**LAPORAN TUGAS AKHIR MOBILE BESERTA PEJELASANNYA**



**DIBUAT OLEH:**

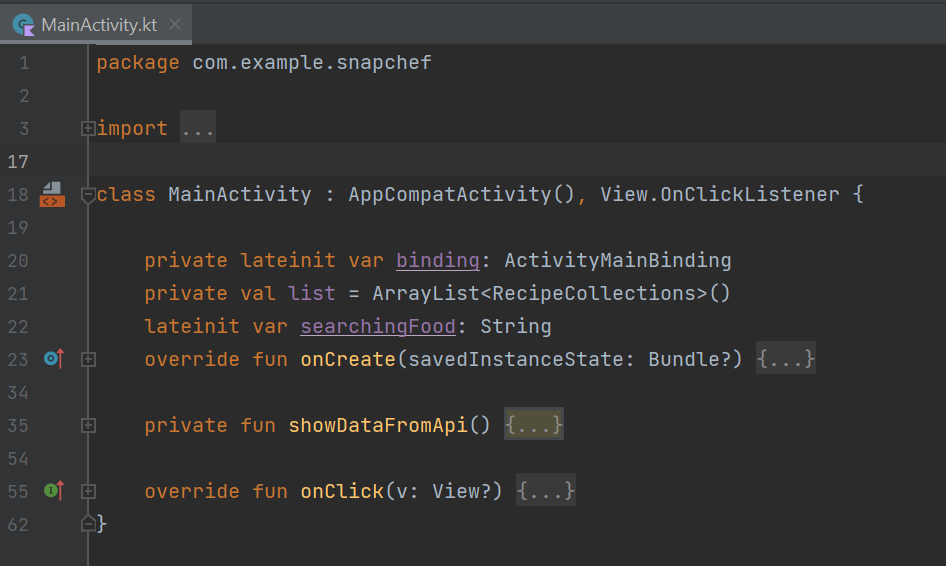
**ALVI SUHARTANTO – 20313015**

**UNIVERSITAS TEKNOKRAT INDONESIA**

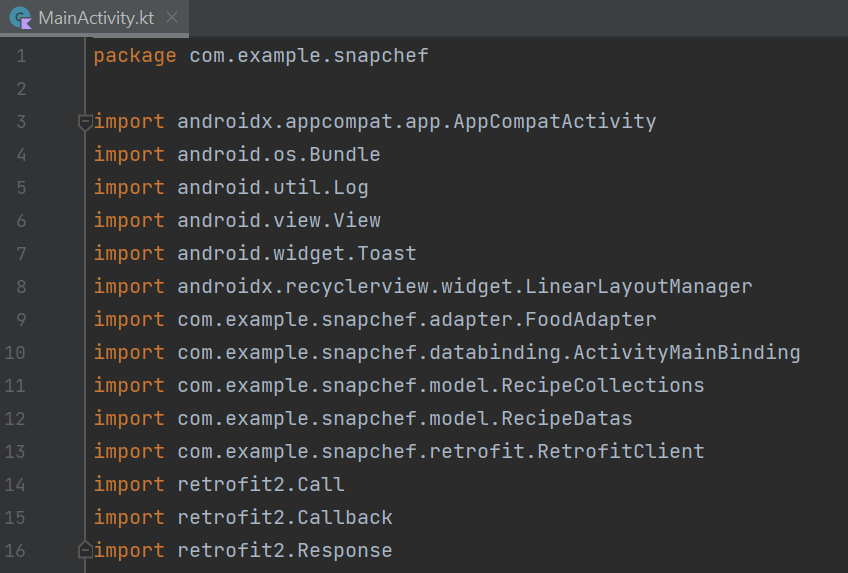
**BANDAR LAMPUNG**

**2023**

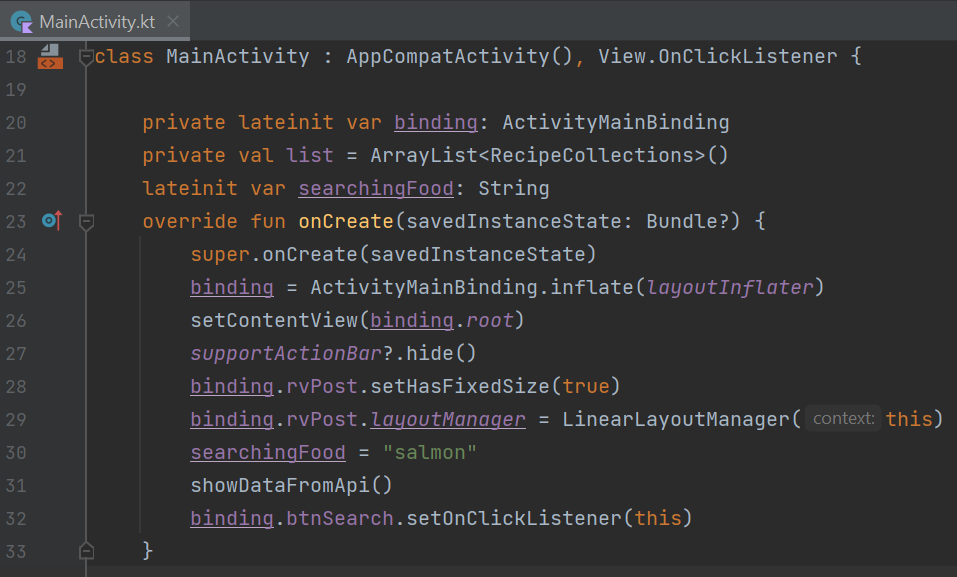
1. **Menambahkan file MainActivity**



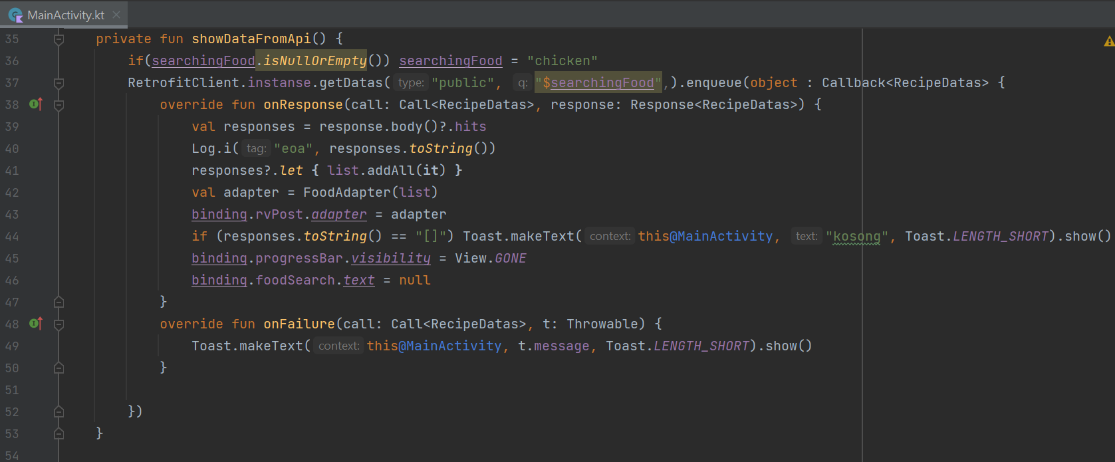
Didalam sebuah file **MainActivity** terdapat beberap fungsi yang di tambah kan diantara lain ada beberpa fungsi yaitu:

****

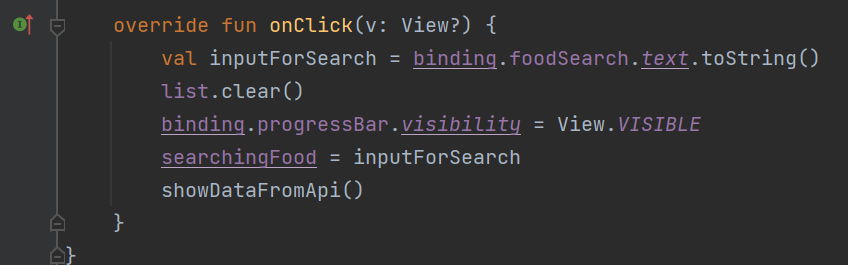
* *Import* yang digunakan dalam file MainActivity di projek kami.

****

* View binding adalah fitur yang memudahkan kami menulis kode yang berinteraksi dengan tampilan. Setelah diaktifkan dalam sebuah modul, view binding akan menghasilkan class binding untuk setiap file tata letak XML yang ada dalam modul tersebut. Instance class binding berisi referensi langsung ke semua tampilan yang memiliki ID di tata letak yang terkait.
* *List* untuk menampilkan list pada RecipCollections secara array atau berulang-berulang.
* *onCreate ()* adalah kondisi awal saat Activity baru diciptakan, inisialisasi savedInstanceState.

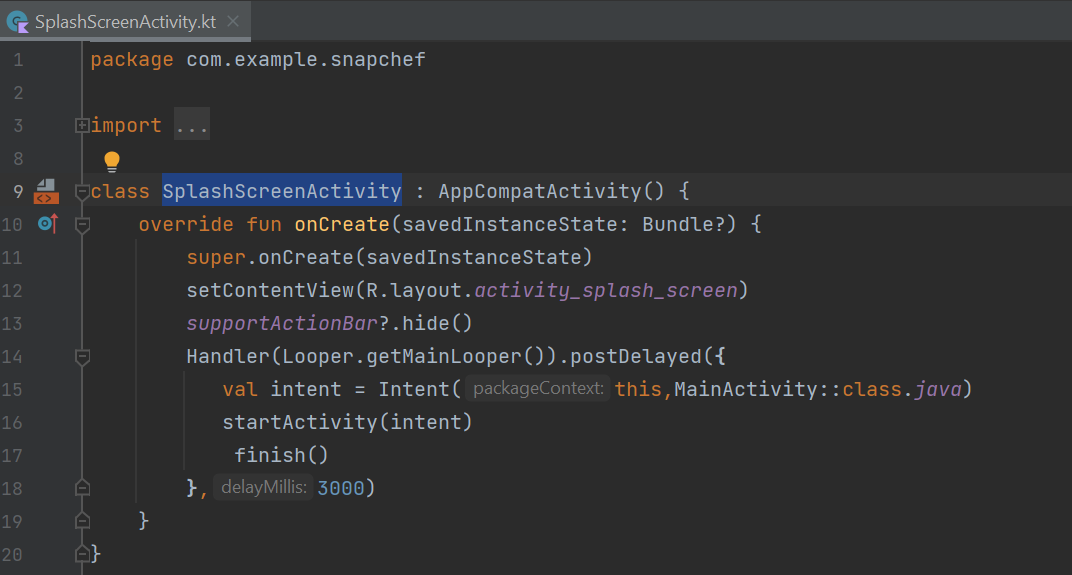


* *showDataFromApi ()* untuk menarik sebuah API yang ingin digunakan dalam aplikasi ini.

**

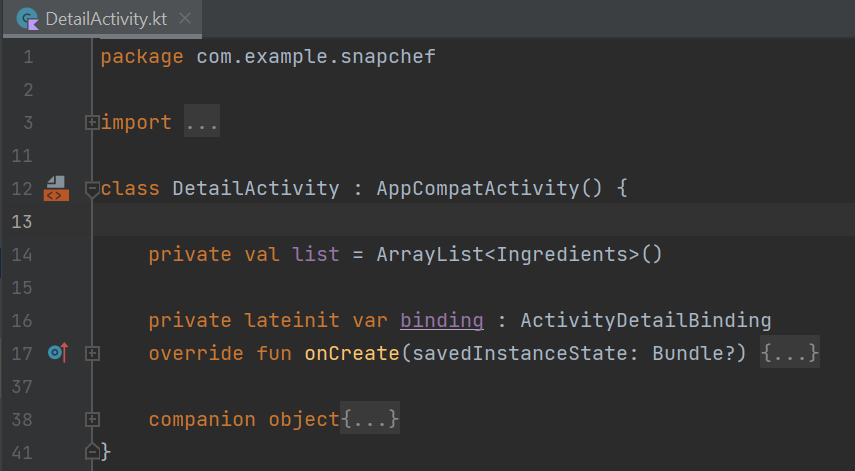
* *onClick* digunakan untuk menentukan fungsi Kotlin yang akan dipanggil dalam aktivitas saat tombol diklik.

1. **Menambahkan file SplashScreenActivity**

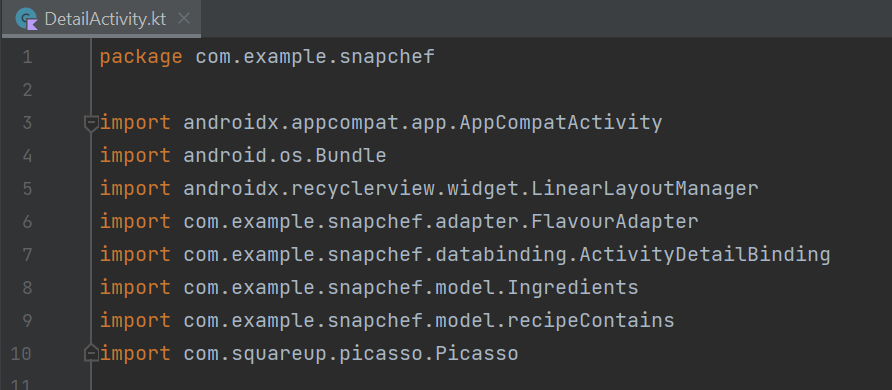
****

* *onCreate ()* adalah kondisi awal saat Activity baru diciptakan, inisialisasi savedInstanceState.

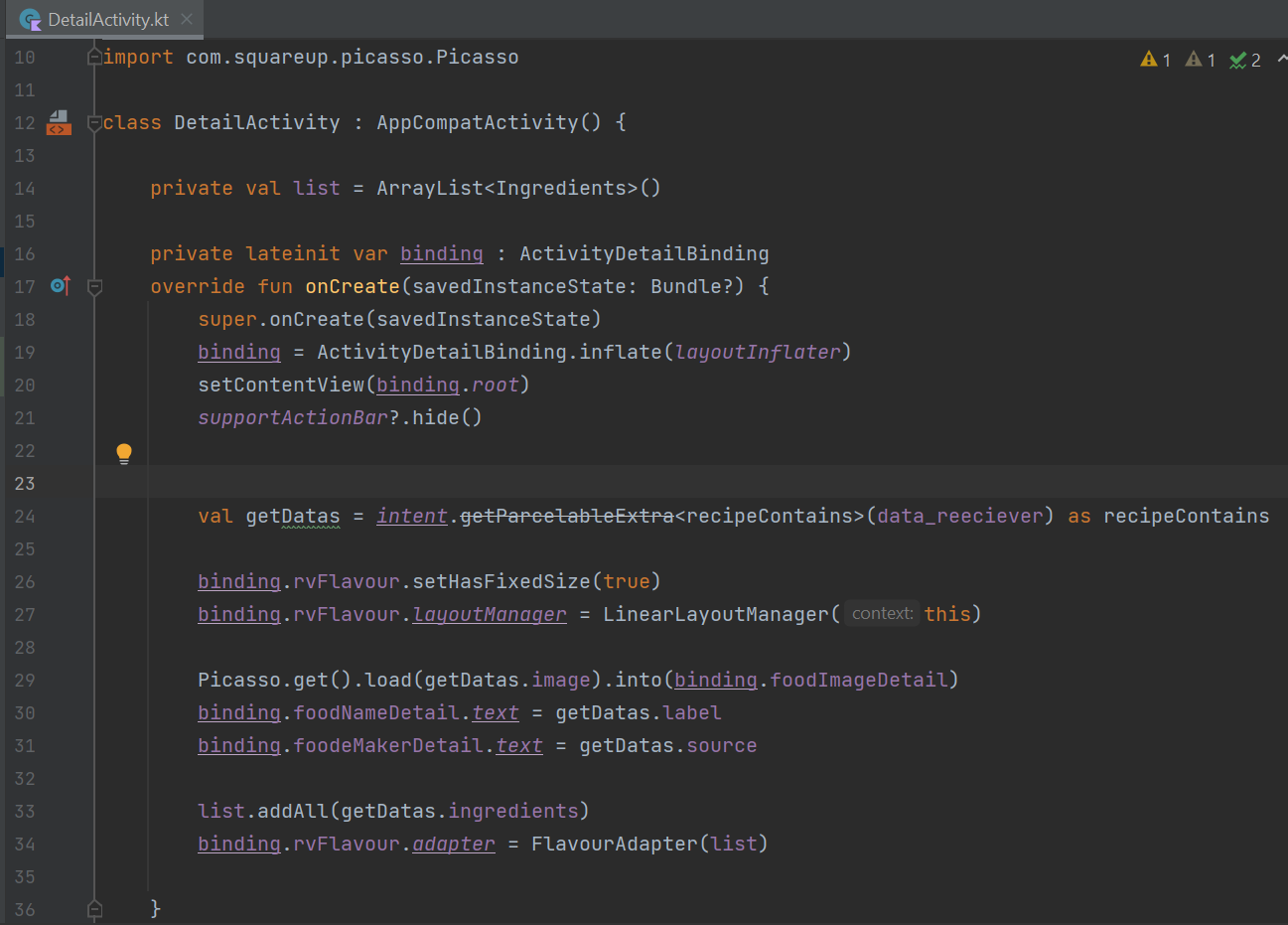
1. **Menambahkan file DetailActivity**

****

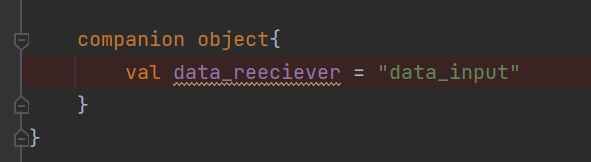
Didalam sebuah file **DetailActivity** terdapat beberap fungsi yang di tambah kan diantara lain ada beberpa fungsi yaitu:



* *Import* yang digunakan dalam file DetailActivity di projek kami.

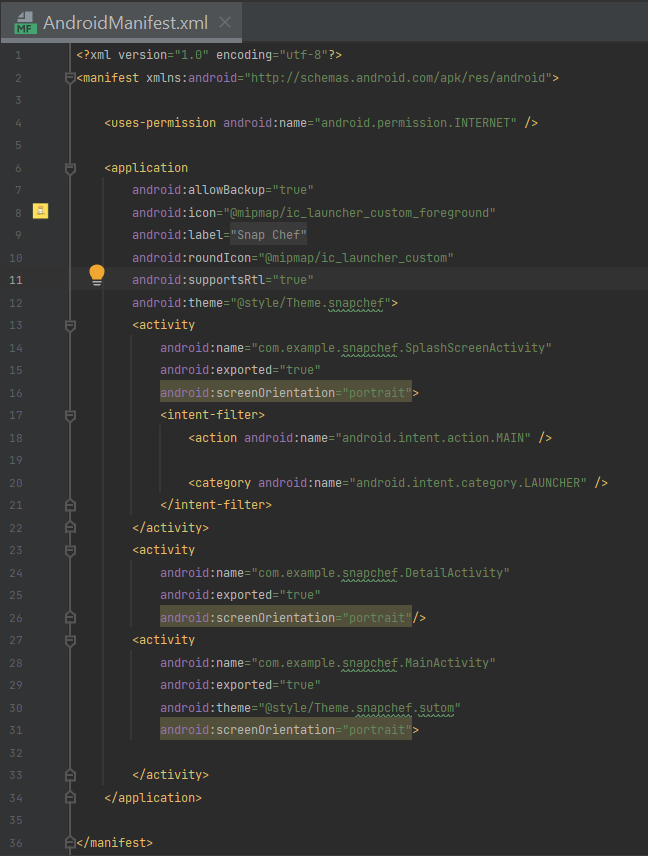
****

* View binding adalah fitur yang memudahkan kami menulis kode yang berinteraksi dengan tampilan. Setelah diaktifkan dalam sebuah modul, view binding akan menghasilkan class binding untuk setiap file tata letak XML yang ada dalam modul tersebut. Instance class binding berisi referensi langsung ke semua tampilan yang memiliki ID di tata letak yang terkait.
* *List* untuk menampilkan list pada Ingredients secara array atau berulang-berulang.
* *onCreate ()* adalah kondisi awal saat Activity baru diciptakan, inisialisasi savedInstanceState.

**

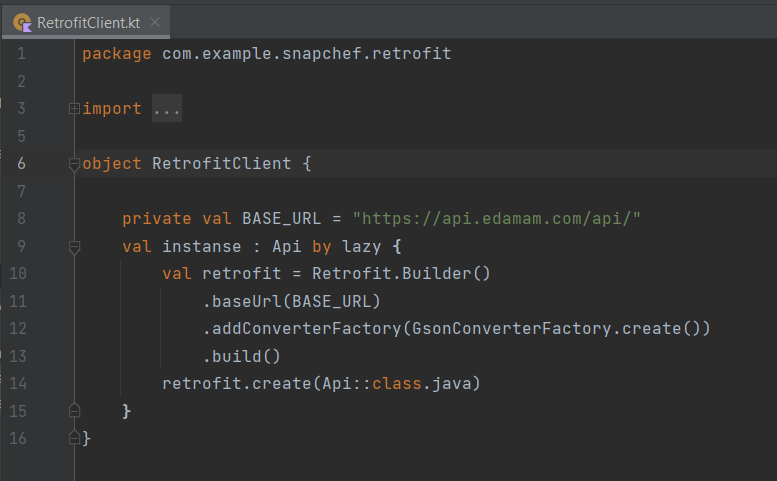
* *companion object* adalah membuat objek kelas anonim, yaitu kelas yang tidak dideklarasikan secara eksplisit dengan classdeklarasi. Kelas semacam itu berguna untuk penggunaan satu kali. Anda dapat mendefinisikannya dari awal, mewarisi dari kelas yang ada, atau mengimplementasikan antarmuka. Contoh kelas anonim juga disebut objek anonim karena ditentukan oleh ekspresi, bukan nama.

1. **Update AndroidManifest**

****

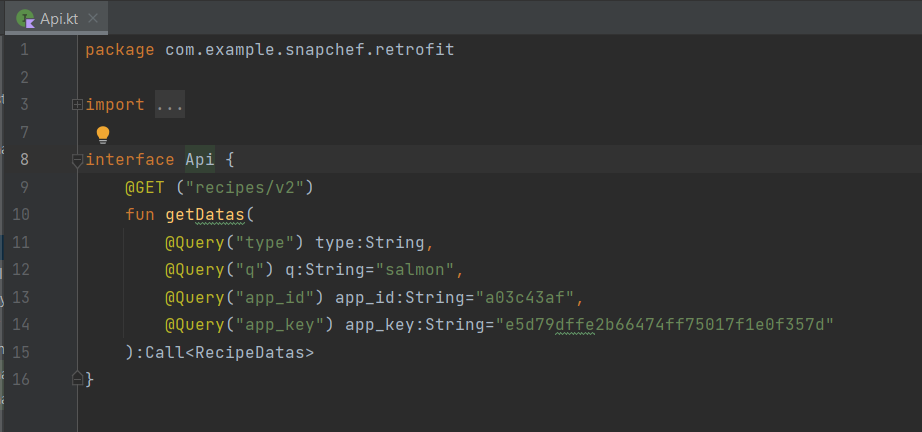
* Mengganti tampilan **LAUNCHER** ke SplashScreenActivity yang awal nya dari MainActivity.

1. **Menambahkan file RetrofitClient**



* *RetrofitClient* adalah sebuah file yang berisi tentang pengaturan database yang di ambil dari sebuah Api yang kami dapatkan dari internet untuk mendapatkan sebuah data resep makanan.

1. **Menambahkan file Api**



* *Api* adalah sebuah file yang berisi tentang pengaturan sebuah output id yang akan di gunakan dalam database yang di ambil dari sebuah Api resep makanan yang sudah di setting pada file *RetrofitClient*.