

## **LAPORAN PRATIKUM MOBILE 3**



**Disusun oleh**  
**Muhammad Zaki Al Hafiz**  
**(2211533004)**

**Dosen Pengampu**  
**Nufriah, S.ST, M.Kom**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**  
**SEPTEMBER 2024**

## 1. Kelas ModelPelanggan

```
package model;

public class ModelPelanggan {

    String id, nama, email, hp;

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getNama() {
        return nama;
    }

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

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getHp() {
        return hp;
    }

    public void setHp(String hp) {
        this.hp = hp;
    }
}
```

Kelas dalam bahasa Java bernama ModelPelanggan, yang termasuk dalam package model. Kelas ini digunakan untuk merepresentasikan data pelanggan dengan empat atribut: id, nama, email, dan hp (nomor handphone).

### 1.1 Deklarasi atribut

```
String id, nama, email, hp;
```

- id: menyimpan identifikasi unik pelanggan.
- nama: menyimpan nama pelanggan.
- email: menyimpan alamat email pelanggan.
- hp: menyimpan nomor handphone pelanggan.

### 1.2 Getter dan Setter

```
public String getId() {  
    return id;  
}  
  
public void setId(String id) {  
    this.id = id;  
}  
  
public String getName() {  
    return nama;  
}  
  
public void setName(String nama) {  
    this.nama = nama;  
}  
  
public String getEmail() {  
    return email;  
}  
  
public void setEmail(String email) {  
    this.email = email;  
}  
  
public String getHp() {  
    return hp;  
}  
  
public void setHp(String hp) { this.hp = hp; }
```

- getId() digunakan untuk mengambil nilai id pelanggan.
- setId(String id) digunakan untuk menetapkan nilai id pelanggan
- getName() digunakan untuk mengambil nama pelanggan.
- setName(String nama) digunakan untuk menetapkan nama pelanggan.

- getEmail() digunakan untuk mengambil alamat email pelanggan.
- setEmail(String email) digunakan untuk menetapkan alamat email pelanggan.
- getHp() digunakan untuk mengambil nomor handphone pelanggan.
- setHp(String hp) digunakan untuk menetapkan nomor handphone pelanggan.

## 2. Kelas PelangganAddActivity

```
package pelanggan;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.example.myapplication.R;
import com.example.myapplication.tambahanPelanggan;

import java.util.UUID;

import database.SQLiteHelper;
import model.ModelPelanggan;

public class PelangganAddActivity extends
AppCompatActivity {
    EditText nama, email, hp;
    Button btnSimpan, btnBatal;
    @Override
```

```

        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            EdgeToEdge.enable(this);
            setContentView(R.layout.activity_pelanggan_add);

            ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
                Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
                v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
                return insets;
            });

            nama = (EditText)
findViewById(R.id.edPelAddNama);
            email = (EditText)
findViewById(R.id.edPelAddEmail);
            hp = (EditText) findViewById(R.id.edPelAddHp);
            btnSimpan = (Button)
findViewById(R.id.btnPelAddSimpan);
            btnBatal = (Button)
findViewById(R.id.btnPelAddBatal);

            SQLiteHelper db;

            db = new SQLiteHelper(PelangganAddActivity.this);

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

```

```

        ModelPelanggan mp = new ModelPelanggan();
        String uniqueID =
UUID.randomUUID().toString();
        mp.setId(""+uniqueID);
        mp.setNama(nama.getText().toString());
        mp.setEmail(email.getText().toString());
        mp.setHp(hp.getText().toString());

        Toast.makeText(PelangganAddActivity.this,
""+mp.getId()+mp.getNama()+mp.getEmail()+mp.getHp(),
Toast.LENGTH_SHORT).show();
        boolean cek = db.insertPelanggan(mp);
        if (cek == true) {

Toast.makeText(PelangganAddActivity.this, "Data
Disimpan", Toast.LENGTH_SHORT).show();
            startActivity(new
Intent(PelangganAddActivity.this,
tambahanPelanggan.class));
            finish();
        }else {

Toast.makeText(PelangganAddActivity.this, "Gagal
Disimpan", Toast.LENGTH_SHORT).show();
        }
    }
});

    btnBatal.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            finish();
        }
    }

```

```

        });

    }

}

```

- `EditText nama, email, hp`: Ini adalah referensi untuk tiga inputan teks yang digunakan untuk mengambil data pelanggan dari pengguna, seperti nama, email, dan nomor handphone.
- `Button btnSimpan, btnBatal`: Dua tombol yang masing-masing digunakan untuk menyimpan data pelanggan (`btnSimpan`) dan membatalkan operasi (`btnBatal`).
- `SQLiteHelper db`: Referensi ke kelas `SQLiteHelper`, yang digunakan untuk berinteraksi dengan database `SQLite`, di mana data pelanggan akan disimpan.
- `EdgeToEdge.enable(this)`: Ini adalah metode untuk mengaktifkan tampilan `edge-to-edge`, memungkinkan tampilan UI untuk mengisi layar sepenuhnya di Android.
- `ViewCompat.setOnApplyWindowInsetsListener`: Digunakan untuk mengatur padding dari tampilan agar sesuai dengan inset (misalnya, sistem bar) pada perangkat.
- `nama, email, dan hp` dihubungkan dengan input teks yang sesuai dari layout XML (`R.layout.activity_pelanggan_add`).
- `btnSimpan` dan `btnBatal` dihubungkan dengan tombol dalam layout yang sesuai.

## 2.1 Logika btnSimpan

```

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

        ModelPelanggan mp = new ModelPelanggan();
    }
}

```

```

        String uniqueID =
UUID.randomUUID().toString();
        mp.setId(""+uniqueID);
        mp.setNama(nama.getText().toString());
        mp.setEmail(email.getText().toString());
        mp.setHp(hp.getText().toString());

        Toast.makeText(PelangganAddActivity.this,
""+mp.getId()+mp.getNama()+mp.getEmail()+mp.getHp(),
Toast.LENGTH_SHORT).show();
        boolean cek = db.insertPelanggan(mp);
        if (cek == true) {

Toast.makeText(PelangganAddActivity.this, "Data
Disimpan", Toast.LENGTH_SHORT).show();
            startActivity(new
Intent(PelangganAddActivity.this,
tambahanPelanggan.class));
            finish();
        }else {

Toast.makeText(PelangganAddActivity.this, "Gagal
Disimpan", Toast.LENGTH_SHORT).show();
        }
    }
});

```

- Sebuah objek ModelPelanggan (mp) dibuat untuk menyimpan data pelanggan yang dimasukkan oleh pengguna.
- `UUID.randomUUID().toString()` digunakan untuk menghasilkan ID pelanggan yang unik.
- Data dari input teks (nama, email, hp) diambil dan dimasukkan ke dalam objek ModelPelanggan.
- Setelah data diisi, fungsi `db.insertPelanggan(mp)` dipanggil



untuk menyimpan data ke database. Jika berhasil, sebuah Toast (notifikasi) ditampilkan dengan pesan "Data Disimpan" dan aplikasi berpindah ke Activity bernama tambahanPelanggan. Jika gagal, ditampilkan pesan "Gagal Disimpan".

## 2.2 Logika btnBatal

```
btnBatal.setOnClickListener(new  
View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        finish();  
    }  
});
```

- Tombol btnBatal hanya menutup Activity ini tanpa melakukan operasi lain. Metode finish() digunakan untuk mengakhiri Activity saat ini.

## 3. Kelas SQLiteHelper

```
package database;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import model.ModelPelanggan;  
  
public class SQLiteHelper extends SQLiteOpenHelper {  
    public static String DATABASE_NAME = "my_laundry.db";  
    public static final int DATABASE_VERSION = 1;  
    public static final String TABLE_PELANGGAN =  
"pelanggan";
```

```

        public static final String KEY_PELANGGAN_ID =
"pelanggan_id";
        public static final String KEY_PELANGGAN_NAMA =
"nama";
        public static final String KEY_PELANGGAN_EMAIL =
"email";
        public static final String KEY_PELANGGAN_HP = "hp";
        private static final String CREATE_TABLE_PELANGGAN =
"CREATE TABLE " + TABLE_PELANGGAN + "("
            + KEY_PELANGGAN_ID + " TEXT, " +
KEY_PELANGGAN_NAMA + " TEXT, "+ KEY_PELANGGAN_EMAIL + "
TEXT, "+KEY_PELANGGAN_HP +" TEXT )";

        public SQLiteHelper(Context context){
            super(context, DATABASE_NAME, null,
DATABASE_VERSION);
        }
        @Override
        public void onCreate(SQLiteDatabase db) {
            db.execSQL(CREATE_TABLE_PELANGGAN);
        }
        @Override
        public void onUpgrade(SQLiteDatabase db, int
oldVersion, int newVersion)
        {
            db.execSQL("DROP TABLE IF EXISTS " +
TABLE_PELANGGAN);
            onCreate(db);
        }
        public boolean insertPelanggan(ModelPelanggan mp){
            SQLiteDatabase database =
this.getReadableDatabase();
            ContentValues contentValues = new
ContentValues();

```

```

        contentValues.put(KEY_PELANGGAN_ID, mp.getId());
        contentValues.put(KEY_PELANGGAN_NAMA,
mp.getNama());
        contentValues.put(KEY_PELANGGAN_EMAIL,
mp.getEmail());
        contentValues.put(KEY_PELANGGAN_HP, mp.getHp());
        long id = database.insert(TABLE_PELANGGAN, null,
contentValues);
        database.close();
        if (id != -1){
            return true;
        }else{
            return false;
        }
    }

    public List<ModelPelanggan> getPelanggan() {
        List<ModelPelanggan> pel = new
ArrayList<ModelPelanggan>();
        String query = "SELECT * FROM " +
TABLE_PELANGGAN;
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(query, null);
        if (cursor.moveToFirst()){
            do{
                ModelPelanggan k = new ModelPelanggan();
                k.setId(cursor.getString(0));
                k.setNama(cursor.getString(1));
                k.setEmail(cursor.getString(2));
                k.setHp(cursor.getString(3));
                pel.add(k);
            }while (cursor.moveToNext());
        }
        return pel;
    }

```

```
}  
}
```

- DATABASE\_NAME: Nama file database (my\_laundry.db).
- TABLE\_PELANGGAN: Nama tabel yang digunakan untuk menyimpan data pelanggan (pelanggan).
- KEY\_PELANGGAN\_ID, KEY\_PELANGGAN\_NAMA, KEY\_PELANGGAN\_EMAIL, KEY\_PELANGGAN\_HP: Masing-masing merupakan nama kolom dalam tabel pelanggan untuk menyimpan ID, nama, email, dan nomor handphone pelanggan.

### 3.1 CREATE TABLE Statement

```
private static final String CREATE_TABLE_PELANGGAN =  
"CREATE TABLE " + TABLE_PELANGGAN + "("  
    + KEY_PELANGGAN_ID + " TEXT, " +  
    KEY_PELANGGAN_NAMA + " TEXT, " + KEY_PELANGGAN_EMAIL +  
    " TEXT, " + KEY_PELANGGAN_HP + " TEXT )";
```

Perintah SQL ini membuat tabel pelanggan dengan empat kolom:

pelanggan\_id, nama, email, dan hp, semuanya bertipe TEXT

### 3.2 Konstruktor

```
public SQLiteHelper(Context context){  
    super(context, DATABASE_NAME, null,  
    DATABASE_VERSION);  
}
```

Konstruktor ini memanggil konstruktor dari kelas SQLiteOpenHelper, yang menginisialisasi database SQLite dengan nama dan versi yang ditentukan.

### 3.3 Metode onCreate

```
public void onCreate(SQLiteDatabase db) {  
    db.execSQL(CREATE_TABLE_PELANGGAN);  
}
```

Metode ini dipanggil ketika database pertama kali dibuat. Di sini, tabel pelanggan dibuat menggunakan perintah SQL yang sudah didefinisikan sebelumnya.

### 3.4 Metode onUpgrade

```

public void onUpgrade(SQLiteDatabase db, int
oldVersion, int newVersion)
{
    db.execSQL("DROP TABLE IF EXISTS " +
TABLE_PELANGGAN);
    onCreate(db);
}

```

Metode ini dipanggil ketika database perlu di-upgrade (misalnya, jika versi database berubah). Tabel yang ada akan dihapus, dan tabel baru akan dibuat.

### 3.5 Metode insertPelanggan

```

public boolean insertPelanggan(ModelPelanggan mp) {
    SQLiteDatabase database =
this.getReadableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(KEY_PELANGGAN_ID, mp.getId());
    contentValues.put(KEY_PELANGGAN_NAMA,
mp.getNama());
    contentValues.put(KEY_PELANGGAN_EMAIL,
mp.getEmail());
    contentValues.put(KEY_PELANGGAN_HP, mp.getHp());
    long id = database.insert(TABLE_PELANGGAN, null,
contentValues);
    database.close();
    if (id != -1) {
        return true;
    } else {
        return false;
    }
}

```

- **Input:** Objek ModelPelanggan yang berisi data pelanggan (ID, nama, email, dan nomor handphone).
- **Proses:** Data pelanggan dimasukkan ke dalam tabel pelanggan menggunakan ContentValues. Fungsi insert() pada SQLite digunakan untuk menyimpan data.

- **Output:** Mengembalikan `true` jika penyimpanan berhasil, dan `false` jika terjadi kesalahan.

### 3.6 Metode `getPelanggan`

```
public List<ModelPelanggan> getPelanggan() {
    List<ModelPelanggan> pel = new
ArrayList<ModelPelanggan>();
    String query = "SELECT * FROM " + TABLE_PELANGGAN;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(query, null);
    if (cursor.moveToFirst()) {
        do{
            ModelPelanggan k = new ModelPelanggan();
            k.setId(cursor.getString(0));
            k.setNama(cursor.getString(1));
            k.setEmail(cursor.getString(2));
            k.setHp(cursor.getString(3));
            pel.add(k);
        }while (cursor.moveToNext());
    }
    return pel;
}
```

- **Proses:** Mengambil semua data pelanggan dari tabel pelanggan menggunakan query `SELECT *`. Hasil query disimpan dalam `Cursor` yang digunakan untuk iterasi baris per baris.
- **Output:** Mengembalikan daftar (`List`) dari objek `ModelPelanggan`, yang berisi semua pelanggan yang ada di database.

## 4. Kelas `tambahanPelanggan`

```
package com.example.myapplication;

import android.annotation.SuppressLint;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
```

```

import android.view.animation.AlphaAnimation;
import android.widget.Button;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

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

import adapter.AdapterPelanggan;
import database.SQLiteHelper;
import model.ModelPelanggan;
import pelanggan.PelangganAddActivity;

public class tambahanPelanggan extends AppCompatActivity
{
    SQLiteHelper db;
    Button btnPelAdd;
    RecyclerView rvPelanggan;
    AdapterPelanggan adapterPelanggan;
    ArrayList<ModelPelanggan> list;
    ProgressDialog progressDialog;
    AlphaAnimation btnAnimasi = new
AlphaAnimation(1F, 0.5F);

    @SuppressWarnings("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

        EdgeToEdge.enable(this);

        setContentView(R.layout.activity_tambahan_pelanggan);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
            (v, insets) -> {
                Insets systemBars
                =insets.getInsets(WindowInsetsCompat.Type.systemBars());
                v.setPadding(systemBars.left,
                    systemBars.top, systemBars.right,
                        systemBars.bottom);
                return insets;
            });
        setContentView();
        eventHandling();
        getData();
    }

    private View.OnClickListener onClickListener = new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                v.startAnimation(btnAnimasi);
                RecyclerView.ViewHolder viewHolder =
                (RecyclerView.ViewHolder)
                    v.getTag();
                int position =
                viewHolder.getAdapterPosition();
                ModelPelanggan mp =
                list.get(position);

                Toast.makeText(tambahanPelanggan.this, "" +mp.getNama(),
                    Toast.LENGTH_SHORT).show();
            }

```



```

        };

private void getData() {
    list.clear();
    showMsg();
    progressDialog.dismiss();
    try {
        List<ModelPelanggan> p = db.getPelanggan();
        if (p.size() > 0){
            for (ModelPelanggan pel : p){
                ModelPelanggan mp = new
ModelPelanggan();

                mp.setId(pel.getId());
                mp.setNama(pel.getNama());
                mp.setEmail(pel.getEmail());
                mp.setHp(pel.getHp());
                list.add(mp);
            }

            adapterPelanggan = new
AdapterPelanggan(this, list);

            adapterPelanggan.notifyDataSetChanged();
            rvPelanggan.setAdapter(adapterPelanggan);

adapterPelanggan.setOnItemClickListener(onClickListener);
        }else{
            Toast.makeText(this, "Data tidak
ditemukan",

                Toast.LENGTH_SHORT).show();
        }
    }catch (Exception e){
    }
}

private void eventHandling() {
    btnPelAdd.setOnClickListener(new
View.OnClickListener() {

```

```

        @Override
        public void onClick(View v) {
            startActivity(new
Intent (tambahanPelanggan.this,
PelangganAddActivity.class));
        }
    });
}

private void setView() {
    db = new SQLiteHelper(this);
    progressDialog = new ProgressDialog(this);
    btnPelAdd = (Button)
findViewById(R.id.btnPelAdd);
    rvPelanggan = (RecyclerView)
findViewById(R.id.rvPelanggan);
    list = new ArrayList<>();
    LinearLayoutManager llm = new
LinearLayoutManager(this);
    llm.setOrientation(LinearLayoutManager.VERTICAL);
    rvPelanggan.setHasFixedSize(true);
    rvPelanggan.setLayoutManager(llm);
}

private void showMsg() {
    progressDialog.setTitle("Informasi");
    progressDialog.setMessage("Loading Data..");
    progressDialog.setCancelable(false);
    progressDialog.show();
}
}
}

```

#### 4.1 Deklarasi Variabel

- SQLiteHelper db: Objek yang digunakan untuk berinteraksi dengan database SQLite.

- Button `btnPelAdd`: Tombol yang memungkinkan pengguna menambahkan pelanggan baru dengan membuka `PelangganAddActivity`.
- RecyclerView `rvPelanggan`: Digunakan untuk menampilkan daftar pelanggan dalam bentuk list.
- AdapterPelanggan `adapterPelanggan`: Adapter yang menghubungkan data pelanggan dengan RecyclerView untuk ditampilkan.
- `ArrayList<ModelPelanggan> list`: List untuk menyimpan data pelanggan.
- ProgressDialog `progressDialog`: Dialog yang muncul saat data sedang di-load.
- AlphaAnimation `btnAnimasi`: Efek animasi saat tombol diklik (mengubah opacity dari 1 menjadi 0.5).

#### 4.2 Metode onCreate

- **Inisialisasi View**: Memanggil `setView()` untuk menginisialisasi komponen UI seperti tombol, RecyclerView, dan lainnya.
- **Event Handling**: Memanggil `eventHandling()` untuk menetapkan tindakan ketika tombol ditekan (seperti menambahkan pelanggan baru).
- **Mengambil Data**: Memanggil `getData()` untuk mengambil data pelanggan dari database dan menampilkannya dalam RecyclerView.

#### 4.3 Metode getData

- **Membersihkan List**: Setiap kali `getData()` dipanggil, daftar list dihapus agar tidak ada duplikasi data.
- **Menampilkan Dialog Progress**: Fungsi `showMsg()` digunakan untuk menampilkan dialog ProgressDialog yang menunjukkan bahwa data sedang di-load.
- **Mengambil Data Pelanggan dari Database**  

```
List<ModelPelanggan> p = db.getPelanggan();
```
- **Error Handling**: Ada blok try-catch untuk menangkap error saat mengambil data dari database.

#### 5. Activity\_Pelanggan\_add.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".tambahanPelanggan">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_margin="@dimen/lg">
        <TextView
            android:textColor="@color/black"
            android:textStyle="bold"
            android:layout_marginBottom="@dimen/lg"
            android:textSize="23sp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="PELANGGAN"/>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Nama"

            android:textColor="@android:color/black"/>

```

```

        <EditText
            android:id="@+id/edPelAddNama"
            android:layout_width="match_parent"
            android:layout_height="48dp"
            android:layout_marginTop="@dimen/s"
            android:background="@drawable/border"

            android:drawableLeft="@drawable/ic_username"
            android:drawablePadding="@dimen/s"
            android:padding="@dimen/md"
            tools:ignore="SpeakableTextPresentCheck"
        />

    </LinearLayout>
    <LinearLayout
        android:layout_marginTop="@dimen/md"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Email"

            android:textColor="@android:color/black"/>

        <EditText
            android:id="@+id/edPelAddEmail"
            android:layout_width="match_parent"
            android:layout_height="48dp"
            android:layout_marginTop="@dimen/s"
            android:background="@drawable/border"
            android:drawableLeft="@drawable/ic_email"
            android:drawablePadding="@dimen/lg"
            android:inputType="textEmailAddress"

```

```

        android:padding="@dimen/md"
        tools:ignore="SpeakableTextPresentCheck"
    />

</LinearLayout>
<LinearLayout
    android:layout_marginTop="@dimen/md"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="HP"

        android:textColor="@android:color/black"/>

    <EditText
        android:id="@+id/edPelAddHp"
        android:layout_width="match_parent"
        android:layout_height="48dp"
        android:layout_marginTop="@dimen/s"
        android:background="@drawable/border"
        android:drawableLeft="@drawable/ic_hp"
        android:drawablePadding="@dimen/s"
        android:inputType="number"
        android:padding="@dimen/md"
        tools:ignore="SpeakableTextPresentCheck"
    />

</LinearLayout>
<LinearLayout
    android:layout_marginTop="@dimen/lg"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"

```

```

        android:weightSum="2">
        <Button
            android:id="@+id/btnPelAddSimpan"
            android:layout_marginRight="@dimen/xs"
            android:background="@color/primary"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Simpan"/>
        <Button
            android:id="@+id/btnPelAddBatal"
            android:layout_marginLeft="@dimen/xs"
            android:textColor="@color/white"
            android:background="@color/black"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Batal"/>
    </LinearLayout>
</LinearLayout>
</LinearLayout>

```

#### 6. Item\_pelanggan.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_marginTop="@dimen/sm"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <androidx.cardview.widget.CardView
        android:id="@+id/cvItemPel"
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        app:cardCornerRadius="@dimen/xs"
    >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="@dimen/lg"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:weightSum="2"
            android:orientation="horizontal">
            <LinearLayout
                android:layout_weight="1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical">
                <TextView
                    android:id="@+id/tvItemPelNama"
                    android:textSize="15sp"
                    android:textStyle="bold"

android:layout_width="wrap_content"

android:layout_height="wrap_content"
                    android:text="FULAN"

android:textColor="@color/primary"/>
                <TextView
                    android:id="@+id/tvItemPelHp"
                    android:textSize="12sp"

android:layout_marginTop="@dimen/xs"

```



```

        android:textColor="@color/black"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"
        android:text="0853xxxx"/>
    </LinearLayout>
</LinearLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>

```

## 7. ModelLayanan

```

package model;

public class ModelLayanan {
    String id, nama, harga;

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getNama() {
        return nama;
    }

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

    public void setHarga(String harga) {
        this.harga = harga;
    }

    public String getHarga() {
        return harga;
    }
}

```

## 8. LayananAddActivity

```

package layanan;

import android.os.Bundle;
import android.widget.EditText;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;

```

```

import androidx.core.view.WindowInsetsCompat;

import com.example.myapplication.R;

public class LayananAddActivity extends AppCompatActivity {
    EditText nama, harga;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_layanan_add);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
        (v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
            return insets;
        });

        nama = (EditText) findViewById(R.id.edNama);
        harga = (EditText) findViewById(R.id.edHarga);
    }
}

```

## 9. Layanan

```

package com.example.myapplication;

import android.annotation.SuppressLint;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.animation.AlphaAnimation;
import android.widget.Button;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

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

import adapter.AdapterLayanan;
import adapter.AdapterPelanggan;
import database.SQLiteHelper;
import database.SQLiteHelper2;
import model.ModelLayanan;
import model.ModelPelanggan;
import pelanggan.PelangganAddActivity;
public class layanan extends AppCompatActivity{
    SQLiteHelper2 db;
    Button btnInfo;
    RecyclerView rvLayanan;
    AdapterLayanan adapterLayanan;
}

```

```

        ArrayList<ModelLayanan> list;
        ProgressDialog progressDialog;
        AlphaAnimation btnAnimasi = new AlphaAnimation(1F, 0.5F);
        @SuppressWarnings("MissingInflatedId")
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            EdgeToEdge.enable(this);
            setContentView(R.layout.activity_layanan);

            ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
                (v, insets) -> {
                    Insets systemBars
                    =insets.getInsets(WindowInsetsCompat.Type.systemBars());
                    v.setPadding(systemBars.left, systemBars.top,
                        systemBars.right,
                            systemBars.bottom);
                    return insets;
                });

            setContentView();
            eventHandling();
            getData();
        }

        private View.OnClickListener onClickListener = new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {
                v.startAnimation(btnAnimasi);
                RecyclerView.ViewHolder viewHolder =
                (RecyclerView.ViewHolder) v.getTag();
                int position = viewHolder.getAdapterPosition();
                ModelLayanan ml = list.get(position);
                Toast.makeText(layanan.this, "Layanan: " + ml.getNama(),
                // Ganti nama
                    Toast.LENGTH_SHORT).show();
            }
        };

        private void getData() {
            list.clear(); // Bersihkan list sebelum mengambil data baru
            showMsg(); // Tampilkan loading dialog
            progressDialog.dismiss(); // Sembunyikan loading dialog
            setelah data berhasil diambil

            try {
                List<ModelLayanan> l = db.getModelLayanan(); // Ambil
                data dari database
                if (l.size() > 0) {
                    for (ModelLayanan lay : l) {
                        ModelLayanan ml = new ModelLayanan();
                        ml.setId(lay.getId());
                        ml.setNama(lay.getNama());
                        ml.setHarga(lay.getHarga());
                        list.add(ml); // Tambahkan data ke list
                    }

                    adapterLayanan = new AdapterLayanan(this, list);
                    adapterLayanan.notifyDataSetChanged();
                    rvLayanan.setAdapter(adapterLayanan);
                }
            }
        }
    }

```

```

adapterLayanan.setOnItemClickListener(onClickListener);
    } else {
        Toast.makeText(this, "Data tidak ditemukan",
Toast.LENGTH_SHORT).show();
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

private void eventHandling() {
    btnInfo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(layanan.this,
LayananAddActivity.class));
        }
    });
}

private void setView() {
    db = new SQLiteHelper2(this); // Inisialisasi SQLiteHelper
    progressDialog = new ProgressDialog(this); // Inisialisasi
ProgressDialog
    btnInfo = findViewById(R.id.btnInfo); // Temukan tombol
tambah pelanggan dari layout
    rvLayanan = findViewById(R.id.rvLayanan); // Temukan
RecyclerView dari layout
    list = new ArrayList<>(); // Inisialisasi list untuk
menampung data pelanggan

    // Inisialisasi LinearLayoutManager untuk RecyclerView
    LinearLayoutManager llm = new LinearLayoutManager(this);
    llm.setOrientation(LinearLayoutManager.VERTICAL); // Mengatur
orientasi menjadi vertikal
    rvLayanan.setHasFixedSize(true); // Optimalkan ukuran
RecyclerView
    rvLayanan.setLayoutManager(llm); // Set layout manager ke
RecyclerView
}

private void showMsg() {
    if (progressDialog == null) {
        progressDialog = new ProgressDialog(this);
        progressDialog.setTitle("Informasi");
        progressDialog.setMessage("Loading Data...");
        progressDialog.setCancelable(false); // Tidak bisa
dibatalkan dengan menekan di luar
    }
    progressDialog.show(); // Tampilkan dialog
}
}
}

```

## 10. AdapterLayanan

```

package adapter;

import android.content.Context;
import android.view.LayoutInflater;

```

```

import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

import com.example.myapplication.R;
import model.ModelLayanan;

public class AdapterLayanan extends
RecyclerView.Adapter<AdapterLayanan.ViewHolder> {
    private List<ModelLayanan> layananList;
    private Context context;

    public AdapterLayanan(Context context, List<ModelLayanan>
layananList) {
        this.context = context;
        this.layananList = layananList;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        // Menginflate layout item_layanan
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.item_layana
n, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        ModelLayanan layanan = layananList.get(position);
        // Menggunakan ID yang sesuai di item_layanan.xml
        holder.tvNamaLayanan.setText(layanan.getNama()); // Ganti ke
getJenisLayanan()
        holder.tvHarga.setText(layanan.getHarga());
    }

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

    public void setOnItemClickListener(View.OnClickListener
onClickListener) {
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        TextView tvNamaLayanan, tvHarga;

        public ViewHolder(View itemView) {
            super(itemView);
            // Menggunakan ID yang sesuai dari item_layanan.xml
            tvNamaLayanan =
itemView.findViewById(R.id.tvItemLayNama); // Ganti ID dengan yang
sesuai
            tvHarga = itemView.findViewById(R.id.tvItemLayHarga); //

```

```

        Ganti ID dengan yang sesuai
    }
}

```

## 11. SQLiteHelper2

```

package database;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

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

import model.ModelLayanan;
import model.ModelPelanggan;

public class SQLiteHelper2 extends SQLiteOpenHelper {
    public static String DATABASE_NAME = "my_laundry2.db";
    public static final int DATABASE_VERSION = 1;
    public static final String TABLE_LAYANAN = "layanan";
    public static final String KEY_LAYANAN_ID = "id";
    public static final String KEY_LAYANAN_NAMA = "nama";
    public static final String KEY_LAYANAN_HARGA = "harga";
    private static final String CREATE_TABLE_LAYANAN = "CREATE TABLE
" + TABLE_LAYANAN + " ("
        + KEY_LAYANAN_ID + " TEXT PRIMARY KEY, " // ID layanan
        sebagai PRIMARY KEY
        + KEY_LAYANAN_NAMA + " TEXT NOT NULL, " // Nama layanan
        + KEY_LAYANAN_HARGA + " TEXT NOT NULL" + // Harga
        layanan
        ");";

    public SQLiteHelper2(Context context){
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(CREATE_TABLE_LAYANAN);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion)
    {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_LAYANAN);
        onCreate(db);
    }
    public boolean insertlayanan(ModelLayanan ml){
        SQLiteDatabase database = this.getReadableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(KEY_LAYANAN_ID, ml.getId());
        contentValues.put(KEY_LAYANAN_NAMA, ml.getNama());
        contentValues.put(KEY_LAYANAN_HARGA, ml.getHarga());
        long id = database.insert(TABLE_LAYANAN, null,
contentValues);
        database.close();
        if (id != -1){
            return true;
        }
    }
}

```

```

        }else{
            return false;
        }
    }
    public List<ModellLayanan> getModelLayanan(){
        List<ModellLayanan> pel = new ArrayList<ModellLayanan>();
        String query = "SELECT * FROM " + TABLE_LAYANAN;
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(query, null);
        if (cursor.moveToFirst()){
            do{
                ModellLayanan k = new ModellLayanan();
                k.setId(cursor.getString(0));
                k.setNama(cursor.getString(1));
                k.setHarga(cursor.getString(2));
                pel.add(k);
            }while (cursor.moveToNext());
        }
        return pel;
    }
}

```