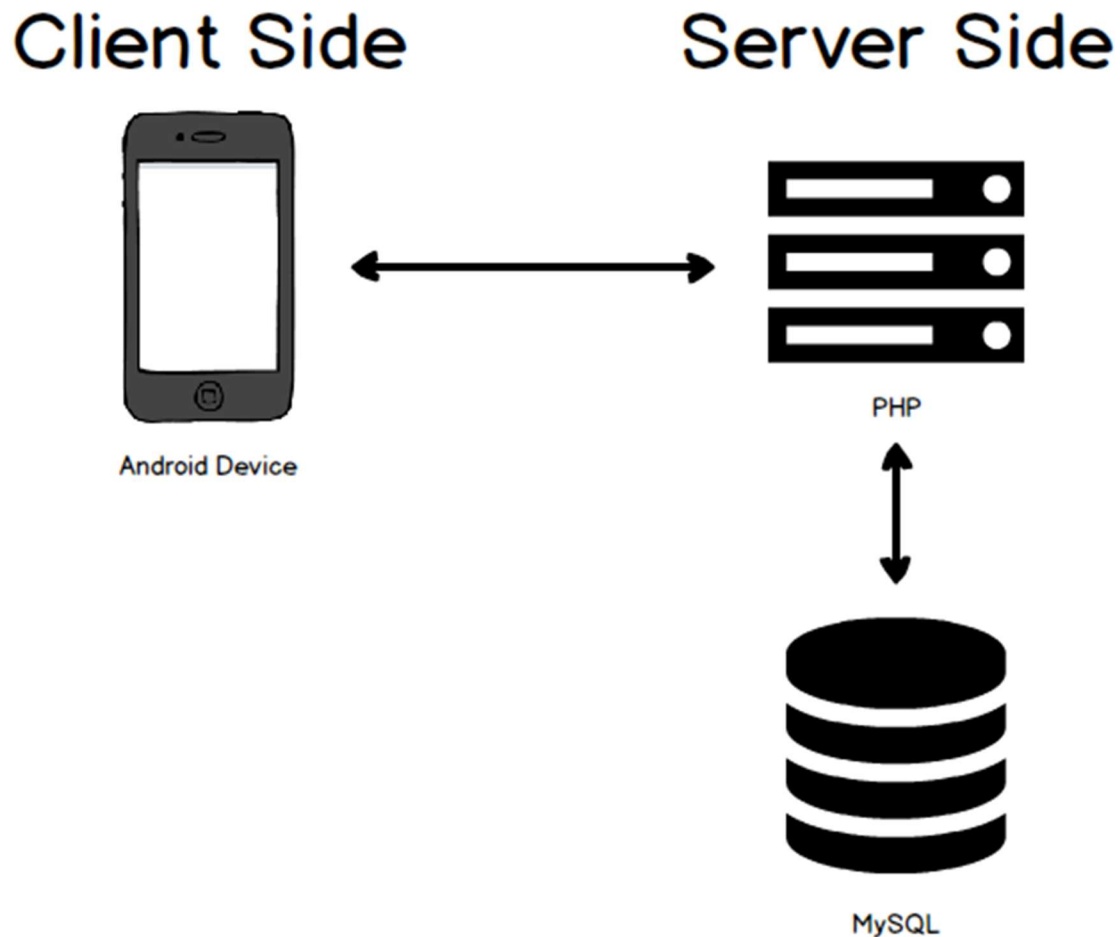


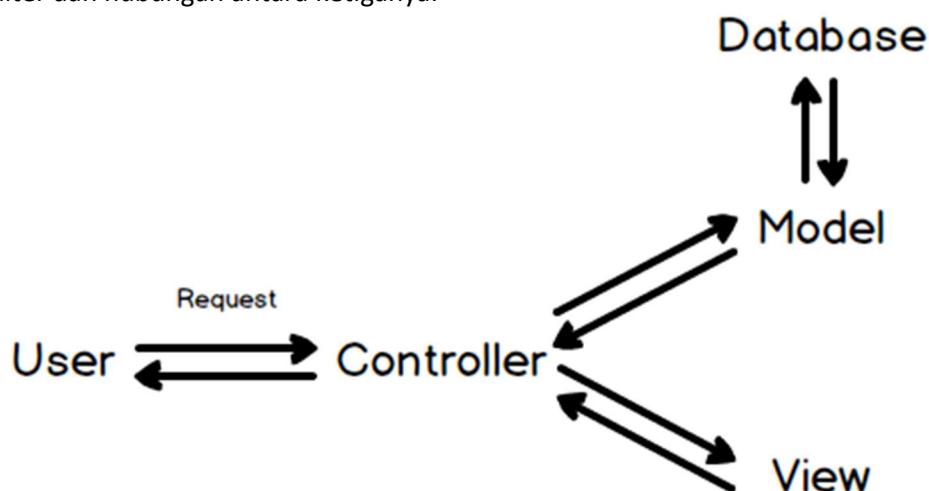
Cara Cepat Menghubungkan Android dengan MySQL dan PHP CodeIgniter

Pada bagian ini, kita akan membuat aplikasi yang dapat dikategorikan sebagai aplikasi *client server database*. Android sebagai *client* akan berhubungan dengan MySQL server dengan perantara PHP, yang dalam hal ini kita menggunakan *framework* CodeIgniter. Konsep dasarnya sebagai berikut :



Gambar 1. Konsep kerja aplikasi Android *client server database*

CodeIgniter sendiri adalah *framework* PHP yang menggunakan sistem *Model View Controller* (MVC). Secara garis besar ada 3 komponen utama, yaitu Model, View, dan Controller. Beginilah gambaran sederhana MVC yang diterapkan CodeIgniter dan hubungan antara ketiganya.



Gambar 2. Konsep hubungan MVC

Babak 1 : Pembuatan Database

Kita akan membuat sebuah database di MySQL dengan perintah berikut :

```
create database contact;
```

Setelah itu kita buat tabel *contact* dengan perintah berikut.

```
create table contact(  
    id int not null AUTO_INCREMENT,  
    prefix varchar(5) not null,  
    first_name varchar(15) not null,  
    last_name varchar(15) not null,  
    email varchar(30) not null,  
    phone varchar(13) not null,  
    PRIMARY KEY (id)  
);
```

Untuk isi data awal, kita isikan dengan perintah berikut ini.

```
INSERT INTO contact SET prefix='Mr.', first_name='Go Frendi', last_name='Gunawan',  
email='frendi@stiki.ac.id', phone='089680868343';
```

```
INSERT INTO contact SET prefix='Mrs.', first_name='Chaulina', last_name='Alfianti',  
email='chaulina@stiki.ac.id', phone='081805199666';
```

Cek dengan menggunakan perintah select berikut ini :

```
MariaDB [contact]> select * from contact;
```

id	prefix	first_name	last_name	email	phone
1	Mr.	Go Frendi	Gunawan	frendi@stiki.ac.id	089680868343
2	Mrs.	Chaulina	Alfianti	chaulina@stiki.ac.id	081805199666

2 rows in set (0.00 sec)

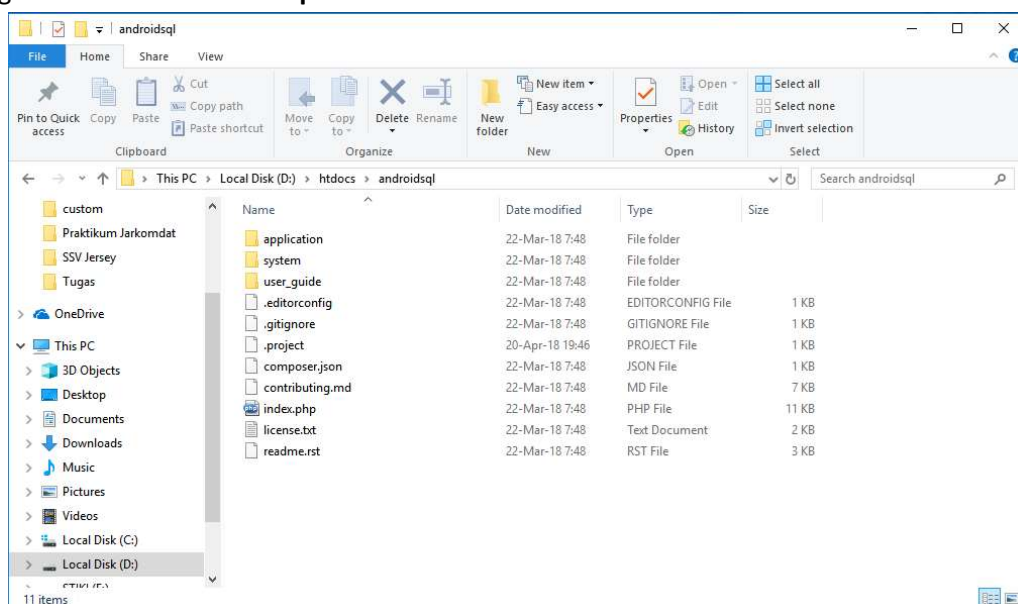
Gambar 3. Hasil *pull* database MySQL melalui *command prompt*

Babak 2 : Pembuatan *Web Services Controller* dengan PHP CodeIgniter

Bagian 1 : Instalasi CodeIgniter

Kita akan menggunakan CodeIgniter 3.1.8. Silahkan mengunduh dari situs resminya www.codeigniter.com.

Sebelumnya kita harus menyiapkan terlebih dahulu direktori *root project* yang akan kita gunakan. Misalnya dalam contoh ini kita gunakan folder **androidsql**.



Gambar 4. Isi direktori *root project* yang telah diinstal CodeIgniter

Bagian 2 : Konfigurasi Database

Langkah awal yang akan kita lakukan adalah konfigurasi database. Buka direktori **application/config** dan buka file **database.php**. *Edit* sehingga menjadi seperti ini :

```
1. $db['default'] = array(
2.     'dsn' => '',
3.     'hostname' => 'localhost',
4.     'username' => 'root',
5.     'password' => '',
6.     'database' => 'contact',
7.     'dbdriver' => 'mysqli',
8.     'dbprefix' => '',
9.     'pconnect' => FALSE,
10.    'db_debug' => (ENVIRONMENT !== 'production'),
11.    'cache_on' => FALSE,
12.    'cachedir' => '',
13.    'char_set' => 'utf8',
14.    'dbcollat' => 'utf8_general_ci',
15.    'swap_pre' => '',
16.    'encrypt' => FALSE,
17.    'compress' => FALSE,
18.    'stricton' => FALSE,
19.    'failover' => array(),
20.    'save_queries' => TRUE
21.);
```

Kemudian di direktori yang sama buka file **autoload.php** dan *edit* baris berikut :

```
1. $autoload['libraries'] = array();
```

menjadi seperti berikut ini :

```
1. $autoload['libraries'] = array('database');
```

Bagian 3 : Menghilangkan index.php

Kita akan menghilangkan index.php pada URL yang digunakan untuk mengakses halaman, sehingga kelihatan “lebih enak dipandang” tanpa mengganggu fungsi *controller* dan *function*-nya.

Buka file **config.php** pada direktori **application/config** dan ubah bagian berikut ini dengan menghilangkan **index.php** pada *script* ini

```
1. $config['index_page'] = 'index.php';
```

sehingga menjadi :

```
1. $config['index_page'] = '';
```

Lalu kita akan membuat sebuah file **.htaccess** yang disimpan di direktori *root project* kita (posisinya sejajar dengan direktori *application*, *system*, dan sebagainya). Isi file **.htaccess** adalah sebagai berikut :

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L]
```

Kita akan coba akses salah satu halaman, misalnya **http://localhost/androidsql/contact**. Jika tidak ada *error*, berarti berhasil.

Bagian 4 : Model

Setelahnya kita akan membuat sebuah model baru di *project* kita, kita beri nama misalnya **M_data**, lalu kita simpan di direktori **application/models**. *Source code*-nya sebagai berikut :

```

1. <?php
2.
3. class M_data extends CI_Model{
4.     function insert($table, $data){
5.         $this->db->insert($table, $data);
6.     }
7.     function where($column, $condition){
8.         $this->db->where($column, $condition);
9.         return $this;
10.    }
11.    function update($table, $data){
12.        $this->db->update($table, $data);
13.    }
14.    function delete($table){
15.        $this->db->delete($table);
16.    }
17.    function getAll($table){
18.        return $this->db->get($table)->result();
19.    }
20.    function get($table){
21.        return $this->db->get($table)->row();
22.    }
23. }
24. ?>

```

Bagian 5 : Controller

Kita akan membuat *controller* untuk meng-*handle* CRUD dari *database* **contact**, seperti berikut ini.

```

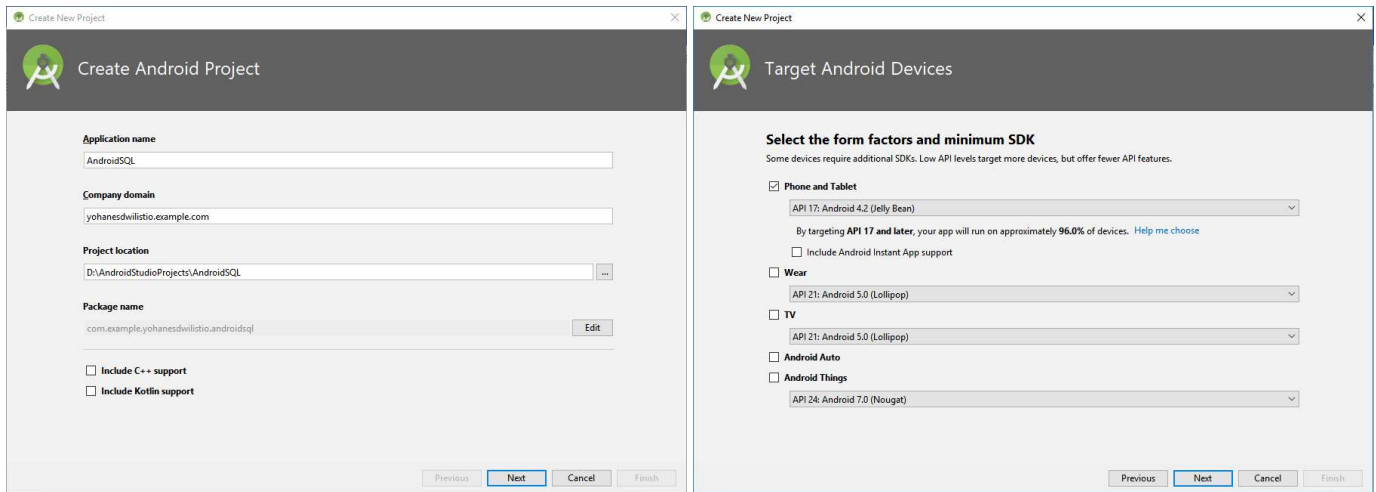
1. <?php
2. defined('BASEPATH') OR exit('No direct script access allowed');
3.
4. class Contact extends CI_Controller{
5.     function __construct(){
6.         parent::__construct();
7.         $this->load->model('m_data');
8.     }
9.     function index(){
10.        $data = $this->m_data->getAll('contact');
11.        echo json_encode($data);
12.    }
13.    function insert(){
14.        $salutation = $this->input->post('salutation');
15.        $first_name = $this->input->post('firstname');
16.        $lastname = $this->input->post('lastname');
17.        $email = $this->input->post('email');
18.        $phone = $this->input->post('phone');
19.        $data = array('prefix' => $salutation, 'first_name' => $first_name, 'last_name' => $lastname, 'email' => $email, 'phone' => $phone);
20.    };
21.    $this->m_data->insert('contact', $data);
22.    echo "success";
23. }
24. function update($id){
25.     $salutation = $this->input->post('salutation');
26.     $first_name = $this->input->post('firstname');
27.     $lastname = $this->input->post('lastname');
28.     $email = $this->input->post('email');
29.     $phone = $this->input->post('phone');
30.     $data = array('prefix' => $salutation, 'first_name' => $first_name, 'last_name' => $lastname, 'email' => $email, 'phone' => $phone);
31. };
32. $this->m_data->where('id', $id)->update('contact', $data);
33. echo "success";
34. }
35. function delete($id){
36.     $this->m_data->where('id', $id)->delete('contact');
37.     echo "success";
38. }
39. }
40. ?>

```

Babak 3 : Tampilan Android dan Java untuk Menampilkan Data

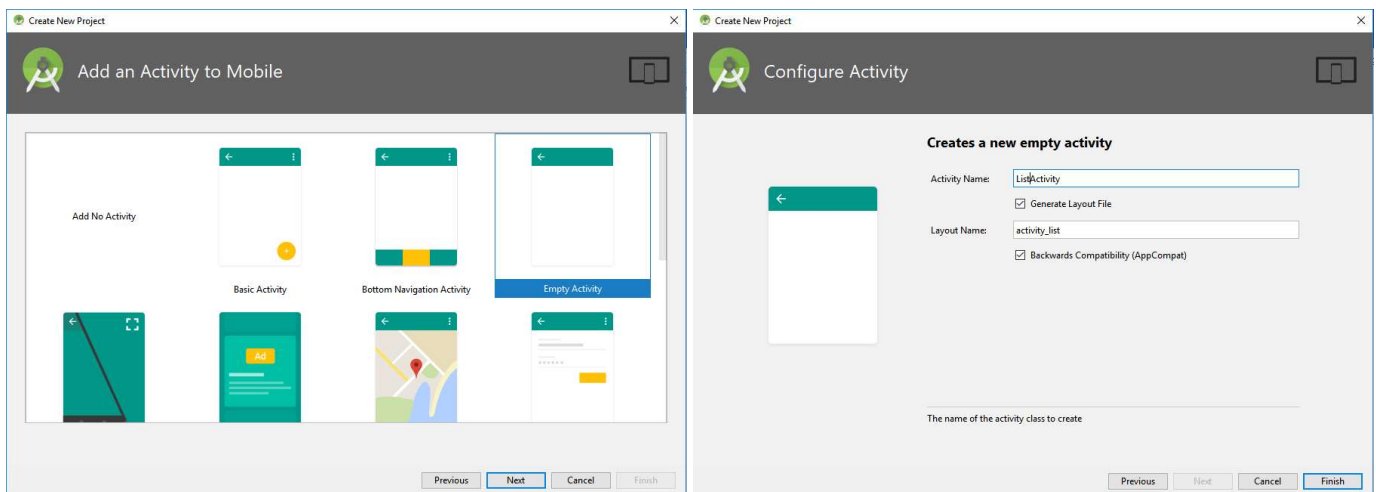
Bagian 1 : Tampilan

Pada bagian ini kita akan membuat sebuah project Android Studio yang baru, kita beri nama AndroidSQL dengan minimum SDK adalah API 17 (Android 4.2 Jelly Bean).



Gambar 5. Persiapan pembuatan *project* Android (memberi nama aplikasi dan minimum SDK)

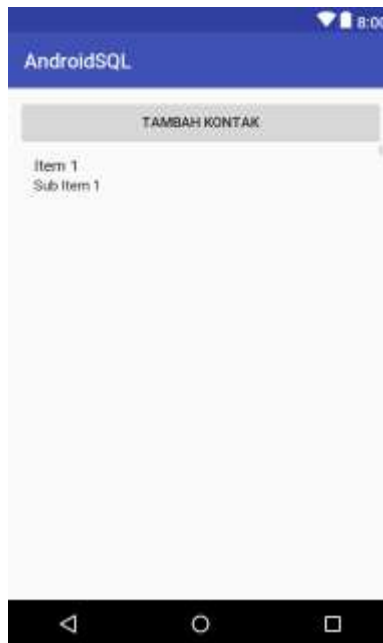
Kita akan membuat *activity* baru. Untuk *activity* pertama ini kita akan membuat tampilan untuk mengambil data dari database yang sudah kita buat sebelumnya. Pilih Empty Activity dan untuk nama *activity* kita beri nama **ListActivity**. Klik Finish.



Gambar 6. Persiapan pembuatan *project* Android (membuat *activity* baru)

Kita tunggu Gradle melakukan proses *build...* dan jika Gradle dalam status *build successfully*, berarti kita bisa mulai mempersiapkan baik tampilan maupun proses *logic* yang akan dilakukan.

Berikut ini adalah tampilan untuk menampilkan data kontak beserta *coding* XML nya.



Gambar 7. Tampilan Android untuk menampilkan semua kontak

activity_list.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="match_parent"
5.     android:layout_height="match_parent"
6.     android:padding="5dp"
7.     tools:context=".ListActivity">
8.
9.     <LinearLayout
10.         android:id="@+id/linearContact"
11.         android:layout_width="match_parent"
12.         android:layout_height="match_parent"
13.         android:orientation="vertical"
14.         android:padding="5dp">
15.
16.         <Button
17.             android:id="@+id/btnAddContact"
18.             android:layout_width="match_parent"
19.             android:layout_height="wrap_content"
20.             android:layout_weight="1"
21.             android:text="Tambah Kontak" />
22.
23.         <ListView
24.             android:id="@+id/listContact"
25.             android:layout_width="match_parent"
26.             android:layout_height="match_parent"
27.             android:layout_weight="1" />
28.     </LinearLayout>
29. </ScrollView>

```

AndroidManifest.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3.     package="com.example.yohanesdwilistio.androidsql">
4.     <uses-permission android:name="android.permission.INTERNET" />
5.     <application
6.         android:allowBackup="true"
7.         android:icon="@mipmap/ic_launcher"
8.         android:label="@string/app_name"
9.         android:roundIcon="@mipmap/ic_launcher_round"
10.        android:supportRtl="true"
11.        android:theme="@style/AppTheme">
12.        <activity android:name=".ListActivity">

```

```

13.         <intent-filter>
14.             <action android:name="android.intent.action.MAIN" />
15.
16.             <category android:name="android.intent.category.LAUNCHER" />
17.         </intent-filter>
18.     </activity>
19. </application>
20.</manifest>

```

Perhatikan baris ke-4 *script* AndroidManifest.xml di atas. Pada baris tersebut terdapat *permission* untuk dapat mengakses internet.

Kita akan menambahkan satu file XML untuk menampilkan detail kontak bila di-klik pada ListView, seperti berikut : **detail_contact.xml**

```

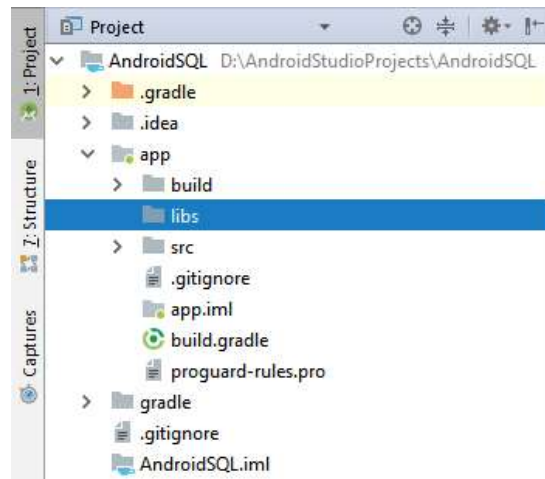
1. <?xml version="1.0" encoding="utf-8"?>
2. <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:layout_width="match_parent"
4.     android:layout_height="match_parent"
5.     android:padding="10dp">
6.     <TableRow>
7.
8.         <TextView
9.             android:id="@+id/textView6"
10.            android:layout_width="match_parent"
11.            android:layout_height="wrap_content"
12.            android:textStyle="bold"
13.            android:text="Email " />
14.
15.         <TextView
16.             android:id="@+id/txtDetailEmail"
17.             android:layout_width="match_parent"
18.             android:layout_height="wrap_content"
19.             android:text="TextView" />
20.     </TableRow>
21.     <TableRow>
22.
23.         <TextView
24.             android:id="@+id/textView8"
25.             android:layout_width="match_parent"
26.             android:layout_height="wrap_content"
27.             android:textStyle="bold"
28.             android:text="Phone " />
29.
30.         <TextView
31.             android:id="@+id/txtDetailPhone"
32.             android:layout_width="match_parent"
33.             android:layout_height="wrap_content"
34.             android:text="TextView" />
35.     </TableRow>
36. </TableLayout>

```

Bagian 2 : Java

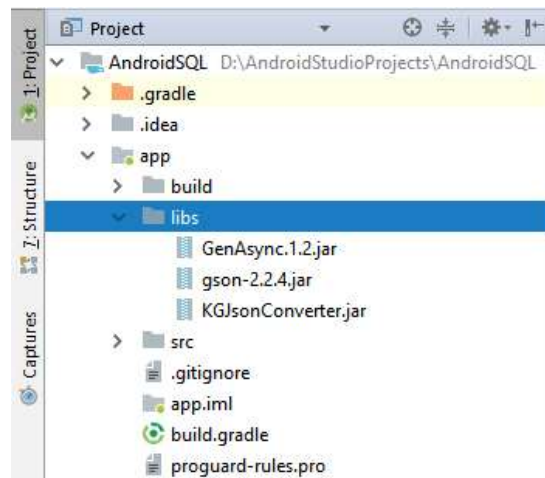
Untuk mengambil data dari database MySQL, kita memerlukan 3 buah library, diantaranya Generic AsyncTask 1.2, GSON, dan KGJJson Converter. Anda bisa mengunduhnya di situs www.github.com/kosalgeek dan download file JAR-nya.

Kita akan *copy* ketiga *library* di atas. Di *navigation pane*, klik pada Android, lalu pilih Project. Akan muncul seperti ini.



Gambar 8. Navigation pane file Project Android.

Kita akan meng-copy ketiga library tersebut pada direktori **libs** di *project* kita.



Gambar 9. Navigation pane file Project Android dengan *library* yang sudah di-copy

Select ketiga file JAR ini, klik kanan lalu pilih Add as Library.

Untuk mengecek apakah ketiga *library* sukses ditambahkan dalam project kita, silahkan cek dengan klik Android di bagian *navigation pane*, lalu pada Gradle Scripts, klik build.gradle (module: app).

Jika menemukan *script* Gradle ini, sebenarnya sudah bisa dikatakan berhasil karena Gradle akan meng-*compile* file JAR yang terdapat pada direktori libs.

```
implementation fileTree(dir: 'libs', include: ['*.jar'])
```

Atau biasanya akan muncul seperti ini di build.gradle

```
compile files('libs/GenAsync.1.2.jar')
compile files('libs/GenAsync.1.2.jar')
compile files('libs/GenAsync.1.2.jar')
```

Sekarang kita buat sebuah *class* baru yang bernama **Contact** dengan penamaan variabel-variabelnya kita samakan dengan nama-nama *field* yang ada di tabel **contact** dari database kita. Mengapa harus sama? Hal ini untuk memudahkan KGJson Converter melakukan *parsing* dari teks JSON ke dalam bentuk *arraylist* dengan tipe data kelas yang kita buat.

```
1. public class Contact {
2.     int id;
3.     String prefix;
4.     String first_name;
5.     String last_name;
6.     String email;
7.     String phone;
8. }
```

Untuk bagian Java ini lumayan panjang dikarenakan kita akan berurusan dengan menampilkan data kontak pada ListView, dan juga menangani kejadian saat salah satu item di ListView diklik sehingga menampilkan detail kontak yang dipilih (belum lagi *event* pada DialogInterface bila tombol Update maupun Delete diklik).

ListActivity.java

```
1.  import android.app.Activity;
2.  import android.app.AlertDialog;
3.  import android.content.DialogInterface;
4.  import android.content.Intent;
5.  import android.support.v7.app.AppCompatActivity;
6.  import android.os.Bundle;
7.  import android.view.View;
8.  import android.view.ViewGroup;
9.  import android.widget.AdapterView;
10. import android.widget.Button;
11. import android.widget.ListView;
12. import android.widget.SimpleAdapter;
13. import android.widget.TextView;
14.
15. import com.kosalgeek.android.json.JsonConverter;
16. import com.kosalgeek.genasync12.AsyncResponse;
17. import com.kosalgeek.genasync12.PostResponseAsyncTask;
18.
19. import java.util.ArrayList;
20. import java.util.HashMap;
21. import java.util.List;
22. import java.util.Map;
23.
24. public class ListActivity extends AppCompatActivity {
25.     private Button btnAdd;
26.     private ArrayList<Contact> userContact = new ArrayList<>();
27.     private ListView listContact;
28.     private Activity activity = this;
29.     private SimpleAdapter adapter;
30.     @Override
31.     protected void onCreate(Bundle savedInstanceState) {
32.         super.onCreate(savedInstanceState);
33.         setContentView(R.layout.activity_list);
34.         btnAdd = findViewById(R.id.btnAddContact);
35.         listContact = findViewById(R.id.listContact);
36.         loadAllContacts();
37.         btnAdd.setOnClickListener(new View.OnClickListener() {
38.             @Override
39.             public void onClick(View v) {
40.                 Intent i = new Intent(activity, FormActivity.class);
41.                 i.putExtra("mode", "insert");
42.                 startActivity(i);
43.             }
44.         });
45.     }
46.
47.     public void loadAllContacts(){
48.         PostResponseAsyncTask taskRead = new PostResponseAsyncTask(ListActivity.this, "Fetching a
49.         ll contacts...", new AsyncResponse() {
50.             @Override
51.             public void processFinish(String s) {
52.                 userContact = new JsonConverter<Contact>().toArrayList(s, Contact.class);
53.                 List<Map<String, String>> contact = insertContactIntoList(userContact);
54.                 adapter = new SimpleAdapter(activity, contact, android.R.layout.simple_list_item_
55.                 2,
56.                 new String[]{"name", "email"}, new int[]{android.R.id.text1, android.R.id
57.                 .text2});
58.                 new ListViewAdjustment().adjustListViewWithSimple(listContact, adapter);
59.                 listContact.setOnItemClickListener(new AdapterView.OnItemClickListener() {
60.                     @Override
61.                     public void onItemClick(AdapterView<?> adapterView, View view, int i, long l)
62.                     {
63.                         registerForContextMenu(listContact);
64.                         listContact.showContextMenu();
65.                         TextView tv = view.findViewById(android.R.id.text1);
66.                         String name = tv.getText().toString();
67.                         ViewGroup viewGroup = view.findViewById(R.id.layoutList);
68.                         View v = getLayoutInflater().inflate(R.layout.detail_contact, viewGroup);
69.                         viewDetailContact(v, name);
70.                     }
71.                 });
72.             }
73.         });
74.     }
75. }
```

```

67.         });
68.     }
69. });
70.     taskRead.execute("http://192.168.43.138/androidsql/contact");
71. }
72.
73.     public List<Map<String, String>> insertContactIntoList(ArrayList<Contact> userContact){
74.         List<Map<String, String>> contact = new ArrayList<>();
75.         if (userContact.size() > 0){
76.             for (int i = 0; i < userContact.size(); i++){
77.                 Map<String, String> contact_detail = new HashMap<>();
78.                 contact_detail.put("name", userContact.get(i).first_name + " " + userContact.get(
79. i).last_name);
80.                 contact_detail.put("email", userContact.get(i).email);
81.                 contact.add(contact_detail);
82.             }
83.         } else {
84.             Map<String, String> contact_detail = new HashMap<>();
85.             contact_detail.put("name", "No contact saved here");
86.             contact_detail.put("email", "Start adding new contact!");
87.             contact.add(contact_detail);
88.         }
89.         return contact;
90.     }
91.     public void viewDetailContact(View v, String name){
92.         TextView txtDetailEmail = v.findViewById(R.id.txtDetailEmail);
93.         TextView txtDetailPhone = v.findViewById(R.id.txtDetailPhone);
94.         String[] names = name.split(" ");
95.         for (int i = 0; i < userContact.size(); i++){
96.             if (userContact.get(i).first_name.contains(names[0]) && userContact.get(i).last_name.
97. contains(names[names.length - 1])){
98.                 txtDetailEmail.setText(userContact.get(i).email);
99.                 txtDetailPhone.setText(userContact.get(i).phone);
100.                 new AlertDialog.Builder(this).setTitle(userContact.get(i).prefix + " " + name).se
101. tView(v).
102.                 setPositiveButton("Update", new DialogInterface.OnClickListener() {
103.                     @Override
104.                     public void onClick(DialogInterface dialog, int which) {
105.                         }
106.                 }).setNegativeButton("Delete", new DialogInterface.OnClickListener() {
107.                     @Override
108.                     public void onClick(DialogInterface dialog, int which) {
109.                         }
110.                 }).show();
111.             }
112.         }
113.     }
114. }

```

Keterangan : Anda dapat mengganti IP *address* 192.168.43.138 dengan IP *address* komputer Anda ketika terhubung ke internet atau IP *address* localhost (10.0.2.2 jika menggunakan emulator bawaan Android Studio).

ListViewAdjustment.java

Buatlah *class* bernama ListViewAdjustment untuk mengatur *height* pada ListView sehingga semua data dapat ditampilkan di layar perangkat Android.

```

1. public class ListViewAdjustment {
2.     public void adjustListViewwithSimple(ListView list, SimpleAdapter adapter){
3.         list.setAdapter(adapter);
4.         int totalHeight = 0;
5.         ListAdapter listAdapter = list.getAdapter();
6.         for (int i = 0; i < listAdapter.getCount(); i++) {
7.             View listItem = listAdapter.getView(i, null, list);
8.             listItem.measure(0, 0);
9.             totalHeight += (listItem.getMeasuredHeight() * 1.05);
10.        }
11.        ViewGroup.LayoutParams params = list.getLayoutParams();
12.        params.height = totalHeight + (list.getDividerHeight() * (listAdapter.getCount()));

```

```
13.         list.setLayoutParams(params);
14.     }
15. }
```

Babak 4 : Insert Data

Kita akan membuat satu *activity* baru untuk melakukan *input* data. Pada kasus ini kita akan menggunakan *activity* ini untuk *insert* maupun *update* dalam satu form.

Bagian 1 : Tampilan

Insert Contact

Salutation ☐ Mr. ☐ Mrs. ☐ Ms.

First Name

Last Name

Email

Phone Number

INSERT

Gambar 10. Tampilan *form* untuk insert dan update data

activity_form.xml

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     xmlns:tools="http://schemas.android.com/tools"
4.     android:layout_width="fill_parent"
5.     android:layout_height="fill_parent"
6.     android:padding="5dp"
7.     tools:context=".FormActivity">
8.     <TableRow>
9.
10.         <TextView
11.             android:id="@+id/textView"
12.             android:layout_width="match_parent"
13.             android:layout_height="wrap_content"
14.             android:layout_gravity="center_vertical"
15.             android:text="Salutation" />
16.
17.         <RadioGroup
18.             android:id="@+id/rgSalutation"
19.             android:layout_width="match_parent"
20.             android:layout_height="wrap_content"
21.             android:orientation="horizontal">
22.
23.             <RadioButton
24.                 android:id="@+id/radioButton"
25.                 android:layout_width="wrap_content"
26.                 android:layout_height="wrap_content"
27.                 android:layout_weight="1"
28.                 android:text="Mr." />
29.
30.             <RadioButton
31.                 android:id="@+id/radioButton2"
32.                 android:layout_width="wrap_content"
33.                 android:layout_height="wrap_content"

```

```

34.         android:layout_weight="1"
35.         android:text="Mrs." />
36.
37.         <RadioButton
38.             android:id="@+id/radioButton3"
39.             android:layout_width="wrap_content"
40.             android:layout_height="wrap_content"
41.             android:layout_weight="1"
42.             android:text="Ms." />
43.     </RadioGroup>
44. </TableRow>
45. <TableRow>
46.     <TextView
47.         android:id="@+id/textView2"
48.         android:layout_width="match_parent"
49.         android:layout_height="wrap_content"
50.         android:text="First Name" />
51.
52.     <EditText
53.         android:id="@+id/txtFname"
54.         android:layout_width="match_parent"
55.         android:layout_height="wrap_content"
56.         android:ems="10"
57.         android:inputType="textPersonName"
58.         android:hint="Input First Name" />
59. </TableRow>
60. <TableRow>
61.     <TextView
62.         android:id="@+id/textView3"
63.         android:layout_width="match_parent"
64.         android:layout_height="wrap_content"
65.         android:text="Last Name" />
66.
67.     <EditText
68.         android:id="@+id/txtLname"
69.         android:layout_width="match_parent"
70.         android:layout_height="wrap_content"
71.         android:ems="10"
72.         android:inputType="textPersonName"
73.         android:hint="Input Last Name"/>
74. </TableRow>
75. <TableRow>
76.     <TextView
77.         android:id="@+id/textView4"
78.         android:layout_width="match_parent"
79.         android:layout_height="wrap_content"
80.         android:text="Email" />
81.
82.     <EditText
83.         android:id="@+id/txtEmail"
84.         android:layout_width="match_parent"
85.         android:layout_height="wrap_content"
86.         android:ems="10"
87.         android:inputType="textEmailAddress"
88.         android:hint="Input Email"/>
89. </TableRow>
90. <TableRow>
91.     <TextView
92.         android:id="@+id/textView5"
93.         android:layout_width="match_parent"
94.         android:layout_height="wrap_content"
95.         android:text="Phone Number" />
96.
97.     <EditText
98.         android:id="@+id/txtPhone"
99.         android:layout_width="match_parent"
100.        android:layout_height="wrap_content"
101.        android:ems="10"
102.        android:inputType="phone"
103.        android:hint="Input Phone Number"/>
104. </TableRow>
105.
106. <Button

```

```

107.         android:id="@+id/btnInsert"
108.         android:layout_width="match_parent"
109.         android:layout_height="wrap_content"
110.         android:text="Insert" />
111. </TableLayout>

```

Jangan lupa untuk memodifikasi bagian **AndroidManifest.xml** di bagian ini :

```

1. <activity android:name=".FormActivity"
2.     android:label="Insert Contact"></activity>

```

Bagian 2 : Java

FormActivity.java

```

1. import android.app.Activity;
2. import android.content.Intent;
3. import android.support.v7.app.AppCompatActivity;
4. import android.os.Bundle;
5. import android.view.View;
6. import android.widget.Button;
7. import android.widget.EditText;
8. import android.widget.RadioGroup;
9. import android.widget.Toast;
10.
11. import com.kosalgeek.genasync12.AsyncResponse;
12. import com.kosalgeek.genasync12.PostResponseAsyncTask;
13.
14. import java.util.HashMap;
15.
16. public class FormActivity extends AppCompatActivity {
17.     String mode;
18.     private RadioGroup salutation;
19.     private EditText firstname;
20.     private EditText lastname;
21.     private EditText email;
22.     private EditText phone;
23.     private Button insert;
24.     private Activity activity = this;
25.     @Override
26.     protected void onCreate(Bundle savedInstanceState) {
27.         super.onCreate(savedInstanceState);
28.         setContentView(R.layout.activity_form);
29.         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
30.         init();
31.         mode = getIntent().getExtras().getString("mode");
32.     }
33.     @Override
34.     public boolean onSupportNavigateUp(){
35.         super.onBackPressed();
36.         return true;
37.     }
38.     private void init(){
39.         salutation = findViewById(R.id.rgSalutation);
40.         firstname = findViewById(R.id.txtFname);
41.         lastname = findViewById(R.id.txtLname);
42.         email = findViewById(R.id.txtEmail);
43.         phone = findViewById(R.id.txtPhone);
44.         insert = findViewById(R.id.btnInsert);
45.         insert.setOnClickListener(new View.OnClickListener() {
46.             @Override
47.             public void onClick(View v) {
48.                 String fname = firstname.getText().toString();
49.                 String lname = lastname.getText().toString();
50.                 String mail = email.getText().toString();
51.                 String phne = phone.getText().toString();
52.                 if (validateDataInput(fname, lname, mail, phne)){
53.                     Toast.makeText(FormActivity.this, "Please enter data correctly", Toast.LENGTH_
LONG).show();
54.                 } else {
55.                     HashMap<String, String> postData = new HashMap<>();

```

```

56.         postData.put("salutation", getSalutation());
57.         postData.put("firstname", fname);
58.         postData.put("lastname", lname);
59.         postData.put("email", mail);
60.         postData.put("phone", phne);
61.         PostResponseAsyncTask taskInsert = new PostResponseAsyncTask(FormActivity.this
, postData, "Adding contacts...", new AsyncResponse() {
62.             @Override
63.             public void processFinish(String s) {
64.                 if (s.equals("success")){
65.                     Toast.makeText(FormActivity.this, "Data added successfully", Toast
.LENGTH_LONG).show();
66.                     Intent i = new Intent(activity, ListActivity.class);
67.                     startActivity(i);
68.                     finish();
69.                 }
70.             }
71.         });
72.         taskInsert.execute("http://192.168.43.138/androidsql/contact/insert");
73.     }
74. }
75. });
76. }
77.
78. private String getSalutation(){
79.     switch (salutation.getCheckedRadioButtonId()){
80.         case R.id.radioButton:
81.             return "Mr.";
82.         case R.id.radioButton2:
83.             return "Mrs.";
84.         case R.id.radioButton3:
85.             return "Ms.";
86.     }
87.     return null;
88. }
89.
90. private boolean validateDataInput(String fname, String lname, String mail, String phone){
91.     return fname.isEmpty() || lname.isEmpty() || (mail.isEmpty() && !Patterns.EMAIL_ADDRESS.ma
tcher(mail).matches()) || phone.isEmpty();
92. }
93. }

```

Babak 5 : Update Data

Untuk melakukan *update* data, kita akan menambahkan pada bagian OnClickListener di DialogInterface saat tombol Update diklik di ListActivity.java, seperti berikut.

```

1. new AlertDialog.Builder(this).setTitle(userContact.get(i).prefix + " " + name).setView(v).
2.     setPositiveButton("Update", new DialogInterface.OnClickListener() {
3.         @Override
4.         public void onClick(DialogInterface dialog, int which) {
5.             Intent j = new Intent(activity, FormActivity.class);
6.             j.putExtra("mode", "update");
7.             j.putExtra("id", userContact.get(finalI).id);
8.             j.putExtra("prefix", userContact.get(finalI).prefix);
9.             j.putExtra("firstname", userContact.get(finalI).first_name);
10.            j.putExtra("lastname", userContact.get(finalI).last_name);
11.            j.putExtra("email", userContact.get(finalI).email);
12.            j.putExtra("phone", userContact.get(finalI).phone);
13.            startActivity(j);
14.        }
15.    }).setNegativeButton("Delete", new DialogInterface.OnClickListener() {
16.        @Override
17.        public void onClick(DialogInterface dialog, int which) {
18.
19.        }
20.    }).show();

```

Tambahkan *script* pada method onCreate di class FormActivity.java sehingga jika dalam mode update, teks pada button akan berubah menjadi "Update", dan form akan terisi dengan data semula.

```

1. if (mode.equals("update")){
2.     insert.setText("Update");
3.     switch (getIntent().getExtras().getString("prefix")){
4.         case "Mr.":
5.             salutation.check(R.id.radioButton);
6.             break;
7.         case "Mrs.":
8.             salutation.check(R.id.radioButton2);
9.             break;
10.        case "Ms.":
11.            salutation.check(R.id.radioButton3);
12.            break;
13.        }
14.        firstname.setText(getIntent().getExtras().getString("firstname"));
15.        lastname.setText(getIntent().getExtras().getString("lastname"));
16.        email.setText(getIntent().getExtras().getString("email"));
17.        phone.setText(getIntent().getExtras().getString("phone"));
18.    }

```

Dan pada method OnClickListener saat tombol ditekan, kita modifikasi juga *source code*-nya pada bagian ini (setelah baris `postData.put("phone", phne);`);

```

1. if (mode.equals("update")){
2.     postData.put("id", String.valueOf(getIntent().getExtras().getInt("id")));
3.     PostResponseAsyncTask taskUpdate = new PostResponseAsyncTask(FormActivity.this, postData, "Updating contacts...", new AsyncResponse() {
4.         @Override
5.         public void processFinish(String s) {
6.             if (s.equals("success")){
7.                 Toast.makeText(FormActivity.this, "Data updated successfully", Toast.LENGTH_LONG).show();
8.                 Intent i = new Intent(activity, ListActivity.class);
9.                 i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
10.                startActivity(i);
11.                finish();
12.            }
13.        }
14.    });
15.    taskUpdate.execute("http://192.168.43.138/androidsql/contact/update" + String.valueOf(getIntent().getExtras().getInt("id")));
16. } else {
17.     PostResponseAsyncTask taskInsert = new PostResponseAsyncTask(FormActivity.this, postData, "Adding contacts...", new AsyncResponse() {
18.         @Override
19.         public void processFinish(String s) {
20.             if (s.equals("success")){
21.                 Toast.makeText(FormActivity.this, "Data added successfully", Toast.LENGTH_LONG).show();
22.                 Intent i = new Intent(activity, ListActivity.class);
23.                 i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
24.                 startActivity(i);
25.                 finish();
26.             }
27.        }
28.    });
29.    taskInsert.execute("http://192.168.43.138/androidsql/contact/insert");
30. }

```

Babak 6 : Delete Data

Untuk proses *delete* kita akan tambahkan *script* di dalam `DialogInterface.OnClickListener` pada *button* Delete di class `ListActivity.java`.

```

1. new AlertDialog.Builder(ListActivity.this).setTitle("Confirm Delete")
2.     .setMessage("Are you sure you want to delete the selected contact?")
3.     .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
4.         @Override
5.         public void onClick(DialogInterface dialog, int which) {
6.             HashMap<String, String> postData = new HashMap<>();

```

```

7.         postData.put("id", String.valueOf(userContact.get(finalI).id));
8.         PostResponseAsyncTask taskUpdate = new PostResponseAsyncTask(ListActivity.this, postData, "Deleting contacts...", new AsyncResponse() {
9.             @Override
10.            public void processFinish(String s) {
11.                if (s.equals("success")){
12.                    Toast.makeText(ListActivity.this, "Data deleted successfully", Toast.LENGTH_LONG).show();
13.                }
14.            }
15.        });
16.        taskUpdate.execute("http://192.168.43.138/androidsql/contact/delete/" + String.valueOf(userContact.get(finalI).id));
17.        loadAllContacts();
18.    }
19.    }).setNegativeButton("No", new DialogInterface.OnClickListener() {
20.        @Override
21.        public void onClick(DialogInterface dialog, int which) {
22.
23.        }
24.    }).show();

```

Untuk *source code* lengkap program ini beserta PHP CodeIgniter dan database MySQL nya, Anda dapat mengunduh melalui GitHub :

- Project **Android Studio** : <https://github.com/yohanesdl9/androidsql>
- Project web **CodeIgniter** (sudah termasuk file database SQL di dalam folder database) : <https://github.com/yohanesdl9/androidsqlweb>