

**PRAKTIKUM
PEMROGRAMAN
MOBILE / TEKNOLOGI
PEMROGRAMAN
MOBILE**

SEMESTER GENAP T.A 2020/2021

LAPORAN PROYEK AKHIR



DISUSUN OLEH:

NAMA	:	UMI ROHMAH NH.	(124190021)
		PANDU DHAULAGIRI	(124190044)
		ARYA ADHI PRADANA	(124190067)
KELAS/PLUG	:	A	
NAMA ASISTEN	:	ANGELICA AMARTYA PUTRI	
		DYAH AYU SUCI ILHAMI	

**PROGRAM STUDI INFORMATIKA/ SISTEM
INFORMASI**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
YOGYAKARTA**

2021

HALAMAN PENGESAHAN

LAPORAN PROYEK AKHIR PRAKTIKUM PEMROGRAMAN MOBILE/TEKNOLOGI PEMROGRAMAN MOBILE

Disusun Oleh:

UMI ROHMAH NH.
124190021

PANDU DHAULAGIRI
124190044

ARYA ADHI PRADANA
124190067

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Pemrograman Mobile

Pada Tanggal:

Asisten Praktikum

Asisten Praktikum

ANGELICA AMARTYA PUTRI
NIM. 124170011

DYAH AYU SUCI ILHAMI
NIM. 124170024

Mengetahui,
Ka. Lab. Digital

Mangaras Yanu Florestiyanto, S.T., M.Eng.
NIK. 2 8201 13 0425 1

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Pemrograman Mobile serta laporan proyek akhir praktikum yang berjudul Aplikasi Qoutes. Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten dosen yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, saya ucapkan terimakasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 17 Juni 2021

Penyusun

DAFTAR ISI

LEMBAR PENGESAHAN	ii
KATA PENGANTAR.....	iii
DAFTAR ISI	iv
BAB I	2
1.1 Rumusan Masalah.....	2
1.2 Tujuan	2
BAB II	3
2.1 Jadwal Pengerjaan & Pembagian Tugas.....	3
2.2 Dasar Teori	4
2.3 Source Code	5
2.4 Pembahasan.....	19
2.5 Output Program	20
BAB III	22
3.1 Kesimpulan	22
3.2 Saran	22
DAFTAR PUSTAKA.....	23

QUOTES OF THE DAY

BAB I

PENDAHULUAN

1.1 Rumusan Masalah

Rumusan masalah dari pengerjaan tugas akhir ini sebagai berikut:

- a. Bagaimana merancang aplikasi *quotes* berbasis android?
- b. Bagaimana mengimplementasikan rancangan menjadi aplikasi *quotes* berbasis android?

1.2 Tujuan

Tujuan yang ingin dicapai pada pengerjaan tugas akhir ini adalah:

- a. Merancang dan membuat desain aplikasi *quotes* berbasis android
- b. Membuat aplikasi sesuai dengan rancangan yang telah dibuat

BAB II

PEMBAHASAN

2.1 Jadwal Pengerjaan dan Pembagian Tugas

Jadwal pengerjaan

TANGGAL	KETERANGAN
2 JUNI 2021	Merancang desain aplikasi (ide aplikasi)
4 JUNI 2021	Membuat desain tampilan aplikasi
11 JUNI 2021	Menyatukan tampilan dalam satu activity dengan fragment
13 JUNI 2021	Implementasi API quotable.io
15 JUNI 2021	Implementasi Room Database
17 JUNI 2021	Pembuatan Laporan

Pembagian Tugas

- Umi Rohmah N.H
 - Membuat ide aplikasi
 - Mendaftar fitur-fitur yang diperlukan
 - Membuat desain tampilan aplikasi
 - Melakukan pengecekan aplikasi
 - Menyusun laporan
- Pandu Dhaulagiri
 - Membuat Class dan Interface yang diperlukan
 - Implementasi fitur-fitur kedalam aplikasi
 - Implementasi Database Aplikasi
 - Implementasi API dan Library
 - Menyusun laporan
- Arya Adhi Pradana
 - Membuat desain Front End dalam bentuk XML
 - Implementasi Front End dalam aplikasi
 - Implementasi Recycler View dengan memanfaatkan Adapter
 - Menghubungkan semua tampilan dalam satu activity main
 - Menyusun laporan

2.2 Dasar Teori

- a. Android Manifest Setiap aplikasi harus memiliki file `AndroidManifest.xml` (bernama persis seperti ini) di direktori akar. File manifest menyediakan informasi penting tentang aplikasi ke sistem Android, yang harus dimiliki sistem agar bisa menjalankan setiap kode aplikasi. Yang dilakukan file manifest diantaranya:
 - i. Menamai paket Java untuk aplikasi yang berfungsi sebagai identifier unik untuk aplikasi.
 - ii. Menjelaskan berbagai komponen aplikasi, yang menyertakan aktivitas, layanan, penerima siaran, dan penyedia materi yang membentuk aplikasi.
 - iii. Menentukan proses yang menjadi host komponen aplikasi.
 - iv. Mendeklarasikan izin aplikasi yang harus dimiliki aplikasi untuk mengakses bagian yang dilindungi pada API dan berinteraksi dengan aplikasi lain.
 - v. Mencantumkan daftar kelas Instrumentation yang memberikan profil dan informasi lain saat aplikasi berjalan.
 - vi. Mendeklarasikan level minimum Android API yang diperlukan aplikasi.
 - vii. Mencantumkan daftar pustaka yang harus ditautkan aplikasi.
- b. Gradle Android Studio menggunakan Gradle sebagai dasar sistem versi, dengan kemampuan khusus Android yang disediakan oleh Plugin Android untuk Gradle. Sistem ini bisa dijalankan sebagai alat terpadu dari menu Android Studio dan secara independen dari baris perintah.
- c. Parent View Merupakan sebuah view yang didalamnya dapat diisi oleh child view ataupun oleh parent view lainnya, beberapa bentuk dari parent view:
 - i. Linear Layout `LinearLayout` adalah sekelompok view yang menyejajarkan semua anak atau child view dalam satu arah, secara vertikal atau horizontal. Kita bisa menetapkan arah layout dengan atribut `android:orientation`.
 - ii. Relative Layout Adalah layout yang menampilkan komponen dalam posisi relatif. Posisi masing-masing Layout dapat ditentukan sebagai relatif terhadap elemen saudara atau pada posisi relatif terhadap area `RelativeLayout` induk. Jadi, dengan menggunakan `RelativeLayout`, kita dapat menyusun suatu komponen secara relatif terhadap komponen lainnya.
- d. Child View Merupakan sebuah view yang letaknya harus berada didalam sebuah parent view, terdapat banyak child view, diantaranya:
 - i. `TextView` : sebuah view untuk menampilkan tulisan.
 - ii. `ImageView` : sebuah view untuk menampilkan gambar.
 - iii. `Button` : Sebuah view untuk melakukan aksi/proses.
 - iv. `EditText` : sebuah form untuk input tulisan.
- e. Intent Intent adalah sebuah kelas dalam programming Android yang berfungsi untuk perpindahan halaman. Intent juga merupakan suatu objek yang terdapat dalam suatu activity dimana objek tersebut dapat komunikasi dengan activity yang lain.

- f. RecyclerView Adalah sebuah library yang digunakan sebagai tempat untuk menampilkan banyak data, mirip seperti ListView, namun RecyclerView lebih banyak memiliki keunggulan dibandingkan dengan ListView, sederhananya adalah RecyclerView digunakan apabila programmer ingin menampilkan 3 sebuah data dengan tampilan yang sama tetapi memiliki konten yang berbeda secara berulang-ulang.
- g. Room Persistence Library Library ini merupakan bagian dari Android Jetpack. Jetpack adalah kumpulan komponen software Android yang memudahkan kita dalam mengembangkan aplikasi android yang hebat.
- h. Application Programming Interface API atau kepanjangan dari Application Programming Interface ialah sekumpulan instruksi program dan protokol yang digunakan untuk membangun aplikasi perangkat lunak. API berperan sebagai pembawa pesan yang menerima permintaan pengguna dan memberitahu sistem apa yang harus dilakukan, lalu memberikan respons yang sesuai untuk permintaan tersebut.
- i. Volley Library: Library ini merupakan networking library yang berfungsi untuk:
 - i. Mengatur masalah *scheduling* dalam network request
 - ii. Mempermudah dalam menghubungkan android dengan api
 - iii. Mengelola permintaan data secara asinkronis

2.3 Source Code

File Java

```
package com.example.tugasakhir;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.os.Bundle;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import
com.google.android.material.bottomnavigation.BottomNavigationView;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity implements
BottomNavigationView.OnNavigationItemSelectedListener{
    private BottomNavigationView navbar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        navbar = findViewById(R.id.main_navbar);
        navbar.setOnNavigationItemSelectedListener(this);
        loadFragment(new Fragment());
        loadFragment(new Home());
    }
    private boolean loadFragment(Fragment fragment) {
        if(fragment != null){
```

```
        getSupportFragmentManager().beginTransaction()
            .replace(R.id.main_frame, fragment)
            .commit();
        return true;
    }return false;
}
@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item) {
    Fragment fragment = new Home();
    switch (item.getItemId()){
        case R.id.ic_home:
            fragment=new Home();
            break;
        case R.id.ic_heart:
            fragment=new Heart();
            break;
    }return loadFragment(fragment);
}
}
```

Source Code 2.3.1 MainActivity.java

```

package com.example.tugasakhir;
import android.app.Dialog;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class AdapterHeart extends
RecyclerView.Adapter<AdapterHeart.ViewHolder> {
    private List<MainData> dataList;
    private Context context;
    private DBKomen database;
    public AdapterHeart(Context context, List<MainData> dataList){
        this.context = context;
        this.dataList = dataList;
        notifyDataSetChanged();
    }
    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        View view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.rv_row,parent,false);
        return new ViewHolder(view);
    }
    public void onBindViewHolder(@NonNull ViewHolder holder, int
position) {
        MainData data = dataList.get(position);
        database = DBKomen.getInstance(context);
        holder.textView.setText(data.getKomentar());
        holder.btEdit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                MainData d=dataList.get(holder.getAdapterPosition());
                int sID= d.getID();
                String sText = d.getKomentar();
                Dialog dialog = new Dialog(context);
                dialog setContentView(R.layout.dialog_update);
                int width = WindowManager.LayoutParams.MATCH_PARENT;
                int height = WindowManager.LayoutParams.WRAP_CONTENT;
                dialog.getWindow().setLayout(width,height);
                dialog.show();
                EditText editText =
dialog.findViewById(R.id.edit_text);
                Button btUpdate =
dialog.findViewById(R.id.bt_update);
                editText.setText(sText);
                btUpdate.setOnClickListener(new
View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        dialog.dismiss();
                        String uText =

```

```

        editText.getText().toString().trim();
        database.mainDao().update(sID, uText);
        dataList.clear();
        dataList.addAll(database.mainDao().getAll());
        notifyDataSetChanged();
    }
    });
}
});

holder.btnDelete.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v){
        MainData d=dataList.get(holder.getAdapterPosition());
        database.mainDao().delete(d);
        int position = holder.getAdapterPosition();
        dataList.remove(position);
        notifyItemRemoved(position);
        notifyItemRangeChanged(position, dataList.size());
    }
});
}
@Override
public int getItemCount() {
    return dataList.size();
}
}
public class ViewHolder extends RecyclerView.ViewHolder {
    TextView textView;
    ImageView btEdit, btDelete;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        textView = itemView.findViewById(R.id.text_view);
        btEdit = itemView.findViewById(R.id.bt_edit);
        btDelete = itemView.findViewById(R.id.bt_delete);
    }
}
}
}

```

Source Code 2.3.2 AdapterHeart.java

```

package com.example.tugasakhir;
import android.content.Context;
import androidx.room.Database;
import androidx.room.Room;
import androidx.room.RoomDatabase;

@Database(entities = {MainData.class}, version = 1, exportSchema =
false)
public abstract class DBKomen extends RoomDatabase {
    private static DBKomen database;
    private static String DATABASE_NAME = "Komen Database";
    public synchronized static DBKomen getInstance(Context context){
        if(database==null){
            database =
Room.databaseBuilder(context.getApplicationContext()
, DBKomen.class, DATABASE_NAME)
        .allowMainThreadQueries()
        .fallbackToDestructiveMigration()
        .build();
        }
        return database;
    }
}

```

```

    }
    public abstract MainDao mainDao();
}

```

Source Code 2.3.3 DBKomen.java

```

package com.example.tugasakhir;
import android.content.Context;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import java.util.ArrayList;
import java.util.List;

public class Heart extends Fragment {
    EditText editText;
    Button btAdd;
    RecyclerView recyclerView;
    List<MainData> dataList = new ArrayList<>();
    LinearLayoutManager layoutManager;
    DBKomen database;
    Context context;
    AdapterHeart adapter;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        View view = inflater.inflate(R.layout.fragment_list,
container, false);
        editText = view.findViewById(R.id.edit_text);
        btAdd = view.findViewById(R.id.bt_add);
        recyclerView = view.findViewById(R.id.recyclerviewid);
        database = DBKomen.getInstance(this.getContext());
        dataList = database.mainDao().getAll();
        layoutManager = new LinearLayoutManager(context);
        adapter = new AdapterHeart(getContext(), dataList);
        recyclerView.setAdapter(adapter);
        recyclerView.setLayoutManager((LinearLayoutManager));
        btAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String sText = editText.getText().toString().trim();
                if(!sText.equals("")){
                    MainData data = new MainData();
                    data.setKomentar(sText);
                    database.mainDao().insert(data);
                    editText.setText("");
                    dataList.clear();
                    dataList.addAll(database.mainDao().getAll());
                    adapter.notifyDataSetChanged();
                }
            }
        });
        return view;
    }
}

```

```

    }
}

```

Source Code 2.3.4 Heart.java

```

package com.example.tugasakhir;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;
import java.util.ArrayList;
import java.util.List;

public class Home extends Fragment {
    EditText editText;
    TextView tvQuote, tvAuth;
    ImageButton btSave, btReset;
    List<MainData> dataList = new ArrayList<>();
    DBKomen database;
    AdapterHeart adapter;
    QuoteDataService quoteDataService;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_home,
container, false);
        tvQuote = view.findViewById(R.id.tv_randQuote);
        tvAuth = view.findViewById(R.id.tv_authQuote);
        btSave = view.findViewById(R.id.btnSave);
        btReset = view.findViewById(R.id.btnReset);
        editText = view.findViewById(R.id.et_komQuote);
        database = DBKomen.getInstance(this.getContext());
        adapter = new AdapterHeart(getContext(), dataList);
        quoteDataService = new QuoteDataService(getContext());
        getQuote();
        btReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                getQuote();
            }
        });
        btSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String sText = editText.getText().toString().trim();
                if(!sText.equals("")){
                    MainData data = new MainData();
                    data.setKomentar(sText);
                    database.mainDao().insert(data);
                    editText.setText("");
                    dataList.clear();
                    dataList.addAll(database.mainDao().getAll());
                    adapter.notifyDataSetChanged();
                }
            }
        });
    }
}

```

```

        });
        return view;
    }
    public void getQuote() {
        quoteDataService.getQuoteContent(new
        QuoteDataService.VolleyResponseListener() {
            @Override
            public void onError(String message) {
                Toast.makeText(getActivity(), "Network Error",
                Toast.LENGTH_SHORT).show();
            }
            @Override
            public void onResponse(String konten) {
                tvQuote.setText(konten);
            }
        });
        quoteDataService.getQuoteAuthor(new
        QuoteDataService.VolleyResponseListener() {
            @Override
            public void onError(String message) {
                Toast.makeText(getActivity(), "Network Error",
                Toast.LENGTH_SHORT).show();
            }
            @Override
            public void onResponse(String konten) {
                tvAuth.setText(konten);
            }
        });
    }
}

```

Source Code 2.3.5 Home.java

```

package com.example.tugasakhir;
import androidx.room.Dao;
import androidx.room.Delete;
import androidx.room.Insert;
import androidx.room.Query;
import java.util.List;
import static androidx.room.OnConflictStrategy.REPLACE;

@Dao
public interface MainDao {
    @Insert(onConflict = REPLACE)
    void insert(MainData mainData);
    @Delete
    void delete(MainData mainData);
    @Query("UPDATE datalokal SET komentar = :sText WHERE ID = :sID")
    void update(int sID, String sText);
    @Query("SELECT * FROM datalokal")
    List<MainData> getAll();
}

```

Source Code 2.3.6 MainDAO.java

```

package com.example.tugasakhir;
import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;

```

```

@Entity(tableName = "DataLokal")
public class MainData {
    @PrimaryKey(autoGenerate=true) private int ID;
    @ColumnInfo (name = "komentar") private String komentar;
    public int getID() {
        return ID;
    }
    public void setID(int ID) {
        this.ID = ID;
    }
    public String getKomentar() {
        return komentar;
    }
    public void setKomentar(String komentar) {
        this.komentar = komentar;
    }
}

```

Source Code 2.3.7 MainData.java

```

package com.example.tugasakhir;
import android.content.Context;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import org.json.JSONException;
import org.json.JSONObject;

public class QuoteDataService {
    Context context;
    public QuoteDataService(Context ctx) {
        this.context = ctx;
    }
    public interface VolleyResponseListener {
        void onError(String message);
        void onResponse(String konten);
    }
    public void getQuoteContent(final VolleyResponseListener
    volleyResponseListener){
        String url ="https://api.quotable.io/random/";
        JsonObjectRequest request = new
        JsonObjectRequest(Request.Method.GET, url, null,
            new Response.Listener<JSONObject>() {
                String konten;
                @Override
                public void onResponse(JSONObject response) {
                    try {
                        JSONObject quoteContent = response;
                        konten =
                        quoteContent.getString("content");
                    } catch (JSONException e) {
                        e.printStackTrace();
                        Toast.makeText(context, e.toString(),
                        Toast.LENGTH_SHORT).show();
                    }
                    volleyResponseListener.onResponse(konten);
                }
            }, error -> {

```



```

        Toast.makeText(context, "Network Error",
Toast.LENGTH_SHORT).show();
        volleyResponseListener.onError("Network Error");
    });
    Simpleton.getInstance(context).addToRequestQueue(request);
}
public void getQuoteAuthor(VolleyResponseListener
volleyResponseListener){
    String url ="https://api.quotable.io/random/";
    JsonObjectRequest request = new
JsonObjectRequest(Request.Method.GET, url, null,
        new Response.Listener<JSONObject>() {
            String author;
            @Override
            public void onResponse(JSONObject response) {
                try {
                    JSONObject quoteContent = response;
                    author =
quoteContent.getString("author");
                } catch (JSONException e) {
                    e.printStackTrace();
                    Toast.makeText(context, e.toString(),
Toast.LENGTH_SHORT).show();
                }
                volleyResponseListener.onResponse(author);
            }
        }, error -> {
            Toast.makeText(context, "Network Error",
Toast.LENGTH_SHORT).show();
            volleyResponseListener.onError("Network Error");
        });
    Simpleton.getInstance(context).addToRequestQueue(request);
}
}

```

Source Code 2.3.8 QuoteDataService.java

```

package com.example.tugasakhir;
import android.content.Context;
import android.graphics.Bitmap;
import android.util.LruCache;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.ImageLoader;
import com.android.volley.toolbox.Volley;

public class Simpleton {
    private static Simpleton instance;
    private RequestQueue requestQueue;
    private static Context ctx;
    private Simpleton(Context context) {
        ctx = context;
        requestQueue = getRequestQueue();
    }
    public static synchronized Simpleton getInstance(Context context)
    {
        if (instance == null) {
            instance = new Simpleton(context);
        }return instance;
    }
    public RequestQueue getRequestQueue() {

```

```

        if (requestQueue == null) {
            requestQueue =
Volley.newRequestQueue(ctx.getApplicationContext());
        }return requestQueue;
    }
    public <T> void addToRequestQueue(Request<T> req) {
        getRequestQueue().add(req);
    }
}

```

Source Code 2.3.9 Simpleton.java

File xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="0dp"
    tools:context=".MainActivity">
    <FrameLayout
        android:id="@+id/main_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1" />

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/main_navbar"
        android:layout_width="match_parent"
        android:layout_height="67dp"
        android:layout_gravity="bottom"
        android:layout_weight="0"
        android:background="@drawable/foto10"
        app:menu="@menu/item_menu" />
</LinearLayout>

```

Source Code 2.3.10 activity_main.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="match_parent"
    android:orientation="vertical"
    android:layout_margin="8dp">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/edit_text"
        android:hint="Enter Text"
        android:padding="12dp"
        android:lines="4"
        android:gravity="top"
        android:background="@android:drawable/editbox_background"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/bt_update"
        android:text="Update"
    >

```

```

        android:layout_marginTop="8dp"/>
    </LinearLayout>

```

Source Code 2.3.11 dialog_update.xml

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Home">
    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/linearLayout"
        android:background="@drawable/foto7"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <androidx.constraintlayout.widget.Guideline
            android:id="@+id/guideline_top"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            app:layout_constraintGuide_percent="0.15" />
        <androidx.constraintlayout.widget.Guideline
            android:id="@+id/guideline_mid"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            app:layout_constraintGuide_percent="0.7" />
        <androidx.constraintlayout.widget.Guideline
            android:id="@+id/guideline_bot"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            app:layout_constraintGuide_percent="0.85" />

        <androidx.constraintlayout.widget.Guideline
            android:id="@+id/guideline_start"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            app:layout_constraintGuide_percent="0.75" />
        <TextView
            android:id="@+id/tv_home"
            android:layout_width="0dp"
            android:layout_height="0dp"
            android:layout_marginStart="16dp"
            android:layout_marginTop="16dp"
            android:layout_marginEnd="16dp"
            android:layout_marginBottom="16dp"
            android:fontFamily="@font/dancing_script"
            android:text="Quotes of The Day"
            android:textSize="24sp"
            app:layout_constraintBottom_toTopOf="@+id/guideline_top"
            app:layout_constraintEnd_toStartOf="@+id/guideline_start"
            app:layout_constraintHorizontal_bias="0.063"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.175" />
        <ImageButton

```

```

        android:id="@+id/btnReset"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:src="@drawable/ic_refresh"
        app:layout_constraintBottom_toTopOf="@+id/guideline_top"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/guideline_start"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/tv_randQuote"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:fontFamily="@font/dancing_script"
        android:text="Loading"
        android:textSize="20sp"
        app:layout_constraintBottom_toTopOf="@+id/guideline_mid"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline_top" />
    <TextView
        android:id="@+id/tv_authQuote"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="16dp"
        android:fontFamily="@font/dancing_script"
        android:includeFontPadding="false"
        android:text="Loading"
        android:textSize="22sp"
        app:layout_constraintBottom_toTopOf="@+id/guideline_bot"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline_mid" />
    <EditText
        android:id="@+id/et_komQuote"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/guideline_start"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline_bot" />
    <ImageButton
        android:id="@+id/btnSave"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"

```

```

        android:layout_marginBottom="16dp"
        android:src="@drawable/ic_save"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/guideline_start"
        app:layout_constraintTop_toTopOf="@+id/guideline_bot" />
    </androidx.constraintlayout.widget.ConstraintLayout>
</FrameLayout>

```

Source Code 2.3.12 fragment_home.xml

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Heart">
    <LinearLayout
        android:background="@drawable/foto6"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <EditText
            android:id="@+id/edit_text"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@android:drawable/editbox_background"
            android:hint="Enter Text"
            android:padding="12dp" />
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:orientation="horizontal">
            <Button
                android:id="@+id/bt_add"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:text="Add" />
        </LinearLayout>
        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/recyclerviewid"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
    </LinearLayout>
</FrameLayout>

```

Source Code 2.3.13 fragment_list.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="4dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```
        android:orientation="horizontal"
        android:padding="16dp">
        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:id="@+id/text_view"
            android:textSize="14sp"/>
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/bt_edit"
            android:layout_marginStart="8dp"
            android:src="@drawable/ic_edit"/>
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/bt_delete"
            android:layout_marginStart="8dp"
            android:src="@drawable/ic_delete"/>
    </LinearLayout>
</androidx.cardview.widget.CardView>
```

Source Code 2.3.14 rv_row.xml

2.4 Pembahasan

Projek ini bertemakan *Quotes* dan memanfaatkan API dari *quotable.io* untuk mendapatkan *quotes* nya. Aplikasi ini memiliki dua halaman utama yaitu *home* dan *notes*. Pada halaman *home*, pengguna akan mendapati sebuah *quotes* acak dan kolom pengisian dibawahnya. Segala hal yang diisi pengguna di kolom pengisian itu akan disimpan dalam list yang dapat dilihat di halaman berikutnya. Pada halaman *notes*, pengguna akan dapat melihat catatan-catatan yang telah mereka buat setelah membaca *quotes*. Fungsi dari *notes* ini adalah untuk mendaftarkan segala hal yang pengguna pelajari atau dapat setelah mendapatkan sebuah *quotes* acak. Pengguna juga bebas mengedit, menambah, dan menghapus *quotes* yang berada di dalam daftar yang ditampilkan dalam halaman kedua.

Aplikasi ini memanfaatkan *quote API* yang didapatkan dari *quotable.io*. Proses penanganan *network* dan data *service* di kelola dengan *library Volley*, yang merupakan bagian dari *Android Jetpack*. Aplikasi ini juga memanfaatkan *Room Database Library* yang mempermudah proses penanganan database lokal dalam android. Kedua *library* ini menciptakan fungsi-fungsi yang sifatnya modular, sehingga tidak '*hard coded*' dan dapat di implementasikan dalam aplikasi yang berbeda-beda dengan API yang berbeda-beda, dengan penyesuaian minim.

Dalam sistem Android nya sendiri, kami memanfaatkan satu buah *activity* yang dibagi menjadi dua *fragment*. Kemudian kelas-kelas lain yang mengelola data-data dari dan ke API serta Database. *Activity* tersebut adalah *MainActivity.java* dengan 2 *fragment* yang diatur oleh *Home.java* serta *Heart.java*. *Fragment home* diatur oleh *Home.java*, dikhususkan untuk menampilkan data dari API. Sedangkan *Fragment notes* diatur oleh *Heart.java*, dikhususkan untuk menampilkan data dari Database.

Tentunya, kedua *fragment* tersebut terhubung satu sama lain. Sebagai contoh, dalam *Fragment Home* terdapat kolom *edit text* yang terhubung ke database yang dapat di tampilkan pada *Fragment Heart*. Cara kerja tiap *fragment class* itu sederhana, mereka hanya memanggil fungsi dari class yang mengelola database dan API untuk menerima atau mengirimkan input.

1. Database

Pada dasarnya, database di kelola oleh 2 buah class dan 1 interface, yakni *MainDao* (interface), *MainData*, dan *DBKomen*. *DBKomen* adalah tempat instansi database dibuat, *MainData* adalah tempat Tabel database dibuat dan dikelola. *MainDao* adalah tempat fungsi-fungsi yang akan digunakan untuk mengurus Query dalam database.

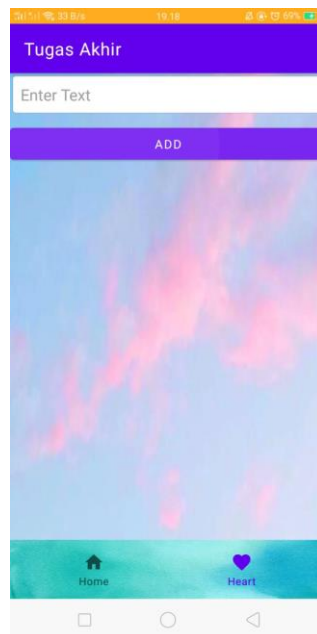
2. API

Pada dasarnya, API di kelola oleh 2 buah class, yakni *QuoteDataService* dan *Simpleton*. *QuoteDataService* adalah class yang melakukan request kepada API serta berisi fungsi-fungsi untuk mengambil data dari API. *Simpleton* adalah class yang mengelola lalu lintas data dari jaringan, agar proses pengambilan data dapat berjalan asinkron, tidak bertabrakan, dan berjalan dengan baik.

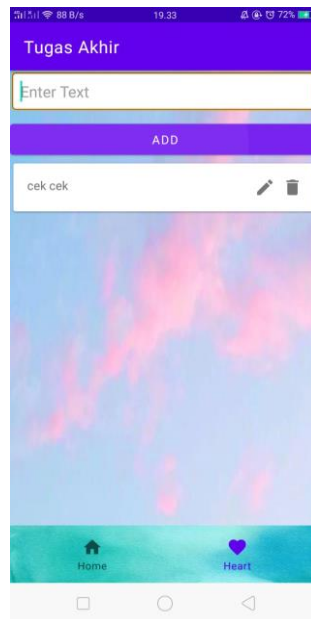
2.5 Output Program



2.5.1 fragment_home.xml



2.5.2 fragment_list.xml



2.5.3 fragment_list.xml

BAB III

PENUTUP

3.1 Kesimpulan

Aplikasi ini di buat untuk memotivasi pengguna agar lebih bersemangat dalam menjalankan aktivitas sehari-harinya. Pengguna juga dapat membuat sebuah catatan / *quotes* dengan lebih praktis yang dapat dibawa kemanapun dan kapanpun, dengan tampilan yang lebih inovatif.

3.2 Saran

Bagi para pembaca diharapkan dapat memberi masukan dan saran bagi penulis agar dapat membuat laporan ini menjadi lebih baik lagi dan sebagaimana mestinya.

DAFTAR PUSTAKA

Modul Praktikum Pemrograman Mobile

<https://developer.android.com>

<https://github.com/lukePeavey/quotable>