

A. Tujuan

Pada modul ini akan membahas cara perangkat android agar terhubung ke jaringan dan melakukan komunikasi dengan server, dengan tujuan agar mahasiswa:

- Mengetahui cara untuk mengakses jaringan melalui aplikasi android yang dikembangkan
- Mengetahui cara kerja Client Server
- Membuat aplikasi client server
- Menampilkan hasil komunikasi dengan server pada recyclerview
- Mengetahui cara membuat Client Server yang menggunakan data gambar untuk komunikasi get, post, put dan delete.

B. DasarTeori

1. Menghubungkan Perangkat Android ke Jaringan / Network

Pada modul kali ini akan dibahas cara untuk membuat aplikasi yang dapat terkoneksi dengan jaringan. Aplikasi yang terkoneksi ke jaringan memiliki beberapa manfaat yaitu dapat berkomunikasi dengan layanan yang lain, misalnya ingin mendapatkan/mengedit/menghapus data yang ada di server, hal ini hanya dapat dilakukan jika aplikasi yang dikembangkan terkoneksi dengan jaringan.

Terdapat 3 modul yang akan membahas membuat aplikasi yang berkaitan dengan jaringan, mulai dari hal yang sederhana agar aplikasi dapat mengakses jaringan sampai pada hal-hal yang lebih kompleks yaitu cara untuk bertransaksi / berkomunikasi dengan server. Data yang digunakan terdiri dari string dan image.

Syarat mutlak jika ingin membuat aplikasi yang terkoneksi ke jaringan adalah harus mendefinisikan uses-permission berikut pada manifest.

<code><uses-permission android:name="android.permission.INTERNET" /></code>	Permission ini berguna agar aplikasi yang dibuat dapat membuka <i>network socket</i>
---	--

Sedangkan uses-permission berikut jika akan membuat aplikasi yang dapat mengetahui *network state*.

<code><uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" ></code>	Permission ini berguna agar aplikasi dapat mengakses informasi tentang jaringan, misalnya apakah perangkat android sedang terkoneksi ke jaringan atau tidak dan juga dapat digunakan untuk mengetahui perangkat android sedang terkoneksi dengan wifi atau tidak.
--	---

2. REST

Pada modul networking ini, akan dikembangkan aplikasi yang memanfaatkan teknologi REST untuk berkomunikasi antar platform yang berbeda yaitu Android dan Rest Server yang ada di laptop masing-masing. Maka kita perlu tahu dahulu pengertian dari REST.

REST (Representational State Transfer) adalah istilah yang diciptakan oleh Roy Fielding untuk mendeskripsikan gaya arsitektur (architecture style) dari sistem jaringan / aplikasi web. REST merupakan sebuah pendekatan untuk membuat web service. Jadi di dalam REST terdapat 2 bagian besar yaitu REST Server yang berfungsi sebagai server untuk memberikan layanan kepada client/ REST Client. Layanan yang dapat diberikan REST Server beraneka ragam tergantung kebutuhan misalnya untuk menyimpan data/gambar seperti instagram.

I. Rest Server

Rest server yang akan dibuat pada modul ini akan menggunakan PHP menggunakan framework codeigniter. Dan menggunakan library Rest Server dari url <https://github.com/chriskacerguis/codeigniter-restserver>.

II. Rest Client -> Retrofit

Retrofit adalah REST Client untuk Android dan Java yang dikembangkan oleh Square. Library ini mempermudah untuk mendapatkan/*retrieve* atau mengunggah/*upload* JSON (atau representasi Rest yang lain) untuk berkomunikasi melalui webservice. Dengan menggunakan Retrofit dapat menggunakan beberapa converter, untuk memproses representasi data antara lain:

- Gson: com.squareup.retrofit:converter-gson
- Jackson: com.squareup.retrofit:converter-jackson
- Moshi: com.squareup.retrofit:converter-moshi
- Protobuf: com.squareup.retrofit:converter-protobuf
- Wire: com.squareup.retrofit:converter-wire
- Simple XML: com.squareup.retrofit:converter-simplexml

Untuk menggunakan Retrofit pada dasarnya memerlukan 3 kelas yaitu:

- Kelas Model yang digunakan untuk memetakan representasi data misalnya JSON
- Interface yang berisi komunikasi / operasi HTTP antara lain post, put, get dan delete
- Dan yang ketiga adalah Retrofit.Builder class adalah yang kelas yang menggunakan kelas Interface. Pada Builder API didefinisikan URL untuk komunikasi / operasi HTTP

Sebagai informasi tambahan kalian dapat membuat Model dari JSON dengan menggunakan URL berikut: <http://www.jsonschema2pojo.org/>.

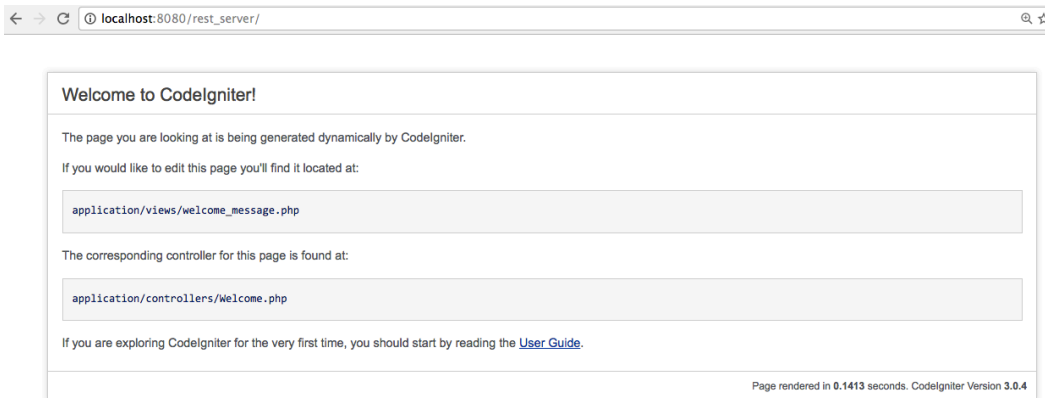
3. Setting Rest Server

I. Pada tahap awal kita akan membuat Rest Server yang akan digunakan oleh Rest Client. Langkah-langkahnya adalah sebagai berikut :

- a. Download code igniter dari url www.codeigniter.com.
- b. Ekstrak file codeigniter yang sudah di download
- c. Rename file tersebut menjadi rest_server
- d. Pindahkan folder tersebut kedalam htdocs
- e. Kemudian coba buka web browser masing-masing dengan menggunakan url : http://localhost:8080/rest_server, maka akan muncul tampilan seperti berikut:

1. silahkan download codeigniter pada situs resminya www.codeigniter.com
2. kemudian ekstrak/ unzip file CodeIgniter-3.1.0.zip.
3. setelah itu silahkan rename folder CodeIgniter-3.1.0 menjadi rest_server
4. pindahkan folder api tersebut ke folder yang akan diindeks oleh web server, misalnya C:\xampp\htdocs.

5. sekarang silahkan bukan web browser anda dan masukan `http://localhost/rest_server/` pada web address maka akan muncul seperti ini :



- Langkah selanjutnya adalah kalian konfigurasi untuk koneksi pada folder `application/config/database.php`.

```
database.php
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => "",
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => "",
    'database' => 'beli_tiket',
    '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
);
```

- Setting selanjutnya yang dilakukan:

 1. Hilangkan `index.php` pada `rest_server`, dengan cara:
 - i. buka `application/config/config.php` dan cari pengaturan `index_page` seperti dibawah ini

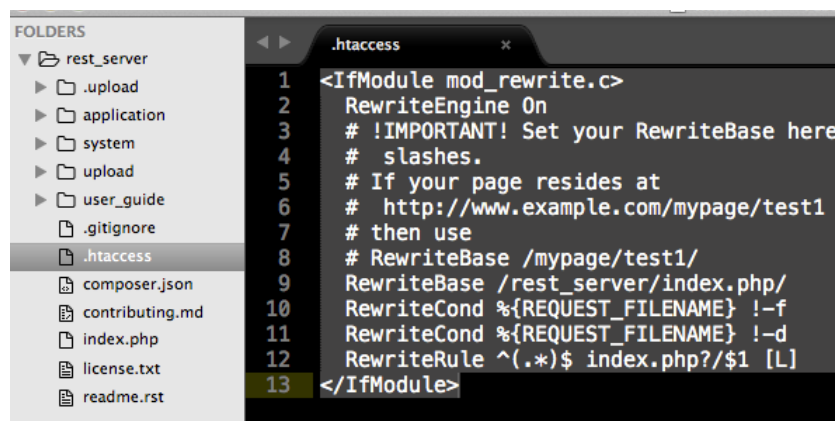
```
$config['index_page'] = 'index.php';
```

hapus `index.php` sehingga menjadi

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

- ii. Selanjutnya buat .htaccess pada direktori root codeigniter. Caranya dengan membuat file baru dengan nama .htaccess (tanpa ekstensi karena htaccess adalah file tanpa esktensi). Dan isi .htaccess seperti berikut:

```
<IfModule mod_rewrite.c>
RewriteEngine On
# !IMPORTANT! Set your RewriteBase here and don't forget trailing and leading
# slashes.
# If your page resides at
# http://www.example.com/mypage/test1
# then use
# RewriteBase /mypage/test1/
RewriteBase /rest_server/index.php/
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?/$1 [L]
</IfModule>
```



Gambar 1. Ilustrasi Letak .htaccess

2. Buat controller untuk get, insert, update tabel pembeli seperti berikut:

```

v beli_tiket pembeli
id_pembeli : char(20)
nama : varchar(30)
alamat : varchar(30)
telpn : varchar(30)
photo_id : varchar(100)

```

```
pembeli.php
<?
require APPPATH . '/libraries/REST_Controller.php';
class pembeli extends REST_Controller {
    // $this->response(array("status"=>"success","result" => $get_pembeli));
    // $this->response(array("status"=>"success"));
    function user_get(){
        $get_pembeli = $this->db->query("SELECT p.id_pembeli, p.nama, p.alamat,
p.telpn, p.photo_id FROM pembeli as p")->result();
        $this->response(array("status"=>"success","result" => $get_pembeli));
    }

    function user_post() {
```

```

$action = $this->post('action');
$data_pembeli = array(
    'id_pembeli' => $this->post('id_pembeli'),
    'nama'       => $this->post('nama'),
    'alamat'     => $this->post('alamat'),
    'telpn'      => $this->post('telpn'),
    'photo_id'   => $this->post('photo_id')
);

if ($action==='post'){
    $this->insertPembeli($data_pembeli);
}else if ($action==='put'){
    $this->updatePembeli($data_pembeli);
}else if ($action==='delete'){
    $this->deletePembeli($data_pembeli);
}else{
    $this->response(array("status"=>"failed","message" => "action harus
diisi"));
}

function insertPembeli($data_pembeli){
    //function upload image
    $uploadaddir = str_replace("application/", "", APPPATH).'upload/';
    if(!file_exists($uploadaddir) && !is_dir($uploadaddir)) {
        echo mkdir($uploadaddir, 0750, true);
    }

    if (!empty($_FILES)){
        $path = $_FILES['photo_id']['name'];
        $ext = pathinfo($path, PATHINFO_EXTENSION);
        // $user_img = time() . rand() . '.' . $ext;
        $user_img = $data_pembeli['id_pembeli'] . '.' . "png";
        $uploadfile = $uploadaddir . $user_img;

        $data_pembeli['photo_id'] = "upload/".$user_img;
    }else{
        $data_pembeli['photo_id']="";
    }

    //////////////////////////////////////
    //////////////////////////////////////
    //cek validasi
    if (empty($data_pembeli['id_pembeli'])){
        $this->response(array('status' => "failed", "message"=>"Id Pembeli harus
diisi"));
    }else if (empty($data_pembeli['nama'])){
        $this->response(array('status' => "failed", "message"=>"nama harus
diisi"));
    }else if (empty($data_pembeli['alamat'])){
        $this->response(array('status' => "failed", "message"=>"alamat harus
diisi"));
    }else if (empty($data_pembeli['telpn'])){
        $this->response(array('status' => "failed", "message"=>"telpn harus
diisi"));
    }
    else{
        $get_pembeli_baseid = $this->db->query("SELECT * FROM pembeli as p WHERE
p.id_pembeli='".$data_pembeli['id_pembeli']."'")->result();
        if(empty($get_pembeli_baseid)){

            $insert= $this->db->insert('pembeli',$data_pembeli);

            if (!empty($_FILES)){
                if ($_FILES["photo_id"]["name"]) {
                    if
(move_uploaded_file($_FILES["photo_id"]["tmp_name"],$uploadfile))
                    {
                        $insert_image = "success";
                    } else{
                        $insert_image = "failed";
                    }
                }
            }
        }
    }
}

```

```

        }else{
            $insert_image = "Image Tidak ada Masukan";
        }
        $data_pembeli['photo_id'] = base_url()."upload/".$user_img;
    }else{
        $data_pembeli['photo_id'] = "";
    }

    if ($insert){
        $this->response(array('status'=>'success','result' =>
array($data_pembeli),"message"=>$insert));
    }
    }else{
        $this->response(array('status' => "failed", "message"=>"Id_pembeli
sudah ada"));
    }
}

function updatePembeli($data_pembeli){
    //function upload image
    $upload_dir = str_replace("application/", "", APPPATH).'upload/';
    if(!file_exists($upload_dir) && !is_dir($upload_dir)) {
        echo mkdir($upload_dir, 0750, true);
    }
    if(!empty($_FILES)){
        $path = $_FILES['photo_id']['name'];
        // $ext = pathinfo($path, PATHINFO_EXTENSION);
        // $user_img = time() . rand() . '.' . $ext;
        $user_img = $data_pembeli['id_pembeli'].'.'.' ".png";
        $uploadfile = $upload_dir . $user_img;

        $data_pembeli['photo_id'] = "upload/".$user_img;
    }

    // $this->response(array(base_url()."upload/".$user_img));
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    //cek validasi
    if (empty($data_pembeli['id_pembeli'])) {
        $this->response(array('status' => "failed", "message"=>"Id Pembeli harus
diisi"));
    }
    }else if (empty($data_pembeli['nama'])) {
        $this->response(array('status' => "failed", "message"=>"nama harus
diisi"));
    }
    }else if (empty($data_pembeli['alamat'])) {
        $this->response(array('status' => "failed", "message"=>"alamat harus
diisi"));
    }
    }else if (empty($data_pembeli['telpon'])) {
        $this->response(array('status' => "failed", "message"=>"telpon harus
diisi"));
    }
    }else{
        $get_pembeli_baseid = $this->db->query("SELECT * FROM pembeli as p WHERE
p.id_pembeli='".$data_pembeli['id_pembeli']."'");
        if(empty($get_pembeli_baseid)){
            $this->response(array('status' => "failed", "message"=>"Id_pembeli
Tidak ada dalam database"));
        }else{
            // $this->response(unlink($uploadfile));
            //cek apakah image
            if (!empty($_FILES["photo_id"]["name"])) {
                if
(move_uploaded_file($_FILES["photo_id"]["tmp_name"],$uploadfile)){
                    $insert_image = "success";
                } else{
                    $insert_image = "failed";
                }
            }
        }
    }else{
        $insert_image = "Image Tidak ada Masukan";
    }
}

```

```

    }

    if ($insert_image=="success"){
        //jika photo di update eksekusi query
        $update= $this->db->query("Update pembeli Set nama
        ='".$data_pembeli['nama'].",'', alamat ='".$data_pembeli['alamat'].",'', telpn
        ='".$data_pembeli['telpn'].'", photo_id ='".$data_pembeli['photo_id'].'" Where id_pembeli
        ='".$data_pembeli['id_pembeli'].'"");

        $data_pembeli['photo_id'] = base_url()."upload/".$user_img;
    }else{
        //jika photo di kosong atau tidak di update eksekusi query
        $update= $this->db->query("Update pembeli Set nama
        ='".$data_pembeli['nama'].",'', alamat ='".$data_pembeli['alamat'].",'', telpn
        ='".$data_pembeli['telpn'].'" Where id_pembeli ='".$data_pembeli['id_pembeli'].'"");

        $getPhotoPath = $this->db->query("SELECT photo_id
        FROM pembeli Where id_pembeli='".$data_pembeli['id_pembeli'].'"")->result();

        if(!empty($getPhotoPath)){
            foreach ($getPhotoPath as $row)
            {
                $user_img = $row->photo_id;
                $data_pembeli['photo_id'] =
                base_url()."upload/".$user_img;
            }
        }

        if ($update){
            $this->response(array('status'=>'success','result' =>
            array($data_pembeli),"message"=>$update));
        }
    }
}

function deletePembeli($data_pembeli){
    if (empty($data_pembeli['id_pembeli'])){
        $this->response(array('status' => "failed", "message"=>"Id Pembeli harus
        diisi"));
    }
    else{
        $getPhotoPath = $this->db->query("SELECT photo_id FROM pembeli Where
        id_pembeli='".$data_pembeli['id_pembeli'].'"")->result();

        if(!empty($getPhotoPath)){
            foreach ($getPhotoPath as $row)
            {
                $path = str_replace("application/", "",
                APPPATH).$row->photo_id;
            }
            //delete image
            unlink($path);

            $this->db->query("Delete From pembeli Where
            id_pembeli='".$data_pembeli['id_pembeli'].'"");
            $this->response(array('status'=>'success',"message"=>"Data id =
            ".$data_pembeli['id_pembeli']. " berhasil di delete "));
        } else{
            $this->response(array('status'=>'fail',"message"=>"Id
            Pembeli tidak ada dalam database"));
        }
    }
}
}

```

4. Rest Client Menggunakan Retrofit Untuk Transaksi Data Image/Citra Pada Device Android dan Menampilkan Hasilnya pada RecyclerView

Pada modul networking ke 1 dan 2 telah dibuat project android yang dapat berkomunikasi dengan server dan menampilkan hasilnya pada recyclerview. Pada modul ketiga ini akan dibahas cara untuk berkomunikasi data berupa file yang mana file tersebut pada project ini adalah image/citra.

I. Ubah Manifest Pada Project Android

Citra yang digunakan transaksi berasal dari external storage perangkat android pengguna, oleh sebab itu pada manifest project android harus ditambahkan permission untuk membaca external storage seperti berikut:

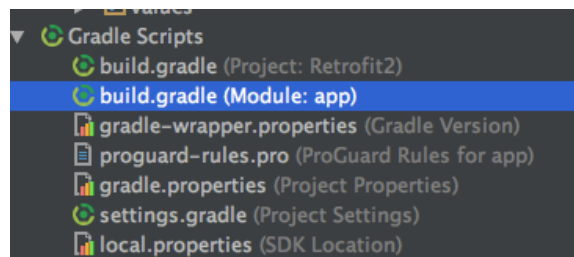
```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

Sehingga secara keseluruhan permission yang digunakan pada project ini adalah:

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

II. Ubah build.gradle

Langkah selanjutnya adalah menambahkan library untuk meload image pada aplikasi yang dikembangkan. Library yang akan digunakan didefinisikan pada file build.gradle.



Gambar 2. Letak build.gradle Pada Project Android

Library yang umum digunakan untuk membantu untuk meload image adalah Picasso dan Glide. Dengan menggunakan library tersebut akan memudahkan programmer untuk menghandle image/citra yang digunakan komunikasi antara client dan server. Kemudahan yang didapat programmer adalah manajemen memory pada aplikasi ketika menampilkan image ke ImageView , upload ke server dan download dari server. Jika tanpa kedua library tersebut developer harus mengatur manajemen memory secara manual dan sering aplikasi yang menggunakan data gambar force close karena leak of memory. Cara menambahkan library Glide dan Picasso ke build.gradle adalah sebagai berikut:

Tabel 1. File build.gradle

build.gradle
<pre>apply plugin: 'com.android.application' android { compileSdkVersion 23 buildToolsVersion "23.0.3" defaultConfig { applicationId "demo.retrofit"</pre>


```

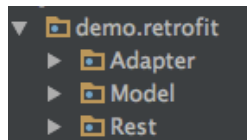
minSdkVersion 14
targetSdkVersion 23
versionCode 1
versionName "1.0"
testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
}
buildTypes {
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
rules.pro'
    }
}
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:23.4.0'
    testCompile 'junit:junit:4.12'

    // retrofit, gson
    compile 'com.google.code.gson:gson:2.6.2'
    compile 'com.squareup.retrofit2:retrofit:2.0.2'
    compile 'com.squareup.retrofit2:converter-gson:2.0.2'
    //recyclerview
    compile 'com.android.support:recyclerview-v7:23.4.0'
    compile 'com.squareup.picasso:picasso:2.5.2'
    compile 'com.github.bumptech.glide:glide:3.7.0'
}

```

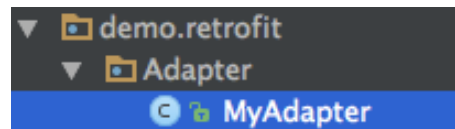
III. Struktur Project Android

Langkah selanjutnya merubah struktur project android sehingga lebih rapi. Karena project android yang dikembangkan semakin besar maka akan lebih jika strukturnya dirubah dengan cara mengelompokkan file yang berfungsi hampir sama kedalam satu package, contohnya adapter, model dan rest.



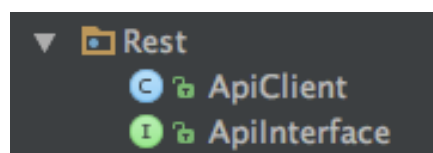
Gambar 3. Struktur File Pada Project Android

Package Adapter berisi adapter dari RecyclerView seperti berikut:



Gambar 4. Struktur Package Adapter Pada Project Android

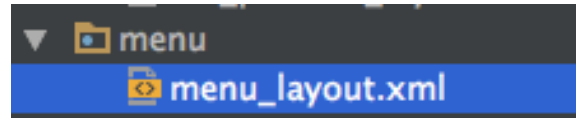
Package model berisi model atau kelas POJO dan Rest berisi file yang digunakan untuk komunikasi dengan server isi adalah sebagai berikut:



Gambar 5. Struktur Package Rest Pada Project Android

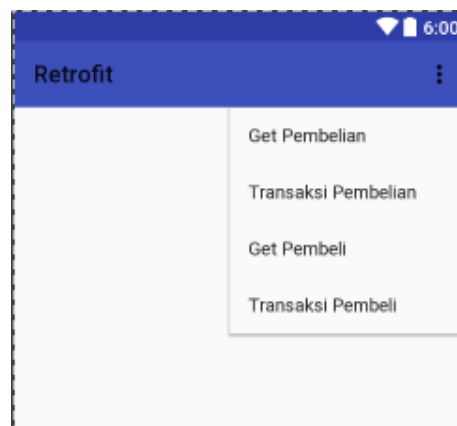
IV. Buat Menu

Karena project semakin besar dan terdiri dari beberapa fungsi maka akan lebih baik dibuat menu, sehingga memudahkan pengguna untuk berpindah activity. Langkah pertama buat menu_layout.xml pada folder menu di dalam folder res.



Gambar 6. Struktur File menu_layout.xml Pada Project Android

```
menu_layout.xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/menuGetPembelian"
        android:title="Get Pembelian"/>
    <item android:id="@+id/menuTransaksiPembelian"
        android:title="Transaksi Pembelian"/>
    <item android:id="@+id/menuGetPembeli"
        android:title="Get Pembeli"/>
    <item android:id="@+id/menuTransaksiPembeli"
        android:title="Transaksi Pembeli"/>
</menu>
```

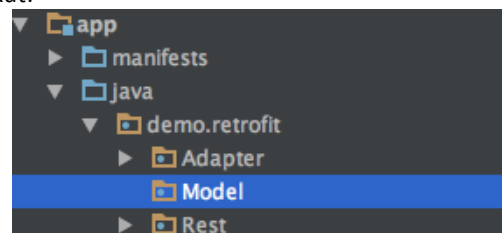


Gambar 7. Preview menu_layout.xml

V. Buat Kelas Pojo untuk data Pembeli

Pada modul ketiga ini akan bertransaksi menggunakan data pembeli maka harus membuat kelas baru untuk menampung data dari table pembeli sehingga memudahkan ketika membuat aplikasi.

Beri nama kelas tersebut kelas Pembeli yang dimasukkan kedalam package model, hasil akhir kelas Pembeli adalah sebagai berikut:



Pembeli.java

```
package demo.retrofit.Model;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

/**
 * Created by alhamdulillah on 11/2/16.
 */

public class Pembeli {
    @SerializedName("id_pembeli")
    private String idPembeli;
    @SerializedName("nama")
    private String nama;
    @SerializedName("alamat")
    private String alamat;
    @SerializedName("telpn")
    private String telpn;
    @SerializedName("photo_id")
    private String photoId;
    private String action;

    public Pembeli() {}

    public Pembeli(String idPembeli, String nama, String alamat, String telpn, String
photoId, String action) {
        this.idPembeli = idPembeli;
        this.nama = nama;
        this.alamat = alamat;
        this.telpn = telpn;
        this.photoId = photoId;
        this.action = action;
    }

    public String getIdPembeli() {
        return idPembeli;
    }

    public void setIdPembeli(String idPembeli) {
        this.idPembeli = idPembeli;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getAlamat() {
        return alamat;
    }

    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    public String getTelpn() {
        return telpn;
    }

    public void setTelpn(String telpn) {
        this.telpn = telpn;
    }

    public String getPhotoId() {
        return photoId;
    }

    public void setPhotoId(String photoId) {
```

```

        this.photoId = photoId;
    }

    public String getAction() {
        return action;
    }

    public void setAction(String action) {
        this.action = action;
    }
}

```

Selanjut buat kelas satu lagi yang digunakan untuk menampung response dari server. Beri nama kelas tersebut ResultPembeli, source code dari kelas ResultPembeli adalah sebagai berikut:

```

ResultPembeli.java
package demo.retrofit.Model;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

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

/**
 * Created by alhamdulillah on 11/2/16.
 */

public class ResultPembeli {
    @SerializedName("status")
    private String status;
    @SerializedName("result")
    private List<Pembeli> result = new ArrayList<Pembeli>();
    @SerializedName("message")
    private String message;
    public ResultPembeli() {}

    public String getStatus() {
        return status;
    }

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

    public List<Pembeli> getResult() {
        return result;
    }

    public void setResult(List<Pembeli> result) {
        this.result = result;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }
}

```

VI. Ubah ApiInterface dan ApiClient pada Package Rest

ApiClient.java diubah Base_URLnya dengan menghilangkan index.php dan sesuaikan ipaddress dan port webserver sesuai dengan komputer masing-masing.

```
ApiClient.java
package demo.retrofit.Rest;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

/**
 * Created by alhamdulillah on 10/23/16.
 */

public class ApiClient {

    public static final String BASE_URL = "http://192.168.1.102:8080/rest_server/";
    private static Retrofit retrofit = null;

    public static Retrofit getClient() {
        if (retrofit==null) {
            Gson gson = new GsonBuilder()
                .setLenient()
                .create();

            retrofit = new Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create(gson))
                .build();
        }
        return retrofit;
    }
}
```

ApiInterface.java diubah isinya menyesuaikan dengan transaksi yang digunakan untuk get, post, put dan delete data pembeli. Source codenya adalah sebagai berikut:

```
ApiInterface.java
package demo.retrofit.Rest;

import demo.retrofit.Model.ResultPembeli;
import demo.retrofit.Model.GetPembelian;
import demo.retrofit.Model.PostPutDelPembelian;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.GET;
import retrofit2.http.HTTP;
import retrofit2.http.Multipart;
import retrofit2.http.POST;
import retrofit2.http.PUT;
import retrofit2.http.Part;

/**
 * Created by alhamdulillah on 10/23/16.
 */

public interface ApiInterface {
    @GET("pembelian/user")
    Call<GetPembelian> getPembelian();

    @FormUrlEncoded
    @POST("pembelian/user")
    Call<PostPutDelPembelian> postPembelian(@Field("id_pembelian") String idPembelian,
    @Field("id_pembeli") String idPembeli,
```

```

@Field("total_harga") String totalHarga,
@Field("tanggal_beli") String tanggalBeli,
@Field("id_tiket") String idTiket);

@FormUrlEncoded
@PUT("pembelian/user")
Call<PostPutDelPembelian> putPembelian(@Field("id_pembelian") String idPembelian,
@Field("id_pembeli") String idPembeli,
@Field("tanggal_beli") String tanggalBeli,
@Field("total_harga") String totalHarga,
@Field("id_tiket") String idTiket);

@FormUrlEncoded
@HTTP(method = "DELETE", path = "pembelian/user", hasBody = true)
Call<PostPutDelPembelian> deletePembelian(@Field("id_pembelian") String idPembelian);

/*****/
@GET("pembeli/user")
Call<ResultPembeli> getPembeli();

@Multipart
@POST("pembeli/user")
Call<ResultPembeli> postPembeli(@Part MultipartBody.Part file, @Part("id_pembeli")
RequestBody idPembeli, @Part("nama") RequestBody nama,
@Part("alamat") RequestBody alamat, @Part("telpn")
RequestBody telpn, @Part("action") RequestBody action);

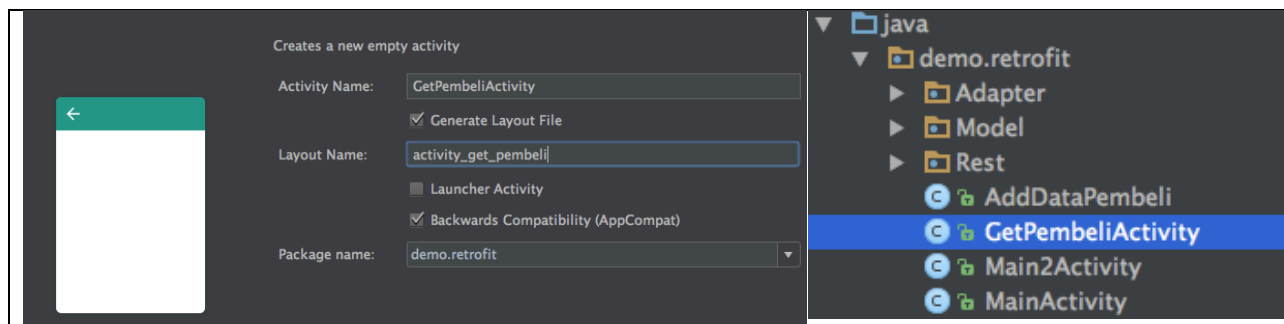
@Multipart
@POST("pembeli/user")
Call<ResultPembeli> putPembeli(@Part MultipartBody.Part file, @Part("id_pembeli")
RequestBody idPembeli, @Part("nama") RequestBody nama,
@Part("alamat") RequestBody alamat, @Part("telpn")
RequestBody telpn, @Part("action") RequestBody action);

@Multipart
@POST("pembeli/user")
Call<ResultPembeli> deletePembeli(@Part("id_pembeli") RequestBody idPembeli,
@Part("action") RequestBody action);
}

```

VII. Kelas untuk menampilkan data Pembeli

Selanjutnya buat kelas untuk menampilkan data yang ada di server, beri nama kelas tersebut `GetPembeliActivity.java`. Ilustrasinya dapat dilihat pada Gambar 8. Pada kelas ini `GetPembeliActivity.java` ini akan berisi `RecyclerView`, oleh karena itu ada 3 komponen yang harus dibuat yaitu `layout`, `layout manager` dan `adapter`.



Gambar 8. Kelas `GetPembeliActivity.java` Pada Project Android

Buat `layout list_pembeli_layout.xml` pada folder `res`. `Layout list_pembeli_layout.xml` digunakan untuk menampilkan setiap item pada `RecyclerView`.

```
list_pembeli_layout.xml
```

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

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp">

        <TextView
            android:id="@+id/tvNama"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:hint="Nama"
            android:textColor="@android:color/holo_green_dark"
            android:textSize="20sp"
            android:textStyle="bold"/>

        <ImageView
            android:id="@+id/imgPembeli"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:layout_below="@+id/tvNama"
            android:layout_marginTop="10dp"
            android:src="@drawable/photoid"/>

        <TextView
            android:id="@+id/tvIdPembeli"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignTop="@+id/imgPembeli"
            android:layout_below="@id/tvIdPembelian"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@+id/imgPembeli"
            android:hint="Id Pembeli"
            android:textSize="20sp"/>

        <TextView
            android:id="@+id/tvAlamat"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tvIdPembeli"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@+id/imgPembeli"
            android:hint="Alamat :"
            android:textSize="20sp"
            android:textStyle="bold"/>

        <TextView
            android:id="@+id/tvAlamatContent"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tvAlamat"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@+id/imgPembeli"
            android:hint="Contoh Jalan danau Maninjau"
            android:textSize="20sp"/>

        <TextView
            android:id="@+id/tvTelpn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tvAlamatContent"
            android:layout_marginTop="5dp"
            android:layout_toRightOf="@+id/imgPembeli"
            android:hint="Telpn :"
```

```

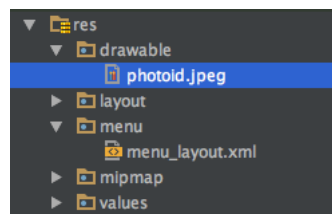
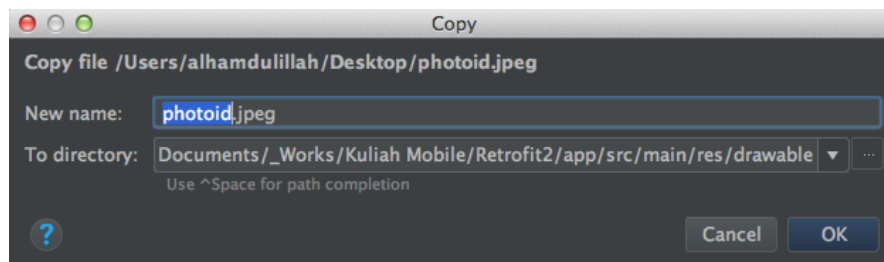
        android:textSize="20sp"
        android:textStyle="bold"/>
    <TextView
        android:id="@+id/tvTelpnContent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tvTelpn"
        android:layout_marginTop="5dp"
        android:layout_toRightOf="@+id/imgPembeli"
        android:hint="Contoh 0861577909"
        android:textSize="20sp"/>

</RelativeLayout>

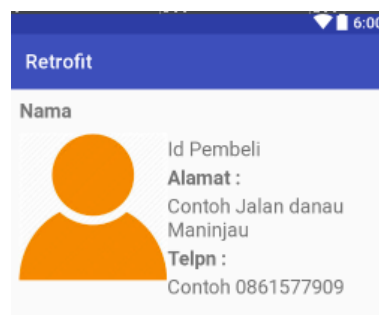
<View
    android:layout_width="match_parent"
    android:layout_height="5dp"
    android:layout_marginTop="10dp"
    android:background="@color/colorPrimary"/>
</LinearLayout>

```

*Jika terjadi error pada list_pembeli_layout.xml, karena pada drawable belum ada photoid. Jadi perlu ditambah file image pada drawable dengan nama photoid.



Ilustrasi dari list_pembeli_layout.xml jika tidak terjadi error dapat dilihat pada Gambar 9

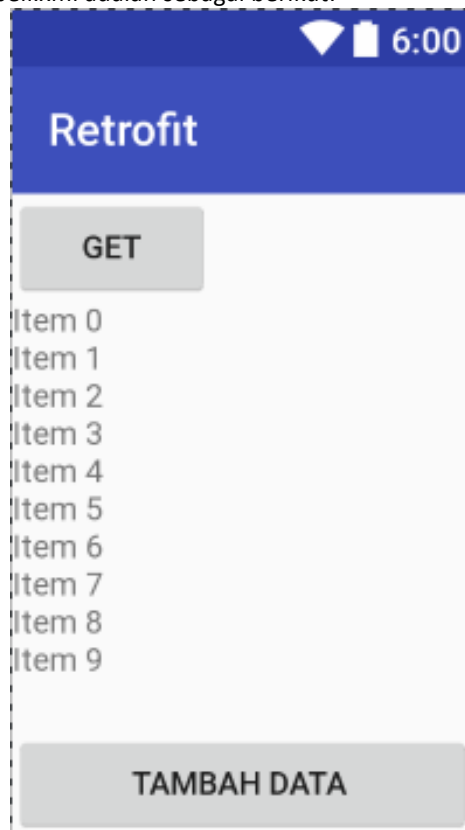


Gambar 9. Preview list_pembeli_layout.xml

Setelah membuat list layout recycler view selanjutnya definisikan RecyclerView pada layout kelas GetPembeliActivity.java yaitu activity_get_pembeli.xml.


```
activity_get_pembeli.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"  
    android:id="@+id/activity_get_pembelian"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context="demo.retrofit.GetPembeliActivity">  
    <Button  
        android:id="@+id/btGet"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Get"/>  
    <android.support.v7.widget.RecyclerView  
        android:id="@+id/recyclerView"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:layout_below="@+id/btGet"  
        android:layout_above="@+id/btAddData">  
    </android.support.v7.widget.RecyclerView>  
    <Button  
        android:id="@+id/btAddData"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="Tambah Data"  
        android:layout_alignParentBottom="true"/>  
</RelativeLayout>
```

Ilustrasi activity_get_pembeli.xml adalah sebagai berikut:



Gambar 10. Preview activity_get_pembeli.xml

Komponen kedua yang harus ketiga menggunakan RecyclerView adalah layout manager. Sourcecode final dari GetPembeliActivity.java adalah

```
GetPembeliActivity.java
package demo.retrofit;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

import java.util.List;

import demo.retrofit.Adapter.PembeliAdapter;
import demo.retrofit.Model.ResultPembeli;
import demo.retrofit.Model.Pembeli;
import demo.retrofit.Rest.ApiClient;
import demo.retrofit.Rest.ApiInterface;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class GetPembeliActivity extends AppCompatActivity {

    RecyclerView mRecyclerView;
    RecyclerView.Adapter mAdapter;
    RecyclerView.LayoutManager mLayoutManager;
    Context mContext;
    ApiInterface mApiInterface;
    Button btAddData, btGet;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_get_pembeli);
        mContext = getApplicationContext();
        mRecyclerView = (RecyclerView) findViewById(R.id.recyclerview);
        mLayoutManager = new LinearLayoutManager(mContext);
        mRecyclerView.setLayoutManager(mLayoutManager);
        btAddData = (Button) findViewById(R.id.btAddData);
        btGet = (Button) findViewById(R.id.btGet);

        btGet.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mApiInterface = ApiClient.getClient().create(ApiInterface.class);
                Call<ResultPembeli> mPembeliCall = mApiInterface.getPembeli();
                mPembeliCall.enqueue(new Callback<ResultPembeli>() {
                    @Override
                    public void onResponse(Call<ResultPembeli> call,
Response<ResultPembeli> response) {
                        Log.d("Get Pembeli", response.body().getStatus());
                        List<Pembeli> listPembeli = response.body().getResult();
                        mAdapter = new PembeliAdapter(listPembeli);
                        mRecyclerView.setAdapter(mAdapter);
                    }

                    @Override
                    public void onFailure(Call<ResultPembeli> call, Throwable t) {
                        Log.d("Get Pembeli", t.getMessage());
                    }
                });
            }
        });
    }
}
```

```

    });

    btAddData.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
//            Intent intent = new Intent(mContext, AddDataPembeli.class);
//            startActivity(intent);
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_layout, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    Intent mIntent;
    switch (item.getItemId()) {
        case R.id.menuGetPembelian:
            mIntent = new Intent(this, MainActivity.class);
            startActivity(mIntent);
            return true;

        case R.id.menuGetPembeli:
            mIntent = new Intent(this, GetPembeliActivity.class);
            startActivity(mIntent);
            return true;

        case R.id.menuTransaksiPembelian:
            mIntent = new Intent(this, Main2Activity.class);
            startActivity(mIntent);
            return true;

        case R.id.menuTransaksiPembeli:
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
}

```

Komponen ketiga yang harus dibuat adalah adapter, oleh karena itu tambahkan pada package adapter kelas PembeliAdapter.java. Sourcecode finalnya adalah sebagai berikut:

```

PembeliAdapter.java
package demo.retrofit.Adapter;

import android.content.Intent;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.bumptech.glide.Glide;

import java.util.List;

import demo.retrofit.Model.Pembeli;
import demo.retrofit.PutDelPembeliActivity;
import demo.retrofit.R;
import demo.retrofit.Rest.ApiClient;

/**
 * Created by 191hamdulillah on 11/2/16.
 */

```

```

public class PembeliAdapter extends RecyclerView.Adapter<PembeliAdapter.PembViewHolder> {
    List<Pembeli> listPembeli;

    public PembeliAdapter(List<Pembeli> listPembeli) {
        this.listPembeli = listPembeli;
    }

    @Override
    public PembeliAdapter.PembViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.list_pembeli_layout, parent,
false);
        PembViewHolder mHolder = new PembViewHolder(view);
        return mHolder;
    }

    @Override
    public void onBindViewHolder(PembeliAdapter.PembViewHolder holder, final int
position) {

        holder.tvIdPembeli.setText(listPembeli.get(position).getIdPembeli());
        holder.tvNama.setText(listPembeli.get(position).getNama());
        holder.tvAlamat.setText(listPembeli.get(position).getAlamat());
        holder.tvTelpn.setText(listPembeli.get(position).getTelpn());
        if (listPembeli.get(position).getPhotoId().length()>0) {
            //
            Picasso.with(holder.itemView.getContext()).load(ApiClient.BASE_URL+listPembeli.get(position)
.getPhotoId())
            //
                .into(holder.mPhotoid);

            Glide.with(holder.itemView.getContext()).load(ApiClient.BASE_URL+listPembeli.get(position)
.getPhotoId())
                .into(holder.mPhotoid);
        }else{
            //
            Picasso.with(holder.itemView.getContext()).load(R.drawable.photoid).into(holder.mPhotoid)
;

            Glide.with(holder.itemView.getContext()).load(R.drawable.photoid).into(holder.mPhotoid);
        }
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(view.getContext(),
PutDelPembeliActivity.class);
                intent.putExtra("id_pembeli",listPembeli.get(position).getIdPembeli());
                intent.putExtra("nama",listPembeli.get(position).getNama());
                intent.putExtra("alamat",listPembeli.get(position).getAlamat());
                intent.putExtra("telpn",listPembeli.get(position).getTelpn());
                intent.putExtra("photo_id",listPembeli.get(position).getPhotoId());
                view.getContext().startActivity(intent);
            }
        });
    }

    @Override
    public int getItemCount() {
        return listPembeli.size();
    }

    public class PembViewHolder extends RecyclerView.ViewHolder {
        ImageView mPhotoid;
        TextView tvIdPembeli, tvNama, tvAlamat, tvTelpn;

        public PembViewHolder(View itemView) {
            super(itemView);
            mPhotoid = (ImageView) itemView.findViewById(R.id.imgPembeli);
            tvIdPembeli = (TextView) itemView.findViewById(R.id.tvIdPembeli);
            tvNama = (TextView) itemView.findViewById(R.id.tvNama);
        }
    }
}

```

```

        tvAlamat = (TextView) itemView.findViewById(R.id.tvAlamatContent);
        tvTelpn = (TextView) itemView.findViewById(R.id.tvTelpnContent);
    }
}

```

- Jika terjadi error source code berikut dapat di comment sementara untuk menjalankan transaksi get pembeli, tetapi jangan lupa di uncomment lagi karena akan digunakan untuk menampilkan detail pembeli.

```

Intent intent = new Intent(view.getContext(), PutDelPembeliActivity.class);
intent.putExtra("id_pembeli", listPembeli.get(position).getIdPembeli());
intent.putExtra("nama", listPembeli.get(position).getNama());
intent.putExtra("alamat", listPembeli.get(position).getAlamat());
intent.putExtra("telpn", listPembeli.get(position).getTelpn());
intent.putExtra("photo_id", listPembeli.get(position).getPhotoId());
view.getContext().startActivity(intent);

```

- Dan pada manifest yang di load pertama adalah GetPembeliActivity, maka ubah file manifest seperti berikut:

AndroidManifest.xml

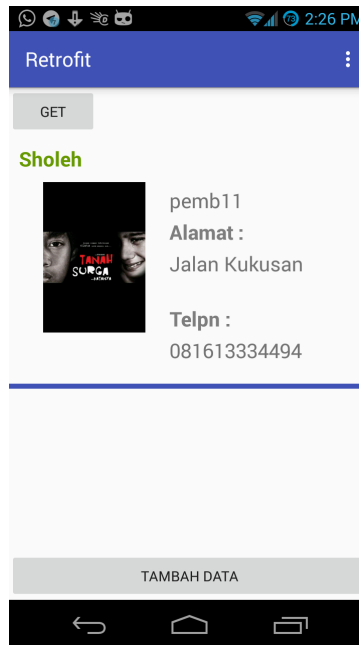
```

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

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            </activity>
        <activity android:name=".Main2Activity">
            </activity>
        <activity android:name=".GetPembeliActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>

```



Gambar 11. Preview Hasil Akhir GetPembeliActivity.java

VIII. Activity untuk menambahkan Data

Pada tahap ini akan dibuat activity untuk menambah data, tambahkan empty activity pada project android dengan nama AddDataPembeli.java. Pada GetPembeliActivity.java sudah ada button untuk menambah data sehingga untuk menghubungkan kedua activity tersebut dapat ditambahkan intent pada button add, menggunakan code berikut:

```
btAddData.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(mContext, AddDataPembeli.class);
        startActivity(intent);
    }
});
```

Setelah membuat activity AddDataPembeli.java, langkah selanjutnya adalah merubah layout activity_add_data_pembeli.xml, seperti tabel berikut:

```
activity_add_data_pembeli.xml
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_post_put_del_pembeli"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scrollbars="vertical"
    tools:context="demo.retrofit.PutDelPembeliActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp">

        <ImageView
            android:id="@+id/imgAddPhotoId"
            android:layout_width="200dp"
```

```

        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:src="@drawable/photoid"/>

<Button
    android:id="@+id/btAddPhotoId"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/imgAddPhotoId"
    android:layout_centerHorizontal="true"
    android:text="Select Photo Id"/>

<TextView
    android:id="@+id/tvAddIdPembeli"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/btAddPhotoId"
    android:layout_marginTop="10dp"
    android:text="ID Pembeli :"
    android:textSize="20dp"/>

<EditText
    android:id="@+id/edtAddIdPembeli"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/tvAddIdPembeli"
    android:layout_toRightOf="@+id/tvAddIdPembeli"
    android:hint="id_pembeli"/>

<TextView
    android:id="@+id/tvAddNama"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvAddIdPembeli"
    android:text="Nama :"
    android:textSize="20dp"/>

<EditText
    android:id="@+id/edtAddNamaPembeli"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/tvAddNama"
    android:layout_alignLeft="@+id/edtAddIdPembeli"
    android:layout_toRightOf="@+id/tvAddNama"
    android:hint="Nama"/>

<TextView
    android:id="@+id/tvAddAlamat"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvAddNama"
    android:text="Alamat :"
    android:textSize="20dp"/>

<EditText
    android:id="@+id/edtAddAlamatPembeli"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/tvAddAlamat"
    android:layout_alignLeft="@+id/edtAddIdPembeli"
    android:layout_toRightOf="@+id/tvAddAlamat"
    android:hint="Alamat"/>

<TextView
    android:id="@+id/tvAddTelpn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvAddAlamat"
    android:text="Telepon :"
    android:textSize="20dp"/>

```

```

<EditText
    android:id="@+id/edtAddTelpnPembeli"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/tvAddTelpn"
    android:layout_alignLeft="@+id/edtAddIdPembeli"
    android:layout_toRightOf="@+id/tvAddTelpn"
    android:hint="Telpn"
    android:inputType="phone"/>

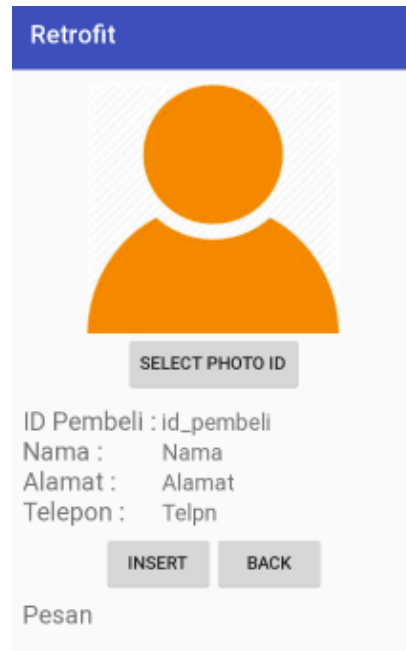
<LinearLayout
    android:id="@+id/LinearLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvAddTelpn"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="5dp"
    android:layout_marginTop="5dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btAddData"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Insert"/>

    <Button
        android:id="@+id/btAddBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Back"/>
</LinearLayout>

<TextView
    android:id="@+id/tvAddMessage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_below="@+id/LinearLayout"
    android:hint="Pesan"
    android:inputType="textMultiLine"
    android:textSize="20dp"/>
</RelativeLayout>
</ScrollView>

```

Gambar 12. Preview activity_add_data_pembeli.xml

Setelah membuat layoutnya, selanjutnya adalah mendefinisikan fungsi dari layout tersebut, misalnya untuk menampilkan data atau untuk menambah data, dan lain sebagainya. Source code akhirnya dapat dilihat pada tabel berikut:

```

AddDataPembeli.java
package demo.retrofit;

import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.bumptech.glide.Glide;

import java.io.File;

import demo.retrofit.Model.ResultPembeli;
import demo.retrofit.Rest.ApiClient;
import demo.retrofit.Rest.ApiInterface;
import okhttp3.MediaType;
import okhttp3.MultipartBody;
import okhttp3.RequestBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class AddDataPembeli extends AppCompatActivity {

    Context mContext;
    ImageView mImageView;

```

```

Button btAddPhotoId, btAddBack, btAddData;
EditText edtAddIdPembeli, edtAddNamaPembeli, edtAddAlamatPembeli;
EditText edtAddTelpnPembeli;
TextView tvAddMessage;
String imagePath = "";

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

    mContext = getApplicationContext();
    mImageView = (ImageView) findViewById(R.id.imgAddPhotoId);
    btAddPhotoId = (Button) findViewById(R.id.btAddPhotoId);
    edtAddIdPembeli = (EditText) findViewById(R.id.edtAddIdPembeli);
    edtAddNamaPembeli = (EditText) findViewById(R.id.edtAddNamaPembeli);
    edtAddAlamatPembeli = (EditText) findViewById(R.id.edtAddAlamatPembeli);
    edtAddTelpnPembeli = (EditText) findViewById(R.id.edtAddTelpnPembeli);

    btAddData = (Button) findViewById(R.id.btAddData);
    btAddBack = (Button) findViewById(R.id.btAddBack);
    tvAddMessage = (TextView) findViewById(R.id.tvAddMessage);

    btAddData.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            ApiInterface mApiInterface =
ApiClient.getClient().create(ApiInterface.class);

            MultipartBody.Part body = null;
            if (!imagePath.isEmpty()){
                //File creating from selected URL
                File file = new File(imagePath);

                // create RequestBody instance from file
                RequestBody requestFile =
RequestBody.create(MediaType.parse("multipart/form-data"), file);

                // MultipartBody.Part is used to send also the actual file name
                body = MultipartBody.Part.createFormData("photo_id", file.getName(),
requestFile);
            }
            RequestBody reqIdPembeli =
MultipartBody.create(MediaType.parse("multipart/form-data"),
(edtAddIdPembeli.getText().toString().isEmpty() == true) ? "" : edtAddIdPembeli.getText().toStr
ing());
            RequestBody reqNama =
MultipartBody.create(MediaType.parse("multipart/form-data"),
(edtAddNamaPembeli.getText().toString().isEmpty()) ? "" : edtAddNamaPembeli.getText().toStrin
g());
            RequestBody reqAlamat =
MultipartBody.create(MediaType.parse("multipart/form-data"),
(edtAddNamaPembeli.getText().toString().isEmpty()) ? "" : edtAddAlamatPembeli.getText().toStr
ing());
            RequestBody reqTelpn =
MultipartBody.create(MediaType.parse("multipart/form-data"),
(edtAddNamaPembeli.getText().toString().isEmpty()) ? "" : edtAddTelpnPembeli.getText().toStri
ng());
            RequestBody reqAction =
MultipartBody.create(MediaType.parse("multipart/form-data"),
"post");
            Call<ResultPembeli> mPembeliCall =
mApiInterface.postPembeli(body, reqIdPembeli, reqNama,
reqAlamat, reqTelpn, reqAction);
            mPembeliCall.enqueue(new Callback<ResultPembeli>() {
                @Override
                public void onResponse(Call<ResultPembeli> call,

```

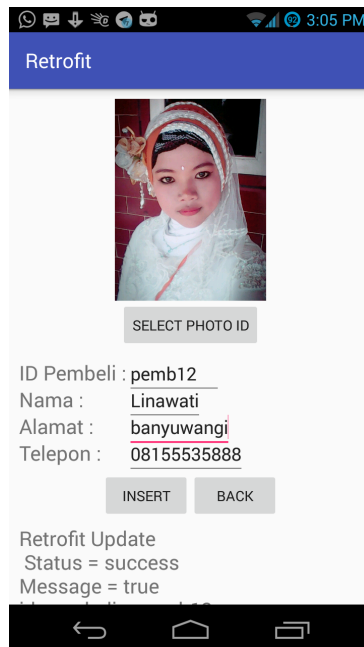
```

Response<ResultPembeli> response) {
//          Log.d("Insert Retrofit",response.body().getStatus());
          if (response.body().getStatus().equals("failed")){
              tvAddMessage.setText("Retrofit Update \n Status =
"+response.body().getStatus()+"\n"+
                                   "Message
"+response.body().getMessage()+"\n");
          }else{
              String detail = "\n"+
                              "id_pembeli
"+response.body().getResult().get(0).getIdPembeli()+"\n"+
                              "agama
"+response.body().getResult().get(0).getNama()+"\n"+
                              "alamat
"+response.body().getResult().get(0).getAlamat()+"\n"+
                              "telpn
"+response.body().getResult().get(0).getTelpn()+"\n"+
                              "photo_id
"+response.body().getResult().get(0).getPhotoId()+"\n";
              tvAddMessage.setText("Retrofit Update \n Status =
"+response.body().getStatus()+"\n"+
                                   "Message = "+response.body().getMessage()+detail);
          }
      }

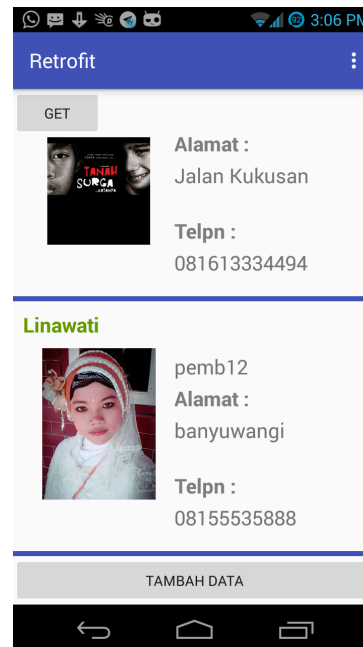
      @Override
      public void onFailure(Call<ResultPembeli> call, Throwable t) {
//          Log.d("Insert Retrofit", t.getMessage());
          tvAddMessage.setText("Retrofit Update \n Status = "+
t.getMessage());
      }
  });
  });
  btAddBack.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
          Intent intent = new Intent(mContext, GetPembeliActivity.class);
          startActivity(intent);
      }
  });
  btAddPhotoId.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
          final Intent galleryIntent = new Intent();
          galleryIntent.setType("image/*");
          galleryIntent.setAction(Intent.ACTION_PICK);
          Intent intentChoose = Intent.createChooser(galleryIntent, "Pilih Gambar
Untuk Di upload");
          startActivityForResult(intentChoose, 10);
      }
  });
  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent data) {
      super.onActivityResult(requestCode, resultCode, data);
      if (resultCode == RESULT_OK && requestCode ==10){
          if (data==null){
              Toast.makeText(mContext, "Gambar Gagal Di load",
Toast.LENGTH_LONG).show();
              return;
          }
          Uri selectedImage = data.getData();
          String[] filePathColumn = {MediaStore.Images.Media.DATA};
          Cursor cursor = getContentResolver().query(selectedImage, filePathColumn,
null, null, null);
          if (cursor != null) {
              cursor.moveToFirst();
              int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
              imagePath =cursor.getString(columnIndex);

```

```
//Picasso.with(mContext).load(new
File(imagePath)).fit().into(mImageView);
Glide.with(mContext).load(new File(imagePath)).into(mImageView);
cursor.close();
}else{
    Toast.makeText(mContext, "Gambar      Gagal      Di      load",
Toast.LENGTH_LONG).show();
}
}
}
```



Gambar 13. Preview Transaksi Insert



Gambar 14. Preview Hasil Telah Masuk ke Server

IX. Buat Activity untuk Transaksi Update dan Delete

*Catatan uncomment code berikut pada PembeliAdapter.java , pada package Adapter.

```
Intent intent = new Intent(view.getContext(), PutDelPembeliActivity.class);
intent.putExtra("id_pembeli",listPembeli.get(position).getIdPembeli());
intent.putExtra("nama",listPembeli.get(position).getNama());
intent.putExtra("alamat",listPembeli.get(position).getAlamat());
intent.putExtra("telpn",listPembeli.get(position).getTelpn());
intent.putExtra("photo_id",listPembeli.get(position).getPhotoId());
view.getContext().startActivity(intent);
```

Tambahkan activity PutDelPembeliActivity.java pada project android. Setelah itu ubah layout seperti berikut:

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

```

<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_post_put_del_pembeli"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scrollbars="vertical"
    tools:context="demo.retrofit.PutDelPembeliActivity">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp">

        <ImageView
            android:id="@+id/imgPhotoId"
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:layout_centerHorizontal="true"
            android:src="@drawable/photoid"/>

        <Button
            android:id="@+id/btPhotoId"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/imgPhotoId"
            android:layout_centerHorizontal="true"
            android:text="Select Photo Id"/>

        <TextView
            android:id="@+id/tvIdPembeli"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/btPhotoId"
            android:layout_marginTop="10dp"
            android:text="ID Pembeli : "
            android:textSize="20dp"/>

        <EditText
            android:id="@+id/edtIdPembeli"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/tvIdPembeli"
            android:layout_toRightOf="@+id/tvIdPembeli"
            android:hint="id_pembeli"
            android:enabled="false"/>

        <TextView
            android:id="@+id/tvNama"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tvIdPembeli"
            android:text="Nama : "
            android:textSize="20dp"/>

        <EditText
            android:id="@+id/edtNamaPembeli"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/tvNama"
            android:layout_alignLeft="@+id/edtIdPembeli"
            android:layout_toRightOf="@+id/tvNama"
            android:hint="Nama"/>

        <TextView
            android:id="@+id/tvAlamat"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tvNama"
            android:text="Alamat : "
            android:textSize="20dp"/>

```

```

<EditText
    android:id="@+id/edtAlamatPembeli"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/tvAlamat"
    android:layout_alignLeft="@+id/edtIdPembeli"
    android:layout_toRightOf="@+id/tvAlamat"
    android:hint="Alamat"/>

<TextView
    android:id="@+id/tvTelpn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvAlamat"
    android:text="Telepon :"
    android:textSize="20dp"/>

<EditText
    android:id="@+id/edtTelpnPembeli"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/tvTelpn"
    android:layout_alignLeft="@+id/edtIdPembeli"
    android:layout_toRightOf="@+id/tvTelpn"
    android:hint="Telpn"
    android:inputType="phone"/>

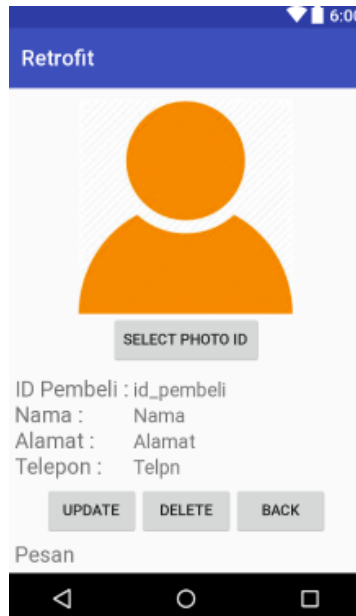
<LinearLayout
    android:id="@+id/LinearLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvTelpn"
    android:layout_centerHorizontal="true"
    android:orientation="horizontal"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp">

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

    <Button
        android:id="@+id/btDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Delete"/>

    <Button
        android:id="@+id/btBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Back"/>
</LinearLayout>
<TextView
    android:id="@+id/tvMessage"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Pesan"
    android:layout_alignParentBottom="true"
    android:layout_below="@+id/LinearLayout"
    android:textSize="20dp"
    android:inputType="textMultiLine"/>
</RelativeLayout>
</ScrollView>

```



Gambar 15. Preview PutDelPembeliActivity.xml

Setelah membuat layout seperti pada Gambar 15, selanjutnya definisikan fungsi dari setiap komponen pada layout tersebut. Source code final dari PutDelPembeliActivity.java dapat dilihat pada tabel dibawah ini.

PutDelPembeliActivity.java
<pre> package demo.retrofit; import android.content.Context; import android.content.Intent; import android.database.Cursor; import android.net.Uri; import android.os.Bundle; import android.provider.MediaStore; import android.support.v7.app.AppCompatActivity; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.ImageView; import android.widget.TextView; import android.widget.Toast; import com.bumptech.glide.Glide; import java.io.File; import demo.retrofit.Model.ResultPembeli; import demo.retrofit.Rest.ApiClient; import demo.retrofit.Rest.ApiInterface; import okhttp3.MediaType; import okhttp3.MultipartBody; import okhttp3.RequestBody; import retrofit2.Call; import retrofit2.Callback; import retrofit2.Response; public class PutDelPembeliActivity extends AppCompatActivity { ImageView mPhotoid; EditText edtIdPembeli, edtNama, edtAlamat, edtTelpn; TextView tvMessage; Context mContext; Button btUpdate, btDelete, btBack, btPhotoId; </pre>

```

String pathImage="";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_put_del_pembeli);
    mContext = getApplicationContext();

    mPhotoid = (ImageView) findViewById(R.id.imgPhotoId);
    edtIdPembeli = (EditText) findViewById(R.id.edtIdPembeli);
    edtNama = (EditText) findViewById(R.id.edtNamaPembeli);
    edtAlamat = (EditText) findViewById(R.id.edtAlamatPembeli);
    edtTelpn = (EditText) findViewById(R.id.edtTelpnPembeli);

    tvMessage = (TextView) findViewById(R.id.tvMessage);

    btUpdate = (Button) findViewById(R.id.btUpdate);
    btDelete = (Button) findViewById(R.id.btDelete);
    btBack = (Button) findViewById(R.id.btBack);
    btPhotoId = (Button) findViewById(R.id.btPhotoId);

    Intent mIntent = getIntent();

    edtIdPembeli.setText(mIntent.getStringExtra("id_pembeli"));
    edtNama.setText(mIntent.getStringExtra("nama"));
    edtAlamat.setText(mIntent.getStringExtra("alamat"));
    edtTelpn.setText(mIntent.getStringExtra("telpn"));

    // if (mIntent.getStringExtra("photo_id").length()>0)
    Picasso.with(mContext).load(ApiClient.BASE_URL +
    mIntent.getStringExtra("photo_id")).into(mPhotoid);
    // else Picasso.with(mContext).load(R.drawable.photoid).into(mPhotoid);
    if (mIntent.getStringExtra("photo_id").length()>0)
    Glide.with(mContext).load(ApiClient.BASE_URL +
    mIntent.getStringExtra("photo_id")).into(mPhotoid);
    else Glide.with(mContext).load(R.drawable.photoid).into(mPhotoid);

    pathImage = mIntent.getStringExtra("photo_id");
    setListener();
}

private void setListener() {
    final ApiInterface mApiInterface =
    ApiClient.getClient().create(ApiInterface.class);

    btUpdate.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            MultipartBody.Part body = null;
            //dicek apakah image sama dengan yang ada di server atau berubah
            //jika sama dikirim lagi jika berbeda akan dikirim ke server
            if
            ((pathImage.contains("upload/"+edtIdPembeli.getText().toString())==false)&&(pathImage.length()>0)){

                //File creating from selected URL
                File file = new File(pathImage);

                // create RequestBody instance from file
                RequestBody requestFile =
                RequestBody.create(MediaType.parse("multipart/form-data"), file);

                // MultipartBody.Part is used to send also the actual file name
                body = MultipartBody.Part.createFormData("photo_id", file.getName(),
                requestFile);
            }

            RequestBody reqIdPembeli =
            MultipartBody.create(MediaType.parse("multipart/form-data"),
            (edtIdPembeli.getText().toString().isEmpty()==true)?"":edtIdPembeli.getText().toString())

```



```

;

        RequestBody reqNama =
            MultipartBody.create(MediaType.parse("multipart/form-data"),
(edtNama.getText().toString().isEmpty()==true)?"":edtNama.getText().toString());

        RequestBody reqAlamat =
            MultipartBody.create(MediaType.parse("multipart/form-data"),
(edtAlamat.getText().toString().isEmpty()==true)?"":edtAlamat.getText().toString());

        RequestBody reqTelpn =
            MultipartBody.create(MediaType.parse("multipart/form-data"),
(edtTelpn.getText().toString().isEmpty()==true)?"":edtTelpn.getText().toString());

        RequestBody reqAction =
            MultipartBody.create(MediaType.parse("multipart/form-data"),
"put");

        Call<ResultPembeli> callUpdate = mApiInterface.putPembeli(body, reqIdPembeli,
reqNama, reqAlamat, reqTelpn,
reqAction);

        callUpdate.enqueue(new Callback<ResultPembeli>() {
            @Override
            public void onResponse(Call<ResultPembeli> call, Response<ResultPembeli>
response) {
                //Log.d("Update Retrofit ", response.body().getStatus());
                if (response.body().getStatus().equals("failed")){
                    tvMessage.setText("Retrofit Update \n Status =
"+response.body().getStatus()+"\n"+
                        "Message = "+response.body().getMessage()+"\n");
                }else{
                    String detail = "\n"+
                        "id_pembeli =
"+response.body().getResult().get(0).getIdPembeli()+"\n"+
                        "agama =
"+response.body().getResult().get(0).getNama()+"\n"+
                        "alamat =
"+response.body().getResult().get(0).getAlamat()+"\n"+
                        "telpn =
"+response.body().getResult().get(0).getTelpn()+"\n"+
                        "photo_id =
"+response.body().getResult().get(0).getPhotoId()+"\n";
                    tvMessage.setText("Retrofit Update \n Status =
"+response.body().getStatus()+"\n"+
                        "Message = "+response.body().getMessage()+detail);
                }
            }

            @Override
            public void onFailure(Call<ResultPembeli> call, Throwable t) {
                //Log.d("Update Retrofit ", t.getMessage());
                tvMessage.setText("Retrofit Update \n Status = "+ t.getMessage());
            }
        });

        btDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                RequestBody reqIdPembeli =
                    MultipartBody.create(MediaType.parse("multipart/form-data"),
(edtIdPembeli.getText().toString().isEmpty()==true)?"":edtIdPembeli.getText().toString())
;

                RequestBody reqAction =
                    MultipartBody.create(MediaType.parse("multipart/form-data"),

```

```

"delete");
        Call<ResultPembeli> callDelete =
mApiInterface.deletePembeli(reqIdPembeli, reqAction);
        callDelete.enqueue(new Callback<ResultPembeli>() {
            @Override
            public void onResponse(Call<ResultPembeli> call,
Response<ResultPembeli> response) {
                tvMessage.setText("Retrofit Delete \n Status =
"+response.body().getStatus()+"\n"+
                    "Message = "+response.body().getMessage()+"\n");
            }

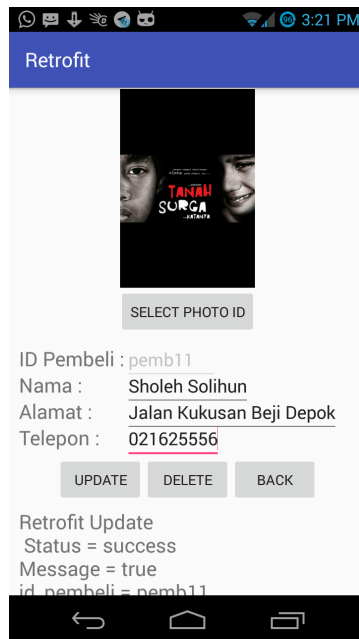
            @Override
            public void onFailure(Call<ResultPembeli> call, Throwable t) {
                tvMessage.setText("Retrofit Delete \n Status = "+
t.getMessage());
            }
        });
    }
});
btBack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent tempIntent = new Intent(mContext, GetPembeliActivity.class);
        startActivity(tempIntent);
    }
});

btPhotoId.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        final Intent galleryIntent = new Intent();
        galleryIntent.setType("image/*");
        galleryIntent.setAction(Intent.ACTION_PICK);
        Intent intentChoose = Intent.createChooser(galleryIntent, "Pilih Gambar
Untuk Di upload");
        startActivityForResult(intentChoose, 10);
    }
});

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK && requestCode == 10) {
        if (data == null) {
            Toast.makeText(mContext, "Gambar Gagal Di load",
Toast.LENGTH_LONG).show();
            return;
        }
        Uri selectedImage = data.getData();
        String[] filePathColumn = {MediaStore.Images.Media.DATA};
        Cursor cursor = getContentResolver().query(selectedImage, filePathColumn,
null, null, null);
        if (cursor != null) {
            cursor.moveToFirst();
            int columnIndex = cursor.getColumnIndex(filePathColumn[0]);
            pathImage = cursor.getString(columnIndex);

            //Picasso.with(mContext).load(new
File(imagePath)).fit().into(mImageView);
            Glide.with(mContext).load(new File(pathImage)).into(mPhotoid);
            cursor.close();
        } else {
            Toast.makeText(mContext, "Gambar Gagal Di load",
Toast.LENGTH_LONG).show();
        }
    }
}
}

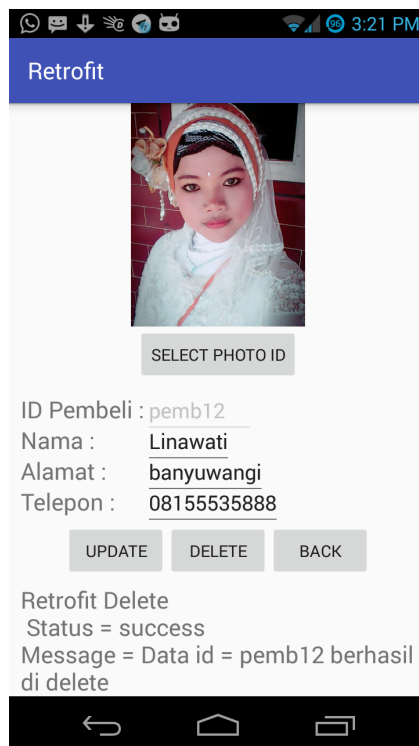
```



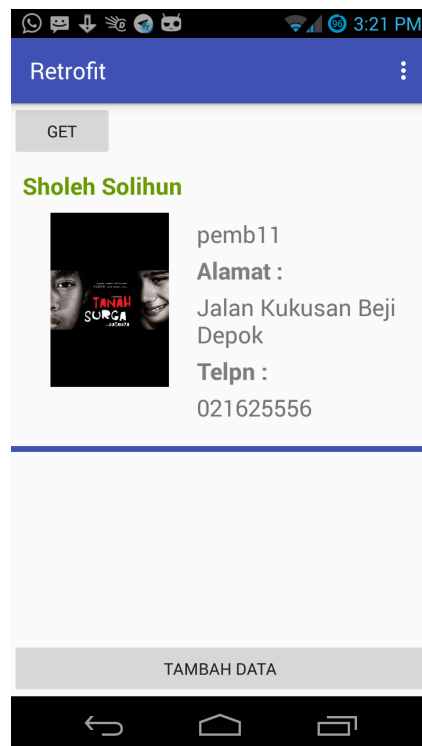
Gambar 16. Preview Transaksi Update



Gambar 17. Preview Transaksi Get Untuk Mengecek Apakah Data Sudah Terupdate



Gambar 18. Preview Transaksi Delete



Gambar 19. Preview Transaksi Get dari Server Untuk Mengecek Apakah Data Sudah Terdelete