including middleware for authentication and CSRF protection. All of these middleware are located in the `app/Http/Middleware` directory.”*

# Redirect Flash saat Register

```
// fungsi kedua untuk menyimpan data ke database
public function register(Request $request)
{
    $user = new User;
    $user->name = $request->name;
    $user->email = $request->email;
    $user->password = bcrypt($request->password);
    $user->save();

    return redirect('/login')->with('success', 'Berhasil mendaftar, silahkan login');
}
```

# Perubahan Login Sistem

Sebelum

```
public function login(Request $request)
{
    // dd($request);
    // login
    $email = $request->email;
    $password = $request->password;

    // check email dan password
    $user = User::where('email', $email)->first(); //f
    $hashedPassword = $user->password;

    if (Hash::check($password, $hashedPassword)) {
        return "Berhasil Login";
    } else {
        return "Gagal Login";
    }

    return $user;
}
```

Sesudah

```
}

public function login(Request $request)
{
    $credentials = $request->validate([
        'email' => ['required', 'email'],
        'password' => ['required'],
    ]);

    if (Auth::attempt($credentials)) {
        $request->session()->regenerate();

        return redirect()->intended('/'); You, 8 h
    }

    return back()->withErrors([
        'email' => 'The provided credentials do not mat
    ])->onlyInput('email');
}
```

# Upload Avatar Image

Siapkan:

- Migration pada table users dengan kolom string avatar
- Controller ProfileController.php
- Route

```
// upload profile avatar
Route::get('/profile/upload', [App\Http\Controllers\ProfileController::class, 'uploadView'])->middleware('auth');
Route::post('/profile/upload', [App\Http\Controllers\ProfileController::class, 'upload'])->middleware('auth');
```

# Syntax untuk membuat migration

```
php artisan make:migration nama_migrationnya_bebas --table=users
```

# Migration untuk membuat Avatar

```
/**
 * Run the migrations.
 */
public function up(): void
{
    Schema::table('users', function (Blueprint $table) {
        $table->string('avatar')->nullable();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::table('users', function (Blueprint $table) {
        $table->dropColumn('avatar');
    });
}
;
```



# Membuat ProfileController

- php artisan make:controller ProfileController

```
class ProfileController extends Controller
{
    public function uploadView()
    {
        return view('profile.upload');
    }

    public function upload(Request $request)
    {
        $this->validate($request, [
            'image' => 'image|file|max:2048',
        ]);

        // simpan gambar pada user yang sedang login
        $user = $request->user();
        $user->avatar = $request->file('image')->store('avatar');
        $user->save();

        return redirect('/profile/upload')->with('success', 'Berhasil mengupload avatar');
    }
}
```

# Membuat View profile/upload.blade.php

```
@extends('layouts.main')
@section('body')
    <div class="d-flex justify-content-center">
        <div class="card mt-5">
            <div class="card-body">
                <h5 class="card-title">Upload</h5>
                <h6 class="card-subtitle mb-2 text-body-secondary">Masukan gambar</h6>
                <div class="row">
                    <form method="POST" action="/profile/upload" enctype="multipart/form-data">
                        @csrf
                        <div class="mb-3">
                            <label for="image" class="form-label">image</label>
                            <input type="file" class="form-control" id="image" name="image">
                        </div>
                        <input type="submit" class="btn btn-primary" id="submit" name="submit" value="Upload">
                    </form>
                </div>
            </div>
        </div>
    </div>
@endsection
```

# Role

Role (hak / wewenang / peran) adalah penamaan hak akses yang ada dalam sistem. Misalnya CEO- Manager- HRD- Karyawan.

# Membuat Kolom Role pada Migration

```
php artisan make:migration alter_table_users_add_role --table=users
```

```
public function up(): void
{
    Schema::table('users', function (Blueprint $table) {
        $table->string('role')->nullable();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::table('users', function (Blueprint $table) {
        $table->dropColumn('role');
    });
}
```

```
php artisan migrate
```

# Tambahkan role “member” saat register

```
// fungsi kedua untuk menyimpan data ke database
public function register(Request $request)
{
    $user = new User;
    $user->name = $request->name;
    $user->email = $request->email;
    $user->password = bcrypt($request->password);
    $user->role = 'member';
    $user->save();

    return redirect('/login')->with('success', 'Berhasil mendaftar, silahkan login');
}
```

# Tambahkan nama role di NavBar

Navbar Home Register

Hi, member Sinta ▾

```
@auth
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown"
      aria-expanded="false">
      Hi, {{ auth()->user()->role }} {{ auth()->user()->name }} You, 1 second ago • Unc
    </a>
    <ul class="dropdown-menu">
      <li><a class="dropdown-item" href="/profile/upload">Upload Avatar</a></li>
      <li><a class="dropdown-item" href="/logout">Logout</a></li>
    </ul>
  </li>
@else
  <li class="nav-item">
```

Sekian