

Baris ini mendefinisikan fungsi untuk mengamankan dan memvalidasi input pengguna dalam aplikasi registrasi. Fungsi `hashPassword` menggunakan algoritma SHA-256 untuk mengubah password menjadi bentuk hash yang aman disimpan, sedangkan fungsi `validateInput` melakukan pengecekan pada username dan password; username hanya boleh berisi huruf alfabet ASCII, password harus memiliki panjang antara 8 sampai 20 karakter, mengandung minimal satu karakter khusus (!@#\$\$%^&*), dan tidak boleh mengandung bagian dari username. Jika semua aturan terpenuhi, fungsi validasi mengembalikan `null` yang berarti input valid, sebaliknya mengembalikan pesan kesalahan spesifik.

```
JS app.js U X
15_Review_Tugas_Besar > JURNAL_2311104013 > JS app.js > ...
1  const fs = require("fs");
2  const readline = require("readline");
3  const crypto = require("crypto");
4
5  // JSON file untuk menyimpan user
6  const USERS_FILE = "users.json";
7
8  // Fungsi untuk hashing password
9  function hashPassword(password) {
10     return crypto.createHash("sha256").update(password).digest("hex");
11 }
12
13 // Fungsi untuk validasi input
14 function validateInput(username, password) {
15     const asciiRegex = /^[A-Za-z]+$/;
16     const specialCharRegex = /[!@#$$%^&*]/;
17
18     if (!asciiRegex.test(username)) {
19         return "Username hanya boleh huruf alfabet ASCII";
20     }
21
22     if (password.length < 8 || password.length > 20) {
23         return "Password harus 8-20 karakter";
24     }
25
26     if (!specialCharRegex.test(password)) {
27         return "Password harus mengandung minimal 1 karakter unik (!@#$$%^&*)";
28     }
29
30     if (password.toLowerCase().includes(username.toLowerCase())) {
31         return "Password tidak boleh mengandung username";
32     }
33
34     return null; // Jika Valid
35 }
36
37 // Fungsi untuk load users
38 function loadUsers() {
```

Pada baris di atas berisi beberapa fungsi utama untuk mengelola data user dan interaksi dengan pengguna lewat command line. Fungsi loadUsers membaca file JSON berisi daftar user, mengembalikan array kosong jika file belum ada. Fungsi saveUser menambahkan user baru ke data yang sudah ada lalu menyimpannya kembali ke file. Fungsi login mencari user yang cocok berdasarkan username dan password yang sudah di-hash. Bagian CLI menggunakan modul readline untuk membuat menu interaktif, meminta input pengguna untuk memilih antara registrasi atau login, dan memanggil fungsi sesuai pilihan tersebut, dengan validasi sederhana untuk pilihan yang tidak valid.

```
JS app.js U X
15_Review_Tugas_Besar > JURNAL_2311104013 > JS app.js > ...
36
37 // Fungsi untuk load users
38 function loadUsers() {
39   if (!fs.existsSync(USERS_FILE)) return [];
40   const data = fs.readFileSync(USERS_FILE);
41   return JSON.parse(data);
42 }
43
44 // Fungsi untuk simpan user
45 function saveUser(username, hashedPassword) {
46   const users = loadUsers();
47   users.push({ username, password: hashedPassword });
48   fs.writeFileSync(USERS_FILE, JSON.stringify(users, null, 2));
49 }
50
51 // Fungsi untuk login
52 function login(username, password) {
53   const users = loadUsers();
54   const hashed = hashPassword(password);
55   return users.find((user) => user.username === username && user.password === hashed);
56 }
57
58 // CLI interface
59 const rl = readline.createInterface({
60   input: process.stdin,
61   output: process.stdout,
62 });
63
64 // Menu
65 function showMenu() {
66   console.log("1. Register");
67   console.log("2. Login");
68   rl.question("Pilih menu (1/2): ", (choice) => {
69     if (choice === "1") {
70       register();
71     } else if (choice === "2") {
72       loginUser();
73     } else {
```

Ln 51, Col 17 Spa

Baris ini mengimplementasikan dua fungsi utama untuk proses interaksi pengguna: register dan loginUser. Fungsi register meminta input username dan password, lalu memvalidasi input tersebut menggunakan validateInput. Jika validasi gagal, program menampilkan pesan error dan berhenti. Jika berhasil, password di-hash dan data user disimpan ke file, kemudian menampilkan pesan sukses. Fungsi loginUser meminta username dan password, kemudian memeriksa kecocokan data melalui fungsi login. Jika cocok, menampilkan pesan selamat datang, jika tidak, menampilkan pesan gagal login. Setelah itu, kedua fungsi menutup interface CLI, dan program memulai dengan memanggil showMenu untuk menampilkan pilihan registrasi atau login.

```
JS app.js U X
15_Review_Tugas_Besar > JURNAL_2311104013 > JS app.js > ...
65 function showMenu() {
68   rl.question("Pilih menu (1/2): ", (choice) => {
72     loginUser();
73   } else {
74     console.log("Pilihan tidak valid.");
75     rl.close();
76   }
77 });
78 }
79
80 // Register
81 function register() {
82   rl.question("Username: ", (username) => {
83     rl.question("Password: ", (password) => {
84       const validationError = validateInput(username, password);
85       if (validationError) {
86         console.log("Error: " + validationError);
87         rl.close();
88         return;
89       }
90
91       const hashed = hashPassword(password);
92       saveUser(username, hashed);
93       console.log("Registrasi berhasil!");
94       rl.close();
95     });
96   });
97 }
98
99 // Login
100 function loginUser() {
101   rl.question("Username: ", (username) => {
102     rl.question("Password: ", (password) => {
103       const result = login(username, password);
104       if (result) {
105         console.log("Login berhasil. Selamat datang, " + username + "!");
106       } else {
107         console.log("Login gagal. Username atau password salah.");
108       }
109       rl.close();
110     });
111   });
112 }
113
114 showMenu();
115
```

Output:

```
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwijanto>
RNAL_2311104013> node app.js
1. Register
2. Login
Pilih menu (1/2): 1
Username: hafizh
Password: 12345
Error: Password harus 8-20 karakter
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwijanto>
RNAL_2311104013> 
```

```
RNAL_2311104013> node app.js
1. Register
2. Login
Pilih menu (1/2): 1
Username: hafizh
Password: tanj12345
Error: Password harus mengandung minimal 1 karakter unik (!@#$$%^&*)
PS E:\KULIAH\Semester 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Hafizh Dwijanto>
RNAL_2311104013> 
```

```
RNAL_2311104013> node app.js
1. Register
2. Login
Pilih menu (1/2): 1
Username: hafizh
Password: tanju1234!
Registrasi berhasil!
PS E:\KULIAH\Semester 4\Konstruksi
RNAL_2311104013> 
```

```
JS app.js U {} users.json U X
15_Review_Tugas_Besar > JURNAL_2311104013 > {} users.json > ...
1 1
2 {
3   "username": "hafizh",
4   "password": "804bc1c905cc37ce38ff3d8c9029f01014854f9e6c9e3b9353eb261d452e2a68"
5 }
6 
```