

Bismillah

Flutter REST API

pembuatan sampai implementasi dari dasar

kali ini saya akan membahas flutter REST API , mulai dari penyiapan bahan, server dan pembuatan API hingga implementasi

Penyiapan Server

untuk penyiapan server bisa macam macam caranya gk hanya satu , bisa pake xampp,lampp , linux server local, php server local, tau pun langsung ke hosting, ataupun nodejs

untuk xampp bisa kunjungi halaman berikut : <https://www.apachefriends.org/index.html>

untuk linux server , yang uda biasa ama linux uda jadi bawaan biasanya contoh distro ubuntu yang biasanya saya gunakan

untuk mac os bisa pake xampp ataupun server local langsung tinggal aktifkan server local dan install mysql + php

lebih lanjut tentang penyiapan server bisa langsung hubugi google , karena kalo dibhas bakalan puuuuuuaannjuuaangggggg ...

kuntuk yang terbiasa menggunakan javascript bisa menggunakan nodejs : <https://nodejs.org/en/> baik windows , linux ataupun mac bisa semua , dan untuk node js sendiri sangat simple tinggal install express js dan mysql karena tidak menggunakan php

untuk tutorial kali ini saya akan menggunakan php dan frameworknya menggunakan slim php 3 : <http://www.slimframework.com/docs/v3/>

karena saya rasa lebih umum dan implementasi di share hosting lebih mudah , walau sebenarnya lebih mudah menggunakan node js dan express js , namun akan terkendala dengan server share hosting yang kebanyakan tiadk mendukung node js dan untuk vps harganya lumayan bisa bikin manyun yang baru belajar

lainnya bisa menggunakan php langsung , laravel , lumen dan framework lainnya

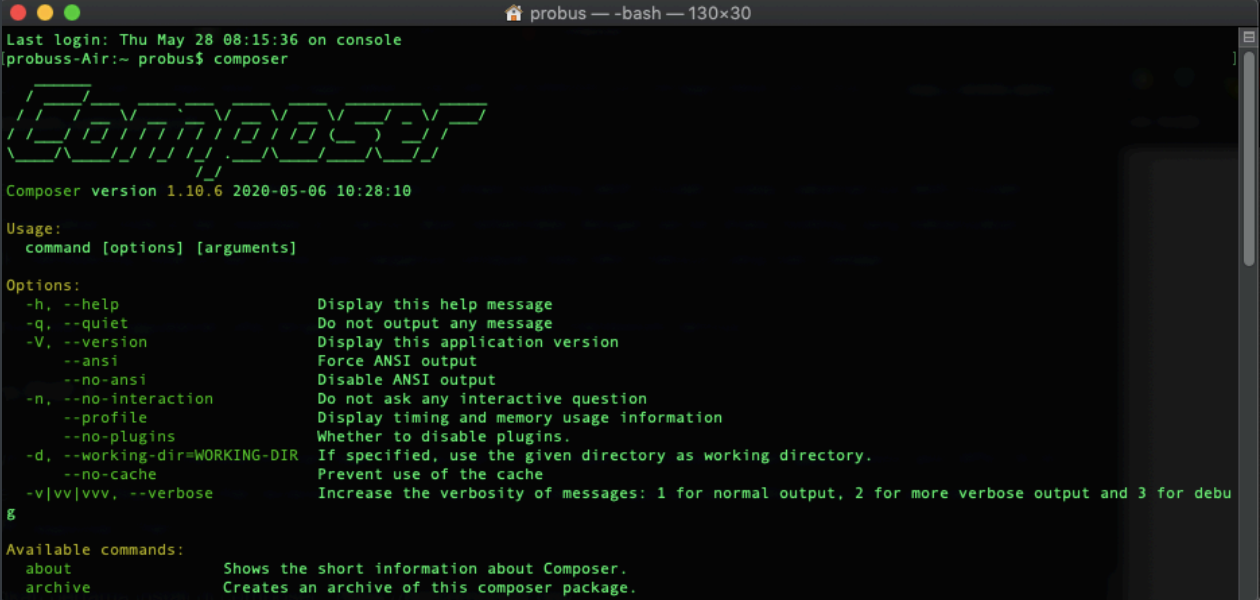
Slim PHP 3

" Slim is a PHP micro framework that helps you quickly write simple yet powerful web applications and APIs. At its core, Slim is a dispatcher that receives an HTTP request, invokes an appropriate callback routine, and returns an HTTP response. That's it."

langkah pertama install di composer : <https://getcomposer.org/>

download dan install sesuai kebutuhan os masing masing , jika gagal jangan sungkan tanya mbh google

pastikan instalasi berhasil dan bisa menjalankan perintah composer di cmd ataupun command line (linux / mac)



```
probuss-Air:~ probus$ composer

Composer version 1.10.6 2020-05-06 10:28:10

Usage:
  command [options] [arguments]

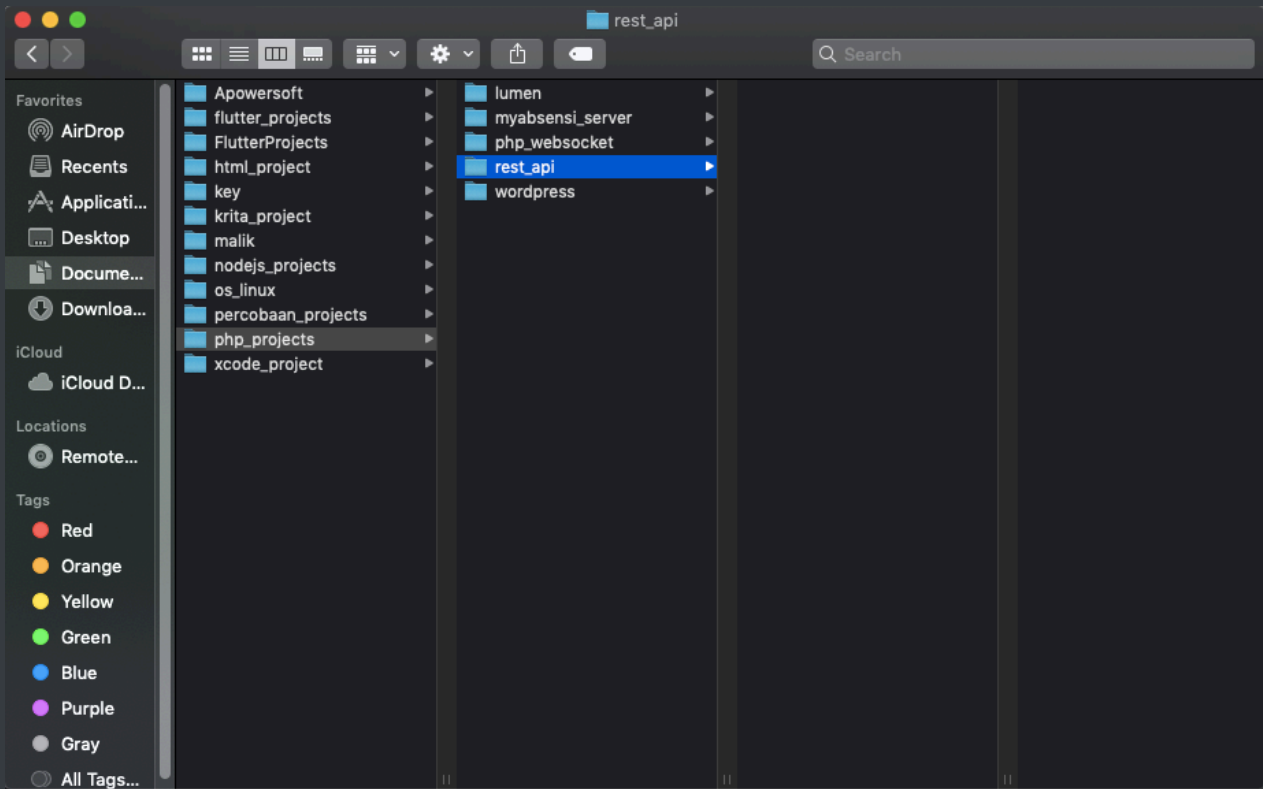
Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache               Prevent use of the cache
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about      Shows the short information about Composer.
  archive    Creates an archive of this composer package.
```

kebetulan saya menggunakan mac jadi saya akan menggunakan command line , di windows mungkin pake cmd , atau ada cara lain bisa mininstall bash linux , terserah aja pake yang mana intnya bisa jalankan composernya

lanjut pergi ke direktori yang ingin di tuju lalu buat folder bernama rest_api > pada kasus saya file akan saya letakkan di > documents/php_projects/rest_api

untuk xampp taro aja langsung di dalam xampp yang bisa dijalankan langsung di xamppnya



jika menggunakan command line akan seperti ini

```
rest_api — -bash — 130x30

search          Searches for packages.
self-update     Updates composer.phar to the latest version.
selfupdate      Updates composer.phar to the latest version.
show            Shows information about packages.
status          Shows a list of locally modified packages, for packages installed from source.
suggests        Shows package suggestions.
u               Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock
file.           Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock
update          Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock
file.           Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock
upgrade         Upgrades your dependencies to the latest version according to composer.json, and updates the composer.lock
file.           Validates a composer.json and composer.lock.
validate        Shows which packages cause the given package to be installed.
why             Shows which packages prevent the given package from being installed.
why-not

[probuss-Air:~ probus$ cd documents
[probuss-Air:documents probus$ ls
Apowersoft      html_project    malik           percobaan_projects
FlutterProjects key             nodejs_projects php_projects
flutter_projects krita_project  os_linux        xcode_project
[probuss-Air:documents probus$ cd php_project
-bash: cd: php_project: No such file or directory
[probuss-Air:documents probus$ cd php_projects
[probuss-Air:php_projects probus$ ls
lumen           myabsensi_server  php_websocket    wordpress
[probuss-Air:php_projects probus$ mkdir -p rest_api
[probuss-Air:php_projects probus$ ls
lumen           myabsensi_server  php_websocket    rest_api          wordpress
[probuss-Air:php_projects probus$ cd rest_api
[probuss-Air:rest_api probus$
```

lalu ketik perintah berikut `composer require slim/slim:3.*`

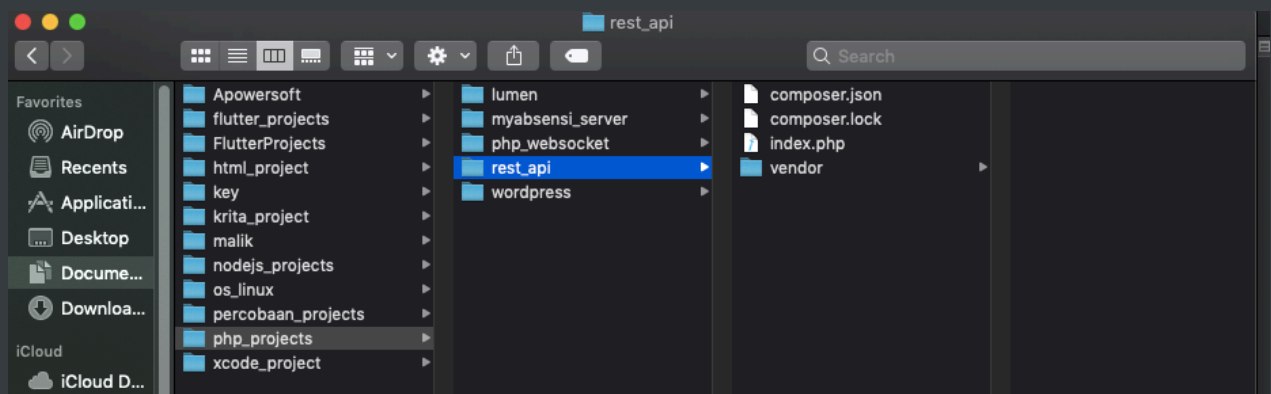
```

[probuss-Air:rest_api probus$ composer require slim/slim:3.*
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 5 installs, 0 updates, 0 removals
  - Installing psr/container (1.0.0): Loading from cache
  - Installing nikic/fast-route (v1.3.0): Loading from cache
  - Installing psr/http-message (1.0.1): Loading from cache
  - Installing pimple/pimple (v3.3.0): Loading from cache
  - Installing slim/slim (3.12.3): Loading from cache
Writing lock file
Generating autoload files
probuss-Air:rest_api probus$

```

selanjutnya kita buat file php nya kl di window tinggal new file 'index.php' mac ama linux tinggal touch index.php

contoh file lengkap bisa di lihat disini : <https://github.com/malikkurosaki/makuronote/tree/master/php/slim>



kaya diatas tuh penampakkannya

selanjutnya kita edit file php nya , bisa pake bracket , vscode ataupun notepad++ atau yang lain juga boleh

lalu copas code berikut

```

<?php
use \Psr\Http\Message\ServerRequestInterface as Request;
use \Psr\Http\Message\ResponseInterface as Response;
use \Slim\Container;
use \Slim\App;

require 'vendor/autoload.php';

$app = new App();

$app->get('/',function(Request $req,Response $res){
    $res->write('apa kabar');

```

```

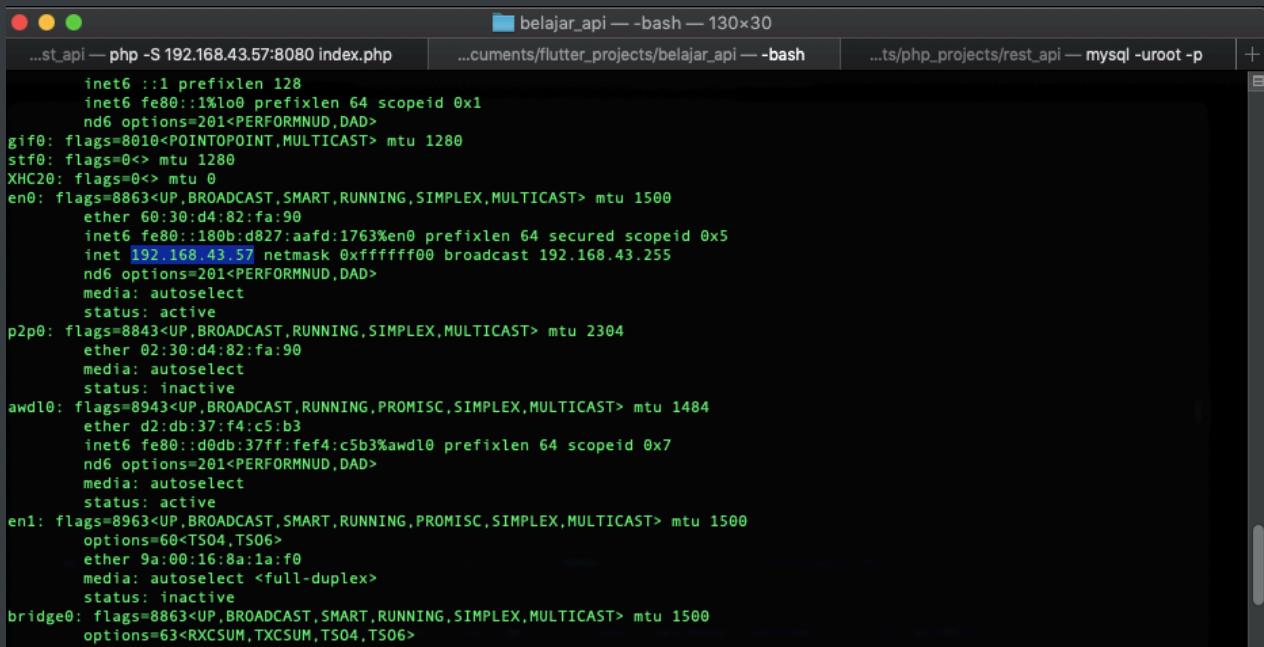
        return $res;
    });

    $app->run();

```

lalu jalankan dengan kode `php -S 192.168.43.57:8080 index.php` di command line

192.168.43.57 ini adalah ip computer , mac dan linux bisa masukkan perintah di commandline
`ifconfig`



```

...st_api — php -S 192.168.43.57:8080 index.php
...cuments/flutter_projects/belajar_api — -bash
...ts/php_projects/rest_api — mysql -uroot -p

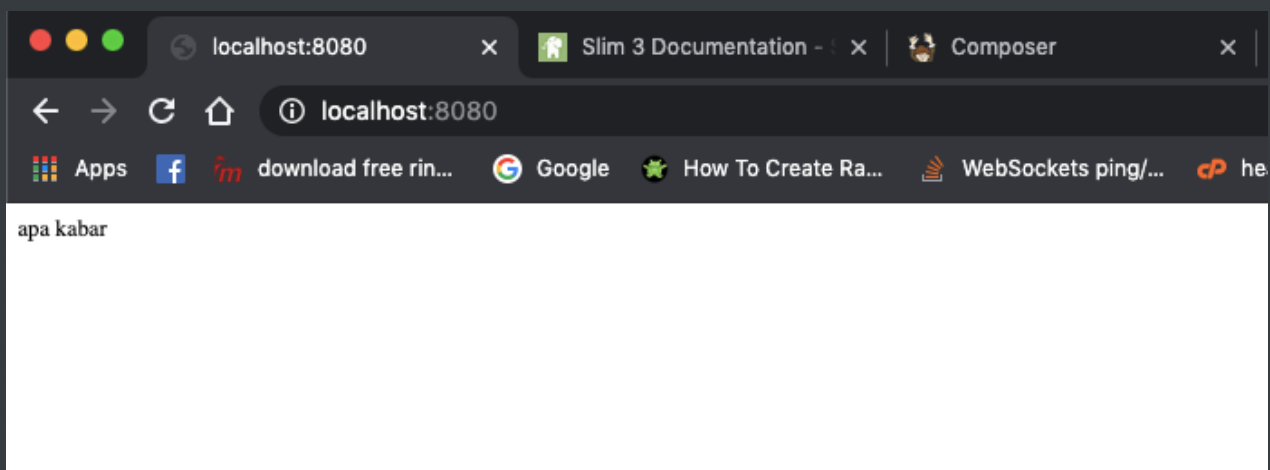
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
XHC20: flags=0<> mtu 0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 60:30:d4:82:fa:90
    inet6 fe80::180b:d827:aafd:1763%en0 prefixlen 64 secured scopeid 0x5
    inet 192.168.43.57 netmask 0xffffff00 broadcast 192.168.43.255
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: active
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
    ether 02:30:d4:82:fa:90
    media: autoselect
    status: inactive
awd10: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1484
    ether d2:db:37:f4:c5:b3
    inet6 fe80::d0db:37ff:fef4:c5b3%awd10 prefixlen 64 scopeid 0x7
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect
    status: active
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=60<TS04,TS06>
    ether 9a:00:16:8a:1a:f0
    media: autoselect <full-duplex>
    status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=63<RXCSUM,TXCSUM,TS04,TS06>

```

di windows pakai `ipconfig`

kl di xampp mungkin langsung aja untuk tutorial lengkap : <http://www.slimframework.com/docs/v3/>

jika berhasil maka akan seperti ini



lanjut ke persiapan mysqlnya

```
mysql> create database belajar_api;
Query OK, 1 row affected (0.00 sec)

mysql>
```

buat database dan bernama belajar_api , di xampp bisa langsung buat di phpmyadmin , jika kurang paham tentang yang ini bisa langsung cari tutornya di youtube atau mbh google

lalu buat table bernama anggota

```
mysql> create table anggota(id int auto_increment primary key,nama_depan varchar(199),nama_belakang varchar(199),panggilan varchar(199),email varchar(199),alamat varchar(199),phone varchar(199));
Query OK, 0 rows affected (0.03 sec)

mysql>
```

lalu isi table itu dengan

```
mysql> insert into anggota values(null,"malik","kurosaki","malik","malik@gmail.com","denpasar","081555555");
Query OK, 1 row affected (0.00 sec)

mysql>
```

kembali ke file index.php

lalu edit filenya sehingga jadi seperti ini

```
<?php
use \Psr\Http\Message\ServerRequestInterface as Request;
use \Psr\Http\Message\ResponseInterface as Response;
use \Slim\Container;
use \Slim\App;

require 'vendor/autoload.php';

$app = new App();
$container = $app->getContainer();

/**
 * deklarasi database mysql
 * host : localhost
 * user : root ( sesuaikan dengan user mysql kalian)
 * pass : makuro123 ( sesuaikan dengan password mysql kalian)
 * database :
 */
$container['db'] = function($container) {
    $con = mysqli_connect('localhost', 'root','makuro123','belajar_api');
    if (!$con) {
        die("Connection failed: " . mysqli_connect_error());
    }
    return $con;
};
```

```

/**
 * main route
 */
$app->get('/',function(Request $req,Response $res){
    $res->write('apa kabar');
    return $res;
});

/**
 * lihat-user
 * implementasi dengan mysql database
 */

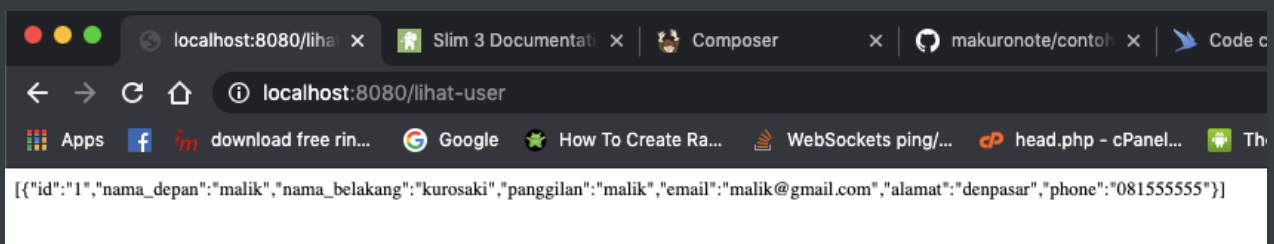
$app->get('/lihat-user',function(Request $req,Response $res) use
($container){
    $sql = "select * from anggota";
    $query = mysqli_query($container->db,$sql);
    $hasil = [];
    while($row = mysqli_fetch_assoc($query)){
        array_push($hasil,$row);
    }
    $res->write(json_encode($hasil));

});

$app->run();

```

jika berhasil maka akan seperti ini



yah kita sedah berhasil membuat api



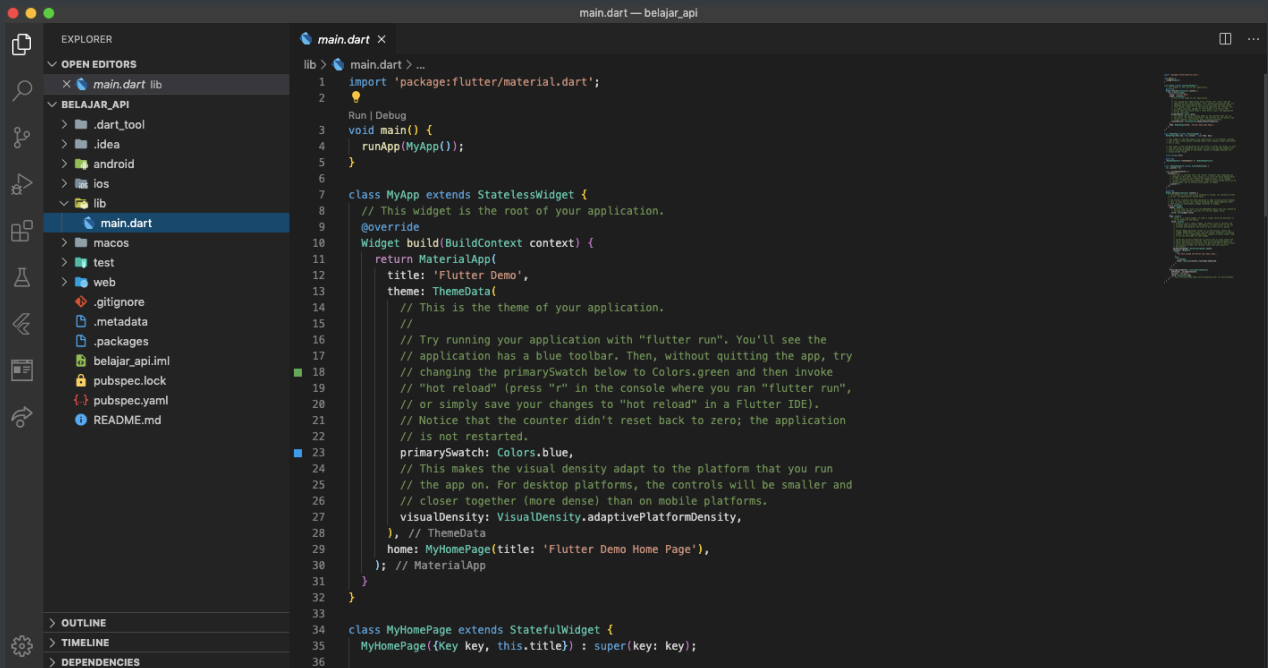
lah kok simple gt doang, ??? ya emang gitu doang .. mau yang lebih. serem ... ? hehe

ok lanjut sekarang implementsi ke flutternya

buat project baru kasi nama 'belajar_api'

```
flutter create --androidx --org dev.malik.belajarapi belajar_api  
cd belajar_api  
code .
```

saya menggunakan `code .` untuk membuka file menggunakan vs code , pastikan sudah menginstall vs code , jika menggunakan android studio bisa langsung create new project belajar_api



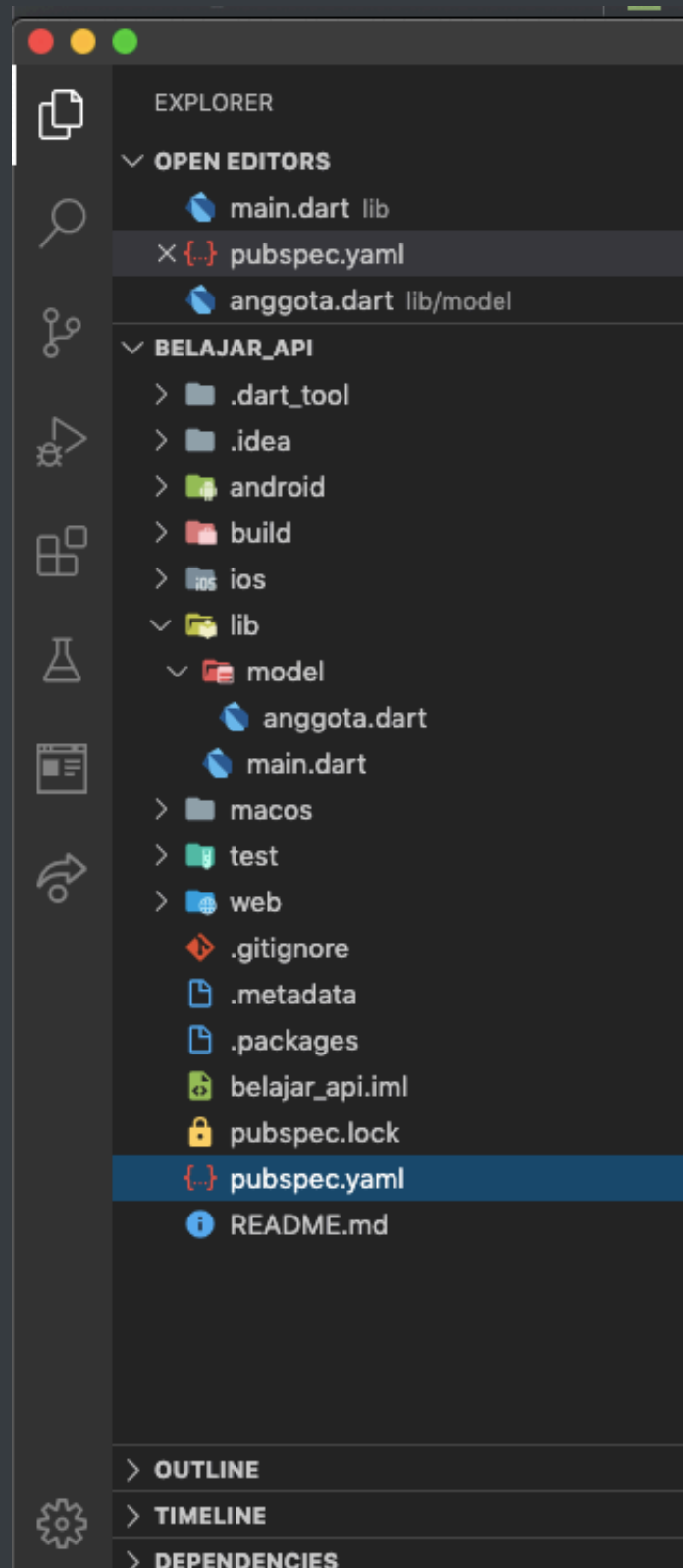
lalu buat folder baru dalam lib folder bernama model didalam model buat file baru namanya anggota didalam file anggota buat class namanya Anggota lalu isikan seperti dibawah

```
class Anggota {  
  String id;  
  String namaDepan;  
  String namaBelakang;  
  String panggilan;  
  String email;  
  String alamat;  
  String phone;  
  
  Anggota(  
    {this.id,  
    this.namaDepan,  
    this.namaBelakang,  
    this.panggilan,  
    this.email,  
    this.alamat,  
    this.phone});  
  
  Anggota.fromJson(Map<String, dynamic> json) {  
    id = json['id'];  
    namaDepan = json['nama_depan'];  
    namaBelakang = json['nama_belakang'];  
    panggilan = json['panggilan'];  
    email = json['email'];  
    alamat = json['alamat'];  
  }  
}
```

```
        phone = json['phone'];
    }

    Map<String, dynamic> toJson() {
        final Map<String, dynamic> data = new Map<String, dynamic>();
        data['id'] = this.id;
        data['nama_depan'] = this.namaDepan;
        data['nama_belakang'] = this.namaBelakang;
        data['panggilan'] = this.panggilan;
        data['email'] = this.email;
        data['alamat'] = this.alamat;
        data['phone'] = this.phone;
        return data;
    }
}
```

cari file yang bernama pubspec.yaml



tambahkan pada bagian dependency

```
dependencies:  
  flutter:  
    sdk: flutter  
  dio:
```

lalu flutter pub get

Lalu di main.dart ubah jadi seperti ini

```
import 'dart:convert';

import 'package:belajar_api/model/anggota.dart';
import 'package:dio/dio.dart';
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MyHome()
    );
  }
}

class MyHome extends StatefulWidget{
  @override
  _MyHomeState createState() => _MyHomeState();
}

class _MyHomeState extends State<MyHome> {

  Future<List<Anggota>> getAnggota()async{
    final res = await new Dio().get('http://192.168.43.57:8080/lihat-user');
    return jsonDecode(res.data).map<Anggota>
((json)=>Anggota.fromJson(json)).toList();
  }
  @override
  Widget build(BuildContext context) {
    print('get anggota');
    return Scaffold(
      body: SafeArea(
        child: SingleChildScrollView(
          child: Container(
```

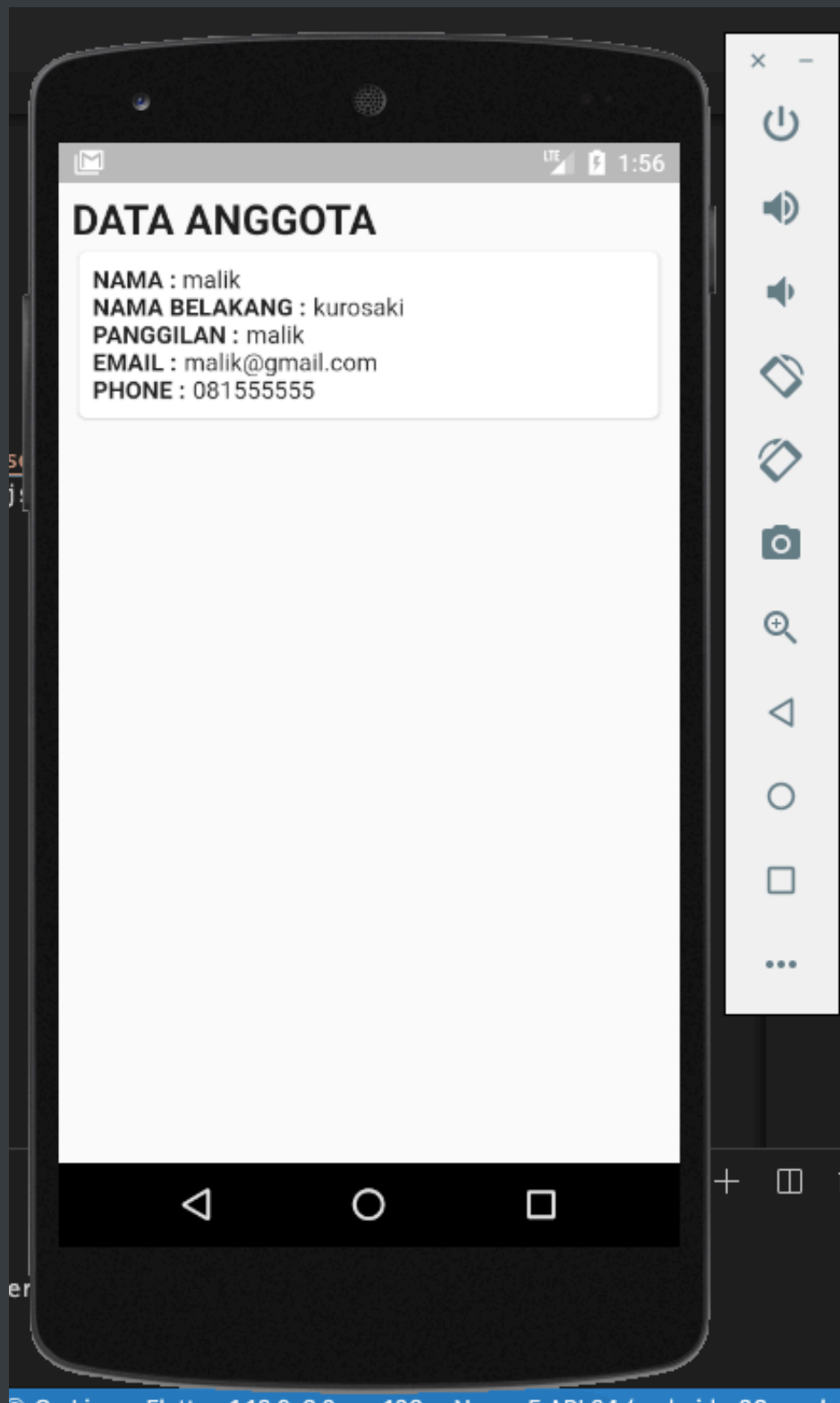
```

padding: EdgeInsets.all(8),
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Text('data anggota'.toUpperCase(),
      style: TextStyle(
        fontSize: 24,
        fontWeight: FontWeight.bold
      ),
    ),
    FutureBuilder(
      future: getAnggota(),
      builder: (context, snapshot){
        if(snapshot.connectionState != ConnectionState.done)
return Center(child: CircularProgressIndicator(),);
        List<Anggota> anggota = snapshot.data;
        return Column(
          children: anggota.map((e)
            => Card(
              child: Container(
                padding: EdgeInsets.all(8),
                child: Column(
                  children: [
                    keterangan('nama', e.namaDepan),
                    keterangan('nama belakang',
e.namaBelakang),

                    keterangan('panggilan', e.panggilan),
                    keterangan('email', e.email),
                    keterangan('phone', e.phone)
                  ],
                ),
              ),
            ).toList(),
          );
        },
      ),
    ],
  ),
),
),
),
),
),
);
}

```

```
Widget keterangan(String title,String text){  
    return Row(  
        children: [  
            Text("$title : ".toUpperCase(),  
                style: TextStyle(  
                    fontWeight: FontWeight.bold  
                ),  
            ),  
            Text(text)  
        ],  
    );  
}
```



jika berhasil akan tampil seperti diatas

selanjutnya kita ke proses pemasukan anggota baru , buka file index.php yang di server

edit hingga seperti berikut

```
<?php
use \Psr\Http\Message\ServerRequestInterface as Request;
use \Psr\Http\Message\ResponseInterface as Response;
```

```

use \Slim\Container;
use \Slim\App;

require 'vendor/autoload.php';

$app = new App();
$container = $app->getContainer();

/**
 * deklarasi database mysql
 * host : localhost
 * user : root ( sesuaikan dengan user mysql kalian)
 * pass : makuro123 ( sesuaikan dengan password mysql kalian)
 * database :
 */
$container['db'] = function($container) {
    $con = mysqli_connect('localhost', 'root','makuro123','belajar_api');
    if (!$con) {
        die("Connection failed: " . mysqli_connect_error());
    }
    return $con;
};

/**
 * main route
 */
$app->get('/',function(Request $req,Response $res){
    $res->write('apa kabar');
    return $res;
});

/**
 * lihat-user
 * implementasi dengan mysql database
 */

$app->get('/lihat-user',function(Request $req,Response $res) use
($container){
    $sql = "select * from anggota";
    $query = mysqli_query($container->db,$sql);
    $hasil = [];
    while($row = mysqli_fetch_assoc($query)){
        array_push($hasil,$row);
    }
}

```



```

        mysqli_close($container->db);
        $res->write(json_encode($hasil));

    });

    /**
     * simpan user baru
     */
    $app->post('/simpan-anggota',function(Request $req,Response $res)use
    ($container){
        $data = $req->getBody();
        $paket = json_decode($data);

        $sql = <<<SQL
        insert into anggota values(
        null,
        "$paket->nama_depan",
        "$paket->nama_belakang",
        "$paket->panggilan",
        "$paket->email",
        "$paket->alamat",
        "$paket->phone"
        )
        SQL;

        if(mysqli_query($container->db,$sql)){
            return $res->write(true);
        }else{
            return $res->write(mysqli_error($container->db));
        }

        mysqli_close($container->db);

    });

    $app->run();

```

pada aplikasi flutter edit main.dart hingga seperti ini

```
import 'dart:convert';

import 'package:belajar_api/model/anggota.dart';
import 'package:dio/dio.dart';
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MyHome()
    );
  }
}

class MyHome extends StatefulWidget{
  @override
  _MyHomeState createState() => _MyHomeState();
}

class _MyHomeState extends State<MyHome> {

  final keyForm = GlobalKey<FormState>();

  static final lsTitle = ['nama depan', 'nama belakang', 'panggilan', 'email', 'alamat', 'phone'];
  final lsForController = List.generate(lsTitle.length, (index)=>TextEditingController());

  Future<List<Anggota>> getAnggota()async{
    final res = await new Dio().get('http://192.168.43.57:8080/lihat-user');
    return jsonDecode(res.data).map<Anggota>((json)=>Anggota.fromJson(json)).toList();
  }
  @override
  Widget build(BuildContext context) {
```

```

print('get anggota');
return Scaffold(
  body: SafeArea(
    child: SingleChildScrollView(
      child: Container(
        padding: EdgeInsets.all(8),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text('data anggota'.toUpperCase(),
              style: TextStyle(
                fontSize: 24,
                fontWeight: FontWeight.bold
              ),
            ),
            FutureBuilder(
              future: getAnggota(),
              builder: (context, snapshot){
                if(snapshot.connectionState != ConnectionState.done)
return Center(child: CircularProgressIndicator(),);
                if(snapshot.data == null) return Center(child:
Text('loading'),);
                List<Anggota> anggota = snapshot.data;
return Column(
  children: anggota.map((e)
    => Card(
      child: Container(
        padding: EdgeInsets.all(8),
        child: Column(
          children: [
            keterangan('nama', e.namaDepan),
            keterangan('nama belakang',
e.namaBelakang),
            keterangan('panggilan', e.panggilan),
            keterangan('email', e.email),
            keterangan('phone', e.phone)
          ],
        ),
      ),
    ),
  ).toList(),
);
  ],
),
),
),
).toList(),
);
},
),
// form input anggota

```

```

Container(
  padding: EdgeInsets.all(16),
  margin: EdgeInsets.only(top:32),
  child: Column(
    children: [
      Text('masukkan anggota baru',
        style: TextStyle(
          fontSize: 24,
          fontWeight: FontWeight.bold
        ),
      ),
      Form(
        key: keyForm,
        child: Column(
          children: List.generate(lsForController.length,
(index) => TextFormField(
          validator: (value) => value.isEmpty?'jangan ada
yang kosong':null,

          controller: lsForController[index],
          decoration: InputDecoration(
            labelText: lsTitle[index]
          ),

        )),
      ),
    ),
    Align(
      alignment: Alignment.centerRight,
      child: OutlineButton(
        child: Text('simpan'),
        onPressed: ()async {
          if(keyForm.currentState.validate()){
            final paketan = new Anggota(
              id: null,
              namaDepan: lsForController[0].text,
              namaBelakang: lsForController[1].text,
              panggilan: lsForController[2].text,
              alamat: lsForController[3].text,
              email: lsForController[4].text,
              phone: lsForController[5].text
            );
            final res = await new
Dio().post('http://192.168.43.57:8080/simpan-anggota',data: paketan);
            print(res.data.toString());
          }

```

```

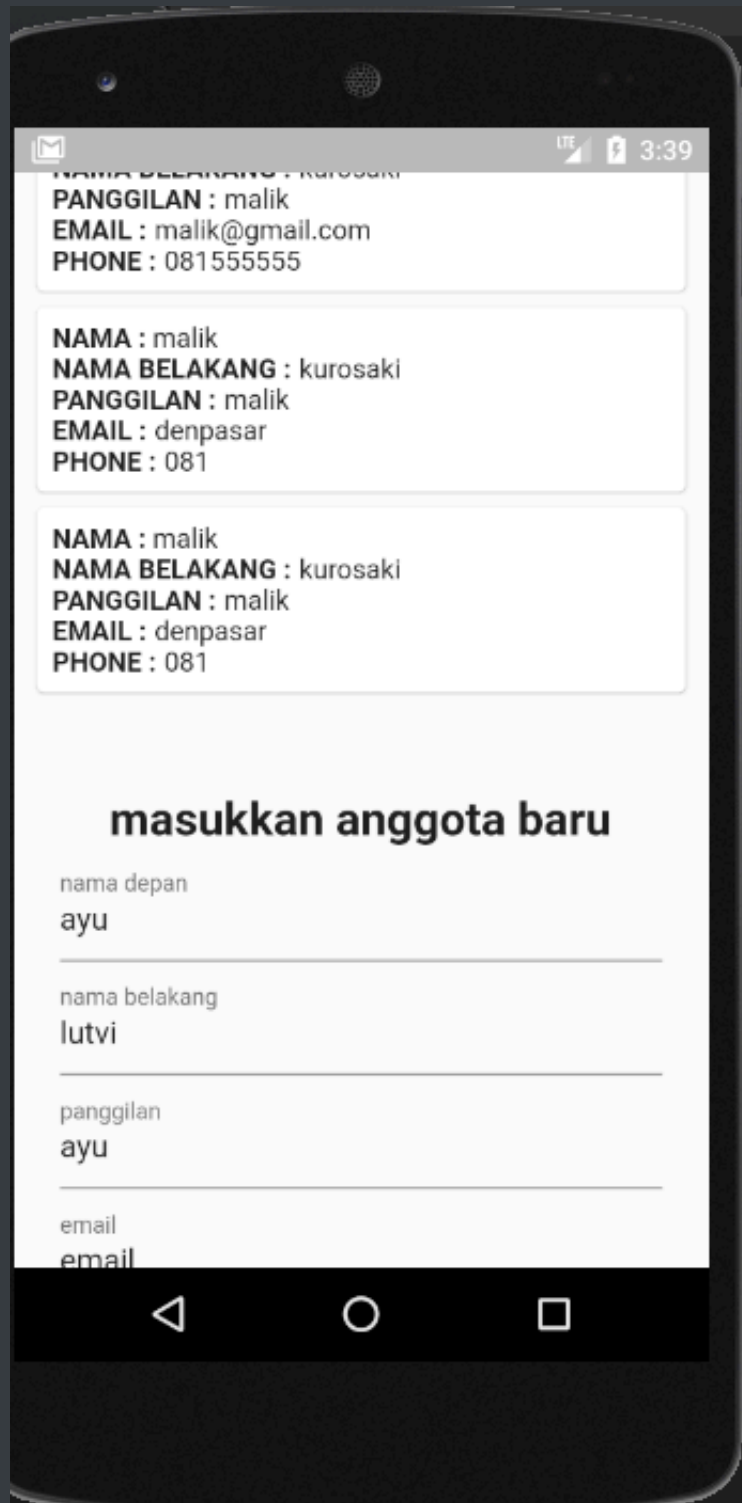
        },
    ),
)
],
),
)
],
),
),
),
),
);
}

Widget keterangan(String title,String text){
    return Row(
        children: [
            Text("$title : ".toUpperCase(),
                style: TextStyle(
                    fontWeight: FontWeight.bold
                ),
            ),
            Text(text),

        ],
    );
}
}

```

jika berhasil penampakan seperti dibawah berikut



tinggal refresh `refresh()` untuk melihat data baru,

pada kode berikut tidak otomatis update , sekarang kita akan membuatnya agat otomatis layout update ketika data ditambahkan

edit pada main.dart hingga seperti dibawah ini

```

import 'dart:convert';

import 'package:belajar_api/model/anggota.dart';
import 'package:dio/dio.dart';
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: MyHome()
    );
  }
}

class MyHome extends StatefulWidget{
  @override
  _MyHomeState createState() => _MyHomeState();
}

class _MyHomeState extends State<MyHome> {

  final keyForm = GlobalKey<FormState>();

  static final lsTitle = ['nama depan', 'nama
belakang', 'panggilan', 'email', 'alamat', 'phone'];
  final lsForController = List.generate(lsTitle.length,
(index)=>TextEditingController());

  Future<List<Anggota>> getAnggota()async{
    final res = await new Dio().get('http://192.168.43.57:8080/lihat-
user');
    return jsonDecode(res.data).map<Anggota>
((json)=>Anggota.fromJson(json)).toList();
  }

  List<Anggota> lsAnggota = <Anggota>[];

  @override

```

```

void initState() {
  Future.microtask(()async{
    List<Anggota> lsAng = await getAnggota();
    if(lsAnggota != lsAng){
      setState(() {
        lsAnggota = lsAng;
      });
    }
  });
  super.initState();
}

@override
Widget build(BuildContext context) {

  print('get anggota');
  return Scaffold(
    body: SafeArea(
      child: SingleChildScrollView(
        child: Container(
          padding: EdgeInsets.all(8),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text('data anggota'.toUpperCase(),
                style: TextStyle(
                  fontSize: 24,
                  fontWeight: FontWeight.bold
                ),
              ),
              Column(
                children: lsAnggota.isEmpty?[Text('loading')]
                  :lsAnggota.map((e)
                    => Card(
                      child: Container(
                        padding: EdgeInsets.all(8),
                        child: Column(
                          children: [
                            keterangan('nama', e.namaDepan),
                            keterangan('nama belakang', e.namaBelakang),
                            keterangan('panggilan', e.panggilan),
                            keterangan('email', e.email),
                            keterangan('phone', e.phone)
                          ],
                        ),
                      ),
                ),
              ),
            ],
          ),
        ),
      ),
    ),
  );
}

```



```

        ),
      ),
    ).toList(),
  ),
  // form input anggota
  Container(
    padding: EdgeInsets.all(16),
    margin: EdgeInsets.only(top: 32),
    child: Column(
      children: [
        Text('masukkan anggota baru',
          style: TextStyle(
            fontSize: 24,
            fontWeight: FontWeight.bold
          ),
        ),
      ],
    ),
    Form(
      key: keyForm,
      child: Column(
        children: List.generate(lsForController.length,
(index) => TextFormField(
          validator: (value) => value.isEmpty?'jangan ada
yang kosong':null,

          controller: lsForController[index],
          decoration: InputDecoration(
            labelText: lsTitle[index]
          ),
        )),
      ),
    ),
    Align(
      alignment: Alignment.centerRight,
      child: OutlineButton(
        child: Text('simpan'),
        onPressed: ()async {
          if(keyForm.currentState.validate()){
            final paketan = new Anggota(
              id: null,
              namaDepan: lsForController[0].text,
              namaBelakang: lsForController[1].text,
              panggilan: lsForController[2].text,
              alamat: lsForController[3].text,
              email: lsForController[4].text,
              phone: lsForController[5].text

```

```

        );
        final res = await new
Dio().post('http://192.168.43.57:8080/simpan-anggota',data: paketan);
        print(res.data);
        final listAng = await getAnggota();
        setState(() {
            lsAnggota = listAng;
        });
    }
},
),
),
],
),
),
],
),
),
),
),
);
}

Widget keterangan(String title,String text){
    return Row(
        children: [
            Text("$title : ".toUpperCase(),
                style: TextStyle(
                    fontWeight: FontWeight.bold
                ),
            ),
            Text(text),
        ],
    );
}
}

```

jika berhasil setiap kali and menambahkan data , maka data akan otomatis terupdate di layout

sekarang lanjut ke penghapusan

edit index.php seperti dibawah

```
<?php
use \Psr\Http\Message\ServerRequestInterface as Request;
use \Psr\Http\Message\ResponseInterface as Response;
use \Slim\Container;
use \Slim\App;

require 'vendor/autoload.php';

$app = new App();
$container = $app->getContainer();

/**
 * deklarasi database mysql
 * host : localhost
 * user : root ( sesuaikan dengan user mysql kalian)
 * pass : makuro123 ( sesuaikan dengan password mysql kalian)
 * database :
 */
$container['db'] = function($container) {
    $con = mysqli_connect('localhost', 'root', 'makuro123', 'belajar_api');
    if (!$con) {
        die("Connection failed: " . mysqli_connect_error());
    }
    return $con;
};

/**
 * main route
 */
$app->get('/', function(Request $req, Response $res){
    $res->write('apa kabar');
    return $res;
});

/**
 * lihat-user
 * implementasi dengan mysql database
 */

$app->get('/lihat-user', function(Request $req, Response $res) use
($container){
```

```

        $sql = "select * from anggota";
        $query = mysqli_query($container->db,$sql);
        $hasil = [];
        while($row = mysqli_fetch_assoc($query)){
            array_push($hasil,$row);
        }
        mysqli_close($container->db);
        $res->write(json_encode($hasil));

    });

    /**
     * simpan user baru
     */
    $app->post('/simpan-anggota',function(Request $req,Response $res)use
    ($container){
        $data = $req->getBody();
        $paket = json_decode($data);

        $sql = <<<SQL
        insert into anggota values(
        null,
        "$paket->nama_depan",
        "$paket->nama_belakang",
        "$paket->panggilan",
        "$paket->email",
        "$paket->alamat",
        "$paket->phone"
        )
        SQL;

        if(mysqli_query($container->db,$sql)){
            return $res->write(true);
        }else{
            return $res->write(mysqli_error($container->db));
        }

        mysqli_close($container->db);

    });

    //delete user
    $app->delete('/hapus-anggota/{id}',function(Request $req,Response
    $res,$args)use ($container){

```

```

        $id = $args['id'];
        $sql = <<<SQL
delete from anggota where id = $id
SQL;
        $query = mysqli_query($container->db,$sql);
        if($query){
            return $res->write(true);
        }else{
            return $res->write(mysqli_error($container->db));
        }
        mysqli_close($container->db);
    });

    $app->run();

```

dan untuk main.dart seperti dibawah

```

import 'dart:convert';

import 'package:belajar_api/model/anggota.dart';
import 'package:dio/dio.dart';
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget{
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            debugShowCheckedModeBanner: false,
            home: MyHome()
        );
    }
}

```

```

class MyHome extends StatefulWidget{
  @override
  _MyHomeState createState() => _MyHomeState();
}

class _MyHomeState extends State<MyHome> {

  final keyForm = GlobalKey<FormState>();

  static final lsTitle = ['nama depan', 'nama
belakang', 'panggilan', 'email', 'alamat', 'phone'];
  final lsForController = List.generate(lsTitle.length,
(index)=>TextEditingController());

  Future<List<Anggota>> getAnggota()async{
    final res = await new Dio().get('http://192.168.43.57:8080/lihat-
user');
    return jsonDecode(res.data).map<Anggota>
((json)=>Anggota.fromJson(json)).toList();
  }

  List<Anggota> lsAnggota = <Anggota>[];

  @override
  void initState() {
    Future.microtask(()async{
      List<Anggota> lsAng = await getAnggota();
      if(lsAnggota != lsAng){
        setState(() {
          lsAnggota = lsAng;
        });
      }
    });
    super.initState();
  }

  @override
  Widget build(BuildContext context) {

    print('get anggota');
    return Scaffold(
      body: SafeArea(
        child: SingleChildScrollView(
          child: Container(
            padding: EdgeInsets.all(8),

```

```

child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Text('data anggota'.toUpperCase(),
      style: TextStyle(
        fontSize: 24,
        fontWeight: FontWeight.bold
      ),
    ),
    Column(
      children: lsAnggota.isEmpty?[Text('loading')]
        :lsAnggota.map((e)
          => listDataAnggota(e)
        ).toList(),
    ),
  ],
)
// form input anggota
Container(
  padding: EdgeInsets.all(16),
  margin: EdgeInsets.only(top:32),
  child: Column(
    children: [
      Text('masukkan anggota baru',
        style: TextStyle(
          fontSize: 24,
          fontWeight: FontWeight.bold
        ),
      ),
      Form(
        key: keyForm,
        child: Column(
          children: List.generate(lsForController.length,
(index) => TextFormField(
          validator: (value) => value.isEmpty?'jangan ada
yang kosong':null,

          controller: lsForController[index],
          decoration: InputDecoration(
            labelText: lsTitle[index]
          ),
        )),
      ),
    ],
  ),
  Align(
    alignment: Alignment.centerRight,
    child: OutlineButton(

```



```

        alignment: Alignment.bottomRight,
        child: OutlineButton(
          child: Text('hapus'),
          onPressed: () async{
            final res = await new
Dio().delete('http://192.168.43.57:8080/hapus-anggota/${e.id}');
            print(res.data);
            final listAng = await getAnggota();
            setState(() {
              lsAnggota = listAng;
            });
          },
        ),
      ),
    ],
  ),
);
}

Widget keterangan(String title,String text){
  return Row(
    children: [
      Text("$title : ".toUpperCase(),
        style: TextStyle(
          fontWeight: FontWeight.bold
        ),
      ),
      Text(text),
    ],
  );
}
}

```

jika berhasil , ketika anda menekan tombol hapus maka list anggota akan otomatis terhapus,

udah cukup segitu ya , untuk update mekanismenya hampir sama kok,

seperti biasa jika ada pertanyaan langsung ke wa 081338929722

atau email : Kurosakiblackangel@gmail.com

atau ketik di pencarian google "malikkurosaki"