

LAPORAN PRAKTIKUM
PENGEMBANGAN APLIKASI BERGERAK
PRAKTIKUM 05 – RECYCLER VIEW



L0122034
BINTANG HARIDA RAMADHAN

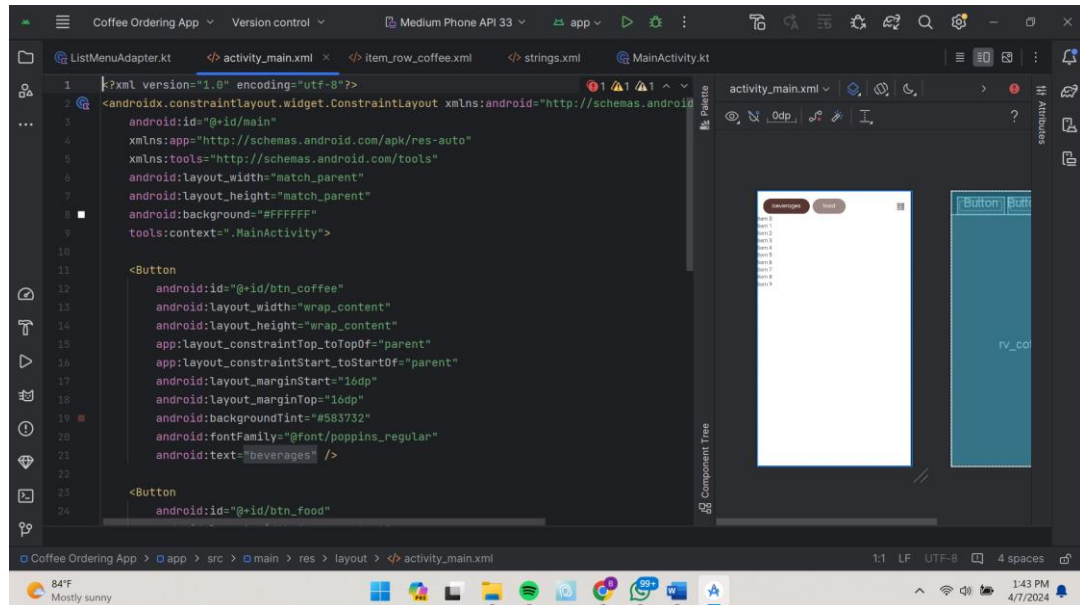
PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET

2024

BAB I

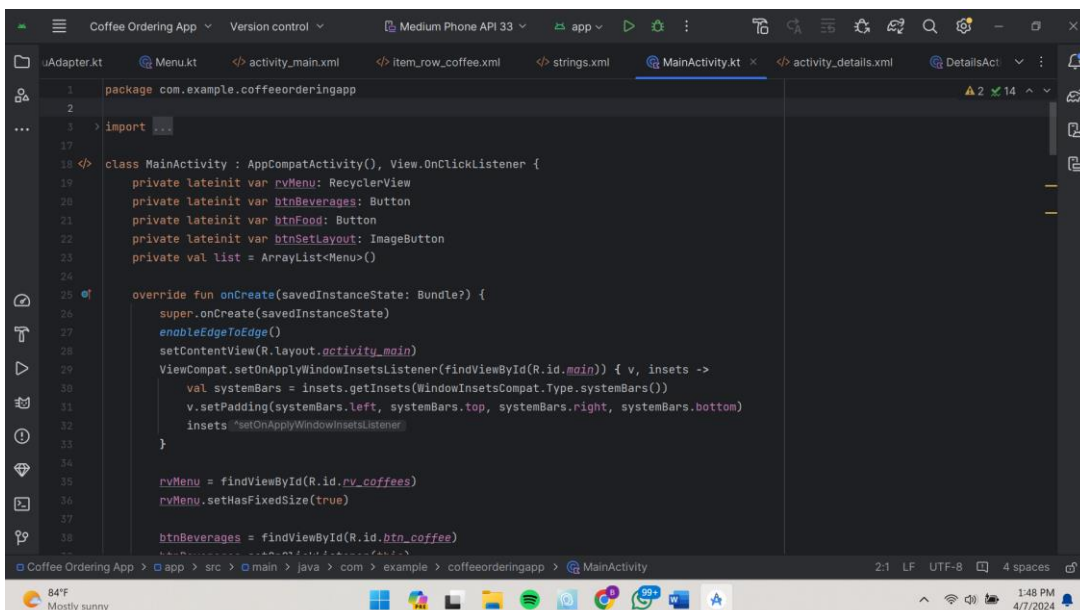
SOURCE CODE

1. File “activity_main.xml”



Halaman ini dibuat menggunakan constraint layout dengan tujuan untuk memudahkan dalam tata letak komponen di dalamnya. Pada bagian ini diberikan beberapa tombol yaitu tombol untuk melihat menu makanan, tombol menu minuman, dan tombol untuk merubah layout. Ketiga tombol ini disusun sejajar secara vertikal pada bagian atas dengan constraint yang sesuai. Untuk tombol layout didapatkan dari google icon yang kemudian disimpan pada folder *drawable*. Di bawah ketiga tombol tersebut diberikan *RecyclerView* yang nantinya akan menampilkan list menu.

2. File “MainActivity.kt”



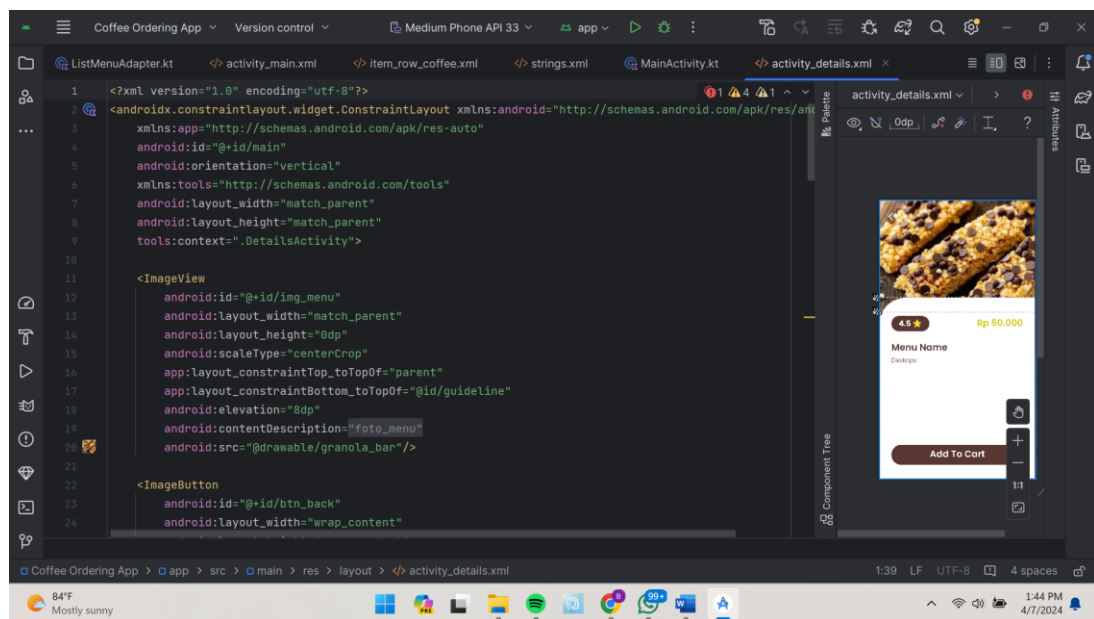
Pada bagian awal definisi class dideklarasikan beberapa *lateinit variable* untuk menampung view yang ada di file “activity_main.xml” sesuai dengan tipe view masing masing, selain itu juga dideklarasikan array yang akan digunakan untuk menampung list menu. Masing masing *variable view* diinisialisasikan dengan memanggil fungsi *findViewById()* yang disertai dengan argumen *id* yang sesuai dengan masing masing view. Untuk *Button* diimplementasikan method *setOnClickListener(this)* yang akan digunakan untuk penentuan event saat tombol tersebut ditekan. Setelah itu, *variable* “list” akan diisi dengan hasil *return value* dari fungsi *getCoffeeList()* sebagai default, karena disini terdapat menu minuman dan makanan. Terakhir, akan dipanggil fungsi *showRecyclerView()*.

Pada fungsi *getCoffeeList()* dan *getFoodList()* diinisialisasikan beberapa *variable* untuk menampung nama menu, harga, deskripsi, dan foto dengan mendapatkannya dari resource dengan *id* yang sesuai. Selanjutnya akan dilakukan looping dari seluruh data tersebut untuk dimasukkan pada list. Terakhir, list tersebut akan digunakan sebagai *return value* yang akan digunakan untuk list yang ada di fungsi utama *onCreate()*. Pada fungsi *showRecyclerView()* akan diatur layoutnya dengan *layoutManager* dan menginisialisasikan adapter yang akan diimplementasikan oleh *recycler view*.

Pada fungsi *onClick* akan diberikan masing masing event bergantung pada tombol mana yang ditekan. Untuk tombol makanan, warna background dari tombol akan ditukar dengan warna yang sedang aktif. Kemudian akan menghapus seluruh list saat ini dan memberikan list yang baru dengan *return value* dari fungsi *getFoodList()* untuk mendapatkan list makanan. Terakhir, akan dipanggil fungsi *showRecyclerView*. Begitupun sebaliknya pada tombol minuman, yang akan memberikan list baru dengan *return value* dari fungsi *getCoffeeList()*.

Untuk *listener* dari tombol set layout akan dilakukan percabangan terlebih dahulu untuk mengecek apakah layout saat ini grid atau bukan, jika iya icon dari set layout akan dirubah menjadi grid icon. Setelah itu layout saat ini juga akan dirubah menjadi linear. Untuk merapikan tampilan dari masing masing card juga dilakukan perubahan constraint dari *TextView* menggunakan fungsi *addOnChildAttachStateChangeListener()* yang akan otomatis dipanggil jika terjadi perubahan keadaan.

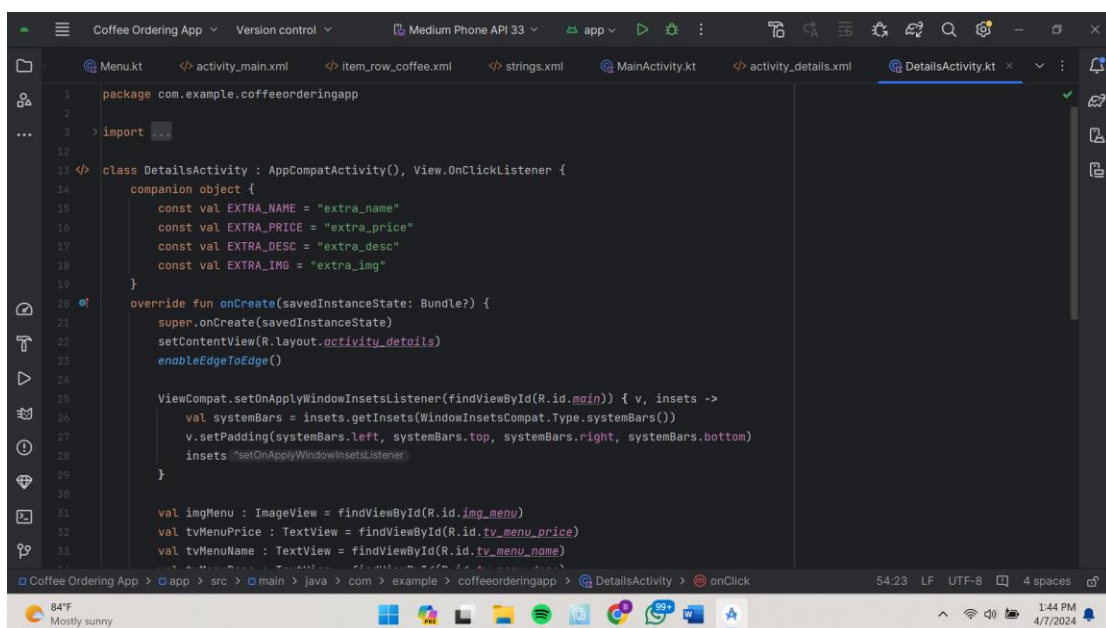
3. File “activity_details.xml”



Pada halaman ini dibuat menggunakan *constraint layout* yang berisikan view untuk menampilkan gambar, tombol kembali, harga, nama menu, deskripsi, rating menu, dan tombol *add to cart*. Constrain dari masing masing *view* tersebut disesuaikan dengan tata letak yang diinginkan. Pada halaman ini, tombol *add to cart* belum difungsikan, jadi hanya sebatas tampilan saja.

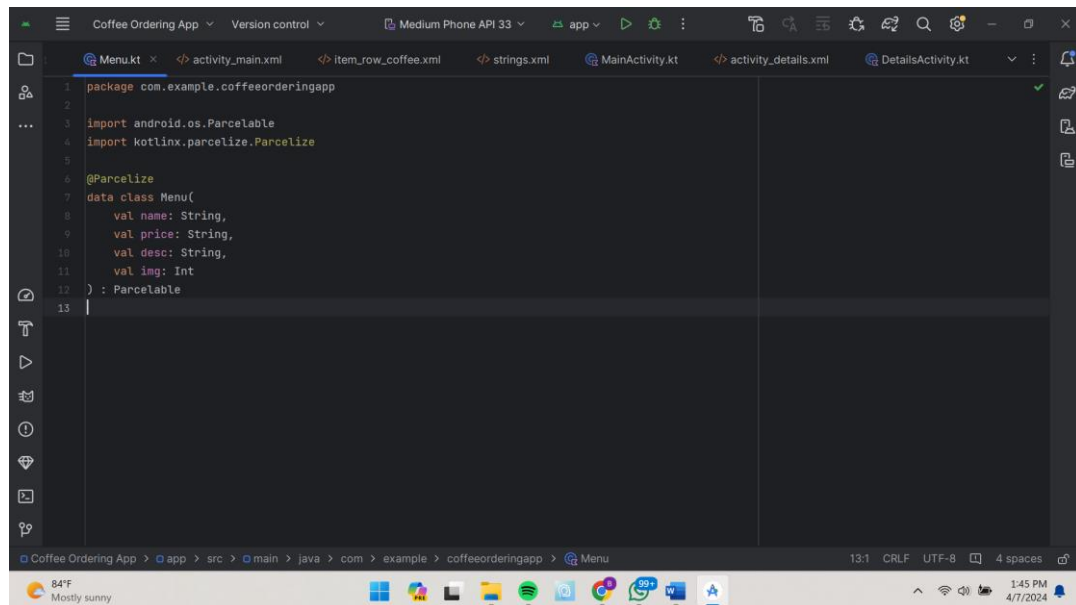
Pada bagian ini juga digunakan *Guideline* untuk menentukan persentase tinggi layar antara *ImageView* dengan *constraint layout details info*. Dengan ini, tinggi dari masing masing *view* akan disesuaikan dengan tinggil layar pengguna. Pada bagian foto diberikan nilai 40% dari tinggi layar dan sisanya akan digunakan untuk menampilkan detail info.

4. File “DetailsActivity.kt”



Pada awal definisi class ini dilakukan inisialisasi *companion object* untuk menerima *intent* dari *adapter*. Selanjutnya, diinisialisasikan variable untuk menampung view yang ada di “activity_details.xml” sesuai dengan *id* masing masing view. Data dari intent akan diterima menggunakan method *intent.getStringExtra()* kemudian disimpan ke masing masing variable data. Variable *view* akan diinisialisasikan dengan value dari variable data yang didapat dari intent tersebut. Untuk *onClickListener* dari tombol kembali akan dilakukan pemanggilan fungsi *finish()*.

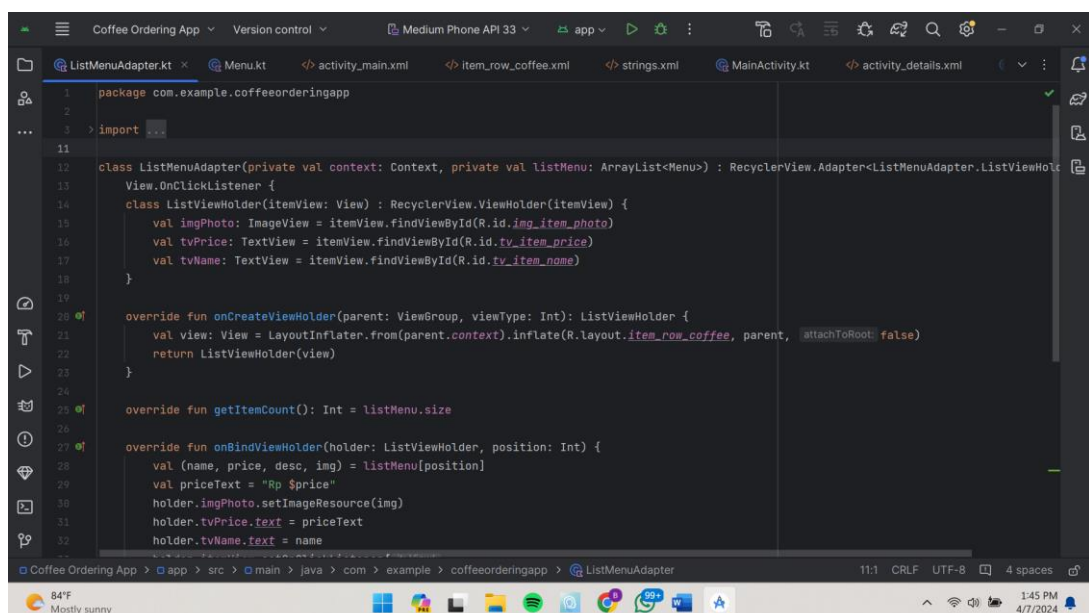
5. File “Menu.kt”



```
1 package com.example.coffeeorderingapp
2
3 import android.os.Parcelable
4 import kotlinx.parcelize.Parcelize
5
6 @Parcelize
7 data class Menu(
8     val name: String,
9     val price: String,
10    val desc: String,
11    val img: Int
12 ) : Parcelable
13
```

Class ini digunakan untuk membuat instance object dari masing masing menu. Menu yang dimaksud disini adalah menu makanan dan menu minuman. *@Parcelable* dan *kotlinx.parcelize.Parcelize* dideklarasikan agar class tersebut dapat dibagi bagian.

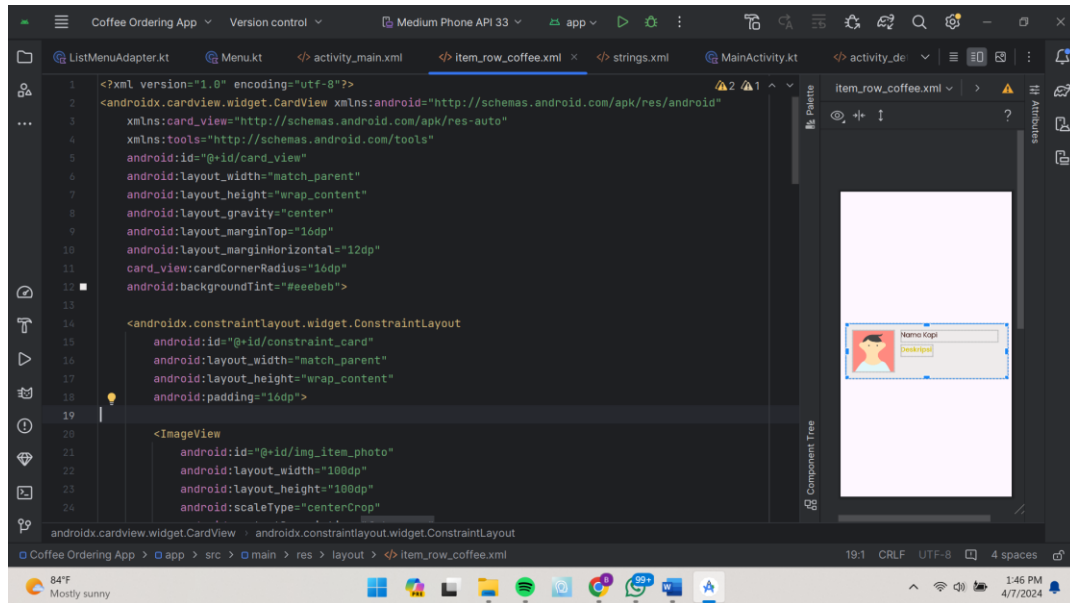
6. File “ListMenuAdapter.kt”



```
1 package com.example.coffeeorderingapp
2
3 import androidx.recyclerview.widget.RecyclerView
4
5 class ListMenuAdapter(private val context: Context, private val listMenu: ArrayList<Menu>) : RecyclerView.Adapter<ListMenuAdapter.ListViewHolder>() {
6     class ListViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
7         val imgPhoto: ImageView = itemView.findViewById(R.id.img_item_photo)
8         val tvPrice: TextView = itemView.findViewById(R.id.tv_item_price)
9         val tvName: TextView = itemView.findViewById(R.id.tv_item_name)
10    }
11
12    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {
13        val view = LayoutInflater.from(parent.context).inflate(R.layout.item_row_coffee, parent, attachToRoot = false)
14        return ListViewHolder(view)
15    }
16
17    override fun getItemCount(): Int = listMenu.size
18
19    override fun onBindViewHolder(holder: ListViewHolder, position: Int) {
20        val (name, price, desc, img) = listMenu[position]
21        val priceText = "Rp $price"
22        holder.imgPhoto.setImageResource(img)
23        holder.tvPrice.text = priceText
24        holder.tvName.text = name
25    }
26}
```

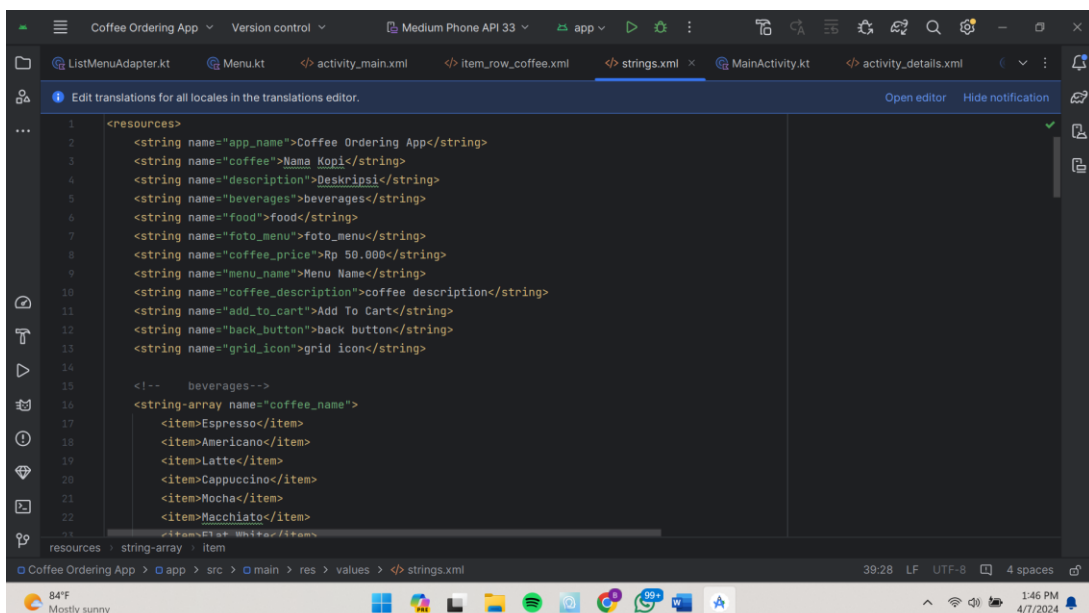
Pada class ini diberikan parameter konstruktor *context* dan *listMenu*. Di dalam definisi class tersebut terdapat class *ViewHolder* yang akan menampung masing masing dari *view* di dalam *card list*. Pada method *onBindViewHolder* akan diambil data dari posisi yang sesuai. data data tersebut selanjutnya akan digunakan untuk memberi nilai *variable view*. Dari masing masing list tersebut diberikan *onClickListener* untuk *intent* menuju *DetailsActivity*. Sebelum *intent* dimulai, data dari masing masing item akan dikirimkan ke *DetailsActivity* agar data tersebut dapat digunakan disana.

7. File “item_row_coffee.xml”



Pada file ini digunakan *CardView*, di dalamnya terdapat lagi *ConstraintLayout* yang menampung *ImageView* dan *TextView*. Constraint dari masing masing *view* diberikan *constraint* yang sesuai dengan tata letak yang diinginkan.

8. File “string.xml”



Di dalam file ini diinisialisasikan daftar nama menu, deskripsi, harga, dan foto. Masing masing dari daftar tersebut diberikan nama yang nantinya akan disesuaikan penggunaannya pada *MainActivity*.

BAB II

USER INTERFACE

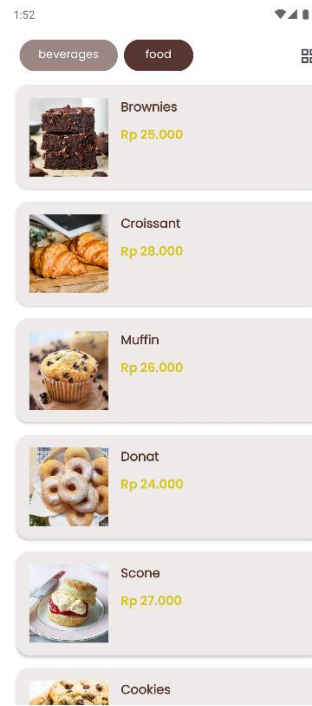
1. Main Activity

- Beverages



Pada bagian ini ditampilkan berbagai macam list menu minuman (sebagai default).

- Food



Ketika user menekan tombol “food” maka *RecyclerView* akan menampilkan berbagai macam list menu makanan.

- Layout Manager



Ketika user menekan icon grid, maka tampilan akan dirubah menjadi grid dengan 2 column. Begitupun sebaliknya, ketika user berada pada layout grid dan menekan icon list maka tampilan akan dirubah menjadi *linear layout*.

2. Details Activity



Pada bagian ini detail informasi dari masing masing menu akan ditampilkan. Disini terdapat informasi rating, nama menu, gambar, harga, dan deskripsi. Untuk data rating disini hanya design saja, sehingga semua item akan memiliki rating yang sama.

BAB III

KESIMPULAN

Untuk dapat menggunakan *RecyclerView* saya memerlukan *item_row_coffee.xml* sebagai design layout per item dari *RecyclerView* tersebut. Sebelumnya juga perlu disiapkan data data (dalam array) yang akan ditampilkan ke masing masing item. Data tersebut akan diolah menggunakan *ListMenuAdapter* dengan menampilkan masing masing dari item sesuai dengan id yang ditentukan pada layout sebelumnya. Selain itu juga diperlukan class *Parcelable* “Menu” yang berfungsi untuk memudahkan menyimpan masing masing data dari *string.xml*

Masing masing item dari *RecyclerView* dapat diberikan event *onClickListener* dengan cara memberikan *setOnClickListener* pada method *onBindViewHolder* dari class *ListMenuAdapter*. *Listner* tersebut akan melakukan intent ke *DetailsActivity* yang disertai dengan pengiriman data melalui companion object.