Nama : Akmal Muhamad Firdaus

NIM : 1301204188

**Jurnal Praktikum Modul 12 dan 13**

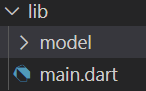
1. **Model**
   1. **Pengenalan Model**

Model mengacu pada bagian dari arsitektur aplikasi yang bertanggung jawab untuk mengelola data dan logika bisnis aplikasi. Model dalam Flutter biasanya digunakan untuk mengatur, memperoleh, dan memanipulasi data, serta menyediakan metode untuk berkomunikasi dengan lapisan lain dalam aplikasi, seperti tampilan (view) dan pengontrol (controller).

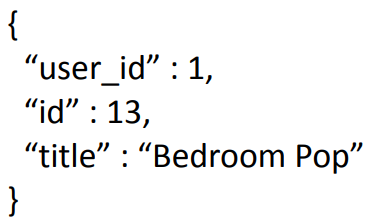
Model sendiri biasanya bagian yang bersentuhan langsung dengan database dan mengkonversinya menjadi class dart yang dapat diakses lebih mudah. Umumnya model akan dibuat dari response JSON.

* 1. **Membuat Model Class**

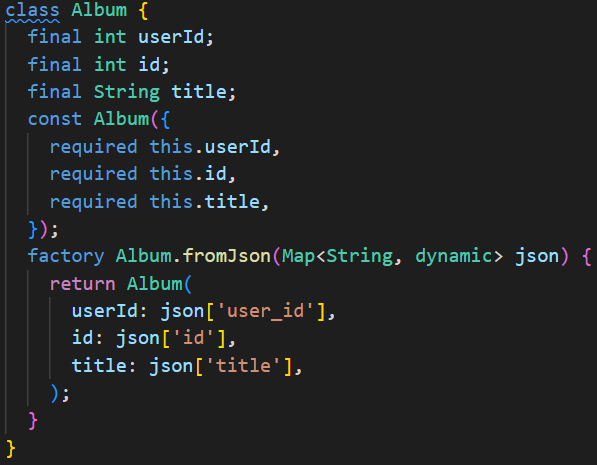
Untuk membuat model biasanya dimasukan kedalam direktori khusus, biasanya bernama model dalam direktori lib.



Setelah itu buat file dart baru pada direktori model. Sebagai contoh response json berikut akan dibuatkan model pada flutter.



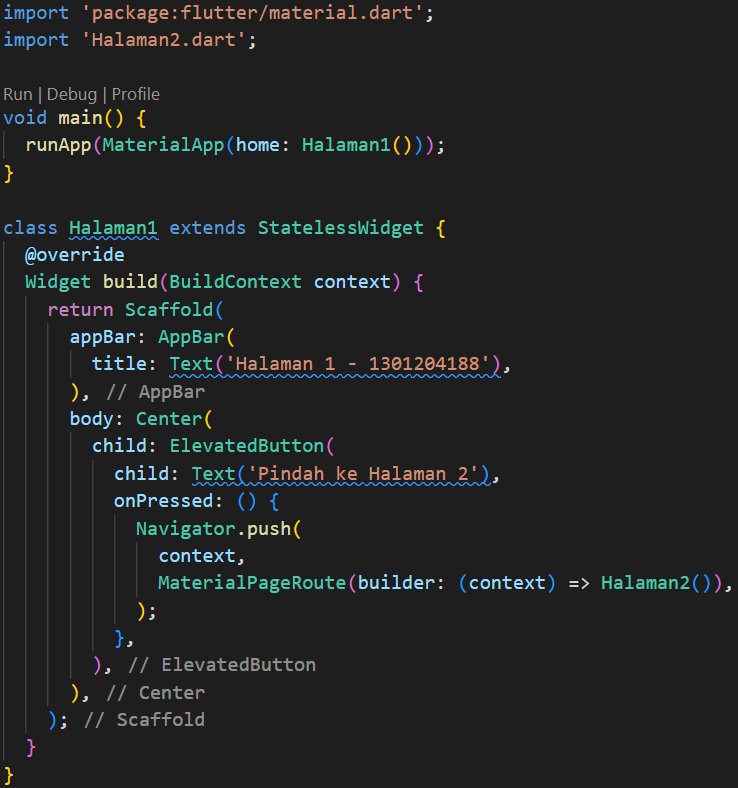
Maka bentuk dari model flutternya adalah sebagai berikut.



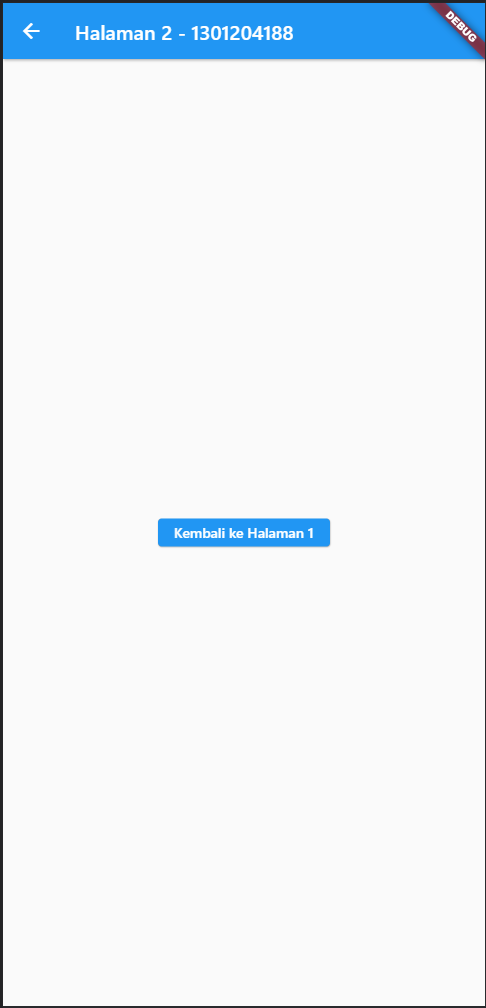
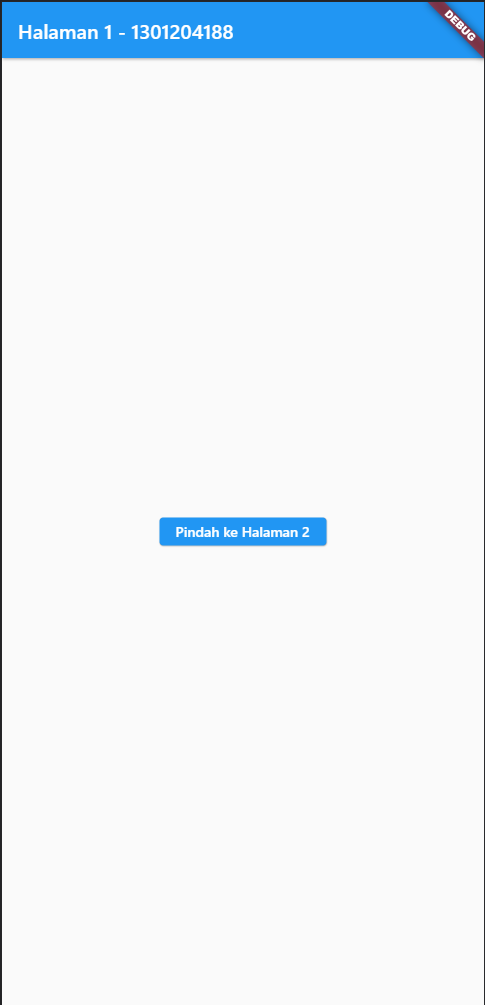
1. **Navigation**
   1. **Navigation Pindah Halaman**

Navigation Pindah Halaman adalah proses berpindah dari satu halaman ke halaman lain dalam aplikasi. Ini melibatkan perpindahan antara tampilan (widget) yang berbeda untuk menyajikan konten atau fitur yang berbeda kepada pengguna.

Flutter menyediakan berbagai cara untuk melakukan navigasi antar halaman, termasuk penggunaan widget Navigator, MaterialPageRoute, PageRouteBuilder, dan banyak lagi. Berikut adalah contoh menggunakan Navigator:

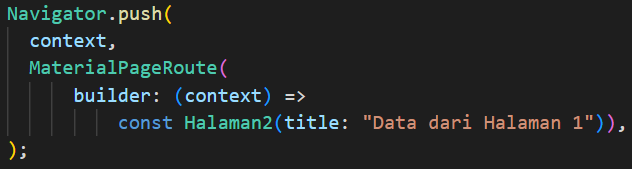


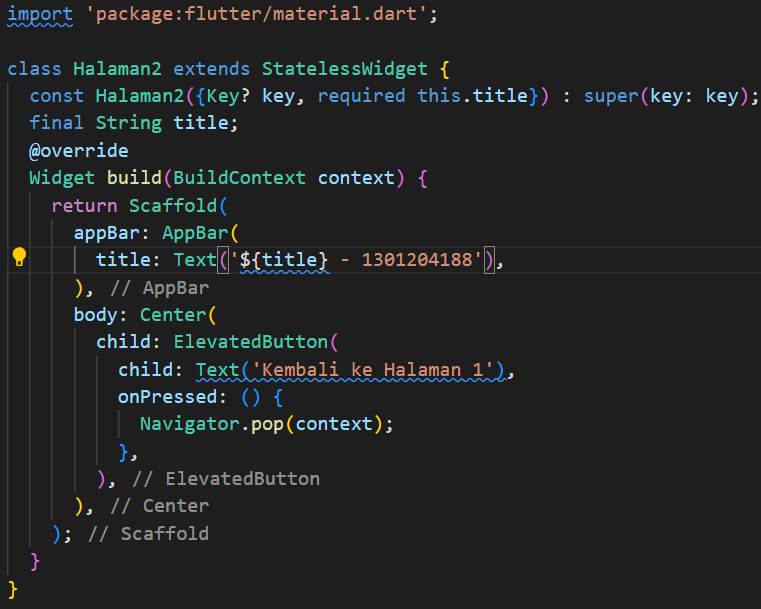


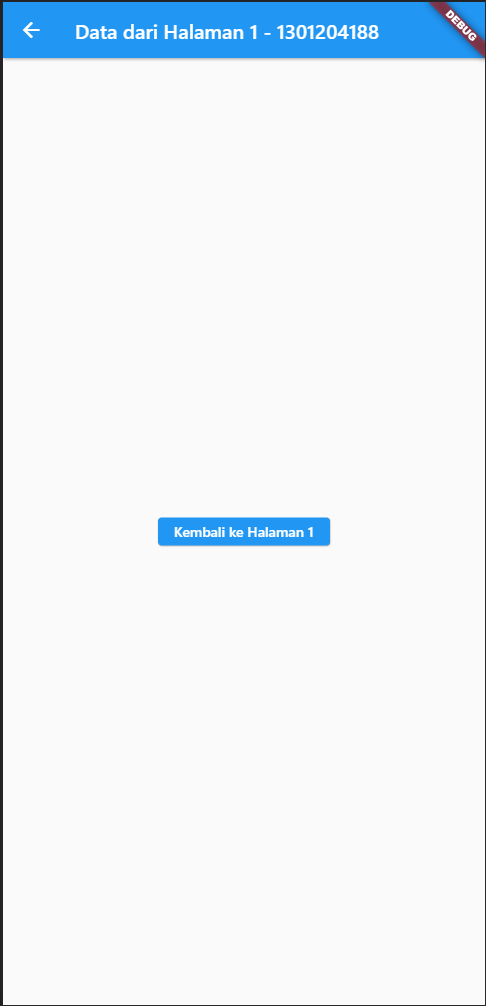


* 1. **Navigation Mengirim Data**

Untuk mengirim data antara halaman dalam aplikasi Flutter, kita dapat menggunakan parameter pada metode navigasi. Berikut adalah cara umum untuk mengirim data antara halaman:



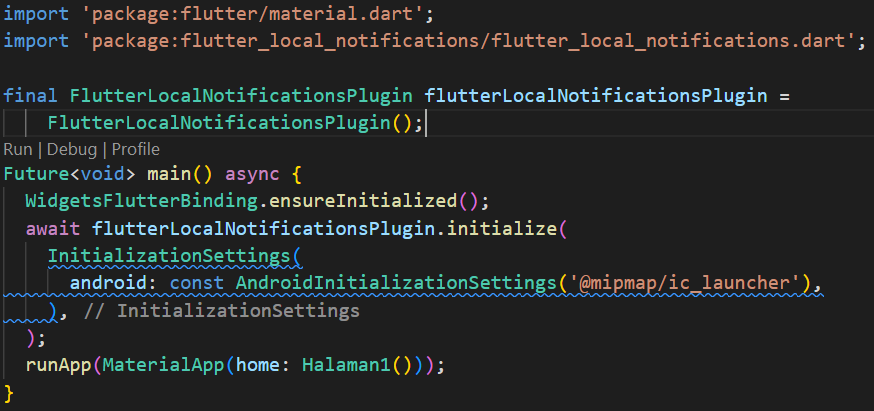




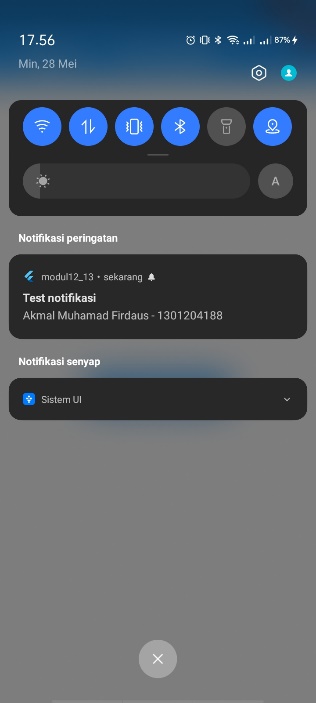
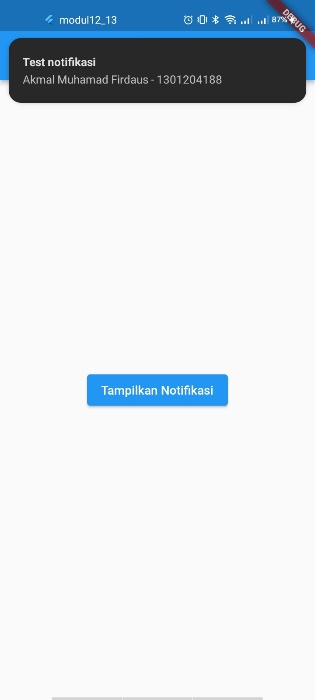
1. **Notification**

Notification pada Flutter adalah pesan yang ditampilkan kepada pengguna sebagai pemberitahuan atau informasi penting di luar jendela utama aplikasi. Notifikasi dapat muncul di bilah notifikasi atau panel notifikasi perangkat, memberikan pengguna informasi penting atau mengingatkan mereka tentang suatu kejadian atau tugas tertentu.

Berikut adalah contoh menggunakan notification pada flutter:



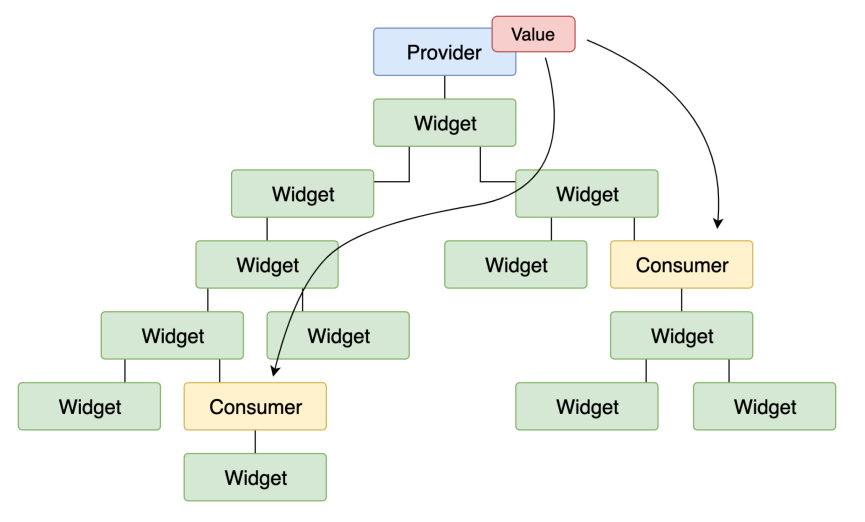




1. **State Management**
   1. **Pengenalan State Management**

State Management adalah cara mengelola dan memperbarui state (data) dalam aplikasi Flutter. State management diperlukan ketika aplikasi memiliki state yang kompleks, seperti data yang perlu diperbarui, disimpan, dan dibagikan di antara berbagai komponen atau widget dalam aplikasi.

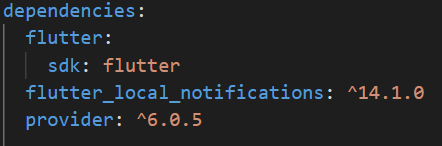
* 1. **State Management Provider**



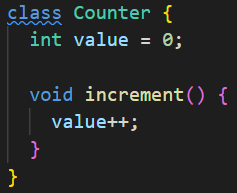
State Management Provider adalah library Flutter yang digunakan untuk manajemen state secara efisien dan sederhana. Library ini memungkinkan kita untuk membagikan dan mengakses state aplikasi secara global di seluruh widget-widget dalam hierarki widget Flutter tanpa harus melewati banyak tingkatan widget.

Library ini menyediakan provider dan consumer untuk menghubungkan state dengan widget yang membutuhkannya. Berikut adalah langkah-langkah dasar penggunaan Provider dalam manajemen state:

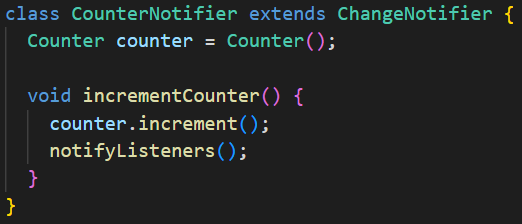
1. Tambahkan dependensi provider ke file pubspec.yaml pada proyek Flutter



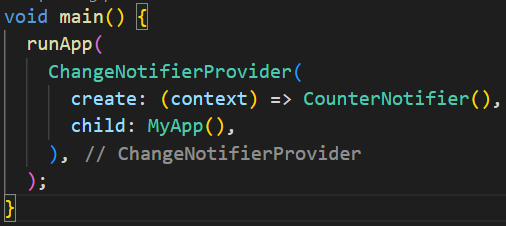
1. Jalankan perintah *flutter pub get* untuk mengunduh dan memperbarui dependensi
2. Buat sebuah class yang akan menjadi state atau model. Misalnya, jika Anda ingin mengelola state counter, Anda dapat membuat class Counter seperti ini:



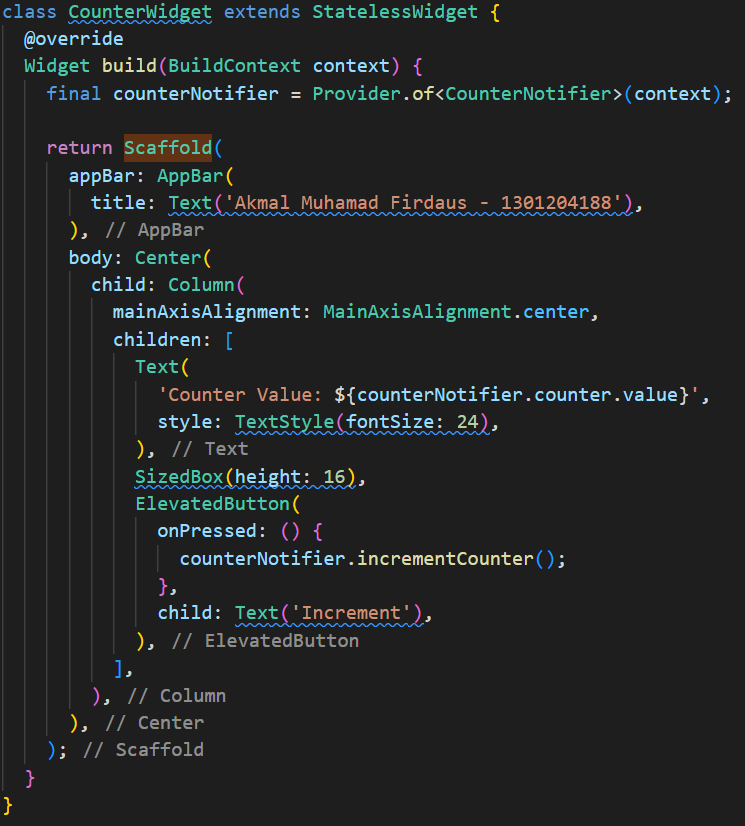
1. Buat class *ChangeNotifier* yang meng-extend class Counter dan akan mengubah state dan memberi tahu widget-widget yang bergantung pada perubahan state:

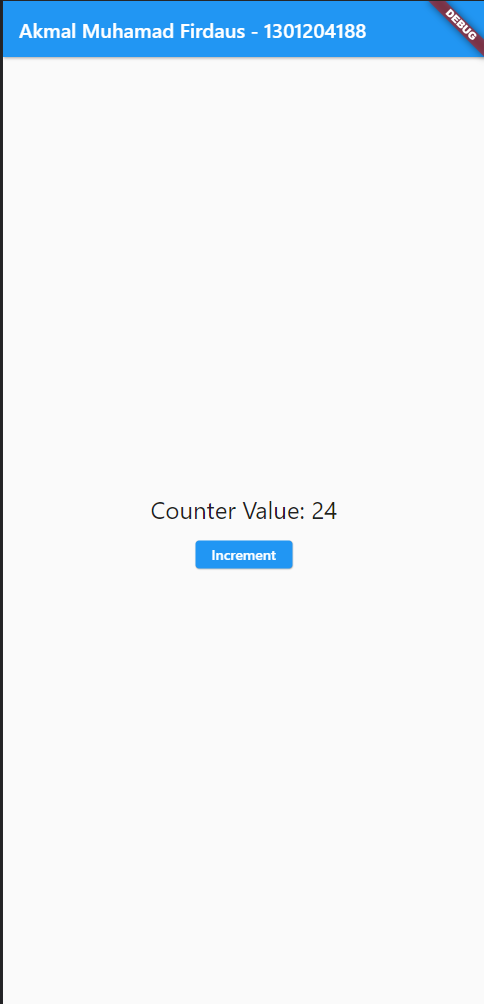
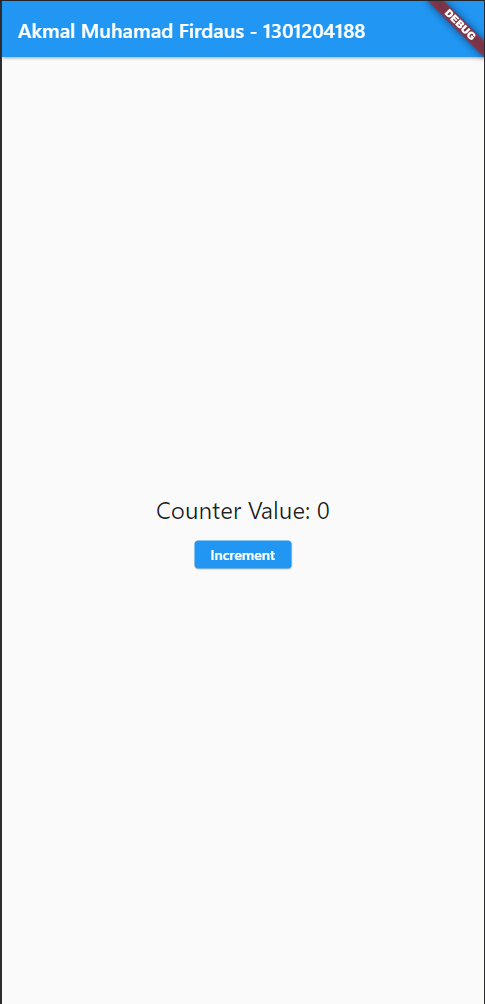


1. Di dalam widget utama aplikasi, wrap widget tersebut dengan *ChangeNotifierProvider*:



1. Sekarang, dalam widget mana pun di dalam aplikasi, kita dapat menggunakan Consumer untuk mengakses dan memperbarui state:



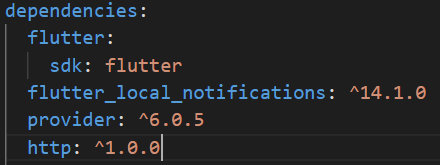


1. **Networking**

Networking adalah proses berkomunikasi antara aplikasi Flutter dengan server atau layanan luar menggunakan protokol HTTP. Dalam pengembangan aplikasi Flutter, terdapat beberapa paket atau library yang dapat digunakan untuk melakukan operasi jaringan, seperti pengambilan data dari API, mengirim permintaan ke server, atau mengunggah file.

Untuk dapat menggunakan library http pada proyek flutter, lakukan langkah-langkah berikut ini:

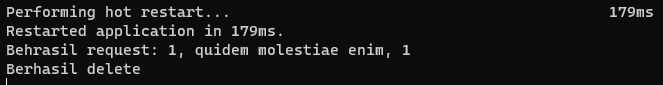
1. Tambahkan dependensi http ke file pubspec.yaml pada proyek Flutter:

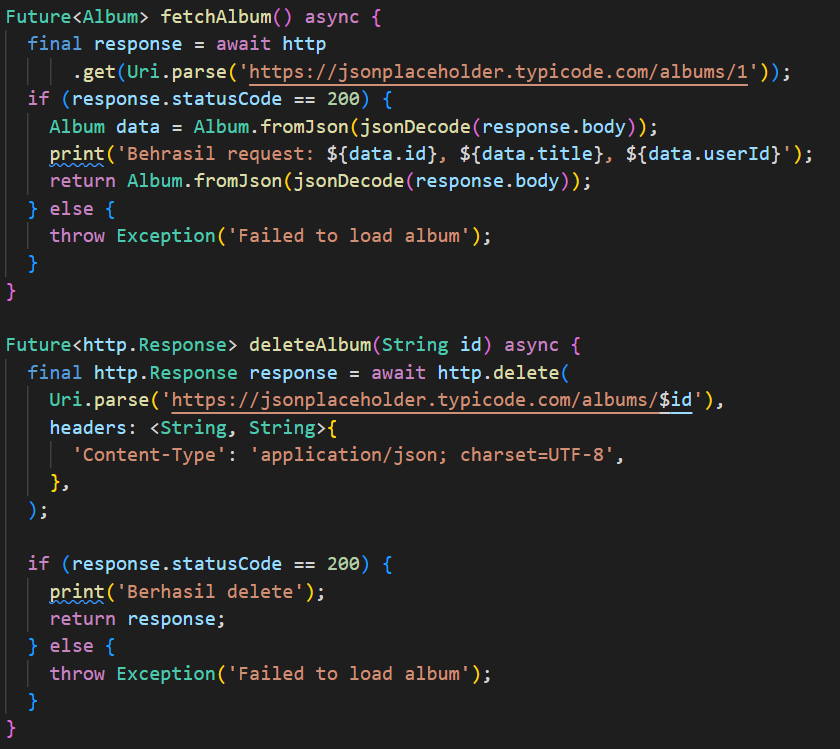


1. Jalankan perintah *flutter pub get* untuk mengunduh dan memperbarui dependensi.
2. Import paket http ke dalam file Dart tempat Anda akan menggunakan operasi jaringan:
   1. **Fetch & Delete Data**

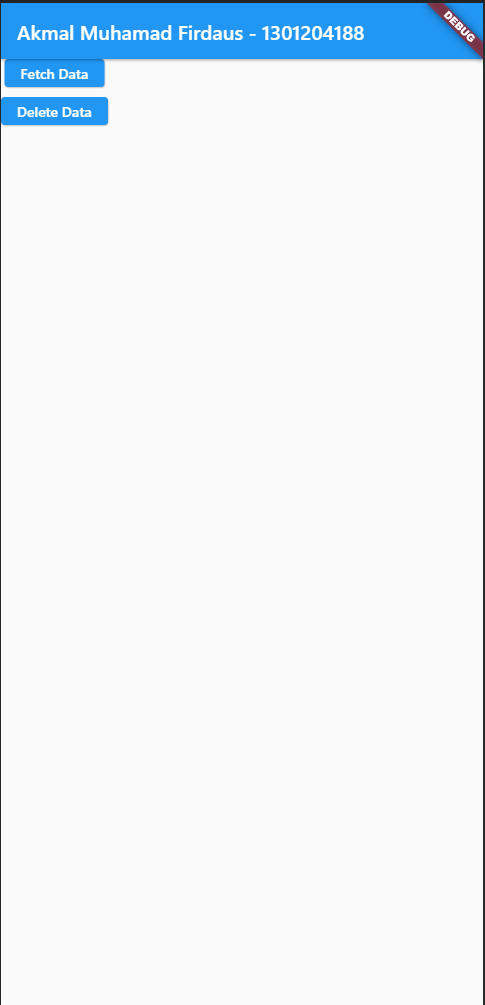
Fetch adalah Pengambilan data (fetching) dan Delete adalah penghapusan data (deleting). Fetch & Delete Data dapat merujuk pada proses mengambil (fetch) data dari server menggunakan permintaan HTTP (misalnya GET request) dan menghapus (delete) data dari server menggunakan permintaan HTTP (misalnya DELETE request).

Berikut adalah contoh Fetch dan Delete data



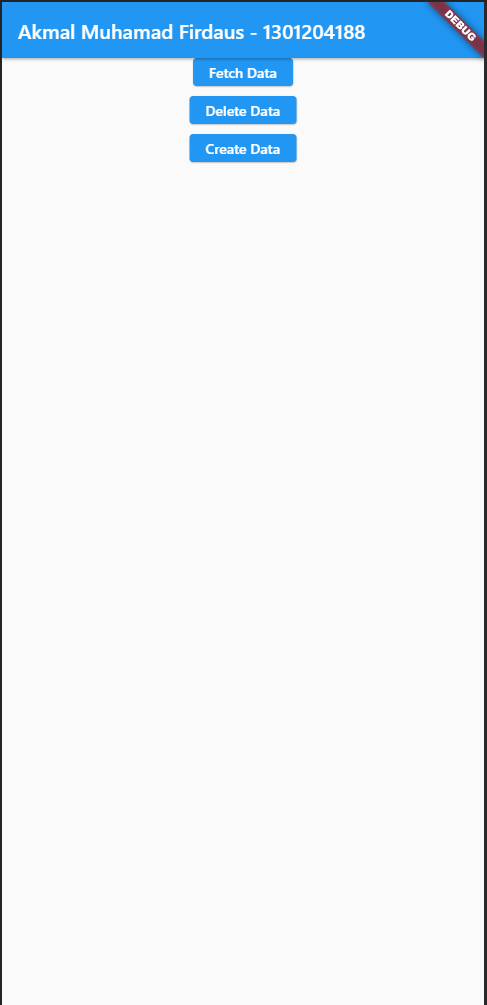


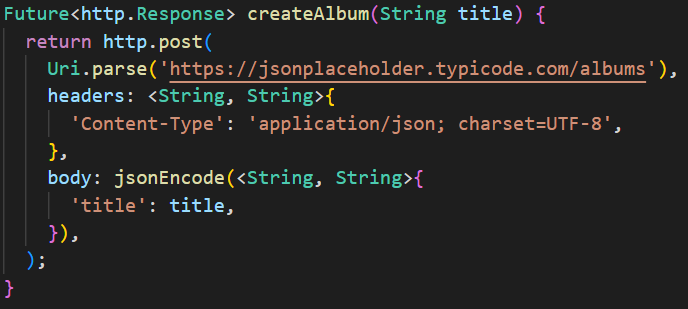


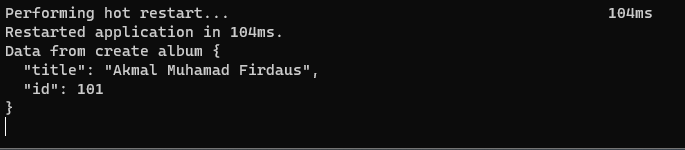


* 1. **Send & Update data**

Untuk melakukan send data dapat menggunakan post sedangkan untuk menupdate data dapat menggunakan method update put. Berikut adalah contohnya:







* 1. **Parse JSON**

Parse json sudah dilakukan sebelumnya pada poin 5.1 saat proses pengambilan data, lalu memparsing json tersebut ke model Album.



