

JURNAL MODUL 15

2311104008

Viona Aziz Syahputri

Link Github

https://github.com/viona123/KPL_Viona-Aziz-Syahputri_2311104008_SE07-01/tree/main/15_Review_Tugas_Besar/Jurnal_GUI_2311104008

login.js

```
1  const fs = require("fs");
2  const readline = require("readline-sync");
3  const { hashPassword } = require("./utils");
4
5  const filePath = "users.json";
6
7  if (!fs.existsSync(filePath)) {
8      console.log("Belum ada data user. Registrasi dulu.");
9      process.exit(1);
10 }
11
12 const users = JSON.parse(fs.readFileSync(filePath));
13 const username = readline.question("Username: ");
14 const password = readline.question("Password: ", { hideEchoBack: true });
15
16 const hashed = hashPassword(password);
17 const match = users.find(user => user.username === username && user.password === hashed);
18
19 if (match) {
20     console.log("Login berhasil!");
21 } else {
22     console.log("Login gagal. Username atau password salah.");
23 }
```

Pertama-tama, dia mengecek dulu apakah file yang isinya data pengguna (users.json) itu udah ada atau belum. Kalau belum ada, berarti belum ada yang daftar, jadi sistem nyuruh kamu daftar dulu terus langsung keluar. Kalau file-nya ada, sistem bakal baca semua data user yang udah tersimpan. Lalu kamu disuruh masukin username dan password. Nah, password yang kamu ketik itu nggak langsung dipakai, tapi diacak dulu biar aman (disebut *hashing*). Terus sistem cek, ada nggak user yang username dan password-nya cocok. Kalau ada, berarti login sukses. Tapi kalau nggak ketemu, berarti gagal login karena salah username atau password.

register.js

```

1  const fs = require("fs");
2  const readline = require("readline-sync");
3  const { hashPassword, isValidUsername, isValidPassword } = require("./utils");
4
5  let users = [];
6  const filePath = "users.json";
7
8  // Baca data lama
9  if (fs.existsSync(filePath)) {
10     users = JSON.parse(fs.readFileSync(filePath));
11 }
12 // Input username dan password
13 const username = readline.question("Masukkan username: ");
14 const password = readline.questionNewPassword("Masukkan password: ", {
15     min: 8,
16     confirmMessage: "Konfirmasi password: ",
17 });
18
19 if (!isValidUsername(username)) {
20     console.log("Username harus 5-20 karakter alfanumerik.");
21     process.exit(1);
22 }
23 if (!isValidPassword(password, username)) {
24     console.log("Password harus 8-20 karakter, mengandung huruf besar, angka, dan simbol.");
25     process.exit(1);
26 }
27
28 if (users.some(user => user.username === username)) {
29     console.log("Username sudah digunakan.");
30     process.exit(1);
31 }
32
33 users.push({ username, password: hashPassword(password) });
34 fs.writeFileSync(filePath, JSON.stringify(users, null, 2));
35
36 console.log("Registrasi berhasil!");

```

Kode ini fungsinya buat daftar akun baru. Pertama, dia coba baca dulu data user lama dari file `users.json`. Kalau ada, dibuka, kalau belum ada ya kosongin dulu. Terus kamu bakal diminta buat masukin username dan password, sekalian konfirmasi password juga. Tapi nggak sembarangan, username-nya harus 5–20 karakter dan cuma boleh huruf atau angka. Password-nya juga harus kuat: panjang 8–20 karakter, ada huruf besar, angka, simbol, dan nggak boleh ada nama username di dalamnya. Kalau ada yang salah, sistem bakal ngasih tahu dan keluar. Kalau semuanya oke dan username belum dipakai orang lain, data kamu bakal disimpan ke file dengan password yang udah diamankan (di-*hash*). Setelah itu, muncul pesan kalau kamu berhasil daftar.

utils.js

```

1  const crypto = require("crypto");
2
3  // Hash password
4  function hashPassword(password) {
5      return crypto.createHash("sha256").update(password).digest("hex");
6  }
7
8  // Validasi username
9  function isValidUsername(username) {
10     const regex = /^[a-zA-Z0-9]{5,20}$/;
11     return regex.test(username);
12 }
13
14 // Validasi password
15 function isValidPassword(password, username) {
16     const regex = /^(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$%^&*]).{8,20}$/;
17     return regex.test(password) && !password.includes(username);
18 }
19
20 module.exports = { hashPassword, isValidUsername, isValidPassword };
21

```

Kode ini isinya kumpulan alat bantu buat login dan registrasi. Yang pertama, ada fungsi buat ngacak password jadi kode unik biar nggak bisa ditebak orang (ini namanya *hashing*). Lalu ada pengecekan buat username, harus antara 5 sampai 20 karakter dan cuma boleh huruf sama angka. Terus ada juga pengecekan password supaya nggak asal-asalan. Password-nya harus cukup panjang, ada huruf besar, angka, dan simbol. Dan yang penting, nggak boleh mengandung username, supaya lebih aman dan nggak gampang dibobol.

users.json

```

1  [
2    {
3      "username": "viona",
4      "password": "8ff8724ef96c8864a89c41b8563f2710b88c28df4c537e19c04d89b5521a4b5f"
5    }
6  ]

```

Setiap user punya username dan password. Tapi tenang, password-nya nggak disimpan dalam bentuk asli, melainkan udah diacak jadi kode panjang (biar aman). Misalnya, user namanya "viona", password-nya nggak kelihatan kayak "1234", tapi udah berubah jadi kumpulan huruf dan angka acak. File ini dipakai buat cek pas login, atau buat mastiin username-nya belum dipakai orang lain pas daftar.

Output

```

Masukkan username: Mimiw
Masukkan password: *****
Konfirmasi password: *****
Registrasi berhasil!

```

Username: Mimiw

Password: ****

Login berhasil!