

App.js

```
15_Module > JS app.js > ...
1  import readline from 'readline';
2  import { loadUsers, saveUsers } from './data.js';
3
4  const rl = readline.createInterface({
5    input: process.stdin,
6    output: process.stdout
7  });
8
9  function validateUsername(username) {
10   const asciiRegex = /^[A-Za-z]+$/;
11   return asciiRegex.test(username) && username.length >= 3 && username.length <= 20;
12 }
13
14 function validatePassword(password, username) {
15   const hasNumber = /[0-9]/.test(password);
16   const hasSymbol = /[!@#$$%^&*]/.test(password);
17   const isLongEnough = password.length >= 8 && password.length <= 20;
18   const containsUsername = password.toLowerCase().includes(username.toLowerCase());
19   return hasNumber && hasSymbol && isLongEnough && !containsUsername;
20 }
21
22 function register() {
23   rl.question('Masukkan username: ', (username) => {
24     if (!validateUsername(username)) {
25       console.log('Username harus 3-20 karakter alfabet ASCII.');
26       return rl.close();
27     }
28   });
29
30   const users = loadUsers();
31   if (users.find(u => u.username === username)) {
32     console.log('Username sudah terdaftar.');
33     return rl.close();
34   }
35
36   rl.question('Masukkan password: ', (password) => {
37     if (!validatePassword(password, username)) {
38       console.log('Password harus 8-20 karakter, mengandung angka, simbol (!@#$$%^&*),');
39       return rl.close();
40     }
41
42     users.push({ username, password });
43     saveUsers(users);
44     console.log('Registrasi berhasil.');
45     rl.close();
46   });
47 });
48
49 function login() {
50   rl.question('Username: ', (username) => {
51     rl.question('Password: ', (password) => {
52       const users = loadUsers();
53       const user = users.find(u => u.username === username && u.password === password);
54     });
55   });
56 }
```

```

56     if (user) {
57         console.log(`Login berhasil. Selamat datang, ${username}.`);
58     } else {
59         console.log('Login gagal. Username atau password salah.');
```

```

60     }
61     rl.close();
62 });
63 });
64 }
65
66 rl.question('Ketik "register" untuk registrasi atau "login" untuk masuk: ', (choice) => {
67     if (choice === 'register') {
68         register();
69     } else if (choice === 'login') {
70         login();
71     } else {
72         console.log('Perintah tidak dikenali.');
```

```

73         rl.close();
74     }
75 });

```

Kode di atas adalah aplikasi desktop berbasis Node.js yang berjalan di terminal dan menyediakan fitur registrasi serta login pengguna menggunakan input username dan password. Data pengguna disimpan dalam file JSON melalui fungsi `loadUsers` dan `saveUsers`. Validasi keamanan diterapkan pada input, termasuk pemeriksaan panjang karakter dan format ASCII untuk username, serta aturan kompleksitas password seperti minimal panjang, mengandung angka, simbol khusus, dan tidak mengandung bagian dari username.

Saat program dijalankan, pengguna diminta memilih antara registrasi atau login. Untuk registrasi, sistem memastikan username belum digunakan dan password memenuhi syarat keamanan sebelum menyimpan data ke file. Untuk login, program mencocokkan username dan password yang dimasukkan dengan data yang telah tersimpan.

Data.js

```

15_Module > JS datajs > ...
1  import fs from 'fs';
2
3  const FILE_PATH = './users.json';
4
5  export function loadUsers() {
6      try {
7          const data = fs.readFileSync(FILE_PATH, 'utf-8');
8          return JSON.parse(data);
9      } catch (err) {
10         return [];
11     }
12 }
13
14 export function saveUsers(users) {
15     fs.writeFileSync(FILE_PATH, JSON.stringify(users, null, 2));
16 }

```

Kode di atas berfungsi untuk membaca dan menyimpan data pengguna dalam format JSON ke dalam file `users.json`. Fungsi `loadUsers` mencoba membaca isi file tersebut dan mengembalikan data yang sudah diparsing menjadi objek JavaScript; jika file tidak ditemukan atau terjadi error, fungsi ini mengembalikan array kosong sebagai fallback. Sedangkan fungsi `saveUsers` menerima data pengguna dalam bentuk array atau objek, lalu menulisnya kembali ke file `users.json`.

Output :

```
PS D:\Campuss\SMT 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Andika Rifki Pratama_231104011_SE0701\15_Module> node app.js
Ketik "register" untuk registrasi atau "login" untuk masuk: register
Masukkan username: pratama
Masukkan password: andika1234!
Registrasi berhasil.

PS D:\Campuss\SMT 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Andika Rifki Pratama_231104011_SE0701\15_Module> node app.js
Ketik "register" untuk registrasi atau "login" untuk masuk: login
Username: pratama
Password: andika1234!
Login berhasil. Selamat datang, pratama.
PS D:\Campuss\SMT 4\Konstruksi Perangkat Lunak\Praktikum\KPL_Andika Rifki Pratama_231104011_SE0701\15_Module>
```