

## LISTING CODING

```
package com.cemerlang.halte;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
public class Utama extends Activity {
    static String pilih, st_lat,st_lon="";
    GpsService    gps;
    String ID, PS;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstancesS
tate);
        setContentView(R.layout.utama
);
        aktifGps();
        findViewById(R.id.button1).set
OnClickListener(new
android.view.View.OnClickListener() {
            @Override
            public void
onClick(View arg0) {
                Intent a=new
Intent(Utama.this,Cari_Lokasi.class);
                a.addFlags(Intent.FLAG_ACTIVITY_CLEA
R_TOP);
                startActivity(a);
            }
        });
        findViewById(R.id.button2).set
OnClickListener(new
android.view.View.OnClickListener() {
            @Override
            public void
onClick(View arg0) {
                Intent a=new
Intent(Utama.this,Lokasi_Peta.class);
                a.addFlags(Intent.FLAG_ACTIVITY_CLEA
R_TOP);
                startActivity(a);
            }
        });
        findViewById(R.id.button3).set
OnClickListener(new
android.view.View.OnClickListener() {
            @Override
            public void
onClick(View arg0) {
                Intent a=new
Intent(Utama.this,Peta.class);
                a.addFlags(Intent.FLAG_ACTIVITY_CLEA
R_TOP);
```

```

        startActivity(a);
    }
});

findViewById(R.id.button4).set
OnClickListener(new
android.view.View.OnClickListener() {
    @Override
    public void
onClick(View arg0) {
        Intent a=new
Intent(Utama.this,Petunjuk.class);
        a.addFlags(Intent.FLAG_ACTIVIT
Y_CLEAR_TOP);
        startActivity(a);
    }
});

findViewById(R.id.button5).set
OnClickListener(new
android.view.View.OnClickListener() {
    @Override
    public void
onClick(View arg0) {
        finish();
    }
});
}

public void aktifGps(){
    gps = new
GpsService(Utama.this);
    if
(gps.canGetLocation())
    {
        startActivity(a);
    }
}

double
latitude = gps.getLatitude();

double
longitude = gps.getLongitude();

st_lat="" + gps.getLatitude();

st_lon="" + gps.getLongitude();
} else
{
    gps.showSettingAlert();
}
}

}

package com.cemerlang.halte;
import android.os.Bundle;
import android.os.Handler;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;

public class Splash extends Activity {
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceS
tate);
        setContentView(R.layout.splash
);
}
}

```

```

        new
Handler().postDelayed(new Runnable(){
        @Override
        public void
run() {
                                Intent
a=new Intent(Splash.this,Utama.class);

a.addFlags(Intent.FLAG_ACTIVITY_CLEA
R_TOP);

        startActivity(a);
        }
}, 1200);
}

```

```

package com.cemerlang.halte;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class Petunjuk extends Activity {
        @Override
        protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.petunj
uk);
        }
}

```

```

package com.cemerlang.halte;
import java.io.IOException;
import java.io.InputStream;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import
org.apache.http.client.ClientProtocolExc
eption;
import
org.apache.http.client.HttpClient;
import
org.apache.http.client.methods.HttpPos
t;
import
org.apache.http.impl.client.DefaultHttp
Client;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import android.os.StrictMode;
import android.util.Log;

import
com.google.android.gms.maps.model.L
atLng;

public class MapDirection {

```

```

        public
List<List<HashMap<String,String>>>
parse(JSONObject jObject){
        List<List<HashMap<String,
String>>> routes = new
ArrayList<List<HashMap<String,String>>
>());
        JSONArray jRoutes =
null;
        JSONArray jLegs = null;
        JSONArray jSteps = null;
        try {
                jRoutes =
jObject.getJSONArray("routes");
                for(int
i=0;i<jRoutes.length();i++){
                        jLegs = (
(JSONObject)jRoutes.get(i)).getJSONArray(
"legs");
                        List
path = new ArrayList<HashMap<String,
String>>());
                        for(int
j=0;j<jLegs.length();j++){
                                jSteps = (
(JSONObject)jLegs.get(j)).getJSONArray(
"steps");
                                for(int
k=0;k<jSteps.length();k++){
                                        String
polyline = "";
                                        polyline
=
(String)((JSONObject)((JSONObject)jStep

```

```

s.get(k)).get("polyline")).get("points");
                                List<LatLng> list
= decodePoly(polyline);
                                for(int l=0;l<list.size();l++){
                                        HashMap<String, String> hm =
new HashMap<String, String>();
                                        hm.put("lat",
Double.toString(((LatLng)list.get(l)).latit
ude) );
                                        hm.put("lng",
Double.toString(((LatLng)list.get(l)).longi
tude) );
                                        path.add(hm);
                                }
                        }
                routes.add(path);
        }
    } catch (JSONException
{
        e.printStackTrace();
    } catch (Exception e){
    }
        return routes;
    }
    private List<LatLng>
decodePoly(String encoded) {
        List<LatLng> poly = new
ArrayList<LatLng>();

```

```

        int index = 0, len =
encoded.length();
        int lat = 0, lng = 0;
        while (index < len) {
            int b, shift = 0, result = 0;
            do {
                b = encoded.charAt(index++) -
63;
                result |= (b & 0x1f) << shift;
                shift += 5;
            } while (b >= 0x20);
            int dlat = ((result & 1) != 0 ?
~(result >> 1) : (result >> 1));
            lat += dlat;
            shift = 0;
            result = 0;
            do {
                b = encoded.charAt(index++) -
63;
                result |= (b & 0x1f) << shift;
                shift += 5;
            } while (b >= 0x20);
            int dlng = ((result & 1) != 0 ?
~(result >> 1) : (result >> 1));
            lng += dlng;

            LatLng p = new LatLng((((double)
lat / 1E5)),
                (((double) lng / 1E5)));
            poly.add(p);
        }

        return poly;
    }

    public int DistanceInfo(LatLng ASAL,
LatLng TUJUAN) {
        StringBuilder stringBuilder = new
StringBuilder();
        int dist = 0;
        try {
            String str_origin =
"origin="+ASAL.latitude+", "+ASAL.longit
ude;
            String str_dest =
"destination="+TUJUAN.latitude+", "+TU
JUAN.longitude;
            String sensor =
"sensor=false";
            String waypoints = "";
            String parameters =
str_origin+"&"+str_dest+"&"+sensor;
            String output = "json";
            String url =
"https://maps.googleapis.com/maps/ap
i/directions/"+output+"?" +parameters;
            HttpPost httpPost =
new HttpPost(url);
            HttpClient client = new
DefaultHttpClient();
            HttpResponse response;
            stringBuilder = new StringBuilder();

```

```

        response =
client.execute(httpPost);

        HttpEntity entity =
response.getEntity();

        InputStream stream =
entity.getContent();

        int b;

        while ((b = stream.read()) != -1) {
            stringBuilder.append((char) b);
        }
    } catch (ClientProtocolException e) {
    } catch (IOException e) {
    }

    JSONObject jsonObject = new
JSONObject();

    try {

        jsonObject = new
JSONObject(stringBuilder.toString());

        JSONArray array =
jsonObject.getJSONArray("routes");

        JSONObject routes =
array.getJSONObject(0);

        JSONArray legs =
routes.getJSONArray("legs");

        JSONObject steps =
legs.getJSONObject(0);

        JSONObject distance =
steps.getJSONObject("distance");

        Log.i("Distance",
distance.toString());

```

```

        dist =
Integer.parseInt(distance.getString("val
ue"));
    } catch (JSONException e) {
        // TODO Auto-generated catch
block
        e.printStackTrace();
    }
    return dist;
}

}

package com.cemerlang.halte;

import android.os.Bundle;

import android.app.Activity;

import android.view.Menu;

public class MainActivity extends
Activity {

    @Override

    protected void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceS
tate);

        setContentView(R.layout.activit
y_main);
    }

    @Override

    public boolean
onOptionsItemSelected(Menu menu) {

        // Inflate the menu; this
adds items to the action bar if it is
present.

```

```

        getMenuInflater().inflate(R.men
u.main, menu);

        return true;
    }
}

package com.cemerlang.halte;
import java.util.ArrayList;
import java.util.HashMap;
import org.json.JSONArray;
import org.json.JSONObject;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.
Editor;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;
import android.widget.Toast;
public class Lokasi_Peta extends Activity
{
    ArrayList<HashMap<String,String>>lst;
    HashMap<String,String>map;
    SimpleAdapter Adapter;

```

```

    ListView list;
    ClassUrl ur=new ClassUrl();
    Cari_Lokasi cr=new Cari_Lokasi();
    static String nama,alamat,lat,lon;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.list_lo
kasi);
        list=(ListView)findViewById(R.id
.listView1);
        new tampil().execute();
    }

    public class tampil extends
AsyncTask<String,String,String>{
        JSONArray ja=null;
        protected void
onPreExecute(){}
        protected String
doInBackground(String... arg0){
            try{
                JSONObject
jp=new JSONObject();
                JSONObject
jo=jp.getJSONObject(ur.link()+"list_hal
te.php");
                ja=jo.getJSONArray("krim");

                lst=new
ArrayList<HashMap<String,String>>();
                for(int i=0; i<=ja.length(); i++){

```

```

        JSONObject
c=ja.getJSONObject(i);

    map=new HashMap<String,String>();
        map.put("a",
c.getString("NamaHalte"));
        map.put("b",
c.getString("Lokasi"));
        map.put("c", c.getString("Lat"));

        map.put("d",
c.getString("Lon"));

        //map.put("e",
c.getString("Deskripsi"));

        lst.add(map);
    }
} catch
(Exception e){} return null;
}

protected void
onPostExecute(String strFromDolnBg){

    Adapter =new
SimpleAdapter(Lokasi_Peta.this,lst,R.lay
out.isi_lokasi,new
String[]{"a","b","c","d"}, new int[]
{R.id.textView1,
R.id.textView2,R.id.textView3,R
.id.textView4});

    list.setAdapter(Adapter);
    list.setTextFilterEnabled(true);
if (cr.cari.toString().trim().length(>0){

    Adapter.getFilter().filter(cr.cari.
toString().toLowerCase().trim());

    }

    list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
@Override

```

```

public
void onItemClick(AdapterView<?> arg0,
View arg1,
int arg2, long arg3) {
    TextView
slat=(TextView)arg1.findViewById(R.id.t
extView3) ;

    TextView
slon=(TextView)arg1.findViewById(R.id.
textView4) ;

    TextView
snama=(TextView)arg1.findViewById(R.i
d.textView1) ;

    TextView
slokasi=(TextView)arg1.findViewById(R.i
d.textView2) ;
    nama=snama.getText().toStrin
g();
    alamat=slokasi.getText().toStrin
g();
    lat=slat.getText().toString();

    lon=slat.getText().toString();

    Intent a=new
Intent(Lokasi_Peta.this,Lokasi_Detail.cl
ass);
    a.addFlags(Intent.FLAG_ACTIVIT
Y_CLEAR_TOP);
    startActivity(a);

    //Toast.makeText(getApplicatio
nContext(), ""+
gambar.toString(),Toast.LENGTH_LONG)
.show();

}

});

}

}

```



```

}

package com.cemerlang.halte;
import java.util.ArrayList;
import java.util.HashMap;
import org.json.JSONArray;
import org.json.JSONObject;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.ListView;
import android.widget.SimpleAdapter;

public class List_Lokasi extends Activity
{
SimpleAdapter Adapter;
ArrayList<HashMap<String,String>>lst;
HashMap<String,String>map;
ListView lv;

@Override
protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.list_lo
kasi);

findViewById(R.id.sinb).setOnCli
ckListener(new
android.view.View.OnClickListener() {

@Override

```

```

public void
onClick(View arg0) {
//
TODO Auto-generated method stub
finish();
}
});
}

public class lokas extends
AsyncTask<String,String,String>{
JSONArray ja=null;
protected void onPreExecute(){}
protected String
doInBackground(String... arg0){
try{
JSONArray
jp=new JSONArray();
JSONObject
jo=jp.getJSONFromUrl("");
lst=new
ArrayList<HashMap<String,String>>();
ja=jo.getJSONArray("kirim");
for(int i=0; i<=
ja.length(); i++){
JSONObject
c=ja.getJSONObject(i);
map=new
HashMap<String,String>();
map.put("id",c.getString("id"));
}
}catch(Exception e){}
return null;
}
}

```

```

    }

    protected void
onPostExecute(String strFormInDoBg){

    lv=(ListView)findViewById(R.id.li
stView1);

        Adapter =new
SimpleAdapter(List_Lokasi.this,lst,R.layo
ut.list_lokasi_detail,new String [] {"a"},

                                new
int[] {R.id.textView1});

        lv.setAdapter(Adapter);

    }
}
}

```

```
package com.cemerlang.halte;
```

```

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import
java.io.UnsupportedEncodingException;

import java.util.List;

import org.apache.http.HttpEntity;

import org.apache.http.HttpResponse;

import org.apache.http.NameValuePair;

import
org.apache.http.client.ClientProtocolExc
eption;

```

```

import
org.apache.http.client.entity.UrlEncode
dFormEntity;

import
org.apache.http.client.methods.HttpGet
;

import
org.apache.http.client.methods.HttpPos
t;

import
org.apache.http.client.utils.URLEncoded
Utils;

import
org.apache.http.impl.client.DefaultHttp
Client;

import org.apache.http.protocol.HTTP;

import org.json.JSONException;

import org.json.JSONObject;

import android.util.Log;

public class JSONParser {

    static InputStream is = null;

    static JSONObject jsonObj = null;

    static String json = "";

    static JSONObject artikel = null;

    // constructor

    public JSONParser() {

    }

    public JSONObject AmbilJson(String
url) {

```

```

        //Memanggil file php di server
    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();
    }

    //membaca JSON yang ditampilkan
file php di server

    try {

        BufferedReader reader = new
BufferedReader(new

```

```

InputStreamReader(is,
HTTP.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("Buffer Error", "Error
converting result " + e.toString());
    }

    //memasukkan data yang di baca
dari file JSON php di server

    try {

        artikel = new JSONObject(json);
    }

    catch (JSONException e) {

        Log.e("JSON Parser", "Error
parsing data " + e.toString());
    }

    return artikel;
}

    public JSONObject
getJSONFromUrl(String url) {

        // Making HTTP request

        try {

```

```

//
defaultHttpClient
    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("Buffer
Error", "Error converting result " +
e.toString());
}
// try parse the string to
a JSON object
try {
    JSONObject(jObj = new
JSONObject(json));
} catch (JSONException
e) {
    Log.e("JSON
Parser", "Error parsing data " +
e.toString());
}
// return JSON String
return jObj;
}

```

```

        public JSONObject
makeHttpRequest(String url, String
method,
                List<NameValuePair>
params) {
                // Making HTTP request
                try {
// check for request method
if (method == "POST") { // request
method is POST
                // defaultHttpClient
                DefaultHttpClient
httpClient = new DefaultHttpClient();
                HttpPost httpPost = new
HttpPost(url);
                httpPost.setEntity(new
UrlEncodedFormEntity(params));
                HttpResponse httpResponse =
httpClient.execute(httpPost);
                HttpEntity httpEntity =
httpResponse.getEntity();
                is = httpEntity.getContent();
                } else if (method == "GET") {
                //
request method is GET
                DefaultHttpClient
httpClient = new DefaultHttpClient();
                String
paramString =
URLEncodedUtils.format(params, "utf-
8");
                url += "?" + paramString;
                HttpGet httpGet = new HttpGet(url);

```

```

                HttpResponse httpResponse =
httpClient.execute(httpGet);
                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("Buffer
Error", "Error converting result " +
e.toString());
    }

    // try parse the string to
a JSON object

    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e("JSON Parser",
"Error parsing data " + e.toString());
    }

    // return JSON String
    return jsonObj;
}

}

package com.cemerlang.halte;
/*
 * Filename   : GpsService.java
 * Projectname : GPSTutorial
 * Date      : Nov 24, 2012 , 6:35:17 PM
 *
 * @author PratamaWijaya
 *
 * class ini digunakan untuk Akses GPS
dan segala macamnya..
 *
 * class ini merupakan turunan dari class
service dan mengImplement
LocationListener
 */

import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import
android.location.LocationListener;
import
android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;

public class GpsService extends Service
implements LocationListener
{
    private final Context
_context;

    // cek apakah GPS aktif ?
    boolean
        isGPSEnable
        = false;

    // cek network aktif ?
    boolean
        isNetworkEnable
        = false;

    boolean
        canGetLocation
        = false;

    Location
        location;

```

```

double
    latitude;

double
    longitude;

// GPS akan update ketika jarak
sudah berubah lebih dari 10 meter

private static final long
MIN_JARAK_GPS_UPDATE
= 10;
// meter

// GPS akan update pada waktu
interval

private static final long
MIN_WAKTU_GPS_UPDATE
= 1000 * 60 * 1;

protected LocationManager
locManager;

public GpsService(Context
context)
{
    _context = context;
    getLocation();
}

private Location getLocation()
{
    try
    {
        locManager =
(LocationManager)
_context.getSystemService(LOCATION_
SERVICE);

        // cek GPS
status

        isGPSEnable =
locManager.isProviderEnabled(Location
Manager.GPS_PROVIDER);

        // cek status
koneksi

        isNetworkEnable =
locManager.isProviderEnabled(Location
Manager.NETWORK_PROVIDER);

        if (!isGPSEnable
&& !isNetworkEnable)
        {
            // tidak
ada koneksi ke GPS dan Jaringan
        } else
        {
            // bisa dapatkan lokasi

            canGetLocation = true;

            // cek
apakah koneksi internet bisa ?

            if (isNetworkEnable)
            {

                // ambil posisi berdasarkan
Network
                locManager.requestLocationUp
dates(LocationManager.NETWORK_PRO
VIDER, MIN_WAKTU_GPS_UPDATE,
                MIN_JARAK_GPS_UPDATE,
this);

                if (locManager
!= null)
                {

```

```

// ambil posisi terakhir user
menggunakan Network
    location =
locManager.getLastKnownLocation(Loc
ationManager.NETWORK_PROVIDER);

    // jika lokasi berhasil didapat
    if (location != null)
        {
            // ambil latitude
            latitude
= location.getLatitude();
            // ambil longitude
            longitude =
location.getLongitude();
        }
    }

    // jika gps bisa digunakan
    if
(isGPSEnable)
    {
        if (location == null)
        {
            // ambil posisi berdasar GPS
            locManager.requestLocationUp
dates(LocationManager.GPS_PROVIDER
, MIN_WAKTU_GPS_UPDATE,
MIN_JARAK_GPS_UPDATE,
this);

            if (locManager != null)
            {
                // dapatkan posisi
                terakhir user menggunakan GPS
                location =
locManager.getLastKnownLocation(Loc
ationManager.GPS_PROVIDER);
                // jika lokasi berhasil didapat
                if (location != null)
                    {
                        // ambil
latitude
                        latitude =
location.getLatitude();
                        // ambil longitude
                        longitude =
location.getLongitude();
                    }
                }
            } catch (Exception e)
            {
                e.printStackTrace();
            }
            return location;
        }
    }
    @Override
    public void
onLocationChanged(Location location)
    {
        // TODO Auto-
generated method stub
    }

```



```

        {
            if (location != null)
                latitude =
location.getLatitude();
                return latitude;
            }
        public void setLatitude(double
latitude)
        {
            this.latitude = latitude;
        }
        public double getLongitude()
        {
            if (location != null)
                longitude =
location.getLongitude();
                return longitude;
            }
        public void setLongitude(double
longitude)
        {
            this.longitude = longitude;
        }
        public boolean
canGetLocation()
        {
            return
this.canGetLocation;
        }
        public void showSettingAlert()

        @Override
        public void
onProviderDisabled(String provider)
        {
            // TODO Auto-
generated method stub
        }
        @Override
        public void
onProviderEnabled(String provider)
        {
            // TODO Auto-
generated method stub
        }
        @Override
        public void
onStatusChanged(String provider, int
status, Bundle extras)
        {
            // TODO Auto-
generated method stub
        }
        @Override
        public IBinder onBind(Intent
intent)
        {
            // TODO Auto-
generated method stub
            return null;
        }
        public double getLatitude()

```

```

        {
            AlertDialog.Builder
alertDialog = new
AlertDialog.Builder(_context);
// title Alernya
        alertDialog.setTitle("GPS
Setting");
            // pesan alert
        alertDialog.setMessage("GPS
tidak aktif. Mau masuk ke setting Menu
?");
        alertDialog.setPositiveButton("S
etting", new
DialogInterface.OnClickListener()
        {
@OVERRIDE
public void onClick(DialogInterface
dialog, int which)
        {
            Intent
intent = new
Intent(Settings.ACTION_LOCATION_SO
URCE_SETTINGS);
            _context.startActivity(intent);
        }
    });
        alertDialog.setNegativeButton("
Cancel", new
DialogInterface.OnClickListener()
        {
@OVERRIDE
        public void
onClick(DialogInterface dialog, int
which)

```

```

        {
            dialog.cancel();
        }
    });
        alertDialog.show();
    }
    public void stopUsingGPS()
    {
        if (locManager != null)
            locManager.removeUpdates(Gp
sService.this);
    }
}
package om.cemerlang.halte;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class ClassUrl extends Activity {
    public String link(){
        String t="http://10.0.2.2/halte/";
        return t;
    }
}
package com.cemerlang.halte;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;

```

```

import android.widget.EditText;

public class Cari_Lokasi extends Activity
{
    EditText edcari;
    static String cari="";

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstancesS
tate);
        setContentView(R.layout.cari_p
eta);
        edcari=(EditText)findViewById(
R.id.edcari);
        findViewById(R.id.sin).setOnClic
kListener(new
android.view.View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                cari=edcari.getText().toString().trim();

                Intent a=new
                Intent(Cari_Lokasi.this,Lokasi_Peta.class
                );
                a.addFlags(Intent.FLAG_ACTIVITY_CLEA
                R_TOP);

                startActivity(a);
            }
        });
    }
}

```