

**PEMROGRAMAN MOBILE
PENGANTAR BAHASA PEMROGRAMAN DART -
BAGIAN 14 PRAKTIKUM 2**



OLEH:

Nama : Agung Rizky S

NIM : 2241720187

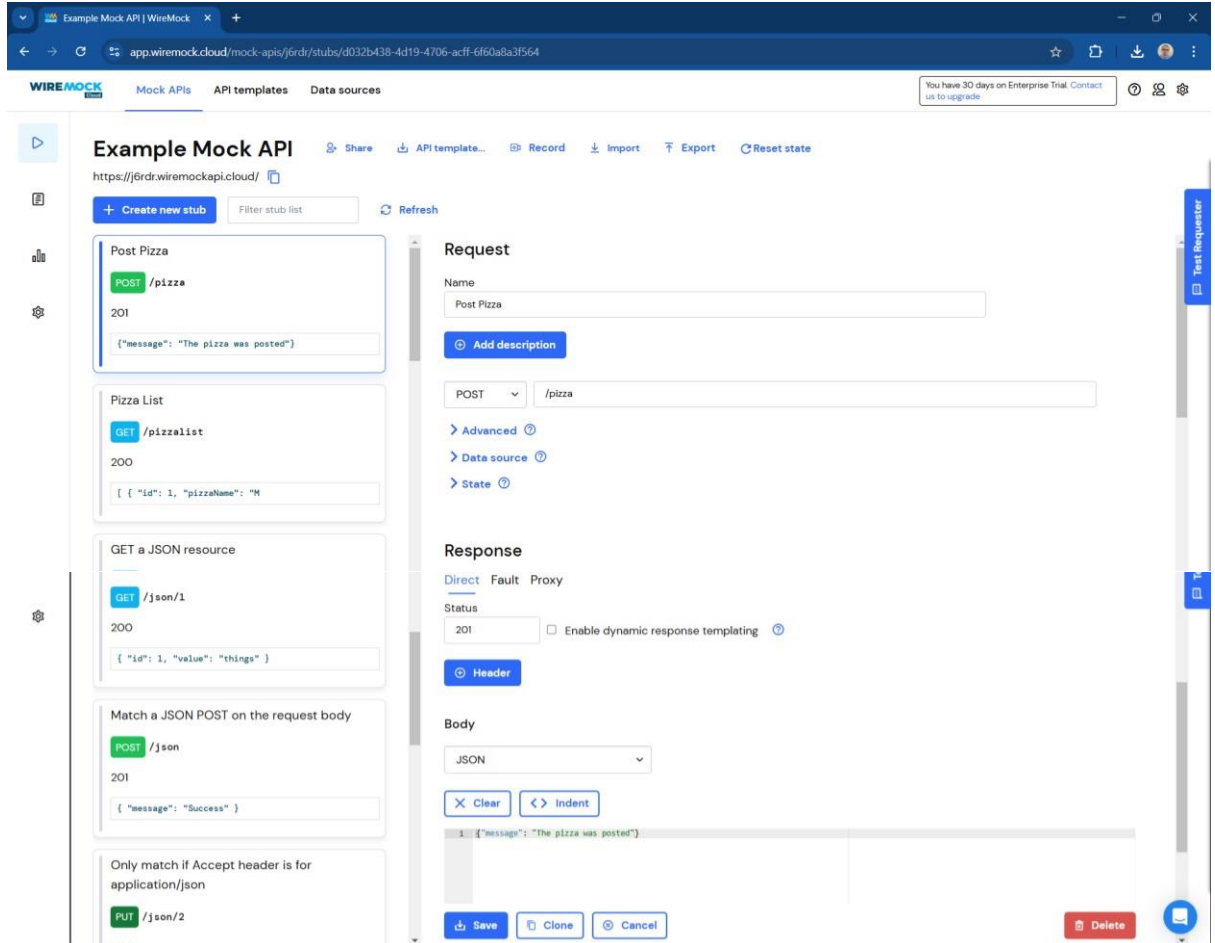
Kelas : TI – 3C

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2024

Praktikum 2: POST-ing data

1. Masuk ke layanan Lab Mock di <https://app.wiremock.cloud/> dan klik bagian Stubs, kemudian, buatlah stub baru.
2. Lengkapi isian seperti gambar berikut:



3. Simpan.
4. Di proyek Flutter, di file `httpHelper.dart`, di kelas `HttpHelper`, buat metode baru bernama `postPizza`, lengkapi kode sebagai berikut.

```
Future<String> postPizza(Pizza pizza) async {
  const postPath = '/pizza';
  String post = json.encode(pizza.toJson());
  Uri url = Uri.https(authority, postPath);
  http.Response r = await http.post(
    url,
    body: post,
  );
  return r.body;
}
```

5. Di dalam proyek, buat sebuah file baru bernama pizza_detail.dart.
6. Di bagian atas file baru, tambahkan impor yang diperlukan.

```
import 'package:flutter/material.dart';
import 'pizza.dart';
import 'httphelper.dart';
```

7. Buat StatefulWidget bernama PizzaDetailScreen.

```
class PizzaDetailScreen extends StatefulWidget {
  const PizzaDetailScreen({super.key});

  @override
  State<PizzaDetailScreen> createState() => _
  PizzaDetailScreenState();
}

class _PizzaDetailScreenState extends State<PizzaDetailScreen> {
  @override
  Widget build(BuildContext context) {
    return Placeholder();
  }
}
```

8. Di bagian atas kelas _PizzaDetailScreenState, tambahkan lima widget TextEditingController. Widget ini akan berisi data untuk objek Pizza yang akan diposting nanti. Juga, tambahkan sebuah String yang akan berisi hasil dari permintaan POST.

```

final TextEditingController txtId = TextEditingController();
final TextEditingController txtName = TextEditingController();
final TextEditingController txtDescription =
    TextEditingController();
final TextEditingController txtPrice = TextEditingController();
final TextEditingController txtImageUrl =
    TextEditingController();
String operationResult = '';

```

9. Override metode dispose() untuk membuang controllers

```

@override
void dispose() {
    txtId.dispose();
    txtName.dispose();
    txtDescription.dispose();
    txtPrice.dispose();
    txtImageUrl.dispose();
    super.dispose();
}

```

10. Dalam metode build() pada kelas, kita return sebuah Scaffold, yang AppBar-nya berisi Teks yang menyatakan “Detail Pizza” dan Body-nya berisi Padding dan SingleChildScrollView yang berisi Column.

```

return Scaffold(
    appBar: AppBar(
        title: const Text('Pizza Detail'),
    ),
    body: Padding(
        padding: const EdgeInsets.all(12),
        child: SingleChildScrollView(
            child: Column(children: []),
        )),
);

```

11. Untuk properti anak dari Column, tambahkan beberapa Text yang akan berisi hasil posting, lima TextFields, masing-masing terikat pada TextEditingController, dan sebuah ElevatedButton untuk menyelesaikan aksi POST (metode postPizza akan dibuat berikutnya). Juga, tambahkan SizedBox untuk memberi jarak pada widget di layar.

```
Text(
  operationResult,
  style: TextStyle(
    backgroundColor: Colors.green[200],
    color: Colors.black),
),
const SizedBox(
  height: 24,
),
TextField(
  controller: txtId,
  decoration: const InputDecoration(hintText: 'Insert ID'),
),
const SizedBox(
  height: 24,
),
TextField(
  controller: txtName,
  decoration: const InputDecoration(hintText: 'Insert Pizza Name'),
),
```

```

const SizedBox(
  height: 24,
),
TextField(
  controller: txtDescription,
  decoration: const InputDecoration(hintText: 'Insert Description'),
),
const SizedBox(
  height: 24,
),
TextField(
  controller: txtPrice,
  decoration: const InputDecoration(hintText: 'Insert Price'),
),
const SizedBox(
  height: 24,
),
TextField(
  controller: txtImageUrl,
  decoration: const InputDecoration(hintText: 'Insert Image Url'),
),
const SizedBox(
  height: 48,
),
ElevatedButton(
  child: const Text('Send Post'),
  onPressed: () {
    postPizza();
  })
],

```

12. Di bagian bawah kelas `_PizzaDetailState`, tambahkan metode `postPizza`.

```

Future postPizza() async {
  HttpHelper helper = HttpHelper();
  Pizza pizza = Pizza();
  pizza.id = int.tryParse(txtId.text);
  pizza.pizzaName = txtName.text;
  pizza.description = txtDescription.text;
  pizza.price = double.tryParse(txtPrice.text);

  pizza.imageUrl = txtImageUrl.text;
  String result = await helper.postPizza(pizza);
  setState(() {
    operationResult = result;
  });
}

```

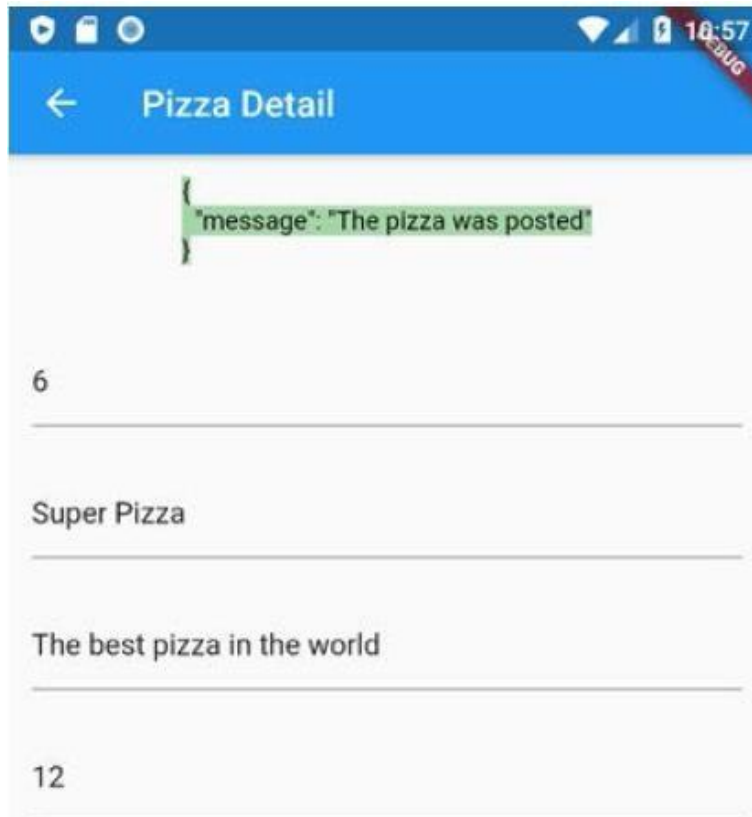
13. Di file main.dart, impor file pizza_detail.dart.
14. Di perancah metode build() dari kelas _MyHomePageState, tambahkan FloatingActionButton yang akan menavigasi ke rute PizzaDetail.

```

floatingActionButton: FloatingActionButton(
  child: const Icon(Icons.add),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => const PizzaDetailScreen()),
    );
  }),

```

15. Jalankan aplikasi. Pada layar utama, tekan FloatingActionButton untuk menavigasi ke rute PizzaDetail
16. Tambahkan detail pizza di kolom teks dan tekan tombol Kirim Postingan. Anda akan melihat hasil yang berhasil, seperti yang ditunjukkan pada gambar berikut.



Jawaban :

Code :

HttpHelper.dart :

```
import 'dart:convert';
import 'dart:io';
import 'package:http/http.dart' as http;
import 'pizza.dart';

class HttpHelper {
  final String authority = '6r4jr.wiremockapi.cloud';
  final String path = 'pizzalist';

  Future<List<Pizza>> getPizzaList() async {
    final Uri url = Uri.https(authority, path);
    try {
      final http.Response result = await http.get(url);
      if (result.statusCode == HttpStatus.ok) {
        final List<dynamic> jsonResponse = json.decode(result.body);
        return jsonResponse.map<Pizza>((item) => Pizza.fromJson(item)).toList();
      } else {
```



```

        return [];
    }
} catch (e) {
    print('Exception: $e');
    return [];
}
}

Future<String> postPizza(Pizza pizza) async {
    const postPath = '/pizza';
    String post = json.encode(pizza.toJson());
    Uri url = Uri.https(authority, postPath);
    try {
        http.Response response = await http.post(
            url,
            headers: {'Content-Type': 'application/json'},
            body: post,
        );
        if (response.statusCode == HttpStatus.ok || response.statusCode == 201) {
            return 'Pizza posted successfully!';
        } else {
            return 'Failed to post pizza: ${response.statusCode}';
        }
    } catch (e) {
        return 'Error posting pizza: $e';
    }
}
}

```

Main.dart :

```

import 'package:flutter/material.dart';
import 'httphelper.dart';
import 'pizza.dart';
import 'pizza_detail.dart';

void main() {
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({super.key});

    @override
    Widget build(BuildContext context) {

```

```

    return MaterialApp(
      title: 'Flutter Pizza API Demo',
      theme: ThemeData(
        primarySwatch: Colors.deepPurple,
      ),
      home: const MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key});

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  Future<List<Pizza>> callPizzas() async {
    HttpHelper helper = HttpHelper();
    return await helper.getPizzaList();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Pizza List')),
      body: FutureBuilder(
        future: callPizzas(),
        builder: (BuildContext context, AsyncSnapshot<List<Pizza>> snapshot) {
          if (snapshot.hasError) {
            return const Center(child: Text('Something went wrong'));
          }
          if (!snapshot.hasData) {
            return const Center(child: CircularProgressIndicator());
          }
          return ListView.builder(
            itemCount: snapshot.data!.length,
            itemBuilder: (BuildContext context, int position) {
              return ListTile(
                title: Text(snapshot.data![position].pizzaName),
                subtitle: Text(
                  '${snapshot.data![position].description} -
\\$${snapshot.data![position].price}',
                ),
              );
            },
          );
        },
      ),
    );
  }
}

```

```

        );
      },
    );
  },
),
floatingActionButton: FloatingActionButton(
  child: const Icon(Icons.add),
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => const PizzaDetailScreen(),
      ),
    );
  },
),
);
}
}

```

Pizza_detaill.dart :

```

import 'package:flutter/material.dart';
import 'pizza.dart';
import 'httphelper.dart';

class PizzaDetailScreen extends StatefulWidget {
  const PizzaDetailScreen({super.key});

  @override
  State<PizzaDetailScreen> createState() => _PizzaDetailScreenState();
}

class _PizzaDetailScreenState extends State<PizzaDetailScreen> {
  final TextEditingController txtId = TextEditingController();
  final TextEditingController txtName = TextEditingController();
  final TextEditingController txtDescription = TextEditingController();
  final TextEditingController txtPrice = TextEditingController();
  final TextEditingController txtImageUrl = TextEditingController();
  String operationResult = '';

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

```

```

    txtName.dispose();
    txtDescription.dispose();
    txtPrice.dispose();
    txtImageUrl.dispose();
    super.dispose();
}

Future postPizza() async {
  HttpHelper helper = HttpHelper();
  Pizza pizza = Pizza(
    id: int.tryParse(txtId.text) ?? 0,
    pizzaName: txtName.text,
    description: txtDescription.text,
    price: double.tryParse(txtPrice.text) ?? 0.0,
    imageUrl: txtImageUrl.text,
  );
  String result = await helper.postPizza(pizza);
  setState(() {
    operationResult = result;
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Pizza Detail'),
    ),
    body: Padding(
      padding: const EdgeInsets.all(12),
      child: SingleChildScrollView(
        child: Column(
          children: [
            Text(
              operationResult,
              style: TextStyle(
                backgroundColor: Colors.green[200],
                color: Colors.black,
              ),
            ),
            const SizedBox(height: 24),
            TextField(
              controller: txtId,
              decoration: const InputDecoration(hintText: "Insert ID"),
            ),
          ],
        ),
      ),
    ),
  );
}

```

```

        const SizedBox(height: 24),
        TextField(
          controller: txtName,
          decoration: const InputDecoration(hintText: 'Insert Pizza Name'),
        ),
        const SizedBox(height: 24),
        TextField(
          controller: txtDescription,
          decoration: const InputDecoration(hintText: 'Insert
Description'),
        ),
        const SizedBox(height: 24),
        TextField(
          controller: txtPrice,
          decoration: const InputDecoration(hintText: 'Insert Price'),
        ),
        const SizedBox(height: 24),

        ElevatedButton(
          child: const Text('Send Post'),
          onPressed: postPizza,
        ),
      ],
    ),
  ),
),
);
}
}

```

Input :

