

# CREATE READ UPDATE DELETE KOTLIN + SQLITE

## ANDROID STUDIO

### 1. Pendahuluan

#### a. CREATE

Dengan fungsi Create, pengguna dapat menambahkan catatan atau entri baru ke database. Fungsi ini meliputi memasukkan data ke dalam database, membuat baris baru dalam tabel, dan menentukan nilai untuk atribut (kolom) dari baris itu.

#### b. READ

Fungsi Read membantu pengguna untuk mengambil atau mengakses data yang ada dari database, seperti saat mencari catatan tertentu atau memperoleh semua catatan yang memenuhi kriteria tertentu.

#### c. UPDATE

Update memungkinkan pengguna untuk mengubah atau mengedit data yang ada di database.

#### d. DELETE

Fungsi Delete memberi akses ke pengguna untuk menghapus data yang tidak diinginkan dari database.

#### e. VIEW BINDING

View Binding adalah sebuah fitur dari android studio yang memudahkan penulisan code berinteraksi dengan sebuah view. View Binding menggantikan code "findViewById".

Untuk mengaktifkan fitur view binding, kita harus melakukan inisiasi view binding di modul-level **build.gradle**.

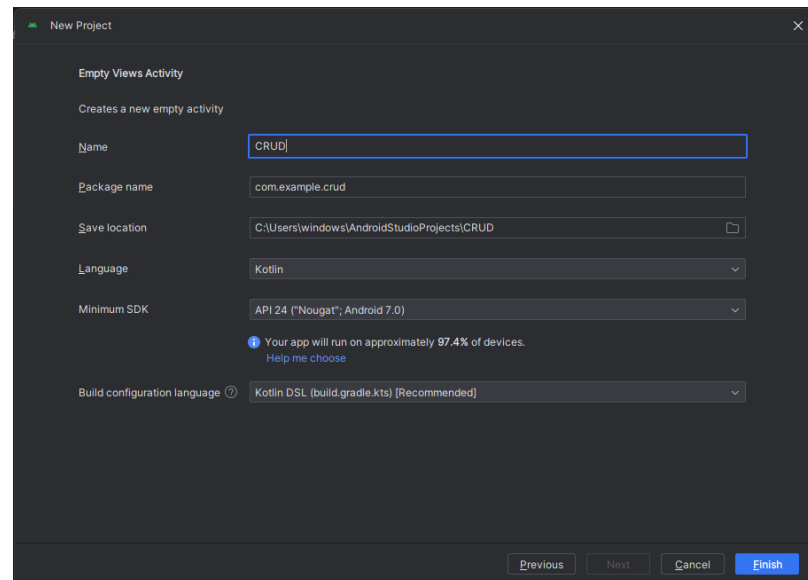
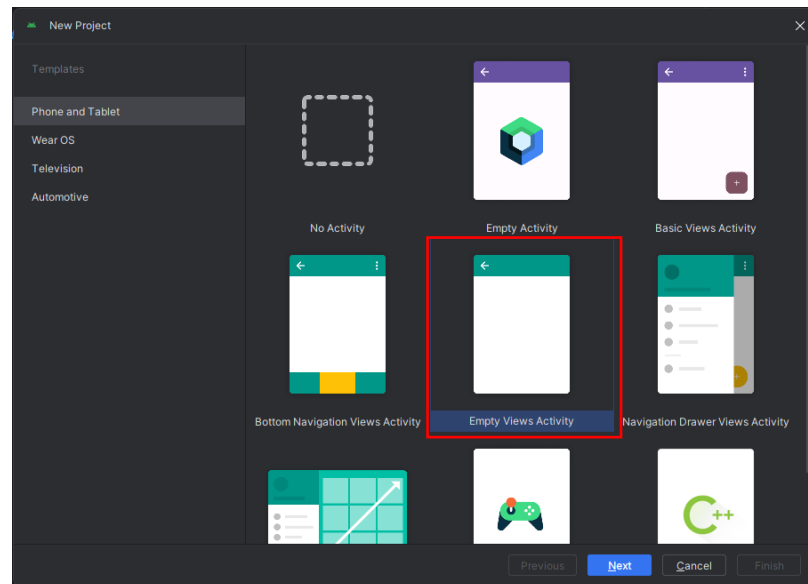
```
Android {  
    .....  
    buildFeatures(){  
        viewBinding = true  
    }  
}
```

### 2. CREATE

Membuat sebuah aplikasi Data Mahasiswa yang akan mengimplementasi fitur-fitur CRUD SQLite + View Binding.

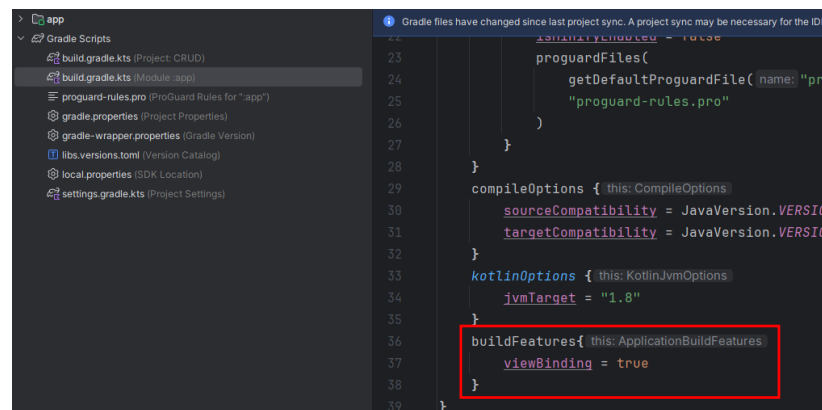
#### a. Membuat New Project di Android Studio

## Pilih Empty Views Activity (Non Jetpack)



### b. Setting Awal Project

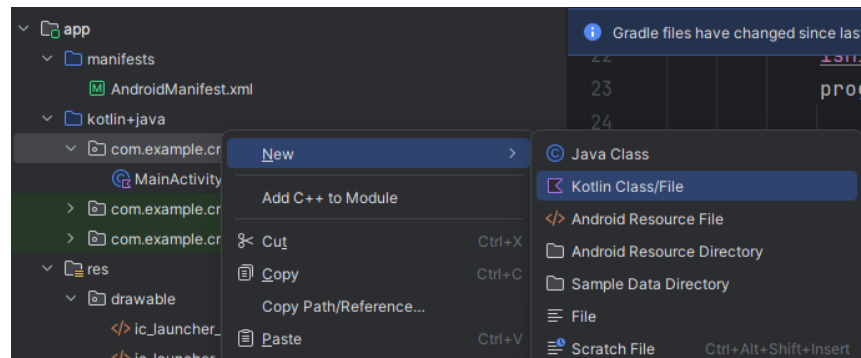
Di file `build.gradle` (Module :app) menambahkan code untuk View Binding



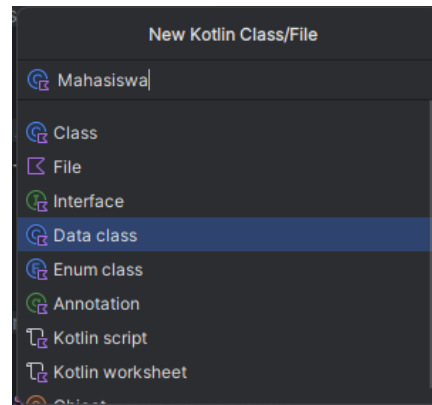
Setelah melakukan inisiasi view binding, klik Sync Now untuk melakukan sinkronisasi dari fitur View Binding.

Sync Now Ignore these changes

### c. Membuat Data Class Mahasiswa



Pilih Data Class lalu enter

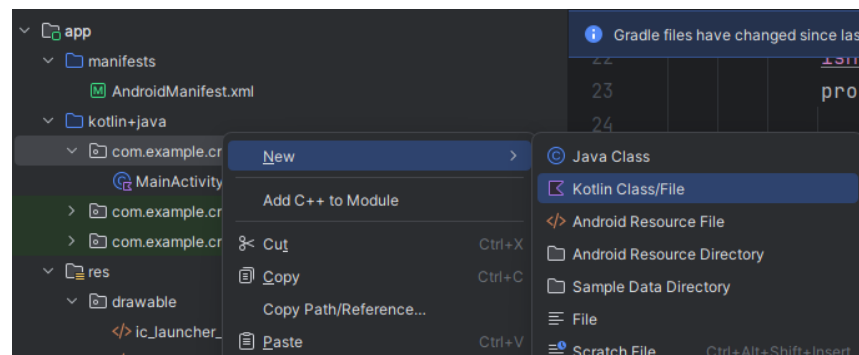


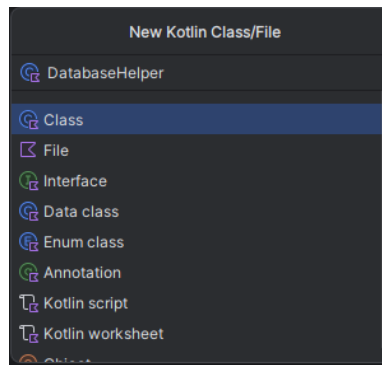
Di dalam data class Mahasiswa kita akan membuat 2 data yaitu nim dengan tipe data String dan nama dengan tipe data String.

```
package com.example.crud

data class Mahasiswa(
    val nim: String,
    val nama: String
)
```

### d. Membuat Database Helper





DatabaseHelper berfungsi untuk melakukan inisiasi koneksi database.  
 Tambahkan kode

```
class DatabaseHelper(context: Context): SQLiteOpenHelper(context,
    DATABASE_NAME, null, DATABASE_VERSION)
```

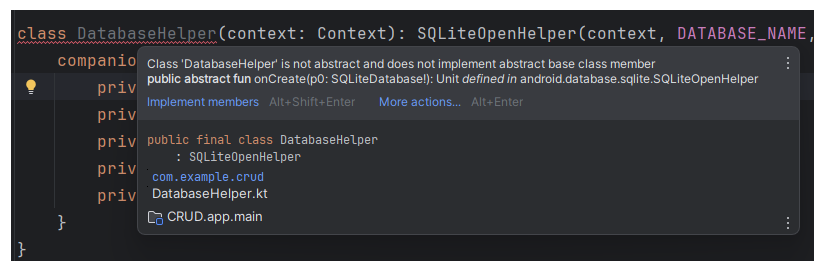
DATABASE\_NAME dan DATABASE\_VERSION error karena kita belum melakukan deklarasi variable tersebut.

Deklarasikan variable tersebut di dalam object

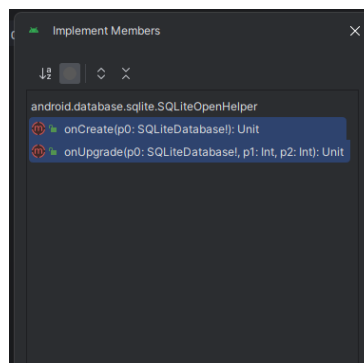
<pre>companion object{     private const val DATABASE_NAME = "mahasiswa.db"     private const val DATABASE_VERSION = 1     private const val TABLE_NAME = "mahasiswa"     private const val COLUMN_NIM = "nim"     private const val COLUMN_NAMA = "nama" }</pre>	Deklarasi nama DB Deklarasi DB version Deklarasi nama tabel Deklarasi kolom 1 Deklarasi kolom 2
---	---

DatabaseHelper tetap merah karena kita harus menambahkan 2 function bawaan.

Klik kursor ke DatabaseHelper dan akan muncul sebuah popup.



Klik Impelent members dan akan muncul Jendela Baru



Lalu klik OK.

Akan muncul 2 function baru.

```

override fun onCreate(p0: SQLiteDatabase?) {
    TODO(reason: "Not yet implemented")
}

override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {
    TODO(reason: "Not yet implemented")
}

```

Ubah function onCreate menjadi

```

override fun onCreate(db: SQLiteDatabase?) {
    val QcreateTable = "CREATE TABLE $TABLE_NAME ($COLUMN_NIM INTEGER PRIMARY KEY, $COLUMN_NAMA TEXT)"
    db?.execSQL(QcreateTable)
}

```

QcreateTable adalah value yang berisikan sebuah Query untuk membuat sebuah table dengan menggunakan variable \$TABLE\_NAME, \$COLUMN\_NIM dan \$COLUMN\_NAMA.

Selanjutnya query tersebut dieksekusi.

Ubah function onUpgrade menjadi

```

override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
    {
        val QdropTable = "DROP TABLE IF EXIST $TABLE_NAME"
        db?.execSQL(QdropTable)
        onCreate(db)
    }
}

```

QdropTable adalah value yang bersihkan sebuah Query untuk menghapus table jika table sudah ada sebelumnya, lalu selanjutnya query dieksekusi.

Menambahkan sebuah function untuk memasukan data Mahasiswa.

<pre> fun InsertMahasiswa(mahasiswa: Mahasiswa){      val db = writableDatabase     val dataMahasiwa = ContentValues().apply {         put(COLUMN_NIM, mahasiswa.nim)         put(COLUMN_NAMA, mahasiswa.nama)     }     db.insert(TABLE_NAME, null, dataMahasiwa)     db.close() } </pre>	Function Mahasiswa yang bersihkan data class Mahasiswa
	Deklarasi db
	Deklarasi dataMahasiswa yang berisikan data class Mahasiswa berupa nim dan nama
	Dieksekusi dengan table tujuan dengan dataMahasiswa

e. Design

Buat sebuah palet warna baru (dapat disesuaikan dengan keinginan)

res-value-colors.xml

```

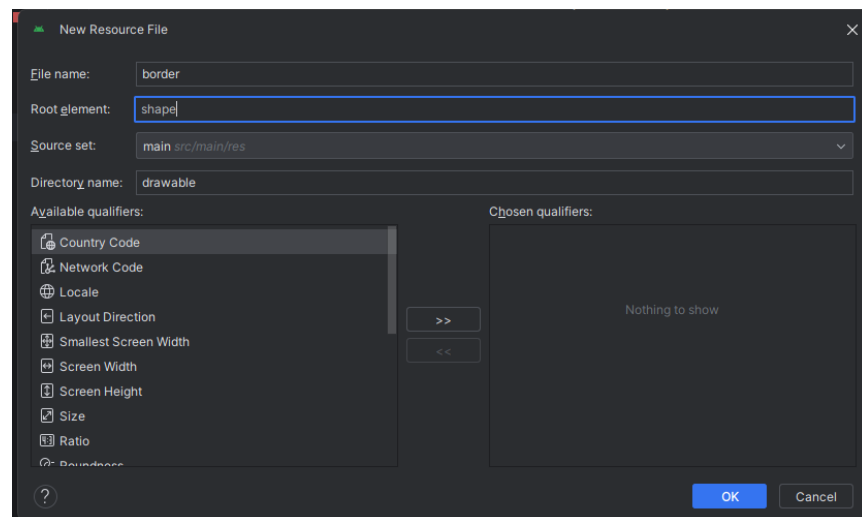
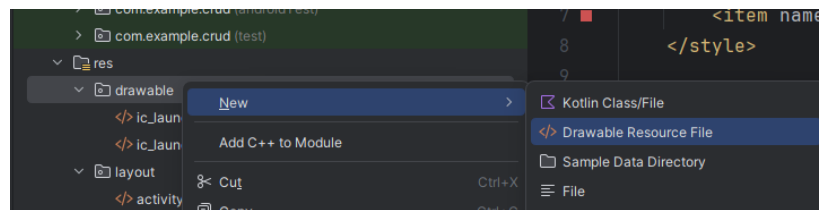
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="black">#FF000000</color>
4      <color name="white">#FFFFFFF</color>
5      <color name="softred">#CC5252</color>
6  </resources>

```

## res-value-themes

```
1 <resources xmlns:tools="http://schemas.android.com/tools">
2 <!-- Base application theme. -->
3 <style name="Base.Theme.CRUD" parent="Theme.Material3.DayNight.NoActionBar">
4 <!-- Customize your light theme here. -->
5 <item name="colorPrimary">@color/softred</item>
6 <item name="colorPrimaryVariant">@color/softred</item>
7 <item name="android:statusBarColor">@color/softred</item>
8 </style>
9
10 <style name="Theme.CRUD" parent="Base.Theme.CRUD" />
11 </resources>
```

Membuat sebuah shape border agar tampilan lebih menarik.

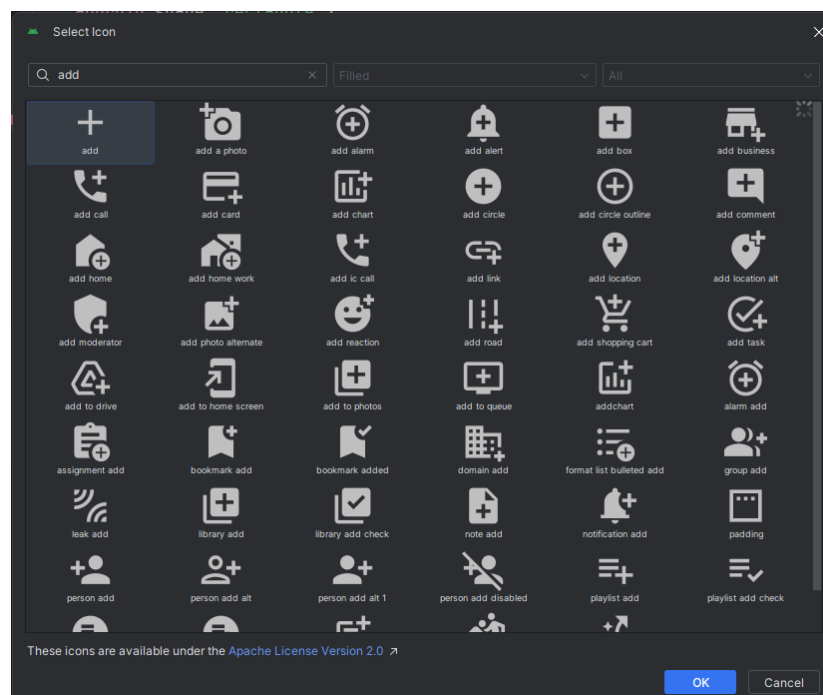
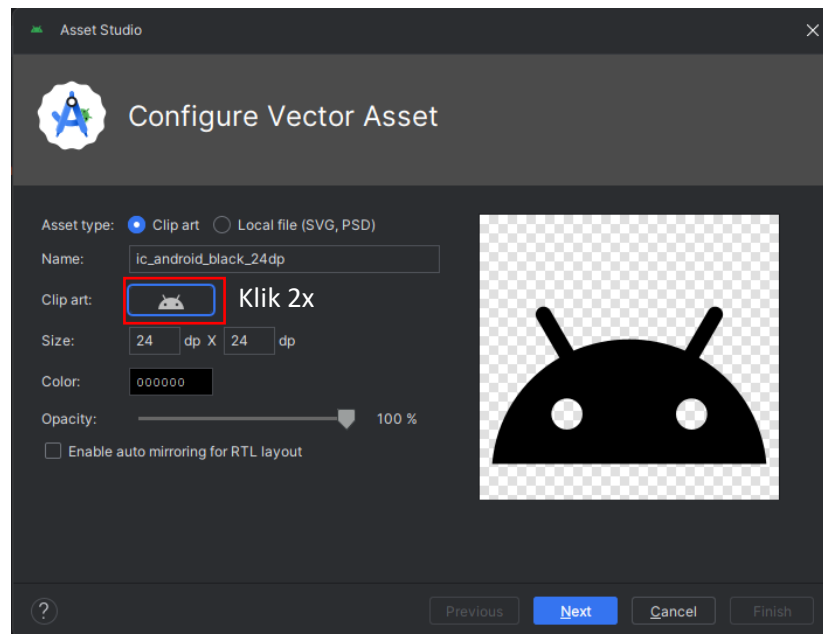
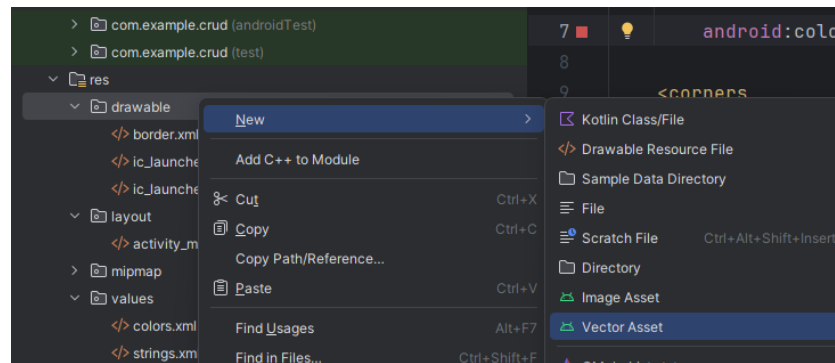


```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <stroke
        android:width="1dp"
        android:color="@color/softred"/>
    <corners
        android:radius="8dp"/>
</shape>
```



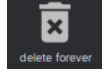
Shape yang dibuat



Tambahkan sebuah vector asset di `res-drawable`. Disini kita akan menambahkan icon add, save, edit dan delete.



Klik OK, next dan finish. Lakukan tahap ini beberapa kali untuk mendapatkan

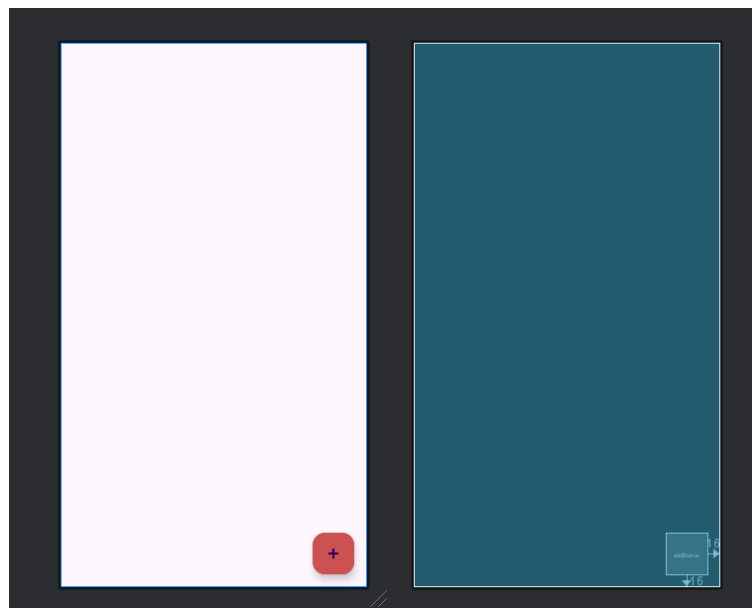
icon save , edit  dan delete . (Bisa menyesuaikan dengan selera masing-masing).

Selanjutnya design `activity_main.xml` untuk menambahkan komponen. Ubah `constraintLayout` menjadi `RelativeLayout` dan tambahkan `floatingActionButton`.

```
<RelativeLayout
    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=".MainActivity">

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/addButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:layout_marginBottom="16dp"
        android:layout_marginTop="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"
        android:src="@drawable/baseline_add_24"
        android:tint="@color/white"
        android:backgroundTint="@color/softred"
    />
</RelativeLayout>
```

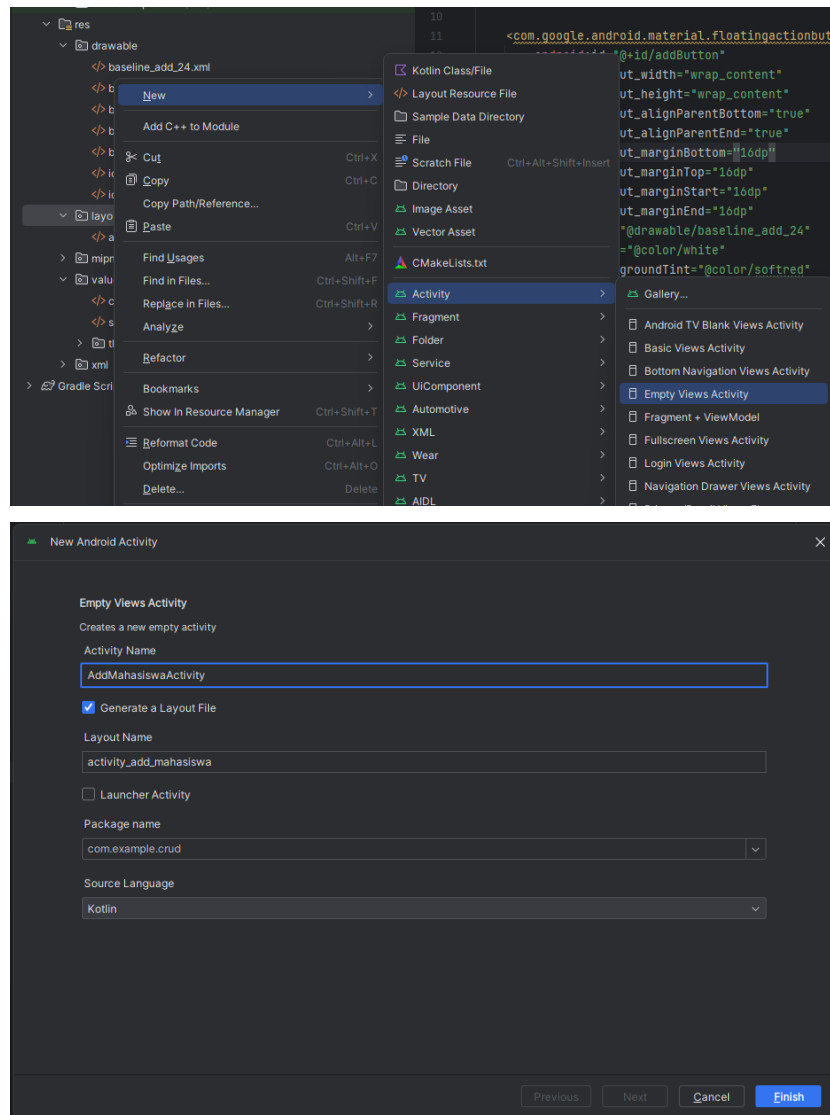
Sehingga tampilan menjadi seperti :



Selanjutnya membuat sebuah tampilan baru untuk melakukan input data mahasiswa.



Buat new empty views activity



Klik finish.

Secara otomatis akan membuat 2 file, yaitu activity dan layout.

Design `activity_add_mahasiswa.xml`

Ubah `constraintLayout` menjadi `RelativeLayout`.

Tambahkan sebuah `TextView`

<pre> &lt;TextView     android:id="@+id/ addMahasiswaHeading"     android:layout_width="wrap_content"     android:layout_height="wrap_content"     android:layout_alignParentTop="true"     android:layout_marginTop="16dp"     android:text="Tambah Masiswa"     android:textColor="@color/softred"     android:textSize="24sp"         /&gt;                     </pre>	<p>ID TextView</p> <p>Membuat TextView berada TOP dengan jarak 16dp Extract string resource Palete warna</p>
---	--

## Tambahkan sebuah ImageView

<pre>&lt;ImageView     android:id="@+id/saveButton"     android:layout_width="wrap_content"     android:layout_height="wrap_content"     android:src="@drawable/baseline_save_alt_24"     android:layout_alignParentEnd="true"     android:layout_marginEnd="6dp"     android:layout_alignTop="@id/addMahasiswaHeading" /&gt;</pre>	<p>ID ImageView</p> <p>Source icon vector Membuat berada di kanan dengan margin 6dp Berata sejajar dengan TextView menggunakan ID</p>
---	---

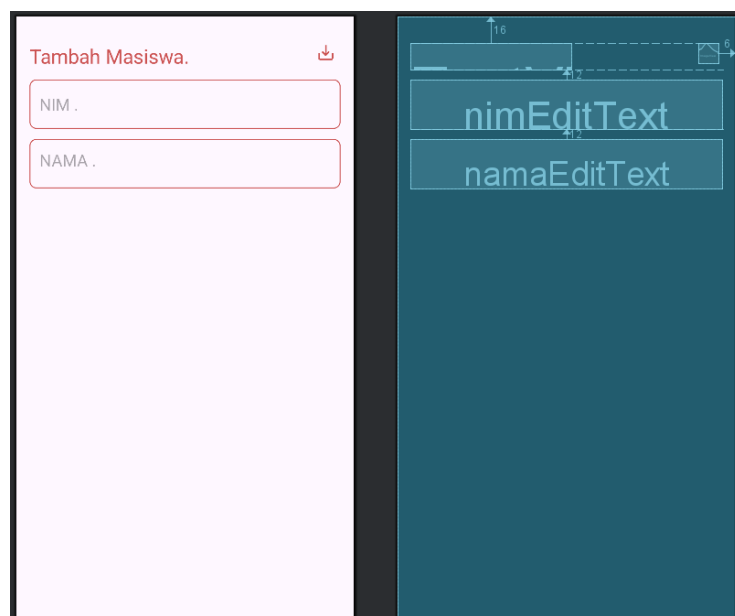
## Tambahkan sebuah EditText untuk menginput NIM

<pre>&lt;EditText     android:id="@+id/nimEditText"     android:layout_width="match_parent"     android:layout_height="60dp"     android:hint="NIM ."     android:padding="12dp"     android:textSize="20sp"     android:layout_below="@id/addMahasiswaHeading"     android:background="@drawable/border"     android:maxLines="1"     android:layout_marginTop="6dp" /&gt;</pre>	<p>ID EditText</p> <p>Tinggi 60dp</p> <p>Ukuran text Posisi editText Menggunakan border</p> <p>Margin atas agar memiliki jarak dengan TextView</p>
---	--

## Tambahkan sebuah EditText untuk menginput NAMA

<pre>&lt;EditText     android:id="@+id/namaEditText"     android:layout_width="match_parent"     android:layout_height="60dp"     android:hint="NAMA ."     android:gravity="top"     android:padding="12dp"     android:textSize="20sp"     android:background="@drawable/red_border"     android:layout_below="@id/namaEditText"     android:layout_marginTop="12dp" /&gt;</pre>	<p>ID EditText</p> <p>Tinggi 60dp</p> <p>Ukuran text Menggunakan border Posisi EditTextnama Margin atas agar memiliki jarak dengan EditText NIM</p>
--	---

Maka tampilan `activity_add_mahasiswa.xml` akan menjadi seperti :



f. Logic Menambahkan Data Mahasiswa

Masuk ke class `AddMahasiswaActivity` dan deklarasikan fitur ViewBinding dan database dengan menambahkan code :

```
private lateinit var binding: ActivityAddMahasiswaBinding
private lateinit var db: DatabaseHelper
```

Di dalam function `onCreate` panggil ViewBinding yang sudah kita deklarasikan dan ubah `setContentView(R.layout.activity_add_mahasiswa)` menjadi `setContentView(binding.root)`  
Lalu panggil database yang sudah dideklarasikan.

```
class AddMahasiswaActivity : AppCompatActivity() {
    .....

    override fun onCreate(savedInstanceState: Bundle?) {
        binding = ActivityAddMahasiswaBinding.inflate(layoutInflater)
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(binding.root)
        db = DatabaseHelper(this)
        .....
    }
}
```

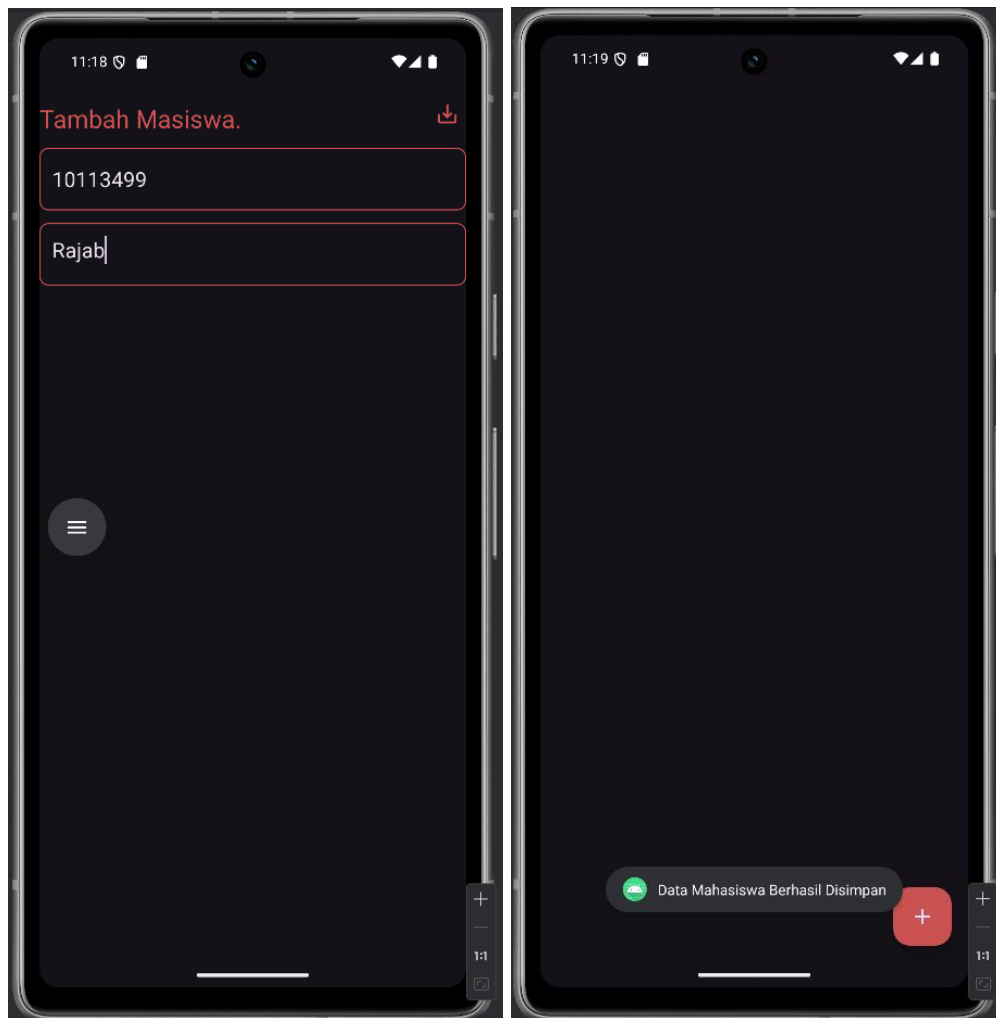
Set button untuk melakukan aksi save data di dalam class `onCreate`

<pre>binding.saveButton.setOnClickListener{     val nim = binding.nimEditText.text.toString()     val nama = binding.namaEditText.text.toString()     val mahasiswa = Mahasiswa(nim, nama)      db.InsertMahasiswa(mahasiswa)      finish()     Toast.makeText(this, "Data Mahasiswa Berhasil     Disimpan", Toast.LENGTH_SHORT).show() }</pre>	<p>Logic jika tombol <code>saveButton</code> di klik</p> <p>Memanggil value dari inputan NIM</p> <p>Memanggil value dari inputan NAMA</p> <p>Memasukan nilai value NIM dan NAMA ke data class <code>Mahasiswa</code></p> <p>Menggunakan function <code>InsertMahasiswa</code> yang berisikan data class <code>mahasiswa</code></p> <p>Memberikan notifikasi jika data berhasil disimpan</p>
---	---

Lalu masuk ke class `ActivityMain` untuk melakukan set `floatingActionButton` yang ada di `activity_main.xml` untuk redirect ke layout `activity_add_mahasiswa.xml`  
Jangan lupa deklarasikan ViewBinding.

<pre>class MainActivity : AppCompatActivity() {     private lateinit var binding: ActivityMainBinding      override fun onCreate(savedInstanceState: Bundle?) {         binding = ActivityMainBinding.inflate(layoutInflater)         super.onCreate(savedInstanceState)         enableEdgeToEdge()         setContentView(binding.root)          binding.addButton.setOnClickListener{             val TambahMahasiswa = Intent(this, AddMahasiswaActivity::class.java)             startActivity(TambahMahasiswa)         }          .....     } }</pre>	<p>Logic jika tombol <code>addButton</code> di klik akan redirect ke <code>ActivityAddMahasiswaActivity</code></p>
--	--

### Tampilan akhir Modul CREATE



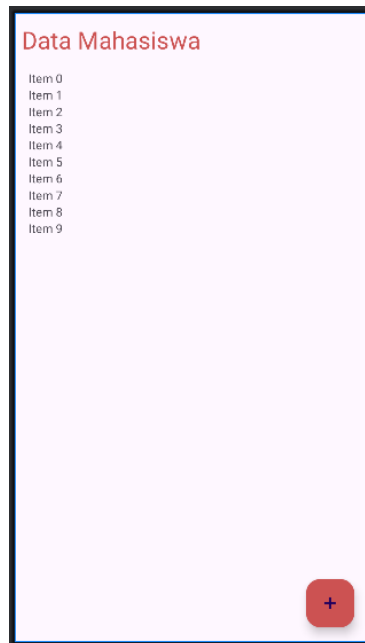
## 3. READ

### a. DESIGN

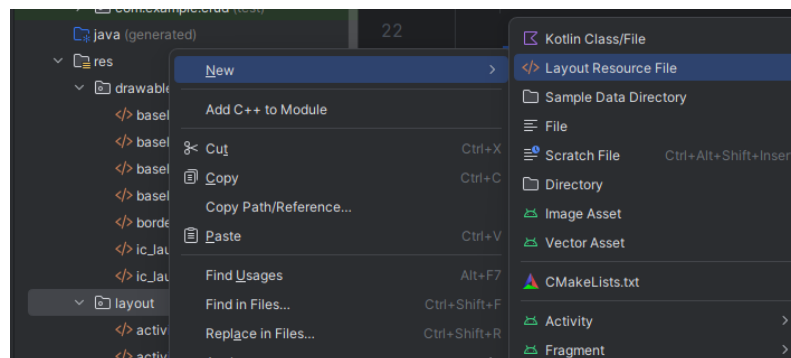
Tambahkan sebuah TextView dan RecyclerView di `activity_main.xml`

```
<TextView
    android:id="@+id/Heading"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:text="@string/_heading_data_mahasiswa"
    android:textSize="28sp"
    android:textColor="@color/softred"
    android:layout_marginStart="8dp"
    android:layout_marginTop="12dp"
/>
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/mahasiswaRecyclerView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/Heading"
    android:padding="16dp"
/>
```

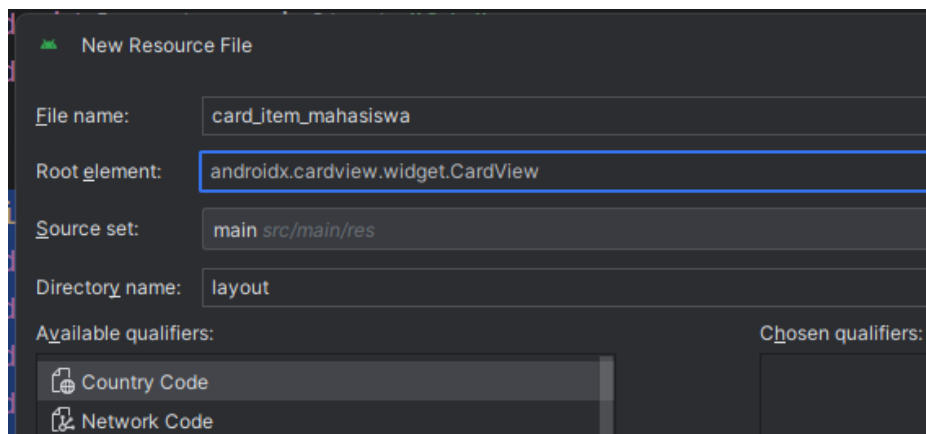
Dengan hasilnya seperti di bawah



Buat sebuah cardView dengan cara klik kanan di Layout – New – Layout Resource File



Beri nama `card_item_mahasiswa` dan root element diganti menjadi `CardView` lalu klik OK



## Tambahkan beberapa code di layout cardView

```
<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"
    app:cardCornerRadius="8dp"
    app:cardElevation="4dp"
    app:cardUseCompatPadding="true"
/>
```

Di dalam cardView tambahkan LinerLayout untuk menampilkan 2 buah TextView yang akan digunakan untuk menampilkan data nim dan nama

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">
    <TextView
        android:id="@+id/nimTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/_nimTV"
        android:textColor="@color/softred"
        android:textSize="22dp"
        android:textStyle="bold"
    />
    <TextView
        android:id="@+id/namaTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/_namaTV"
        android:textColor="@color/softred"
        android:textSize="18dp"
    />
</LinearLayout>
```

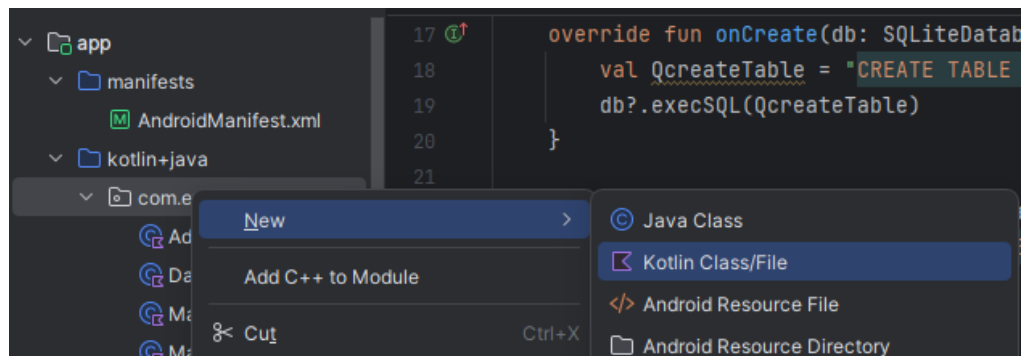
### b. Logic Read

Masuk ke **DatabaseHelper** dan tambahkan sebuah function yang berfungsi untuk mengambil semua data yang ada di tabel mahasiswa

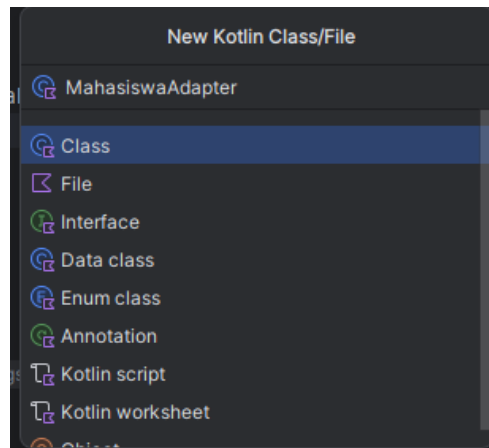
<pre>fun getAllMahasiswa(): List&lt;Mahasiswa&gt;{     val mahasiswaList = mutableListOf&lt;Mahasiswa&gt;()     val db = readableDatabase     val QgetAllData = "SELECT * FROM \$TABLE_NAME"     val cursor = db.rawQuery(QgetAllData, null)      while (cursor.moveToNext()){          val nim = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NIM))         val nama = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NAMA))         val datamahasiswa = Mahasiswa(nim, nama)         mahasiswaList.add(datamahasiswa)     }     cursor.close()     db.close()     return mahasiswaList }</pre>	<p>Function</p> <p>Deklarasi mutableList</p> <p>Deklatasi koneksi database</p> <p>Query mendapatkan data</p> <p>Value untuk menentukan lokasi data</p> <p>Looping jika val cursor masih bergerak ke data selanjutnya</p> <p>Mengambil data nim</p> <p>Mengambil data nama</p> <p>Menyimpan data nim dan nama</p> <p>Ditambahkan ke dalam mutableList</p> <p>Hasil dari function</p>
--	---

Selanjutnya membuat sebuah adapter untuk memasukan data mahasiswa ke dalam recycleView

Buat class baru



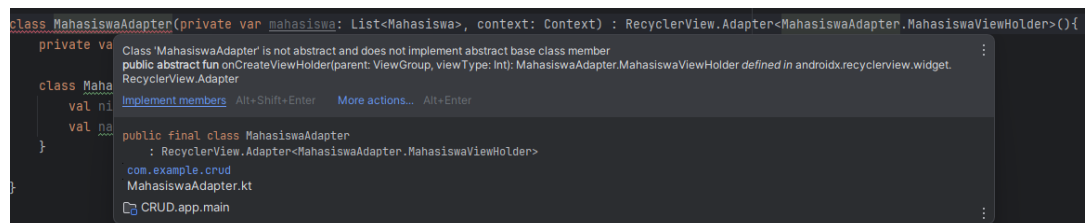
Lalu beri nama MahasiswaAdapter



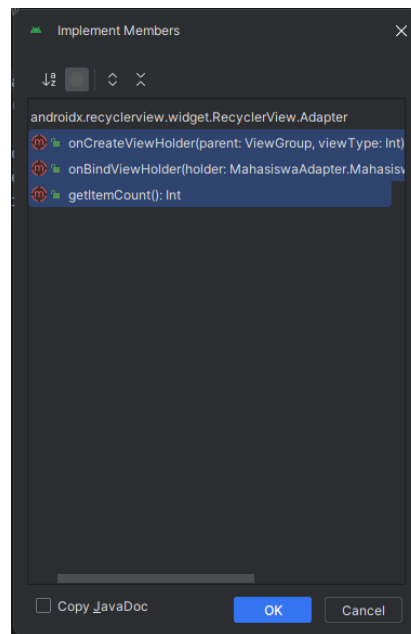
Lalu class ubah class dan tambahkan class MahasiswaViewHolder yang berisi nim dan nama yang ditampilkan ke card\_item\_mahasiswa.xml

```
class MahasiswaAdapter(private var mahasiswa: List<Mahasiswa>, context: Context) :  
    RecyclerView.Adapter<MahasiswaAdapter.MahasiswaViewHolder>() {  
    private val db : DatabaseHelper = DatabaseHelper(context)  
  
    class MahasiswaViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
        val nimTextView : TextView = itemView.findViewById(R.id.nimTextView)  
        val namaTextView : TextView = itemView.findViewById(R.id.namaTextView)  
    }  
}
```

Dekatkan kursor ke class MahasiswaAdapter dan implementasikan member dari class



Lalu klik OK



Lalu masukan code ke dalam 3 class tersebut

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MahasiswaViewHolder {
    val view = LayoutInflater.from(parent.context).inflate(R.layout.card_item_mahasiswa, parent, false)
    return MahasiswaViewHolder(view)
}

override fun getItemCount(): Int = mahasiswa.size

override fun onBindViewHolder(holder: MahasiswaViewHolder, position: Int) {
    val mahasiswa = mahasiswa[position]
    holder.nimTextView.text = mahasiswa.nim
    holder.namaTextView.text = mahasiswa.nama
}
```

Tambahkan function untuk melakukan refreshdata

```
fun refreshData(newMahasiswa: List<Mahasiswa>){
    mahasiswa = newMahasiswa
    notifyDataSetChanged()
}
```

Masuk ke dalam **MainActivity** deklarasikan koneksi database dan mahasiswa adapter

```
private lateinit var db: DatabaseHelper
private lateinit var mahasiswaAdapter: MahasiswaAdapter
```

Tambahkan code untuk memanggil getAllMahasiswa dari **DatabaseHelper** dan binding data ke dalam recyclerView menggunakan **card\_item\_mahasiswa.xml**

```
db = DatabaseHelper(this)
mahasiswaAdapter = MahasiswaAdapter(db.getAllMahasiswa(), this)

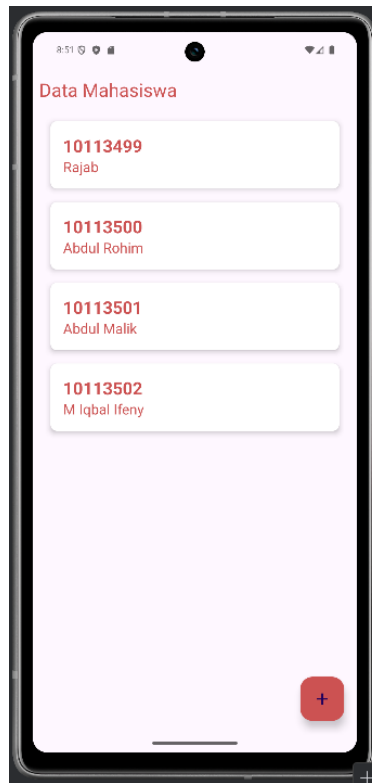
binding.mahasiswaRecyclerView.layoutManager = LinearLayoutManager(this)
binding.mahasiswaRecyclerView.adapter = mahasiswaAdapter
```



Di luar function onCreate panggil function refreshdata

```
override fun onResume() {  
    super.onResume()  
    mahasiswaAdapter.refreshData(db.getAllMahasiswa())  
}
```

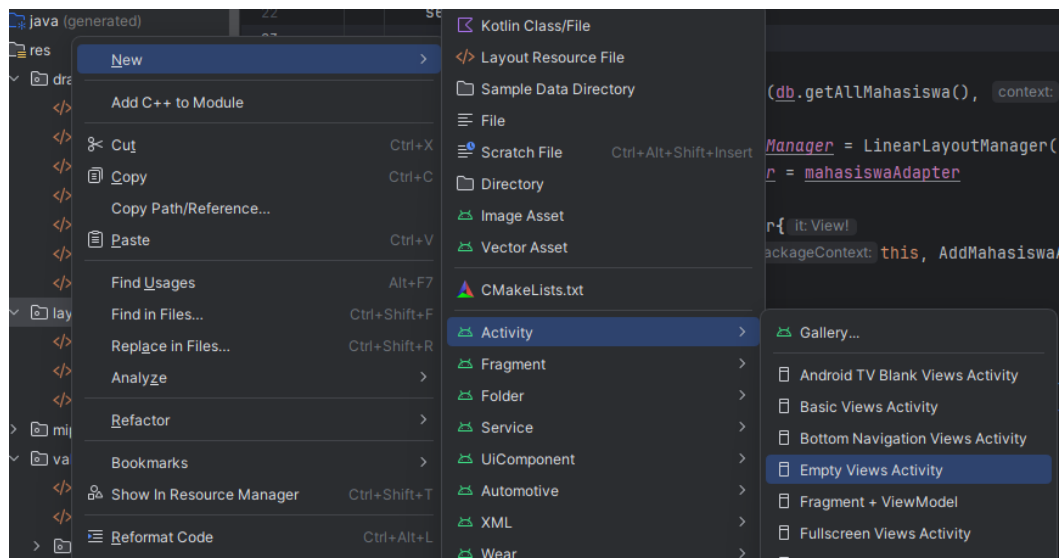
Tampilan Modul READ



#### 4. UPDATE

##### a. DESIGN

Buat sebuah Empty View Activity dengan nama activity UpdateMahasiswaActivity dan otomatis sebuah layout akan terbuat dengan nama activity\_update\_mahasiswa.xml



**Empty Views Activity**  
Creates a new empty activity

Activity Name  
UpdateMahasiswaActivity

☒ Generate a Layout File

Layout Name  
activity\_update\_mahasiswa

☐ Launcher Activity

Package name  
com.example.crud

Source Language  
Kotlin

Lalu klik finish

Masuk ke dalam file `activity_update_mahasiswa.xml` lalu masukan semua code design yang ada di `activity_add_mahasiswa.xml`, rubah nama id dari setiap elemen.

- addMahasiswaHeading menjadi editMahasiswaHeading
- Jangan lupa rubah text yang ada di editMahasiswaHeading menjadi "Edit Mahasiswa"
- saveButton menjadi editSaveButton
- Jangan lupa rubah alignTop yang awalnya addMahasiswaHeading menjadi editMahasiswaHeading
- nimEditText menjadi editNimEditText
- Jangan lupa rubah layout\_below yang awalnya addMahasiswaHeading menjadi editMahasiswaHeading
- namaEditText menjadi editNamaEditText
- Jangan lupa rubah layout\_below yang awalnya nimEditText menjadi editNimEditText

## Code

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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:padding="16dp"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AddMahasiswaActivity">

    <TextView
        android:id="@+id/editMahasiswaHeading"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="16dp"
        android:text="Edit Mahasiswa"
        android:textColor="@color/softred"
        android:textSize="24sp"
    />
```

```

<ImageView
    android:id="@+id/editSaveButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/baseline_save_alt_24"
    android:layout_alignParentEnd="true"
    android:layout_alignTop="@id/editMahasiswaHeading"
    android:layout_marginEnd="6dp"
/>
<EditText
    android:id="@+id/editNimEditText"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:hint="NIM ."
    android:padding="12dp"
    android:textSize="20sp"
    android:layout_below="@id/editMahasiswaHeading"
    android:background="@drawable/border"
    android:maxLines="1"
    android:layout_marginTop="12dp"
/>
<EditText
    android:id="@+id/editNamaEditText"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:hint="NAMA ."
    android:gravity="top"
    android:padding="12dp"
    android:textSize="20sp"
    android:background="@drawable/border"
    android:layout_below="@id/editNimEditText"
    android:layout_marginTop="12dp"
/>
</RelativeLayout>

```

Sehingga tampilannya akan menjadi seperti

Masuk ke dalam `card_item_mahasiswa.xml` dan masukan TextView NIM ke dalam sebuah RelativeLayout

```

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="4dp"
    >
    <TextView
        android:id="@+id/nimTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/_nimTV"
        android:textColor="@color/softred"
        android:textSize="22dp"
        android:textStyle="bold"
    />
</RelativeLayout>

```

Lalu tambahkan 2 ImageView yang memanggil drawable edit hapus ke dalam RelativeLayout yang baru dan beri id updateButton dan deleteButton

```

<ImageView
    android:id="@+id/updateButton"
    android:layout_width="24dp"
    android:layout_height="24dp"
    android:layout_alignParentEnd="true"
    android:layout_marginEnd="30dp"
    android:src="@drawable/baseline_edit_24"
/>

<ImageView
    android:id="@+id/deleteButton"
    android:layout_width="24dp"
    android:layout_height="24dp"
    android:layout_alignParentEnd="true"
    android:src="@drawable/baseline_delete_forever_24"
/>

```

Untuk source/src disesuaikan dengan @drawable yang dibuat di project masing-masing.

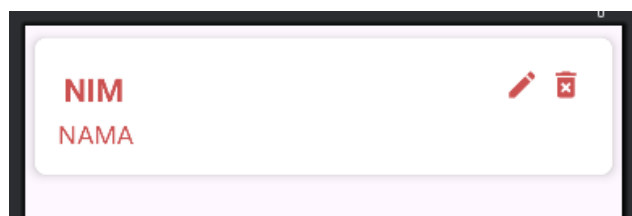
Untuk merubah warna dari imageView bisa masuk ke dalam icon @drawable temukan "tint" dan rubah warna sesuai dengan keinginan masing-masing

```

android:tint="@color/softred"
,2h8c1.1,0 2,-0.9 2,-2L18,7L6,7

```

Untuk tampilan card\_item\_mahasiswa.xml akan menjadi seperti



## b. Logic

Masuk ke **MahasiswaAdapter** dan tambahkan sebuah value yang berfungsi untuk mendeklarasikan **ImageView** yang nantinya akan digunakan untuk melakukan save button di dalam class **MahasiswaViewHolder**

```
class MahasiswaViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
    .....  
    val updateButton : ImageView = itemView.findViewById(R.id.updateButton)  
}
```

tambahkan code di dalam function **onBindViewHolder** untuk melakukan redirect button edit ke halaman Edit Mahasiswa yang membawa nim ("nimmahasiswa") untuk dijadikan kunci ketika melakukan update, karena ketika melakukan update dibutuhkan 2 buah nim yaitu nim yang lama dan nim yang baru. Nim lama dibutuhkan untuk mencari data mana yang akan diupdate dan nim baru akan dimasukan ke dalam data untuk menggantikan nim yang lama.

```
override fun onBindViewHolder(holder: MahasiswaViewHolder, position: Int) {  
    val mahasiswa = mahasiswa[position]  
    holder.nimTextView.text = mahasiswa.nim  
    holder.namaTextView.text = mahasiswa.nama  
  
    holder.updateButton.setOnClickListener {  
        val intent = Intent(holder.itemView.context, UpdateMahasiswaActivity::class.java).apply {  
            putExtra("nimmahasiswa", mahasiswa.nim)  
        }  
        holder.itemView.context.startActivity(intent)  
    }  
}
```

**putExtra** berfungsi untuk membawa value nim ke tampilan baru dengan nama variable "nimmahasiswa"

Masuk ke dalam **UpdateMahasiswaActivity** dan deklarasikan 2 buah variable baru yaitu variable database dan variable **Fnim** (FindNIM) yang nantinya akan digunakan untuk menangkap **putExtra** dari **MahasiswaAdapter** ("nimmahasiswa")

```
private lateinit var db: DatabaseHelper  
private var Fnim: String = ""
```

Di dalam function **onCreate** tambahkan kode untuk memanggil database dan menangkap data "nimmahasiswa"

```
override fun onCreate(savedInstanceState: Bundle?) {  
    .....  
    db = DatabaseHelper(this)  
    Fnim = intent.getStringExtra("nimmahasiswa").toString()  
    .....  
}
```

Masuk ke dalam **DatabaseHelper** lalu tambahkan 1 buah function **getMahasiswaByNIM** yang berfungsi untuk mengambil data mahasiswa untuk ditampilkan di tampilan **activity\_update\_mahasiswa.xml** dengan kunci pencarian nim yang sebelumnya didapatkan dari **putExtra Fnim** -> "nimmahasiswa"

```

fun getMhswbyNIM (nim: String): Mahasiswa{
    val db = readableDatabase
    val query = "SELECT * FROM $TABLE_NAME WHERE $COLUMN_NIM = $nim"
    val cursor = db.rawQuery(query, null)
    cursor.moveToFirst()

    val nim = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NIM))
    val nama = cursor.getString(cursor.getColumnIndexOrThrow(COLUMN_NAMA))

    cursor.close()
    db.close()
    return Mahasiswa(nim, nama)
}

```

Selanjutnya tambahkan 1 buah function `updateMahasiswa` yang berisikan 2 buah parameter, parameter pertama digunakan untuk mengambil data nim lama yang akan digunakan untuk melakukan pencarian, yang sebelumnya didapatkan dari `Fnim` -> "nimmahasiswa", parameter kedua adalah data Mahasiswa baru dengan menggunakan class data `Mahasiswa(nim, nama)`

```

fun updateMahasiswa(oldNIM: String, mahasiswa: Mahasiswa){
    val db = writableDatabase
    val values = ContentValues().apply {
        put(COLUMN_NIM, mahasiswa.nim)
        put(COLUMN_NAMA, mahasiswa.nama)
    }
    val where = "$COLUMN_NIM = ?"
    val arg = arrayOf("$oldNIM")
    db.update(TABLE_NAME, values, where, arg)
    db.close()
}

```

Kembali ke `UpdateMahasiswaActivity` dan panggil function `getMhswbyNIM` untuk mendapatkan data mahasiswa yang akan diedit dan tambahkan `setOnClickListener` untuk button `editSaveButton`

```

override fun onCreate(savedInstanceState: Bundle?){
    .....
    db = DatabaseHelper(this)
    Fnim = intent.getStringExtra("nimmahasiswa").toString()

    val mhsw = db.getMhswbyNIM(Fnim)
    binding.editNimEditText.setText(mhsw.nim)
    binding.editNamaEditText.setText(mhsw.nama)

    binding.editSaveButton.setOnClickListener{
        val newNIM = binding.editNimEditText.text.toString()
        val newNAMA = binding.editNamaEditText.text.toString()
        val updateMhsw = Mahasiswa(newNIM, newNAMA)
        db.updateMahasiswa(Fnim, updateMhsw)
        finish()
        Toast.makeText(this, "Data Berhasil Diupdate",
            Toast.LENGTH_SHORT).show()
    }
    .....
}

```

## 5. DELETE

### a. LOGIC

Masuk ke **MahasiswaAdapter** dan tambahkan sebuah value yang berfungsi untuk mendaklarasikan **ImageView** yang nanti akan digunakan untuk melakukan delete button di dalam class **MahasiswaViewHolder**

```
class MahasiswaViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView){
    .....
    val deleteButton : ImageView = itemView.findViewById(R.id.deleteButton)
}
```

Masuk ke dalam **DatabaseHelper** dan tambahkan function yang berfungsi untuk menghapus data mahasiswa dengan 1 buah parameter yaitu nim

```
fun deleteMahasiswa(nim: String){
    val db = writableDatabase
    val where = "$COLUMN_NIM = ?"
    val arg = arrayOf("$nim")
    db.delete(TABLE_NAME, where, arg)
    db.close()
}
```

Selanjutnya masuk ke dalam **MahasiswaAdapter** dan tambahkan code untuk melakukan aksi delete dari **imageView deleteButton**

```
holder.deleteButton.setOnClickListener{
    db.deleteMahasiswa(mahasiswa.nim)
    refreshData(db.getAllMahasiswa())
    Toast.makeText(holder.itemView.context, "Data Mahasiswa Berhasil Dihapus !", Toast.LENGTH_SHORT).show()
}
```

