Lampiran 1

LISTING PROGRAM

1. Class Login.java

```
package com.sikurro.siamumjember;

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

import com.sikurro.siamumjember.datalite.DataMhsw;
import com.sikurro.siamumjember.librarry.UseFunctions;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class LoginActivity extends Activity implements
OnClickListener {

	Typeface font;
	TextView header;
	Button btnLogin, btnLupaPassword;
	ProgressDialog progressDialog;
	EditText inputNIM, inputPassword;
	String MhswID, Password;
	boolean status;
	private final String msgLupa = "Silahkan Anda
menghubungi pihak Pusat Data Informasi (PDI) untuk segera
dapat perbaikan password Sistem Informasi Akademik
Mahasiswa.";

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		// TODO Auto-generated method stub
		super.onCreate(savedInstanceState);
```

```
      setContentView(com.sikurro.siamumjember.R.layout.login
);
            inisialisasi();
            try {
                  new resetData().execute();
                  Log.d("reset data :", "sukses1");
            } catch (Exception e) {
                  // TODO: handle exception
                  Log.e("reset data :", e.getMessage());
            }
      }

      void inisialisasi() {
            header = (TextView) findViewById(R.id.headerLog);
            btnLogin = (Button) findViewById(R.id.btnLogin);
            btnLogin.setOnClickListener(this);

            btnLupaPassword = (Button)
findViewById(R.id.btnLupaPassword);
            btnLupaPassword.setOnClickListener(this);

            inputNIM = (EditText)
findViewById(R.id.txtLoginNim);
            inputPassword = (EditText)
findViewById(R.id.txtLoginPassword);

            font = Typeface.createFromAsset(getAssets(),
"fonts/bandless.ttf");
            header.setTypeface(font);
      }

      @Override
      protected void onPause() {
            // TODO Auto-generated method stub
            super.onPause();
            finish();
      }

      class loginAsy extends AsyncTask<String, String,
String> {

            @Override
            protected void onPreExecute() {
                  // TODO Auto-generated method stub
                  super.onPreExecute();
                  pd();
            }

            @Override
```

```
            protected String doInBackground(String... params)
{
                  // TODO Auto-generated method stub
                  try {
                        login();
                        setCeklogin(false);
                  } catch (Exception e) {
                        setCeklogin(true);
                        progressDialog.dismiss();
                        // TODO: handle exception
                  }

                  return null;
            }

            @Override
            protected void onPostExecute(String result) {
                  // TODO Auto-generated method stub
                  super.onPostExecute(result);

                  if (isCeklogin()) {

      Toast.makeText(getApplicationContext(), "Login Error",
                              Toast.LENGTH_LONG).show();
                  } else {
                        progressDialog.dismiss();
                        if (isStatus()) {

      Toast.makeText(getApplicationContext(),
                                    "Incorrect Nim or
Password!", Toast.LENGTH_LONG)
                                    .show();

                        }
                  }

            }

      }

      protected void openActivty() {
            try {

                  @SuppressWarnings("rawtypes")
                  Class ourClass = Class

      .forName("com.sikurro.siamumjember.slidingmenu.MainAct
ivity");

                  Intent ourIntent = new
Intent(LoginActivity.this, ourClass);
```

```
                startActivity(ourIntent);

            } catch (Exception e) {
                // TODO: handle exception
                e.printStackTrace();
            }
        }

        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub

            switch (v.getId()) {
            case R.id.btnLogin:

                try {

                    new loginAsy().execute();

                } catch (Exception e) {
                    // TODO: handle exception
                }

                break;

            case R.id.btnLupaPassword:
                AlertDialog.Builder alert = new
AlertDialog.Builder(
                        LoginActivity.this);
                alert.setTitle("Lupa Password?");
                alert.setIcon(R.drawable.ic_detil_mhsw);
                alert.setMessage(msgLupa);
                alert.setPositiveButton("Ok",
                        new
DialogInterface.OnClickListener() {

                            @Override
                            public void
onClick(DialogInterface dialog, int which) {
                                // TODO Auto-
generated method stub
                                dialog.cancel();
                            }
                        });
                alert.show();

                break;
            default:
                break;
            }
```

```
	}

	protected void login() {
		getText();

		UseFunctions useFunctions = new UseFunctions();
		Log.d("Button", "Login");

		JSONObject json = useFunctions.loginMhsw(MhswID,
Password);

		try {
			if (json.getString(KEY_SUCCESS) != null) {

				String res =
json.getString(KEY_SUCCESS);
				if (Integer.parseInt(res) == 1) {
					// user successfully logged in
					// Store user details in SQLite
Database
					DataMhsw dataMhsw = new
DataMhsw();
					JSONObject json_mhsw =
json.getJSONObject("mhsw");

					// Clear all previous data in
database

	useFunctions.logoutMhsw(getApplicationContext());


	dataMhsw.setMhswID(json_mhsw.getString(KEY_MHSWID));

	dataMhsw.setTahunID(json_mhsw.getString(KEY_TAHUNID));

	dataMhsw.setNama(json_mhsw.getString(KEY_NAMA));

	dataMhsw.setFoto(json_mhsw.getString(KEY_FOTO));

	dataMhsw.setStatusMhsw(json_mhsw.getString(KEY_STATUSM
HSW));

	dataMhsw.setProgram(json_mhsw.getString(KEY_PROGRAM));

	dataMhsw.setProdi(json_mhsw.getString(KEY_PRODI));

	dataMhsw.setPenasehatAkademik(json_mhsw

	.getString(KEY_PENASEHATAKADEMIK));

	dataMhsw.setKelamin(json_mhsw.getString(KEY_KELAMIN));
```

```
        dataMhsw.setTempatLahir(json_mhsw
        .getString(KEY_TEMPATLAHIR));
        dataMhsw.setTanggalLahir(json_mhsw
        .getString(KEY_TANGGALLAHIR));
        dataMhsw.setAgama(json_mhsw.getString(KEY_AGAMA));
        dataMhsw.setAlamat(json_mhsw.getString(KEY_ALAMAT));
        dataMhsw.setKota(json_mhsw.getString(KEY_KOTA));
        dataMhsw.setRT(json_mhsw.getString(KEY_RT));
        dataMhsw.setRW(json_mhsw.getString(KEY_RW));
        dataMhsw.setKodePos(json_mhsw.getString(KEY_KODEPOS));
        dataMhsw.setPropinsi(json_mhsw.getString(KEY_PROPINSI)
);
        dataMhsw.setTelepon(json_mhsw.getString(KEY_TELEPON));
        dataMhsw.setHandphone(json_mhsw.getString(KEY_HANDPHON
E));
        dataMhsw.setEmail(json_mhsw.getString(KEY_EMAIL));

        dataMhsw.addDataMhsw(getApplicationContext());
                              setStatus(false);
                              openActivty();
                        } else {
                              setStatus(true);
                        }
                  }

            } catch (JSONException e) {
                  // TODO: handle exception
                  Log.e("Error Login : ", e.getMessage());
            }

        }

        protected void getText() {
              MhswID = inputNIM.getText().toString();
              Password = inputPassword.getText().toString();
        }
```

```
      class resetData extends AsyncTask<String, String,
String> {
            @Override
            protected void onPreExecute() {
                  // TODO Auto-generated method stub
                  super.onPreExecute();
                  pd();

            }

            @Override
            protected String doInBackground(String... params)
{
                  // TODO Auto-generated method stub

                  UseFunctions functions = new
UseFunctions();

      functions.logoutMhsw(getApplicationContext());
                  Log.d("reset data :", "sukses1");
                  return null;
            }

            @Override
            protected void onPostExecute(String result) {
                  // TODO Auto-generated method stub
                  super.onPostExecute(result);
                  progressDialog.dismiss();
            }

      }

      protected void pd() {
            progressDialog = new
ProgressDialog(LoginActivity.this);
            progressDialog.setMessage("Please Wait...");
            progressDialog.setIndeterminate(false);
            progressDialog.show();
      }

      public boolean isStatus() {
            return status;
      }

      public void setStatus(boolean status) {
            this.status = status;
      }

      public boolean isCeklogin() {
            return ceklogin;
```

```
	}

	public void setCeklogin(boolean ceklogin) {
		this.ceklogin = ceklogin;
	}

	boolean ceklogin = false;

	// JSON Response node name
	private static String KEY_SUCCESS = "success";
	private static final String KEY_MHSWID = "MhswID";
	private static final String KEY_TAHUNID = "TahunID";
	private static final String KEY_NAMA = "Nama";
	private static final String KEY_FOTO = "Foto";
	private static final String KEY_STATUSMHSW =
"StatusMhsw";
	private static final String KEY_PROGRAM = "Program";
	private static final String KEY_PRODI = "Prodi";
	private static final String KEY_PENASEHATAKADEMIK =
"PenasehatAkademik";
	private static final String KEY_KELAMIN = "Kelamin";
	private static final String KEY_TEMPATLAHIR =
"TempatLahir";
	private static final String KEY_TANGGALLAHIR =
"TanggalLahir";
	private static final String KEY_AGAMA = "Agama";
	private static final String KEY_ALAMAT = "Alamat";
	private static final String KEY_KOTA = "Kota";
	private static final String KEY_RT = "RT";
	private static final String KEY_RW = "RW";
	private static final String KEY_KODEPOS = "KodePos";
	private static final String KEY_PROPINSI = "Propinsi";
	private static final String KEY_TELEPON = "Telepon";
	private static final String KEY_HANDPHONE =
"Handphone";
	private static final String KEY_EMAIL = "Email";

}
```

2. Class Use_Function.java

```
package com.sikurro.siamumjember.librarry;

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

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONObject;
```

```
import android.content.Context;
import android.util.Log;

public class UseFunctions {

	private JSONParser jsonParser;
//192.168.221.1
	//10.0.2.2
	private static String apiURL =
"http://192.168.221.1/siamuj_api/";

	private static String tag_login = "login";
	private static String tag_krs = "krs";
	private static String tag_nilai = "nilaisemester";
	private static String tag_bayar = "detilbayar";
	private static String tag_rhs = "rekaphasil";
	private static String tag_ubah_pass = "ubahpassword";

	public UseFunctions() {
		jsonParser = new JSONParser();
	}

	public JSONObject loginMhsw(String MhswID, String
Password) {

		List<NameValuePair> params = new
ArrayList<NameValuePair>();
		params.add(new BasicNameValuePair("tag",
tag_login));
		params.add(new BasicNameValuePair("MhswID",
MhswID));
		params.add(new BasicNameValuePair("Password",
Password));
		JSONObject json =
jsonParser.getJSONFromUrl(apiURL, params);

		Log.e("JSON Mhsw", json.toString());
		return json;

	}

	public JSONObject krs(String MhswID) {
		List<NameValuePair> params = new
ArrayList<NameValuePair>();
		params.add(new BasicNameValuePair("tag",
tag_krs));
		params.add(new BasicNameValuePair("MhswID",
MhswID));
		JSONObject json =
jsonParser.getJSONFromUrl(apiURL, params);
```

```
            Log.e("JSON KRS", json.toString());
            return json;
      }

      public JSONObject nilaiSemester(String MhswID, String
TahunID) {
            List<NameValuePair> params = new
ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair("tag",
tag_nilai));
            params.add(new BasicNameValuePair("MhswID",
MhswID));
            params.add(new BasicNameValuePair("TahunID",
TahunID));
            JSONObject json =
jsonParser.getJSONFromUrl(apiURL, params);

            Log.e("JSON KRS", json.toString());
            return json;
      }

      public JSONObject rekapHasil(String MhswID) {
            List<NameValuePair> params = new
ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair("tag",
tag_rhs));
            params.add(new BasicNameValuePair("MhswID",
MhswID));
            JSONObject json =
jsonParser.getJSONFromUrl(apiURL, params);

            Log.e("JSON KRS", json.toString());
            return json;
      }

      public JSONObject detilBayar(String MhswID) {
            List<NameValuePair> params = new
ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair("tag",
tag_bayar));
            params.add(new BasicNameValuePair("MhswID",
MhswID));
            JSONObject json =
jsonParser.getJSONFromUrl(apiURL, params);

            Log.e("JSON KRS", json.toString());
            return json;
      }

      public JSONObject ubahPassword(String MhswID, String
passwordLama, String passwordBaru) {
```

```
            List<NameValuePair> params = new
ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair("tag",
tag_ubah_pass));
            params.add(new BasicNameValuePair("MhswID",
MhswID));
            params.add(new BasicNameValuePair("PasswordLama",
passwordLama));
            params.add(new BasicNameValuePair("PasswordBaru",
passwordBaru));
            JSONObject json =
jsonParser.getJSONFromUrl(apiURL, params);

            Log.e("JSON KRS", json.toString());
            return json;
      }

      public boolean logoutMhsw(Context context) {

            DatabaseHandler db = new
DatabaseHandler(context);
            db.resetAllTable();
            db.close();
            return true;
      }

      public boolean isMhswLoggedIn(Context context) {

            DatabaseHandler db = new
DatabaseHandler(context);
            int count = db.getRowCountMhsw();
            if (count > 0) {
                  return true;
            }

            return false;
      }

      public boolean isCekKRS(Context context) {
            DatabaseHandler db = new
DatabaseHandler(context);
            int count = db.getRowCountKRS();
            if (count > 0) {
                  return true;
            }
            return false;
      }

      public boolean isCekRHS(Context context) {
            DatabaseHandler db = new
DatabaseHandler(context);
```

```
            int count = db.getRowCountRHS();
            if (count > 0) {
                  return true;
            }
            return false;
      }

      public boolean isCekPembayaran(Context context) {
            DatabaseHandler db = new
DatabaseHandler(context);
            int count = db.getRowCountBayar();
            if (count > 0) {
                  return true;
            }
            return false;
      }


      public boolean isCekNilai(Context context) {
            DatabaseHandler db = new
DatabaseHandler(context);
            int count = db.getRowCountNilai();
            if (count > 0) {
                  return true;
            }
            return false;
      }

}
```