

**LAPORAN PRAKTIKUM**  
**PENGEMBANGAN APLIKASI BERGERAK**  
**RESPONSI 1**



**L0122034**  
**BINTANG HARIDA RAMADHAN**

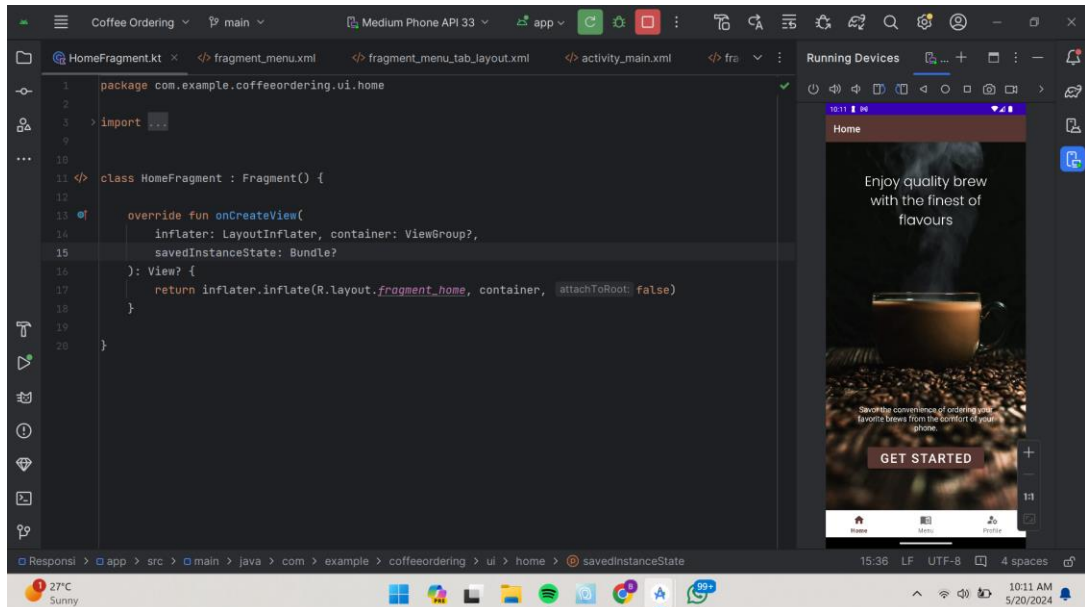
**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN SAINS DATA**  
**UNIVERSITAS SEBELAS MARET**

**2024**

# BAB I

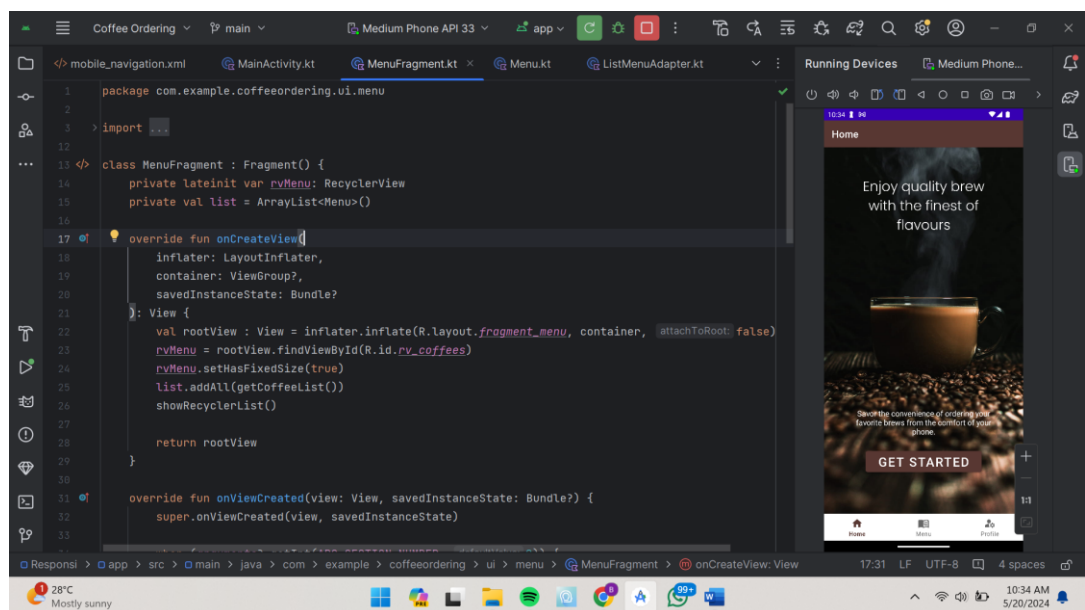
## SOURCE CODE

### 1. File “HomeFragment.kt”



Source code ini digunakan untuk menentukan aksi pada layout Home Fragnmet. Tetapi karena pada layout ini tidak memiliki aksi apa apa, sehingga pada file ini hanya terdapat default function dari pembuatan fragment. Button yang ada pada layout ini hanya pemanis saja dan tidak memiliki aksi apapun jika di tekan.

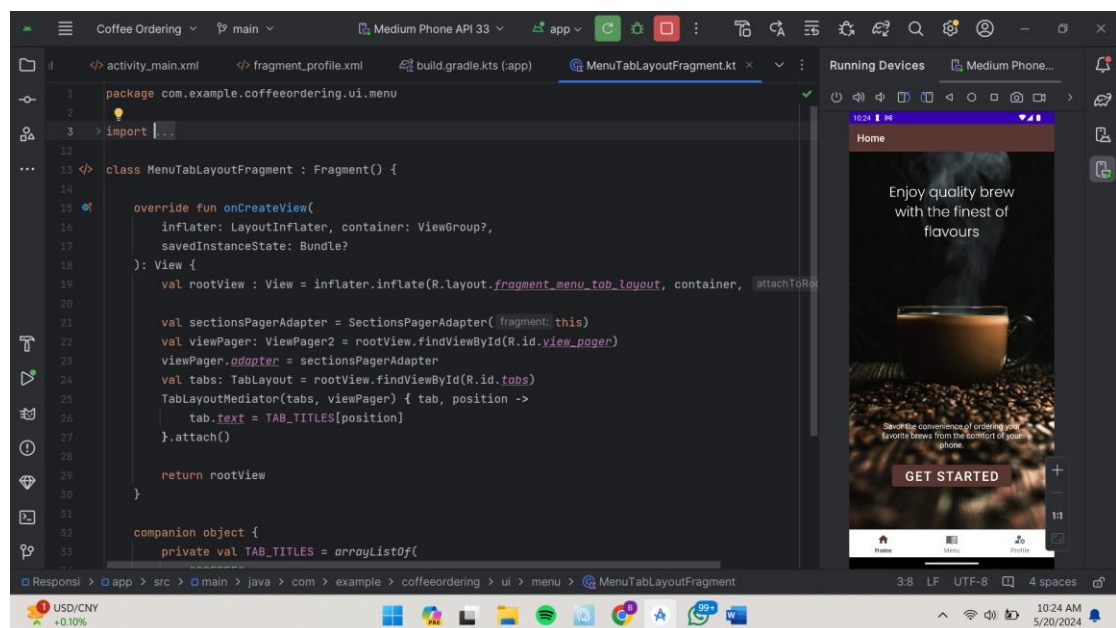
### 2. File “MenuFragment.kt”



Source code di atas adalah implementasi dari MenuFragment, yaitu fragment yang menampilkan daftar item menu dalam sebuah RecyclerView. Fragment ini menginisialisasi RecyclerView (rvMenu) dan sebuah ArrayList bernama list untuk menyimpan data item menu. Dalam method onCreateView, RecyclerView diinisialisasi. Daftar item menu kopi kemudian ditambahkan ke dalam list menggunakan metode getCoffeeList(), dan showRecyclerView() dipanggil untuk menampilkan daftar item menu dalam RecyclerView. Method onViewCreated memeriksa argumen fragment dan mengisi list dengan data yang sesuai berdasarkan nilai ARG\_SECTION\_NUMBER; jika nilainya 0, list diisi dengan item kopi, sedangkan jika nilainya 1, list diisi dengan item makanan. Setelah mengisi list, metode showRecyclerView() dipanggil untuk memperbarui RecyclerView.

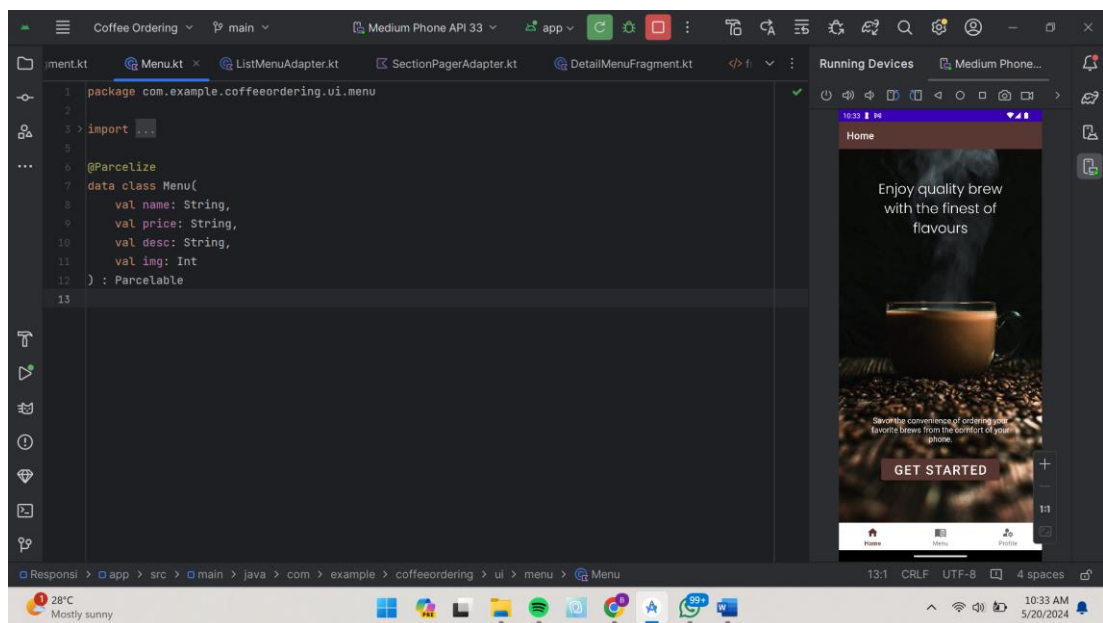
Method getCoffeeList() dan getFoodList() digunakan untuk mengambil data item kopi dan makanan dari resources aplikasi dan mengembalikannya dalam bentuk ArrayList<Menu>. Metode showRecyclerView() mengatur layout manager untuk RecyclerView dan menginisialisasi ListMenuAdapter dengan konteks fragment dan data item menu, kemudian menentukannya sebagai adapter untuk RecyclerView. Kode ini memungkinkan fragment untuk menampilkan daftar item menu yang berbeda berdasarkan nilai dari argumen yang diterimanya (ARG\_SECTION\_NUMBER), sehingga dapat menampilkan daftar kopi atau daftar makanan sesuai dengan kebutuhan aplikasi.

### 3. File “MenuTabLayoutFragment.kt”



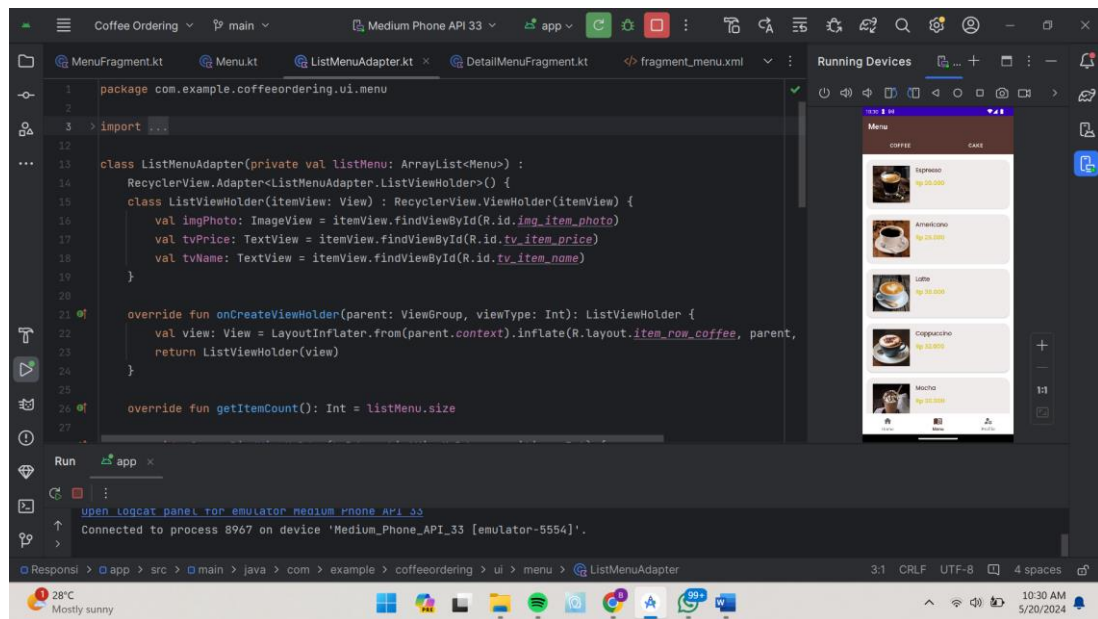
MenuTabLayoutFragment merupakan fragment yang menampilkan layout dengan tab navigasi menggunakan TabLayout dan ViewPager2. Di dalam metode onCreateView, layout fragment (fragment\_menu\_tab\_layout.xml) diinflasi dan komponen-komponen utama seperti ViewPager2 dan TabLayout diinisialisasi. SectionsPagerAdapter, sebuah adapter yang mengelola fragment-fragment untuk setiap halaman/tab, diinstansiasi dan diatur sebagai adapter untuk ViewPager2. TabLayout kemudian dihubungkan dengan ViewPager2 menggunakan TabLayoutMediator, yang menyinkronkan tab dengan halaman yang ditampilkan di ViewPager2. Judul-judul untuk setiap tab diambil dari array TAB\_TITLES, yang didefinisikan dalam companion object. Dengan mengaitkan tab dengan posisi mereka di ViewPager2, pengguna dapat menavigasi antara kategori menu seperti kopi dan kue dengan mudah. Kode ini memanfaatkan fitur modern Android untuk menyediakan navigasi berbasis tab dalam sebuah fragment, sehingga meningkatkan pengalaman pengguna dengan cara yang terstruktur dan intuitif.

#### 4. File “Menu.kt”



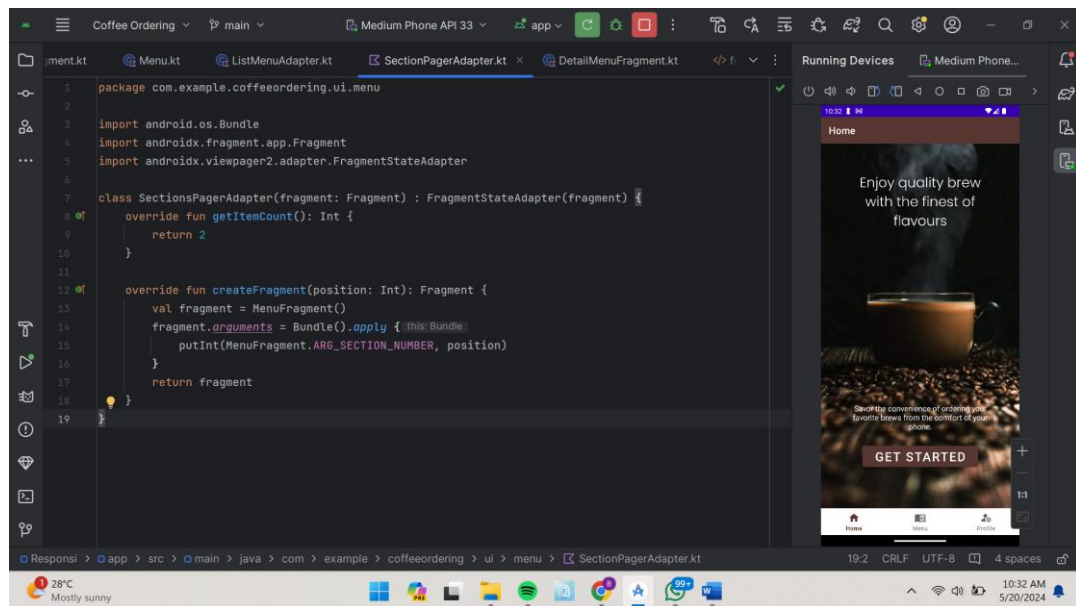
Kelas data Menu didefinisikan dengan menggunakan anotasi @Parcelize dari library Kotlin Parcelable Extensions (kotlin-parcelize). Kelas ini memiliki empat properti: name, price, desc, dan img. Properti name, price, dan desc adalah string, sedangkan img adalah integer yang mungkin merujuk ke resource ID gambar. Dengan anotasi @Parcelize, Kotlin secara otomatis menghasilkan kode yang diperlukan untuk mengimplementasikan interface Parcelable, yang memungkinkan objek Menu untuk di-parcel dan di-unparcel.

## 5. File “ListMenuAdapter.kt”



ListMenuAdapter adalah adapter untuk RecyclerView yang menampilkan daftar item menu. Kelas ini menerima `ArrayList<Menu>` sebagai parameter untuk menampilkan daftar menu. Setiap item dalam daftar menu dihubungkan dengan layout `item_row_coffee` melalui ViewHolder pattern yang diimplementasikan oleh `ViewHolder`. Pada metode `onCreateViewHolder`, layout item diinflasi dan sebuah `ViewHolder` dikembalikan. Metode `getItemCount` mengembalikan jumlah item dalam daftar menu. Pada metode `onBindViewHolder`, data dari objek `Menu` yang sesuai dengan posisi saat ini diambil dan diikat ke tampilan item seperti `ImageView` dan `TextView` di `ViewHolder`. Ketika sebuah item diklik, adapter membuat `Bundle` yang berisi data menu (seperti nama, harga, deskripsi, dan gambar), dan menggunakan `findNavController` untuk menavigasi ke `DetailMenuFragment`, sambil membawa bundle tersebut. Ini memungkinkan aplikasi untuk menampilkan detail menu ketika sebuah item dalam daftar diklik, dengan menggunakan Navigation Component dari Android Jetpack untuk mengelola navigasi antar fragment.

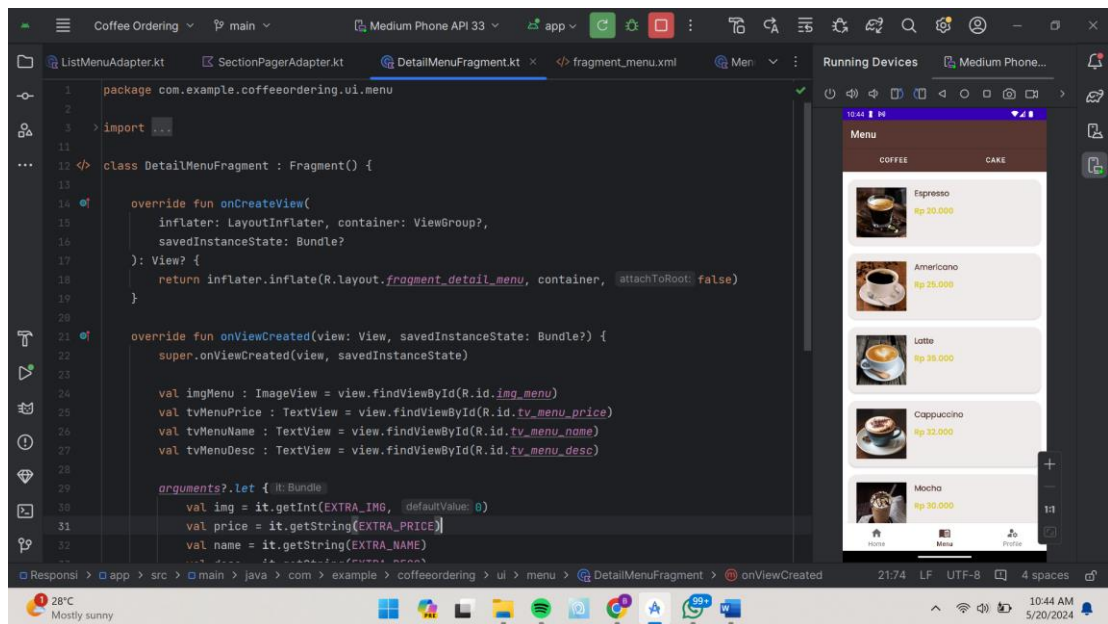
## 6. File “SectionPagerAdapter.kt”



SectionsPagerAdapter adalah sebuah adapter yang mengelola dan menampilkan Fragment dalam ViewPager2. Kelas ini menerima sebuah Fragment sebagai parameter dalam konstruktornya, yang digunakan oleh FragmentStateAdapter untuk menampilkan fragmen. Metode getItemCount mengembalikan jumlah total halaman (fragmen) yang akan ditampilkan, yang dalam hal ini adalah dua. Metode createFragment digunakan untuk membuat dan mengembalikan sebuah Fragment berdasarkan posisi saat ini dari ViewPager2. Dalam hal ini, setiap kali createFragment dipanggil, sebuah MenuFragment baru dibuat dan argumen position disematkan ke dalam Bundle sebagai ARG\_SECTION\_NUMBER. Nilai position ini menentukan data yang akan ditampilkan dalam MenuFragment, memungkinkan setiap halaman untuk menampilkan konten yang berbeda. Adapter ini memungkinkan pengguna untuk menggeser antara dua halaman (tab), yang masing-masing menampilkan daftar menu yang berbeda.

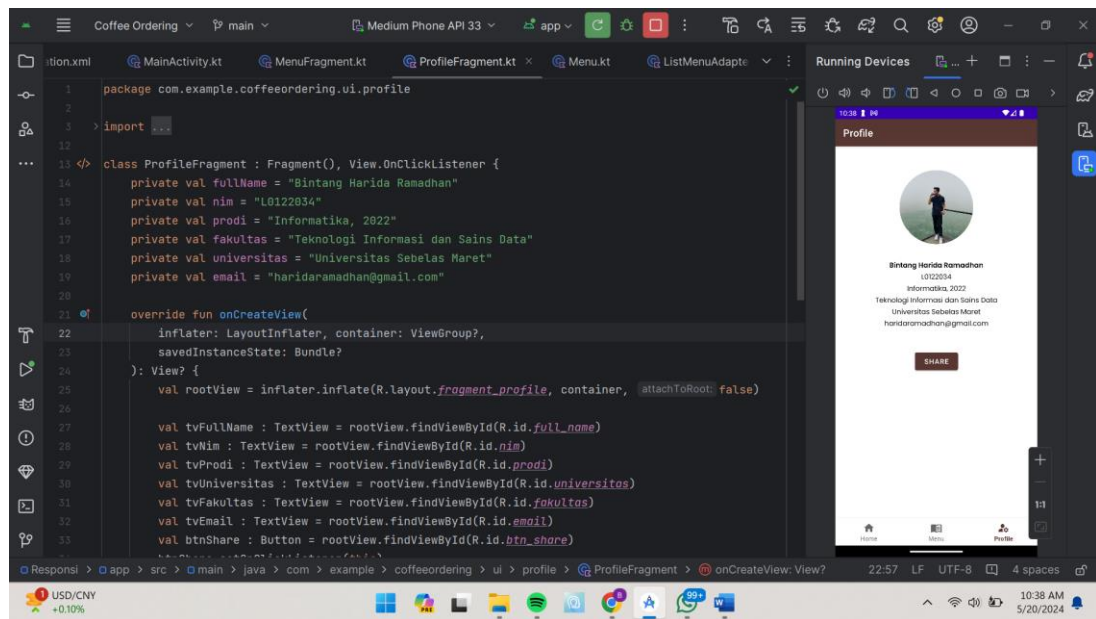


## 7. File “DetailMenuFragment.kt”



DetailMenuFragment adalah Fragment yang berfungsi untuk menampilkan detail dari item menu tertentu yang dipilih oleh pengguna. Dalam metode onCreateView, layout fragment\_detail\_menu di-inflate untuk digunakan sebagai tampilan fragment ini. Pada metode onViewCreated, berbagai View seperti ImageView untuk gambar menu (imgMenu), dan TextView untuk harga (tvMenuPrice), nama (tvMenuName), dan deskripsi menu (tvMenuDesc) diinisialisasi dengan memanggil findViewById pada View yang di-inflate. Fragment ini kemudian memeriksa apakah ada Bundle argumen yang diterima (melalui arguments). Jika ada, nilai-nilai seperti gambar (img), harga (price), nama (name), dan deskripsi (desc) diambil dari argumen tersebut dan digunakan untuk mengisi View yang sesuai. Gambar diatur dengan setImageResource, dan teks diatur dengan setText. companion object mendefinisikan konstanta untuk kunci argumen (EXTRA\_NAME, EXTRA\_PRICE, EXTRA\_DESC, EXTRA\_IMG) yang digunakan untuk mengambil data dari Bundle.

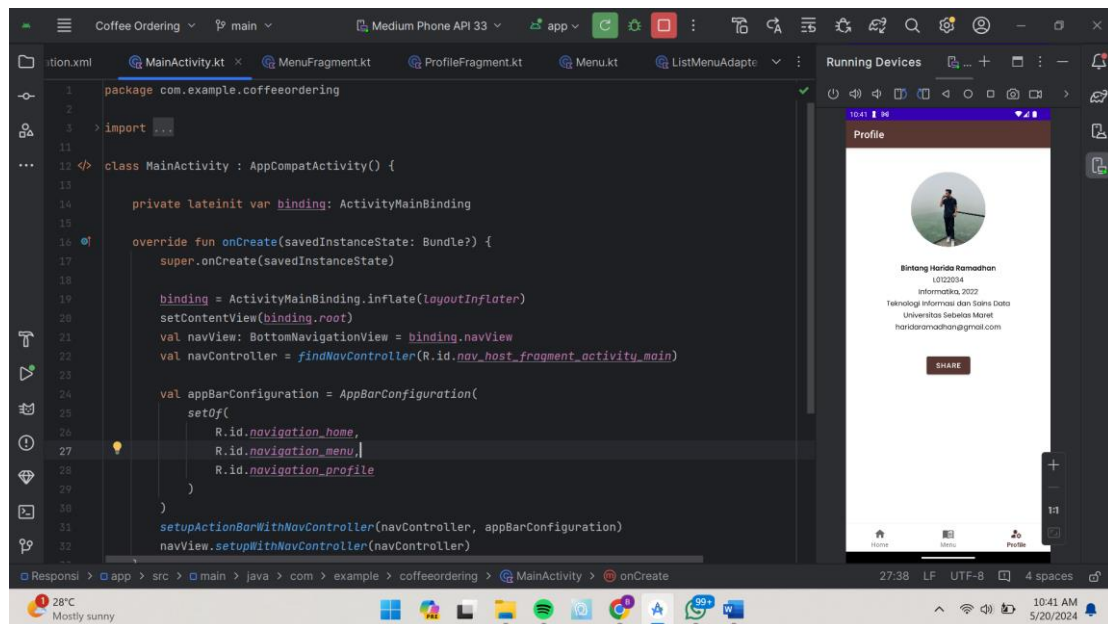
## 8. File "ProfileFragment.kt"



ProfileFragment adalah sebuah Fragment dalam aplikasi Android yang berfungsi untuk menampilkan informasi profil pengguna. Informasi ini termasuk nama lengkap, NIM, program studi, fakultas, universitas, dan email pengguna. Data ini disimpan dalam variabel instance dalam kelas. Saat ProfileFragment dibuat, layout `fragment_profile` diinflasi, dan referensi ke berbagai elemen UI seperti `TextView` dan `Button` diambil dari layout tersebut. `TextView` kemudian diisi dengan data profil yang sesuai. Tombol berbagi (`btn_share`) diatur dengan `OnClickListener` yang mendefinisikan perilaku ketika tombol tersebut ditekan. Ketika tombol berbagi ditekan, sebuah `Intent` dibuat dengan aksi `Intent.ACTION_SEND`, tipe `"text/plain"`, dan beberapa ekstra seperti email tujuan, subjek, dan teks yang berisi data profil. `Intent` ini kemudian digunakan untuk memulai aktivitas baru yang memungkinkan pengguna memilih aplikasi untuk berbagi informasi profil.



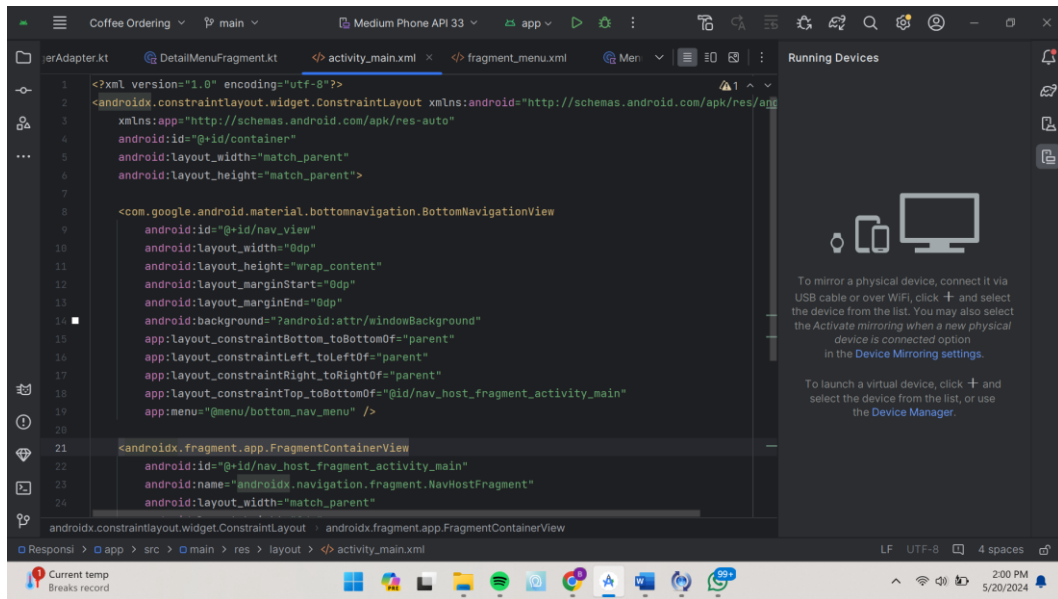
## 9. File “MainActivity.kt”



MainActivity adalah aktifitas utama dari aplikasi dengan menampilkan navigasi dasar menggunakan BottomNavigationView. Pada metode onCreate, ActivityMainBinding diinisialisasi untuk menghubungkan layout dengan aktivitas ini. Layout diatur dengan memanggil setContentView pada root view dari binding. BottomNavigationView (navView) diambil dari binding dan dihubungkan dengan NavController, yang dikelola oleh NavHostFragment yang terdapat dalam layout utama (activity\_main).

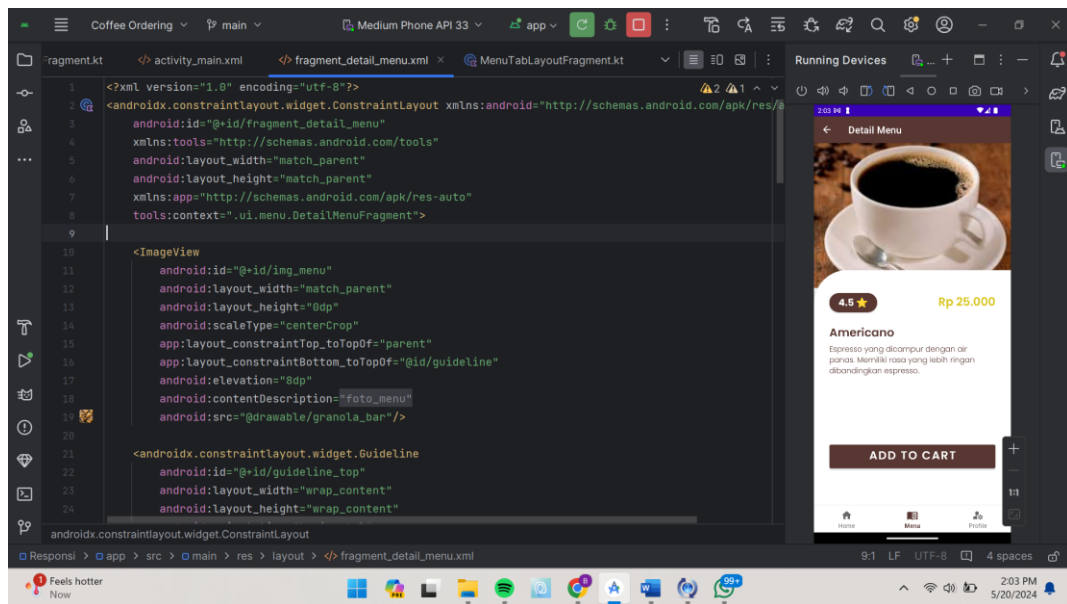
Sebuah AppBarConfiguration dibuat untuk menentukan fragment mana saja yang harus dianggap sebagai destinasi tingkat atas, dalam hal ini termasuk R.id.navigation\_home, R.id.navigation\_menu, dan R.id.navigation\_profile. Metode setupActionBarWithNavController digunakan untuk menyelaraskan ActionBar dengan NavController, dan navView.setupWithNavController menghubungkan BottomNavigationView dengan NavController, memungkinkan navigasi diatur sesuai dengan destinasi yang ditentukan. Fungsi onSupportNavigateUp dioverride untuk mendukung navigasi naik ketika tombol naik di toolbar ditekan, menggunakan NavController untuk menangani navigasi.

## 10. File “main\_activity.xml”



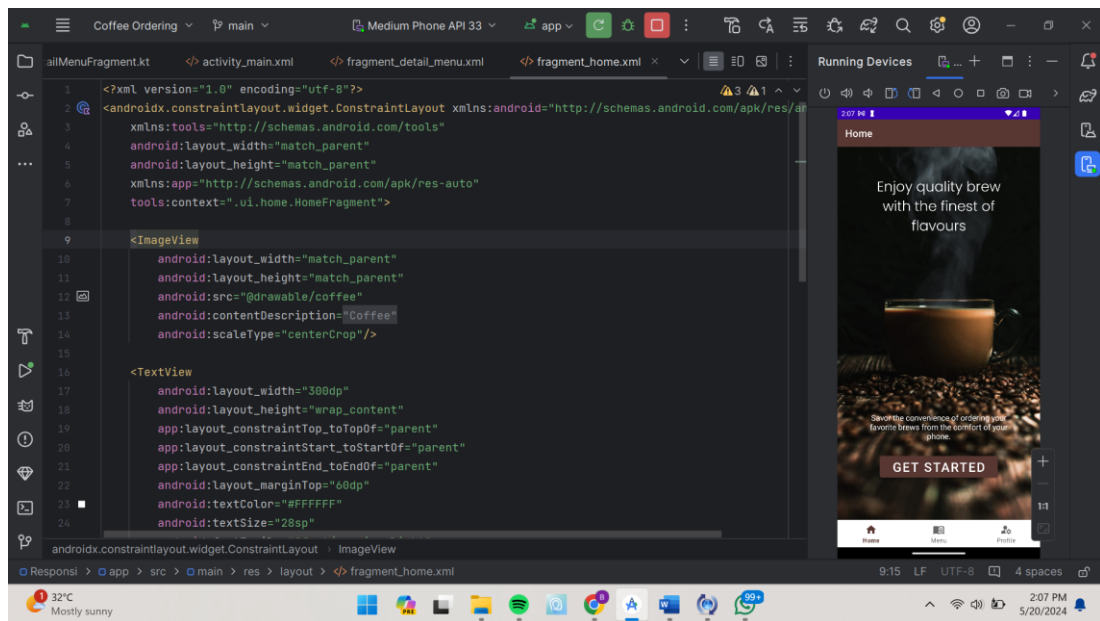
Source code di atas mendefinisikan layout utama untuk sebuah activity menggunakan ConstraintLayout sebagai root layout. Di dalamnya terdapat dua elemen utama, yaitu BottomNavigationView dan NavHostFragment. BottomNavigationView dengan ID nav\_view terletak di bagian bawah layar dan digunakan untuk navigasi antar fragment dalam aplikasi, dengan menu yang ditentukan di file bottom\_nav\_menu.xml. NavHostFragment dengan ID nav\_host\_fragment\_activity\_main berfungsi sebagai container untuk menampilkan fragment sesuai dengan navigasi yang ditentukan dalam file mobile\_navigation.xml. NavHostFragment ini diatur sebagai default navigation host menggunakan atribut app:defaultNavHost="true", dan posisinya diatur dengan constraint agar berada di atas BottomNavigationView serta memenuhi sisa ruang yang tersedia di layar.

## 11. File “fragment\_detail\_menu.xml”



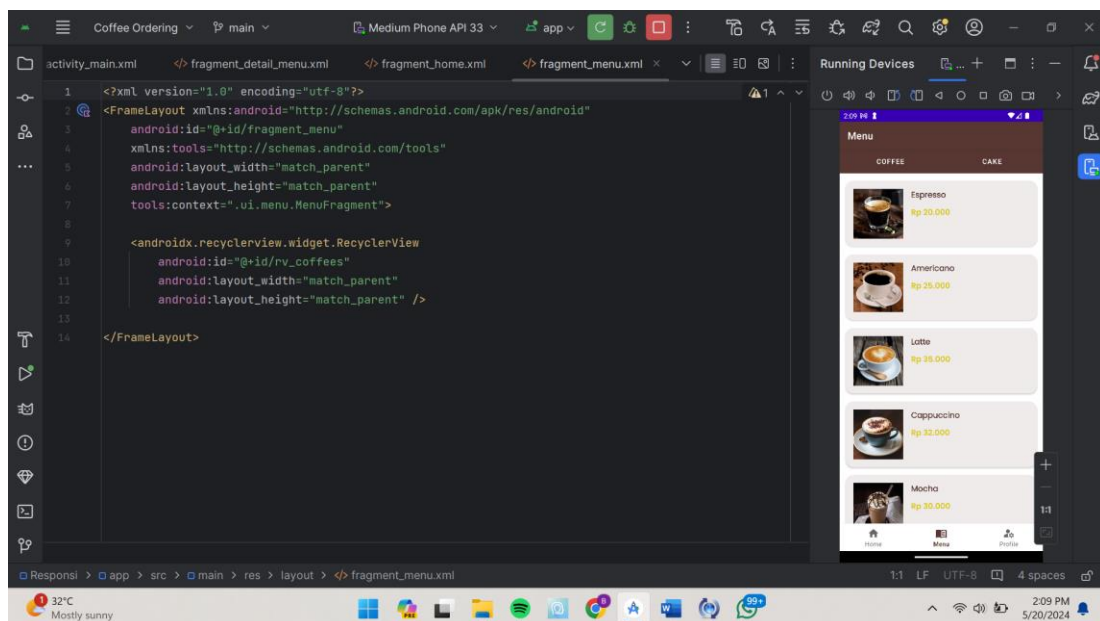
Kode XML ini mendefinisikan tata letak untuk DetailMenuFragment dalam aplikasi Android menggunakan ConstraintLayout. Layout ini menampilkan gambar menu di bagian atas, diikuti dengan informasi detail tentang menu tersebut yang termasuk rating, harga, nama, dan deskripsi. Gambar menu menggunakan ImageView dengan skala jenis centerCrop, sementara detail informasi ditampilkan dalam ConstraintLayout yang diatur di bawah guideline. Informasi tentang rating menu ditampilkan dalam CardView, dan harga, nama, serta deskripsi ditampilkan dalam TextView. Deskripsi menu di dalam ScrollView memungkinkan konten teks panjang untuk digulir. Tombol "Add to Cart" diletakkan di bagian bawah tata letak, memungkinkan pengguna untuk menambahkan item ke keranjang belanja mereka. Guideline digunakan untuk membantu pengaturan tata letak secara proporsional.

## 12. File “fragment\_home.xml”



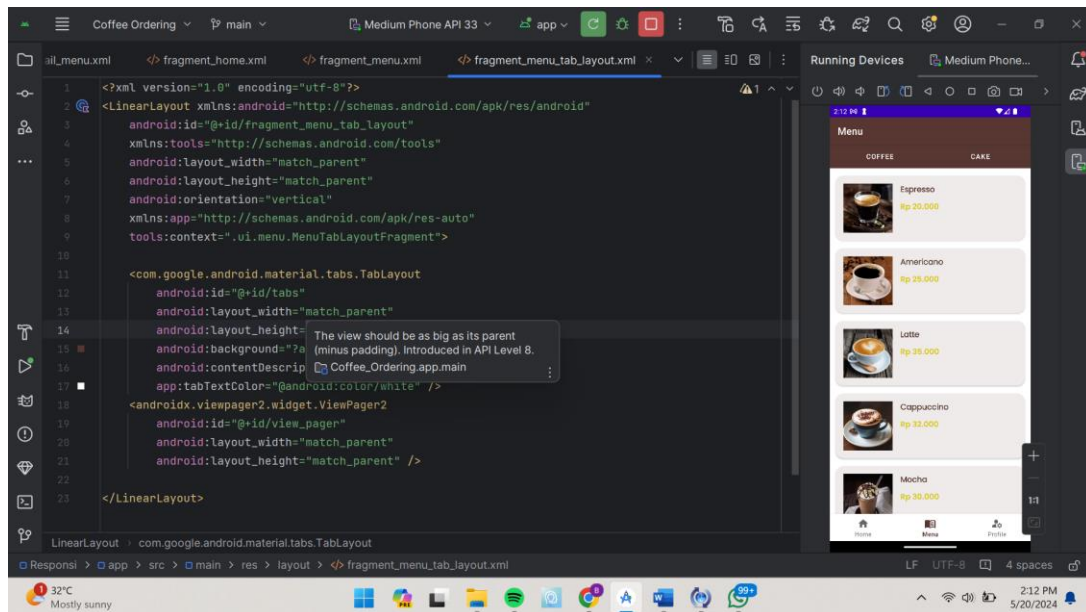
Layout XML ini mendefinisikan tampilan untuk HomeFragment dalam aplikasi pemesanan kopi. Menggunakan ConstraintLayout sebagai root, layout ini berisi elemen-elemen berikut: sebuah ImageView yang menampilkan gambar kopi sebagai latar belakang dengan skala tengah potong; dua TextView yang menampilkan deskripsi toko kopi dan aplikasi dengan teks berwarna putih, salah satunya berada di bagian atas layar dengan margin atas 60dp dan yang lainnya di atas tombol; serta sebuah Button berukuran 240dp lebar yang menampilkan teks "Get Started" di bagian bawah layar dengan margin bawah 80dp. Semua elemen diatur untuk berada di tengah secara horizontal.

### 13. File “fragment menu.xml”



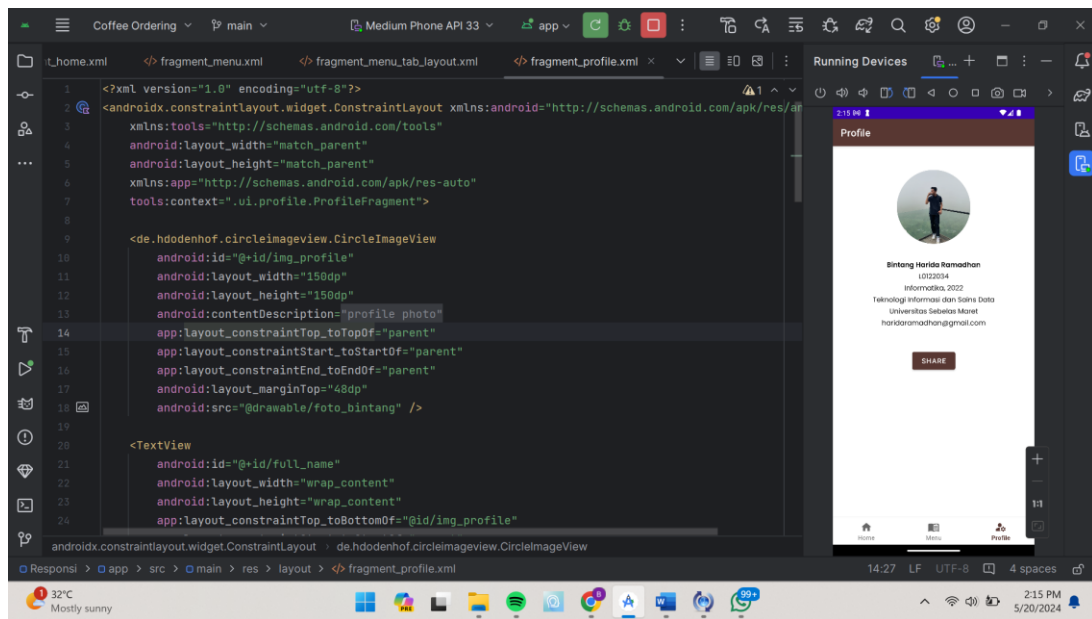
Layout XML ini mendefinisikan tampilan untuk MenuFragment dalam aplikasi pemesanan kopi. Menggunakan FrameLayout sebagai root, layout ini hanya berisi satu elemen RecyclerView yang memiliki ID rv\_coffees. RecyclerView ini diatur untuk mengisi seluruh ukuran lebar dan tinggi layar, memungkinkan menampilkan daftar item (menu kopi) dalam tampilan fragmen tersebut.

#### 14. File “fragment\_menu\_tab\_layout.xml”



Layout XML ini mendefinisikan tampilan untuk MenuTabLayoutFragment dalam aplikasi pemesanan kopi. Layout ini menggunakan LinearLayout dengan orientasi vertikal sebagai root dan terdiri dari dua elemen utama: TabLayout dan ViewPager2. TabLayout dengan ID tabs berada di bagian atas dan menampilkan tab untuk navigasi antar halaman (dengan teks berwarna putih dan latar belakang berwarna utama aplikasi). Di bawahnya, ViewPager2 dengan ID view\_pager diatur untuk mengisi sisa ruang layar, memungkinkan penggeseran antar halaman fragmen yang sesuai dengan tab yang dipilih.

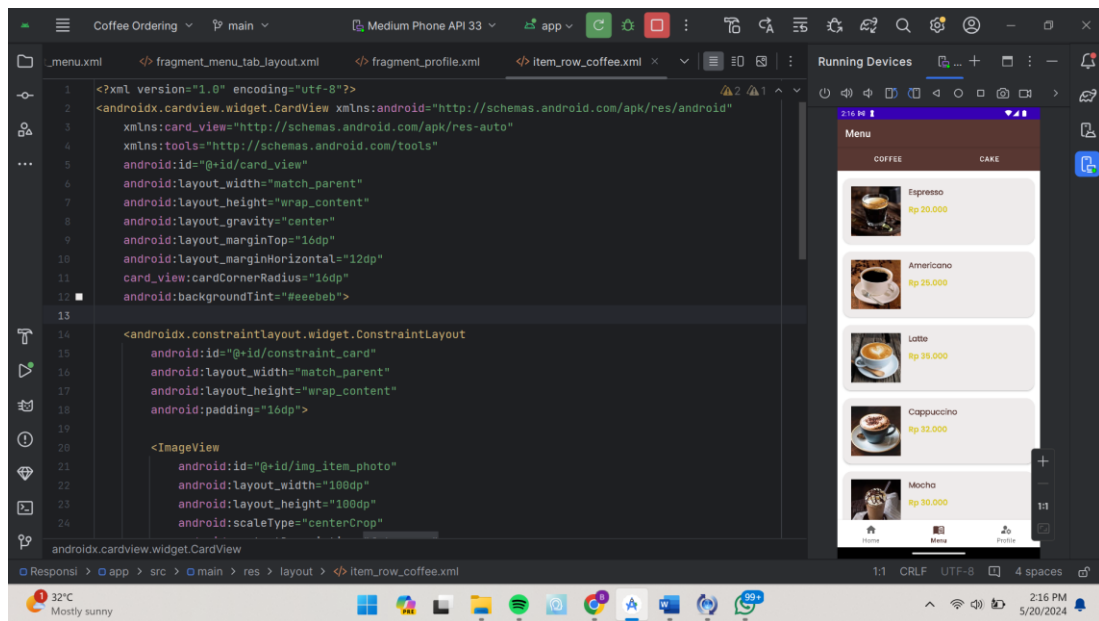
## 15. File “fragment\_profile.xml”



Layout XML ini mendefinisikan tampilan untuk ProfileFragment dalam aplikasi. Layout menggunakan ConstraintLayout sebagai root dan terdiri dari berbagai elemen UI untuk menampilkan informasi profil pengguna. Di bagian atas, terdapat CircleImageView untuk menampilkan foto profil dengan ID img\_profile. Di bawahnya, ada beberapa TextView untuk menampilkan nama lengkap (full\_name), NIM (nim), program studi (prodi), fakultas (fakultas), universitas (universitas), dan email (email). Setiap TextView diatur untuk menampilkan teks dengan font khusus dan warna hitam. Di bagian bawah, ada Button dengan ID btn\_share yang memungkinkan pengguna untuk berbagi informasi profil mereka. Semua elemen diatur dengan constraint untuk menjaga tata letak yang responsif dan terpusat secara vertikal.

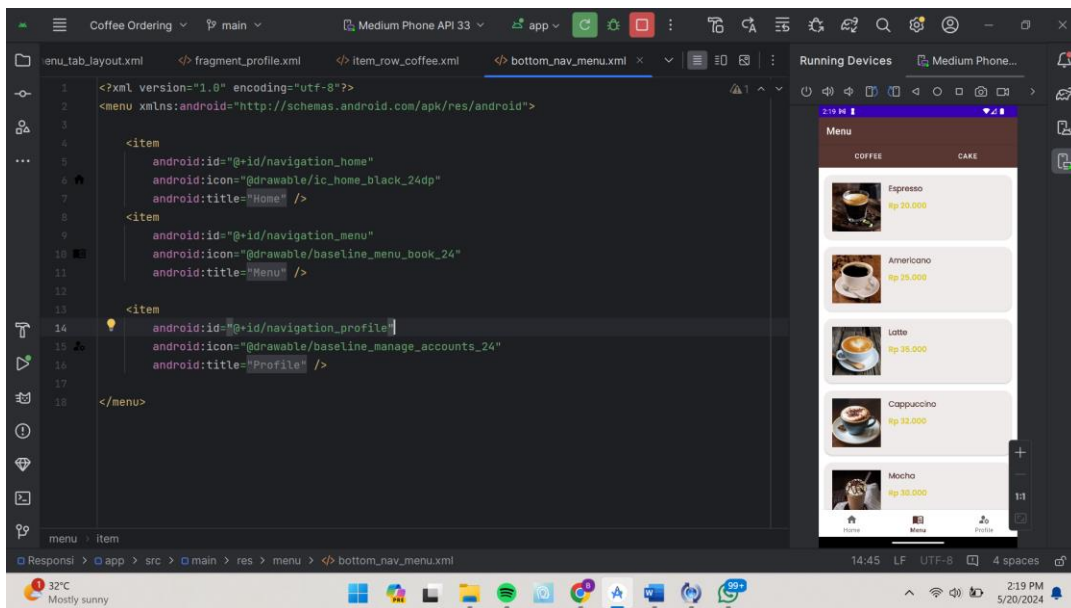


## 16. File “item\_row\_coffee.xml”



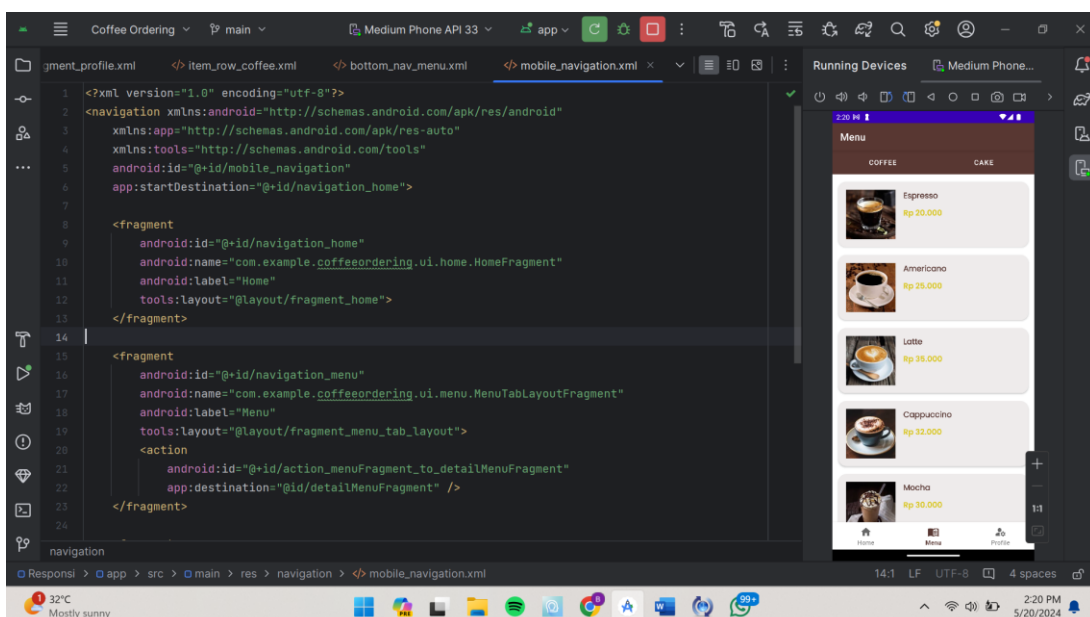
Layout XML ini mendefinisikan tampilan untuk item dalam daftar menggunakan CardView. CardView ini memiliki margin dan sudut melengkung, serta latar belakang berwarna abu-abu terang. Di dalam CardView, terdapat ConstraintLayout yang mengatur tata letak elemen-elemen UI. Terdapat ImageView untuk menampilkan gambar item di sisi kiri, dengan ukuran 100dp x 100dp dan contentDescription yang disesuaikan. Di sebelah kanan gambar, terdapat dua TextView. TextView pertama (tv\_item\_name) menampilkan nama item dengan teks berukuran 16sp, bergaya tebal, dan warna coklat gelap. TextView kedua (tv\_item\_price) berada di bawah nama item, menampilkan harga dengan teks berukuran 16sp, berwarna emas, dan bergaya semi tebal. Semua elemen diatur menggunakan constraint untuk memastikan tata letak yang rapi dan responsif.

## 17. File “bottom\_nav\_menu.xml”



File XML ini mendefinisikan menu navigasi bawah untuk aplikasi Android. Menu ini berisi tiga item navigasi yang ditampilkan sebagai ikon dengan teks di bawahnya. Item pertama adalah `navigation_home`, yang menggunakan ikon `ic_home_black_24dp` dan menampilkan judul "Home". Item kedua adalah `navigation_menu`, yang menggunakan ikon `baseline_menu_book_24` dan menampilkan judul "Menu". Item ketiga adalah `navigation_profile`, yang menggunakan ikon `baseline_manage_accounts_24` dan menampilkan judul "Profile". Menu ini akan digunakan untuk berpindah antara tiga halaman utama aplikasi: Home, Menu, dan Profile.

## 18. File “mobile\_navigation.xml”



File XML ini mendefinisikan navigasi dalam aplikasi Android menggunakan komponen Navigation dari Jetpack. File ini mendeklarasikan `nav_graph` dengan titik awal `navigation_home`, yang merujuk pada `HomeFragment`. Selain itu, terdapat tiga fragmen lainnya: `navigation_menu` yang merujuk pada `MenuTabLayoutFragment`, `detailMenuFragment` untuk detail menu, dan `navigation_profile` yang merujuk pada `ProfileFragment`. Navigasi dari `MenuTabLayoutFragment` ke `DetailMenuFragment` diatur dengan `action_menuFragment_to_detailMenuFragment`, yang memungkinkan perpindahan ke detail menu dari menu utama. Setiap fragmen juga dilengkapi dengan label dan layout yang sesuai untuk menentukan tampilan UI mereka.

## BAB II

### USER INTERFACE

#### 1. Home



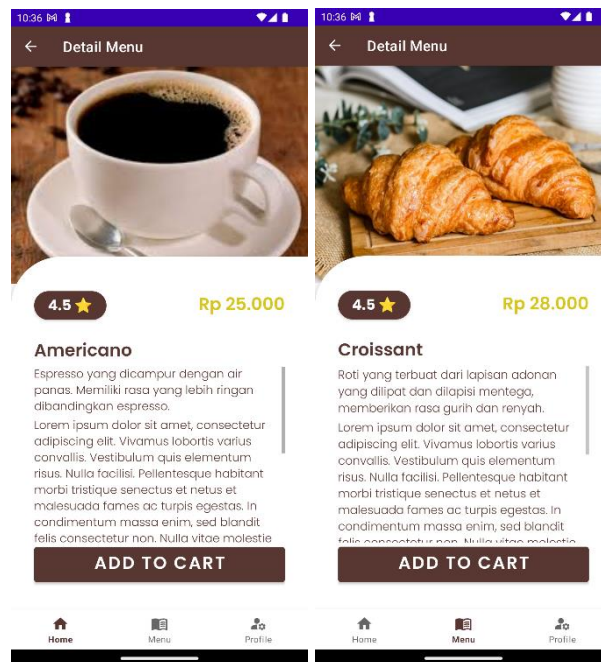
Pada fragment ini ditampilkan text view yang berisi garis besar dari aplikasi yang saya buat.

#### 2. Menu



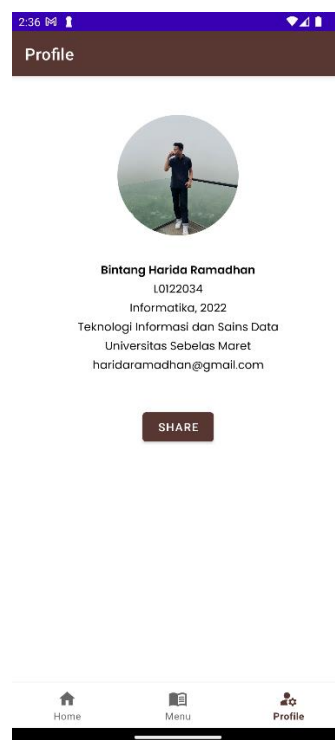
Pada fragment ini ditampilkan RecyclerView yang masing masing itemnya akan menampilkan gambar, nama, dan harga dari masing masing menu. Disini menggunakan tab layout yaitu tab Coffee dan tab Cake.

### 3. Detail Menu



Pada fragment ini ditampilkan gambar dalam ukuran besar, nama, harga, dan deskripsi mengenai menu sesuai dengan item menu yang dipilih pada Recycler View.

### 4. Profile



Pada fragment ini ditampilkan foto dan data diri saya. Selain itu juga terdapat button Share yang akan melakukan intent ke default sharing application pada android.

### **BAB III**

#### **KESIMPULAN**

Aplikasi ini adalah aplikasi pemesanan kopi pada Android. Kode Kotlin berfungsi sebagai backend yang mengatur logika dan fungsi dari aplikasi. Terdapat beberapa kelas penting yang digunakan untuk mengelola data dan tampilan, seperti Menu, ListMenuAdapter, SectionsPagerAdapter, dan ProfileFragment. Kelas-kelas ini membantu dalam pengelolaan dan penampilan data menu kopi kepada pengguna.

RecyclerView digunakan dalam MenuFragment untuk menampilkan daftar menu kopi, memberikan pengguna pengalaman memilih menu dengan mudah dan intuitif. Selain itu, ViewPager2 dan TabLayout digunakan dalam MenuTabLayoutFragment untuk membuat antarmuka tab yang interaktif, memungkinkan pengguna untuk beralih antara kategori menu yang berbeda dengan lancar.

Pada ProfileFragment, terdapat fitur untuk berbagi informasi profil menggunakan Intent, yang memungkinkan pengguna untuk berbagi detail profil mereka dengan aplikasi lain. MainActivity bertindak sebagai pusat pengelolaan navigasi antara berbagai fragmen menggunakan BottomNavigationView, memastikan pengalaman navigasi yang mulus dan konsisten bagi pengguna.

Kode XML melengkapi fungsi Kotlin dengan mendefinisikan tata letak dan tampilan antarmuka pengguna. Setiap fragmen memiliki layout XML yang sesuai, seperti fragment\_detail\_menu.xml untuk menampilkan detail menu, fragment\_menu\_tab\_layout.xml untuk tata letak tab menu, dan fragment\_profile.xml untuk profil pengguna. CardView digunakan dalam fragment\_menu.xml untuk menampilkan item menu dalam format yang menarik dan informatif.

Selain itu, activity\_main.xml mendefinisikan tata letak utama aplikasi, termasuk BottomNavigationView yang memungkinkan pengguna untuk menavigasi antar fragmen seperti Home, Menu, dan Profile. Struktur navigasi ini juga didefinisikan dalam mobile\_navigation.xml, yang mengatur alur navigasi dan hubungan antar fragmen dalam aplikasi.