LAMPIRAN

Lampiran-1 Listing program aplikasi main.dart menggunakan *visual studio code*

```
import 'dart:async';

import
'package:flutter/material.dart
';

import
'package:monitor_sampah/home.d
art';

import
'package:firebase_core/firebas
e_core.dart';

import
'package:flutter/services.dart
';

Future<void> main() async {

WidgetsFlutterBinding.ensureIn
itialized();

SystemChrome.setSystemUIOverla
yStyle(const
SystemUiOverlayStyle(

    statusBarColor:
Color(0xFFB71C1C),

    statusBarIconBrightness:
Brightness.light,

    // transparent status bar

  ));

  await
Firebase.initializeApp();

  runApp(

    MaterialApp(

        title:          'TRASH
MONITORING',

        theme: ThemeData(

          fontFamily:
'Signika',

          primaryColor:
Colors.red.shade900,

          splashColor:
Colors.red.shade900,

          colorScheme:


ColorScheme.fromSwatch().copyW
ith(secondary:
Colors.red.shade900),

        ),

        home:            const
OkeyPage(),


debugShowCheckedModeBanner:
false),

  );

}

class     OkeyPage     extends
StatefulWidget {

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

  @override

  _OkeyPageState createState()
=> _OkeyPageState();

}

class  _OkeyPageState  extends
State<OkeyPage> {

  final     myController     =
TextEditingController();

  final    myController2    =
TextEditingController();

  @override

  void dispose() {

    // Clean up the controller
when the widget is disposed.

    myController.dispose();

    myController2.dispose();
```

```
    super.dispose();
  }
  @override
  Widget    build(BuildContext
context) {
    return Scaffold(
      backgroundColor:
Colors.white,
      // appBar: AppBar(
      //        title:   const
Text("Login Page"),
      // ),
      body: Container(
        width:
double.infinity,
        height:
double.infinity,
        decoration:      const
BoxDecoration(
          image:
DecorationImage(
            image:
AssetImage('assets/gambar/bg.p
ng'),
            fit: BoxFit.cover,
          ),
        ),
        child: Center(
          child:
SingleChildScrollView(
            child: Column(
              children:
<Widget>[
                Padding(
                  padding:
const     EdgeInsets.only(top:
60.0, bottom: 10.0),
                  child:
Center(
                    child:
SizedBox(
                      width:
170,
height: 170,
/*decoration: BoxDecoration(
color: Colors.red,
borderRadius:
BorderRadius.circular(50.0)),*
/
                      child:
Image.asset('assets/gambar/air
.png')),
                  ),
                ),
                //       const
Padding(
                //    padding:
EdgeInsets.only(top:      6.0,
bottom: 0),
                //      child:
Center(
                //      child:
SizedBox(
                //
/*decoration: BoxDecoration(
                //
color: Colors.red,
                //
borderRadius:
BorderRadius.circular(50.0)),*
/
                //
child: Text(
                //
"SMART FARM",
```

```
                //
style: TextStyle(
                //
fontWeight: FontWeight.w900,
                //
fontSize: 20,
                //         ),
                //       ),
                //     ),
                //   ),
                // ),
                //       const
Padding(
                //    padding:
EdgeInsets.only(top:      6.0,
bottom: 50.0),
                //      child:
Center(
                //      child:
SizedBox(
                //
/*decoration: BoxDecoration(
                //
color: Colors.red,
                //
borderRadius:
BorderRadius.circular(50.0)),*
/
                //
child: Text(
                //
"BROILER CHICKEN",
                //
style: TextStyle(
                //
fontWeight: FontWeight.w900,
                //
fontSize: 15,
                //         ),
                //       ),
                //     ),
                //   ),
                // ),
                Padding(
                  //padding:
const
EdgeInsets.only(left:15.0,righ
t: 15.0,top:0,bottom: 0),
                  padding:
const
EdgeInsets.symmetric(horizonta
l: 25),
                  child:
TextField(

controller: myController,

decoration:              const
InputDecoration(

border: OutlineInputBorder(),

labelText: 'Username',

hintText:  'Silahkan  masukkan
username anda'),
                  ),
                ),
                Padding(
                  padding:
const EdgeInsets.only(
                      left:
25.0,  right:  25.0,  top:  25,
bottom: 25),
                  //padding:
EdgeInsets.symmetric(horizonta
l: 15),
                  child:
TextField(

controller: myController2,

obscureText: true,
```

```
decoration:              const
InputDecoration(

border: OutlineInputBorder(),

focusColor: Colors.amber,

hoverColor: Colors.amber,

labelText:          'Password',
hintText:  'Silahkan  masukkan
password anda'),

                  ),
                ),
                // TextButton(
                //
onPressed: () {
                //     // TODO
FORGOT  PASSWORD  SCREEN  GOES
HERE
                //   },
                //      child:
const Text(
                //     'Forgot
Password',
                //      style:
TextStyle(color:  Colors.blue,
fontSize: 15),
                //   ),
                // ),
                Container(
                  height: 50,
                  width: 250,
                  decoration:
BoxDecoration(
                      color:
Colors.red.shade900,

borderRadius:
BorderRadius.circular(20)),
                    child:
TextButton(

                      onPressed:
() {
                        if
(myController.text == "ar" &&

myController2.text == "123") {

Navigator.push(context,

MaterialPageRoute(builder: (_)
=> MyHomePage()));
                        }   else
if (myController.text == "" ||

myController2.text == "") {

showDialog(

context: context,

builder: (context) {

return const AlertDialog(

title: Center(

child: Icon(

Icons.dangerous_rounded,

color:     Color.fromARGB(255,
143, 0, 0),

size: 50.0,

),

),

// Retrieve the text the that
user has entered by using the
```

```
// TextEditingController.

content: Text(

"Username atau Password tidak boleh kosong",

textAlign: TextAlign.center,

),
                                );
                              },
                            );
                          } else {

showDialog(

context: context,

builder: (context) {

return const AlertDialog(

title: Center(

child: Icon(

Icons.dangerous_rounded,

color:     Color.fromARGB(255, 143, 0, 0),

size: 50.0,

),

),

// Retrieve the text the that user has entered by using the
```

```
// TextEditingController.

content: Text(

"Username atau Password Anda salah",

textAlign: TextAlign.center,

),
                                );
                              },
                            );
                          }
                        },
                        child:
const Text(
                          'Login',
                          style:
TextStyle(
                            color:
Colors.white,

fontSize: 20,

fontWeight: FontWeight.w700,
                          ),
                        ),
                      ),
                    ),
                    const
SizedBox(
                      height: 130,
                    ),
                    //       const
Text('New     User?     Create
Account')
                  ],
```

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

Lampiran-2 Listing program aplikasi home menggunakan *visual studio code*

```
import 'dart:async';
import 'dart:io';

import
'package:firebase_core/firebas
e_core.dart';
import
'package:firebase_database/fir
ebase_database.dart';
// import
'package:firebase_database/ui/
firebase_animated_list.dart';
import
'package:flutter/foundation.da
rt' show kIsWeb;
import
'package:flutter/material.dart
';
import
'package:percent_indicator/per
cent_indicator.dart';
// import
'package:flutter_switch/flutte
r_switch.dart';
import
'package:intl/intl.dart';
import
'package:webview_flutter/webvi
ew_flutter.dart';
import
'package:flutter_map/flutter_m
ap.dart';
import
'package:flutter_map/plugin_ap
i.dart';
import
'package:latlong2/latlong.dart
';
import
'package:location/location.dar
t';
import
'package:flutter/services.dart
';
import
'package:maps_launcher/maps_la
uncher.dart';

Future<void> main() async {

WidgetsFlutterBinding.ensureIn
itialized();
  await
Firebase.initializeApp();
  runApp(
    const MaterialApp(
      title: 'TRASH MONITOR',
      home: MyHomePage(),
    ),
  );
}

class MyHomePage extends
StatefulWidget {
  static const String route =
'/live_location';

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

  @override
  _MyHomePageState
createState() {
    return _MyHomePageState();
  }
}

class _MyHomePageState extends
State<MyHomePage> {
  final myController =
TextEditingController();
  final myController2 =
TextEditingController();
  final myController3 =
TextEditingController();
  final myController4 =
TextEditingController();
  final myController5 =
TextEditingController();
  final myController6 =
TextEditingController();

  late bool status;
  late bool status2;
  late bool status3;
  late bool status4;

  String _longitude1 = "";
  String _longitude2 = "";
  String _latitude1 = "";
  String _latitude2 = "";
  String _nilai1 = "";
  String _nilai2 = "";

  late DatabaseReference
_nilai1Ref;
  late DatabaseReference
_nilai2Ref;
```

```
  late DatabaseReference
_longitude1Ref;
  late DatabaseReference
_latitude1Ref;
  late DatabaseReference
_longitude2Ref;
  late DatabaseReference
_latitude2Ref;

  late DatabaseReference
_messagesRef;
  late DatabaseReference
_messages2Ref;
  late
StreamSubscription<DatabaseEve
nt> _nilai1Subscription;
  late
StreamSubscription<DatabaseEve
nt> _nilai2Subscription;
  late
StreamSubscription<DatabaseEve
nt> _longitude1Subscription;
  late
StreamSubscription<DatabaseEve
nt> _longitude2Subscription;
  late
StreamSubscription<DatabaseEve
nt> _latitude2Subscription;
  late
StreamSubscription<DatabaseEve
nt> _latitude1Subscription;
  late
StreamSubscription<DatabaseEve
nt> _messagesSubscription;
  late
StreamSubscription<DatabaseEve
nt> _messages2Subscription;

  bool _anchorToBottom =
false;

  final String _kTestKey =
'Hello';
  final String _kTestValue =
'world!';
  FirebaseException? _error;
  bool initialized = false;

  String convertToIdr(dynamic
number, int decimalDigit) {
    NumberFormat
currencyFormatter =
NumberFormat.currency(
      locale: 'id',
      symbol: 'Rp ',
      decimalDigits:
decimalDigit,
    );
    return
currencyFormatter.format(numbe
r);
  }

  String greeting() {
    var hour =
DateTime.now().hour;
    if (hour < 11) {
      return 'Selamat Pagi';
    }
    if (hour < 16) {
      return 'Selamat Siang';
    }
    if (hour < 18) {
      return 'Selamat Sore';
    }
    return 'Selamat Malam';
  }

  String splitt(String
text_split) {
    final h_split =
text_split.split('%');
    return h_split[1];
  }

  String splitt2(String
text_split) {
    final h_split =
text_split.split('%');
    return h_split[2];
  }

  String bulan() {
    var now = DateTime.now();
    var formatter =
DateFormat('MMMM');
    String formattedDate =
formatter.format(now);
    return formattedDate; //
2016-01-25
  }

  LocationData?
_currentLocation;
  late final MapController
_mapController;

  bool _liveUpdate = false;
  bool _permission = false;

  String? _serviceError = '';
```

```
  int interActiveFlags =
InteractiveFlag.all;

  final Location
_locationService = Location();

  @override
  void initState() {
    init();
    super.initState();
    if (Platform.isAndroid)
WebView.platform =
AndroidWebView();
    _mapController =
MapController();
    initLocationService();
  }

  void initLocationService()
async {
    await
_locationService.changeSetting
s(
      accuracy:
LocationAccuracy.high,
      interval: 1000,
    );

    LocationData? location;
    bool serviceEnabled;
    bool serviceRequestResult;

    try {
      serviceEnabled = await
_locationService.serviceEnable
d();

      if (serviceEnabled) {
        final permission =
await
_locationService.requestPermis
sion();
        _permission =
permission ==
PermissionStatus.granted;

        if (_permission) {
          location = await
_locationService.getLocation()
;
          _currentLocation =
location;

_locationService.onLocationCha
nged
.listen((LocationData result)
async {
            if (mounted) {
              setState(() {

_currentLocation = result;

                // If Live
Update is enabled, move map
center
                if
(_liveUpdate) {

_mapController.move(

LatLng(_currentLocation!.latit
ude!,

_currentLocation!.longitude!),

_mapController.zoom);
                }
              });
            }
          });
        }
      } else {
        serviceRequestResult =
await
_locationService.requestServic
e();
        if
(serviceRequestResult) {

initLocationService();
          return;
        }
      }
    } on PlatformException
catch (e) {

debugPrint(e.toString());
      if (e.code ==
'PERMISSION_DENIED') {
        _serviceError =
e.message;
      } else if (e.code ==
'SERVICE_STATUS_ERROR') {
        _serviceError =
e.message;
      }
      location = null;
    }
  }
```

```
  Future<void> init() async {
    _nilai1Ref =
FirebaseDatabase.instance.ref(
'tps1/tinggi');
    _nilai2Ref =
FirebaseDatabase.instance.ref(
'tps2/tinggi');
    _longitude1Ref =
FirebaseDatabase.instance.ref(
'longitude1');
    _longitude2Ref =
FirebaseDatabase.instance.ref(
'longitude2');
    _latitude1Ref =
FirebaseDatabase.instance.ref(
'latitude1');
    _latitude2Ref =
FirebaseDatabase.instance.ref(
'latitude2');

    final database =
FirebaseDatabase.instance;

    _messagesRef =
database.ref('/');
    _messages2Ref =
database.ref('/');


database.setLoggingEnabled(fal
se);

    if (!kIsWeb) {

database.setPersistenceEnabled
(true);

database.setPersistenceCacheSi
zeBytes(10000000);
    }

    if (!kIsWeb) {
      await
_nilai1Ref.keepSynced(true);
    }
    if (!kIsWeb) {
      await
_nilai2Ref.keepSynced(true);
    }

    if (!kIsWeb) {
      await
_longitude1Ref.keepSynced(true
);
    }
    if (!kIsWeb) {
      await
_longitude2Ref.keepSynced(true
);
    }
    if (!kIsWeb) {
      await
_latitude1Ref.keepSynced(true)
;
    }
    if (!kIsWeb) {
      await
_latitude2Ref.keepSynced(true)
;
    }

    setState(() {
      initialized = true;
    });

    _nilai1Subscription =
_nilai1Ref.onValue.listen(
      (DatabaseEvent event) {
        setState(() {
          _error = null;
          _nilai1 =
(event.snapshot.value ?? 0) as
String;
        });
      },
      onError: (Object o) {
        final error = o as
FirebaseException;
        setState(() {
          _error = error;
        });
      },
    );

    _nilai2Subscription =
_nilai2Ref.onValue.listen(
      (DatabaseEvent event) {
        setState(() {
          _error = null;
          _nilai2 =
(event.snapshot.value ?? 0) as
String;
        });
      },
      onError: (Object o) {
        final error = o as
FirebaseException;
        setState(() {
          _error = error;
        });
      },
    );
```

```
    _longitude1Subscription =
_longitude1Ref.onValue.listen(
      (DatabaseEvent event) {
        setState(() {
          _error = null;
          _longitude1 =
(event.snapshot.value ?? 0) as
String;
        });
      },
      onError: (Object o) {
        final error = o as
FirebaseException;
        setState(() {
          _error = error;
        });
      },
    );

    _longitude2Subscription =
_longitude2Ref.onValue.listen(
      (DatabaseEvent event) {
        setState(() {
          _error = null;
          _longitude2 =
(event.snapshot.value ?? 0) as
String;
        });
      },
      onError: (Object o) {
        final error = o as
FirebaseException;
        setState(() {
          _error = error;
        });
      },
    );

    _latitude1Subscription =
_latitude1Ref.onValue.listen(
      (DatabaseEvent event) {
        setState(() {
          _error = null;
          _latitude1 =
(event.snapshot.value ?? 0) as
String;
        });
      },
      onError: (Object o) {
        final error = o as
FirebaseException;
        setState(() {
          _error = error;
        });
      },
    );

    _latitude2Subscription =
_latitude1Ref.onValue.listen(
      (DatabaseEvent event) {
        setState(() {
          _error = null;
          _latitude2 =
(event.snapshot.value ?? 0) as
String;
        });
      },
      onError: (Object o) {
        final error = o as
FirebaseException;
        setState(() {
          _error = error;
        });
      },
    );

    final messagesQuery =
_messagesRef.limitToLast(10);

    _messagesSubscription =
messagesQuery.onChildAdded.lis
ten(
      (DatabaseEvent event) {
        print('Child added:
${event.snapshot.value}');
      },
      onError: (Object o) {
        final error = o as
FirebaseException;
        print('Error:
${error.code}
${error.message}');
      },
    );

    final messages2Query =
_messages2Ref.limitToLast(10);

    _messages2Subscription =
messages2Query.onChildAdded.li
sten(
      (DatabaseEvent event) {
        print('Child added:
${event.snapshot.value}');
      },
      onError: (Object o) {
        final error = o as
FirebaseException;
        print('Error:
${error.code}
${error.message}');
```

```
      },
    );
  }

  @override
  void dispose() {
    super.dispose();
    myController.dispose();
    myController2.dispose();
    myController3.dispose();
    myController4.dispose();
    myController5.dispose();
    myController6.dispose();


_messagesSubscription.cancel()
;

_messages2Subscription.cancel(
);

_nilai1Subscription.cancel();

_nilai2Subscription.cancel();

_longitude1Subscription.cancel
();

_longitude2Subscription.cancel
();

_latitude1Subscription.cancel(
);

_latitude2Subscription.cancel(
);
  }

  @override
  Widget build(BuildContext
context) {
    LatLng currentLatLng;

    // Until currentLocation
is initially updated, Widget
can locate to 0, 0
    // by default or store
previous location value to
show.
    if (_currentLocation !=
null) {
      currentLatLng =

LatLng(_currentLocation!.latit
ude!,
_currentLocation!.longitude!);
    } else {
      currentLatLng =
LatLng(0, 0);
    }
    if (!initialized) return
Container();

    return Scaffold(
      // appBar: AppBar(
      //   title: const
Text('Flutter Database
Example'),
      // ),
      body: SafeArea(
        child: Column(
          children: [
            Container(
              decoration:
BoxDecoration(
                image: const
DecorationImage(
                    image:
AssetImage('assets/gambar/head
_kuning.png'),
                    fit:
BoxFit.fill),
                color:
Colors.red.shade900,
                borderRadius:
const BorderRadius.only(

bottomRight:
Radius.circular(20.0),

bottomLeft:
Radius.circular(20.0)),
                boxShadow:
const [
                  BoxShadow(
                    color:
Colors.grey,
                    offset:
Offset(0.0, 0.0),

blurRadius: 3.0,

spreadRadius: 3.0,
                    ),
                ],
              ),
              // padding:
EdgeInsets.all(20),
              child:
Container(
                // margin:
EdgeInsets.all(10),
```

```
                padding: const
EdgeInsets.only(
                    top: 5,
left: 15, right: 15, bottom:
15),
                child: Column(
mainAxisAlignment:
MainAxisAlignment.spaceBetween
,
                  children:
<Widget>[
                    Row(
mainAxisAlignment:
MainAxisAlignment.spaceBetween
,
children: [
                      Text(
greeting(),
style: const TextStyle(
fontWeight: FontWeight.w800,
fontSize: 17,
color: Colors.white,
                        ),
                      ),
                      const
Text(
"TRASH MONITORING",
style: TextStyle(
fontWeight: FontWeight.w800,
fontSize: 17,
color: Colors.white),
                        ),
                      ],
                    ),
                    Row(
mainAxisAlignment:
MainAxisAlignment.spaceBetween
,
children: [
                      Row(
children: const [
Icon(
Icons.person,
size: 20,
color: Colors.white,
                                ),
Text(
"SUARDI",
style: TextStyle(
fontSize: 15,
fontStyle: FontStyle.normal,
fontWeight: FontWeight.w800,
color: Colors.white,
),
                                ),
                              ],
                            ),
                            const
Text(
"KOTA PAREPARE",
style: TextStyle(
fontSize: 15,
color: Colors.white,
fontWeight: FontWeight.w800),
                              ),
                            ],
                          ),
                          Row(
mainAxisAlignment:
MainAxisAlignment.spaceBetween
,
children: [
                            Row(
children: const [
```

```
Icon(
Icons.phone,
size: 20,
color: Colors.white,
),
Text(
"085123654987",
style: TextStyle(
fontSize: 15,
fontWeight: FontWeight.w800,
color: Colors.white,
),
),
],
),
const
Text(
"",
style: TextStyle(
fontSize: 15,
color: Colors.white,
fontWeight: FontWeight.w800),
),
],
),
],
),
),
),
Column(
children: [
const
SizedBox(
height: 10,
),
Container(
height:
MediaQuery.of(context).size.he
ight * 0.83,
decoration:
const BoxDecoration(
color:
Color.fromARGB(255, 238, 238,
238),
borderRadius:
BorderRadius.vertical(top:
Radius.circular(30))),
// ignore:
sort_child_properties_last
child:
DefaultTabController(
length: 3,
child:
Column(
children: <Widget>[
TabBar(
padding: const
EdgeInsets.all(0),
labelColor: Colors.black,
labelStyle: const TextStyle(
fontSize: 10,
fontWeight: FontWeight.bold,
),
unselectedLabelColor:
const Color.fromARGB(255, 196,
196, 196),
indicatorSize:
TabBarIndicatorSize.label,
indicator: BoxDecoration(
borderRadius:
BorderRadius.circular(10),
color: Colors.white),
tabs: <Widget>[
SizedBox(
height: 50,
width: 70,
child: Tab(
```

```
child: Column(
mainAxisAlignment:
MainAxisAlignment.center,
children: [
Icon(
Icons.monitor,
color: Colors.red.shade900,
),
const Text("Monitoring"),
],
),
),
),


SizedBox(
height: 50,
width: 70,
child: Tab(
child: Column(
mainAxisAlignment:
MainAxisAlignment.center,
children: [
Icon(
Icons.map_rounded,
color: Colors.red.shade900,
),
const Text(
"MAPS",
),
],
),
),
SizedBox(
height: 50,
width: 70,
child: Tab(
child: Column(
mainAxisAlignment:
MainAxisAlignment.center,
children: [
Icon(
Icons.settings,
color: Colors.red.shade900,
),
const Text(
"Pengaturan",
),
],
),
),
),
),
//
SizedBox(
//
height: 50,
//
width: 70,
//
child: Tab(
//
child: Column(
//
mainAxisAlignment:
MainAxisAlignment.center,
```

```
                                //
children: [                     //
Icon(                           //
Icons.history,                  //
color: Colors.red.shade900,     //
),                              //
Text(                           //
"History",                      //
),                              //
],                              //
),                              //
),                              //
),                                ],
                                ),
Expanded(
child: TabBarView(
children: <Widget>[
ListView(
children: [
Column(
children: [
Container(
decoration: const
BoxDecoration(
color: Color.fromARGB(
255, 238, 238, 238),
),
padding: const
EdgeInsets.only(
top: 10,
bottom: 5,
right: 5,
left: 5,
),
child: Container(
decoration: BoxDecoration(
color: Colors.white,
boxShadow: const [
BoxShadow(
color: Color.fromARGB(
255, 225, 225, 225),
offset: Offset(0.0, 0.0),
blurRadius: 2.0,
spreadRadius: 2.0,
),
],
borderRadius:
BorderRadius.circular(
20) // use instead of
BorderRadius.all(Radius.circul
ar(20))
),
// margin: EdgeInsets.all(10),
padding: const
EdgeInsets.all(15),
child: Column(
children: [
const Row(
mainAxisAlignment:
MainAxisAlignment
```

```
.spaceBetween,
children: [
Text(
"TPS I",
style: TextStyle(
fontSize: 18,
color: Colors.black,
fontWeight:
FontWeight.w800),
),
Icon(Icons.line_weight),
],
),
Row(
mainAxisAlignment:
MainAxisAlignment
.spaceBetween,
children: const [
Text(
"(Ketinggian Sampah)",
style: TextStyle(
fontSize: 10,
color: Colors.black,
fontStyle:
FontStyle.italic,
fontWeight:
FontWeight.normal),
),
],
),
const Padding(
padding: EdgeInsets.only(
bottom: 10)),
Row(
mainAxisAlignment:
MainAxisAlignment.center,
children: [
CircularPercentIndicator(
radius: 60.0,
lineWidth: 12.0,
// animation: true,
// animationDuration: 1000,
percent: (double.parse(
splitt(_nilai1))) /
29,
// percent: 10 / 100,
center: Column(
mainAxisAlignment:
MainAxisAlignment
.center,
children: [
Text(
(double.parse(splitt(
_nilai1)))
.toStringAsFixed(
2),
```

```
style: const TextStyle(
fontSize: 35,
color:
Colors.black,
fontWeight:
FontWeight
.w800),
),
const Text(
"CM (Sensor 1)",
style: TextStyle(
fontSize: 10,
color: Colors.black,
),
),
],
),
circularStrokeCap:
CircularStrokeCap.round,
backgroundColor:
const Color.fromARGB(
255, 225, 225, 225),
progressColor:
Colors.red.shade900,
),

SizedBox(width: 10),

// TPS I sensor 2
CircularPercentIndicator(
radius: 60.0,
lineWidth: 12.0,
// animation: true,
// animationDuration: 1000,
percent: (double.parse(
splitt2(_nilai1))) /
29,
// percent: 10 / 100,
center: Column(
mainAxisAlignment:
MainAxisAlignment
.center,
children: [
Text(
(double.parse(splitt2(
_nilai1)))
.toStringAsFixed(
2),
style: const TextStyle(
fontSize: 35,
color:
Colors.black,
fontWeight:
FontWeight
.w800),
),
const Text(
```

```
"CM (Sensor 2)",
style: TextStyle(
fontSize: 10,
color: Colors.black,
),
),
],
),
circularStrokeCap:
CircularStrokeCap.round,
backgroundColor:
const Color.fromARGB(
255, 225, 225, 225),
progressColor:
Colors.red.shade900,
),
],
),
SizedBox(height: 10),
ElevatedButton(
style:
ElevatedButton.styleFrom(
primary:
Colors.red, // background
onPrimary: Colors
.white, // foreground
),
onPressed: () =>
MapsLauncher.launchCoordinates
(
double.parse(_latitude1),
double.parse(_longitude1),
'TPS I' +
" (" +
((double.parse(splitt(
_nilai1)) +
double.parse(
splitt2(
_nilai1))) /
2)
.toStringAsFixed(
2) +
" CM)"),
child: Text('Open Maps'),
),
],
),
),
),
Container(
decoration: const
BoxDecoration(
color: Color.fromARGB(
255, 238, 238, 238),
),
padding: const
EdgeInsets.only(
top: 10,
```

```
bottom: 5,
right: 5,
left: 5,
),
child: Container(
decoration: BoxDecoration(
color: Colors.white,
boxShadow: const [
BoxShadow(
color: Color.fromARGB(
255, 225, 225, 225),
offset: Offset(0.0, 0.0),
blurRadius: 2.0,
spreadRadius: 2.0,
),
],
borderRadius:
BorderRadius.circular(
20) // use instead of
BorderRadius.all(Radius.circul
ar(20))
),
// margin: EdgeInsets.all(10),
padding: const
EdgeInsets.all(15),
child: Column(
children: [
Row(
mainAxisAlignment:
MainAxisAlignment
.spaceBetween,
children: const [
Text(
"TPS II",
style: TextStyle(
fontSize: 18,
color: Colors.black,
fontWeight:
FontWeight.w800),
),
Icon(Icons
.line_weight_rounded),
],
),
Row(
mainAxisAlignment:
MainAxisAlignment
.spaceBetween,
children: const [
Text(
"(Ketinggian Sampah)",
style: TextStyle(
fontSize: 10,
color: Colors.black,
fontStyle:
FontStyle.italic,
fontWeight:
FontWeight.normal),
```

```
),
],
),

const Padding(
padding: EdgeInsets.only(
bottom: 10)),
Row(
mainAxisAlignment:
MainAxisAlignment.center,
children: [
CircularPercentIndicator(
radius: 60.0,
lineWidth: 12.0,
// animation: true,
// animationDuration: 1000,
percent: (double.parse(
splitt(_nilai2))) /
29,
// percent: 10 / 100,
center: Column(
mainAxisAlignment:
MainAxisAlignment
.center,
children: [
Text(
(double.parse(splitt(
_nilai2)))
.toStringAsFixed(
2),
style: const TextStyle(
fontSize: 35,
color:
Colors.black,
fontWeight:
FontWeight
.w800),
),
const Text(
"CM (Sensor 1)",
style: TextStyle(
fontSize: 10,
color: Colors.black,
),
),
],
),
circularStrokeCap:
CircularStrokeCap.round,
backgroundColor:
const Color.fromARGB(
255, 225, 225, 225),
progressColor:
Colors.red.shade900,
),
```

```
SizedBox(width: 10),

// TPS I sensor 2
CircularPercentIndicator(
radius: 60.0,
lineWidth: 12.0,
// animation: true,
// animationDuration: 1000,
percent: (double.parse(
splitt2(_nilai2))) /
29,
// percent: 10 / 100,
center: Column(
mainAxisAlignment:
MainAxisAlignment
.center,
children: [
Text(
(double.parse(splitt2(
_nilai2)))
.toStringAsFixed(
2),
style: const TextStyle(
fontSize: 35,
color:
Colors.black,
fontWeight:
FontWeight
.w800),
),
const Text(
"CM (Sensor 2)",
style: TextStyle(
fontSize: 10,
color: Colors.black,
),
),
],
),
circularStrokeCap:
CircularStrokeCap.round,
backgroundColor:
const Color.fromARGB(
255, 225, 225, 225),
progressColor:
Colors.red.shade900,
),
],
),
SizedBox(height: 10),
ElevatedButton(
style:
ElevatedButton.styleFrom(
primary:
Colors.red, // background
onPrimary: Colors
```

```
.white, // foreground

),

onPressed: () =>
MapsLauncher.launchCoordinates
(

double.parse(_latitude2),

double.parse(_longitude2),

'TPS II' +

" (" +

((double.parse(splitt(

_nilai2)) +

double.parse(

splitt2(

_nilai2))) /

2)

.toStringAsFixed(

2) +

" CM)"),

child: Text('Open Maps'),

),

// const Padding(

//     padding:
EdgeInsets.only(

//         bottom: 10)),

// Column(

//   mainAxisAlignment:

//
MainAxisAlignment.center,

//   children: <Widget>[

//     Text(

//       "Jumlah Yang harus
dibayar sebesar ",

//       style: const
TextStyle(

//         fontSize: 12,

//         fontWeight:

//
FontWeight.normal,

//         color:
Colors.black,

//       ),

//     ),

//     Text(

//       _nilai1.toString(),

//       style: TextStyle(

//         fontSize: 12,

//         fontWeight:

//
FontWeight.bold,

//         color:
Colors.black,

//       ),

//     ),

//   ],

// ),

],

),

),

),

],

),
```

```
],

), // Container(


//Peta

FlutterMap(

options: MapOptions(

center:
LatLng(_currentLocation!.latit
ude!,

_currentLocation!.longitude!),

// center:
LatLng(_currentLocation!.latit
ude!,

//
_currentLocation!.longitude!),

zoom: 15,

),

// nonRotatedChildren: [

//
AttributionWidget.defaultWidge
t(

//     source:
(double.parse(_latitude1) *

//
double.parse(_longitude1))

//         .toString(),

//     onSourceTapped: null,

//   ),

// ],

children: [

TileLayer(

urlTemplate:

'https://tile.openstreetmap.or
g/{z}/{x}/{y}.png',

userAgentPackageName:
'com.example.app',

),

MarkerLayer(

markers: [

Marker(

point: LatLng(

_currentLocation!.latitude!,

_currentLocation!.longitude!),

width: 80,

height: 80,

builder: (context) => Icon(

Icons.person_pin_circle_sharp,

color: Colors.red,

size: 50.0,

),

),

Marker(

point:
LatLng(double.parse(_latitude1
),

double.parse(_longitude1)),

width: 80,

height: 80,

builder: (context) => Icon(

Icons.pin_drop_rounded,

color: Colors.red,

size: 50.0,
```

```
),
),
Marker(
point:
LatLng(double.parse(_latitude2
),
double.parse(_longitude2)),
width: 80,
height: 80,
builder: (context) => Icon(
Icons.pin_drop_rounded,
color: Colors.red,
size: 50.0,
),
),
],
),
],
),
//  WebView(
//   initialUrl:
'https://google.com',
// ),

//Pengaturan
ListView(
children: [
Column(
children: [
Container(
decoration: const
BoxDecoration(
color: Color.fromARGB(
255, 238, 238, 238),
),
padding: const
EdgeInsets.only(
top: 10,
bottom: 5,
right: 5,
left: 5,
),
child: Container(
decoration: BoxDecoration(
color: Colors.white,
boxShadow: const [
BoxShadow(
color: Color.fromARGB(
255, 225, 225, 225),
offset: Offset(0.0, 0.0),
blurRadius: 2.0,
spreadRadius: 2.0,
),
],
borderRadius:
BorderRadius.circular(
20) // use instead of
BorderRadius.all(Radius.circul
ar(20))
),
```

```
// margin: EdgeInsets.all(10),

padding: const
EdgeInsets.all(20),

child: Column(

children: [

Row(

mainAxisAlignment:

MainAxisAlignment

.spaceBetween,

children: [

const Text(

"Pengaturan Lokasi TPS",

style: TextStyle(

fontSize: 18,

color: Colors.black,

fontWeight:

FontWeight.w700),

),

Icon(

Icons.date_range_rounded,

color: Colors.red.shade900,

),

],

),

],

),

),

),
```

```
// const Padding(

//   padding:
EdgeInsets.all(5),

// ),


Container(

decoration: const
BoxDecoration(

color: Color.fromARGB(

255, 238, 238, 238),

),

padding: const
EdgeInsets.all(5),

child: Container(

decoration: BoxDecoration(

color: Colors.white,

boxShadow: const [

BoxShadow(

color: Color.fromARGB(

255, 225, 225, 225),

offset: Offset(0.0, 0.0),

blurRadius: 2.0,

spreadRadius: 2.0,

),

],

borderRadius:
BorderRadius.circular(

20) // use instead of
BorderRadius.all(Radius.circul
ar(20))

),
```

```
// margin: EdgeInsets.all(10),

padding:

const EdgeInsets.only(top:
20),

child: Column(

children: [

const Text(

"TPS I",

style: TextStyle(

fontSize: 15,

fontStyle: FontStyle.normal,

fontWeight: FontWeight.w800,

color: Colors.black,

),

),

Padding(

padding:

const EdgeInsets.all(5)),

Padding(

// padding: const
EdgeInsets.only(left:15.0,righ
t: 15.0,top:0,bottom: 0),

padding:

const EdgeInsets.symmetric(

horizontal: 25),

child: TextField(

controller: myController3,

decoration: const
InputDecoration(

border:

OutlineInputBorder(),

labelText: 'Latitude',

hintText:

'Silahkan masukkan Latitude
TPS I'),

),

),

Padding(

//padding: const
EdgeInsets.only(left:15.0,righ
t: 15.0,top:0,bottom: 0),

padding: const
EdgeInsets.only(

left: 25.0,

right: 25.0,

top: 25,

bottom: 25),

child: TextField(

controller: myController4,

decoration: const
InputDecoration(

border:

OutlineInputBorder(),

labelText: 'Longitude',

hintText:

'Silahkan masukkan Longitude
TPS II'),

),

),

const Text(

"TPS II",
```

```
style: TextStyle(

fontSize: 15,

fontStyle: FontStyle.normal,

fontWeight: FontWeight.w800,

color: Colors.black,

),

),

Padding(

padding:

const EdgeInsets.all(5)),

Padding(

// padding: const
EdgeInsets.only(left:15.0,righ
t: 15.0,top:0,bottom: 0),

padding:

const EdgeInsets.symmetric(

horizontal: 25),

child: TextField(

controller: myController5,

decoration: const
InputDecoration(

border:

OutlineInputBorder(),

labelText: 'Latitude',

hintText:

'Silahkan masukkan Latitude
TPS II'),

),

),

Padding(

//padding: const
EdgeInsets.only(left:15.0,righ
t: 15.0,top:0,bottom: 0),

padding: const
EdgeInsets.only(

left: 25.0,

right: 25.0,

top: 25,

bottom: 25),

child: TextField(

controller: myController6,

decoration: const
InputDecoration(

border:

OutlineInputBorder(),

labelText: 'Longitude',

hintText:

'Silahkan masukkan Longitude
TPS II'),

),

),

Row(

mainAxisAlignment:

MainAxisAlignment.end,

children: [

Container(

margin:

const EdgeInsets.only(

right: 25),

height: 40,
```

```
width: 100,
decoration: BoxDecoration(
color:
Colors.red.shade900,
borderRadius:
BorderRadius
.circular(5)),
child: TextButton(
onPressed: () {
_messagesRef
.child("latitude1")
.set(myController3
.text);
_messagesRef
.child("longitude1")
.set(myController4
.text);
_messagesRef
.child("latitude2")
.set(myController5
.text);
_messagesRef
.child("longitude2")
.set(myController6
.text);
},
child: const Text(
'SAVE',
style: TextStyle(
color: Colors.white,
fontSize: 18,
fontWeight:
FontWeight.w700,
),
),
),
),
],
),
const Padding(
padding:
EdgeInsets.only(bottom: 25),
),
],
                                    ),
                                  ),
                                ),

                                Container(
                                  decoration:
const BoxDecoration(
                                    color:
Color.fromARGB(
                                      255,
238, 238, 238),
                                  ),
                                  padding:
const EdgeInsets.only(
```

```
                top: 5,
                bottom: 5,
                right: 5,
                left: 5,
              ),
              child:
Container(
              decoration:
BoxDecoration(
                color:
Colors.white,
boxShadow: const [
BoxShadow(
                color:
Color.fromARGB(
255, 225, 225, 225),
                offset:
Offset(0.0, 0.0),
blurRadius: 2.0,
spreadRadius: 2.0,
                ),
              ],
borderRadius:
BorderRadius.circular(
                20) //
use instead of
BorderRadius.all(Radius.circular(20)
)
              ),
// margin:
EdgeInsets.all(10),
              padding:
const EdgeInsets.all(20),
              child:
Column(
                children:
[
                  Row(
mainAxisAlignment:
MainAxisAlignment
.spaceBetween,
children: [
                    const
Text(
"Pengaturan Profile",
style: TextStyle(
fontSize: 18,
color: Colors.black,
fontWeight:
FontWeight.w700),
```

```
                    ),
                    Icon(
Icons.person_pin,
color: Colors.red.shade900,
                    ),
                  ],
                ),
              ],
            ),
          ),
          // const
Padding(
          //   padding:
EdgeInsets.all(5),
          // ),

          Container(
            decoration:
const BoxDecoration(
              color:
Color.fromARGB(
                255,
238, 238, 238),
            ),
            padding:
const EdgeInsets.all(5),
            child:
Container(
              decoration:
BoxDecoration(
                color:
Colors.white,
boxShadow: const [
BoxShadow(
                  color:
Color.fromARGB(
255, 225, 225, 225),
                  offset:
Offset(0.0, 0.0),
blurRadius: 2.0,
spreadRadius: 2.0,
                ),
              ],
borderRadius:
BorderRadius.circular(
                20) //
use instead of
BorderRadius.all(Radius.circular(20)
)
              ),
              // margin:
EdgeInsets.all(10),
              padding:
                const
EdgeInsets.only(top: 20),
```

```
                child:
Column(
                children:
[
Padding(
//padding: const
EdgeInsets.only(left:15.0,right:
15.0,top:0,bottom: 0),
padding:
const EdgeInsets.symmetric(
horizontal: 25),
                child:
TextField(
controller: myController,
decoration: const InputDecoration(
border:
OutlineInputBorder(),
labelText: 'Nama',
hintText:
'Silahkan masukkan Nama Anda'),
                ),
              ),
Padding(
padding: const EdgeInsets.only(
                left:
25.0,
right: 25.0,
                top:
25,
bottom: 25),
//padding:
EdgeInsets.symmetric(horizontal:
15),
                child:
TextField(
controller: myController2,
decoration: const InputDecoration(
border:
OutlineInputBorder(),
focusColor: Colors.amber,
hoverColor: Colors.amber,
labelText: 'No. HP',
hintText:
```

```
'Silahkan masukkan No. HP Anda'),
                                    ),
                                ),
                                //
TextButton(
                                //
onPressed: () {
                                //    //
TODO FORGOT PASSWORD
SCREEN GOES HERE
                                //   },
                                //
child: const Text(
                                //
'Forgot Password',
                                //
style: TextStyle(color: Colors.blue,
fontSize: 15),
                                //   ),
                                // ),
                                Row(

mainAxisAlignment:

MainAxisAlignment.end,

children: [

Container(

margin:

const EdgeInsets.only(

right: 25),

height: 40,

width: 100,

decoration: BoxDecoration(

color:

Colors.red.shade900,

borderRadius:

BorderRadius

.circular(5)),

child: TextButton(

onPressed: () {

_messages2Ref

.child("nama")

.set(myController

.text);

_messages2Ref
```

```
.child("hp")
.set(myController2
.text);
                                    },
child: const Text(
'SAVE',
style: TextStyle(
color: Colors.white,
fontSize: 18,
fontWeight:
FontWeight.w700,
                                      ),
                                    ),
                                  ),
                                ),
                              ],
                            ),
                            const
Padding(
padding:
EdgeInsets.only(bottom: 25),
                                      ),
                                    ],
                                  ),
                                ),
                              ),
                              Container(
                                height: 350,
                                decoration:
const BoxDecoration(
                                  color:
Color.fromARGB(
                                    255,
238, 238, 238),
                                ),
                                padding:
const EdgeInsets.only(
                                  top: 5,
                                  bottom: 5,
                                  right: 5,
                                  left: 5,
                                ),
                              ),
                            ],
                          ),
                        ],
                      ),
                    ],
                  ),
```

```
                ),
              ],
            ),
          ),
          padding: const EdgeInsets.all(20.0),
        ),
        //kelelmbapan


        // Center(
        //   child: _error == null
        //       ? Text(
        //           'Button tapped $_relay3 time${_relay3 == 1 ? '' : 's'}.\n\n'
        //           'This includes all devices, ever.',
        //         )
        //       : Text(
        //           'Error retrieving button tap count:\n${_error!.message}',
        //         ),
        // ),
        // Center(
        //   child: _error == null
        //       ? Text(
        //           'Button tapped $_nilai1 time${_relay3 == 1 ? '' : 's'}.\n\n'
        //           'This includes all devices, ever.',
        //         )
        //       : Text(
        //           'Error retrieving button tap count:\n${_error!.message}',
        //         ),
        // ),
            ],
          ),
        ),
      ),
      // floatingActionButton: FloatingActionButton(
      //   onPressed: _increment,
      //   tooltip: 'Increment',
      //   child: const Icon(Icons.add),
      // ),
    );
  }
}
```

Lampiran-3 Listing program aplikasi map menggunakan *visual studio code*

```
import 'package:flutter/material.dart';
import
'package:flutter_map/flutter_map.dar
t';
import
'package:flutter_map/plugin_api.dart'
;
import
'package:latlong2/latlong.dart';
import
'package:location/location.dart';
import 'package:flutter/services.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends
StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context)
{
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.red,
      ),
      home: const MyHomePage(title:
'Flutter Demo Home Page'),
    );
  }
}

class MyHomePage extends
StatefulWidget {
  static const String route =
'/live_location';

  const MyHomePage({super.key,
required this.title});
  final String title;
  @override
  State<MyHomePage> createState()
=> _MyHomePageState();
}

class _MyHomePageState extends
State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  LocationData? _currentLocation;
  late final MapController
_mapController;

  bool _liveUpdate = false;
  bool _permission = false;

  String? _serviceError = '';

  int interActiveFlags =
InteractiveFlag.all;

  final Location _locationService =
Location();

  @override
  void initState() {
    super.initState();
    _mapController =
MapController();
    initLocationService();
  }

  void initLocationService() async {
    await
_locationService.changeSettings(
      accuracy:
LocationAccuracy.high,
      interval: 1000,
    );
```

```
    LocationData? location;
    bool serviceEnabled;
    bool serviceRequestResult;

    try {
      serviceEnabled = await
_locationService.serviceEnabled();

      if (serviceEnabled) {
        final permission = await
_locationService.requestPermission()
;
        _permission = permission ==
PermissionStatus.granted;

        if (_permission) {
          location = await
_locationService.getLocation();
          _currentLocation = location;

_locationService.onLocationChange
d
              .listen((LocationData result)
async {
            if (mounted) {
              setState(() {
                _currentLocation = result;

                // If Live Update is
enabled, move map center
                if (_liveUpdate) {
                  _mapController.move(

LatLng(_currentLocation!.latitude!,

_currentLocation!.longitude!),
                      _mapController.zoom);
                }
              });
            }
          });
        }
      } else {
        serviceRequestResult = await
_locationService.requestService();
        if (serviceRequestResult) {
          initLocationService();
          return;
        }
      }
    } on PlatformException catch (e) {
      debugPrint(e.toString());
      if (e.code ==
'PERMISSION_DENIED') {
        _serviceError = e.message;
      } else if (e.code ==
'SERVICE_STATUS_ERROR') {
        _serviceError = e.message;
      }
      location = null;
    }
  }

  @override
  Widget build(BuildContext context)
{
    LatLng currentLatLng;

    // Until currentLocation is initially
updated, Widget can locate to 0, 0
    // by default or store previous
location value to show.
    if (_currentLocation != null) {
      currentLatLng =

LatLng(_currentLocation!.latitude!,
_currentLocation!.longitude!);
    } else {
      currentLatLng = LatLng(0, 0);
    }
    return Scaffold(
      appBar: AppBar(
        // Here we take the value from
the MyHomePage object that was
created by
        // the App.build method, and
use it to set our appbar title.
        title: Text(widget.title),
      ),
      body: FlutterMap(
        options: MapOptions(
```

```
        center:
LatLng(_currentLocation!.latitude!,
_currentLocation!.longitude!),
        zoom: 9.2,
      ),
      nonRotatedChildren: [
AttributionWidget.defaultWidget(
        source:        'OpenStreetMap
contributors',
        onSourceTapped: null,
      ),
      ],
      children: [
       TileLayer(
        urlTemplate:
'https://tile.openstreetmap.org/{z}/{x
}/{y}.png',
        userAgentPackageName:
'com.example.app',
       ),
       MarkerLayer(
        markers: [
         Marker(
          point: LatLng(
_currentLocation!.latitude!,
_currentLocation!.longitude!),
          width: 80,
          height: 80,
          builder: (context) => Icon(
Icons.person_pin_circle_sharp,
           color: Colors.red,
           size: 50.0,
          ),
         ),
         Marker(
          point:  LatLng(-4.0215663,
119.6715051),
          width: 80,
          height: 80,
          builder: (context) => Icon(
           Icons.pin_drop_rounded,
           color: Colors.red,
           size: 50.0,
          ),
         ),
        ],
       ),
      ],
     ),
     floatingActionButton:
FloatingActionButton(
      onPressed: _incrementCounter,
      tooltip: 'Increment',
      child: const Icon(Icons.add),
     ),
    );
  }
}
```