

# RecyclerView

# Pengertian

RecyclerView adalah komponen View pada Android yang memungkinkan kita untuk menampilkan daftar dinamis yang kita inginkan pada aplikasi yang akan kita buat, seperti:

1. Widget untuk menampilkan daftar data.
2. "Recycles" (reuses) item views untuk membuat scrolling lebih cepat.
3. Dapat menentukan sebuah layout untuk setiap item yang ada pada dataset
4. Mendukung animasi dan transisi.

# Membuat RecyclerView

RecyclerView bisa dibuat menggunakan tiga komponen utama dari RecyclerView, yakni:

1. RecyclerView : Berfungsi untuk menampilkan daftar beserta layout XML yang akan menjadi tampilan setiap baris daftar yang kita buat.
2. Adapter : Berfungsi untuk memproses data-data yang akan ditampilkan dalam daftar yang akan dibuat.
3. LayoutManager : Berfungsi untuk mengatur tata letak daftar seperti apa yang akan kita tampilkan.

# Kegunaan

Ada RecyclerView buat menampilkan daftar, Adapter buat memproses data dan LayoutManager untuk tata letak.

Nah, tiga sekawan udah kita resapi. Biar makin familiar, kita coba praktek aja yuk!

Cara membuat RecyclerView



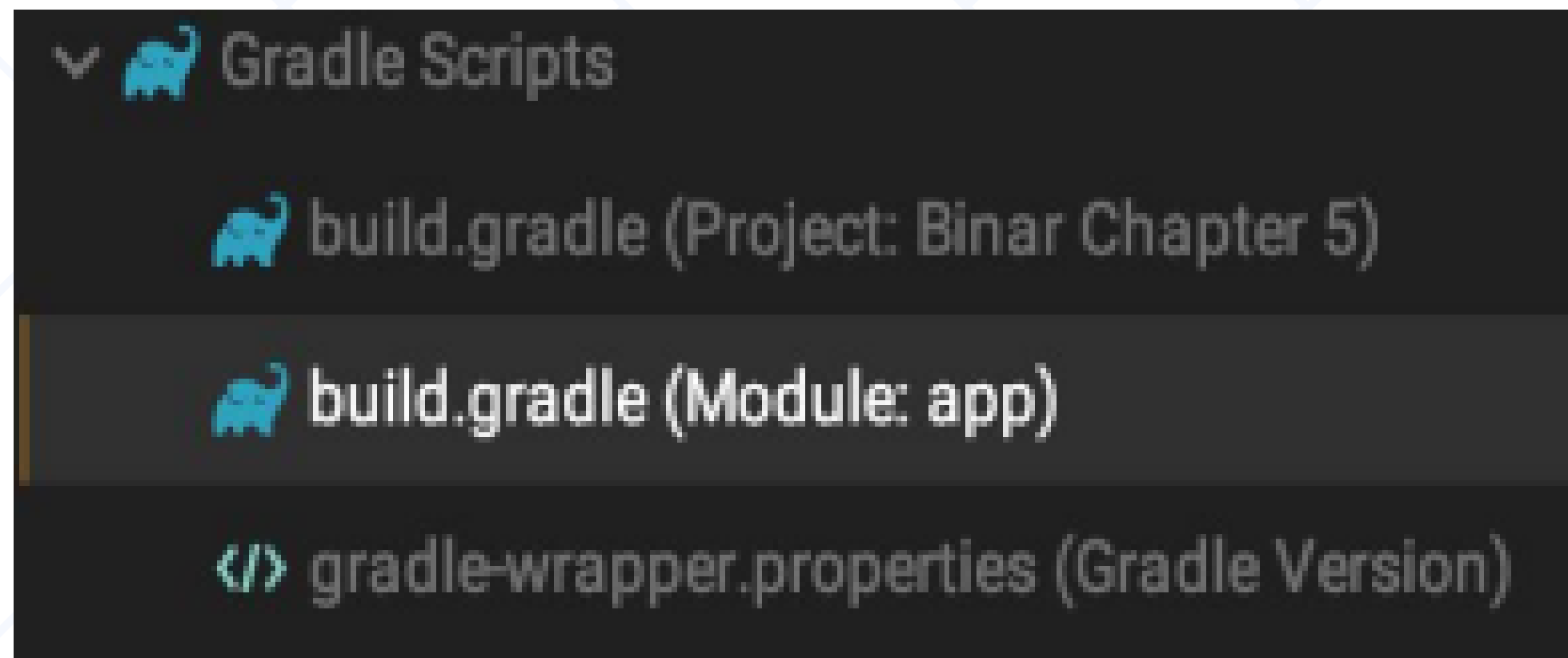
**Persiapan**

# Pertama

Pertama, sebelum kita menggunakan RecyclerView, pastikan project kita sudah menambahkan dependency RecyclerView di file Gradle level app.

## Kedua

Kedua, buka build.gradle (Module: app)



## Ketiga

Ketiga, pada bagian dependencies, tambahkan:

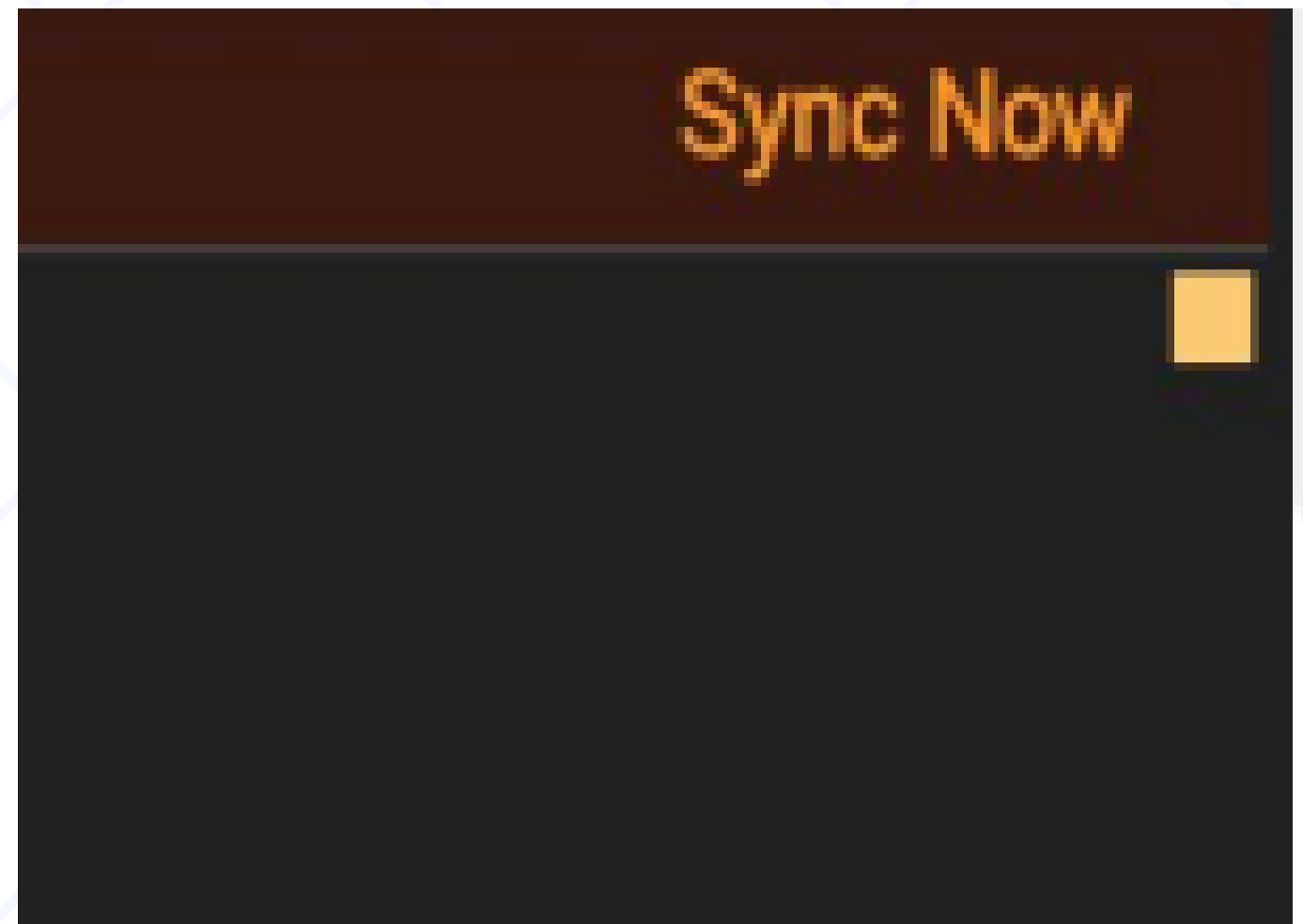
`implementation'androidx.recyclerview:recyclerview:version_code'`

```
// RecyclerView  
implementation 'androidx.recyclerview:recyclerview:1.2.0'
```



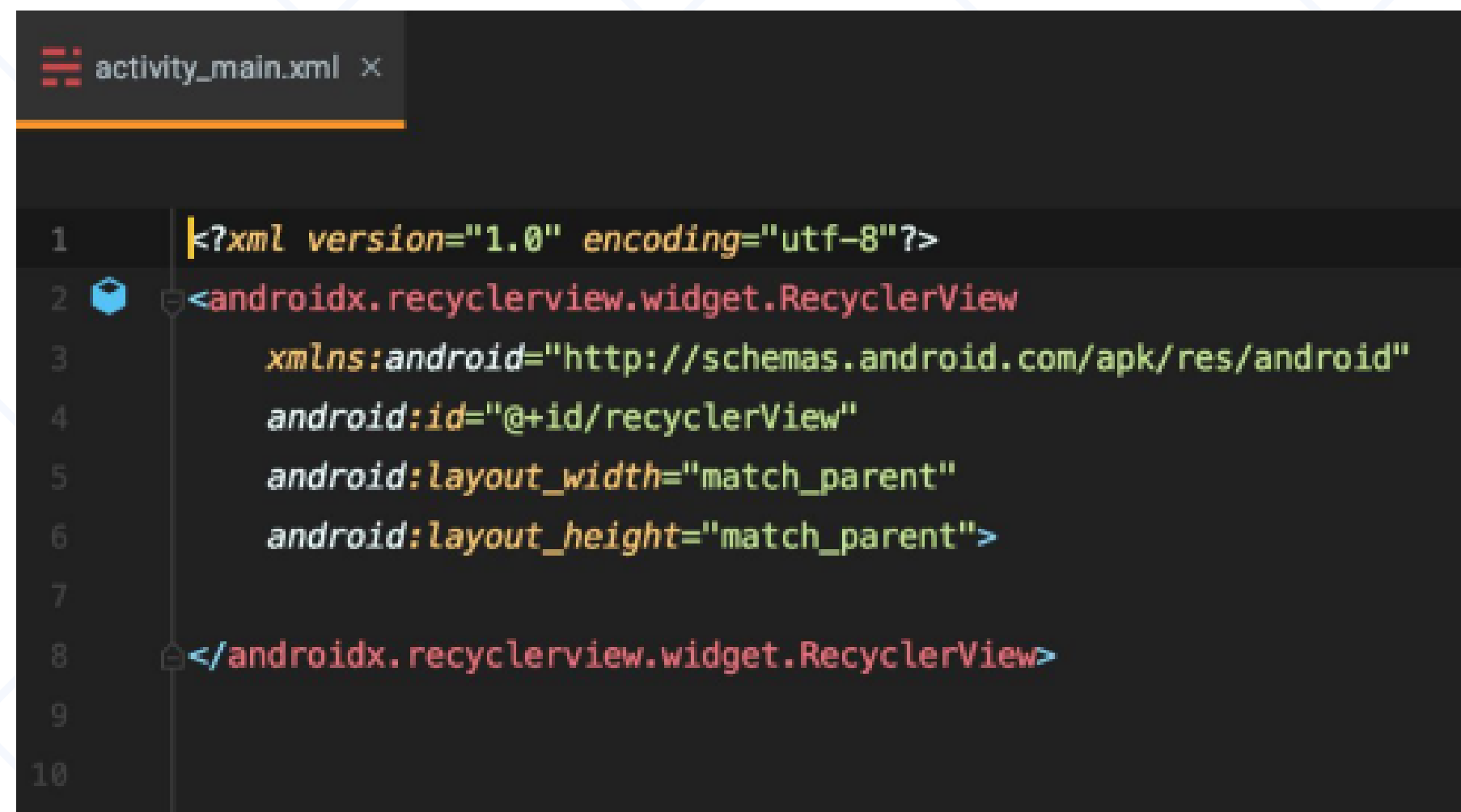
## Keempat

Keempat, klik “Sync-Now” yang muncul pada bagian atas kanan jendela build.gradle. Lalu tunggu sampai proses sync selesai.



## Kelima

Kelima, setelah dependency ditambahkan, kita bisa menggunakan View RecyclerView di Layout XML kita! jangan lupa untuk memberikan ID pada RecyclerView yang akan dibuat.

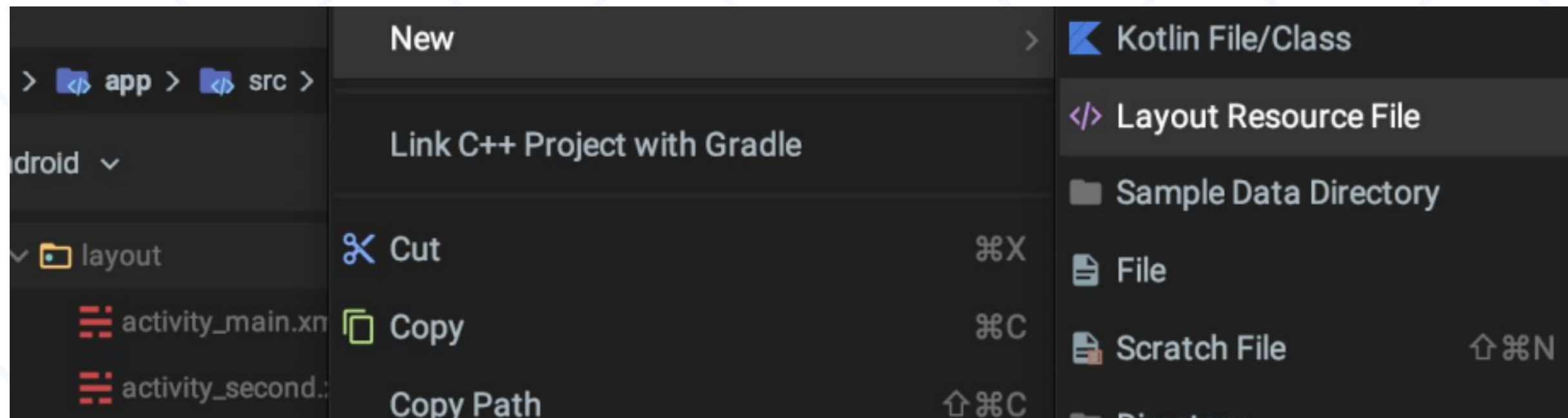


```
activity_main.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.recyclerview.widget.RecyclerView
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:id="@+id/recyclerView"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent">
7
8  </androidx.recyclerview.widget.RecyclerView>
9
10
```

Setelah buat kode RecyclerView, selanjutnya kita buat Item XML untuk layoutnya! Caranya kayak gini:

1. Klik kanan Folder Layout pada Project kita
2. Arahkan ke New
3. Klik Layout Resource File

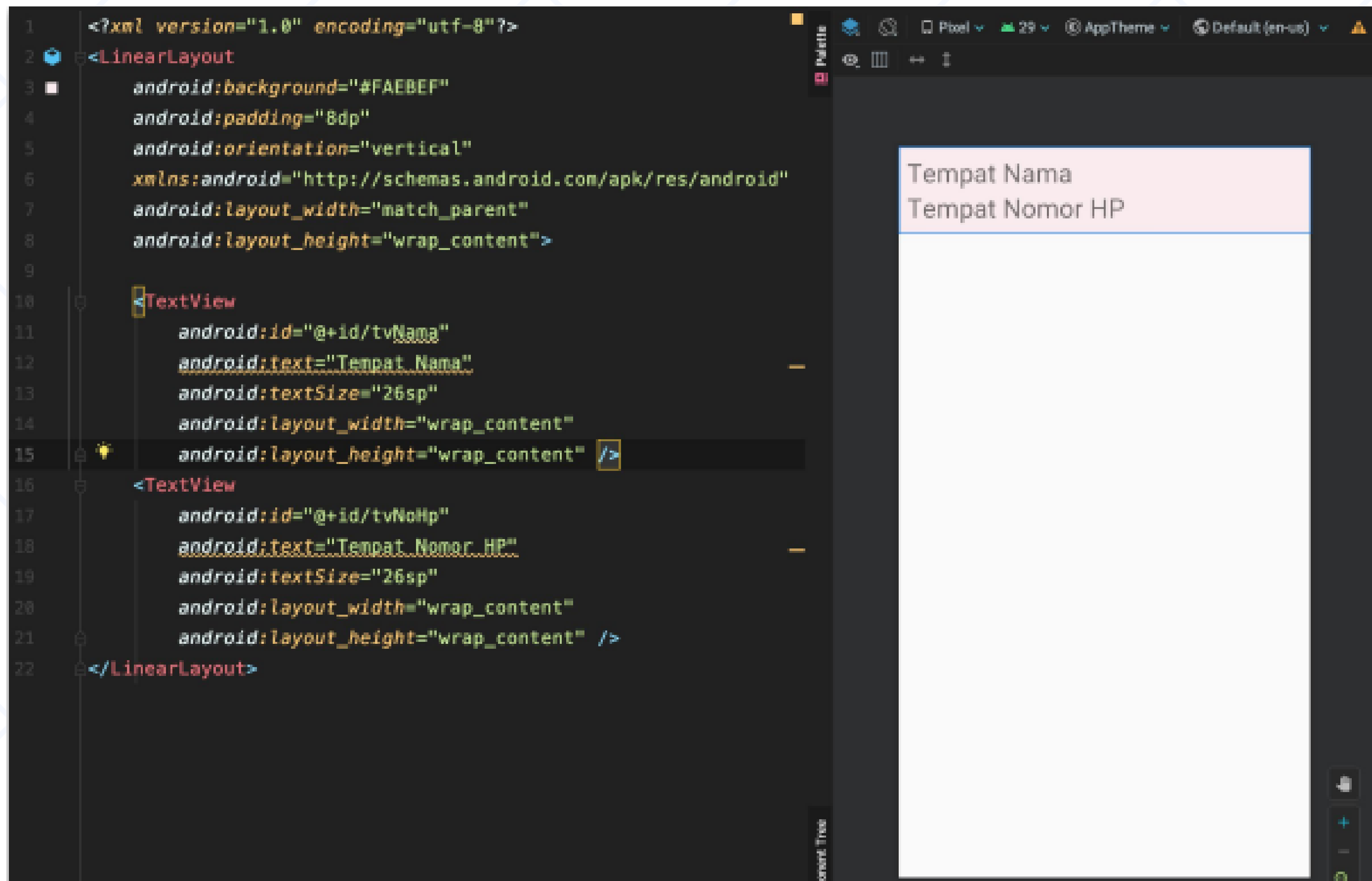
Oya, Kamu bisa memberikan nama layout XML yang diinginkan yaa. Misalnya, kita buat namanya `my_contact_item.xml`



## Ketujuh

Setelah kita kasih nama, dilanjutkan dengan membuat layout seperti gambar di samping.

Kalo udah gitu, teks Tempat Nama dan Tempat Nomor HP bakal berubah-ubah sesuai daftar data yang kita sediakan.



# Kedelapan

Caranya bikin Adapter yaitu klik kanan package tempat file yang akan dibuat

1. Arahkan ke New
2. Klik Kotlin File/Class

Kita buat nama POJO nya dengan : `MyContact.kt`

Di Kotlin, kita bisa membuat Adapter dengan menggunakan data class.

```
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class ContactAdapter(private val listContact: ArrayList<MyContact>)
    : RecyclerView.Adapter<ContactAdapter.ViewHolder>() {

    // Class Holder
    class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val nama = itemView.findViewById<TextView>(R.id.tvNama)
        val noHp = itemView.findViewById<TextView>(R.id.tvNoHp)
    }

    // Membuat Holder
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val view = LayoutInflater.from(parent.context)
            .inflate(R.layout.my_contact_item, parent, attachToRoot: false)
        return ViewHolder(view)
    }

    // Melakukan penentuan data yang akan ditampilkan pada setiap item / baris
    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        holder.nama.text = listContact[position].nama
        holder.noHp.text = listContact[position].noHp
    }

    // Memberitahu banyaknya data yang akan ditampilkan
    override fun getItemCount(): Int {
        return listContact.size
    }
}
```



# Kesembilan

Pada komponen terakhir, kita siapin `LayoutManager` pada `Activity` sebelum semua selesai.

Karena kita ingin menampilkan daftarnya sebagai daftar biasa dengan arah Vertikal, kali ini kita menggunakan `LinearLayoutManager`.

# 16

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

        val listContact = arrayListOf(
            MyContact( nama: "Sabrina", noHp: "081977823093"),
            MyContact( nama: "Pak De", noHp: "08284578987"),
            MyContact( nama: "Pak Le", noHp: "081377864586"),
            MyContact( nama: "Bu De", noHp: "087844556688"),
            MyContact( nama: "Bu Le", noHp: "08196727523"),
            MyContact( nama: "Kido", noHp: "08284578987"),
            MyContact( nama: "Sidu", noHp: "08131664586"),
            MyContact( nama: "Binarian", noHp: "087847756688"),
            MyContact( nama: "My hubby", noHp: "08187930923"),
            MyContact( nama: "Sayangku", noHp: "08284878987"),
            MyContact( nama: "Mantan", noHp: "081377873586"),
        )

        // Membuat adapter
        val adapter = ContactAdapter(listContact)

        // Membuat Layout Manager
        val layoutManager = LinearLayoutManager(context: this, LinearLayoutManager.VERTICAL, reverseLayout: false)

        // Membuat RecyclerView
        val recyclerView = findViewById<RecyclerView>(R.id.recyclerView)

        // Set LayoutManager pada RecyclerView
        recyclerView.layoutManager = layoutManager

        // Set Adapter pada RecyclerView
        recyclerView.adapter = adapter
    }
}
```

# Kesepuluh

Run Aplikasi. Kita bisa ubah ubah tampilan daftarnya menggunakan `my_contact_item.xml` untuk mempercantik tampilan daftar kita.

