

```

package com.app.maptemp;

import java.text.DecimalFormat;
import java.util.List;
import java.util.concurrent.ExecutionException;

import com.app.maptemp.R;
import com.google.android.gms.maps.*;
import com.google.android.gms.maps.GoogleMap.OnMapClickListener;
import com.google.android.gms.maps.GoogleMap.OnMapLongClickListener;
import com.google.android.gms.maps.GoogleMap.OnMarkerClickListener;
import com.google.android.gms.maps.GoogleMap.OnMyLocationButtonClickListener;
import com.google.android.gms.maps.model.*;

import android.location.Location;
import android.os.Bundle;
import android.app.Activity;
import android.content.Context;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.View;
import android.view.ViewGroup;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.PopupWindow;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity implements OnMapClickListener,
    OnMapLongClickListener, OnMyLocationButtonClickListener,
    OnMarkerClickListener {
    private TextView textKeterangan;
    private GoogleMap map;
    private Koneksi koneksi;
    private List<Lokasi> listLokasi;
    private double lat;
    private double lng;
    String[] items = new String[] { "Big City", "Bustour", "Flag-Export",
        "Hut", "Information", "Landmark", "Marina", "Mosque", "Pin-
export",
        "Small City", "Star", "You Are Here" };
    int[] icons = new int[] {R.drawable.bigcity, R.drawable.bustour,
R.drawable.flag_export,
        R.drawable.hut, R.drawable.information, R.drawable.Landmark,
R.drawable.marina_2, R.drawable.mosquee, R.drawable.pin_export,
        R.drawable.smallcity, R.drawable.star_3,
R.drawable.you_are_here_2};
    // static final HttpTransport HTTP_TRANSPORT =
    // AndroidHttp.newCompatibleTransport();
    // static final JsonFactory JSON_FACTORY = new JacksonFactory();

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    map = ((MapFragment) getSupportFragmentManager().findFragmentById(R.id.map))
        .getMap();
    textKeterangan = (TextView) findViewById(R.id.text_keterangan);

    // mengambil semua lokasi pada database dan menampilkan sebagai marker
    // di map
    koneksi = new Koneksi(this);
    listLokasi = koneksi.getAllLokasi();

    for (Lokasi lokasi : listLokasi) {
        LatLng posisi = new LatLng(lokasi.getLat(), lokasi.getLng());
        int icon = getIcon(lokasi.getIcon());
        map.addMarker(new MarkerOptions()
            .title(lokasi.getNama())
            .snippet(lokasi.getDeskripsi())
            .position(posisi)
            .icon(BitmapDescriptorFactory
                .fromResource(icon)));

        if (lokasi.getKode() == 1) {
            map.moveCamera(CameraUpdateFactory.newLatLngZoom(posisi,
13));

                lat = lokasi.getLat();
                lng = lokasi.getLng();

                prosesCuaca(lat, lng);
            }
        }
        map.setMyLocationEnabled(true);
        map.setOnMapClickListener(this);
        map.setOnMapLongClickListener(this);
        map.setOnMyLocationButtonClickListener(this);
        map.setOnMarkerClickListener(this);
    }

    private int getIcon(String icon) {
        int result = 0;
        for (int i = 0; i < items.length; i++) {
            if (items[i].equals(icon)){
                result = icons[i];
            }
        }
        if (result==0) result = R.drawable.map_icon_1;
        return result;
    }

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

```

```

        return true;
    }

    @Override
    public void onMapLongClick(LatLng point) {
        try {
            Toast.makeText(this, "Tambah Lokasi Baru: " + point.toString(),
                Toast.LENGTH_SHORT).show();
            // textKeterangan.setText("Lokasi: " + point.toString());
            map.addMarker(new MarkerOptions().position(point).title(
                point.toString()));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onMapClick(LatLng point) {
        try {
            Toast.makeText(this, "Menuju Lokasi: " + point.toString(),
                Toast.LENGTH_SHORT).show();
            textKeterangan.setText("Lokasi: " + point.toString());
            map.animateCamera(CameraUpdateFactory.newLatLngZoom(point, 14));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    public boolean onMyLocationButtonClick() {
        Location location = map.getMyLocation();

        if (location != null) {
            Toast.makeText(this, "Go to my location", Toast.LENGTH_SHORT)
                .show();
            LatLng point = new LatLng(location.getLatitude(),
                location.getLongitude());
            map.animateCamera(CameraUpdateFactory.newLatLngZoom(point, 14));
            popUpWindow(this, map);
            prosesCuaca(location.getLatitude(), location.getLongitude());
        } else {
            Toast.makeText(this, "My location Failed", Toast.LENGTH_SHORT)
                .show();
        }

        return true;
    }

    // menampilkan layar popup
    private void popUpWindow(final Context context, final GoogleMap map) {
        LayoutInflater inflater = (LayoutInflater) this
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        final View pview = inflater.inflate(R.layout.pop_up, null, false);
        final PopupWindow pw = new PopupWindow(pview, 500, 500, true);
    }

```

```

        pw.showAtLocation(this.findViewById(R.id.map), Gravity.CENTER, 0, 0);
        Button buttonSave = (Button) pview.findViewById(R.id.button_save);
        Button buttonBatal = (Button) pview.findViewById(R.id.button_batal);
        final Spinner spinnerIcon = (Spinner)
pview.findViewById(R.id.spinner_icon);
        final EditText textNamaLokasi = (EditText)
pview.findViewById(R.id.text_nama_lokasi);
        final EditText textDeskripsi = (EditText)
pview.findViewById(R.id.text_deskripsi);

//        spinnerIcon.setAdapter(new MyCustomAdapter(this, R.layout.row, items));

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
                android.R.layout.simple_spinner_item, items);

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

        buttonSave.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                // mengambil dan menyimpan inputan yang dilakukan oleh
                // dalam variabel
                String namaLokasi = textNamaLokasi.getText().toString();
                String deskripsi = textDeskripsi.getText().toString();
                String icon = spinnerIcon.getSelectedItem().toString();

                if (namaLokasi.isEmpty()) { // apabila nama lokasi tidak
                    Toast.makeText(context, "Nama Lokasi Harus diisi!",
                            Toast.LENGTH_SHORT).show();
                } else {
                    Location location = map.getMyLocation();

                    // menentukan atribut yang akan disimpan di database
                    Lokasi lokasi = new Lokasi();
                    lokasi.setNama(namaLokasi);
                    lokasi.setDeskripsi(deskripsi);
                    lokasi.setLat(location.getLatitude());
                    lokasi.setLng(location.getLongitude());
                    lokasi.setIcon(icon);

                    // menambahkan data lokasi ke dalam database
                    Koneksi koneksi = new Koneksi(context);
                    boolean hasil = koneksi.insertLokasi(lokasi);

                    // apabila data berhasil disimpan ke dalam database,
                    // membuat marker baru
                    if (hasil) {
                        int iconCode = getIcon(lokasi.getIcon());
                        Toast.makeText(context,

```



```

        if (list.indexOf(string) < list.size() - 1)
            builder.append("\n");
    }
    textKeterangan.setText(builder.toString());
} catch (InterruptedException e) {
    e.printStackTrace();
} catch (ExecutionException e) {
    e.printStackTrace();
}
}

@Override
public boolean onMarkerClick(Marker marker) {
    LatLng latLng = marker.getPosition();
    Location myLocation = map.getLocation();
    double myLat = myLocation.getLatitude();
    double myLng = myLocation.getLongitude();
    LatLng myLatLng = new LatLng(myLat, myLng);
    double result = CalculationByDistance(myLatLng, latLng);

    DecimalFormat df = new DecimalFormat("###.###");
    String jarak = df.format(result)+"KM";
    if (marker.getTitle().isEmpty()){
        Toast.makeText(this, "Jarak Dari My Location Ke "+myLat+", "+myLng+"
Sebesar "+jarak, Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(this, "Jarak Dari My Location Ke "+marker.getTitle()+"
Sebesar "+jarak, Toast.LENGTH_LONG).show();
    }

    textKeterangan.setText("Jarak: "+jarak);

    return false;
}

public double CalculationByDistance(LatLng StartP, LatLng EndP) {
    int Radius=6371;//radius of earth in Km
    double lat1 = StartP.latitude;
    double lat2 = EndP.latitude;
    double lon1 = StartP.longitude;
    double lon2 = EndP.longitude;
    double dLat = Math.toRadians(lat2-lat1);
    double dLon = Math.toRadians(lon2-lon1);
    double a = Math.sin(dLat/2) * Math.sin(dLat/2) +
    Math.cos(Math.toRadians(lat1)) * Math.cos(Math.toRadians(lat2)) *
    Math.sin(dLon/2) * Math.sin(dLon/2);
    double c = 2 * Math.asin(Math.sqrt(a));
    double valueResult= Radius*c;
    double km=valueResult/1;
    DecimalFormat newFormat = new DecimalFormat("#####");
    int kmInDec = Integer.valueOf(newFormat.format(km));
    double meter=valueResult%1000;
    int meterInDec= Integer.valueOf(newFormat.format(meter));
    // Log.i("Radius Value", ""+valueResult+" KM "+kmInDec+" Meter
"+meterInDec);
}

```

```

        return Radius * c;
    }

    public class MyCustomAdapter extends ArrayAdapter<String> {
objects) {
        public MyCustomAdapter(Context context, int textViewResourceId, String[]
            super(context, textViewResourceId, objects);
            // TODO Auto-generated constructor stub
        }

        @Override
parent) {
        public View getDropDownView(int position, View convertView, ViewGroup
            // TODO Auto-generated method stub
            return super.getDropDownView(position, convertView, parent);
        }

        @Override
        public View getView(int position, View convertView, ViewGroup parent) {
            // TODO Auto-generated method stub
            return super.getView(position, convertView, parent);
        }

parent) {
        public View getCustomView(int position, View convertView, ViewGroup
            // TODO Auto-generated method stub
            // return super.getView(position, convertView, parent);

            LayoutInflater inflater = getLayoutInflater();
            View row = inflater.inflate(R.layout.row, parent, false);
            TextView label = (TextView) row.findViewById(R.id.text_list);
            label.setText(items[position]);

            ImageView icon = (ImageView) row.findViewById(R.id.image_icon);

            icon.setImageResource(icons[position]);

            return row;
        }
    }
}

package com.app.maptemp;

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

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

import android.os.AsyncTask;

```

```

public class AsyncTaskParseJson extends AsyncTask<String, String, Cuaca> {
    final String TAG = "AsyncTaskParseJson.java";

    // contacts JSONArray
    JSONArray dataJsonArr = null;

    @Override
    protected Cuaca doInBackground(String... arg0) {
        Cuaca cuaca = new Cuaca();
        String url = arg0[0];
        try {

            // instantiate our json parser
            JsonParser jParser = new JsonParser();

            // get json string from url
            JSONObject json = jParser.getJSONFromUrl(url);
            cuaca.setLokasi(json.getString("name"));

            JSONArray weather = json.getJSONArray("weather");

            List<String> list = new ArrayList<String>();
            // loop through all users
            for (int i = 0; i < weather.length(); i++) {

                JSONObject c = weather.getJSONObject(i);

                // Storing each json item in variable
                list.add(c.getString("main"));

            }

            JSONObject main = json.getJSONObject("main");
            double temperature = main.getDouble("temp");

            JSONObject wind = json.getJSONObject("wind");
            double speed = wind.getDouble("speed");

            cuaca.setListCuaca(list);
            cuaca.setTemp(temperature);
            cuaca.setSpeed(speed);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        return cuaca;
    }
}

```



```

package com.app.maptemp;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.ArrayList;
import java.util.List;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteStatement;
import android.widget.Toast;

public class Koneksi {
    private static final String DB_NAME = "db_map";
    private static final int DB_VERSION = 1;

    private final Context context;
    private DatabaseHelper helper;
    private SQLiteDatabase db;

    public Koneksi(Context context) {
        this.context = context;
        helper = new DatabaseHelper(this.context);
        try {
            copyDataBase();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private class DatabaseHelper extends SQLiteOpenHelper{

        public DatabaseHelper(Context context) {
            super(context, DB_NAME, null, DB_VERSION);
        }

        @Override
        public void onCreate(SQLiteDatabase db) {
            // db.execSQL(CREATE_ARAB);
            // createDB();
            System.out.println("oncreate");
        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
        {
            // db.execSQL("DROP TABLE IF EXISTS " + DB_TABLE);
            // onCreate(db);
        }
    }
}

```

```

public void copyDataBase() throws IOException{
    InputStream myInput = context.getAssets().open(DB_NAME);

    String DB_PATH =
context.getFilesDir().getParentFile().getPath()+"/databases/";
    File fileOutput = new File(DB_PATH);
    if (!fileOutput.exists()){
        fileOutput.mkdirs();
    }

    fileOutput = new File(DB_PATH, DB_NAME);
    if (!fileOutput.exists()){
        OutputStream myOutput = new
FileOutputStream(fileOutput.getAbsolutePath());

        byte[] buffer = new byte[1024];
        int length;
        while ((length = myInput.read(buffer))>0){
            myOutput.write(buffer, 0, length);
        }
        myOutput.flush();
        myOutput.close();
    }

    myInput.close();
}

public Koneksi open(){
    db = helper.getWritableDatabase();
    return this;
}

public void close(){
    helper.close();
}

public List<Lokasi> getAllLokasi(){
    List<Lokasi> list = new ArrayList<Lokasi>();
    try {
        String sql = "SELECT kode_lokasi, nama_lokasi, deskripsi, lat, lng, icon
FROM lokasi";

        db = helper.getWritableDatabase();
        Cursor cursor = db.rawQuery(sql, null);

        if (cursor.moveToFirst()) {
            do {
                Lokasi lokasi = new Lokasi();
                lokasi.setKode(cursor.getInt(0));
                lokasi.setNama(cursor.getString(1));
                lokasi.setDeskripsi(cursor.getString(2));
                lokasi.setLat(cursor.getDouble(3));
                lokasi.setLng(cursor.getDouble(4));
            }
        }
    }
}

```

```

        lokasi.setIcon(cursor.getString(5));
        list.add(lokasi);
    } while (cursor.moveToNext());
    }
    db.close();
} catch (Exception e) {
    Toast.makeText(context, "Error "+e.getMessage(),
Toast.LENGTH_LONG).show();
}
return list;
}

public Lokasi getLokasi(int kodeLokasi){
    Lokasi lokasi = new Lokasi();
    try {
        String sql = "SELECT kode_lokasi, nama_lokasi, deskripsi, lat, lng, icon
FROM lokasi WHERE kode_lokasi="+kodeLokasi;

        db = helper.getWritableDatabase();
        Cursor cursor = db.rawQuery(sql, null);

        if (cursor.moveToFirst()) {
            do {
                lokasi.setKode(cursor.getInt(0));
                lokasi.setNama(cursor.getString(1));
                lokasi.setDeskripsi(cursor.getString(2));
                lokasi.setLat(cursor.getDouble(3));
                lokasi.setLng(cursor.getDouble(4));
                lokasi.setIcon(cursor.getString(5));
            } while (cursor.moveToNext());
        }
        db.close();
    } catch (Exception e) {
        Toast.makeText(context, "Error "+e.getMessage(),
Toast.LENGTH_LONG).show();
    }
    return lokasi;
}

public boolean insertLokasi(Lokasi lokasi){
    boolean result = false;
    try {
        String sql = "INSERT INTO lokasi (nama_lokasi, deskripsi, lat, lng, icon)
VALUES (?, ?, ?, ?, ?)";
        db = helper.getWritableDatabase();
        SQLiteStatement stmt = db.compileStatement(sql);
        stmt.bindString(1, lokasi.getNama());
        stmt.bindString(2, lokasi.getDeskripsi());
        stmt.bindDouble(3, lokasi.getLat());
        stmt.bindDouble(4, lokasi.getLng());
        stmt.bindString(5, lokasi.getIcon());
        long insert = stmt.executeInsert();
        if (insert > 0){
            result = true;
        }
    }
}

```

```

        db.close();
    } catch (Exception e) {
        Toast.makeText(context, "Error " + e.getMessage(),
Toast.LENGTH_LONG).show();
    }
    return result;
}
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/text_list"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/tambah_lokasi" />

    <EditText
        android:id="@+id/text_nama_lokasi"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/masukkan_nama_lokasi" >

        <requestFocus />
    </EditText>

    <EditText
        android:id="@+id/text_deskripsi"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/masukkan_deskripsi" />

    <Spinner
        android:id="@+id/spinner_icon"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/button_save"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="32dp"
            android:text="@string/simpan" />

```

```

        <Button
            android:id="@+id/button_batal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_alignParentTop="true"
            android:layout_marginRight="32dp"
            android:text="@string/batal" />
    </RelativeLayout>

</LinearLayout>

package com.app.maptemp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;

import android.util.Log;

public class JsonParser {

    final String TAG = "JsonParser.java";

    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    public JSONObject getJSONFromUrl(String url) {

        // make HTTP request
        try {

            DefaultHttpClient httpClient = new DefaultHttpClient();
            HttpPost httpPost = new HttpPost(url);

            HttpResponse httpResponse = httpClient.execute(httpPost);
            HttpEntity httpEntity = httpResponse.getEntity();
            is = httpEntity.getContent();

        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}

```

```

    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {
        BufferedReader reader = new BufferedReader(new InputStreamReader(is,
"iso-8859-1"), 8);
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "n");
        }
        is.close();
        json = sb.toString();

    } catch (Exception e) {
        Log.e(TAG, "Error converting result " + e.toString());
    }

    // try parse the string to a JSON object
    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e(TAG, "Error parsing data " + e.toString());
    }

    // return JSON String
    return jsonObj;
}
}

```

```

package com.app.maptemp;

```

```

import java.util.List;

```

```

public class Cuaca {
    private String lokasi;
    private List<String> listCuaca;
    private double speed;
    private double temp;
    public List<String> getListCuaca() {
        return listCuaca;
    }
    public void setListCuaca(List<String> listCuaca) {
        this.listCuaca = listCuaca;
    }
    public double getSpeed() {
        return speed;
    }
    public void setSpeed(double speed) {

```

```

        this.speed = speed;
    }
    public double getTemp() {
        return temp;
    }
    public void setTemp(double temp) {
        this.temp = temp;
    }
    public String getLocation() {
        return lokasi;
    }
    public void setLocation(String lokasi) {
        this.lokasi = lokasi;
    }
}

```

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.app.maptemp"
    android:versionCode="1"
    android:versionName="1.0" >

    <!-- Min SDK should be min 11 -->
    <uses-sdk
        android:minSdkVersion="11"
        android:targetSdkVersion="19" />

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

    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_Logo"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.app.maptemp.SplashScreen"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.app.maptemp.MainActivity"

```

```

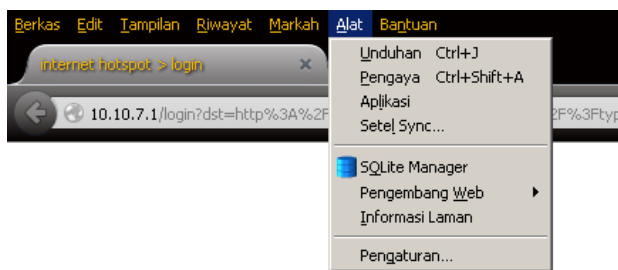
        android:label="@string/app_name" />

    <meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="AIzaSyCvF06p1C1zCCegfEGJZCDiPg255Z1ssvo" />
</application>

</manifest>

```

Gambar langkah memasukan data dari SQLite database



A screenshot of the 'Add New Record' dialog box in SQLite Manager. The 'Table Name' is set to 'lokasi'. The dialog lists six fields to be entered:

Field ID	Field Name (Type)	Value
1.	kode_lokasi (INTEGER)	NULL
2.	nama_lokasi (VARCHAR)	Empty string
3.	deskripsi (TEXT)	Null
4.	lat (DOUBLE)	Empty string
5.	lng (DOUBLE)	Empty string
6.	icon (VARCHAR)	Null

At the bottom of the dialog are 'OK' and 'Batal' buttons.

SQLite Manager - E:\ta ku\TA BARU\Workspace\maptemp\assets\db_map

Database Table Index View Trigger Tools Help

Directory (Select Profile Database) Go

db_map

Structure Browse & Search Execute SQL DB Settings

TABLE lokasi Search Show All Add Duplicate Edit Delete

kode_lokasi	nama_lokasi	deskripsi	lat	lng	icon	
2	Taman Pintar	Tentang Taman Pintar Yogy...	-7.7991276	110.3666382	Hut	
3	Tugu Jogja	Jln.Pangeran mangkubumi	-7.7829166	110.367027	Landmark	
4	Rumah Saya	Gatak,Gari,Wonosari	-7.9426	110.59898	Star	
5	Alakom	lobby kampus	-7.79272	110.40872	Information	