

Laporan Praktikum
Mata Kuliah Pemrograman Web



Pertemuan 6. Praktikum 6
"SESSION"

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

Disusun Oleh :
Angga Ardiansyah
2307300/A3

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

I. PENDAHULUAN

Session adalah mekanisme yang digunakan dalam aplikasi web untuk menyimpan informasi pengguna secara sementara saat mereka berinteraksi dengan aplikasi tersebut. Dalam konteks Node.js, session digunakan untuk melacak dan mengingat status pengguna di antara permintaan HTTP (request) yang berbeda.

Berikut adalah beberapa poin penting tentang session:

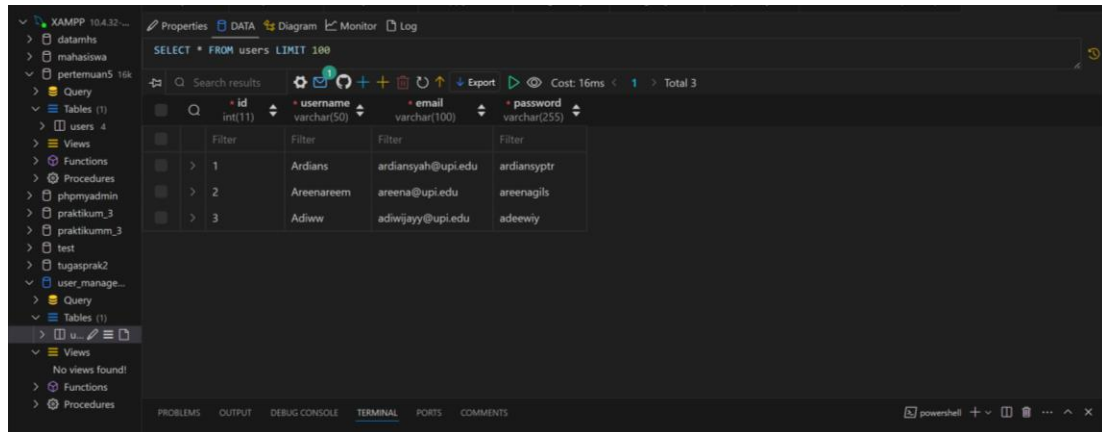
1. **Penyimpanan Informasi Sementara:** Ketika pengguna login, informasi seperti ID pengguna, username, atau token disimpan dalam session, sehingga server dapat mengingat pengguna tersebut selama sesi mereka berlangsung.
2. **Terikat pada Pengguna:** Setiap pengguna yang terhubung ke aplikasi memiliki session unik. Ketika pengguna melakukan request baru (misalnya berpindah halaman), session yang terkait dengannya akan diidentifikasi dan diakses.
3. **Berbasis Cookie:** Session biasanya disimpan di server, dan untuk melacak session tersebut, browser pengguna akan menyimpan cookie yang berisi ID session. Server akan memeriksa cookie ini untuk mengidentifikasi session pengguna.
4. **Contoh Penggunaan:**
 - Ketika pengguna login, aplikasi menyimpan informasi login dalam session.
 - Pada request berikutnya, aplikasi dapat memeriksa session untuk memastikan bahwa pengguna sudah login.
 - Jika pengguna logout, session akan dihapus, sehingga pengguna harus login kembali.

II. ALAT DAN BAHAN

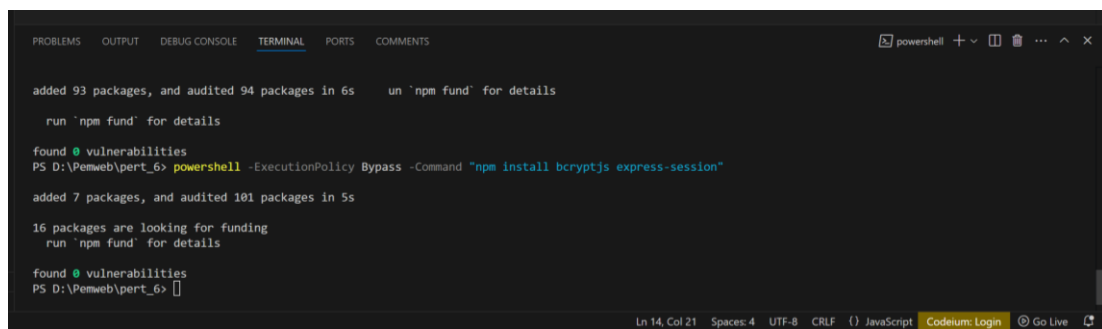
1. Laptop
2. Mouse
3. Visual Code Studio
4. Node js
5. XAMPP
6. Browser

III. LANGKAH KERJA

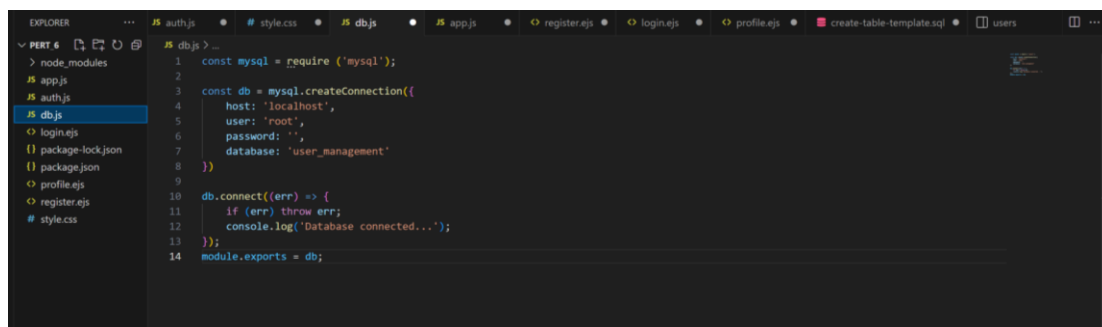
1. Buat database pada mysql.



2. Inisialisasi node js agar terbuat file package.json yang berfungsi sebagai manifest project serta install dependencies pada terminal mengelola instalasi dan update package.



3. Buat file “db.js” untuk mengatur database ke koneksi.



4. Buat file “app.js” untuk menginisialisasi server express dan mengatur middleware.
 - Server express adalah sebuah kerangka kerja (framework) yang menjadi fondasi untuk membangun aplikasi web menggunakan Node.js.
 - Middleware adalah Perangkat lunak yang bertindak sebagai "perantara" antara aplikasi, data, dan pengguna.

```

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
7  const app = express();
8
9  //Set EJS sebagai template engine
10 app.set('view engine', 'ejs');
11
12 // Middleware
13 app.use(bodyParser.json());
14 app.use(bodyParser.urlencoded({ extended: true }));
15 app.use(session({
16   secret: 'secret',
17   resave: false,
18   saveUninitialized: true
19 }));
20
21 // Set static folder
22 app.use(express.static(path.join(__dirname, 'public')));
23
24 // Middleware to check login status
25 app.use((req, res, next) => {
26   if (!req.session.user && req.path !== '/auth/login' && req.path !== '/auth/register') {
27     // If the user is not logged in and trying to access any other page except login/register
28     return res.redirect('/auth/login');
29   } else if (req.session.user && req.path === '/') {
30     // If user is logged in and tries to access the root route, redirect to profile
31     return res.redirect('/auth/profile');
32   }
33   next();
34 });
35
36 // Routes
37 app.use('/auth', authRoutes);
38
39 // Root Route: Redirect to /auth/login or /auth/profile based on session
40 app.length('/', (req, res) => {
41   if (req.session.user) {

```

5. Buat folder views untuk tampilan pada register, login, dan profile.

- Register

```

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>Register</title>
7  </head>
8  <body>
9    <div class="container">
10     <h2>Register</h2>
11     <form action="/auth/register" method="POST">
12       <label for="username">Username</label>
13       <input type="text" id="username" name="username" required>
14
15       <label for="email">Email</label>
16       <input type="email" id="email" name="email" required>
17
18       <label for="password">Password</label>
19       <input type="password" id="password" name="password" required>
20
21       <button type="submit">Register</button>
22     </form>
23     <p>Already have an account? <a href="/auth/login">Login here</a></p>
24   </div>
25 </body>
26 </html>

```

- 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  </head>
8  <body>
9    <div class="container">
10     <h2>Login</h2>
11     <form action="/auth/login" method="POST">
12       <label for="username">Username</label>
13       <input type="text" id="username" name="username" required>
14
15       <label for="email">Email</label>
16       <input type="email" id="email" name="email" required>
17
18       <label for="password">Password</label>
19       <input type="password" id="password" name="password" required>
20
21       <button type="submit">Register</button>
22     </form>
23     <p>Already have an account? <a href="/auth/login">Login here</a></p>
24   </div>
25 </body>
26 </html>

```

- 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 </head>
8 <body>
9   <div class="container">
10    <h2>Welcome, <%= user.username %></h2>
11    <p>Email: <%= user.email %></p>
12    <a href="/auth/login"></a>
13  </div>
14 </body>
15 </html>

```

6. Buat rute login, logout, dan regiter. File “auth.js” pada folder “routes” berfungsi untuk mengelola autentifikasi pengguna.

```

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   const hashedPassword = bcrypt.hashSync(password, 10);
16
17   const query = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";
18   db.query(query, [username, email, hashedPassword], (err, result) => {
19     if (err) throw err;
20     res.redirect('/auth/login');
21   });
22 });
23
24 // Render halaman login
25 router.get('/login', (req, res) => {
26   res.render('login');
27 });
28
29 // Proses login use
30 router.post('/login', (req, res) => {
31   const { username, password } = req.body;
32
33   const query = "SELECT * FROM users WHERE username = ?";
34   db.query(query, [username], (err, result) => {
35     if (err) throw err;
36     if (result.length > 0) {
37       const user = result[0];
38
39       if (bcrypt.compareSync(password, user.password)) {
40         req.session.user = user;
41         res.redirect('/auth/profile');
42       }
43     }
44   });
45 });

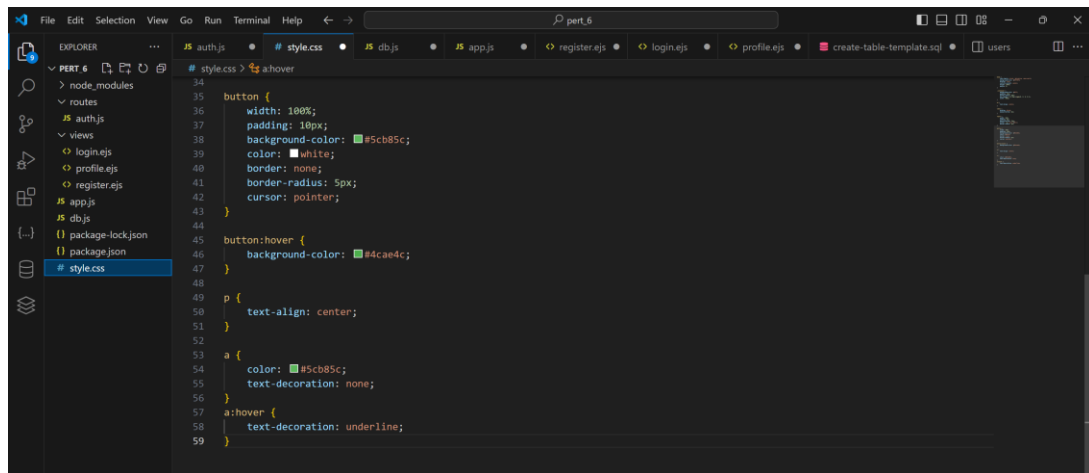
```

7. Buat CSS agar tampilan menjadi lebih menarik.

```

1 body {
2   font-family: Arial, Helvetica, sans-serif;
3   background-color: #f0f0f0;
4   display: flex;
5   justify-content: center;
6   height: 100vh;
7   margin: 0;
8 }
9
10 .container {
11   background-color: #fff;
12   padding: 20px;
13   border-radius: 10px;
14   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
15   width: 300px;
16 }
17
18 h2 {
19   text-align: center;
20 }
21
22 label {
23   display: block;
24   margin-bottom: 5px;
25 }
26
27 input {
28   width: 100%;
29   padding: 8px;
30   margin-bottom: 15px;
31   border: 1px solid #ccc;
32   border-radius: 5px;
33 }

```



IV. KESIMPULAN

Melalui praktikum ini mahasiswa dapat menambah wawasan tentang session, bahwa mekanisme ini dapat menyimpan informasi pengguna secara sementara saat mereka berinteraksi dengan aplikasi tersebut. Alur mekanisme ini dimulai dari proses registrasi, kita mempelajari bagaimana data pengguna baru diolah dan disimpan dengan aman. Selanjutnya ke proses login, kita melihat bagaimana session bekerja sebagai penjaga pintu virtual. Kemudian yang terakhir yakni logout, kita mempelajari pentingnya penghapusan session yang bersih dan aman. Selain itu, praktikum juga mengajarkan kita tentang penanganan error dan validasi input.