

Setelah itu ubah **widget** menjadi `GestureDetector` dan kita juga menambahkan `onTap` yang kemudian akan membuka halaman Detail Produk, sehingga kode untuk Class `ItemProduk` menjadi

```
class ItemProduk extends StatelessWidget {
  final String? kodeProduk;
  final String? namaProduk;
  final int? harga;

  const ItemProduk({Key? key, this.kodeProduk, this.namaProduk, this.harga})
    : super(key: key);

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      child: Card(
        child: ListTile(
          title: Text(namaProduk.toString()),
          subtitle: Text(harga.toString()),
        ),
      ),
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => ProdukDetail(
              kodeProduk: kodeProduk,
              namaProduk: namaProduk,
              harga: harga,
            )),
        );
      },
    );
  }
}
```

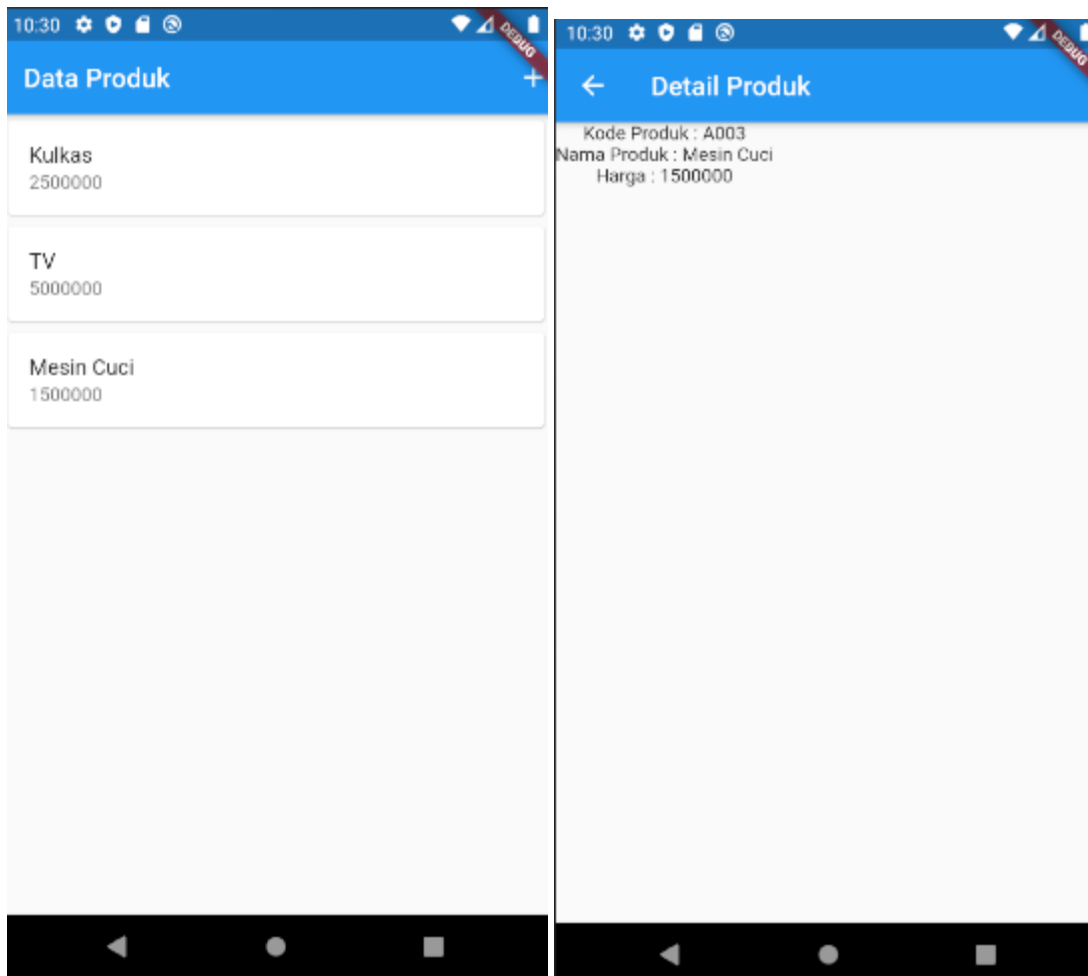
Adapun untuk keseluruhan kode pada `produk_page.dart` sebagai berikut

```
1. import 'package:aplikasi_flutter_pertamaku/ui/produk_detail.dart';
2. import 'package:aplikasi_flutter_pertamaku/ui/produk_form.dart';
3. import 'package:flutter/material.dart';
4.
5. class ProdukPage extends StatefulWidget {
6.   const ProdukPage({Key? key}) : super(key: key);
7.
8.   @override
9.   _ProdukPageState createState() => _ProdukPageState();
10. }
11.
12. class _ProdukPageState extends State<ProdukPage> {
13.   @override
14.   Widget build(BuildContext context) {
15.     return Scaffold(
16.       appBar: AppBar(
17.         title: const Text('Data Produk'),
18.         actions: [
19.           GestureDetector(
20.             // menampilkan icon +
21.             child: const Icon(Icons.add),
22.             onTap: () async {
23.               // berpindah ke halaman ProdukForm
```

```

24.         Navigator.push(
25.             context,
26.             MaterialPageRoute(
27.                 builder: (context) => const ProdukForm()));
28.         })
29.     ],
30. ),
31. body: ListView(
32.     children: const [
33.         // List 1
34.         ItemProduk(
35.             kodeProduk: "A001",
36.             namaProduk: "Kulkas",
37.             harga: 2500000,
38.         ),
39.         // List 2
40.         ItemProduk(
41.             kodeProduk: "A002",
42.             namaProduk: "TV",
43.             harga: 5000000,
44.         ),
45.         // List 3
46.         ItemProduk(
47.             kodeProduk: "A003",
48.             namaProduk: "Mesin Cuci",
49.             harga: 1500000,
50.         ),
51.     ],
52. ),
53. );
54. }
55. }
56.
57. class ItemProduk extends StatelessWidget {
58.     final String? kodeProduk;
59.     final String? namaProduk;
60.     final int? harga;
61.
62.     const ItemProduk({Key? key, this.kodeProduk, this.namaProduk, this.harga})
63.         : super(key: key);
64.
65.     @override
66.     Widget build(BuildContext context) {
67.         return GestureDetector(
68.             child: Card(
69.                 child: ListTile(
70.                     title: Text(namaProduk.toString()),
71.                     subtitle: Text(harga.toString()),
72.                 ),
73.             ),
74.             onTap: () {
75.                 Navigator.push(
76.                     context,
77.                     MaterialPageRoute(
78.                         builder: (context) => ProdukDetail(
79.                             kodeProduk: kodeProduk,
80.                             namaProduk: namaProduk,
81.                             harga: harga,
82.                         ));
83.             },
84.         );
85.     }
86. }

```



Membuat projek flutter yang terhubung dengan API

Apa itu API

API atau Application Programming Interface adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. Jadi, API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform.

Perumpamaan yang bisa digunakan untuk menjelaskan API adalah seorang pelayan di restoran. Tugas pelayan tersebut adalah menghubungkan tamu restoran dengan juru masak. Tamu cukup memesan makanan sesuai daftar menu yang ada dan pelayan memberitahukannya ke juru masak. Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan.

Itulah gambaran tugas dari API dalam pengembangan aplikasi.