Sinkronisasi Data Lokal dan Server dengan alarm Manager.

1. Buat project dengan nama SchedulerDataBarang, isi comapany domain dengan agusharyanto.net klik next terus sampai   Pada pilihan activity pileh Empty Activity klik Next terus sampai finish. Maka project akan tercreate secara otomatis.

2. Pada gradle module:app tambahkan library recylerview, cardview dan volley

```
compile 'com.android.support:recyclerview-v7:25.3.1'
compile 'com.android.support:cardview-v7:25.3.1'
compile 'com.mcxiaoke.volley:library-aar:1.0.+'
```

3. Disini kita nantinya akan menampilkan data dari database sqlite dengan menggunakan recycler view. untuk itu pada **activity_main** ketik code ini

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="net.agusharyanto.schedulerdatabarang.MainActivity">
    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerViewBarang"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true" />
</RelativeLayout>
```

4. kita sipakan juga layout yang digunakan untuk baris pada recyclerview. buat layout baru dengan nama row_barang

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="0dp">
    <android.support.v7.widget.CardView
        android:id="@+id/cv"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        >
        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:padding="10dp"
            >
            <TextView
                android:id="@+id/textViewRowKode"
                android:layout_width="100dp"
                android:layout_height="wrap_content"
                android:layout_alignParentLeft="true"
                android:layout_alignParentStart="true"
                android:layout_alignParentTop="true"
                android:text="Kode" />
            <TextView
                android:id="@+id/textViewRowNama"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentTop="true"
                android:layout_toEndOf="@+id/textViewRowKode"
                android:layout_toRightOf="@+id/textViewRowKode"
                android:text="Nama" />
            <TextView
                android:id="@+id/textViewRowHarga"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentEnd="true"
                android:layout_alignParentRight="true"
                android:layout_alignParentTop="true"
                android:text="Harga" />
        </RelativeLayout>
    </android.support.v7.widget.CardView>
</LinearLayout>
```

5. Kalau di pemrograman java yang berkonsep OOP kita harus memandang segala sesuatu sebagai object. untuk itu buat class baru dengan nama **Barang**

```
package net.agusharyanto.schedulerdatabarang;
import java.io.Serializable;
/**
 * Created by agus on 9/27/17.
 */
public class Barang implements Serializable{
    private String id="";
    private String kode="";
    private String nama="";
    private String harga ="";
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getKode() {
        return kode;
    }
    public void setKode(String kode) {
        this.kode = kode;
    }
    public String getNama() {
        return nama;
    }
    public void setNama(String nama) {
        this.nama = nama;
    }
    public String getHarga() {
        return harga;
    }
    public void setHarga(String harga) {
        this.harga = harga;
    }
    @Override
    public String toString() {
        return "Barang{" +
                "id='" + id + '\'' +
                ", kode='" + kode + '\'' +
                ", nama='" + nama + '\'' +
                ", harga='" + harga + '\'' +
                '}';
    }
}
```

5. Sekarang kita buat databasenya, buat class baru dengan nama **DatabaseHelper**

```
package net.agusharyanto.schedulerdatabarang;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import java.util.ArrayList;
/**
 * Created by agus on 10/25/17.
 */
public class DatabaseHelper extends SQLiteOpenHelper {
    private final static String DATABASE_NAME ="dbbarang";
    private final static int DATABASE_VERSION = 1;
    private final static String BARANG_TABLE = "tbl_barang";
    private final static String FIELD_ID="_id";
    private final static String FIELD_KODE ="kode";
    private final static String FIELD_NAMA="nama";
    private final static String FIELD_HARGA ="harga";
    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        // creat table dan init data
        String SQL_CREATE_TABLE = "create table "+ BARANG_TABLE +"
("+FIELD_ID+" integer primary key, "
                + FIELD_KODE + " text not null, "+FIELD_NAMA+ " text not null,"
                + FIELD_HARGA +" text not null);";
        db.execSQL(SQL_CREATE_TABLE);
        initData(db);
    }
    private void initData(SQLiteDatabase db){
        ContentValues contentValues = new ContentValues();
        contentValues.put(FIELD_ID,"1");
        contentValues.put(FIELD_KODE,"B001");
        contentValues.put(FIELD_NAMA,"Mouse M238");
        contentValues.put(FIELD_HARGA,"150000");
        db.insert(BARANG_TABLE,null,contentValues);
        ContentValues contentValues1 = new ContentValues();
        contentValues.put(FIELD_ID,"2");
        contentValues1.put(FIELD_KODE,"B002");
        contentValues1.put(FIELD_NAMA,"Mouse B175");
        contentValues1.put(FIELD_HARGA, "100000");
        db.insert(BARANG_TABLE, null, contentValues1);
        ContentValues contentValues2 = new ContentValues();
        contentValues.put(FIELD_ID,"3");
        contentValues2.put(FIELD_KODE,"B003");
        contentValues2.put(FIELD_NAMA,"Mouse M170");
        contentValues2.put(FIELD_HARGA, "120000");
        db.insert(BARANG_TABLE,null,contentValues2);
    }
    public ArrayList<Barang> getDataBarang(SQLiteDatabase db){
        ArrayList<Barang> barangArrayList = new ArrayList<Barang>();
        String[] allColumns = {FIELD_ID, FIELD_KODE, FIELD_NAMA, FIELD_HARGA};
        Cursor cursor =
db.query(BARANG_TABLE,allColumns,null,null,null,null,null);
        cursor.moveToFirst();
        while (!cursor.isAfterLast()) {
            Barang barang = cursorToBarang(cursor);
            barangArrayList.add(barang);
            cursor.moveToNext();
        }
        return  barangArrayList;
    }
    public long updateBarang(Barang barang, SQLiteDatabase db) {
        ContentValues initialValues = new ContentValues();
        initialValues.put(FIELD_KODE, barang.getKode());
        initialValues.put(FIELD_NAMA, barang.getNama());
        initialValues.put(FIELD_HARGA, barang.getHarga());
        long rowaffect =db.update(BARANG_TABLE, initialValues, FIELD_ID + "=" +
barang.getId(), null);
        return rowaffect;
    }
    public void deleteBarang(Barang barang, SQLiteDatabase db) {
        String id = barang.getId();
        db.delete(BARANG_TABLE, FIELD_ID + " = " + id, null);
    }
    private Barang cursorToBarang(Cursor cursor) {
        Barang barang = new Barang();
        barang.setId(cursor.getString(0));
        barang.setKode(cursor.getString(1));
        barang.setNama(cursor.getString(2));
        barang.setHarga(cursor.getString(3));
        return barang;
    }
    public long insertBarang(Barang barang, SQLiteDatabase db) {
        ContentValues initialValues = new ContentValues();
        initialValues.put(FIELD_ID, barang.getId());
        initialValues.put(FIELD_KODE, barang.getKode());
        initialValues.put(FIELD_NAMA, barang.getNama());
        initialValues.put(FIELD_HARGA, barang.getHarga());
        /*kita gunakan insertWithConflict dengan
SQLiteDatabase.CONFLICT_REPLACE
        agar jika ada duplikat data maka akan direplace (update) dan
        jika tidak duplikat maka insert seperti biasa
         */
        long insertId = db.insertWithOnConflict(BARANG_TABLE, null,
initialValues, SQLiteDatabase.CONFLICT_REPLACE);
        return insertId;
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }
}
```

7. Untuk menampilkan data di RecyclerView tentunya kita butuh adapter, untuk itu buat class baru dengan nama **BarangAdapter**

```
package net.agusharyanto.schedulerdatabarang;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import java.util.List;
/**
 * Created by agus on 9/26/17.
 */
public class BarangAdapter  extends
RecyclerView.Adapter<BarangAdapter.BarangViewHolder> {
    List<Barang> barangs;
    public BarangAdapter(List<Barang> barangs) {
        this.barangs = barangs;
    }
    @Override
    public BarangViewHolder onCreateViewHolder(ViewGroup viewGroup, int i) {
        View v =
LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.row_barang,
viewGroup, false);
        BarangViewHolder barangViewHolder = new BarangViewHolder(v);
        return barangViewHolder;
    }
    @Override
    public void onBindViewHolder(BarangViewHolder barangViewHolder, int i) {
        barangViewHolder.barangName.setText(barangs.get(i).getNama());
        barangViewHolder.barangHarga.setText(barangs.get(i).getHarga());
        barangViewHolder.barangKode.setText(barangs.get(i).getKode());
    }
    @Override
    public int getItemCount() {
        return barangs.size();
    }
    public Barang getItem(int position) {
        return barangs.get(position);
    }
    @Override
    public void onAttachedToRecyclerView(RecyclerView recyclerView) {
        super.onAttachedToRecyclerView(recyclerView);
    }
    public static class BarangViewHolder extends RecyclerView.ViewHolder {
        CardView cv;
        TextView barangName;
        TextView barangHarga;
        TextView barangKode;
        BarangViewHolder(View itemView) {
            super(itemView);
            cv = (CardView) itemView.findViewById(R.id.cv);
            barangName = (TextView)
itemView.findViewById(R.id.textViewRowNama);
            barangHarga = (TextView)
itemView.findViewById(R.id.textViewRowHarga);
            barangKode = (TextView)
itemView.findViewById(R.id.textViewRowKode);
        }
    }
}
```

8. Pada MainActivity.java ketikan kode berikut

```
package net.agusharyanto.schedulerdatabarang;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.support.v4.content.LocalBroadcastManager;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
```

```java
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;
public class MainActivity extends AppCompatActivity {
    private RecyclerView mRecyclerView;
    private BarangAdapter rvAdapter;
    private RecyclerView.LayoutManager mLayoutManager;
    private List<Barang> barangList = new ArrayList<Barang>();
    private DatabaseHelper databaseHelper;
    private SQLiteDatabase db;
    private BroadcastReceiver receiver;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        databaseHelper = new DatabaseHelper(MainActivity.this);
        db = databaseHelper.getWritableDatabase();
        mRecyclerView = (RecyclerView) findViewById(R.id.recyclerViewBarang);
        mRecyclerView.setHasFixedSize(true);
        mLayoutManager = new LinearLayoutManager(this);
        mRecyclerView.setLayoutManager(mLayoutManager);
        gambarDatakeRecyclerView();
        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.SECOND, 10);
        Intent intent = new Intent(this, BarangService.class);
        PendingIntent pintent = PendingIntent.getService(this, 0, intent, 0);
        AlarmManager alarm = (AlarmManager)
getSystemService(Context.ALARM_SERVICE);
        //for 3 mint 3*60*1000
        alarm.setRepeating(AlarmManager.RTC_WAKEUP, cal.getTimeInMillis(),
                3*60*1000, pintent);
        //Receiver untuk menerima perintah dari service
        receiver = new BroadcastReceiver() {
            @Override
            public void onReceive(Context context, Intent intent) {
                String s = intent.getStringExtra(BarangService.BARANG_MESSAGE);
                Log.d("TAG","receives:"+s);
                gambarDatakeRecyclerView();
            }
        };
    }
    //Dimethod onResume ini kita register receiver agar bisa menangkap
broadcast dari service
    @Override
    protected void onResume() {
         super.onResume();
        LocalBroadcastManager.getInstance(this).registerReceiver((receiver),
                new IntentFilter(BarangService.BARANG_RESULT)
        );
    }
    //Dimethod ini kita unregister receivernya agar kalau activity tidak aktif
tidak perlu menangkap broadcast
    @Override
    protected void onPause() {
        LocalBroadcastManager.getInstance(this).unregisterReceiver(receiver);
        super.onPause();
    }
    private void gambarDatakeRecyclerView(){
        barangList.clear();
        barangList = databaseHelper.getDataBarang(db);
        rvAdapter = new BarangAdapter(barangList);
        mRecyclerView.setAdapter(rvAdapter);
    }
}
```

9. Buat Service dengan nama BarangService lalu ketikan kode berikut

```
package net.agusharyanto.schedulerdatabarang;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.os.IBinder;
import android.support.v4.content.LocalBroadcastManager;
import android.util.Log;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
public class BarangService extends Service {
    private ArrayList<Barang> barangList = new ArrayList<Barang>();
    private DatabaseHelper databaseHelper;
    private SQLiteDatabase db;
    private Context context;
    LocalBroadcastManager broadcaster;
    @Override
    public IBinder onBind(Intent intent) {
        // TODO: Return the communication channel to the service.
        throw new UnsupportedOperationException("Not yet implemented");
    }
    @Override
    public void onCreate() {
        // TODO Auto-generated method stub
        super.onCreate();
        broadcaster = LocalBroadcastManager.getInstance(this);
        databaseHelper = new DatabaseHelper(this);
        db= databaseHelper.getWritableDatabase();
        context = this;
    }
    @Override
    public void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        db.close();
        databaseHelper.close();
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // TODO Auto-generated method stub
        Log.d("TAG","Service Running");
            loadDataServerVolley();
        return super.onStartCommand(intent, flags, startId);
    }
    private void loadDataServerVolley(){
        String url = "http://192.168.0.102/barang/listdata.php";
        StringRequest postRequest = new StringRequest(Request.Method.POST, url,
                new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        processResponse(response);
                    }
                },
                new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        error.printStackTrace();
                    }
                }
        ) {
            @Override
            protected Map<String, String> getParams()
            {
                Map<String, String>  params = new HashMap<>();
                return params;
            }
        };
        Volley.newRequestQueue(this).add(postRequest);
    }
    private void processResponse(String response){
        try {
            JSONObject jsonObj = new JSONObject(response);
            JSONArray jsonArray = jsonObj.getJSONArray("data");
           // Log.d("TAG", "data length: " + jsonArray.length());
            Barang objectbarang = null;
            barangList.clear();
            for(int i = 0; i < jsonArray.length(); i++){
                JSONObject obj = jsonArray.getJSONObject(i);
                objectbarang= new Barang();
                objectbarang.setId(obj.getString("id"));
                objectbarang.setNama(obj.getString("nama"));
                objectbarang.setKode(obj.getString("kode"));
                objectbarang.setHarga(obj.getString("harga"));
                databaseHelper.insertBarang(objectbarang, db);
            }
            sendResult("update");
        } catch (JSONException e) {
            Log.d("BarangService", "errorJSON");
        }
    }
    static final public String BARANG_RESULT =
"net.agusharyanto.schedulerdatabarang.REQUEST_PROCESSED";
    static final public String BARANG_MESSAGE =
"net.agusharyanto.schedulerdatabarang.BARANG_MSG";
    //Method yang akan mentrigger method yang dipanggil dibroadcast receiver
onrecive diactivity
    public void sendResult(String message) {
        Intent intent = new Intent(BARANG_RESULT);
        if(message != null)
            intent.putExtra(BARANG_MESSAGE, message);
        broadcaster.sendBroadcast(intent);
    }
}
```

10. Tambahkan permission INTERNET pada AndroidManifest.xml, ketikan kode berikut

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.agusharyanto.schedulerdatabarang">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service
            android:name=".BarangService"
            android:enabled="true"
            android:exported="true" />
    </application>
</manifest>
```

11. Selesai sudah mari kita jalankan programnya dan perhatikan bahwa scheduler jalan setiap 3 menit sekali, rubah data diserver dan perhatikan bahwa data diserver sudah sama dengan dilokal.