

PEMROGRAMAN WEB LANJUT
JOBSHEET 9 - AUTHENTICATION, MIDDLEWARE



Disusun oleh:
Stefanus Ageng Budi Utomo (2241720126)

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN AJARAN 2023/2024

Praktikum - Auth dan middleware

1. Sebelum melangkah lebih lanjut setting file pada config/auth.php ubah pada bagian model menjadi model yang telah kita buat sebelumnya untuk menyimpan username dan password

```
'providers' => [
    'users' => [
        'driver' => 'eloquent',
        'model' => App\Models\UserModel::class,
    ],

    // 'users' => [
    //     'driver' => 'database',
    //     'table' => 'users',
    // ],
],
```

2. Membuat middleware dengan perintah pada cmd atau terminal

```
PS D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS> php artisan make:middleware CekLogin
```

```
INFO Middleware [D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS\app\Http\Middleware\CekLogin.php] created successfully.
```

Setelah itu cek didalam folder app/Http/Middleware apakah sudah ada file bernama Cek_login.php

```

1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Symfony\Component\HttpFoundation\Response;
8
9  class Cek_Login
10 {
11     /**
12      * Handle an incoming request.
13      *
14      * @param  \Closure(\Illuminate\Http\Request):
15      * (\Symfony\Component\HttpFoundation\Response) $next
16      */
17     public function handle(Request $request, Closure $next): Response
18     {
19         return $next($request);
20     }
21 }

```

3. Kemudian tambahkan kode dibawah ini, letakkan di function handle()

```

auth.php M Cek_Login.php U X
app > Http > Middleware > Cek_Login.php > Cek_Login > handle
1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Auth;
8  use Symfony\Component\HttpFoundation\Response;
9
10 class Cek_Login
11 {
12     /**
13      * Handle an incoming request.
14      *
15      * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
16      */
17     public function handle(Request $request, Closure $next, $roles): Response
18     {
19         // return $next($request);
20         // cek apakah sudah login atau belum. jika belum kembali ke halaman login
21         if (!Auth::check()) {
22             return redirect('login');
23         }
24         // simpan data user pada variable $user
25         $user = Auth::user();
26         // jika user memiliki level sesuai pada kolom, lanjutkan request
27         if ($user->level_id == $roles) {
28             return $next($request);
29         }
30         // jika tidak memiliki akses maka kembalikan ke halaman login
31         return redirect('login')->with('error', 'Maaf anda tidak memiliki akses');
32     }
33 }

```

4. Ketika sudah membuat middleware, langkah berikutnya harus registrasikan pada kernel.php. Agar middleware dapat dibaca oleh sistem. Buka folder App/Http/Kernel.php dan tambahkan code pada variabel \$middlewareAliases seperti dibawah

```

app > Http > Kernel.php > Kernel > $middlewareAliases
7  class Kernel extends HttpKernel
8  {
9      /**
10       * The application's global HTTP middleware stack.
11       *
12       * @var array
13       */
14     protected $middleware = [
15         // ...
16     ];
17
18     /**
19      * The application's route level HTTP middleware stack.
20      *
21      * @var array
22      */
23     protected $middlewareAliases = [
24         'auth' => \App\Http\Middleware\Authenticate::class,
25         'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
26         'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
27         'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
28         'can' => \Illuminate\Auth\Middleware\Authorize::class,
29         'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
30         'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
31         'precognitive' => \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class,
32         'signed' => \App\Http\Middleware\ValidateSignature::class,
33         'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
34         'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
35         // menambah middleware alias dengan nama cek_login
36         'cek_login' => \App\Http\Middleware\Cek_Login::class,
37     ];
38 }

```

5. Membuat Controller Auth. Pada langkah ini kita akan membuat controller auth, yang dimana di dalamnya terdapat proses autentikasi dari request login. Jalankan perintah make:controller

```

PS D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS> php artisan make:controller AuthController
INFO Controller [D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS\app\Http\Controllers\AuthController.php] created successfully.

```

Buka file App/Http/Controllers/AuthController.php yang sudah kita buat. Lalu kita akan isi perintah code untuk membuat proses autentikasi , verifikasi, dan logout

6. Kemudian buat code AuthController.php seperti dibawah ini

```

app > Http > Controllers > AuthController.php > AuthController
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\UserModel;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Auth;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Support\Facades\Validator;
10
11 class AuthController extends Controller
12 {
13     //
14     public function index()
15     {
16         // kita ambil data user lalu simpan pada variabel $user
17         $user = Auth::user();
18
19         // kondisi jika user nya ada
20         if ($user) {
21             // jika user nya memiliki level admin
22             if ($user->level_id == '1') {
23                 return redirect()->intended('admin');
24             }
25             // jika user nya memiliki level manager
26             else if ($user->level_id == '2') {
27                 return redirect()->intended('manager');
28             }
29         }
30         return view('login');
31     }
32

```

```

33     public function proses_login(Request $request)
34     {
35         // kita buat validasi pada saat tombol login di klik
36         // validasinya username dan password wajib diisi
37         $request->validate([
38             'username' => 'required',
39             'password' => 'required',
40         ]);
41         // ambil data request username & password saja
42         $credential = $request->only('username', 'password');
43         // cek jika data username dan password valid(sesuai) dengan data
44         if (Auth::attempt($credential)) {
45             // kalau berhasil simpan data usernya di variable $user
46             $user = Auth::user();
47             // cek lagi jika level user admin maka arahkan ke halaman admin
48             if ($user->level_id == '1') {
49                 return redirect()->intended('admin');
50             }
51
52             // tapi jika level usernya user biasa maka arahkan ke halaman user
53             elseif ($user->level_id == '2') {
54                 return redirect()->intended('manager');
55             }
56             // jika belum ada role maka ke halaman /
57             return redirect()->intended('/');
58         }
59         // jika ga ada data user yang valid maka kembalikan lagi ke halaman login
60         // pastikan kirim pesan error juga kalau login gagal ya
61         return redirect('login')
62             ->withInput()
63             ->withErrors(['login_gagal' => 'Pastikan kembali username dan password yang dimasukan sudah benar']);
64
65

```

```

66     public function register()
67     {
68         // tampilkan view register
69         return view('register');
70     }
71
72     // aksi form register
73     public function proses_register(Request $request)
74     {
75         // kita buat validasi untuk proses register
76         // validasinya yaitu semua field wajib diisi
77         // validasinya username itu harus unique atau tidak boleh duplicate username
78         $validator = Validator::make($request->all(), [
79             'nama' => 'required',
80             'username' => 'required|unique:m_user',
81             'password' => 'required',
82         ]);
83
84         if ($validator->fails()) {
85             return redirect('/register')->withErrors($validator)->withInput();
86         }
87
88         // kalau berhasil isi level & hash passwordnya biar secure
89         $request['level_id'] = '2';
90         $request['password'] = Hash::make($request->password);
91
92         // masukkan semua data pada request ke table user
93         UserModel::create($request->all());
94
95         // kalau berhasil arahkan ke halaman login
96         return redirect()->route('login');
97     }
98
99     public function logout(Request $request)
100    {
101        // logout harus menghapus sessionnya
102        $request->session()->flush();
103
104        // jalankan juga fungsi logout auth
105        Auth::logout();
106        // kembali ke halaman login
107        return redirect('login');
108    }
109 }

```

7. Kemudian tambahkan kode pada route pada file routes/web.php seperti kode program dibawah

```

// Kita atur juga untuk middleware menggunakan group pada routing
// didalamnya terdapat group untuk mengecek kondisi login
// jika user yang login merupakan admin maka akan diarahkan ke AdminController
// jika user yang login merupakan manager maka akan diarahkan ke UserController
Route::group(['middleware' => ['auth']], function () {
    Route::group(['middleware' => ['cek_login:1']], function () {
        Route::resource('admin', AdminController::class);
    });
    Route::group(['middleware' => ['cek_login:2']], function () {
        Route::resource('manager', ManagerController::class);
    });
});

```

8. Kemudian buat file controller dengan nama AdminController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah dibawah

```

PS D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS> php artisan make:controller AdminController

INFO Controller [D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS\app\Http\Controllers\AdminController.php] created successfully.

PS D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS>

```

9. Buka folder App/Http/Controllers dan isi file AdminController seperti dibawah

```

app > Http > Controllers > AdminController.php > AdminController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class AdminController extends Controller
8  {
9      //
10     public function index()
11     {
12         return view('admin');
13     }
14 }
15

```

10. Kemudian buat file controller dengan nama ManagerController sebagai halaman yang boleh diakses oleh admin saja. Jalankan perintah : `php artisan make:controller ManagerController`

```

PS D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS> php artisan make:controller ManagerController

INFO Controller [D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS\app\Http\Controllers\ManagerController.php] created successfully.

PS D:\Abi\kuliah\semester 4\PBL\Code\PWL_POS>

```

11. Buka folder `App/Http/Controllers` dan isi file `ManagerController` seperti dibawah

```

app > Http > Controllers > ManagerController.php > ManagerController > index
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class ManagerController extends Controller
8  {
9      //
10     public function index()
11     {
12         return view('manager');
13     }
14 }
15

```

12. Pada langkah terakhir yaitu membuat layout sederhana
13. Buatlah halaman login dengan membuka folder `resources/views` dengan nama file `login.blade.php`. Lalu kita isi filenya seperti contoh dibawah

```

resources > views > login.blade.php > form > div.row > div.col-7 > div.check-primary
1  @extends('adminlte::auth.auth-page', ['auth_type' => 'login'])
2
3  @section('admin_css_pre')
4      <link rel="stylesheet" href="{{ asset('vendor/icheck-bootstrap/icheck-bootstrap.min.css') }}">
5  @stop
6
7  @php($login_url = View::getSection('login_url') ?? config('adminlte.login_url', 'login'))
8  @php($register_url = View::getSection('register_url') ?? config('adminlte.register_url', 'register'))
9  @php($password_reset_url = View::getSection('password_reset_url') ?? config('adminlte.password_reset_url', 'password/reset'))
10
11  @if (config('adminlte.use_route_url', false))
12      @php($login_url = $login_url ? route($login_url) : '')
13      @php($register_url = $register_url ? route($register_url) : '')
14      @php($password_reset_url = $password_reset_url ? route($password_reset_url) : '')
15  @else
16      @php($login_url = $login_url ? url($login_url) : '')
17      @php($register_url = $register_url ? url($register_url) : '')
18      @php($password_reset_url = $password_reset_url ? url($password_reset_url) : '')
19  @endif
20
21  @section('auth_header', __('adminlte::adminlte.login_message'))
22
23  @section('auth_body')
24      @error('login_gagal')
25          <div class="alert alert-warning alert-dismissible fade show" role="alert">
26              <span class="alert-inner--text"><strong>Warning!</strong>{{ $message }}</span>
27              <button type="button" class="close" data-dismiss="alert" aria-label="Close">
28                  <span aria-hidden="true">&times;</span>
29              </button>
30          </div>
31      @enderror
32      <form action="{{ url('proses_login') }}" method="post">
33          @csrf
34
35          <!-- email field -->
36          <div class="input-group mb-3">
37              <input type="text" name="username" class="form-control @error('username') is-invalid @enderror"
38                  value="{{ old('username') }}" placeholder="Username" autofocus>
39
40              <div class="input-group-append">
41                  <div class="input-group-text">
42                      <span class="fas fa-envelope {{ config('adminlte.classes_auth_icon', '') }}"></span>
43                  </div>
44              </div>

```

```

45
46          @error('username')
47              <span class="invalid-feedback" role="alert">
48                  <strong>{{ $message }}</strong>
49              </span>
50          @enderror
51      </div>
52
53      <!-- Password field -->
54      <div class="input-group mb-3">
55          <input type="password" name="password" class="form-control @error('password') is-invalid @enderror"
56              placeholder="{{ __('adminlte::adminlte.password') }}">
57
58          <div class="input-group-append">
59              <div class="input-group-text">
60                  <span class="fas fa-lock {{ config('adminlte.classes_auth_icon', '') }}"></span>
61              </div>
62          </div>
63
64          @error('password')
65              <span class="invalid-feedback" role="alert">
66                  <strong>{{ $message }}</strong>
67              </span>
68          @enderror
69      </div>
70
71      <!-- login field -->
72      <div class="row">
73          <div class="col-7">
74              <div class="icheck-primary" title="{{ __('adminlte::adminlte.remember_me_hint') }}">
75                  <input type="checkbox" name="remember" id="remember" {{ old('remember') ? 'checked' : '' }}>
76
77                  <label for="remember">
78                      {{ __('adminlte::adminlte.remember_me') }}
79                  </label>
80              </div>
81          </div>
82

```

```

83 <div class="col-5">
84 <button type="submit"
85     class="btn btn-block {{ config('adminlte.classes_auth_btn', 'btn-flat btn-primary') }}">
86
87 <span class="fas fa-sign-in-alt"></span>
88 {{ __('adminlte::adminlte.sign_in') }}
89 </button>
90 </div>
91 </div>
92 </form>
93 @stop
94
95 @section('auth_footer')
96 @if ($register_url)
97 <p class="my-0">
98 <a href="{{ route('register') }}">
99 {{ __('adminlte::adminlte.register_a_new_membership') }}
100 </a>
101 </p>
102 @endif
103 @stop
104

```

14. Buatlah halaman register dengan membuka folder resources/views dengan nama file register.blade.php. Lalu kita isi filenya seperti contoh dibawah


```

45         <span class="fas fa-user" {{ config('adminlte.classes_auth_icon', '') }}></span>
46     </div>
47 </div>
48
49 @error('username')
50     <span class="invalid-feedback" role="alert">
51         <strong>{{ $message }}</strong>
52     </span>
53 @enderror
54 </div>
55
56 {{-- Password field --}}
57 <div class="input-group mb-3">
58     <input type="password" name="password" class="form-control @error('password') is-invalid @enderror"
59         placeholder="{{ __('adminlte::adminlte.password') }}">
60
61     <div class="input-group-append">
62         <div class="input-group-text">
63             <span class="fas fa-lock" {{ config('adminlte.classes_auth_icon', '') }}></span>
64         </div>
65     </div>
66
67     @error('password')
68         <span class="invalid-feedback" role="alert">
69             <strong>{{ $message }}</strong>
70         </span>
71     @enderror
72 </div>
73
74 {{-- register button --}}
75 <button type="submit" class="btn btn-block" {{ config('adminlte.classes_auth_btn', 'btn-flat btn-primary') }}>
76     <span class="fas fa-user-plus"></span>
77     {{ __('adminlte::adminlte.register') }}
78 </button>
79 </form>
80 @stop
81
82 @section('auth_footer')
83     @if ($register_url)
84         <p class="my-0">
85             <a href="{{ route('login') }}">
86                 {{ __('adminlte::adminlte.i_already_have_a_membership') }}
87             </a>
88         </p>
89     @endif
90 @stop

```

15. Buatlah halaman register dengan membuka folder resources/views dengan nama file admin.blade.php. Lalu kita isi filenya seperti contoh dibawah

```

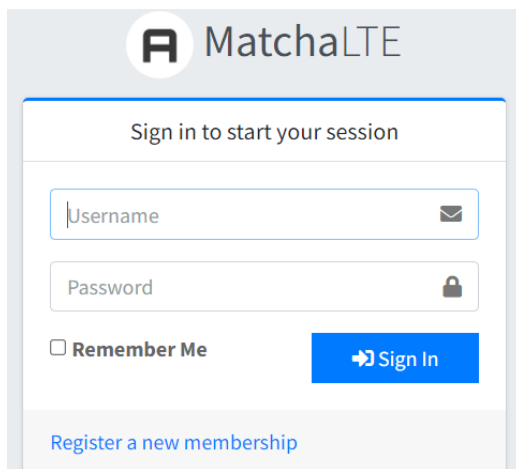
resources > views > admin.blade.php > div.container > div.card > div.card-body
1  @extends('layout.app')
2
3  @section('subtitle', 'Admin')
4  @section('content_header_title', 'Home')
5  @section('content_header_subtitle', 'Admin')
6
7  @section('content')
8      <div class="container">
9          <div class="card">
10             <div class="card-header">
11                 Tampilan <?php echo Auth::user()->level_id == 1 ? 'Admin' : 'Manager'; ?>
12             </div>
13             <div class="card-body">
14                 <h1>Login Sebagai:
15                 <?php echo Auth::user()->level_id == 1 ? 'Admin' : 'Manager'; ?>
16                 </h1>
17                 <a href="{{ route('logout') }}">Logout</a>
18             </div>
19         </div>
20     </div>
21 @endsection
22
23 @push('js')
24 @endpush

```

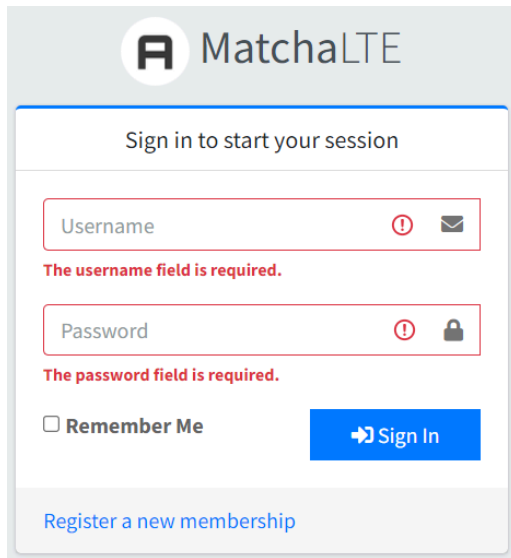
16. Buatlah halaman register dengan membuka folder resources/views dengan nama file manager.blade.php. Lalu kita isi filenya seperti contoh dibawah

```
resources > views > manager.blade.php > ...
1  @extends('layout.app')
2
3  @section('subtitle', 'Manager')
4  @section('content_header_title', 'Home')
5  @section('content_header_subtitle', 'Manager')
6
7  @section('content')
8      <div class="container">
9          <div class="card">
10             <div class="card-header">
11                 Tampilan <?php echo Auth::user()->level_id == 1 ? 'Admin' : 'Manager'; ?>
12             </div>
13             <div class="card-body">
14                 <h1>Login Sebagai:
15                 <?php echo Auth::user()->level_id == 1 ? 'Admin' : 'Manager'; ?>
16                 </h1>
17                 <a href="{{ route('logout') }}">Logout</a>
18             </div>
19         </div>
20     </div>
21 @endsection
22
23 @push('js')
24 @endpush
```

17. Jalankan kode diatas dan buat laporan dengan isi screenshoot hasil dan beri penjelasan pada setiap screenshoot hasil dan commit perubahan pada git.
Tampilan awal




Ketika form kosong di submit




MatchaLTE

Sign in to start your session

Username ! 

The username field is required.

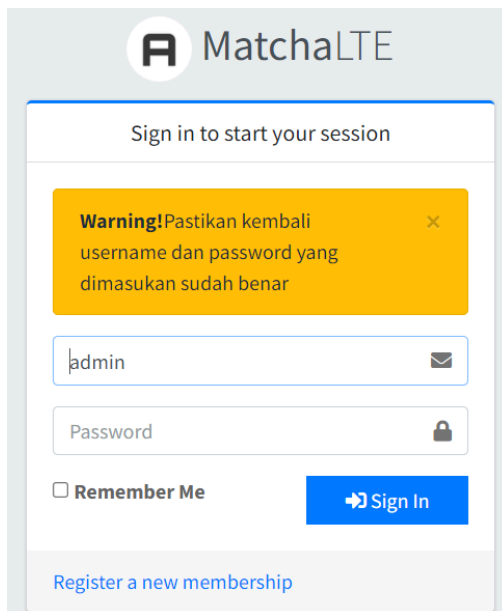
Password ! 

The password field is required.

☐ Remember Me ➔ Sign In

[Register a new membership](#)


Ketika password salah




MatchaLTE

Sign in to start your session

Warning! Pastikan kembali username dan password yang dimasukan sudah benar ×

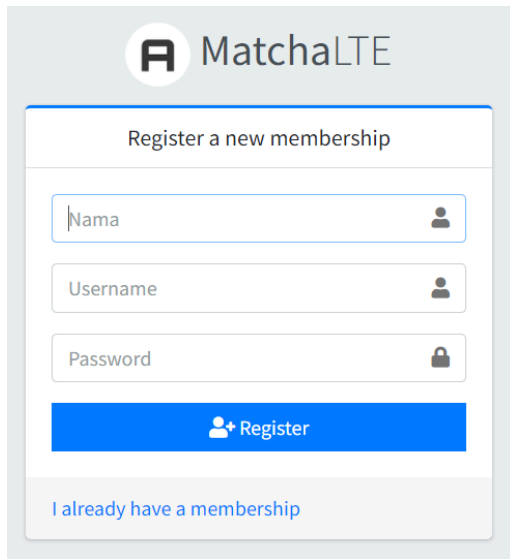
admin 

Password 

☐ Remember Me ➔ Sign In

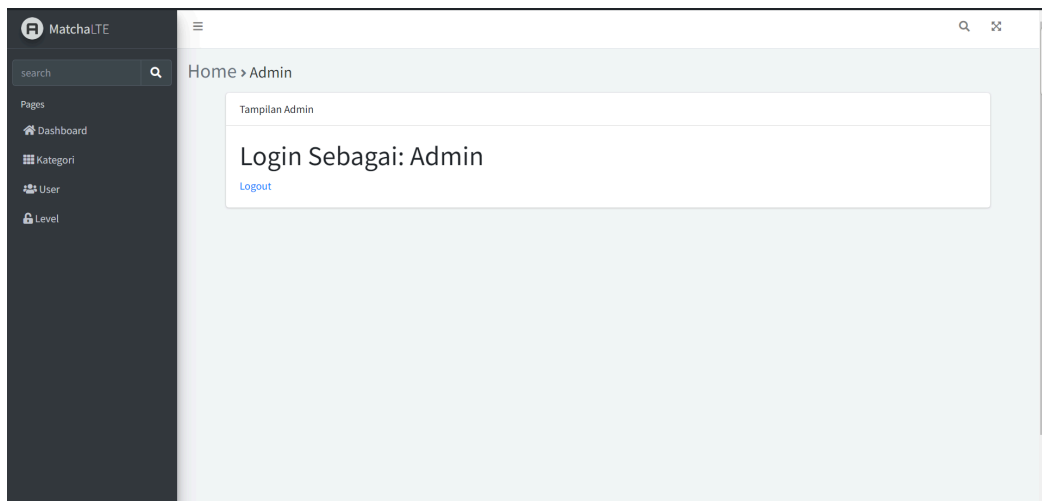
[Register a new membership](#)

Tampilan register



The image shows a registration form for MatchaLTE. At the top, there is a logo with a stylized 'A' inside a circle, followed by the text 'MatchaLTE'. Below the logo, the heading 'Register a new membership' is centered. The form consists of three input fields: 'Nama' (Name), 'Username', and 'Password'. Each field has a small user icon to its right. Below the password field is a blue button with a white user icon and the text 'Register'. At the bottom of the form, there is a link that says 'I already have a membership'.

jika benar menggunakan akun admin



jika benar menggunakan akun manager

