**3. METODE PENELITIAN**

* 1. **Perencanaan**

Aplikasi IN OUT merupakan aplikasi absensi karyawan berbasis android ynag menggunakan flutter dan database Firebase.Aplikasi ini memungkinkan admin untuk mendata dan mengenfisensi proses absensi karyawan.

Aplikasi ini dirancang uuntuk membantu pengguna dalam melakukan absensi secara real time dan mudah selain itu memudahkan admin agar absensi dapat berjalan secara terorgaganisir serta memudahkan dalam urusan pendataan.

Terdapat fitur penambahan pengguna (Add User) Fitur ini digunakan oleh admin untuk mendaftarkan data karyawan ke dalam sistem. Karyawan tidak melakukan pendaftaran secara mandiri. Semua data akun dibuat langsung oleh admin. Admin akan mengisi formulir input yang tersedia pada antarmuka sistem, lalu sistem akan menyimpan data tersebut ke dalam database. Karyawan kemudian dapat menggunakan informasi yang diberikan (username dan password) untuk login ke dalam sistem.Untuk user terdapat tombol clock in dan clock out untuk absen. Setiap kali karyawan menekan tombol Clock In atau Clock Out, sistem akan menyimpan waktu absensi ke dalam basis data. Sebagai bentuk validasi kehadiran, sistem menyediakan fitur foto selfie saat karyawan melakukan absensi. Fitur ini berguna untuk mencegah kecurangan seperti titip absen dan meningkatkan akuntabilitas absensi.

* 1. **Analisi Kebutuhan Aplikasi**

1. Perangkat Keras yang Digunakan

Dalam pembuatan aplikasi ini penulisan memerlukan peangkat keras berupa Laptop deangan spesifikasi berikut:

1. Processor:intel(R)core(TM) i7-10510U CPU @ 1.80GHz
2. Ram: 8.00 GB
3. Storage 512 GB SSD
4. VGA: Interl(R) UHF Graphics
5. Perangkat lunak yang digunakan

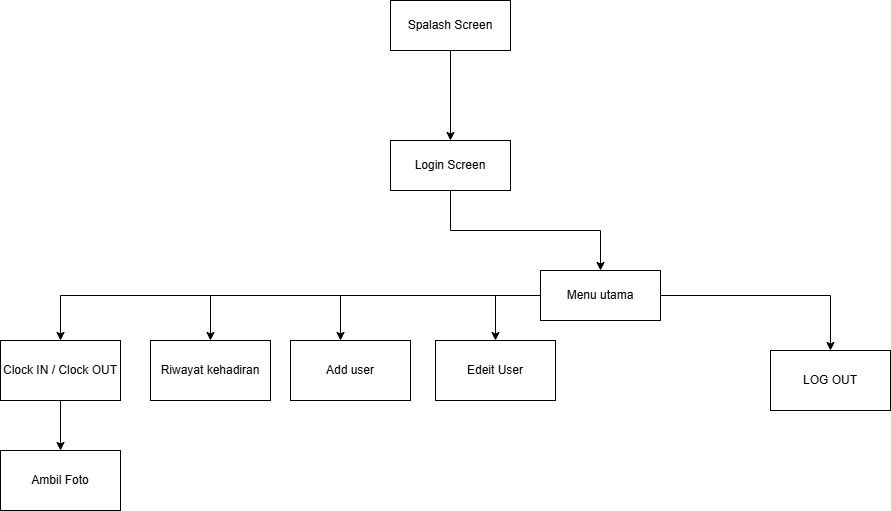
Adapun perangkat lunak yang digunakan untuk menunjang perangkat keras yang menunjang perangkat keras yang telah disebutkan adalah sebagai berikut:

1. Operating System Windows 11 Home 64-Bit
2. Visual studio Code
3. Android studio
   1. **Perancangan Aplikasi**

Bab Perancangan aplikasi ini bertujuan untuk menggambarkan alur pembuatan program dan bagaimana program ini berjalan beserta fitur fitur di dalamnya.

* + 1. **Struktur Navigasi**

Berikut merupakan struktur navigasi dari aplikasi IN OUT yang berjenis campuran linier dan hirarki. Terdapat pada gambar 3.5.

****

Gambar3.5 Struktur Navigasi

* + 1. **Use Case Diagram**

Pada use case daiagram di bawah terdapat relsi interaksi antara user dan sisten pada aplikasi. Terdapat dua actor yang terlibat dalam aplikasi ini yaitu Admin dan User Proses interaksi dimulai dengan membuka aplikasi , kemudia User dan Admin akan diarahkan ke halaman splash screen Setelah 3 detik user dan adamin akan di arahkan ke halaman login. Jika User belum memiliki akin User harus menghubungi admin untuk mendaftarkan User.Jika admin sudah login maka admin dapat memiliki fitur yaitu membuat akun,melihat riwayat kehadiran semua Karyawan,mengedit akun User karyawan. sedangkan User biasa clock in untuk absen datang dan clock out untuk absen pulang disertakan foto ketika sudah mengklik clock in atau clock out dan melihat kapan User clock in dan kapan User clock out.



Gambar 3.6 Use Case

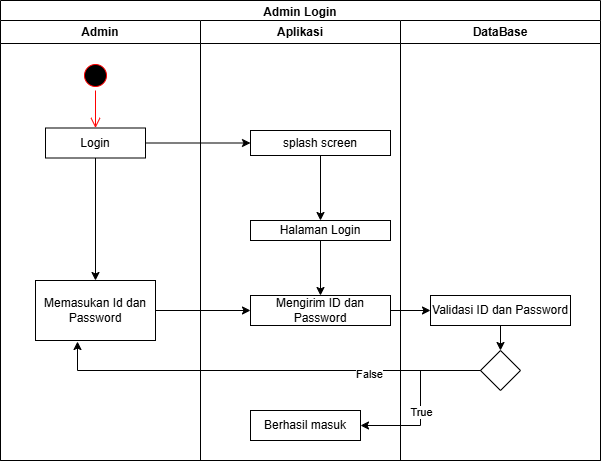
* + 1. **Activity Diagram**

Activity diagram merupakan sebuah runtutan proses dari sistem yang digambarkan secar vertikal .Runtutan proses ini dapat berupa proses bisnis yang terdapat pada dalam sistem.Berikut activity diagram dari aplikasi IN OUT

* + - 1. **Activity Daigram Admin**

Activity diagram dibawah merupakan aktivitas yang dulakuakan oleh admin dan objek dalam sistem.Aplikasi ini memiliki tiga fitur yaitu fitur melihat report karyawan ,mengubah akun karyawan dan mebuat akun karyawan.

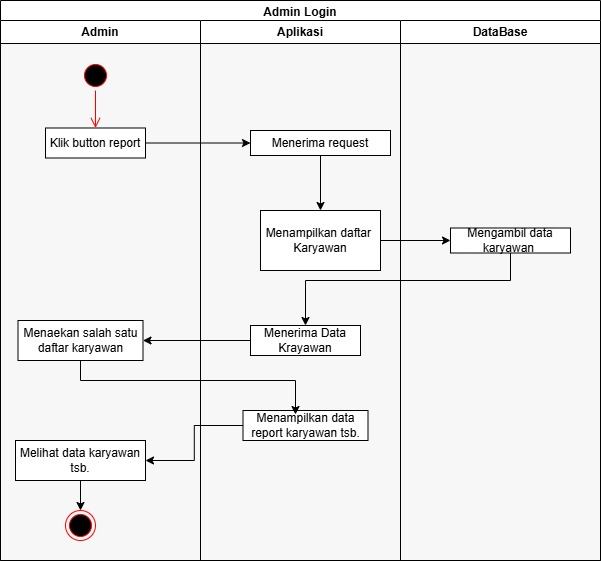
1.**Admin Login**

****

Gambar3.7

Pada gambar 3.7 menunjukan proses login yang dilakukan oleh admin, dimulai oleh admin mengisi ID dan password kemudian admin menekan tombol masik. Setelah itu data email dan password diterima oleh sistem dan mengirimnya ke database untuk diverifikasi jika sudah benar maka admin dapat masuk ke sistem, jika tidak , maka admin tidak dapat masuk ke sistem dan akan teteap di halaman login

* + - 1. **Admin Melihat report**



Gambar3.8 Activity Admin Melihat Report

Pada gambar 3.8 menunjukkan proses melihat report karyawan oleh admin,

dengan cara admin mengklik button report, kemudian aplikasi akan

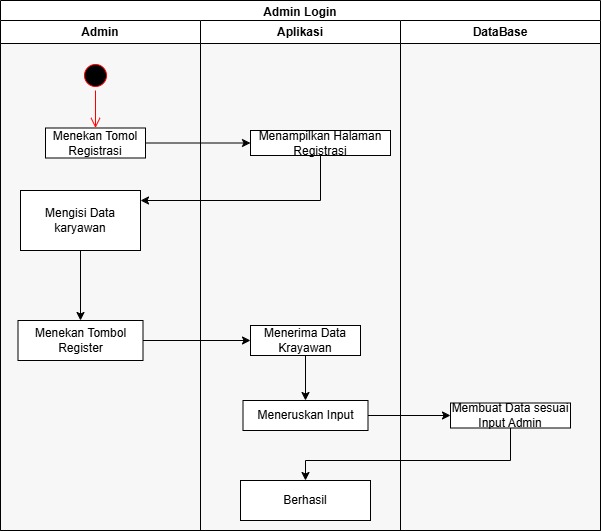
menampilkan daftar karyawan. Setelah itu admin dapat memilih data

karyawan yang ingin dilihat dengan mengklik nama karyawan, data report

diterima oleh sistem dan mengirimnya ke database untuk diambil datanya,

jika sudah maka akan muncul data report karyawan yang telah dipilih.

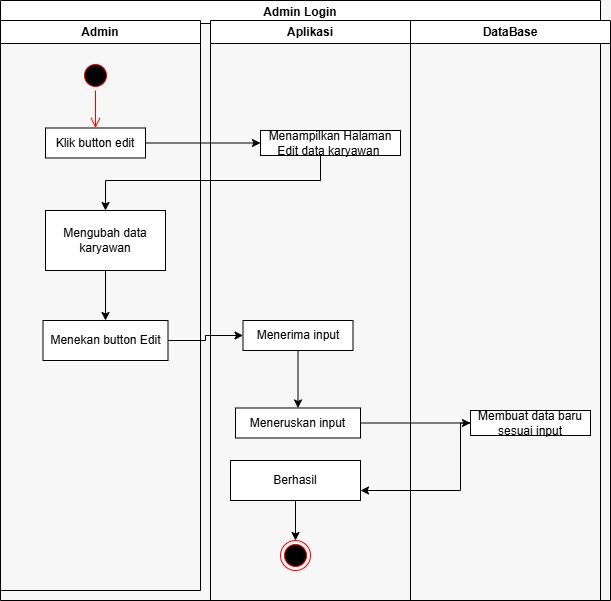
* + - 1. **Admin Create Account**

****

Gambar 3.9 Activity diagram Admin Mebuat Account

Pada gambar 3,9 menunjukan proses admin dalam membuat akun untuk karyawan dengan mengklik button Registrasi ,kemudain aplikasi akan menampilkan form dalam pembuatan akun. Setelah itu admin akan mengisi form tersebut , jika sudah admin akan mengkilik button Registrasi.

1. **Admin Mengedit Akun**



Gambar3.10 Activity Mengedit Akun User

Pada gambar 3.10 menunjukkan proses admin dalam mengubah akun untuk

karyawan, dengan cara admin mengklik button edit account, kemudian

aplikasi akan menampilkan form dalam pengubahan akun. Setelah itu

admin dapat akan mengisi form tersebut, jika sudah admin mengklik button edit.

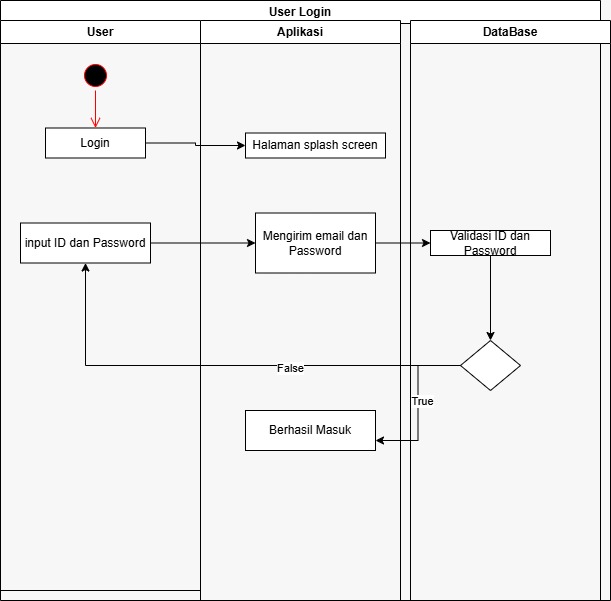
1. **Activity Diagram User**

Activity diagram dibawah merupakan aktivitas yang dilakukan oleh

user dan objek dalam sistem. Aplikasi ini memiliki dua fitur, yaitu fitur

mengunggah foto absen, dan melihat riwyat kehadiran diri sendiri.

1. **User Login**



Gambar3.11 Activity Login Akun User

Pada gambar 3.11 menunjukkan proses login yang dilakukan oleh user,

dimulai oleh user mengisi inputan Usernmae dan password, kemudian user

menekan button masuk. Setelah itu data Username dan password diterima

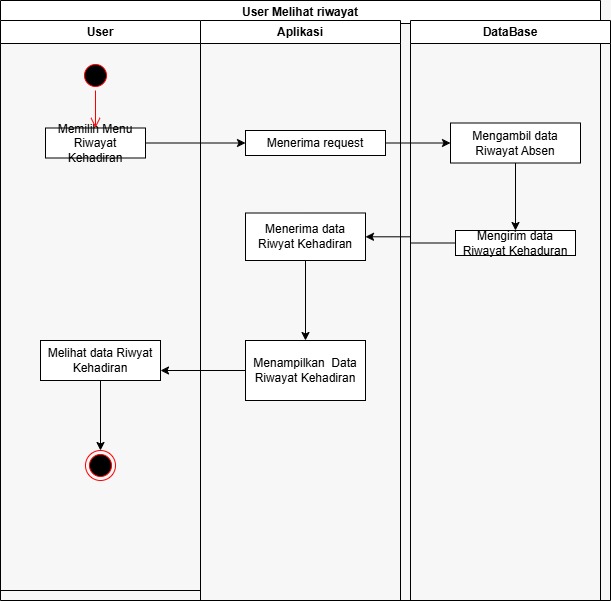
oleh sistem dan mengirimnya ke database untuk diverifikasi, jika sudah

benar maka User dapat masuk ke sistem, jika tidak, maka User tidak

dapat masuk ke sistem dan sistem akan tetap menunjukkan menu

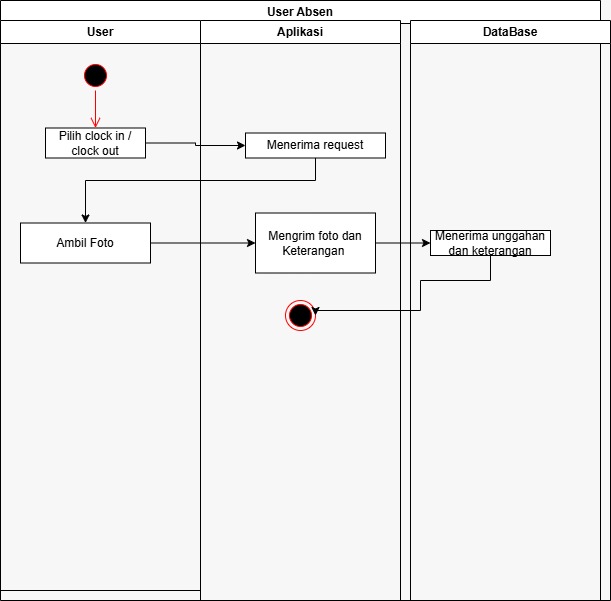
login.

1. **Riwayat User**



Pada gambar 3.12 menunjukan proses melihat riwayat user dengan cara user mengklik button riwayat kemudian aplikasi akan pergi ke halaman riwayat setelah itu user dapat melihat riwayat nya sendiri yang di dapat dari data riwayat kehaduran

1. **User Absen**



Pada gambar 3.13 menunjukkan proses mengunggah Foto absen oleh

karyawan, dengan cara user mengklik button Clock In atau Clock out, kemudian aplikasi

akan otomatis membuka kamera untuk mengunggah foto. Setelah itu user dapat

mengklik simpan untuk menyimpan foto waktu hari dan tanggal yang ingin disimpan

* 1. **Implementasi**

Bagian sub bab ini akan menjelaskan implementasi penyusunan prigram

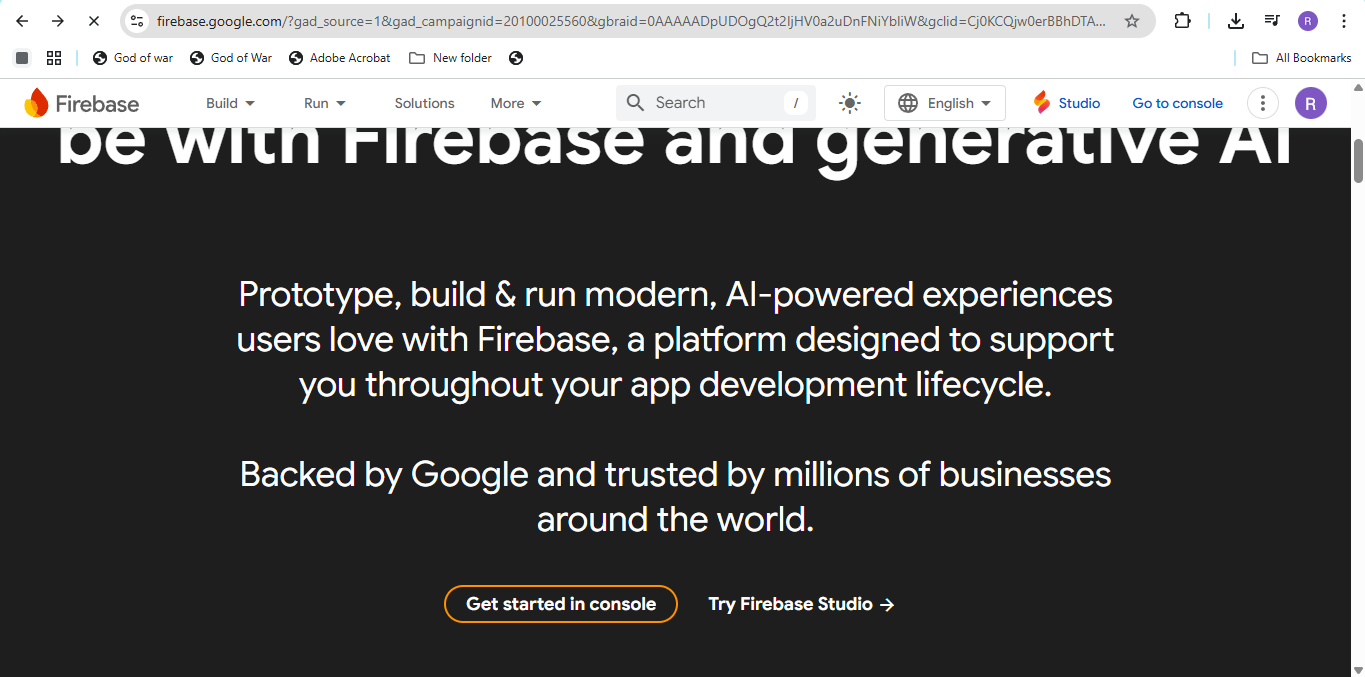
Aplikaisi “IN OUT”.

* + 1. **Implementasi Database dengan Firestore**

Database yang digunakan pada program ini adalah Firebase Firestore.

Untuk membuat database Firebase, buka situs resmi Firebase terlebih dahulu lalu

pilih "Get started" seperti pada gambar 3.14.

****

**Gambar 3.14**

* + 1. **Implementasi Halaman Splash screen**

Halaman pertama dari aplikasi ini merupakan halaman *splash screen* yang berfungsi sebagai tampilan awal sementara. Halaman ini ditampilkan selama 3 detik sebelum sistem secara otomatis mengarahkan pengguna ke halaman login Tampilan Halaman ini persis seperti gambar3.15.

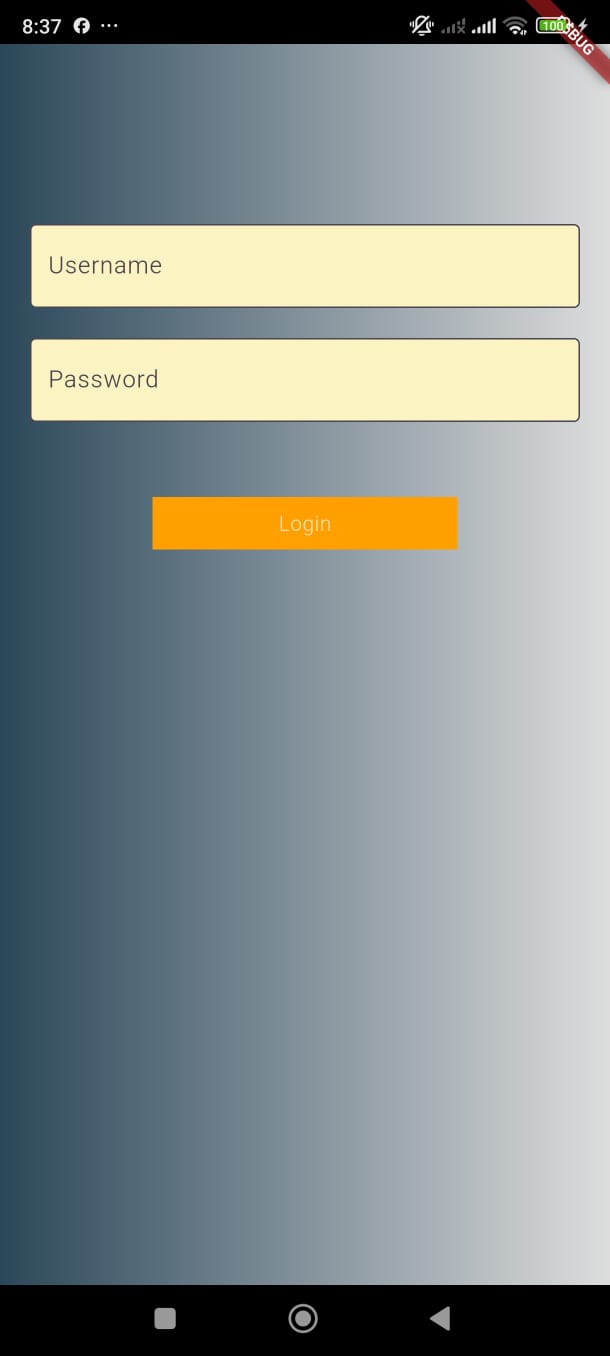
****

Gambar3.15 splash scren

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:penelitian\_fluter/home\_page.dart';  import 'package:penelitian\_fluter/login\_page.dart';  import 'package:penelitian\_fluter/register\_page.dart';  class SplashScreen extends StatefulWidget {  const SplashScreen({super.key});  @override  State<SplashScreen> createState() => \_SplashScreenState();  }  class \_SplashScreenState extends State<SplashScreen> {  void initState() {  // TODO: implement initState  super.initState();  WidgetsBinding.instance.addPostFrameCallback((\_) {  pindahKeHomePage();  });  }  Future<void> pindahKeHomePage() async {  {  await Future.delayed(Duration(seconds: 3));  Navigator.pushReplacement(  context,  MaterialPageRoute(  builder: (context) {  return HomePage();  },  ),  );  }  }  @override  Widget build(BuildContext context) {  return SafeArea(  child: Scaffold(  body: Center(  child: Container(  width: 200,  height: 200,  child: Image(image: AssetImage("assets/Gambar.jpg")),  ),  ),  ),  );  }  } |

**3.4.3 Implementasi Halaman Login screen**

Halaman login screen merupakan halaman untuk memverivikasi pengguna yang ingin masuk ke dalam aplikasi tersebut. Tampilan halaman login dapat dilihat pada gambar 3.16.

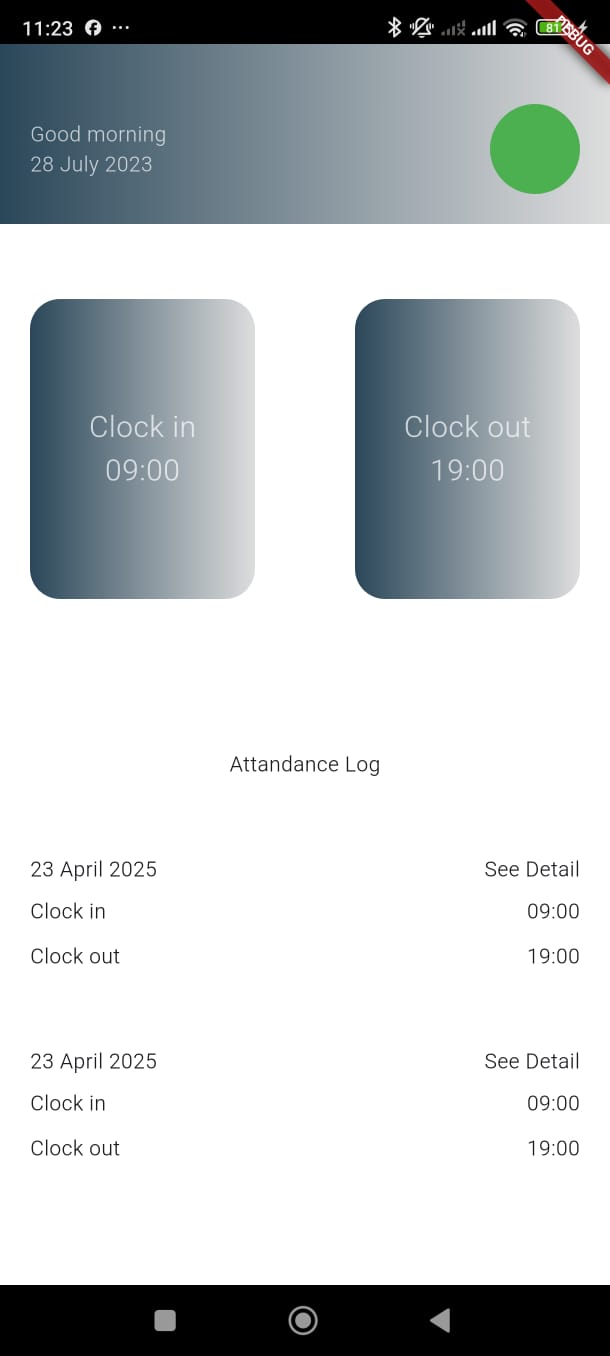
****

Gambar 3.16

|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:penelitian\_fluter/home\_page.dart';  class LoginPage extends StatefulWidget {  const LoginPage({super.key});  @override  State<LoginPage> createState() => \_LoginPageState();  }  class \_LoginPageState extends State<LoginPage> {  TextEditingController controller = TextEditingController();  TextEditingController password = TextEditingController();  @override  Widget build(BuildContext context) {  return SafeArea(  child: Scaffold(  body: Container(  decoration: BoxDecoration(  gradient: LinearGradient(  colors: <Color>[Color(0xff2A4759), Color(0xffDDDDDD)],  ),  ),  child: Container(  margin: EdgeInsets.only(top: 120, right: 20, left: 20),  child: Center(  child: Column(  children: [  TextField(  decoration: InputDecoration(  hintText: 'Username',  fillColor: Color(0xffFBF3C1),  filled: true,  border: OutlineInputBorder(),  ),  controller: controller,  ),  SizedBox(height: 20),  TextField(  obscureText: true,  decoration: InputDecoration(  hintText: 'Password',  fillColor: Color(0xffFBF3C1),  filled: true,  border: OutlineInputBorder(),  ),  controller: password,  ),  child: Container(  margin: EdgeInsets.only(top: 50),  width: MediaQuery.of(context).size.width / 2,  height: 35,  color: Color(0xffFF9F00),  child: Center(  child: Text(  'Login',  style: TextStyle(color: Colors.white70),  ),  ),  ),  ),  ],  ),  ),  ),  ),  ),  );  }  } |

* + 1. **Implementasi HomePage**

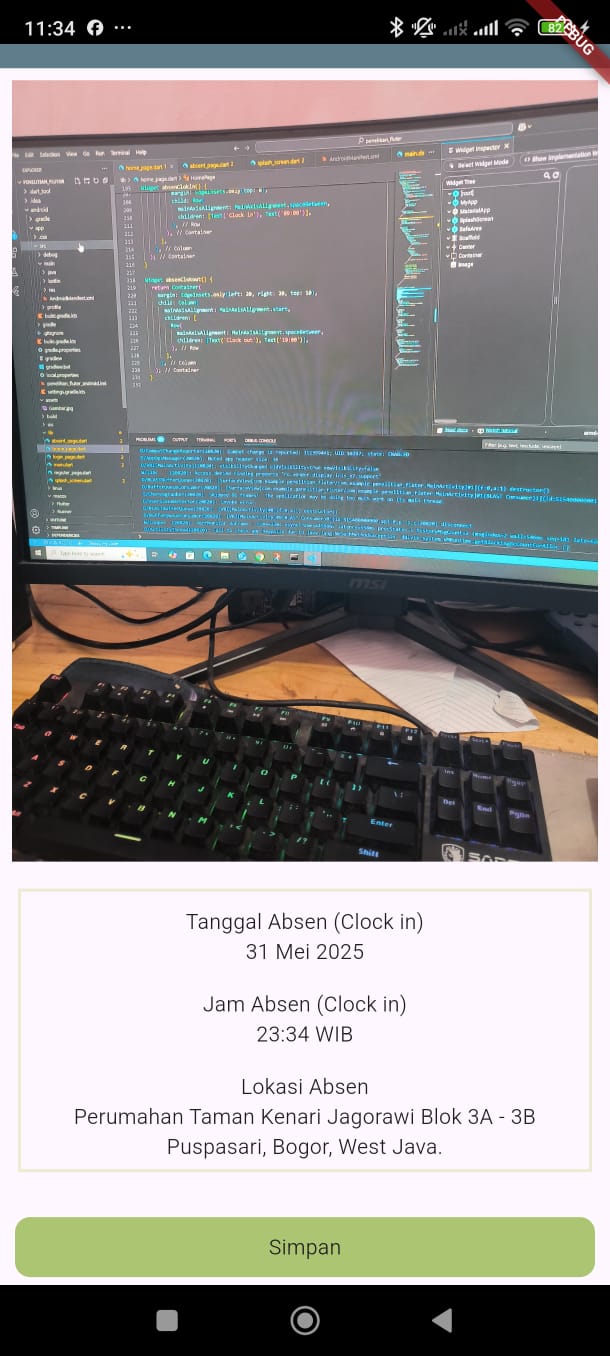
Homepage ada halaman utama untuk aplikasi ini home page bisa di akses apabila User/Admin berhasil login



|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:penelitian\_fluter/absent\_page.dart';  class HomePage extends StatefulWidget {  const HomePage({super.key});  @override  State<HomePage> createState() => \_HomePageState();  }  class \_HomePageState extends State<HomePage> {  @override  Widget build(BuildContext context) {  Widget headBar() {  return Container(  width: MediaQuery.of(context).size.width,  // ignore: sort\_child\_properties\_last  child: Container(  margin: EdgeInsets.only(left: 20, top: 40, bottom: 20),  width: MediaQuery.of(context).size.width,  child: Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [  Column(  crossAxisAlignment: CrossAxisAlignment.start,  children: [  Text('Good morning', style: TextStyle(color: Colors.white70)),  Text('28 July 2023', style: TextStyle(color: Colors.white70)),  ],  ),  Container(  margin: EdgeInsets.only(right: 20),  width: 60,  height: 60,  decoration: BoxDecoration(  shape: BoxShape.circle,  color: Colors.green,  ),  ),  ],  ),  ),  decoration: BoxDecoration(  gradient: LinearGradient(  colors: <Color>[Color(0xff2A4759), Color(0xffDDDDDD)],  ),  ),  );  }  Widget tombol({Color? warna = Colors.black12, String? tulisan}) {  return Container(  width: MediaQuery.of(context).size.width - 20,  height: 50,  color: warna,  margin: EdgeInsets.only(top: 40),  child: Center(child: Text(tulisan ?? 'Tombol1')),  );  }  return MaterialApp(  home: SafeArea(  child: Scaffold(  backgroundColor: Colors.white,  body: ListView(  children: <Widget>[  Column(  children: [  headBar(),  Container(  margin: EdgeInsets.only(left: 20, top: 50, right: 20),  child: Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [  GestureDetector(  onTap: () {  Navigator.push(  context,  MaterialPageRoute(  builder: (context) {  return AbsentPage(flow: 'Clock in');  },  ),  );  },  child: kOtak(),  ),  GestureDetector(  onTap: () {  Navigator.push(  context,  MaterialPageRoute(  builder: (context) {  return AbsentPage(flow: 'Clock Out');  },  ),  );  },  child: kOtak(text: 'Clock out', jam: '19:00'),  ),  ],  ),  ),  // tombol(warna: Colors.red),  // tombol(tulisan: 'Tombol2'),  // tombol(tulisan: 'Tombol3'),  // tombol(tulisan: 'Tombol4'),  Container(  margin: EdgeInsets.only(top: 100, left: 50, right: 50),  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [Text('Attandance Log')],  ),  ),  absenClokin(), absenClokout(),  absenClokin(), absenClokout(),  ],  ),  ],  ),  ),  ),  );  }  }  Widget kOtak({String? text, String? jam}) {  return Container(  child: Column(  mainAxisAlignment: MainAxisAlignment.center,  children: [  Text(  text ?? 'Clock in',  style: TextStyle(color: Colors.white70, fontSize: 20),  ),  Text(  jam ?? '09:00',  style: TextStyle(color: Colors.white70, fontSize: 20),  ),  ],  ),  width: 150,  height: 200,  decoration: BoxDecoration(  gradient: LinearGradient(  colors: <Color>[Color(0xff2A4759), Color(0xffDDDDDD)],  ),  borderRadius: BorderRadius.circular(20),  ),  );  }  Widget kOtakBiru() {  return Container(  width: 180,  height: 200,  decoration: BoxDecoration(  color: Colors.blue,  borderRadius: BorderRadius.circular(20),  ),  );  }  Widget kOtakMerah() {  return Container(  width: 180,  height: 200,  decoration: BoxDecoration(  color: Colors.red,  borderRadius: BorderRadius.circular(20),  ),  );  }  Widget kOtakHijau() {  return Container(  width: 180,  height: 200,  decoration: BoxDecoration(  color: Colors.green,  borderRadius: BorderRadius.circular(20),  ),  );  }  Widget absenClokin() {  return Container(  margin: EdgeInsets.only(top: 50, left: 20, right: 20),  child: Column(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [  Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [Text('23 April 2025'), Text('See Detail')],  ),  Container(  margin: EdgeInsets.only(top: 8),  child: Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [Text('Clock in'), Text('09:00')],  ),  ),  ],  ),  );  }  Widget absenClokout() {  return Container(  margin: EdgeInsets.only(left: 20, right: 20, top: 10),  child: Column(  mainAxisAlignment: MainAxisAlignment.start,  children: [  Row(  mainAxisAlignment: MainAxisAlignment.spaceBetween,  children: [Text('Clock out'), Text('19:00')],  ),  ],  ),  );  } |

* + 1. **Implementasi Absen Page**

Absen page adalah halaman dimana User akan absen Clock in/Clock out setelah user mengklik button Clock in/Clock out aplikasi akan langsung membuka kamera dan minta foto untuk dismpan ke data base bersama waktu hari dan tanggal

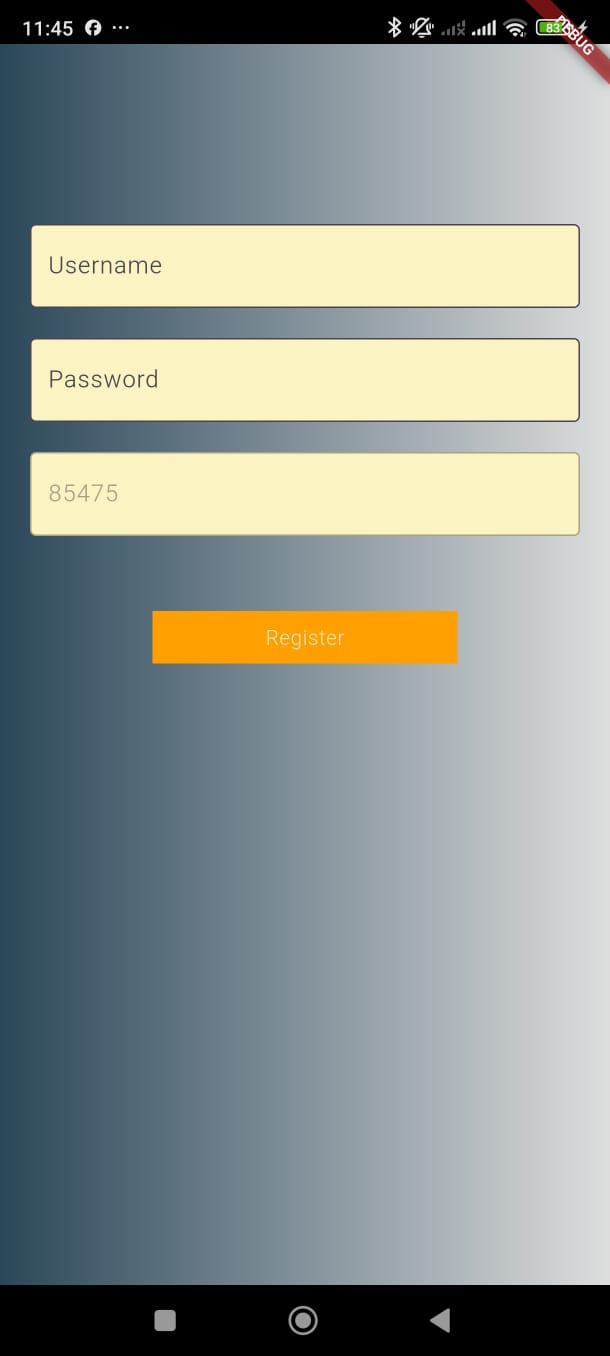


|  |
| --- |
| import 'dart:developer';  import 'dart:io';  import 'package:flutter/material.dart';  import 'package:geolocator/geolocator.dart';  import 'package:image\_picker/image\_picker.dart';  import 'package:intl/intl.dart';  import 'package:intl/date\_symbol\_data\_local.dart';  import 'package:nominatim\_flutter/model/request/reverse\_request.dart';  import 'package:nominatim\_flutter/nominatim\_flutter.dart';  class AbsentPage extends StatefulWidget {  final String flow;  const AbsentPage({required this.flow, super.key});  @override  State<AbsentPage> createState() => \_AbsentPageState();  }  class \_AbsentPageState extends State<AbsentPage> {  bool isLoading = true;  String location = "Unknown";  bool? isInRange;  File? \_image;  @override  void initState() {  gatheringInformation();  super.initState();  }  Future<void> \_getCurrentLocation() async {  bool serviceEnabled;  LocationPermission permission;  // Check if location services are enabled  serviceEnabled = await Geolocator.isLocationServiceEnabled();  if (!serviceEnabled) {  setState(() {  location = "Location services are disabled.";  });  return;  }  // Check permission  permission = await Geolocator.checkPermission();  if (permission == LocationPermission.denied) {  permission = await Geolocator.requestPermission();  if (permission == LocationPermission.denied) {  setState(() {  location = "Location permission denied.";  });  return;  }  }  if (permission == LocationPermission.deniedForever) {  setState(() {  location = "Location permission permanently denied.";  });  return;  }  // Get current position  Position position = await Geolocator.getCurrentPosition(  desiredAccuracy: LocationAccuracy.high,  );  final reverseRequest = ReverseRequest(  lat: position.latitude,  lon: position.longitude,  addressDetails: true,  extraTags: true,  nameDetails: true,  );  final reverseResult = await NominatimFlutter.instance.reverse(  reverseRequest: reverseRequest,  language: 'en-US,en;q=0.5', // Specify the desired language(s) here  );  print(  'longitude latitdue """ ${position.latitude} ${{position.longitude}}',  );  setState(() {  location =  '${reverseResult.address?['road']} ${reverseResult.address?['village']}, ${reverseResult.address?['county']}, ${reverseResult.address?['state']}.';  });  }  Future<bool> isWithin500Meters(double refLat, double refLng) async {  // Get current position  Position position = await Geolocator.getCurrentPosition(  desiredAccuracy: LocationAccuracy.high,  );  // Calculate distance in meters  double distanceInMeters = Geolocator.distanceBetween(  refLat,  refLng,  position.latitude,  position.longitude,  );  print('Distance: $distanceInMeters meters');  if (distanceInMeters >= 500) {  ScaffoldMessenger.of(context).showSnackBar(  SnackBar(  content: Text(  'Lokasi Anda saat ini berada di luar jangkauan 500 meter dari area kantor yang telah ditentukan.',  ),  duration: Duration(seconds: 2),  ),  );  }  // Check if within 500 meters  return distanceInMeters <= 500;  }  Future<void> \_takePhoto() async {  final picker = ImagePicker();  final XFile? pickedFile = await picker.pickImage(  source: ImageSource.camera,  );  if (pickedFile != null) {  setState(() {  \_image = File(pickedFile.path);  });  }  }  Future<void> gatheringInformation() async {  isLoading = true;  await Future.delayed(Duration(seconds: 2));  setState(() {});  await \_takePhoto();  await \_getCurrentLocation();  isInRange = await isWithin500Meters(-6.4810887, 106.8737162);  isLoading = false;  setState(() {});  }  @override  Widget build(BuildContext context) {  return SafeArea(  child: Scaffold(  body: SingleChildScrollView(  child: Column(  crossAxisAlignment: CrossAxisAlignment.center,  children: [  Container(  height: 80,  width: MediaQuery.of(context).size.width,  color: Colors.blueGrey,  child: Center(  child: Text(  widget.flow,  style: TextStyle(color: Colors.white, fontSize: 20),  ),  ),  ),  (isLoading == true)  ? CircularProgressIndicator()  : Column(  children: [  Padding(  padding: const EdgeInsets.all(8.0),  child: Image.file(\_image!),  ),  Container(  decoration: BoxDecoration(  borderRadius: BorderRadius.circular(10),  ),  margin: EdgeInsets.symmetric(horizontal: 12),  width: MediaQuery.of(context).size.width,  child: Container(  margin: EdgeInsets.symmetric(vertical: 10),  child: Column(  crossAxisAlignment: CrossAxisAlignment.center,  children: [  Container(  decoration: BoxDecoration(  border: Border.all(  color: Color(0xffEAEBD0),  width: 2,  ),  ),  child: Column(  children: [  detailDataWidget(  'Tanggal Absen (${widget.flow})',  DateFormat(  'dd MMMM yyyy',  'id\_ID',  ).format(DateTime.now()),  ),  detailDataWidget(  'Jam Absen (${widget.flow})',  '${DateFormat('HH:mm', 'id\_ID').format(DateTime.now())} WIB',  ),  detailDataWidget('Lokasi Absen', location),  ],  ),  ),  ],  ),  ),  ),  Visibility(  visible: isInRange == false,  child: GestureDetector(  onTap: () async {  await gatheringInformation();  },  child: tombolWidget(tulisan: 'Muat Ulang Lokasi'),  ),  ),  Visibility(  visible: isInRange == true,  child: tombolWidget(  tulisan: 'Simpan',  warna: Color(0xffACC572),  ),  ),  SizedBox(height: 20),  ],  ),  ],  ),  ),  ),  );  }  Widget detailDataWidget(String title, String data) {  return Container(  margin: EdgeInsets.only(top: 10, left: 20, right: 20, bottom: 5),  child: Column(  crossAxisAlignment: CrossAxisAlignment.center,  children: [  Text(title, textAlign: TextAlign.center),  Text(data, textAlign: TextAlign.center),  ],  ),  );  }  Widget tombolWidget({Color? warna = Colors.black12, String? tulisan}) {  return Container(  width: MediaQuery.of(context).size.width - 20,  decoration: BoxDecoration(  color: warna,  borderRadius: BorderRadius.circular(10),  ),  height: 40,  margin: EdgeInsets.only(top: 20),  child: Center(child: Text(tulisan ?? 'Tombol1')),  );  }  } |

* + 1. **Implementasi Register Page**

Register page adalah halama untuk Admin menambahkan user baru ke sistem

Register hanya bisa dilakukan oleh Admin saja



|  |
| --- |
| import 'package:flutter/material.dart';  import 'package:penelitian\_fluter/home\_page.dart';  import 'package:penelitian\_fluter/login\_page.dart';  import 'package:penelitian\_fluter/register\_page.dart';  class SplashScreen extends StatefulWidget {  const SplashScreen({super.key});  @override  State<SplashScreen> createState() => \_SplashScreenState();  }  class \_SplashScreenState extends State<SplashScreen> {  void initState() {  // TODO: implement initState  super.initState();  WidgetsBinding.instance.addPostFrameCallback((\_) {  pindahKeHomePage();  });  }  Future<void> pindahKeHomePage() async {  {  await Future.delayed(Duration(seconds: 3));  Navigator.pushReplacement(  context,  MaterialPageRoute(  builder: (context) {  return RegisterPage();  },  ),  );  }  }  @override  Widget build(BuildContext context) {  return SafeArea(  child: Scaffold(  body: Center(  child: Container(  width: 200,  height: 200,  child: Image(image: AssetImage("assets/Gambar.jpg")),  ),  ),  ),  );  }  } |