

## MODUL 5

### RecyclerView



#### CAPAIAN PEMBELAJARAN

---

1. Mahasiswa dapat merepresentasikan data dengan menggunakan komponen recyclerview



#### KEBUTUHAN ALAT/BAHAN/SOFTWARE

---

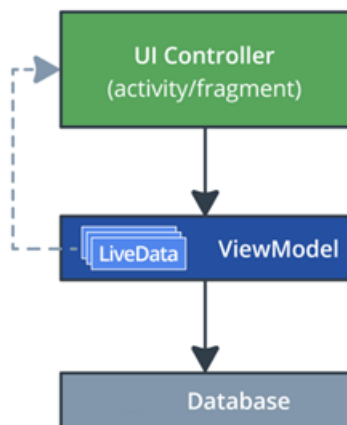
1. Android Studio 3.4.
2. Handphone Android versi 7.0 (Nougat)
3. Kabel data USB.
4. Driver ADB.



#### DASAR TEORI

---

RecyclerView adalah tampilan yang menggunakan arsitektur yang disederhanakan dengan UI controller, ViewModel, dan LiveData.



Menampilkan list atau grid data adalah salah satu tugas UI paling umum di Android. Daftar bervariasi dari yang sederhana hingga yang sangat kompleks. Daftar tampilan teks mungkin menampilkan data sederhana, seperti daftar belanja. Daftar yang kompleks, seperti daftar tujuan

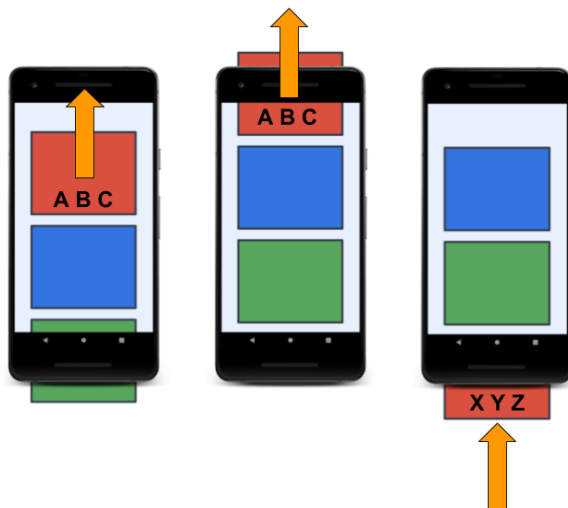
liburan yang beranotasi, dapat menunjukkan kepada pengguna banyak detail di dalam scrolling grid dengan header. Untuk mendukung semua kasus penggunaan ini, Android menyediakan widget RecyclerView.



Manfaat terbesar dari RecyclerView adalah sangat efisien untuk daftar besar:

- Secara default, RecyclerView hanya berfungsi untuk memproses atau menggambar item yang saat ini terlihat di layar. Misalnya, jika list memiliki seribu elemen tetapi hanya 10 elemen yang terlihat, RecyclerView hanya berfungsi untuk menggambar 10 item di layar. Ketika pengguna melakukan scroll, RecyclerView mengetahui item baru apa yang seharusnya ada di layar dan tidak cukup berfungsi untuk menampilkan item itu.
- Ketika suatu item scroll dari layar, tampilan item tersebut didaur ulang. Itu berarti item diisi dengan konten baru yang scroll ke layar. Perilaku RecyclerView ini menghemat banyak waktu pemrosesan dan membantu scroll list dengan lancar.
- Ketika suatu item berubah, alih-alih menggambar ulang seluruh daftar, RecyclerView dapat memperbarui satu item itu. Ini adalah keuntungan efisiensi yang sangat besar ketika menampilkan daftar item kompleks!

Dalam urutan yang ditunjukkan di bawah ini, kita dapat melihat bahwa satu tampilan telah diisi dengan data, ABC. Setelah itu tampilan bergulir dari layar, RecyclerView menggunakan kembali tampilan untuk data baru, XYZ.

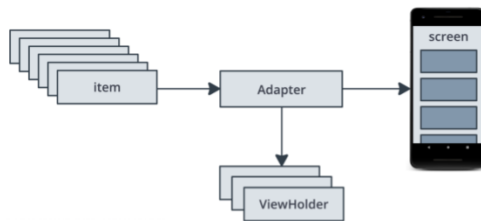


#### Adapter pattern

Jika kita pernah bepergian antar negara yang menggunakan soket listrik yang berbeda, kita mungkin tahu bagaimana kita bisa mencolokkan perangkat kita ke outlet dengan menggunakan adaptor. Adaptor memungkinkan kita mengonversi satu jenis steker ke yang lain, yang benar-benar mengubah satu antarmuka menjadi yang lain. Pola adaptor dalam rekayasa perangkat lunak

membantu objek bekerja dengan API lain. RecyclerView menggunakan adaptor untuk mengubah data aplikasi menjadi sesuatu yang dapat ditampilkan RecyclerView, tanpa mengubah cara aplikasi menyimpan dan memproses data. Untuk aplikasi pelacak tidur, kita membuat adaptor yang mengadaptasi data menjadi sesuatu yang RecyclerView tahu cara menampilkannya, tanpa mengubah ViewModel.

Mengimplementasikan sebuah RecyclerView



Untuk menampilkan data dalam RecyclerView, memerlukan bagian-bagian berikut:

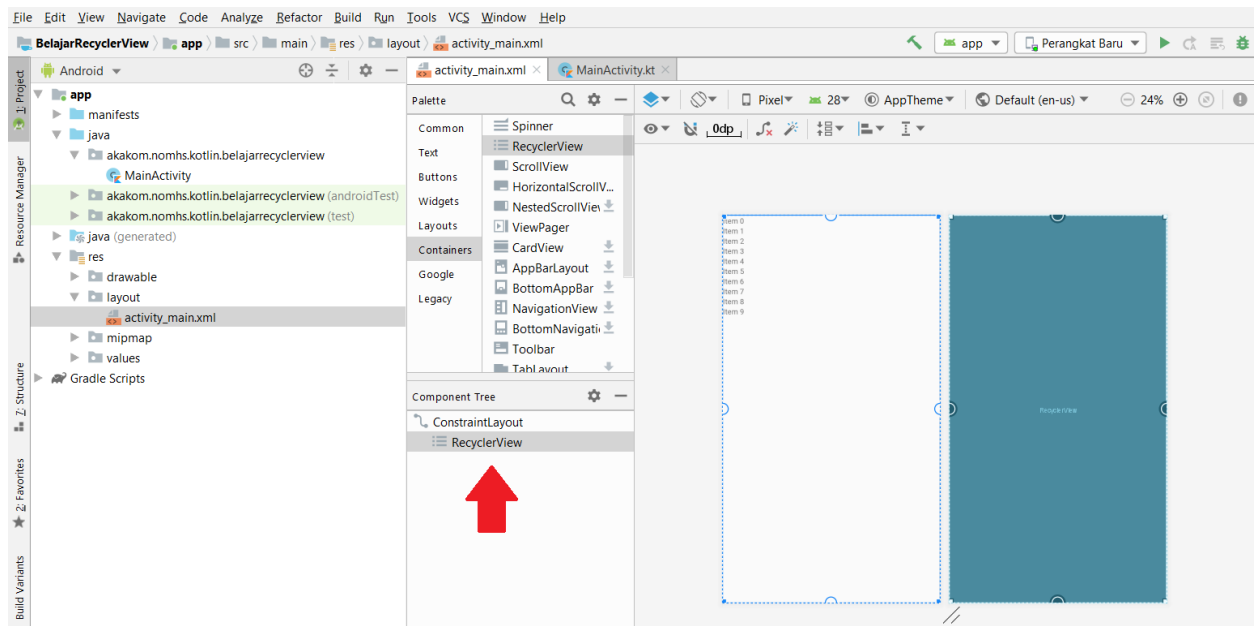
- Data untuk ditampilkan.
- Mesin virtual RecyclerView didefinisikan dalam file layout, untuk bertindak sebagai wadah untuk tampilan.
- Layout untuk satu item data.  
Jika semua item list terlihat sama, kita dapat menggunakan layout yang sama untuk semuanya, tetapi itu tidak wajib. Layout item harus dibuat secara terpisah dari layout fragmen, sehingga tampilan satu item pada satu waktu dapat dibuat dan diisi dengan data.
- Layout Manager.  
Layout Manager menangani organisasi (layout) komponen UI dalam tampilan.
- View holder.  
view holder extends kelas ViewHolder. Ini berisi informasi tampilan untuk menampilkan satu item dari layout item. Penampil tampilan juga menambahkan informasi yang digunakan RecyclerView untuk memindahkan tampilan di layar secara efisien.
- Adaptor.  
Adaptor menghubungkan data kita ke RecyclerView. Ini menyesuaikan data sehingga dapat ditampilkan di ViewHolder. RecyclerView menggunakan adaptor untuk mengetahui cara menampilkan data di layar.



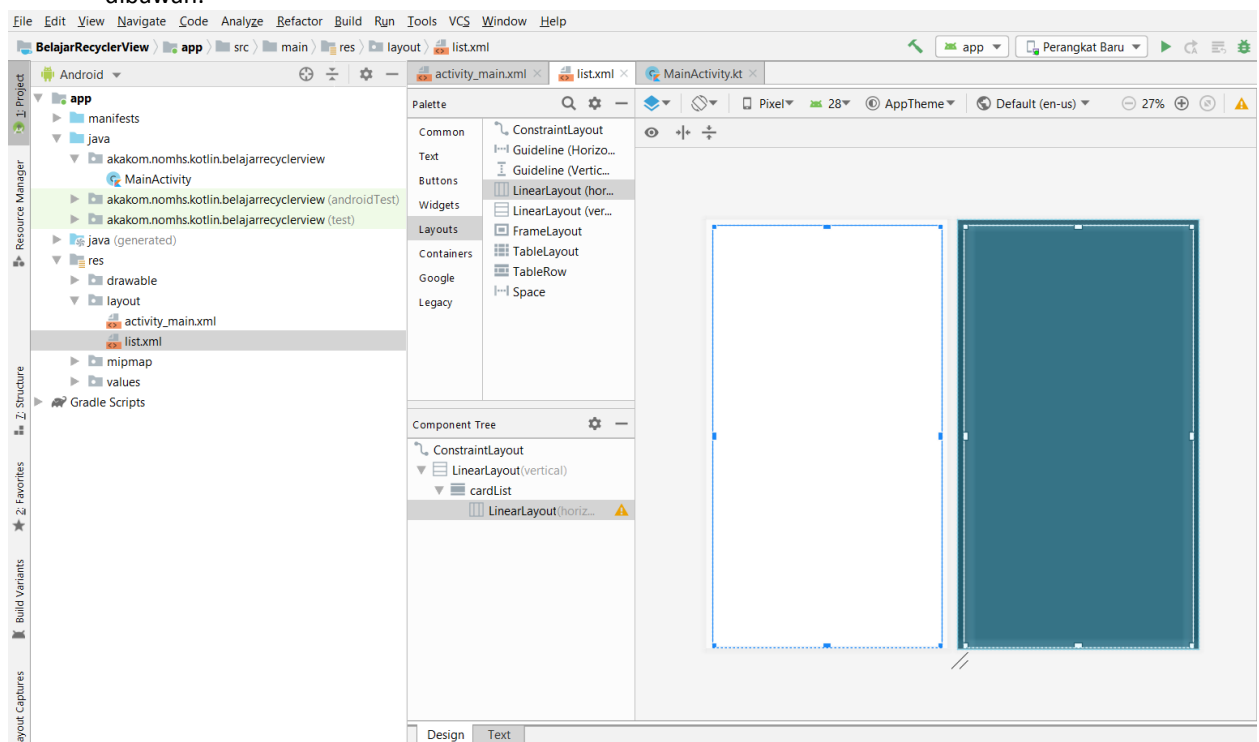
## PRAKTIK

---

1. Buat Project baru. (dalam contoh ini diberi nama BelajarRecyclerView)
2. Pada layout\_main.xml hapus komponen yang sudah ada (biasanya TextView).
3. Dari tab Palette, pilih Container, kemudian pilih RecyclerView
4. Klik dan drag ke Componen Tree, masukkan dibawah ConstrainLayout. Hasilnya seperti dibawah. (Perhatikan panah merah)



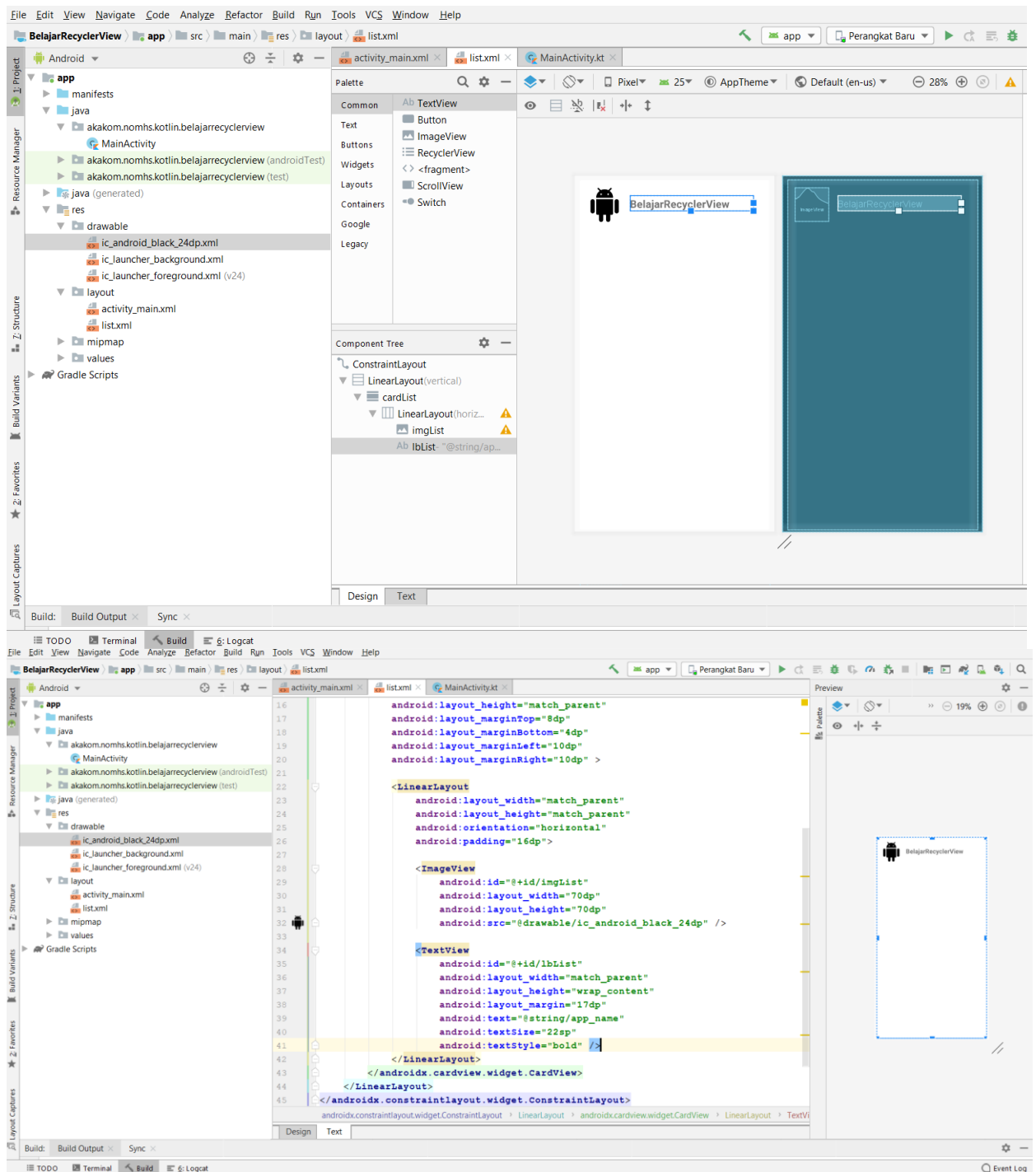
5. Buat sebuah file dibawah layout, beri nama list.xml
6. Tambahkan LinearLayout (vertical) didalam Container ConstraintLayout. Kemudian tambahkan cardaView dari Palette Container. Kemudian tambahkan LinearLayout(horisontal). Sampai sejauh ini, hasilnya dapat dilihat seperti gambar dibawah.



7. Kemudian, buka tab Text, ubah sehingga menjadi seperti dibawah ini.

```
activity_main.xml x list.xml x MainActivity.kt x Users.kt x Adapter.kt x
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     android:layout_width="match_parent"
6     android:layout_height="wrap_content">
7     <LinearLayout
8         android:layout_width="match_parent"
9         android:layout_height="match_parent"
10        android:orientation="vertical">
11
12        <android.support.v7.widget.CardView
13            android:layout_width="match_parent"
14            android:layout_height="match_parent"
15            android:id="@+id/cardList"
16            android:layout_marginTop="8dp"
17            android:layout_marginBottom="4dp"
18            android:layout_marginLeft="10dp"
19            android:layout_marginRight="10dp">
20
21            <LinearLayout
22                android:layout_width="match_parent"
23                android:layout_height="match_parent"
24                android:orientation="horizontal"
25                android:padding="16dp">
26
27            </LinearLayout>
28        </android.support.v7.widget.CardView>
29    </LinearLayout>
30</android.support.constraint.ConstraintLayout>
```

8. Kemudian tambahkan Image Button dan TextView dari tab Design dan kemudian modifikasi beberapa atribut dari tab Text. Perhatikan tampilan kedua tab dibawah.



9. Buka file MainActivity.kt, modifikasilah menjadi seperti berikut ini. List digunakan untuk membuat daftar String yang akan ditampilkan pada RecyclerView. Kemudian daftar tersebut dimasukkan ke dalam Adapter dan kemudian Adapter dipasang pada RecyclerView.

```

Adapter.kt x MainActivity.kt x list.xml x Users.kt x activity_main.xml x
9 class MainActivity : AppCompatActivity() {
10     val list = ArrayList<Users>()
11     val listUsers = arrayOf(
12         "Alpha",
13         "Bravo",
14         "Charlie",
15         "Delta",
16         "Echo",
17         "Foxtrot",
18         "Golf",
19         "Hotel",
20         "India",
21         "Juliet"
22     )
23     override fun onCreate(savedInstanceState: Bundle?) {
24         super.onCreate(savedInstanceState)
25         setContentView(R.layout.activity_main)
26         mRecyclerView.setHasFixedSize(true)
27         mRecyclerView.layoutManager = LinearLayoutManager(context, this)
28         for (i in 0 until listUsers.size) {
29             list.add(Users(listUsers.get(i)))
30             if (listUsers.size - 1 == i) {
31                 val adapter = Adapter(list)
32                 adapter.notifyDataSetChanged()
33                 mRecyclerView.adapter = adapter
34             }
35         }
36     }
37 }

```

- Setelah itu buat file kotlin dibawah package yang sama dengan file MainActivity.kt dengan nama Users.kt dan Adapter.kt. Program Adapter.kt ini digunakan untuk membuat Adapter dari RecyclerView. Dan class Users.kt digunakan untuk menampung daftar.

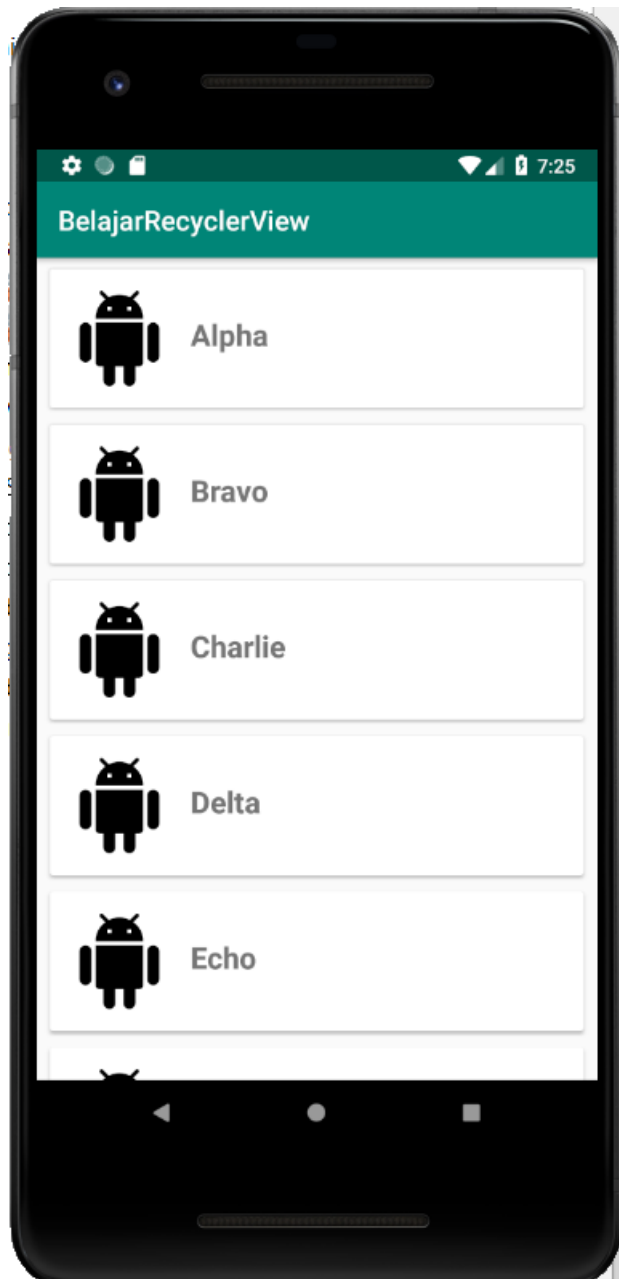
```

Adapter.kt x MainActivity.kt x list.xml x Users.kt x activity_main.xml x
9 class Adapter(private val list: ArrayList<Users>) : RecyclerView.Adapter<Adapter.Holder>() {
10
11     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): Holder {
12         return Holder(LayoutInflater.from(parent.context).inflate(R.layout.list, parent, attachToRoot: false))
13     }
14
15     override fun getItemCount(): Int = list.size
16
17     override fun onBindViewHolder(holder: Holder, position: Int) {
18         holder.view.lbList.text = list.get(position).name
19     }
20
21     class Holder(val view: View) : RecyclerView.ViewHolder(view)
22 }

```

```
Adapter.kt x MainActivity.kt x list.xml x Users.kt x activity_main.xml x
3 data class Users (val name:String?)
4
5
6
```

11. Jalankan. Dan anda akan mendapatkan hasil seperti berikut ini. Scroll ke bawah untuk mendapatkan list berikutnya.







## LATIHAN

---

1. Modifikasilah aplikasi dengan menambahkan Toast jika salah satu list dipilih.



## TUGAS

---

1. Buat aplikasi baru dengan menerapkan RecyclerView



## REFERENSI

---

1. <https://kotlinlang.org/docs/reference/>
2. <https://developer.android.com/kotlin>
3. <https://developer.android.com/courses/kotlin-android-fundamentals/toc>
4. <https://codelabs.developers.google.com/android-kotlin-fundamentals/>
5. <https://developer.android.com/kotlin/learn>
6. <https://developer.android.com/kotlin/resources>