

PRAKTIKUM #10

Latihan Web Service (API), MySQL, PHP

1. Buat database di MySQL dengan nama : coba

2. Buat Tabel note_app :

Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	Auto Incr?	Z
id	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
title	varchar	50		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
content	text			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
* date	date			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Buttons: Create Table..., Advanced Properties..., Preview..., Insert, Delete, Cancel. ☒ Hide Language Options

3. Isikan beberapa data:

	id	title	content		date
<input type="checkbox"/>	1	Belajar CRUD	Belajar CRUD MySQL Flutter	26B	2023-11-27
<input checked="" type="checkbox"/>	2	Belajar API	Belajar lagi	12B	2023-11-27
<input type="checkbox"/>	3	Isi data update	uji coba isi data postman kemudian d...	44B	2023-11-27
<input type="checkbox"/>	4	API & JSON	Belajar API, database MySQL dan PHP	35B	2023-11-27
<input type="checkbox"/>	5	Coba Hapus	Data ini akan di Hapus	22B	2023-11-27

4. Buat folder di xampp\htdocs\latihan\note_app

5. Buat file PHP berikut di folder **note_app** :

List.php

```
<?php
    $connection = new mysqli("localhost","root","","coba");
    $data       = mysqli_query($connection, "select * from note_app");
    $data       = mysqli_fetch_all($data, MYSQLI_ASSOC);
    echo json_encode($data);
?>
```

create.php

```
<?php
    $connection = new mysqli("localhost", "root", "", "coba");
    $title      = $_POST['title'];
    $content    = $_POST['content'];
    $date       = date('Y-m-d');
    $result     = mysqli_query($connection, "insert into note_app set
title='$title', content='$content', date='$date'");
    if($result){
```

```

        echo json_encode([
            'message' => 'Data input successfully'
        ]);
    }else{
        echo json_encode([
            'message' => 'Data Failed to input'
        ]);
    }
?>

```

update.php

```

<?php
    $connection = new mysqli("localhost", "root", "", "coba");
    $title      = $_POST['title'];
    $content    = $_POST['content'];
    $id         = $_POST['id'];
    $result = mysqli_query($connection, "update note_app set
title='$title', content='$content' where id='$id'");
    if($result){
        echo json_encode([
            'message' => 'Data edit successfully'
        ]);
    }else{
        echo json_encode([
            'message' => 'Data Failed to update'
        ]);
    }
?>

```

delete.php

```

<?php
    $connection = new mysqli("localhost", "root", "", "coba");
    $id = $_POST['id'];
    $result = mysqli_query($connection, "delete from note_app where
id=".$id);
    if($result){
        echo json_encode([
            'message' => 'Data delete successfully'
        ]);
    }else{
        echo json_encode([
            'message' => 'Data Failed to delete'
        ]);
    }
?>

```

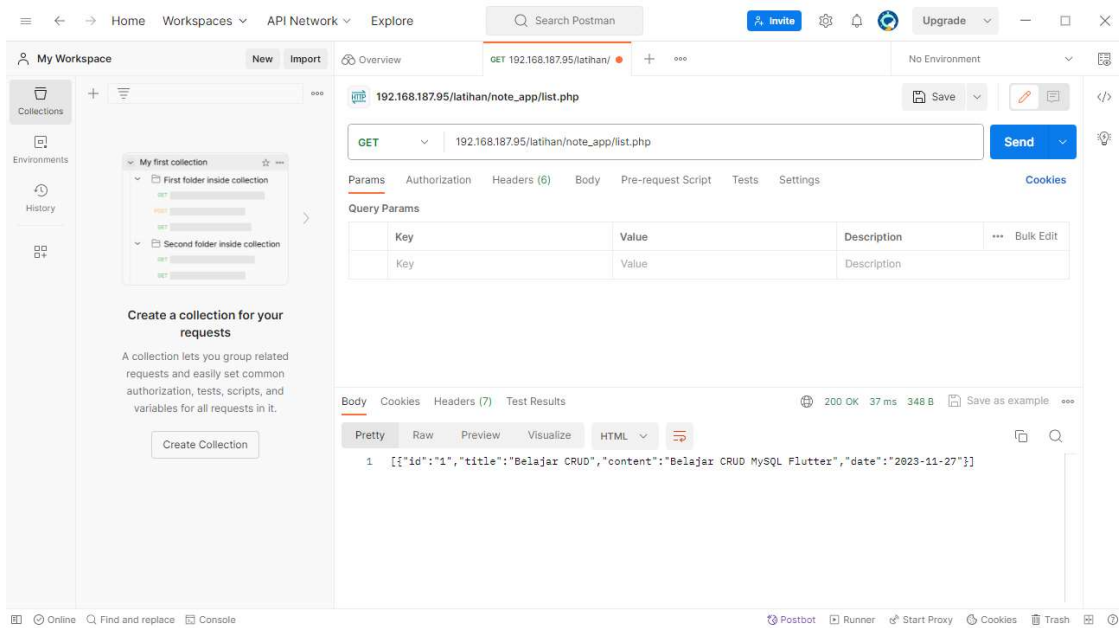
detail.php

```

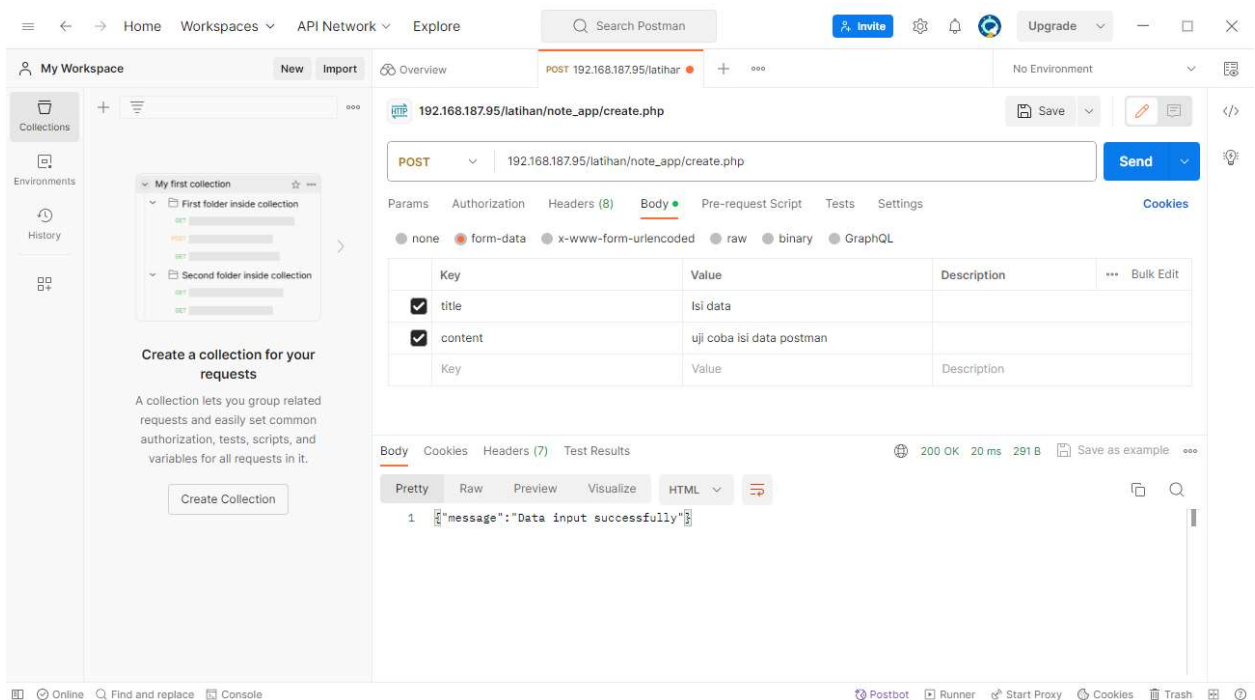
<?php
    $connection = new mysqli("localhost","root","","coba");
    $data      = mysqli_query($connection, "select * from note_app
where id=".$_GET['id']);
    $data      = mysqli_fetch_array($data, MYSQLI_ASSOC);
    echo json_encode($data);
?>

```

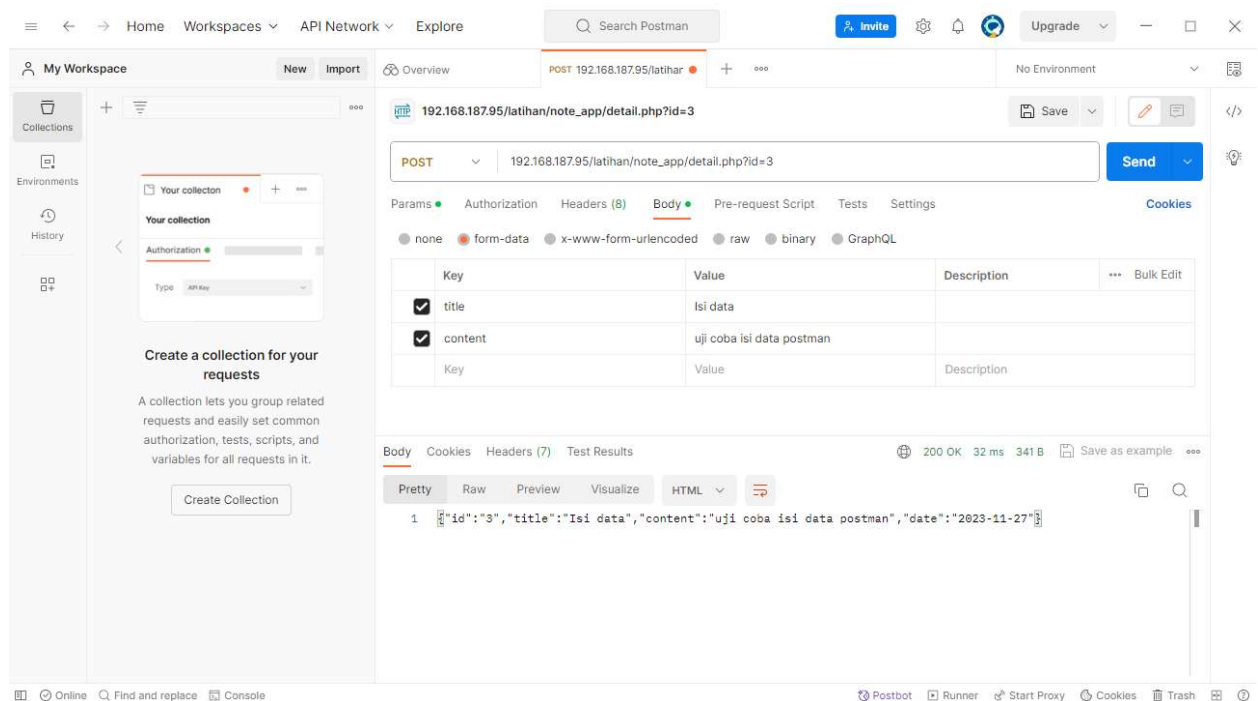
6. Uji di **Postman** : **List.php**



Create.php



Detail.php



7. Buat Project baru : note_app_mysql
8. Tambahkan dependency di pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  flutter_staggered_grid_view: ^0.6.1  
  http: ^0.13.4
```

Kemudian simpan atau jalankan perintah **flutter pub get** untuk mendapatkan package

9. Buka **main.dart**, edit script code nya :

```
// ignore_for_file: use_key_in_widget_constructors, prefer_const_constructors  
  
import 'package:flutter/material.dart';  
import 'home.dart';  
  
void main() => runApp(App());  
  
class App extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      // themeMode: ThemeMode.dark,  
      theme: ThemeData(  
        primaryColor: Colors.black,  
        scaffoldBackgroundColor: Colors.blueGrey.shade900,  

```

```

        appBarTheme:
            AppBarTheme(backgroundColor: Colors.transparent, elevation: 0)),
        title: 'Flutter + PHP CRUD',
        initialRoute: '/',
        routes: {
            '/': (context) => Home(),
        },
    );
}
}

```

10. Buat file **home.dart** di folder lib.

```

import 'package:flutter/material.dart';
import 'package:flutter_staggered_grid_view/flutter_staggered_grid_view.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
import 'add.dart';
import 'edit.dart';

class Home extends StatefulWidget {
  @override
  HomeState createState() => HomeState();
}

class HomeState extends State<Home> {
  //make list variable to accomodate all data from database
  List _get = [];

  //make different color to different card
  final _lightColors = [
    Colors.amber.shade300,
    Colors.lightGreen.shade300,
    Colors.lightBlue.shade300,
    Colors.orange.shade300,
    Colors.pinkAccent.shade100,
    Colors.tealAccent.shade100
  ];

  @override
  void initState() {
    super.initState();
    //in first time, this method will be executed
    _getData();
  }
}

```

```

Future _getData() async {
  try {
    final response = await http.get(Uri.parse(
      //you have to take the ip address of your computer.
      //because using localhost will cause an error
      "http://192.168.187.95/latihan/note_app/list.php"));

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      // entry data to variabel list _get
      setState(() {
        _get = data;
      });
    }
  } catch (e) {
    print(e);
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Note List'),
    ),
    //if not equal to 0 show data
    //else show text "no data available"
    body: _get.length != 0
      //we use masonry grid to make masonry card style
      ? MasonryGridView.count(
        crossAxisCount: 2,
        itemCount: _get.length,
        itemBuilder: (context, index) {
          return GestureDetector(
            onTap: () {
              Navigator.push(
                context,
                //routing into edit page
                //we pass the id note
                MaterialPageRoute(
                  builder: (context) => Edit(
                    id: _get[index]['id'],

```

```

        ));
    },
    child: Card(
      //make random color to eveery card
      color: _lightColors[index % _lightColors.length],
      child: Container(
        //make 2 different height
        constraints:
          BoxConstraints(minHeight: (index % 2 + 1) * 85),
        padding: EdgeInsets.all(15),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              '${_get[index]['date']}',
              style: TextStyle(color: Colors.black),
            ),
            SizedBox(height: 10),
            Text(
              '${_get[index]['title']}',
              style: TextStyle(
                color: Colors.black,
                fontSize: 20,
                fontWeight: FontWeight.bold,
              ),
            ),
          ],
        ),
      ),
    ),
  ],
),
);
},
)
: Center(
  child: Text(
    "No Data Available",
    style: TextStyle(
      color: Colors.white,
      fontSize: 20,
      fontWeight: FontWeight.bold,
    ),
  ),
),
),
),
floatingActionButton: FloatingActionButton(
  backgroundColor: Colors.black,

```

```

        child: Icon(Icons.add),
        onPressed: () {
            Navigator.push(
                context,
                //routing into add page
                MaterialPageRoute(builder: (context) => Add()));
        },
    ),
);
}
}

```

11. Buat file **add.dart** di folder **lib**.

```

// ignore_for_file: unused_field, prefer_const_constructors, avoid_print

import 'dart:convert';

import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

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

    @override
    State<Add> createState() => _AddState();
}

class _AddState extends State<Add> {
    final _formKey = GlobalKey<FormState>();

    //inisialize field
    var title = TextEditingController();
    var content = TextEditingController();

    Future _onSubmit() async {
        try {
            return await http.post(
                Uri.parse("http://192.168.187.95/latihan/note_app/create.php"),
                body: {
                    "title": title.text,
                    "content": content.text,
                },
            ).then((value) {
                //print message after insert to database
            });
        } catch (e) {
            //print error message
        }
    }
}

```



```

        //you can improve this message with alert dialog
        var data = jsonDecode(value.body);
        print(data["message"]);

        Navigator.of(context)
            .pushNamedAndRemoveUntil('/', (Route<dynamic> route) => false);
    });
} catch (e) {
    print(e);
}
}

```

```

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("Create New Note"),
        ),
        body: Form(
            key: _formKey,
            child: Container(
                padding: EdgeInsets.all(20.0),
                child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                        Text(
                            'Title',
                            style: TextStyle(
                                color: Colors.white,
                                fontSize: 16,
                                fontWeight: FontWeight.bold,
                            ),
                        ),
                        SizedBox(height: 5),
                        TextFormField(
                            controller: title,
                            decoration: InputDecoration(
                                hintText: "Type Note Title",
                                border: OutlineInputBorder(
                                    borderRadius: BorderRadius.circular(15.0),
                                ),
                                fillColor: Colors.white,
                                filled: true),
                            style: const TextStyle(
                                fontWeight: FontWeight.bold,

```

```

        fontSize: 16,
      ),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Note Title is Required!';
        }
        return null;
      },
    ),
    SizedBox(height: 20),
    Text(
      'Content',
      style: TextStyle(
        color: Colors.white,
        fontSize: 16,
        fontWeight: FontWeight.bold,
      ),
    ),
  ),
  SizedBox(height: 5),
  TextFormField(
    controller: content,
    keyboardType: TextInputType.multiline,
    minLines: 5,
    maxLines: null,
    decoration: InputDecoration(
      hintText: 'Type Note Content',
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(15.0),
      ),
      fillColor: Colors.white,
      filled: true),
    style: const TextStyle(
      fontWeight: FontWeight.bold,
      fontSize: 16,
    ),
    validator: (value) {
      if (value!.isEmpty) {
        return 'Note Content is Required!';
      }
      return null;
    },
  ),
),
SizedBox(height: 15),
ElevatedButton(
  style: ElevatedButton.styleFrom(

```

```

        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(15),
        ),
      ),
      child: Text(
        "Submit",
        style: TextStyle(color: Colors.white),
      ),
      onPressed: () {
        //validate
        if (_formKey.currentState!.validate()) {
          //send data to database with this method
          _onSubmit();
        }
      },
    ),
  ],
),
),
),
);
}
}

```

12. Buat file **edit.dart** di **folder lib**.

```

import 'dart:convert';

import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

class Edit extends StatefulWidget {
  Edit({required this.id});

  String id;

  @override
  State<Edit> createState() => _EditState();
}

class _EditState extends State<Edit> {
  final _formKey = GlobalKey<FormState>();

  //inisialize field
  var title = TextEditingController();
  var content = TextEditingController();

```

```

@override
void initState() {
  super.initState();
  //in first time, this method will be executed
  _getData();
}

//Http to get detail data
Future _getData() async {
  try {
    final response = await http.get(Uri.parse(
      //you have to take the ip address of your computer.
      //because using localhost will cause an error
      //get detail data with id
      "http://192.168.187.95/latihan/note_app/detail.php?id='${widget.id}'"))
    ;

    // if response successful
    if (response.statusCode == 200) {
      final data = jsonDecode(response.body);

      setState(() {
        title = TextEditingController(text: data['title']);
        content = TextEditingController(text: data['content']);
      });
    }
  } catch (e) {
    print(e);
  }
}

Future _onUpdate(context) async {
  try {
    return await http.post(
      Uri.parse("http://192.168.187.95/latihan/note_app/update.php"),
      body: {
        "id": widget.id,
        "title": title.text,
        "content": content.text,
      },
    ).then((value) {
      //print message after insert to database
      //you can improve this message with alert dialog
      var data = jsonDecode(value.body);
    });
  }
}

```

```

        print(data["message"]);

        Navigator.of(context)
            .pushNamedAndRemoveUntil('/', (Route<dynamic> route) => false);
    });
} catch (e) {
    print(e);
}
}

Future _onDelete(context) async {
    try {
        return await http.post(
            Uri.parse("http://192.168.1.4/latihan/note_app/delete.php"),
            body: {
                "id": widget.id,
            },
        ).then((value) {
            //print message after insert to database
            //you can improve this message with alert dialog
            var data = jsonDecode(value.body);
            print(data["message"]);

            // Remove all existing routes until the home.dart, then rebuild Home.
            Navigator.of(context)
                .pushNamedAndRemoveUntil('/', (Route<dynamic> route) => false);
        });
    } catch (e) {
        print(e);
    }
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("Creat New Note"),
            // ignore: prefer_const_literals_to_create_immutables
            actions: [
                Container(
                    padding: EdgeInsets.only(right: 20),
                    child: IconButton(
                        onPressed: () {
                            showDialog(
                                context: context,

```

```
builder: (BuildContext context) {  
    //show dialog to confirm delete data  
    return AlertDialog(  
        content: Text('Are you sure you want to delete this?'),  
        actions: <Widget>[  
            ElevatedButton(  
                child: Icon(Icons.cancel),  
                onPressed: () => Navigator.of(context).pop(),  
            ),  
            ElevatedButton(  
                child: Icon(Icons.check_circle),  
                onPressed: () => _onDelete(context),  
            ),  
        ],  
    );  
},  
);  
},  
icon: Icon(Icons.delete)),  
)  
],  
),  
body: Form(  
    key: _formKey,  
    child: Container(  
        padding: EdgeInsets.all(20.0),  
        child: Column(  
            crossAxisAlignment: CrossAxisAlignment.start,  
            children: [  
                Text(  
                    'Title',  
                    style: TextStyle(  
                        color: Colors.white,  
                        fontSize: 16,  
                        fontWeight: FontWeight.bold,  
                    ),  
                ),  
                SizedBox(height: 5),  
                TextFormField(  
                    controller: title,  
                    decoration: InputDecoration(  
                        hintText: "Type Note Title",  
                        border: OutlineInputBorder(  
                            borderRadius: BorderRadius.circular(15.0),  
                        ),  
                    ),
```

```

        fillColor: Colors.white,
        filled: true),
    style: const TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 16,
    ),
    validator: (value) {
        if (value!.isEmpty) {
            return 'Note Title is Required!';
        }
        return null;
    },
),
 SizedBox(height: 20),
Text(
    'Content',
    style: TextStyle(
        color: Colors.white,
        fontSize: 16,
        fontWeight: FontWeight.bold,
    ),
),
SizedBox(height: 5),
TextFormField(
    controller: content,
    keyboardType: TextInputType.multiline,
    minLines: 5,
    maxLines: null,
    decoration: InputDecoration(
        hintText: 'Type Note Content',
        border: OutlineInputBorder(
            borderRadius: BorderRadius.circular(15.0),
        ),
        fillColor: Colors.white,
        filled: true),
    style: const TextStyle(
        fontWeight: FontWeight.bold,
        fontSize: 16,
    ),
    validator: (value) {
        if (value!.isEmpty) {
            return 'Note Content is Required!';
        }
        return null;
    },
),

```

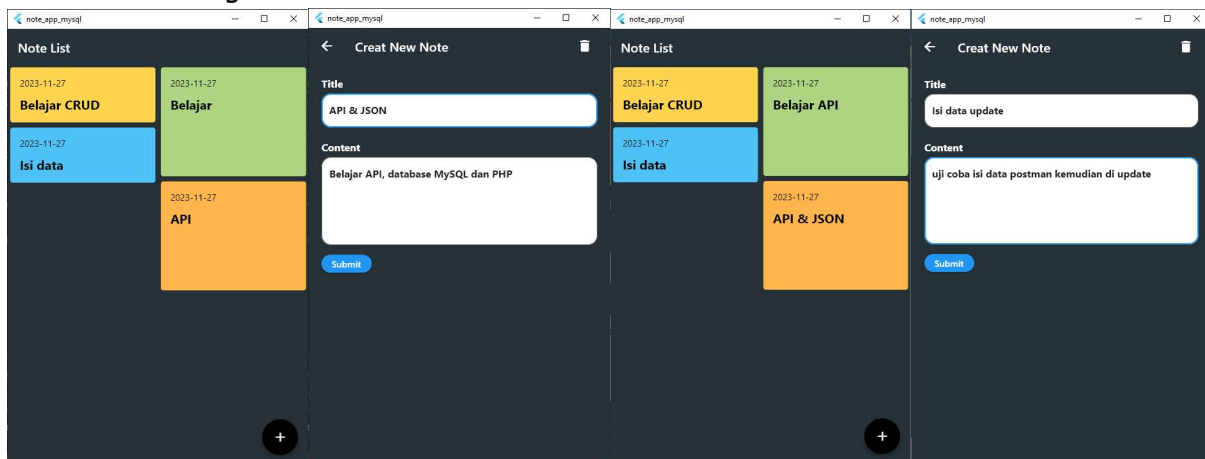
```

    ),
    SizedBox(height: 15),
    ElevatedButton(
      style: ElevatedButton.styleFrom(
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(15),
        ),
      ),
      child: Text(
        "Submit",
        style: TextStyle(color: Colors.white),
      ),
      onPressed: () {
        //validate
        if (_formKey.currentState!.validate()) {
          //send data to database with this method
          _onUpdate(context);
        }
      },
    ),
  ],
),
),
),
),
);
}
}

```

13. Jangan lupa ubah IP address menjadi IP address komputer masing-masing.

14. Jalankan dengan device chrome.



Sumber :

<https://www.sahretech.com/2022/03/crud-flutter-php-mysql-pertama.html>

<https://www.sahretech.com/2022/03/crud-flutter-php-mysql-kedua.html>

Tugas :

- Buat program dari tabel barang

KdBrg	NmBrg	HrgBeli	HrgJual	Stok
B0001	Buku	5000	6000	100
B0002	Spidol	3500	4000	50

- Menu
 - Tambah Data
 - Edit Data
 - Hapus Data
 - Tampil Data