BAB 2 DAFTAR LAMPIRAN

Lampiran - 1. *Source Code* Absensi.dart

```
import 'dart:async';
import
'package:absensi/bloc/geolocat
ion_bloc.dart';
import
'package:absensi/services/geol
ocator_service.dart';
import
'package:absensi/styles/consta
nt.dart';
import
'package:absensi/widgets/mapAb
sen.dart';
import
'package:flutter/material.dart
';
import
'package:flutter_bloc/flutter_
bloc.dart';
import
'package:google_maps_flutter/g
oogle_maps_flutter.dart';

class AbsensiScreen extends
StatelessWidget {
  const AbsensiScreen({Key?
key, required this.absen}) :
super(key: key);

  final String absen;

  @override
  Widget build(BuildContext
context) {
    final
Completer<GoogleMapController>
_controller = Completer();

    return
BlocBuilder<GeolocationBloc,
GeolocationState>(
      builder: (context,
state) {
        return Scaffold(
          appBar: AppBar(
            title: Text(
              'Lokasi Anda',
              style:
primaryStyle.copyWith(
                fontSize: 22,
                fontWeight:
FontWeight.bold,
              ),
            ),
            actions: <Widget>[
              IconButton(
                icon: const
Icon(

Icons.location_searching,
                  color:
Colors.white,
                ),
                onPressed: ()
async {
                  final
geoDevice =
                      await
GeolocatorService().getDeviceL
ocation();

                  await
GeolocatorService().toMyPositi
on(

LatLng(geoDevice.latitude,
geoDevice.longitude),

_controller,
                  );

                  final double
_distance =
GeolocatorService().inRadius(

state.geolocation,

LatLng(geoDevice.latitude,
geoDevice.longitude));


ScaffoldMessenger.of(context).
showSnackBar(
                    SnackBar(
                      content:
Text(

"Anda berada ${(_distance /
```

```
1000).round()} KM dari lokasi
absen"),
                    ),
                  );
                },
              ),
              IconButton(
                icon: const
Icon(
Icons.refresh,
                  color:
Colors.white,
                ),
                onPressed: ()
async {

ScaffoldMessenger.of(context).
showSnackBar(
                    const
SnackBar(
                      content:
Text('Memperbaharui lokasi
anda'),
                    ),
                  );

                  final
geoDevice =
                      await
GeolocatorService().getDeviceL
ocation();
context.read<GeolocationBloc>(
).add(ChangePosition(

LatLng(geoDevice.latitude,
geoDevice.longitude)));
                    // LatLng(-
3.9843736145260973,
119.65054606109925)));
                  },
                )
              ],
            ),
            body: MapAbsen(
              absen: absen,
              state: state,
              controller:
_controller,
            ),
          );
        },
      );
    }
}
```

Lampiran – 2. Source Code Cuti.dart

```
import
'package:absensi/bloc/cuti_blo
c.dart';
import
'package:absensi/styles/consta
nt.dart';
import
'package:absensi/widgets/cardC
uti.dart';
import
'package:absensi/widgets/modal
Cuti.dart';
import
'package:flutter/material.dart
';
import
'package:flutter_bloc/flutter_
bloc.dart';
import
'package:lottie/lottie.dart';
import
'package:pull_to_refresh/pull_
to_refresh.dart';

class CutiScreen extends
StatelessWidget {
  const CutiScreen({Key? key})
: super(key: key);

  @override
  Widget build(BuildContext
context) {
    final _cutiBloc =
BlocProvider.of<CutiBloc>(cont
ext);

    RefreshController
_refreshController =

RefreshController(initialRefre
sh: false);

    return Scaffold(
      appBar: AppBar(
        title: Text(
          "Cuti",
          style:
primaryStyle.copyWith(
            fontSize: 22,
            fontWeight:
FontWeight.bold,
          ),
        ),
      ),
      floatingActionButton:
FloatingActionButton(
        backgroundColor:
primaryColor,
        child:
Icon(Icons.add),
        onPressed: () {

modalBottomSheetCuti(context);
        },
      ),
      body:
BlocListener<CutiBloc,
CutiState>(
        listener: (context,
state) {
          print("=====>
$state");
          if (state is
CutiSuccess || state is
CutiFailure) {

_refreshController.refreshComp
leted();
          }
        },
        child:
BlocBuilder<CutiBloc,
CutiState>(
          builder: (context,
state) {
            return state is
CutiLoading
                ?
Center(child:
Lottie.asset('assets/animation
s/loading.json'))
                : state is
CutiIsEmpty
                    ?
Center(child: _empty())
                    :
SmartRefresher(

controller:
_refreshController,

onRefresh: () async {

_cutiBloc.add(CutiLoaded());

                              //
await
```

```
Future.delayed(Duration(millis
econds: 1000));
                        },
                        child:
ListView.builder(
                            //
itemCount: 5,

itemCount: state.cutis.length,

itemBuilder: (context, index)
{

return CardCuti(

tanggal:
state.cutis[index].tanggal,

keterangan:
state.cutis[index].keterangan,

diterima:
state.cutis[index].diterima,

);

}),
                                );
                                  },
                                ),
                              ),
                            );
                          }
                        }

                        Widget _empty() {
                          return Column(
                            children: <Widget>[
                              Lottie.asset(

                        'assets/animations/empty.json'
                        ,
                                height: 250,
                              ),
                              Text('Anda belum
                        memiliki data cuti')
                            ],
                          );
                        }
```

Lampiran – 3. Source Code History.dart

```
import
'package:absensi/bloc/histori_
bloc.dart';
import
'package:absensi/styles/consta
nt.dart';
import
'package:absensi/widgets/cardH
istori.dart';
import
'package:flutter/material.dart
';
import
'package:flutter_bloc/flutter_
bloc.dart';
import
'package:lottie/lottie.dart';
import
"package:absensi/app/string_ex
tension.dart";

class HistoriScreen extends
StatelessWidget {
  const HistoriScreen({Key?
key}) : super(key: key);

  @override
  Widget build(BuildContext
context) {
    MediaQueryData mediaQuery
= MediaQuery.of(context);

    List<String> listBulan = [
      'Januari',
      'Februari',
      'Maret',
      'April',
      'Mei',
      'Juni',
      'Juli',
      'Agustus',
      'September',
      'Oktober',
      'November',
      'Desember'
    ];

    final int tahun =
DateTime.now().year;

    List<String> listTahun =
        List.generate(3,
(index) => (tahun -
index).toString());

    List<String> listTotal =
["hadir", "alpa", "izin",
'cuti'];
    List<Color> listColor = [
      primaryColor,
      redColor,
      blueColor,
      orangeColor,
    ];

    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Histori Absen',
          style:
primaryStyle.copyWith(
            fontSize: 22,
            fontWeight:
FontWeight.bold,
          ),
        ),
      ),
      body:
BlocBuilder<HistoriBloc,
HistoriState>(
        builder: (context,
state) {
          print(state);
          return Column(
            children:
<Widget>[
              Padding(
                padding: const
EdgeInsets.all(20),
                child: Row(

mainAxisAlignment:
MainAxisAlignment.spaceAround,
                  children:
<Widget>[

_dropDownButton(
                    value:
state.bulan,
                    list:
listBulan,
                    hint: '-
Pilih Bulan -',
                    width:
(mediaQuery.size.width / 100)
* 45.0,

callback: (String? value)
async {
```

```
context
.read<HistoriBloc>()
.add(ChangeBulan(value ??
''));
                        if
(state.tahun != '')
context.read<HistoriBloc>().ad
d(HistoriLoaded(
bulan: value ?? '', tahun:
state.tahun));
                      },
                    ),
SizedBox(width: 5),
_dropDownButton(
                      value:
state.tahun,
                      list:
listTahun,
                      hint: '-
Pilih Tahun -',
                      width:
(mediaQuery.size.width / 100)
* 40.0,
callback: (String? value) {
context
.read<HistoriBloc>()
.add(ChangeTahun(value ??
''));
                        if
(state.bulan != '')
context.read<HistoriBloc>().ad
d(HistoriLoaded(
bulan: state.bulan, tahun:
value ?? ''));
                      },
                    ),
                  ],
                ),
              ),
              state is
HistoriIsLoaded
                  ?
state.historis.length > 0
                                    ?
Expanded(
child: Column(
children: <Widget>[
GridView.count(
crossAxisCount: 2,
childAspectRatio: (1 / .3),
padding: const
EdgeInsets.symmetric(
horizontal: 20,
),
shrinkWrap: true,
children: List.generate(4,
(index) {
return Padding(
padding: const
EdgeInsets.all(8.0),
child: _infoTotal(
listTotal[index].capitalize()
+
" : " +
state.total
.toJson()[listTotal[index]]
.toString(),
listColor[index],
),
);
}),
),
Expanded(
child: ListView.builder(
```

```
padding: EdgeInsets.all(10),

shrinkWrap: true,

itemCount:
state.historis.length,

itemBuilder: (context, index)
{

return CardHistori(

histori:
state.historis[index],

);

}),

),
                            ],
                          ),
                        )
                      :
_empty()
                : state is
HistoriLoading
                    ?
Lottie.asset('assets/animation
s/loading.json')
                    :
_search(),
            ],
          );
        },
      ),
    );
  }
}

Widget _empty() {
  return Column(
    children: <Widget>[
      Lottie.asset(

'assets/animations/empty.json'
,
        height: 250,
      ),
      Text('Tidak ada data
yang ditemukan')
    ],
  );
}

Widget _search() {
  return Column(
    mainAxisAlignment:
MainAxisAlignment.center,
    crossAxisAlignment:
CrossAxisAlignment.center,
    children: <Widget>[
      Lottie.asset(

'assets/animations/search.json
',
        height: 300,
      ),
      Text('Silahkan pilih
bulan dan tahun terlebih
dahulu')
    ],
  );
}

Widget _dropDownButton({
  required String? value,
  required List<String> list,
  required String hint,
  required double width,
  required Function(String?)
callback,
}) {
  return Container(
    decoration:
kBoxDecorationStyle.copyWith(
      color: secondaryColor,
    ),
    alignment:
Alignment.centerLeft,
    height: 60.0,
    width: width,
    child: Center(
      child: SizedBox(
        width: width,
        child: ButtonTheme(
          alignedDropdown:
true,
          child:
DropdownButtonHideUnderline(
            child:
DropdownButton<String>(
              value: value ==
'' ? null : value,

iconEnabledColor:
Colors.white,
              hint: Text(hint,
                  style:
TextStyle(
```

```
                    fontSize:
15,
                    color:
Colors.white,
                  )),
              onChanged:
callback,
              items:
list.map((String value) {
                return
DropdownMenuItem<String>(
                  value:
value,
                  child: Text(
                    value,
                    style:
TextStyle(

fontSize: 15,
                      color:
Colors.black,

fontWeight: FontWeight.bold,
                    ),
                  ),
                );
              }).toList(),
            ),
                ),
              ),
            ),
          ),
        );
}

Widget _infoTotal(String text,
Color color) {
  return Container(
    decoration: BoxDecoration(
      color: color,
      borderRadius:
BorderRadius.circular(15),
    ),
    child: Center(
      child: Padding(
        padding: const
EdgeInsets.all(8.0),
        child: Text(
          text,
          style:
primaryStyle.copyWith(fontWeig
ht: FontWeight.bold),
        ),
      ),
```

Lampiran – 4. *Source Code* Home.dart

```
import
'package:absensi/bloc/absen_bl
oc.dart';

import
'package:absensi/bloc/authenti
cation_bloc.dart';

import
'package:absensi/bloc/cuti_blo
c.dart';

import
'package:absensi/bloc/geolocat
ion_bloc.dart';

import
'package:absensi/bloc/histori_
bloc.dart';

import
'package:absensi/bloc/izin_blo
c.dart';

import
'package:absensi/bloc/login_bl
oc.dart';

import
'package:absensi/models/absen_
models.dart';

import
'package:absensi/models/respon
seApi_models.dart';

import
'package:absensi/services/geol
ocator_service.dart';

import
'package:absensi/styles/consta
nt.dart';

import
'package:absensi/widgets/heade
rHome.dart';

import
'package:absensi/widgets/infoA
bsen.dart';

import
'package:absensi/widgets/modal
Absen.dart';

import
'package:flutter/material.dart
';

import
'package:flutter_bloc/flutter_
bloc.dart';

import
'package:google_maps_flutter/g
oogle_maps_flutter.dart';

import
'package:lottie/lottie.dart';

import
'package:pull_to_refresh/pull_
to_refresh.dart';

import
'package:shimmer/shimmer.dart'
;



class HomeScreen extends
StatelessWidget {

  const HomeScreen({Key? key})
: super(key: key);



  @override

  Widget build(BuildContext
context) {

    final _authBloc =
BlocProvider.of<Authentication
Bloc>(context);
```

```
    final _absenBloc =
BlocProvider.of<AbsenBloc>(con
text);


    RefreshController
_refreshController =

RefreshController(initialRefre
sh: false);


    int popped = 0;


    return
BlocListener<AbsenBloc,
AbsenState>(
      listener: (context,
state) {
        if (state is
AbsenFailure) {

print(state.message.split('
')[0]);
          if
(state.message.split(' ')[0]
== 'Token') {

BlocProvider.of<LoginBloc>(con
text).add(LoginUnauthorized())
;

_authBloc.add(UserLoggedOut())
;

Navigator.of(context).pushRepl
acementNamed('/login');
          }
        }
      },
      child: WillPopScope(
        onWillPop: () async {
          DateTime initTime =
DateTime.now();
          popped += 1;
          if (popped >= 2)
return true;
          await
ScaffoldMessenger.of(context)
              .showSnackBar(
                SnackBar(
                  content:
Text('Tekan sekali lagi untuk
keluar'),
                  duration:
const Duration(seconds: 2),
                ),
              )
              .closed;

          if
(DateTime.now().difference(ini
tTime) >= Duration(seconds:
2)) {
            popped = 0;
          }
          return false;
        },
        child: Scaffold(

resizeToAvoidBottomInset:
false,
```

```
          appBar: AppBar(
automaticallyImplyLeading:
false,
            elevation: 0,
            title: Text(
              'Absensi',
              style:
primaryStyle.copyWith(
                fontSize: 22,
                fontWeight:
FontWeight.bold,
              ),
            ),
            actions: <Widget>[
              PopupMenuButton(
                icon:
Icon(Icons.more_vert),
                itemBuilder:
(context) => [
PopupMenuItem<int>(
                    value: 0,
                    child:
Text("Tentang"),
                  ),
PopupMenuItem<int>(
                    value: 1,
                    child:
Row(
children: [
                        Icon(
Icons.logout,
color: primaryColor,
                        ),
                        const
SizedBox(
width: 7,
                        ),
Text("Logout")
                      ],
                    ),
                  ),
                ],
                onSelected:
(item) {
                  switch
(item) {
                    case 0:
Navigator.of(context).pushName
d('/tentang');
                      break;
                    case 1:
BlocProvider.of<HistoriBloc>(c
ontext).add(HistoriReset());
BlocProvider.of<IzinBloc>(cont
ext).add(IzinReset());
```

```
BlocProvider.of<CutiBloc>(cont
ext).add(CutiReset());

_authBloc.add(UserLoggedOut())
;

Navigator.of(context).pushRepl
acementNamed('/login');
                      break;
                    default:
                  }
                },
              ),
            ],
          ),
          body:
SmartRefresher(
            controller:
_refreshController,
            onRefresh: ()
async {

_absenBloc.add(AbsenLoaded());


              await
Future.delayed(Duration(millis
econds: 1000));

_refreshController.refreshComp
leted();
            },
            child:
SingleChildScrollView(
              child: Column(
                children:
<Widget>[

BlocBuilder<AuthenticationBloc
, AuthenticationState>(
                    builder:
(context, state) {
                      return
HeaderHome(
                        state:
state,
                      );
                    },
                  ),
                  Padding(
                    padding:
const EdgeInsets.all(15),
                    child:
BlocBuilder<AbsenBloc,
AbsenState>(
                      builder:
(context, state) {

print("===> $state");

                        if
(state is AbsenLoading)

return Column(

crossAxisAlignment:
CrossAxisAlignment.start,

children: [
```

```
Row(
mainAxisAlignment:
MainAxisAlignment.center,
crossAxisAlignment:
CrossAxisAlignment.center,
children: [
_buttonAbsenLoading(),
_buttonAbsenLoading(),
],
),
SizedBox(height: 10),
Row(
mainAxisAlignment:
MainAxisAlignment.center,
crossAxisAlignment:
CrossAxisAlignment.center,
children: [
_buttonAbsenLoading(size: 80),
_buttonAbsenLoading(size: 80),
_buttonAbsenLoading(size: 80),
],
),
SizedBox(height: 40),
Text(
'Jadwal Absen',
style:
kHeaderStyle.copyWith(fontSize
: 18),
),
Text('-'),
SizedBox(height: 20),
_shimmerLoading(),
SizedBox(height: 20),
_shimmerLoading(),
                                ],
                              );
                            else
if (state is AbsenIsLoaded) {
AbsenModel absen =
state.absen;
                              bool
btnDatangDisabled = false;
```

```
                        bool
btnPulangDisabled = false;

                        if
(absen.jamPulang != "") {

DateTime now = new
DateTime.now();

List tanggal = absen.tanggal

.split('-')

.map((e) => int.parse(e))

.toList();

List jam = absen.jamPulang

.split(':')

.map((e) => int.parse(e))

.toList();

DateTime jamPulang = new
DateTime(

tanggal[2],

tanggal[1],

tanggal[0],

jam[0],
                                jam[1],

                                );

                                btnDatangDisabled =
                                jamPulang.isBefore(now) ||

                                absen.waktuDatang.isNotEmpty;

                                btnPulangDisabled =
                                jamPulang.isAfter(now) ||

                                absen.waktuPulang.isNotEmpty;

                                }

                                bool
                                btnDisabled =

                                absen.libur || absen.izin ||
                                absen.cuti;



                                return Column(

                                crossAxisAlignment:
                                CrossAxisAlignment.start,

                                children: [

                                SizedBox(height: 15),

                                Row(

                                mainAxisAlignment:
                                MainAxisAlignment.center,
```

```
crossAxisAlignment:
CrossAxisAlignment.center,

children: <Widget>[

_buttonAbsen(

disabled:

btnDisabled ||
btnDatangDisabled,

size: 25,

icon: Icons.login,

text: 'Absen Datang',

callback: () async {

try {

await GeolocatorService()

.getPermission();

final geoDevice =

await GeolocatorService()

.getDeviceLocation();

context.read<GeolocationBloc>(
).add(

ChangePosition(LatLng(

geoDevice.latitude,

geoDevice.longitude)));

Navigator.of(context)

.pushNamed('/absensiDatang');

} catch (e) {

modalBottomSheetAbsen(

context,

"",

ResponseApiModel(

error: true,

message: e.toString()),

() {});

}

}),

SizedBox(width: 20),

_buttonAbsen(

disabled:
```

```
btnDisabled ||
btnPulangDisabled,

size: 25,

icon: Icons.logout,

text: 'Absen Pulang',

callback: () async {

try {

await GeolocatorService()

.getPermission();


final geoDevice =

await GeolocatorService()

.getDeviceLocation();

context.read<GeolocationBloc>(
).add(

ChangePosition(LatLng(

geoDevice.latitude,

geoDevice.longitude)));

Navigator.of(context)

.pushNamed('/absensiPulang');

} catch (e) {

modalBottomSheetAbsen(

context,

"",

ResponseApiModel(

error: true,

message: e.toString()),

() {});

}

}),

],

),

SizedBox(height: 25),

Row(

mainAxisAlignment:
MainAxisAlignment.center,

crossAxisAlignment:
CrossAxisAlignment.center,
```

```
children: <Widget>[
_buttonAbsen(
size: 15,
icon:
Icons.compare_arrows_rounded,
text: 'Izin',
callback: () async {
Navigator.of(context)
.pushNamed('/izin');
}),
SizedBox(width: 20),
_buttonAbsen(
size: 15,
icon:
Icons.calendar_today_rounded,
text: 'Cuti',
callback: () async {
Navigator.of(context)
.pushNamed('/cuti');
}),
SizedBox(width: 20),
_buttonAbsen(
size: 15,
icon: Icons.auto_stories,
text: 'Histori Absen',
callback: () {
Navigator.of(context)
.pushNamed('/histori');
}),
],
),
SizedBox(height: 40),
Text(
'Jadwal Absen',
style:
kHeaderStyle.copyWith(fontSize
: 18),
),
```

```
Text(state.absen.tanggal),

SizedBox(height: 20),

InfoAbsen(

icon: Icons.login,

tipe: 'datang',

absen: absen,

),

SizedBox(height: 20),

InfoAbsen(

icon: Icons.logout,

tipe: 'pulang',

absen: absen,

)

                            ],

                          );

                        } else
if (state is AbsenFailure) {

return Column(

mainAxisSize:
MainAxisSize.min,

crossAxisAlignment:
CrossAxisAlignment.center,

children: <Widget>[

Lottie.asset(

'assets/animations/error.json'
,

height: 200,

),

Text(

state.message,

style: kHeaderStyle,

)

                                ],

                              );

                            }

                            return
Container();

                          },

                        ),

                      ),

                    ],

                  ),

                ),

              ),
```

```
        ),
      ),
    );
  }
}

Widget _buttonAbsen({
  bool disabled = false,
  required double size,
  required IconData icon,
  required String text,
  required Function callback,
}) {
  return Column(
    children: <Widget>[
      TextButton(
        style:
TextButton.styleFrom(
          backgroundColor:
disabled ? secondaryColor :
primaryColor,
          padding:
EdgeInsets.all(size),
          shape:

RoundedRectangleBorder(borderR
adius:
BorderRadius.circular(15.0)),
        ),
        child: Icon(
          icon,
          color: Colors.white,
          size: 50,
        ),
        onPressed: () =>
disabled ? null : callback(),
      ),
      Text(text)
    ],
  );
}

Widget
_buttonAbsenLoading({double
size = 100}) {
  return Shimmer.fromColors(
    baseColor: secondaryColor,
    highlightColor:
Colors.white,
    child: Padding(
      padding:
EdgeInsets.symmetric(vertical:
15, horizontal: 10),
      child: Container(
        height: size,
        width: size,
        decoration:
BoxDecoration(
          color:
secondaryColor,
          borderRadius:
BorderRadius.circular(15),
        ),
```

```
      ),
    ),
  );
}

Widget _shimmerLoading() {
  return Shimmer.fromColors(
    baseColor: Colors.white,
    highlightColor:
primaryColor,
    child: Container(
      height: 60,
      width: double.infinity,
      decoration:
BoxDecoration(
        color: bgColor,
        borderRadius:
BorderRadius.circular(15),
        boxShadow: [
          BoxShadow(
            color:
Colors.grey,
            offset:
Offset(0.0, 1.0), //(x,y)
            blurRadius: 3.0,
          ),
        ],
      ),
    ),
  );
}
```

Lampiran – 5 *Source Code* Izin.dart

```
import
'package:absensi/bloc/izin_blo
c.dart';

import
'package:absensi/styles/consta
nt.dart';

import
'package:absensi/widgets/cardI
zin.dart';

import
'package:absensi/widgets/modal
Izin.dart';

import
'package:flutter/material.dart
';

import
'package:flutter_bloc/flutter_
bloc.dart';

import
'package:intl/intl.dart';

import
'package:lottie/lottie.dart';



class IzinScreen extends
StatelessWidget {

  const IzinScreen({Key? key})
: super(key: key);



  @override

  Widget build(BuildContext
context) {

    return Scaffold(

      appBar: AppBar(

        title: Text(

          "Izin",

          style:
primaryStyle.copyWith(

            fontSize: 22,

            fontWeight:
FontWeight.bold,

          ),

        ),

      ),

      floatingActionButton:
FloatingActionButton(

        backgroundColor:
primaryColor,

        child:
Icon(Icons.add),

        onPressed: () {


modalBottomSheetIzin(context);

        },

      ),

      body:
BlocBuilder<IzinBloc,
IzinState>(

        builder: (context,
state) {

          print("=====>
$state");

          return state is
IzinLoading
```

```
              ? Center(child:
Lottie.asset('assets/animation
s/loading.json'))

              : state is
IzinIsEmpty

                  ?
Center(child: _empty())

                  :
ListView.builder(

                      //
itemCount: 5,

itemCount: state.izins.length,

itemBuilder: (context, index)
{

                        return
CardIzin(

tanggal: DateFormat('dd MMMM
yyyy kk:mm a').format(

state.izins[index].createdAt!.
add(

Duration(hours: 8),

                            ),

                          ),

keterangan:
state.izins[index].keterangan,
                            );

                        });

                    },

                ),

            );

        }

    }

Widget _empty() {

  return Column(

    children: <Widget>[

      Lottie.asset(

'assets/animations/empty.json'
,

        height: 250,

      ),

      Text('Anda belum
memiliki data izin')

    ],

  );

}
```

Lampiran – 6. *Source Code* Login.dart

```
import
'package:absensi/styles/consta
nt.dart';

import
'package:absensi/widgets/login
Form.dart';

import
'package:flutter/material.dart
';



class LoginScreen extends
StatelessWidget {

  const LoginScreen({Key?
key}) : super(key: key);



  @override

  Widget build(BuildContext
context) {

    int popped = 0;



    return WillPopScope(

      onWillPop: () async {

        DateTime initTime =
DateTime.now();

        popped += 1;

        if (popped >= 2)
return true;

        await
ScaffoldMessenger.of(context)

            .showSnackBar(

              SnackBar(

                content:
Text('Tekan sekali lagi untuk
keluar'),

                duration:
const Duration(seconds: 2),

              ),

            )

            .closed;



        if
(DateTime.now().difference(ini
tTime) >= Duration(seconds:
2)) {

          popped = 0;

        }

        return false;

      },

      child: Scaffold(

        body: Stack(

          children: <Widget>[

            Container(

              height:
double.infinity,

              width:
double.infinity,

              decoration:
BoxDecoration(

                gradient:
LinearGradient(

                  begin:
Alignment.topCenter,

                  end:
Alignment.bottomCenter,

                  colors: [


primaryColor,


secondaryColor,


dangerColor,
```

```
                  ],
                ),
              ),
            ),
            Container(
              height:
double.infinity,
              child:
SingleChildScrollView(
                physics:
AlwaysScrollableScrollPhysics(
),
                padding:
EdgeInsets.symmetric(
                  horizontal:
40.0,
                  vertical:
200.0,
                ),
                child: Column(

mainAxisAlignment:
MainAxisAlignment.center,
                  children:
<Widget>[
                    Image(
                      image:
AssetImage("assets/images/Abse
nsi Karyawan.png"),
                    ),
                    //
SizedBox(height: 20),
                    // Text(
                    //
"Absensi Karyawan",
                    //
textAlign: TextAlign.center,
                    //
style: primaryStyle.copyWith(
                    //
fontSize: 30.0,
                    //
fontWeight: FontWeight.bold,
                    //   ),
                    // ),

SizedBox(height: 30.0),

LoginForm()
                  ],
                ),
              ),
            )
          ],
        ),
      ),
    );
  }
}
```

Lampiran – 7. *Source Code* Tentang.dart

```
import 'package:absensi/styles/constant.dart';

import 'package:flutter/material.dart';



class TentangScreen extends StatelessWidget {

  const TentangScreen({Key? key}) : super(key: key);



  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text(

          "Tentang",

          style: primaryStyle.copyWith(

            fontSize: 22,

            fontWeight: FontWeight.bold,

          ),

        ),

      ),

      body: Container(

        color: Colors.white,

        child: Padding(

          padding: const EdgeInsets.symmetric(vertical: 20, horizontal: 30),

          child: Column(


crossAxisAlignment: CrossAxisAlignment.start,

            children: [

              SizedBox(height: 20),

              Center(

                child: Image(

                  image: AssetImage("assets/images/logo.png"),

                  width: 150,

                ),

              ),

              SizedBox(height: 20),

              Center(

                child: Text(

                  'Aplikasi Absensi Karyawan Menggunakan Geolocation dan Finger Print Berbasis Android',

                  textAlign: TextAlign.center,

                  style: primaryStyle.copyWith(

                    fontSize: 18,
```

```
                    color:
Colors.black87,

fontWeight: FontWeight.bold,
                  ),
                ),
              ),
              SizedBox(height:
30),
              Center(
                child: Text(
                  'Anggita
Arfina Arfah \n217280058',
                  style:
TextStyle(
                    fontSize:
20,

fontWeight: FontWeight.bold,
                  ),
                  textAlign:
TextAlign.center,
                ),
              ),
              SizedBox(height:
35),
              Column(

crossAxisAlignment:
CrossAxisAlignment.start,
                children:
<Widget>[
_dosenInfo("Pembimbing", "1",
"Muh. Zainal, ST., MT"),

SizedBox(height: 10),
                  _dosenInfo(

"Pembimbing", "2", "Ir. Untung
Suwardoyo, S.Kom, MT"),

SizedBox(height: 10),

_dosenInfo("Penguji", "1",
"Ahmad Selao, STP., M.Sc."),

SizedBox(height: 10),

_dosenInfo("Penguji", "2",
"Masnur, ST., M.Kom"),
                ],
              ),
            ],
          ),
        ),
      ),
    );
  }
}

Widget _dosenInfo(String
dosen, String angka, String
nama) {
  return Row(
    children: [
```

```
      Text("$dosen $angka"),

      SizedBox(width: dosen ==
"Pembimbing" ? 20 : 52),

      Text(":"),

      SizedBox(width: 20),

      Expanded(child:
Text(nama)),

    ],

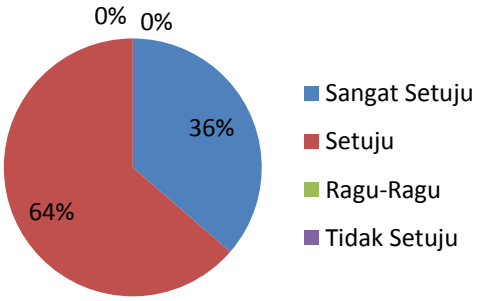
  );

}
```

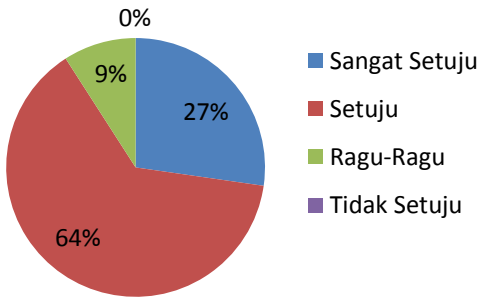
Lampiran – 8. Rekapitulasi Hasil Kuesioner

Pernyataan	Sangat Setuju	Setuju	Ragu-Ragu	Tidak Setuju
Aplikasi Absensi yang di usulkan sangat mudah digunakan dan tidak memerlukan sosialisasi cara penggunaan	4	7	0	0
Pendeteksi lokasi pada aplikasi, sangat akurat	3	7	1	0
Perekapan data absensi menjadi lebih cepat dan tepat dengan menggunakan aplikasi absensi karyawan yang di usulkan	6	5	0	0
Penetapan waktu absen pada aplikasi, sesuai dengan SOP Perusahaan	3	7	1	0
Maps dalam aplikasi absensi ini, dapat dengan tepat mendeteksi lokasi keberadaan saya	3	5	3	0
Penggunaan aplikasi absensi ini sangat meng efesienkan waktu	6	5	0	0
Aplikasi absensi ini memudahkan karyawan dalam melakukan absensi	8	3	0	0

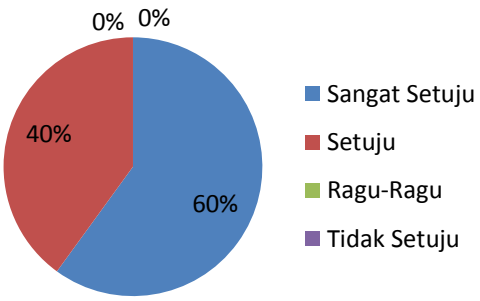
Aplikasi Absensi yang di usulkan sangat mudah digunakan dan tidak memerlukan sosialisasi cara penggunaan



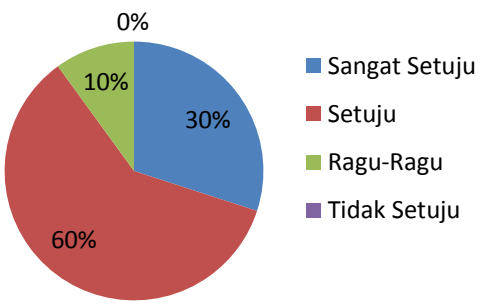
Pendeteksi lokasi pada aplikasi, sangat akurat



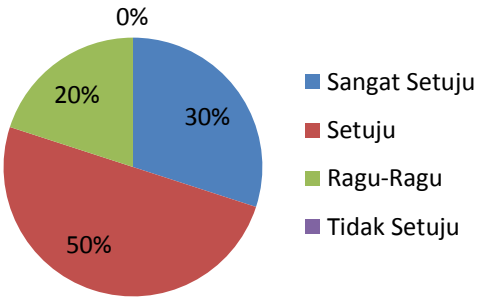
Perekapan data absensi menjadi lebih cepat dan tepat dengan menggunakan aplikasi absensi karyawan yang di usulkan



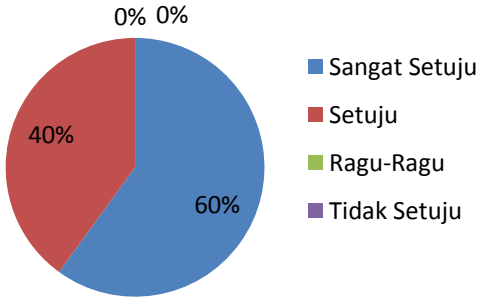
Penetapan waktu absen pada aplikasi, sesuai dengan SOP Perusahaan

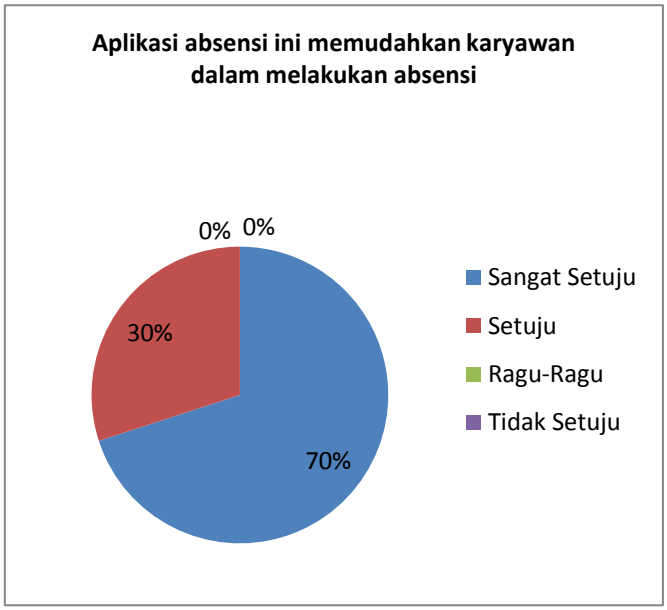


Maps dalam aplikasi absensi ini, dapat dengan tepat mendeteksi lokasi keberadaan saya



Penggunaan aplikasi absensi ini sangat meng efesienkan waktu




Aplikasi absensi ini memudahkan karyawan dalam melakukan absensi
0%
0%
30%
70%
Sangat Setuju
Setuju
Ragu-Ragu
Tidak Setuju