

Laporan Praktikum
Mata Kuliah Pemograman Berorientasi Objek



Pertemuan 6 Tugas ke 6
“CRUD Node JS Css,
Session”

Dosen Pengampu :
Willdan Aprizal Arifin, S.Pd., M.Kom

Disusun Oleh :
Muhammad Zikri Alhaq (2308504)

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA

2024

Tujuan

Tujuan dari proyek Pada umumnya, metrik ini juga mencakup kegiatan seperti: pencarian informasi, mengisi formulir untuk membaca konten, scrolling halaman website, menambahkan item ke keranjang belanja, mencari tiket transportasi secara online, dan lain sebagainya.

Dasar Teori

Web session, atau sesi web adalah jumlah waktu yang dihabiskan oleh pengguna untuk menjelajahi website tertentu, mulai dari saat mereka memasuki halaman pertama, hingga mereka meninggalkan website tersebut.

Sesi web juga didefinisikan sebagai serangkaian tindakan yang dilakukan secara berurutan oleh pengunjung website dalam jangka waktu tertentu.

Pada umumnya, metrik ini juga mencakup kegiatan seperti: pencarian informasi, mengisi formulir untuk membaca konten, scrolling halaman website, menambahkan item ke keranjang belanja, mencari tiket transportasi secara online, dan lain sebagainya.

Setiap interaksi yang Anda lakukan terhadap suatu website dicatat sebagai sesi web ke dalam properti situs tersebut.

Langkah-Langkah Percobaan

Proyek ini terdiri dari beberapa file penting:

- app.js: File utama yang mengatur server dan rute aplikasi.
- views/index.ejs: Halaman utama yang menampilkan daftar produk.
- views/edit.ejs: Halaman untuk mengedit produk.
- public/style.css: File CSS untuk styling aplikasi.
- package.json dan package-lock.json: File yang mengelola dependensi proyek.

1. File app.js

```
- const express = require('express');  
- const bodyParser = require('body-parser');  
- const session = require('express-session');  
- const authRoutes = require('./routes/auth');  
- const path = require('path');  
- const app = express();
```

- Penjelasan

Bagian ini mengimpor modul-modul yang diperlukan untuk aplikasi.

Express digunakan sebagai framework web, body-parser untuk mengurai body permintaan HTTP, express-session untuk mengelola sesi pengguna, dan path untuk menangani jalur file.

```
- app.set('view engine', 'ejs');
```

- Baris ini mengatur EJS sebagai mesin template untuk merender tampilan HTML dinamis.

```
- app.use(bodyParser.json());  
- app.use(bodyParser.urlencoded({ extended: true }));  
- app.use(session({  
-   secret: 'your-secret-key',  
-   resave: false,  
-   saveUninitialized: true,  
- }));
```

- Ini mengonfigurasi middleware. Body-parser digunakan untuk mengurai data JSON dan form, sementara express-session diatur untuk mengelola sesi pengguna dengan kunci rahasia.

```
- app.use(express.static(path.join(__dirname, 'public')));
```

- Baris ini mengatur folder 'public' sebagai lokasi untuk menyimpan file statis seperti CSS dan JavaScript.

```
- app.use((req, res, next) => {
-   if (!req.session.user && req.path !== '/auth/login' && req.path !==
-     '/auth/register') {
-     return res.redirect('/auth/login');
-   } else if (req.session.user && req.path === '/') {
-     return res.redirect('/auth/profile');
-   }
-   next();
- });
```

- Ini adalah middleware untuk memeriksa status login pengguna. Jika pengguna belum login dan mencoba mengakses halaman selain login atau register, mereka akan diarahkan ke halaman login. Jika pengguna sudah login dan mencoba mengakses halaman utama, mereka akan diarahkan ke halaman profil

```
- app.use('/auth', authRoutes);
```

- Ini mengarahkan semua rute yang dimulai dengan '/auth' ke modul authRoutes.

```
- app.get('/', (req, res) => {
-   if (req.session.user) {
-     return res.redirect('/auth/profile');
-   }
-   return res.redirect('/auth/login');
- });
```

- Rute ini menangani permintaan ke halaman utama. Jika pengguna sudah login, mereka diarahkan ke halaman profil; jika belum, mereka diarahkan ke halaman login

```
- app.listen(3000, () => {
-   console.log('http://localhost:3000');
- });
```

- Ini memulai server pada port 3000 dan mencetak URL server ke konsol.

2. File auth.ejs

```
- const express = require('express');
- const router = express.Router();
- const bcrypt = require('bcryptjs');
- const db = require('../config/db');
```

- Bagian ini mengimpor modul yang diperlukan. Express.Router() digunakan untuk membuat router terpisah, bcrypt untuk hashing password, dan db untuk koneksi database.

```
- router.get('/register', (req, res) => {
-   res.render('register');
- });
```

- Rute ini merender halaman pendaftaran ketika pengguna mengakses '/auth/register'.

```
- router.post('/register', (req, res) => {
-   const { username, email, password } = req.body;
-   if (!username || !email || !password) {
-     return res.send('Please fill in all fields.');
```

```
-   }
-   const hashedPassword = bcrypt.hashSync(password, 10);
-   const query = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";
-   db.query(query, [username, email, hashedPassword], (err, result) => {
-     if (err) throw err;
-     res.redirect('/auth/login');
```

```
-   });
- });
```

- Rute ini menangani proses pendaftaran. Ia memeriksa apakah semua field terisi, mengenkripsi password, dan menyimpan data pengguna ke database. Setelah berhasil, pengguna diarahkan ke halaman login.

```
- router.get('/login', (req, res) => {
-   res.render('login');
- });
```

- Rute ini merender halaman login ketika pengguna mengakses '/auth/login'.

```
router.post('/login', (req, res) => {
  const { username, password } = req.body;
  const query = "SELECT * FROM users WHERE username = ?";
  db.query(query, [username], (err, result) => {
    if (err) throw err;
    if (result.length > 0) {
      const user = result[0];
      if (bcrypt.compareSync(password, user.password)) {
        req.session.user = user;
        req.session.loginTime = new Date();
        return res.redirect('/auth/profile');
      }
    }
    res.send('Invalid username or password');
  });
});
```

- Rute ini menangani proses login. Ia memeriksa kredensial pengguna, membandingkan password yang dimasukkan dengan hash yang tersimpan, dan jika cocok, menyimpan informasi pengguna dalam sesi dan mengarahkan ke halaman profil.

```
router.get('/profile', (req, res) => {
  if (req.session.user) {
    let sessionDuration = 0;
    if (req.session.loginTime) {
      const loginTime = new Date(req.session.loginTime);
      const currentTime = new Date();
      sessionDuration = Math.round((currentTime - loginTime) / 1000);
    }
    res.render('profile', {
      user: req.session.user,
      sessionDuration: sessionDuration
    });
  } else {
    res.redirect('/auth/login');
  }
});
```

- Rute ini menampilkan halaman profil pengguna. Ia menghitung durasi sesi pengguna dan merender halaman profil dengan informasi pengguna dan durasi sesi.

```
router.get('/logout', (req, res) => {
  req.session.destroy();
  res.redirect('/auth/login');
});
```

- Rute ini menangani proses logout dengan menghancurkan sesi pengguna dan mengarahkan kembali ke halaman login.

```
router.post('/change-password', (req, res) => {
  if (!req.session.user) {
    return res.redirect('/auth/login');
  }
  const { currentPassword, newPassword } = req.body;
  const userId = req.session.user.id;
  const query = "SELECT password FROM users WHERE id = ?";
  db.query(query, [userId], (err, result) => {
    if (err) throw err;
    if (result.length > 0) {
      const user = result[0];
      if (bcrypt.compareSync(currentPassword, user.password)) {
        const hashedNewPassword = bcrypt.hashSync(newPassword, 10);
```

```

-         const updateQuery = "UPDATE users SET password = ? WHERE id = ?";
-         db.query(updateQuery, [hashedNewPassword, userId], (updateErr,
updateResult) => {
-             if (updateErr) throw updateErr;
-             res.send('Password berhasil diubah');
-         });
-         } else {
-             res.send('Password saat ini salah');
-         }
-         } else {
-             res.send('User tidak ditemukan');
-         }
-     });
- });

```

- Rute ini menangani perubahan password pengguna. Ia memeriksa apakah password saat ini benar, dan jika ya, mengupdate password dengan yang baru setelah di-hash.

3. File db.js

```

- const mysql = require('mysql');
- const db = mysql.createConnection({
-     host: 'localhost',
-     user: 'root',
-     password: '',
-     database: 'user_management'
- });
- db.connect((err) => {
-     if (err) {
-         console.error('Error connecting to the database:', err);
-         throw err;
-     }
-     console.log('Database connected!');
- });
- module.exports = db;

```

- File ini mengatur koneksi ke database MySQL. Ia menggunakan modul mysql untuk membuat koneksi dengan parameter yang sesuai (host, user, password, dan nama database). Setelah koneksi berhasil, ia mencetak pesan ke konsol. Objek koneksi db kemudian diekspor untuk digunakan di file lain

4. File login.ejs

```

- <!DOCTYPE html>
- <html lang="en">
- <head>
-     <meta charset="UTF-8">
-     <meta name="viewport" content="width=device-width, initial-scale=1.0">
-     <title>Login</title>
-     <link rel="stylesheet" href="/styleslogin.css">
- </head>
- <body>
-     <div class="container">
-         <h2>Login</h2>
-         <form action="/auth/login" method="POST">
-             <label for="username">Username</label>
-             <input type="text" id="username" name="username" required>
-
-             <label for="password">Password</label>
-             <input type="password" id="password" name="password" required>
-
-             <button type="submit">Login</button>
-         </form>
-     </div>
- </body>
- </html>

```

```

-         </form>
-         <p>Don't have an account? <a href="/auth/register">Register here</a></p>
-     </div>
- </body>
- </html>

```

- File ini adalah template EJS untuk halaman login. Ia memuat stylesheet CSS khusus untuk login dan menampilkan form login dengan field untuk username dan password. Juga terdapat link untuk mendaftar bagi pengguna yang belum memiliki akun.

5. Register.ejs

```

- <!DOCTYPE html>
- <html lang="en">
- <head>
-     <meta charset="UTF-8">
-     <meta name="viewport" content="width=device-width, initial-scale=1.0">
-     <title>Register</title>
-     <link rel="stylesheet" href="/stylesregister.css">
- </head>
- <body>
-     <div class="container">
-         <h2>Register</h2>
-         <form id="registerForm" action="/auth/register" method="POST" onsubmit="return
validateForm()">
-             <label for="username">Username</label>
-             <input type="text" id="username" name="username" required>
-
-             <label for="email">Email</label>
-             <input type="email" id="email" name="email" required>
-
-             <label for="password">Password</label>
-             <input type="password" id="password" name="password" required>
-
-             <button type="submit">Register</button>
-         </form>
-         <p>Already have an account? <a href="/auth/login">Login here</a></p>
-         <p id="errorMessage" style="color:red;"></p>
-     </div>
-
-     <script>
-         function validateForm() {
-             const username = document.getElementById("username").value;
-             const email = document.getElementById("email").value;
-             const password = document.getElementById("password").value;
-             const errorMessage = document.getElementById("errorMessage");
-
-             if (!username || !email || !password) {
-                 errorMessage.textContent = "Please fill in all fields.";
-                 return false;
-             }
-             return true;
-         }
-     </script>
- </body>
- </html>

```

- File ini adalah template EJS untuk halaman pendaftaran. Ia memuat stylesheet CSS khusus untuk pendaftaran dan menampilkan form pendaftaran dengan field untuk username, email, dan password. Terdapat juga validasi form menggunakan JavaScript untuk memastikan semua field terisi sebelum form dikirim.

6. File profile.ejs

```
- <!DOCTYPE html>
- <html lang="en">
- <head>
-   <meta charset="UTF-8">
-   <meta name="viewport" content="width=device-width, initial-scale=1.0">
-   <title>Profile</title>
-   <link rel="stylesheet" href="/stylesprofile.css">
- </head>
- <body>
-   <div class="container">
-     <h2>Welcome, <%= user.username %></h2>
-     <p>Email: <%= user.email %></p>
-     <div class="session-info">
-       <p>Session Duration: <%= sessionDuration %> seconds</p>
-     </div>
-
-     <h3>Change Password</h3>
-     <form action="/auth/change-password" method="POST">
-       <label for="currentPassword">Current Password:</label>
-       <input type="password" id="currentPassword" name="currentPassword"
required>
-
-       <label for="newPassword">New Password:</label>
-       <input type="password" id="newPassword" name="newPassword" required>
-
-       <button type="submit">Change Password</button>
-     </form>
-
-     <form action="/auth/logout" method="GET">
-       <button type="submit">Logout</button>
-     </form>
-   </div>
- </body>
- </html>
```

- File ini adalah template EJS untuk halaman profil pengguna. Ia menampilkan informasi pengguna seperti username dan email, durasi sesi, serta form untuk mengubah password. Juga terdapat tombol logout untuk mengakhiri sesi pengguna.

7. File styleslogin.css

```
- /* Global Styles */
- * {
-   margin: 0;
-   padding: 0;
-   box-sizing: border-box;
- }
-
- body {
-   font-family: 'Arial', sans-serif;
-   background-color: #f5f5f5;
-   display: flex;
-   justify-content: center;
-   align-items: center;
-   min-height: 100vh;
- }
-
- .container {
-   background-color: #fff;
-   padding: 30px;
```

```

-     border-radius: 8px;
-     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
-     max-width: 400px;
-     width: 100%;
-     text-align: center;
- }
-
- h2 {
-     margin-bottom: 20px;
-     font-size: 24px;
-     color: #333;
- }
-
- /* Input Fields */
- input {
-     width: 100%;
-     padding: 10px;
-     margin: 10px 0;
-     border: 1px solid #ddd;
-     border-radius: 5px;
-     font-size: 16px;
-     outline: none;
-     transition: border-color 0.3s ease;
- }
-
- input:focus {
-     border-color: #007bff;
- }
-
- label {
-     display: block;
-     margin-bottom: 5px;
-     font-size: 14px;
-     color: #555;
-     text-align: left;
- }
-
- /* Buttons */
- button {
-     width: 100%;
-     padding: 12px;
-     background-color: #007bff;
-     color: white;
-     border: none;
-     border-radius: 5px;
-     cursor: pointer;
-     font-size: 16px;
-     transition: background-color 0.3s ease;
-     margin-top: 15px;
- }
-
- button:hover {
-     background-color: #0056b3;
- }
-
- /* Links */
- a {
-     color: #007bff;
-     text-decoration: none;
-     font-size: 14px;
- }

```



```

- a:hover {
-     text-decoration: underline;
- }
-
- /* Form Spacing */
- form {
-     margin-top: 20px;
- }
-
- p {
-     margin-top: 20px;
-     color: #666;
- }
-
- /* Responsive Design */
- @media (max-width: 768px) {
-     .container {
-         padding: 20px;
-     }
-
-     h2 {
-         font-size: 22px;
-     }
-
-     input, button {
-         font-size: 14px;
-         padding: 10px;
-     }
- }

```

- File CSS ini mengatur tampilan untuk halaman pendaftaran. Gaya-gaya yang digunakan sangat mirip dengan halaman login, dengan beberapa penyesuaian khusus untuk halaman pendaftaran.

8. File stylesregister.css

```

- /* Global Styles for Register Page */
- * {
-     margin: 0;
-     padding: 0;
-     box-sizing: border-box;
- }
-
- body {
-     font-family: 'Arial', sans-serif;
-     background-color: #f5f5f5;
-     display: flex;
-     justify-content: center;
-     align-items: center;
-     min-height: 100vh;
-     margin: 0;
- }
-
- .container {
-     background-color: #fff;
-     padding: 30px;
-     border-radius: 8px;
-     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
-     max-width: 400px;
-     width: 100%;
-     text-align: center;
- }

```

```

- h2 {
-     margin-bottom: 20px;
-     font-size: 24px;
-     color: #333;
- }
-
- /* Input Fields */
- input {
-     width: 100%;
-     padding: 10px;
-     margin: 10px 0;
-     border: 1px solid #ddd;
-     border-radius: 5px;
-     font-size: 16px;
-     outline: none;
-     transition: border-color 0.3s ease;
- }
-
- input:focus {
-     border-color: #007bff;
- }
-
- label {
-     display: block;
-     margin-bottom: 5px;
-     font-size: 14px;
-     color: #555;
-     text-align: left;
- }
-
- /* Buttons */
- button {
-     width: 100%;
-     padding: 12px;
-     background-color: #007bff;
-     color: white;
-     border: none;
-     border-radius: 5px;
-     cursor: pointer;
-     font-size: 16px;
-     transition: background-color 0.3s ease;
-     margin-top: 15px;
- }
-
- button:hover {
-     background-color: #0056b3;
- }
-
- /* Links */
- a {
-     color: #007bff;
-     text-decoration: none;
-     font-size: 14px;
- }
-
- a:hover {
-     text-decoration: underline;
- }
-
- /* Form Spacing */
- form {

```

```

-     margin-top: 20px;
- }
-
- p {
-     margin-top: 20px;
-     color: #666;
- }
-
- /* Responsive Design for Register Page */
- @media (max-width: 768px) {
-     .container {
-         padding: 20px;
-     }
-
-     h2 {
-         font-size: 22px;
-     }
-
-     input, button {
-         font-size: 14px;
-         padding: 10px;
-     }
- }

```

- File CSS ini mengatur tampilan untuk halaman pendaftaran. Gaya-gaya yang digunakan sangat mirip dengan halaman login, dengan beberapa penyesuaian khusus untuk halaman pendaftaran.

9. File stylesprofile.css

```

- /* Global Styles untuk Halaman Profil */
- * {
-     margin: 0;
-     padding: 0;
-     box-sizing: border-box;
- }
-
- body {
-     font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
-     background-color: #f0f2f5;
-     display: flex;
-     justify-content: center;
-     align-items: center;
-     min-height: 100vh;
-     margin: 0;
-     color: #333;
- }
-
- .container {
-     background-color: #ffffff;
-     padding: 40px;
-     border-radius: 15px;
-     box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);
-     max-width: 500px;
-     width: 100%;
-     text-align: left;
- }
-
- h2 {
-     font-size: 28px;
-     margin-bottom: 20px;
-     color: #1877f2;
- }

```

```

-     text-align: center;
- }
-
- p {
-     font-size: 16px;
-     margin-bottom: 15px;
-     line-height: 1.5;
- }
-
- h3 {
-     font-size: 22px;
-     margin-top: 30px;
-     margin-bottom: 15px;
-     color: #1877f2;
- }
-
- form {
-     margin-top: 20px;
- }
-
- label {
-     display: block;
-     margin-bottom: 5px;
-     font-weight: bold;
- }
-
- input[type="password"] {
-     width: 100%;
-     padding: 12px;
-     margin-bottom: 15px;
-     border: 1px solid #ddd;
-     border-radius: 5px;
-     font-size: 16px;
-     transition: border-color 0.3s ease;
- }
-
- input[type="password"]:focus {
-     border-color: #1877f2;
-     outline: none;
- }
-
- button {
-     background-color: #1877f2;
-     color: white;
-     border: none;
-     padding: 12px 20px;
-     border-radius: 5px;
-     cursor: pointer;
-     font-size: 16px;
-     transition: background-color 0.3s ease;
-     width: 100%;
-     margin-bottom: 15px;
- }
-
- button:hover {
-     background-color: #166fe5;
- }
-
- /* Gaya untuk tombol Logout */
- form:last-child button {
-     background-color: #e4e6eb;
-     color: #050505;

```

```

-   }
-
-   form:last-child button:hover {
-       background-color: #d8dadf;
-   }
-
-   /* Informasi sesi */
-   .session-info {
-       background-color: #e7f3ff;
-       padding: 15px;
-       border-radius: 5px;
-       margin-bottom: 20px;
-       font-size: 14px;
-       color: #1877f2;
-   }
-
-   /* Responsive Design */
-   @media (max-width: 768px) {
-       .container {
-           padding: 30px;
-           margin: 20px;
-       }
-
-       h2 {
-           font-size: 24px;
-       }
-
-       h3 {
-           font-size: 20px;
-       }
-
-       input[type="password"], button {
-           font-size: 14px;
-           padding: 10px;
-       }
-   }

```

- File CSS ini mengatur tampilan untuk halaman profil pengguna. Ia menggunakan skema warna yang berbeda (terinspirasi dari Facebook) dan memiliki gaya yang lebih kompleks untuk menampilkan informasi profil, form perubahan password, dan tombol logout. File ini juga mencakup desain responsif untuk perangkat mobile.
- Penjelasan ini mencakup semua file utama dalam proyek Anda, memberikan gambaran menyeluruh tentang struktur dan fungsi aplikasi web yang Anda buat.

Kesimpulan

Proyek ini mendemonstrasikan implementasi dasar sistem otentikasi dan manajemen pengguna, yang dapat menjadi fondasi untuk aplikasi web yang lebih kompleks. Dengan struktur yang modular dan penggunaan teknologi modern, proyek ini menyediakan kerangka kerja yang solid untuk pengembangan lebih lanjut, seperti penambahan fitur-fitur baru atau peningkatan keamanan.