LAMPIRAN

Lampiran 1 Source Code

Adapter.java

```
package com.example.reskrim;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.CheckBox;

import
androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class adapter extends
RecyclerView.Adapter<adapter.ViewHolder
> {
   private Context context;
   private LayoutInflater inflater;
   private List<HashMap<String, Object>>
data;

   public adapter(Context context,
List<HashMap<String, Object>> data) {
      this.context = context;
      this.inflater =
LayoutInflater.from(context);
      this.data = data;
   }

   public class ViewHolder extends
RecyclerView.ViewHolder {
      public View layout;
      CheckBox cbactivitieslistreg;

      public ViewHolder(View v) {
         super(v);
         layout = v;
         cbactivitieslistreg =
v.findViewById(R.id.checkBox);
      }
   }

   @Override
   public ViewHolder
onCreateViewHolder(ViewGroup parent, int
viewType) {
      View v =
inflater.inflate(R.layout.row_checkboxes,
parent, false);
      return new ViewHolder(v);
   }

   @Override
   public void onBindViewHolder(final
ViewHolder holder, int position) {
      HashMap<String, Object> data_tampil
= data.get(holder.getAdapterPosition());
      String nama = (String)
data_tampil.get(koneksi.nama);
      boolean isChecked = (boolean)
data_tampil.get("isChecked");

      holder.cbactivitieslistreg.setText(nama);

holder.cbactivitieslistreg.setChecked(isChec
ked);


holder.cbactivitieslistreg.setOnCheckedChan
geListener(null); // Clear any previous
listener

holder.cbactivitieslistreg.setChecked(isChec
ked);

holder.cbactivitieslistreg.setOnCheckedChan
geListener((buttonView, isChecked1) -> {
         data_tampil.put("isChecked",
isChecked1);
      });
   }

   @Override
   public int getItemCount() {
      return data.size();
   }

   public List<String> getSelectedItems() {
      List<String> selectedItems = new
ArrayList<>();
      for (HashMap<String, Object> item :
data) {
         if ((boolean) item.get("isChecked")) {
            selectedItems.add((String)
item.get(koneksi.nama));
```

```
            }
        }
        return selectedItems;
    }
}
```

Adapter_DataPengaduan.java

```
package com.example.reskrim;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import
android.graphics.drawable.ColorDrawable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import
androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;


import java.util.ArrayList;
import java.util.HashMap;

import
cn.pedant.SweetAlert.SweetAlertDialog;

public class adapter_datapengaduan extends
RecyclerView.Adapter<RecyclerView.View
Holder> {
    private Context context;
    private LayoutInflater inflater;
    private          ArrayList<HashMap<String,
String>> data;


    public      adapter_datapengaduan(Context
context,           ArrayList<HashMap<String,
String>> data) {
        this.context = context;
        inflater  =  LayoutInflater.from(context);
        this.data = data;
    }

    @Override
    public              RecyclerView.ViewHolder
onCreateViewHolder(ViewGroup parent, int
viewType) {
        View                v                =
LayoutInflater.from(parent.getContext()).infl
ate(R.layout.content_list_order, null);
        MyHolder holder= new MyHolder(v);

        return holder;
    }

    @Override
    public                                  void
onBindViewHolder(RecyclerView.ViewHol
der holder, int position) {
        MyHolder          myHolder=(MyHolder)
holder;
        HashMap<String, String> data_tampil=
new HashMap<String, String>();
        data_tampil=data.get(position);
        String
tanggal=data_tampil.get(koneksi.tanggal);
        String id=data_tampil.get(koneksi.id);
        String kat=data_tampil.get(koneksi.ket);
        String
urai=data_tampil.get(koneksi.uraian);
        String
st=data_tampil.get(koneksi.status);
        String
nama_pelanggar=data_tampil.get(koneksi.na
ma);
        String
nama_petugas=data_tampil.get(koneksi.user
name);
        String
gmbar=data_tampil.get(koneksi.gambar);
        String itp=data_tampil.get(koneksi.itp);
        String idp=data_tampil.get(koneksi.idp);
        String lati=data_tampil.get(koneksi.lati);
        String
longi=data_tampil.get(koneksi.longi);
        String
alamat=data_tampil.get(koneksi.alamat);
        String
nohp=data_tampil.get(koneksi.no_hp);
        String
stnk=data_tampil.get(koneksi.stnk);
        String
merk=data_tampil.get(koneksi.merk);
        String
plat=data_tampil.get(koneksi.plat);
        String
warna=data_tampil.get(koneksi.warna);
        String
jadwal=data_tampil.get(koneksi.jadwal);
        String
```

```
lokasi=data_tampil.get(koneksi.lokasi);
      String
tujuan=data_tampil.get(koneksi.tujuan);


myHolder.tanggal.setText(data_tampil.get(k
oneksi.tanggal));

myHolder.status.setText(data_tampil.get(kon
eksi.status));

myHolder.id_tilang.setText(data_tampil.get(
koneksi.id));

myHolder.kategori.setText(data_tampil.get(k
oneksi.ket));

       if (st.equals("Proses")){

myHolder.status.setTextColor(Color.parseCo
lor("#655802"));

myHolder.ln.setBackgroundResource(R.dra
wable.bg_proses1);
       }else
if(st.equals("Ditanggapi")||st.equals("Selesai
")){

myHolder.status.setTextColor(Color.parseCo
lor("#146502"));

myHolder.ln.setBackgroundResource(R.dra
wable.bg_proses3);
       }else if (st.equals("Ditolak")){

myHolder.status.setTextColor(Color.parseCo
lor("#650202"));

myHolder.ln.setBackgroundResource(R.dra
wable.bg_proses2);
       }

       myHolder.vw.setOnClickListener(new
View.OnClickListener() {
           @Override
           public void onClick(View view) {
            CharSequence[] item = {"Lihat"};
            AlertDialog.Builder request = new
AlertDialog.Builder(view.getContext())
                   .setTitle("Pilihan")
                   .setItems(item,                 new
DialogInterface.OnClickListener() {
                       @Override
                       public                      void
onClick(DialogInterface dialogInterface, int
i) {
                           switch (i){
                             case 0:


                                 Intent intent = new
Intent(context, Detail_Tilang.class);

intent.putExtra(koneksi.id, id);

intent.putExtra(koneksi.tanggal, tanggal);

intent.putExtra(koneksi.username,
nama_petugas);

intent.putExtra(koneksi.nama,
nama_pelanggar);

intent.putExtra(koneksi.alamat, alamat);

intent.putExtra(koneksi.no_hp, nohp);

intent.putExtra(koneksi.stnk, stnk);

intent.putExtra(koneksi.merk, merk);

intent.putExtra(koneksi.plat, plat);

intent.putExtra(koneksi.warna, warna);

intent.putExtra(koneksi.jadwal, jadwal);

intent.putExtra(koneksi.lokasi, lokasi);

intent.putExtra(koneksi.tujuan, tujuan);

intent.putExtra(koneksi.ket, kat);

intent.putExtra(koneksi.uraian, urai);

intent.putExtra(koneksi.status, st);

intent.putExtra(koneksi.gambar, gmbar);


intent.addFlags(Intent.FLAG_ACTIVITY_
NEW_TASK);

context.startActivity(intent);
                                 break;

                       }
                   }
               });
           request.create();
```

```
            request.show();
            }
        });



    }



    @Override
    public int getItemCount() {
        return data.size();
    }

   class MyHolder extends RecyclerView.ViewHolder {
        TextView tanggal, status, id_tilang, kategori,uraian, tanggap;

        LinearLayout ln, vw;
        public MyHolder(View itemView){
            super(itemView);
            tanggal = (TextView) itemView.findViewById(R.id.pg_tgl);
            status = (TextView) itemView.findViewById(R.id.pg_proses);
            id_tilang = (TextView) itemView.findViewById(R.id.pd_id);
            kategori = (TextView) itemView.findViewById(R.id.pg_kategori);
            uraian = (TextView) itemView.findViewById(R.id.pg_isi);
            tanggap = (TextView) itemView.findViewById(R.id.pg_tanggapan)
;

            ln=(LinearLayout) itemView.findViewById(R.id.Lny);
            vw=(LinearLayout) itemView.findViewById(R.id.view);
        }

        }

}
```

Adapter_Detail.java

```
package com.example.reskrim;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import
androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.HashMap;

public class adapter_detail extends RecyclerView.Adapter<RecyclerView.View
Holder> {
    private Context context;
    private LayoutInflater inflater;
    private ArrayList<HashMap<String, String>> data;
    ArrayList<HashMap<String, String>> tampil=new ArrayList<HashMap<String, String>>();


    public adapter_detail(Context context, ArrayList<HashMap<String, String>> data)
{
        this.context = context;
        inflater = LayoutInflater.from(context);
        this.data = data;
    }

    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v =
LayoutInflater.from(parent.getContext()).infl
ate(R.layout.content_detail, null);
        MyHolder holder= new MyHolder(v);

        return holder;
    }

    @Override
    public void onBindViewHolder(RecyclerView.ViewHol
der holder, int position) {
        MyHolder myHolder=(MyHolder) holder;
        HashMap<String, String> data_tampil= new HashMap<String, String>();
        data_tampil=data.get(position);


myHolder.nama.setText(data_tampil.get(kon
eksi.nama));
```

```
myHolder.keterangan.setText(data_tampil.ge
t(koneksi.ket));


    }

    @Override
    public int getItemCount() {
        return data.size();
    }

   class MyHolder extends
RecyclerView.ViewHolder {
        TextView nama, keterangan;

        public MyHolder(View itemView){
            super(itemView);
             nama = (TextView)
itemView.findViewById(R.id.dt_nm);
             keterangan = (TextView)
itemView.findViewById(R.id.dt_ket);

        }

        }

}
```

Detail_Tilang.java

```
package com.example.reskrim;

import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.recyclerview.widget.LinearLayout
Manager;
import
androidx.recyclerview.widget.RecyclerView;

import android.app.Dialog;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import
android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TableRow;
import android.widget.TextView;
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.bumptech.glide.Glide;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import
cn.pedant.SweetAlert.SweetAlertDialog;

public class Detail_Tilang extends
AppCompatActivity {
    String idd;
    ImageView gmb1;
    TextView id1, tgl1, nama_petugas1,
nama_pelanggar1, alamat1, nohp1, stnk1,
merk1, plat1,
        warna1, jadwal1, lokasi_sidang1,
tujuan1, keterangan1, kategori1, status1;
    RecyclerView rv;
    Button btnnotif;

    TableRow jadwalSidangLayout,
lokasiSidangLayout;
    ArrayList<HashMap<String, String>>
tampil_detail = new ArrayList<>();
    adapter_detail adapter;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_detail_tila
ng);

        gmb1 =
findViewById(R.id.pg_gambar);
        id1 = findViewById(R.id.pg_id);
        tgl1 = findViewById(R.id.pg_tgl);
        nama_petugas1 =
```

```
findViewById(R.id.pg_namapetugas);
    nama_pelanggar1 =
findViewById(R.id.pg_namapelanggar);
    alamat1 =
findViewById(R.id.pg_alamat);
    nohp1 = findViewById(R.id.pg_nohp);
    stnk1 = findViewById(R.id.pg_stnk);
    merk1 = findViewById(R.id.pg_merk);
    plat1 = findViewById(R.id.pg_plat);
    warna1 =
findViewById(R.id.pg_warna);
    jadwal1 =
findViewById(R.id.pg_jadwal);
    lokasi_sidang1 =
findViewById(R.id.pg_lokasi);
    tujuan1 =
findViewById(R.id.pg_tujuan);
    kategori1 =
findViewById(R.id.pg_kategori);
    keterangan1 =
findViewById(R.id.pg_isi);
    status1 = findViewById(R.id.pg_status);
    btnnotif =
findViewById(R.id.btn_notif);
    jadwalSidangLayout =
findViewById(R.id.jadwalsidangtr);
    lokasiSidangLayout =
findViewById(R.id.lokasisidangtr);
    rv = findViewById(R.id.rv_detail);
    rv.setHasFixedSize(true);
    LinearLayoutManager lim = new
LinearLayoutManager(getApplicationConte
xt());

lim.setOrientation(LinearLayoutManager.V
ERTICAL);
    rv.setLayoutManager(lim);

    Intent intent = getIntent();
    idd = intent.getStringExtra(koneksi.id);

    // Debug log to check URL
    String imageUrl =
intent.getStringExtra(koneksi.gambar);
    Log.d("Detail_Tilang", "URL Gambar: "
+ imageUrl);

    // Load image with Glide and add
placeholders and error handling
    Glide.with(Detail_Tilang.this)
        .load(imageUrl)

        .into(gmb1);

    id1.setText(idd);

    tgl1.setText(intent.getStringExtra(koneksi.ta
nggal));

    nama_petugas1.setText(intent.getStringExtra
(koneksi.username));

    nama_pelanggar1.setText(intent.getStringEx
tra(koneksi.nama));

    alamat1.setText(intent.getStringExtra(konek
si.alamat));

    nohp1.setText(intent.getStringExtra(koneksi.
no_hp));

    stnk1.setText(intent.getStringExtra(koneksi.s
tnk));

    merk1.setText(intent.getStringExtra(koneksi.
merk));

    plat1.setText(intent.getStringExtra(koneksi.p
lat));

    warna1.setText(intent.getStringExtra(koneks
i.warna));

    jadwal1.setText(intent.getStringExtra(konek
si.jadwal));

    lokasi_sidang1.setText(intent.getStringExtra
(koneksi.lokasi));

    tujuan1.setText(intent.getStringExtra(koneks
i.tujuan));

    kategori1.setText(intent.getStringExtra(kone
ksi.ket));

    keterangan1.setText(intent.getStringExtra(ko
neksi.uraian));

    status1.setText(intent.getStringExtra(koneksi
.status));

    String kategori =
kategori1.getText().toString();
    if (kategori.equalsIgnoreCase("Slip
Merah")) {

jadwalSidangLayout.setVisibility(View.VISI
BLE);

lokasiSidangLayout.setVisibility(View.VISI
```

```
BLE);

        // Mendapatkan nilai jadwal1 dan lokasi_sidang1
        String jadwal = jadwal1.getText().toString(); // Ganti dengan ID yang sesuai
        String lokasi_sidang = lokasi_sidang1.getText().toString(); // Ganti dengan ID yang sesuai

        if (jadwal.equals("30 November -0001")) {
          jadwal1.setText("Tunggu jadwal sidang dari admin");
        }

        if (lokasi_sidang.isEmpty()) {
          lokasi_sidang1.setText("Tunggu lokasi sidang dari admin");
        }
      } else if (kategori.equalsIgnoreCase("Slip Biru")) {

lokasiSidangLayout.setVisibility(View.GONE);

jadwalSidangLayout.setVisibility(View.GONE);
      }

      String keterangan = keterangan1.getText().toString();
      if(keterangan.isEmpty()){
        keterangan1.setText("-");
      }

      btnnotif.setOnClickListener(v -> kirimNotifikasi());

      tampil_pasal();
    }

    private void tampil_pasal() {
      Dialog dialog = new Dialog(this);

dialog.setContentView(R.layout.loading);
      if (dialog.getWindow() != null) {

dialog.getWindow().setBackgroundDrawable(new ColorDrawable(0));
      }
      dialog.show();
      dialog.setCancelable(false);

      StringRequest penjualan = new StringRequest(Request.Method.POST, koneksi.tampil,
          new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
              Log.e("TAG", "onResponse: " + response);
              try {
                JSONObject jsonObject = new JSONObject(response);
                if (jsonObject.getInt("hasil") == 1) {
                  JSONArray data = jsonObject.getJSONArray("list");
                  for (int i = 0; i < data.length(); i++) {
                    JSONObject c = data.getJSONObject(i);

                    // Memastikan kunci 'nama' ada
                    String a = c.has(koneksi.nama) ? c.getString(koneksi.nama) : null;

                    // Memastikan kunci 'ket' ada
                    String b = c.has(koneksi.ket) ? c.getString(koneksi.ket) : null;

                    HashMap<String, String> map = new HashMap<>();
                    map.put(koneksi.nama, a);
                    map.put(koneksi.ket, b);
                    tampil_detail.add(map);
                  }
                  adapter = new adapter_detail(getApplicationContext(), tampil_detail);
                  rv.setAdapter(adapter);
                } else if (jsonObject.getInt("hasil") == 0) {
                  new SweetAlertDialog(Detail_Tilang.this, SweetAlertDialog.ERROR_TYPE)
                      .setTitleText("Maaf!!!")
                      .setContentText("Data Tidak Ditemukan")

.setConfirmClickListener(sDialog -> {
                        Intent hc = new Intent(Detail_Tilang.this,
```

```
List_Pengaduan.class);
                        finish();
                        startActivity(hc);
                      })
                      .show();
                  }
                } catch (JSONException e) {
                  e.printStackTrace();
                }
                dialog.dismiss();
              }
            }, new Response.ErrorListener() {
          @Override
          public void onErrorResponse(VolleyError error) {
            Log.e("Detail", "onErrorResponse: " + error.getMessage());
            new SweetAlertDialog(Detail_Tilang.this, SweetAlertDialog.ERROR_TYPE)
                .setTitleText("Oops...")
                .setContentText("Jaringan tidak ada" + error.getMessage())
                .show();
            dialog.dismiss();
          }
        }) {
          @Override
          protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> param = new HashMap<>();
            param.put(koneksi.username, idd);
            param.put(koneksi.id, "data_pasal");
            return param;
          }
        };

        RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
        requestQueue.add(penjualan);
    }

    private void kirimNotifikasi() {
        StringRequest request = new StringRequest(Request.Method.POST, "http://etilang.us.to/layanan/kirim_notif.php",
            new Response.Listener<String>() {
              @Override
              public void onResponse(String response) {
                // Handle response from PHP script (if any)
                Log.d("NotificationResponse", response);
                // Add any UI feedback if needed
              }
            }, new Response.ErrorListener() {
          @Override
          public void onErrorResponse(VolleyError error) {
            // Handle errors
            Log.e("NotificationError", "Error sending notification: " + error.getMessage());
          }
        }) {
          @Override
          protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> params = new HashMap<>();
            params.put("id_tilang", id1.getText().toString());
            params.put("tgl", tgl1.getText().toString());
            params.put("nama_petugas", nama_petugas1.getText().toString());
            params.put("nama_pelanggar", nama_pelanggar1.getText().toString());
            params.put("alamat", alamat1.getText().toString());
            params.put("nohp", nohp1.getText().toString());
            params.put("stnk", stnk1.getText().toString());
            params.put("merk", merk1.getText().toString());
            params.put("plat", plat1.getText().toString());
            params.put("warna", warna1.getText().toString());
            params.put("jadwal", jadwal1.getText().toString());
            params.put("lokasi_sidang", lokasi_sidang1.getText().toString());
            params.put("tujuan", tujuan1.getText().toString());
            params.put("keterangan", keterangan1.getText().toString());
            params.put("kategori", kategori1.getText().toString());
            params.put("status", status1.getText().toString());
            return params;
          }
        };

        RequestQueue queue =
```

```
Volley.newRequestQueue(getApplicationCo
ntext());
        queue.add(request);
    }
}
```

Edit_Profil.java

```
package com.example.reskrim;

import
androidx.appcompat.app.AppCompatActivit
y;

import android.app.DatePickerDialog;
import android.app.Dialog;
import android.content.Intent;
import android.content.SharedPreferences;
import
android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.bumptech.glide.Glide;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import  java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

import
cn.pedant.SweetAlert.SweetAlertDialog;

public class Edit_Profil extends
AppCompatActivity {
    String user_;
    EditText nama, username, alamat, tempat,
tanggal, nohp;
    Spinner jenis_k;
    ImageView dt, foto;
    Button update;
    String jenis;
    String us,nm,jk,tl,np,alm,ft, tp;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_edit_profi
l);
        bacaPreferensi();
        profil();
        nama=findViewById(R.id.m_nama);
        username=findViewById(R.id.m_user);
        alamat=findViewById(R.id.m_alamat);
        tempat=findViewById(R.id.m_tempat);
        tanggal=findViewById(R.id.m_tgl);
        nohp=findViewById(R.id.m_no);
        jenis_k=findViewById(R.id.spin_jk);
        dt=findViewById(R.id.r_date);
        update=findViewById(R.id.m_btn);
        foto=findViewById(R.id.ep_profil);

        Calendar calendar =
Calendar.getInstance();
        final int year =
calendar.get(Calendar.YEAR);
        final int month =
calendar.get(Calendar.MONTH);
        final int day =
calendar.get(Calendar.DAY_OF_MONTH);

        dt.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                DatePickerDialog
datePickerDialog = new DatePickerDialog(
                        Edit_Profil.this, new
DatePickerDialog.OnDateSetListener() {
                    @Override
                    public void
onDateSet(DatePicker view, int year, int
month, int dayOfMonth) {
                        month = month + 1;
                        String date = dayOfMonth +
"-" + month + "-" + year;
                        tanggal.setText(date);
                    }
                }, year, month, day
                );
                datePickerDialog.show();
            }
        });
```

```
        update.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
                update_profil();
            }
        });
    }

    private void update_profil() {
        String c=nama.getText().toString();
        String
jenis_kelamin=jenis_k.getSelectedItem().toS
tring();
        String d=tempat.getText().toString();
        String e=tanggal.getText().toString();
        String f=alamat.getText().toString();
        String h=nohp.getText().toString();
        if (jenis_kelamin.equals("Laki-Laki")){
            jenis="L";
        }else
if(jenis_kelamin.equals("Perempuan")){
            jenis="P";
        }
        if (c.equals("")) {
            nama.setError("Belum diisi");
            nama.requestFocus();
        }else if (d.equals("")) {
            tempat.setError("Belum diisi");
            tempat.requestFocus();
        }else if (e.equals("")) {
            tanggal.setError("Belum diisi");
            tanggal.requestFocus();
        }else if (f.equals("")) {
            alamat.setError("Belum diisi");
            alamat.requestFocus();
        }else if (h.equals("")) {
            nohp.setError("Belum diisi");
            nohp.requestFocus();
        }else {
            Dialog dialog = new
Dialog(Edit_Profil.this);

dialog.setContentView(R.layout.loading);
            if (dialog.getWindow() != null) {

dialog.getWindow().setBackgroundDrawabl
e(new ColorDrawable(0));
            }
            dialog.show();
            dialog.setCancelable(false);
            StringRequest update = new
StringRequest(Request.Method.POST,
koneksi.edit,
                    new
Response.Listener<String>() {
                @Override
                public void
onResponse(String response) {
                    dialog.dismiss();
                    try {
                        JSONObject
jsonObject = new JSONObject(response);
                        if
(jsonObject.getInt("hasil") == 1) {
                            dialog.dismiss();
                            new
SweetAlertDialog(Edit_Profil.this,
SweetAlertDialog.SUCCESS_TYPE)

.setTitleText("Data Terkirim")

.setContentText("Data Berhasil Tersimpan")

.setConfirmText("Ok")

.setConfirmClickListener(new
SweetAlertDialog.OnSweetClickListener() {
                                @Override
                                public void
onClick(SweetAlertDialog sDialog) {
                                    Intent hc
= new Intent(Edit_Profil.this, Profil.class);

startActivity(hc);
                                    finish();


                                }
                            })
                            .show();
                        } else if
(jsonObject.getInt("hasil") == 0) {
                            dialog.dismiss();
                            new
SweetAlertDialog(Edit_Profil.this,
SweetAlertDialog.ERROR_TYPE)

.setTitleText("Oops...")

.setContentText("Data Gagal Tersimpan")
                                .show();
                        }
                    } catch (JSONException
e) {
                        e.printStackTrace();
                    }
                }
            }, new
Response.ErrorListener() {
```

```
            @Override
            public void
onErrorResponse(VolleyError error) {
                new
SweetAlertDialog(Edit_Profil.this,
SweetAlertDialog.ERROR_TYPE)

.setContentText("Jaringan tidak ada")
                        .show();
                dialog.dismiss();
            }
        }) {
            @Override
            protected Map<String, String>
getParams() throws AuthFailureError {
                Map<String, String> param =
new HashMap<>();
                param.put(koneksi.username,
user_);
                param.put(koneksi.nama, c);
                param.put(koneksi.tempat, d);
                param.put(koneksi.tanggal,
e);
                param.put(koneksi.alamat, f);
                param.put(koneksi.no_hp, h);

param.put(koneksi.jenis_kelamin, jenis);
                param.put(koneksi.id,
"editdata");
                return param;
            }
        };
        RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationCo
ntext());
        requestQueue.add(update);


    }
}


private void bacaPreferensi() {
    SharedPreferences pref =
getSharedPreferences("akun",
MODE_PRIVATE);
    user_ =
pref.getString(koneksi.username, "0");
}
private void profil() {
    StringRequest cari = new
StringRequest(Request.Method.POST,
koneksi.profil,
        new Response.Listener<String>()
{
            @Override
            public void onResponse(String
response) {
                try {
                    JSONObject jsonObject
=new JSONObject(response);
                    if
(jsonObject.getInt("hasil")==1) {
                        JSONArray
hasil=jsonObject.getJSONArray("Profil");
                        for (int i = 0; i <
hasil.length(); i++) {
                            JSONObject c =
hasil.getJSONObject(i);
                            nm =
c.getString(koneksi.nama);
                            jk =
c.getString(koneksi.jenis_kelamin);
                            tl =
c.getString("tanggall");
                            tp =
c.getString(koneksi.tempat);
                            alm =
c.getString(koneksi.alamat);
                            np =
c.getString(koneksi.no_hp);
                            ft =
c.getString(koneksi.gambar);

                        }
                        nama.setText(nm);
                        if (jk.equals("L")){

jenis_k.setSelection(0);
                        }else if(jk.equals("P")){

jenis_k.setSelection(1);
                        }

 Glide.with(Edit_Profil.this).load(ft)
                                .into(foto);
                        alamat.setText(alm);
                        nohp.setText(np);
                        tempat.setText(tp);
                        tanggal.setText(tl);
                    }
                }catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener() {
    @Override
    public void
onErrorResponse(VolleyError error) {
        new
SweetAlertDialog(Edit_Profil.this,
SweetAlertDialog.ERROR_TYPE)
```

```
                .setTitleText("Oops...")
                .setContentText("Jaringan
Tidak Ada")
                .show();
        }
    }) {
        @Override
        protected Map<String, String>
getParams() throws AuthFailureError {
            Map<String, String> param = new
HashMap<>();
            param.put(koneksi.username,
user_);
            return param;
        }
    };
    RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationCo
ntext());
    requestQueue.add(cari);
  }

  @Override
  public void onBackPressed() {
    startActivity(new
Intent(getApplicationContext(),
Profil.class));
    finish();
    super.onBackPressed();
  }
}
```

Home.java

```
package com.example.reskrim;

import
androidx.appcompat.app.AppCompatActivit
y;
import androidx.cardview.widget.CardView;

import android.app.AlertDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import
android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.os.Handler;
import android.text.format.DateFormat;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.animation.Animation;
import
android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.TextView;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.bumptech.glide.Glide;
import
com.google.android.material.snackbar.Snack
bar;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;

import
cn.pedant.SweetAlert.SweetAlertDialog;


public class Home extends
AppCompatActivity implements
View.OnClickListener {
  String user_,nama_, hariIni, tgll, nm, ft;
  ImageView foto,keluar;
  TextView username, dat;
  CardView profil, pengaduan, history,
tentang;
  Animation animTv;
  boolean doubleBackToExitPressedOnce =
false;
  @Override
  protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);

setContentView(R.layout.activity_home);
    bacaPreferensi();
    data_profil();
    profil=findViewById(R.id.cv_profil);

pengaduan=findViewById(R.id.cv_pengadu
an);

history=findViewById(R.id.cv_history);
```

```
keluar=findViewById(R.id.img_keluar);
        dat=findViewById(R.id.tvDate);

username=findViewById(R.id.td_nama);
        foto=findViewById(R.id.img_profil);
        tentang=findViewById(R.id.cv_about);

        Date dateNow = Calendar.getInstance().getTime();
        animTv = AnimationUtils.loadAnimation(this, R.anim.anim_tv);
        hariIni = (String) DateFormat.format("EEEE", dateNow);
        dat.setAnimation(animTv);
        if (hariIni.equalsIgnoreCase("sunday")) {
            hariIni = "Minggu";
        } else if (hariIni.equalsIgnoreCase("monday")) {
            hariIni = "Senin";
        } else if (hariIni.equalsIgnoreCase("tuesday")) {
            hariIni = "Selasa";
        } else if (hariIni.equalsIgnoreCase("wednesday")) {
            hariIni = "Rabu";
        } else if (hariIni.equalsIgnoreCase("thursday")) {
            hariIni = "Kamis";
        } else if (hariIni.equalsIgnoreCase("friday")) {
            hariIni = "Jumat";
        } else if (hariIni.equalsIgnoreCase("saturday")) {
            hariIni = "Sabtu";
        }

        getToday();

        profil.setOnClickListener(this);
        pengaduan.setOnClickListener(this);
        history.setOnClickListener(this);
        keluar.setOnClickListener(this);
        tentang.setOnClickListener(this);
    }

    private void data_profil() {
        StringRequest cari = new StringRequest(Request.Method.POST, koneksi.profil,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        try {
                            JSONObject jsonObject =new JSONObject(response);
                            if (jsonObject.getInt("hasil")==1) {
                                JSONArray hasil=jsonObject.getJSONArray("Profil");
                                for (int i = 0; i < hasil.length(); i++) {
                                    JSONObject c = hasil.getJSONObject(i);
                                    ft = c.getString(koneksi.gambar);
                                    nm = c.getString(koneksi.nama);
                                }
                                username.setText(nm);

                                Glide.with(Home.this).load(ft)
                                        .into(foto);

                            }
                        }catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Log.e("YSPL", "onErrorResponse: " + error.getMessage());
                new SweetAlertDialog(Home.this, SweetAlertDialog.ERROR_TYPE)
                        .setTitleText("Oops...")
                        .setContentText("Jaringan Tidak Ada")
                        .show();
            }
        }) {
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String, String> param = new HashMap<>();
                param.put(koneksi.username, user_);
                return param;
            }
        };
        RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
```

```
        requestQueue.add(cari);
    }

    private void bacaPreferensi() {
        SharedPreferences pref = getSharedPreferences("akun", MODE_PRIVATE);
        user_ = pref.getString(koneksi.username, "0");
        nama_ = pref.getString(koneksi.nama, "0");
    }

    private void getToday() {
        Date date = Calendar.getInstance().getTime();
        String tanggal = (String) DateFormat.format("d", date);
        String monthNumber = (String) DateFormat.format("M", date);
        String year = (String) DateFormat.format("yyyy", date);

        int month = Integer.parseInt(monthNumber);
        String bulan = null;
        if (month == 1) {
            bulan = "Januari";
        } else if (month == 2) {
            bulan = "Februari";
        } else if (month == 3) {
            bulan = "Maret";
        } else if (month == 4) {
            bulan = "April";
        } else if (month == 5) {
            bulan = "Mei";
        } else if (month == 6) {
            bulan = "Juni";
        } else if (month == 7) {
            bulan = "Juli";
        } else if (month == 8) {
            bulan = "Agustus";
        } else if (month == 9) {
            bulan = "September";
        } else if (month == 10) {
            bulan = "Oktober";
        } else if (month == 11) {
            bulan = "November";
        } else if (month == 12) {
            bulan = "Desember";
        }
        tgll= tanggal + " " + bulan + " " + year;
        String formatFix = hariIni + ", " +tgll ;
        dat.setText(formatFix);
    }
```

```
    @Override
    public void onClick(View v) {
        switch (v.getId()){
            case R.id.cv_profil:
                Intent pro=new Intent(Home.this, Profil.class);

overridePendingTransition(R.anim.slide_in_left, R.anim.stay);
                startActivity(pro);
                finish();
                break;
            case R.id.cv_pengaduan:
                Intent pg=new Intent(Home.this, Pengaduan.class);

overridePendingTransition(R.anim.slide_in_left, R.anim.stay);
                startActivity(pg);
                finish();
                break;
            case R.id.cv_history:
                Intent ch=new Intent(Home.this, List_Pengaduan.class);

overridePendingTransition(R.anim.slide_in_left, R.anim.stay);
                startActivity(ch);
                finish();
                break;
            case R.id.cv_about:
                tentang_aplikasi(v);
                break;

            case R.id.img_keluar:
                new SweetAlertDialog(Home.this, SweetAlertDialog.BUTTON_CONFIRM)
                        .setTitleText("KELUAR")
                        .setContentText("Kamu yakin ingin keluar ?")
                        .setConfirmText("Ok")

.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {
                            @Override
                            public void onClick(SweetAlertDialog sDialog) {
                                Intent hc=new Intent(Home.this, Login.class);
                                SharedPreferences pref = getSharedPreferences("akun",MODE_PRIVATE);

SharedPreferences.Editor editor = pref.edit();
```

```
editor.putString(koneksi.username, "0");

editor.putString(koneksi.nama, "0");
                        editor.commit();
                        startActivity(hc);
                        finish();


                    }
                })
                .setCancelButton("Batal",
SweetAlertDialog::dismissWithAnimation)
                .show();
            break;
        }
    }

    private void tentang_aplikasi(View view) {
        AlertDialog.Builder builder = new
 AlertDialog.Builder(view.getRootView().get
Context());
        View dialogview =
LayoutInflater.from(view.getRootView().get
Context()).inflate(R.layout.pop_about, null);
        builder.setView(dialogview);
        final AlertDialog alertDialog =
builder.create();

alertDialog.getWindow().setBackgroundDra
wable(new
ColorDrawable(Color.TRANSPARENT));
        alertDialog.show();

    }

    @Override
    public void onBackPressed() {
        if (doubleBackToExitPressedOnce) {
            super.onBackPressed();
            return;
        }

        this.doubleBackToExitPressedOnce =
true;


Snackbar.make(findViewById(R.id.img_kel
uar), "Please click BACK again to exit",
Snackbar.LENGTH_SHORT).show();

        new Handler().postDelayed(new
Runnable() {

            @Override
            public void run() {

doubleBackToExitPressedOnce=false;
            }
        }, 2000);
    }
}
```

Koneksi.java

```
package com.example.reskrim;


public class koneksi {
    public static final String internet =
"http://etilang.us.to/layanan/";
    public static final String edit =
internet+"edit.php";
    public static final String simpan =
internet+"simpan.php";
    public static final String smp_ch =
internet+"sch.php";
    public static final String login =
internet+"login.php";
    public static final String profil =
internet+"profil.php";
    public static final String tampil =
internet+"tampil.php";
    public static final String rating =
internet+"rate.php";


    public static final String username =
"user";
    public static final String password =
"pass";
    public static final String longi = "long";
    public static final String lati = "lat";
    public static final String alm = "alm";
    public static final String id = "akses";
    public static final String nama = "nama";
    public static final String tempat =
"tempat";
    public static final String tanggal =
"tanggal";
    public static final String alamat = "alamat";
    public static final String no_hp = "no";
    public static final String gambar =
"gambar";
    public static final String jenis_kelamin =
"jk";
    public static final String stnk = "stnk";
    public static final String merk = "merk";
    public static final String plat = "plat";
    public static final String warna = "warna";
    public static final String pasal = "pasal";
    public static final String ket = "ket";
    public static final String uraian = "uraian";
```

```
    public static final String jadwal = "jadwal";
    public static final String lokasi = "lokasi";
    public static final String tujuan = "tujuan";
    public static final String status = "status";

    public static final String itp = "itp";
    public static final String idp = "idp";


}
```

List_Pengaduan.java

```
package com.example.reskrim;

import
androidx.appcompat.app.AppCompatActivit
y;
import
androidx.recyclerview.widget.LinearLayout
Manager;
import
androidx.recyclerview.widget.RecyclerView;

import android.app.Dialog;
import android.content.Intent;
import android.content.SharedPreferences;
import
android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.util.Log;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import
cn.pedant.SweetAlert.SweetAlertDialog;

public class List_Pengaduan extends
AppCompatActivity {
    RecyclerView rv;
    ArrayList<HashMap<String, String>>
tampil_datapengaduan = new
ArrayList<HashMap<String, String>>();
    adapter_datapengaduan adapter;
    String user_;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_list_penga
duan);
        bacaPreferensi();
        tampil_pengaduan();
        rv = findViewById(R.id.rv_pengaduan);
        rv.setHasFixedSize(true);
        LinearLayoutManager lim = new
LinearLayoutManager(getApplicationConte
xt());

lim.setOrientation(LinearLayoutManager.V
ERTICAL);
        rv.setLayoutManager(lim);
    }
    private void bacaPreferensi() {
        SharedPreferences pref =
getSharedPreferences("akun",
MODE_PRIVATE);
        user_ =
pref.getString(koneksi.username, "0");
    }
    private void tampil_pengaduan() {
        Dialog dialog = new Dialog(this);

dialog.setContentView(R.layout.loading2);
        if (dialog.getWindow() != null) {

dialog.getWindow().setBackgroundDrawabl
e(new ColorDrawable(0));
        }
        dialog.show();
        dialog.setCancelable(false);
        final StringRequest list_guru = new
StringRequest(Request.Method.POST,
koneksi.tampil,
                new
Response.Listener<String>() {
                    @Override
                    public void
onResponse(String response) {
                        Log.e("TAG",
"onResponse: "+response);
                        try {
                            JSONObject jsonObject
= new JSONObject(response);
```

```
                    if (jsonObject.getInt("hasil") == 1) {
                        JSONArray hasil = jsonObject.getJSONArray("Pengaduan");

                        tampil_datapengaduan.clear();
                        for (int i = 0; i < hasil.length(); i++) {
                            JSONObject c = hasil.getJSONObject(i);
                            String tgl = c.getString(koneksi.tanggal);
                            String id = c.getString(koneksi.id);
                            String kategori = c.getString(koneksi.ket);
                            String keterangan = c.getString(koneksi.uraian);
                            String status = c.getString(koneksi.status);
                            String nama_pelanggar = c.getString(koneksi.nama);
                            String nama_petugas = c.getString(koneksi.username);
                            String gmb = c.getString(koneksi.gambar);
                            String itp = c.getString(koneksi.itp);
                            String idp = c.getString(koneksi.idp);
                            String lat = c.getString(koneksi.lati);
                            String lng = c.getString(koneksi.longi);
                            String alamat = c.getString(koneksi.alamat);
                            String no = c.getString(koneksi.no_hp);
                            String stnk = c.getString(koneksi.stnk);
                            String merk = c.getString(koneksi.merk);
                            String plat = c.getString(koneksi.plat);
                            String warna = c.getString(koneksi.warna);
                            String jadwal = c.getString(koneksi.jadwal);
                            String lokasi = c.getString(koneksi.lokasi);
                            String tujuan = c.getString(koneksi.tujuan);

                            HashMap<String, String> map = new HashMap<String, String>();

                            map.put(koneksi.tanggal, tgl);
                            map.put(koneksi.id, id);

                            map.put(koneksi.ket, kategori);

                            map.put(koneksi.uraian, keterangan);

                            map.put(koneksi.status, status);

                            map.put(koneksi.nama, nama_pelanggar);

                            map.put(koneksi.username, nama_petugas);

                            map.put(koneksi.gambar, gmb);
                            map.put(koneksi.itp, itp);
                            map.put(koneksi.idp, idp);
                            map.put(koneksi.lati, lat);
                            map.put(koneksi.longi, lng);
                            map.put(koneksi.alamat, alamat);
                            map.put(koneksi.no_hp, no);
                            map.put(koneksi.stnk, stnk);
                            map.put(koneksi.merk, merk);
                            map.put(koneksi.plat, plat);
                            map.put(koneksi.warna, warna);
                            map.put(koneksi.jadwal, jadwal);
                            map.put(koneksi.lokasi, lokasi);
                            map.put(koneksi.tujuan, tujuan);

                            tampil_datapengaduan.add(map);
                        }
                        adapter = new adapter_datapengaduan(getApplicationContext(), tampil_datapengaduan);

                        rv.setAdapter(adapter);
                        dialog.dismiss();
                    } else if (jsonObject.getInt("hasil") == 0) {
                        dialog.dismiss();
```

```
                        new
SweetAlertDialog(List_Pengaduan.this,
SweetAlertDialog.ERROR_TYPE)

.setTitleText("Maaf!!!")

.setContentText("Data Tidak Ditemukan")

.setConfirmClickListener(new
SweetAlertDialog.OnSweetClickListener() {
                                @Override
                                public void
onClick(SweetAlertDialog sDialog) {

startActivity(new
Intent(getApplicationContext(),
Home.class));
                                    finish();
                                }
                            })
                            .show();
                    }
                } catch (JSONException e)
{
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener()
{
            @Override
            public                      void
onErrorResponse(VolleyError error) {

                new
SweetAlertDialog(List_Pengaduan.this,
SweetAlertDialog.ERROR_TYPE)
                    .setTitleText("Oops...")
                    .setContentText("Jaringan
Tidak Ada")
                    .show();
                dialog.dismiss();
            }
        }){
            @Override
            protected Map<String, String>
getParams() throws AuthFailureError {
                Map<String, String> param =
new HashMap<>();
                param.put(koneksi.id,
"pengaduan");
                param.put(koneksi.username,
user_);
                return param;
            }
        };
        RequestQueue requestQueue =
Volley.newRequestQueue(this);
        requestQueue.add(list_guru);
    }

  @Override
  public void onBackPressed() {
    startActivity(new
Intent(getApplicationContext(),
Home.class));
    finish();
    super.onBackPressed();
  }
}
```

LocationPickerActivity.java

```
// The MIT License (MIT)

// Copyright (c) 2018 Intuz Solutions Pvt Ltd.

// Permission is hereby granted, free of
charge, to any person obtaining a copy of this
software and associated documentation files
// (the "Software"), to deal in the Software
without restriction, including without
limitation the rights to use, copy, modify,
// merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to
permit persons to whom the Software is
// furnished to do so, subject to the following
conditions:

// THE SOFTWARE IS PROVIDED "AS
IS", WITHOUT WARRANTY OF ANY
KIND, EXPRESS OR IMPLIED,
INCLUDING BUT NOT LIMITED TO THE
WARRANTIES OF
// MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT
SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE
// LIABLE FOR ANY CLAIM, DAMAGES
OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF
OR IN
// CONNECTION WITH THE SOFTWARE
OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

package com.example.reskrim;

import android.Manifest;
```

```
import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Intent;
import android.content.IntentSender;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

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

import
com.google.android.gms.common.api.ApiEx
ception;
import
com.google.android.gms.common.api.Resol
vableApiException;
import
com.google.android.gms.common.api.Status
;
import
com.google.android.gms.location.FusedLoca
tionProviderClient;
import
com.google.android.gms.location.LocationA
vailability;
import
com.google.android.gms.location.LocationC
allback;
import
com.google.android.gms.location.LocationR
equest;
import
com.google.android.gms.location.LocationR
esult;
import
com.google.android.gms.location.LocationS
ervices;
import
com.google.android.gms.location.LocationS
ettingsRequest;
import
com.google.android.gms.location.LocationS
ettingsResponse;
import
com.google.android.gms.location.LocationS
ettingsStates;
import
com.google.android.gms.location.LocationS
ettingsStatusCodes;
import
com.google.android.gms.maps.CameraUpda
te;
import
com.google.android.gms.maps.CameraUpda
teFactory;
import
com.google.android.gms.maps.GoogleMap;
import
com.google.android.gms.maps.MapFragmen
t;
import
com.google.android.gms.maps.OnMapRead
yCallback;
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.tasks.OnComplete
Listener;
import
com.google.android.gms.tasks.OnFailureList
ener;
import
com.google.android.gms.tasks.OnSuccessLi
stener;
import com.google.android.gms.tasks.Task;
import
com.google.android.libraries.places.api.Plac
```

```
es;
import
com.google.android.libraries.places.api.mod
el.Place;
import
com.google.android.libraries.places.widget.
Autocomplete;
import
com.google.android.libraries.places.widget.
AutocompleteActivity;
import
com.google.android.libraries.places.widget.
model.AutocompleteActivityMode;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Locale;
import java.util.regex.Matcher;
import java.util.regex.Pattern;


public class LocationPickerActivity extends
AppCompatActivity implements
OnMapReadyCallback {
   private static final int
REQUEST_CHECK_SETTINGS = 2;
   private final String TAG =
LocationPickerActivity.class.getSimpleNam
e();
   private String userAddress = "";
   private String userState = "";
   private String userCity = "";
   private String userPostalCode = "";
   private String userCountry = "";
   private String userAddressline2 = "";
   private String userAddressline1 = "";
   private Bundle addressBundle;
   private List addressdetails;
   private double mLatitude;
   private double mLongitude;
   private String userCountryISOCode = null;
   private String place_id = "";
   private String place_url = " ";
   private GoogleMap mMap;
   private static final int
REQUEST_ID_MULTIPLE_PERMISSION
S = 2;
   private boolean
mLocationPermissionGranted;
   private TextView imgSearch;
   private TextView citydetail;
   private EditText addressline1;
   private EditText addressline2;
   int
PLACE_AUTOCOMPLETE_REQUEST_C
ODE = 1;

   //inital zoom
   private float previousZoomLevel = -1.0f;
   private boolean isZooming = false;

   //Declaration of
FusedLocationProviderClient
   private FusedLocationProviderClient
fusedLocationProviderClient;
   private List<AsyncTask> filterTaskList =
new ArrayList<>();
   String regex = "^(-?\\d+(\\.\\d+)?),\\s*(-
?\\d+(\\.\\d+)?)$";
   Pattern latLongPattern =
Pattern.compile(regex);
   private int doAfterPermissionProvided,
doAfterLocationSwitchedOn = 1;
   private double currentLatitude;
   private double currentLongitude;
   private LocationRequest locationRequest;
   private LocationCallback
locationCallback;

   @Override
   protected void onCreate(@Nullable
Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);

      if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.KITKAT) {

getWindow().setFlags(WindowManager.Lay
outParams.FLAG_TRANSLUCENT_STAT
US,
WindowManager.LayoutParams.FLAG_TR
ANSLUCENT_STATUS);
      }

setContentView(R.layout.activity_location_
picker);
      if(getSupportActionBar()!=null)
         getSupportActionBar().hide();
      ImageView imgCurrentloc =
findViewById(R.id.imgCurrentloc);
      Button txtSelectLocation =
findViewById(R.id.fab_select_location);
      ImageView directionTool =
findViewById(R.id.direction_tool);
      ImageView googleMapTool =
findViewById(R.id.google_maps_tool);

      imgSearch =
```

```
findViewById(R.id.imgSearch);
        addressline2                    =
findViewById(R.id.addressline2);
        citydetail                      =
findViewById(R.id.citydetails);


        // Initialize bundle
        addressBundle = new Bundle();

        //intitalization                of
FusedLocationProviderClient
        fusedLocationProviderClient     =
LocationServices.getFusedLocationProvider
Client(this);

        //Prepare for Request for current
location
        getLocationRequest();

        //define callback of location request
        locationCallback = new
LocationCallback() {
            @Override
            public                      void
onLocationAvailability(LocationAvailability
locationAvailability) {
                Log.d(TAG,
"onLocationAvailability:
isLocationAvailable = " +
locationAvailability.isLocationAvailable());
            }

            @Override
            public                      void
onLocationResult(LocationResult
locationResult) {
                Log.d(TAG, "onLocationResult: "
+ locationResult);
                if (locationResult == null) {
                    return;
                }

                //show location on map
                switch
(doAfterLocationSwitchedOn) {
                    case 1:

startParsingAddressToShow();
                        break;
                    case 2:
                        //on click of imgCurrent

showCurrentLocationOnMap(false);
                        break;
                    case 3:
                        //on Click of Direction Tool

showCurrentLocationOnMap(true);
                        break;
                }

                //Location fetched, update listener
can be removed

fusedLocationProviderClient.removeLocatio
nUpdates(locationCallback);

            }

        };

        // Try to obtain the map from the
SupportMapFragment.
        MapFragment mapFragment =
(MapFragment)
getFragmentManager().findFragmentById(R
.id.map);
        mapFragment.getMapAsync(this);
        //if you want to open the location on the
LocationPickerActivity through intent
        Intent i = getIntent();
        if (i != null) {
            Bundle extras = i.getExtras();
            if (extras != null) {
                userAddress             =
extras.getString(MapUtility.ADDRESS);
                //temp -> get lat , log from db
                mLatitude               =
getIntent().getDoubleExtra(MapUtility.LATI
TUDE, 0);
                mLongitude              =
getIntent().getDoubleExtra(MapUtility.LON
GITUDE, 0);
                userCountryISOCode      =
extras.getString(MapUtility.COUNTRY_IS
O_CODE, null);
            }
        }

        if (savedInstanceState != null) {
            mLatitude                   =
savedInstanceState.getDouble("latitude");
            mLongitude                  =
savedInstanceState.getDouble("longitude");
            userAddress                 =
savedInstanceState.getString("userAddress")
;
            currentLatitude             =
 savedInstanceState.getDouble("currentLatitu
```

```
de");
        currentLongitude = savedInstanceState.getDouble("currentLongitude");
        userCountryISOCode = savedInstanceState.getString("userCountryISOCode", null);
    }

    if (!MapUtility.isNetworkAvailable(this)) {
        MapUtility.showToast(this, "Please Connect to Internet");
    }


    imgSearch.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (!Places.isInitialized()) {

Places.initialize(LocationPickerActivity.this.getApplicationContext(),MapUtility.apiKey);
            }

            // Set the fields to specify which types of place data to return.
            List<Place.Field> fields = Arrays.asList(Place.Field.ID, Place.Field.NAME, Place.Field.ADDRESS, Place.Field.LAT_LNG);


            // Start the autocomplete intent.
            Intent intent = new Autocomplete.IntentBuilder(

AutocompleteActivityMode.FULLSCREEN, fields)

.setCountry(userCountryISOCode)

.build(LocationPickerActivity.this);

LocationPickerActivity.this.startActivityForResult(intent, PLACE_AUTOCOMPLETE_REQUEST_CODE);
        }
    });


txtSelectLocation.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent();
            // add data into intent and send back to Parent Activity

intent.putExtra(MapUtility.ADDRESS, imgSearch.getText().toString().trim());

intent.putExtra(MapUtility.LATITUDE, mLatitude);

intent.putExtra(MapUtility.LONGITUDE, mLongitude);

intent.putExtra("fullAddress",addressBundle);
            intent.putExtra("id", place_id);//if you want place id
            intent.putExtra("url", place_url);//if you want place url

LocationPickerActivity.this.setResult(Activity.RESULT_OK, intent);

LocationPickerActivity.this.finish();
        }
    });

    imgCurrentloc.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

LocationPickerActivity.this.showCurrentLocationOnMap(false);
            doAfterPermissionProvided = 2;
            doAfterLocationSwitchedOn = 2;
        }
    });

    // google maps tools
    directionTool.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

LocationPickerActivity.this.showCurrentLocationOnMap(true);
            doAfterPermissionProvided = 3;
            doAfterLocationSwitchedOn = 3;
        }
    });
```

```
googleMapTool.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
          // Default google map
          Intent intent = new
Intent(Intent.ACTION_VIEW, Uri.parse(

"http://maps.google.com/maps?q=loc:" +
mLatitude + ", " + mLongitude + ""));

LocationPickerActivity.this.startActivity(int
ent);
        }
      });

      try {

Toast.makeText(getApplicationContext(),thi
s.getResources().getString(R.string.edittext_
hint),Toast.LENGTH_SHORT).show();
      }catch (Exception e){

Toast.makeText(this,this.getResources().get
String(R.string.edittext_hint),Toast.LENGT
H_SHORT).show();

      }

   }


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

fusedLocationProviderClient.removeLocatio
nUpdates(locationCallback);
   }

   @Override
   protected void onActivityResult(int
requestCode, int resultCode, Intent data) {
      //after a place is searched
      super.onActivityResult(requestCode,
resultCode, data);
      if (requestCode ==
PLACE_AUTOCOMPLETE_REQUEST_C
ODE) {
        if (resultCode == RESULT_OK) {
          Place place =
Autocomplete.getPlaceFromIntent(data);
          userAddress = place.getAddress();
          //
addressdetails=place.getAddressComponen t
s();
          imgSearch.setText("" +
userAddress);
          mLatitude =
place.getLatLng().latitude;
          mLongitude =
place.getLatLng().longitude;
          place_id = place.getId();
          place_url =
String.valueOf(place.getWebsiteUri());

          addMarker();

        } else if (resultCode ==
AutocompleteActivity.RESULT_ERROR) {
          // TODO: Handle the error.
          Status status =
Autocomplete.getStatusFromIntent(data);
          Log.i(TAG,
status.getStatusMessage());
        } else if (resultCode ==
RESULT_CANCELED) {
          // The user canceled the operation.
        }
      } else if (requestCode ==
REQUEST_CHECK_SETTINGS) {
        //after location switch on dialog
shown
        if (resultCode != RESULT_OK) {
          //Location not switched ON

Toast.makeText(LocationPickerActivity.this,
"Location Not Available..",
Toast.LENGTH_SHORT).show();

        } else {
          // Start location request listener.
          //Location will be received
onLocationResult()
          //Once loc recvd, updateListener
will be turned OFF.
          Toast.makeText(this, "Fetching
Location...",
Toast.LENGTH_LONG).show();
          startLocationUpdates();

        }
      }
   }

   private boolean
checkAndRequestPermissions() {
      int locationPermission =
ContextCompat.checkSelfPermission(this,
```

```
Manifest.permission.ACCESS_FINE_LOC
ATION);
      int coarsePermision =
ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_L
OCATION);
      List<String> listPermissionsNeeded =
new ArrayList<>();

      if (locationPermission !=
PackageManager.PERMISSION_GRANTE
D) {

listPermissionsNeeded.add(Manifest.permiss
ion.ACCESS_FINE_LOCATION);
      }
      if (coarsePermision !=
PackageManager.PERMISSION_GRANTE
D) {

listPermissionsNeeded.add(Manifest.permiss
ion.ACCESS_COARSE_LOCATION);
      }

      if (!listPermissionsNeeded.isEmpty()) {

ActivityCompat.requestPermissions(this,
listPermissionsNeeded.toArray(new
String[0]),
REQUEST_ID_MULTIPLE_PERMISSION
S);
         return false;
      }

      //getSettingsLocation();
      return true;

   }

   private void
showCurrentLocationOnMap(final boolean
isDirectionClicked) {

      if (checkAndRequestPermissions()) {


@SuppressLint("MissingPermission")
         Task<Location> lastLocation =
fusedLocationProviderClient.getLastLocatio
n();

lastLocation.addOnSuccessListener(this,
new OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location
location) {
               if (location != null) {
                  mMap.clear();
                  if (isDirectionClicked) {
                     currentLatitude =
location.getLatitude();
                     currentLongitude =
location.getLongitude();
                     //Go to Map for Directions
                     Intent intent = new
Intent(Intent.ACTION_VIEW, Uri.parse(

"http://maps.google.com/maps?saddr=" +
currentLatitude + ", " + currentLongitude +
"&daddr=" + mLatitude + ", " + mLongitude
+ ""));

LocationPickerActivity.this.startActivity(int
ent);
                  } else {
                     //Go to Current Location
                     mLatitude =
location.getLatitude();
                     mLongitude =
location.getLongitude();

LocationPickerActivity.this.getAddressByG
eoCodingLatLng();
                  }

               } else {
                  //Gps not enabled if loc is null

LocationPickerActivity.this.getSettingsLocat
ion();

Toast.makeText(LocationPickerActivity.this,
"Location not Available",
Toast.LENGTH_SHORT).show();

               }
            }
         });

lastLocation.addOnFailureListener(new
OnFailureListener() {
            @Override
            public void onFailure(@NonNull
Exception e) {
               //If perm provided then gps not
enabled
//             getSettingsLocation();

Toast.makeText(LocationPickerActivity.this,
"Location Not Availabe",
```

```
Toast.LENGTH_SHORT).show();

            }
        });
    }

}
public Bitmap resizeMapIcons(String iconName, int width, int height){
    Bitmap imageBitmap = BitmapFactory.decodeResource(getResources(),getResources().getIdentifier(iconName, "drawable", getPackageName()));
    Bitmap resizedBitmap = Bitmap.createScaledBitmap(imageBitmap, width, height, false);
    return resizedBitmap;
}
private void addMarker() {
    CameraUpdate cameraUpdate;
    String SPACE = " , ";
    LatLng coordinate = new LatLng(mLatitude, mLongitude);
    if (mMap != null) {
        MarkerOptions markerOptions;
        try {
            mMap.clear();
            imgSearch.setText("" + userAddress);
            markerOptions = new MarkerOptions().position(coordinate).title(userAddress).icon(BitmapDescriptorFactory.fromBitmap(resizeMapIcons("ic_pointer",100,100)));
            if(isZooming) {
              //  camera will not Update
                cameraUpdate = CameraUpdateFactory.newLatLngZoom(coordinate, mMap.getCameraPosition().zoom);
            }else {
                // camera will Update zoom
                cameraUpdate = CameraUpdateFactory.newLatLngZoom(coordinate, 18);


            }


mMap.animateCamera(cameraUpdate);

mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

            Marker marker = mMap.addMarker(markerOptions);
                //marker.showInfoWindow();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
        try{
            userAddressline2 = userAddressline2.substring(0, userAddressline2.indexOf(userCity));
            //  userAddressline.replace(userCity,"");
            //  userAddressline.replace(userPostalCode,"");
            //  userAddressline.replace(userState,"");
            //  userAddressline.replace(userCountry,"");
        }catch (Exception ex){Log.d(TAG,"address error "+ex);}

        try {

addressline2.setText(userAddressline2);

citydetail.setText(userCity+SPACE+userPostalCode+SPACE+userState+SPACE+userCountry);

        }catch (Exception ex){ex.printStackTrace();}

    }


    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        mMap.clear();

mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);

mMap.getUiSettings().setMapToolbarEnabled(false);
        if (mMap.isIndoorEnabled()) {
            mMap.setIndoorEnabled(false);
        }


        mMap.setInfoWindowAdapter(new GoogleMap.InfoWindowAdapter() {

            // Use default InfoWindow frame
```

```
        @Override
        public View getInfoWindow(Marker
arg0) {
            return null;
        }

        // Defines the contents of the
InfoWindow
        @Override
        public View getInfoContents(Marker
arg0) {
            View v =
getLayoutInflater().inflate(R.layout.info_wi
ndow_layout, null);

            // Getting the position from the
marker
            LatLng latLng =
arg0.getPosition();
            mLatitude = latLng.latitude;
            mLongitude = latLng.longitude;
            TextView tvLat =
v.findViewById(R.id.address);
            tvLat.setText(userAddress);
            return v;

        }
    });

mMap.getUiSettings().setZoomControlsEna
bled(true);

    // Setting a click event handler for the
map
    mMap.setOnMapClickListener(new
GoogleMap.OnMapClickListener() {
        @Override
        public void onMapClick(LatLng
latLng) {
            mMap.clear();
            mLatitude = latLng.latitude;
            mLongitude = latLng.longitude;
            Log.e("latlng", latLng + "");
            isZooming=true;

LocationPickerActivity.this.addMarker();
            if
(!MapUtility.isNetworkAvailable(LocationPi
ckerActivity.this)) {

MapUtility.showToast(LocationPickerActivi
ty.this, "Please Connect to Internet");
            }

LocationPickerActivity.this.getAddressByG
eoCodingLatLng();

        }
    });

    if (checkAndRequestPermissions()) {
        startParsingAddressToShow();
    } else {
        doAfterPermissionProvided = 1;
    }

}

  private void getSettingsLocation() {
    LocationSettingsRequest.Builder
builder = new
LocationSettingsRequest.Builder()

.addLocationRequest(locationRequest);

    Task<LocationSettingsResponse> result
=

LocationServices.getSettingsClient(this).che
ckLocationSettings(builder.build());

    result.addOnCompleteListener(new
OnCompleteListener<LocationSettingsResp
onse>() {
        @Override
        public void onComplete(@NonNull
Task<LocationSettingsResponse> task) {
            try {
                LocationSettingsResponse
response =
task.getResult(ApiException.class);
                // All location settings are
satisfied. The client can initialize location
                // requests here.
                //...
                if (response != null) {
                    LocationSettingsStates
locationSettingsStates =
response.getLocationSettingsStates();
                    Log.d(TAG,
"getSettingsLocation: " +
locationSettingsStates);

LocationPickerActivity.this.startLocationUp
dates();

                }
            } catch (ApiException exception) {
                Log.d(TAG,
"getSettingsLocation: " + exception);
```

```
                switch
(exception.getStatusCode()) {
                    case
LocationSettingsStatusCodes.RESOLUTIO
N_REQUIRED:
                        // Location settings are not
satisfied. But could be fixed by showing the
                        // user a dialog.
                        try {
                            // Cast to a resolvable
exception.
                            ResolvableApiException
resolvable = (ResolvableApiException)
exception;
                            // Show the dialog by
calling startResolutionForResult(),
                            // and check the result in
onActivityResult().

resolvable.startResolutionForResult(

LocationPickerActivity.this,

REQUEST_CHECK_SETTINGS);
                        } catch
(IntentSender.SendIntentException e) {
                            // Ignore the error.
                        } catch
(ClassCastException e) {
                            // Ignore, should be an
impossible error.
                        }
                        break;
                    case
LocationSettingsStatusCodes.SETTINGS_C
HANGE_UNAVAILABLE:
                        // Location settings are not
satisfied. However, we have no way to fix the
                        // settings so we won't show
the dialog.
                        //...
                        break;
                }
            }
        }
    });
  }

  /**
   * Show location from intent
   */
  private void startParsingAddressToShow()
{
    //get address from intent to show on map
    if (userAddress == null ||
userAddress.isEmpty()) {

        //if intent does not have address,
        //cell is blank
        showCurrentLocationOnMap(false);

    } else

        //check if address contains lat long,
then extract
        //format will be lat,lng i.e
19.23234,72.65465
        if
(latLongPattern.matcher(userAddress).match
es()) {

            Pattern p = Pattern.compile("(-
?\\d+(\\.\\d+)?)");  // the pattern to search for
            Matcher m =
p.matcher(userAddress);

            // if we find a match, get the group
            int i = 0;
            while (m.find()) {
              // we're only looking for 2s
group, so get it
              if (i == 0)
                mLatitude =
Double.parseDouble(m.group());
              if (i == 1)
                mLongitude =
Double.parseDouble(m.group());

              i++;

            }
            //show on map

getAddressByGeoCodingLatLng();
            addMarker();

        } else {
            //get latlong from String address
via reverse geo coding
            //Since lat long not present in db
            if (mLatitude == 0 && mLongitude
== 0) {

getLatLngByRevGeoCodeFromAdd();
            } else {
              // Latlong is more accurate to get
exact point on map ,
              // String address might not be
sufficient (i.e Mumbai, Mah..etc)
              addMarker();
```

```
            }
        }

    }

    @Override
    protected void onSaveInstanceState(@NonNull Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putDouble("latitude", mLatitude);
        outState.putDouble("longitude", mLongitude);
        outState.putString("userAddress", userAddress);

outState.putBundle("addressBundle",addressBundle);
        outState.putDouble("currentLatitude", currentLatitude);

outState.putDouble("currentLongitude", currentLongitude);

outState.putString("userCountryISOCode", userCountryISOCode);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                // app icon in action bar clicked; goto parent activity.
                this.finish();
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }

    @Override
    public void onBackPressed() {
        finish();
    }

    private void getAddressByGeoCodingLatLng() {

        //Get string address by geo coding from lat long
        if (mLatitude != 0 && mLongitude != 0) {

            if (MapUtility.popupWindow != null && MapUtility.popupWindow.isShowing()) {
                MapUtility.hideProgress();
            }

            Log.d(TAG, "getAddressByGeoCodingLatLng: START");
            //Cancel previous tasks and launch this one
            for (AsyncTask prevTask : filterTaskList) {
                prevTask.cancel(true);
            }

            filterTaskList.clear();
            GetAddressFromLatLng asyncTask = new GetAddressFromLatLng();
            filterTaskList.add(asyncTask);

asyncTask.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR, mLatitude, mLongitude);
        }
    }

    private void getLatLngByRevGeoCodeFromAdd() {

        //Get string address by geo coding from lat long
        if (mLatitude == 0 && mLongitude == 0) {

            if (MapUtility.popupWindow != null && MapUtility.popupWindow.isShowing()) {
                MapUtility.hideProgress();
            }

            Log.d(TAG, "getLatLngByRevGeoCodeFromAdd: START");
            //Cancel previous tasks and launch this one
            for (AsyncTask prevTask : filterTaskList) {
                prevTask.cancel(true);
            }

            filterTaskList.clear();
            GetLatLngFromAddress asyncTask =
```

```
new GetLatLngFromAddress();
            filterTaskList.add(asyncTask);

asyncTask.executeOnExecutor(AsyncTask.T
HREAD_POOL_EXECUTOR,
userAddress);
        }
    }

    @SuppressLint("StaticFieldLeak")
    private class GetAddressFromLatLng
extends AsyncTask<Double, Void, Bundle> {
        Double latitude, longitude;
        @Override
        protected void onPreExecute() {
            super.onPreExecute();

MapUtility.showProgress(LocationPickerAc
tivity.this);
        }

        @Override
        protected                    Bundle
doInBackground(Double... doubles) {
            try {

                latitude = doubles[0];
                longitude = doubles[1];
                Geocoder geocoder;
                List<Address> addresses;
                geocoder          =          new
Geocoder(LocationPickerActivity.this,
Locale.getDefault());
                StringBuilder      sb      =      new
StringBuilder();

                //get location from lat long if
address string is null
                addresses                    =
geocoder.getFromLocation(latitude,
longitude, 1);

                if (addresses != null &&
addresses.size() > 0) {

                    String address =
addresses.get(0).getAddressLine(0);
                    if (address != null)

addressBundle.putString("addressline2",addr
ess);
                        sb.append(address).append("
");

                    String city =
addresses.get(0).getLocality();
                    if (city != null)

addressBundle.putString("city",city);
                        sb.append(city).append("  ");


                    String state =
addresses.get(0).getAdminArea();
                    if (state != null)

addressBundle.putString("state",state);
                        sb.append(state).append("  ");


                    String country =
addresses.get(0).getCountryName();
                    if (country != null)

 addressBundle.putString("country",country);
                        sb.append(country).append("
");

                    String postalCode =
addresses.get(0).getPostalCode();
                    if (postalCode != null)

addressBundle.putString("postalcode",postal
Code);

sb.append(postalCode).append("  ");
                    // return sb.toString();


addressBundle.putString("fulladdress",sb.to
String());

                    return addressBundle;
                } else {
                    return null;
                }
            } catch (IOException e) {
                e.printStackTrace();

addressBundle.putBoolean("error",true);
                return addressBundle;
                //return roundAvoid(latitude) + ","
+ roundAvoid(longitude);

            }

            // return bu;
        }
```

```
        @Override
        // setting address into different components
        protected void onPostExecute(Bundle userAddress) {
            super.onPostExecute(userAddress);


LocationPickerActivity.this.userAddress = userAddress.getString("fulladdress");
            LocationPickerActivity.this.userCity = userAddress.getString("city");
            LocationPickerActivity.this.userState = userAddress.getString("state");

LocationPickerActivity.this.userPostalCode = userAddress.getString("postalcode");

LocationPickerActivity.this.userCountry = userAddress.getString("country");

LocationPickerActivity.this.userAddressline2 = userAddress.getString("addressline2");
            MapUtility.hideProgress();
            addMarker();
        }
    }

    private class GetLatLngFromAddress extends AsyncTask<String, Void, LatLng> {

        @Override
        protected void onPreExecute() {
            super.onPreExecute();

MapUtility.showProgress(LocationPickerActivity.this);
        }

        @Override
        protected LatLng doInBackground(String... userAddress) {
            LatLng latLng = new LatLng(0, 0);

            try {

                Geocoder geocoder;
                List<Address> addresses;
                geocoder = new Geocoder(LocationPickerActivity.this, Locale.getDefault());

                //get location from lat long if address string is null
                addresses = geocoder.getFromLocationName(userAddress[0], 1);

                if (addresses != null && addresses.size() > 0) {
                    latLng = new LatLng(addresses.get(0).getLatitude(), addresses.get(0).getLongitude());
                }
            } catch (Exception ignored) {
            }
            return latLng;
        }


        @Override
        protected void onPostExecute(LatLng latLng) {
            super.onPostExecute(latLng);

LocationPickerActivity.this.mLatitude = latLng.latitude;

LocationPickerActivity.this.mLongitude = latLng.longitude;
            MapUtility.hideProgress();
            addMarker();
        }
    }


    double roundAvoid(double value) {
        double scale = Math.pow(10, 6);
        return Math.round(value * scale) / scale;
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        for (AsyncTask task : filterTaskList) {
            task.cancel(true);
        }
    }

    @SuppressLint("MissingSuperCall")
    @Override
    public void onRequestPermissionsResult(int requestCode,
                                           @NonNull String permissions[],
                                           @NonNull int[] grantResults) {
        mLocationPermissionGranted = false;
```

```
        switch (requestCode) {
            case
REQUEST_ID_MULTIPLE_PERMISSION
S: {
                // If request is cancelled, the result
arrays are empty.
                if (grantResults.length > 0
                        && grantResults[0] ==
PackageManager.PERMISSION_GRANTE
D) {
                    mLocationPermissionGranted =
true;
                }
            }

        }
    }

    @Override
    protected void onResume() {
        super.onResume();
        //Do tasks for which permission was
granted by user in onRequestPermission()
        if (!isFinishing() &&
mLocationPermissionGranted) {
            // perform action required b4 asking
permission
            mLocationPermissionGranted =
false;
            switch (doAfterPermissionProvided)
{
                case 1:
                    startParsingAddressToShow();
                    break;
                case 2:

showCurrentLocationOnMap(false);
                    break;
                case 3:

 showCurrentLocationOnMap(true);
                    break;
            }

        }

    }

    private void startLocationUpdates() {

        if
(ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOC
ATION) !=
PackageManager.PERMISSION_GRANTE
D &&
ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_L
OCATION) !=
PackageManager.PERMISSION_GRANTE
D) {

Toast.makeText(LocationPickerActivity.this,
"Location not Available",
Toast.LENGTH_SHORT).show();
            return;
        }


fusedLocationProviderClient.requestLocatio
 nUpdates(locationRequest,
                locationCallback,
                null /* Looper */)
                .addOnSuccessListener(new
OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void
aVoid) {
                        Log.d(TAG,
"startLocationUpdates: onSuccess: ");
                    }
                })
                .addOnFailureListener(new
OnFailureListener() {
                    @Override
                    public void
onFailure(@NonNull Exception e) {
                        if (e instanceof ApiException)
{
                            Log.d(TAG,
"startLocationUpdates: " + ((ApiException)
e).getMessage());
                        } else {
                            Log.d(TAG,
"startLocationUpdates: " + e.getMessage());
                        }
                    }
                });

    }

    private void getLocationRequest() {
        locationRequest = new
LocationRequest();
        locationRequest.setInterval(10000);

locationRequest.setFastestInterval(3000);

locationRequest.setPriority(LocationRequest
.PRIORITY_HIGH_ACCURACY);
```

```
    }

}
```

Login.java

```
package com.example.reskrim;

import
androidx.appcompat.app.AppCompatActivit
y;

import android.app.Dialog;
import android.content.Intent;
import android.content.SharedPreferences;
import
android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import
com.google.firebase.iid.FirebaseInstanceId;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

import
cn.pedant.SweetAlert.SweetAlertDialog;

public class Login extends
AppCompatActivity {
    EditText username, password;
    Button masuk;
    String user_, nama_;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_login);
        bacaPreferensi();
        username =
findViewById(R.id.ed_username);
        password =
findViewById(R.id.ed_password);
        masuk = findViewById(R.id.bt_masuk);



        masuk.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                login();
            }
        });

        if (user_.equals("0")) {

        } else {
            startActivity(new
Intent(getApplicationContext(),
Home.class));
            finish();
        }
    }
    private void bacaPreferensi() {
        SharedPreferences pref =
getSharedPreferences("akun",
MODE_PRIVATE);
        user_ =
pref.getString(koneksi.username, "0");
        nama_ = pref.getString(koneksi.nama,
"0");
    }
    private void login() {
        String token =
FirebaseInstanceId.getInstance().getToken();

        String
ur=username.getText().toString();
        String
pw=password.getText().toString();

        if (ur.equals("")) {
            username.setError("Belum diisi");
            username.requestFocus();
        } else if (pw.equals("")) {
            password.setError("Belum diisi");
            password.requestFocus();
        } else {
            Dialog dialog=new Dialog(this);

dialog.setContentView(R.layout.loading);
```

```
        if (dialog.getWindow()!=null){

dialog.getWindow().setBackgroundDrawabl
e(new ColorDrawable(0));
        }
        dialog.show();
        dialog.setCancelable(false);
        StringRequest simpan = new
StringRequest(Request.Method.POST,
koneksi.login,
                new
Response.Listener<String>() {
                    @Override
                    public void
onResponse(String response) {
                        try {
                            JSONObject jsonObject
=new JSONObject(response);
                            if
(jsonObject.getInt("hasil")==1) {
                            JSONArray
hasil=jsonObject.getJSONArray("login");
                            for (int i = 0; i <
hasil.length(); i++) {
                                JSONObject c =
hasil.getJSONObject(i);
                                String id =
c.getString(koneksi.username);
                                String nm =
c.getString(koneksi.nama);
                                startActivity(new
Intent(getApplicationContext(),
Home.class));
                                SharedPreferences
pref = getSharedPreferences("akun",
MODE_PRIVATE);

SharedPreferences.Editor editor = pref.edit();

editor.putString(koneksi.username,
id.toString());

editor.putString(koneksi.nama,
nm.toString());

                                editor.commit();
                                finish();
                            }
                        }else if
(jsonObject.getInt("hasil")==0)  {
                            new
SweetAlertDialog(Login.this,
SweetAlertDialog.ERROR_TYPE)

.setTitleText("Oops...")
.setContentText("Username atau Password
anda salah!")
                                    .show();
                            dialog.dismiss();
                        }
                    }catch (JSONException e) {
                        e.printStackTrace();
                    }

                    }
                }, new Response.ErrorListener()
{
                    @Override
                    public void
onErrorResponse(VolleyError error) {
                        Log.e("YWAN",
"onErrorResponse: " + error.getMessage());
                        new
SweetAlertDialog(Login.this,
SweetAlertDialog.ERROR_TYPE)
                            .setTitleText("Oops...")
                            .setContentText("" +
error.getMessage())
                            .show();
                        dialog.dismiss();
                    }
                }) {
                    @Override
                    protected Map<String, String>
getParams() throws AuthFailureError {
                        Map<String, String> param =
new HashMap<>();
                        param.put(koneksi.username,
ur);
                        param.put(koneksi.password,
pw);
                        param.put("token", token);
                        return param;
                    }
                };
                RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationCo
ntext());
                requestQueue.add(simpan);
            }
        }
}
```

Mainactivity.java

```
package com.example.reskrim;

import
androidx.appcompat.app.AppCompatActivit
```

```
y;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;


public class MainActivity extends AppCompatActivity implements View.OnClickListener {

   private TextView txtLatLong;
   private TextView txtAddress;
   private static final int ADDRESS_PICKER_REQUEST = 1020;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);
      MapUtility.apiKey = getResources().getString(R.string.api_key);

findViewById(R.id.btnAddressPicker).setOnClickListener(this);
      txtLatLong = findViewById(R.id.txtLatLong);
      txtAddress = findViewById(R.id.txtAddress);
   }
   @Override
   public void onClick(View view) {
      switch (view.getId()) {
         case R.id.btnAddressPicker:
            Intent intent = new Intent(MainActivity.this, LocationPickerActivity.class);
            startActivityForResult(intent, ADDRESS_PICKER_REQUEST);
            break;
      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent data) {
      super.onActivityResult(requestCode, resultCode, data);

      if (requestCode == ADDRESS_PICKER_REQUEST) {
         try {
            if (data != null && data.getStringExtra(MapUtility.ADDRESS) != null) {
               // String address = data.getStringExtra(MapUtility.ADDRESS);
               double currentLatitude = data.getDoubleExtra(MapUtility.LATITUDE, 0.0);
               double currentLongitude = data.getDoubleExtra(MapUtility.LONGITUDE, 0.0);
               Bundle completeAddress =data.getBundleExtra("fullAddress");
               /* data in completeAddress bundle
               "fulladdress"
               "city"
               "state"
               "postalcode"
               "country"
               "addressline1"
               "addressline2"
                */

               txtLatLong.setText(new StringBuilder().append("http://www.google.com/maps/place/").append(currentLatitude).append

(",").append(currentLongitude).toString());
            }
         } catch (Exception ex) {
            ex.printStackTrace();
         }
      }
   }
}
```

Pengaduan.java

```
package com.example.reskrim;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.Manifest;
import android.app.AlertDialog;
import android.app.DatePickerDialog;
```

```
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import
android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.bumptech.glide.Glide;
import
com.bumptech.glide.load.engine.DiskCache
Strategy;
import
com.bumptech.glide.request.target.SimpleTa
rget;
import
com.bumptech.glide.request.transition.Trans
ition;
import
com.google.android.material.textfield.TextIn
putLayout;
import com.karumi.dexter.Dexter;
import com.karumi.dexter.PermissionToken;
import
com.karumi.dexter.listener.PermissionDenie
dResponse;
import
com.karumi.dexter.listener.PermissionGrante
dResponse;
import
com.karumi.dexter.listener.PermissionReque
st;
import
com.karumi.dexter.listener.single.Permission
Listener;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;

import
cn.pedant.SweetAlert.SweetAlertDialog;
import es.dmoral.toasty.Toasty;
import
pl.aprilapps.easyphotopicker.EasyImage;

public class Pengaduan extends
AppCompatActivity {
   ImageView loc, date, foto;
   EditText user, tanggal,nama, alamat,hp,
stnk,merk,plat,warna,keterangan;
   List<String> datakategori = new
ArrayList<String>();
   List<String> datatujuan = new
ArrayList<String>();
   String lng, lti;
   Button save;
   CheckBox cek;
   TextInputLayout
tujuanPembayaranLayout;

   String user_, nama_;
   Spinner kate, tujuan;
   public static final int
REQUEST_CODE_CAMERA = 001;
   public static final int
REQUEST_CODE_GALLERY = 002;
   Bitmap bitmap;
   String gmb="0";
   private static final int
ADDRESS_PICKER_REQUEST = 1020;

   RecyclerView rv;
   ArrayList<HashMap<String, Object>>
tampil_datapengaduan = new
ArrayList<HashMap<String, Object>>();
   private adapter madapter;
   @Override
```

```
   protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_pengaduan);
        bacaPreferensi();
        tampil_kategori();
        tampil_tujaun();

       tampil_pasal();
       rv=findViewById(R.id.recyclerView);;
       rv.setHasFixedSize(true);
       LinearLayoutManager lim = new LinearLayoutManager(getApplicationContext());

lim.setOrientation(LinearLayoutManager.VERTICAL);
       rv.setLayoutManager(lim);


        MapUtility.apiKey = getResources().getString(R.string.api_key);
        loc=findViewById(R.id.r_loc);
       cek=findViewById(R.id.ceklok);
       save=findViewById(R.id.m_btn);
       user=findViewById(R.id.m_user);
       kate=findViewById(R.id.spin_ket);
       date=findViewById(R.id.r_date);
       foto=findViewById(R.id.foto);

tujuan=findViewById(R.id.spin_tujuan);
       tanggal=findViewById(R.id.m_tgl);

nama=findViewById(R.id.ed_pelanggar);
       alamat=findViewById(R.id.ed_alamat);
       hp=findViewById(R.id.ed_no);
       stnk=findViewById(R.id.ed_stnk);
       merk=findViewById(R.id.ed_merk);
       plat=findViewById(R.id.ed_plat);
       warna=findViewById(R.id.ed_warna);

keterangan=findViewById(R.id.ed_uraian);
       tujuanPembayaranLayout = findViewById(R.id.tujuanPembayaranLayout);



       kate.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
           @Override
           public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
               String selectedKate=parent.getItemAtPosition(position).toString();
               handleKateSelection(selectedKate);
           }

           private void handleKateSelection(String selectedKate) {
               if (selectedKate.equals("Slip Merah")) {

tujuanPembayaranLayout.setVisibility(View.GONE);
               } else if (selectedKate.equals("Slip Biru")) {

tujuanPembayaranLayout.setVisibility(View.VISIBLE);
               }
           }

           @Override
           public void onNothingSelected(AdapterView<?> parent) {

           }
       });


       foto.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
               izinaplikasi();
               pilih_gambar();
           }
       });
       Calendar calendar = Calendar.getInstance();
       final int year = calendar.get(Calendar.YEAR);
       final int month = calendar.get(Calendar.MONTH);
       final int day = calendar.get(Calendar.DAY_OF_MONTH);

       SimpleDateFormat dateFormat = new SimpleDateFormat("d-M-yyyy", Locale.getDefault());
       String currentDate = dateFormat.format(calendar.getTime());

       tanggal.setText(currentDate);
```

```
        tanggal.setFocusable(false);

tanggal.setFocusableInTouchMode(false);
        tanggal.setClickable(false);
        date.setEnabled(false);
        date.setFocusable(false);
        date.setClickable(false);

        date.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
              DatePickerDialog datePickerDialog
= new DatePickerDialog(
                    Pengaduan.this, new
DatePickerDialog.OnDateSetListener() {
                  @Override
                  public void onDateSet(DatePicker
view, int year, int month, int dayOfMonth) {
                    month = month + 1;
                    String date = dayOfMonth + "-" +
month + "-" + year;
                    tanggal.setText(date);
                  }
                }, year, month, day
                );
                datePickerDialog.show();
              }
            });


        save.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
//              simpan_pengaduan();
//
selectedlist=madapter.listofselectedactivites(
);
              simpan_tilang();
//
//            Log.d("list",selectedlist.toString()) ;
//
//                Toast.makeText(Pengaduan.this,
""+selectedlist.toString(),
Toast.LENGTH_SHORT).show();
            }
          });

        loc.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
              Intent intent = new
Intent(Pengaduan.this,
LocationPickerActivity.class);
                startActivityForResult(intent,
ADDRESS_PICKER_REQUEST);
            }
          });

        user.setText(nama_);
    }

    private void tampil_tujaun() {
      StringRequest stringRequest = new
StringRequest(Request.Method.POST,
koneksi.tampil,
            new Response.Listener<String>() {
              @Override
              public void onResponse(String
response) {

                try {
                  Log.e("TAG", "onResponse:
"+response );
                  JSONObject jsonObject =new
JSONObject(response);
                  if (jsonObject.getInt("hasil")
== 1){
                    JSONArray result =
jsonObject.getJSONArray("Tujuan");
                    datatujuan.clear();
                    for (int i = 0; i <
result.length(); i++) {
                      JSONObject c =
result.getJSONObject(i);

datatujuan.add(c.getString(koneksi.nama));
                    }
                    ArrayAdapter<String>
adapter = new
ArrayAdapter<String>(Pengaduan.this,
R.layout.item_spinner, datatujuan);

adapter.setDropDownViewResource(android
.R.layout.simple_spinner_dropdown_item);
                    tujuan.setAdapter(adapter);

                  } else {

Toast.makeText(getApplicationContext(),
"Tidak ada",
Toast.LENGTH_SHORT).show();
                  }
                } catch (JSONException e) {
                  e.printStackTrace();
                }

              }
```

```
            },
            new Response.ErrorListener() {
              @Override
              public void
onErrorResponse(VolleyError error) {
                Log.e("TAG",
"onErrorResponse: "+error.getMessage());

Toast.makeText(getApplicationContext(),
String.valueOf(error),
Toast.LENGTH_SHORT).show();
              }
            }){
          @Override
          protected Map<String, String>
getParams() throws AuthFailureError {
            Map<String, String> param = new
HashMap<>();
            param.put(koneksi.id, "tujuan");
            return param;
          }
        };
        RequestQueue requestQueue =
Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }


// private void simpan() {
//
//                List<String> selectedlist =
madapter.listofselectedactivites();
//                          Toast.makeText(this,
selectedlist.toString(),
Toast.LENGTH_SHORT).show();
//       final ProgressDialog progressDialog =
new ProgressDialog(this);
//       progressDialog.setMessage("Mengirim
data. . .");
//     progressDialog.show();
//             StringRequest simpan = new
StringRequest(Request.Method.POST,
koneksi.smp_ch,
//           new Response.Listener<String>() {
//             @Override
//               public void onResponse(String
response) {
//               progressDialog.dismiss();
//              Log.e("YSPL", "onResponse: " +
response);
//               try {
//                 JSONObject jsonObject = new
JSONObject(response);
//                if (jsonObject.getInt("hasil") ==
1) {
//                                          new
SweetAlertDialog(Pengaduan.this,
SweetAlertDialog.SUCCESS_TYPE)
//                                .setTitleText("Data
Terkirim")
//                        .setContentText("Pengaduan
Anda Telah Terkirim!")
//                           .setConfirmText("Ok")
//
.setConfirmClickListener(new
SweetAlertDialog.OnSweetClickListener() {
//                             @Override
//                                    public void
onClick(SweetAlertDialog sDialog) {
//                                 Intent hc = new
Intent(Pengaduan.this, Home.class);
//                              startActivity(hc);
//                              finish();
//                            }
//                          })
//                          .show();
//               } else if (response.contains("0"))
{
//                                          new
SweetAlertDialog(Pengaduan.this,
SweetAlertDialog.ERROR_TYPE)
//                        .setContentText("Pengaduan
gagal")
//                           .show();
//                      progressDialog.dismiss();
//                 }
//               } catch (JSONException e) {
//                 e.printStackTrace();
//               }
//             }
//           }, new Response.ErrorListener() {
//         @Override
//                                    public void
onErrorResponse(VolleyError error) {
//                              Log.e("Pengaduan",
"onErrorResponse: " + error.getMessage());
//                                          new
SweetAlertDialog(Pengaduan.this,
SweetAlertDialog.ERROR_TYPE)
//                   .setContentText("Jaringan tidak
ada")
//                  .show();
//           progressDialog.dismiss();
//         }
//     }) {
//         @Override
//                protected Map<String, String>
getParams() throws AuthFailureError {
//             Map<String, String> param = new
HashMap<>();
```

```
//                  param.put(koneksi.nama,
selectedlist.toString());
//
//          return param;
//       }
//    };
//            RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationCo
ntext());
//    requestQueue.add(simpan);
// }


   private void tampil_pasal() {
     Dialog dialog = new Dialog(this);

dialog.setContentView(R.layout.loading2);
     if (dialog.getWindow() != null) {

dialog.getWindow().setBackgroundDrawabl
e(new ColorDrawable(0));
     }
     dialog.show();
     dialog.setCancelable(false);
     final StringRequest list_guru = new
StringRequest(Request.Method.POST,
koneksi.tampil,
          new Response.Listener<String>() {
            @Override
            public void onResponse(String
response) {
              Log.e("TAG", "onResponse: " +
response);
              try {
                JSONObject jsonObject = new
JSONObject(response);
                if (jsonObject.getInt("hasil")
== 1) {
                  JSONArray hasil =
jsonObject.getJSONArray("Pasal");

tampil_datapengaduan.clear();
                  for (int i = 0; i <
hasil.length(); i++) {
                    JSONObject c =
hasil.getJSONObject(i);
                    String nm =
c.getString(koneksi.nama);
                    HashMap<String, Object>
map = new HashMap<>();
                    map.put(koneksi.nama,
nm);
                    map.put("isChecked",
false);
 tampil_datapengaduan.add(map);
                  }
                  madapter = new
adapter(getApplicationContext(),
tampil_datapengaduan);
                  rv.setAdapter(madapter);
                  dialog.dismiss();
                } else if
(jsonObject.getInt("hasil") == 0) {
                  dialog.dismiss();
                  new
SweetAlertDialog(Pengaduan.this,
SweetAlertDialog.ERROR_TYPE)
                      .setTitleText("Maaf!!!")
                      .setContentText("Data
Tidak Ditemukan")

.setConfirmClickListener(new
SweetAlertDialog.OnSweetClickListener() {
                        @Override
                        public void
onClick(SweetAlertDialog sDialog) {
                          startActivity(new
Intent(getApplicationContext(),
Home.class));
                          finish();
                        }
                      })
                      .show();
                }
              } catch (JSONException e) {
                e.printStackTrace();
              }
            }
          }, new Response.ErrorListener() {
        @Override
        public void
onErrorResponse(VolleyError error) {
          new
SweetAlertDialog(Pengaduan.this,
SweetAlertDialog.ERROR_TYPE)
              .setTitleText("Oops...")
              .setContentText("Jaringan Tidak
Ada")
              .show();
          dialog.dismiss();
        }
      }) {
        @Override
        protected Map<String, String>
getParams() throws AuthFailureError {
          Map<String, String> param = new
HashMap<>();
          param.put(koneksi.id, "pasal");
          return param;
```

```
        }
      };
      RequestQueue requestQueue = Volley.newRequestQueue(this);
      requestQueue.add(list_guru);
    }

    private void simpan_tilang() {
      List<String> selectedPasal = madapter.getSelectedItems();
      String tgl = tanggal.getText().toString();
      String nm = nama.getText().toString();
      String alm = alamat.getText().toString();
      String nhp = hp.getText().toString();
      String sim = stnk.getText().toString();
      String mrk = merk.getText().toString();
      String plt = plat.getText().toString();
      String wrn = warna.getText().toString();
      String ket = keterangan.getText().toString();

      if (tgl.equals("")) {
        tanggal.setError("Belum diisi");
        tanggal.requestFocus();
      } else if (nm.equals("")) {
        nama.setError("Belum diisi");
        nama.requestFocus();
      } else if (alm.equals("")) {
        alamat.setError("Belum diisi");
        alamat.requestFocus();
      } else if (nhp.equals("")) {
        hp.setError("Belum diisi");
        hp.requestFocus();
      } else if (sim.equals("")) {
        stnk.setError("Belum diisi");
        stnk.requestFocus();
      } else if (mrk.equals("")) {
        merk.setError("Belum diisi");
        merk.requestFocus();
      } else if (plt.equals("")) {
        plat.setError("Belum diisi");
        plat.requestFocus();
      } else if (wrn.equals("")) {
        warna.setError("Belum diisi");
        warna.requestFocus();
      } else {
        if (gmb.equals("0")) {
          Toasty.info(Pengaduan.this, "Silahkan Lengkapi Data Terlebih Dahulu",

Toasty.LENGTH_SHORT).show();
        } else {
          if (!cek.isChecked()) {
            Toasty.info(Pengaduan.this, "Lokasi Belum Ditentukan",
Toasty.LENGTH_SHORT).show();
          } else {
            final ProgressDialog progressDialog = new ProgressDialog(this);

progressDialog.setMessage("Mengirim data. . .");
            progressDialog.show();
            StringRequest simpan = new StringRequest(Request.Method.POST, koneksi.simpan,
                new Response.Listener<String>() {
                  @Override
                  public void onResponse(String response) {
                    progressDialog.dismiss();
                    Log.e("Tilang", "onResponse: " + response);
                    try {
                      JSONObject jsonObject = new JSONObject(response);
                      if (jsonObject.getInt("hasil") == 1) {
                        new SweetAlertDialog(Pengaduan.this, SweetAlertDialog.SUCCESS_TYPE)
                            .setTitleText("Data Terkirim")

.setContentText("Data Anda Telah Terkirim!")

.setConfirmText("Ok")

.setConfirmClickListener(new SweetAlertDialog.OnSweetClickListener() {
                              @Override
                              public void onClick(SweetAlertDialog sDialog) {
                                Intent hc = new Intent(Pengaduan.this, SignatureActivity.class);

hc.putExtra("no_hp", nhp);

startActivity(hc);
                                finish();
                              }
                            })
                            .show();
                      } else {
                        SweetAlertDialog dialog = new
```

```
SweetAlertDialog(Pengaduan.this,
SweetAlertDialog.SUCCESS_TYPE)
                                .setTitleText("Data
Terkirim")

.setContentText("Data          Anda          Telah
Terkirim!")

.setConfirmText("Ok")

.setConfirmClickListener(new
SweetAlertDialog.OnSweetClickListener() {
                                    @Override
                                    public          void
onClick(SweetAlertDialog sDialog) {
                                      Intent hc = new
Intent(Pengaduan.this,
SignatureActivity.class);

hc.putExtra("no_hp", nhp);

startActivity(hc);
                                      finish();
                                    }
                                  });
                              dialog.show();
                            }
                          } catch (JSONException e)
{
                            e.printStackTrace();
                          }
                        }
                      },                                new
Response.ErrorListener() {
                    @Override
                    public                            void
onErrorResponse(VolleyError error) {
                      Log.e("Tilang",
"onErrorResponse: " + error.getMessage());
                      new
SweetAlertDialog(Pengaduan.this,
SweetAlertDialog.ERROR_TYPE)
                          .setContentText("Jaringan
tidak ada")
                          .show();
                      progressDialog.dismiss();
                    }
                  }) {
                    @Override
                    protected Map<String, String>
getParams() throws AuthFailureError {
                      Map<String, String> param =
new HashMap<>();
                      param.put(koneksi.username,
user_);
                      param.put(koneksi.tanggal,
tgl);
                      param.put(koneksi.longi, lng);
                      param.put(koneksi.lati, lti);
                      param.put(koneksi.nama, nm);
                      param.put(koneksi.alamat,
alm);
                      param.put(koneksi.no_hp,
nhp);
                      param.put(koneksi.stnk, sim);
                      param.put(koneksi.merk, mrk);
                      param.put(koneksi.plat, plt);
                      param.put(koneksi.warna,
wrn);
                      param.put(koneksi.pasal,
selectedPasal.toString());
                      param.put(koneksi.ket,
kate.getSelectedItem().toString());
                      param.put(koneksi.uraian, ket);
                      param.put(koneksi.tujuan,
tujuan.getSelectedItem().toString());
                      param.put(koneksi.gambar,
imageTostring(bitmap));
                      return param;
                    }
                  };
                  RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationCo
ntext());
                  requestQueue.add(simpan);
                }
              }
            }
          }
          private void bacaPreferensi() {
            SharedPreferences pref =
getSharedPreferences("akun",
MODE_PRIVATE);
            user_ =
pref.getString(koneksi.username, "0");
            nama_ = pref.getString(koneksi.nama,
"0");
          }
          private void izinaplikasi() {
            Dexter.withActivity(this)

.withPermission(Manifest.permission.CAM
ERA)
                .withListener(new
PermissionListener() {
                  @Override
                  public                          void
onPermissionGranted(PermissionGrantedRe
sponse response) {
                  }
```

```
            @Override
            public void
onPermissionDenied(PermissionDeniedResp
onse response) {

Toast.makeText(Pengaduan.this, "You should
accept permission",
Toast.LENGTH_SHORT).show();

            }

            @Override
            public void
onPermissionRationaleShouldBeShown(Per
missionRequest permission,
PermissionToken token) {

            }
        }).check();
    }
    private void pilih_gambar() {
        CharSequence[] item = {"Kamera",
"Galeri"};
        AlertDialog.Builder request = new
AlertDialog.Builder(this)
            .setTitle("Tambah Gambar")
            .setItems(item, new
DialogInterface.OnClickListener() {
                @Override
                public void
onClick(DialogInterface dialogInterface, int
i) {
                    switch (i){
                        case 0:
                            //Membuka Kamera
Untuk Mengambil Gambar

EasyImage.openCamera(Pengaduan.this,
REQUEST_CODE_CAMERA);
                            break;
                        case 1:
                            //Membuaka Galeri
Untuk Mengambil Gambar

EasyImage.openGallery(Pengaduan.this,
REQUEST_CODE_GALLERY);
                            break;
                    }
                }
            });
        request.create();
        request.show();
    }
    @Override
    protected void onActivityResult(int
requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode,
resultCode, data);

EasyImage.handleActivityResult(requestCo
de, resultCode, data, this, new
EasyImage.Callbacks() {
            @Override
            public void
onImagePickerError(Exception e,
EasyImage.ImageSource source, int type) {
                //Method Ini Digunakan Untuk
Menghandle Error pada Image
            }

            @Override
            public void onImagePicked(File
imageFile, EasyImage.ImageSource source,
int type) {
                //Method Ini Digunakan Untuk
Menghandle Image
                switch (type) {
                    case
REQUEST_CODE_CAMERA:
                        Glide.with(Pengaduan.this)
                            .asBitmap()
                            .load(imageFile)

.diskCacheStrategy(DiskCacheStrategy.ALL
)
                            .into(new
SimpleTarget<Bitmap>() {
                                @Override
                                public void
onResourceReady(@NonNull Bitmap
resource, @Nullable Transition<? super
Bitmap> transition) {

foto.setImageBitmap(resource);
                                    bitmap = resource;
                                    gmb="1";
                                }
                            });
                        break;

                    case
REQUEST_CODE_GALLERY:
                        Glide.with(Pengaduan.this)
                            .asBitmap()
                            .load(imageFile)

.diskCacheStrategy(DiskCacheStrategy.ALL
)
                            .into(new
```

```
SimpleTarget<Bitmap>() {
                        @Override
                        public void
onResourceReady(@NonNull Bitmap
resource, @Nullable Transition<? super
Bitmap> transition) {

foto.setImageBitmap(resource);
                            bitmap = resource;
                            gmb="1";
                        }
                    });

                break;
            }
        }

        @Override
        public void
onCanceled(EasyImage.ImageSource source,
int type) {

        }
    });
    if (requestCode ==
ADDRESS_PICKER_REQUEST) {
        try {
            if (data != null &&
data.getStringExtra(MapUtility.ADDRESS)
!= null) {
                // String address =
 data.getStringExtra(MapUtility.ADDRESS);
                double currentLatitude =
 data.getDoubleExtra(MapUtility.LATITUDE
, 0.0);
                double currentLongitude =
data.getDoubleExtra(MapUtility.LONGITU
DE, 0.0);
                Bundle completeAddress
=data.getBundleExtra("fullAddress");
                /* data in completeAddress
bundle
                "fulladdress"
                "city"
                "state"
                "postalcode"
                "country"
                "addressline1"
                "addressline2"
                 */

//                              lokasi.setText(new
StringBuilder().append("http://www.google.
com/maps/place/").append(currentLatitude).
append
//
(",").append(currentLongitude).toString());
                cek.setChecked(true);
                cek.setText("Lokasi
Terupdate");
                lng=new
StringBuilder().append(currentLongitude).to
String();
                lti=new
StringBuilder().append(currentLatitude).toSt
ring();

            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
private void tampil_kategori() {
    StringRequest stringRequest = new
StringRequest(Request.Method.POST,
koneksi.tampil,
        new Response.Listener<String>()
{
            @Override
            public void onResponse(String
response) {

                try {
                    Log.e("TAG",
"onResponse: "+response );
                    JSONObject jsonObject
=new JSONObject(response);
                    if
(jsonObject.getInt("hasil") == 1){
                        JSONArray result =
jsonObject.getJSONArray("Kategori");
                        datakategori.clear();
                        for (int i = 0; i <
result.length(); i++) {
                            JSONObject c =
result.getJSONObject(i);

datakategori.add(c.getString(koneksi.nama))
;
                        }
                        ArrayAdapter<String>
adapter = new
ArrayAdapter<String>(Pengaduan.this,
R.layout.item_spinner, datakategori);

adapter.setDropDownViewResource(android
.R.layout.simple_spinner_dropdown_item);

kate.setAdapter(adapter);
```

```
                    } else {

Toast.makeText(getApplicationContext(),
"Tidak                                ada",
Toast.LENGTH_SHORT).show();
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }

            }
        },
        new Response.ErrorListener() {
            @Override
            public                          void
onErrorResponse(VolleyError error) {
              Log.e("TAG",
"onErrorResponse: "+error.getMessage());

Toast.makeText(getApplicationContext(),
String.valueOf(error),
Toast.LENGTH_SHORT).show();
            }
        }){
      @Override
      protected Map<String, String>
getParams() throws AuthFailureError {
        Map<String, String> param = new
HashMap<>();
        param.put(koneksi.id, "kategori");
        return param;
      }
    };
    RequestQueue requestQueue =
Volley.newRequestQueue(this);
    requestQueue.add(stringRequest);
  }



  private String imageTostring(Bitmap bm){
    ByteArrayOutputStream
outputStream=new
ByteArrayOutputStream();

bm.compress(Bitmap.CompressFormat.JPE
G, 25, outputStream);

byte[]imaBytes=outputStream.toByteArray()
;
    String                    encodeImage=
Base64.encodeToString(imaBytes,Base64.D
EFAULT);
    return encodeImage;
  }

  @Override
  public void onBackPressed() {
    startActivity(new
Intent(getApplicationContext(),
Home.class));
    finish();
    super.onBackPressed();
  }
}
```

Profil.java

```
package com.example.reskrim;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import
androidx.appcompat.app.AppCompatActivit
y;

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.Color;
import
android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.util.Base64;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.bumptech.glide.Glide;
import
com.bumptech.glide.load.engine.DiskCache
Strategy;
import
com.bumptech.glide.request.target.SimpleTa
rget;
```

```
import
com.bumptech.glide.request.transition.Trans
ition;


import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.util.HashMap;
import java.util.Map;

import
cn.pedant.SweetAlert.SweetAlertDialog;
import es.dmoral.toasty.Toasty;
import
pl.aprilapps.easyphotopicker.EasyImage;


public class Profil extends
AppCompatActivity {
   TextView user, nama, jekel, ttl, nohp,
alamat, gp;
   String us,nm,jk,tl,np,alm,ft, tp;
   String user_;
   ImageView foto, edit, img_g;
   public static final int
REQUEST_CODE_CAMERA = 001;
   public static final int
REQUEST_CODE_GALLERY = 002;
   Bitmap bitmap;
   @Override
   protected void onCreate(Bundle
savedInstanceState) {
      super.onCreate(savedInstanceState);

setContentView(R.layout.activity_profil);
      bacaPreferensi();
      profil();
      user = findViewById(R.id.p_user);
      nama = findViewById(R.id.p_nama);
      jekel = findViewById(R.id.p_jk);
      ttl = findViewById(R.id.p_ttl);
      nohp = findViewById(R.id.p_nohp);
      alamat = findViewById(R.id.p_alamat);
      foto = findViewById(R.id.img_profil);
      edit = findViewById(R.id.p_edit);
      gp = findViewById(R.id.p_gpass);
      img_g=findViewById(R.id.img_ganti);

      img_g.setOnClickListener(new
View.OnClickListener() {
         @Override
         public void onClick(View view) {
            pilih_gambar();
         }
      });
      gp.setOnClickListener(new
View.OnClickListener() {
         @Override
         public void onClick(View view) {
            ganti_pass(view);
         }
      });

      edit.setOnClickListener(new
View.OnClickListener() {
         @Override
         public void onClick(View view) {
            edit_profil();
         }
      });
   }

   private void edit_profil() {
      startActivity(new Intent(Profil.this,
Edit_Profil.class));

overridePendingTransition(R.anim.slide_in_
right, R.anim.stay);
   }

   private void ganti_pass(View view) {
      AlertDialog.Builder builder=new
AlertDialog.Builder(view.getRootView().get
Context());
      View dialogview=
LayoutInflater.from(view.getRootView().get
Context()).inflate(R.layout.popup_pass,
null);
      builder.setView(dialogview);
      builder.setCancelable(true);
      final AlertDialog
alertDialog=builder.create();

alertDialog.getWindow().setBackgroundDra
wable(new
ColorDrawable(Color.TRANSPARENT));
      alertDialog.show();
      EditText
pl=dialogview.findViewById(R.id.p_lama);
      EditText
pb=dialogview.findViewById(R.id.p_baru);
      EditText
pf=dialogview.findViewById(R.id.p_konf);
      Button
sv=dialogview.findViewById(R.id.pop_simp
an);
```

```
      sv.setOnClickListener(new
View.OnClickListener() {
          @Override
          public void onClick(View view) {
              String pala=pl.getText().toString();
              String
paba=pb.getText().toString();
              String
pako=pf.getText().toString();

              if (pala.equals("")) {
                 pl.setError("Belum diisi");
                 pl.requestFocus();
              } else if (paba.equals("")) {
                 pb.setError("Belum diisi");
                 pb.requestFocus();
              }else if (pako.equals("")) {
                 pf.setError("Belum diisi");
                 pf.requestFocus();
              }else {
                 if (paba.equals(pako)) {
                    Dialog dialog = new
Dialog(Profil.this);

dialog.setContentView(R.layout.loading);
                    if (dialog.getWindow() !=
null) {

dialog.getWindow().setBackgroundDrawabl
e(new ColorDrawable(0));
                    }
                    dialog.show();
                    dialog.setCancelable(false);
                    StringRequest registrasi =
new StringRequest(Request.Method.POST,
koneksi.edit,
                            new
Response.Listener<String>() {
                               @Override
                               public void
onResponse(String response) {
                                  dialog.dismiss();
                                  try {
                                     JSONObject
jsonObject = new JSONObject(response);
                                     if
(jsonObject.getInt("hasil") == 1) {

dialog.dismiss();

alertDialog.dismiss();

Toasty.success(Profil.this, "Password
Berhasil Diganti",

Toasty.LENGTH_SHORT).show();
                                     } else if
(jsonObject.getInt("hasil") == 0) {

dialog.dismiss();

alertDialog.dismiss();

Toasty.warning(Profil.this, "Password Gagal
Diganti",

Toasty.LENGTH_SHORT).show();
                                     }else if
(jsonObject.getInt("hasil") == 2) {

dialog.dismiss();

Toasty.warning(Profil.this, "Password Lama
Salah",

Toasty.LENGTH_SHORT).show();
                                     }
                                  } catch
(JSONException e) {

e.printStackTrace();
                                  }
                               }
                            }, new
Response.ErrorListener() {
                               @Override
                               public void
onErrorResponse(VolleyError error) {
                                  new
SweetAlertDialog(Profil.this,
SweetAlertDialog.ERROR_TYPE)

.setContentText("Jaringan tidak ada")
                                     .show();
                                  dialog.dismiss();
                               }
                            }) {
                               @Override
                               protected Map<String,
String> getParams() throws AuthFailureError
{
                                  Map<String, String>
param = new HashMap<>();

param.put(koneksi.password, pala);

param.put(koneksi.alamat, paba);

param.put(koneksi.username, user_);
```

```
                    param.put(koneksi.id,
"gpass");
                    return param;
                }
            };
            RequestQueue requestQueue
=
Volley.newRequestQueue(getApplicationCo
ntext());
            requestQueue.add(registrasi);
          }else {
            pf.setError("Password Tidak
Sama");
            pf.requestFocus();
          }
        }

      }
    });

  }

  private void pilih_gambar() {
    CharSequence[] item = {"Kamera",
"Galeri"};
    AlertDialog.Builder request = new
AlertDialog.Builder(this)
        .setTitle("Tambah Gambar")
        .setItems(item, new
DialogInterface.OnClickListener() {
          @Override
          public void
onClick(DialogInterface dialogInterface, int
i) {
            switch (i){
              case 0:
                //Membuka Kamera
Untuk Mengambil Gambar

EasyImage.openCamera(Profil.this,
REQUEST_CODE_CAMERA);
                break;
              case 1:
                //Membuaka Galeri
Untuk Mengambil Gambar

EasyImage.openGallery(Profil.this,
REQUEST_CODE_GALLERY);
                break;
            }
          }
        });
    request.create();
    request.show();
  }

  @Override
  protected void onActivityResult(int
requestCode, int resultCode, @Nullable
Intent data) {
    super.onActivityResult(requestCode,
resultCode, data);

EasyImage.handleActivityResult(requestCo
de, resultCode, data, this, new
EasyImage.Callbacks() {
      @Override
      public void
onImagePickerError(Exception e,
EasyImage.ImageSource source, int type) {
        //Method Ini Digunakan Untuk
Menghandle Error pada Image
      }

      @Override
      public void onImagePicked(File
imageFile, EasyImage.ImageSource source,
int type) {
        //Method Ini Digunakan Untuk
Menghandle Image
        switch (type) {
          case
REQUEST_CODE_CAMERA:
            Glide.with(Profil.this)
                .asBitmap()
                .load(imageFile)

.diskCacheStrategy(DiskCacheStrategy.ALL
)
                .into(new
SimpleTarget<Bitmap>() {
                  @Override
                  public void
onResourceReady(@NonNull Bitmap
resource, @Nullable Transition<? super
Bitmap> transition) {

foto.setImageBitmap(resource);
                    bitmap = resource;
                    ganti_foto();
                  }
                });
            break;

          case
REQUEST_CODE_GALLERY:
            Glide.with(Profil.this)
                .asBitmap()
                .load(imageFile)
```

```
.diskCacheStrategy(DiskCacheStrategy.ALL
)
                        .into(new
SimpleTarget<Bitmap>() {
                            @Override
                            public void
onResourceReady(@NonNull Bitmap
resource, @Nullable Transition<? super
Bitmap> transition) {

foto.setImageBitmap(resource);
                                bitmap = resource;
                                ganti_foto();
                            }
                        });

                    break;
                }
            }

            @Override
            public void
onCanceled(EasyImage.ImageSource source,
int type) {

            }
        });
    }

    private void ganti_foto() {
        Dialog dialog = new Dialog(Profil.this);

dialog.setContentView(R.layout.loading);
        if (dialog.getWindow() != null) {

dialog.getWindow().setBackgroundDrawabl
e(new ColorDrawable(0));
        }
        dialog.show();
        dialog.setCancelable(false);
        StringRequest registrasi = new
StringRequest(Request.Method.POST,
koneksi.edit,
                new Response.Listener<String>()
{
                    @Override
                    public void onResponse(String
response) {
                        dialog.dismiss();
                        try {
                            JSONObject jsonObject =
new JSONObject(response);
                            if
(jsonObject.getInt("hasil") == 1) {
                                dialog.dismiss();

Toasty.success(Profil.this, "Foto Profil
diganti",

Toasty.LENGTH_SHORT).show();
                            } else if
(jsonObject.getInt("hasil") == 0) {
                                dialog.dismiss();

Toasty.warning(Profil.this, "Foto Profil gagal
diganti",

Toasty.LENGTH_SHORT).show();
                            }
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                }, new Response.ErrorListener() {
            @Override
            public void
onErrorResponse(VolleyError error) {
                new SweetAlertDialog(Profil.this,
SweetAlertDialog.ERROR_TYPE)
                        .setContentText("Jaringan
tidak ada")
                        .show();
                dialog.dismiss();
            }
        }) {
            @Override
            protected Map<String, String>
getParams() throws AuthFailureError {
                Map<String, String> param = new
HashMap<>();
                param.put(koneksi.username,
user_);
                param.put(koneksi.gambar,
imageTostring(bitmap));
                param.put(koneksi.id, "gfoto");
                return param;
            }
        };
        RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationCo
ntext());
        requestQueue.add(registrasi);
    }

    private String imageTostring(Bitmap bm){
        ByteArrayOutputStream
outputStream=new
ByteArrayOutputStream();

bm.compress(Bitmap.CompressFormat.JPE
```

```
G, 25, outputStream);

byte[]imaBytes=outputStream.toByteArray()
;
      String                encodeImage=
Base64.encodeToString(imaBytes,Base64.D
EFAULT);
      return encodeImage;
   }
   private void bacaPreferensi() {
      SharedPreferences          pref      =
getSharedPreferences("akun",
MODE_PRIVATE);
      user_                                =
pref.getString(koneksi.username,  "0");
   }
   private void profil() {
      StringRequest      cari      =      new
StringRequest(Request.Method.POST,
koneksi.profil,
            new   Response.Listener<String>()
{
               @Override
               public  void  onResponse(String
response) {
                  try {
                     JSONObject      jsonObject
=new JSONObject(response);
                     if
(jsonObject.getInt("hasil")==1) {
                        JSONArray
hasil=jsonObject.getJSONArray("Profil");
                        for  (int  i  =  0;  i  <
hasil.length(); i++) {
                           JSONObject      c      =
hasil.getJSONObject(i);
                           nm                   =
c.getString(koneksi.nama);
                           jk                   =
c.getString(koneksi.jenis_kelamin);
                           tl                   =
c.getString(koneksi.tanggal);
                           tp                   =
c.getString(koneksi.tempat);
                           alm                  =
c.getString(koneksi.alamat);
                           np                   =
c.getString(koneksi.no_hp);
                           ft                   =
c.getString(koneksi.gambar);

                        }
                        nama.setText(nm);

Glide.with(Profil.this).load(ft)
                              .into(foto);
                        if (jk.equals("L")){
                           jekel.setText("Laki    -
Laki");
                        }else if(jk.equals("P")){

jekel.setText("Perempuan");
                        }
                        user.setText(user_);
                        ttl.setText(tp+", "+tl);
                        nama.setText(nm);
                        alamat.setText(alm);
                        nohp.setText(np);
                     }
                  }catch (JSONException e) {
                     e.printStackTrace();
                  }
               }
            }, new Response.ErrorListener() {
         @Override
         public                               void
onErrorResponse(VolleyError error) {
            new  SweetAlertDialog(Profil.this,
SweetAlertDialog.ERROR_TYPE)
                  .setTitleText("Oops...")
                  .setContentText("Jaringan
Tidak Ada")
                  .show();
         }
      }) {
         @Override
         protected      Map<String,      String>
getParams() throws AuthFailureError {
            Map<String, String> param = new
HashMap<>();
            param.put(koneksi.username,
user_);
            return param;
         }
      };
      RequestQueue      requestQueue      =
Volley.newRequestQueue(getApplicationCo
ntext());
      requestQueue.add(cari);
   }

   @Override
   public void onBackPressed() {
      startActivity(new
Intent(getApplicationContext(),
Home.class));
      finish();
      super.onBackPressed();
   }
}
```

SignatureActivity.java

```
package com.example.reskrim;

import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import
androidx.appcompat.app.AppCompatActivit
y;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import
com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import
com.github.gcacace.signaturepad.views.Sign
aturePad;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.ByteArrayOutputStream;
import java.util.HashMap;
import java.util.Map;

public class SignatureActivity extends
AppCompatActivity {

    private SignaturePad signaturePad;
    private Button btnClear;
    private Button btnSave;
    private String noHp; // Variable to hold the
offender's name

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

setContentView(R.layout.activity_signature)
;

        // Get the offender's name from the
previous activity
        noHp =
getIntent().getStringExtra("no_hp");

        // Initialize views
        signaturePad =
findViewById(R.id.signaturePad);
        btnClear =
findViewById(R.id.btnClear);
        btnSave = findViewById(R.id.btnSave);

        // Disable save button initially
        btnSave.setEnabled(false);

        // Set up button click listeners
        btnClear.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                signaturePad.clear();
                btnSave.setEnabled(false);
            }
        });

        btnSave.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                saveSignature();
            }
        });

        // Set up signature pad listeners
        signaturePad.setOnSignedListener(new
SignaturePad.OnSignedListener() {
            @Override
            public void onStartSigning() {
                // Enable the clear and save buttons
when user starts signing
                btnClear.setEnabled(true);
                btnSave.setEnabled(true);
            }

            @Override
            public void onSigned() {
                // Implementation if needed when
user completes signing
            }

            @Override
            public void onClear() {
                // Disable save button when
signature is cleared
                btnSave.setEnabled(false);
            }
        });
    }

    private void saveSignature() {
        // Get signature bitmap from
SignaturePad
```

```
        Bitmap signatureBitmap = signaturePad.getSignatureBitmap();

        // Convert bitmap to base64 string
        ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
        signatureBitmap.compress(Bitmap.CompressFormat.PNG, 100, byteArrayOutputStream);
        byte[] imageBytes = byteArrayOutputStream.toByteArray();
        final String encodedImage = Base64.encodeToString(imageBytes, Base64.DEFAULT);

        // Create a StringRequest to send data to PHP backend
        StringRequest stringRequest = new StringRequest(Request.Method.POST, "http://etilang.us.to/layanan/save_signature.php",
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        try {
                            // Log response for debugging
                            Log.d("SignatureActivity", "Response: " + response);

                            // Handle JSON response
                            JSONObject jsonResponse = new JSONObject(response);
                            String status = jsonResponse.getString("status");
                            String message = jsonResponse.getString("message");

                            // Show toast message based on response status
                            Toast.makeText(SignatureActivity.this, message, Toast.LENGTH_SHORT).show();

                            // Handle success or failure response accordingly
                            if ("success".equals(status)) {
                                // Signature image saved successfully
                                // Redirect to Home Activity
                                Intent intent = new Intent(SignatureActivity.this, Home.class);
                                startActivity(intent);
                                finish();
                            } else {
                                // Error saving signature image
                                // Handle error case
                            }
                        } catch (JSONException e) {
                            e.printStackTrace();
                            Toast.makeText(SignatureActivity.this, "Error parsing JSON response", Toast.LENGTH_SHORT).show();
                        }
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        error.printStackTrace();
                        Toast.makeText(SignatureActivity.this, "Error saving signature image", Toast.LENGTH_SHORT).show();
                    }
                }) {
            @Override
            protected Map<String, String> getParams() {
                Map<String, String> params = new HashMap<>();
                params.put("no_hp", noHp); // Sending offender' number
                params.put("signatureImage", encodedImage);
                return params;
            }
        };

        // Add the request to the RequestQueue
        RequestQueue requestQueue = Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
    }

}
```

Edit.php

```
<?php
require_once('koneksi.php');
```

```
$respon=array();

if
($_SERVER['REQUEST_METHOD']=='PO
ST'){
   $id=$_POST['akses'];

   if ($id=="gfoto") {
      $user=$_POST['user'];
      $gambar= $_POST['gambar'];
      $url=date("d.m.yy-h.i.sa").".jpeg";

      $update=mysqli_query($con, "UPDATE
 `tb_masyarakat` SET `foto`='$url' WHERE
`username`='$user'");
                  $upload=$link.date("d.m.yy-
h.i.sa").".jpeg";
      if($update){
                     file_put_contents($upload,
base64_decode($gambar));
         $respon['hasil']=1;
         $respon['pesan']='Update Berhasil';

      }else{
         $respon['hasil']=0;
         $respon['pesan']='Update Gagal';

      }
   }elseif ($id=="gpass") {
      $pbaru=$_POST['alamat'];
      $plama= $_POST['pass'];
      $user=$_POST['user'];
               $sql="SELECT * FROM
`tb_masyarakat` WHERE username='$user'";
         $hasil= mysqli_query($con, $sql);
         if(mysqli_num_rows($hasil)>0){
                                          while
 ($row=mysqli_fetch_array($hasil)){
               $idu=$row['password'];
            }
         }

         if ($idu==$plama) {
                  $update=mysqli_query($con,
"UPDATE `tb_masyarakat` SET
`password`='$pbaru' WHERE
username='$user'");
            if($update){
               $respon['hasil']=1;
                     $respon['pesan']='Update
Berhasil';

            }else{
               $respon['hasil']=0;
               $respon['pesan']='Update Gagal';
            }
         }else{
            $respon['hasil']=2;
            $respon['pesan']='Update Gagal';
         }
   }elseif ($id=="editdata") {
      $user=$_POST['user'];
      $nama=$_POST['nama'];
      $jk=$_POST['jk'];
      $tempat_lahir=$_POST['tempat'];
                  $tanggal_lahir=date('Y-m-d',
strtotime($_POST['tanggal']));
      $alamat=$_POST['alamat'];
      $no=$_POST['no'];

      $update=mysqli_query($con, "UPDATE
 `tb_masyarakat` SET
`nama`='$nama',`jenis_kelamin`='$jk',`temp
at_lahir`='$tempat_lahir',`tanggal_lahir`='$ta
nggal_lahir',`alamat`='$alamat',`no_hp`='$no
' WHERE `username`='$user'");
      if($update){
         $respon['hasil']=1;
         $respon['pesan']='Update Berhasil';
      }else{
         $respon['hasil']=0;
         $respon['pesan']='Update Gagal';

      }

   }

}else{
   $respon['hasil']=3;
      $respon['pesan']='Data Tidak diterima';
}
echo json_encode($respon);

 ?>
```

Kirim_notif.php

```
<?php
// Konfigurasi koneksi database
$servername = "localhost";
$username = "dragonxp_etilang"; // Ubah
sesuai dengan username database Anda
$password = "Okitosenpai123."; // Ubah
sesuai dengan password database Anda
$dbname = "dragonxp_etilang"; // Ubah
sesuai dengan nama database Anda

// Buat koneksi
```

```
$conn = new mysqli($servername, $username, $password, $dbname);

// Cek koneksi
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Ambil data dari request
$tgl = isset($_POST['tgl']) ? $_POST['tgl'] : '';
$nama_petugas = isset($_POST['nama_petugas']) ? $_POST['nama_petugas'] : '';
$nama_pelanggar = isset($_POST['nama_pelanggar']) ? $_POST['nama_pelanggar'] : '';
$alamat = isset($_POST['alamat']) ? $_POST['alamat'] : '';
$nohp = isset($_POST['nohp']) ? $_POST['nohp'] : '';
$stnk = isset($_POST['stnk']) ? $_POST['stnk'] : '';
$merk = isset($_POST['merk']) ? $_POST['merk'] : '';
$plat = isset($_POST['plat']) ? $_POST['plat'] : '';
$warna = isset($_POST['warna']) ? $_POST['warna'] : '';
$jadwal = isset($_POST['jadwal']) ? $_POST['jadwal'] : '';
$lokasi_sidang = isset($_POST['lokasi_sidang']) ? $_POST['lokasi_sidang'] : '';
$tujuan = isset($_POST['tujuan']) ? $_POST['tujuan'] : '';
$kategori = isset($_POST['kategori']) ? $_POST['kategori'] : '';
$keterangan = isset($_POST['keterangan']) ? $_POST['keterangan'] : '';
$status = isset($_POST['status']) ? $_POST['status'] : '';
$id_tilang = isset($_POST['id_tilang']) ? $_POST['id_tilang'] : ''; // Ambil id_tilang dari request

// Pastikan id_tilang ada
if (empty($id_tilang)) {
    die("Error: id_tilang tidak ditemukan.");
}

// Ambil data denda dari tabel pasal
$sql = "SELECT * FROM tb_pelanggaran
    JOIN pasal ON tb_pelanggaran.id_pasal = pasal.id_pasal
    WHERE id_laporan = '$id_tilang'";
$result = $conn->query($sql);

// Debugging
if ($result === false) {
    die("Error: " . $conn->error);
}

$denda = '';
$pasal = '';

if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
        $denda .= "- " . $row['keterangan'] . "\n";
        $pasal .= "- " . $row['nama_pasal'] . "\n";
    }
} else {
    $denda = "Data denda tidak ditemukan.";
    $pasal = "Pasal tidak ditemukan.";
}

// Format pesan yang akan dikirim
$message = "Yth Bpk/Ibu Di Tempat \n" .
    "Pesan Ini Dikirim Oleh Satlantas Polres Parepare:\n" .
    "\n" .
    "Berikut Ini Detail Tilang Anda:\n" .
    "\n" .
    "Tanggal: $tgl\n" .
    "Nama Petugas: $nama_petugas\n" .
    "Nama Pelanggar: $nama_pelanggar\n" .
    "Alamat: $alamat\n" .
    "No HP: $nohp\n" .
    "NIK: $stnk\n" .
    "Merk: $merk\n" .
    "Plat: $plat\n" .
    "Warna: $warna\n";

if ($kategori == "Slip Merah") {
    $message .= "Pasal:\n".
        "$pasal" .
        "Kategori: $kategori\n" .
        "Jadwal Sidang: $jadwal\n" .
        "Lokasi Sidang: $lokasi_sidang\n" .
        "Status: $status\n";
} elseif ($kategori == "Slip Biru") {
    $message .= "Pasal:\n".
        "$pasal" .
        "Kategori: $kategori\n" .
        "Denda: \n".
        "$denda" .
        "Tujuan: $tujuan\n" .
        "Status: $status\n";
```

```
}

// Token dan target nomor WhatsApp
$token = "V5EYmv+_JE1Y#mfL_2R+";
$target = $nohp; // Ganti dengan nomor tujuan

$curl = curl_init();

curl_setopt_array($curl, array(
        CURLOPT_URL => 'https://api.fonnte.com/send',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_ENCODING => '',
  CURLOPT_MAXREDIRS => 10,
  CURLOPT_TIMEOUT => 0,
  CURLOPT_FOLLOWLOCATION => true,
      CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array(
    'target' => $target,
    'message' => $message,
  ),
  CURLOPT_HTTPHEADER => array(
    'Authorization: ' . $token
  ),
));

$response = curl_exec($curl);
if (curl_errno($curl)) {
  $error_msg = curl_error($curl);
}
curl_close($curl);

if (isset($error_msg)) {
  echo $error_msg;
}
echo $response;

// Tutup koneksi database
$conn->close();
?>
```

Koneksi.php

```
<?php
$con=mysqli_connect('localhost','dragonxp_etilang','Okitosenpai123.','dragonxp_etilang');
$link="foto_user/";
$lg="http://etilang.us.to/layanan/foto_user/";
$fp="http://etilang.us.to/layanan/foto_pengaduan/";
$gp="foto_pengaduan/";
if (!$con) {
        die("Connection failed: " . mysqli_connect_error());
}

?>
```

Login.php

```
<?php
require_once('koneksi.php');
$respon=array();
if ($_SERVER['REQUEST_METHOD']=='POST'){

  $user=$_POST['user'];
  $pass=$_POST['pass'];
  $token=$_POST['token'];

   $edit="UPDATE `tb_masyarakat` SET `token`='$token' WHERE username='$user'";
  $hedit=mysqli_query($con, $edit);
if ($hedit) {
  $sql="SELECT * FROM `tb_masyarakat` WHERE username='$user' and password='$pass'";
  $hasil= mysqli_query($con, $sql);
  if(mysqli_num_rows($hasil)>0){
    $respon["login"]=array();
                                    while ($row=mysqli_fetch_array($hasil)){
      $index=array();
      $index["nama"] = $row['nama'];
                        $index["user"] = $row['username'];
          array_push($respon["login"], $index);
      $respon["hasil"]=1;
      $respon["pesan"]="sukses";

    }echo json_encode($respon);

  }else{
    $respon["hasil"]=0;
      $respon["pesan"]="Gagal";
```

```
        echo json_encode($respon);
    }
    mysqli_close($con);

}

 }
 ?>
```

Profil.php

```
<?php
require_once('koneksi.php');

if($con){
    $id=$_POST['user'];
      $sql=mysqli_query($con, "SELECT * FROM `tb_masyarakat` WHERE username='$id'") or die (mysqli_error());
    if(mysqli_num_rows($sql)>0){
       $respon["Profil"]=array();
        while ($row=mysqli_fetch_array($sql))
{
          $hasil=array();
          $hasil["nama"] = $row['nama'];
          $hasil["jk"] = $row['jenis_kelamin'];
                          $hasil["tempat"] = $row['tempat_lahir'];
              $hasil["tanggal"] = date('d F Y', strtotime($row['tanggal_lahir']));
             $hasil["tanggall"] = date('d-m-Y', strtotime($row['tanggal_lahir']));
          $hasil["alamat"] = $row['alamat'];
          $hasil["no"] = $row['no_hp'];
                          $hasil["gambar"] = $lg.$row['foto_user'];
          array_push($respon["Profil"], $hasil);
          $respon["hasil"]=1;
       }
       echo json_encode($respon);
    }else{
       $respon["hasil"]=0;
              $respon["pesan"]="Hasil Tidak Ditemukan";
       echo json_encode($respon);
    }
}else{
          echo json_encode(array('respon'=> 'koneksi gagal'));
}
mysqli_close($con);


 ?>
```

Register.php

```
<?php
require_once('koneksi.php');
$respon=array();

if
($_SERVER['REQUEST_METHOD']=='POST'){
    $user=$_POST['user'];
    $nama=$_POST['nama'];
    $jk=$_POST['jk'];
    $tempat_lahir=$_POST['tempat'];
                $tanggal_lahir=date('Y-m-d', strtotime($_POST['tanggal']));
    $alamat=$_POST['alamat'];
    $no=$_POST['no'];
    $pw=$_POST['pass'];
    $gambar= $_POST['gambar'];
    $url=date("d.m.yy-h.i.sa").".jpeg";

     $register=mysqli_query($con, "INSERT INTO `tb_masyarakat`(`username`, `nama`, `jenis_kelamin`, `tempat_lahir`, `tanggal_lahir`, `alamat`, `no_hp`, `password`, `foto_user`,`token`) VALUES ('$user','$nama','$jk','$tempat_lahir','$tanggal_lahir','$alamat','$no','$pw','$url','0')");
                $upload=$link.date("d.m.yy-h.i.sa").".jpeg";
    if($register){
                  file_put_contents($upload, base64_decode($gambar));
       $respon['hasil']=1;
       $respon['pesan']='Registrasi Berhasil';

    }else{
       $respon['hasil']=0;
       $respon['pesan']='Registrasi Gagal';

    }
}else{
    $respon['hasil']='0';
      $respon['pesan']='Data Tidak diterima';
}
echo json_encode($respon);

 ?>
```

Save_signature.php

```
<?php
require_once('koneksi.php');

// Path to save signature images
```

```
$uploadDirectory = 'foto_signature/';

// Validate if request is POST
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['signatureImage']) && isset($_POST['no_hp'])) {

    // Check if directory exists or create it
    if (!file_exists($uploadDirectory)) {
        mkdir($uploadDirectory, 0777, true);
    }

    // Decode base64 encoded image
                $encodedImage = $_POST['signatureImage'];
                $decodedImage = base64_decode($encodedImage);

    // Generate unique file name for image
     $fileName = uniqid() . '.png'; // Change extension as needed

    // Path to save image
        $uploadPath = $uploadDirectory . $fileName;

    // Save image to server
          if (file_put_contents($uploadPath, $decodedImage)) {
        // Image saved successfully

          // Save image filename to database tb_tilang
        $conn = $con;

        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        // Prepare statement
            $stmt = $conn->prepare("UPDATE tb_tilang SET signature = ?, updated_at = NOW() WHERE no_hp = ?");

        // Bind parameters
            $stmt->bind_param("ss", $fileName, $_POST['no_hp']); // Save only filename

        // Execute statement
        if ($stmt->execute()) {
            $response = array(
                'status' => 'success',
                'message' => 'Tanda tangan berhasil disimpan.',
                'signature_path' => $uploadPath // Optional: Return full path if needed
            );
        } else {
            $response = array(
                'status' => 'error',
                'message' => 'Gagal menyimpan tanda tangan ke database.'
            );
        }

        // Close statement and connection
        $stmt->close();
        $conn->close();

    } else {
        // Failed to save image
        $response = array(
            'status' => 'error',
            'message' => 'Gagal menyimpan tanda tangan di server.'
        );
    }
} else {
    // Invalid request
    $response = array(
        'status' => 'error',
        'message' => 'Request tidak valid.'
    );
}

// Return JSON response
header('Content-Type: application/json');
echo json_encode($response);
?>
```

Simpan.php

```
<?php
require_once('koneksi.php');
$respon=array();
date_default_timezone_set('Asia/Jakarta');

if ($_SERVER['REQUEST_METHOD']=='POST'){

    $user=$_POST['user'];
    $alamat=$_POST['alamat'];
    $kate=$_POST['ket'];
    $longi=$_POST['long'];
    $lati=$_POST['lat'];
```

```
                    $tgl=date('Y-m-d',
strtotime($_POST['tanggal']));
   $nama=$_POST['nama'];
   $no=$_POST['no'];
   $stnk=$_POST['stnk'];
   $merk=$_POST['merk'];
   $plat=$_POST['plat'];
   $warna=$_POST['warna'];
   $pasal=$_POST['pasal'];
   $uraian=$_POST['uraian'];
   $tujuan=$_POST['tujuan'];
   $foto=$_POST['gambar'];
   $date=date("d.m.yy-h.i.sa");
   $jam=date("G:i:s");
     $res = preg_replace(array('/^\[/','/\]$/'),
'',$pasal);
   $arr_kalimat = explode (", ",$res);

       $sql="SELECT max(id_tilang) as
maxKode FROM tb_tilang";
   $hasil= mysqli_query($con, $sql);
   if(mysqli_num_rows($hasil)>0){
                                        while
($row=mysqli_fetch_array($hasil)){
        $kode=$row['maxKode'];
        $noUrut = (int) substr($kode, 3, 3);
        $noUrut++;
        $index=array();
        $char = "TL";
            $id_lapor=  $char.sprintf("%03s",
$noUrut);

     }
   }

    $aks="SELECT * FROM `tb_bidang`
WHERE nama_bidang='$kate'";
   $hasil_aks= mysqli_query($con, $aks);
   if(mysqli_num_rows($hasil_aks)>0){
                                        while
($row=mysqli_fetch_array($hasil_aks)){
        $id_kate=$row["kode_kategori"];
     }
   }

            $tjn="SELECT       *       FROM
`tujuan_pembayaran`                WHERE
nama_tujuan='$tujuan'";
   $hasil_tjn= mysqli_query($con, $tjn);
   if(mysqli_num_rows($hasil_tjn)>0){
                                        while
($row=mysqli_fetch_array($hasil_tjn)){
        $id_tjn=$row["id_tujuan"];
     }
   }

  foreach ($arr_kalimat as $key => $value) {
        $aks="SELECT * FROM `pasal`
WHERE nama_pasal='$value'";
         $hasil_pasal=  mysqli_query($con,
$aks);
     if(mysqli_num_rows($hasil_pasal)>0){
                                        while
($row=mysqli_fetch_array($hasil_pasal)){
          $id_pasal=$row["id_pasal"];
        }
     }

       mysqli_query($con, "INSERT INTO
`tb_pelanggaran`(`id_laporan`,      `id_pasal`)
VALUES ('$id_lapor','$id_pasal')");
   }
   $url=$id_lapor.".jpeg";

     $lapor=mysqli_query($con,  "INSERT
INTO `tb_tilang`(`id_tilang`, `jam`, `tgl`,
`username`, `tgl_pelanggaran`, `lati`, `longi`,
`nama_pelanggar`, `alamat`, `no_hp`, `stnk`,
`merk`, `plat`, `warna`, `kode_kategori`,
`keterangan`, `tujuan`, `gambar`, `status`,
`id_tanggapan`)                    VALUES
('$id_lapor','$jam',CURRENT_DATE,'$user'
,'$tgl','$lati','$longi','$nama','$alamat','$no','$
stnk','$merk','$plat','$warna','$id_kate','$urai
an','$id_tjn','$url','Proses','-')");
   $upload=$gp.$url;
   if($lapor){
                    file_put_contents($upload,
base64_decode($foto));
     $respon['hasil']=1;
     $respon['pesan']='Simpan Berhasil';
     echo json_encode($respon);
   }else{
     $respon['hasil']=0;
     $respon['pesan']='Simpan Gagal';
     echo json_encode($respon);
   }
}else{
   $respon['hasil']=3;
   $respon['pesan']='Data Tidak diterima';
   echo json_encode($respon);
}
echo json_encode($respon);

?>
```

Tampil.php

```
<?php
require_once('koneksi.php');
```

```
if($con){
   $jenis=$_POST['akses'];
   if ($jenis=="kategori") {
       $sql=mysqli_query($con, "SELECT *
FROM `tb_bidang`");
      if(mysqli_num_rows($sql)>0){
         $respon["Kategori"]=array();
                                          while
($rows=mysqli_fetch_array($sql)) {
            $hasil=array();
             $hasil["nama"]=$rows["nama_bid
ang"];
               array_push($respon["Kategori"],
$hasil);
            $respon["hasil"]=1;
          }
         echo json_encode($respon);
      }else{
         $respon["hasil"]=0;
              $respon["pesan"]="Hasil Tidak
Ditemukan";
         echo json_encode($respon);
      }
   }elseif ($jenis=="tujuan"){
       $sql=mysqli_query($con, "SELECT *
FROM `tujuan_pembayaran`");
      if(mysqli_num_rows($sql)>0){
         $respon["Tujuan"]=array();
                                          while
($rows=mysqli_fetch_array($sql)) {
            $hasil=array();
            $hasil["nama"]=$rows["nama_tuju
an"];
                array_push($respon["Tujuan"],
$hasil);
            $respon["hasil"]=1;
          }
         echo json_encode($respon);
      }else{
         $respon["hasil"]=0;
              $respon["pesan"]="Hasil Tidak
Ditemukan";
         echo json_encode($respon);
      }
   }elseif ($jenis=="pasal"){
       $sql=mysqli_query($con, "SELECT *
FROM `pasal`");
      if(mysqli_num_rows($sql)>0){
         $respon["Pasal"]=array();
                                          while
($rows=mysqli_fetch_array($sql)) {
            $hasil=array();
             $hasil["nama"]=$rows["nama_pas
al"];
                array_push($respon["Pasal"],
$hasil);
            $respon["hasil"]=1;
          }
         echo json_encode($respon);
      }else{
         $respon["hasil"]=0;
              $respon["pesan"]="Hasil Tidak
Ditemukan";
         echo json_encode($respon);
      }
   }elseif ($jenis=="pengaduan") {
      $user=$_POST['user'];
       $sql=mysqli_query($con, "SELECT *,
tb_tilang.alamat as almt, tb_tilang.no_hp as
nhp,tb_tilang.keterangan as ket FROM
`tb_tilang`
              LEFt JOIN tb_tanggapan on
tb_tilang.id_tanggapan=tb_tanggapan.id_tan
ggapan
                    JOIN tb_bidang on
tb_tilang.kode_kategori=tb_bidang.kode_kat
egori
                  JOIN tb_masyarakat on
tb_tilang.username=tb_masyarakat.usernam
e
               JOIN tujuan_pembayaran on
tb_tilang.tujuan=tujuan_pembayaran.id_tuju
an WHERE tb_tilang.username='$user'");
      if(mysqli_num_rows($sql)>0){
         $respon["Pengaduan"]=array();
                                          while
($rows=mysqli_fetch_array($sql)) {
            $hasil=array();
              $hasil["tanggal"]=date('d F Y',
strtotime($rows['tgl_pelanggaran']));
            $hasil["akses"]=$rows["id_tilang"];
             $hasil["ket"]=$rows["nama_bidan
g"];
            $hasil["uraian"]=$rows["ket"];
            $hasil["status"]=$rows["status"];
            $hasil["gambar"]=$fp.$rows["gam
bar"];
             $hasil["nama"]=$rows["nama_pel
anggar"];
             $hasil["itp"]=$rows["isi_tanggapa
n"];
             $hasil["idp"]=$rows["id_tanggapa
n"];
            $hasil["user"]=$rows["nama"];
            $hasil["lat"]=$rows["lati"];
            $hasil["long"]=$rows["longi"];
            $hasil["alamat"]=$rows["almt"];
            $hasil["no"]=$rows["nhp"];
            $hasil["stnk"]=$rows["stnk"];
```

```
            $hasil["merk"]=$rows["merk"];
            $hasil["plat"]=$rows["plat"];
            $hasil["warna"]=$rows["warna"];
                $hasil["jadwal"]=date('d F Y',
strtotime($rows['jadwal']));
            $hasil["lokasi"]=$rows["lokasi"];
             $hasil["tujuan"]=$rows["nama_tuj
uan"];
            array_push($respon["Pengaduan"],
$hasil);
            $respon["hasil"]=1;
          }
          echo json_encode($respon);
       }else{
          $respon["hasil"]=0;
                $respon["pesan"]="Hasil  Tidak
Ditemukan";
          echo json_encode($respon);
       }
    }elseif ($jenis=="data_pasal") {
       $user=$_POST['user'];
         $sql=mysqli_query($con,  "SELECT
*FROM `tb_pelanggaran`
                          JOIN     pasal     on
tb_pelanggaran.id_pasal=pasal.id_pasal
WHERE id_laporan='$user'");
       if(mysqli_num_rows($sql)>0){
          $respon["list"]=array();
                                          while
($rows=mysqli_fetch_array($sql))  {
             $hasil=array();
            $hasil["ket"]=$rows["keterangan"];
             $hasil["nama"]=$rows["nama_pas
al"];
            array_push($respon["list"], $hasil);
            $respon["hasil"]=1;
          }
          echo json_encode($respon);
       }else{
          $respon["hasil"]=0;
                $respon["pesan"]="Hasil  Tidak
Ditemukan";
          echo json_encode($respon);
       }
    }
}else{
          echo     json_encode(array('respon'=>
'koneksi gagal'));
}
mysqli_close($con);


?>
```