

LATIHAN WEBVIEW

Android System Webview adalah sistem komponen yang merupakan bagian penting dari aplikasi peramban atau browser Chrome yang memungkinkan aplikasi dapat menampilkan konten website pada perangkat. Komponen ini akan secara otomatis terpasang secara pre-installed atau bawaan dari pabrikan pada setiap perangkat Android, yang nantinya selama masa penggunaan aplikasi ini akan meminta diupdate atau diperbaharui agar memiliki keamanan terbaru serta perbaikan bug lainnya demi kenyamanan pengguna. Sistem ini pertama kali hadir di Android 4.4 (KitKat) yang awalnya menggunakan basis open source dari Google Chrome. Namun, semenjak kehadiran Android generasi kelima atau Android 5.0 (Lollipop), sistem ini dijadikan sebuah modul aplikasi yang terpisah dari Google Chrome sehingga dapat terus diperbarui di Playstore tanpa memerlukan update firmware terbaru.

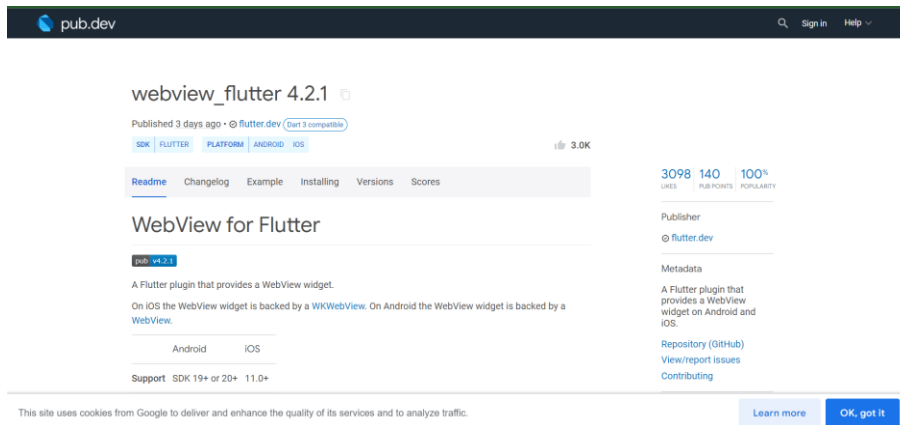
bahwa aplikasi ini berfungsi untuk menampilkan konten website secara langsung dari aplikasi pihak ke-3 yang menggunakan komponen Chrome tanpa harus membuka aplikasi browser terlebih dahulu.

Maksudnya, jika kalian mendapat kiriman tautan atau link lewat aplikasi semisal Facebook, Line, whatsapp dan aplikasi lainnya, maka ketika kalian mengklik link atau tautan tersebut sistem akan memuat konten web nantinya akan tampil pada In-App tanpa kalian harus membuka browser Chrome terlebih dahulu.

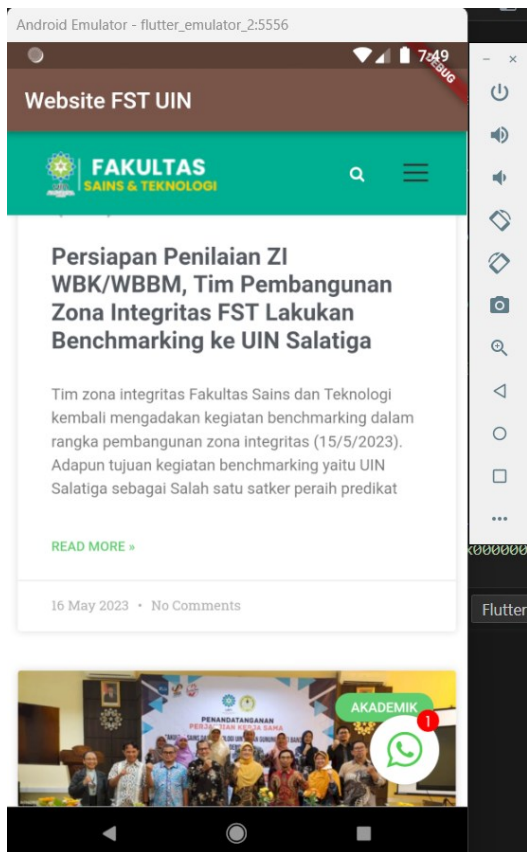
Dengan demikian, maka user experience atau pengalaman pengguna akan jadi lebih baik dan tidak repot harus berganti aplikasi ketika harus membuka tautan website yang dikirim lewat aplikasi lain.

Latihan 1: Membuat Webview

Kali ini kita akan belajar bagaimana membuat webview. Ketika aplikasi mobile di jalankan maka akan muncul suatu web. Pada aplikasi mobile kali ini dibuta untuk menampilkan website portal web FST UIN Bandung. Sehingga ketika aplikasi dijalankan, secara otomatis akan membuka situs dari FST UIN. Pada flutter kalian dapat membaca dokumentasi package webview pada link: https://pub.dev/packages/webview_flutter



Jika ada erorr dan sebagainya sesuaikan dengan versi flutter yang kalian gunakan. Versi flutter yang digunakan pada latihan ini di 3.7 dengan webview_flutter 4.0.1.



Langkah-langkahnya adalah sebagai berikut:

1. Buatlah new project aplikasi flutter dengan penamaan bebas.
2. Tambahkan dependencies di pubspec webview flutter.

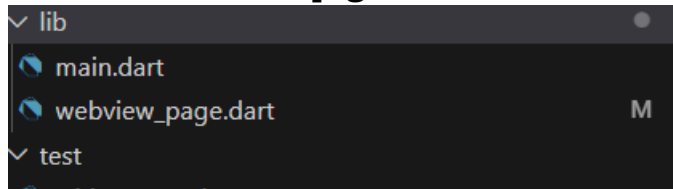
```

# dependencies can be manually updated by changing
# the latest version available on pub.dev. To see w
# versions available, run `flutter pub outdated`.
dependencies:
  cupertino_icons: ^1.0.2
  flutter:
    sdk: flutter
  webview_flutter: ^4.0.1

dev_dependencies:

```

3. Buat new file **webview_page.dart**



4. Edit file **main.dart** menjadi kode program berikut

```

5. import 'package:flutter/material.dart';
6. import 'package:flutter_webview_example/webview_page.dart';
7.
8. void main() {
9.   runApp(const MyApp());
10.}
11.
12. class MyApp extends StatelessWidget {
13.   const MyApp({super.key});
14.
15.   @override
16.   Widget build(BuildContext context) {
17.     return MaterialApp(
18.       title: 'Flutter Demo',
19.       theme: ThemeData(
20.         primarySwatch: Colors.green,
21.       ),
22.       home: const WebViewPage(),
23.     );
24.   }
25.}
26.

```

5. Tuliskan kode program berikut pada **webview_page.dart**. Silahkan untuk bebas mengubah link web yang akan ditampilkan pada aplikasi android.

```
import 'package:flutter/material.dart';
import 'package:webview_flutter/webview_flutter.dart';

class WebViewPage extends StatefulWidget {
  const WebViewPage({Key? key}) : super(key: key);

  @override
  State<WebViewPage> createState() => _WebViewPageState();
}

class _WebViewPageState extends State<WebViewPage> {
  WebViewController? _controller;

  @override
  void initState() {
    super.initState();
    _controller = WebViewController()
      ..setJavaScriptMode(JavascriptMode.unrestricted)
      ..setBackgroundColor(const Color(0x00000000))
      ..setNavigationDelegate(
        NavigationDelegate(
          onProgress: (int progress) {
            // Update loading bar.
          },
          onPageStarted: (String url) {},
          onPageFinished: (String url) {},
          onWebResourceError: (WebResourceError error) {},
          onNavigationRequest: (NavigationRequest request) {
            if (request.url.startsWith('https://www.youtube.com/')) {
              return NavigationDecision.prevent;
            }
            return NavigationDecision.navigate;
          },
        ),
      )
      ..loadRequest(Uri.parse('https://fst.uinsgd.ac.id/'));
  }

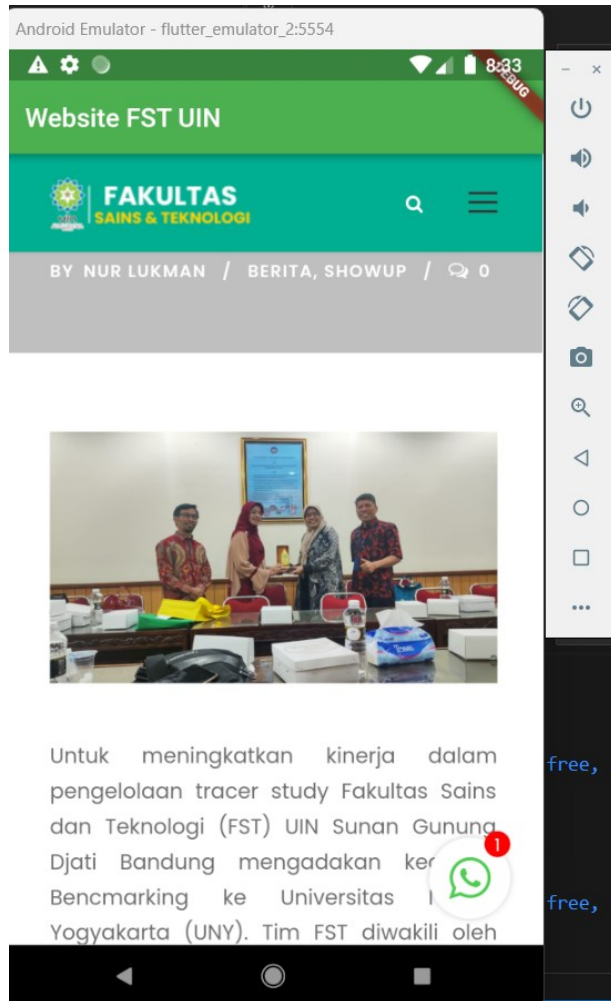
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Website FST UIN"),
      ),
    );
  }
}
```

```

        actions: const [],
      ),
      body: WebViewWidget(controller: _controller!));
    }
  }
}

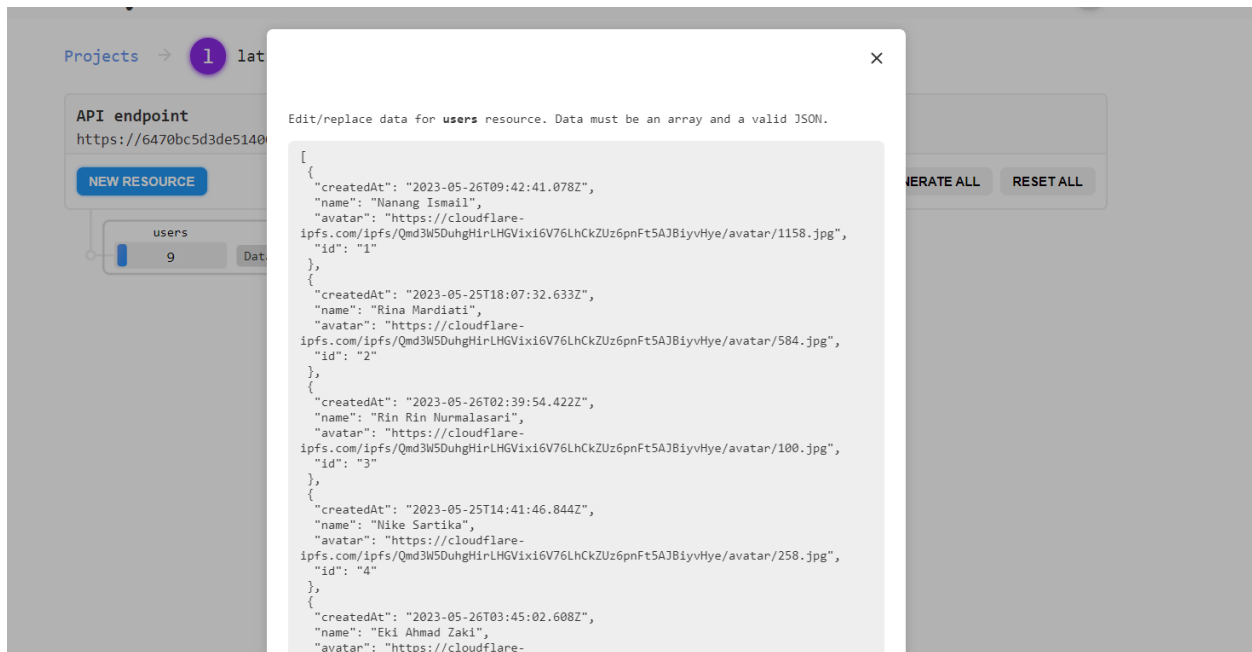
```

Jalankan pada android emulator dan hasil yang akan muncul seperti dibawah ini:



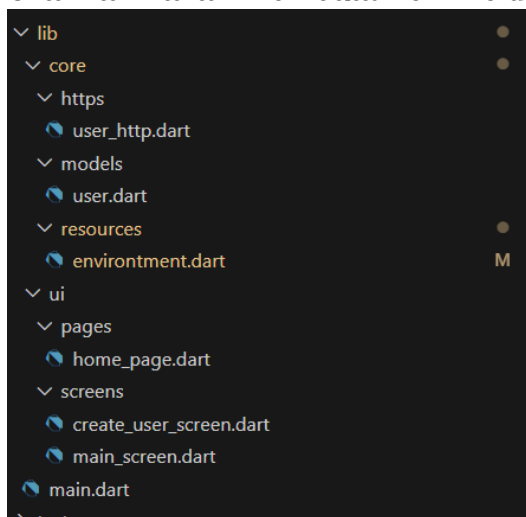
LATIHAN HTTP GET API (CREATE,READ, UPDATE,DELETE)

Pada latihan kali ini kita akan melakukan get data dari API yang telah saya buat di url: <https://6470bc5d3de51400f724d8e5.mockapi.io/api/v1/users> . Data ini akan digunakan pada latihan kali ini.



Langkah-langkah:

1. Buat project baru aplikasi flutter.
2. Silahkan kalian membuat new file dengan struktur folder seperti berikut:



3. Jangan lupa untuk tambahkan dependencies http ya karena pada tutorial kali ini kita akan gunakan http. Dapat di tambahkan bisa **via pubspec**.

```
2  description: A new Flutter project.
3  publish_to: 'none'
4  version: 1.0.0+1
5
6  environment:
7    sdk: ">=2.15.1 <3.0.0"
8
9  dependencies:
10   cupertino_icons: ^1.0.2
11   flutter:
12     sdk: flutter
13   fluttertoast: ^8.0.8
14   http: ^0.13.4
15
16  dev_dependencies:
17   flutter_lints: ^1.0.0
18   flutter_test:
19     sdk: flutter
20
21  flutter:
22    uses-material-design: true
23    # assets:
24
```

4. Edit file **main.dart** menjadi seperti ini :

```
5. import 'package:flutter/material.dart';
6. import 'ui/screens/main_screen.dart';
7.
8. void main() {
9.   runApp(Root());
10.}
11.
12. class Root extends StatelessWidget {
13.   const Root({Key? key}) : super(key: key);
14.
15.   @override
16.   Widget build(BuildContext context) {
17.     return MaterialApp(
18.       debugShowCheckedModeBanner: false,
19.       title: "Latihan HTTP",
20.       theme: ThemeData.light().copyWith(
21.         appBarTheme: AppBarTheme(backgroundColor: Colors.white, elevation: 0),
22.         scaffoldBackgroundColor: Colors.white,
23.         iconTheme: IconThemeData(color: Colors.black),

```

```

24.     ),
25.     home: MainScreen(),
26.   );
27. }
28. }
29.

```

4. Pada folder **models**, edit bagian file **users.dart** yang sudah dibuat. Pada kode program ini memuat model/kelas untuk user yang kita akan panggil data-datanya.

```

1  class User {
2    User({
3      this.createdAt,
4      this.name,
5      this.avatar,
6      this.id,
7    });
8
9    DateTime? createdAt;
10   String? name;
11   String? avatar;
12   String? id;
13
14   factory User.fromJson(Map<String, dynamic> json) => User(
15     createdAt: json["createdAt"] == null ? null : DateTime.parse(json["createdAt"]),
16     name: json["name"] == null ? null : json["name"],
17     avatar: json["avatar"] == null ? null : json["avatar"],
18     id: json["id"] == null ? null : json["id"],
19   );
20
21   Map<String, dynamic> toJson() => {
22     "createdAt": createdAt == null ? null : createdAt!.toIso8601String(),
23     "name": name == null ? null : name,
24     "avatar": avatar == null ? null : avatar,
25     "id": id == null ? null : id,
26   };
27 }
28

```

5. Pada folder **resources** terdapat file **enviromtment.dart**. Tulis kode program berikut pada enviromtment.dart untuk memasukkan url API yang sudah dibuat sebelumnya.

```

lib > core > resources > enviromtment.dart > ...
1  const String endpoint =
2    "https://6470bc5d3de51400f724d8e5.mockapi.io/api/v1/users";
3

```

6. Edit file pada folder http, edit file **user_http.dart** dan tambahkan kode program berikut untuk menjalankan get, post, dll proses crud ada pada file ini.

```

import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:my_http_mockapi/core/models/user.dart';
import 'package:my_http_mockapi/core/resources/enviromtment.dart';

class UserHttp {

```



```

Future<List<User>> listOfUsers() async {
  try {
    var resp = await http.get(Uri.parse(endpoint));
    if (resp.statusCode == 200) {
      List map = jsonDecode(resp.body);
      // print(map);
      List<User> users = map.map((e) => User.fromJson(e)).toList();
      return users;
    } else {
      return [];
    }
  } catch (e) {
    print(e);
  }
  return [];
}

```

```

Future users() async {
  try {
    var resp = await http.get(Uri.parse(endpoint));
    if (resp.statusCode == 200) {
      List map = jsonDecode(resp.body);
      List<User> users = map.map((e) => User.fromJson(e)).toList();
      return users;
    } else {
      return [];
    }
  } catch (e) {
    print(e);
  }
}

```

```

Future addUsers(String name) async {
  try {
    var body = {"name": name};
    var resp = await http.post(Uri.parse(endpoint), body: body);
    if (resp.statusCode <= 201) {
      return true;
    } else {
      return false;
    }
  } catch (e) {
    print(e);
  }
}

```

```

    Future<bool> addUser(String name, {DateTime? date, String avatar = "avatar",
String id = "1"}) async {
    try {
        Map<String, dynamic> userMap = User(name: name, createdAt: DateTime.now(),
avatar: avatar, id: id).toJson();

        var resp = await http.post(Uri.parse(endpoint), body: userMap);
        print("resp ${resp.body}");
        if (resp.statusCode <= 201) {
            return true;
        }
    } catch (e) {
        print(e);
    }
    return false;
}

Future updateUsers(String name) async {
    try {
        var body = {"name": name};
        var resp = await http.put(Uri.parse(endpoint), body: body);
        if (resp.statusCode <= 201) {
            return true;
        } else {
            return false;
        }
    } catch (e) {
        print(e);
    }
}

Future deleteUsers(String id) async {
    try {
        var resp = await http.delete(Uri.parse("${endpoint}/${id}"));
        if (resp.statusCode <= 201) {
            return true;
        } else {
            return false;
        }
    } catch (e) {
        print(e);
    }
}
}

```

7. Selanjutnya pada menu page, terdapat file **home_page.dart**. Tambahkan kode program berikut pada **home_page.dart**

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:my_http_mockapi/core/https/user_http.dart';
import 'package:my_http_mockapi/core/models/user.dart';

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return FutureBuilder<List<User>>(
      future: UserHttp().listOfUsers(),
      builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
          return Center(child: CircularProgressIndicator());
        }
        if (!snapshot.hasData && snapshot.hasError) {
          return Center(child: CircularProgressIndicator());
        }
        return ListView(
          children: snapshot.data!.map((e) => itemCategory(e, context)).toList(),
        );
      },
    );
  }

  Widget itemCategory(User data, BuildContext context) {
    return ListTile(
      title: Text(data.name ?? ""),
      trailing: GestureDetector(onTap: () => _showCupertinoDialog(data, context),
        child: Icon(Icons.more_horiz)),
    );
  }

  void _showCupertinoDialog(User data, BuildContext context) {
    showDialog(
```

```

        context: context,
        builder: (context) {
          return CupertinoAlertDialog(
            title: Text('Delete User?'),
            content: Text('Cannot restored if you select yes'),
            actions: [
              TextButton(
                onPressed: () async {
                  var resp = await UserHttp().deleteUsers(data.id!);
                  if (resp) {
                    Navigator.pop(context);
                    Fluttertoast.showToast(msg: "Success Delete User");
                    setState(() {});
                  }
                },
                child: Text('Yes')),
              TextButton(onPressed: () => Navigator.pop(context), child:
Text('No')),
            ],
          );
        });
      }
    }
  }
}

```

8. Edit file **screens/create_user_screen.dart** dan tambahkan kode berikut. Pada file ini kita akan menambahkan listview.

```

import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:my_http_mockapi/core/https/user_http.dart';

class CreateUserScreen extends StatefulWidget {
  const CreateUserScreen({Key? key}) : super(key: key);

  @override
  _CreateUserScreenState createState() => _CreateUserScreenState();
}

class _CreateUserScreenState extends State<CreateUserScreen> {
  final TextEditingController _etName = TextEditingController();

  @override
  void dispose() {
    _etName.dispose();
  }
}

```

```

    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: IconButton(icon: Icon(Icons.arrow_back, color: Colors.black),
onPressed: () => Navigator.pop(context)),
      ),
      body: ListView(
        padding: EdgeInsets.all(20),
        children: [
          TextField(
            controller: _etName,
          ),
          SizedBox(height: 100),
          SizedBox(
            height: 40,
            width: double.infinity,
            child: ElevatedButton(
              child: Text("Save"),
              onPressed: () async {
                if (_etName.text.trim().length == 0) return;
                // var resp = await UserHttp().addUsers(_etName.text);
                var resp = await UserHttp().addUser(_etName.text);
                if (resp) {
                  Navigator.pop(context);
                  Fluttertoast.showToast(msg: "Success Add New User");
                }
              },
            ),
          ),
        ],
      ),
    );
  }
}

```

9. Pada file **screens/main_screen.dart** tambahkan kode berikut:

```

import 'package:flutter/material.dart';
import 'package:my_http_mockapi/ui/pages/home_page.dart';

import 'create_user_screen.dart';

```

```

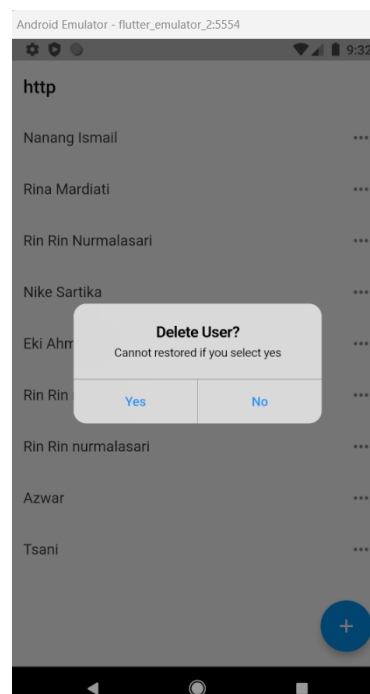
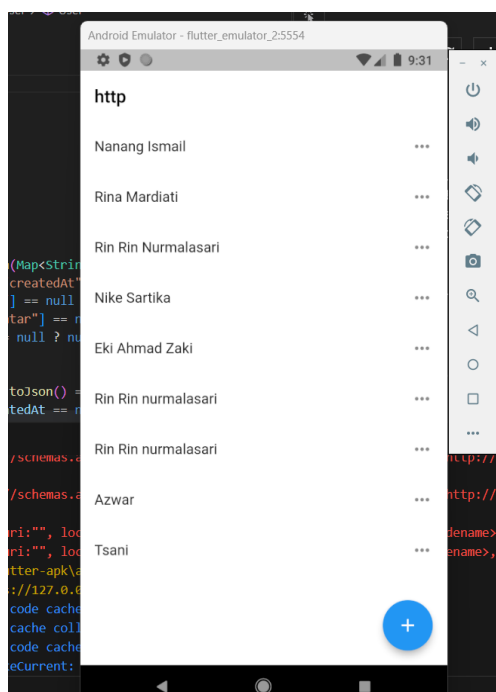
class MainScreen extends StatefulWidget {
  const MainScreen({Key? key}) : super(key: key);

  @override
  State<MainScreen> createState() => _MainScreenState();
}

class _MainScreenState extends State<MainScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("http", style: TextStyle(color: Colors.black))),
      body: HomePage(),
      floatingActionButton: FloatingActionButton(
        child: Icon(Icons.add),
        onPressed: () => Navigator.push(context, MaterialPageRoute(builder:
(context) => CreateUserScreen())).then((value) => setState(() {})),
      ),
    );
  }
}

```

Jika tidak ada erorr, jalankan kode program dan akan tampil seperti di bawah ini. Aplikasi dapat menambah, menghapus user.



Tugas dikumpulkan **hari jumat, 2 Juni 2023 Pukul: 23:59 WIB.**

- Kode program dimasukan ke dalam github
- Record hasil running program dengan nama file video **nim_nama_latihan1/2.mp4**

Jika ada yang ingin ditanyakan atau diskusi dapat dilakukan di grup WA. Terimakasih