

# Bermain API Kotlin dan CI

1. Pertama kita ekstrak codeigniter yang telah kita download tadi lalu gantilah namanya tersebut bebas sesuai keinginan disini saya mengganti dengan nama **aplikasi**
2. Letakan didalam folder webservice kita jika kalian menggunakan xampp letakan di dalam folder htdocs
3. Setelah itu kita edit file **autoload.php** yang ada didalam folder application > config dan carilah kata autoload libraries dan edit menjadi seperti ini

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

4. Lalu masih berada di dalam folder config kita edit juga file **config.php** dan

**database.php** kita edit menjadi seperti ini

```
//config.php...$config['base_url'] =  
'server_api';...//database.php...$active_group = 'default';  
$query_builder = TRUE;$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',          //hostname akses ke mysql  
    'username' => 'root',                //username masuk ke mysql  
    'password' => '',                    //password mysql  
    'database' => 'db_crud',             //nama database yang akan kita buat  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,  
    'cachedir' => '',  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => '',  
    'encrypt' => FALSE,  
    'compress' => FALSE,  
    'stricton' => FALSE,  
    'failover' => array(),
```

```
'save_queries' => TRUE
);
```

5. Setelah itu kita pindah ke dalam folder controler tepatnya di **application > controller** lalu kita buat file baru disana dengan nama **ServerApi.php**

ServerApi.php

```
<?php
```

```
defined('BASEPATH') OR exit('No direct script access allowed');
```

```
class ServerApi extends CI_Controller {
```

```
    // fungsi untuk CREATE
```

```
    public function addStaff()
```

```
    {
```

```
        // deklarasi variable
```

```
        $name = $this->input->post('name');
```

```
        $hp = $this->input->post('hp');
```

```
        $alamat = $this->input->post('alamat');
```

```
        // isikan variabel dengan nama file
```

```
        $data['staff_name'] = $name;
```

```
        $data['staff_hp'] = $hp;
```

```
        $data['staff_alamat'] = $alamat;
```

```
        $q = $this->db->insert('tb_staff', $data);
```

```
        // check insert berhasil apa nggak
```

```
        if ($q) {
```

```
            $response['pesan'] = 'insert berhasil';
```

```
            $response['status'] = 200;
```

```
        } else {
```

```
            $response['pesan'] = 'insert error';
```

```
            $response['status'] = 404;
```

```
        }
```

```
        echo json_encode($response);
```

```
    }
```

```
    // fungsi untuk READ
```

```
    public function getDataStaff()
```

```
    {
```

```
        $q = $this->db->get('tb_staff');
```

```
        if ($q -> num_rows() > 0) {
```

```
            $response['pesan'] = 'data ada';
```

```
            $response['status'] = 200;
```

```
            // 1 row
```

```
            $response['staff'] = $q->row();
```

```
            $response['staff'] = $q->result();
```

```
        } else {
```

```

        $response['pesan'] = 'data tidak ada';
        $response['status'] = 404;
    }
    echo json_encode($response);
}

// fungsi untuk DELETE
public function deleteStaff()
{
    $id = $this->input->post('id');
    $this->db->where('staff_id', $id);
    $status = $this->db->delete('tb_staff');
    if ($status == true) {
        $response['pesan'] = 'hapus berhasil';
        $response['status'] = 200;
    } else {
        $response['pesan'] = 'hapus error';
        $response['status'] = 404;
    }
    echo json_encode($response);
}

// fungsi untuk UPDATE
public function updateStaff()
{
    // deklarasi variable
    $id = $this->input->post('id');
    $name = $this->input->post('name');
    $hp = $this->input->post('hp');
    $alamat = $this->input->post('alamat');
    $this->db->where('staff_id', $id);
    // isikan variabel dengan nama file
    $data['staff_name'] = $name;
    $data['staff_hp'] = $hp;
    $data['staff_alamat'] = $alamat;
    $q = $this->db->update('tb_staff', $data);
    // check insert berhasil apa nggak
    if ($q) {
        $response['pesan'] = 'update berhasil';
        $response['status'] = 200;
    } else {
        $response['pesan'] = 'update error';
        $response['status'] = 404;
    }
    echo json_encode($response);
}
}

```

6. Setelah selesai membuat web service kita masuk kedalam database kita lalu konfigurasi dengan aturan seperti ini  
nama database **dbcrud** dengan nama table **tb\_staff**

7. Lalu kita buat 4 Field Table dengan aturan seperti dibawah ini

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
<input type="checkbox"/>	1	<b>staff_id</b>	int(11)		Tidak	Tidak ada		AUTO_INCREMENT	Ubah  Hapus  Lainnya
<input type="checkbox"/>	2	<b>staff_name</b>	varchar(30)	latin1_swedish_ci	Tidak	Tidak ada			Ubah  Hapus  Lainnya
<input type="checkbox"/>	3	<b>staff_hp</b>	varchar(13)	latin1_swedish_ci	Tidak	Tidak ada			Ubah  Hapus  Lainnya
<input type="checkbox"/>	4	<b>staff_alamat</b>	text	latin1_swedish_ci	Tidak	Tidak ada			Ubah  Hapus  Lainnya

Column

8. Setelah itu kita isi kan dengan data dummy terlebih dahulu untuk mengecek apakah web service API sudah berjalan baik atau belum

Setelah itu kita cek dengan mengetikan [http://ip\\_server/ciapi/index.php/ServerApi/getDataStaff](http://ip_server/ciapi/index.php/ServerApi/getDataStaff)

Sekarang kita pindah ke Android Studio nya kita buat project baru dengan nama bebas lalu ikuti langkah dibawah ini :

1. Setelah selesai build gradle kita perlu menambahkan beberapa library maka kita masukan library itu kedalam dependencies berikut daftar librarynya :

```
//ini berada di build.gradle//Retrofit
implementation 'com.squareup.retrofit2:retrofit:2.6.1'
implementation 'com.squareup.retrofit2:converter-
gson:2.6.1'//Interceptor logging test
implementation("com.squareup.okhttp3:logging-
interceptor:4.2.0")//anko
implementation "org.jetbrains.anko:anko:0.10.8"
//anko library
implementation "org.jetbrains.anko:anko-commons:0.10.8"
implementation 'androidx.recyclerview:recyclerview:1.0.0'
implementation 'androidx.cardview:cardview:1.0.0'
```

2. Setelah itu kita edit file **activity\_main.xml** dengan mengetikan code berikut :

activity\_main.xml

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

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView

        android:id="@+id/rvCategory"
        tools:listitem="@layout/item_data"

        app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/btnTambah">
    </androidx.recyclerview.widget.RecyclerView>
    <Button

        android:id="@+id/btnTambah"
        android:text="Tambah"
```

```

        android:background="@android:color/holo_green_dark"
        android:textColor="@android:color/white"
        android:layout_width="match_parent"
        android:layout_alignParentBottom="true"
        android:layout_height="wrap_content"/>

</RelativeLayout>

```

## 2. Lalu kita buat file xml baru dengan nama **item\_data.xml** lalu kita edit menjadi seperti ini :

```

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

<LinearLayout
    android:padding="16dp"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <androidx.cardview.widget.CardView
        android:background="#dddddd"
        app:cardCornerRadius="16dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
            android:background="#dddddd"
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <LinearLayout
                android:background="#dddddd"
                android:padding="8dp"
                android:orientation="horizontal"
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <TextView
                    android:text="Nama      :"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"/>
                <TextView

```

```

        android:layout_marginLeft="10dp"
        android:id="@+id/tvName"
        android:text="Muhamad "
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
<LinearLayout
    android:background="#dddddd"
    android:layout_marginLeft="8dp"
    android:layout_marginBottom="8dp"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:text="No Hp   :"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <TextView
        android:layout_marginLeft="10dp"
        android:id="@+id/tvPhone"
        android:text="Muhamad "
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
<LinearLayout
    android:layout_marginBottom="10dp"
    android:layout_marginLeft="8dp"
    android:background="#dddddd"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:text="Alamat   :"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <TextView
        android:lines="3"
        android:layout_marginLeft="10dp"
        android:id="@+id/tvAddress"
        android:text="Muhamad "
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>

```

```

        <Button
            android:textColor="@android:color/white"
            android:text="Hapus"
            android:background="@android:color/holo_red_light"
            android:id="@+id/btnHapus"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"/>
    </LinearLayout>

</androidx.cardview.widget.CardView>

</LinearLayout>

```

4. Setelah itu kita buat Activity baru dengan memilih jenis activity nya yaitu Empty Activity lalu beri nama **UpdateAddActivity** setelah terbuat kita edit file xml nya dengan mengetikan code berikut ini :  
activity\_update\_add.xml

```

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

<LinearLayout
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".UpdateAddActivity">

    <EditText
        android:id="@+id/etName"
        android:hint="Name"
        android:layout_marginBottom="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <EditText

```



```

        android:id="@+id/etPhone"
        android:hint="Phone"
        android:layout_marginBottom="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/etAlamat"
        android:hint="Alamat"
        android:lines="3"
        android:layout_marginBottom="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

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

</LinearLayout>

```

5. Setelah itu kita buat package baru dengan nama **adapter**, **model**, **presenter**, dan **network** lalu kita buat beberapa file untuk di tempat di package package tadi dengan nama dan jenisnya sebagai berikut :

- a. DataAdapter (Class) > package **adapter**
- b. DataItem, ResultStaff, ResultStatus (Class) > package **model**
- c. NetworkConfig (Object) > package **network**
- d. CrudView (Interface) dan Presenter(Class) > package **presenter**

6. Lalu kita edit satu persatu file tadi pertama kita edit file DataAdapter terlebih dahulu dengan mengetikan code berikut :  
DataAdapter

```
package
com.example.crudmysql.ada
pter
```

```
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.example..crudmysql.R
import com.example..crudmysql.model.DataItem
import kotlinx.android.synthetic.main.item_data.view.*
import org.jetbrains.anko.sdk27.coroutines.onClick

class DataAdapter(val data: List<DataItem>?, private val click:
onClickItem) : RecyclerView.Adapter<DataAdapter.MyHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType:
Int): MyHolder {
        val view =
LayoutInflater.from(parent.context).inflate(R.layout.item_data,par
ent,false)
        return MyHolder(view)
    }

    override fun getItemCount() = data?.size ?: 0

    override fun onBindViewHolder(holder: MyHolder, position: Int)
{
        holder.onBind(data?.get(position))
        holder.itemView.onClick {
            click.clicked(data?.get(position))
        }
        holder.itemView.btnHapus.setOnClickListener {
            click.delete(data?.get(position))
        }
    }

    class MyHolder(itemView: View) :
RecyclerView.ViewHolder(itemView) {
        fun onBind(get: DataItem?) {
            itemView.tvName.text = get?.staffName
            itemView.tvPhone.text = get?.staffHp
            itemView.tvAddress.text = get?.staffAlamat
        }
    }

    interface onClickItem{
```

```

        fun clicked (item: DataItem?)
        fun delete(item: DataItem?)

    }
}

```

## 7. lalu kita edit file DataItem, ResultStaff, dan ResultStatus dengan mengetikan code berikut :

```

//DataItempackage com.example.crudmysql.model

import com.google.gson.annotations.SerializedName
import java.io.Serializable

class DataItem : Serializable{

    @field:SerializedName("staff_name")
    val staffName: String? = null

    @field:SerializedName("staff_id")
    val staffId: String? = null

    @field:SerializedName("staff_hp")
    val staffHp: String? = null

    @field:SerializedName("staff_alamat")
    val staffAlamat: String? = null
}//ResultStaffpackage com.example..crudmysql.model

import com.google.gson.annotations.SerializedName

data class ResultStaff (
    @field:SerializedName("pesan")
    val pesan: String? = null,

    @field:SerializedName("staff")
    val staff: List<DataItem>? = null,

    @field:SerializedName("status")
    val status: Int? = null
)//ResultStatuspackage com.example..crudmysql.model

class ResultStatus {
    val pesan : String? = null
    val status : Int? = null
}

```

## 8. Kemudian kita edit juga file NetworkConfig dengan mengetikan code berikut :

NetworkConfig

```
package
com.example.crudmysql.network

import com.example.crudmysql.model.ResultStatus
import com.example.crudmysql.model.ResultStaff
import okhttp3.OkHttpClient
import okhttp3.logging.HttpLoggingInterceptor
import retrofit2.Call
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import retrofit2.http.Field
import retrofit2.http.FormUrlEncoded
import retrofit2.http.GET
import retrofit2.http.POST

object NetworkConfig {

    fun getInterceptor() : OkHttpClient {
        val logging = HttpLoggingInterceptor()
        logging.level = HttpLoggingInterceptor.Level.BODY

        val okHttpClient = OkHttpClient.Builder()
            .addInterceptor(logging)
            .build()

        return okHttpClient
    }

    //Retrofit

    fun getRetrofit(): Retrofit {
        return Retrofit.Builder()

        .baseUrl("http://ipserver/namaaplikasi/index.php/ServerApi/")
            .client(getInterceptor())

        .addConverterFactory(GsonConverterFactory.create())
            .build()
    }
}
```

```

        fun getService() =
getRetrofit().create(StaffService::class.java)
    }
    interface StaffService{

        //Fungsi Create Data
        @FormUrlEncoded
        @POST("addStaff")
        fun addStaff(@Field("name") name : String,
                    @Field("hp") hp : String,
                    @Field("alamat") alamat : String) :
        Call<ResultStatus>

        //Fungsi Get Data
        @GET("getDataStaff")

        fun getData() : Call<ResultStaff>

        //Fungsi Delete Data
        @FormUrlEncoded
        @POST("deleteStaff")
        fun deleteStaff(@Field("id") id: String?) :
        Call<ResultStatus>

        //Fungsi Update Data
        @FormUrlEncoded
        @POST("updateStaff")
        fun updateStaff(@Field("id") id: String,
                    @Field("name") name: String,
                    @Field("hp") hp : String,
                    @Field("alamat") alamat : String) :
        Call<ResultStatus>
    }

```

## 9. Lalu kita edit File CrudView dan Presenter dengan mengetikkan code berikut :

```

//CrudView package com.example.crudmysql.presenter

import com.example.crudmysql.model.DataItem

interface CrudView {
    //Untuk get data

```

```

    fun onSuccessGet(data: List<DataItem>?)
    fun onFailedGet(msg : String)

    //Untuk Delete
    fun onSuccessDelete(msg: String)
    fun onErrorDelete(msg: String)

    //Untuk Add
    fun successAdd(msg : String)
    fun errorAdd(msg: String)

    //Untuk Update
    fun onSuccessUpdate(msg : String)
    fun onErrorUpdate(msg : String)

}

```

## Presenter

```

package com.example..crudmysql.presenter

import android.util.Log
import com.example..crudmysql.model.ResultStaff
import com.example..crudmysql.model.ResultStatus
import com.example..crudmysql.network.NetworkConfig
import retrofit2.Call
import retrofit2.Response

class Presenter (val crudView: CrudView) {

    //Fungsi GetData
    fun getData(){
        NetworkConfig.getService().getData()
            .enqueue(object : retrofit2.Callback<ResultStaff>{
                override fun onFailure(call: Call<ResultStaff>, t: Throwable) {
                    crudView.onFailedGet(t.localizedMessage)
                    Log.d("Error", "Error Data")
                }

                override fun onResponse(call: Call<ResultStaff>, response:
                Response<ResultStaff>) {
                    if(response.isSuccessful){
                        val status = response.body()?.status
                        if (status == 200){
                            val data = response.body()?.staff
                            crudView.onSuccessGet(data)
                        } else{
                            crudView.onFailedGet("Error $status")
                        }
                    }
                }
            })
    }
}

```

```

    }
  }
})
}

```

//Add data

```

fun addData(name : String, hp : String, alamat : String){
    NetworkConfig.getService()
        .addStaff(name, hp, alamat)
        .enqueue(object : retrofit2.Callback<ResultStatus>{
            override fun onFailure(call: Call<ResultStatus>, t: Throwable) {
                crudView.errorAdd(t.localizedMessage)
            }

            override fun onResponse(call: Call<ResultStatus>, response:
Response<ResultStatus>) {
                if (response.isSuccessful && response.body()?.status == 200) {
                    crudView.successAdd(response.body()?.pesan ?: "")
                } else {
                    crudView.errorAdd(response.body()?.pesan ?: "")
                }
            }
        })
}

```

//Hapus Data

```

fun hapusData(id: String?){
    NetworkConfig.getService()
        .deleteStaff(id)
        .enqueue(object : retrofit2.Callback<ResultStatus>{
            override fun onFailure(call: Call<ResultStatus>, t: Throwable) {
                crudView.onErrorDelete(t.localizedMessage)
            }

            override fun onResponse(call: Call<ResultStatus>, response:
Response<ResultStatus>) {
                if (response.isSuccessful && response.body()?.status == 200){
                    crudView.onSuccessDelete(response.body()?.pesan ?: "")
                } else {
                    crudView.onErrorDelete(response.body()?.pesan ?: "")
                }
            }
        })
}

```

```

//Update Data
fun updateData(id: String, name: String, hp: String, alamat: String){
    NetworkConfig.getService()
        .updateStaff(id, name, hp, alamat)
        .enqueue(object : retrofit2.Callback<ResultStatus>{
            override fun onFailure(call: Call<ResultStatus>, t: Throwable) {
                crudView.onErrorUpdate(t.localizedMessage)
            }

            override fun onResponse(call: Call<ResultStatus>, response:
Response<ResultStatus>) {
                if (response.isSuccessful && response.body()?.status == 200){
                    crudView.onSuccessUpdate(response.body()?.pesan ?: "")
                }else{
                    crudView.onErrorUpdate(response.body()?.pesan ?: "")
                }
            }
        })
    }
}

```

10. Kemudian kita edit file MainActivity dengan mengetikan code berikut :

MainActivity

```

package
com.example.crudmysql

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.example.crudmysql.adapter.DataAdapter
import com.example.crudmysql.model.DataItem
import com.example.crudmysql.presenter.CrudView
import com.example.crudmysql.presenter.Presenter
import kotlinx.android.synthetic.main.activity_main.*
import org.jetbrains.anko.alert
import org.jetbrains.anko.startActivity

class MainActivity : AppCompatActivity(), CrudView {

    private lateinit var presenter: Presenter
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}

```



```

        presenter = Presenter(this)
        presenter.getData()

        btnTambah.setOnClickListener {
            startActivity<UpdateAddActivity>()
            finish()
        }
    }

    override fun onSuccessGet(data: List<DataItem>?) {
        rvCategory.adapter = DataAdapter(data, object :
        DataAdapter.OnItemClickListener {
            override fun clicked(item: DataItem?) {
                startActivity<UpdateAddActivity>("dataItem" to item)
            }

            override fun delete(item: DataItem?) {
                presenter.hapusData(item?.staffId)
                startActivity<MainActivity>()
                finish()
            }
        })
    }

    override fun onFailedGet(msg: String) {
    }

    override fun onSuccessDelete(msg: String) {
        presenter.getData()
    }

    override fun onErrorDelete(msg: String) {
        alert {
            title = "Error Delete Data"
        }.show()
    }

    override fun successAdd(msg: String) {
    }

    override fun errorAdd(msg: String) {
    }

```

```

        override fun onSuccessUpdate(msg: String) {
        }

        override fun onErrorUpdate(msg: String) {
        }
    }
}

```

## 11. Dan terakhir kita edit file UpdateAddActivitynya dengan mengetikan code berikut :

UpdateAddActivity

```

package
com.example.crudmysql

import android.annotation.SuppressLint
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.example..crudmysql.model.DataItem
import com.example..crudmysql.presenter.CrudView
import com.example..crudmysql.presenter.Presenter
import kotlinx.android.synthetic.main.activity_update_add.*
import org.jetbrains.anko.sdk27.coroutines.onClick
import org.jetbrains.anko.startActivity

@Suppress("SENSELESS_COMPARISON")
class UpdateAddActivity : AppCompatActivity(), CrudView {

    private lateinit var presenter: Presenter
    @SuppressLint("SetTextI18n")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_update_add)

        presenter = Presenter(this)
        val itemDataItem = intent.getSerializableExtra("dataItem")

        if (itemDataItem == null){
            btnAction.text = "Tambah"
            btnAction.onClick {
                presenter.addData(
                    etName.text.toString(),
                    etPhone.text.toString(),
                    etAlamat.text.toString()
                )
            }
        }
    }
}

```

```

    }

    }else if (itemDataItem != null){
        btnAction.text = "Update"
        val item = itemDataItem as DataItem?
        etName.setText(item?.staffName.toString())
        etPhone.setText(item?.staffHp.toString())
        etAlamat.setText(item?.staffAlamat.toString())
        btnAction.onClick {
            presenter.updateData(
                item?.staffId ?: "",
                etName.text.toString(),
                etPhone.text.toString(),
                etAlamat.text.toString())
            finish()
        }
    }
}

override fun successAdd(msg: String) {
    startActivity<MainActivity>()
    finish()
}

override fun errorAdd(msg: String) {}

override fun onSuccessUpdate(msg: String) {
    startActivity<MainActivity>()
    finish()
}

override fun onErrorUpdate(msg: String) {}

override fun onSuccessGet(data: List<DataItem>?) {}

override fun onFailedGet(msg: String) {}

override fun onSuccessDelete(msg: String) {}

override fun onErrorDelete(msg: String) {}
}

```

12. Jangan lupa kita masukan permission internet pada Manifest.xml dan untuk pengecekan pastikan smartphone kalian terkoneksi dengan jaringan yang sama dengan jaringan laptop

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