**UAS**

**PRAKTIKUM PEMROGRAMAN MOBILE**

**JUDUL JOBSHEET**

**.**

**Untuk Memenuhi Ujian Akhir Semester**

**Mata Kuliah Praktikum Pemrograman Mobile**

**Dosen Pengampu: Safri Adam, S.Kom., M.Kom**

****

**Disusun Oleh:**

**Aryo Wibowo (NIM: 3202116031)**

**Alda Rahmanita (NIM: 3202116076)**

**Uchi Hardiana (NIM: 3202116077)**

**Walillah (NIM: 3202116032)**

**Ifdul Rahman (NIM:3202116033)**

**PROGRAM STUDI D3 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI PONTIANAK**

**2023**

**KATA PENGANTAR**

Puji dan syukur kami panjatkan kepada Allah SWT atas rahmat dan karunia-nya sehingga Ujian Akhir Semester” dapat terselesaikan dengan baik. ini merupakan salah satu UAS yang diberikan oleh dosen pengampu mata kuliah Praktikum Pemrograman Mobile kepada mahasiswa Program Studi D3 Teknik Informatika Jurusan Teknik Elektro.

Uas ini kami membahas tentang ..? di aplikasi mobile. Demikian laporan ini kami buat semoga bermanfaat.

Pontianak, 18 Agustus 2023

Penyusun,

(Kelompok ?)

**UJIAN AKHIR SEMESTER**

**Data**

**Source Code :**



Keterangan :

Kode tersebut menunjukkan penggunaan Hive untuk menyimpan dan memuat data dalam bentuk objek Note dengan memanfaatkan kotak (box) Hive. Ini adalah pendekatan efisien untuk mengelola data lokal dalam aplikasi Flutter.

**Model => Note data.dart**

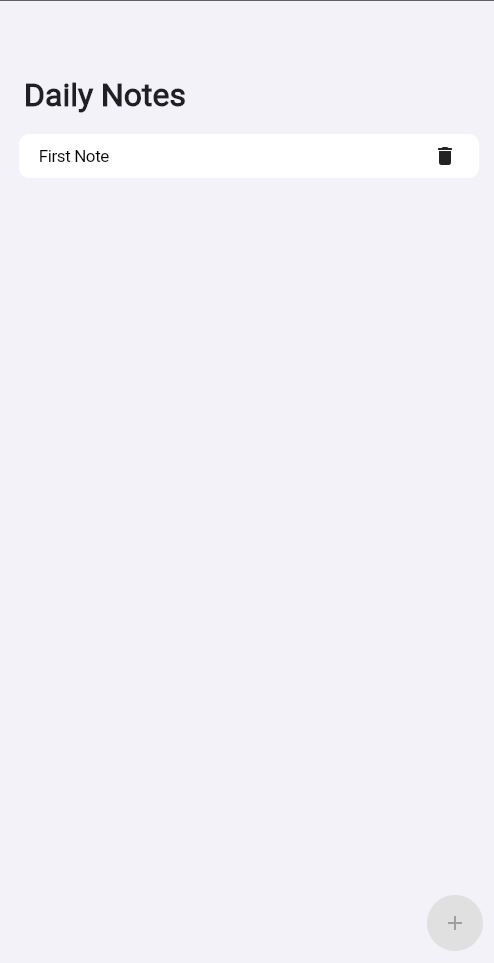
**Source Code :**

****

**Keterangan :**

Pada Kode Note.data.dart di atas meruupakan bagian dari sebuah aplikasi Flutter yang mengelola data catatan (notes) menggunakan Hive database. Ini merangkum kelas NoteData yang berperan sebagai penyedia data dan pengelola perubahan (ChangeNotifier) untuk catatan-catatan dalam aplikasi.

**Output :**

****

**Pages => editing\_note\_page.dart**

**Source Code :**

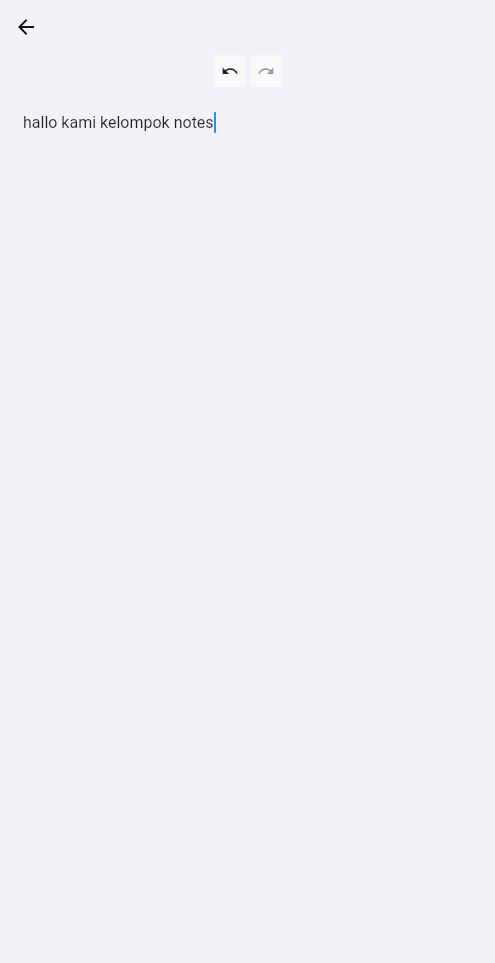
****

**Keterangan :**

Pada Kode di atas digunakan untuk mengedit catatan (notes) menggunakan editor teks yang disediakan oleh pustaka flutter\_quill. Ini merangkum halaman EditingNotePage yang berfungsi untuk menambah atau mengedit catatan dalam aplikasi.

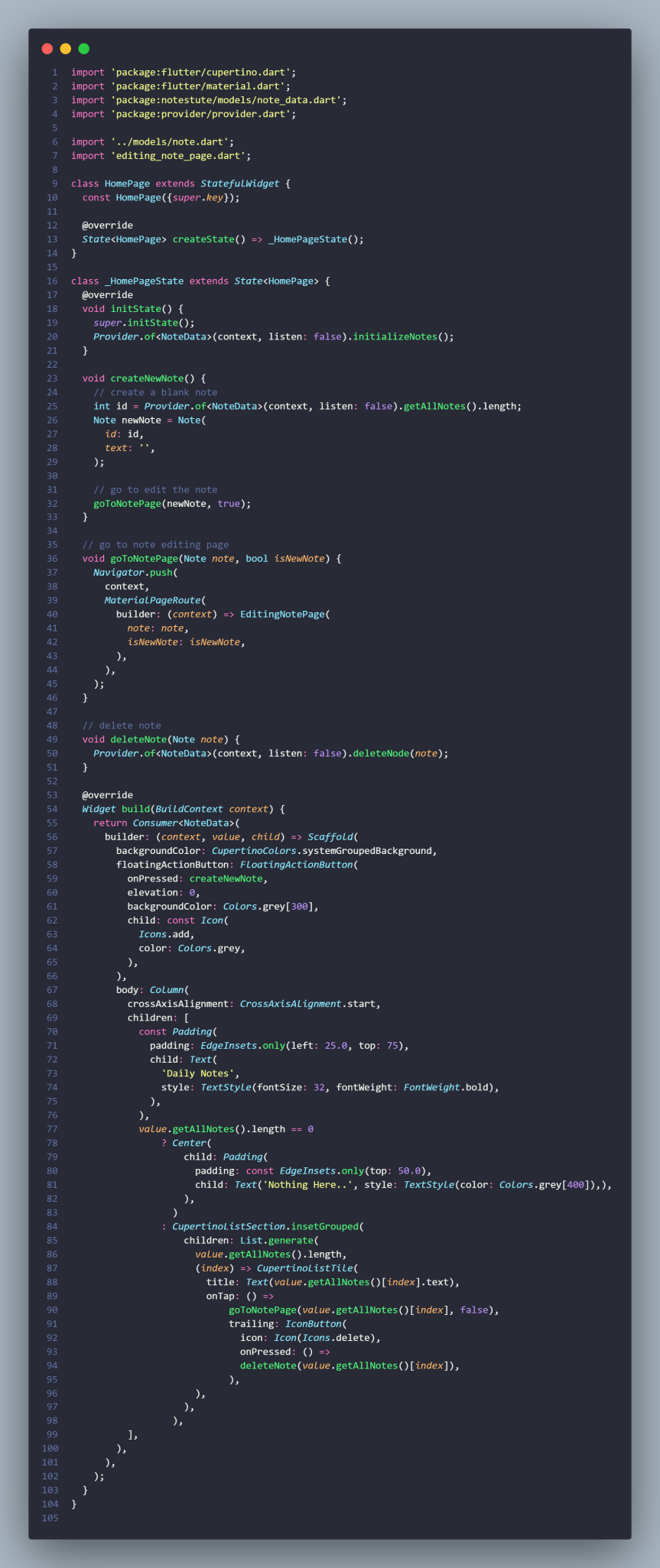
Pada dasarnya, kelas EditingNotePage ini mengatur tampilan dan logika interaksi untuk mengedit dan menyimpan catatan menggunakan editor teks Quill. Hal ini juga berinteraksi dengan NoteData melalui penyedia data (Provider) untuk menambah atau mengupdate catatan.

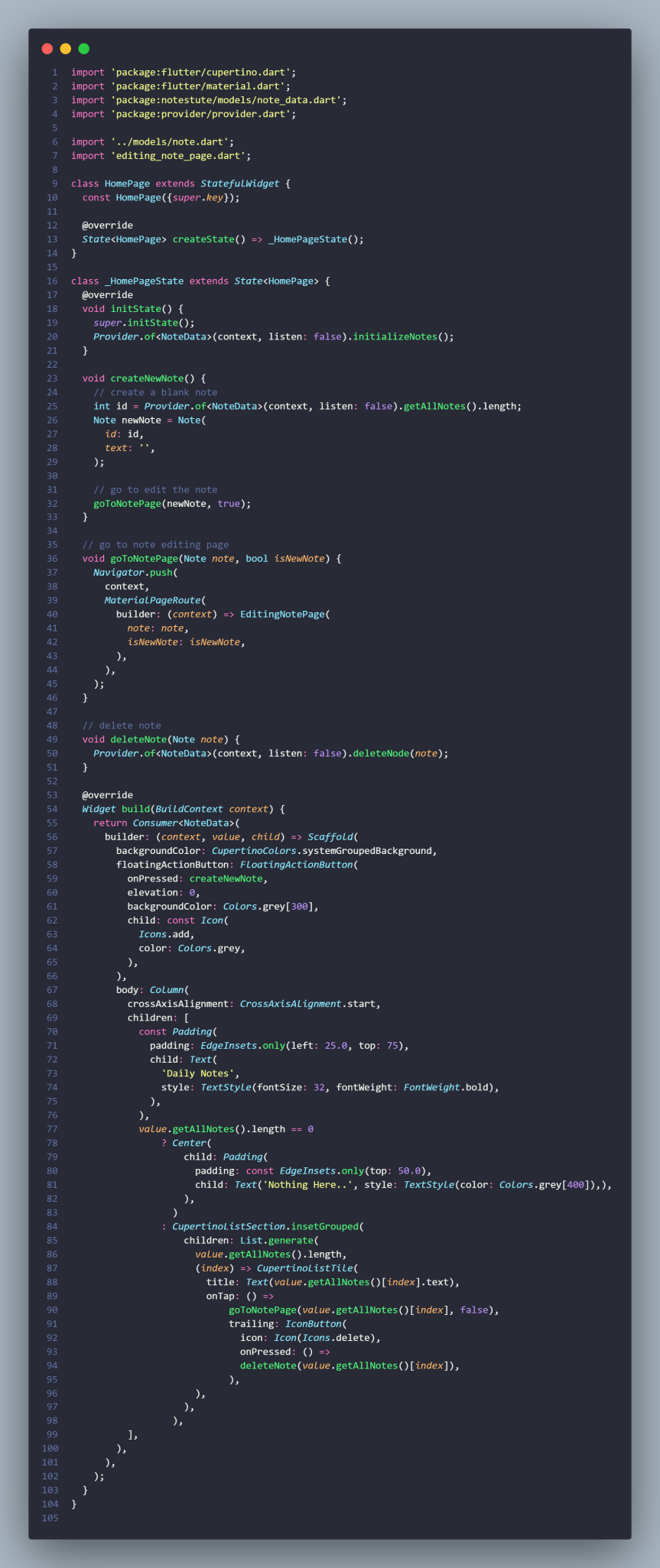
**Output :**

****

**Pages => home\_page.dart**

**Source Code :**

****

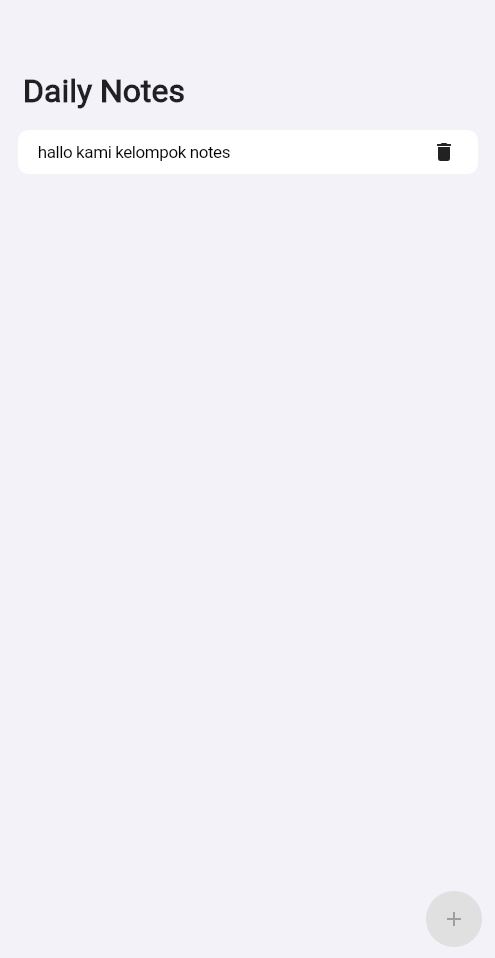
****

**Keterangan :**

Kode di atas adalah bagian dari aplikasi Flutter yang menampilkan daftar catatan (notes) pada halaman utama (HomePage). Ini menggunakan penyedia data NoteData dari Provider untuk mengelola catatan-catatan dalam aplikasi dan memungkinkan pengguna menambah, mengedit, dan menghapus catatan.

Kelas HomePage ini mengatur tampilan dan interaksi pada halaman utama, serta menggunakan Consumer untuk mendengarkan perubahan data catatan dan merefresh tampilan ketika terjadi perubahan.

**Output :**

****

**Pages => main.dart**

**Source Code :**

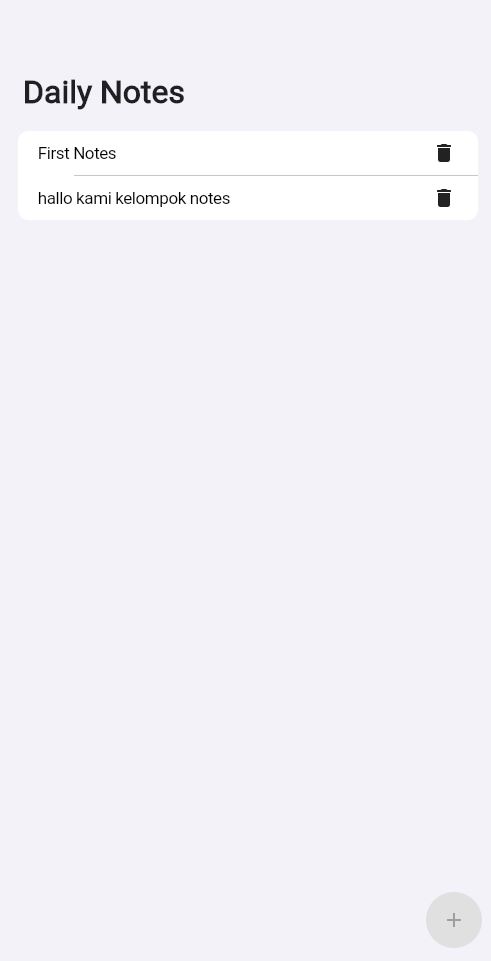
****

**Keterangan :**

Kode di atas menggunakan Hive database untuk mengelola catatan (notes) dan menyediakan tampilan melalui MaterialApp. Ini merangkum langkah-langkah utama dalam inisialisasi dan peluncuran aplikasi.

Dengan menggabungkan pustaka Hive, penyedia data Provider, dan tampilan menggunakan MaterialApp, kode ini menginisialisasi dan menjalankan aplikasi Flutter yang dapat mengelola catatan dan menampilkannya pada halaman utama (HomePage).

**Output :**



**TEST=> model =>android.device\_info\_fake.dart**

**Source code :**

****

**Keterangan :**

Kode di atas mendefinisikan sejumlah variabel konstan yang menyimpan informasi palsu (dummy) tentang perangkat Android dan lingkungan sistem. Ini biasanya digunakan untuk pengujian atau simulasi dalam pengembangan aplikasi.

**TEST=> model =>android.device\_info\_test.dart**

**Source code :**

****

**Keterangan :**

Kode di atas adalah bagian dari pengujian unit untuk kelas AndroidDeviceInfo dalam pustaka device\_info\_plus. Ini merangkum serangkaian pengujian yang memeriksa konversi dari dan ke objek AndroidDeviceInfo menggunakan data palsu (dummy).

**TEST=>Model => ios\_device\_info\_test.dart**

**Source Code :**

****

**Keterangan :**

Kode di atas adalah bagian dari pengujian unit untuk kelas IosDeviceInfo dalam pustaka device\_info\_plus. Ini merangkum serangkaian pengujian yang memeriksa konversi dari dan ke objek IosDeviceInfo menggunakan data palsu (dummy) untuk lingkungan sistem iOS.

**TEST=>Model => Linux\_device\_info\_test.dart**

****

**Keterangan :**

Kode di atas adalah bagian dari pengujian unit untuk kelas LinuxDeviceInfo dalam pustaka device\_info\_plus. Ini merangkum satu pengujian yang memeriksa konversi dari objek LinuxDeviceInfo ke peta (map) dengan menggunakan data yang ditentukan.

**TEST=>Model => macos\_device\_info\_test.dart**

**Source Code :**

****

**Keterangan :**

Kode di atas adalah bagian dari pengujian unit untuk kelas MacOsDeviceInfo dalam pustaka device\_info\_plus. Ini merangkum serangkaian pengujian yang memeriksa konversi dari dan ke objek MacOsDeviceInfo menggunakan data yang ditentukan.

**TEST=>Model => windows\_device\_info\_test.dart**

**Source Code :**

****

**Keterangan :**

Kode di atas adalah bagian dari pengujian unit untuk kelas WindowsDeviceInfo dalam pustaka device\_info\_plus. Ini merangkum satu pengujian yang memeriksa konversi dari objek WindowsDeviceInfo ke peta (map) dengan menggunakan data yang ditentukan.

**MACOS=> Classes => Src => device\_info\_plus.pudspec**

**Source Code :**

****

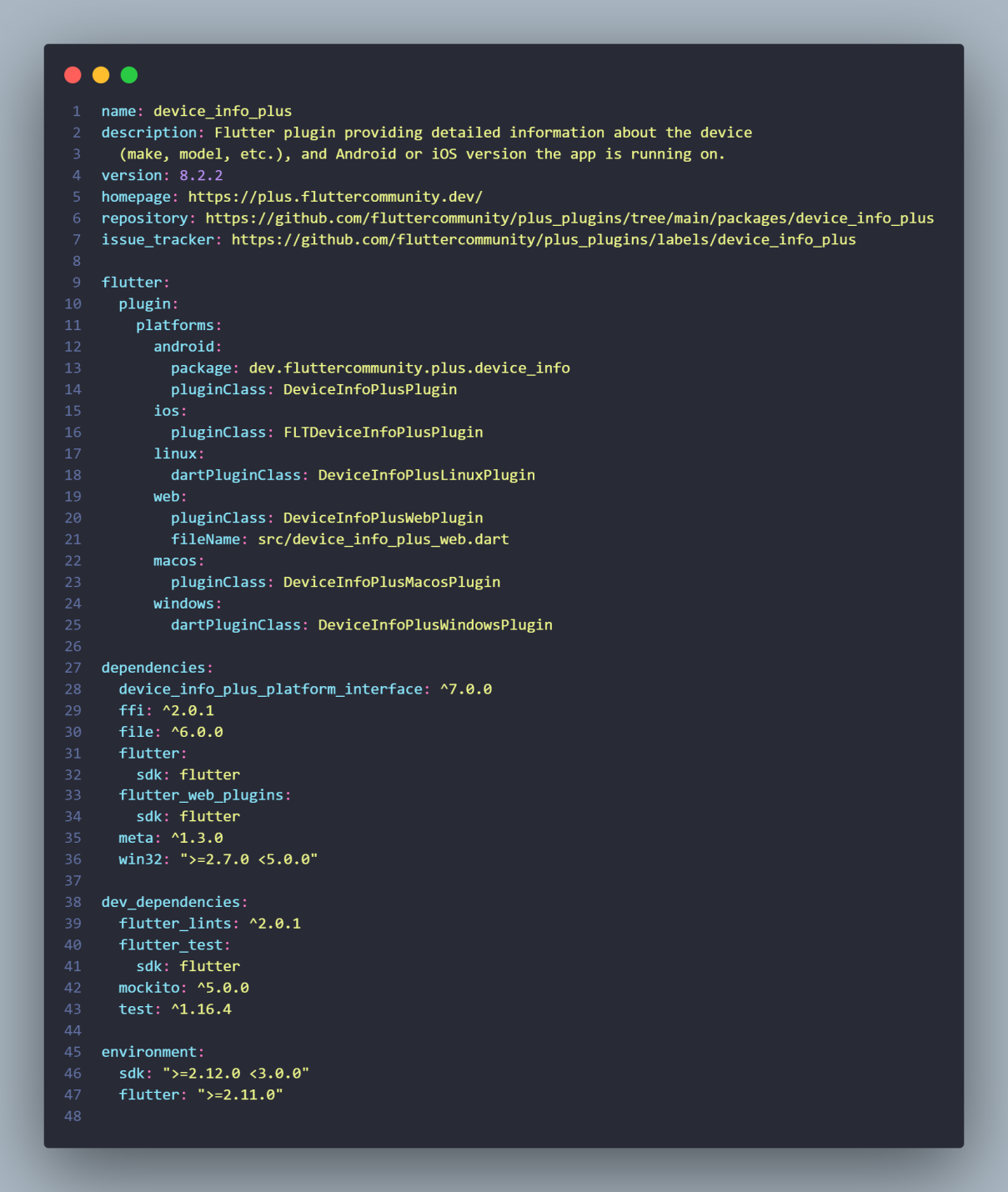
****

**Keterangan :**

Kode tersebut adalah bagian dari pengujian unit untuk kelas WebBrowserInfo dalam pustaka device\_info\_plus. Ini merangkum dua pengujian yang memeriksa fungsionalitas kelas WebBrowserInfo dalam mengambil informasi tentang peramban web.

**Pubspec.yaml**

**Source Code :**

****

**SEKIAN DAN TERIMAKASIH**