HomeActivity.java

```
package
com.kelas_c.a220280075.bhabinkamtibnas.
activity;

import android.annotation.SuppressLint;
import                android.os.Bundle;
import               android.os.Handler;
import            android.view.MenuItem;
import               android.view.View;
import          android.widget.TextView;
import             android.widget.Toast;

import     androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.viewpager2.widget.ViewPager2;

import
com.google.android.material.bottomnavig
ation.BottomNavigationView;
import
com.kelas_c.a220280075.bhabinkamtibnas.
BackToLogin;
import
com.kelas c.a220280075.bhabinkamtibnas.
R;
import
com.kelas_c.a220280075.bhabinkamtibnas.
SessionManager;

public   class   HomeActivity   extends
AppCompatActivity                    {

    private    ViewPager2    viewPager;
    private        BottomNavigationView
bottomNavigationView;
    String          user,          nama;
    SessionManager       sessionManager;
    private   boolean   klikduakali   =
false;

    @Override
    protected    void    onCreate(Bundle
savedInstanceState)                  {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_home);

        viewPager                    =
findViewById(R.id.view_pager);
        bottomNavigationView         =
findViewById(R.id.bottom_navigation);

        HomePager    adapter    =    new
HomePager(this);
        viewPager.setAdapter(adapter);

        sessionManager       =       new
SessionManager(HomeActivity.this);
if(!sessionManager.isLoggedIn()){

BackToLogin.kembaliKeHalamanMasuk(HomeA
ctivity.this);
        }
        user                         =
sessionManager.getUserDetail().get(Sess
ionManager.UserAnggota);
        nama                         =
sessionManager.getUserDetail().get(Sess
ionManager.NamaAnggota);


bottomNavigationView.setOnNavigationIte
mSelectedListener(new
BottomNavigationView.OnNavigationItemSe
lectedListener()                     {
            @Override
            public              boolean
onNavigationItemSelected(@NonNull
MenuItem            item)            {
                int       itemId       =
item.getItemId();
                if       (itemId       ==
R.id.home)                           {

viewPager.setCurrentItem(0);
                    return         true;
                } else if (itemId ==
R.id.profil)                         {

viewPager.setCurrentItem(1);
                    return         true;
                }
                return             false;
            }
        });


viewPager.registerOnPageChangeCallback(
new ViewPager2.OnPageChangeCallback() {
            @Override
            public                  void
onPageSelected(int      position)    {

super.onPageSelected(position);
                switch    (position)    {
                    case             0:

bottomNavigationView.setSelectedItemId(
R.id.home);
                        break;
                    case             1:

bottomNavigationView.setSelectedItemId(
R.id.profil);
                        break;
                }
            }
        });
    }

    @Override
    public    void    onBackPressed()    {
        if        (klikduakali)        {
```

```
            super.onBackPressed();
            return;
        }
        klikduakali = true;
        showToast("Tekan sekali lagi untuk keluar.", R.layout.custom_toast_info);
        new Handler().postDelayed(() -> klikduakali = false, 2000);
    }

    private void showToast(String message, int layoutId) {
        final Toast toast = new Toast(this);

toast.setDuration(Toast.LENGTH_LONG);
        @SuppressLint("InflateParams") View custom_view = getLayoutInflater().inflate(layoutId, null);
        TextView tvMessage = custom_view.findViewById(R.id.toast_message);
        tvMessage.setText(message);
        toast.setView(custom_view);
        toast.show();
    }
}
```

HomePager.java

```
package com.kelas_c.a220280075.bhabinkamtibnas.activity;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentActivity;
import androidx.viewpager2.adapter.FragmentStateAdapter;

import com.kelas_c.a220280075.bhabinkamtibnas.fragment.HomeFragment;
import com.kelas_c.a220280075.bhabinkamtibnas.fragment.ProfilFragment;

public class HomePager extends FragmentStateAdapter {

    public HomePager(@NonNull FragmentActivity fragmentActivity) {
        super(fragmentActivity);
    }

    @NonNull
    @Override
    public Fragment createFragment(int position) {
        switch (position) {
            case 0:
                return new HomeFragment();
            case 1:
                return new ProfilFragment();
            default:
                return new HomeFragment();
        }
    }

    @Override
    public int getItemCount() {
        return 2;
    }
}
```

LaporanActivity.java

```
package com.kelas_c.a220280075.bhabinkamtibnas.activity;

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.PorterDuffColorFilter;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.text.Editable;
import android.text.TextUtils;
import android.text.TextWatcher;
import android.view.Gravity;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import
```

```
androidx.appcompat.app.ActionBar;
import
androidx.appcompat.app.AlertDialog;
import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.appcompat.widget.Toolbar;
import
androidx.core.app.ActivityCompat;
import
androidx.core.content.ContextCompat;

import
com.google.android.gms.location.FusedLo
cationProviderClient;
import
com.google.android.gms.location.Locatio
nServices;
import
com.google.android.gms.maps.CameraUpdat
eFactory;
import
com.google.android.gms.maps.GoogleMap;
import
com.google.android.gms.maps.OnMapReadyC
allback;
import
com.google.android.gms.maps.model.Bitma
pDescriptor;
import
com.google.android.gms.maps.model.Bitma
pDescriptorFactory;
import
com.google.android.gms.maps.model.LatLn
g;
import
com.google.android.gms.maps.model.Marke
rOptions;
import
com.google.android.material.floatingact
ionbutton.ExtendedFloatingActionButton;
import
com.kelas_c.a220280075.bhabinkamtibnas.
R;
import
com.google.android.gms.maps.MapView;
import
com.kelas_c.a220280075.bhabinkamtibnas.
SessionManager;
import
com.kelas_c.a220280075.bhabinkamtibnas.
api.ApiClient;
import
com.kelas_c.a220280075.bhabinkamtibnas.
api.ApiInterface;
import
com.kelas c.a220280075.bhabinkamtibnas.
model.Data;

import
org.jetbrains.annotations.NotNull;

import                    java.io.File;
import             java.io.IOException;
import                   java.util.List;
import                java.util.Objects;

import                okhttp3.MediaType;
import            okhttp3.MultipartBody;
import              okhttp3.RequestBody;
import                   retrofit2.Call;
import                retrofit2.Callback;
import                retrofit2.Response;

public  class  LaporanActivity  extends
AppCompatActivity             implements
OnMapReadyCallback                    {

    private       GoogleMap       mMap;
    private FusedLocationProviderClient
fusedLocationProviderClient;
    private     static     final     int
LOCATION_PERMISSION_REQUEST_CODE  =  1;
    private       MapView       mapView;
    private EditText deskripsiEditText;
    private  EditText  latitudeEditText;
    private EditText longitudeEditText;
    private   EditText   alamatEditText;
    private    ImageView    unggahfoto;
    private   ProgressDialog   progress;
    String   user,   nama,   deskripsi,
alamat,   lati,   longi,   part_image;
    ApiInterface          apiInterface;
    SessionManager       sessionManager;
    private     static     final     int
CAMERA_REQUEST          =          100;
    private     static     final     int
STORAGE_REQUEST          =          101;
    private     static     final     int
IMAGE_CAPTURE_CODE         =         102;
    private     static     final     int
PICK_IMAGE_CODE          =          103;
    private final int MAX_CHAR_LIMIT =
200;
    private        Uri        imageUri;

    @Override
    protected    void    onCreate(Bundle
savedInstanceState)                   {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_lapora
n);

        sessionManager       =       new
SessionManager(LaporanActivity.this);
        user                          =
sessionManager.getUserDetail().get(Sess
ionManager.UserAnggota);
        nama                          =
sessionManager.getUserDetail().get(Sess
ionManager.NamaAnggota);

        progress          =          new
ProgressDialog(LaporanActivity.this);
        progress.setCancelable(false);
        progress.setMessage("Tunggu
Sebentar...");
```

```
        mapView                         =
findViewById(R.id.mapView);

mapView.onCreate(savedInstanceState);
        mapView.getMapAsync(this);

        deskripsiEditText               =
findViewById(R.id.deskripsilaporan);
        TextView  deskripsilaporanError
=
findViewById(R.id.deskripsilaporanError
);

deskripsiEditText.addTextChangedListene
r(new          TextWatcher()          {
            @Override
            public                  void
beforeTextChanged(CharSequence  s,  int
start,   int   count,   int   after)   {

            }

            @Override
            public                  void
onTextChanged(CharSequence    s,    int
start,   int   before,   int   count)   {

clearErrorState(deskripsiEditText,
deskripsilaporanError,
R.drawable.baseline_assignment_add);
            }

            @Override
            public                  void
afterTextChanged(Editable      s)      {
                if     (s.length()     >
MAX_CHAR_LIMIT)                         {

deskripsilaporanError.setText("Deskrips
i  laporan  maksimal  200  karakter!");

deskripsilaporanError.setVisibility(Vie
w.VISIBLE);
                }         else          {

deskripsilaporanError.setVisibility(Vie
w.GONE);
                }
            }
        });
        latitudeEditText                =
findViewById(R.id.latitudeEditText);
        longitudeEditText               =
findViewById(R.id.longitudeEditText);
        alamatEditText                  =
findViewById(R.id.alamat);
        unggahfoto                      =
findViewById(R.id.imgunggahberkas);
        unggahfoto.setOnClickListener(v
->                tampilGambarDialog());
        ExtendedFloatingActionButton
btnadd  =  findViewById(R.id.btnInsert);
        btnadd.setOnClickListener(new
View.OnClickListener()                  {
            @Override
            public void onClick(View v)
{

if(unggahfoto.getDrawable()   ==   null){
                    showToast("Unggah
Foto                        Kegiatan!",
R.layout.custom_toast_info);
                }else{
                    deskripsi           =
deskripsiEditText.getText().toString().
trim();
                    alamat              =
alamatEditText.getText().toString().tri
m();
                    lati                =
latitudeEditText.getText().toString().t
rim();
                    longi               =
longitudeEditText.getText().toString().
trim();

                    if
(TextUtils.isEmpty(deskripsi))          {

setErrorState(deskripsiEditText,
deskripsilaporanError,
R.drawable.baseline_assignment_add,
"Masukkan     Deskripsi     Laporan!");
                        return;
                    }
                    unggahfoto();
                }
            }
        });

        fusedLocationProviderClient     =
LocationServices.getFusedLocationProvid
erClient(this);

        Toolbar         toolbar         =
findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        ActionBar       mAction         =
getSupportActionBar();
        assert    mAction    !=    null;
        TextView  titleTextView  =  new
TextView(getApplicationContext());
        titleTextView.setText("Buat
Laporan".toUpperCase());
        titleTextView.setTextSize(17);

titleTextView.setTextColor(Color.WHITE)
;

titleTextView.setGravity(Gravity.CENTER
);

mAction.setDisplayOptions(ActionBar.DIS
PLAY_SHOW_CUSTOM);

mAction.setCustomView(titleTextView,
new
ActionBar.LayoutParams(ActionBar.Layout
Params.WRAP_CONTENT,
```

```
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

mAction.setDisplayHomeAsUpEnabled(true)
;

toolbar.setNavigationOnClickListener(v
->                    onBackPressed());
    }

    @Override
    public    void    onMapReady(@NonNull
GoogleMap          googleMap)          {
        mMap        =        googleMap;
        getCurrentLocation();
    }

    private void getCurrentLocation() {
        if
(ActivityCompat.checkSelfPermission(thi
s,
Manifest.permission.ACCESS_FINE_LOCATIO
N) != PackageManager.PERMISSION_GRANTED
&&

ActivityCompat.checkSelfPermission(this
,
Manifest.permission.ACCESS_COARSE_LOCAT
ION)                                  !=
PackageManager.PERMISSION_GRANTED)     {

ActivityCompat.requestPermissions(this,
new
String[]{Manifest.permission.ACCESS_FIN
E_LOCATION},
LOCATION_PERMISSION_REQUEST_CODE);
            return;
        }


fusedLocationProviderClient.getLastLoca
tion().addOnCompleteListener(task -> {
            if (task.isSuccessful() &&
task.getResult()     !=     null)     {
                Location    location    =
task.getResult();
                LatLng currentLatLng =
new      LatLng(location.getLatitude(),
location.getLongitude());


latitudeEditText.setText(String.valueOf
(currentLatLng.latitude));

latitudeEditText.setEnabled(false);

latitudeEditText.setFocusable(false);

longitudeEditText.setText(String.valueO
f(currentLatLng.longitude));

longitudeEditText.setEnabled(false);

longitudeEditText.setFocusable(false);

mMap.moveCamera(CameraUpdateFactory.new
LatLngZoom(currentLatLng,          17));

@SuppressLint("UseCompatLoadingForDrawa
bles") BitmapDrawable bitmapdraww    =
(BitmapDrawable)
getResources().getDrawable(R.drawable.i
coppol);
                Bitmap b = bitmapdraww
.getBitmap();
                Bitmap smallMarker =
Bitmap.createScaledBitmap(b, 123, 123,
false);
                BitmapDescriptor icon =
BitmapDescriptorFactory.fromBitmap(smal
lMarker);
                LatLng myLocation = new
LatLng(location.getLatitude(),
location.getLongitude());
                mMap.addMarker(new
MarkerOptions().position(myLocation).ic
on(icon).title("Lokasi     Anda     saat
ini"));

                Geocoder geocoder = new
Geocoder(getApplicationContext());
                try                    {
                    List<Address>
addresses                             =
geocoder.getFromLocation(location.getLa
titude(), location.getLongitude(), 1);
                    if   (addresses   !=
null   &&   addresses.size()   >   0)   {
                        Address address
=                     addresses.get(0);
                        StringBuilder
sb      =      new      StringBuilder();
                        for (int i = 0;
i  <=  address.getMaxAddressLineIndex();
i++)                                  {

sb.append(address.getAddressLine(i)).ap
pend("\n");
                        }
                        String
fullAddress       =       sb.toString();

alamatEditText.setText(fullAddress);

alamatEditText.setEnabled(false);

alamatEditText.setFocusable(false);
                    }
                } catch (IOException e)
{

e.printStackTrace();
                }
            }
        });
    }

    @Override
    public                            void
```

```
onRequestPermissionsResult(int
requestCode,      @NonNull      String[]
permissions,       @NonNull       int[]
grantResults)                          {

super.onRequestPermissionsResult(reques
tCode,    permissions,    grantResults);
        if        (requestCode        ==
LOCATION_PERMISSION_REQUEST_CODE)      {
            if (grantResults.length > 0
&&          grantResults[0]          ==
PackageManager.PERMISSION_GRANTED)     {
                getCurrentLocation();
            }
        }   else   if   (requestCode   ==
CAMERA_REQUEST)                        {
            if (grantResults.length > 0
&&          grantResults[0]          ==
PackageManager.PERMISSION_GRANTED)     {
                openCamera();
            }           else           {
                showToast("Harap
Aktifkan          Izin          Kamera",
R.layout.custom_toast_info);
            }
        }   else   if   (requestCode   ==
STORAGE_REQUEST)                       {
            if (grantResults.length > 0
&&          grantResults[0]          ==
PackageManager.PERMISSION_GRANTED)     {
                openGallery();
            }           else           {
                showToast("Harap
Aktifkan       Izin       Penyimpanan",
R.layout.custom_toast_info);
            }
        }
    }

    @Override
    protected    void    onResume()    {
        super.onResume();
        mapView.onResume();
    }

    @Override
    protected    void    onPause()    {
        super.onPause();
        mapView.onPause();
    }

    @Override
    protected    void    onDestroy()    {
        super.onDestroy();
        mapView.onDestroy();
    }

    @Override
    public    void    onLowMemory()    {
        super.onLowMemory();
        mapView.onLowMemory();
    }

    private    void    unggahfoto()    {
        File     imagefile     =     new
File(part_image);
        if    (!imagefile.exists())    {
            showToast("File      gambar
tidak                    ditemukan!",
R.layout.custom_toast_gagal);
            return;
        }
        progress.show();
        RequestBody    deskripsilap    =
RequestBody.create(MultipartBody.FORM,
deskripsi);
        RequestBody     alamatlap     =
RequestBody.create(MultipartBody.FORM,
alamat);
        RequestBody       latlap       =
RequestBody.create(MultipartBody.FORM,
lati);
        RequestBody      longlap      =
RequestBody.create(MultipartBody.FORM,
longi);
        RequestBody       idnya       =
RequestBody.create(MultipartBody.FORM,
user);
        RequestBody      reqBody      =
RequestBody.create(MediaType.parse("mul
tipart/form-file"),          imagefile);
        MultipartBody.Part  partImage  =
MultipartBody.Part.createFormData("Ungg
ahFotoLokasi",      imagefile.getName(),
reqBody);

        apiInterface                   =
ApiClient.getApiClient().create(ApiInte
rface.class);
        Call<Data>      masukCall      =
apiInterface.buatlaporanRespone(deskrip
silap,   alamatlap,   latlap,   longlap,
idnya,                    partImage);
        masukCall.enqueue(new
Callback<Data>()                       {
            @Override
            public                 void
onResponse(@NotNull   Call<Data>   call,
@NotNull   Response<Data>   response)   {
                progress.hide();
                if
(response.isSuccessful()             &&
response.body()      !=      null)      {
                    Data responseData =
response.body();
                    if
(responseData.isStatus())              {

showToast(responseData.getMessage(),
R.layout.custom_toast_sukses);
                        Intent it = new
Intent(LaporanActivity.this,
HomeActivity.class);

it.putExtra("SELECTED_TAB",          0);

startActivity(it);
                        finish();
                    }       else       {
```

```
showToast(responseData.getMessage(),
R.layout.custom_toast_gagal);
                    }
                } else {
                    showToast("Gagal
mengunggah data!",
R.layout.custom_toast_gagal);
                }
            }

            @Override
            public void
onFailure(@NotNull Call<Data> call,
@NotNull Throwable t) {
                progress.hide();
                showToast("Tidak Ada
Koneksi Internet, Silahkan coba lagi!",
R.layout.custom_toast_gagal);
            }
        });
    }

    private void tampilGambarDialog() {
        String[] options = {"Kamera",
"Galeri"};
        AlertDialog.Builder builder =
new
AlertDialog.Builder(LaporanActivity.thi
s);
        builder.setTitle("Pilih berkas
dari:");
        builder.setItems(options,
(dialog, which) -> {
            if (which == 0) {
                if
(!checkCameraPermission()) {

requestCameraPermission();
                } else {
                    openCamera();
                }
            } else if (which == 1) {
                if
(!checkStoragePermission()) {

requestStoragePermission();
                } else {
                    openGallery();
                }
            }
        });
        builder.create().show();
    }

    private Boolean
checkStoragePermission() {
        if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.TIRAMISU) {
            return
ContextCompat.checkSelfPermission(Lapor
anActivity.this,
Manifest.permission.READ_MEDIA_IMAGES)
== PackageManager.PERMISSION_GRANTED &&

ContextCompat.checkSelfPermission(Lapor
anActivity.this,
Manifest.permission.READ_MEDIA_VIDEO)
== PackageManager.PERMISSION_GRANTED &&

ContextCompat.checkSelfPermission(Lapor
anActivity.this,
Manifest.permission.READ_MEDIA_AUDIO)
==   PackageManager.PERMISSION_GRANTED;
        } else {
            return
ContextCompat.checkSelfPermission(Lapor
anActivity.this,
Manifest.permission.WRITE_EXTERNAL_STOR
AGE) ==
PackageManager.PERMISSION_GRANTED &&

ContextCompat.checkSelfPermission(Lapor
anActivity.this,
Manifest.permission.READ_EXTERNAL_STORA
GE) ==
PackageManager.PERMISSION_GRANTED;
        }
    }

    private void
requestStoragePermission() {
        if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.TIRAMISU) {
            requestPermissions(new
String[]{

Manifest.permission.READ_MEDIA_IMAGES,

Manifest.permission.READ_MEDIA_VIDEO,

Manifest.permission.READ_MEDIA_AUDIO
            }, STORAGE_REQUEST);
        } else {
            requestPermissions(new
String[]{

Manifest.permission.WRITE_EXTERNAL_STOR
AGE,

Manifest.permission.READ_EXTERNAL_STORA
GE
            }, STORAGE_REQUEST);
        }
    }

    private Boolean
checkCameraPermission() {
        return
ContextCompat.checkSelfPermission(Lapor
anActivity.this,
Manifest.permission.CAMERA) ==
PackageManager.PERMISSION_GRANTED;
    }

    private void
requestCameraPermission() {
        requestPermissions(new
String[]{Manifest.permission.CAMERA},
CAMERA_REQUEST);
    }
```

```
    private void openCamera() {
        ContentValues values = new ContentValues();
        values.put(MediaStore.Images.Media.TITLE, "New Picture");
        values.put(MediaStore.Images.Media.DESCRIPTION, "From the Camera");
        imageUri = LaporanActivity.this.getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, values);
        Intent takePicture = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        takePicture.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
        startActivityForResult(takePicture, IMAGE_CAPTURE_CODE);
    }

    private void openGallery() {
        Intent pickPhoto = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        startActivityForResult(pickPhoto, PICK_IMAGE_CODE);
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == Activity.RESULT_OK) {
            if (requestCode == IMAGE_CAPTURE_CODE) {
                unggahfoto.setImageURI(imageUri);
                part_image = getPathFromUri(imageUri);
            } else if (requestCode == PICK_IMAGE_CODE && data != null) {
                Uri dataImage = data.getData();
                unggahfoto.setImageURI(dataImage);
                part_image = getPathFromUri(dataImage);
            }
        }
    }

    private String getPathFromUri(Uri uri) {
        String[] projection = {MediaStore.Images.Media.DATA};
        Cursor cursor = LaporanActivity.this.getContentResolver().query(uri, projection, null, null, null);
        if (cursor != null) {
            cursor.moveToFirst();
            int index = cursor.getColumnIndexOrThrow(projection[0]);
            String path = cursor.getString(index);
            cursor.close();
            return path;
        }
        return null;
    }

    private void setErrorState(EditText editText, TextView errorTextView, int drawableId, String errorMessage) {
        editText.setBackgroundResource(R.drawable.custom_edittext_required);
        Drawable draw = ContextCompat.getDrawable(LaporanActivity.this, drawableId);
        Objects.requireNonNull(draw).setColorFilter(new PorterDuffColorFilter(Color.parseColor("#FFCC0000"), PorterDuff.Mode.SRC_IN));
        editText.setCompoundDrawablesWithIntrinsicBounds(draw, null, null, null);
        errorTextView.setVisibility(View.VISIBLE);
        errorTextView.setText(errorMessage);
    }

    private void clearErrorState(EditText editText, TextView errorTextView, int drawableId) {
        editText.setBackgroundResource(R.drawable.custom_edittext);
        Drawable draw = ContextCompat.getDrawable(LaporanActivity.this, drawableId);
        Objects.requireNonNull(draw).setColorFilter(new PorterDuffColorFilter(Color.parseColor("#2E190A"), PorterDuff.Mode.SRC_IN));
        editText.setCompoundDrawablesWithIntrinsicBounds(draw, null, null, null);
        errorTextView.setVisibility(View.GONE);
    }

    private void showToast(String
```

```
message,      int      layoutId)      {
        final   Toast   toast   =   new
Toast(LaporanActivity.this);

toast.setDuration(Toast.LENGTH_LONG);
        @SuppressLint("InflateParams")
View            custom_view            =
getLayoutInflater().inflate(layoutId,
null);
        TextView       tvMessage       =
custom_view.findViewById(R.id.toast_mes
sage);
        tvMessage.setText(message);
        toast.setView(custom view);
        toast.show();
    }
}
```

LoginActivity.java

```
package
com.kelas c.a220280075.bhabinkamtibnas.
activity;

import android.annotation.SuppressLint;
import          android.content.Intent;
import          android.graphics.Color;
import     android.graphics.PorterDuff;
import
android.graphics.PorterDuffColorFilter;
import
android.graphics.drawable.Drawable;
import               android.os.Bundle;
import           android.text.Editable;
import          android.text.TextUtils;
import         android.text.TextWatcher;
import
android.text.method.PasswordTransformat
ionMethod;
import               android.view.View;
import            android.widget.Button;
import          android.widget.CheckBox;
import    android.widget.CompoundButton;
import          android.widget.EditText;
import          android.widget.TextView;
import            android.widget.Toast;

import     androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.core.content.ContextCompat;

import
com.kelas_c.a220280075.bhabinkamtibnas.
R;
import
com.kelas_c.a220280075.bhabinkamtibnas.
SessionManager;
import
com.kelas c.a220280075.bhabinkamtibnas.
api.ApiClient;
import
com.kelas c.a220280075.bhabinkamtibnas.
api.ApiInterface;
import
com.kelas_c.a220280075.bhabinkamtibnas.
model.Data;
import
com.kelas c.a220280075.bhabinkamtibnas.
model.ResultData;

import               java.util.Objects;

import                  retrofit2.Call;
import              retrofit2.Callback;
import              retrofit2.Response;

public   class   LoginActivity   extends
AppCompatActivity                    {

    private  EditText  usernameEditText;
    private  EditText  passwordEditText;
    String         user,          pass;
    ApiInterface          apiInterface;
    SessionManager       sessionManager;

    @Override
    protected    void    onCreate(Bundle
savedInstanceState)                  {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_login)
;

        usernameEditText              =
findViewById(R.id.username);
        passwordEditText              =
findViewById(R.id.password);
        CheckBox showPasswordCheckBox =
findViewById(R.id.showPasswordCheckBox)
;
        Button       loginButton       =
findViewById(R.id.loginButton);
        TextView  usernameErrorTextView
=     findViewById(R.id.usernameError);
        TextView  passwordErrorTextView
=     findViewById(R.id.passwordError);

usernameEditText.addTextChangedListener
(new          TextWatcher()          {
            @Override
            public                  void
beforeTextChanged(CharSequence  s,  int
start,   int   count,   int   after)   {

            }

            @Override
            public                  void
onTextChanged(CharSequence    s,    int
start,   int   before,   int   count)   {

clearErrorState(usernameEditText,
usernameErrorTextView,
R.drawable.baseline_person);
```

```
            }

            @Override
            public                  void
afterTextChanged(Editable      s)      {

            }
        });

passwordEditText.addTextChangedListener
(new           TextWatcher()           {
            @Override
            public                  void
beforeTextChanged(CharSequence   s,   int
start,   int   count,   int   after)   {

            }

            @Override
            public                  void
onTextChanged(CharSequence     s,     int
start,   int   before,   int   count)   {

clearErrorState(passwordEditText,
passwordErrorTextView,
R.drawable.baseline_lock);
            }

            @Override
            public                  void
afterTextChanged(Editable      s)      {

            }
        });


loginButton.setOnClickListener(new
View.OnClickListener()                  {
            @Override
            public void onClick(View v)
{
                user                    =
usernameEditText.getText().toString().t
rim();
                pass                    =
passwordEditText.getText().toString().t
rim();

                boolean isUsernameEmpty
=               TextUtils.isEmpty(user);
                boolean isPasswordEmpty
=               TextUtils.isEmpty(pass);

                if  (isUsernameEmpty  ||
isPasswordEmpty)                        {
                    if
(isUsernameEmpty)                       {

setErrorState(usernameEditText,
usernameErrorTextView,
R.drawable.baseline_person,    "Masukkan
Username!");
                    }       else        {

clearErrorState(usernameEditText,
usernameErrorTextView,
R.drawable.baseline_person);
                    }

                    if
(isPasswordEmpty)                       {

setErrorState(passwordEditText,
passwordErrorTextView,
R.drawable.baseline_lock,      "Masukkan
Password!");
                    }       else        {

clearErrorState(passwordEditText,
passwordErrorTextView,
R.drawable.baseline_lock);
                    }
                    return;
                }
                masuk(user,       pass);
            }
        });


showPasswordCheckBox.setOnCheckedChange
Listener(new
CompoundButton.OnCheckedChangeListener(
)                                       {
            @Override
            public                  void
onCheckedChanged(CompoundButton
buttonView,    boolean    isChecked)    {
                if     (isChecked)     {

passwordEditText.setTransformationMetho
d(null);
                }         else          {

passwordEditText.setTransformationMetho
d(new   PasswordTransformationMethod());
                }
            }
        });
    }

    private   void   masuk(String   user,
String              pass)               {
        apiInterface                    =
ApiClient.getApiClient().create(ApiInte
rface.class);
        Call<Data>      masukCall       =
apiInterface.masukRespone(user,   pass);
        masukCall.enqueue(new
Callback<Data>()                        {
            @Override
            public                  void
onResponse(@NonNull   Call<Data>   call,
@NonNull   Response<Data>   response)   {
                if
(response.isSuccessful()              &&
response.body()      !=      null)      {
                    Data responseData =
response.body();
                    if
(responseData.isStatus())               {
```

```
                        ResultData
rsData = responseData.getResultData();
                        if (rsData !=
null)                                   {

sessionManager          =          new
SessionManager(LoginActivity.this);

sessionManager.createLoginSession(

rsData.getUserAnggota(),

rsData.getNamaAnggota()
                                );
                                Intent
intent = new Intent(LoginActivity.this,
HomeActivity.class);

startActivity(intent);
                                finish();
                            }
                        }       else       {

showToast(responseData.getMessage(),
R.layout.custom_toast_gagal);
                        }
                    }         else         {
                        showToast("Terjadi
kesalahan,    silakan    coba    lagi.",
R.layout.custom_toast_gagal);
                    }
                }

                @Override
                public                  void
onFailure(@NonNull    Call<Data>    call,
@NonNull       Throwable       t)       {
                    showToast("Tidak    ada
koneksi internet, silahkan coba lagi!",
R.layout.custom_toast_gagal);
                }
            });
        }

    private void setErrorState(EditText
editText, TextView errorTextView, int
drawableId,   String   errorMessage)   {

editText.setBackgroundResource(R.drawab
le.custom_edittext_required);
        Drawable         draw         =
ContextCompat.getDrawable(LoginActivity
.this,                      drawableId);

Objects.requireNonNull(draw).setColorFi
lter(new
PorterDuffColorFilter(Color.parseColor(
"#FFCC0000"), PorterDuff.Mode.SRC_IN));

editText.setCompoundDrawablesWithIntrin
sicBounds(draw,   null,   null,   null);

errorTextView.setVisibility(View.VISIBL
E);
errorTextView.setText(errorMessage);
    }

    private                         void
clearErrorState(EditText      editText,
TextView errorTextView, int drawableId)
{

editText.setBackgroundResource(R.drawab
le.custom_edittext);
        Drawable         draw         =
ContextCompat.getDrawable(LoginActivity
.this,                      drawableId);

Objects.requireNonNull(draw).setColorFi
lter(new
PorterDuffColorFilter(Color.parseColor(
"#2E190A"),    PorterDuff.Mode.SRC_IN));

editText.setCompoundDrawablesWithIntrin
sicBounds(draw,   null,   null,   null);

errorTextView.setVisibility(View.GONE);
    }

    private    void    showToast(String
message,      int      layoutId)      {
        final   Toast   toast   =   new
Toast(getApplicationContext());

toast.setDuration(Toast.LENGTH_LONG);
        @SuppressLint("InflateParams")
View            custom_view            =
getLayoutInflater().inflate(layoutId,
null);
        TextView       tvMessage       =
custom view.findViewById(R.id.toast mes
sage);
        tvMessage.setText(message);
        toast.setView(custom_view);
        toast.show();
    }
}
```

LokasiActivity.java

```
package
com.kelas_c.a220280075.bhabinkamtibnas.
activity;

import android.annotation.SuppressLint;
import                android.os.Bundle;
import                 android.util.Log;
import           android.graphics.Color;
import          android.graphics.Bitmap;
import
android.graphics.drawable.BitmapDrawabl
e;
import                 android.Manifest;
import          android.content.Context;
import
android.content.pm.PackageManager;
import             android.view.Gravity;
```

```
import android.view.MenuItem;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.text.Html;
import android.util.TypedValue;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.ActionBar;
import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.appcompat.widget.Toolbar;
import
androidx.core.app.ActivityCompat;

import
com.google.android.gms.location.FusedLo
cationProviderClient;
import
com.google.android.gms.location.Locatio
nServices;
import
com.google.android.gms.maps.CameraUpdat
eFactory;
import
com.google.android.gms.maps.GoogleMap;
import
com.google.android.gms.maps.OnMapReadyC
allback;
import
com.google.android.gms.maps.SupportMapF
ragment;
import
com.google.android.gms.maps.model.Bitma
pDescriptor;
import
com.google.android.gms.maps.model.Bitma
pDescriptorFactory;
import
com.google.android.gms.maps.model.LatLn
g;
import
com.google.android.gms.maps.model.Marke
r;
import
com.google.android.gms.maps.model.Marke
rOptions;
import
com.google.android.gms.maps.model.Polyl
ineOptions;
import com.google.maps.DirectionsApi;
import com.google.maps.GeoApiContext;
import
com.google.maps.android.PolyUtil;
import
com.google.maps.errors.ApiException;
import
com.google.maps.model.DirectionsResult;
import
com.google.maps.model.TravelMode;
import
com.kelas_c.a220280075.bhabinkamtibnas.
R;

import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
import java.util.Objects;

public class LokasiActivity extends
AppCompatActivity implements
OnMapReadyCallback {

    private static final int
REQUEST_LOCATION_PERMISSION = 1;
    private GoogleMap gmap;
    private FusedLocationProviderClient
fusedLocationClient;
    private String idlaporan,
tgllaporan, alamat, latitude,
longitude;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_lokasi
);

        Bundle extras =
getIntent().getExtras();
        if (extras != null) {
            idlaporan =
extras.getString("IdLaporan");
            tgllaporan =
extras.getString("TanggalLaporan");
            alamat =
extras.getString("AlamatLaporan");
            latitude =
extras.getString("LatLaporan");
            longitude =
extras.getString("LongLaporan");
        }

        fusedLocationClient =
LocationServices.getFusedLocationProvid
erClient(this);
        requestLocationPermission();

        Toolbar toolbar =
findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        ActionBar mAction =
getSupportActionBar();
        assert mAction != null;
        TextView titleTextView = new
TextView(getApplicationContext());
        titleTextView.setText("Detail
Lokasi".toUpperCase());
        titleTextView.setTextSize(17);

titleTextView.setTextColor(Color.WHITE)
;
```

```
titleTextView.setGravity(Gravity.CENTER
);

mAction.setDisplayOptions(ActionBar.DIS
PLAY_SHOW_CUSTOM);

mAction.setCustomView(titleTextView,
new
ActionBar.LayoutParams(ActionBar.Layout
Params.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

mAction.setDisplayHomeAsUpEnabled(true)
;

toolbar.setNavigationOnClickListener(v
-> onBackPressed());
    }

    @Override
    public boolean
onOptionsItemSelected(@NonNull MenuItem
item) {
        super.onBackPressed();
        return
super.onOptionsItemSelected(item);
    }

    private void
requestLocationPermission() {
        if
(ActivityCompat.checkSelfPermission(thi
s,
Manifest.permission.ACCESS_FINE_LOCATIO
N) !=
PackageManager.PERMISSION_GRANTED) {

ActivityCompat.requestPermissions(this,
new
String[]{Manifest.permission.ACCESS_FIN
E_LOCATION},
REQUEST_LOCATION_PERMISSION);
        } else {
            initMap();
        }
    }

    @Override
    public void
onRequestPermissionsResult(int
requestCode, @NonNull String[]
permissions, @NonNull int[]
grantResults) {

super.onRequestPermissionsResult(reques
tCode, permissions, grantResults);
        if (requestCode ==
REQUEST_LOCATION_PERMISSION) {
            if (grantResults.length > 0
&& grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                initMap();
            } else {
                Log.e("MapsActivity",
"Izin lokasi ditolak.");
            }
        }
    }

    private void initMap() {
        SupportMapFragment mapFragment
= (SupportMapFragment)
getSupportFragmentManager().findFragmen
tById(R.id.maps);
        if (mapFragment != null) {

mapFragment.getMapAsync(this);
        }
    }

    @SuppressLint({"MissingPermission",
"PotentialBehaviorOverride"})
    @Override
    public void onMapReady(@NonNull
GoogleMap googleMap) {
        gmap = googleMap;

        if (latitude != null &&
longitude != null) {
            LatLng mapindo = new
LatLng(Double.parseDouble(latitude),
Double.parseDouble(longitude));

@SuppressLint("UseCompatLoadingForDrawa
bles") BitmapDrawable bitmapdraw =
(BitmapDrawable)
getResources().getDrawable(R.drawable.m
ap);
            Bitmap b =
bitmapdraw.getBitmap();
            Bitmap smallMarker =
Bitmap.createScaledBitmap(b, 100, 100,
false);
            BitmapDescriptor icon =
BitmapDescriptorFactory.fromBitmap(smal
lMarker);

            MarkerOptions markerOptions
= new
MarkerOptions().position(mapindo).icon(
icon);
            Marker marker =
gmap.addMarker(markerOptions);
            String titleText =
                    "<div style='text-
align: center;'>" +
                    "<b>Lokasi
Kejadian</b>" +
                    "</div>" +
                    "<div
style='text-align: left;'>" +
                    "<small>ID
Laporan: <b>" + idlaporan +
"</b></small><br/>" +

"<small>Tanggal: <b>" +
formatDateToIndonesian(tgllaporan) +
"</b></small><br/>" +
```

```
"<small>Alamat:    <b>"    +    alamat    +
"</b></small>"                              +
                                "</div>";

Objects.requireNonNull(marker).setTitle
(titleText);

gmap.setInfoWindowAdapter(new
GoogleMap.InfoWindowAdapter()          {
                @Override
                public              View
getInfoWindow(@NonNull Marker marker) {
                    return          null;
                }

                @Override
                public              View
getInfoContents(@NonNull Marker marker)
{
                    Context mContext =
getApplicationContext();
                    LinearLayout info =
new              LinearLayout(mContext);

info.setOrientation(LinearLayout.VERTIC
AL);
                    TextView   title   =
new                 TextView(mContext);

title.setText(Html.fromHtml(marker.getT
itle()));

title.setTextColor(Color.BLACK);

title.setBackgroundColor(Color.WHITE);

title.setTextSize(TypedValue.COMPLEX_UN
IT_SP,                                17);

info.addView(title);
                    return          info;
                }
            });

gmap.moveCamera(CameraUpdateFactory.new
LatLngZoom(mapindo,                 16));
        }


fusedLocationClient.getLastLocation().a
ddOnSuccessListener(this, location -> {
            if  (location  !=  null)  {

@SuppressLint("UseCompatLoadingForDrawa
bles") BitmapDrawable bitmapdraww    =
(BitmapDrawable)
getResources().getDrawable(R.drawable.i
coppol);
                Bitmap b = bitmapdraww
.getBitmap();
                Bitmap   smallMarker   =
Bitmap.createScaledBitmap(b,  123,  123,
false);
                BitmapDescriptor icon =
BitmapDescriptorFactory.fromBitmap(smal
lMarker);
                LatLng myLocation = new
LatLng(location.getLatitude(),
location.getLongitude());
                gmap.addMarker(new
MarkerOptions().position(myLocation).ic
on(icon).title("Lokasi     Anda     saat
ini"));
                if (latitude != null &&
longitude       !=       null)       {

addRoute(myLocation,                new
LatLng(Double.parseDouble(latitude),
Double.parseDouble(longitude)));
                }
            }          else          {
                Log.e("MapsActivity",
"Lokasi        tidak        tersedia.");
            }
        });
    }

    private                         String
formatDateToIndonesian(String
originalDate)                           {
        SimpleDateFormat originalFormat
=   new   SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault());
        SimpleDateFormat targetFormat =
new  SimpleDateFormat("dd  MMMM  yyyy",
new         Locale("id",         "ID"));
        Date                        date;
        String   formattedDate   =   "";
        try                             {
            date                        =
originalFormat.parse(originalDate);
            formattedDate               =
targetFormat.format(date);
        } catch  (ParseException  e)  {
            e.printStackTrace();
        }
        return            formattedDate;
    }

    private     void     addRoute(LatLng
origin,     LatLng     destination)     {
        new      Thread(()      ->      {
            try                         {
                GeoApiContext
geoApiContext          =          new
GeoApiContext.Builder().apiKey("AIzaSyA
BGq7KsTzC3lhlJvGvfBqr3GHITiDWX4A").buil
d();
                DirectionsResult result
=
DirectionsApi.newRequest(geoApiContext)

.mode(TravelMode.DRIVING)
                        .origin(new
com.google.maps.model.LatLng(origin.lat
itude,                origin.longitude))

.destination(new
com.google.maps.model.LatLng(destinatio
```

```
n.latitude,      destination.longitude))
                        .await();
                runOnUiThread(() -> {
                    if (result != null
&&    result.routes    !=    null    &&
result.routes.length     >     0     &&
result.routes[0].overviewPolyline    !=
null)                                {
                        PolylineOptions
polylineOptions = new PolylineOptions()

.addAll(PolyUtil.decode(result.routes[0
].overviewPolyline.getEncodedPath()))

.color(Color.BLUE)

.width(10); // You can change the width
as                                needed

gmap.addPolyline(polylineOptions);
                    }       else       {

Log.e("MapsActivity", "Tidak ada hasil
rute           yang           valid.");
                    }
                });
            }  catch  (ApiException  |
InterruptedException | IOException e) {
                e.printStackTrace();
                Log.e("MapsActivity",
"Gagal    memperoleh    rute:    "    +
e.getMessage());
            }
        }).start();
    }
}
```

PreviewGambarActivity.java

```
package
com.kelas c.a220280075.bhabinkamtibnas.
activity;

import                              static
com.kelas c.a220280075.bhabinkamtibnas.
api.ApiClient.BASE_URL;

import          android.content.Intent;
import          android.graphics.Color;
import               android.os.Bundle;
import            android.view.Gravity;
import     android.widget.LinearLayout;
import         android.widget.TextView;

import
androidx.appcompat.app.ActionBar;
import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.appcompat.widget.Toolbar;

import        com.bumptech.glide.Glide;
import
com.bumptech.glide.load.engine.DiskCach
eStrategy;
import
com.github.chrisbanes.photoview.PhotoVi
ew;
import
com.kelas c.a220280075.bhabinkamtibnas.
R;

public    class    PreviewGambarActivity
extends       AppCompatActivity       {

    String                          idlap;

    @Override
    protected    void    onCreate(Bundle
savedInstanceState)                    {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_previe
w_gambar);

        Intent    get    =    getIntent();
        idlap                          =
get.getStringExtra("IdLaporan");

        PhotoView       imgidlap       =
findViewById(R.id.imgdetail);

Glide.with(PreviewGambarActivity.this).
load(BASE_URL   +   "laporan/"   +   idlap
+".jpg").skipMemoryCache(true).diskCach
eStrategy(DiskCacheStrategy.NONE).into(
imgidlap);

        Toolbar        toolbar        =
findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        ActionBar       mAction       =
getSupportActionBar();
        assert    mAction    !=    null;
        TextView  titleTextView  =  new
TextView(getApplicationContext());
        titleTextView.setText("Preview
Gambar".toUpperCase());
        titleTextView.setTextSize(17);

titleTextView.setTextColor(Color.WHITE)
;

titleTextView.setGravity(Gravity.CENTER
);
        TextView subtitleTextView = new
TextView(getApplicationContext());
        subtitleTextView.setText("ID
Lapor:        "        +        idlap);

subtitleTextView.setTextSize(13);

subtitleTextView.setTextColor(Color.WHI
TE);

subtitleTextView.setGravity(Gravity.CEN
```

```
TER);
        LinearLayout titleLayout = new
LinearLayout(getApplicationContext());

titleLayout.setOrientation(LinearLayout
.VERTICAL);
        LinearLayout.LayoutParams
params               =               new
LinearLayout.LayoutParams(LinearLayout.
LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT)
;
        params.gravity                  =
Gravity.CENTER;

titleLayout.addView(titleTextView,
params);

titleLayout.addView(subtitleTextView,
params);

mAction.setDisplayOptions(ActionBar.DIS
PLAY_SHOW_CUSTOM);

mAction.setCustomView(titleLayout,   new
ActionBar.LayoutParams(ActionBar.Layout
Params.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));

mAction.setDisplayHomeAsUpEnabled(true)
;

toolbar.setNavigationOnClickListener(v
->                      onBackPressed());
    }
}
```

SplashActivity.java

```
package
com.kelas_c.a220280075.bhabinkamtibnas.
activity;

import android.annotation.SuppressLint;
import          android.content.Intent;
import               android.os.Bundle;
import              android.os.Handler;
import             android.view.Window;
import      android.view.WindowManager;

import
androidx.appcompat.app.AppCompatActivit
y;

import
com.kelas c.a220280075.bhabinkamtibnas.
R;

@SuppressLint("CustomSplashScreen")
public   class   SplashActivity   extends
AppCompatActivity                       {

    private     static     final     int
SPLASH_DURATION         =         1000;

    @Override
    protected    void    onCreate(Bundle
savedInstanceState)                     {

super.onCreate(savedInstanceState);


requestWindowFeature(Window.FEATURE_NO_
TITLE);

getWindow().setFlags(WindowManager.Layo
utParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCR
EEN);


setContentView(R.layout.activity splash
);
        new Handler().postDelayed(() ->
{
            Intent    intent    =    new
Intent(SplashActivity.this,
HomeActivity.class);
            startActivity(intent);
            finish();
        },             SPLASH_DURATION);
    }
}
```

HomePragment.java

```
package
com.kelas_c.a220280075.bhabinkamtibnas.
fragment;

import android.annotation.SuppressLint;
import          android.content.Intent;
import          android.graphics.Color;
import
android.graphics.drawable.Drawable;
import               android.os.Bundle;

import     androidx.annotation.NonNull;
import
androidx.appcompat.app.ActionBar;
import
androidx.appcompat.app.AlertDialog;
import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.appcompat.widget.Toolbar;
import
androidx.core.graphics.drawable.Drawabl
eCompat;
import  androidx.fragment.app.Fragment;
import
androidx.recyclerview.widget.LinearLayo
utManager;
import
```

```
androidx.recyclerview.widget.RecyclerVi
ew;
import
androidx.swiperefreshlayout.widget.Swip
eRefreshLayout;

import                android.text.Html;
import             android.view.Gravity;
import      android.view.LayoutInflater;
import                android.view.Menu;
import        android.view.MenuInflater;
import             android.view.MenuItem;
import                android.view.View;
import           android.view.ViewGroup;
import            android.widget.Button;
import         android.widget.ImageView;
import    android.widget.RelativeLayout;
import          android.widget.TextView;
import             android.widget.Toast;

import
com.google.android.material.floatingact
ionbutton.ExtendedFloatingActionButton;
import
com.kelas c.a220280075.bhabinkamtibnas.
activity.PreviewGambarActivity;
import
com.kelas_c.a220280075.bhabinkamtibnas.
R;
import
com.kelas c.a220280075.bhabinkamtibnas.
SessionManager;
import
com.kelas_c.a220280075.bhabinkamtibnas.
activity.LaporanActivity;
import
com.kelas c.a220280075.bhabinkamtibnas.
activity.LoginActivity;
import
com.kelas_c.a220280075.bhabinkamtibnas.
activity.LokasiActivity;
import
com.kelas c.a220280075.bhabinkamtibnas.
adapter.HomeAdapter;
import
com.kelas_c.a220280075.bhabinkamtibnas.
api.ApiClient;
import
com.kelas_c.a220280075.bhabinkamtibnas.
api.ApiInterface;
import
com.kelas_c.a220280075.bhabinkamtibnas.
model.Data;
import
com.kelas_c.a220280075.bhabinkamtibnas.
model.ResultData;

import
org.jetbrains.annotations.NotNull;

import                    java.util.List;

import                    retrofit2.Call;
import                retrofit2.Callback;
import                retrofit2.Response;

public    class    HomeFragment    extends
Fragment                                {

    private  RecyclerView  recyclerView;
    private           SwipeRefreshLayout
mSwipeRefreshLayout;
    private RelativeLayout errorlayout;
    private      ImageView      imgerror;
    private     TextView     errortitle,
errormessage;
    private       Button       btnretry;
    HomeAdapter                 adapter;
    HomeAdapter.ItemClickListener
itemClickListener;
    List<ResultData>     resultDataList;
    String          user,          nama;
    ApiInterface           apiInterface;
    SessionManager       sessionManager;

    @Override
    public                              View
onCreateView(LayoutInflater    inflater,
ViewGroup       container,        Bundle
savedInstanceState)                     {
        View          rootView          =
inflater.inflate(R.layout.fragment_home
,           container,           false);

        sessionManager       =       new
SessionManager(requireContext());
        user                            =
sessionManager.getUserDetail().get(Sess
ionManager.UserAnggota);
        nama                            =
sessionManager.getUserDetail().get(Sess
ionManager.NamaAnggota);

        errorlayout                     =
rootView.findViewById(R.id.rel_error);
        imgerror                        =
rootView.findViewById(R.id.imgerror);
        errortitle                      =
rootView.findViewById(R.id.tverror);
        errormessage                    =
rootView.findViewById(R.id.tvsuberror);
        btnretry                        =
rootView.findViewById(R.id.btnretry);

        mSwipeRefreshLayout             =
rootView.findViewById(R.id.swaplistlapo
ranhari);
        recyclerView                    =
rootView.findViewById(R.id.listviewlapo
ranhari);

recyclerView.setLayoutManager(new
LinearLayoutManager(requireContext()));

recyclerView.setHasFixedSize(true);
        tampillaporanhari(user);

mSwipeRefreshLayout.setColorSchemeResou
rces(R.color.colorPrimary);
```

```
mSwipeRefreshLayout.setOnRefreshListener(() -> tampillaporanhari(user));

        itemClickListener = new HomeAdapter.ItemClickListener() {
            @Override
            public void OnItemClickLokasi(View view, int position) {
                String idlaporan = resultDataList.get(position).getIdLaporan();
                String almtlaporan = resultDataList.get(position).getAlamatLaporan();
                String latlap = resultDataList.get(position).getLatLaporan();
                String longlap = resultDataList.get(position).getLongLaporan();
                String tgllap = resultDataList.get(position).getTanggalLaporan();
                Intent it = new Intent(requireContext(), LokasiActivity.class);

it.putExtra("IdLaporan", idlaporan);

it.putExtra("AlamatLaporan", almtlaporan);

it.putExtra("LatLaporan", latlap);

it.putExtra("LongLaporan", longlap);

it.putExtra("TanggalLaporan", tgllap);
                startActivity(it);
            }

            @Override
            public void OnItemClickHapusLaporan(View view, int position) {
                String idlaporan = resultDataList.get(position).getIdLaporan();
                String hexColor = "#B10000";
                int color = Color.parseColor(hexColor);
                String kata = "Hapus Laporan dengan ID ";
                String kdnya = "<font color='#0384DA'><b>" + idlaporan + "</b></font> ?";
                AlertDialog alertDialog = new AlertDialog.Builder(requireContext())

.setTitle("Konfirmasi Hapus!")

.setMessage(Html.fromHtml(kata + kdnya))

.setPositiveButton("Ya", (dialog, which) -> {

hapuslaporan(idlaporan, user);

resultDataList.remove(position);

adapter.notifyItemRemoved(position);
                        })

.setNegativeButton("Tidak", null)

.setIcon(R.drawable.ic_close)
                        .show();
                Drawable icon = getResources().getDrawable(R.drawable.ic_info);

DrawableCompat.setTint(icon, color);

alertDialog.setIcon(icon);
            }
        };

        ExtendedFloatingActionButton btnadd = rootView.findViewById(R.id.btnAdd);
        btnadd.setOnClickListener(v -> startActivity(new Intent(requireContext(), LaporanActivity.class)));

        Toolbar toolbar = rootView.findViewById(R.id.toolbar);
        AppCompatActivity activity = (AppCompatActivity) getActivity();
        if (activity != null) {

activity.setSupportActionBar(toolbar);
            ActionBar actionBar = activity.getSupportActionBar();
            if (actionBar != null) {

actionBar.setDisplayShowTitleEnabled(true);

actionBar.setTitle("Laporan".toUpperCase());
                TextView titleTextView = new TextView(activity);

titleTextView.setText(actionBar.getTitle());

titleTextView.setTextSize(17);

titleTextView.setTextColor(Color.WHITE);

titleTextView.setGravity(Gravity.CENTER);

actionBar.setDisplayOptions(ActionBar.DISPLAY_SHOW_CUSTOM);
```

```
actionBar.setCustomView(titleTextView,
new
ActionBar.LayoutParams(ActionBar.Layout
Params.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));
            }
        }

        setHasOptionsMenu(true);
        return                  rootView;
    }

    @Override
    public                          void
onCreateOptionsMenu(Menu          menu,
MenuInflater         inflater)         {

inflater.inflate(R.menu.top_menu,
menu);
    }

    @Override
    public                          boolean
onOptionsItemSelected(MenuItem  item)  {
        if      (item.getItemId()      ==
R.id.logout)                            {
            new
AlertDialog.Builder(requireContext())

.setTitle("Konfirmasi          Keluar!")
                    .setMessage("Apakah
Anda      yakin      ingin      keluar?")

.setPositiveButton("Ya",        (dialog,
which)                ->                {

sessionManager.logoutSession();
                        Intent intent =
new             Intent(requireContext(),
LoginActivity.class);

intent.setFlags(Intent.FLAG_ACTIVITY_NE
W_TASK                                  |
Intent.FLAG_ACTIVITY_CLEAR_TASK);

startActivity(intent);

requireActivity().finish();
                    })

.setNegativeButton("Tidak",        null)

.setIcon(R.drawable.ic_info)
                    .show();
            return                  true;
        }
        return
super.onOptionsItemSelected(item);
    }

    private                         void
tampillaporanhari(String     user)     {
errorlayout.setVisibility(View.GONE);
        showLoading();
        apiInterface                    =
ApiClient.getApiClient().create(ApiInte
rface.class);
        Call<List<ResultData>>  call  =
apiInterface.getLaporanHari(user);
        call.enqueue(new
Callback<List<ResultData>>()          {

            @Override
            public                  void
onResponse(@NonNull
Call<List<ResultData>>  call,  @NonNull
Response<List<ResultData>> response)  {
                hideLoading();

if(response.isSuccessful()            &&
response.body()        !=        null){

onGetResult(response.body());
                }else{
                    String   errorCode;
                    switch
(response.code()){
                        case       404:
                            errorCode  =
"404            not            found";
                            break;
                        case       500:
                            errorCode  =
"500          server          broken";
                            break;
                        default:
                            errorCode  =
"Unknown                        error";
                            break;
                    }

showErrorMessage(R.drawable.no_result,
"No  Result",  "Please  Try  Again!-"+
errorCode);
                }
            }

            @Override
            public                  void
onFailure(@NonNull
Call<List<ResultData>>  call,  @NonNull
Throwable             t)            {
                hideLoading();

onErrorLoading(t.toString());
            }
        });
    }

    private  void  hapuslaporan(String
idlaporan,     String     user)     {
        apiInterface                    =
ApiClient.getApiClient().create(ApiInte
rface.class);
        Call<Data>     masukCall     =
apiInterface.hapuslaporanRespone(idlapo
ran,                            user);
```

```
        masukCall.enqueue(new
Callback<Data>()                        {
            @Override
            public                  void
onResponse(@NotNull   Call<Data>   call,
@NotNull   Response<Data>   response)   {
                if(response.body()   !=
null   &&   response.isSuccessful()   &&
response.body().isStatus())             {

showToast(response.body().getMessage(),
R.layout.custom_toast_sukses);
                }else{
                    assert
response.body()         !=         null;

showToast(response.body().getMessage(),
R.layout.custom_toast_gagal);
                }
            }


@SuppressLint("SetTextI18n")
            @Override
            public                  void
onFailure(@NotNull   Call<Data>   call,
@NotNull       Throwable       t)       {
                showToast("Tidak    ada
koneksi internet, silahkan coba lagi!",
R.layout.custom_toast_gagal);
            }
        });
    }

    private    void    showLoading()    {

mSwipeRefreshLayout.setRefreshing(true)
;
    }

    private    void    hideLoading()    {

mSwipeRefreshLayout.setRefreshing(false
);
    }


@SuppressLint("NotifyDataSetChanged")
    private                         void
onGetResult(List<ResultData> results) {
        adapter          =          new
HomeAdapter(getContext(),       results,
itemClickListener);
        adapter.notifyDataSetChanged();

recyclerView.setAdapter(adapter);
        resultDataList    =    results;
    }

    private  void  onErrorLoading(String
message)                                {

showErrorMessage(R.drawable.no_result,
"Oops...", "Network failure, Please Try
Again!.\n"+                   message);
    }

    private  void  showErrorMessage(int
imageView,    String    title,    String
message){

        if(errorlayout.getVisibility()
==                          View.GONE){

errorlayout.setVisibility(View.VISIBLE)
;
        }


imgerror.setImageResource(imageView);
        errortitle.setText(title);
        errormessage.setText(message);


btnretry.setOnClickListener(view     ->
tampillaporanhari(user));

    }

    private    void    showToast(String
message,      int      layoutId)      {
        final   Toast   toast   =   new
Toast(requireContext());

toast.setDuration(Toast.LENGTH_LONG);
        @SuppressLint("InflateParams")
View           custom_view           =
getLayoutInflater().inflate(layoutId,
null);
        TextView       tvMessage       =
custom_view.findViewById(R.id.toast_mes
sage);
        tvMessage.setText(message);
        toast.setView(custom_view);
        toast.show();
    }
}
```

ProfilFragment.java

```
package
com.kelas c.a220280075.bhabinkamtibnas.
fragment;

import                                static
com.kelas c.a220280075.bhabinkamtibnas.
api.ApiClient.BASE_URL;

import                 android.Manifest;
import  android.annotation.SuppressLint;
import             android.app.Activity;
import     android.app.DatePickerDialog;
import       android.app.ProgressDialog;
import    android.content.ContentValues;
import           android.content.Intent;
import
android.content.pm.PackageManager;
import          android.database.Cursor;
```

```
import android.graphics.Color;
import android.graphics.PorterDuff;
import
android.graphics.PorterDuffColorFilter;
import
android.graphics.drawable.Drawable;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.ActionBar;
import
androidx.appcompat.app.AlertDialog;
import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.appcompat.widget.Toolbar;
import
androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;
import
androidx.recyclerview.widget.LinearLayo
utManager;
import
androidx.recyclerview.widget.RecyclerVi
ew;
import
androidx.swiperefreshlayout.widget.Swip
eRefreshLayout;

import android.os.Handler;
import android.provider.MediaStore;
import android.text.Editable;
import android.text.TextUtils;
import android.text.TextWatcher;
import
android.text.method.PasswordTransformat
ionMethod;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;
import
com.bumptech.glide.load.engine.DiskCach
eStrategy;
import
com.google.android.material.dialog.Mate
rialAlertDialogBuilder;
import
com.kelas_c.a220280075.bhabinkamtibnas.
activity.PreviewGambarActivity;
import
com.kelas_c.a220280075.bhabinkamtibnas.
R;
import
com.kelas_c.a220280075.bhabinkamtibnas.
SessionManager;
import
com.kelas_c.a220280075.bhabinkamtibnas.
activity.LoginActivity;
import
com.kelas_c.a220280075.bhabinkamtibnas.
activity.LokasiActivity;
import
com.kelas_c.a220280075.bhabinkamtibnas.
adapter.ProfilAdapter;
import
com.kelas_c.a220280075.bhabinkamtibnas.
api.ApiClient;
import
com.kelas_c.a220280075.bhabinkamtibnas.
api.ApiInterface;
import
com.kelas c.a220280075.bhabinkamtibnas.
model.Data;
import
com.kelas_c.a220280075.bhabinkamtibnas.
model.ResultData;

import
org.jetbrains.annotations.NotNull;

import java.io.File;
import java.util.Calendar;
import java.util.List;
import java.util.Objects;

import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class ProfilFragment extends
Fragment implements
View.OnClickListener {

    private RecyclerView recyclerView;
    private SwipeRefreshLayout
mSwipeRefreshLayout;
    private RelativeLayout errorlayout;
    private ImageView imgerror;
    private TextView errortitle,
errormessage;
    private Button btnretry;
    private TextView tvnama,
itemcoutlaporan, tvtanggal;
    private ImageView viewfotoanggota;
    private ImageView unggahfoto;
    private ProgressDialog progress;
    ProfilAdapter adapter;
    ProfilAdapter.ItemClickListener
itemClickListener;
```

```
    List<ResultData>     resultDataList;
    String   user,   nama,   part_image;
    ApiInterface          apiInterface;
    SessionManager       sessionManager;
    private     static     final     int
CAMERA_REQUEST          =          100;
    private     static     final     int
STORAGE_REQUEST          =          101;
    private     static     final     int
IMAGE_CAPTURE_CODE        =        102;
    private     static     final     int
PICK_IMAGE_CODE          =          103;
    private        Uri        imageUri;

    @Override
    public                          View
onCreateView(LayoutInflater   inflater,
ViewGroup       container,       Bundle
savedInstanceState)                    {
        View         rootView         =
inflater.inflate(R.layout.fragment_profil,          container,          false);

        sessionManager       =       new
SessionManager(requireContext());
        user                           =
sessionManager.getUserDetail().get(SessionManager.UserAnggota);
        nama                           =
sessionManager.getUserDetail().get(SessionManager.NamaAnggota);

        errorlayout                    =
rootView.findViewById(R.id.rel_error);
        imgerror                       =
rootView.findViewById(R.id.imgerror);
        errortitle                     =
rootView.findViewById(R.id.tverror);
        errormessage                   =
rootView.findViewById(R.id.tvsuberror);
        btnretry                       =
rootView.findViewById(R.id.btnretry);

        tvtanggal                      =
rootView.findViewById(R.id.tvtanggal);
        tvtanggal.setFocusable(false);

tvtanggal.setOnClickListener(new
View.OnClickListener()                 {
            @Override
            public void onClick(View v)
{
                final   Calendar   c   =
Calendar.getInstance();
                int       tahun       =
c.get(Calendar.YEAR);
                int       bulan       =
c.get(Calendar.MONTH);
                int      tanggal      =
c.get(Calendar.DAY_OF_MONTH);
                DatePickerDialog
datePickerDialog         =         new
DatePickerDialog(requireContext(),
R.style.CustomDatePickerDialog,   (view,
year,    month,    dayOfMonth)    ->    {
                    String
formattedDate = String.format("%d-%02d-
%02d", year, month + 1, dayOfMonth);

tvtanggal.setText(formattedDate);
                },     tahun,     bulan,
tanggal);
                int    colorAccent    =
ContextCompat.getColor(requireContext()
,                 R.color.colorAccent);

datePickerDialog.setOnShowListener(dial
og                 ->                 {

datePickerDialog.getButton(DatePickerDialog.BUTTON_POSITIVE).setTextColor(colorAccent);

datePickerDialog.getButton(DatePickerDialog.BUTTON_NEGATIVE).setTextColor(colorAccent);
                });

datePickerDialog.show();
            }
        });
        mSwipeRefreshLayout            =
rootView.findViewById(R.id.swaplistlaporanall);
        recyclerView                   =
rootView.findViewById(R.id.listviewlaporanall);

recyclerView.setLayoutManager(new
LinearLayoutManager(requireContext()));

recyclerView.setHasFixedSize(true);
        tampillaporan(user);

mSwipeRefreshLayout.setColorSchemeResources(R.color.colorPrimary);

mSwipeRefreshLayout.setOnRefreshListener(()                ->                {
            tampillaporan(user);
            tvtanggal.setText("");
        });

        itemClickListener    =    (view,
position)             ->             {
            String      idlaporan      =
resultDataList.get(position).getIdLaporan();
            String     almtlaporan     =
resultDataList.get(position).getAlamatLaporan();
            String       latlap       =
resultDataList.get(position).getLatLaporan();
            String       longlap       =
resultDataList.get(position).getLongLaporan();
            String       tgllap       =
resultDataList.get(position).getTanggalLaporan();
```

```
            Intent     it     =     new
Intent(requireContext(),
LokasiActivity.class);
            it.putExtra("IdLaporan",
idlaporan);

it.putExtra("AlamatLaporan",
almtlaporan);
            it.putExtra("LatLaporan",
latlap);
            it.putExtra("LongLaporan",
longlap);

it.putExtra("TanggalLaporan",   tgllap);
            startActivity(it);
        };

        progress          =          new
ProgressDialog(requireContext());
        progress.setCancelable(false);
        progress.setMessage("Tunggu
Sebentar...");

        viewfotoanggota                =
rootView.findViewById(R.id.img_foto_ang
gota);

Glide.with(requireContext()).load(BASE_
URL    +    "anggota/"    +    user    +
".jpg").placeholder(R.drawable.user).sk
ipMemoryCache(true).diskCacheStrategy(D
iskCacheStrategy.NONE).into(viewfotoang
gota);
        tvnama                         =
rootView.findViewById(R.id.txtnama);
        tvnama.setText(nama);

        itemcoutlaporan                =
rootView.findViewById(R.id.itemcoutlapo
ran);
        badgelaporab(user);
        Handler     handler     =     new
Handler();
        handler.postDelayed(new
Runnable()                             {
            @Override
            public    void    run()    {
                badgelaporab(user);

handler.postDelayed(this,         1000);
            }
        },                             1000);

        ImageView     imgubahfoto     =
rootView.findViewById(R.id.img_ubah_fot
o);

imgubahfoto.setOnClickListener(this);
        ImageView     imgubahnama     =
rootView.findViewById(R.id.img_ubah_dat
a);

imgubahnama.setOnClickListener(this);
        ImageView     caritanggal     =
rootView.findViewById(R.id.caritanggal)
;

caritanggal.setOnClickListener(this);

        Toolbar        toolbar        =
rootView.findViewById(R.id.toolbar);
        AppCompatActivity   activity   =
(AppCompatActivity)       getActivity();
        if    (activity    !=    null)    {

activity.setSupportActionBar(toolbar);
            ActionBar    actionBar    =
activity.getSupportActionBar();
            if   (actionBar   !=   null)   {

actionBar.setDisplayShowTitleEnabled(tr
ue);

actionBar.setTitle("Profil".toUpperCase
());
                TextView   titleTextView
=        new        TextView(activity);

titleTextView.setText(actionBar.getTitl
e());

titleTextView.setTextSize(17);

titleTextView.setTextColor(Color.WHITE)
;

titleTextView.setGravity(Gravity.CENTER
);

actionBar.setDisplayOptions(ActionBar.D
ISPLAY_SHOW_CUSTOM);

actionBar.setCustomView(titleTextView,
new
ActionBar.LayoutParams(ActionBar.Layout
Params.WRAP_CONTENT,
ActionBar.LayoutParams.WRAP_CONTENT,
Gravity.CENTER));
            }
        }

        setHasOptionsMenu(true);
        return                    rootView;
    }

    @Override
    public                              void
onCreateOptionsMenu(Menu           menu,
MenuInflater         inflater)         {

inflater.inflate(R.menu.top_menu,
menu);
    }

    @Override
    public                            boolean
onOptionsItemSelected(MenuItem  item)  {
        if      (item.getItemId()      ==
R.id.logout)                           {
            new
```

```
AlertDialog.Builder(requireContext())

.setTitle("Konfirmasi          Keluar!")
                    .setMessage("Apakah
Anda      yakin      ingin      keluar?")

.setPositiveButton("Ya",        (dialog,
which)                ->              {

sessionManager.logoutSession();
                        Intent intent =
new             Intent(requireContext(),
LoginActivity.class);

intent.setFlags(Intent.FLAG_ACTIVITY_NE
W_TASK                                  |
Intent.FLAG_ACTIVITY_CLEAR_TASK);

startActivity(intent);

requireActivity().finish();
                    })

.setNegativeButton("Tidak",        null)

.setIcon(R.drawable.ic_info)
                    .show();
            return                  true;
        }
        return
super.onOptionsItemSelected(item);
    }

    @Override
    public   void   onClick(View   v)   {
        if          (v.getId()          ==
R.id.img ubah foto)                     {
            formubahfoto();
        }   else   if   (v.getId()   ==
R.id.img_ubah_data)                     {
            profilDialog();
        }   else   if   (v.getId()   ==
R.id.caritanggal)                       {
            String      tgllapor      =
tvtanggal.getText().toString().trim();
            if
(!TextUtils.isEmpty(tgllapor))          {

tampillaporanBerdasarkanTanggal(user,
tgllapor);
            }           else            {
                showToast("Silahkan
Pilih                       Tanggal",
R.layout.custom toast info);
            }
        }
    }

    private   void   formubahfoto()   {
        LayoutInflater    inflater    =
getLayoutInflater();
        View        customView        =
inflater.inflate(R.layout.item_dialog_u
bahfoto,                        null);
        TextView      juduldialog      =
customView.findViewById(R.id.txtdialjud
ul);
        unggahfoto                      =
customView.findViewById(R.id.imgunggahb
erkas);

unggahfoto.setOnClickListener(v3      ->
tampilGambarDialog());
        Button       btnsimpan       =
customView.findViewById(R.id.btnsimpan)
;
        juduldialog.setText("Ubah
Foto");

juduldialog.setTextAlignment(View.TEXT_
ALIGNMENT_CENTER);

juduldialog.setTextColor(Color.parseCol
or("#FFFFFF"));

juduldialog.setBackgroundColor(Color.pa
rseColor("#6F543C"));
        AlertDialog   dialog   =   new
MaterialAlertDialogBuilder(requireConte
xt(),
R.style.RoundedMaterialDialog).setView(
customView).show();
        btnsimpan.setOnClickListener(v1
->                                      {
            if(unggahfoto.getDrawable()
==                                  null){
                showToast("Belum    Ada
File          Yang          Dipilih",
R.layout.custom_toast_info);
            }else{
                unggahfoto();
                dialog.dismiss();
            }
        });

        dialog.setCancelable(false);

customView.findViewById(R.id.tutupdialo
g).setOnClickListener(v2              ->
dialog.dismiss());
    }

    private   void   profilDialog()   {
        String[]   options   =   {"Nama",
"Password"};
        AlertDialog.Builder   builder   =
new
AlertDialog.Builder(requireContext());
        builder.setTitle("Ubah Data:");
        builder.setItems(options,
(dialog,        which)        ->        {
            if    (which    ==    0)    {
                formubahdata();
            } else if (which == 1) {
                formubahpassword();
            }
        });
        builder.create().show();
    }
```

```
    private void formubahdata() {
        LayoutInflater inflater = getLayoutInflater();
        View customView = inflater.inflate(R.layout.item_dialog_ubahdata, null);
        TextView juduldialog = customView.findViewById(R.id.txtdialjudul);
        EditText nama = customView.findViewById(R.id.nama);
        TextView namaError = customView.findViewById(R.id.namaError);
        nama.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
                clearErrorState(nama, namaError, R.drawable.baseline_person);
            }

            @Override
            public void afterTextChanged(Editable s) {

            }
        });
        Button btnsimpan = customView.findViewById(R.id.btnsimpan);
        juduldialog.setText("Ubah Nama");

judul dialog.setTextAlignment(View.TEXT_ALIGNMENT_CENTER);

judul dialog.setTextColor(Color.parseColor("#FFFFFF"));

judul dialog.setBackgroundColor(Color.parseColor("#6F543C"));
        AlertDialog dialog = new MaterialAlertDialogBuilder(requireContext(), R.style.RoundedMaterialDialog).setView(customView).show();
        btnsimpan.setOnClickListener(v1 -> {
            String inputnama = nama.getText().toString().trim();
            if (TextUtils.isEmpty(inputnama)) {
                setErrorState(nama, namaError, R.drawable.baseline_person, "Masukkan Nama!");
                return;
            }
            ubahdata(inputnama, user);
            dialog.dismiss();
        });
        dialog.setCancelable(false);

dialog.findViewById(R.id.tutupdialog).setOnClickListener(v2 -> dialog.dismiss());
    }

    private void formubahpassword() {
        LayoutInflater inflater = getLayoutInflater();
        View customView = inflater.inflate(R.layout.item_dialog_ubahpass, null);
        TextView juduldialog = customView.findViewById(R.id.txtdialjudul);
        EditText passlama = customView.findViewById(R.id.passwordlama);
        TextView passlamaError = customView.findViewById(R.id.passwordlamaError);

passlama.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {

            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {

clearErrorState(passlama, passlamaError, R.drawable.baseline_lock);
            }

            @Override
            public void afterTextChanged(Editable s) {

            }
        });
        EditText passbaru = customView.findViewById(R.id.passwordbaru);
        TextView passbaruError = customView.findViewById(R.id.passwordbaruError);

passbaru.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int
```

```
start, int count, int after) {

            }

            @Override
            public void
onTextChanged(CharSequence s, int
start, int before, int count) {

clearErrorState(passbaru,
passbaruError,
R.drawable.baseline_lock);
            }

            @Override
            public void
afterTextChanged(Editable s) {

            }
        });
        EditText passbaruulang =
customView.findViewById(R.id.passwordba
ruulangi);
        TextView passbaruulangError =
customView.findViewById(R.id.passwordba
ruulangiError);

passbaruulang.addTextChangedListener(ne
w TextWatcher() {
            @Override
            public void
beforeTextChanged(CharSequence s, int
start, int count, int after) {

            }

            @Override
            public void
onTextChanged(CharSequence s, int
start, int before, int count) {

clearErrorState(passbaruulang,
passbaruulangError,
R.drawable.baseline_lock);
            }

            @Override
            public void
afterTextChanged(Editable s) {

            }
        });
        CheckBox
showPasswordlamaCheckBox =
customView.findViewById(R.id.showPasswo
rdlamaCheckBox);

showPasswordlamaCheckBox.setOnCheckedCh
angeListener(new
CompoundButton.OnCheckedChangeListener(
) {
            @Override
            public void
onCheckedChanged(CompoundButton
buttonView, boolean isChecked) {
                if (isChecked) {

passlama.setTransformationMethod(null);
                } else {

passlama.setTransformationMethod(new
PasswordTransformationMethod());
                }
            }
        });
        CheckBox
showPasswordBaruCheckBox =
customView.findViewById(R.id.showPasswo
rdBaruCheckBox);

showPasswordBaruCheckBox.setOnCheckedCh
angeListener(new
CompoundButton.OnCheckedChangeListener(
) {
            @Override
            public void
onCheckedChanged(CompoundButton
buttonView, boolean isChecked) {
                if (isChecked) {

passbaru.setTransformationMethod(null);
                } else {

passbaru.setTransformationMethod(new
PasswordTransformationMethod());
                }
            }
        });
        CheckBox
showPasswordBaruUlangiCheckBox =
customView.findViewById(R.id.showPasswo
rdBaruUlangiCheckBox);

showPasswordBaruUlangiCheckBox.setOnChe
ckedChangeListener(new
CompoundButton.OnCheckedChangeListener(
) {
            @Override
            public void
onCheckedChanged(CompoundButton
buttonView, boolean isChecked) {
                if (isChecked) {

passbaruulang.setTransformationMethod(n
ull);
                } else {

passbaruulang.setTransformationMethod(n
ew PasswordTransformationMethod());
                }
            }
        });
        Button btnsimpan =
customView.findViewById(R.id.btnsimpan)
;
        juduldialog.setText("Ubah
Password");

juduldialog.setTextAlignment(View.TEXT_
ALIGNMENT_CENTER);
```

```
judulldialog.setTextColor(Color.parseColor("#FFFFFF"));

judulldialog.setBackgroundColor(Color.parseColor("#6F543C"));
        AlertDialog dialog = new MaterialAlertDialogBuilder(requireContext(), R.style.RoundedMaterialDialog).setView(customView).show();
        btnsimpan.setOnClickListener(v1 ->                                  {
            String       pslama       = passlama.getText().toString().trim();
            String       psbaru       = passbaru.getText().toString().trim();
            String     psbaruulang     = passbaruulang.getText().toString().trim();

            boolean  ispasslamaEmpty  = TextUtils.isEmpty(pslama);
            boolean  ispassbaruEmpty  = TextUtils.isEmpty(psbaru);
            boolean ispassbaruulangEmpty                    = TextUtils.isEmpty(psbaruulang);

            if    (ispasslamaEmpty    || ispassbaruEmpty                          || ispassbaruulangEmpty)                {
                if  (ispasslamaEmpty)  {

setErrorState(passlama,  passlamaError, R.drawable.baseline_lock,      "Masukkan Password                          Lama!");
                }         else         {

clearErrorState(passlama, passlamaError, R.drawable.baseline_lock);
                }

                if  (ispassbaruEmpty)  {

setErrorState(passbaru,  passbaruError, R.drawable.baseline_lock,      "Masukkan Password                          Baru!");
                }         else         {

clearErrorState(passbaru, passbaruError, R.drawable.baseline lock);
                }

                if (ispassbaruulangEmpty)                {

setErrorState(passbaruulang, passbaruulangError, R.drawable.baseline_lock,      "Masukkan Ulangi         Password         Baru!");
                }         else         {

clearErrorState(passbaruulang, passbaruulangError, R.drawable.baseline_lock);
                }
                return;
            }

            if (!isValidPassword(psbaru))            {
                setErrorState(passbaru, passbaruError, R.drawable.baseline_lock,      "Password harus  memiliki  Huruf  Besar,  Kecil, Angka           dan           Simbol!");
                return;
            }           else           {

clearErrorState(passbaru, passbaruError, R.drawable.baseline lock);
            }

            if (!psbaru.equals(psbaruulang))          {

setErrorState(passbaruulang, passbaruulangError, R.drawable.baseline_lock,      "Password Baru  dan  Konfirmasi  tidak  cocok!");
                return;
            }           else           {

clearErrorState(passbaruulang, passbaruulangError, R.drawable.baseline_lock);
            }
            ubahpass(pslama,     psbaru, psbaruulang,                          user);
            dialog.dismiss();
        });
        dialog.setCancelable(false);

dialog.findViewById(R.id.tutupdialog).setOnClickListener(v2                    -> dialog.dismiss());
    }

    private                         boolean isValidPassword(String    password)    {
        String     passwordPattern     = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[@$!%*?&])[A-Za-z\\d@$!%*?&]{8,}$";
        return password.matches(passwordPattern);
    }

    private      void      unggahfoto(){
        progress.show();
        File     imagefile     =     new File(part_image);
        RequestBody       idnya       = RequestBody.create(MultipartBody.FORM,user);
        RequestBody      reqBody      =
```

```
RequestBody.create(MediaType.parse("multipart/form-file"),imagefile);
        MultipartBody.Part partImage = MultipartBody.Part.createFormData("UnggahFotoAnggota",
imagefile.getName(),reqBody);

        apiInterface                    =
ApiClient.getApiClient().create(ApiInterface.class);
        Call<Data>      masukCall      =
apiInterface.unggahfotoRespone(idnya,partImage);
        masukCall.enqueue(new
Callback<Data>()                        {
            @Override
            public                  void
onResponse(@NotNull   Call<Data>   call,
@NotNull   Response<Data>   response)   {
                progress.hide();
                if(response.body()    !=
null   &&   response.isSuccessful()   &&
response.body().isStatus())             {

showToast(response.body().getMessage(),
R.layout.custom_toast_sukses);

Glide.with(requireContext()).load(BASE_URL    +    "anggota/"    +    user    +
".jpg").placeholder(R.drawable.user).skipMemoryCache(true).diskCacheStrategy(DiskCacheStrategy.NONE).into(viewfotoanggota);
                }else{

showToast(response.body().getMessage(),
R.layout.custom toast gagal);
                }
            }


@SuppressLint("SetTextI18n")
            @Override
            public                  void
onFailure(@NotNull   Call<Data>   call,
@NotNull       Throwable       t)       {
                progress.hide();
                showToast("Tidak    Ada
Koneksi Internet, Silahkan coba lagi!",
R.layout.custom_toast_gagal);
            }
        });
    }

    private                         void
updateProfileInfo(String   newName)   {
        Handler     handler     =     new
Handler();
        handler.post(()       ->       {
            tvnama.setText(newName);
        });
    }

    private     void     ubahdata(String
namaubah,      String      iduser)      {
        progress.show();
        apiInterface                    =
ApiClient.getApiClient().create(ApiInterface.class);
        Call<Data>      masukCall      =
apiInterface.ubahdataRespone(namaubah,
iduser);
        masukCall.enqueue(new
Callback<Data>()                        {
            @Override
            public                  void
onResponse(@NotNull   Call<Data>   call,
@NotNull   Response<Data>   response)   {
                progress.hide();
                if(response.body()    !=
null   &&   response.isSuccessful()   &&
response.body().isStatus())             {
                    sessionManager     =
new    SessionManager(requireContext());

sessionManager.createLoginSession(user,
namaubah);

showToast(response.body().getMessage(),
R.layout.custom_toast_sukses);

updateProfileInfo(namaubah);
                }else{

showToast(Objects.requireNonNull(response.body()).getMessage(),
R.layout.custom_toast_sukses);
                    formubahdata();
                }
            }


@SuppressLint("SetTextI18n")
            @Override
            public                  void
onFailure(@NotNull   Call<Data>   call,
@NotNull       Throwable       t)       {
                progress.hide();
                showToast("Tidak    ada
koneksi internet, silahkan coba lagi!",
R.layout.custom_toast_gagal);
            }
        });
    }

    private     void     ubahpass(String
passlama,   String   passbaru,   String
ulangpassbaru,     String     user)     {
        progress.show();
        apiInterface                    =
ApiClient.getApiClient().create(ApiInterface.class);
        Call<Data>      masukCall      =
apiInterface.ubahpassRespone(passlama,
passbaru,      ulangpassbaru,      user);
        masukCall.enqueue(new
Callback<Data>()                        {
            @Override
            public                  void
onResponse(@NotNull   Call<Data>   call,
```

```
@NotNull Response<Data> response) {
                progress.hide();
                if(response.body() != null && response.isSuccessful() && response.body().isStatus()) {

showToast(response.body().getMessage(), R.layout.custom_toast_sukses);
                }else{

showToast(Objects.requireNonNull(response.body()).getMessage(), R.layout.custom_toast_gagal);
                    formubahpassword();
                }
            }


@SuppressLint("SetTextI18n")
            @Override
            public void onFailure(@NotNull Call<Data> call, @NotNull Throwable t) {
                progress.hide();
                showToast("Tidak ada koneksi internet, silahkan coba lagi!", R.layout.custom_toast_gagal);
            }
        });
    }

    private void badgelaporab(String user){
        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
        Call<List<ResultData>> call = apiInterface.getCountLaporan(user);
        call.enqueue(new Callback<List<ResultData>>() {

            @Override
            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {

if(response.isSuccessful() && response.body() != null){
                    int jml = response.body().get(0).getIsiLaporan();

itemcoutlaporan.setText(jml + " laporan");
                }else{
                    showToast("Gagal meload data", R.layout.custom_toast_gagal);
                }
            }

            @Override
            public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {
                showToast(t.toString(), R.layout.custom_toast_gagal);
            }
        });
    }

    private void tampillaporan(String user) {

errorlayout.setVisibility(View.GONE);
        showLoading();
        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
        Call<List<ResultData>> call = apiInterface.getLaporan(user);
        call.enqueue(new Callback<List<ResultData>>() {

            @Override
            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {
                hideLoading();

if(response.isSuccessful() && response.body() != null){

onGetResult(response.body());
                }else{
                    String errorCode;
                    switch (response.code()){
                        case 404:
                            errorCode = "404 not found";
                            break;
                        case 500:
                            errorCode = "500 server broken";
                            break;
                        default:
                            errorCode = "Unknown error";
                            break;
                    }

showErrorMessage(R.drawable.no_result, "No Result", "Please Try Again!-"+errorCode);
                }
            }

            @Override
            public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {
                hideLoading();

onErrorLoading(t.toString());
            }
        });
```

```
    }

    private void tampillaporanBerdasarkanTanggal(String user, String tgl) {
        showLoading();
        apiInterface = ApiClient.getApiClient().create(ApiInterface.class);
        Call<List<ResultData>> call = apiInterface.getCariLaporan(user, tgl);
        call.enqueue(new Callback<List<ResultData>>() {
            @Override
            public void onResponse(@NonNull Call<List<ResultData>> call, @NonNull Response<List<ResultData>> response) {
                hideLoading();
                if (response.isSuccessful() && response.body() != null) {

                    onGetResult(response.body());
                } else {
                    String errorCode;
                    switch (response.code()){
                        case 404:
                            errorCode = "404 not found";
                            break;
                        case 500:
                            errorCode = "500 server broken";
                            break;
                        default:
                            errorCode = "Unknown error";
                            break;
                    }

                    showErrorMessage(R.drawable.no_result, "No Result", "Please Try Again!-"+errorCode);
                }
            }

            @Override
            public void onFailure(@NonNull Call<List<ResultData>> call, @NonNull Throwable t) {
                hideLoading();

                onErrorLoading(t.toString());
            }
        });
    }

    private void showLoading() {

        mSwipeRefreshLayout.setRefreshing(true);
    }

    private void hideLoading() {

        mSwipeRefreshLayout.setRefreshing(false);
    }


    @SuppressLint("NotifyDataSetChanged")
    private void onGetResult(List<ResultData> results) {
        adapter = new ProfilAdapter(getContext(), results, itemClickListener);
        adapter.notifyDataSetChanged();

        recyclerView.setAdapter(adapter);
        resultDataList = results;
    }

    private void onErrorLoading(String message) {

        showErrorMessage(R.drawable.no_result, "Oops...", "Network failure, Please Try Again!.\n"+ message);
    }

    private void showErrorMessage(int imageView, String title, String message){

        if(errorlayout.getVisibility() == View.GONE){

            errorlayout.setVisibility(View.VISIBLE);
        }


        imgerror.setImageResource(imageView);
        errortitle.setText(title);
        errormessage.setText(message);


        btnretry.setOnClickListener(view -> tampillaporan(user));

    }

    private void tampilGambarDialog() {
        String[] options = {"Kamera", "Galeri"};
        AlertDialog.Builder builder = new AlertDialog.Builder(requireContext());
        builder.setTitle("Pilih berkas dari:");
        builder.setItems(options, (dialog, which) -> {
            if (which == 0) {
                if (!checkCameraPermission()) {

                    requestCameraPermission();
```

```
                } else {
                    openCamera();
                }
            } else if (which == 1) {
                if
(!checkStoragePermission())           {

requestStoragePermission();
                } else {
                    openGallery();
                }
            }
        });
        builder.create().show();
    }

    private                          Boolean
checkStoragePermission()              {
        if   (Build.VERSION.SDK_INT   >=
Build.VERSION_CODES.TIRAMISU)         {
            return
ContextCompat.checkSelfPermission(requi
reContext(),
Manifest.permission.READ_MEDIA_IMAGES)
== PackageManager.PERMISSION_GRANTED &&

ContextCompat.checkSelfPermission(requi
reContext(),
Manifest.permission.READ_MEDIA_VIDEO)
== PackageManager.PERMISSION_GRANTED &&

ContextCompat.checkSelfPermission(requi
reContext(),
Manifest.permission.READ_MEDIA_AUDIO)
==    PackageManager.PERMISSION_GRANTED;
        }             else            {
            return
ContextCompat.checkSelfPermission(requi
reContext(),
Manifest.permission.WRITE_EXTERNAL_STOR
AGE)                                  ==
PackageManager.PERMISSION_GRANTED     &&

ContextCompat.checkSelfPermission(requi
reContext(),
Manifest.permission.READ_EXTERNAL_STORA
GE)                                   ==
PackageManager.PERMISSION_GRANTED;
        }
    }

    private                           void
requestStoragePermission()            {
        if   (Build.VERSION.SDK_INT   >=
Build.VERSION_CODES.TIRAMISU)         {
            requestPermissions(new
String[]{

Manifest.permission.READ_MEDIA_IMAGES,

Manifest.permission.READ_MEDIA_VIDEO,

Manifest.permission.READ_MEDIA_AUDIO
            },         STORAGE_REQUEST);
        }             else            {
            requestPermissions(new
String[]{

Manifest.permission.WRITE_EXTERNAL_STOR
AGE,

Manifest.permission.READ_EXTERNAL_STORA
GE
            },         STORAGE_REQUEST);
        }
    }

    private                          Boolean
checkCameraPermission()               {
        return
ContextCompat.checkSelfPermission(requi
reContext(),
Manifest.permission.CAMERA)           ==
PackageManager.PERMISSION_GRANTED;
    }

    private                           void
requestCameraPermission()             {
        requestPermissions(new
String[]{Manifest.permission.CAMERA},
CAMERA_REQUEST);
    }

    private    void    openCamera()    {
        ContentValues   values   =   new
ContentValues();

values.put(MediaStore.Images.Media.TITL
E,            "New            Picture");

values.put(MediaStore.Images.Media.DESC
RIPTION,     "From     the     Camera");
        imageUri                      =
requireContext().getContentResolver().i
nsert(MediaStore.Images.Media.EXTERNAL_
CONTENT_URI,                    values);
        Intent    takePicture    =    new
Intent(MediaStore.ACTION_IMAGE_CAPTURE)
;

takePicture.putExtra(MediaStore.EXTRA_O
UTPUT,                          imageUri);

startActivityForResult(takePicture,
IMAGE_CAPTURE_CODE);
    }

    private    void    openGallery()    {
        Intent    pickPhoto    =    new
Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTEN
T_URI);

startActivityForResult(pickPhoto,
PICK_IMAGE_CODE);
    }

    @Override
    public                            void
onRequestPermissionsResult(int
```

```
requestCode,      @NonNull      String[]
permissions,       @NonNull       int[]
grantResults)                              {

super.onRequestPermissionsResult(reques
tCode,    permissions,    grantResults);
        if        (requestCode        ==
CAMERA_REQUEST)                            {
            if (grantResults.length > 0
&&          grantResults[0]          ==
PackageManager.PERMISSION_GRANTED)     {
                openCamera();
            }          else          {
                showToast("Harap
Aktifkan          Izin          Kamera",
R.layout.custom_toast_info);
            }
        }  else  if  (requestCode  ==
STORAGE_REQUEST)                           {
            if (grantResults.length > 0
&&          grantResults[0]          ==
PackageManager.PERMISSION_GRANTED)     {
                openGallery();
            }          else          {
                showToast("Harap
Aktifkan       Izin       Penyimpanan",
R.layout.custom_toast_info);
            }
        }
    }

    @Override
    public  void  onActivityResult(int
requestCode,  int  resultCode,  Intent
data)                                      {

super.onActivityResult(requestCode,
resultCode,                        data);
        if        (resultCode        ==
Activity.RESULT_OK)                        {
            if      (requestCode      ==
IMAGE_CAPTURE_CODE)                        {

unggahfoto.setImageURI(imageUri);
                part_image             =
getPathFromUri(imageUri);
            } else if (requestCode  ==
PICK_IMAGE_CODE  &&  data  !=  null)  {
                Uri     dataImage     =
data.getData();

unggahfoto.setImageURI(dataImage);
                part_image             =
getPathFromUri(dataImage);
            }
        }
    }

    private  String  getPathFromUri(Uri
uri)                                       {
        String[]      projection      =
{MediaStore.Images.Media.DATA};
        Cursor         cursor         =
requireContext().getContentResolver().q
uery(uri,   projection,   null,   null,
null);
        if    (cursor    !=    null)    {
            cursor.moveToFirst();
            int         index         =
cursor.getColumnIndexOrThrow(projection
[0]);
            String        path        =
cursor.getString(index);
            cursor.close();
            return                 path;
        }
        return                     null;
    }

    private void setErrorState(EditText
editText,  TextView  errorTextView,  int
drawableId,   String   errorMessage)   {

editText.setBackgroundResource(R.drawab
le.custom_edittext_required);
        Drawable         draw         =
ContextCompat.getDrawable(requireContex
t(),                        drawableId);

Objects.requireNonNull(draw).setColorFi
lter(new
PorterDuffColorFilter(Color.parseColor(
"#FFCC0000"), PorterDuff.Mode.SRC_IN));

editText.setCompoundDrawablesWithIntrin
sicBounds(draw,   null,   null,   null);

errorTextView.setVisibility(View.VISIBL
E);

errorTextView.setText(errorMessage);
    }

    private                        void
clearErrorState(EditText      editText,
TextView errorTextView, int drawableId)
{

editText.setBackgroundResource(R.drawab
le.custom_edittext);
        Drawable         draw         =
ContextCompat.getDrawable(requireContex
t(),                        drawableId);

Objects.requireNonNull(draw).setColorFi
lter(new
PorterDuffColorFilter(Color.parseColor(
"#2E190A"),    PorterDuff.Mode.SRC_IN));

editText.setCompoundDrawablesWithIntrin
sicBounds(draw,   null,   null,   null);

errorTextView.setVisibility(View.GONE);
    }

    private    void    showToast(String
message,      int      layoutId)      {
        final   Toast   toast   =   new
Toast(requireContext());
```

```
toast.setDuration(Toast.LENGTH_LONG);
        @SuppressLint("InflateParams")
View            custom_view          =
getLayoutInflater().inflate(layoutId,
null);
        TextView       tvMessage       =
custom_view.findViewById(R.id.toast_mes
sage);
        tvMessage.setText(message);
        toast.setView(custom_view);
        toast.show();
    }
}
```

BackToLogin.java

```
package
com.kelas_c.a220280075.bhabinkamtibnas;

import            android.app.Activity;
import          android.content.Context;
import           android.content.Intent;

import
com.kelas_c.a220280075.bhabinkamtibnas.
activity.LoginActivity;

public      class      BackToLogin      {
    public          static          void
kembaliKeHalamanMasuk(Context   context)
{
        Intent      intent      =      new
Intent(context,    LoginActivity.class);

intent.addFlags(Intent.FLAG_ACTIVITY_CL
EAR_TOP                                 |
Intent.FLAG_ACTIVITY_NEW_TASK);
        context.startActivity(intent);
        if      (context      instanceof
Activity)                               {
            ((Activity)
context).finish();
        }
    }
}
```

SessionManager.java

```
package
com.kelas_c.a220280075.bhabinkamtibnas;

import          android.content.Context;
import
android.content.SharedPreferences;
import
android.preference.PreferenceManager;

import               java.util.HashMap;

public     class     SessionManager     {
    private    final    SharedPreferences
sharedPreferences;
    private                          final
SharedPreferences.Editor         editor;

    public     static     final     String
IS_LOGGED_IN       =       "isLoggedIn";
    public     static     final     String
UserAnggota        =        "useranggota";
    public     static     final     String
NamaAnggota        =        "namaanggota";

    public    SessionManager    (Context
context){
        sharedPreferences               =
PreferenceManager.getDefaultSharedPrefe
rences(context);
        editor                          =
sharedPreferences.edit();
    }

    public                           void
createLoginSession(String   useranggota,
String                    namaanggota){
        editor.putBoolean(IS_LOGGED_IN,
true);
        editor.putString(UserAnggota,
useranggota);
        editor.putString(NamaAnggota,
namaanggota);
        editor.commit();
    }

    public    HashMap<String,    String>
getUserDetail(){
        HashMap<String, String> user =
new                          HashMap<>();
        user.put(UserAnggota,
sharedPreferences.getString(UserAnggota
,                                  null));
        user.put(NamaAnggota,
sharedPreferences.getString(NamaAnggota
,                                  null));
        return                        user;
    }

    public     void     logoutSession(){
        editor.clear();
        editor.commit();
    }

    public     boolean     isLoggedIn(){
        return
sharedPreferences.getBoolean(IS_LOGGED_
IN,                                false);
    }
}
```