

Tutorial Aplikasi CRUD Android Studio & PHP MySQL

Pada tutorial kali ini kita akan membuat aplikasi **CRUD (Create, Read, Update, Delete)** Android menggunakan database MySQL. Ini merupakan hal dasar yang sangat penting dari aplikasi yang memiliki komunikasi dengan database.

Untuk tutorial kali ini kamu yang ingin membuat aplikasi CRUD Android menggunakan Database MySQL setidaknya harus memiliki sedikit pemahaman pemrograman PHP dan memahami konsep dari database, terutamanya MySQL. Bukan sebuah pekerjaan yang mudah untuk memahami konsep komunikasi data antara aplikasi dengan database ini, terutama database MySQL dimana database itu sendiri terletak di Server bukan di perangkat Android itu sendiri.

Agar memudahkan kalian dalam memahami pembuatan aplikasi ini kita akan membuatnya dengan sangat sederhana tanpa harus merubah atau membuat tampilan yang begitu rumit. Kita akan membuat dengan tampilan sederhana agar kalian terfokus pada inti dari aplikasi ini. Setelah kalian memahami konsep dasar atau cara kerja dari aplikasi ini maka kita akan mencoba merubah tampilannya sedikit lebih menarik dengan menambahkan efek *Floating Action Bar* atau yang lainnya. Tetapi **pada tutorial kali ini kita hanya terfokus pada hal intinya tanpa mempedulikan tampilan** (untuk saat ini). Agar tidak lebih menambah kalian bingung dengan penjelasan lainnya, mari kita langsung saja bahas bagaimana konsep pembuat dari aplikasi ini.

Untuk pembuatan **aplikasi CRUD ini kita akan membagi pekerjaan menjadi 3 bagian**, antara lain yaitu:

1. **Pembuatan database pada MySQL.**
2. **Pembuatan File PHP.**
3. **Pembuatan Project Android.**

Hal tersebut kita lakukan agar memudahkan alur berpikir serta tidak membingungkan karena telah kita atur alur pengerjaannya. Sebelum kita mulai mengerjakan projectnya kita perlu menentukan tema aplikasi yang akan kita bangun.

Disini kita akan membuat aplikasi dengan kasus data pegawai, oleh sebab itu mari kita langsung bahas pekerjaan tersebut satu per satu.

1. Pembuatan Database MySQL

Pada tahap ini kalian sangat perlu memperhatikan penamaan dari database, table, dan field yang kalian bangun. Sebelumnya jika kalian belum memahami konsep MySQL alangkah lebih baiknya jika mempelajarinya terlebih dahulu. Tidak harus memiliki kemampuan yang *expert* dalam hal MySQL, kalian hanya perlu memahami bagaimana cara membuat *database, table, dan fieldnya*. Serta sedikit memahami perintah SQL.

Jika kalian telah memahami bagaimana membuat database di MySQL silahkan membuat database dengan nama **db_android** dengan nama table **tb_pegawai** dan struktur tablenya sendiri seperti berikut:

| # | Name | Type | Collation | Attributes | Null | Default | Extra |
|---|---------------|--------------|-------------------|------------|------|---------|----------------|
| 1 | id | int(11) | | | No | None | AUTO_INCREMENT |
| 2 | nama | varchar(100) | latin1_swedish_ci | | No | None | |
| 3 | posisi | varchar(100) | latin1_swedish_ci | | No | None | |
| 4 | gaji | varchar(100) | latin1_swedish_ci | | No | None | |

Setelah kalian membangun databasenya mari kita lanjut kepekerjaan kedua yaitu pembuatan File PHP nya.

2. Pembuatan File PHP

Pembuatan file PHP disini digunakan sebagai web service untuk parsing data aplikasi Android. Pada tahap pembuatan File PHP ini, ada beberapa File yang harus kita buat, berikut ini file php yang perlu kalian buat:

Pertama kita perlu untuk membuat koneksi dengan database, oleh sebab itu kita buat file php dengan nama **koneksi.php** dan tulis programnya seperti di bawah ini:

```
<?php
/*
penulis: Muhammad yusuf
website: http://www.kodingindonesia.com/

Nama Database yang telah dibuat bernama "db_android"
Jika database yang kalian buat tidak sama dengan yang dibuat oleh penulis maka ganti nama database
sesuai dengan nama database yang kalian buat
*/

//Mendefinisikan Konstanta
define('HOST','localhost');
define('USER','root');
define('PASS','');
define('DB','db_android');

//membuat koneksi dengan database
$con = mysqli_connect(HOST,USER,PASS,DB) or die('Unable to Connect');
?>
```

Setelah kita membuat koneksi dengan database. Selanjutnya kita buat file untuk memasukkan data atau Maka mari kita buat file php dengan nama **tambahPgw.php** kemudian Copy Script di bawah ini:

```
<?php

/*

penulis: Muhammad yusuf
website: http://www.kodingindonesia.com/

*/

if($_SERVER['REQUEST_METHOD']=='POST'){

    //Mendapatkan Nilai Variable
    $name = $_POST['name'];
    $desg = $_POST['desg'];
    $sal = $_POST['salary'];

    //Pembuatan Syntax SQL
    $sql = "INSERT INTO tb_pegawai (nama,posisi,gaji) VALUES ('$name','$desg','$sal')";

    //Import File Koneksi database
    require_once('koneksi.php');

    //Eksekusi Query database
    if(mysqli_query($con,$sql)){
        echo 'Berhasil Menambahkan Pegawai';
    }else{
        echo 'Gagal Menambahkan Pegawai';
    }

    mysqli_close($con);
}

?>
```

Setelah kita membuat File untuk memasukkan data atau Create. Maka selanjutnya kita perlu File PHP untuk dapat menampilkan daftar data pegawai yang ada di database (**READ**), oleh sebab itu mari kita buat file php dengan nama **tampilSemuaPgw.php** dan Copy Script di bawah ini:

```

<?php

/*

penulis: Muhammad yusuf
website: http://www.kodingindonesia.com/

*/

//Import File Koneksi Database
require_once('koneksi.php');

//Membuat SQL Query
$sql = "SELECT * FROM tb_pegawai";

//Mendapatkan Hasil
$r = mysqli_query($con,$sql);

//Membuat Array Kosong
$result = array();

while($row = mysqli_fetch_array($r)){

    //Memasukkan Nama dan ID kedalam Array Kosong yang telah dibuat
    array_push($result,array(
        "id"=>$row['id'],
        "name"=>$row['nama']
    ));
}

//Menampilkan Array dalam Format JSON
echo json_encode(array('result'=>$result));

mysqli_close($con);
?>

```

Selain kita butuh untuk menampilkan semua data pegawai kita juga perlu untuk menampilkan detail pegawai yang kita pilih atau kita *select (READ)*. Oleh sebab itu mari kita buat file php dengan nama **tampilPgw.php** dan Copy Script di bawah ini:

```

<?php
    /*
    penulis: Muhammad yusuf
    website: http://www.kodingindonesia.com/
    */

    //Mendapatkan Nilai Dari Variable ID Pegawai yang ingin ditampilkan
    $id = $_GET['id'];

    //Importing database
    require_once('koneksi.php');

    //Membuat SQL Query dengan pegawai yang ditentukan secara spesifik sesuai ID
    $sql = "SELECT * FROM tb_pegawai WHERE id=$id";

    //Mendapatkan Hasil
    $r = mysqli_query($con, $sql);

    //Memasukkan Hasil Kedalam Array
    $result = array();
    $row = mysqli_fetch_array($r);
    array_push($result, array(
        "id"=>$row['id'],
        "name"=>$row['nama'],
        "desg"=>$row['posisi'],
        "salary"=>$row['gaji']
    ));

    //Menampilkan dalam format JSON
    echo json_encode(array('result'=>$result));

    mysqli_close($con);
?>

```

Sampai tahap ini kita telah membuat file PHP untuk **Create, Reade**. Selanjutnya mari kita buat file php untuk **Update**. Dalam kasus ini untuk Update data pegawai, maka mari kita buat file php dengan nama **updatePgw.php** dan Copy Script di bawah ini:

```

<?php

```

```

/*

penulis: Muhammad yusuf
website: http://www.kodingindonesia.com/

*/
if($_SERVER['REQUEST_METHOD']=='POST'){
    //Mendapatkan Nilai Dari Variable
    $id = $_POST['id'];
    $name = $_POST['name'];
    $desg = $_POST['desg'];
    $sal = $_POST['salary'];

    //import file koneksi database
    require_once('koneksi.php');

    //Membuat SQL Query
    $sql = "UPDATE tb_pegawai SET nama = '$name', posisi = '$desg', gaji = '$sal' WHERE id = $id;";

    //Meng-update Database
    if(mysqli_query($con,$sql)){
        echo 'Berhasil Update Data Pegawai';
    }else{
        echo 'Gagal Update Data Pegawai';
    }

    mysqli_close($con);
}
?>

```

Yang terakhir mari kita buat file PHP untuk **Delete**. Dalam kasus ini kita akan membuat file php untuk delete data pegawai, silahkan buat file php dengan nama **hapusPgw.php** dan Copy Script di bawah ini:

```
<?php
```

```
/*
```

```

penulis: Muhammad yusuf
website: http://www.kodingindonesia.com/

*/

//Mendapatkan Nilai ID
$id = $_GET['id'];

//Import File Koneksi Database
require_once('koneksi.php');

//Membuat SQL Query
$sql = "DELETE FROM tb_pegawai WHERE id=$id;";

//Menghapus Nilai pada Database
if(mysqli_query($con,$sql)){
echo 'Berhasil Menghapus Pegawai';
}else{
echo 'Gagal Menghapus Pegawai';
}

mysqli_close($con);
?>

```

Kita telah membuat File PHP untuk masing-masing proses **Create, Read, Update, Delete**. Maka pekerjaan kita untuk membuat file php telah selesai. File PHP yang telah kita buat tersebut simpan dalam satu folder bernama **pegawai**. Karena disini penulis menggunakan **XAMPP** maka directory untuk penyimpanan file php tersebut terdapat di HTDOC, adapun directory penyimpanannya seperti berikut: **(C:\xampp\htdocs\Android\pegawai)**. Untuk mengakses di Android sendiri kita perlu mengetahui **IP dari Komputer kita**, agar kita bisa mengetahui berapa IP komputer kita mungkin kita bisa gunakan perintah di CMD dengan perintah **ipconfig** maka akan tampil berapa IP kita seperti contoh berikut ini:

```

Connection-specific DNS Suffix . : local
Link-local IPv6 Address . . . . . : fe80::c8b8:9512:df02:3fbf%3
IPv4 Address. . . . . : 192.168.1.9
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1

```

Pada contoh di atas IP yang tertera adalah 192.168.1.9 maka path untuk script kita nanti di Android Studio adalah

http://192.168.1.9/Android/pegawai/nama_file.php

Bagaimana? Apakah sudah pusing? Hehehehe.... Untuk saran, kalian ikuti saja terlebih dahulu sesuai dengan yang ada. Kalian ikuti dari mulai nama database, table, field, dan juga nama file-file serta nama variable nya. Setelah itu tinggal kalian jalankan programnya dan lihat serta pahami apa yang terjadi, ketika kalian mulai paham maka kalian coba untuk edit-edit sedikit.

Agar tidak lama-lama membaca prolog mari kita langsung ke langkah selanjutnya yaitu membuat Project **Android Studio**.

3. Pembuatan Project Pada Android Studio

Pada pekerjaan ini merupakan pekerjaan intinya, karena memang kita membuat aplikasi Android ya. Silahkan kalian buat Project baru pada Android Studio, dicontoh ini menggunakan nama **MyCRUD** pada project yang dibuat.

Jika project baru telah terbuka maka langkah pertama yang perlu kalian lakukan yaitu silahkan buka File Manifest dan tambah kan perintah berikut:

```
1 <uses-permission android:name="android.permission.INTERNET" />
```

Perintah di atas digunakan untuk ijin internet meskipun project yang akan kita buat nanti adalah bersifat Local tetap perlu untuk menambahkan perintah tersebut. Untuk peletakan script di atas kalian letakkan seperti gambar berikut berikut:


```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.kodingindonesia.mycrud">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="MyCRUD"
        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>
        <activity android:name=".TampilSemuaPgw" />
        <activity android:name=".ListView" />
        <activity android:name=".TampilPegawai"></activity>
    </application>

</manifest>

```

Sekarang silahkan buat Java Class baru di dalam package yang ada dengan nama **konfigurasi**. Dan kalian ikuti kode programnya sebagai berikut:

Sebagai catatan jangan lupa perhatikan bagian URL (Pemanggilan file PHP yang ada di Server).

Sesuaikan Dengan IP kalian masing-masing

```

package com.kodingindonesia.mycrud;

/**
 * Created by muhammadyusuf on 01/19/2017.
 * kodingindonesia
 */

public class konfigurasi {

```

```

//Dibawah ini merupakan Pengalamatan dimana Lokasi Skrip CRUD PHP disimpan
//Pada tutorial Kali ini, karena kita membuat localhost maka alamatnya tertuju ke IP komputer
//dimana File PHP tersebut berada
//PENTING! JANGAN LUPA GANTI IP SESUAI DENGAN IP KOMPUTER DIMANA DATA PHP BERADA
public static final String URL_ADD="http://192.168.1.9/Android/pegawai/tambahPgw.php";
public static final String URL_GET_ALL = "http://192.168.1.9/Android/pegawai/tampilSemuaPgw.php";
public static final String URL_GET_EMP = "http://192.168.1.9/Android/pegawai/tampilPgw.php?id=";
public static final String URL_UPDATE_EMP = "http://192.168.1.9/Android/pegawai/updatePgw.php";
public static final String URL_DELETE_EMP = "http://192.168.1.9/Android/pegawai/hapusPgw.php?id=";

//Dibawah ini merupakan Kunci yang akan digunakan untuk mengirim permintaan ke Skrip PHP
public static final String KEY_EMP_ID = "id";
public static final String KEY_EMP_NAMA = "name";
public static final String KEY_EMP_POSISI = "desg"; //desg itu variabel untuk posisi
public static final String KEY_EMP_GAJIH = "salary"; //salary itu variabel untuk gaji

//JSON Tags
public static final String TAG_JSON_ARRAY="result";
public static final String TAG_ID = "id";
public static final String TAG_NAMA = "name";
public static final String TAG_POSISI = "desg";
public static final String TAG_GAJIH = "salary";

//ID karyawan
//emp itu singkatan dari Employee
public static final String EMP_ID = "emp_id";
}

```

Setelah selesai membuat java class konfigurasi, selanjutnya silahkan buat java class kembali dengan nama **RequestHandler**. Java Class ini kita gunakan untuk menangani permintaan jaringan. Silahkan ikuti kode program untuk java class **RequestHandler** seperti di bawah ini:

```

package com.kodingindonesia.mycrud;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;

```

```
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.HashMap;
import java.util.Map;

import javax.net.ssl.HttpURLConnection;

/**
 * Created by muhammadyusuf on 01/19/2017.
 * kodingindonesia
 */

public class RequestHandler {
    //Metode Untuk mengirim httpPostRequest
    //Metode ini mengambil 2 Argumen
    //Metode Pertama adalah URL dari Skrip yang digunakan untuk mengirimkan permintaan
    //Yang lainnya adalah HashMap dengan nilai pasangan nama yang berisi data yang akan dikirim dengan permintaan
    public String sendPostRequest(String requestURL,
                                   HashMap<String, String> postDataParams) {

        //Membuat URL
        URL url;

        //Objek StringBuilder untuk menyimpan pesan diambil dari server
        StringBuilder sb = new StringBuilder();
        try {
            //Inisialisasi URL
            url = new URL(requestURL);

            //Membuat Koneksi HttpURLConnection
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();

            //Konfigurasi koneksi
            conn.setReadTimeout(15000);
            conn.setConnectTimeout(15000);
            conn.setRequestMethod("POST");
            conn.setDoInput(true);
```

```

conn.setDoOutput(true);

//Membuat Keluaran Stream
OutputStream os = conn.getOutputStream();

//Menulis Parameter Untuk Permintaan
//Kita menggunakan metode postDataString yang didefinisikan di bawah ini
BufferedWriter writer = new BufferedWriter(
    new OutputStreamWriter(os, "UTF-8"));
writer.write(postDataString(postDataParams));

writer.flush();
writer.close();
os.close();
int responseCode = conn.getResponseCode();

if (responseCode == HttpURLConnection.HTTP_OK) {

    BufferedReader br = new BufferedReader(new InputStreamReader(conn.getInputStream()));
    sb = new StringBuilder();
    String response;
    //Reading server response
    while ((response = br.readLine()) != null){
        sb.append(response);
    }
}

} catch (Exception e) {
    e.printStackTrace();
}
return sb.toString();
}

public String sendGetRequest(String requestURL){
    StringBuilder sb = new StringBuilder();
    try {
        URL url = new URL(requestURL);
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(con.getInputStream()));
    }
}

```

```

        String s;
        while((s=bufferedReader.readLine())!=null){
            sb.append(s+"\n");
        }
    }catch(Exception e){
    }
    return sb.toString();
}

public String sendGetRequestParam(String requestURL, String id){
    StringBuilder sb =new StringBuilder();
    try {
        URL url = new URL(requestURL+id);
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(con.getInputStream()));

        String s;
        while((s=bufferedReader.readLine())!=null){
            sb.append(s+"\n");
        }
    }catch(Exception e){
    }
    return sb.toString();
}

private String getPostDataString(HashMap<String, String> params) throws UnsupportedEncodingException {
    StringBuilder result = new StringBuilder();
    boolean first = true;
    for (Map.Entry<String, String> entry : params.entrySet()) {
        if (first)
            first = false;
        else
            result.append("&");

        result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
    }
}

```

```
        return result.toString();
    }
}
```

Selanjutnya mari kita rubah tampilan dari **activity_main.xml** yang telah terbentuk sejak project pertama kali dibangun. Pada file xml ini kita akan gunakan sebagai tampilan menambah data pegawai (**ADD**). selain xml ini kita nantinya akan membuat tampilan lain atau file xml lainnya untuk menampilkan data semua pegawai dan untuk menampilkan data dari pegawai yang di pilih/select. Untuk file xml **activity_main.xml** silahkan ikuti kode programnya seperti berikut:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama Pegawai" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editTextName" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Posisi" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:id="@+id/editTextDesg" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Gajih" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editTextSalary" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Tambah Pegawai"
    android:id="@+id/buttonAdd" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Daftar Pegawai"
    android:id="@+id/buttonView" />

</LinearLayout>
```

Dari kode program xml di atas akan menghasilkan tampilan seperti berikut:

Selanjutnya silahkan ikuti kode program untuk **MainActivity.java** seperti berikut ini:

```
package com.kodingindonesia.mycrud;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```



```
import java.util.HashMap;

/**
 * Created by muhammadyusuf on 01/19/2017.
 * kodingindonesia
 */

public class MainActivity extends AppCompatActivity implements View.OnClickListener{

    //Dibawah ini merupakan perintah untuk mendefinikan View
    private EditText editTextName;
    private EditText editTextDesg;
    private EditText editTextSal;

    private Button buttonAdd;
    private Button buttonView;

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

        //Inisialisasi dari View
        editTextName = (EditText) findViewById(R.id.editTextName);
        editTextDesg = (EditText) findViewById(R.id.editTextDesg);
        editTextSal = (EditText) findViewById(R.id.editTextSalary);

        buttonAdd = (Button) findViewById(R.id.buttonAdd);
        buttonView = (Button) findViewById(R.id.buttonView);

        //Setting listeners to button
        buttonAdd.setOnClickListener(this);
        buttonView.setOnClickListener(this);
    }

    //Dibawah ini merupakan perintah untuk Menambahkan Pegawai (CREATE)
    private void addEmployee(){
```

```

final String name = editTextName.getText().toString().trim();
final String desg = editTextDesg.getText().toString().trim();
final String sal = editTextSal.getText().toString().trim();

class AddEmployee extends AsyncTask<Void,Void,String>{

    ProgressDialog loading;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        loading = ProgressDialog.show(MainActivity.this, "Menambahkan...", "Tunggu...", false, false);
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        loading.dismiss();
        Toast.makeText(MainActivity.this, s, Toast.LENGTH_LONG).show();
    }

    @Override
    protected String doInBackground(Void... v) {
        HashMap<String,String> params = new HashMap<>();
        params.put(konfigurasi.KEY_EMP_NAMA, name);
        params.put(konfigurasi.KEY_EMP_POSISI, desg);
        params.put(konfigurasi.KEY_EMP_GAJIH, sal);

        RequestHandler rh = new RequestHandler();
        String res = rh.sendPostRequest(konfigurasi.URL_ADD, params);
        return res;
    }
}

AddEmployee ae = new AddEmployee();
ae.execute();

@Override

```

```

public void onClick(View v) {
    if(v == buttonAdd){
        addEmployee();
    }

    if(v == buttonView){
        startActivity(new Intent(this,TampilSemuaPgw.class));
    }
}
}

```

Selanjutnya silahkan kalian buat file xml baru dengan nama **TampilanSemuaPgw** maka secara otomatis namanya akan menjadi **activity_tampil_semua_pgw.xml**. File xml ini digunakan untuk menampilkan semua data pegawai. Setelah file xml telah terbentuk selanjutnya silahkan ikuti kode program seperti berikut:

```

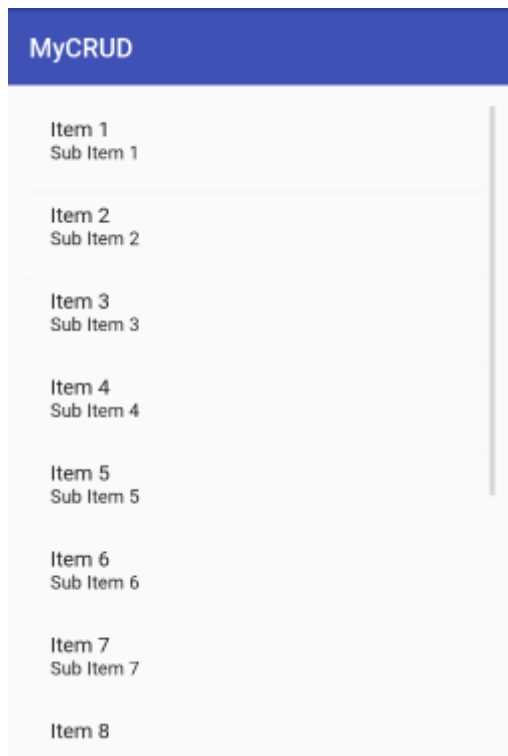
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context="com.kodingindonesia.mycrud.TampilSemuaPgw">

    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/listView" />

</LinearLayout>

```

Dari kode program xml di atas akan menghasilkan tampilan seperti berikut:



Karena kita membuat ListView, maka kita perlu untuk membuat lagi satu Layout atau tampilan xml untuk menampilkan ListVeiw tersebut. Silahkan buat file xml dengan nama **list_item.xml** dan ikuti kode program di bawah ini:

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

    <TextView
        android:id="@+id/id"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
```

```
<TextView
    android:id="@+id/name"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

</LinearLayout>
```

Setelah kita membuat file xml **activity_tampil_semua_pgw.xml** secara langsung file Java Class baru akan terbentuk dengan nama **tampilSemuaPgw.java** silahkan kalian salin kode program dibawah berikut kedalam Java Class **tampilSemuaPgw.java**

```
package com.kodingindonesia.mycrud;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.SimpleAdapter;

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

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

/**
 * Created by muhammadyusuf on 01/19/2017.
 * kodingindonesia
 */

public class TampilSemuaPgw extends AppCompatActivity implements ListView.OnItemClickListener{
```

```

private ListView listView;

private String JSON_STRING;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tampil_semua_pgw);
    listView = (ListView) findViewById(R.id.listView);
    listView.setOnItemClickListener(this);
    getJSON();
}

private void showEmployee(){
    JSONObject jsonObject = null;
    ArrayList<HashMap<String,String>> list = new ArrayList<HashMap<String, String>>();
    try {
        jsonObject = new JSONObject(JSON_STRING);
        JSONArray result = jsonObject.getJSONArray(konfigurasi.TAG_JSON_ARRAY);

        for(int i = 0; i<result.length(); i++){
            JSONObject jo = result.getJSONObject(i);
            String id = jo.getString(konfigurasi.TAG_ID);
            String name = jo.getString(konfigurasi.TAG_NAMA);

            HashMap<String,String> employees = new HashMap<>();
            employees.put(konfigurasi.TAG_ID,id);
            employees.put(konfigurasi.TAG_NAMA,name);
            list.add(employees);
        }

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

    ListAdapter adapter = new SimpleAdapter(
        TampilSemuaPgw.this, list, R.layout.list_item,
        new String[]{konfigurasi.TAG_ID,konfigurasi.TAG_NAMA},

```

```

        new int[] {R.id.id, R.id.name});

listView.setAdapter(adapter);
}

private void getJSON() {
    class GetJSON extends AsyncTask<Void,Void,String>{

        ProgressDialog loading;
        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            loading = ProgressDialog.show(TampilSemuaPgw.this, "Mengambil Data", "Mohon Tunggu...", false, false);
        }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            loading.dismiss();
            JSON_STRING = s;
            showEmployee();
        }

        @Override
        protected String doInBackground(Void... params) {
            RequestHandler rh = new RequestHandler();
            String s = rh.sendGetRequest(konfigurasi.URL_GET_ALL);
            return s;
        }
    }
    GetJSON gj = new GetJSON();
    gj.execute();
}

@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    Intent intent = new Intent(this, TampilPegawai.class);
    HashMap<String,String> map = (HashMap)parent.getItemAtPosition(position);
    String empId = map.get(konfigurasi.TAG_ID).toString();

```

```

        intent.putExtra(konfigurasi.EMP_ID, empId);
        startActivity(intent);
    }
}

```

Selanjutnya kita buat file xml untuk menampilkan data pegawai yang dipilih/select dengan nama tampil **tampilPegawai**, dari nama tersebut akan secara otomatis terbentuk nama untuk xml dengan nama **activity_tampil_pegawai.xml** dan selanjutnya salin kode program berikut:

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ID Pegawai" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editTextId" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama Pegawai" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editTextName" />

    <TextView
        android:layout_width="wrap_content"

```



```
        android:layout_height="wrap_content"
        android:text="Posisi" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editTextDesg" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Salary" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editTextSalary" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Update Pegawai"
    android:id="@+id/buttonUpdate" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Hapus Pegawai"
    android:id="@+id/buttonDelete" />

</LinearLayout>
```

Dari kode program di atas akan menghasilkan tampilan seperti berikut:

MyCRUD

ID Pegawai

Nama Pegawai

Posisi

Salary

UPDATE PEGAWAI

HAPUS PEGAWAI

Dari file xml yang kita buat sebelumnya akan menghasilkan file Java Class dengan nama **tampilPegawai.java** selanjutnya silahkan salin kode program di bawah ini kedalam file java class **tampilPegawai.java**

```
package com.kodingindonesia.mycrud;

import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

```
import android.widget.Toast;

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

import java.util.HashMap;

/**
 * Created by muhammadyusuf on 01/19/2017.
 * kodingindonesia
 */

public class TampilPegawai extends AppCompatActivity implements View.OnClickListener{

    private EditText editTextId;
    private EditText editTextName;
    private EditText editTextDesg;
    private EditText editTextSalary;

    private Button buttonUpdate;
    private Button buttonDelete;

    private String id;

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

        Intent intent = getIntent();

        id = intent.getStringExtra(konfigurasi.EMP_ID);

        editTextId = (EditText) findViewById(R.id.editTextId);
        editTextName = (EditText) findViewById(R.id.editTextName);
        editTextDesg = (EditText) findViewById(R.id.editTextDesg);
        editTextSalary = (EditText) findViewById(R.id.editTextSalary);
    }
}
```

```

        buttonUpdate = (Button) findViewById(R.id.buttonUpdate);
        buttonDelete = (Button) findViewById(R.id.buttonDelete);

        buttonUpdate.setOnClickListener(this);
        buttonDelete.setOnClickListener(this);

        editTextId.setText(id);

        getEmployee();
    }

    private void getEmployee(){
        class GetEmployee extends AsyncTask<Void,Void,String>{
            ProgressDialog loading;
            @Override
            protected void onPreExecute() {
                super.onPreExecute();
                loading = ProgressDialog.show(TampilPegawai.this, "Fetching...", "Wait...", false, false);
            }

            @Override
            protected void onPostExecute(String s) {
                super.onPostExecute(s);
                loading.dismiss();
                showEmployee(s);
            }

            @Override
            protected String doInBackground(Void... params) {
                RequestHandler rh = new RequestHandler();
                String s = rh.sendGetRequestParam(konfigurasi.URL_GET_EMP,id);
                return s;
            }
        }
        GetEmployee ge = new GetEmployee();
        ge.execute();
    }

    private void showEmployee(String json){

```

```

try {
    JSONObject jsonObject = new JSONObject(json);
    JSONArray result = jsonObject.getJSONArray(konfigurasi.TAG_JSON_ARRAY);
    JSONObject c = result.getJSONObject(0);
    String name = c.getString(konfigurasi.TAG_NAMA);
    String desg = c.getString(konfigurasi.TAG_POSISI);
    String sal = c.getString(konfigurasi.TAG_GAJIH);

    editTextName.setText(name);
    editTextDesg.setText(desg);
    editTextSalary.setText(sal);

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

private void updateEmployee() {
    final String name = editTextName.getText().toString().trim();
    final String desg = editTextDesg.getText().toString().trim();
    final String salary = editTextSalary.getText().toString().trim();

    class UpdateEmployee extends AsyncTask<Void,Void,String>{
        ProgressDialog loading;
        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            loading = ProgressDialog.show(TampilPegawai.this, "Updating...", "Wait...", false, false);
        }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            loading.dismiss();
            Toast.makeText(TampilPegawai.this, s, Toast.LENGTH_LONG).show();
        }

        @Override

```

```

protected String doInBackground(Void... params) {
    HashMap<String,String> hashMap = new HashMap<>();
    hashMap.put(konfigurasi.KEY_EMP_ID,id);
    hashMap.put(konfigurasi.KEY_EMP_NAMA,name);
    hashMap.put(konfigurasi.KEY_EMP_POSISI,desg);
    hashMap.put(konfigurasi.KEY_EMP_GAJIH,salary);

    RequestHandler rh = new RequestHandler();

    String s = rh.sendPostRequest(konfigurasi.URL_UPDATE_EMP,hashMap);

    return s;
}

UpdateEmployee ue = new UpdateEmployee();
ue.execute();
}

private void deleteEmployee(){
    class DeleteEmployee extends AsyncTask<Void,Void,String> {
        ProgressDialog loading;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            loading = ProgressDialog.show(TampilPegawai.this, "Updating...", "Tunggu...", false, false);
        }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            loading.dismiss();
            Toast.makeText(TampilPegawai.this, s, Toast.LENGTH_LONG).show();
        }

        @Override
        protected String doInBackground(Void... params) {
            RequestHandler rh = new RequestHandler();

```

```

        String s = rh.sendGetRequestParam(konfigurasi.URL_DELETE_EMP, id);
        return s;
    }
}

DeleteEmployee de = new DeleteEmployee();
de.execute();
}

private void confirmDeleteEmployee(){
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
    alertDialogBuilder.setMessage("Apakah Kamu Yakin Ingin Menghapus Pegawai ini?");

    alertDialogBuilder.setPositiveButton("Ya",
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface arg0, int arg1) {
                deleteEmployee();
                startActivity(new Intent(TampilPegawai.this, TampilSemuaPgw.class));
            }
        });

    alertDialogBuilder.setNegativeButton("Tidak",
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface arg0, int arg1) {

            }
        });

    AlertDialog alertDialog = alertDialogBuilder.create();
    alertDialog.show();
}

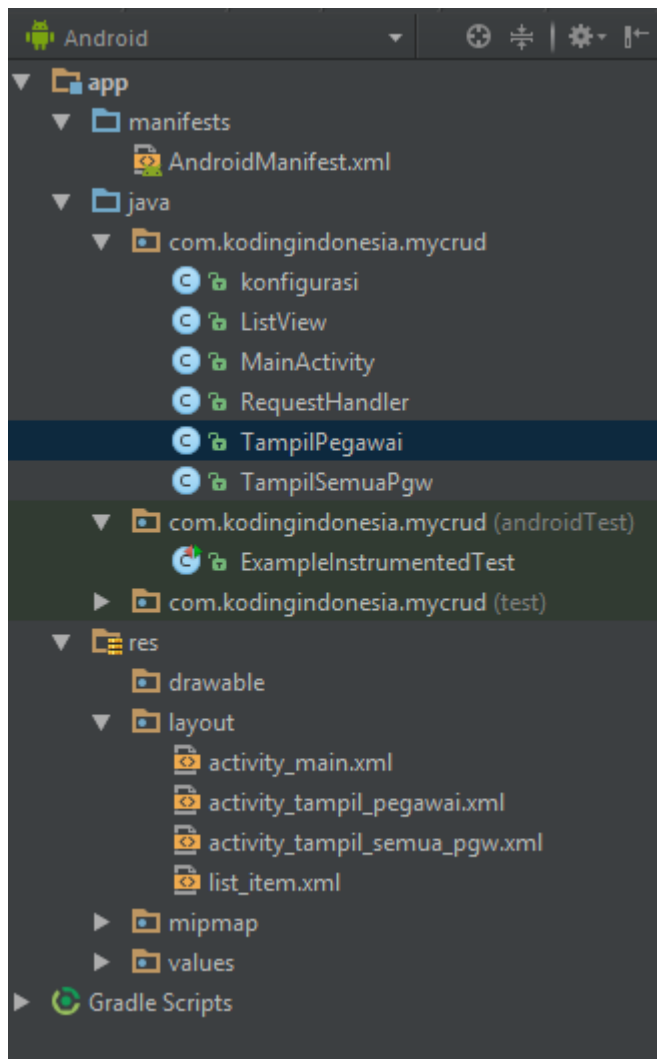
@Override
public void onClick(View v) {
    if(v == buttonUpdate){
        updateEmployee();
    }
}

```

```
        if(v == buttonDelete){  
            confirmDeleteEmployee();  
        }  
    }  
}
```

Setelah kalian selesai mengikuti langkah-langkah di atas semua maka program Android siap dijalankan di emulator atau dismartphone Android langsung.

Jika masih ada yang error silahkan Cek kembali **File Java Class** dan **xml** nya sudah benar atau belum. dari project Android yang telah kita buat akan menciptakan susunan file **Java Class**serta **xml** seperti gambar berikut:



Penting:

Jangan lupa untuk menghidupkan **Apache** dan **MySQL** pada **xampp** atau **wamp** atau web service lainnya.