

Bagian Ke 12 NodeJS Fullstack Dev

Passport & Login

Sekarang kita akan membuat login page dan profile page.

Buat folder baru pada root folder dan bernama "views".

Kemudian buat 2 file baru yaitu

```
views/login.ejs
```

dan

```
views/profile.ejs
```

serta

```
views/home.ejs
```

kemudian

```
views/layout.ejs
```

Kemudian copas konten file layout.ejs dari project latihan yang lalu:

```
<!doctype html>  
<html lang="en">
```

```
<head>
    <% include ./partials/head %>
</head>

<body>

    <% include ./partials/navbar %>

    <%- body -%>

    <% include ./partials/footer %>

    <% include ./partials/javascript %>
</body>
</html>
```

Copas juga seluruh isi folder partials dibawah views.

Update file login.ejs, profile.ejs, home.ejs dengan menambahkan kode sbb:

```
<% layout('layout') %>
```

Gunanya untuk merender template dari 'layout.ejs' ke seluruh halaman web.

Kembali ke server.js dan buatlah route untuk halaman profile:

```
app.get('/profile', function(req, res, next){
    res.render('profile')
});
```

Tidak lupa kita menambahkan middleware untuk passport initialize yang

telah dibuat sebelumnya:

```
app.use(passport.initialize());  
app.use(passport.session());
```

Baris pertama kita memberikan initial kepada passport. Baris kedua kita menyimpannya ke dalam session.

Selanjutnya kita require file passport.js didalam file server.js:

```
var passportConf = require('./passport');
```

pada file profile.ejs tambahkan:

```
<h1>Profile</h1>
```

pada file home.ejs tambahkan:

```
<h1>Profile</h1>
```

Kemudian run the server !

```
nodemon server
```

Buka browser localhost:3000

Menambahkan CSS

Buat folder baru beri nama "public".

Buat folder dibawah nya public/css.

Buat file baru beri nama public/css/custom.css dan update code css sbb:

```
body {  
    padding-top: 50px;  
    padding-bottom: 20px;  
}
```

Update link css pada file partials/head.ejs:

```
<link href="/css/custom.css" rel="stylesheet">
```

Kemudian update server.js tambahkan code pada middleware sbb:

```
app.use(express.static(__dirname + '/public'));
```

Untuk memberitahukan express mencari static folder yaitu "public"

Kemudian test di browser lagi localhost:3000.

Membuat form login

Update file login.ejs sbb:

```
<form action="" method="post">  
    <input type="text" name="email"></input>  
    <input type="text" name="password"></input>  
  
    <button class="btn btn-
```

```
primary">Login</button>
</form>
```

Test di localhost:3000/login

Membuat halaman logout

Pada halaman server.js buat buat method logout dengan get sbb:

```
app.get('/logout', function(req, res, next){
    req.logout();
    res.redirect('/');
});
```

Create new user

Kita harus membuat user baru dengan atribut email dan password (sebelumnya kita buat name dan age).

Copas method post create user pada pelajaran sebelumnya:

```
app.post('/create-user', function(req, res, next){
    var user = new User();
    user.email = req.body.email;
    user.password = req.body.password;
    user.save(function(err){
        if (err) console.log (err)
        res.json(user);
    });
});
```

Jangan lupa require user:

```
var User = require('./models/user');
```

Buka postman dan lakukan

setup sebagai post localhost:3000/create-user

Isi form dengan:

email : john@gmail.com

password : abc123

Kemudian klik tombol "Send"

Error to fix:

Pada file passport.js perbaiki passport.deserialize seperti sbb:

```
passport.deserialize(function(id, done){  
    User.findById(id, function(err, user){  
        done(err, user);  
    });  
});
```

Bila user sudah dibuat pada Postman kemudian coba untuk login di localhost:3000/login.

Continue next video

Copyright ©GandrungStudio 2018 Dilarang memperbanyak dan mendistribusikan Materi Training ini tanpa ijin dari PT.Gandrung Media Corporation. Pelanggaran akan hak cipta ini akan kami ajukan ke Institusi Hukum yang berlaku.