

LAMPIRAN

Lampiran 1 Berita Acara Ujian Skripsi



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

Nama : Moh. Harifin
Nim : 2118131
Jurusan : Teknik Informatika S-1
Judul : Pengembangan Aplikasi Presensi Siswa Dengan Metode Location Based Services (Lbs) Berbasis Android Di Smk Islam Al-Futuhiyyah

Dipertahankan Dihadapan Majelis Penguji Skripsi Jenjang Strata Satu(S-1)
Pada

Hari : Rabu
Tanggal : 22 Januari 2025
Nilai : 84 (A)

Panitia Ujian Skripsi :
Ketua Majelis Penguji


(Yosep Agus Pranoto ST., MT)

NIP.P 1031000432

Anggota Penguji :

Dosen Penguji I


Ahmad Faisol/ST.MT.


NIP.P 1031000431

Dosen Penguji II


Eko Heri Susanto S.Kom., M.Kom.

NIP.P 103 24 00605

Lampiran 2 Formulir Perbaikan Skripsi



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang


FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika , maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Moh. Harifin
 NIM : 2118131
 JURUSAN : Teknik Informatika S-1
 JUDUL : PENGEMBANGAN APLIKASI PRESENSI SISWA DENGAN METODE LOCATION BASED SERVICES (LBS) BERBASIS ANDROID DI SMK ISLAM AL-FUTUHIYYAH


No.	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	22 Januari 2025	1. Setup Lokasi menggunakan peta 2. Perbaiki izin yang error 3. Tambahkan filter / pencarian di presensi guru 4. Tambahkan prosentase kehadiran pada report 5. Orang tua juga membutuhkan Riwayat /report kehadiran siswa	
2.	Penguji II	22 Januari 2025	Pada latar belakang disebutkan "bagaimana keefektifan pengembangan sistem presensi LBS jika di terapkan di SMK Islam Al-Futuhiyyah" Tetapi pada bab 4 yaitu implementasi dan pengujian tidak disebutkan skenario uji coba untuk mengukur efektif / tidaknya, belum disebutkan. Lalu pada bab Kesimpulan hanya disebutkan hasil kuisisioner itu bisa menyatakan bahwa penelitian ini sudah efektif? parameter efektif itu apa saja ?	

Dosen Penguji I




Ahmad Faisol, ST.MT.
NIP.P 1031000431

Dosen Penguji II



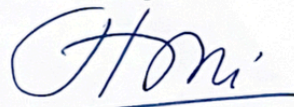
Eko Heri Susanto S.Kom., M.Kom.
NIP. P 1032400605

Dosen Pembimbing I




Nurlaily Vandyansyah, ST.MT
NIP.P 1031900557

Dosen Pembimbing II



Febriana Santi W. S.Kom., M.Kom.
NIP.P 1031000425

Lampiran 3 SK Dosen Pembimbing Utama

	<p>PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG INSTITUT TEKNOLOGI NASIONAL MALANG FAKULTAS TEKNOLOGI INDUSTRI FAKULTAS TEKNIK SIPIL DAN PERENCANAAN PROGRAM PASCASARJANA MAGISTER TEKNIK</p>
PT. BNI (PERSERO) MALANG BANK NIAGA MALANG	Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III.INF/TA/2024
Lampiran : ---
Perihal : Pembimbing Utama Skripsi

Kepada : **Yth. Bpk/Ibu Nurlaily Vendyansyah ST., MT.**
Dosen Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

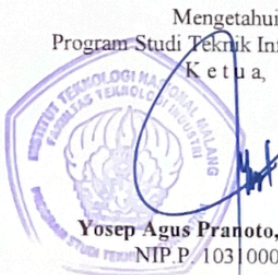
Nama : Mohammad Harifin
Nim : 2118131
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

21 Agustus 2024 s/d 21 Pebruari 2025


Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,


Yosep Agus Pranoto, ST., MT.
NIP.P. 1031000432

Form S-4a

Lampiran 4 SK Dosen Pembimbing Pendamping

	PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG INSTITUT TEKNOLOGI NASIONAL MALANG FAKULTAS TEKNOLOGI INDUSTRI FAKULTAS TEKNIK SIPIL DAN PERENCANAAN PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG BANK NIAGA MALANG	Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III INF/TA/2024
Lampiran : ---
Perihal : Pembimbing Pendamping Skripsi

Kepada : **Yth. Bpk/Ibu Febriana Santi Wahyuni S.Kom., M.Kom.**
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :


Nama : Mohammad Harifin
Nim : 2118131
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

21 Agustus 2024 s/d 21 Pebruari 2025

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua


Yosep Agus Pranoto, ST., MT.
NIP. P. 1031000432

Form S-4a

Lampiran 5 Formulir Bimbingan Skripsi Utama

FORMULIR BIMBINGAN SKRIPSI

Nama : Mohammad Harifin
 Nim : 2118131
 Masa Bimbingan :
 Judul Skripsi : Pengembangan Aplikasi Presensi Siswa Dengan Metode Location Based Services (Lbs) Berbasis Android Di Smk Islam Al-Futuhiyyah Kota Sumenep

No.	Tanggal	Uraian	Paraf Pembimbing
1.	18/11/24	Wawancara instansi yg membutuhkan yakni Yayasan; Analisa Kebutuhannya; Bab 1: Latar Belakang ditulis dg SW+1H; Rumusan & Tujuan;	
2.		Desain Sistem; Perancangan Sistem; Bab 1 Teori disesuaikan dg judul.	
3.	3/11/2024	Rumusan masalah dipertajam; Batasan Masalah dipertajam; Sistematisa Penulisan disesuaikan dg template penulisan skripsi;	
4.		Bab II: Tinjauan pustaka <Pendidikan Terkait> sebaiknya 10 tahun terakhir; Belum ada/	
5.		Teori masih terlalu sedikit LBS, GPS, Map, Sistem Presensi, Profil Yayasan, Teori Havershine	
6.		mohon dibuat flowchart umum, Android, website. Bab III: Analisis Kebutuhan Fungsi disesuaikan	
7.		dg 3 role id dan sesuaikan dg sitemap/Struktur menu Diagram Blok sistem; Flowchart sistem disesuaikan	
8.		dg Struktur Menu; TIDAK PERLU DFD; Desain Database; ERD; Perancangan Algoritma Aplikasi Presensi	
9.	17/11/24	bimbingan jurnal	
10.	20/11/24	pengisian jurnal	

Malang,..... 2024

Dosen Pembimbing

(Nurlaily Vendyansyah, S.T,M.T)

NIP. 1031900557





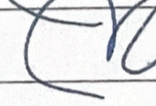
FORMULIR BIMBINGAN SKRIPSI

Nama : Mohammad Harifin

Nim : 2118131

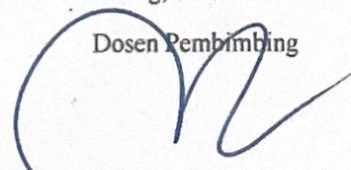
Masa Bimbingan :

Judul Skripsi : Pengembangan Aplikasi Presensi Siswa Dengan Metode location Based services berbasis Android Di smk Islam Al-Fulhiyyah

No.	Tanggal	Uraian	Paraf Pembimbing
1.	29/11/24	Demo program	
2.	05/11/24	laporan	
3.	10/12/24	revisi jurnal	
4.	13/01/24	konsul kompre	
5.	16/01/24	laporan	
6.			
7.			
8.			
9.			
10.			

Malang,..... 2024

Dosen Pembimbing



(Nurlaily Vandyansyah, S.T,M.T)

NIP. 1031900557

Lampiran 6 Formulir Bimbingan Skripsi Pendamping

FORMULIR BIMBINGAN SKRIPSI

Nama : Mohammad Harifin

Nim : 2118131

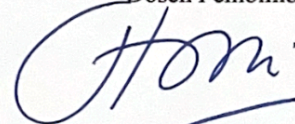
Masa Bimbingan :

Judul Skripsi : Pengembangan Aplikasi Presensi Siswa Dengan Metode Location Based Services (Lbs) Berbasis Android Di Smk Islam Al-Futuhiyyah

No.	Tanggal	Uraian	Paraf Pembimbing
1.	17/10/24	Lanjutkan program, konsultasi tesikan ds dosen penguji caminar proposal	hri
2.	24/10/24	Urgulkan UI desain front end	hri
3.	31/10/24	Tambahkan fitur up Guru pd front-end. ATT Seminar Progress.	hri
4.	24/11/24	Lakukan pengujian secara ds smk kasus.	hri
5.	29/11/24	Buat dokumentasi pengujian	hri
6.	4/12/24	Revisi laporan	hri
7.	6/12/24	ATT Seminar Hasil	hri
8.	9/1/25	Revisi laporan	hri
9.	13/1/25	Revisi laporan	hri
10.	15/1/25	ATT Uji Komprehensif	hri

Malang, 15/1/2024

Dosen Pembimbing



(Febriana Santi W. S.Kom., M.Kom)

NIP. 1031000425

Lampiran 7 Surat Izin Publikasi Data



**YAYASAN PENDIDIKAN ISLAM AL FUTUHIYYAH
SMK ISLAM AL FUTUHIYYAH ARJASA**

Status/NSS : Terakreditasi/402052830002

Jl. Pahlawan Lembungan Kalinganyar, Arjasa. E-mail.futuhiyyahs@gmail.com. Pos 69491

SURAT PERNYATAAN

Yang bertandatangan di bawah ini :

Nama : H. Mathanin, S.E
Jabatan : Kepala Sekolah
Instansi/Lembaga : SMK ISLAM AL-FUTUHIYYAH
No. Telepon/HP : 0818384967
Alamat : Jl. Pahlawan Lembungan Kalinganyar, Arjasa, kabupaten sumenep, jawa timur

Dengan ini menyatakan bersedia menjadi mitra dalam pembuatan skripsi dari

Nama : Mohammad Harifin
NIM : 2118131
Judul Skripsi : Pengembangan Aplikasi Presensi Siswa Dengan Metode Location Based Services (Lbs) Berbasis Android Di Smk Islam Al-Futuhiyyah

Dan saya menyatakan bahwa data yang digunakan dalam skripsi tersebut adalah benar dan boleh dipublikasikan. Demikian surat pernyataan ini saya buat tanpa ada paksaan dan tekanan dari pihak manapun serta dapat dipergunakan sebagaimana mestinya.

Catatan :

Sumenep, 17 januari 2025

Yang membuat pernyataan



H. MATHANIN, S.E)

Lampiran 8 Dokumentasi

Pada beberapa gambar di bawah merupakan hasil dokumentasi untuk melakukan proses presensi dari pengguna guru dan siswa.





Lampiran 9 Source Code

1. GetUserLocation.js

```
import React, {useEffect, useState} from 'react';
import {PermissionsAndroid, Alert} from 'react-native';
import Geolocation from 'react-native-geolocation-service';

const requestLocationPermission = async () => {
  try {
    const granted = await PermissionsAndroid.request(
      PermissionsAndroid.PERMISSIONS.ACCESS_FINE_LOCATION,
      {
        title: 'Akses Lokasi Diperlukan',
        message:
          'Aplikasi ini memerlukan akses lokasi untuk fungsi
presensi.',
        buttonNeutral: 'Nanti',
        buttonNegative: 'Tolak',
        buttonPositive: 'Oke',
      },
    );
    if (granted === PermissionsAndroid.RESULTS.GRANTED) {
      return true;
    } else {
      Alert.alert(
        'Izin Lokasi Ditolak',
        'Aplikasi memerlukan izin lokasi untuk melanjutkan.',
        [{text: 'Oke'}],
      );
      return false;
    }
  } catch (err) {
    console.warn(err);
    return false;
  }
};

const getUserLocation = async () => {
  const hasPermission = await requestLocationPermission();
  if (hasPermission) {
    return new Promise((resolve, reject) => {
      Geolocation.getCurrentPosition(
        position => {
          const {latitude, longitude} = position.coords;

          // Periksa apakah lokasi palsu
          if (position.mocked) {
            console.log('Lokasi palsu terdeteksi!');
            Alert.alert(
              'Peringatan',
              'Lokasi perangkat terdeteksi sebagai lokasi palsu.
Harap nonaktifkan Mock Location.',
            );
          }
        },
      );
    });
  }
};
```

```

        reject(new Error('Lokasi palsu terdeteksi'));
    } else {
        // Lokasi valid
        resolve({latitude, longitude});
        console.log('Lokasi pengguna:', position);
    }
},
error => {
    // console.error('Error mendapatkan lokasi pengguna:',
error);
    Alert.alert(
        'Error',
        'Gagal mendapatkan lokasi pengguna atau izin lokasi
tidak diberikan.',
    );
    reject(error);
},
{enableHighAccuracy: true, timeout: 15000, maximumAge:
10000}
);
});
} else {
    console.log('Izin lokasi tidak diberikan.');
```

```

return null;
}
};

export default getUserLocation;

```

2. UseUserData.js

```

import { useState, useEffect } from 'react';
import AsyncStorage from '@react-native-async-storage/async-
storage';
import { NativeEventEmitter, NativeModules } from 'react-native';

const userEventEmitter = new
NativeEventEmitter(NativeModules.ToastExample);

const useUserData = () => {
    const [userInfo, setUserInfo] = useState({
        nama: '',
        profileImage: require('../assets/images/noimage.jpg'),
        role: '',
    });

    const fetchUserData = async () => {
        try {
            const userData = await AsyncStorage.getItem('userData');
            if (userData) {
                const parsedData = JSON.parse(userData);
                const imageUrl = parsedData.foto || null; // URL lengkap
dari API
                setUserInfo({
                    ...parsedData,
```

```

        profileImage: imageUrl ? { uri: imageUrl } :
require('../assets/images/noimage.jpg'),
    });
    }
    } catch (error) {
        console.error('Error fetching user data:', error);
    }
};

const refreshUserData = async () => {
    await fetchUserData();
    userEventEmitter.emit('userDataUpdated'); // Emit event untuk
notifikasi data pengguna diperbarui
};

const checkAndResetLogin = async () => {
    try {
        const lastLogin = await
AsyncStorage.getItem('lastLoginTime');
        const currentDate = new Date();

        if (lastLogin) {
            const lastLoginDate = new Date(lastLogin);
            const resetTime = new Date();
            resetTime.setHours(4, 0, 0, 0);

            if (currentDate > resetTime && lastLoginDate < resetTime)
{
                await AsyncStorage.removeItem('userData');
                await AsyncStorage.removeItem('lastLoginTime');
                setUserInfo({ nama: '', profileImage:
require('../assets/images/noimage.jpg') });
                userEventEmitter.emit('userDataUpdated'); // Emit
event jika data direset
            }
        }
    } catch (error) {
        console.error('Error checking reset login:', error);
    }
};

useEffect(() => {
    checkAndResetLogin();
    fetchUserData();

    const subscription =
userEventEmitter.addListener('userDataUpdated', fetchUserData);

    return () => subscription.remove(); // Hapus listener saat
unmount
}, []);

return { userInfo, refreshUserData };
};

```

```
export default useUserData;
```

3. Haversine.js

```
// fungsi ubah ke radian
const toRad = (value) => (value * Math.PI) / 180;

// Menghitung jarak antara dua titik koordinat latitude dan
longitude dengan rumus Haversine
const haversine = (lat1, lon1, lat2, lon2) => {
  // Jari-jari bumi dalam kilometer
  const R = 6371;

  // mengubah coordinate ke radian
  lat1 = toRad(lat1);
  lon1 = toRad(lon1);
  lat2 = toRad(lat2);
  lon2 = toRad(lon2);

  // Menghitung selisih latitude dan longitude
  const dLat = lat2 - lat1;
  const dLon = lon2 - lon1;

  // Rumus Haversine
  const a =
    Math.sin(dLat / 2) * Math.sin(dLat / 2) +
    Math.cos(lat1) * Math.cos(lat2) *
    Math.sin(dLon / 2) * Math.sin(dLon / 2);
  const c = 2 * Math.asin(Math.sqrt(a));

  // Menghitung jarak dalam meter
  const distance = R * c * 1000; // mengkonversi ke meter
  return distance;
};

export default haversine;
```

4. Presensi/index.js

```
import React, {useEffect, useState} from 'react';
import {
  View,
  Alert,
  ActivityIndicator,
  Text,
  TouchableOpacity,
} from 'react-native';
import MapView, {Circle, Marker, PROVIDER_GOOGLE} from 'react-native-maps';
import axios from 'axios';
import {API_URL} from '../api/api';
import {Picker} from '@react-native-picker/picker';
import getUserLocation from '../hooks/getUserLocation';
import haversine from '../hooks/haversine';
```

```

import useUserData from '../hooks/useUserData';
import styles from './styles';
import {useNavigation} from '@react-navigation/native';

const InsertPresensi = () => {
  const navigation = useNavigation();
  const {userInfo, refreshUserData} = useUserData();
  const [lokasiData, setLokasiData] = useState([]);
  const [userLocation, setUserLocation] = useState(null);
  const [loading, setLoading] = useState(true);
  const [selectedSubject, setSelectedSubject] = useState({
    id: '', // Untuk menyimpan id mapel
    jadwal_id: '', // Untuk menyimpan id jadwal
  });

  const [subjects, setSubjects] = useState([]);
  const [isPresensiOpen, setIsPresensiOpen] = useState(false);

  const fetchUserLocation = async () => {
    try {
      setLoading(true);

      // Dapatkan posisi GPS pengguna
      const loc = await getUserLocation();
      setUserLocation(loc || null);
    } catch (error) {
      // console.error('Error mendapatkan lokasi pengguna:',
error);
      setUserLocation(null);
    } finally {
      setLoading(false);
    }
  };

  useEffect(() => {
    if (userInfo && userInfo.id_kelas) {
      fetchSubjects();
      fetchLokasi();
      fetchUserLocation();
      const intervalId = setInterval(() => {
        fetchUserLocation();
        fetchSubjects();
        fetchLokasi();
      }, 30000);
      return () => clearInterval(intervalId);
    }
  }, [userInfo]);

  const fetchPresensiStatus = async subjectId => {
    try {
      const url = API_URL.PRESENSI_STATUS(
        userInfo.id_kelas,
        userInfo.id_siswa,

```



```

        subjectId,
    );

    const response = await axios.get(url);
    setIsPresensiOpen(response.data.isPresensi);
  } catch (error) {
    Alert.alert('Gagal memeriksa status presensi.');
```

```

    setIsPresensiOpen(false);
  }
};

const fetchSubjects = async () => {
  try {
    const url = API_URL.JADWAL_BY_KELAS(userInfo.id_kelas);
    const response = await axios.get(url);
    if (response.data && response.data.data) {
      const mataPelajaran = response.data.data.map(item => ({
        id: item.id_mapel,
        id_jadwal: item.id_jadwal,
        name: item.mata_pelajaran,
      }));
      setSubjects(mataPelajaran);
    } else {
      Alert.alert('Error', 'Mata pelajaran tidak ditemukan.');
```

```

    }
  } catch (error) {
    Alert.alert('Error', 'Gagal mengambil data mata pelajaran.');
```

```

  }
};

const fetchLokasi = async () => {
  try {
    const url = API_URL.KELAS_BY_ID(userInfo.id_kelas);
    const response = await axios.get(url);
    if (response.data && response.data.data) {
      const {lokasi, radius} = response.data.data;
      setLokasiData([[lokasi, radius]]);
    } else {
      Alert.alert('Error', 'Data lokasi tidak ditemukan.');
```

```

    }
  } catch (error) {
    Alert.alert('Error', 'Gagal mengambil data lokasi.');
```

```

  }
};

const handleSubjectChange = (subjectId) => {
  const selectedSubjectData = subjects.find(subject =>
subject.id === subjectId);
  setSelectedSubject({
    id: selectedSubjectData.id,
    jadwal_id: selectedSubjectData.id_jadwal,
  });
  fetchPresensiStatus(selectedSubjectData.id);
};

```

```

};

const handlePresent = async () => {
  if (!selectedSubject.id || !selectedSubject.jadwal_id) {
    Alert.alert('Peringatan', 'Silakan pilih mata pelajaran
sebelum presensi.');
```

```

    return;
  }
  if (userLocation) {
    try {

      let canPresent = false;

      // Periksa apakah pengguna berada dalam radius lokasi
presensi
      lokasiData.forEach(item => {
        const [latitude, longitude] =
item.lokasi.split(',').map(parseFloat);
        const radiusInMeters = parseFloat(item.radius);
        const distance = haversine(
          userLocation.latitude,
          userLocation.longitude,
          latitude,
          longitude,
        );

        if (distance <= radiusInMeters) {
          canPresent = true;
        }
      });

      if (canPresent) {
        try {
          const url = API_URL.MELAKUKAN_PRESENSI;
          const response = await axios.post(url, {
            jadwal_id: selectedSubject.jadwal_id,
            siswa_id: userInfo.id_siswa,
            latitude: userLocation.latitude,
            longitude: userLocation.longitude,
          });

          if (response.data.message) {
            Alert.alert('Presensi', response.data.message);
          }

          setSelectedSubject({ id: '', jadwal_id: '' });
          setIsPresensiOpen(false);
          navigation.navigate('Home');
        } catch (error) {
          Alert.alert('Error', 'Gagal melakukan presensi.');
```

```

        }
      } else {
        Alert.alert('Presensi', 'Anda berada di luar radius
presensi.');
```

```

    }
    } catch (error) {
        console.error('Error mendeteksi mock location:', error);
        Alert.alert('Error', 'Gagal mendeteksi mock location.');
```

```

    }
    } else {
        Alert.alert('Error', 'Lokasi pengguna tidak ditemukan.');
```

```

    }
};

return (
    <View style={styles.container}>
        <View style={styles.header}>
            <Text style={styles.headerText}>Presensi</Text>
        </View>

        <View style={styles.mapContainer}>
            {loading ? (
                <ActivityIndicator size="large" color="green" />
            ) : (
                <MapView
                    provider={PROVIDER_GOOGLE}
                    style={styles.map}
                    initialRegion={{
                        latitude: userLocation ? userLocation.latitude : -
6.1753924,
                        longitude: userLocation ? userLocation.longitude :
106.8271528,
                        latitudeDelta: 0.0001,
                        longitudeDelta: 0.0001,
                    }}>
                    {lokasiData.map((item, index) => (
                        <Circle
                            key={index}
                            center={{
                                latitude:
parseFloat(item.lokasi.split(',')[0]),
                                longitude:
parseFloat(item.lokasi.split(',')[1]),
                            }}
                            radius={parseFloat(item.radius)}
                            strokeColor="rgba(76, 175, 80, 0.5)"
                            fillColor="rgba(76, 175, 80, 0.2)"
                        />
                    ))}
                    {userLocation && (
                        <Marker coordinate={userLocation} title="Lokasi
Anda" />
                    )}
                </MapView>
            )}
        </View>

        <View style={styles.subjectContainer}>

```

```

        <Picker
            selectedValue={selectedSubject.id}
            style={styles.picker}
            onChange={handleSubjectChange}
            itemStyle={{color: 'black'}}>
        <Picker.Item label="Pilih Mata Pelajaran" value="" />
        {subjects.map(subject => (
            <Picker.Item
                key={subject.id}
                label={subject.name}
                value={subject.id}
            />
        ))}
    </Picker>

    <TouchableOpacity
        style={[
            styles.presentButton,
            {backgroundColor: isPresensiOpen ? 'green' :
'gray'},
        ]}
        onPress={handlePresent}
        disabled={!isPresensiOpen}>
    <Text
style={styles.presentButtonText}>Presensi</Text>
    </TouchableOpacity>
    </View>
</View>
);
};

export default InsertPresensi;

```

5. Http/Controllers/API/PresensiController.php

```

<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Models\Jadwal;
use App\Models\Kelas;
use App\Models\Presensi;
use App\Models\Siswa;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Mail;
use App\Mail\KirimEmail as MailKirimEmail;
use App\Models\Guru;

class PresensiController extends Controller
{
    public function updatePresensiStatus(Request $request,
$id)

```

```

    {
        // Validasi status yang akan diupdate
        $request->validate([
            'status' => 'required|in:Hadir,Tidak
            Hadir,Izin,Sakit,Belum Absen'
        ]);

        try {
            // Ambil data presensi berdasarkan ID
            $presensi = Presensi::findOrFail($id);

            // cek tanggal presensi
            $tanggal = date('Y-m-d');
            if ($presensi->tanggal != $tanggal) {
                return response()->json([
                    'status' => 'error',
                    'message' => 'Tidak dapat mengubah status
                    presensi pada tanggal yang berbeda.',
                ], 400);
            }
            // Update status presensi
            $presensi->status = $request->status;
            $presensi->sesi_status = 'Ditutup';
            $presensi->save();

            return response()->json([
                'status' => 'success',
                'message' => 'Status presensi berhasil diubah',
                'data' => $presensi,
            ]);
        } catch (\Exception $e) {
            return response()->json([
                'status' => 'error',
                'message' => 'Terjadi kesalahan saat mengubah
                status presensi',
                'error' => $e->getMessage(),
            ], 500);
        }
    }

    public function getPresensi(Request $request)
    {
        // Ambil data dari query string
        $id_jadwal = $request->query('id_jadwal');
        $pertemuan = $request->query('pertemuan');
        $id_guru = $request->query('id_guru');
        $nis = $request->query('nis');

        try {
            // Query builder untuk presensi
            $query = Presensi::query();

```

```

// Filter berdasarkan id_jadwal
if ($id_jadwal) {
    $query->where('jadwal_id', $id_jadwal);
}

// Filter berdasarkan pertemuan
if ($pertemuan) {
    $query->where('keterangan', 'LIKE',
"$pertemuan%");
}

// Filter berdasarkan id_guru
if ($id_guru) {
    $jadwalList = Jadwal::where('guru_id',
$id_guru)->pluck('id');
    $query->whereIn('jadwal_id', $jadwalList);
}

// Filter berdasarkan NIS jika NIS tidak kosong
if (!empty($nis)) {
    $siswaList = Siswa::where('nis', 'LIKE',
"%$nis%")->pluck('id');
    if ($siswaList->isEmpty()) {
        return response()->json([
            'status' => 'error',
            'message' => 'Tidak ada siswa yang
cocok dengan NIS yang dimasukkan.',
        ], 404);
    }
    $query->whereIn('siswa_id', $siswaList);
}

// Eksekusi query
$presensi = $query->with(['siswa.user',
'jadwal.mataPelajaran'])
->orderBy('tanggal', 'desc')
->get();

// Jika tidak ada data presensi
if ($presensi->isEmpty()) {
    return response()->json([
        'status' => 'error',
        'message' => 'Data presensi tidak
ditemukan.',
    ], 404);
}

// Format data presensi
$presensiFormatted = $presensi->map(function
($item) {
    $siswa = $item->siswa;
    $jadwal = $item->jadwal;
    return [
        'id' => $item->id,

```

```

        'nis' => $siswa ? $siswa->nis : 'NIS tidak
tersedia',
        'kelas' => $siswa ? $siswa->kelas->nama :
'Kelas tidak tersedia',
        'mata_pelajaran' => $jadwal ? $jadwal-
>mataPelajaran->nama : 'Mata Pelajaran tidak tersedia',
        'tanggal' => $item->tanggal,
        'jam' => $item->jam,
        'status' => $item->status,
        'siswa_nama' => $siswa ? $siswa->nama :
'Nama tidak tersedia',
        'jadwal_mapel' => $jadwal ? $jadwal-
>mataPelajaran->nama : 'Mata Pelajaran tidak tersedia',
        'keterangan' => $item->keterangan,
    ];
});

return response()->json([
    'status' => 'success',
    'message' => 'Data berhasil diambil',
    'data' => $presensiFormatted,
]);
} catch (\Exception $e) {
    return response()->json([
        'status' => 'error',
        'message' => 'Terjadi kesalahan saat mengambil
data presensi.',
        'error' => $e->getMessage(),
    ], 500);
}

public function getPresensiBySiswaId($siswa_id)
{
    // Mengambil data presensi berdasarkan siswa_id dengan
pengecualian status "Belum Absen"
    $presensi = Presensi::where('siswa_id', $siswa_id)
        ->where('status', '!=', 'Belum Absen') //
Menambahkan filter untuk status selain "Belum Absen"
        ->with(['jadwal.mataPelajaran', 'siswa.user'])
        ->orderBy('tanggal', 'desc')
        ->get();

    // jika presensi tidak ditemukan
    if ($presensi->isEmpty()) {
        return response()->json([
            'status' => 'success',
            'message' => 'Data presensi tidak ditemukan.',
            'data' => [],
        ]);
    }

    $formattedPresensi = $presensi->map(function ($item) {
        // jika foto user tidak ada, gunakan foto default

```

```

        if ($item->siswa->user->foto == null) {
            $foto = url('default.jpg');
        } else {
            $foto = url('foto/' . $item->siswa->user->foto);
        }
        return [
            'user_image' => $foto,
            'nama_siswa' => $item->siswa->user->nama,
            'mata_pelajaran' => $item->jadwal->mataPelajaran->nama,
            'jadwal_pembelajaran' => [
                'jam_mulai' => $item->jadwal->jam_mulai,
                'jam_selesai' => $item->jadwal->jam_selesai,
            ],
            'tanggal_presensi' => $item->tanggal,
            'jam_presensi' => $item->jam,
            'status' => $item->status,
        ];
    });

    return response()->json([
        'status' => 'success',
        'data' => $formattedPresensi,
    ]);
}

public function getPresensiByGuru($id_guru)
{
    // Validasi apakah $id_guru ada di tabel 'guru'
    $guru = Guru::find($id_guru);
    if (!$guru) {
        return response()->json([
            'status' => 'error',
            'message' => 'Guru tidak ditemukan.',
        ], 404);
    }

    try {
        $jadwalList = Jadwal::where('guru_id', $id_guru)->get();
        $presensiList = Presensi::whereIn('jadwal_id', $jadwalList->pluck('id'))
            ->with(['siswa.user', 'jadwal.mataPelajaran'])
            ->where('status', '!=', 'Belum Absen')
            ->orderBy('tanggal', 'desc')
            ->get();

        // Jika presensi tidak ditemukan
        if ($presensiList->isEmpty()) {
            return response()->json([
                'status' => 'success',
                'message' => 'Data presensi tidak ditemukan.',
            ]);
        }
    } catch (Exception $e) {
        return response()->json([
            'status' => 'error',
            'message' => $e->getMessage(),
        ], 500);
    }
}

```



```

        'data' => [],
    });
}

// Format data presensi dengan pengecekan foto
$formattedPresensi = $presensiList->map(function
($item) {
    // jk foto user tidak ada, gunakan foto default
    $foto = $item->siswa->user->foto ? url('foto/'
. $item->siswa->user->foto) : url('default.jpg');
    return [
        'user_image' => $foto,
        'nis' => $item->siswa->nis,
        'kelas' => $item->siswa->kelas->nama,
        'nama_siswa' => $item->siswa->user->nama,
        'mata_pelajaran' => $item->jadwal-
>mataPelajaran->nama,
        'jadwal_pembelajaran' => [
            'jam_mulai' => $item->jadwal-
>jam_mulai,
            'jam_selesai' => $item->jadwal-
>jam_selesai,
        ],
        'tanggal_presensi' => $item->tanggal,
        'jam_presensi' => $item->jam,
        'status' => $item->status,
    ];
});

return response()->json([
    'status' => 'success',
    'message' => 'Data presensi berhasil diambil.',
    'data' => $formattedPresensi,
]);
} catch (\Exception $e) {
    return response()->json([
        'status' => 'error',
        'message' => 'Terjadi kesalahan saat mengambil
data presensi.',
        'error' => $e->getMessage(),
    ], 500);
}
}
public function getPertemuan($jadwal_id)
{

    // Ambil keterangan unik berdasarkan jadwal
    $pertemuan = Presensi::where('jadwal_id', $jadwal_id)
->select('keterangan')
->distinct()
->pluck('keterangan');

    return response()->json([

```

```

        'message' => 'Daftar pertemuan berhasil diambil',
        'data' => $pertemuan
    });
}
public function BukaPresensi(Request $request)
{
    $request->validate([
        'jadwal_id' => 'required|exists:jadwal,id',
        'id_kelas' => 'required|exists:kelas,id',
        'keterangan' => 'nullable|string',
    ]);

    $tanggal = date('Y-m-d');

    DB::beginTransaction();

    try {
        $jadwal = Jadwal::find($request->jadwal_id);
        $kelas = Kelas::find($request->id_kelas);

        // cek apakah ada presensi pada tanggal ini dengan
        status "Ditutup" dengan status presensi Belum Absen
        $cekpresensi = Presensi::where('jadwal_id',
        $jadwal->id)
            ->where('tanggal', $tanggal)
            ->where('sesi_status', 'Ditutup')
            ->where('status', 'Belum Absen')
            ->get();
        if ($cekpresensi->isNotEmpty()) {
            // Jika ada presensi "Ditutup", ubah statusnya
            menjadi "Dibuka"
            foreach ($cekpresensi as $presensi) {
                $presensi->update([
                    'sesi_status' => 'Dibuka',
                    'waktu_buka' => date('H:i:s')
                ]);
            }
            DB::commit();
            return response()->json([
                'message' => 'Presensi berhasil dibuka
kembali.',
            ], 200);
        }
        // cek apakah jadwal dan tanggal ini sudah ada
        presensi
        $presensiHariIni = Presensi::where('jadwal_id',
        $jadwal->id)
            ->where('tanggal', $tanggal)
            // keterangan presensi harus sama dengan yang
            diinputkan
            ->exists();
        if ($presensiHariIni) {
            DB::rollBack();
            return response()->json([

```

```

        'message' => 'Presensi sudah ada',
    ], 409);
    }
    // cek presensi berdasarkan jadwal dan keterangan
    yang sama
    $presensiKeterangan = Presensi::where('jadwal_id',
    $jadwal->id)
        ->where('keterangan', $request->keterangan)
        ->exists();
    if ($presensiKeterangan) {
        DB::rollBack();
        return response()->json([
            'message' => 'Presensi dengan keterangan
    yang sama sudah ada pada pertemuan ini.',
        ], 409);
    }
    // cek apakah ada siswa di kelas ini
    $siswaKelas = Siswa::where('kelas_id', $kelas-
    >id)->exists();
    if (!$siswaKelas) {
        DB::rollBack();
        return response()->json([
            'message' => 'Tidak ada siswa di kelas
    ini.',
        ], 404);
    }
    // jika tidak ada presensi sama sekali pada tanggal
    ini, buat presensi baru
    $siswaList = Siswa::where('kelas_id', $kelas->id)-
    >get();
    foreach ($siswaList as $siswa) {
        Presensi::create([
            'tanggal' => $tanggal,
            'jam' => date('H:i:s'),
            'keterangan' => $request->keterangan ??
    'Pertemuan baru',
            'sesi_status' => 'Dibuka',
            'waktu_buka' => date('H:i:s'),
            'siswa_id' => $siswa->id,
            'jadwal_id' => $jadwal->id,
            'lokasi' => "Guru: " . ($request->kodeGuru
    ?? 'Tidak Diketahui') . " - Kelas: " . $kelas->nama,
        ]);
    }
    DB::commit();
    return response()->json([
        'message' => 'Presensi baru berhasil dibuka
    untuk pertemuan ini.',
    ], 201);
    } catch (\Exception $e) {
        DB::rollBack();
        return response()->json([
            'message' => 'Terjadi kesalahan saat membuka
    presensi.',

```

```

        'error' => $e->getMessage(),
    ], 500);
    }
}
public function MelakukanPresensi(Request $request)
{
    $request->validate([
        'jadwal_id' => 'required|exists:jadwal,id',
        'siswa_id' => 'required|exists:siswa,id',
        'latitude' => 'required|numeric',
        'longitude' => 'required|numeric',
    ]);

    $tanggal = date('Y-m-d');
    $jadwal_id = $request->jadwal_id;
    $siswa_id = $request->siswa_id;

    try {
        $presensi = Presensi::where('jadwal_id',
$jadwal_id)
            ->where('siswa_id', $siswa_id)
            ->where('tanggal', $tanggal)
            ->where('sesi_status', 'Dibuka')
            ->where('status', 'Belum Absen')
            ->first();

        if (!$presensi) {
            return response()->json([
                'message' => 'Presensi tidak tersedia atau
sesi sudah ditutup.',
            ], 404);
        }

        // Update status presensi dan lokasi
        $presensi->update([
            'status' => 'Hadir',
            'lokasi' => "{$request->latitude}, {$request-
>longitude}",
            'sesi_status' => 'Ditutup',
            'jam' => date('H:i:s'),
        ]);

        // jika presensi berhasil, kirim notifikasi ke
orang tua
        dispatch(function () use ($siswa_id, $tanggal) {
            $this->Notif_Email_Ortu($siswa_id, $tanggal);
            $this->Notif_WA_Ortu($siswa_id, $tanggal);
        });

        return response()->json([
            'message' => 'Presensi berhasil dilakukan',
            'data' => $presensi
        ], 200);
    }
}

```

```

    } catch (\Exception $e) {
        return response()->json([
            'message' => 'Gagal melakukan presensi',
            'error' => $e->getMessage(),
        ], 500);
    }
}
// cek status presensi siswa berdasarkan kelas dan jadwal
dan id_mapel
public function cekStatusPresensi($id_kelas, $id_siswa,
$id_mapel)
{
    $tanggal = date('Y-m-d');
    $jadwal = Jadwal::where('kelas_id', $id_kelas)-
>where('mapel_id', $id_mapel)->first();

    if (!$jadwal) {
        return response()->json([
            'isPresensi' => false
        ], 404);
    }

    // Periksa status presensi yang sesuai dengan kondisi
yang diberikan
    $isPresensi = Presensi::where('jadwal_id', $jadwal-
>id)
        ->where('siswa_id', $id_siswa)
        ->where('tanggal', $tanggal)
        ->where('status', 'Belum Absen')
        ->where('sesi_status', 'Dibuka')
        ->exists();

    return response()->json([
        'isPresensi' => $isPresensi
    ], 200);
}
// kirim notifikasi ke email oran_tua saat siswa melakukan
presensi
protected function Notif_Email_Ortu($siswa_id, $tanggal)
{
    $siswa = Siswa::with('user', 'kelas')-
>find($siswa_id);
    if (!$siswa || !$siswa->email_orangtua) {
        return false;
    }

    $jam = date('H:i:s');
    $orangTuaEmail = $siswa->email_orangtua;

    $data_email = [
        'subject' => 'Notifikasi Presensi Siswa',
        'sender_name' => 'smki@gmail.com',

```

```

        'isi' => "Anak Anda, {$siswa->user->nama}, telah
berhasil melakukan presensi pada tanggal {$tanggal} pukul
{$jam}.",
        'siswa' => $siswa,
        'tanggal' => $tanggal,
        'jam' => $jam,
    ];

    Mail::to($orangTuaEmail)->send(new
MailKirimEmail($data_email));

    return true;
}
// kirim notifikasi wa
public function Notif_WA_Ortu($siswa_id, $tanggal)
{
    $siswa = Siswa::with('user', 'kelas')->find($siswa_id);
    $jam = date('H:i:s');
    $pesan = "Notifikasi Presensi Siswa - SMKI AL-
FUTUHIYYAH\n\n" .
        "Yth. {$siswa->nama_orangtua},\n\n" .
        "Kami informasikan bahwa anak Anda, {$siswa->user-
>nama}, telah berhasil melakukan presensi pada:\n" .
        "- Tanggal: {$tanggal}\n" .
        "- Jam: {$jam}\n" .
        "- Kelas: {$siswa->kelas->nama}\n\n" .
        "Mohon pastikan bahwa anak Anda telah hadir di
kelas tepat waktu. Terima kasih atas perhatian dan
kerjasamanya.\n\n" .
        "Salam hangat,\n" .
        "Operator Sekolah\n" .
        "SMKI AL-FUTUHIYYAH";
    $nomorHp = $siswa->no_hp_orangtua;
    $token = 'DEvC9ApZhTWyhd7e5Wcb';
    $curl = curl_init();
    curl_setopt_array($curl, array(
        CURLOPT_URL => 'https://api.fonnte.com/send',
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_ENCODING => '',
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 0,
        CURLOPT_FOLLOWLOCATION => true,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => 'POST',
        CURLOPT_POSTFIELDS => array('target' => $nomorHp,
'message' => $pesan),
        CURLOPT_HTTPHEADER => array(
            'Authorization: ' . $token
        ),
    ));
    curl_exec($curl);
    curl_close($curl);
}

```

```

        // \Log::info('WhatsApp notification response: ' .
$response);
    }
    public function tutupOtomatisPresensi()
    {
        $currentTime = date('H:i:s');
        $tanggalHariIni = date('Y-m-d');

        // Ambil semua presensi dengan status 'Dibuka' pada
        hari ini yang sudah melebihi 10 menit
        $presensiList = Presensi::where('sesi_status',
'Dibuka')
            ->where('tanggal', $tanggalHariIni)
            ->whereRaw("TIMEDIFF('$currentTime', waktu_buka)
>= '00:10:00'") // 10 menit setelah waktu_buka
            ->get();

        foreach ($presensiList as $presensi) {
            $presensi->update([
                'sesi_status' => 'Ditutup'
            ]);
        }

        return response()->json([
            'message' => count($presensiList) . ' presensi
berhasil ditutup otomatis.'
        ]);
    }
    public function tutupPresensiOtomatisLewatHari()
    {
        $tanggalKemarin = date('Y-m-d', strtotime('-1 day'));

        // Ambil semua presensi dengan status 'Belum Absen'
        yang memiliki tanggal presensi sebelum hari ini
        $presensiList = Presensi::where('sesi_status',
'Ditutup')
            ->where('tanggal', '<', date('Y-m-d'))
            ->where('status', 'Belum Absen')
            ->get();

        foreach ($presensiList as $presensi) {
            $presensi->update([
                'status' => 'Tidak Hadir',
            ]);
        }

        return response()->json([
            'message' => count($presensiList) . ' presensi
berhasil diubah menjadi "Tidak Hadir".'
        ]);
    }
}

```

Lampiran 10 Form Uji Plagiasi



INSTITUT TEKNOLOGI NASIONAL MALANG
PERPUSTAKAAN PUSAT

Jln. Bendungan Sigura-gura No.2 Malang 65145
Telp. (0341) 551431 Pcs. 163-146-147 Fax. (0341) 553015 Website: library.itn.ac.id

FORM UJI PLAGIASI UNTUK MAHASISWA

Yang bertandatangan di bawah ini, Mahasiswa Institut Teknologi Nasional Malang:

Nama : MOH. HARIFIN
NIM : 2118131
Fakultas / Jurusan : Teknologi Industri / Teknik Informatika
Email : arifinrx@gmail.com
No. Tlp : 085231735580
Judul/ Jml artikel : Pambangan Aplikasi presensi siswa Dengan Metode Location Based services (LBS) Berbasis Android Di smk AL-Pukuhiyah

Karya ilmiah yang bersangkutan di atas melalui proses cek plagiasi menggunakan aplikasi trunitin dengan hasil kemiripan (Similarity) sebesar.....%

Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana mestinya.

Mahasiswa

MOH. HARIFIN
2118131

Malang, 05.02.2022

Pelaksana,

Retno W. Satrio