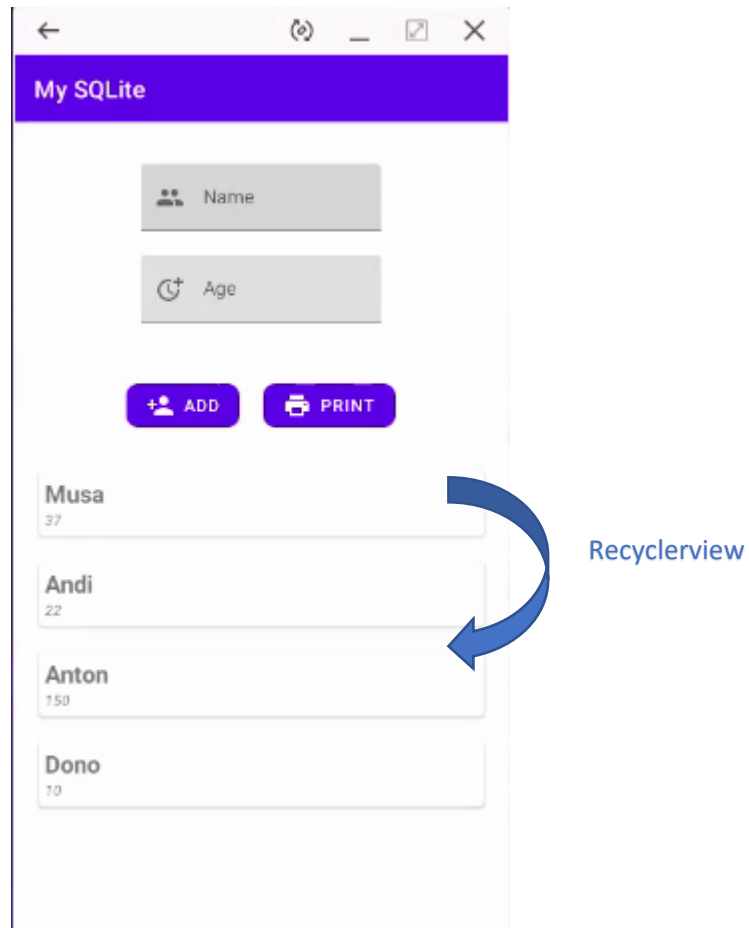


PRAKTIKUM

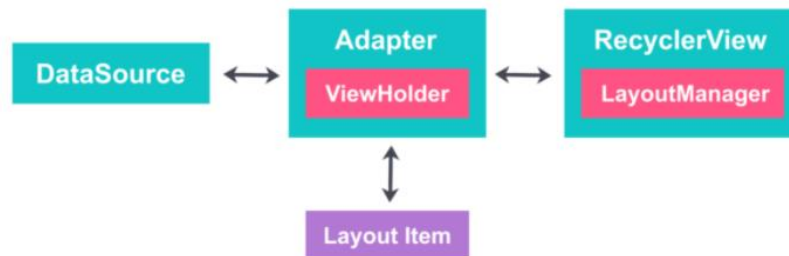
SQLITE & RECYCLERVIEW

Pada praktikum kali ini, kita akan coba membuat aplikasi sederhana dengan menggunakan RecyclerView.

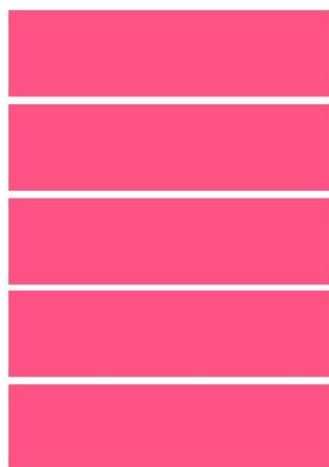


RecyclerView adalah sebuah komponen tampilan (widget) yang lebih canggih ketimbang pendahulunya, yakni ListView. Gambar di bawah menunjukkan perbedaan ListView dan RecyclerView, jika Anda menggunakan ListView objek dari tiap item akan dibuat semuanya dari awal sampai akhir. Sedangkan jika Anda menggunakan RecyclerView, objek yang dibuat hanya sebatas ukuran layar dan beberapa di atas dan di bawahnya saja. Selanjutnya ia menggunakan kembali item yang sudah tidak terlihat.

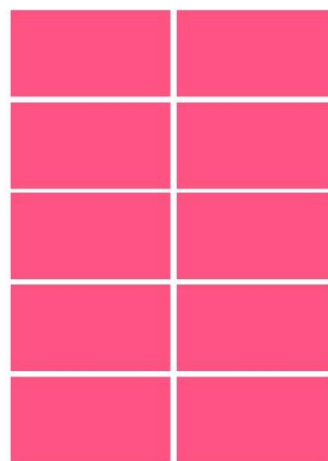
Sebenarnya masalah pada ListView dapat diatasi dengan menggunakan ViewHolder pattern., namun penggunaannya belum diharuskan di ListView. Berbeda dengan RecyclerView yang mewajibkan menggunakan ViewHolder.



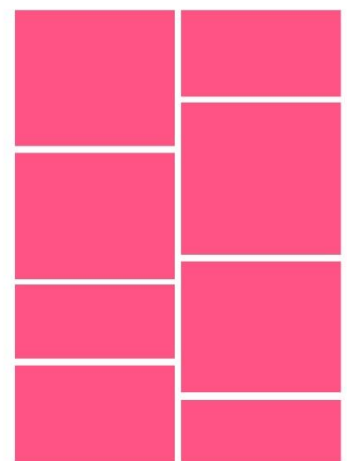
1. **RecyclerView dan LayoutManager:** Komponen antarmuka yang bertugas untuk menampilkan data set yang dimiliki di dalamnya. Layoutmanager akan mengatur posisi tampilan data baik itu secara list (vertikal), grid (baris dan kolom) atau staggered grid (grid yang memiliki susunan tak seragam / tak beraturan).



LinearLayoutManager



GridLayoutManager



StaggeredLayoutManager

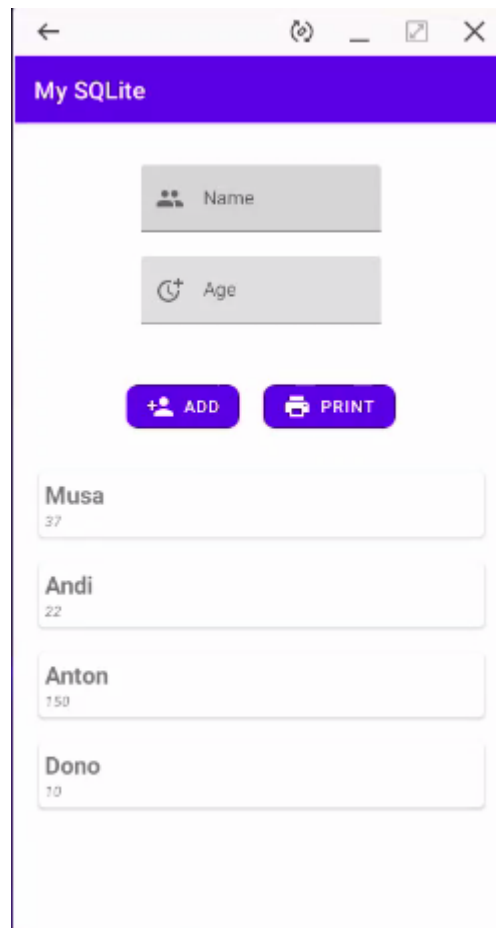
2. **Adapter:** Komponen yang akan mengatur bagaimana menampilkan data set ke dalam **RecyclerView**. Di sinilah terjadi proses pengisian tampilan (**ViewInflate**) dari file layout xml untuk tiap elemen dari data yang sebelumnya terpasang (*bind*) ke dalam **RecyclerView**.
3. **Dataset:** Kumpulan data yang dimiliki dan ingin ditampilkan. Bisa berupa array, list maupun obyek map.

4. **Item Animator:** Ini yang spesial. Kita bisa pasang animasi untuk tiap item di dalamnya. Contoh animasi yang umum seperti penambahan (*add*) dan penghapusan (*removal*) item. Kita akan mempelajari hal ini pada materi terpisah.

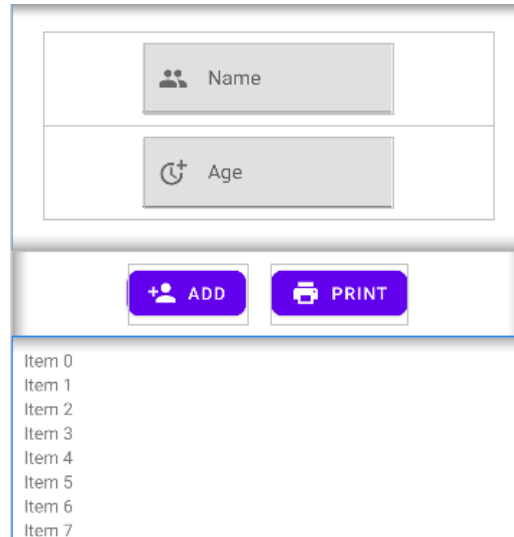
Pada praktikum kali ini kita akan mempelajari cara menampilkan data biodata seperti Nama dan Usia ke dalam sebuah RecyclerView. Beberapa poin yang akan diulas dalam materi ini adalah :

1. Bagaimana menggunakan RecyclerView?
2. Berbagai macam RecyclerView seperti list, grid, dan card.
3. Membuat onClick pada RecyclerView.

Kita akan coba membuat program seperti di bawah ini :



1. Buatlah tampilan seperti di bawah ini, nama layout yang digunakan sebagai contoh, activity_main.xml :



Berikut kodenya :

```
<?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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="25dp">
        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Name"
            android:gravity="center"
            android:padding="10dp"
            android:id="@+id/tiName"
            app:startIconDrawable="@drawable/ic_user"
            app:endIconMode="clear_text">
            <com.google.android.material.textfield.TextInputEditText
                android:layout_width="200dp"
                android:layout_height="wrap_content"
            />
        </com.google.android.material.textfield.TextInputLayout>
        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Age"
        >
```

```

        android:gravity="center"
        android:padding="10dp"
        android:id="@+id/tiAge"
        app:startIconDrawable="@drawable/ic_age"
        app:endIconMode="clear_text">
        <com.google.android.material.textfield.TextInputEditText
            android:layout_width="200dp"
            android:layout_height="wrap_content"
            android:inputType="number"/>
    </com.google.android.material.textfield.TextInputLayout>

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center">
    <com.google.android.material.button.MaterialButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add"
        android:layout_margin="10dp"
        android:id="@+id/btnAdd"
        app:icon="@drawable/ic_add"
        app:cornerRadius="10dp" />
    <com.google.android.material.button.MaterialButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Print"
        android:layout_margin="10dp"
        android:id="@+id/btnPrint"
        app:icon="@drawable/ic_print"
        app:cornerRadius="10dp" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:id="@+id/root_layout">

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/listView"
        android:clickable="true"
        android:padding="10dp"
        android:outlineAmbientShadowColor="@color/purple_200"
        android:outlineSpotShadowColor="@color/purple_200"

        style="@style/TextAppearance.AppCompat.Widget.Button.Colored"/>
</LinearLayout>

</LinearLayout>

```

2. Buatlah *class* dengan nama DBHelper.kt untuk mendeklarasikan database, tabel beserta atributnya. Pada *class* DBHelper, kita dapat mendefinisikan beberapa fungsi untuk mencari, menghapus, menambah, dan mengubah data yang ada di SQLite.

Berikut full code yang ada pada class DBHelper.kt

```
package com.aurumsystem.mysqlite

import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class DBHelper(context: Context, factory:
SQLiteDatabase.CursorFactory?)
    : SQLiteOpenHelper(context, DATABASE_NAME, factory,
DATABASE_VERSION)
{
    //membuat basis data dan tabel
    companion object{
        private const val DATABASE_NAME = "dbData.db"
        private const val DATABASE_VERSION = 1
        private const val TABLE_NAME = "tbUser"
        public const val ID_COL = "id"
        public const val NAME_COL = "name"
        public const val AGE_COL = "age"
        public var getAge = "getage"
    }

    override fun onCreate(db: SQLiteDatabase?) {
        val query = ("CREATE TABLE " +TABLE_NAME+ "(" + ID_COL +
            " INTEGER PRIMARY KEY, " +
            NAME_COL + " TEXT," +
            AGE_COL + " TEXT" + ")")
        db!!.execSQL(query)
    }

    override fun onUpgrade(db: SQLiteDatabase?, p1: Int, p2: Int) {
        db!!.execSQL("DROP TABLE IF EXISTS" + TABLE_NAME)
        onCreate(db)
    }

    //membuat fungsi menambahkan data

    fun addData(name: String?, age: String?) : Boolean{
        val values = ContentValues()
        values.put(NAME_COL, name)
        values.put(AGE_COL, age)
        val db = this.writableDatabase
        var cursor: Cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME+
            " WHERE NAME = '$name'", null)
        if(cursor.moveToFirst()){
            return false
        }
    }
}
```

```

    }
    else{
        db.insert(TABLE_NAME,null, values)
        return true
    }
}

//menambahkan fungsi untuk menampilkan semua data
@SuppressLint("Range")
fun getAllData(): ArrayList<DBModel>{
    val db: SQLiteDatabase = this.readableDatabase
    //val arrayList = ArrayList<String>()
    val arrayList = arrayListOf<DBModel>()
    var Nama: String
    val res: Cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME,
null)
    if (res.moveToFirst()){
        do {
            Nama = res.getString(res.getColumnIndex("name"))
            getAge = res.getString(res.getColumnIndex("age"))
            arrayList.add(DBModel(Nama, getAge))
        }while(res.moveToNext())
    }
    return arrayList
}

//membuat fungsi untuk menghapus data berdasarkan parameter
fun deleteData(Name: String){
    val db: SQLiteDatabase = this.writableDatabase
    db.execSQL("DELETE FROM " + TABLE_NAME + " WHERE name =
'$Name'")
    db.close()
}

//membuat fungsi untuk mencari data berdasarkan parameter
@SuppressLint("Range")
fun searchData(Name:String): ArrayList<DBModel>{
    val db: SQLiteDatabase = this.readableDatabase
    val arrayList = arrayListOf<DBModel>()
    val res: Cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME + "
WHERE " + NAME_COL + " = '$Name' " , null)
    if (res.moveToFirst()){
        do {
            getAge = res.getString(res.getColumnIndex("age"))
        }while(res.moveToNext())
    }
    return arrayList
}

//membuat fungsi untuk mengubah data berdasarkan parameter
fun updateData(Name: String, Age: Int){
    val db: SQLiteDatabase = this.writableDatabase
    db.execSQL("UPDATE " + TABLE_NAME + " SET age = $Age WHERE name
= '$Name'")
}

```

```

        db.close()
    }

    //membuat fungsi untuk mencari data berdasarkan parameter
    @SuppressWarnings("Range")
    fun SearchDataByName(Name:String): ArrayList<DBModel>{
        val db: SQLiteDatabase = this.readableDatabase
        val arrayList = arrayListOf<DBModel>()
        var Nama: String
        var Usia: String
        val res: Cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME + "
WHERE " + NAME_COL + " like '%$Name%' " , null)
        if (res.moveToFirst()){
            do {
                Nama = res.getString(res.getColumnIndex("name"))
                Usia = res.getString(res.getColumnIndex("age"))
                arrayList.add(DBModel(Nama, Usia))

            }while(res.moveToNext())
        }
        return arrayList
    }
}

```

Di sini kita sudah membuat banyak fungsi CRUD yang terdiri dari: membuat basis data dan tabel, serta fungsi CRUD.

- Langkah selanjutnya kita akan membuat tampilan RecyclerView untuk mengatur posisi masing-masing view/komponen. Buatlah layout dengan nama item_data.xml dengan kode sebagai berikut:

```

<?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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="horizontal"
    android:padding="25dp"
    app:cardCornerRadius="5dp"
    android:layout_margin="10dp"
    android:clickable="true"
    android:foreground="?attr/selectableItemBackground"
    android:foregroundTint="@color/purple_200"
    android:focusable="true"
    android:id="@+id/cvData">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"

```



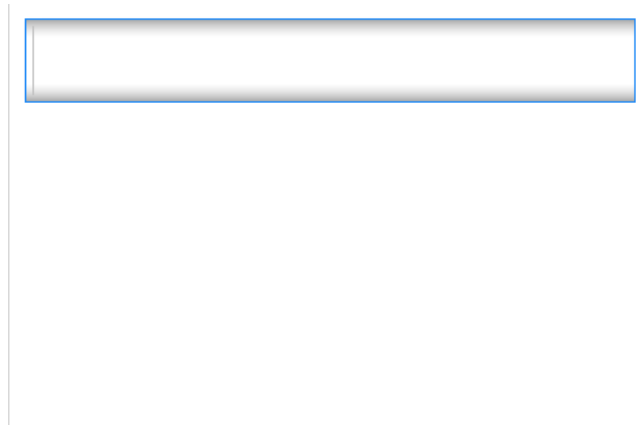
```

        android:padding="5dp"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tvNama"
        android:textSize="20sp"
        android:textStyle="bold"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/tvAge"
        android:textSize="12sp"
        android:textStyle="italic"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

```



- Langkah selanjutnya kita akan coba membuat model data, buatlah class dengan nama DBModel.kt.

```

package com.aurumsystem.mysqlite

data class DBModel (var Nama: String, var Usia: String)

```

- Langkah berikutnya adalah membuat *Adapter* yang mengatur koneksi ke dalam basisdata serta tabel dengan interface kita, nama file : DBAdapter.kt

```

package com.aurumsystem.mysqlite

import android.content.ComponentCallbacks
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView

```

```

import androidx.cardview.widget.CardView
import androidx.recyclerview.widget.RecyclerView

class DBAdapter(private val
listData:ArrayList<DBModel>):RecyclerView.Adapter<DBAdapter.CardViewHol
der>() {
    private lateinit var onItemClickCallback: OnItemClickCallback
    interface OnItemClickCallback {
        fun onItemClick(data: DBModel)
    }
    fun setOnItemClickCallback(onItemClickCallback:
OnItemClickCallback) {
        this.onItemClickCallback = onItemClickCallback
    }
    inner class
CardViewHolder(itemV:View):RecyclerView.ViewHolder(itemV) {
        var Nama: TextView = itemV.findViewById(R.id.tvNama)
        var Usia: TextView = itemV.findViewById(R.id.tvAge)
        var cvData: CardView = itemV.findViewById(R.id.cvData)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
DBAdapter.CardViewHolder {
        val view:View =
LayoutInflater.from(parent.context).inflate(R.layout.item_data,
parent,false)
        return CardViewHolder(view)
    }

    override fun onBindViewHolder(holder: DBAdapter.CardViewHolder,
position: Int) {
        val data = listData[position]
        holder.Nama.text = data.Nama
        holder.Usia.text = data.Usia
        holder.itemView.setOnClickListener {
onItemClickCallback.onItemClick(listData[holder.adapterPosition])
        }
    }

    override fun getItemCount(): Int {
        return listData.size
    }
}

```

6. Buatlah tampilan untuk proses update data. Skenarionya adalah ketika RecyclerView tersebut di klik, maka akan membawa data berupa Nama dan Usia ke Activity berikutnya (activity_update.xml) dengan kode berikut :

```

<?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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:orientation="vertical">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="25dp">
    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:gravity="center"
        android:padding="10dp"
        android:id="@+id/tiName"
        app:startIconDrawable="@drawable/ic_user">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="200dp"
        android:layout_height="wrap_content"/>
    </com.google.android.material.textfield.TextInputLayout>
    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Age"
        android:gravity="center"
        android:padding="10dp"
        android:id="@+id/tiAge"
        app:startIconDrawable="@drawable/ic_age">

    <com.google.android.material.textfield.TextInputEditText
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:inputType="number"/>
    </com.google.android.material.textfield.TextInputLayout>

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center">
    <com.google.android.material.button.MaterialButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Update"
        android:layout_margin="10dp"
        android:id="@+id/btnUpdate"
        app:icon="@drawable/ic_update"
        app:cornerRadius="10dp" />
    <com.google.android.material.button.MaterialButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

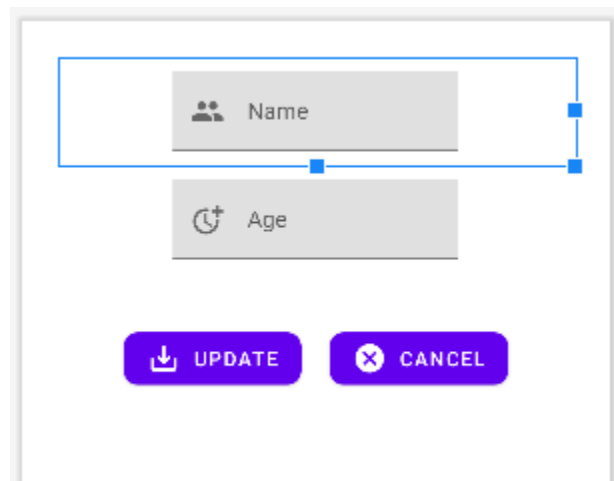
```

```

        android:text="Cancel"
        android:layout_margin="10dp"
        android:id="@+id/btnCancel"
        app:icon="@drawable/ic_cancel"
        app:cornerRadius="10dp" />
    </LinearLayout>

</LinearLayout>

```



7. Pada bagian **MainActivity.kt**, tuliskan kode berikut untuk menghubungkan antara interface, controller, dan basisdata dengan kode berikut :

```

package com.aurumsystem.mysqlite

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.text.Editable
import android.text.TextWatcher
import android.view.Display
import android.widget.*
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.android.material.dialog.MaterialAlertDialogBuilder
import com.google.android.material.textfield.TextInputLayout

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val btnAdd:Button = findViewById(R.id.btnAdd)
        val btnPrint:Button = findViewById(R.id.btnPrint)
        val tiName: TextInputLayout = findViewById(R.id.tiName)
        val tiAge: TextInputLayout = findViewById(R.id.tiAge)
        var name = tiName.editText?.text.toString()
        var age = tiAge.editText?.text.toString()
    }
}

```

```

val listView: RecyclerView = findViewById(R.id.listView)
val db = DBHelper(this, null)

var arrayList: ArrayList<DBModel> = arrayListOf()
listView.setHasFixedSize(true)
arrayList.addAll(db.getAllData())
listView.layoutManager = LinearLayoutManager(this)
var CardData = DBAdapter(arrayList)
listView.adapter = CardData

fun Display() {
    arrayList.clear()
    arrayList.addAll(db.getAllData())

    listView.invalidate()
    listView.refreshDrawableState()
    listView.adapter = CardData
    Toast.makeText(this, "Data displayed",
    Toast.LENGTH_SHORT).show()
}

fun showSelectedData(data: DBModel) {
    //Toast.makeText(this, "Kamu memilih " + data>Nama,
    Toast.LENGTH_SHORT).show()
    var params: String = data>Nama
    db.SearchData(params)
    val context = this;
    MaterialAlertDialogBuilder(context).apply{
        setTitle("Data : $params, Age : ${DBHelper.getAge}")
        setIcon(R.drawable.ic_user)
        setMessage("What do you want to do?")
        setPositiveButton("Delete") {_, _ ->

            db.deleteData(params)
            Toast.makeText(this@MainActivity, "Data has been
deleted", Toast.LENGTH_SHORT).show()
            Display()
        }
        setNegativeButton("Update") {_, _ ->

            val moveIntent = Intent(this@MainActivity,
UpdateActivity::class.java).apply {
                putExtra(UpdateActivity>Nama, params)
                putExtra(UpdateActivity>Age, DBHelper.getAge)
            }

            //moveIntent.putExtra(UpdateActivity>Age,
tiAge.editText?.text?.toString())
            startActivity(moveIntent)
            finish()
        }
        setNeutralButton("Cancel") {_, _ ->

        }
    }.create().show()
}

```

```

    }

    CardData.setOnItemClickCallback(object :
DBAdapter.OnItemClickCallback{
        override fun onItemClick(data: DBModel) {
            showSelectedData(data)
        }
    })

    tiName.editText!!.addTextChangedListener(object : TextWatcher{
        override fun beforeTextChanged(p0: CharSequence?, p1: Int,
p2: Int, p3: Int) {

        }

        override fun onTextChanged(p0: CharSequence?, p1: Int, p2:
Int, p3: Int) {
            if(tiName.editText?.text?.isBlank() == true){
                Display()
            }
            if(tiName.editText?.text?.isNotBlank() == true)
            {
                val arrlist:ArrayList<String> =
db.SearchDataByName(tiName.editText?.text?.toString()).toString() as
ArrayList<String>
                val arrayAdapter: ArrayAdapter<*> =
ArrayAdapter<Any?>(this@MainActivity,
                    android.R.layout.simple_list_item_1, arrlist as
List<Any?>)
                arrayList.clear()

arrayList.addAll(db.SearchDataByName(tiName.editText?.text?.toString()).
toString()))

                arrayAdapter.notifyDataSetChanged()
                listView.invalidate()
                listView.refreshDrawableState()
                listView.adapter = CardData
                //Toast.makeText(this@MainActivity, "Data found",
Toast.LENGTH_SHORT).show()
            }

        }

        override fun afterTextChanged(p0: Editable?) {

        }

    })

    btnPrint.setOnClickListener {
        Display()
    }
    btnAdd.setOnClickListener {
        if(tiName.editText?.text?.isNotEmpty() == true &&
tiAge.editText?.text?.isNotEmpty() == true){

```

```

        if (db.addData(tiName.editText?.text.toString(),
tiAge.editText?.text.toString())) {
            Toast.makeText(this, "Inserted",
Toast.LENGTH_SHORT).show()
            Display()
        }
        else
        {
            Toast.makeText(this, "Not Inserted",
Toast.LENGTH_SHORT).show()
        }
    }
    else{
        Toast.makeText(this, "Cannot left blank",
Toast.LENGTH_SHORT).show()
    }
}

}

}

```

8. Pada class **UpdateActivity.kt**, tuliskan kode berikut untuk memanggil fungsi perubahan data yang ada di **DBHelper.kt**

UpdateActivity.kt

```

package com.aurumsystem.mysqlite

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.google.android.material.button.MaterialButton
import com.google.android.material.textfield.TextInputLayout

class UpdateActivity : AppCompatActivity() {
    companion object{
        const val Nama = "nama"
        const val Age = "age"
    }
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_update)
        val btnCancel: MaterialButton = findViewById(R.id.btnCancel)
        val tiName: TextInputLayout = findViewById(R.id.tiName)
        val tiAge: TextInputLayout = findViewById(R.id.tiAge)
        val btnUpdate: MaterialButton = findViewById(R.id.btnUpdate)
        val nama = intent.getStringExtra(Nama);
        tiName.editText?.setText(nama)
        val db = DBHelper(this, null)

        val age = intent.getStringExtra(Age);
        tiAge.editText?.setText(age)
    }
}

```

```

        btnCancel.setOnClickListener {
            finish()
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
        }
        btnUpdate.setOnClickListener {
            db.updateData(tiName.editText?.text?.trim().toString(),
tiAge.editText?.text?.trim().toString().toInt())

            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
            finish()
        }
    }
}

```

Jalankan aplikasi Anda.

Source code dapat dilihat di Elearning.