

Jurnal Modul 15

Aplikasi desktop ini dibangun menggunakan **Electron**, sebuah framework yang memungkinkan pembuatan aplikasi desktop berbasis web dengan JavaScript, HTML, dan CSS. Aplikasi ini memiliki dua fitur utama yaitu **registrasi user** dan **login user**, dengan data yang disimpan secara lokal dalam file **JSON**. Untuk menjalankan aplikasi, pengguna perlu memastikan telah menginstal **Node.js** dan **Electron** melalui terminal atau command prompt. Setelah itu, dengan berada di direktori project, perintah `npm install` digunakan untuk mengunduh dependensi, diikuti `npm start` untuk menjalankan aplikasi. Saat aplikasi dibuka, pengguna disajikan antarmuka sederhana yang berisi form registrasi dan login. Pada saat registrasi, sistem akan melakukan validasi terhadap input password agar memenuhi kriteria panjang minimal 8 karakter dan maksimal 20 karakter, mengandung minimal satu angka, satu simbol khusus seperti `!@#$%^&*`, serta tidak mengandung kata dari username. Jika validasi lolos, password di-hash menggunakan algoritma **SHA256** sebelum disimpan ke file `userData.json` agar aman. Saat login, aplikasi mencocokkan hash dari password yang dimasukkan dengan hash yang tersimpan, dan menampilkan pesan berhasil atau gagal sesuai hasilnya.

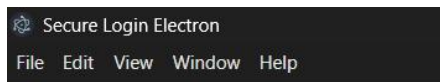
Secara struktur, file utama aplikasi terdiri dari `main.js` yang mengatur window Electron, `index.html` sebagai antarmuka, `renderer.js` untuk menangani interaksi form dan logika validasi, serta `userData.json` sebagai penyimpanan data user. Aplikasi ini dapat dikembangkan lebih lanjut dengan berbagai fitur tambahan, seperti sistem **session login**, **encrypt file JSON** agar lebih aman, **pengaturan reset password**, serta integrasi ke database lokal seperti **SQLite** atau database online jika ingin dihubungkan dengan server. Selain itu, antarmuka dapat diperkaya menggunakan framework UI seperti **Bootstrap** atau **Tailwind CSS**, dan fitur notifikasi login berhasil atau gagal bisa ditambahkan melalui modul Electron seperti **dialog** atau **notification API**. Dengan menerapkan prinsip **Secure Coding Practices**, aplikasi ini tidak hanya aman digunakan, tetapi juga fleksibel untuk dikembangkan menjadi sistem login desktop yang lebih kompleks ke depannya.

✅ Secure Coding yang Dipenuhi:

Faktor	Implementasi
Input Validation	Regex validasi username & password, panjang karakter
Handling Invalid Data	Pesan error spesifik tampil di UI
Password Hashing	SHA-256 via Web Crypto API
Password Rules	Minimal 8 karakter, 1 angka, 1 simbol, tidak mengandung username
Secure Bridge	Pakai <code>preload.js</code> + <code>contextBridge</code> untuk keamanan komunikasi renderer-main

Outputnya running kodingan memakai register & login serta file user.json:

Tampilan default



Registrasi User

<input type="text"/>
<input type="password"/>
<input type="button" value="Register"/>

Login User

<input type="text"/>
<input type="password"/>
<input type="button" value="Login"/>

Tampilan saat registrasi berhasil



Registrasi User


<input type="text" value="Hisyam"/>
<input type="password" value="....."/>
<input type="button" value="Register"/>

Login User

<input type="text"/>
<input type="password"/>
<input type="button" value="Login"/>

Registrasi berhasil!

Tampilan saat registrasi gagal

 Secure Login Electron
File Edit View Window Help

Registrasi User

Login User

Password harus 8-20 karakter, minimal 1 angka dan 1 simbol (!@#\$%^&*).

Tampilan saat login berhasil

 Secure Login Electron
File Edit View Window Help

Registrasi User

Login User

Login berhasil!

Tampilan saat login gagal

 Secure Login Electron
File Edit View Window Help

Registrasi User

Login User

Username atau password salah.

Tampilan saat hasil json telah teregistrasi

```
1 {  
2   "users": [  
3     {  
4       "username": "Hisyam",  
5       "password": "e66860546f18cdbbcd86b35e18b525bffc67f772c650cedfe3ff7a0026fa1dee"  
6     }  
7   ]  
8 }
```