

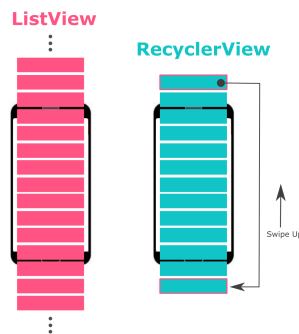
Modul 5 Recycler View

Apa itu Recycler View?

- Sebuah komponen yang mirip dengan listview
- listview user interface pada Android Studio yang menampilkan item-item dari sekumpulan daftar yang tersusun berbaris ke bawah atau kesamping dengan tampilan yang dapat scroll.

Apa perbedaan ListView dan RecyclerView?

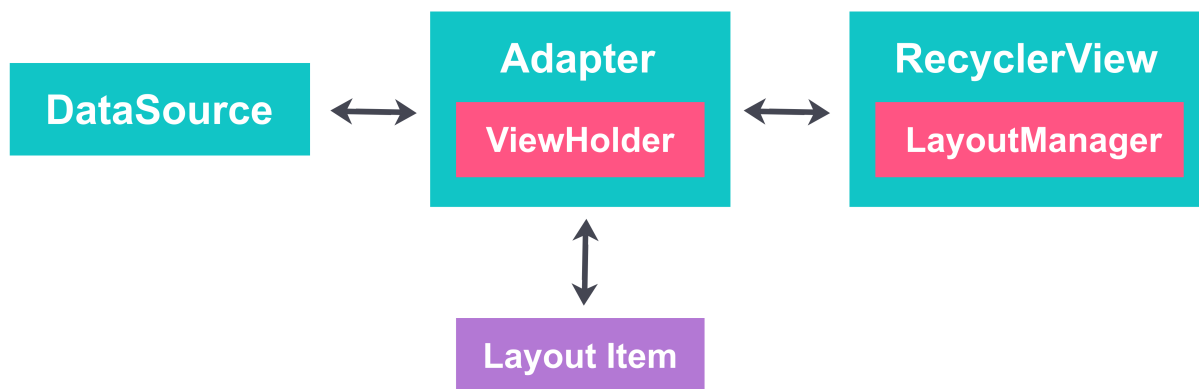
- ListView akan menampilkan objek dari tiap item dari awal hingga akhir.
- RecyclerView akan menampilkan objek hanya sebatas ukuran layar dan beberapa diatas dan dibawahnya saja



Kapan sih RecyclerView digunakan?

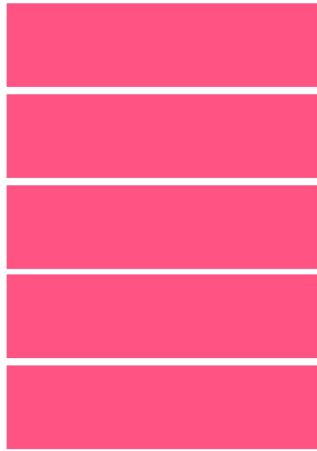
- Apabila kita memiliki data yang sedikit tidak akan terasa karena memori masih sanggup dalam menanggapi aplikasinya. Sedangkan apabila kita memiliki setidaknya 1000 data akan memakan memori yang cukup banyak.
- Menggunakan RecyclerView aplikasi akan menjadi lebih efisien dan tidak menghabiskan memori.

Penarapan RecyclerView

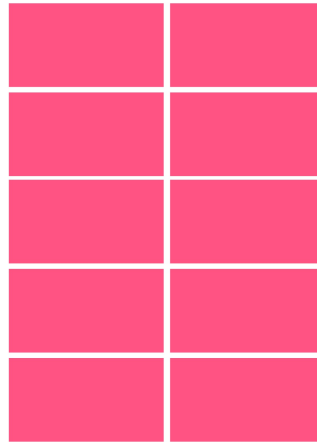


1. **RecyclerView dan LayoutManager:**

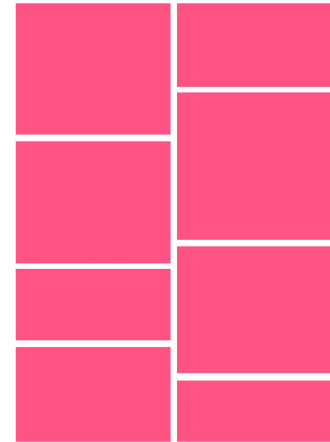
Komponen antarmuka yang bertugas untuk menampilkan data set yang dimiliki di dalamnya. Layoutmanager akan mengatur posisi tampilan data baik itu secara list (vertikal), grid (baris dan kolom) atau staggered grid (grid yang memiliki susunan tak seragam / tak beraturan).



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.

Langkah-langkah mengimplementasikan RecyclerView

Langkah-langkah mengimplementasikan recyclerview sebagai berikut :

1. Tambahkan `dependencies` komponen recyclerview pada file `build.gradle` level modul.
2. Tambahkan obyek `RecyclerView` di berkas layout xml dari activity / fragment.
3. Definiskan model kelas (POJO) yang akan digunakan sebagai *data source*.
4. Buat berkas layout xml untuk baris item di `RecyclerView`.
5. Buat sebuah kelas adapter yang inherit ke `RecyclerView.Adapter` dan `ViewHolder` untuk menampilkan tiap elemen data.
6. Definiskan obyek `RecyclerView` berikut dengan bentuk yang diinginkan (bisa dalam bentuk list, grid, atau staggered) dan selanjutnya pasang obyek adapter (*binding*) agar bisa menampilkan koleksi data ke dalam `RecyclerView`.

Code Lab RecyclerView

Sebelum mengimplementasikan recyclerview maka kita harus menambahkan depedencinya pada build.gradle (module : app

```
implementation 'androidx.recyclerview:recyclerview:1.1.0'
```

Membuat RecyclerView pada xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv_heroes"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:listitem="@layout/item_row_hero" />

</RelativeLayout>
```

Membuat bagian detail pada saat menampilkan recycler viewnya agar memiliki bentuk yang sama

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="16dp">

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/img_item_photo"
        android:layout_width="55dp"
        android:layout_height="55dp"
        android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp"
        tools:src="@android:color/darker_gray" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_toEndOf="@id/img_item_photo"
        android:layout_toRightOf="@id/img_item_photo"
        android:orientation="vertical">

        <TextView
            android:id="@+id/tv_item_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="8dp"
            android:textSize="16sp"
            android:textStyle="bold"
            tools:text="@string/heroes_name" />

        <TextView
            android:id="@+id/tv_item_detail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ellipsize="end"
            android:maxLines="2"
            tools:text="@string/detail" />

    </LinearLayout>

</LinearLayout>
```

Dilanjutkan dengan membuat adapter, namun sebelumnya kita akan membuat data object dan dataclass agar memudahkan nantinya dan lebih dinamis

```
data class Hero(
    var name: String = "",
    var detail: String = "",
    var photo: Int = 0
)
```

```

object HeroesData {
    private val heroNames = arrayOf("Ahmad Dahlan",
        "Ahmad Yani",
        "Sutomo",
        "Gatot Soebroto",
        "Ki Hadjar Dewantara",
        "Mohammad Hatta",
        "Soedirman",
        "Soekarno",
        "Soepomo",
        "Tan Malaka")

    private val heroDetails = arrayOf("Salah seorang ulama dan khatib terkemuka di Masjid Besar Kasultanan Yogyakarta pada masa itu, dan ib
        "Jenderal TNI Anumerta Ahmad Yani (juga dieja Achmad Yani; lahir di Purworejo, Jawa Tengah, 19 Juni 1922 – meninggal di Lubang
        "Sutomo (lahir di Surabaya, Jawa Timur, 3 Oktober 1920 – meninggal di Padang Arafah, Arab Saudi, 7 Oktober 1981 pada umur 61 ta
        "Jenderal TNI (Purn.) Gatot Soebroto (lahir di Sumpiuh, Banyumas, Jawa Tengah, 10 Oktober 1907 – meninggal di Jakarta, 11 Juni
        "Raden Mas Soewardi Soerjaningrat (EBI: Suwardi Suryaningrat, sejak 1922 menjadi Ki Hadjar Dewantara, EBI: Ki Hajar Dewantara,
        "Dr.(HC) Drs. H. Mohammad Hatta (lahir dengan nama Mohammad Athar, dikenal sebagai Bung Hatta; lahir di Fort de Kock (sekarang
        "Jenderal Besar Raden Soedirman (EYD: Sudirman; lahir 24 Januari 1916 – meninggal 29 Januari 1950 pada umur 34 tahun) adalah se
        "Dr.(H.C.) Ir. H. Soekarno (ER, EYD: Sukarno, nama lahir: Koesno Sosrodihardjo) (lahir di Surabaya, Jawa Timur, 6 Juni 1901 – m
        "Prof. Mr. Dr. Soepomo (Ejaan Soewandi: Supomo; lahir di Sukoharjo, Jawa Tengah, 22 Januari 1903 – meninggal di Jakarta, 12 Sep
        "Tan Malaka atau Ibrahim gelar Datuk Sutan Malaka (lahir di Nagari Pandam Gadang, Suliki, Lima Puluh Kota, Sumatera Barat, 2 Ju

    private val heroesImages = intArrayOf(R.drawable.ahmad_dahlan,
        R.drawable.ahmad_yani,
        R.drawable.bung_tomo,
        R.drawable.gatot_subroto,
        R.drawable.ki_hadjar_dewantara,
        R.drawable.mohammad_hatta,
        R.drawable.sudirman,
        R.drawable.sukarno,
        R.drawable.supomo,
        R.drawable.tan_malaka)

    val listData: ArrayList<Hero>
    get() {
        val list = arrayListOf<Hero>()
        for (position in heroNames.indices) {
            val hero = Hero()
            hero.name = heroNames[position]
            hero.detail = heroDetails[position]
            hero.photo = heroesImages[position]
            list.add(hero)
        }
        return list
    }
}

```

Lalu dilanjutkan dengan membuat adapter

```

class ListHeroAdapter(val listHero: ArrayList<Hero>) : RecyclerView.Adapter<ListHeroAdapter.ViewHolder>() {

    override fun onCreateViewHolder(viewGroup: ViewGroup, i: Int): ViewHolder {
        val view: View = LayoutInflater.from(viewGroup.context).inflate(R.layout.item_row_hero, viewGroup, false)
        return ViewHolder(view)
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        val hero = listHero[position]
        Glide.with(holder.itemView.context)
            .load(hero.photo)
            .apply(RequestOptions().override(55, 55))
            .into(holder.imgPhoto)
        holder.tvName.text = hero.name
        holder.tvDetail.text = hero.detail
    }

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

    inner class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        var tvName: TextView = itemView.findViewById(R.id.tv_item_name)
        var tvDetail: TextView = itemView.findViewById(R.id.tv_item_detail)
        var imgPhoto: ImageView = itemView.findViewById(R.id.img_item_photo)
    }
}

```

```
}  
}
```

Bedah code

```
override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
    val hero = listHero[position]  
    Glide.with(holder.itemView.context)  
        .load(hero.photo)  
        .apply(RequestOptions().override(55, 55))  
        .into(holder.imgPhoto)  
    holder.tvName.text = hero.name  
    holder.tvDetail.text = hero.detail  
}
```

Pada bagian ini kita akan menampilkan data berdasarkan posisi objeknya.

Pada bagian MainActivity

```
class MainActivity : AppCompatActivity() {  
    private lateinit var rvHeroes: RecyclerView  
    private var list: ArrayList<Hero> = arrayListOf()  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        rvHeroes = findViewById(R.id.rv_heroes)  
        rvHeroes.setHasFixedSize(true)  
  
        list.addAll(HeroesData.listData)  
        showRecyclerView()  
    }  
  
    private fun showRecyclerView() {  
        rvHeroes.layoutManager = LinearLayoutManager(this)  
        val listHeroAdapter = ListHeroAdapter(list)  
        rvHeroes.adapter = listHeroAdapter  
    }  
}
```

Bedah Code

```
private fun showRecyclerView() {  
    rvHeroes.layoutManager = LinearLayoutManager(this)  
    val listHeroAdapter = ListHeroAdapter(list)  
    rvHeroes.adapter = listHeroAdapter  
}
```

Pada bagian diatas kita menghubungkan antara adapter dengan layout.

```
list.addAll(HeroesData.listData)  
showRecyclerView()
```

Dengan memanggil function pada showRecyclerView maka recyclerview pada view akan diisi datanya dengan adapter.