

**Laporan Praktikum**  
**Mata Kuliah Pemrograman Web**



**Pertemuan 5**

**“CRUD**

**Session”**

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

**Disusun Oleh :**

Hilwa Nur Falah

(2312509)

**PROGRAM STUDI SISTEM INFORMASI KELAUTAN**  
**UNIVERSITAS PENDIDIKAN INDONESIA**  
**2024**

## **I. PENDAHULUAN**

Dalam pengembangan aplikasi web, manajemen data sangat penting, terutama untuk aplikasi yang memerlukan operasi CRUD (Create, Read, Update, Delete). CRUD adalah konsep dasar dari aplikasi berbasis database yang memungkinkan pengguna untuk menambah, melihat, memperbarui, dan menghapus data. Menurut Gonzalez & Smith (2018), CRUD merupakan prinsip yang mendasar dalam interaksi antara sistem informasi dan database karena memastikan bahwa data dapat dikelola dengan efisien. Selain itu, implementasi session pada aplikasi web membantu menjaga status pengguna selama interaksi mereka dengan aplikasi, seperti yang diuraikan oleh Kim (2020), session memungkinkan autentikasi pengguna dan pelacakan aktivitas secara aman.

## **II. TUJUAN**

1. Memahami cara implementasi CRUD.
2. Mempelajari cara membuat aplikasi web yang mendukung operasi CRUD menggunakan Node.js dan Express.
3. Mengimplementasikan sistem session untuk autentikasi dan pelacakan pengguna.
4. Memahami cara menghubungkan aplikasi dengan database untuk mengelola data pengguna secara dinamis..

## **III. ALAT DAN BAHAN**

- Laptop.
- Node.js dan npm
- MySQL sebagai basis data
- Visual Studio Code
- Koneksi internet untuk menjalankan localhost.

### **Framework:**

- Express.js dan Express-session
- Bcrypt dan Mysql
- Body-parser

## IV. LANGKAH KERJA

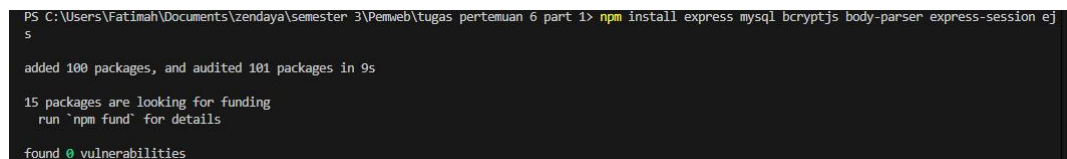
1. Install Node.js dan MySQL pada perangkat.
2. Buat direktori proyek baru dan inisialisasi dengan *npm init-y*.



```
PS C:\Users\Fatimah\Documents\zendaya\semester 3\Pemweb\tugas pertemuan 6 part 1> npm init -y
Wrote to C:\Users\Fatimah\Documents\zendaya\semester 3\Pemweb\tugas pertemuan 6 part 1\package.json:

{
  "name": "tugas-pertemuan-6-part-1",
  "version": "1.0.0",
  "main": "app.js",
  "scripts": {
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

3. Instal modul yang dibutuhkan: *npm install express bcryptjs express-session mysql2 body-parser*.

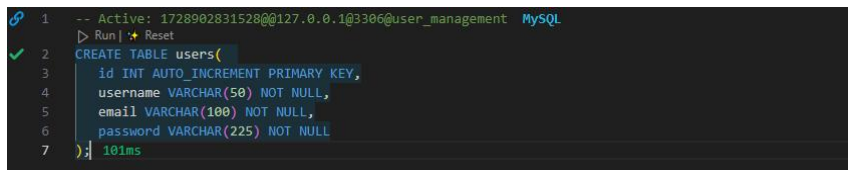


```
PS C:\Users\Fatimah\Documents\zendaya\semester 3\Pemweb\tugas pertemuan 6 part 1> npm install express mysql bcryptjs body-parser express-session ej s
added 100 packages, and audited 101 packages in 9s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

4. Buat database baru di MySQL dan buat tabel *users*..



```
-- Active: 1728902831528@@@127.0.0.1@3306@user_management MySQL
> Run | ⚙ Reset
CREATE TABLE users(
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL,
  email VARCHAR(100) NOT NULL,
  password VARCHAR(225) NOT NULL
);
101ms
```

5. Membuat CRUD di *auth.js*. Membuat rute untuk register, login, profile, dan logout seperti yang sudah diimplementasikan sebelumnya.

```
7 router.get('/register', (req, res) => {
8   res.render('register');
9 });
10
11 router.post('/register', (req, res) => {
12   const { username, email, password } = req.body;
13
14   const hashedPassword = bcrypt.hashSync(password, 10);
15
16   const query = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";
17   db.query(query, [username, email, hashedPassword], (err, result) => {
18     if (err) throw err;
19     res.redirect('/auth/login');
20   });
21 });
22
23 router.get('/login', (req, res) => {
24   res.render('login');
25 });
26
27 router.post('/login', (req, res) => {
28   const { username, password } = req.body;
29
30   const query = "SELECT * FROM users WHERE username = ?";
31   db.query(query, [username], (err, result) => {
32     if (err) throw err;
33     if (result.length > 0) {
34       const user = result[0];
35       if (bcrypt.compareSync(password, user.password)) {
36         req.session.user = user;
37         res.redirect('/auth/profile');
38       } else {
39         // ...
40       }
41     }
42   });
43 });
```

6. Ikuti langkah seperti pada modul lalu jalankan dengan perintah *node app.js*.

## V. ANALISIS

Selama proses pengembangan dan praktikum yang mengacu pada modul Session, beberapa tantangan yang dihadapi termasuk kesalahan dalam koneksi database dan error dalam manajemen session. Solusi yang diterapkan antara lain memastikan bahwa modul yang dibutuhkan telah terinstal dengan benar dan jalur file dalam proyek sudah benar. Selain itu, perlu diperhatikan pentingnya enkripsi kata sandi menggunakan *bcryptjs* untuk memastikan keamanan data pengguna.

## VI. KESIMPULAN

Pembuatan aplikasi CRUD dengan session berhasil dilakukan dengan menggunakan Node.js dan Express. Implementasi session memungkinkan aplikasi untuk mengenali pengguna dan mengatur hak akses sesuai dengan status login mereka. Penggunaan enkripsi untuk kata sandi menambah lapisan keamanan dalam sistem. Dengan memahami dasar-dasar ini, pengembang dapat membangun aplikasi yang lebih kompleks dan aman di masa mendatang.