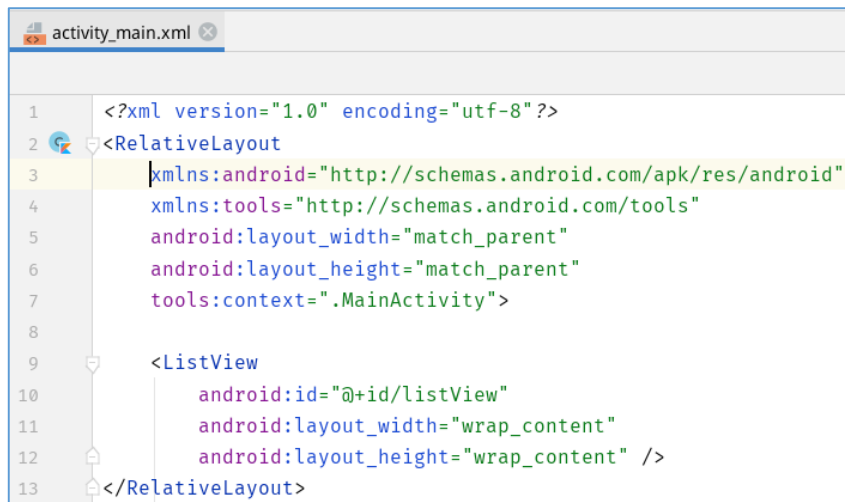


Pertemuan 9

ListView

Pada pertemuan 5, kita akan membuat sebuah aplikasi yang dapat menampilkan data dalam bentuk list. Modul ini akan mengajarkan membuat ListView dengan menggunakan metode **Simple List Item** yang menggunakan format yang telah disediakan oleh Android Studio.

Untuk mencoba, silahkan buka sebuah proyek baru bernama „**BelajarListView**“. Selanjutnya jangan lupa untuk menambahkan „**buildFeatures viewBinding**“ di dalam **build.gradle**. Kemudian silahkan buka file **activity_main.xml**. Pada file ini, kita akan menambahkan sebuah objek **ListView** ke dalam tampilan halaman aplikasi kita.



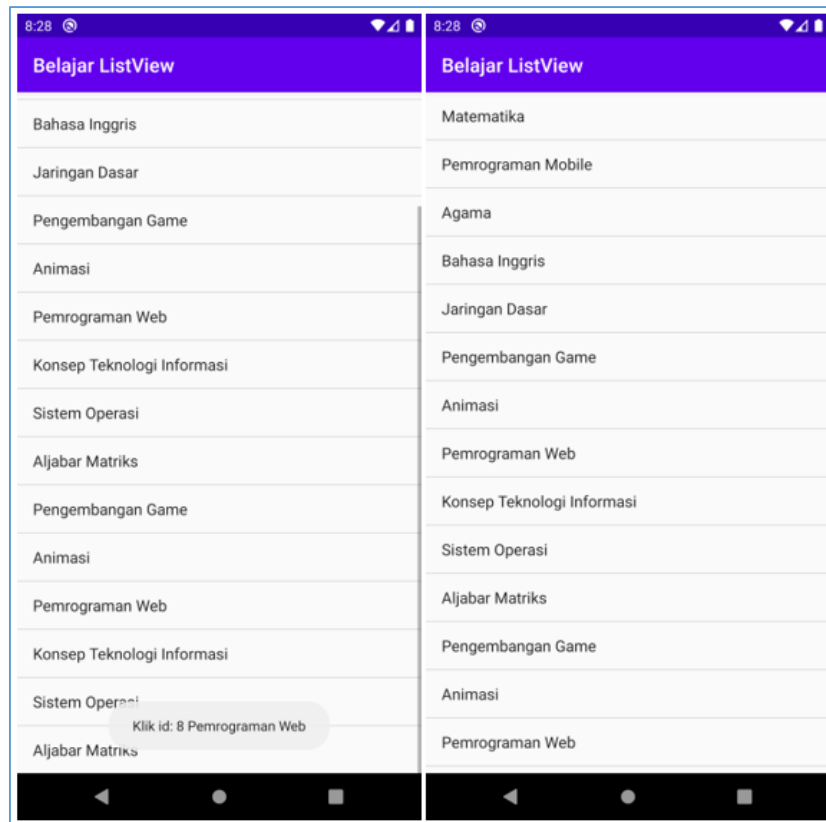
Gambar 1. Halaman activity_main.xml

Setelahnya, kita akan membuat daftar teks yang akan ditampilkan dalam bentuk **List**. Pada contoh ini menggunakan daftar Mata Kuliah yang disimpan kedalam sebuah **array** bernama **mataPelajaran**. Ketika baris list tersebut di tekan, maka akan menampilkan pop up teks yang berisi id dan juga nama mata pelajaran.

```
MainActivity.kt x
1 package com.example.simplelistview
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import android.widget.AdapterView
7 import android.widget.AdapterView.OnItemClickListener
8 import android.widget.Toast
9 import com.example.simplelistview.databinding.ActivityMainBinding
10
11 class MainActivity : AppCompatActivity() {
12     private lateinit var binding : ActivityMainBinding
13
14     val mataPelajaran = arrayOf(
15         "Matematika",
16         "Pemrograman Mobile",
17         "Agama",
18         "Bahasa Inggris",
19         "Jaringan Dasar",
20         "Pengembangan Game",
21         "Animasi",
22         "Pemrograman Web",
23         "Konsep Teknologi Informasi",
24         "Sistem Operasi",
25         "Aljabar Matriks",
26         "Pengembangan Game",
27         "Animasi",
28         "Pemrograman Web",
29         "Konsep Teknologi Informasi",
30         "Sistem Operasi",
31         "Aljabar Matriks",
32     )
33
34     override fun onCreate(savedInstanceState: Bundle?) {
35         super.onCreate(savedInstanceState)
36         binding = ActivityMainBinding.inflate(layoutInflater)
37         setContentView(binding.root)
38
39         val arrayAdapter = ArrayAdapter<String>(context: this,
40             android.R.layout.simple_list_item_1, mataPelajaran)
41         binding.listView.adapter = arrayAdapter
42
43         binding.listView.setOnItemClickListener { parent: AdapterView<*>, view: View?,
44             posisi: Int, id: Long ->
45
46             Toast.makeText(
47                 context: this, text: "Klik id: " + (posisi+1) + " " +
48                     mataPelajaran[posisi], Toast.LENGTH_SHORT
49             ).show()
50         }
51     }
52 }
```

Gambar 2. Halaman MainActivity.kt

Jika sudah selesai, maka silahkan coba jalankan aplikasi. Tampilan akan muncul seperti pada gambar di bawah ini. Ketika setiap baris di tekan, maka akan menampilkan informasi yang sudah kita set di kode program di atas ini.



Gambar 3. Hasil Aplikasi ListView

Recycler View

Beberapa tahun belakangan cara pembuatan list dengan menggunakan ListView sudah mulai ditinggalkan. Contoh di atas merupakan sebuah contoh sederhana, kode program akan menjadi lebih panjang ketika kita akan membuat modifikasi tampilan dengan menggunakan ListView. Salah satu kekurangannya adalah pada penggunaan ListView cukup memakan memori penggunaan pada aplikasi. Sebab, saat dilakukan proses scrolling menggunakan ListView, maka objek yang tidak tampil di layar akan tetap menahan memori yang dipakai. Sementara, objek yang baru terbentuk, akan mengambil memori baru.

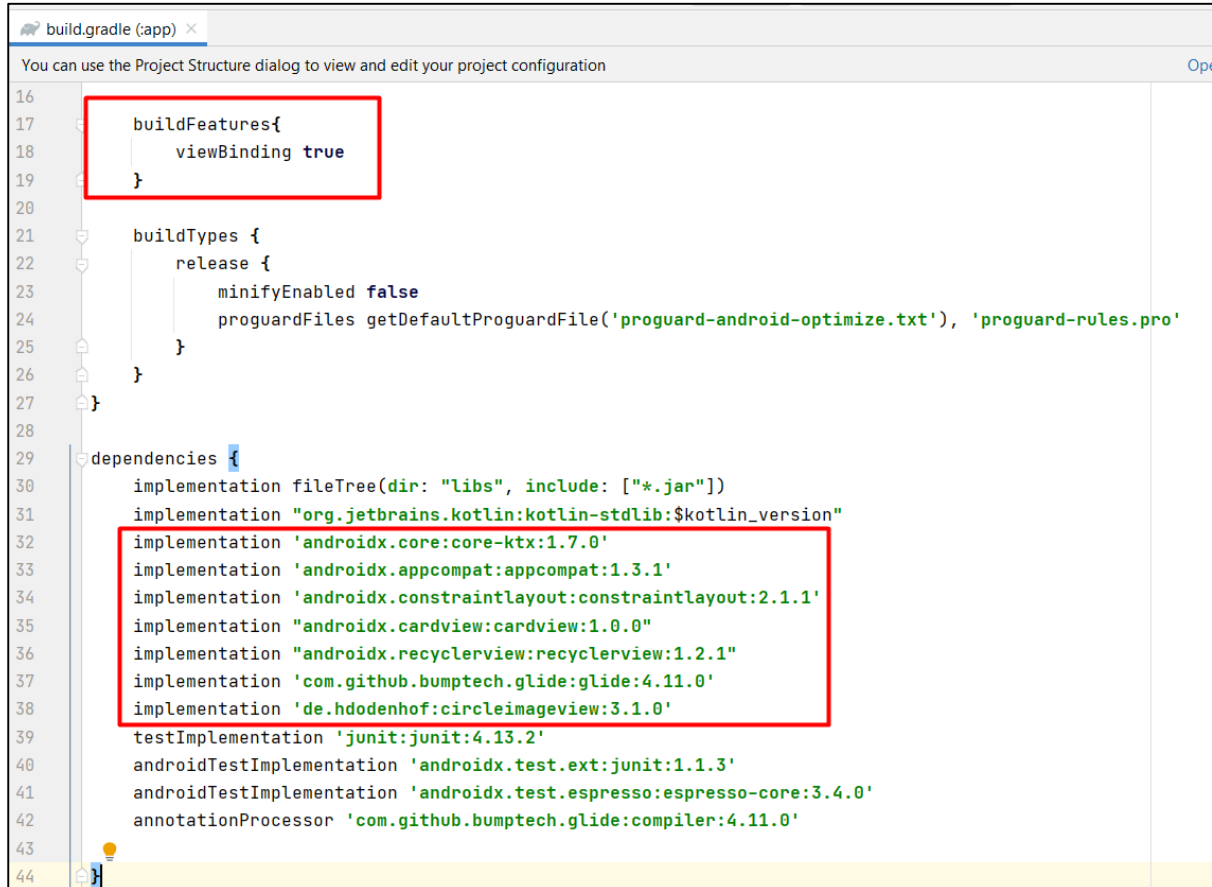
Hal ini tidak terjadi ketika menggunakan Recycler View (RV). Dengan RV, maka ketika halaman di scroll, maka objek yang tidak terlihat di layar akan melepaskan memorinya untuk dapat digunakan oleh objek lain yang terbentuk. Sehingga seberapa banyakpun objek list yang akan ditampilkan, maka jumlah memori akan tetap sama. Recycler View merupakan library resmi yang disediakan oleh Google, sehingga untuk dapat menggunakannya perlu menambahkan pada dependencies. Penulisan kode programnya juga lebih sederhana jika dibandingkan dengan penggunaan ListView.

Silahkan buat sebuah proyek baru bernama „**BelajarRecyclerView**“. Kemudian karena kita akan membutuhkan koneksi internet untuk mengambil objek gambar dari internet, maka kita perlu menambahkan hak akses untuk penggunaan internet di **AndroidManifest.xml**.



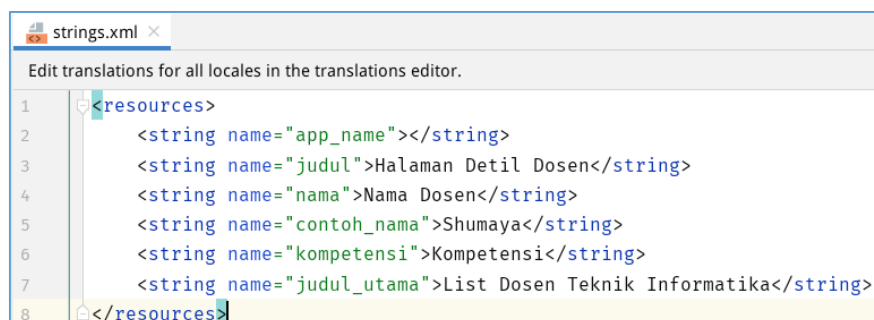
Gambar 4. Halaman AndroidManifest.xml

Selanjutnya untuk dapat menggunakan kotlin synthetic, maka kita harus menambahkan plugin **kotlin-android-extensions** pada file **build.gradle** baris kode atas. Setelah itu, tambahkan juga baris kode dependencies terutama untuk **RV, cardview, glide, dll** seperti pada gambar di bawah ini.



Gambar 5. Halaman build.gradle

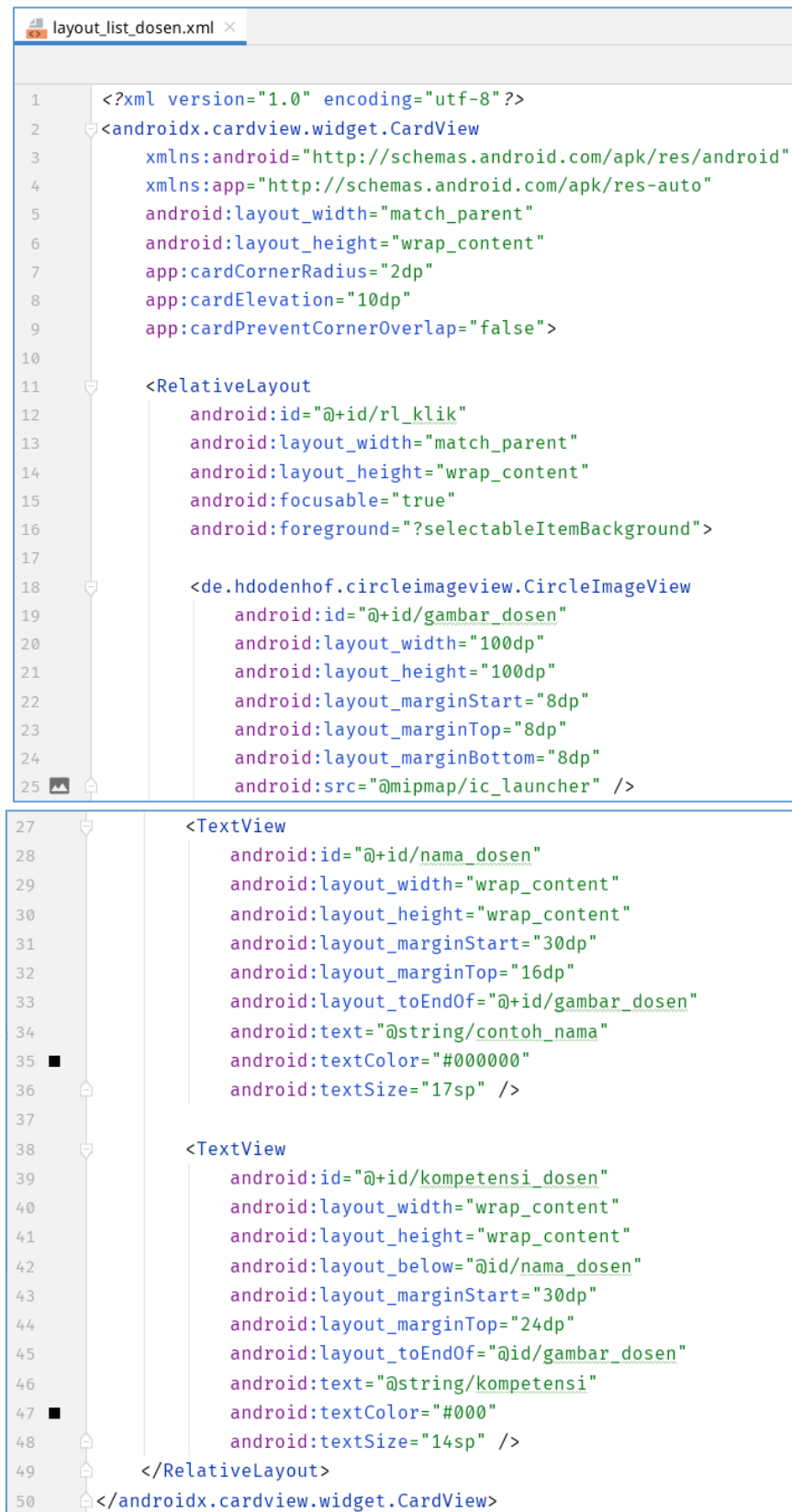
Langkah selanjutnya adalah mendaftarkan teks ke dalam file **strings.xml** untuk beberapa objek yang digunakan dalam halaman aplikasi.



Gambar 6. Halaman strings.xml

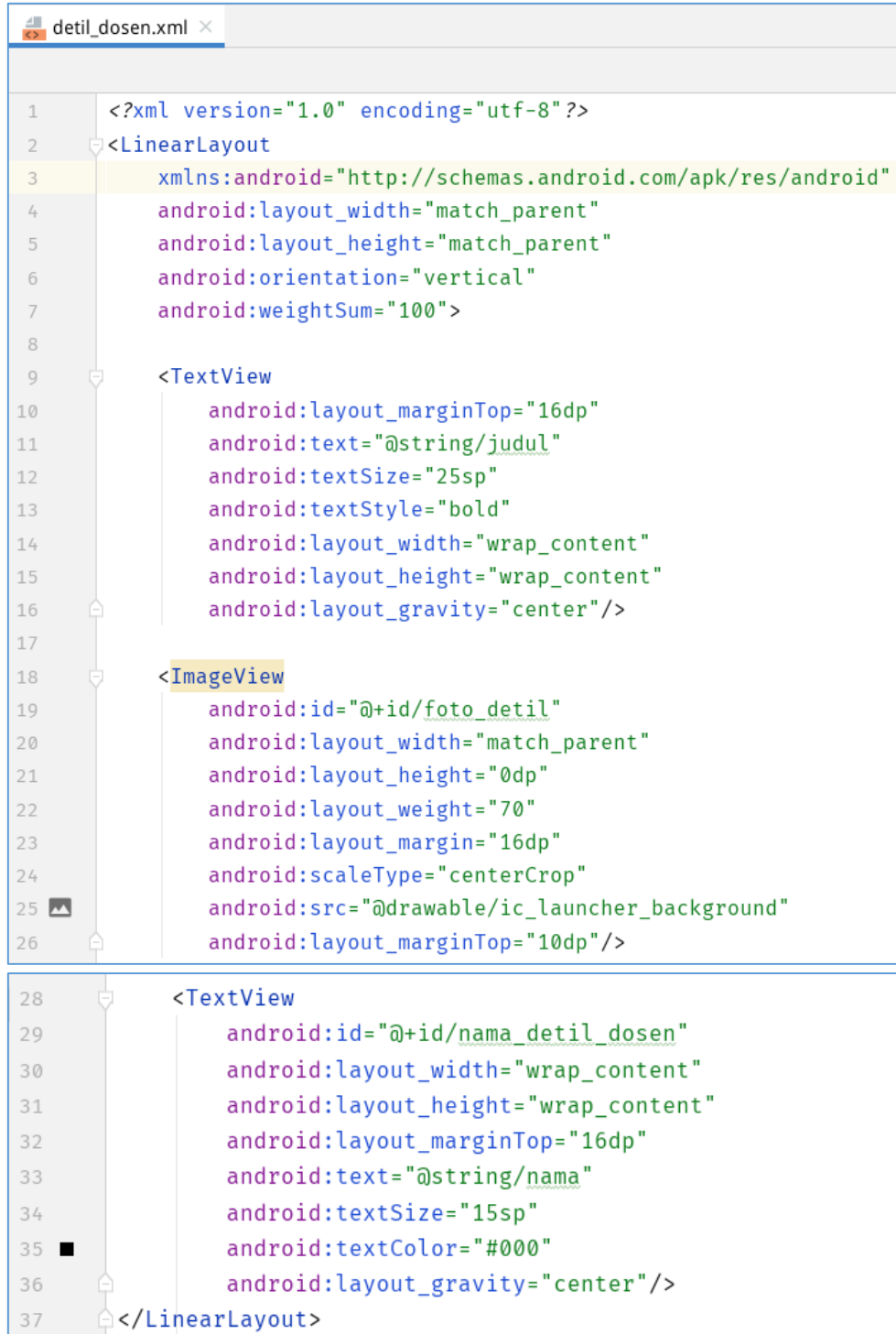
Langkah selanjutnya adalah membuat sebuah layout baru. Tambahkan sebuah layout kedalam folder **Layout** dan beri nama **layout_list_dosen.xml**. Halaman ini merupakan tampilan UI berbentuk List yang di modifikasi. Kita akan menambahkan sebuah Image Profile dengan menggunakan library

circleImageView untuk memperindah tampilan aplikasi. Silahkan coba tambahkan kode xml seperti pada gambar di bawah ini.



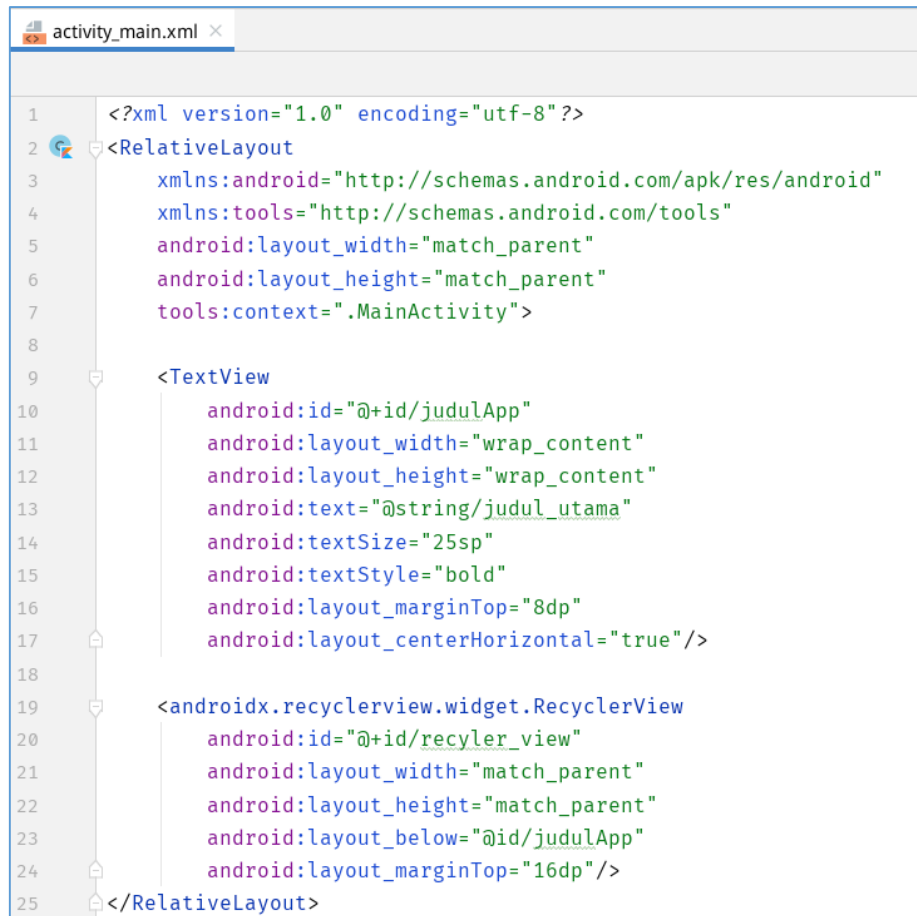
Gambar 7. Halaman layout_list_dosen.xml

Selanjutnya kita akan membuat tampilan halaman detail pada saat list dosen tersebut ditekan. Pada halaman ini, kita akan menambahkan ImageView untuk menampilkan gambar dosen secara penuh. Selain itu juga akan menambahkan sebuah TextView untuk menampilkan nama dosen tersebut.



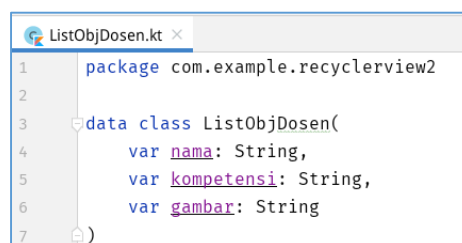
Gambar 8. Halaman detil_dosen.xml

Terakhir kita akan mengimplementasikan sebuah ViewGroup bernama **Recycler View**. RV lah yang akan menampung list yang tadi telah kita modifikasi dan menampilkan data yang sudah di set pada file kotlin nya. Silahkan buka file **activity_main.xml**, kemudian tambahkan kode xml berikut ini.



Gambar 9. Halaman activity_main.xml

Setelah selesai memodifikasi halaman tampilan, langkah selanjutnya adalah membuat kode kotlin untuk logika programnya. Sebelumnya, kita akan mendefinisikan objek yang akan kita tampilkan pada aplikasi, yaitu nama dosen, kompetensi dan foto dari dosen tersebut. Silahkan tambahkan sebuah file kotlin baru, beri nama **ListObjDosen.kt**. Kemudian daftarkan datanya.



Gambar 10. Halaman ListObjDosen.kt

Setelah itu, kita akan menuliskan sumber data yang akan ditampilkan. Biasanya anda dapat mengambil data melalui API. Akan tetapi pada contoh ini, kita akan melakukan set statis untuk *source* yang kita gunakan pada aplikasi. Silahkan tambah sebuah file kotlin baru bernama **"SumberData.kt"**. Kemudian anda bisa *copy paste* kode program di bawah ini.


```
package com.example.recyclerview2

class SumberData {
    companion object {

        fun buatSetData(): ArrayList<ListObjDosen> {
            val list = ArrayList<ListObjDosen>()
            //list bisa didapatkan/copas di web pcr ya.
            https://pcr.ac.id/prodi/301/teknik-informatika/profil
            list.add(
                ListObjDosen(
                    "Ananda, S.Kom,M.T.",
                    "Digital Image Processing",
                    "https://pcr.ac.id/assets/images/pegawai/AND20170207022257.jpg"
                )
            )
            list.add(
                ListObjDosen(
                    "Silvana Rasio Henim, S.ST, M.T.",
                    "Programming",
                    "https://pcr.ac.id/assets/images/pegawai/SRH20190718105457.jpg"
                )
            )
            list.add(
                ListObjDosen(
                    "Agus Urip Ari Wibowo, S.T.,M.T.",
                    "IoT",
                    "https://pcr.ac.id/assets/images/pegawai/AUA20190718114257.jpg"
                )
            )
            list.add(
                ListObjDosen(
                    "Ardianto Wibowo, S.Kom.,M.T.",
                    "Data Engineering",
                    "https://pcr.ac.id/assets/images/pegawai/ARW20190718113926.jpg"
                )
            )
            list.add(
                ListObjDosen(
                    "Erwin Setyo Nugroho, S.T.,M.Eng",
                    "Computer Networking & Administration",
                    "https://pcr.ac.id/assets/images/pegawai/ESN20190718113952.jpg"
                )
            )
            list.add(
                ListObjDosen(
                    " Ibnu Surya, S.T.,M.T.",
                    "Operating System",
                    "https://pcr.ac.id/assets/images/pegawai/ISA20190718114016.jpg"
                )
            )
        }
    }
}
```

Gambar 11. Halaman SumberData

```
list.add(
    ListObjDosen(
        "Kartina Diah Kusuma Wardhani, S.T.,M.T.",
        "Artificial Intelligent",
        "https://pcr.ac.id/assets/images/pegawai/DYH20170214111439.jpg"
    )
)
list.add(
    ListObjDosen(
        "Maksum Ro'is Adin Saf, S.Kom., M.Eng.",
        "Web Semantik",
        "https://pcr.ac.id/assets/images/pegawai/MRA20190718114446.jpg"
    )
)
list.add(
    ListObjDosen(
        "Meilany Dewi, S.T.,M.T.",
        "Computer Animation",
        "https://pcr.ac.id/assets/images/pegawai/MEL20190718114526.jpg"
    )
)
list.add(
    ListObjDosen(
        "Muhammad Arif Fadhly Ridha, S.Kom.,M.T.",
        "Cloud Computing",
        "https://pcr.ac.id/assets/images/pegawai/MAF20190718114601.jpg"
    )
)
list.add(
    ListObjDosen(
        "Muhammad Ihsan Zul, S.Pd., M.Eng.",
        "Machine Learning",
        "https://pcr.ac.id/assets/images/pegawai/MIZ20190718115103.jpg"
    )
)
list.add(
    ListObjDosen(
        "Rahmat Suhatman, S.T.M.T.",
        "Certified Ethical Hacking",
        "https://pcr.ac.id/assets/images/pegawai/RMT20190718113858.jpg"
    )
)
list.add(
    ListObjDosen(
        "Rika Perdana Sari, S.T.,M.Eng.",
        "Management System Information",
        "https://pcr.ac.id/assets/images/pegawai/RPS20190718114751.jpg"
    )
)
)
```

```
        list.add(
            ListObjDosen(
                "Shumaya Resty Ramadhani, S.ST., M.Sc.",
                "Mobile and Desktop Technology",
                "https://pcr.ac.id/assets/images/pegawai/SRR20190718114820.jpg"
            )
        )
        list.add(
            ListObjDosen(
                "Dr. Juni Nurma Sari, S.Kom.,M.MT.",
                "Programming",
                "https://pcr.ac.id/assets/images/pegawai/JNS20190718115229.jpg"
            )
        )
        list.add(
            ListObjDosen(
                "Yuliska, S.T., M.Eng.",
                "Natural Language Processing",
                "https://pcr.ac.id/assets/images/pegawai/YLK20191106020236.jpg"
            )
        )
        list.add(
            ListObjDosen(
                "Khairul Umam Syaliman, S.T., M.Kom.",
                "Data Mining",
                "https://pcr.ac.id/assets/images/pegawai/KUS20191106020716.jpg"
            )
        )
        list.add(
            ListObjDosen(
                "Jan Alif Kreshna, S.ST., M.Sc.",
                "Artificial Intelligent",
                "https://pcr.ac.id/assets/images/pegawai/JAK20170214113137.jpg"
            )
        )
        return list
    }
}
```

Setelah itu, agar gambar yang tampil pada list yang kita gunakan menjadi lebih rapi dan tidak berdempet satu sama lain, maka kita dapat menambahkan jarak atau dekorasi terhadap RV. Silahkan buat sebuah class baru bernama **"DekorasiSpasiGambar.kt"**. Kemudian tambahkan kode berikut ini.

```
DekorasiSpasiGambar.kt
1 package com.example.recyclerview2
2
3 import android.graphics.Rect
4 import android.view.View
5 import androidx.recyclerview.widget.RecyclerView
6
7 class DekorasiSpasiGambar(private val padding: Int):
8     RecyclerView.ItemDecoration() {
9     override fun getItemOffsets(
10         outRect: Rect,
11         view: View,
12         parent: RecyclerView,
13         state: RecyclerView.State
14     ) {
15         super.getItemOffsets(outRect, view, parent, state)
16         outRect.top = padding
17     }
18 }
```

Gambar 12. Halaman DekorasiSpasiGambar.kt

Kemudian, tambah lagi sebuah file baru bernama **“DetilDosen.kt”**. Halaman ini akan berisi logika program yang mengatur proses tampilnya data dosen berdasarkan list mana yang dipilih. Silahkan tambahkan kode program di bawah ini.

```
DetilDosen.kt
1 package com.example.recyclerview2
2 import android.os.Bundle
3 import androidx.appcompat.app.AppCompatActivity
4 import com.bumptech.glide.Glide
5 import com.bumptech.glide.request.RequestOptions
6 import com.example.recyclerview2.databinding.DetilDosenBinding
7
8 class DetilDosen : AppCompatActivity() {
9     private lateinit var binding: DetilDosenBinding
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        binding = DetilDosenBinding.inflate(layoutInflater)
13        val view = binding.root
14        setContentView(view)
15        if (intent.hasExtra( name: "namanya")) {
16            val nama: String = this.intent.getStringExtra( name: "namanya").toString()
17            val foto: String = this.intent.getStringExtra( name: "fotonya").toString()
18            setDetil(foto, nama)
19        }
20    }
21    fun setDetil(foto: String, nama: String) {
22        val requestOp = RequestOptions()
23            .placeholder(R.drawable.ic_launcher_background)
24            .error(R.drawable.ic_launcher_background)
25
26        binding.namaDetilDosen.text = nama
27        Glide.with( activity: this)
28            .load(foto)
29            .apply(requestOp)
30            .centerCrop()
31            .into(binding.fotoDetil)
32    }
33 }
```

Gambar 13. Halaman DetilDosen.kt

Selanjutnya kita akan membuat sebuah file adapter yang berfungsi sebagai jembatan antara view dan datanya. Silahkan buat sebuah file kotlin baru bernama **“DoseRecyclerAdapter.kt”**, kemudian tambahkan kode program berikut ini. Pada halaman ini kita juga akan memberikan aksi

ketika list tersebut ditekan dan kemana akan diarahkan. Ada data yang bergerak juga pada proses tersebut.

```
DosenRecyclerAdapter.kt x
1  package com.example.recyclerview2
2
3  import android.content.Intent
4  import android.view.LayoutInflater
5  import android.view.ViewGroup
6  import android.widget.ImageView
7  import android.widget.RelativeLayout
8  import android.widget.TextView
9  import android.widget.Toast
10 import androidx.recyclerview.widget.RecyclerView
11 import com.bumptech.glide.Glide
12 import com.bumptech.glide.request.RequestOptions
13 import com.example.recyclerview2.databinding.LayoutListDosenBinding
14
15 class DosenRecyclerAdapter : RecyclerView.Adapter<RecyclerView.ViewHolder>() {
16     private var items: List<ListObjDosen> = ArrayList()
17
18     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): DosenViewHolder {
19         val binding = LayoutListDosenBinding.inflate(LayoutInflater.from(parent.context),
20             parent, attachToParent: false)
21         return DosenViewHolder(binding)
22     }
23
24     override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
25         when (holder) {
26             is DosenViewHolder -> {
27                 holder.bind(items.get(position))
28                 holder.klik.setOnClickListener { it: View!
29                     holder.kalau_diklik(items.get(position))
30                 }
31             }
32         }
33     }
34 }
```

```
35 fun submitList(listDosen: List<ListObjDosen>) {  
36     items = listDosen  
37 }  
38 override fun getItemCount(): Int {  
39     return items.size  
40 }  
41  
42 class DosenViewHolder constructor(val binding: LayoutListDosenBinding) :  
43     RecyclerView.ViewHolder(binding.root) {  
44  
45     val foto_dosen: ImageView = binding.gambarDosen  
46     val nama_dosen: TextView = binding.namaDosen  
47     val kom_dosen: TextView = binding.kompetensiDosen  
48     var klik: RelativeLayout = binding.rlKlik  
49  
50     fun bind(listObjDosen: ListObjDosen) {  
51         nama_dosen.setText(listObjDosen.nama)  
52         kom_dosen.setText(listObjDosen.kompetensi)  
53  
54         val requestOp = RequestOptions()  
55         .placeholder(R.drawable.ic_launcher_background)  
56         .error(R.drawable.ic_launcher_background)  
57  
58         Glide.with(itemView.context)  
59             .applyDefaultRequestOptions(requestOp)  
60             .load(listObjDosen.gambar)  
61             .into(foto_dosen)  
62     }  
63  
64     fun kalau_diklik(get: ListObjDosen) {  
65         Toast.makeText(itemView.context, text: "Kamu memilih : ${get.nama}",  
66             Toast.LENGTH_SHORT)  
67         .show()  
68  
69         val intent = Intent(itemView.context, DetilDosen::class.java)  
70         intent.putExtra(name: "namanya", get.nama)  
71         intent.putExtra(name: "kompetensinya", get.kompetensi)  
72         intent.putExtra(name: "fotonya", get.gambar)  
73         itemView.context.startActivity(intent)  
74     }  
75 }  
76 }  
77 }
```

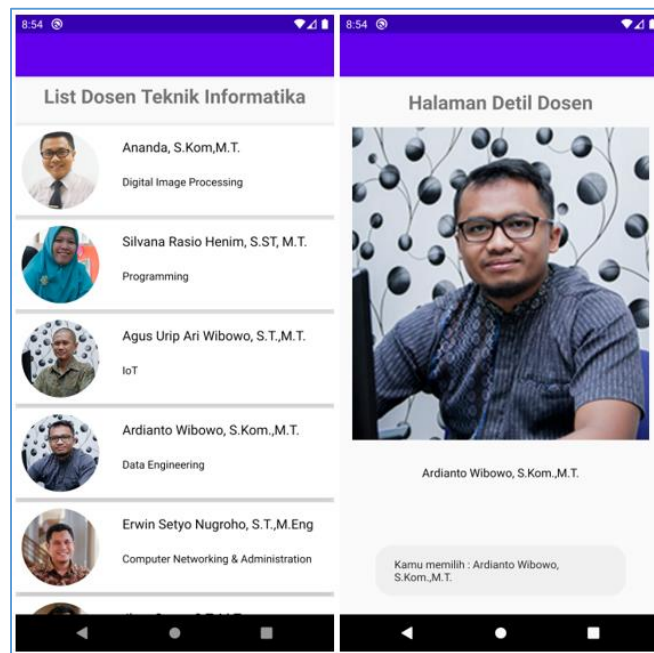
Gambar 14. Halaman DosenRecyclerViewAdapter.kt

Terakhir silahkan buka file **MainActivity.kt** untuk menambahkan dataset yang sudah terbentuk tadi untuk ditampilkan ke halaman utama. Silahkan tambahkan kode program di bawah ini.

```
MainActivity.kt
1 package com.example.recyclerview2
2 import androidx.appcompat.app.AppCompatActivity
3 import android.os.Bundle
4 import androidx.recyclerview.widget.LinearLayoutManager
5 import com.example.recyclerview2.databinding.ActivityMainBinding
6
7 class MainActivity : AppCompatActivity() {
8     private lateinit var dosenAdapter: DosenRecylcerAdapter
9     private lateinit var binding : ActivityMainBinding
10
11     override fun onCreate(savedInstanceState: Bundle?) {
12         super.onCreate(savedInstanceState)
13         binding = ActivityMainBinding.inflate(layoutInflater)
14         setContentView(binding.root)
15         initRecyclerView()
16         tambahDataSet()
17     }
18
19     private fun tambahDataSet() {
20         val data = SumberData.buatSetData()
21         dosenAdapter.submitList(data)
22     }
23
24     private fun initRecyclerView() {
25         binding.recyclerview.apply { this: RecyclerView
26             layoutManager = LinearLayoutManager( context: this@MainActivity)
27             val spacingAtas = DekorasiSpasiGambar( padding: 20)
28             addItemDecoration(spacingAtas)
29             dosenAdapter = DosenRecylcerAdapter()
30             adapter = dosenAdapter
31         }
32     }
33 }
```

Gambar 15. Halaman MainActivity.kt

Silahkan jalankan aplikasi anda. Jika berhasil, maka akan muncul tampilan seperti di bawah ini. Jika list nya di klik maka akan membuka sebuah halaman baru berisi detail dari dosen tersebut.



Gambar 16. Halaman Hasil Aplikasi

Latihan Aplikasi

1. Silahkan tambahkan detil dosen berupa **NIP dosen**, **Nama MK** yang di ampu, dan **ruangan** dosen tersebut di laporan yang akan kalian kumpulkan (ini ditambahkan pada proyek „**BelajarRecyclerView**” pada materi di bawah ini).
2. Buat sebuah aplikasi android kotlin sederhana untuk membuat katalog film yang menampilkan list movie seperti pada contoh dibawah. Terdapat poster, judul, rating, dan nama pemain. Beri nama Proyek “**LatihanMovieKatalog**”. Kemudian ketika salah satu item film di klik maka akan masuk ke halaman yang berisi detil informasi seperti poster (berukuran lebih besar), durasi, genre, Bahasa, synopsis, dan list pemeran. Minimal ada 6 film yang lengkap informasinya. Silahkan cari gambar dan informasi melalui internet. Buat tampilan dengan UI yang menarik. **Latihan ini silahkan di upload projeknya di classroom.**

