

Nama : Izzatunnisa Arwadi

NPM : 1806269726

Kelas : Ekstensi

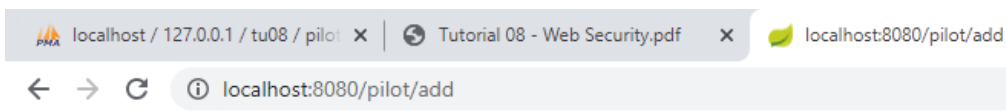
Latihan

1. Tambahkan user dengan role ADMIN, agar fitur add, view, update dan delete pilot hanya dapat diakses oleh admin.

Answer:

```
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests()
        .antMatchers("/css/**").permitAll()
        .antMatchers("/js/**").permitAll()
        .antMatchers("/flight/**").hasAnyAuthority("PILOT")
        .antMatchers("/flight/**").hasAnyAuthority("ADMIN")
        .antMatchers("/pilot/**").hasAnyAuthority("ADMIN")
}
```

Pada method configure ditambahkan antMatchers untuk pilot dan inputkan ADMIN sebagai parameter pada hasAnyAuthority



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Tue Apr 14 08:25:21 ICT 2020

There was an unexpected error (type=Forbidden, status=403).

Forbidden

Ketika role UMUM melakukan akses pilot, maka akan ditampilkan whitelabel error page dengan error forbidden

A screenshot of a web form titled 'Tambah User Baru'. It contains three input fields: 'User Name' with the value 'izzarwd', 'Password:' with the value '1234', and 'Role:' with the value 'ADMIN'. Below the fields is a blue button labeled 'Simpan'.

Menambahkan user dengan role ADMIN

Ketika dilakukan akses halaman pilot dengan role ADMIN, halaman berhasil dilakukan akses.

2. Tambahkan fitur update password yang dapat diakses oleh semua role user

Answer:

```

1 package com.apap.tu08.model;
2
3 public class PasswordModel {
4     private String passwordLama;
5     private String passwordBaru;
6     private String konfirmasiPassword;
7 }

```

Dilakukan pembuatan model, untuk menyimpan variable-variabel password

```

@RequestMapping(value = "/changePassword", method = RequestMethod.POST)
private String changePassword(@Principal principal, @ModelAttribute PasswordModel password, Model model) {
    UserRoleModel userSekarang = userRoleDb.findByUsername(principal.getName());

    PasswordEncoder bcpe = new BCryptPasswordEncoder();

    Boolean pencocokan1 = bcpe.matches(password.getPasswordLama(), userSekarang.getPassword());
    Boolean pencocokan2 = password.getPasswordBaru().equals(password.getKonfirmasiPassword());

    if (pencocokan1 && pencocokan2) {
        UserRoleModel user = new UserRoleModel();

        user.setId(userSekarang.getId());
        user.setUsername(userSekarang.getUsername());
        user.setRole(userSekarang.getRole());
        user.setPassword(password.getPasswordBaru());

        userService.addUser(user);

        model.addAttribute("success", "Success Change Password!");
    } else {
        List<String> pesan = new ArrayList<String>();

        if (!pencocokan1) {
            pesan.add("Passwrod Salah");
        }
        if (!pencocokan2) {
            pesan.add("Password Tidak Cocok");
        }
        model.addAttribute("error", pesan);
    }

    return "home";
}

```

Pada UserRoleController dilakukan pembuatan method changePassword dengan memanfaatkan penggunaan model Password, principal serta PasswordEncoder. Model password digunakan untuk mendapatkan ketiga variable password (lama, baru, konfirmasi). Principal digunakan untuk mengetahui user yang login saat ini dan PasswordEncoder digunakan untuk melakukan encode pada password.

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
<title>Update Password</title>
<object th:include="fragments/fragment :: css" th:remove="tag"></object>
<object th:include="fragments/fragment :: js" th:remove="tag"></object>
</head>

<body>
<nav th:replace="fragments/fragment :: navbar"></nav>

<div class="container-fluid"><br>
<h2 th:text="Hello" + ${HttpServletRequest.remoteUser} + "!">login as</h2>
<div class="col-md-6">
<div class="card">
<div class="card-header">Update Password</div>
<div th:if="${error}">
<p style="color: red">Error: </p>
<p th:each="error: ${error}" style="color: red" th:text="${error}"></p>
</div>
<div th:if="${success}">
<p style="color: green">Password Berhasil Diubah!</p>
</div>
<div th:if="${success}">
<p style="color: green">Password Berhasil Diubah!</p>
</div>
<div class="card-body">
<form th:action="@{/user/changePassword}" method="post">
<div class="row form-group">
<label class="col-sm-4"> Password Lama: </label> <input
class="col-sm-8 form-control input-sm" type="password" name="passwordLama" required>
</div>
<div class="row form-group">
<label class="col-sm-4"> Password Baru: </label> <input
class="col-sm-8 form-control input-sm" type="password" name="passwordBaru" required>
</div>
<div class="row form-group">
<label class="col-sm-4"> Konfirmasi Password Baru: </label> <input
class="col-sm-8 form-control input-sm" type="password" name="konfirmasiPassword"
required>
</div>
<div>
<button type="submit" class="btn btn-primary">Simpan</button>
</div>
</form>
</div>
</div>
</div>
</div>
</body>
</html>

```

Dilakukan pembuatan halaman html untuk update password

```

<li class="nav-item active">
<a class="nav-link" th:href="@{/updatePassword}">Update Password</a>
</li>

```

Pada fragment ditambahkan menu update password

Ketika dilakukan klik pada menu update password, akan ditampilkan halaman seperti di atas

Ketika password berhasil diubah terdapat notifikasinya

3. Pada saat pembuatan user baru dan update password user, tambahkan ketentuan agar password user mengandung angka dan huruf serta minimal memiliki 8 karakter.