

**Laporan Praktikum**

**Mata Kuliah**

**Pemrograman Web**



**Tugas Pertemuan 6**

**“SESSION”**

Dosen Pengampu:

Willdan Aprizal Arifin, S.Pd., M.Kom.

Disusun Oleh:

Habibirrohimi

2300149

**PROGRAM STUDI SISTEM INFORMASI KELAUTAN**

**UNIVERSITAS PENDIDIKAN INDONESIA**

**2024**

## **I. PENDAHULUAN**

Dalam dunia teknologi yang terus berkembang, kemampuan untuk membuat aplikasi web yang dapat mengelola data dengan efisien menjadi sangat penting. Praktikum ini bertujuan untuk memperkenalkan mahasiswa pada pengembangan aplikasi web menggunakan teknologi modern seperti Node.js dan MySQL.

Fokus utama praktikum ini adalah membuat sistem manajemen logistik sederhana. Melalui proyek ini, mahasiswa akan belajar cara:

1. Menghubungkan aplikasi web dengan database
2. Membuat fitur untuk menambah, membaca, mengubah, dan menghapus data (CRUD)
3. Membuat tampilan web yang menarik dan responsif

Proyek ini tidak hanya mengajarkan teknik pemrograman, tetapi juga memberikan pengalaman praktis dalam membangun aplikasi yang bisa digunakan dalam situasi nyata. Dengan menyelesaikan praktikum ini, mahasiswa diharapkan mendapatkan pemahaman yang lebih baik tentang cara kerja aplikasi web modern dan keterampilan yang diperlukan untuk mengembangkannya.

## **II. ALAT DAN BAHAN**

### **2.1 Alat Dan Bahan**

- Node.js
- MySQL
- Express.js
- Body-parser
- EJS (Embedded JavaScript templates)
- Laragon (sebagai server lokal)
- Tailwind CSS (untuk styling)

### III. PENJELASAN

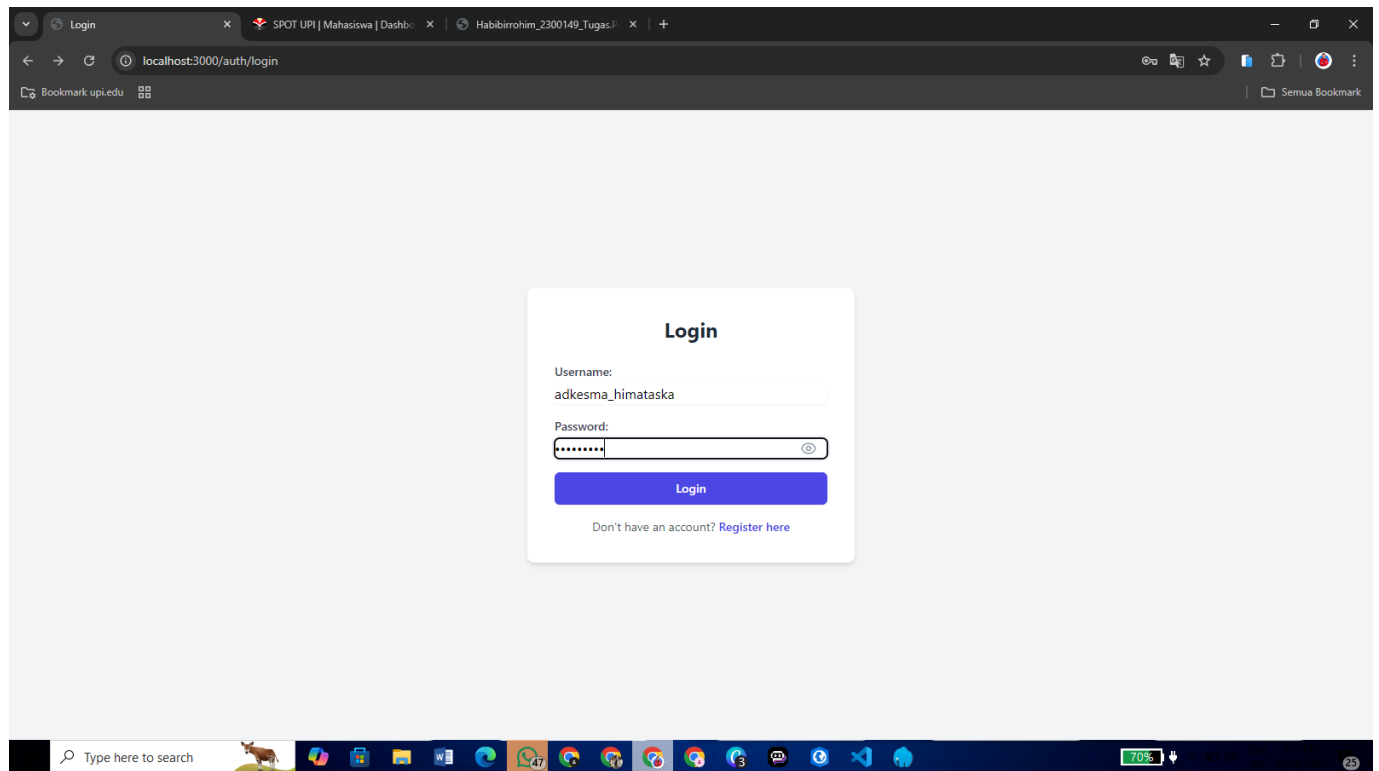
Pemahaman saya pada praktikum hari ini  
Membuat Sistem Login dengan Node.js

1. Session seperti ingatan sementara komputer tentang siapa kita. Saat kita login ke sebuah website, komputer "mengingat" kita sampai kita logout. Ini membantu website tahu siapa kita tanpa harus login terus-menerus.
2. Menggunakan Session di Node.js:  
Kita pakai alat bernama 'express-session' untuk membuat session. Ini seperti memberi komputer buku catatan untuk menulis siapa saja yang sedang login.
3. Menyusun Proyek:  
Kita belajar menyusun file-file proyek dengan rapi. Ini seperti menyusun lemari, di mana setiap laci punya isi yang berbeda (misalnya, satu laci untuk koneksi database, satu lagi untuk tampilan website).
4. Menghubungkan ke Database:  
Kita menggunakan MySQL sebagai tempat menyimpan data pengguna. Ini seperti buku besar yang menyimpan semua informasi pengguna.
5. Sistem Login dan Daftar:  
Saat pengguna daftar, kita simpan password mereka dengan cara aman (diacak). Saat login, kita periksa apakah username dan password benar. Kalau benar, kita buat session untuk mengingat bahwa pengguna ini sudah login. Saat logout, kita hapus ingatan ini.
6. Membuat Tampilan Website:  
Kita pakai EJS untuk membuat halaman web. Ini membantu kita membuat halaman yang bisa berubah-ubah isinya tergantung siapa yang login.
7. Menjaga Keamanan:  
Kita buat aturan agar halaman tertentu hanya bisa dilihat oleh pengguna yang sudah login.
8. Membuat Tampilan Depan:  
Kita buat halaman-halaman sederhana untuk login, daftar, dan profil pengguna menggunakan HTML dan CSS.
9. Mengatur Alur Website:  
Kita buat "jalan" di website untuk login, daftar, dan logout. Ini memastikan pengguna bisa dengan mudah melakukan apa yang mereka inginkan.  
Apa yang Kita Pelajari pada praktikum ini:
  1. Pentingnya mengingat pengguna yang login untuk membuat website lebih nyaman dipakai.
  2. Cara membuat sistem login yang aman dengan Node.js.
  3. Menggunakan database untuk menyimpan informasi pengguna.
  4. Menjaga keamanan informasi penting seperti password.
  5. Menyusun proyek dengan rapi agar mudah dikerjakan.

6. Menggabungkan bagian yang dilihat pengguna (tampilan) dengan bagian yang bekerja di belakang layar.
7. Membuat aturan agar website aman dan berfungsi dengan baik.

## Tampilan Ouput nya

### Login



## Register

Register

Username  
Habibirroh

Email  
hhabirroh@gmail.com

Password  
\*\*\*\*\*

Register

Already have an account? [Login here](#)

## Tampilan Profile

Welcome, Habibirroh

Email: hhabirroh@gmail.com

Logout

## Halaman Login

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Login</title>
7   <script src="https://cdn.tailwindcss.com"></script>
8   <script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/js/all.min.js"></script>
9 </head>
10 <body class="bg-gray-100 flex items-center justify-center min-h-screen">
11   <div class="bg-white p-8 rounded-lg shadow-md w-96">
12     <div class="text-2xl font-bold mb-6 text-center text-gray-800">Welcome, <%= user.username %></div>
13     <div class="text-gray-600 mb-4 text-center">Don't have an account? <a href="/auth/register" class="text-blue-500 font-medium">Register here</a></div>
14     <div class="text-center">
15       <input type="text" class="w-full p-2 border border-transparent rounded-md shadow-sm text-sm font-medium text-white bg-red-600 hover:bg-red-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-white">
16     </div>
17     <div class="text-center">
18       <input type="password" class="w-full p-2 border border-transparent rounded-md shadow-sm text-sm font-medium text-white bg-red-600 hover:bg-red-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-white">
19     </div>
20     <div class="text-center">
21       <button type="button" class="w-full p-2 border border-transparent rounded-md shadow-sm text-sm font-medium text-white bg-red-600 hover:bg-red-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-white">Login</button>
22     </div>
23   </div>
24 </body>
25 </html>
```

## Halaman Profile

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Profile</title>
7   <script src="https://cdn.tailwindcss.com"></script>
8 </head>
9 <body class="bg-gray-100 flex items-center justify-center min-h-screen">
10   <div class="bg-white p-8 rounded-lg shadow-md w-96">
11     <div class="text-2xl font-bold mb-6 text-center text-gray-800">Welcome, <%= user.username %></div>
12     <div class="text-gray-600 mb-4 text-center">Don't have an account? <a href="/auth/register" class="text-blue-500 font-medium">Register here</a></div>
13     <div class="text-center">
14       <input type="text" class="w-full p-2 border border-transparent rounded-md shadow-sm text-sm font-medium text-white bg-red-600 hover:bg-red-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-white">
15     </div>
16   </div>
17 </body>
18 </html>
```

## Halaman Register



## Routers/Auth 1

```
1  const express = require('express');
2  const router = express.Router();
3  const bcrypt = require('bcryptjs');
4  const db = require('../config/db');
5
6  // Render halaman register
7  router.get('/register', (req, res) => {
8    res.render('register');
9  });
10
11 // Proses register user
12 router.post('/register', (req, res) => {
13   const { username, email, password } = req.body;
14
15   // Validasi server-side untuk memastikan input tidak kosong
16   if (!username || !email || !password) {
17     return res.send('Please fill in all fields.');
```



## Routers/Auth 2

```
1 // Render halaman profil
2 router.get('/profile', (req, res) => {
3   if (req.session.user) {
4     res.render('profile', { user: req.session.user });
5   } else {
6     res.redirect('/auth/login');
7   }
8 });
9
10 // ... (kode lainnya tetap sama)
11
12 // Proses Logout
13 router.get('/logout', (req, res) => {
14   req.session.destroy((err) => {
15     if (err) {
16       console.error('Error destroying session:', err);
17     }
18     res.redirect('/auth/login');
19   });
20 });
21
22 module.exports = router;
23
24 // routes/auth.js
25 router.post('/login', (req, res) => {
26   const { username, password } = req.body;
27
28   // Proses Login
29   db.query("SELECT * FROM user WHERE username = ?", [username], (err, result) => {
30     if (err) throw err;
31
32     if (result.length > 0) {
33       const user = result[0];
34
35       // Periksa password
36       if (bcrypt.compareSync(password, user.password)) {
37         // Simpan informasi user di session
38         req.session.user = user;
39
40         // Arahkan user ke halaman profile
41         res.redirect('/auth/profile');
42       } else {
43         res.send('Password salah');
44       }
45     } else {
46       res.send('User tidak ditemukan');
47     }
48   });
49 });
50
51 // routes/auth.js
52 router.get('/profile', (req, res) => {
53   // Periksa apakah user telah Login
54   if (req.session.user) {
55     // Tampilkan halaman profile
56     res.render('profile', { user: req.session.user });
57   } else {
58     // Arahkan user ke halaman Login
59     res.redirect('/auth/login');
60   }
61 });
```

## Views/APP.JS

```
1  const express = require('express');
2  const bodyParser = require('body-parser');
3  const session = require('express-session');
4  const authRoutes = require('./routes/auth');
5  const path = require('path');
6  const app = express();
7
8  // Set EJS sebagai template engine
9  app.set('view engine', 'ejs');
10 app.set('views', path.join(__dirname, 'views'));
11
12 // Middleware
13 app.use(bodyParser.json());
14 app.use(bodyParser.urlencoded({ extended: true }));
15 app.use(session({
16   secret: 'your-secret-key',
17   resave: false,
18   saveUninitialized: true,
19 }));
20
21 // Set static folder
22 app.use(express.static(path.join(__dirname, 'public')));
23
24 // Routes
25 app.use('/auth', authRoutes);
26
27 // Root Route
28 app.get('/', (req, res) => {
29   if (req.session.user) {
30     return res.redirect('/auth/profile');
31   }
32   return res.redirect('/auth/login');
33 });
34
35 // Menjalankan Server
36 app.listen(3000, () => {
37   console.log('Server running on http://localhost:3000');
38 });
39
40
41
```

## **V. KESIMPULAN**

Praktikum ini berhasil memberikan pengalaman berharga dalam pengembangan aplikasi web full-stack. Mahasiswa telah berhasil:

1. Menguasai dasar-dasar Node.js, Express.js, dan MySQL.
2. Belajar menghubungkan backend dengan database.
3. Mengembangkan frontend menggunakan EJS dan Tailwind CSS.
4. Memahami alur kerja pengembangan aplikasi web modern.

Keterampilan yang diperoleh dalam praktikum ini membentuk dasar yang kuat untuk pengembangan aplikasi web yang lebih kompleks di masa depan. Mahasiswa kini memiliki pemahaman praktis tentang cara membangun aplikasi web yang fungsional dan responsif.