PERTEMUAN 11

Menambahkan Depedencies

Pada Editor Offline (VSCode)

Dalam pembuatan aplikasi ini dibutuhkan depedensi untuk menerima dan mengirim request ke Rest API (**dio**). Pastikan komputer atau laptop terhubung ke internet, buka file **pubspec.yaml** kemudian pada bagian **dependencies** tambahkan kode berikut

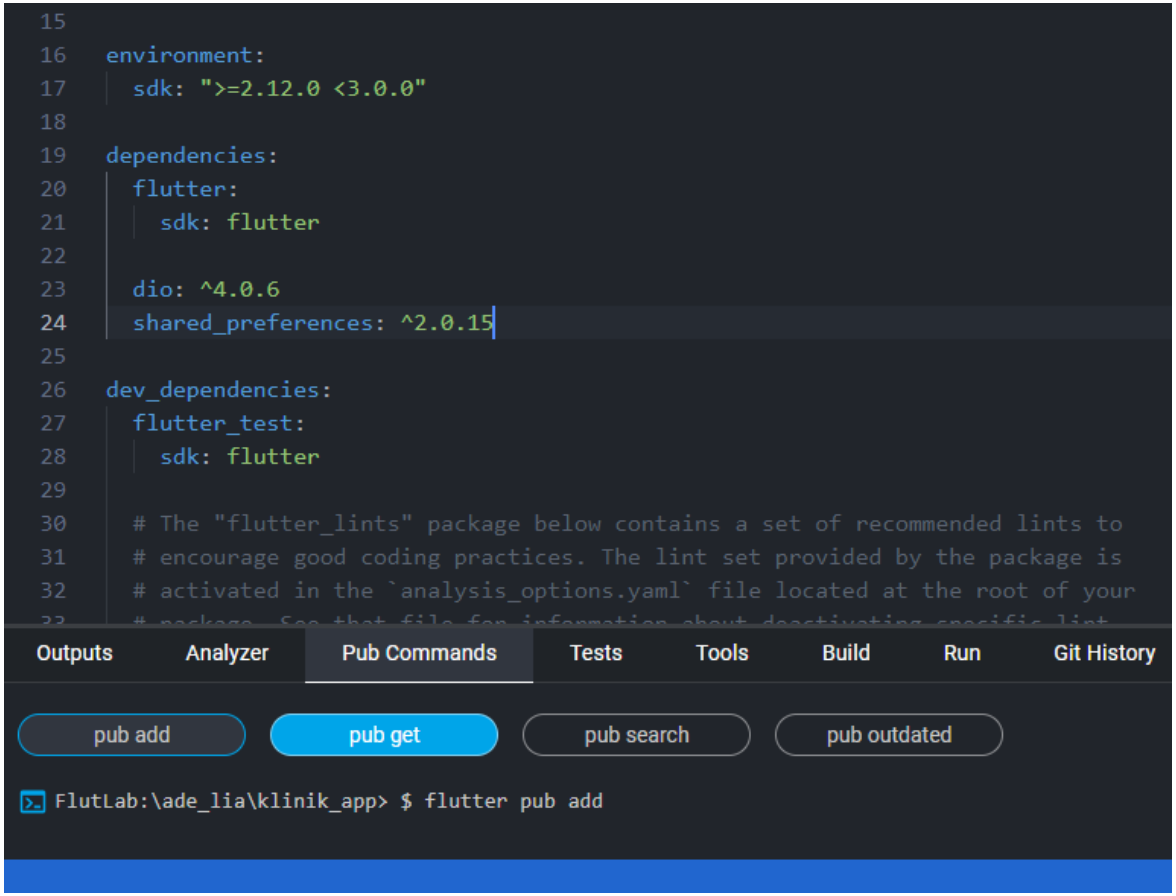
```
29. dependencies:
30.   flutter:
31.     sdk: flutter
32.
33.
34.   # The following adds the Cupertino Icons font to your application.
35.   # Use with the CupertinoIcons class for iOS style icons.
36.   cupertino_icons: ^1.0.2
37.   dio: ^4.0.6
38.   shared_preferences: ^2.0.15
39.
40. dev_dependencies:
41.   flutter_test:
42.     sdk: flutter
```

Simpan file tersebut, dan secara otomatis VS Code akan mengunduh depedencies tersebut, jika tidak berhasil, untuk mengunduh depedencies atau package yang telah ditambahkan, dapat dilakukan dengan membuka **CommandPrompt** kemudian masuk ke folder projek dan ketikkan **flutter pub get** kemudian tekan Enter

```
[perpustakaan_app] flutter pub get
Running "flutter pub get" in perpustakaan_app...
exit code 0
```

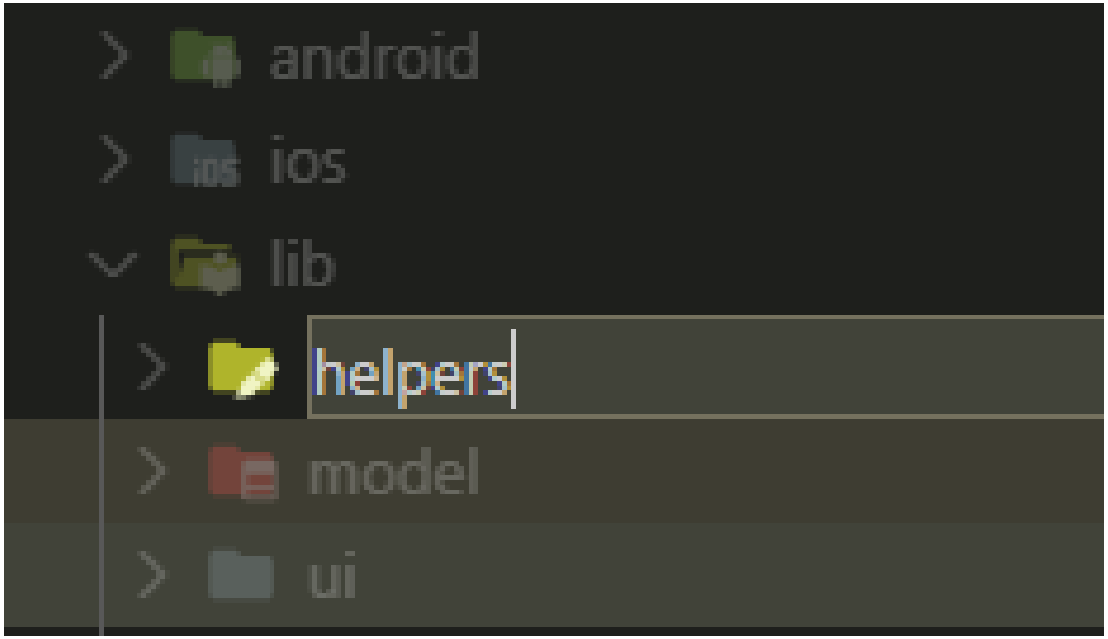
Pada Editor Online (flutlab.io)

Buka file **pubspec.yaml** dibagian bawah **depedencies:** tambahkan depedensi dio kemudian simpan, dibagian bawah editor terdapat tab **Pub Commands**, pilih tab tersebut kemudian klik **pub get**



Membuat Class Untuk Menyimpan Token

Pada bagian ini kita akan membuat class yang akan menyimpan token dan data pengguna saat berhasil login. Buat sebuah folder dengan nama helpers



Kemudian pada folder helpers buat sebuah file dengan nama **user_info.dart** dan masukkan kode berikut

```dart
import 'package:shared_preferences/shared_preferences.dart';

const String TOKEN = "token";
const String USER_ID = "userID";
const String USERNAME = "username";

class UserInfo {
  Future setToken(String value) async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
    return pref.setString(TOKEN, value);
  }

  Future<String?> getToken() async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
    return pref.getString(TOKEN);
  }

  Future setUserID(String value) async {
    final SharedPreferences pref = await SharedPreferences.getInstance();
    return pref.setString(USER_ID, value);
  }
```

```
22.
23.   Future<String> getUserID() async {
24.     final SharedPreferences pref = await SharedPreferences.getInstance();
25.     return pref.getString(USER_ID).toString();
26.   }
27.
28.   Future setUsername(String value) async {
29.     final SharedPreferences pref = await SharedPreferences.getInstance();
30.     return pref.setString(USERNAME, value);
31.   }
32.
33.   Future<String> getUsername() async {
34.     final SharedPreferences pref = await SharedPreferences.getInstance();
35.     return pref.getString(USERNAME).toString();
36.   }
37.
38.   Future<void> logout() async {
39.     final SharedPreferences pref = await SharedPreferences.getInstance();
40.     pref.clear();
41.   }
42. }
```

Mengarahkan Halaman Awal ke Login

Buka kembali file **main.dart** kita akan memodifikasi file tersebut dengan kondisi jika belum login maka akan membuka halaman login, namun jika sudah login maka akan membuka halaman beranda

```
1.  import 'package:flutter/material.dart';
2.  import '/helpers/user_info.dart';
3.  import '/ui/beranda.dart';
4.  import '/ui/login.dart';
5.
6.  Future<void> main() async {
7.    WidgetsFlutterBinding.ensureInitialized();
8.    var token = await UserInfo().getToken();
9.    print(token);
10.   runApp(MaterialApp(
11.     title: "Klinik APP",
12.     debugShowCheckedModeBanner: false,
13.     home: token == null ? Login() : Beranda(),
14.   ));
15. }
```

Persiapan Koneksi ke API

Buat sebuah fie dengan nama **api_client.dart** pada folder **helpers** kemudian ketikkan kode berikut

```dart
import 'package:dio/dio.dart';

final Dio dio = Dio(BaseOptions(
    baseUrl: 'https://63588b60c27556d2893f7a12.mockapi.io/',
    connectTimeout: 5000,
    receiveTimeout: 3000));

class ApiClient {
  Future<Response> get(String path) async {
    try {
      final response = await dio.get(Uri.encodeFull(path));
      return response;
    } on DioError catch (e) {
      throw Exception(e.message);
    }
  }

```

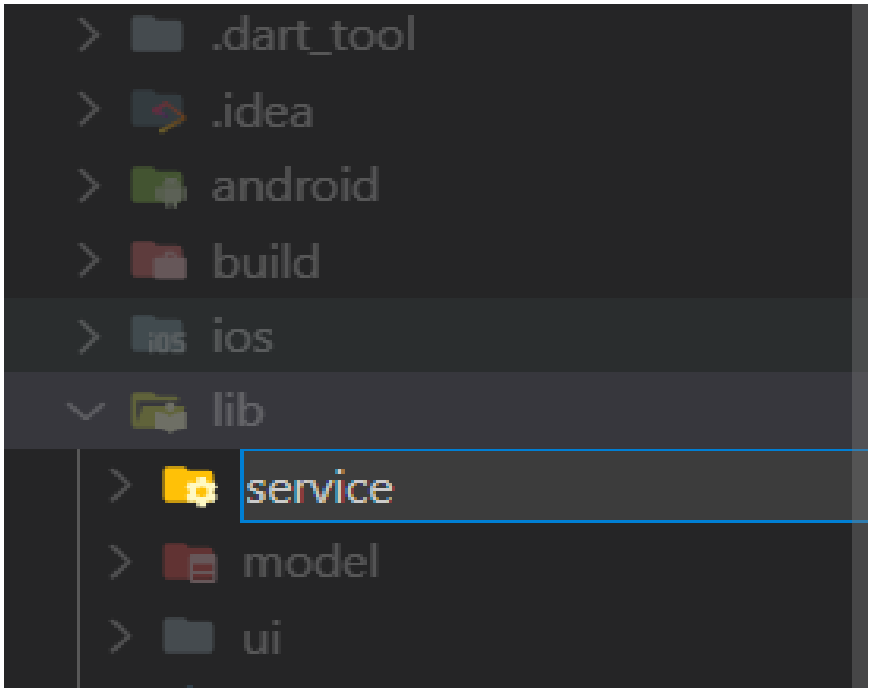
```
18.  Future<Response> post(String path, dynamic data) async {
19.    try {
20.      final response = await dio.post(Uri.encodeFull(path), data: data);
21.      return response;
22.    } on DioError catch (e) {
23.      throw Exception(e.message);
24.    }
25.  }
26.
27.  Future<Response> put(String path, dynamic data) async {
28.    try {
29.      final response = await dio.put(Uri.encodeFull(path), data: data);
30.      return response;
31.    } on DioError catch (e) {
32.      throw Exception(e.message);
33.    }
```

```
34.   }
35.
36.   Future<Response> delete(String path) async {
37.     try {
38.       final response = await dio.delete(Uri.encodeFull(path));
39.       return response;
40.     } on DioError catch (e) {
41.       throw Exception(e.message);
42.     }
43.   }
44. }
```

Pada baris 4 (`baseUrl: 'https://63588b60c27556d2893f7a12.mockapi.io/'`) Sesuaikan dengan API Endpoint pada mockapi.io yang telah dibuat sebelumnya

Membuat Service Login

Buat sebuah folder dengan nama **service** pada folder **lib**



Pada bagian ini kita akan membuat dummy login. Buat sebuah file pada folder **service** dengan nama **login_service.dart** kemudian ketikkan kode berikut

```
1.  import '../helpers/user_info.dart';
2.
3.  class LoginService {
4.    Future<bool> login(String username, String password) async {
5.      bool isLogin = false;
6.      if (username == 'admin' && password == 'admin') {
7.        await UserInfo().setToken("admin");
8.        await UserInfo().setUserID("1");
9.        await UserInfo().setUsername("admin");
10.       isLogin = true;
11.     }
12.     return isLogin;
13.   }
14. }
```

Kemudian buka file **login.dart** yang ada pada folder **ui**. Pada bagian fungsi tombol login kita ubah menjadi seperti berikut

```
Widget _tombolLogin() {
    return Container(
        width: MediaQuery.of(context).size.width,
        child: ElevatedButton(
            child: Text("Login"),
            onPressed: () async {
```

```
            String username = _usernameCtrl.text;
            String password = _passwordCtrl.text;
            await LoginService().login(username, password).then((value) {
              if (value == true) {
                Navigator.pushReplacement(context,
                    MaterialPageRoute(builder: (context) => Beranda()));
              } else {
                AlertDialog alertDialog = AlertDialog(
                  content: const Text("Username atau Password Tidak Valid"),
                  actions: [
                    ElevatedButton(
                      onPressed: () {
                        Navigator.pop(context);
                      },
                      child: const Text("OK"),
                      style: ElevatedButton.styleFrom(primary: Colors.green),
                    )
                  ],
                );
                showDialog(
                    context: context, builder: (context) => alertDialog);
              }
            });
          }));
}
```

Membuat Service Poli

Buat sebuah file pada folder **service** dengan nama **poli_service.dart** kemudian ketikkan kode berikut

```
1.  import 'package:dio/dio.dart';
2.  import '../helpers/api_client.dart';
3.  import '../model/poli.dart';
4.
5.  class PoliService {
6.    Future<List<Poli>> listData() async {
7.      final Response response = await ApiClient().get('poli');
8.      final List data = response.data as List;
9.      List<Poli> result = data.map((json) => Poli.fromJson(json)).toList();
10.     return result;
11.   }
12.
13.   Future<Poli> simpan(Poli poli) async {
14.     var data = poli.toJson();
15.     final Response response = await ApiClient().post('poli', data);
16.     Poli result = Poli.fromJson(response.data);
17.     return result;
18.   }
```

```
19.
20.   Future<Poli> ubah(Poli poli, String id) async {
21.     var data = poli.toJson();
22.     final Response response = await ApiClient().put('poli/${id}', data);
23.     print(response);
24.     Poli result = Poli.fromJson(response.data);
25.     return result;
26.   }
27.
28.   Future<Poli> getById(String id) async {
29.     final Response response = await ApiClient().get('poli/${id}');
30.     Poli result = Poli.fromJson(response.data);
31.     return result;
32.   }
33.
34.   Future<Poli> hapus(Poli poli) async {
35.     final Response response = await ApiClient().delete('poli/${poli.id}');
36.     Poli result = Poli.fromJson(response.data);
37.     return result;
38.   }
39. }
```

TUGAS

Buat kode untuk Service Pegawai dan Pasien