

LISTING PROGRAM

1. com.mylbsisp
 - a. CariActivity.java

```
package com.mylbsisp;

import com.mylbsisp.R.id;

import android.os.Bundle;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;

import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class CariActivity extends Activity {
    public static String ipAddress = "anam.byethost13.com";
    public static String serverURL = "http://" + ipAddress + "/";
    ProgressDialog pd;
    EditText acari;

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

        acari = (EditText) findViewById(R.id.txtcari);

        Button btnCari = (Button) findViewById(R.id.btnCari);

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

                Intent i = new
                Intent(getApplicationContext(), HasilCariActivity.class);
                //String Cari =
                ((EditText) view).findViewById(R.id.cari).getText().toString();
                String Cari = acari.getText().toString();
                i.putExtra("hasil_cari", Cari);
                //HasilCariActivity aCariAktiviti =

                CariActivity.this;

                startActivity(i);
            }
        });
    }
}
```

```

    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
    it is present.
    getMenuInflater().inflate(R.menu.cari, menu);
    return true;
}
}

```

b. DashboardActivity.java

```

package com.mylbsisp;

import com.mylbsisp.library.Karyawan;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class DashboardActivity extends Activity {
    public static Karyawan aKaryawan=null;

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

        setContentView(R.layout.dashboard);

        Intent intent = getIntent();
        // aKaryawan =
        (Karyawan)intent.getSerializableExtra("Karyawan");

        Button btn_jenis = (Button) findViewById(R.id.btn_jenis);
        Button btn_pelanggan = (Button)
        findViewById(R.id.btn_pelanggan);
        Button btn_cari = (Button) findViewById(R.id.btn_cari);
        Button btn_tentang = (Button)
        findViewById(R.id.btn_tentang);

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

```

```

        Intent i = new
Intent(getApplicationContext(), JenisAktiviti.class);
        Bundle b = new Bundle();
        i.putExtras(b);
        startActivity(i);
    }
});
btn_pelanggan.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {

        Intent i = new
Intent(getApplicationContext(), PelangganAktiviti.class);
        Bundle b = new Bundle();
        i.putExtras(b);
        startActivity(i);
    }
});
btn_cari.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent i = new
Intent(getApplicationContext(), CariAktiviti.class);
        Bundle b = new Bundle();
        i.putExtras(b);
        startActivity(i);
    }
});
btn_tentang.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent i = new
Intent(getApplicationContext(), TentangAktiviti.class);
        Bundle b = new Bundle();
        i.putExtras(b);
        startActivity(i);
    }
});
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
it is present.
    getMenuInflater().inflate(R.menu.dashboard, menu);
    return true;
}
}
}

```

c. HasilActivity.java

```

package com.mylbsisp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.params.ConnManagerParams;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.mylbsisp.library.Cari;
import com.mylbsisp.library.Jenis;
import com.mylbsisp.library.Pelanggan;

import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class HasilCariActivity extends Activity {
    public static String ipAddress = "anam.byethost13.com";
    public static String serverURL = "http://"+ipAddress+"/";
    public static Pelanggan aPelanggan = null;
    protected static CariActivity aCariAktiviti;
    ProgressDialog pd;
    private ListView lv;

```

```

    private CustomAdapter adapter;
    public String hasil_cari;
    ArrayList<Pelanggan> listPelanggan= new ArrayList<Pelanggan>();

    @Override
    public void onBackPressed() {
        // TODO Auto-generated method stub
        super.onBackPressed();
    }

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

        Bundle b = getIntent().getExtras();
        hasil_cari = b.getString("hasil_cari");

        aPelanggan = new Pelanggan();
        showCari();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if
        it is present.
        getMenuInflater().inflate(R.menu.hasil_cari, menu);
        return true;
    }

    private void showCari(){
        String urlString= serverURL+"cari.php";
        ConnectivityManager connMgr = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
        if (networkInfo != null && networkInfo.isConnected()) {
            DownloadDataCari dt = new DownloadDataCari();
            dt.execute(urlString);
            pd = ProgressDialog.show(this, "Loading", "Data sedang di
            proses...\nTunggu sebentar...", true);
            //pb.setVisibility(ProgressBar.VISIBLE);
        } else {
            Toast.makeText(this.getContext(), "Gagal Cari!!
            karna tidak ada koneksi.", Toast.LENGTH_LONG).show();
        }
    }

    public void TampilkanHasilCari (Cari aCari){
        if (aCari!=null){

        }else{

```

```

        Toast.makeText(this, getContext(), "Tidak
ada data", Toast.LENGTH_LONG).show();
    }
}

public void setAdapter() {
    lv = (ListView) findViewById(R.id.listView1);
    adapter = new CustomAdapter(HasilCariActivity.this,
        R.id.listView1,
        listPelanggan);
    lv.setAdapter(adapter);
    lv.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView? arg0, View arg1, int
position,
            long arg3){
            Object o = lv.getItemAtPosition(position);
            // Here i want have values of the clicked row, - like
the name and/or id... - but i get the following:
            // Output o.toString():
com.example.customlistview.MainActivity$Person@41252310
            Log.i("TEST", o.toString());

            aPelanggan = (Pelanggan) o;
            Toast.makeText(getContext(), "Sa
"+aPelanggan.toString(), Toast.LENGTH_LONG).show();
            String urlString = serverURL+"carijenis.php";
            Intent i = new
Intent(getContext(), TesActivity.class);
            TesActivity.jnsmap = TesActivity.cari;
            Bundle b = new Bundle();
            i.putExtras(b);
            startActivity(i);
        }
    });
}
}

```

```

private class DownloadDataCari extends AsyncTask<String, String,
String> {
    final int HTTP_TIMEOUT = 3 * 1000;
    HttpClient mHttpClient;
    private boolean adaDataPelanggan = false;
    @Override
    protected String doInBackground(String... params) {
        String[] urls;
        urls = params;
    }
}

```

```

        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may
be invalid.";
        }
    }

    private HttpClient getHttpClient() {
        if (mHttpClient == null) {
            mHttpClient = new DefaultHttpClient();
            final HttpParams params = mHttpClient.getParams();
            HttpConnectionParams.setConnectionTimeout(params,
HTTP_TIMEOUT);
            HttpConnectionParams.setSoTimeout(params,
HTTP_TIMEOUT);
            ConnManagerParams.setTimeout(params, HTTP_TIMEOUT);
        }
        return mHttpClient;
    }

    private String downloadUrl(String myurl) throws IOException
{
        BufferedReader in = null;
        try {
            HttpClient client = getHttpClient();
            HttpPost request = new HttpPost(myurl);
            String hcari = hasil_cari;

            //String cari = "amira";
            ArrayList<NameValuePair> postParameters = new
ArrayList<NameValuePair>();
            if (hcari.equals(""))
            {
                String cari = "ZZZZZZZZZZZZZZZZZZZZ";
                postParameters.add(new BasicNameValuePair("nama",
cari));
            }
            else{
                String cari = hasil_cari;
                postParameters.add(new
BasicNameValuePair("nama", cari));
            }
            UriEncodedFormEntity formEntity = new
UriEncodedFormEntity(postParameters, "utf-8");
            request.setEntity(formEntity);
            HttpResponse response = client.execute(request);
            Log.d("executeHttpPost ", response.toString());
            in = new BufferedReader(new
InputStreamReader(response.getEntity().getContent()));

```

```

        StringBuffer sb = new StringBuffer("");
        String line = "";
        String NL = System.getProperty("line.separator");
        while ((line = in.readLine()) != null) {
            sb.append(line + NL);
        }
        in.close();

        String result = sb.toString();

        Log.d("hasil", result);
        return result;
    } catch (Exception e) {
        return null;
    }
}

@Override
protected void onPostExecute(String result) {
    JSONArray jsonArray;
    Pelanggan aPel = null;
    try {
        jsonArray = new JSONArray(result);
        if (jsonArray.length() == 0) {
            adaDataPelanggan = false;
        } else {
            adaDataPelanggan = true;

            for (int i = 0; i < jsonArray.length(); i++)
            {
                JSONObject jobject =
                jsonArray.getJSONObject(i);

                String aldpelanggan =
                jobject.getString("id_pelanggan");
                String aldjenisJB =
                jobject.getString("id_jenisbrlgn");
                String aNama =
                jobject.getString("nama_pelanggan");
                String aAlamat =
                jobject.getString("alamat_pelanggan");
                String aNotelp =
                jobject.getString("notelp_pelanggan");
                String aEmail =
                jobject.getString("email_pelanggan");
                String aLatitude =
                jobject.getString("latitude");
                String aLongitude =
                jobject.getString("longitude");
                String aBTS = jobject.getString("bts");
                String aStatus =
                jobject.getString("status_pelanggan");
            }
        }
    }
}

```

```

        aPel = new Pelanggan();
        aPel.setIdPelanggan(aIdPelanggan);
        aPel.setIdJenisJB(aIdJenisJB);
        aPel.setName(aNama);
        aPel.setAlamat(aAlamat);
        aPel.setNotelp(aNotelp);
        aPel.setEmail(aEmail);
        aPel.setLatitude(aLatitude);
        aPel.setLongitude(aLongitude);
        aPel.setBTS(aBTS);
        aPel.setStatus(aStatus);

        listPelanggan.add(aPel);

        Log.d("cek ", aPel.toString());
    }}

} catch (JSONException e) {
    Log.d("Error login - json", e.getMessage());
} finally{
    //TampilkanHasilCari (aCari);
    if (adaDataPelanggan) {
        setAdapter();
    }else{
        onBackPressed();

        Toast.makeText(HasilCariActivity.this, getContext(),
        "Maaf data yang anda cari tidak ada dalam database.",
        Toast.LENGTH_LONG).show();
    }

    pd.dismiss();
    // pb.setVisibility(ProgressBar.INVISIBLE);
}
}

}

public class CustomAdapter extends ArrayAdapter<Pelanggan>{
    private ArrayList<Pelanggan> entries;
    private Activity activity;
    public CustomAdapter(Activity aaq, int textViewResourceId,
    ArrayList<Pelanggan> entries) {
        super(aaq, textViewResourceId, entries);
        this.entries = entries;
        this.activity = aaq;
    }
}

```



```

import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class JenisActivity extends Activity {
    public static String ipAddress = "anam.byethost13.com";
    public static String serverURL = "http://"+ipAddress+"/";
    public static Jenis aJenis = null;
    ProgressDialog pd;
    private ListView lv;
    private CustomAdapter adapter;

    public String hasil_cari;

    ArrayList<Jenis> listJenis= new ArrayList<Jenis>();
    public static ArrayList<Pelanggan> listPelanggan= new
ArrayList<Pelanggan>();
    public String DaftarJenis;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.jenis);
        aJenis = new Jenis();
        showJenis();
    }

    private void showJenis(){
        String urlString = serverURL+"jenis.php";
        ConnectivityManager connMgr = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
        if (networkInfo != null && networkInfo.isConnected()) {
            DownloadDataJenis dt = new DownloadDataJenis();
            dt.execute(urlString);
            pd = ProgressDialog.show(this, "Loading", "Data sedang di
proses...\nTunggu sebentar...", true);
            //pb.setVisibility(ProgressBar.VISIBLE);
        } else {
            Toast.makeText(this.getApplicationContext(), "Gagal
Tampil!! karna tidak ada koneksi.", Toast.LENGTH_LONG).show();
        }
    }

    @Override

```

```

    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if
        it is present.
        getMenuInflater().inflate(R.menu.jenis, menu);
        return true;
    }

    public void TampilkanJenis(Jenis aJenis){
        if (aJenis!=null){

        }else{
            Toast.makeText(this.getApplicati onContext(), "Ti dak
            ada data", Toast.LENGTH_LONG).show();
        }
    }

    public void setAdapter() {
        lv = (ListVi ew) findVi ewByI d(R.i d. l i s t Vi ew 1);
        adapter = new CustomAdapter(Jeni sActi vi ty. thi s,
            R.i d. l i s t Vi ew 1,
            l i s tJeni s);
        lv.setAdapter(adapter);
        lv.setOnl temCl i ckLi stener(new
        AdapterVi ew.Onl temCl i ckLi stener() {
            @Ovverri de
            public void onl temCl i ck(AdapterVi ew<?> arg0, Vi ew arg1, int
            posi ti on,
                long arg3) {
                Object o = lv.getl temAtPosi ti on(posi ti on);
                // Here i want have values of the clicked row, - like
                the name and/or i d... , - but i get the following:
                // Output o.toString():
                com.examp l e. customl i stvi ew. Mai nActi vi ty$Person@41252310
                Log.i("TEST", o.toString());

                aJeni s = (Jeni s) o;
                Toast.makeText(getApplicati onContext(), "Sa
                "+aJeni s.toString(), Toast.LENGTH_LONG).show();
                String urlString = serverURL+"cari jeni s. php";
                Downl oadDataCari JB dt = new Downl oadDataCari JB();
                dt.execute(stringUrl);

                // asytask pelanggan pake kuery berdasarkan idjenis
                // field pub static list pelanggan
            }
        });
    }
}

```

```

    private class DownloadDataJenis extends AsyncTask<String, String,
String> {

        final int HTTP_TIMEOUT = 3 * 1000;
        HttpClient mHttpClient;
        @Override
        protected String doInBackground(String... params) {
            String[] urls;
            urls = params;
            try {
                return downloadUrl(urls[0]);

            } catch (IOException e) {
                return "Unable to retrieve web page. URL may
be invalid.";
            }
        }

        private HttpClient getHttpClient() {
            if (mHttpClient == null) {
                mHttpClient = new DefaultHttpClient();
                final HttpParams params = mHttpClient.getParams();
                HttpConnectionParams.setConnecti onTimeout(params,
HTTP_TIMEOUT);
                HttpConnectionParams.setSoTimeout(params,
HTTP_TIMEOUT);
                ConnManagerParams.setTimeout(params, HTTP_TIMEOUT);
            }
            return mHttpClient;
        }

        private String downloadUrl (String myurl) throws IOException
{
            BufferedReader in = null;
            try {
                HttpClient client = getHttpClient();
                HttpPost request = new HttpPost(myurl);
                ArrayList<NameValuePair> postParameters = new
ArrayList<NameValuePair>();
                URLEncodedFormEntity formEntity = new
URLEncodedFormEntity(postParameters, "utf-8");
                request.setEntity(formEntity);
                HttpResponse response = client.execute(request);
                Log.d("executeHttpPost ", response.toString());
                in = new BufferedReader(new
InputStreamReader(response.getEntity().getContent()));
                StringBuffer sb = new StringBuffer("");
                String line = "";
                String NL = System.getProperty("line.separator");
                while ((line = in.readLine()) != null) {

```

```

        sb.append(line + NL);
    }
    in.close();

    String result = sb.toString();

    Log.d("hasil", result);
    return result;
    } catch (Exception e) {
        return null;
    }
}

```

```

@Override
protected void onPostExecute(String result) {
    JSONArray jsonArray;
    Jenis aJenis = null;
    try {
        jsonArray = new JSONArray(result);
        for (int i = 0; i < jsonArray.length(); i++)
        {
            JSONObject jobject =
            jsonArray.getJSONObject(i);

            String aldJB =
            jobject.getString("id_jenisbrlggn");
            String aNamaJB =
            jobject.getString("nama_jenisbrlggn");
            String aNilai =
            jobject.getString("nilai_jenisbrlggn");

            aJenis = new Jenis();
            aJenis.setIdJB(aldJB);
            aJenis.setNamaJB(aNamaJB);
            aJenis.setNilai(aNilai);
            listJenis.add(aJenis);

            Log.d("cek ", aJenis.toString());
        }
        //statusLogin(aMahasiswa);
    } catch (JSONException e) {
        Log.d("Error login - json", e.getMessage());
    } finally{
        TampilkanJenis(aJenis);
        setAdapter();
        pd.dismiss();
        // pb.setVisibility(ProgressBar.INVISIBLE);
    }
}

```

```

    }

    public class CustomAdapter extends ArrayAdapter<Jenis>{
        private ArrayList<Jenis> entries;
        private Activity activity;
        public CustomAdapter(Activity aaq, int textViewResourceId,
ArrayList<Jenis> entries) {
            super(aaq, textViewResourceId, entries);
            this.entries = entries;
            this.activity = aaq;
        }

        public View getView(int position, View convertView, ViewGroup
parent) {
            View v = convertView;
            TextView txtNama;

            LayoutInflater vi =(LayoutInflater)
activity.getSystemService(Context. LAYOUT_INFLATER_SERVICE);
            v = vi.inflate(R.layout. list_jenis_layout, parent, false);
            txtNama= (TextView) v.findViewById(R.id. listNama);
            final Jenis jns = entries.get(position);
            if (jns!=null) {
                txtNama.setText(jns.getNamaJB());
            }
            return v;
        }
    }

    private class DownloadDataCari JB extends AsyncTask<String,
String, String> {

        final int HTTP_TIMEOUT = 3 * 1000;
        HttpClient mHttpClient;
        @Override
        protected String doInBackground(String... params) {
            String[] urls;
            urls = params;
            try {
                return downloadUrl(urls[0]);
            } catch (IOException e) {
                return "Unable to retrieve web page. URL may
be invalid.";
            }
        }
    }

```

```

    }

    private HttpClient getHttpClient() {
        if (mHttpClient == null) {
            mHttpClient = new DefaultHttpClient();
            final HttpParams params = mHttpClient.getParams();
            HttpConnectionParams.setConnecti onTimeout(params,
HTTP_TIMEOUT);
            HttpConnectionParams.setSoTimeout(params,
HTTP_TIMEOUT);
            ConnManagerParams.setTimeout(params, HTTP_TIMEOUT);
        }
        return mHttpClient;
    }

    private String downloadUrl (String myurl) throws IOException
{
        BufferedReader in = null;
        try {
            HttpClient client = getHttpClient();
            HttpPost request = new HttpPost(myurl);
            String id = aJenis.getIdJB();
            ArrayList<NameValuePair> postParameters = new
ArrayList<NameValuePair>();
            postParameters.add(new BasicNameValuePair("jenis",
id));
            UriEncodedFormEntity formEntity = new
UriEncodedFormEntity(postParameters, "utf-8");
            request.setEntity(formEntity);
            HttpResponse response = client.execute(request);
            Log.d("executeHttpPost ", response.toString());
            in = new BufferedReader(new
InputStreamReader(response.getEntity().getContent()));
            StringBuffer sb = new StringBuffer("");
            String line = "";
            String NL = System.getProperty("line.separator");
            while ((line = in.readLine()) != null) {
                sb.append(line + NL);
            }
            in.close();

            String result = sb.toString();

            Log.d("hasil j b p", result);
            return result;
        } catch (Exception e) {
            return null;
        }
    }
}

```

@Override

e. LoginActivity.java

```
package com.mylbsisp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.params.ConnManagerParams;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.google.android.gms.internal.ak;
import com.mylbsisp.library.Karyawan;

import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.view.Window;
import android.widget.EditText;
import android.widget.Toast;

public class LoginActivity extends Activity {
    public static String ipAddress = "anam.byethost13.com";
    public static String serverURL = "http://" + ipAddress + "/";

    EditText un, pw;

    ProgressDialog pd;

    private static Karyawan akaryawan;

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

```

        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.login);

        un = (EditText)findViewById(R.id.username);
        pw = (EditText)findViewById(R.id.password);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if
        it is present.
        getMenuInflater().inflate(R.menu.login, menu);
        return true;
    }

    public void klikLogin(View view){
        String urlString = serverURL+"login.php";
        ConnectivityManager connMgr = (ConnectivityManager)
            getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
        if (networkInfo != null && networkInfo.isConnected()){
            DownloadDataLogin dt = new DownloadDataLogin();
            dt.execute(urlString);
            pd = ProgressDialog.show(this, "Login", "Sedang di
            proses...\nTunggu sebentar...", true);
            //pb.setVisibility(ProgressBar.VISIBLE);
        }
    }

    public void statusLogin(boolean alogin) {
        if (alogin) {
            Toast.makeText(this.getContext(),
            "Selamat datang "+aKaryawan.getNama()+"\nTunggu sebentar..",
            Toast.LENGTH_LONG).show();
            Intent s = new Intent (this,
            DashboardActivity.class);
            DashboardActivity.aKaryawan = this.aKaryawan;
            startActivity(s);
            finish();
        }else{
            Toast.makeText(this.getContext(), "Gagal
            Login!! Kesalahan pada Username dan Password.",
            Toast.LENGTH_LONG).show();
        }
    }

    private class DownloadDataLogin extends AsyncTask<String, String,
    String>{

        final int HTTP_TIMEOUT = 30 * 1000;
        HttpClient mHttpClient;
        @Override
        protected String doInBackground(String... params) {
            String[] urls;

```

```

        urls = params;
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may
be invalid.";
        }
    }

    private String downloadUrl(String myurl) throws IOException
{
    BufferedReader in = null;
    try {
        HttpClient client = getHttpClient();
        HttpPost request = new HttpPost(myurl);
        String uname = un.getText().toString();
        String passwd = pw.getText().toString();
        ArrayList<NameValuePair> postParameters = new
ArrayList<NameValuePair>();
        postParameters.add(new BasicNameValuePair("username",
uname));
        postParameters.add(new BasicNameValuePair("password",
passwd));
        UrlEncodedFormEntity formEntity = new
UrlEncodedFormEntity(postParameters, "utf-8");
        request.setEntity(formEntity);
        HttpResponse response = client.execute(request);
        Log.d("executeHttpPost ", response.toString());
        in = new BufferedReader(new
InputStreamReader(response.getEntity().getContent()));
        StringBuffer sb = new StringBuffer("");
        String line = "";
        String NL = System.getProperty("line.separator");
        while ((line = in.readLine()) != null) {
            sb.append(line + NL);
        }
        in.close();

        String result = sb.toString();

        Log.d("hasil ", result);
        return result;
    } catch (Exception e) {
        return null;
    }
}

private HttpClient getHttpClient() {
    if (mHttpClient == null) {
        mHttpClient = new DefaultHttpClient();
        final HttpParams params = mHttpClient.getParams();
        HttpConnectionParams.setConnectionTimeout(params,
HTTP_TIMEOUT);
    }
}

```

```

        HttpConnecti onParams. setSoTi meout(params,
HTTP_TI MEOUT);
        ConnManagerParams. setTi meout(params, HTTP_TI MEOUT);
    }
    return mHttpCl ient;
}

@Override
protected void onPostExecute(String result) {
    JSONArray jsonArray;
    aKaryawan = new Karyawan();
    boolean login = false;
    try {
        jsonArray = new JSONArray(result);
        for (int i = 0; i < jsonArray.length(); i++)
        {
            JSONObject jObject =
j sonArray. getJSONObje ct(i);
            String al dKaryawan=
j Object. getStri ng("i d_karyawan");
            String aUsername=
j Object. getStri ng("username_karyawan");
            String aNama =
j Object. getStri ng("nama_karyawan");

            if (aUsername. equals(un. getText(). toString())) {
                aKaryawan. setI dKaryawan(al dKaryawan);
                aKaryawan. setUsername(aUsername);
                aKaryawan. setNama(aNama);

                Log. d("cek
", aKaryawan. toString());

                login = true;

                //break;
            }else{
                login = false;

                //aKaryawan =null;
            }
        }
    } catch (JSONException e) {
        Log. d("Error login - json", e. getMessage());
    }finally{
        statusLogi n(login);
        pd. di smi ss();
        // pb. setVi si bi li ty(ProgressBar. INVI SI BLE);
    }
}

```

```

    }
}
}

```

f. PelangganActivity.java

```

package com.mylbsi.sp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.params.ConnManagerParams;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.mylbsi.sp.library.Pelanggan;

import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

public class PelangganActivity extends Activity {
    public static String ipAddress = "anam.byethost13.com";
    public static String serverURL = "http://"+ipAddress+"/";
    public static Pelanggan aPelanggan= null;
    ProgressDialog pd;
    private ListView lv;

```

```

private CustomAdapter adapter;
ArrayList<Pelanggan> listPelanggan= new ArrayList<Pelanggan>();
public String DaftarPelanggan;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.pelanggan);
    aPelanggan = new Pelanggan();
    showPelanggan();
}
private void showPelanggan(){
    String urlString= serverURL+"pelanggan.php";
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected()) {
        DownloadDataPelanggan dt = new DownloadDataPelanggan();
        dt.execute(urlString);
        pd = ProgressDialog.show(this, "Loading", "Data sedang di
proses...\nTunggu sebentar...", true);
        //pb.setVisibility(ProgressBar.VISIBLE);
    } else {
        Toast.makeText(this.getApplicationContext(), "Gagal
Tampil!! karna tidak ada koneksi.", Toast.LENGTH_LONG).show();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
it is present.
    getMenuInflater().inflate(R.menu.pelanggan, menu);
    return true;
}

public void TampilkanPelanggan(Pelanggan aPelanggan){
    if (aPelanggan!=null){

    }else{
        Toast.makeText(this.getApplicationContext(), "Tidak
ada data", Toast.LENGTH_LONG).show();
    }
}

public void setAdapter() {
    lv = (ListView) findViewById(R.id.listView1);
    adapter = new CustomAdapter(PelangganActivity.this,
R.id.listView1,
listPelanggan);
}

```

```

        lv.setAdapter(adapter);
        lv.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int
position,
        long arg3) {
        Object o = lv.getItemAtPosition(position);
        // Here i want have values of the clicked row, - like
the name and/or id... - but i get the following:
        // Output o.toString():
com.example.customlistview.MainActivity$Person@41252310
        Log.i("TEST", o.toString());

        aPelanggan = (Pelanggan) o;
        Toast.makeText(getApplicationContext(), "Sa
"+aPelanggan.toString(), Toast.LENGTH_LONG).show();
        Intent i = new
Intent(getApplicationContext(), TesAktiviti.class);
        TesAktiviti.jnsmap = TesAktiviti.pelanggan;
        Bundle b = new Bundle();
        i.putExtras(b);
        startActivity(i);
    }
});
}

```

```

private class DownloadDataPelanggan extends AsyncTask<String,
String, String> {

    final int HTTP_TIMEOUT = 3 * 1000;
    HttpClient mHttpClient;
    @Override
    protected String doInBackground(String... params) {
        String[] urls;
        urls = params;
        try {
            return downloadUrl(urls[0]);

        } catch (IOException e) {
            return "Unable to retrieve web page. URL may
be invalid.";
        }
    }
}

```

```

private HttpClient getHttpClient() {
    if (mHttpClient == null) {
        mHttpClient = new DefaultHttpClient();
        final HttpParams params = mHttpClient.getParams();
        HttpConnectionParams.setConnectionTimeout(params,
HTTP_TIMEOUT);
    }
}

```



```

        String aNama =
j Object.getString("nama_pelanggan");
        String aAlamat =
j Object.getString("alamat_pelanggan");
        String aNotelp =
j Object.getString("notelp_pelanggan");
        String aEmail =
j Object.getString("email_pelanggan");
        String aLatitude =
j Object.getString("latitude");
        String aLongitude =
j Object.getString("longitude");
        String aBTS = j Object.getString("bts");
        String aStatus =
j Object.getString("status_pelanggan");

        aPelanggan = new Pelanggan();
        aPelanggan.setIdPelanggan(alDpelanggan);
        aPelanggan.setIdJenisJB(alDjenisJB);
        aPelanggan.setNama(aNama);
        aPelanggan.setAlamat(aAlamat);
        aPelanggan.setNotelp(aNotelp);
        aPelanggan.setEmail(aEmail);
        aPelanggan.setLatitude(aLatitude);
        aPelanggan.setLongitude(aLongitude);
        aPelanggan.setBTS(aBTS);
        aPelanggan.setStatus(aStatus);

        listPelanggan.add(aPelanggan);

        Log.d("cek ", aPelanggan.toString());
    }

} catch (JSONException e) {
    Log.d("Error login - json", e.getMessage());
}finally{
    TampilPelanggan(aPelanggan);
    setAdapter();
    pd.dismiss();
    // pb.setVisibility(ProgressBar.INVISIBLE);
}
}

}

public class CustomAdapter extends ArrayAdapter<Pelanggan>{
    private ArrayList<Pelanggan> entri es;
    private Activity activity;
    public CustomAdapter(Activity aaq, int textViewResourceId,
ArrayList<Pelanggan> entri es) {

```

```

        super(aaq, textViewResourceld, entries);
        this.entries = entries;
        this.activity = aaq;
    }

    public View getView(int position, View convertView, ViewGroup
parent) {
        View v = convertView;
        TextView txtNama;

        LayoutInflater vi =(LayoutInflater)
activity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        v = vi.inflate(R.layout.list_pelanggan_layout,
parent, false);
        txtNama= (TextView) v.findViewById(R.id.listNama);
        final Pelanggan plg = entries.get(position);
        if (plg!=null) {
            txtNama.setText("Nama Pelanggan : "+plg.getNama()+"
| Alamat : "+plg.getAlamat());
        }
        return v;
    }
}
}
}

```

g. TentangActivity.java

```

package com.mylbsisp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class TentangActivity extends Activity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if
it is present.
        getMenuInflater().inflate(R.menu.tentang, menu);
        return true;
    }
}
}

```

h. TesActivity.java

```
package com.mylbsisp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.lang.annotation.RetentionPolicy;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.ArrayList;
import java.util.ListIterator;
import java.util.HashMap;
import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.params.ConnManagerParams;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.google.android.gms.internal.ak;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.OnInfoWindowClickListener;
import com.google.android.gms.maps.GoogleMap.OnMyLocationChangeListener;
import com.google.android.gms.maps.model.BitmapDescriptor;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.PolylineOptions;
import com.google.android.gms.maps.MapFragment;
import com.mylbsisp.library.DirectionsJSONParser;
import com.mylbsisp.library.Jenis;
import com.mylbsisp.library.Karyawan;
import com.mylbsisp.library.Pelanggan;

import android.location.Location;
import android.os.AsyncTask;
import android.os.Bundle;
```

```

import android.app.Activity;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.graphics.Color;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class TesActivity extends Activity implements
OnMyLocationChangeListener {
    private Pelanggan aPelanggan = null;
    public static int jnsmap =0;
    public static int pelanggan =1;
    public static int cari =2;
    public static int jenis =3;
    private ArrayList<Pelanggan> listPelanggan;
    private ArrayList<Pelanggan> listPelangganMarker;
    ProgressDialog pd;
    //private Jenis aJenis = null;
    private GoogleMap map;
    private Double myLatitude ;
    private Double myLongitude;
    private LatLng myLatLng;
    protected Location tujuan=null;
    private Location myLocation;
    protected Float jarak;
    public static String ipAddress = "anam.byethost13.com";
    public static String serverURL = "http://"+ipAddress+"/";
    Karyawan aKaryawan;
    private int intervalUpdate = 0;
    private int intervalUpdate2 = 0;

    //static final LatLng kantor = new LatLng(-7.752496, 110.382358);
    //static final LatLng rumah = new LatLng(-7.788591, 110.373673);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tes);
        pd = ProgressDialog.show(this, "Loading", "Data sedang di
proses...\nTunggu sebentar...", true);
        if (jnsmap==jenis) {
            listPelanggan = new ArrayList<Pelanggan>();
            listPelanggan = JenisActivity.listPelanggan;
            Toast.makeText(getApplicationContext(), "Sa
"+listPelanggan.size()+"\n jns="+jnsmap, Toast.LENGTH_LONG).show();
            showMap(jnsmap);
        }else if (jnsmap==pelanggan) {

```

```

        aPel anggan = new Pel anggan();
        aPel anggan = Pel angganActi vi ty. aPel anggan;
        Toast. makeText(getAppl i cati onContext(), "Sa
"+aPel anggan. toString() +"\n j ns="+j nsm ap,
Toast. LENGTH_LONG). show();
        showMap(j nsm ap);
    }else i f (j nsm ap==cari) {

        aPel anggan = new Pel anggan();
        aPel anggan = Hasi l Cari Acti vi ty. aPel anggan;
        Toast. makeText(getAppl i cati onContext(), "Sa
"+aPel anggan. toString() +"\n j ns="+j nsm ap,
Toast. LENGTH_LONG). show();
        showMap(j nsm ap);
    }

//     aJeni s = new Jeni s();
//     aJeni s = Jeni sActi vi ty. aJeni s;
//     Toast. makeText(getAppl i cati onContext(), "Mi
"+aJeni s. toString(), Toast. LENGTH_LONG). show();
}
private void showMap(int jeni s){

    map = ((MapFragment) getFragmentManager()
        . findFragmentByI d(R. i d. map)). getMap();

    i f (jeni s==pel anggan) {
        map. moveCamera(CameraUpdateFactory. newLatLngZoom(
            new
LatLng(Double e. parseDouble(aPel anggan. getLati tude(). trim()),
            Double e. parseDouble(aPel anggan. getLongi tude(). trim()), 18));

        map. addMarker(new MarkerOpti ons()

            . i con(Bi tmapDescri ptorFactory. fromResource(R. drawabl e. marker))
            . anchor(0. 0f, 1. 0f)
            . posi ti on(new
LatLng(Double e. parseDouble(aPel anggan. getLati tude(). trim()),
            Double e. parseDouble(aPel anggan. getLongi tude(). trim()))
            . sni ppet((aPel anggan. getAl amat(). toString()))
            . ti tle(aPel anggan. getNama(). toString()));

        //map. setMapType(Googl eMap. MAP_TYPE_NORMAL);

    }else i f (jeni s == thi s. jeni s) {
        //for (Pel anggan aPel : l i stPel anggan) {
        //
        }
}

```

```

        int ii = 1;
        ListPelangganMarker = new ArrayList<Pelanggan>();
        for (ListIterator<Pelanggan> iter =
ListPelanggan.listIterator(); iter.hasNext(); ) {
            Pelanggan aPel = iter.next();
            Marker a;
            a=map.addMarker(new MarkerOptions()

                .icon(BitmapDescriptorFactory.fromResource(R.drawable.marker))
                .anchor(0.0f, 1.0f)
                .position(new
LatLng(Double.parseDouble(aPel.getLatitude()),

                    Double.parseDouble(aPel.getLongitude())))
                .snippet((aPel.getAlamat().toString()))
                .title((aPel.getNama().toString())));

            aPel.setIdMarker(a.getId());
            ListPelangganMarker.add(aPel);
            Log.i("data list ", "pelanggan "+ii +"
:\n"+aPel.toString());
            ii++;

            map.moveCamera(CameraUpdateFactory.newLatLngZoom(
                new
LatLng(Double.parseDouble(aPel.getLatitude().trim()),

                    Double.parseDouble(aPel.getLongitude().trim()), 18));

            }

        }else if (jenis == this.cari){
            map.moveCamera(CameraUpdateFactory.newLatLngZoom(
                new
LatLng(Double.parseDouble(aPelanggan.getLatitude().trim()),

                    Double.parseDouble(aPelanggan.getLongitude().trim()), 18));
            map.addMarker(new MarkerOptions()

                .icon(BitmapDescriptorFactory.fromResource(R.drawable.marker))
                .anchor(0.0f, 1.0f)
                .position(new
LatLng(Double.parseDouble(aPelanggan.getLatitude()),

                    Double.parseDouble(aPelanggan.getLongitude())))
                .snippet((aPelanggan.getAlamat().toString()))
                .title((aPelanggan.getNama().toString())));

            }

        pd.dismiss();

        map.setMapType(GoogleMap.MAP_TYPE_NORMAL);

```

```

// Enabling MyLocation Layer of Google Map
map.setMyLocationEnabled(true);

// Setting event handler for location change
map.setOnMyLocationChangeListener(this);

map.setOnInfoWindowClickListener(new
OnInfoWindowClickListener() {

    @Override
    public void onInfoWindowClick(final Marker m) {

        tujan = new Location("");

tujan.setLatitude(m.getPosition().latitude);

tujan.setLongitude(m.getPosition().longitude);
        jarak = myLocation.distanceTo(tujan);
        intervalUpdate = 0;
        intervalUpdate2 = 0;
        String urlString = serverURL+"updatelog.php";
        UpdateLog dt = new UpdateLog();
        dt.execute(urlString);

        Log.d("url update", ""+urlString);

        Toast.makeText(getApplicationContext(),
"Marker "+m.getTitle(), Toast.LENGTH_LONG).show();

        // Create custom dialog object
        final Dialog dialog = new Dialog(TesActivity.this);
        // Include dialog.xml file
        dialog.setContentView(R.layout.menu_dialog);
        // Set dialog title
        dialog.setTitle("Custom Dialog");

        dialog.show();

        Button infoButton = (Button)
dialog.findViewById(R.id.info);
        Button directionButton = (Button)
dialog.findViewById(R.id.direction);
        // if decline button is clicked, close the custom
dialog
        infoButton.setOnClickListener(new OnClickListener()
{

    @Override
    public void onClick(View v) {
        // Close dialog
        showdialoginfodetail();
    }
}
}

```

```

        private void showDetail () {
            final Dialog dialogDetail =
            new Dialog(TesActivity.this);
            final Pelanggan aPelangganDialog
            = new Pelanggan();

            dialogDetail setContentView(R.layout.info_detail);

            DialogDetail Pelanggan";
            dialogDetail.setTitle("Info
            Detail Pelanggan");

            TextView nama = (TextView)
            dialogDetail.findViewById(R.id.nama);
            TextView alamat = (TextView)
            dialogDetail.findViewById(R.id.alamat);
            TextView notelpn = (TextView)
            dialogDetail.findViewById(R.id.notelp);
            Button tutup = (Button)
            dialogDetail.findViewById(R.id.tutup);

            // pelanggan set idmarker =
            marker.getId
            //kalo jns=jenis maka
            aPelanggan=listpelanggan di kelas ini.getId(m.getId)
            if (jnsmap ==
            TesActivity.jenis){
                for
            (ListIterator<Pelanggan> iter = listPelangganMarker.listIterator();
            iter.hasNext(); ) {
                    Pelanggan aPel =
            iter.next();
                    if
            (aPel.getIdMarker().trim().equals(m.getId().trim())){
                                aPelanggan =
            aPel;

            nama.setText(aPelanggan.getNama());
            alamat.setText(aPelanggan.getAlamat());
            notelpn.setText(aPelanggan.getNotelp());
            aPelanggan.toString();
            Log.d("aPel ",
            }
            Log.d(" list aPel ",
            aPel.getIdMarker());
            }
            m.getId());
            Log.d("marker",

```



```

        String sensor = "sensor=false";

        // Travelling Mode
        String mode = "mode=driving";

        // Building the parameters to the web
service
        String parameters =
str_origin+"&"+str_dest+"&"+sensor+"&"+mode;

        // Output format
        String output = "json";

        // Building the url to the web
service
        String url =
"https://maps.googleapis.com/maps/api/directions/"+output+"?"+parameters+"alternatives=false";
        Log.i("url ", url);

        DownloadTask downloadTask = new
DownloadTask();

        // Start downloading json data from Google
Directions API
        downloadTask.execute(url);
        dialog.dismiss();
    }
    });
}

@Override
public void onBackPressed() {
    JenisAktiviti.listPelanggan =new ArrayList<Pelanggan>();
    super.onBackPressed();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if
it is present.
    getMenuInflater().inflate(R.menu. tes, menu);
    return true;
}

private String downloadUrl (String strUrl) throws IOException{
    String data = "";
    InputStream iStream = null;
    HttpURLConnection urlConnection = null;
    try{
        URL url = new URL(strUrl);

        // Creating an http connection to communicate with url

```

```

        urlConnection = (HttpURLConnection)
url.openConnection();

        // Connecting to url
urlConnection.connect();

        // Reading data from url
InputStream inputStream = urlConnection.getInputStream();

        BufferedReader br = new BufferedReader(new
InputStreamReader(inputStream));

        StringBuffer sb = new StringBuffer();

        String line = "";
        while( ( line = br.readLine()) != null){
            sb.append(line);
        }

        data = sb.toString();

        br.close();

    }catch(Exception e){
        Log.d("Exception while downloading url", e.toString());
    }finally{
        inputStream.close();
        urlConnection.disconnect();
    }
    }
    return data;
}
private class DownloadTask extends AsyncTask<String, Void,
String>{

    // Downloading data in non-ui thread
@Override
protected String doInBackground(String... url) {

        // For storing data from web service
String data = "";

        try{
            // Fetching the data from web service
            data = downloadUrl(url[0]);
        }catch(Exception e){
            Log.d("Background Task", e.toString());
        }
        return data;
    }

    // Executes in UI thread, after the execution of
// doInBackground()
@Override
protected void onPostExecute(String result) {

```

```

        super.onPostExecute(result);

        ParserTask parserTask = new ParserTask();

        // Invokes the thread for parsing the JSON data
        parserTask.execute(result);
    }
}

private class ParserTask extends AsyncTask<String, Integer,
List<List<HashMap<String, String>>> >{

    // Parsing the data in non-ui thread
    @Override
    protected List<List<HashMap<String, String>>>
doInBackground(String... jsonData) {

        JSONObject jsonObject;
        List<List<HashMap<String, String>>> routes = null;

        try{
            jsonObject = new JSONObject(jsonData[0]);
            DirectionsJSONParser parser = new
DirectionsJSONParser();

            // Starts parsing data
            routes = parser.parse(jsonObject);
        }catch(Exception e){
            e.printStackTrace();
        }
        Log.d("hasmapparsing", routes.toString());
        return routes;
    }

    // Executes in UI thread, after the parsing process
    @Override
    protected void onPostExecute(List<List<HashMap<String,
String>>> result) {
        ArrayList<LatLng> points = null;
        PolylineOptions lineOptions = null;
        MarkerOptions markerOptions = new MarkerOptions();

        Log.d("result size", result.size()+"");

        // Traversing through all the routes
        for(int i=0; i<result.size(); i++){
            points = new ArrayList<LatLng>();
            lineOptions = new PolylineOptions();

            // Fetching i-th route
            List<HashMap<String, String>> path = result.get(i);

            Log.d("paths size", path.size()+"");

```

```

        // Fetching all the points in i-th route
        for(int j=0; j<path.size(); j++){
            HashMap<String, String> point = path.get(j);

            double lat =
Double.parseDouble(point.get("lat"));
            double lng =
Double.parseDouble(point.get("lng"));
            LatLng position = new LatLng(lat, lng);

            points.add(position);
        }

        // Adding all the points in the route to LineOptions
        lineOptions.addAll(points);
        lineOptions.width(2);
        // Changing the color polyline according to the mode

        lineOptions.color(Color.RED);
        lineOptions.width(10);
    }

    if(result.size()<1){
        Toast.makeText(getApplicationContext(), "No Points",
Toast.LENGTH_SHORT).show();
        return;
    }

    // Drawing polyline in the Google Map for the i-th route
    map.addPolyline(lineOptions);
}
}

public class UpdateLog extends AsyncTask<String, String, String>{

    final int HTTP_TIMEOUT = 30 * 1000;
    HttpClient mHttpClient;
    @Override
    protected String doInBackground(String... params) {
        String[] urls;
        urls = params;
        try {
            return downloadUrl(urls[0]);
        } catch (IOException e) {
            return "Unable to retrieve web page. URL may
be invalid.";
        }
    }
}

```

```

        private String downloadUrl (String myurl) throws IOException
    {
        BufferedReader in = null;
        try {
            HttpClient client = getHttpClient();
            HttpPost request = new HttpPost(myurl);
            String id_karyawan =
DashboardActi vity. akaryawan.getIdKaryawan(). toString();
            String post_latitude = myLocation.getLatitude()+"";
            String post_longitude = myLocation.getLongitude()+"";
            String post_latitude_tujuan = tujuan.getLatitude()+"";
            String post_longitude_tujuan =
tujuan.getLongitude()+"";

            Log.d("id_karyawan post ", id_karyawan);
            Log.d("post_latitude post ", post_latitude);
            Log.d("post_longitude post ", post_longitude);
            Log.d("post_latitude_tujuan post ", post_latitude_tujuan);
            Log.d("post_longitude_tujuan post ", post_longitude_tujuan);
            ArrayList<NameValuePair> postParameters = new
ArrayList<NameValuePair>();
            postParameters.add(new
BasicNameValuePair("id_karyawan", id_karyawan));
            postParameters.add(new BasicNameValuePair("latitude",
post_latitude));
            postParameters.add(new
BasicNameValuePair("longitude", post_longitude));
            postParameters.add(new
BasicNameValuePair("latitude_tujuan", post_latitude_tujuan));
            postParameters.add(new
BasicNameValuePair("longitude_tujuan", post_longitude_tujuan));
            URLEncodedFormEntity formEntity = new
URLEncodedFormEntity(postParameters, "utf-8");
            request.setEntity(formEntity);
            HttpResponse response = client.execute(request);
            Log.d("executeHttpPost ", response.toString());
            in = new BufferedReader(new
InputStreamReader(response.getEntity().getContent()));
            StringBuffer sb = new StringBuffer("");
            String line = "";
            String NL = System.getProperty("line.separator");
            while ((line = in.readLine()) != null) {
                sb.append(line + NL);
            }
            in.close();

            String result = sb.toString();

            Log.d("hasil ", result);
            return result;
        } catch (Exception e) {
            return null;
        }
        finally {
    }
    }

```

```

    }
}

private HttpClient getHttpClient() {
    if (mHttpClient == null) {
        mHttpClient = new DefaultHttpClient();
        final HttpParams params = mHttpClient.getParams();
        HttpConnectionParams.setConnectionTimeout(params,
HTTP_TIMEOUT);
        HttpConnectionParams.setSoTimeout(params,
HTTP_TIMEOUT);
        ConnManagerParams.setTimeout(params, HTTP_TIMEOUT);
    }
    return mHttpClient;
}

}

private void myToast(String pesan) {
    Toast.makeText(getApplicationContext(), ""+pesan,
Toast.LENGTH_LONG).show();
}

}
}

```

2. com.mylbisp.library

a) Cari.java

```

package com.mylbisp.library;

import java.util.ArrayList;

public class Cari {
    String Nama;
    String BTS;
    String Notelp;
    ArrayList<Cari> ListCari;

    public Cari() {
        // TODO Auto-generated constructor stub
    }

    public Cari(String aNama, String aBTS, String aNotelp,
ArrayList<Cari> ListCari) {
        this.Nama = aNama;
    }
}

```

```

        this.BTS = aBTS;
        this.NoteIp = aNoteIp;
        this.ListCari = ListCari;
    }

    public String getNama() {
        return Nama;
    }
    public String getBTS() {
        return BTS;
    }
    public String getNoteIp() {
        return NoteIp;
    }
    public ArrayList<Cari> getListCari(){
        return ListCari;
    }

    public void setNama(String Nama) {
        this.Nama = Nama;
    }
    public void setBTS(String BTS) {
        this.BTS = BTS;
    }
    public void setNoteIp(String NoteIp) {
        this.NoteIp = NoteIp;
    }
    public void setListCari(ArrayList<Cari> ListCari) {
        this.ListCari = ListCari;
    }

    @Override
    public String toString() {
        String cari = "\nNama : "+this.Nama+
            "\nBTS : "+this.BTS+
            "\nNoteIp : "+this.NoteIp+
            "\nListCari : "+this.ListCari;

        return cari;
    }
}

```

b) DashboardLayout.java

```

package com.mylbsisp.library;

/*
 * Copyright 2011 Google Inc.

```

```
*
* Licensed under the Apache License, Version 2.0 (the
"License");
* you may not use this file except in compliance with the
License.
* You may obtain a copy of the License at
*
*   http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing,
software
* distributed under the License is distributed on an "AS IS"
BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either express or implied.
* See the License for the specific language governing
permissions and
* limitations under the License.
*/
```

```
import android.content.Context;
import android.util.AttributeSet;
import android.view.View;
import android.view.ViewGroup;
```

```
/**
 * Custom layout that arranges children in a grid-like
manner, optimizing for even horizontal and
 * vertical whitespace.
 */
```

```
public class DashboardLayout extends ViewGroup {
```

```
    private static final int
UNEVEN_GRID_PENALTY_MULTIPLIER = 10;
```

```
    private int mMaxChildWidth = 0;
    private int mMaxChildHeight = 0;
```

```

public DashboardLayout(Context context) {
    super(context, null);
}

public DashboardLayout(Context context, AttributeSet
attrs) {
    super(context, attrs, 0);
}

public DashboardLayout(Context context, AttributeSet
attrs, int defStyle) {
    super(context, attrs, defStyle);
}

@Override
protected void onMeasure(int widthMeasureSpec, int
heightMeasureSpec) {
    mMaxChildWidth = 0;
    mMaxChildHeight = 0;

    // Measure once to find the maximum child size.

    int childWidthMeasureSpec =
MeasureSpec.makeMeasureSpec(
    MeasureSpec.getSize(widthMeasureSpec),
MeasureSpec.AT_MOST);
    int childHeightMeasureSpec =
MeasureSpec.makeMeasureSpec(
    MeasureSpec.getSize(widthMeasureSpec),
MeasureSpec.AT_MOST);

    final int count = getChildCount();
    for (int i = 0; i < count; i++) {
        final View child = getChildAt(i);
        if (child.getVisibility() == GONE) {
            continue;
        }
    }
}

```

```

        child.measure(childWidthMeasureSpec,
childHeightMeasureSpec);

        mMaxChildWidth = Math.max(mMaxChildWidth,
child.getMeasuredWidth());
        mMaxChildHeight = Math.max(mMaxChildHeight,
child.getMeasuredHeight());
    }

    // Measure again for each child to be exactly the same
size.

    childWidthMeasureSpec =
MeasureSpec.makeMeasureSpec(
        mMaxChildWidth, MeasureSpec.EXACTLY);
    childHeightMeasureSpec =
MeasureSpec.makeMeasureSpec(
        mMaxChildHeight, MeasureSpec.EXACTLY);

    for (int i = 0; i < count; i++) {
        final View child = getChildAt(i);
        if (child.getVisibility() == GONE) {
            continue;
        }

        child.measure(childWidthMeasureSpec,
childHeightMeasureSpec);
    }

    setMeasuredDimension(
        resolveSize(mMaxChildWidth, widthMeasureSpec),
        resolveSize(mMaxChildHeight,
heightMeasureSpec));
}

@Override

```

```

protected void onLayout(boolean changed, int l, int t, int r,
int b) {
    int width = r - l;
    int height = b - t;

    final int count = getChildCount();

    // Calculate the number of visible children.
    int visibleCount = 0;
    for (int i = 0; i < count; i++) {
        final View child = getChildAt(i);
        if (child.getVisibility() == GONE) {
            continue;
        }
        ++visibleCount;
    }

    if (visibleCount == 0) {
        return;
    }

    // Calculate what number of rows and columns will
    optimize for even horizontal and
    // vertical whitespace between items. Start with a 1 x N
    grid, then try 2 x N, and so on.
    int bestSpaceDifference = Integer.MAX_VALUE;
    int spaceDifference;

    // Horizontal and vertical space between items
    int hSpace = 0;
    int vSpace = 0;

    int cols = 1;
    int rows;

    while (true) {
        rows = (visibleCount - 1) / cols + 1;

```

```

        hSpace = ((width - mMaxChildWidth * cols) / (cols +
1));
        vSpace = ((height - mMaxChildHeight * rows) /
(rows + 1));

        spaceDifference = Math.abs(vSpace - hSpace);
        if (rows * cols != visibleCount) {
            spaceDifference *=
UNEVEN_GRID_PENALTY_MULTIPLIER;
        }

        if (spaceDifference < bestSpaceDifference) {
            // Found a better whitespace squareness/ratio
            bestSpaceDifference = spaceDifference;

            // If we found a better whitespace squareness and
there's only 1 row, this is
            // the best we can do.
            if (rows == 1) {
                break;
            }
        } else {
            // This is a worse whitespace ratio, use the
previous value of cols and exit.
            --cols;
            rows = (visibleCount - 1) / cols + 1;
            hSpace = ((width - mMaxChildWidth * cols) / (cols
+ 1));
            vSpace = ((height - mMaxChildHeight * rows) /
(rows + 1));
            break;
        }

        ++cols;
    }

    // Lay out children based on calculated best-fit number
of rows and cols.

```

```

        // If we chose a layout that has negative horizontal or
        vertical space, force it to zero.
        hSpace = Math.max(0, hSpace);
        vSpace = Math.max(0, vSpace);

        // Re-use width/height variables to be child
        width/height.
        width = (width - hSpace * (cols + 1)) / cols;
        height = (height - vSpace * (rows + 1)) / rows;

        int left, top;
        int col, row;
        int visibleIndex = 0;
        for (int i = 0; i < count; i++) {
            final View child = getChildAt(i);
            if (child.getVisibility() == GONE) {
                continue;
            }

            row = visibleIndex / cols;
            col = visibleIndex % cols;

            left = hSpace * (col + 1) + width * col;
            top = vSpace * (row + 1) + height * row;

            child.layout(left, top,
                (hSpace == 0 && col == cols - 1) ? r : (left +
width),
                (vSpace == 0 && row == rows - 1) ? b : (top
+ height));
            ++visibleIndex;
        }
    }
}

```

c) DirectionJSONParser.java

```
package com.mylbisp.library;
```

```

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

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

import com.google.android.gms.maps.model.LatLng;

public class DirectionsJSONParser {

    /** Receives a JSONObject and returns a list of lists
    containing latitude and longitude */
    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");

            /** Traversing all routes */
            for(int i=0;i<jRoutes.length();i++){
                jLegs = (
                (JSONObject)jRoutes.get(i)).getJSONArray("legs");
                List path = new ArrayList<HashMap<String,
                String>>());

                /** Traversing all legs */
                for(int j=0;j<jLegs.length();j++){

```

```

        jSteps = (
(JSONObject)jLegs.get(j)).getJSONArray("steps");

        /** Traversing all steps */
        for(int k=0;k<jSteps.length();k++){
            String polyline = "";
            polyline =
(String)((JSONObject)((JSONObject)jSteps.get(k)).get("p
olyline")).get("points");
            List<LatLng> list = decodePoly(polyline);

            /** Traversing all points */
            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)).latitude) );
                hm.put("lng",
Double.toString(((LatLng)list.get(l)).longitude) );
                path.add(hm);
            }
            routes.add(path);
        }
    }
} catch (JSONException e) {
    e.printStackTrace();
} catch (Exception e){
}
return routes;
}

/**
 * Method to decode polyline points
 * Courtesy :
jeffreysambells.com/2010/05/27/decoding-polylines-
from-google-maps-direction-api-with-java
 */

```

```

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;
}
}

```

d) Jenis.java

```
package com.mylbisp.library;
```

```

import java.io.Serializable;
import java.util.ArrayList;

public class Jenis implements Serializable {
    String IdJB;
    String NamaJB;
    String Nilai;
    ArrayList<Jenis> ListJenis;

    public Jenis() {
        // TODO Auto-generated constructor stub
    }

    public Jenis(String aIdJB, String aNamaJB, String
aNilai, ArrayList<Jenis> ListJenis) {
        // TODO Auto-generated constructor stub
        this.IdJB = aIdJB;
        this.NamaJB = aNamaJB;
        this.Nilai = aNilai;
        this.ListJenis = ListJenis;
    }
    public String getIdJB() {
        return IdJB;
    }
    public String getNamaJB() {
        return NamaJB;
    }
    public String getNilai() {
        return Nilai;
    }
    public ArrayList<Jenis> getListJenis(){
        return ListJenis;
    }

    public void setIdJB(String IdJB) {
        this.IdJB = IdJB;
    }

```

```

    }
    public void setNamaJB(String NamaJB) {
        this.NamaJB = NamaJB;
    }
    public void setNilai(String Nilai) {
        this.Nilai = Nilai;
    }
    public void setListJenis(ArrayList<Jenis> ListJenis) {
        this.ListJenis = ListJenis;
    }
}

```

```

@Override
public String toString() {
    String jenis = "IdJB : "+this.IdJB+
        "\nNamaJB : "+this.NamaJB+
        "\nNilai : "+this.Nilai+
        "\nListJenis : "+this.ListJenis;

    return jenis;
}
}

```

e) JSONParser.java

```

package com.mylbisp.library;

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 {

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

    // constructor
    public JSONParser() {

    }

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

    // return JSON String
    return jObj;

}

}

```

f) Pelanggan.java

```

package com.mylbisp.library;

import java.io.Serializable;
import java.util.ArrayList;

public class Pelanggan implements Serializable {
    String IdPelanggan;

```

```

String IdJenisJB;
String Nama;
String Alamat;
String Notelp;
String Email;
String Latitude;
String Longitude;
String BTS;
String Status;
ArrayList<Pelanggan> ListPelanggan;
private String idMarker;

public Pelanggan(){

}

public Pelanggan(String aIdPelanggan, String
aIdJenisJB, String aNama, String aAlamat, String aNotelp,
String aEmail, String aLatitude, String aLongitude, String
aBTS, String aStatus, ArrayList<Pelanggan>
ListPelanggan){
    this.IdPelanggan = aIdPelanggan;
    this.IdJenisJB = aIdJenisJB;
    this.Nama = aNama;
    this.Alatamat = aAlamat;
    this.Notelp = aNotelp;
    this.Email = aEmail;
    this.Latitude = aLatitude;
    this.Longitude = aLongitude;
    this.BTS = aBTS;
    this.Status = aStatus;
    this.ListPelanggan = ListPelanggan;
}
public String getIdPelanggan() {
    return IdPelanggan;
}
public String getIdJenisJB(){
    return IdJenisJB;
}

```

```

}

public String getNama() {
    return Nama;
}
public String getAlamat() {
    return Alamat;
}
public String getNotelp() {
    return Notelp;
}
public String getEmail() {
    return Email;
}
public String getLatitude() {
    return Latitude;
}
public String getLongitude() {
    return Longitude;
}
public String getBTS() {
    return BTS;
}
public String getStatus() {
    return Status;
}
public ArrayList<Pelanggan> getListPelanggan(){
    return ListPelanggan;
}

```

```

public void setIdPelanggan(String IdPelanggan) {
    this.IdPelanggan = IdPelanggan;
}
public void setIdJenisJB(String IdJenisJB){
    this.IdJenisJB = IdJenisJB;
}

```

```

    }
    public void setName(String Nama) {
        this>Nama = Nama;
    }
    public void setAddress(String Alamat) {
        this.Alatamat = Alamat;
    }
    public void setNotelp(String Notelp) {
        this.Notelp = Notelp;
    }
    public void setEmail(String Email) {
        this.Email = Email;
    }
    public void setLatitude(String Latitude) {
        this.Latitude = Latitude;
    }
    public void setLongitude(String Longitude) {
        this.Longitude = Longitude;
    }
    public void setBTS(String BTS) {
        this.BTS = BTS;
    }
    public void setStatus(String Status) {
        this.Status = Status;
    }
    public void setListPelanggan(ArrayList<Pelanggan>
ListPelanggan) {
        this.ListPelanggan = ListPelanggan;
    }

    public String toString() {
        String pelanggan = "Nama : "+this>Nama+
            "\nAlamat : "+this.Alatamat+
            "\nNotelp : "+this.Notelp+
            "\nBTS : "+this.BTS+
            "\nLat : "+this.Latitude+
            "\nLog : "+this.Longitude+

```

```

        "\nListPelanggan :
"+this.ListPelanggan;
        return pelanggan;
    }

    public void setIdMarker(String id) {
        this.idMarker = id;
    }

    public String getIdMarker() {
        return idMarker;
    }
}

```

g) KoneksiHTTP.java

```

package com.mylbsisp.library;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.ArrayList;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import
org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import
org.apache.http.conn.params.ConnManagerParams;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;

public class KoneksiHTTP {
    public static final int HTTP_TIMEOUT = 3000;
    private static HttpClient client;

    private static HttpClient getHttpClient() {

```

```

        if (client == null) {
            client = new DefaultHttpClient();
            final HttpParams parameterHttp =
client.getParams();

            HttpConnectionParams.setConnectionTimeout(param
eterHttp,

                HTTP_TIMEOUT);

            ConnManagerParams.setTimeout(parameterHttp,
HTTP_TIMEOUT);
        }
        return client;
    }

    public static String eksekusiHttpPost(String url,
        ArrayList<NameValuePair>
postParameter) throws Exception {
        BufferedReader in = null;

        try {
            HttpClient klien = getHttpClient();
            HttpPost req = new HttpPost(url);
            UriEncodedFormEntity formEntity = new
UriEncodedFormEntity(
                postParameter);
            req.setEntity(formEntity);
            HttpResponse resp = klien.execute(req);

            in = new BufferedReader(new
InputStreamReader(resp.getEntity()
                .getContent()));
            StringBuffer stringBuffer = new
StringBuffer("");
            String line = "";
            String NL =
System.getProperty("line.separator");
            while ((line = in.readLine()) != null) {

```

```

        stringBuffer.append(line + NL);
    }
    in.close();
    String hasil = stringBuffer.toString();
    return hasil;
} finally {
    if (in != null) {
        in.close();
    }
}
}
}
}

```

h) Karyawan.java

```
package com.mylbsi.sp.library;
```

```
import java.io.Serializable;
```

```

public class Karyawan implements Serializable {
    String IdKaryawan;
    String Username;
    String Password;
    String Nama;
    String Alamat;
    String Notelp;
    String Email;

    public Karyawan() {
        // TODO Auto-generated constructor stub
    }

    public Karyawan(String aldKaryawan, String aUsername, String
aPassword, String aNama, String aAlamat, String aNotelp, String aEmail)
{
        // TODO Auto-generated constructor stub
        this.IdKaryawan = aldKaryawan;
        this.Username = aUsername;
        this.Password = aPassword;
        this.Nama = aNama;
        this.Alatmat = aAlamat;
        this.Notelp = aNotelp;
        this.Email = aEmail;
    }

    public String getIdKaryawan() {

```

```

        return IdKaryawan;
    }
    public String getUsername() {
        return Username;
    }
    public String getPassword() {
        return Password;
    }
    public String getNama() {
        return Nama;
    }
    public String getAlamat() {
        return Alamat;
    }
    public String getNotelp() {
        return Notelp;
    }
    public String getEmail() {
        return Email;
    }
}

    public void setIdKaryawan(String IdKaryawan) {
        this.IdKaryawan = IdKaryawan;
    }
    public void setUsername(String Username) {
        this.Username = Username;
    }
    public void setPassword(String Password) {
        this.Password = Password;
    }
    public void setNama(String Nama) {
        this.Nama = Nama;
    }
    public void setAlamat(String Alamat) {
        this.Alatamat = Alamat;
    }
    public void setNotelp(String Notelp) {
        this.Notelp = Notelp;
    }
    public void setEmail(String Email) {
        this.Email = Email;
    }
}

@Override
    public String toString() {
        String karyawan = "Nama : "+this.Nama+
            "\nNotelp : "+this.Notelp;

        return karyawan;
    }
}

```

3. j