

SURAT KETERANGAN HASIL PENGECEKAN TURNITIN



UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN

Gedung Rektorat Lantai 3,
Jl. Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

SURAT KETERANGAN HASIL PENGECEKAN TURNITIN

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul : IMPLEMENTASI ALGORITMA TRIPLE EXPONENTIAL
SMOOTHING & TRANSFORMER LAPORAN PENGADUAN MASYARAKAT
BERBASIS WEB UNTUK KEPOLISIAN DI WILAYAH JAKARTA UTARA
Penulis : **Kanro**
NIM : **2020230078**
Tgl pemeriksaan : 10 Februari 2025

Dengan hasil Tingkat Kesamaan (*similarity index*) **14%**

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 10 Februari 2025

Ka.UPT Perpustakaan Unsada

Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi

Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan
dan Pasca Sarjana

LAMPIRAN

```
import streamlit as st
import pyrebase
import admin.index as admin_index # Mengimpor file
admin/index.py setelah login sebagai admin
import public.index as public_index # Mengimpor file
public/index.py setelah login sebagai public

# Konfigurasi Firebase untuk pyrebase
firebaseConfig = {
    "apiKey": "AIzaSyADwagZsQxhQBQpwvubO9ulz7QHruJyVkm",
    "authDomain": "lapor-cde21.firebaseio.com",
    "databaseURL": "https://lapor-cde21-default-
rtadb.firebaseio.com",
    "projectId": "lapor-cde21",
    "storageBucket": "lapor-cde21.appspot.com",
    "messagingSenderId": "89266309551",
    "appId": "1:89266309551:web:b9f1dc04ac75721922c75a",
    "measurementId": "G-4NRHYVEE7Y"
}

firebase = pyrebase.initialize_app(firebaseConfig)
auth = firebase.auth()
db = firebase.database()

# Fungsi untuk registrasi pengguna
def register_user(email, password, role):
    try:
        user =
        auth.create_user_with_email_and_password(email,
password)
        user_id = user['localId']
        # Menyimpan informasi pengguna di database
        db.child("users").child(user_id).set({
            "email": email,
            "role": role
        })
        st.success("Registrasi berhasil! Silakan login.")
    except Exception as e:
        st.error(f"Terjadi kesalahan saat registrasi: {e}")

# Fungsi untuk login pengguna
def login_user(email, password):
    try:
        user = auth.sign_in_with_email_and_password(email,
password)
        user_id = user['localId']

        # Mendapatkan informasi peran pengguna
        user_info =
        db.child("users").child(user_id).get().val()
        role = user_info.get("role", "Public") # Default ke
Public jika role tidak ditemukan
```

```

st.session_state['logged_in'] = True
st.session_state['user'] = user
st.session_state['role'] = role
st.success(f"Selamat datang, {email}!")

# Menentukan halaman berdasarkan peran
if role == "Admin":
    st.session_state['page'] = 'admin'
else:
    st.session_state['page'] = 'public'

st.rerun() # Reload halaman agar login form tidak
muncul
except Exception as e:
    st.error(f"Terjadi kesalahan saat login: {e}")

# Fungsi utama aplikasi login dan registrasi
def main():
    # Inialisasi session state jika belum ada
    if 'logged_in' not in st.session_state:
        st.session_state['logged_in'] = False
    if 'page' not in st.session_state:
        st.session_state['page'] = 'login'

    # Menampilkan halaman sesuai dengan status login
    if st.session_state['logged_in']:
        if st.session_state['page'] == 'admin':
            admin_index.main() # Menampilkan halaman admin
            setelah login sebagai admin
        elif st.session_state['page'] == 'public':
            public_index.main() # Menampilkan halaman public
            setelah login sebagai public
        else:
            option = st.selectbox("Pilih opsi", ["Login",
"Registrasi"])

            email = st.text_input("Email")
            password = st.text_input("Password", type="password")

            if option == "Registrasi":
                role = st.radio("Pilih jenis akun:", ("Admin",
"Public"))
                if st.button("Daftar"):
                    register_user(email, password, role)
            elif option == "Login":
                if st.button("Login"):
                    login_user(email, password)

if __name__ == "__main__":
    main()

```