

LAMPIRAN



**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang**

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Hirarki Ardi Pratama Wendri
Nim : 1618043
Jurusan : Teknik Informatika S-1
Judul : Penerapan Location Based Services Untuk Pencarian Lokasi
Rapat Menggunakan Metode Design Sprint

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

Hari : Senin
Tanggal : 20 Juli 2020
Nilai : 90 (A)

**Panitia Ujian Skripsi
Ketua Majelis Penguji**

Survo Adi Wibowo,S.T.M.T.
NIP .P.1031100438



**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang**

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Hirarki Ardi Pratama Wendri
Nim : 1618043
Jurusan : Teknik Informatika S-1
Judul : Penerapan Location Based Services Untuk Pencarian
Lokasi Rapat Menggunakan Metode Design Sprint

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

Hari : Senin
Tanggal : 20 Juli 2020
Nilai : 90

Panitia Ujian Skripsi
Dosen Pembimbing I

Joseph Dedy Irawan, S.T, M.T
NIP. 197404162005011002



**PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang**

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Hirarki Ardi Pratama Wendri
Nim : 1618043
Jurusan : Teknik Informatika S-1
Judul : Penerapan Location Based Services Untuk Pencarian
Lokasi Rapat Menggunakan Metode Design Sprint

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

Hari : Senin
Tanggal : 20 Juli 2020
Nilai : 90

Panitia Ujian Skripsi
Dosen Pembimbing II

Ahmad Faisol, S.T, M.T

NIP.P 1031000431



FORMULIR PERBAIKAN SKRIPSI

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

NAMA : Hirarki Ardi Pratama Wendri
NIM : 1618043
JURUSAN : Teknik Informatika S-1
JUDUL : PENERAPAN LOCATION BASED SERVICES UNTUK
PENCARIAN LOKASI RAPAT MENGGUNAKAN
METODE DESIGN SPRINT

No.	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	20 Juli 2020	1. Responden untuk kuesioner ditambah 2. Revisi laporan	

Dosen Penguji I

Febriana Santi W, S.Kom, M.kom

NIP.P 1031000425



FORMULIR PERBAIKAN SKRIPSI

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

NAMA : Hirarki Ardi Pratama Wendri
NIM : 1618043
JURUSAN : Teknik Informatika S-1
JUDUL : PENERAPAN LOCATION BASED SERVICES UNTUK
PENCARIAN LOKASI RAPAT MENGGUNAKAN
METODE DESIGN SPRINT

No.	Penguji	Tanggal	Uraian	Paraf
2.	Penguji II	20 Juli 2020	1. Perbaikan Kesimpulan 2. Pengujian pada location based services perlu dibuat lebih rinci lagi.	

Dosen Penguji II

Renaldi Primaswara P , S.Kom, M.Kom

NIP.P 1031900558



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang

FORMULIR PERBAIKAN SKRIPSI

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

NAMA : Hirarki Ardi Pratama Wendri
NIM : 1618043
JURUSAN : Teknik Informatika S-1
JUDUL : PENERAPAN LOCATION BASED SERVICES UNTUK
PENCARIAN LOKASI RAPAT MENGGUNAKAN METODE
DESIGN SPRINT

No.	Penguji	Tanggal	Uraian
1.	Penguji I	20 Juli 2020	1. Responden untuk kuesioner ditambah 2. Revisi laporan
2.	Penguji II	20 Juli 2020	1. Perbaikan Kesimpulan 2. Pengujian pada location based services perlu dibuat lebih rinci lagi.

Dosen Pembimbing I


Joseph Dedy Irawan, S.T. M.T
NIP. 197404162005011002



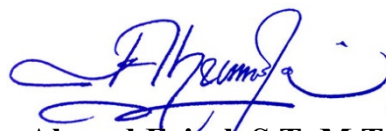
FORMULIR PERBAIKAN SKRIPSI

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

NAMA : Hirarki Ardi Pratama Wendri
NIM : 1618043
JURUSAN : Teknik Informatika S-1
JUDUL : PENERAPAN LOCATION BASED SERVICES UNTUK
PENCARIAN LOKASI RAPAT MENGGUNAKAN METODE
DESIGN SPRINT

No.	Penguji	Tanggal	Uraian
1.	Penguji I	20 Juli 2020	1. Responden untuk kuesioner ditambah 2. Revisi laporan
2.	Penguji II	20 Juli 2020	1. Perbaikan Kesimpulan 2. Pengujian pada location based services perlu dibuat lebih rinci lagi.

Dosen Pembimbing II


Ahmad Faisal, S.T, M.T

NIP.P 1031000431



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

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

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

Malang, 24 April 2020

Nomor : ITN-04-128/III-T.INF/TA/2020
Lampiran : ---
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Josep Dedy Irawan, ST, MT
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

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

Nama : Hirarki Ardi Pratama Wendri
Nim : 1618043
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

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

20 Februari 2020 S/D 20 Agustus 2020

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,



Suryo Adi Wibowo, ST., MT.
NIP.P. 1031100438



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

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

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

Malang, 24 April 2020

Nomor : ITN-04-128/III-T.INF/TA/2020
Lampiran : ---
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Ahmad Faisol, ST, MT
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

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

Nama : Hirarki Ardi Pratama Wendri
Nim : 1618043
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

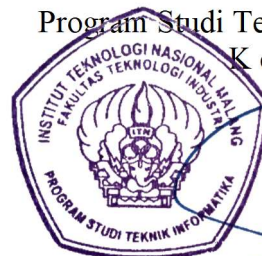
Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

20 Februari 2020 S/D 20 Agustus 2020

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,



Suryo Adi Wilowo, ST., MT.
NIP.P. 1031100438



INSTITUT TEKNOLOGI NASIONAL MALANG

Fakultas Teknologi Industri

Program Studi Teknik Informatika S1

FORMULIR BIMBINGAN SKRIPSI

Nama : HIRARKI ARDI PRATAMA WENDRI
Nim : 1618043
Masa Bimbingan : 6 Bulan
Judul Skripsi : Penerapan Location Based Service untuk Pencarian Lokasi Rapat Menggunakan Metode Design Sprint

No.	Tanggal	Uraian	Paraf Pembimbing
1.	27-04-2020	Tambahkan sistematika penulisan, BAB II Tambahkan teori Pendukung, BAB III Lanjutkan	
2.	1-05-2020	Rapikan penulisan laporan, tambahkan screenshot yang sudah selesai dikerjakan	
3.	2-05-2020	Tambahkan sistematika penulisan, ACC Seminar Progress	
4.	20-05-2020	Perbaiki sistematika penulisan	
5.	25-05-2020	Tambahkan teori pendukung	
6.	10-06-2020	Revisi makalah seminar hasil	
7.	14-06-2020	ACC Semhas	
8.	7-07-2020	Perancangan disain Sprint	
9.	14-07-2020	Perbaiki penulisan laporan	
10.	15-07-2020	ACC Kompre	

Malang, 15 Juli 2020

Dosen Pembimbing

(Joseph Dedy Irawan, S.T.MT.)
NIP. 197404162005011002



INSTITUT TEKNOLOGI NASIONAL MALANG

Fakultas Teknologi Industri

Program Studi Teknik Informatika S1

FORMULIR BIMBINGAN SKRIPSI

Nama : Hirarki Ardi Pratama Wendri
Nim : 1618043
Masa Bimbingan : 20 Februari 2020 s/d 20 Agustus 2020
Judul Skripsi : Penerapan Location Based Service Untuk Pencarian Lokasi Rapat Menggunakan Metode Design Sprint

No.	Tanggal	Uraian	Paraf Pembimbing
1.	25/04/2020	Perbaiki Rumusan masalah nomor 1, langsung fokus pada penerapan metode pada fitur pencarian.	
2.	26/04/2020	Perbaikan laporan seminar progress	
3.	27/04/2020	Pengecekan laporan seminar-progress	
4.	3/06/2020	Bimbingan mengenai location based services	
5.	4/06/2020	Menambahkan fitur absensi rapat	
6.	5/06/2020	Revisi Makalah seminar hasil	
7.	15/06/2020	Acc Makalah Seminar Hasil	
8.	15/07/2020	Revisi Program Semar	
9.	15/07/2020	Revisi Penulisan Halaman Laporan	
10.	15/07/2020	ACC Laporan Skripsi dan Ujian Komprehensif	

Malang, 16 Juli 2020

Dosen Pembimbing

(Ahmad Faisol, S.T, M.T)

NIP.P 1031000431

Nama Bapak / Ibu	Pekerjaan	Pertanyaan 1	Pertanyaan 2.	Pertanyaan 3
Ahmad Faisol	Dosen	Menarik	Ya sangat membantu, untuk bisa mencari jalur tercepat menuju ke lokasi rapat	Absensi rapat dengan fitur face recognition, kelola user profile, penjadwalan rapat dengan metode tertentu
Abdul Wahid	Dosen	Menarik	Iya sangat membantu	Iya
Karina Auliasari	Dosen	Menarik	Saya belum bisa memakai fitur pencarian lokasi, karena pada saat selesai menginstall aplikasi tidak ada panduan atau notifikasi keterangan cara Login, pada saat saya mencoba menginputkan uname&pwd berupa NIP dan berhasil login juga kesulitan pada saat pengisian data tempat rapat aplikasi terus saja memberikan notifikasi belum diisi padahal sudah saya isi.	Mohon ditambahkan video singkat pemakaian aplikasi pada bagian bawah deskripsi pada google play
Ali	Dosen	Menarik	Iya. Tapi, ruang rapat di ITN tidak banyak	ya
Joseph Dedy Irawan	Dosen	Menarik	Ya	Ya
Febriana	Dosen	Biasa	sebaiknya disediakan apk ny shg bs di coba secara langsung	ada menu untuk menambahkan lokasi baru
Rendi	Dosen	Biasa	Cukup membantu untuk dosen baru, Belum pernah mencoba.	Fitur live streaming rapat di semar
Puguh	Karyawan	Menarik	Sangat membantu	iya
Imam Sudjono	Karyawan	Menarik	Ya, sangat membantu dalam melaksanakan rapat	sudah cukup membantu

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?><manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="id.ac.itn.semar">
<!--          The ACCESS_COARSE/FINE_LOCATION permissions are not
required to use          Google Maps Android API v2, but you must
specify either coarse or fine          location permissions for the
'MyLocation' functionality.          -->

<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<application          android:name=".Semar"
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:requestLegacyExternalStorage="true"
android:networkSecurityConfig="@xml/network_security_config"
android:supportsRtl="true"          android:theme="@style/AppTheme">
<!--          The API key for Google Maps-based APIs is defined
as a string resource.          (See the file
"res/values/google_maps_api.xml").          Note that the API
key is linked to the encryption key used to sign the APK.
You need a different API key for each encryption key, including
the release key that is used to          sign the APK for
publishing.          You can define the keys for the debug and
release targets in src/debug/ and src/release/.          -->
<meta-data
android:name="com.google.android.geo.API_KEY"
android:value="@string/google_maps_key" />          <activity
android:name=".Activity.MapsActivity"
android:theme="@style/Theme.MaterialComponents.Light.DarkActionBar
.Bridge"          android:label="@string/title_activity_maps"/>
<service          android:name=".Service.MyMeetingService"
android:exported="false" />          <service
android:name=".Service.MyMeetingService"
android:exported="false" />          <activity
android:name=".Activity.DetailMeetingActivity"
android:launchMode="singleTask" />          <activity
android:name=".Activity.MainActivity"
android:theme="@style/AppTheme.NoActionBar" />          <activity
android:name=".Activity.GenerateMeetingActivity"
android:screenOrientation="portrait"
android:theme="@style/Theme.AppCompat.NoActionBar" />
<activity          android:name=".Activity.LoginActivity"
android:screenOrientation="portrait"
android>windowSoftInputMode="adjustResize"
android:theme="@style/LoginTheme">          <intent-filter>
```

```

<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>          </activity>          <meta-data
android:name="com.onesignal.NotificationOpened.DEFAULT"
android:value="DISABLE" />          <provider
android:name="androidx.core.content.FileProvider"
android:authorities="{applicationId}.provider"
android:exported="false"
android:grantUriPermissions="true">          <meta-data
android:name="android.support.FILE_PROVIDER_PATHS"
android:resource="@xml/provider_paths" />          </provider>
</application></manifest>

```

DetailMeetingActivity.java

```

package id.ac.itn.semar.Activity;

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.ProgressDialog;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProviders;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.location.Location;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Environment;
import android.os.Looper;
import android.provider.MediaStore;

import androidx.annotation.Nullable;
import androidx.core.content.ContextCompat;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.provider.Settings;
import android.text.TextUtils;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.RelativeLayout;
import android.widget.Spinner;

```

```
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;
import androidx.lifecycle.Observer;
import androidx.lifecycle.ViewModelProvider;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.github.irshulx.Editor;
import com.github.irshulx.models.EditorTextStyle;
import
com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.gson.JsonObject;

import butterknife.BindView;
import butterknife.ButterKnife;
import id.ac.itn.semar.Adapter.BerkasAdapter;
import id.ac.itn.semar.Helper.DateFormatter;
import id.ac.itn.semar.Helper.LocationHelper;
import id.ac.itn.semar.Model.FileResponse;
import id.ac.itn.semar.Model.ListBerkasModel;
import id.ac.itn.semar.Model.MeetingModel;

import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Objects;

import id.ac.itn.semar.Helper.SharedPrefManager;
import id.ac.itn.semar.Model.FileResponse;
import id.ac.itn.semar.Model.ListBerkasModel;
import id.ac.itn.semar.Model.MeetingModel;
import id.ac.itn.semar.Model.NotulenModel;
import id.ac.itn.semar.Model.PegawaiModel;
import id.ac.itn.semar.Model.PrepareMeetingModel;
import id.ac.itn.semar.Model.ResultModel;
import id.ac.itn.semar.R;
import id.ac.itn.semar.Rest.ApiClient;
import id.ac.itn.semar.Rest.ApiService;
import id.ac.itn.semar.ViewModel.BerkasViewModel;
```



```

import id.ac.itn.semar.ViewModel.MeetingViewModel;
import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DetailMeetingActivity extends AppCompatActivity
implements View.OnClickListener {
    private String TAG =
DetailMeetingActivity.class.getSimpleName();
    private static final int REQUEST_WRITE_PERMISSION = 786;
    private static final int REQUEST_IMAGE_DATA = 0;
    private static final int REQUEST_FILE_DATA = 1;
    private static final int REQUEST_EDIT_MEETING = 2;
    private static final int PERMISSION_LOCATION_ID = 30;
    private static final int MULTIPLE_PERMISSION_ID = 99;
    private static boolean DATA_UPDATED = false;

    private String idMeeting, userEntri, fileJns, fileInfo,
fileName, mediaPath, fileFormat;
    private String latitude, longitude;
    private double userLat, userLong;
    @BindView(R.id.rv_list_berkas)
RecyclerView rvBerkas;
    private BerkasAdapter adapter;
    private BerkasViewModel berkasViewModel;
    @BindView(R.id.tv_detail_judul)
TextView tvJudul;
    @BindView(R.id.tv_jam_start)
TextView tvJamStart;
    @BindView(R.id.tv_jam_end)
TextView tvJamEnd;
    @BindView(R.id.tv_detail_ruang)
TextView tvRuang;
    @BindView(R.id.map_button)
ImageButton ibMap;
    @BindView(R.id.tv_detail_tanggal)
TextView tvTanggal;
    @BindView(R.id.edt_acara)
EditText edtAcara;
    @BindView(R.id.btnKonfirmasi)
Button btnKonfirmasi;
    EditText edtFileInfo;
    EditText edtFile;
    Spinner sFileJns;
    Spinner sFormatFile;
    Editor htmlViewer, htmlEditor;
    ArrayAdapter<String> itemFileJns, itemFormatFile;
    private static final int BUFFER_SIZE = 1024 * 2;
    ProgressDialog progressDialog;
    boolean editPermission = false;
    private MeetingViewModel meetingVm;

    private LocationHelper locationHelper;
    FusedLocationProviderClient mFusedLocationProviderClient;

    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_detail_meeting);

    Objects.requireNonNull(getSupportActionBar()).setElevation(0);
    meetingVm = new
    ViewModelProvider(this).get(MeetingViewModel.class);
    locationHelper = new
    LocationHelper(getApplicationContext(), this);
    mFusedLocationProviderClient =
    LocationServices.getFusedLocationProviderClient(this);

    getSupportActionBar().setElevation(0);
    meetingVm =
    ViewModelProviders.of(this).get(MeetingViewModel.class);
    meetingVm.getMeetingById().observe(this, getMeeting);
    if (getSupportActionBar() != null) {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setTitle("Detail Rapat");
    }

    ButterKnife.bind(this);
    //Melakukan pengecekan versi android dan perizinan akses
    berkas penyimpanan / storage
    //      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M &&
    ContextCompat.checkSelfPermission(this,
    //      Manifest.permission.READ_EXTERNAL_STORAGE) !=
    PackageManager.PERMISSION_GRANTED) {
    //          //Minta izin
    //          requestPermissions(new
    String[]{android.Manifest.permission.WRITE_EXTERNAL_STORAGE},
    REQUEST_WRITE_PERMISSION);
    //      } else {
    //          prepareView();
    //      }
    if (checkAndRequestPermissions() &&
    Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        requestPermissions(new
    String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
    REQUEST_WRITE_PERMISSION);

    locationHelper.requestPermission(PERMISSION_LOCATION_ID);
        } else {
            prepareView();
            getLastLocation();
        }
    }

    private LocationCallback mLocationCallback = new
    LocationCallback(){
        @Override
        public void onLocationResult(LocationResult
    locationResult) {
            super.onLocationResult(locationResult);
            Location lastLocation =
    locationResult.getLastLocation();
            userLat = lastLocation.getLatitude();
            userLong = lastLocation.getLongitude();
        }
    }

```

```

};

private boolean checkAndRequestPermissions() {
    int filePermission =
ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE);
    int locationPermission =
ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION);

    List<String> listPermissionsNeeded = new ArrayList<>();
    if (locationPermission !=
PackageManager.PERMISSION_GRANTED) {

listPermissionsNeeded.add(Manifest.permission.ACCESS_FINE_LOCATION
);
    }
    if (filePermission != PackageManager.PERMISSION_GRANTED) {

listPermissionsNeeded.add(Manifest.permission.WRITE_EXTERNAL_STORA
GE);
    }
    if (!listPermissionsNeeded.isEmpty()) {
        ActivityCompat.requestPermissions(this,
listPermissionsNeeded.toArray(new
String[listPermissionsNeeded.size()]), MULTIPLE_PERMISSION_ID);
        return false;
    }
    return true;
}

@SuppressWarnings("MissingPermission")
private void getLastLocation(){
    Log.d("ScanAbsenActivity", "getLastLocation: checking
permission");
    if (locationHelper.isLocationEnabled()) {
        Log.d("ScanAbsenActivity", "getLastLocation: location
enabled");

mFusedLocationProviderClient.getLastLocation().addOnCompleteListener
(new OnCompleteListener<Location>() {
        @Override
        public void onComplete(@NonNull Task<Location>
task) {
            Log.d("ScanAbsenActivity", "getLastLocation:
onComplete executed");
            Location location = task.getResult();
            if (location == null) {
                Log.d("ScanAbsenActivity", "onComplete:
location is null");
                requestNewLocationData();
            } else {
                Log.d("ScanAbsenActivity", "onComplete:
location is available -> lat = "
                    + location.getLatitude() + ", long
= " + location.getLongitude());
                userLat = location.getLatitude();
                userLong = location.getLongitude();
            }
        }
    });
}

```

```

        }
    }
    });
    } else { //jika pengaturan location belum dihidupkan
        Log.d("ScanAbsenActivity", "getLastLocation: location
disabled");
        AlertDialog.Builder builder = new
AlertDialog.Builder(this);
        builder.setTitle("Pemberitahuan")
            .setMessage("Anda harus mengaktifkan
pengaturan lokasi terlebih dahulu!")
            .setCancelable(false)
            .setPositiveButton("OK", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialog, int which) {
                    Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                    startActivity(intent);
                }
            });

        AlertDialog dialog = builder.create();
        dialog.show();
    }

    @SuppressWarnings("MissingPermission")
    private void requestNewLocationData() {
        LocationRequest mLocationRequest = new LocationRequest();

mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURAC
Y);

        mLocationRequest.setInterval(0);
        mLocationRequest.setFastestInterval(0);
        mLocationRequest.setNumUpdates(1);

        mFusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);

mFusedLocationProviderClient.requestLocationUpdates(mLocationReque
st, mLocationCallback,
        Looper.myLooper()
    );
    }

    private Observer<MeetingModel> getMeeting = new
Observer<MeetingModel>() {
        @Override
        public void onChanged(MeetingModel meetingModel) {
            if (meetingModel != null) {
                meeting = meetingModel;
                latitude = meeting.getLatitude();
                longitude = meeting.getLongitude();

                setContent(true);
                invalidateOptionsMenu();
            }
        }
    }
}

```

```

        } else {
            Toast.makeText(getApplicationContext(), "Kesalahan
internal, hubungi admin !", Toast.LENGTH_LONG).show();
        }
    }
};

private void prepareView() {
    progressDialog = new ProgressDialog(this);
    progressDialog.setTitle("Menyiapkan data");
    progressDialog.setMessage("Harap tunggu ....");
    progressDialog.setCancelable(false);

    ibMap.setOnClickListener(this);
    btnKonfirmasi.setOnClickListener(this);

    PegawaiModel pegawai =
SharedPrefManager.getInstance(DetailMeetingActivity.this).getPegaw
ai();
    userEntri = pegawai.getIdPeg();
    getMeeting();
}

MeetingModel meeting = null;

private void getMeeting() {
    boolean isNotif = getIntent().getBooleanExtra("isNotif",
false);

    if (isNotif) {
        idMeeting = getIntent().getStringExtra("IdMeeting");
        meetingVm.setMeetingById(idMeeting, userEntri);
    } else {
        Bundle bundle = getIntent().getExtras();
        assert bundle != null;
        meeting = (MeetingModel)
bundle.getSerializable("extra_meeting");
        assert meeting != null;
        idMeeting = meeting.getIdMeeting();
        latitude = meeting.getLatitude();
        longitude = meeting.getLongitude();

        Log.d(TAG, "getMeeting: location, latitude : " +
latitude + ", longitude : " + longitude);

        setContent(true);
        prepareBerkas();
    }
}

private void setContent(boolean firstCall) {
    DateFormatter df = new DateFormatter();

tvTanggal.setText(df.getDateLongIndo(meeting.getTgMeeting()));

tvJamStart.setText(df.getTimeFormat(meeting.getTsStart()));
tvJamEnd.setText(df.getTimeFormat(meeting.getTsEnd()));
tvJudul.setText(meeting.getJdlMeeting());
}

```

```

        tvRuang.setText(meeting.getLokasi());
        edtAcara.setText(meeting.getAcaraMeeting());

        if (firstCall) {
            //Mengecek izin user untuk mengedit meeting
            if (userEntri.equals(meeting.getUserEntri())) { //Jika
yg membuka detail meeting adalah orang yg sama yg membuat rapat
                editPermission = true;
            }
        }

@Override
public boolean onSupportNavigateUp() {
    finish();
    return super.onSupportNavigateUp();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_detail_meeting,
menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public void invalidateOptionsMenu() {
    super.invalidateOptionsMenu();
}

@Override
public boolean onPrepareOptionsMenu(Menu menu) {
    MenuItem itemedit = menu.findItem(R.id.action_edit);
    MenuItem itemup = menu.findItem(R.id.action_upload);

    if (editPermission) {
        itemedit.setVisible(true);
        itemup.setVisible(true);
    } else {
        // disabled
        itemedit.setVisible(false);
        itemup.setVisible(false);
    }
    return super.onPrepareOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            onBackPressed();
            break;
        case R.id.action_edit:
            requestEditMeeting();
            break;
        case R.id.action_upload:
            showDialogUpload();
            break;
        case R.id.action_notulen:

```

```

        requestNotulenData();
        break;
    }
    return super.onOptionsItemSelected(item);
}

@Override
public void onRequestPermissionsResult(int requestCode,
String[] permissions, int[] grantResults) {
    if (requestCode == MULTIPLE_PERMISSION_ID) { //Jika izin
diterima
        Map<String, Integer> perms = new HashMap<>();
        // Initialize the map with both permissions
        perms.put(Manifest.permission.WRITE_EXTERNAL_STORAGE,
PackageManager.PERMISSION_GRANTED);
        perms.put(Manifest.permission.ACCESS_FINE_LOCATION,
PackageManager.PERMISSION_GRANTED);
        // Fill with actual results from user
        if (grantResults.length > 0) {
            for (int i = 0; i < permissions.length; i++)
                perms.put(permissions[i], grantResults[i]);
            // Check for both permissions
            if
(perms.get(Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED
                &&
perms.get(Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED) {
                // process the normal flow
                //else any one or both the permissions are not
granted
                prepareView();
                invalidateOptionsMenu();
                getLastLocation();
            } else {
                //permission is denied (this is the first
time, when "never ask again" is not checked) so ask again
explaining the usage of permission
                // shouldShowRequestPermissionRationale will
return true
                //show the dialog or snackbar saying its
necessary and try again otherwise proceed with setup.
                if
(ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) ||
ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.ACCESS_FINE_LOCATION)) {
                    confirmationDialog("Untuk memakai aplikasi
ini, diperlukan izin layanan lokasi dan Berkas",
                        new
DialogInterface.OnClickListener() {
                            @Override
                            public void
onClick(DialogInterface dialog, int which) {
                                switch (which) {
                                    case
DialogInterface.BUTTON_POSITIVE:
checkAndRequestPermissions();

```

```

        break;
        case
DialogInterface.BUTTON_NEGATIVE:
        // proceed with
        // logic by disabling the related features or quit the app.
        finish();
        break;
    }
    });
}
//permission is denied (and never ask again is
checked)
//shouldShowRequestPermissionRationale will
return false
else {
    Toast.makeText(this, "Silahkan aktifkan
izin layanan lokasi dan kamera dipengaturan anda",
Toast.LENGTH_LONG)
        .show();
    //proceed with logic by disabling the
related features or quit the app.
    finish();
}
}
}
}

private void confirmationDialog(String message,
DialogInterface.OnClickListener okListener) {
    new AlertDialog.Builder(this)
        .setMessage(message)
        .setPositiveButton("OK", okListener)
        .setNegativeButton("Cancel", okListener)
        .create()
        .show();
}

@Override
protected void onActivityResult(int requestCode, int
resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    try {
        // When an Image is picked
        switch (requestCode) {
            case REQUEST_IMAGE_DATA:
                if (resultCode == RESULT_OK && null != data) {
                    // Get the Image from data
                    Uri selectedImage = data.getData();
                    String[] filePathColumn =
{MediaStore.Images.Media.DATA};

                    assert selectedImage != null;
                    Cursor cursor =
getContentResolver().query(selectedImage, filePathColumn,
null, null, null);
                    assert cursor != null;

```



```

        cursor.moveToFirst();

        int columnIndex =
cursor.getColumnIndex(filePathColumn[0]);
        mediaPath = cursor.getString(columnIndex);
        edtFile.setText(mediaPath);
        cursor.close();
    }
    break;
    case REQUEST_FILE_DATA:
        assert data != null;
        Uri uri = data.getData();

        String path =
getFileFromURI(DetailMeetingActivity.this, uri);
        edtFile.setText(path);
        break;
    case REQUEST_EDIT_MEETING:
        if (resultCode == RESULT_OK) {
            assert data != null;
            Bundle bundle = data.getExtras();
            assert bundle != null;
            meeting = (MeetingModel)
bundle.getSerializable("extra_meeting");
            DATA_UPDATED = true;
            setContent(false);
        }
        break;
    }
} catch (Exception e) {
    Log.e("onARException", "problem: " + e.getMessage());
}
}

private void requestNotulenData() {
    progressDialog.show();

    ApiService service =
ApiClient.getClient().create(ApiService.class);

    Call<NotulenModel> call =
service.getNotulen(meeting.getIdMeeting());

    call.enqueue(new Callback<NotulenModel>() {
        @Override
        public void onResponse(@NonNull Call<NotulenModel>
call, @NonNull Response<NotulenModel> response) {
            progressDialog.dismiss();

            assert response.body() != null;
            if (response.body().getStatus()) { //jika notulen
ditemukan
                String idNotulen =
response.body().getIdNotulen();
                String isiNotulen =
response.body().getIsiNotulen();

                showDialogNotulen("edit", idNotulen,
isiNotulen);
            }
        }
    });
}

```

```

        } else { //jika notulen tidak ditemukan
            showDialogNotulen("null", "null", "null");
        }
    }

    @Override
    public void onFailure(@NonNull Call<NotulenModel>
call, @NonNull Throwable t) {
        progressDialog.dismiss();
        Toast.makeText(getApplicationContext(), "Kesalahan
data!", Toast.LENGTH_SHORT).show();
    }
});
}

private void requestEditMeeting() {
    progressDialog.show();

    ApiService service =
ApiClient.getClient().create(ApiService.class);

    Call<PrepareMeetingModel> call =
service.prepareMeeting(meeting.getIdMeeting());

    call.enqueue(new Callback<PrepareMeetingModel>() {
        @Override
        public void onResponse(@NonNull
Call<PrepareMeetingModel> call, @NonNull
Response<PrepareMeetingModel> response) {
            progressDialog.dismiss();

            assert response.body() != null;
            if (response.body().getStatus()) {
                JsonObject jsonObject =
response.body().getData();

                Intent intent = new
Intent(getApplicationContext(), GenerateMeetingActivity.class);
                intent.putExtra("act", "edit");
                intent.putExtra("data",
String.valueOf(jsonObject));
                startActivityForResult(intent,
REQUEST_EDIT_MEETING);
            } else {
                Toast.makeText(getApplicationContext(),
"Kesalahan !", Toast.LENGTH_SHORT).show();
            }
        }

        @Override
        public void onFailure(@NonNull
Call<PrepareMeetingModel> call, @NonNull Throwable t) {
            progressDialog.dismiss();
            Toast.makeText(getApplicationContext(), "Kesalahan
data!", Toast.LENGTH_SHORT).show();
        }
    });
}
}

```

```

private void showDialogNotulen(String act, String idNotulen,
String isiNotulen) {
    AlertDialog.Builder dialogBuilder = new
AlertDialog.Builder(this);
    final View myView =
getLayoutInflater().inflate(R.layout.form_dialog_notulen, null);

    htmlViewer = myView.findViewById(R.id.htmlViewer);
    htmlEditor = myView.findViewById(R.id.htmlEditor);
    RelativeLayout HtmlEditorContainer =
myView.findViewById(R.id.HtmlEditorContainer);

    setupHtmlEditor(myView);

    if (editPermission) { //jika punya akses edit
        //hide viewer
        htmlViewer.setVisibility(View.GONE);

        //jika ada data notulen, tampilkan isinya
        if (act.equals("edit")) htmlEditor.render(isiNotulen);
        else htmlEditor.render();

        dialogBuilder.setPositiveButton("Simpan", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialogInterface, int i) {

                String updateNotulen =
htmlEditor.getContentAsHTML();

                if (updateNotulen.equals("<p data-
tag=\"input\" style=\"color:#000000;\"></p>")) { //zero
                    Toast.makeText(getApplicationContext(),
"Anda belum mengetik apapun :)", Toast.LENGTH_LONG).show();
                } else {
                    if (act.equals("null")) { //save
                        saveNotulen(updateNotulen);
                    } else { //edit
                        editNotulen(idNotulen, updateNotulen);
                    }
                }
            }
        });
        dialogBuilder.setNegativeButton("Batal", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialogInterface, int i) {
                dialogInterface.dismiss();
            }
        });
    } else { //jika akses hanya melihat
        //hide editor
        HtmlEditorContainer.setVisibility(View.GONE);

        if (act.equals("null"))
            htmlViewer.render("<p>Belum ada notulen yang
disimpan, cek lagi nanti ...</p>");
    }
}

```

```

        else htmlViewer.render(isiNotulen); //jika sudah ada
        notulen, tampilkan isinya

        dialogBuilder.setPositiveButton("Tutup", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
        DialogInterface, int i) {
                dialogInterface.dismiss();
            }
        });

        dialogBuilder.setView(myView);
        AlertDialog alertDialog = dialogBuilder.create();
        alertDialog.setCancelable(false);
        alertDialog.show();
    }

    private void setupHtmlEditor(View myView) {
        myView.findViewById(R.id.action_h1).setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                htmlEditor.updateTextStyle(EditorTextStyle.H1);
            }
        });

        myView.findViewById(R.id.action_h2).setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                htmlEditor.updateTextStyle(EditorTextStyle.H2);
            }
        });

        myView.findViewById(R.id.action_h3).setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                htmlEditor.updateTextStyle(EditorTextStyle.H3);
            }
        });

        myView.findViewById(R.id.action_bold).setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                htmlEditor.updateTextStyle(EditorTextStyle.BOLD);
            }
        });

        myView.findViewById(R.id.action_Italic).setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {

```

```

htmlEditor.updateTextStyle(EditorTextStyle.ITALIC);
    }
    });

myView.findViewById(R.id.action_indent).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
htmlEditor.updateTextStyle(EditorTextStyle.INDENT);
    }
    });

myView.findViewById(R.id.action_outdent).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {

htmlEditor.updateTextStyle(EditorTextStyle.OUTDENT);
    }
    });

myView.findViewById(R.id.action_bulleted).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        htmlEditor.insertList(false);
    }
    });

myView.findViewById(R.id.action_unordered_numbered).setOnClickList
ener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        htmlEditor.insertList(true);
    }
    });

myView.findViewById(R.id.action_insert_link).setOnClickListener(ne
w View.OnClickListener() {
    @Override
    public void onClick(View v) {
        htmlEditor.insertLink();
    }
    });

myView.findViewById(R.id.action_erase).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        htmlEditor.clearAllContents();
    }
    });

```

```

myView.findViewById(R.id.action_blockquote).setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {

htmlEditor.updateTextStyle(EditorTextStyle.BLOCKQUOTE);
    }
});

}

private void showDialogUpload() {
    AlertDialog.Builder dialogBuilder = new
AlertDialog.Builder(this);
    final View myView =
getLayoutInflater().inflate(R.layout.form_upload_dialog, null);
    edtFileInfo = myView.findViewById(R.id.etFileInfo);
    edtFile = myView.findViewById(R.id.etFileName);

    String[] mimeTypes =
        {"application/msword",
"application/vnd.openxmlformats-
officedocument.wordprocessingml.document", // .doc & .docx
        "application/vnd.ms-excel",
"application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet", // .xls & .xlsx
        "text/plain",
        "application/pdf"};

    edtFile.setOnClickListener(view -> {
        switch
(String.valueOf(sFormatFile.getSelectedItemId())) {
            case "1":
                Intent intent = new Intent();
                intent.setAction(Intent.ACTION_GET_CONTENT);
                intent.addCategory(Intent.CATEGORY_OPENABLE);

                intent.setType("*/*");
                intent.putExtra(Intent.EXTRA_MIME_TYPES,
mimeTypes);

                startActivityForResult(intent,
REQUEST_FILE_DATA);
                break;
            case "2":
                Intent galleryIntent = new
Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                startActivityForResult(galleryIntent,
REQUEST_IMAGE_DATA);
                break;
            default:
                Toast.makeText(getApplicationContext(), "Pilih
Format File terlebih dahulu !", Toast.LENGTH_SHORT).show();
                break;
        }
    });
}

```

```

        sFileJns = myView.findViewById(R.id.sp_file_jns);
        itemFileJns = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item,

getResources().getStringArray(R.array.jenis_file));

itemFileJns.setDropDownViewResource(android.R.layout.simple_spinner_
r_dropdown_item);
        sFileJns.setAdapter(itemFileJns);

        sFormatFile = myView.findViewById(R.id.sp_file_format);
        itemFormatFile = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item,

getResources().getStringArray(R.array.format_file));

itemFormatFile.setDropDownViewResource(android.R.layout.simple_spi
nner_dropdown_item);
        sFormatFile.setAdapter(itemFormatFile);

        dialogBuilder.setPositiveButton("Upload", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface,
int i) {
                fileFormat =
String.valueOf(sFormatFile.getSelectedItemId());
                fileJns =
String.valueOf(sFileJns.getSelectedItemId());
                fileInfo = edtFileInfo.getText().toString();
                fileName = edtFile.getText().toString();

                if (fileFormat.equals("0")) {
                    Toast.makeText(DetailMeetingActivity.this,
"Pilih Format File Terlebih Dahulu",
Toast.LENGTH_SHORT).show();
                }

                if (fileJns.equals("0")) {
                    Toast.makeText(DetailMeetingActivity.this,
"Pilih File Jenis Terlebih Dahulu",
Toast.LENGTH_SHORT).show();
                }

                if (TextUtils.isEmpty(fileInfo)) {
                    edtFileInfo.setError("Isikan file info!");
                    edtFileInfo.requestFocus();
                    return;
                }

                if (TextUtils.isEmpty(fileName)) {
                    edtFile.setError("Pilih File terlebih
dahulu");

                    edtFile.requestFocus();
                    return;
                }

                uploadFile(idMeeting, fileJns, fileInfo,

```

```

userEntri, Uri.parse(fileName));

        dialogInterface.dismiss();
    }
});

        dialogBuilder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface,
int i) {
                dialogInterface.dismiss();
            }
        });

        dialogBuilder.setView(myView);
AlertDialog alertDialog = dialogBuilder.create();
alertDialog.show();
    }

    private void saveNotulen(String isiNotulen) {
        progressDialog.show();

        ApiService service =
ApiClient.getClient().create(ApiService.class);

        Call<NotulenModel> call =
service.saveNotulen(meeting.getIdMeeting(), isiNotulen,
userEntri);

        call.enqueue(new Callback<NotulenModel>() {
            @Override
            public void onResponse(@NonNull Call<NotulenModel>
call, @NonNull Response<NotulenModel> response) {
                progressDialog.dismiss();

                assert response.body() != null;
                Toast.makeText(getApplicationContext(),
response.body().getMsg(), Toast.LENGTH_LONG).show();
            }

            @Override
            public void onFailure(@NonNull Call<NotulenModel>
call, @NonNull Throwable t) {
                progressDialog.dismiss();
                Toast.makeText(getApplicationContext(), "Kesalahan
data!", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void editNotulen(String idNotulen, String isiNotulen)
{
        progressDialog.show();

        ApiService service =
ApiClient.getClient().create(ApiService.class);

        Call<NotulenModel> call = service.editNotulen(idNotulen,

```



```

isiNotulen);

        call.enqueue(new Callback<NotulenModel>() {
            @Override
            public void onResponse(@NonNull Call<NotulenModel>
call, @NonNull Response<NotulenModel> response) {
                progressDialog.dismiss();

                assert response.body() != null;
                Toast.makeText(getApplicationContext(),
response.body().getMsg(), Toast.LENGTH_LONG).show();
            }

            @Override
            public void onFailure(@NonNull Call<NotulenModel>
call, @NonNull Throwable t) {
                progressDialog.dismiss();
                Toast.makeText(getApplicationContext(), "Kesalahan
data!", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void uploadFile(String idMeeting, String fileJns,
String fileInfo, String userEntri, Uri
fileUri) {
        progressDialog.show();
        File file = new
File(Objects.requireNonNull(fileUri.getPath()));

        //Create RequestBody for file
        RequestBody requestBody =
RequestBody.create(MediaType.parse("*/*"), file);
        MultipartBody.Part fileToUpload =
MultipartBody.Part.createFormData("file", file.getName(),
requestBody);
        RequestBody idRapat =
RequestBody.create(MediaType.parse("text/plain"), idMeeting);
        RequestBody jnsFile =
RequestBody.create(MediaType.parse("text/plain"), fileJns);
        RequestBody infoFile =
RequestBody.create(MediaType.parse("text/plain"), fileInfo);
        RequestBody nameFile =
RequestBody.create(MediaType.parse("text/plain"), file.getName());
        RequestBody entriUser =
RequestBody.create(MediaType.parse("text/plain"), userEntri);

        ApiService service =
ApiClient.getClient().create(ApiService.class);

        Call<FileResponse> call = service.uploadFileRapat(idRapat,
jnsFile, infoFile, nameFile,
entriUser, fileToUpload);

        call.enqueue(new Callback<FileResponse>() {
            @Override
            public void onResponse(@NonNull Call<FileResponse>
call, @NonNull Response<FileResponse> response) {
                progressDialog.dismiss();

```

```

        assert response.body() != null;
        if (response.body().isStatus()) {
            prepareBerkas();
        }
        Toast.makeText(DetailMeetingActivity.this,
response.body().getMessage(),
            Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onFailure(@NonNull Call<FileResponse>
call, @NonNull Throwable t) {
        progressDialog.dismiss();
        Log.e("onFailureUpload", "error: " +
t.getMessage());
    }
});
}

    public static String getFileFromURI(Context context, Uri
contentUri) {
        //copy file and send new file path
        String fileName = getFileName(contentUri);
        File wallpaperDirectory = new File(
            Environment.getExternalStorageDirectory() +
"/upload_document");
        // have the object build the directory structure, if
needed.
        if (!wallpaperDirectory.exists()) {
            wallpaperDirectory.mkdirs();
        }
        if (!TextUtils.isEmpty(fileName)) {
            File copyFile = new File(wallpaperDirectory +
File.separator + fileName);
            // create folder if not exists

            copy(context, contentUri, copyFile);
            return copyFile.getAbsolutePath();
        }
        return null;
    }

    public static String getFileName(Uri uri) {
        if (uri == null) return null;
        String fileName = null;
        String path = uri.getPath();
        assert path != null;
        int cut = path.lastIndexOf('/');
        if (cut != -1) {
            fileName = path.substring(cut + 1);
        }
        return fileName;
    }

    public static void copy(Context context, Uri srcUri, File
dstFile) {
        try {
            InputStream inputStream =
context.getContentResolver().openInputStream(srcUri);

```

```

        if (inputStream == null) return;
        OutputStream outputStream = new
FileOutputStream(dstFile);
        copystream(inputStream, outputStream);
        inputStream.close();
        outputStream.close();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

    public static int copystream(InputStream input, OutputStream
output) throws Exception, IOException {
        byte[] buffer = new byte[BUFFER_SIZE];

        BufferedInputStream in = new BufferedInputStream(input,
BUFFER_SIZE);
        BufferedOutputStream out = new
BufferedOutputStream(output, BUFFER_SIZE);
        int count = 0, n = 0;
        try {
            while ((n = in.read(buffer, 0, BUFFER_SIZE)) != -1) {
                out.write(buffer, 0, n);
                count += n;
            }
            out.flush();
        } finally {
            try {
                out.close();
            } catch (IOException e) {
                Log.e(e.getMessage(), String.valueOf(e));
            }
            try {
                in.close();
            } catch (IOException e) {
                Log.e(e.getMessage(), String.valueOf(e));
            }
        }
        return count;
    }

    private void prepareBerkas() {
        BerkasViewModel berkasViewModel = new
ViewModelProvider(this).get(BerkasViewModel.class);
        berkasViewModel.loadBerkas(meeting);
        berkasViewModel.getBerkas().observe(this, getBerkas);
    }

    private Observer<ListBerkasModel> getBerkas = new
Observer<ListBerkasModel>() {
        @Override
        public void onChanged(@Nullable ListBerkasModel
listBerkasModel) {
            if (listBerkasModel != null) {
                adapter = new
BerkasAdapter(getApplicationContext(),
listBerkasModel.getDataBerkas(), editPermission);
            }
        }
    }

```

```

        rvBerkas.setLayoutManager(new
LinearLayoutManager(getApplicationContext()));
        rvBerkas.setAdapter(adapter);
        adapter.notifyDataSetChanged();
    }
}
};

@Override
public void onBackPressed() {
    if (isTaskRoot() || DATA_UPDATED) {
        Intent intent = new Intent(this, MainActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        super.onBackPressed();
    } else {
        super.onBackPressed();
    }
}

@Override
public void onClick(View v) {
    if (v == ibMap) {
        Intent intentLokasi = new
Intent(DetailMeetingActivity.this, MapsActivity.class);
        intentLokasi.putExtra("extra_lat", latitude);
        intentLokasi.putExtra("extra_long", longitude);
        intentLokasi.putExtra("extra_lokasi",
tvRuang.getText().toString());
        startActivity(intentLokasi);
    }
    if (v == btnKonfirmasi) {
        if (checkPosisi()) { // jika berada di lokasi rapat
            progressDialog.show();
            String idPeg =
SharedPreferences.getInstance(this).getPegawai().getIdPeg();
            ApiService service =
ApiClient.getClient().create(ApiService.class);

            Call<ResultModel> callResult =
service.editKonfirmasi(idMeeting, idPeg);

            callResult.enqueue(new Callback<ResultModel>() {
                @Override
                public void onResponse(Call<ResultModel> call,
Response<ResultModel> response) {
                    progressDialog.dismiss();
                    if (response.isSuccessful() &&
response.body() != null) {
                        if (response.body().getStatus()) {
//
btnKonfirmasi.setVisibility(View.GONE);

Toast.makeText(DetailMeetingActivity.this,
response.body().getMsg(),
Toast.LENGTH_LONG).show();
} else {

```

```

Toast.makeText (DetailMeetingActivity.this,
                response.body().getMsg(),
Toast.LENGTH_LONG).show();
        }
        } else {

Toast.makeText (DetailMeetingActivity.this,
                "Upps terjadi kesalahan,
silahkan cek koneksi anda.", Toast.LENGTH_SHORT).show();
        }
        }

@Override
public void onFailure(Call<ResultModel> call,
Throwable t) {
        progressDialog.dismiss();
        Log.d(TAG, "onFailure: error -> " +
t.getMessage());
        Toast.makeText (DetailMeetingActivity.this,
                "Upps terjadi kesalahan, silahkan
hubung admin", Toast.LENGTH_SHORT).show();
        }
        });
        } else { // jika tidak berada dilokasi rapat
        Toast.makeText (this, "Anda belum sampai dilokasi
rapat", Toast.LENGTH_SHORT).show();
        }
        }
    }

private boolean checkPosisi(){
    float[] jarak = new float[1];
    Log.d(TAG, "onClick: user coordinate -> lat: " + userLat +
", long: " + userLong);
    Location.distanceBetween(userLat, userLong,
        Double.parseDouble(latitude),
Double.parseDouble(longitude), jarak);
    Log.d(TAG, "onClick: jarak antara user dan lokasi rapat
dalam km -> " + (jarak[0]/1000));
    return (jarak[0] / 1000) < 0.5;
    }
}

```

MapsActivity.java

```

package id.ac.itn.semar.Activity;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentActivity;

import android.content.Intent;
import android.location.Location;
import android.net.Uri;
import android.os.Bundle;

```

```

import android.util.Log;
import android.view.View;
import android.widget.Toast;

import com.akexorcist.googleplay.direction.DirectionCallback;
import com.akexorcist.googleplay.direction.GoogleDirection;
import com.akexorcist.googleplay.direction.model.Direction;
import com.github.clans.fab.FloatingActionButton;
import com.github.clans.fab.FloatingActionMenu;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import
com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.CameraPosition;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.material.textfield.TextInputEditText;

import java.util.Objects;

import id.ac.itn.semar.Helper.LocationHelper;
import id.ac.itn.semar.R;

public class MapsActivity extends AppCompatActivity implements
    OnMapReadyCallback,
    GoogleApiClient.ConnectionCallbacks,
    GoogleApiClient.OnConnectionFailedListener,
    View.OnClickListener{
    private String TAG = MapsActivity.class.getSimpleName();
    private TextInputEditText tieLokasiRapat;
    SupportMapFragment mapFragment;
    FloatingActionButton fabCurrent, fabDirection, fabRapat;
    FloatingActionMenu famMap;

    private GoogleMap mMap;

    private double longitude, latitude;
    private double userLongitude, userLatitude;

    private GoogleApiClient googleApiClient;

    private void getCurrentLocation(){
        Log.d(TAG, "getCurrentLocation: executed");
        Location location =
LocationServices.FusedLocationApi.getLastLocation(googleApiClient)
;
        if (location != null) {

```

```

        userLongitude = location.getLongitude();
        userLatitude = location.getLatitude();

        Log.d(TAG, "getCurrentLocation: user coordinat ->
latitude: " + userLatitude
        + ", longitude: " + userLongitude);
    }
}

private void getMeetingLocation(){
    mMap.clear();

    moveMap(longitude, latitude);
}

private void moveMap(double mLong, double mLat){
    Log.d(TAG, "moveMap: executed");
    LatLng latLng = new LatLng(mLat, mLong);

    mMap.addMarker(new MarkerOptions()
        .position(latLng)
        .draggable(true)
        .title("Lokasi Rapat"));

    CameraPosition cameraPosition = new
CameraPosition.Builder()
        .target(latLng) // Sets the center of the map
to location user
        .zoom(17) // Sets the tilt of
the camera to 30 degrees
        .build(); // Creates a
CameraPosition from the builder

mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPos
ition));
// Toast.makeText(this, latitude+ ", " + longitude,
Toast.LENGTH_SHORT).show();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);
    if (getSupportActionBar() != null) {
        getSupportActionBar().setTitle("Lokasi Rapat");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    tieLokasiRapat = findViewById(R.id.et_lokasi_rapat);
    fabCurrent = findViewById(R.id.fab_current);
    fabDirection = findViewById(R.id.fab_direction);
    fabRapat = findViewById(R.id.fab_destination);
    famMap = findViewById(R.id.fam_map);
}

```

```

        fabCurrent.setOnClickListener(this);
        fabDirection.setOnClickListener(this);
        fabRapat.setOnClickListener(this);

        if (getIntent() != null) {
            latitude =
Double.parseDouble(Objects.requireNonNull(getIntent().getStringExtra("extra_lat")));
            longitude =
Double.parseDouble(Objects.requireNonNull(getIntent().getStringExtra("extra_long")));
            String lokasiRapat =
getIntent().getStringExtra("extra_lokasi");

            tieLokasiRapat.setText(lokasiRapat);
            Log.d(TAG, "onCreate: location from detailIntent ->
latitude : " + latitude + ", longitude : " + longitude);
        }

        // Obtain the SupportMapFragment and get notified when the
map is ready to be used.
        mapFragment = (SupportMapFragment)
getSupportFragmentManager()
            .findFragmentById(R.id.map);
        assert mapFragment != null;
        mapFragment.getMapAsync(this);

        googleApiClient = new GoogleApiClient.Builder(this)
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .addApi(LocationServices.API)
            .build();
    }

    @Override
    public boolean onSupportNavigateUp() {
        finish();
        return super.onSupportNavigateUp();
    }

    @Override
    protected void onStart() {
        Log.d(TAG, "onStart: executed");
        googleApiClient.connect();
        super.onStart();
    }

    @Override
    protected void onStop() {
        Log.d(TAG, "onStop: executed");
        googleApiClient.disconnect();
    }

```



```

        super.onStop();
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be
    used.
     * This is where we can add markers or lines, add listeners or
    move the camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the
    user will be prompted to install
     * it inside the SupportMapFragment. This method will only be
    triggered once the user has
     * installed Google Play services and returned to the app.
    */
    @Override
    public void onMapReady(GoogleMap googleMap) {
        Log.d(TAG, "onMapReady: executed");
        mMap = googleMap;

        // Add a marker in Sydney and move the camera
        LatLng malang = new LatLng(latitude, longitude);

        mMap.addMarker(new MarkerOptions()
            .position(malang)
            .draggable(true)
            .title("Marker in Malang"));

        CameraPosition cameraPosition = new
        CameraPosition.Builder()
            .target(malang) // Sets the center of the map
            to location user
            .zoom(20) // Sets the tilt of
            the camera to 30 degrees
            .build(); // Creates a
        CameraPosition from the builder

        //
        mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPos
        ition));

        mMap.moveCamera(CameraUpdateFactory.newCameraPosition(cameraPositi
        on));
    }

    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.fab_current:
                mMap.clear();
                getCurrentLocation();
                moveMap(userLongitude, userLatitude);
        }
    }

```

```

        break;
        case R.id.fab_direction:
//          getDestDirection(userLatitude, userLongitude,
latitude, longitude);
            openInGm(longitude, latitude);
            break;
        case R.id.fab_destination:
            getMeetingLocation();
            break;
    }
    famMap.close(true);
}

@Override
public void onConnected(@Nullable Bundle bundle) {
    Log.d(TAG, "onConnected: executed");
    getCurrentLocation();
}

@Override
public void onConnectionSuspended(int i) {
    Log.d(TAG, "onConnectionSuspended: executed");
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult
connectionResult) {
    Log.d(TAG, "onConnectionFailed: executed");
}

private void getDestDirection(double userLat, double userLong,
double destLat, double destLong){
    Log.d(TAG, "getDestDirection: check coordinat user -> lat:
"+ userLat + ", long: " +userLong);
    Log.d(TAG, "getDestDirection: check coordinat meeting ->
lat " + destLat + ", long: " + destLong);
    LatLng userCoordinat = new LatLng(userLat, userLong);
    LatLng meetingCoordinat = new LatLng(destLat, destLong);

    GoogleDirection.withServerKey(getResources().getString(R.string.go
ogle_maps_key))
        .from(userCoordinat)
        .to(meetingCoordinat)
        .execute(new DirectionCallback() {
            @Override
            public void onDirectionSuccess(Direction
direction) {
                Log.d(TAG, "onDirectionSuccess: executed,
direction ok? " + direction.getErrorMessage());
            }

            @Override
            public void onDirectionFailure(Throwable t) {

```

```
        Log.d(TAG, "onDirectionFailure: executed,
something went wrong -> "
                + t.getMessage());
    }
    });
}

private void openInGm(double meetingLong, double meetingLat){
    Uri locationUri = Uri.parse("google.navigation:q=" +
meetingLat + "," + meetingLong);
    Intent mapIntent = new Intent(Intent.ACTION_VIEW,
locationUri);
    mapIntent.setPackage("com.google.android.apps.maps");

    if (mapIntent.resolveActivity(getPackageManager()) !=
null) {
        startActivity(mapIntent);
    } else {
        Log.d(TAG, "openInGm: google map app not installed");
    }
}
}
```