

Kelvin Ferdinan_2311104009

https://github.com/mispipin/KPL_Kelvin-Ferdinan_2311104009_s1se0701

index.html

```
15_Desktop_App > index.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8" />
5      <title>Simple Auth App</title>
6      <style>
7          body { font-family: sans-serif; padding: 20px; max-width: 400px; margin: auto; }
8          input, button { display: block; width: 100%; margin: 10px 0; padding: 8px; }
9          .status { margin-top: 15px; color: red; font-weight: bold; }
10     </style>
11 </head>
12 <body>
13     <h2>Register</h2>
14     <input type="text" id="reg-username" placeholder="Username" />
15     <input type="password" id="reg-password" placeholder="Password" />
16     <button onclick="register()">Register</button>
17
18     <h2>Login</h2>
19     <input type="text" id="log-username" placeholder="Username" />
20     <input type="password" id="log-password" placeholder="Password" />
21     <button onclick="login()">Login</button>
22
23     <div class="status" id="status"></div>
24
25     <script src="app.js"></script>
26 </body>
27 </html>
28
```

App.js

```
15_Desktop_App > appjs > ...
1  function hashPassword(password) {
2      return btoa(password); // NOT secure, hanya base64 (untuk demo)
3  }
4
5  function validate(username, password) {
6      const asciiOnly = /^[A-Za-z]+$/;
7      const specialChar = /[!@#%&*]/;
8      const digit = /\d/;
9      const errors = [];
10
11     if (!asciiOnly.test(username)) errors.push("Username hanya huruf A-Z");
12     if (username.length < 4 || username.length > 20) errors.push("Username harus 4-20 karakter");
13     if (password.length < 8 || password.length > 20) errors.push("Password harus 8-20 karakter");
14     if (!specialChar.test(password)) errors.push("Password perlu karakter unik (!@#%&*");
15     if (!digit.test(password)) errors.push("Password harus mengandung angka");
16     if (password.toLowerCase().includes(username.toLowerCase())) errors.push("Password tidak boleh mengandung username");
17
18     return errors;
19 }
20
21 function register() {
22     const username = document.getElementById("reg-username").value.trim();
23     const password = document.getElementById("reg-password").value.trim();
24     const users = JSON.parse(localStorage.getItem("users") || "[]");
25
26     const errors = validate(username, password);
27     if (errors.length > 0) return showStatus(errors.join("\n"));
28
29     if (users.find(u => u.username === username)) {
30         return showStatus("Username sudah digunakan");
31     }
32
33     users.push({ username, password: hashPassword(password) });
34     localStorage.setItem("users", JSON.stringify(users));
35     showStatus("Registrasi berhasil!", "green");
36 }
37
38 function login() {
39     const username = document.getElementById("log-username").value.trim();
40     const password = document.getElementById("log-password").value.trim();
41     const users = JSON.parse(localStorage.getItem("users") || "[]");
42
43     const user = users.find(u => u.username === username);
44     if (!user) return showStatus("Username tidak ditemukan");
45     if (user.password !== hashPassword(password)) return showStatus("Password salah");
46
47     showStatus("Login berhasil!", "green");
48 }
49
50 function showStatus(message, color = "red") {
51     const status = document.getElementById("status");
52     status.innerText = message;
53     status.style.color = color;
54 }
55
56
```