

LAPORAN PRAKTIKUM
PEMROGRAMAN NATIVE MOBILE 1



Disusun Oleh :

Cornelia Anggry Ayu Oktavia

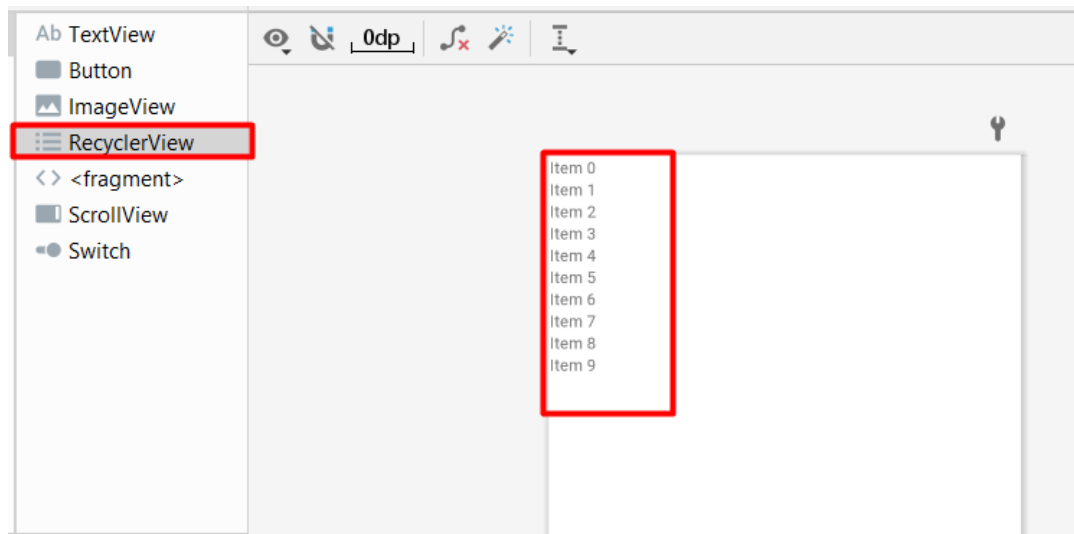
213110008

LABORATORIUM UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

TAHUN 2022/2023

RECYCLER VIEW DENGAN CLICK LISTENER

1. Import RecyclerView ke dalam projek



Tambahkan RecyclerView kedalam program tunggu hingga proses selesai

2. Cek pada layout main



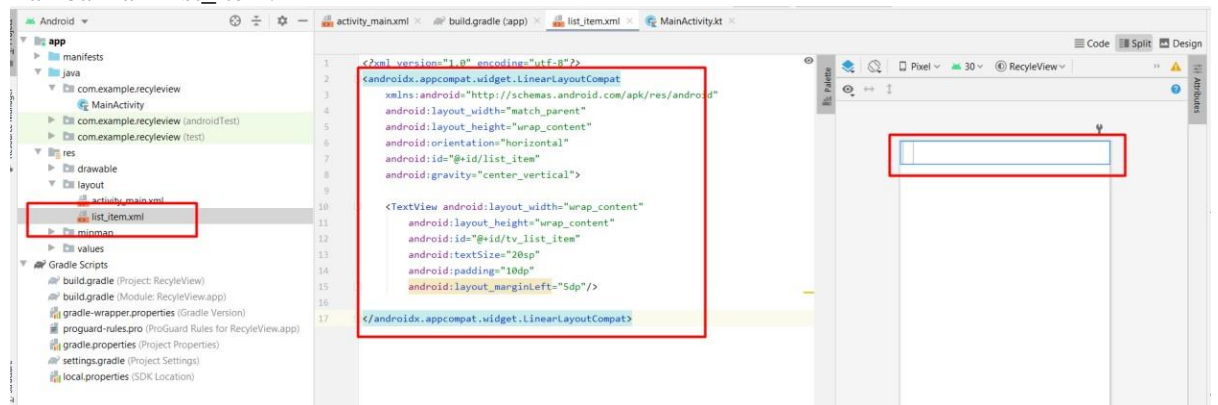
Pada layout main akan ditambahkan kode program otomatis seperti gambar di atas dimana ada lebar dan tinggi, dimana tingginya menyesuaikan dengan layout dan lebarnya menyesuaikan dengan tinggi layout

3. Cek build.Gradle



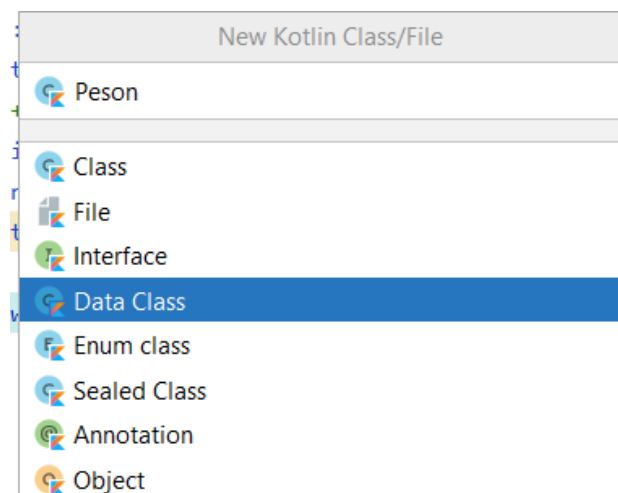
Pada bagian ini ditambahkan dependency recycleView yang mana menggunakan versi 1.2.1

4. Tambahkan List_item.xml



Tambahkan listItem yang mana nanti akan digunakan untuk mengisi data dari RecycleView, list ini akan di panggil dan di tampilkan berulang kali didalam RecycleView

5. Buat data Class



Data class ini berfungsi untuk menjadi model, dimana model ini nantinya yang akan digunakan untuk mengisi adapter dan ditampilkan pada RecycleView

```

1 package com.example.recycleview
2
3 data class Person(
4
5     val name: String = ""
6 )

```

Pada data class ini hanya terdapat 1 variable yaitu name dengan tipe data String dan value awal adalah “ ”

6. Tambahkan Adapter dengan name MyCutomAdapter

```

1 package com.example.recycleview
2
3 import androidx.recyclerview.widget.RecyclerView
4
5 class MyCustomAdapter(val items: List<Person>)
6     : RecyclerView.Adapter<MyCustomAdapter.ViewHolder>() {
7
8
9

```

Buat seperti kode diatas, kemudian tambahkan data implentasi member adapter, antara lain onCreateViewHolder, onBindViewHolder dan getItemCount

The screenshot shows an IDE with a project structure on the left, a code editor in the center, and an 'Implement Members' dialog box on the right.

Project Structure (Left):

- java
 - com.example.recycleview
 - MainActivity
 - MyCustomAdapter
 - Person
 - com.example.recycleview (androidTest)
 - com.example.recycleview (test)
- res
 - drawable
 - layout
 - activity_main.xml
 - list_item.xml
 - mipmap
 - values
- Gradle Scripts
 - build.gradle (Project: RecycleView)
 - build.gradle (Module: RecycleView.app)
 - gradle-wrapper.properties (Gradle Version)
 - proguard-rules.pro (ProGuard Rules for RecycleView.app)
 - gradle.properties (Project Properties)
 - settings.gradle (Project Settings)
 - local.properties (SDK Location)

Code Editor (Center):

```

3 import androidx.recyclerview.widget.RecyclerView
4
5 class MyCustomAdapter(val items: List<Person>)
6     : RecyclerView.Adapter<MyCustomAdapter.ViewHolder>() {
7
8
9
10
11
12
13
14
15

```

Implement Members Dialog (Right):

The dialog shows the 'androidx.recyclerview.widget.RecyclerView.Adapter' interface with the following methods to be implemented:

- onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder
- onBindViewHolder(holder: MyCustomAdapter.ViewHolder, position: Int)
- getItemCount(): Int

At the bottom of the dialog, there are buttons: ☐ Copy JavaDoc, OK, Select None, and Cancel.

7. Berikut penjelasan pada bagian Implentasi member dari Adapter

```

class MyCustomAdapter(val items: List<Person>)
:RecyclerView.Adapter<MyCustomAdapter.ViewHolder>() {

    //Fungsi Yang pertama kali di jalankan ketika Adapter dipanggil
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        TODO( reason: "Not yet implemented")
    }

    //Fungsi yang berguna untuk mengisi data
    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        TODO( reason: "Not yet implemented")
    }

    //fungsi yang digunakan untuk menghitung jumlah data
    override fun getItemCount(): Int {
        TODO( reason: "Not yet implemented")
    }

    //Fungsi yang digunakan untuk menginisialisasi data
    class ViewHolder(val view:View):RecyclerView.ViewHolder(view) {
    }
}

```

8. Tambahkan beberapa kode seperti dibawah ini

```

class MyCustomAdapter(val items: List<Person>)
:RecyclerView.Adapter<MyCustomAdapter.ViewHolder>() {

    //Fungsi Yang pertama kali di jalankan ketika Adapter dipanggil
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val view = LayoutInflater.from(parent.context).inflate(R.layout.list_item,parent, attachToRoot: false)
        return ViewHolder(view)
    }

    //Fungsi yang berguna untuk mengisi data
    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        holder.bind(items[position])
    }

    //fungsi yang digunakan untuk menghitung jumlah data
    override fun getItemCount(): Int {
        return items.size
    }

    //Fungsi yang digunakan untuk menginisialisasi data
    class ViewHolder(val view:View):RecyclerView.ViewHolder(view) {
        fun bind(person: Person){
            itemView.tv_list_item.text =person.name
        }
    }
}

```

Pada onCreate tambahkan dimana layout tersebut akan di isi, pada baigan onBindViewHolder tambahkan holder text untuk mengisi data dan pada bagian item tambahkan perintah untuk menghitung jumlah data

9. Pada MainActivity tambahkan beberapa kode di bawa ini

```

package com.example.recyleview

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.recyclerview.widget.LinearLayoutManager
import kotlinx.android.synthetic.main.activity_main.*

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

        val persons = ArrayList<Person>()
        persons.add(Person( name: "Lilis Pujihati"))
        persons.add(Person( name: "Rudi Hartono"))
        persons.add(Person( name: "Heri Sukresno"))
        persons.add(Person( name: "Nurul Lindawati"))
        persons.add(Person( name: "Susni Susanti"))
        persons.add(Person( name: "Sigit Santosa"))
        persons.add(Person( name: "Ahmad Nawawi"))
        persons.add(Person( name: "Dian Nitami"))
        my_recycler_view.layoutManager = LinearLayoutManager( context: this)
        my_recycler_view.setHasFixedSize (true)
        my_recycler_view.adapter = MyCustomAdapter(persons)
    }
}

```

Berikut beberapa perintah yang digunakan

`val persons = ArrayList<Person>()` → digunakan untuk membuat variable persons yang mana isinya arraylist dari model Person

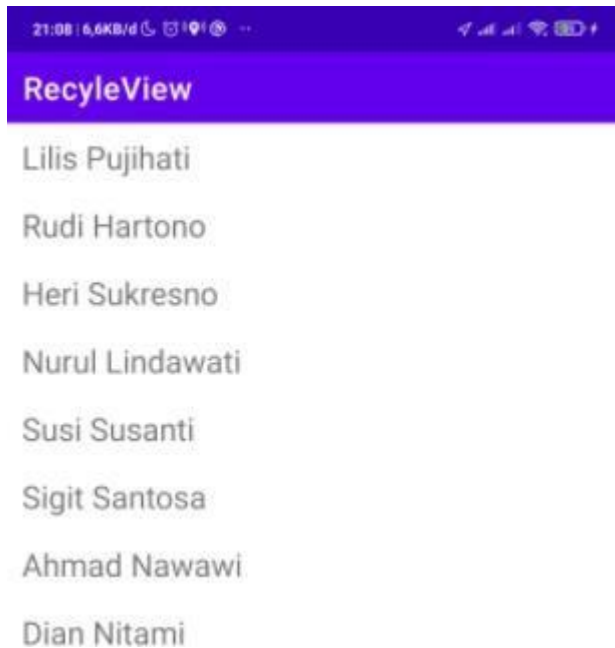
`persons.add(Person("Lilis Pujihati"))` → digunakan untuk menambahkan data dengan nilai Lilis Pujihati

`my_recycler_view.layoutManager = LinearLayoutManager(this)` → digunakan untuk inisialisasi Layout meneger pada recyleview dengan name id my_recycle_view

`my_recycler_view.setHasFixedSize (true)` → digunakan untuk mengset Value dengan FixedSize

`my_recycler_view.adapter = MyCustomAdapter(persons)` → digunakan untuk memanggil adapter

10. Hasilnya seperti gambar dibawah ini



Hasilnya akan menampilkan list dengan data seperti diatas

11. Tambahkan Fungsi itemclick pada mainActivity

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        val persons = ArrayList<Person>()  
        persons.add(Person( name: "Lilis Pujihati"))  
        persons.add(Person( name: "Rudi Hartono"))  
        persons.add(Person( name: "Heri Sukresno"))  
        persons.add(Person( name: "Nurul Lindawati"))  
        persons.add(Person( name: "Susi Susanti"))  
        persons.add(Person( name: "Sigit Santosa"))  
        persons.add(Person( name: "Ahmad Nawawi"))  
        persons.add(Person( name: "Dian Nitami"))  
        my_recycler_view.layoutManager = LinearLayoutManager( context: this)  
        my_recycler_view.setHasFixedSize (true)  
        my_recycler_view.adapter = MyCustomAdapter(persons)  
    }  
    private fun personItemClicked(person: Person) {  
        Toast.makeText( context: this, text: "Clicked: ${person.name}", Toast.LENGTH_SHORT)  
            .show()  
    }  
}
```

Fungsi ini digunakan untuk menampilkan Toas di bagian nama yang di klik

12. Tambahkan variable klik pada adapter

```
class MyCustomAdapter(val items: List<Person>
, val clickListener: (Person) -> Unit):
    RecyclerView.Adapter<MyCustomAdapter.ViewHolder>() {
```

Variable yang di tambahkan berisikan unit

13. Tambahkan Parameter di ViewHolder

```
class ViewHolder(val view:View):RecyclerView.ViewHolder(view) {
    fun bind(person: Person, clickListener: (Person) -> Unit){
        itemView.tv_list_item.text =person.name
        itemView.setOnClickListener { clickListener(person)}
    }
}
```

Parameter ini digunakan untuk menjalankan fungsi ketika di klik

14. Tambahkan parameter di onBindViewHolder

```
//Fungsi yang berguna untuk mengisi data
override fun onBindViewHolder(holder: ViewHolder, position: Int) {
    holder.bind(items[position], clickListener)
}
```

Parameter clickListener ini digunakan untuk memanggil fungsi clickListener dari holder

15. Tambahkan fungsi pemanggil di MainActivity

```
my_recycler_view.adapter = MyCustomAdapter(persons, {person -> personItemClicked(person)} )
```

Fungsi di atas digunakan untuk memanggil fungsi personItemClicked dengan parameternya adalah person

16. Hasilnya Sebagai berikut

RecycleView

Lilis Pujihati

Rudi Hartono

Heri Sukresno

Nurul Lindawati

Susi Susanti

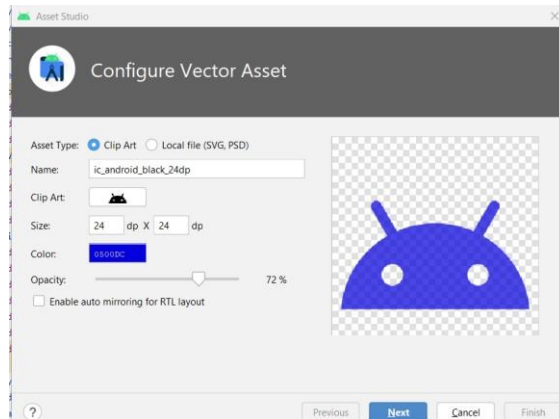
Sigit Santosa

Ahmad Nawawi

Dian Nitami

Clicked: Sigit Santosa

17. Tambahkan Vektor asset



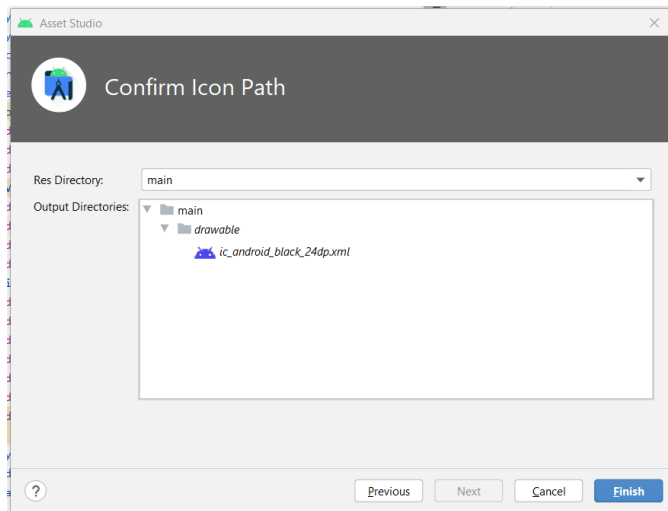
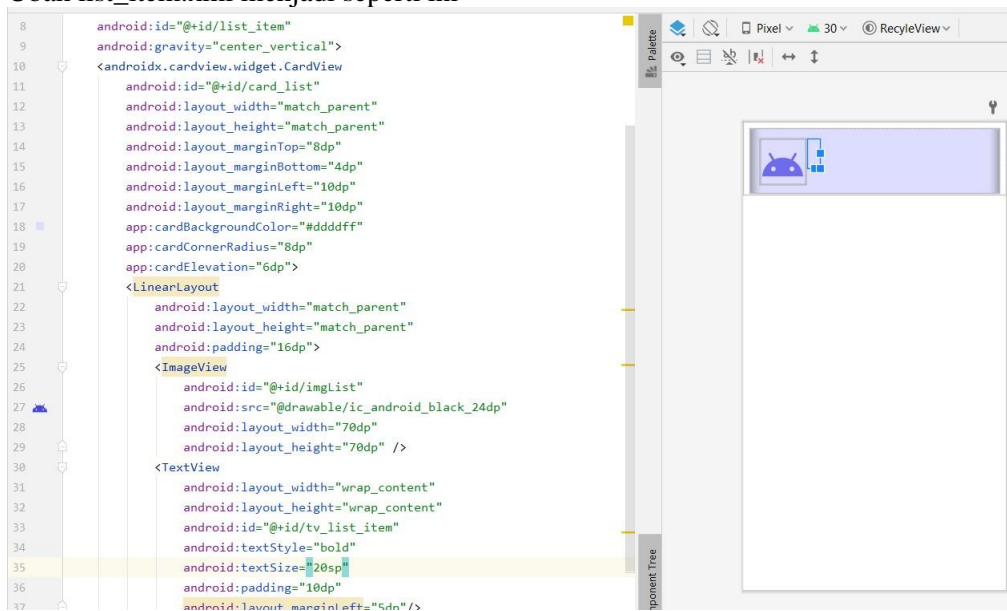


Image vector ini nantinya akan digunakan untuk sebuah tampilan di list item

18. Ubah list_item.xml menjadi seperti ini



Yang mana terdapat sebuah image dan text view yang mana imagenya akan diisi dengan sebuah asset image

19. Pada Build gradel tambahkan cardView

```
dependencies {

    implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
    implementation 'androidx.core:core-ktx:1.5.0'
    implementation 'androidx.appcompat:appcompat:1.3.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'

    implementation 'androidx.recyclerview:recyclerview:1.2.1'
    implementation 'androidx.cardview:cardview:1.0.0'
}
```

Card view ini nanti digunakan untuk membuat card dari tampilan RecyclerView

20. Tambahkan Tipe Layout pada MainActivity

```
my_recycler_view.layoutManager = LinearLayoutManager(context, LinearLayoutManager.HORIZONTAL, reverseLayout: false)
my_recycler_view.setHasFixedSize(true)
my_recycler_view.adapter = MyCustomAdapter(persons, {person -> personItemClicked(person)})

private fun personItemClicked(person: Person) {
```

Yang mana layout akan diatur secara horizontal

21. Hasilnya akan seperti ini



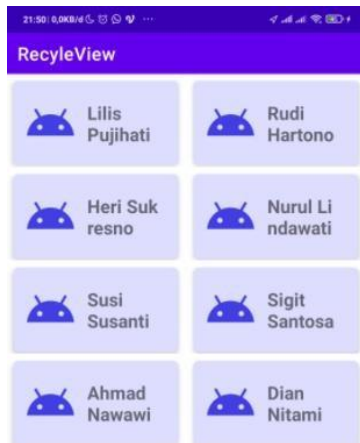
Tampilan dari linierlayout tipe horizontal adalah cardView akan ditampilkan secara horizontal hingga akan bergeser ke kanan dan kekiri

22. Ubah LinierLayoutManager menjadi gridLayoutManager

```
val persons = ArrayList<Person>()
persons.add(Person(name: "Lilis Pujihati"))
persons.add(Person(name: "Rudi Hartono"))
persons.add(Person(name: "Heri Sukresno"))
persons.add(Person(name: "Nurul Lindawati"))
persons.add(Person(name: "Susi Susanti"))
persons.add(Person(name: "Sigit Santosa"))
persons.add(Person(name: "Ahmad Nawawi"))
persons.add(Person(name: "Dian Nitami"))
my_recycler_view.layoutManager = GridLayoutManager(context, this, spanCount: 2)
my_recycler_view.adapter = MyCustomAdapter(persons, {person -> personItemClicked(person)})
```

Perintah diatas digunakan untuk mengatur layout adapter menjadi GridLayout adapter dengan tampilan setiap 2 barisnya terdiri dari 2 span

23. Hasilnya akan menjadi seperti ini



Gambar diatas merupakan hasil dari gridLayout yang mana menampilkan hasil secara grid dengan 2 gambar per baris

Latihan

Tambahkan model nama

```
data class Person(  
    val name: String = "",  
    val umur: String = ""  
)
```

Pada model Person tambahkan variable baru dengan nama umur dengan value ""

Tambahkan fungsi list

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="30dp"  
        android:id="@+id/tv_list_item"  
        android:textSize="20sp"  
        android:text="testnama"  
        android:layout_marginLeft="5dp"/>  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="30dp"  
        android:id="@+id/tv_list_name"  
        android:textStyle="bold"  
        android:text="testnomer"  
        android:textSize="20sp"  
        android:layout_marginLeft="5dp"/>  
</LinearLayout>
```

Tambahkan satu textView dalam Layout, yang mana digunakan untuk menampung nomer

Tambahkan di Fungsi

```

class ViewHolder(val view:View):RecyclerView.ViewHolder(view) {
    fun bind(person: Person, clickListener: (Person) -> Unit){
        itemView.tv_list_item.text =person.name
        itemView.tv_list_name.text=person.umur
        itemView.setOnClickListener { clickListener(person)}
    }
}

```

Pada adapter di ViewHolder tambah satu lagi dimana ditambahkan sebuah view yang diisi dengan umur

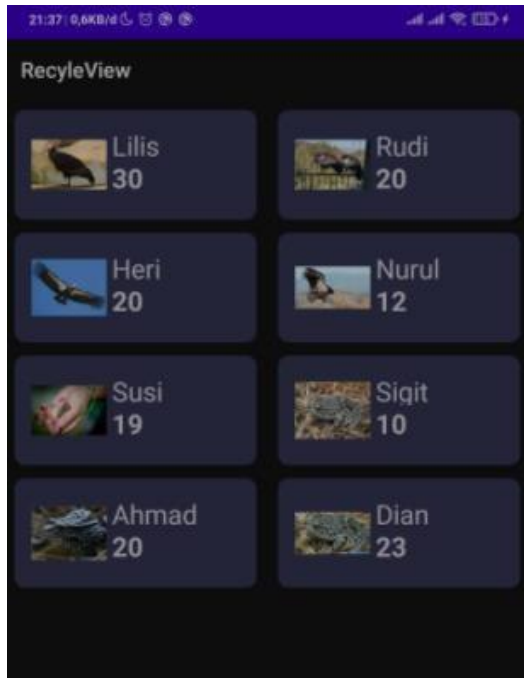
Berikut hasilnya



Hasilnya menampilkan nama dan umur dari data yang dimasukkan

TUGAS

Menambahkan gambar di dalam recyleView



1. Tambahkan Glide di build gradel

```
implementation 'com.github.bumptech.glide:glide:4.11.0'
```

2. Tambahkan variable didalam model untuk menampung gambar

```
data class Person(  
    val name: String = "",  
    val umur: String = "",  
    val img: Int = 0  
)
```

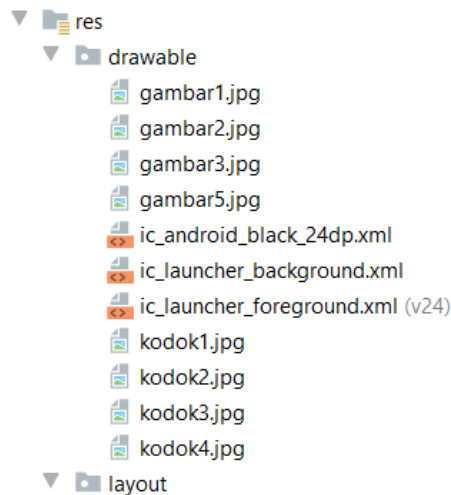
Tipe gambar int karena langsung mengambil data

3. Tambahkan glide di Adapter

```
class ViewHolder(val view: View): RecyclerView.ViewHolder(view) {  
    fun bind(person: Person, clickListener: (Person) -> Unit) {  
        itemView.tv_list_item.text = person.name  
        itemView.tv_list_name.text = person.umur  
        Glide.with(itemView.context)  
            .load(person.img)  
            .apply(RequestOptions().override(width = 120, height = 100))  
            .into(itemView.imgList)  
        itemView.setOnClickListener {  
            clickListener(person)  
        }  
    }  
}
```

Tambahkan perintah glide di adapter, yang mana digunakan untuk membaca gambar, pada bagian apply .override digunakan untuk mengatur lebar dan tinggi dari gambar

4. Tambahkan gambar kedalam Draweble



5. Pada MainActivity pangil data

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

        val persons = ArrayList<Person>()
        persons.add(Person( name: "Lilis Pujihati", umur: "30", R.drawable.gambar1 ))
        persons.add(Person( name: "Rudi Hartono", umur: "20", R.drawable.gambar2))
        persons.add(Person( name: "Heri Sukresno", umur: "20", R.drawable.gambar3))
        persons.add(Person( name: "Nurul Lindawati", umur: "12", R.drawable.gambar5))
        persons.add(Person( name: "Susi Susanti", umur: "19", R.drawable.kodok1))
        persons.add(Person( name: "Sigit Santosa", umur: "10", R.drawable.kodok2))
        persons.add(Person( name: "Ahmad Nawawi", umur: "20", R.drawable.kodok3))
        persons.add(Person( name: "Dian Nitami", umur: "23", R.drawable.kodok4))
        my_recycler_view.layoutManager = GridLayoutManager( context: this, spanCount: 2)
        my_recycler_view.adapter = MyCustomAdapter(persons, {person -> personItemClicked(person) })
    }

    private fun personItemClicked(person: Person) {
        Toast.makeText( context: this, text: "Clicked: ${person.name}", Toast.LENGTH_SHORT)
            .show()
    }
}
```

KESIMPULAN

RecyclerView digunakan untuk menampilkan data yang mana data yang ditampilkan memiliki Format yang sama, Implementasi recyleView terdiri dari 3 komponen yaitu onCreate, onBind dan getCount. Sedang untuk mengatur tampilan dapat menggunakan linier layout manager atau menggunakan gridlayout, untuk menampilkan gambar menggunakan glid