

Perancangan Sistem *Authentication* pada Website Monitoring Suhu Mesin *Ball Tea*

1st Yeni Octaviani
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

yenioctaviani@student.telkomuniversit
y.ac.id

2nd Asep Suhendi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

suhendi@telkomuniversity.ac.id

3rd Nurwulan Fitriyanti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

nurwulanf@telkomuniversity.ac.id

Abstrak — Di Indonesia, teh menjadi salah satu minuman yang sering dikonsumsi oleh masyarakat Indonesia. Pada salah satu tahap pengolahan daun teh, yaitu pengeringan akhir menggunakan mesin ball tea masih dilakukan pemantauan suhu secara manual dengan termometer analog sehingga menyebabkan ketidakefisienan dan keterlambatan untuk mendeteksi masalah. Untuk mengatasi masalah ini, kami merancang sistem yang dapat memantau suhu pada mesin ball tea menggunakan sensor RTD (*Resistance Temperature Detector*) PT-100. Hasil monitoring suhu akan ditampilkan pada website dalam bentuk data dan grafik. Dengan memanfaatkan *firebase authentication*, sistem menyediakan login yang aman pada website. *Firestore* digunakan untuk menyimpan data suhu secara terenkripsi, memberikan lapisan keamanan yang kuat pada informasi yang dihasilkan. Pada hasil perancangan dan pengujian yang sudah dilakukan terlihat bahwa kedua database dapat terhubung dan berfungsi sesuai dengan harapan sistem. *Firebase Authentication* dapat berfungsi dengan baik dan Database *Firestore* dapat menerima dan mengirim data yang berisi identitas pengguna pada website.

Kata kunci— *website*, *monitoring suhu*, *firebase*, *Firestore*, *reactjs*

I. PENDAHULUAN

Minuman teh memiliki popularitas tinggi di kalangan masyarakat Indonesia dan proses pengolahannya melibatkan berbagai tahapan seperti pelayuan, pendinginan, penggulungan, dan pengeringan menggunakan mesin Ball Tea. Pada tahap pengeringan terakhir, mesin Ball Tea digunakan untuk mengurangi kadar air pada daun teh hingga mencapai 5-6%, dengan durasi proses 8-10 jam dan suhu optimal berkisar antara 100-130°C.

Dalam dunia industri, proses monitoring suhu merupakan hal yang penting untuk dilakukan. Dengan adanya monitoring suhu, data yang diperoleh dapat digunakan untuk melakukan *quality control* dan analisa jika terjadi suatu kendala [7]. Monitoring suhu akan lebih efisien jika dilakukan dengan website, sehingga monitoring dapat dilakukan dimana saja. Pencatatan suhu yang dilakukan secara berkala dan terus-menerus dengan manual masih kurang akurat dan membuat pekerjaan tersebut tidak efisien.

Penelitian ini bertujuan untuk merancang sistem *authentication* pada website yang akan digunakan untuk memonitoring suhu. Dengan adanya *authentication*, dapat

memberikan keamanan dan memastikan bahwa hanya pengguna yang sah yang dapat mengakses data suhu pada mesin *ball tea*. Kombinasi antara *firebase authentication* dan *Firestore* memberikan solusi yang aman dan efisien untuk manajemen pengguna dan penyimpanan data pada platform *firebase*.

II. KAJIAN TEORI

Teh menjadi salah satu komoditi yang memiliki peranan penting di Indonesia. Produksi teh Indonesia kebanyakan diekspor ke berbagai negara tujuan, seperti Rusia, Malaysia, Amerika Serikat, China, dan Taiwan. Secara keseluruhan, Indonesia menyumbang sekitar 2% terhadap total produksi teh di dunia. Berdasarkan data pada tahun 2018 - 2021, Indonesia menempati urutan ketujuh sebagai negara yang paling banyak memproduksi teh di dunia, dengan produksi sebanyak 589.750 ton. Pada proses pengolahan daun teh, Suhu pada pengeringan kedua cenderung tidak diperhatikan secara berkala sehingga menyebabkan kualitas daun teh yang kurang baik untuk dijual dan gosong. Jika kualitas produk menurun, maka perusahaan akan mengalami kerugian.

Pada perancangan sistem ini membutuhkan beberapa platform, yaitu antara lain:

A. ReactJs

ReactJS merupakan *front-end library* yang berguna bagi pengembang web untuk membangun antarmuka pengguna dan sering digunakan sebagai pendukung dari *web-framework*. Beberapa keunggulan *ReactJS* meliputi kecepatan, kemudahan penggunaan, dan kemampuan skalabilitas. Dengan *ReactJS*, pengembang memiliki kemampuan untuk membuat komponen antarmuka pengguna yang lebih interaktif, memiliki keadaan (*stateful*), dan *reusable*.

B. Firebase Authentication

Firebase Authentication merupakan suatu sistem membantu sistem autentikasi yang aman sekaligus meningkatkan proses masuk dan registrasi akun pengguna. Sistem ini memungkinkan aplikasi untuk menyimpan data pengguna dengan aman di *cloud* dan memberikan pengalaman pribadi yang sama di semua perangkat. Pada saat registrasi akun, pengguna akan memasukkan email dan

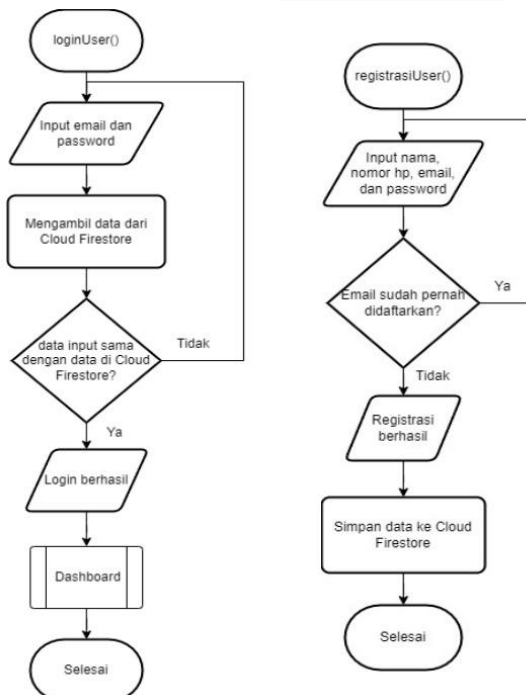
password. Data tersebut akan dikirimkan ke database. *Firebase Authentication* juga mendukung autentikasi dengan Google. Dalam hal ini, *Firebase* akan mengirim token ID Google pengguna ke server. Token tersebut kemudian akan ditukarkan dengan kredensial pengguna.

C. Cloud Firestore

Cloud Firestore merupakan bagian dari layanan *Firebase* di *Google Cloud Platform*, adalah penyimpanan data database NoSQL yang memberikan keunggulan dalam kinerja dan skalabilitas. Berbeda dengan database relasional, database NoSQL ini mengorganisir informasi tanpa hubungan langsung antar item data. Ketika data disimpan di *Cloud Firestore* melalui *Firebase*, akses data dapat dilakukan dari mana saja, dan dengan menggunakan stream, data dapat diperbarui secara real-time.

III. METODE

Pada metode penelitian ini akan dijelaskan cara kerja sistem *firebase authentication* dan *cloud firestore* serta implementasi dan pengujian pada *website*. Gambar 1 merupakan flowchart cara kerja dari sistem autentikasi dan penyimpanan data pengguna. Langkah pertama pengguna diarahkan pada laman registrasi untuk membuat akun dengan memasukkan nama, nomor hp, email, dan password. Data yang diinputkan akan otomatis terkirim pada *cloud firestore* dan data autentikasi yaitu email dan password akan masuk ke *firebase authentication*. Pada *cloud firestore* dan *firebase authentication* memiliki UserID yang sama sehingga pada fitur profile dapat menampilkan data pengguna yang sesuai.



GAMBAR 1
Flowchart login dan register

Adapun sintaksis JavaScript yang ditulis pada ReactJS. Konfigurasi *firebase* untuk menginisialisasi dan menghubungkan *firebase* dengan *website*. Konfigurasi ini

mencakup informasi seperti API, ID proyek, dan informasi autentikasi serta *firestore*.

```

1 // Import { initializeApp } from "firebase/app";
2 // Import { getAuth } from "firebase/auth";
3 import { firebase } from "firebase/app";
4 import { getAuth } from "firebase/auth";
5
6 const firebaseConfig = {
7   apiKey: "AIzaSyBmQWj5j2AoljF8_vv_vgl48RtKtc",
8   authDomain: "monsu-7a532.firebaseio.com",
9   projectId: "monsu-7a532",
10  storageBucket: "monsu-7a532.appspot.com",
11  messagingSenderId: "99262294288",
12  appId: "1:99262294288:web:6c6abbf3b46278ee328bd",
13  measurementId: "G-0JG810XG0R",
14 };
15
16 // const app = initializeApp(firebaseConfig);
17 // const auth = getAuth();
18 const app = firebase.initializeApp(firebaseConfig);
19 const auth = getAuth(app);
20 const db = firebase.firestore();
21
22 export { auth, db };
  
```

Gambar 2
Sintaks konfigurasi firebase

IV. HASIL DAN PEMBAHASAN

A. Implementasi Fitur Login

Pada fitur login, digunakan metode 'signInWithEmailAndPassword' seperti pada gambar 3 yang berfungsi untuk memverifikasi identitas pengguna dan memberikan akses ke *website* setelah pengguna memasukan email dan password yang sesuai dengan data yang ada pada *firebase authentication*.

```

1 import { signInWithEmailAndPassword } from "firebase/auth";
2 import React, { useState } from "react";
3 import { Link, useNavigate } from "react-router-dom";
4 import { auth } from "../firebase/auth";
5 import Navbar from "../components/Navbar/Navbar";
6 import img from "../assets/Bg_lp_fix.png";
7
8 const login = () => {
9   const [email, setEmail] = useState("");
10  const [password, setPassword] = useState("");
11  const [error, setError] = useState("");
12  const navigate = useNavigate();
13
14  const handleSubmIt = async (e) => {
15    e.preventDefault();
16    try {
17      const userCredential = await signInWithEmailAndPassword(
18        auth,
19        email,
20        password,
21      );
22      const user = userCredential.user;
23      localStorage.setItem("token", user.accessToken);
24      localStorage.setItem("user", JSON.stringify(user));
25      navigate("/dashboard");
26    } catch (error) {
27      setError("Email atau password kamu salah");
28    }
29  };
30 }
  
```

GAMBAR 3
Sintaks fitur login

Identifier	Providers	Created	Signed In	User UID
sven.nine@...		Jan 23, ...	Jan 23, ...	sYnCl8nWbgf...
sven.nine20...		Jan 23, ...	Jan 23, ...	miq4skLBSZUEq...
088899998...		Jan 2, 2...	Jan 2, 2...	pGVueWnyz4FX...

GAMBAR 4
Data autentikasi pada firebase

B. Implementasi Register

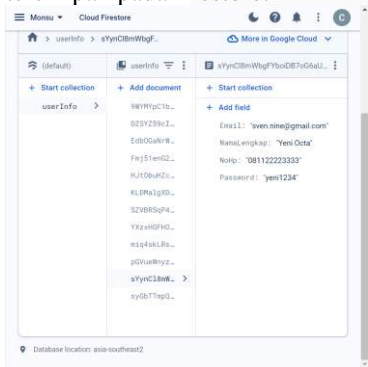
Pada fitur register, digunakan metode 'createUserWithEmailAndPassword' untuk membuat pengguna baru dengan menggunakan email dan password. Data yang masuk akan terkirim ke *firebase authentication*. Adapun metode 'doc' dan 'set' yang digunakan untuk

membuat dan mengakses data pada firestore. Sintaks ‘doc’ digunakan untuk referensi data dengan ID data “userInfo” dan “user.uid” untuk menyamakan data dengan firebase authentication. Sintaks ‘set’ digunakan untuk menetapkan nilai pada data yang akan dikirim.

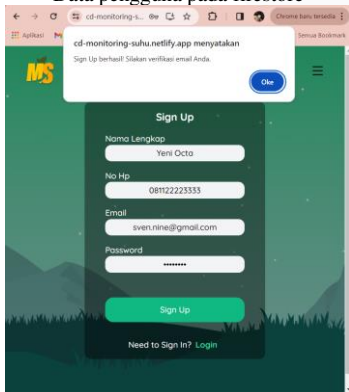
```
9  const SignUp = () => {
10   const [namaLengkap, setNamaLengkap] = useState("");
11   const [noHp, setNoHp] = useState("");
12   const [email, setEmail] = useState("");
13   const [password, setPassword] = useState("");
14   const [error, setError] = useState(null);
15   const navigate = useNavigate();
16
17   const handleSubmit = async (e) => {
18     e.preventDefault();
19     try {
20       const userCredential = await createUserWithEmailAndPassword(
21         auth,
22         email,
23         password,
24       );
25       const user = userCredential.user;
26       const dbref = doc(db, "userInfo", user.uid);
27       await setDoc(dbref, {
28         NamaLengkap: namaLengkap,
29         NoHp: noHp,
30         Email: email,
31         Password: password,
32       });
33       alert("Sign In Successfully");
34       navigate("/login");
35     } catch (error) {
36       setError(error.message);
37       console.error("Error signing up:", error);
38     }
39     setEmail("");
40     setNamaLengkap("");
41     setNoHp("");
42     setPassword("");
43   };
44 }
```

GAMBAR 5
Sintaks fitur register

Gambar 6 merupakan hasil implementasi fitur register yang menunjukkan terdapat banyak data yang sudah tersimpan pada firestore dan tampilan dari website apabila data sudah terkirim dan tersimpan pada firestore.



GAMBAR 6
Data pengguna pada firestore



GAMBAR 7
Tampilan pengguna berhasil register

C. Pengujian

Pada pengujian web menggunakan pengujian black-box. Black-box merupakan pengujian yang digunakan untuk

menguji perangkat lunak tanpa memerinci rincian perangkat lunak. Dalam pengujian black-box bertujuan untuk mengetahui seluruh fitur dapat berfungsi sesuai dengan yang diharapkan. Hasil pengujian pada halaman login dapat dilihat pada tabel 1

TABEL 1
Pengujian pada halaman login

Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian
Belum melakukan Sign Up	Klik “Create Account”	Menampilkan halaman sign up	Sesuai harapan
Username dan Password tidak diisi kemudian klik “Login”	Username: Kosong Password: Kosong	Tidak dapat masuk dan menampilkan tanda yang harus diisi	Sesuai harapan
Username dan Password tidak sesuai atau belum melakukan Sign Up kemudian klik “Login”	Username: user@gmail.com Password: pass	Tidak dapat login dan akan menampilkan notifikasi “email atau password kamu salah”	Sesuai harapan
Username dan Password sesuai tetapi belum melakukan verifikasi e-mail kemudian klik “Login”	Username: user@gmail.com Password: pass	Tidak dapat login dan akan menampilkan notifikasi “Silahkan verifikasi e-mail Anda sebelum login”	Sesuai harapan
Username dan Password yang sesuai dengan data Sign Up, telah melakukan verifikasi email, kemudian klik “Login”	Username: user@gmail.com Password: user123	Sistem menerima akses login dan akan menampilkan halaman dashboard	Sesuai harapan

Hasil pengujian pada halaman sign up dapat dilihat pada tabel 2

TABEL 2
Hasil pengujian pada halaman sign up

Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian
Pengisian sign up tidak lengkap, lalu klik “Sign Up”	Nama lengkap: Kosong No Hp: Kosong Email: Kosong Password: Kosong	Tidak dapat sign up dan menampilkan tanda yang harus diisi	Sesuai harapan
Pengisian password kurang dari 6 karakter, lalu klik “Sign Up”	Nama Lengkap: user No Hp: 12345 Email: user@gmail.com Password: user	Tidak dapat melakukan sign up dan menampilkan notifikasi “Password harus lebih dari 6 karakter”	Sesuai harapan

Pengisian <i>Sign Up</i> secara lengkap dan sesuai, kemudian klik " <i>Sign Up</i> "	<i>Sign Up</i> terisi semua dan sesuai	Sistem menerima akses <i>sign up</i> dan akan menampilkan notifikasi " <i>Sign Up</i> berhasil"	Sesuai harapan
Sudah pernah mengisi <i>Sign Up</i>	Klik " <i>Login</i> " pada halaman <i>sign up</i>	Akan menampilkan halaman <i>login</i>	Sesuai harapan

V. KESIMPULAN

Hasil perancangan yang telah dilakukan terlihat bahwa kedua database dapat terhubung dan berfungsi sesuai dengan harapan sistem. Firebase authentication dan cloud firestore dapat diimplementasikan pada fitur login dan register, khususnya untuk menerima dan menyimpan data yang berisi identitas pengguna. Hasil pengujian menggunakan *black box* dengan beberapa *test case* didapatkan hasil respon website sesuai dengan yang diharapkan.

REFERENSI

- [1] N. Miskiyah. (2022, November) "Sistem Monitoring Pengeringan Teh Hijau Tahap Dua (Ball Tea) Berbasis Internet of Things (IoT)". Available: <https://elibrary.unikom.ac.id/> [22/01/2024].
- [2] F. D. Silalahi, J. Dian and N. (2021, Oktober), "Implementasi Internet Of Things (Iot) Dalam Monitoring Suhu Dan Kelembaban Ruang Produksi Obat Non Steril Menggunakan Arduino Berbasis Web," *Jurnal Penelitian Ilmu dan Teknologi Komputer*, vol. 13, no. 2, pp. 62-68. Available: <https://doi.org/10.5281/3662.jupiter.2021.10> [22/01/2024]
- [3] D. Manumono and Listiyani. (2022, September), "Kajian Perkembangan Teh di Indonesia," *Journal of Agribusiness Plantations*, vol. 2, no. 2, pp. 133-146. Available: <https://doi.org/10.55180/aft.v2i2.281> [22/01/2024]
- [4] S. Febriani and F. Purwaningtias. (2022, Februari), "Implementasi Platform As A Service (PAAS) Pada Aplikasi Getfix Berbasis Cloud Computing," *Jurnal Sains dan Informatika*, vol. 8, no. 2, pp. 86-95. Available: <https://proceeding.unpkediri.ac.id/index.php/stains/article/view/1444> [22/01/2024]
- [5] J. Panjaitan and A. F. Pakpahan. (2021, april), "Perancangan Sistem E-reporting Menggunakan ReactJS dan Firebase," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, pp. 20-34. Available: <https://doi.org/10.28932/jutisi.v7i1.3098> [22/01/2024]
- [6] R. s. Huda, A. F. Kurniawan and M. (2022, Febuari), "Penerapan Firebase Pada Aplikasi E-Wisata Berbasis Android," *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS TAHUN*, vol. 1, pp. 247-257. Available: <https://proceeding.unpkediri.ac.id/index.php/stains/article/view/1444> [22/01/2024]