

**LAPORAN TUGAS**

**APLIKASI MOBILE PRAKTIKUM 3**

“Database SQLite (Create dan Read Data)”



Oleh:

**Octavio Yusda Wildani**

**NIM. 2211533020**

Dosen Pengampu Mata Kuliah

**Nurfiah, S.ST, M.Kom**

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**PADANG**

**2024**

## **A. Pendahuluan**

Modul praktikum ini berfokus pada pengembangan aplikasi mobile berbasis Android menggunakan bahasa pemrograman Java. Salah satu aspek penting dalam pengembangan aplikasi adalah pengelolaan data. Pada modul ini, kita mempelajari cara menggunakan dan memanipulasi database SQLite di Android. Praktikum ini memberikan langkah-langkah dalam membuat koneksi, mengelola database SQLite, dan menampilkan data yang tersimpan menggunakan RecyclerView.

## **B. Tujuan**

1. Membuat database dan tabel SQLite pada aplikasi Android.
2. Melakukan operasi dasar seperti menambah (insert) dan membaca (read) data dari database SQLite.
3. Menampilkan data dari database SQLite dalam bentuk daftar menggunakan RecyclerView.

## **C. Teori**

SQLite adalah sistem manajemen basis data (DBMS) yang tertanam (embedded) dan umum digunakan di perangkat Android. Data disimpan secara lokal dalam file teks dan tidak memerlukan koneksi internet untuk diakses. SQLite mendukung berbagai operasi database seperti menambah, menghapus, memperbarui, dan mengambil data (CRUD - Create, Read, Update, Delete).

RecyclerView adalah komponen tampilan di Android yang digunakan untuk menampilkan daftar data dengan lebih efisien dibandingkan ListView. RecyclerView memberikan fleksibilitas dalam pengaturan tampilan dan performa yang lebih optimal, terutama untuk jumlah data yang besar. Pada praktikum ini, kita belajar membuat model data, mengelola database dengan menggunakan SQLiteHelper, serta menampilkan data dalam daftar melalui RecyclerView.

## **D. Langkah-langkah Praktikum**

1. Model Pelanggan

Kita akan membuat package baru bernama "model" dan menambahkan class baru bernama ModelPelanggan. Di dalam class ini, kita akan membuat method getter dan setter seperti pada contoh di bawah.

```
ModelPelanggan.java x
1 package id.my.lauvi.laundryapps.model;
2
3 public class ModelPelanggan { 21 usages
4     String id, nama, email, hp; 2 usages
5
6     public String getId(){ return id; }
7     public void setId(String id){ this.id = id; }
8
9     public String getNama() { return nama ;} 4 usages
10    public void setNama(String nama) { this.nama = nama; } 3 usages
11
12    public String getEmail() { return email ;} 3 usages
13    public void setEmail(String email) { this.email = email; } 3 usages
14
15    public String getHp() { return hp ;} 4 usages
16    public void setHp(String hp) { this.hp = hp; } 3 usages
17 }
18
```

## 2. Membuat Class Database Helper

Selanjutnya, kita membuat package baru bernama "database" dan membuat class Java baru bernama SQLiteHelper. Class ini bertujuan untuk membuat dan memodifikasi database SQLite serta melakukan operasi CRUD (Create, Read, Update, Delete).

```
package id.my.lauvi.laundryapps.database;

import ...

public class SQLiteHelper extends SQLiteOpenHelper { 9 usages
    public static String DATABASE_NAME = "my_laundry.db"; 1 usage
    public static final int DATABASE_VERSION = 1; 1 usage
    public static final String TABLE_PELANGGAN = "pelanggan"; 6 usages
    public static final String KEY_PELANGGAN_ID = "pelanggan_id"; 4 usages
    public static final String KEY_PELANGGAN_NAMA = "nama"; 2 usages
    public static final String KEY_PELANGGAN_EMAIL = "email"; 2 usages
    public static final String KEY_PELANGGAN_HP = "hp"; 2 usages
    private static final String CREATE_TABLE_PELANGGAN = "CREATE TABLE " + 1 usage
        TABLE_PELANGGAN + "("
        + KEY_PELANGGAN_ID + " TEXT, " + KEY_PELANGGAN_NAMA + " TEXT, " +
        KEY_PELANGGAN_EMAIL + " TEXT, " + KEY_PELANGGAN_HP + " TEXT )";
    public SQLiteHelper(Context context){ 3 usages
        super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) { db.execSQL(CREATE_TABLE_PELANGGAN); }
    @Override no usages
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
    {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_PELANGGAN);
        onCreate(db);
    }
}
```

```

public boolean insertPelanggan(ModelPelanggan mp){ 1 usage
    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, nullColumnHack: 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, selectionArgs: 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;
}

public ModelPelanggan getPelangganById(String id){ no usages
    SQLiteDatabase db = this.getReadableDatabase();
    String query = "SELECT * FROM " + TABLE_PELANGGAN + " WHERE " + KEY_PELANGGAN_ID + " = ?";
    Cursor cursor = db.rawQuery(query, new String[]{id});
    ModelPelanggan pelanggan = null;
    if (cursor.moveToFirst()){
        pelanggan = new ModelPelanggan();
        pelanggan.setId(cursor.getString(0));
        pelanggan.setNama(cursor.getString(1));
        pelanggan.setEmail(cursor.getString(2));
        pelanggan.setHp(cursor.getString(3));
    }
    cursor.close();
    return pelanggan;
}

public boolean deletePelanggan(String id) { 1 usage
    SQLiteDatabase db = this.getWritableDatabase();
    int result = db.delete(TABLE_PELANGGAN, whereClause: KEY_PELANGGAN_ID + " = ?", new String[]{id});
    db.close();
    return result > 0;
}

```

### 3. Membuat Data Pelanggan

Kita akan membuat activity baru bernama PelangganAddActivity untuk menginput data 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"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:background="@color/white"
    tools:context=".pelanggan.PelangganAddActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_margin="16dp">
        <TextView
            android:textColor="@color/hijau1"
            android:textStyle="bold"
            android:layout_marginBottom="16dp"
            android:textSize="23sp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="PELANGGAN"/>

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    <LinearLayout
        <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="wrap_content"
                android:layout_marginTop="12dp"
                android:background="@drawable/border"
                android:drawableLeft="@drawable/ic_username"
                android:drawablePadding="7dp"
                android:padding="12dp"
                android:hint="Masukkan nama anda"/>
        </LinearLayout>
    </LinearLayout>

```

```

47         <LinearLayout
48             android:layout_marginTop="12dp"
49             android:layout_width="match_parent"
50             android:layout_height="wrap_content"
51             android:orientation="vertical">
52             <TextView
53                 android:layout_width="match_parent"
54                 android:layout_height="wrap_content"
55                 android:text="Email"
56                 android:textColor="@android:color/black"/>
57
58             <EditText
59                 android:id="@+id/edPelAddEmail"
60                 android:layout_width="match_parent"
61                 android:layout_height="wrap_content"
62                 android:layout_marginTop="12dp"
63                 android:background="@drawable/border"
64                 android:drawableLeft="@drawable/ic_email"
65                 android:drawablePadding="8dp"
66                 android:inputType="textEmailAddress"
67                 android:padding="12dp"
68                 android:hint="Masukkan email anda"/>
69         </LinearLayout>

```

```

71         <LinearLayout
72             android:layout_marginTop="12dp"
73             android:layout_width="match_parent"
74             android:layout_height="wrap_content"
75             android:orientation="vertical">
76             <TextView
77                 android:layout_width="match_parent"
78                 android:layout_height="wrap_content"
79                 android:text="HP"
80                 android:textColor="@android:color/black"/>
81
82             <EditText
83                 android:id="@+id/edPelAddHp"
84                 android:layout_width="match_parent"
85                 android:layout_height="wrap_content"
86                 android:layout_marginTop="8dp"
87                 android:background="@drawable/border"
88                 android:drawableLeft="@drawable/ic_hp"
89                 android:drawablePadding="8dp"
90                 android:inputType="number"
91                 android:padding="12dp"
92                 android:hint="Masukkan nomor anda" />
93         </LinearLayout>

```

```


94      <LinearLayout
95      |
96      |     android:layout_width="match_parent"
97      |     android:layout_height="wrap_content"
98      |     android:orientation="horizontal"
99      |     android:weightSum="2">
100     <Button
101     |     android:id="@+id/btnPelAddSimpan"
102     |     android:layout_width="match_parent"
103     |     android:layout_height="wrap_content"
104     |     android:layout_marginRight="4dp"
105     |     android:layout_weight="1"
106     |     android:backgroundTint="@color/hijau1"
107     |     android:text="Simpan"
108     |     android:textColor="@color/white" />
109     <Button
110     |     android:id="@+id/btnPelAddBatal"
111     |     android:layout_marginLeft="4dp"
112     |     android:textColor="@color/red"
113     |     android:backgroundTint="@color/hijau1"
114     |     android:layout_width="match_parent"
115     |     android:layout_height="wrap_content"
116     |     android:layout_weight="1"
117     |     android:text="Batal"/>
118     </LinearLayout>
119 </LinearLayout>

```


- Output

**PELANGGAN**


Nama

 Masukkan nama anda

Email

 Masukkan email anda

HP

 Masukkan nomor anda

SIMPAN

BATAL

- Java

```

PelangganAddActivity.java x
1 package id.my.lauvi.laundryapps.pelanggan;
2
3 > import ...
21
22 </> public class PelangganAddActivity extends AppCompatActivity {
23
24     EditText nama, email, hp; 2 usages
25     Button btnSimpan, btnBatal; 2 usages
26     SQLiteHelper db; 2 usages
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         EdgeToEdge.enable(this);
32         setContentView(R.layout.activity_pelanggan_add_activity);
33         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
34             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
35             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
36             return insets;
37         });
38
39         nama = (EditText) findViewById(R.id.edPelAddNama);
40         email = (EditText) findViewById(R.id.edPelAddEmail);
41         hp = (EditText) findViewById(R.id.edPelAddHp);
42         btnSimpan = (Button) findViewById(R.id.btnPelAddSimpan);
43         btnBatal = (Button) findViewById(R.id.btnPelAddBatal);
44
45         db = new SQLiteHelper(context: PelangganAddActivity.this);
46         btnSimpan.setOnClickListener(new View.OnClickListener() {
47             @Override
48             public void onClick(View view) {
49                 ModelPelanggan mp = new ModelPelanggan();
50                 String uniqueID = UUID.randomUUID().toString();
51                 mp.setId(uniqueID);
52                 mp.setNama(nama.getText().toString());
53                 mp.setEmail(email.getText().toString());
54                 mp.setHp(hp.getText().toString());
55
56                 Toast.makeText(context: PelangganAddActivity.this,
57                     text: ""+mp.getId()+mp.getNama()+mp.getEmail()+mp.getHp(),
58                     Toast.LENGTH_SHORT).show();
59                 boolean cek = db.insertPelanggan(mp);
60                 if(cek == true){
61                     Toast.makeText(context: PelangganAddActivity.this,
62                         text: "Data berhenti disimpan", Toast.LENGTH_SHORT).show();
63                     startActivity(new Intent(packageContext: PelangganAddActivity.this,
64                         PelangganActivity.class));
65                     finish();
66                 }else {
67                     Toast.makeText(context: PelangganAddActivity.this, text: "Data gagal disimpan",
68                         Toast.LENGTH_SHORT).show();
69                 }
70             }
71         });
72         btnBatal.setOnClickListener(new View.OnClickListener() {
73             @Override
74             public void onClick(View view) { finish(); }
75         });
76     }
77 }
78
79 }

```



#### 4. Menambahkan Permission

Permission untuk menggunakan database SQLite pada Android perlu ditambahkan di Android Manifest. Permission yang diperlukan adalah `READ_EXTERNAL_STORAGE`.

```
AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools">
4     <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

#### 5. Menampilkan Data Pelanggan

Setelah data berhasil disimpan di database, kita akan menampilkan data pelanggan menggunakan RecyclerView.

- XML

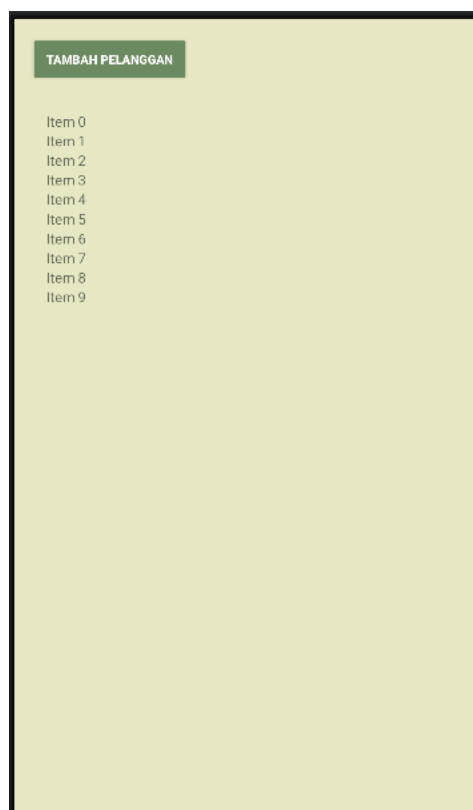
```
activity_pelanggan.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical"
9     android:background="@color/ivory"
10    tools:context=".pelanggan.PelangganActivity">
11    <LinearLayout
12        android:layout_margin="16dp"
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:orientation="vertical">
16
17        <Button
18            android:id="@+id/btnPelAdd"
19            android:layout_width="wrap_content"
20            android:layout_height="wrap_content"
21            android:layout_marginBottom="12dp"
22            android:backgroundTint="@color/hijau1"
23            android:textColor="@color/white"
24            android:padding="16dp"
25            android:text="Tambah Pelanggan"
26            android:textSize="12sp"/>
```

```

21         android:layout_marginBottom="12dp"
22         android:backgroundTint="@color/hijau1"
23         android:textColor="@color/white"
24         android:padding="16dp"
25         android:text="Tambah Pelanggan"
26         android:textSize="12sp"
27         android:textStyle="bold">
28     </Button>
29     <androidx.recyclerview.widget.RecyclerView
30         android:id="@+id/rvPelanggan"
31         android:layout_width="match_parent"
32         android:layout_height="0dp"
33         android:layout_weight="1"
34         android:padding="16dp">
35     </androidx.recyclerview.widget.RecyclerView>
36 </LinearLayout>
37
38 </LinearLayout>

```

- Output



- Item Pelanggan

Selanjutnya kita akan membuat layout baru bernama item\_pelanggan untuk tampilan item dari list yang ditampilkan.

FULAN  
0853xxxx

- Custom Adapter

Kita akan membuat Custom Adapter dengan menambahkan Java class baru bernama AdapterPelanggan di package "adapter".

```
AdapterPelanggan.java x
1 package id.my.lauvi.laundryapps.adapter;
2
3 > import ...
15
16 public class AdapterPelanggan extends RecyclerView.Adapter<AdapterPelanggan.ViewHolder> {
17     private static final String TAG = AdapterPelanggan.class.getSimpleName();
18     private Context context;
19     private List<ModelPelanggan> list;
20     private View.OnClickListener onItemClick;
21     public void setOnItemClickListener(View.OnClickListener itemClickListener) {
22         onItemClick = itemClickListener;
23     }
24     public AdapterPelanggan(Context context, List<ModelPelanggan> list) {
25         this.context = context;
26         this.list = list;
27     }
28     @Override
29     public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
30         View view =
31             LayoutInflater.from(parent.getContext()).inflate(R.layout.item_pelanggan,
32                 parent, attachToRoot: false);
33         return new ViewHolder(view);
34     }
35
36     @Override
37     public void onBindViewHolder(ViewHolder holder, int position) {
38         ModelPelanggan item = list.get(position);
39         holder.tvNama.setText(item.getNama());
40         holder.tvHp.setText(item.getHp());
41     }
42     @Override
43     public int getItemCount() { return list.size(); }
44     public void clear() {
45         int size = this.list.size();
46         this.list.clear();
47         notifyItemRangeRemoved(0, size);
48     }
49     public class ViewHolder extends RecyclerView.ViewHolder {
50         TextView tvNama, tvHp;
51         public ViewHolder(View itemView) {
52             super(itemView);
53             tvNama = (TextView) itemView.findViewById(R.id.tvItemPelNama);
54             tvHp = (TextView) itemView.findViewById(R.id.tvItemPelHp);
55             itemView.setTag(this);
56             itemView.setOnClickListener(onItemClick);
57         }
58     }
59 }
```

- Pelanggan Activity

```
1 package id.my.lauvi.laundryapps.pelanggan;
2
3 > import ...
27
28 <> public class PelangganActivity extends AppCompatActivity{
29     SQLiteHelper DB; no usages
30     TextView btnPelAdd; 1 usage
31     RecyclerView rvPelanggan; 4 usages
32     AdapterPelanggan adapterPelanggan; 4 usages
33     ArrayList<ModelPelanggan> list; 5 usages
34     ProgressDialog progressDialog; 6 usages
35     AlphaAnimation btnAnimasi = new AlphaAnimation(1F,0.5F); 1 usage
36
37     @SuppressWarnings("MissingInflatedId")
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         setContentView(R.layout.activity_pelanggan);
42
43         // Handling insets for edge-to-edge layout
44         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
45             v.setPadding(insets.getInsets(WindowInsetsCompat.Type.systemBars()).left,
46                 insets.getInsets(WindowInsetsCompat.Type.systemBars()).top,
47                 insets.getInsets(WindowInsetsCompat.Type.systemBars()).right,
48                 insets.getInsets(WindowInsetsCompat.Type.systemBars()).bottom);
49             return insets;
50         });
51
52         setView();
53         eventHandling();
54         getData();
55     }
56     private void setView(){ 1 usage
57         db = new SQLiteHelper( context: this);
58         progressDialog = new ProgressDialog( context: this);
59         btnPelAdd = findViewById(R.id.btnPelAdd);
60         rvPelanggan = findViewById(R.id.btnPelanggan);
61         list = new ArrayList<>();
62
63         LinearLayoutManager = new LinearLayoutManager( context: this);
64         llm.setOrientation(LinearLayoutManager.VERTICAL);
65         rvPelanggan.setLayoutManager(llm);
66         rvPelanggan.setHasFixedSize(true);
67     }
68
69     private void showMsg(){ 1 usage
70         progressDialog.setTitle("Informasi");
71         progressDialog.setMessage("Loading Data..");
72         progressDialog.setCancelable(false);
73         progressDialog.show();
74     }
```

```

76     private void getData(){ 1 usage
77         list.clear();
78         showMsg();
79         try {
80             List<ModelPelanggan> pelangganList = db.getPelanggan();
81             if (pelangganList.size() > 0) {
82                 list.addAll(pelangganList);
83
84                 adapterPelanggan = new AdapterPelanggan( context: this, list);
85                 rvPelanggan.setAdapter(adapterPelanggan);
86                 adapterPelanggan.setOnItemClickListener(onClickListener);
87                 adapterPelanggan.notifyDataSetChanged();
88             } else {
89                 Toast.makeText( context: this, text: "Data pelanggan tidak ditemukan", Toast.LENGTH_SHORT).show();
90             }
91         } catch (Exception e) {
92             e.printStackTrace();
93         } finally {
94             progressDialog.dismiss();
95         }
96     }

```

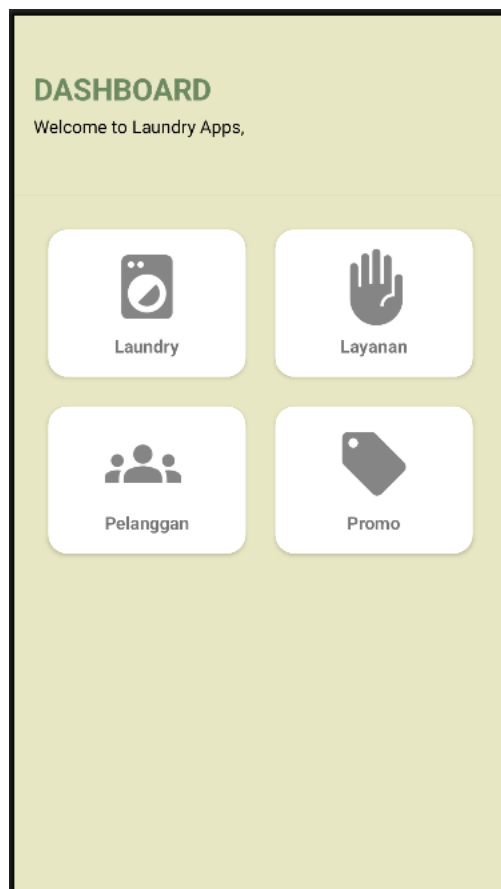
```

98     private View.OnClickListener onClickListener = new View.OnClickListener() { no usages
99         @Override
100         public void onClick(View view) {
101             view.startAnimation(btnAnimasi);
102             RecyclerView.ViewHolder viewHolder = (RecyclerView.ViewHolder) view.getTag();
103             int position = viewHolder.getAdapterPosition();
104             ModelPelanggan mp = list.get(position);
105             Toast.makeText( context: PelangganActivity.this, text: "Pelanggan: " + mp.getNama(), Toast.LENGTH_SHORT).show();
106         }
107     };

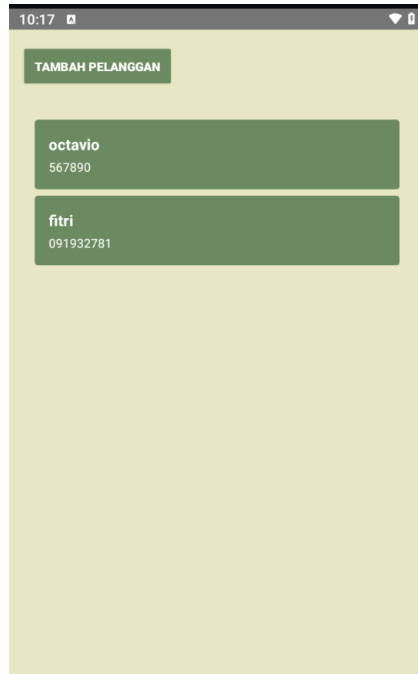
```

- Simulasi

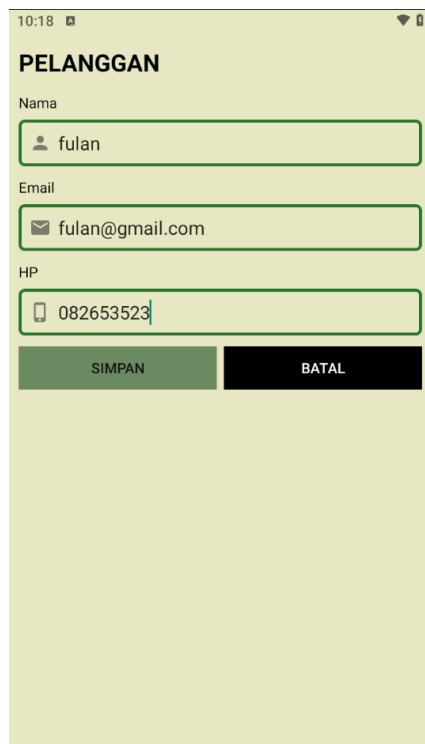
1. Halaman utama setelah login



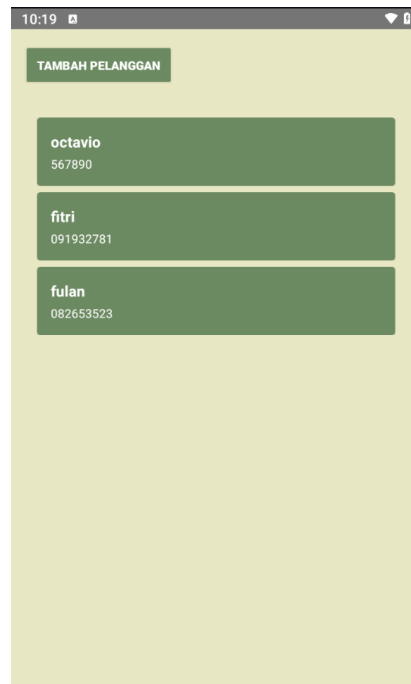
2. Halaman pelanggan ditampilkan setelah kita mengklik ikon pelanggan pada dashboard. Pada halaman ini, kita bisa menambahkan data pelanggan dengan mengklik tombol dibagian atas.



3. Selanjutnya akan dialihkan ke halaman Add Pelanggan yang memungkinkan kita untuk memasukkan data pelanggan dan menyimpannya dengan mengklik tombol simpan.

A screenshot of a mobile application interface for adding a new customer. The title 'PELANGGAN' is at the top. Below it, there are three input fields: 'Nama' (Name) with the value 'fulan', 'Email' with the value 'fulan@gmail.com', and 'HP' (Phone Number) with the value '082653523'. At the bottom, there are two buttons: 'SIMPAN' (Save) and 'BATAL' (Cancel).

4. Data pelanggan yang berhasil ditambahkan akan muncul dengan pemberitahuan dan ditampilkan pada daftar.



## E. Tugas

### 1. Membuat Model Layanan

Seperti model pelanggan, kita juga perlu membuat model layanan. Model ini membutuhkan tiga variabel: ID, Tipe Layanan, dan Harga. Pada model Layanan.java, kita akan menggunakan method getter dan setter.

```
package id.my.lauvi.laundryapps.model;

public class ModelLayanan {
    String id, tipe, harga;

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

    public String getType(){ return tipe;}
    public void setType(String tipe) { this.tipe = tipe ;}

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

## 2. Membuat Class Database Helper

Membuat class SQLiteHelper2 di dalam package “database” yang difokuskan pada fungsi layanan.

```
package id.my.lauvi.laundryapps.database;

> import ...

public class SQLiteHelper2 extends SQLiteOpenHelper { 9 usages
    public static String DATABASE_NAME = "my_laundry.db"; 1 usage
    public static final int DATABASE_VERSION = 3; 1 usage
    public static final String TABLE_LAYANAN = "layan"; 5 usages
    public static final String KEY_LAYANAN_ID = "layan_id"; 3 usages
    public static final String KEY_LAYANAN_TIPE = "tipeLayanan"; 2 usages
    public static final String KEY_LAYANAN_HARGA = "harga"; 2 usages
    private static final String CREATE_TABLE_LAYANAN = "CREATE TABLE " 1 usage
        + TABLE_LAYANAN + "(" + KEY_LAYANAN_ID +
        " TEXT, " + KEY_LAYANAN_TIPE + " TEXT, " + KEY_LAYANAN_HARGA + " TEXT )";

    > public SQLiteHelper2(Context context) { super(context, DATABASE_NAME, factory: null, DATABASE_VERSION
    @Override
    > public void onCreate(SQLiteDatabase db) { db.execSQL(CREATE_TABLE_LAYANAN); }
    @Override no usages
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_LAYANAN);
        onCreate(db);
    }
}
```

```
public boolean insertLayanan(ModelLayanan ml) { 1 usage
    SQLiteDatabase database = this.getReadableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(KEY_LAYANAN_ID, ml.getId());
    contentValues.put(KEY_LAYANAN_TIPE, ml.getTipe());
    contentValues.put(KEY_LAYANAN_HARGA, ml.getHarga());
    long id = database.insert(TABLE_LAYANAN, nullColumnHack: null, contentValues);
    database.close();
    if(id != -1){
        return true;
    }else {
        return false;
    }
}
```



```

public List<ModelLayanan> getLayanan(){
    List<ModelLayanan> lay = new ArrayList<>();
    String query = "SELECT * FROM " + TABLE_LAYANAN;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(query, selectionArgs: null);
    if (cursor.moveToFirst()){
        do{
            ModelLayanan k = new ModelLayanan();
            k.setId(cursor.getString(i: 0));
            k.setTipe(cursor.getString(i: 1));
            k.setHarga(cursor.getString(i: 2));
            lay.add(k);
        }while (cursor.moveToNext());
    }
    return lay;
}

```

### 3. Membuat Data Layanan

Membuat activity baru bernama LayananAddActivity untuk menambahkan layanan ke halaman LayananActivity. File XML-nya serupa dengan pelanggan, menggunakan RecyclerView.

- Output

11:19

**LAYANAN**

Layanan

Masukkan layanan terbaru

Harga

\$ Harga yang ditentukan

SIMPAN BATAL

- LayananAddActivity

```
package id.my.lauvi.laundryapps.layanan;

> import ...

> public class LayananAddActivity extends AppCompatActivity {
    EditText edTipe, edHarga; 2 usages
    Button btnLaySimpan, btnLayBatal; 2 usages
    SQLiteHelper2 db; 2 usages

    @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;
        });

        edTipe = (EditText) findViewById(R.id.edLayAddLayanan);
        edHarga = (EditText) findViewById(R.id.edLayAddHarga);
        btnLaySimpan = (Button) findViewById(R.id.btnLayAddSimpan);
        btnLayBatal = (Button) findViewById(R.id.btnLayAddBatal);
    }
}
```

```
db = new SQLiteHelper2(context: LayananAddActivity.this);
btnLaySimpan.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ModellLayanan ml = new ModellLayanan();
        String uniqueID = UUID.randomUUID().toString();
        ml.setId(uniqueID);
        ml.setTipe(edTipe.getText().toString());
        ml.setHarga(edHarga.getText().toString());

        Toast.makeText(context: LayananAddActivity.this,
            text: ""+ml.getId()+ml.getTipe()+ml.getHarga(), Toast.LENGTH_SHORT).show();
        boolean cek = db.insertLayanan(ml);
        if(cek == true){
            Toast.makeText(context: LayananAddActivity.this,
                text: "Data berhasil ditambahkan", Toast.LENGTH_SHORT).show();
            startActivity(new Intent(packageContext: LayananAddActivity.this,
                LayananActivity.class));
            finish();
        } else {
            Toast.makeText(context: LayananAddActivity.this, text: "Data gagal ditambahkan",
                Toast.LENGTH_SHORT).show();
        }
    }
});
```

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

- Item Layanan

Membuat layout baru bernama item\_layanan untuk menampilkan item dari list layanan.



- Custom Adapter

Membuat adapter untuk layanan. Buat class Java baru bernama AdapterLayanan di package "adapter" yang sudah ada. Adapter ini akan menghubungkan data layanan ke RecyclerView.

```
package id.my.lauvi.laundryapps.adapter;

> import ...

public class AdapterLayanan extends RecyclerView.Adapter<AdapterLayanan.ViewHolder> { 5 usages
    private static final String TAG = AdapterLayanan.class.getSimpleName(); no usages
    private Context context; 1 usage
    private List<ModelLayanan> list; 5 usages
    private View.OnClickListener onItemClick; 2 usages
    public void setOnItemClickListener(View.OnClickListener listener, 1 usage
                                     onItemClickListener){
        onItemClick = onItemClickListener;
    }
    public AdapterLayanan(Context context, List<ModelLayanan> list){ 1 usage
        this.context = context;
        this.list = list;
    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType){
        View view =
            LayoutInflater.from(parent.getContext()).inflate(R.layout.item_layanan,
                parent, attachToRoot: false);
        return new ViewHolder(view);
    }
}
```

```

@Override
public void onBindViewHolder(ViewHolder holder, int position){
    ModelLayanan item = list.get(position);
    holder.tvTipe.setText(item.getTipe());
    holder.tvHarga.setText(item.getHarga());
}

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

public void clear(){ no usages
    int size = this.list.size();
    this.list.clear();
    notifyItemRangeRemoved( positionStart: 0, size);
}

public class ViewHolder extends RecyclerView.ViewHolder{ 4 usages
    TextView tvTipe, tvHarga; 2 usages

    public ViewHolder(View itemView){ 1 usage
        super(itemView);
        tvTipe = (TextView) itemView.findViewById(R.id.tvLayanan);
        tvHarga = (TextView) itemView.findViewById(R.id.tvHarga);
        itemView.setTag(this);
        itemView.setOnClickListener(onItemClicked);
    }
}

```

- Layanan Activity

```

package id.my.lauvi.laundryapps.layanan;

import ...

public class LayananActivity extends AppCompatActivity {
    SQLiteHelper2 db; 2 usages
    TextView btnLayAdd; 2 usages
    RecyclerView rvLayanan; 4 usages
    AdapterLayanan adapterLayanan; 4 usages
    ArrayList<ModelLayanan> list; 5 usages
    ProgressDialog progressDialog; 6 usages
    AlphaAnimation btnAnimasi = new AlphaAnimation(1f,0.5f); 2 usages

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: 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() { 1 usage
    @Override
    public void onClick(View view) {
        view.startAnimation(btnAnimasi);
        RecyclerView.ViewHolder viewHolder = (RecyclerView.ViewHolder)
            view.getTag();
        int position = viewHolder.getAdapterPosition();
        ModelLayanan mp = list.get(position);
        Intent intent= new Intent( packageContext: LayananActivity.this, LayananEditActivity.class);
        intent.putExtra(Constant.ID, mp.getId());
        intent.putExtra(Constant.TIPE, mp.getTipe());
        intent.putExtra(Constant.HARGA, mp.getHarga());
        startActivity(intent);
    }
};

private void setView() { 1 usage
    db = new SQLiteHelper2( context: this);
    progressDialog = new ProgressDialog( context: this);
    btnLayAdd = (TextView) findViewById(R.id.btnLayAdd);
    rvLayanan =(RecyclerView) findViewById(R.id.rvLayanan);
    list = new ArrayList<>();
    LinearLayoutManager llm = new LinearLayoutManager( context: this);
    llm.setOrientation(LinearLayoutManager.VERTICAL);
    rvLayanan.setHasFixedSize(true);
    rvLayanan.setLayoutManager(llm);
}

```

```

private void showMsg() { 1 usage
    progressDialog.setTitle("Informasi");
    progressDialog.setMessage("Loading Data...");
    progressDialog.setCancelable(false);
    progressDialog.show();
}

private void eventHandling() { 1 usage
    btnLayAdd.setOnClickListener(v -> {
        v.startAnimation(btnAnimasi);
        startActivityForResult(new Intent( packageContext: LayananActivity.this, LayananAddActivity.class));
    });
}

// Override onActivityResult
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 1 && resultCode == RESULT_OK) {
        getData(); // Panggil getData() untuk memperbarui tampilan
    }
}

private void getData () { 3 usages
    list.clear();
    showMsg();
    progressDialog.dismiss();
    try{

```

```

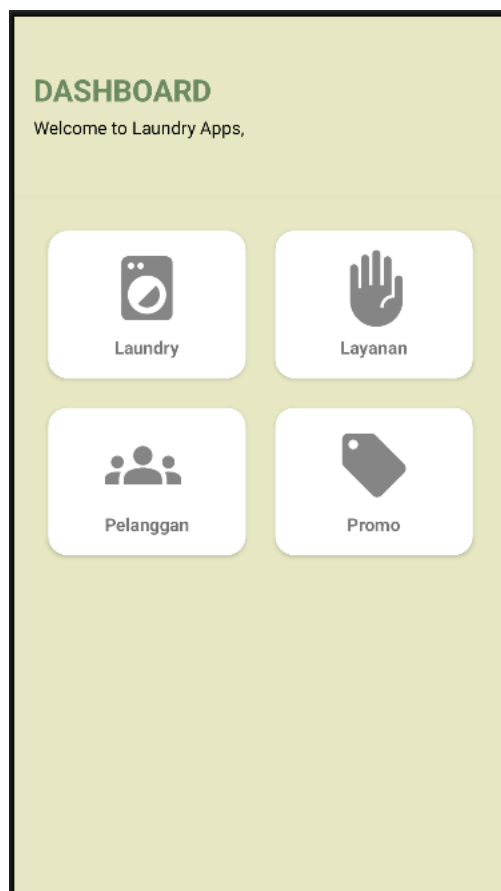
List<ModelLayanan> l = db.getLayanan();
if (l.size() > 0){
    for (ModelLayanan lay : l){
        ModelLayanan ml = new ModelLayanan();
        ml.setId((lay.getId()));
        ml.setTipe((lay.getTipe()));
        ml.setHarga((lay.getHarga()));
        list.add(ml);
    }
    adapterLayanan = new AdapterLayanan( context: this, list);
    adapterLayanan.notifyDataSetChanged();
    rvLayanan.setAdapter(adapterLayanan);
    adapterLayanan.setOnItemClickListener(onClickListener);
}else{
    Toast.makeText( context: this, text: "Data tidak ditemukan", Toast.LENGTH_SHORT).show();
}
}
} catch (Exception e ){
}
}

@Override
protected void onResume() {
    super.onResume();
    getData(); // Memuat ulang data setelah penghapusan atau pembaruan
}

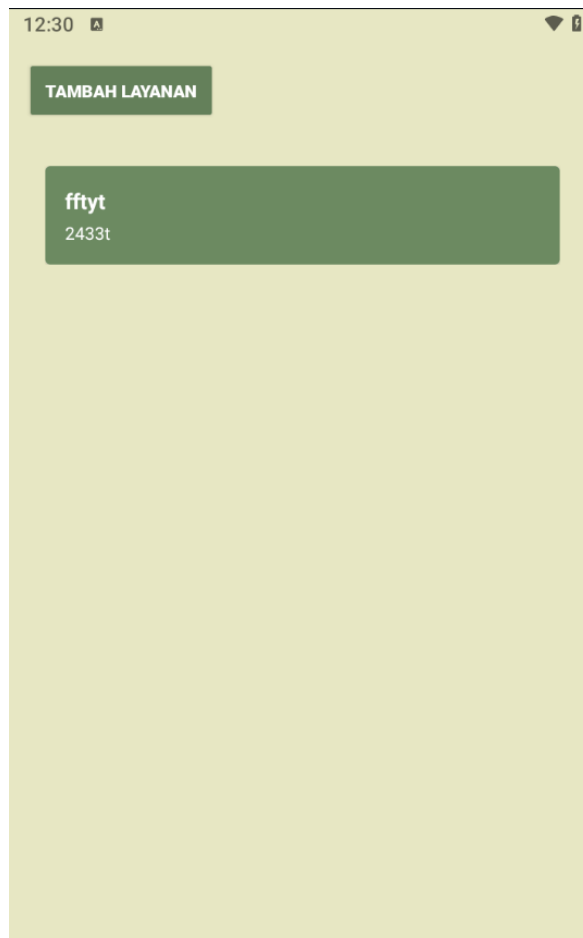
```

- Simulasi

1. Halaman utama setelah login



2. Halaman layanan ditampilkan setelah kita mengklik widget layanan. Pada halaman ini, kita bisa menambahkan layanan dengan mengklik tombol tambah layanan.



3. Selanjutnya kita diarahkan ke Halaman Add Layanan yang memungkinkan kita menambahkan layanan dan harga per kg. Setelah mengisi data, kita bisa menyimpan layanan dengan mengklik tombol simpan, dan akan diarahkan kembali ke halaman LayananActivity.

12:32

## LAYANAN

Layanan

 gosok

Harga

\$ 12000

SIMPAN BATAL

4. Di halaman LayananActivity, data layanan baru akan muncul pada daftar dengan notifikasi sukses.

12:32

TAMBAH LAYANAN

**fftyt**  
2433t

**gosok**  
12000