

PEMROGRAMAN FLUTTER DENGAN WEB API LARAVEL DASAR



Disusul Oleh :

Mokhamad Ramdhani Raharjo, S.Kom., M.Kom

FAKULTAS TEKNOLOGI INFORMASI

**UNIVERSITAS ISLAM KALIMANTAN MUHAMMAD ARSYAD AL BANJARI
BANJARMASIN**

2024

DAFTAR ISI

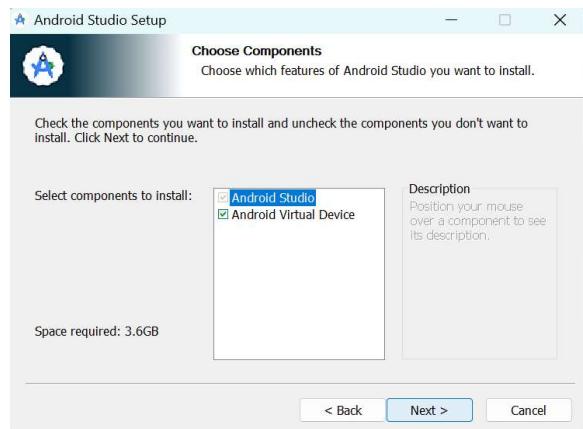
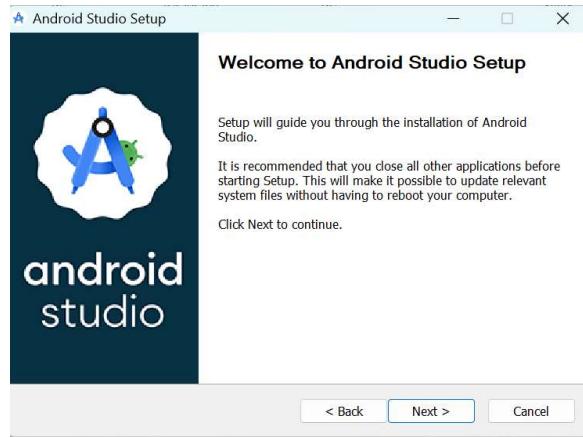
PERTEMUAN I.....	3
1.1 PERSIAPAN TOOLS	3
1.2 PERMBUATAN PROJECT WEB-API SEKOLAH.....	13
1.3 PENGUJIAN ENDPOINT DAN INSTALASI POSTMAN	17
1.4 PEMBUATAN PROJECT AWAL FLUTTER.....	20
PERTEMUAN II	27
2.1 Kode Tambah Data Mahasiswa.....	27
2.2 Kode View Mahasiswa	29
2.3 Kode Ubah Data Mahasiswa.....	31
PERTEMUAN III.....	36
3.1 Persiapan Model ,Controller, Route dan Migration Matakuliah	36
3.2 Persiapan Model, Controller dan View Mata Kuliah pada Flutter Mobile.....	40
3.3 Report PDF data Mata Kuliah.....	47
PERTEMUAN IV	50
4.1 Persiapan Menu Side atau Drawer Menu	50
4.2 Persiapan Web API Jadwal	52
4.3 Persiapan Model,Controller dan View Pada Flutter.....	55
4.4 Persiapan Slide Menu	60

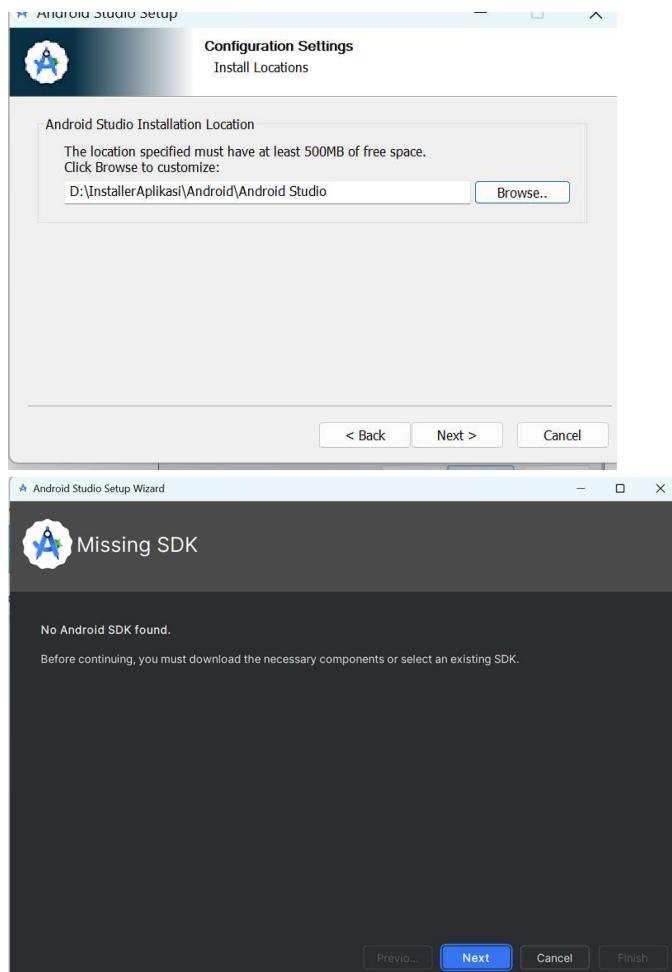
PERTEMUAN I

1.1 PERSIAPAN TOOLS

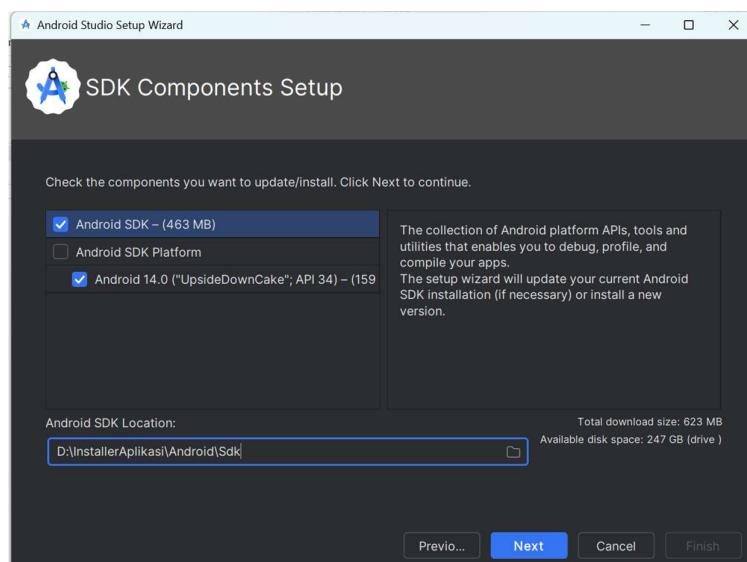
▪ INTALASI ANDROID STUDIO

- Silahkan download dilink berikut ini :
<https://developer.android.com/studio>
- Tahapan Instalasi Android Studio

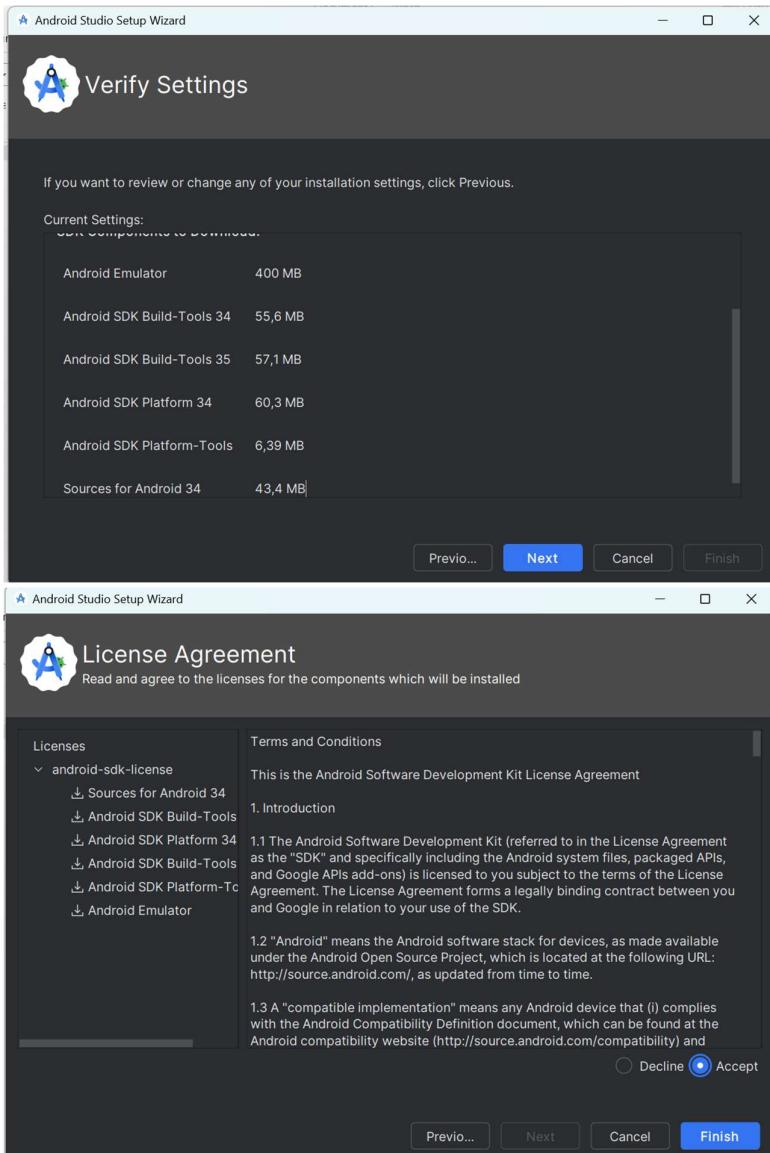




Klik Tombol Next untuk instalasi SDK Android

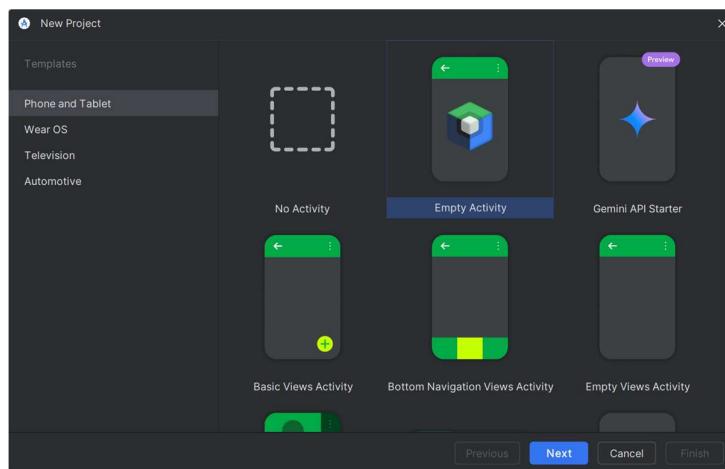
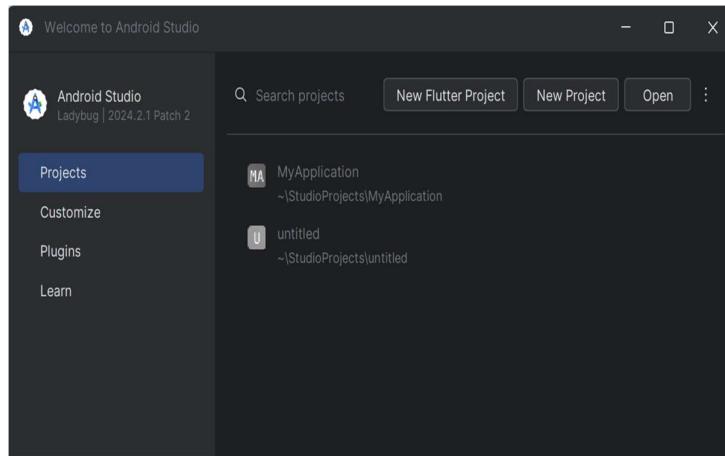


Silahkan tentukan lokasi instalasi Sdk

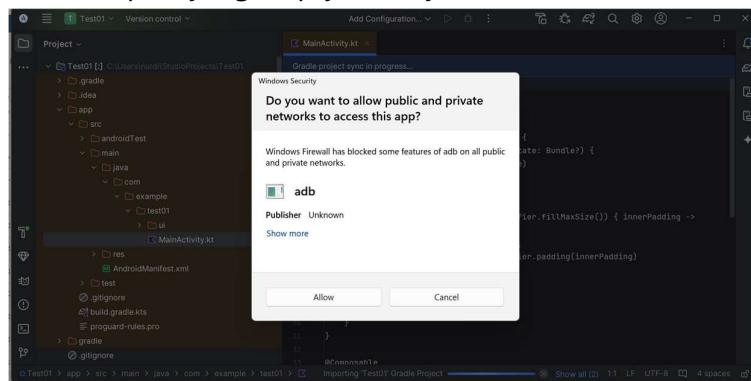


Pilih Accept dan Klik Tombol Finish dan tunggu sampai proses download selesai

- Project Baru Pada Android Studio
Pada Instalasi pertama android studio buat project baru dengan nama “test01” klik tombol New Project



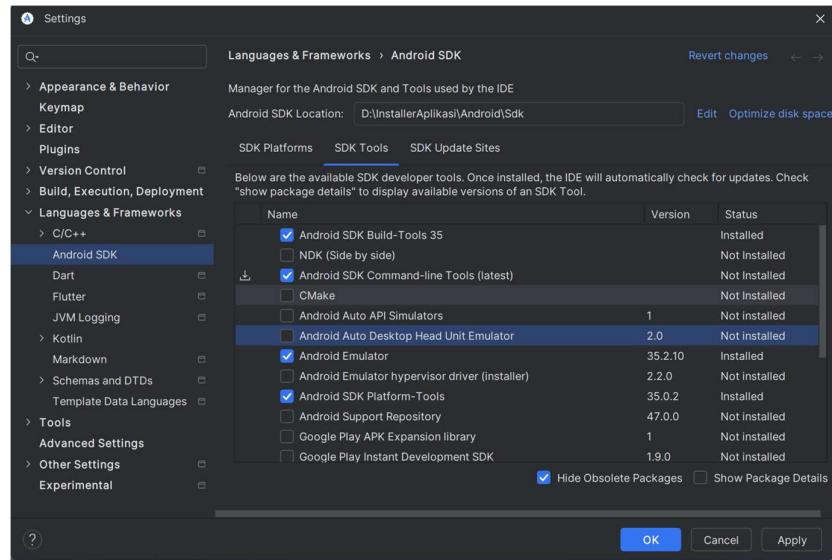
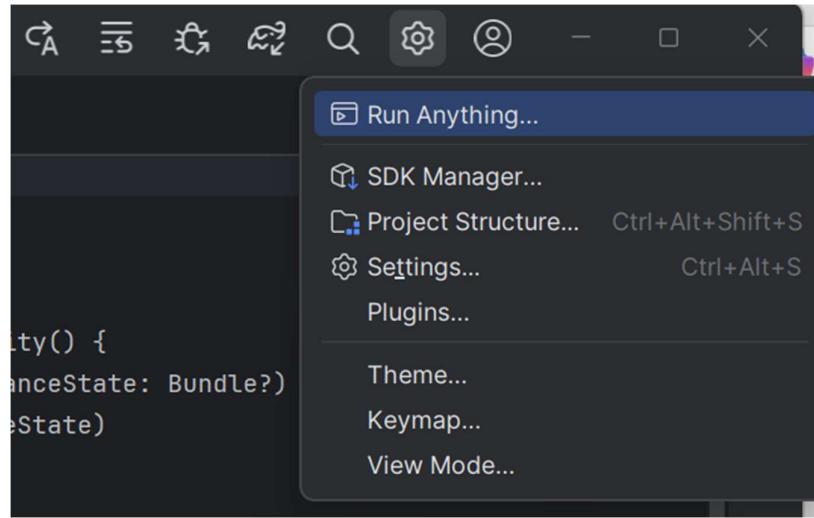
Silahkan pilih yang Empty Activity kemudian klik tombol Next



Klik Allow untuk menginjinkan adb

■ KONFIGURASI SDK

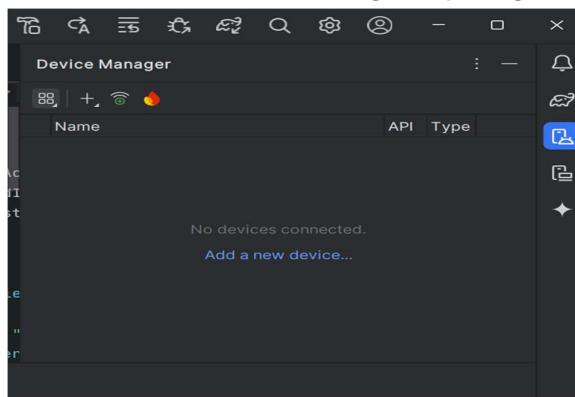
Klik menu icon setting kemudian pilih menu SDK Manager



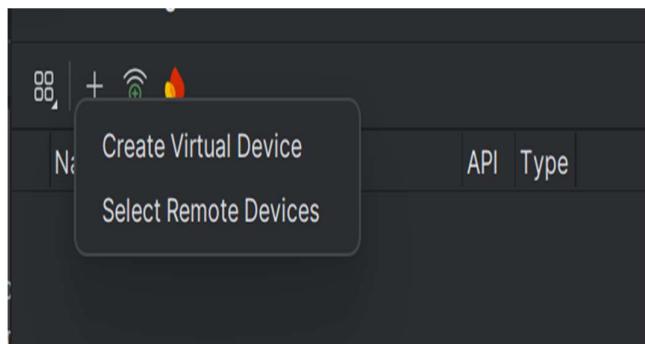
Silahkan Centang seperti tampilan gambar diatas, jika sudah klik tombol apply. Tunggu sampai proses selesai dan klik tombol OK.

▪ KONFIGURASI EMULATOR

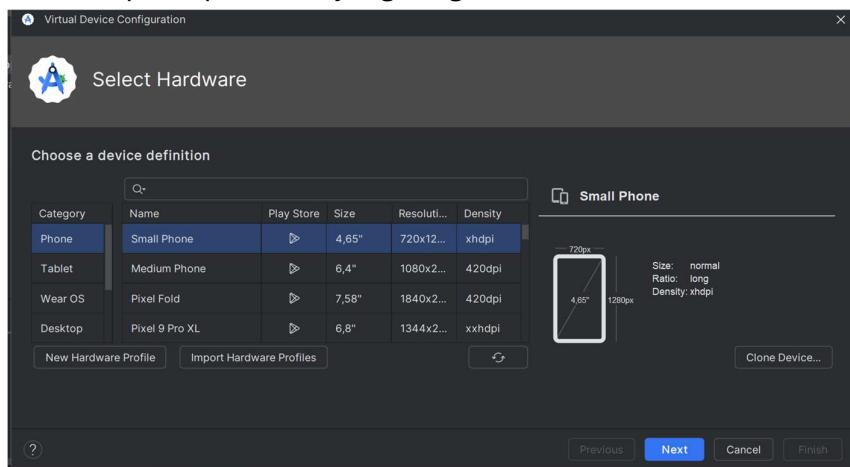
Klik menu ikon Device Manager seperti gambar dibawah ini:



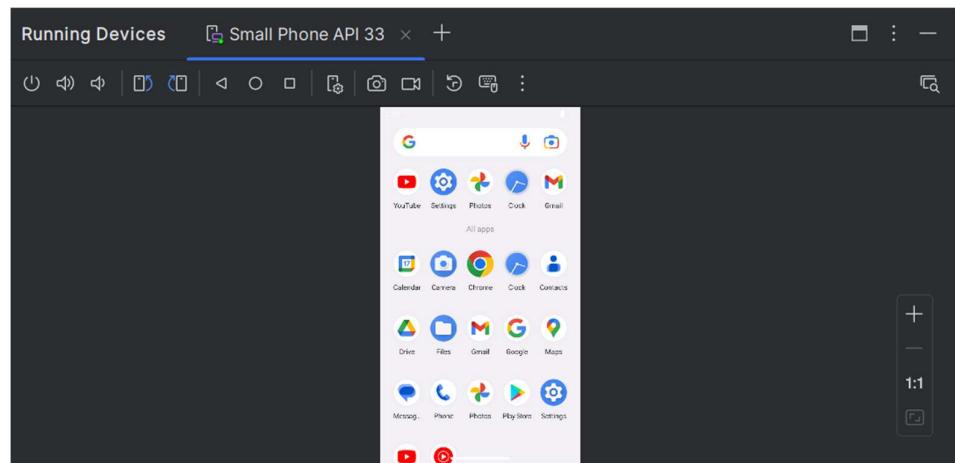
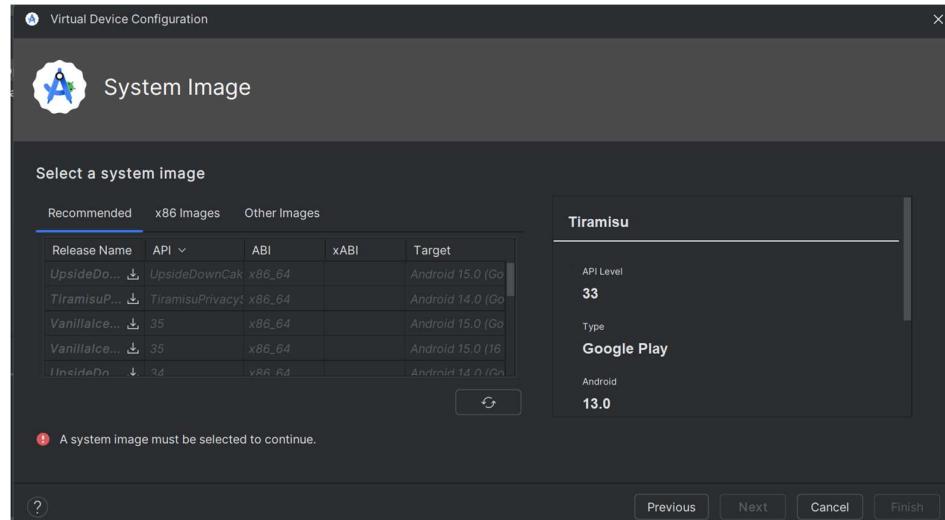
Klik Icon Plus kemudian klik Create Virtual Device



Silahkan pilih tipe device yang diingin diinstall Kemudian Klik Tombol Next

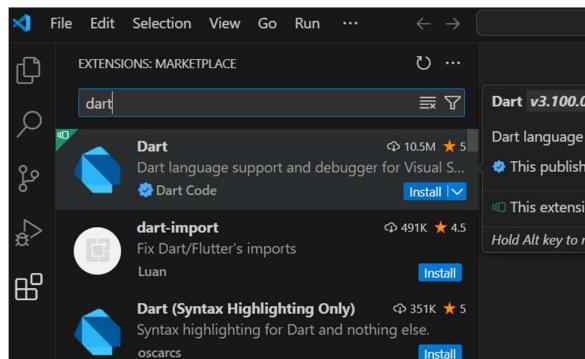


Silahkan pilih System Image yang ingin diinstall pada emulator , klik tombol next untuk proses lanjut.



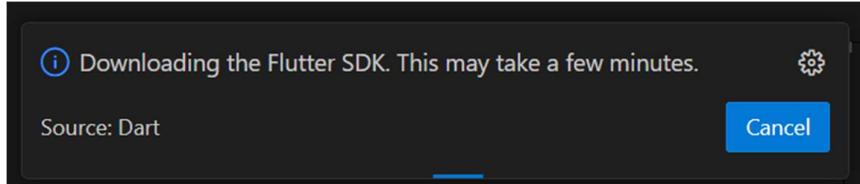
▪ INSTALASI FLUTTER

- Jalankan aplikasi VS Code pada menu extention ketikan Dart kemudian install:



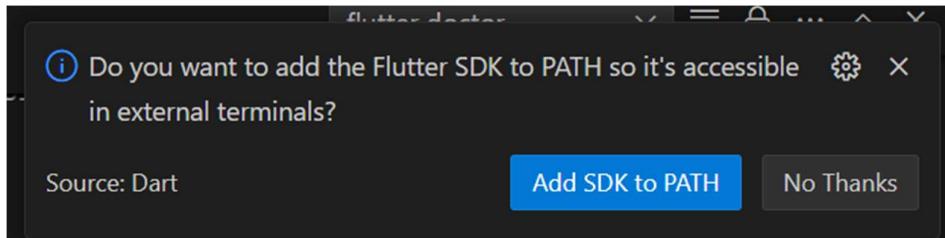
- Pada vs code jika belum ada Dart SDK maka klik tombol Download SDK

- Arahkan lokasi folder ketempat yang diinginkan



Silahkan tunggu sampai proses selesai

- Klik Add SDK to PATH



- Tekan Ctrl + Shift + P pada VS Code pilih Run Flutter Doctor

Usahakan list semua kondisi tercentang semua jika masih ada yang silang atau tanda seru maka silahkan diperbaiki seperti contoh dibawah ini memiliki kesalahan, maka kita perbaiki:

```
...
[flutter] flutter doctor -v
[!] Flutter (Channel stable, 3.24.5, on Microsoft Windows [Version 10.0.22631.4460], locale id-ID)
  • Flutter version 3.24.5 on channel stable at D:\Installer\Aplikasi\Android\flutter
  ! The flutter binary is not on your path. Consider adding D:\Installer\Aplikasi\Android\flutter\bin to your path.
  ! The dart binary is not on your path. Consider adding D:\Installer\Aplikasi\Android\flutter\bin to your path.
  • Upstream repository https://github.com/flutter/flutter.git
  • Framework revision dec2ee5clf (5 days ago), 2024-11-13 11:13:06 -0800
  • Engine revision a18df97ca5
  • Dart version 3.5.4
  • DevTools version 2.37.3
  • If those were intentional, you can disregard the above warnings; however it is recommended to use "git" directly to perform update checks and upgrades.

[v] Windows Version (Installed version of Windows is version 10 or higher)

[X] Android toolchain - develop for Android devices
  X Unable to locate Android SDK.
    Install Android Studio from: https://developer.android.com/studio/index.html
    On first launch it will assist you in installing the Android SDK components.
    (or visit https://Flutter.dev/to/windows-android-setup for detailed instructions).
    If the Android SDK has been installed to a custom location, please use
      `flutter config --android.sdk` to update to that location.

[v] Chrome - develop for the web
  • Chrome at C:\Program Files\Google\Chrome\Application\chrome.exe

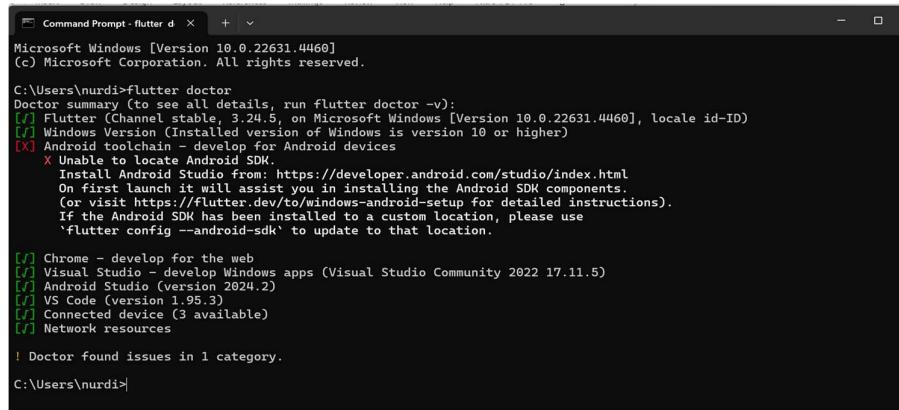
[v] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.11.5)
  • Visual Studio at C:\Program Files\Microsoft Visual Studio\2022\Community
  • Visual Studio Community 2022 version 17.11.35327.3
  • Windows 10 SDK version 10.0.22621.0

[v] Android Studio (version 2024.2)
  • Android Studio at D:\Installer\Aplikasi\Android\Android Studio
  • Flutter plugin can be installed from:
    | https://plugins.jetbrains.com/plugin/9212-flutter
  • Dart plugin can be installed from:
    | https://plugins.jetbrains.com/plugin/6351-dart
  • Java version OpenJDK Runtime Environment (build 21.0.3+12282718-b509.11)

[v] VS Code (version 1.95.3)
  • VS Code at C:\Users\nurdii\AppData\Local\Programs\Microsoft VS Code
  • Flutter extension version 3.100.0
```

- Untuk penyelesaian SDK flutter belum terdaftar pada path windows secara otomatis bisa ditambahkan manual pada edit Environment Variable – System Variable – Path – Edit – New - Tambahkan lokasi flutter\bin. Jika sudah ditambahkan maka akan bisa meyelesaikan Android toolchain. (Jika sudah centang semua abaikan tahap ini). Jika Sudah ditambahkan maka

pengecekan flutter doctor bisa dijalankan Command Prompt Windows seperti berikut ini:



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nurdi>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.24.5, on Microsoft Windows [Version 10.0.22631.4460], locale id-ID)
[!] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices
    ✘ Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/to/windows-android-setup for detailed instructions).
      If the Android SDK has been installed to a custom location, please use
        'flutter config --android-sdk' to update to that location.

[!] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.11.5)
[!] Android Studio (version 2024.2)
[!] VS Code (version 1.95.3)
[!] Connected device (3 available)
[!] Network resources

! Doctor found issues in 1 category.

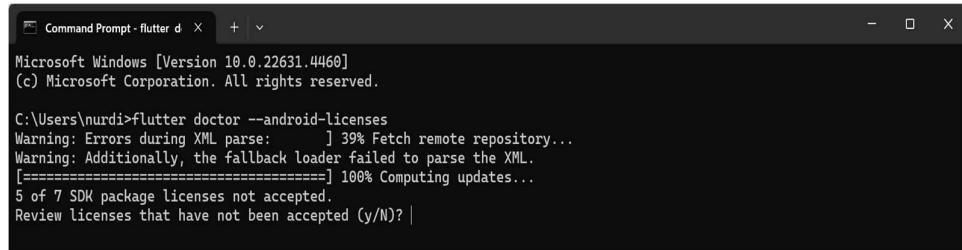
C:\Users\nurdi>
```

- Jalankan cmd kemudian ketikan seperti berikut:



```
C:\Users\nurdi>flutter config --android-sdk D:\InstallerAplikasi\Android\Sdk
Setting "android-sdk" value to "D:\InstallerAplikasi\Android\Sdk".
```

Ketikan lagi pada cmd seperti dibawah ini, kemudian berikan persetujuan 'y' untuk menerima konfigurasi :



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nurdi>flutter doctor --android-licenses
Warning: Errors during XML parse: [ 3% Fetch remote repository...
Warning: Additionally, the fallback loader failed to parse the XML.
[=====] 100% Computing updates...
5 of 7 SDK package licenses not accepted.
Review licenses that have not been accepted (y/N)? |
```

```
C:\Users\nurdi>flutter doctor -v
operable program or batch file.

C:\Users\nurdi>flutter doctor -v
[✓] Flutter (Channel stable, 3.24.5, on Microsoft Windows [Version 10.0.22631.4460], locale id-ID)
  • Flutter version 3.24.5 on channel stable at D:\InstallerAplikasi\Android\flutter
  • Upstream repository https://github.com/flutter/flutter.git
  • Framework revision dec2ee5c1f (5 days ago), 2024-11-13 11:13:06 -0800
  • Engine revision a18df97ca5
  • Dart version 3.5.4
  • DevTools version 2.37.3

[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
  • Android SDK at D:\InstallerAplikasi\Android\Sdk
  • Platform android-34, build-tools 34.0.0
  • Java binary at: D:\InstallerAplikasi\Android\Android Studio\jbr\bin\java
  • Java version OpenJDK Runtime Environment (build 21.0.3+12282718-b509.11)
  • All Android licenses accepted.

[✓] Chrome - develop for the web
  • Chrome at C:\Program Files\Google\Chrome\Application\chrome.exe

[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.11.5)
  • Visual Studio at C:\Program Files\Microsoft Visual Studio\2022\Community
  • Visual Studio Community 2022 version 17.11.35327.3
  • Windows 10 SDK version 10.0.22621.0

[✓] Android Studio (version 2024.2)
  • Android Studio at D:\InstallerAplikasi\Android\Android Studio
  • Flutter plugin can be installed from:
    https://plugins.jetbrains.com/plugin/9212-flutter
  • Dart plugin can be installed from:
    https://plugins.jetbrains.com/plugin/6351-dart
  • Java version OpenJDK Runtime Environment (build 21.0.3+12282718-b509.11)

[✓] VS Code (version 1.95.3)
  • VS Code at C:\Users\nurdi\AppData\Local\Programs\Microsoft VS Code
  • Flutter extension version 3.100.0

[✓] Connected device (3 available)
  • Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.22631.4460]
  • Chrome (web)      • chrome   • web-javascript • Google Chrome 130.0.6723.119
  • Edge (web)        • edge     • web-javascript • Microsoft Edge 131.0.2903.51

[✓] Network resources
  • All expected network resources are available.

• No issues found!

C:\Users\nurdi>
```

```
Command Prompt - flutter doctor + - v

operable program or batch file.

C:\Users\nurdii>flutter doctor -v
[✓] Flutter (Channel stable, 3.24.5, on Microsoft Windows [Version 10.0.22631.4460], locale id-ID)
  • Flutter version 3.24.5 on channel stable at D:\InstallerAplikasi\Android\flutter
  • Upstream repository https://github.com/flutter/flutter.git
  • Framework revision dec2ee5c1f (5 days ago), 2024-11-13 11:13:06 -0800
  • Engine revision a18df97ca5
  • Dart version 3.5.4
  • DevTools version 2.37.3

[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
  • Android SDK at D:\InstallerAplikasi\Android\Sdk
  • Platform android-34, build-tools 34.0.0
  • Java binary at: D:\InstallerAplikasi\Android\Android Studio\jbr\bin\java
  • Java version OpenJDK Runtime Environment (build 21.0.3+12282718-b509.11)
  • All Android licenses accepted.

[✓] Chrome - develop for the web
  • Chrome at C:\Program Files\Google\Chrome\Application\chrome.exe

[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.11.5)
  • Visual Studio at C:\Program Files\Microsoft Visual Studio\2022\Community
  • Visual Studio Community 2022 version 17.11.35327.3
  • Windows 10 SDK version 10.0.22621.0

[✓] Android Studio (version 2024.2)
  • Android Studio at D:\InstallerAplikasi\Android\Android Studio
  • Flutter plugin can be installed from:
    https://plugins.jetbrains.com/plugin/9212-flutter
  • Dart plugin can be installed from:
    https://plugins.jetbrains.com/plugin/6351-dart
  • Java version OpenJDK Runtime Environment (build 21.0.3+12282718-b509.11)

[✓] VS Code (version 1.95.3)
  • VS Code at C:\Users\nurdii\AppData\Local\Programs\Microsoft VS Code
  • Flutter extension version 3.100.0

[✓] Connected device (3 available)
  • Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.22631.4460]
  • Chrome (web)      • chrome   • web-javascript • Google Chrome 130.0.6723.119
  • Edge (web)        • edge     • web-javascript • Microsoft Edge 131.0.2903.51

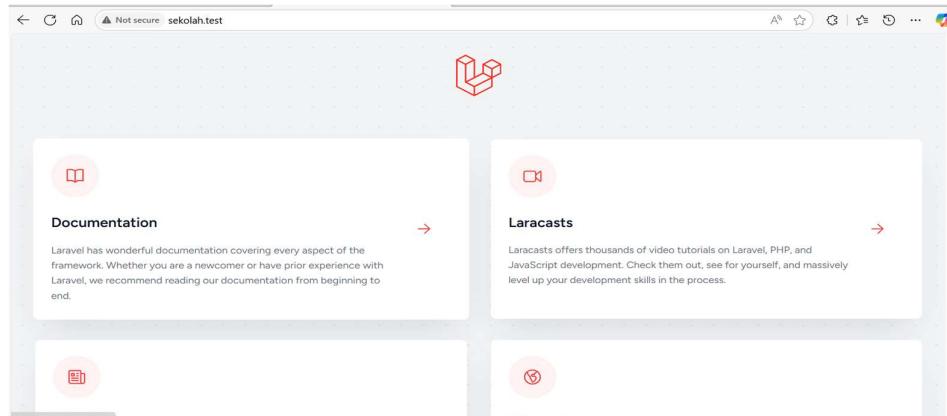
[✓] Network resources
  • All expected network resources are available.

• No issues found!

C:\Users\nurdii|
```

1.2 PERMBUATAN PROJECT WEB-API SEKOLAH

- Jalankan Laragon buat project dengan nama ‘mahasiswa-api’



- Silahkan buka file .env dan konfigurasi koneksi database seperti berikut ini :

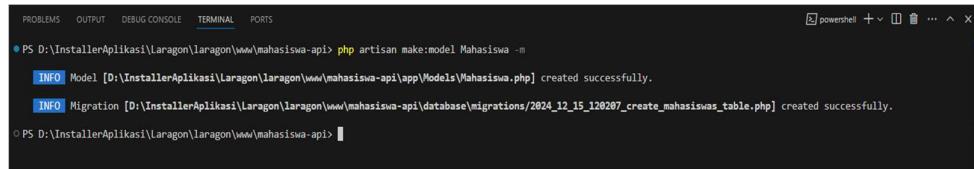
```

0
1 DB_CONNECTION=mysql
2 DB_HOST=127.0.0.1
3 DB_PORT=3306
4 DB_DATABASE=mahasiswa-api
5 DB_USERNAME=root
6 DB_PASSWORD=
7

```

- Persiapan Model dan Migrasi

Ketikan Migration pada terminal VS Code
php artisan make:model Mahasiswa -m

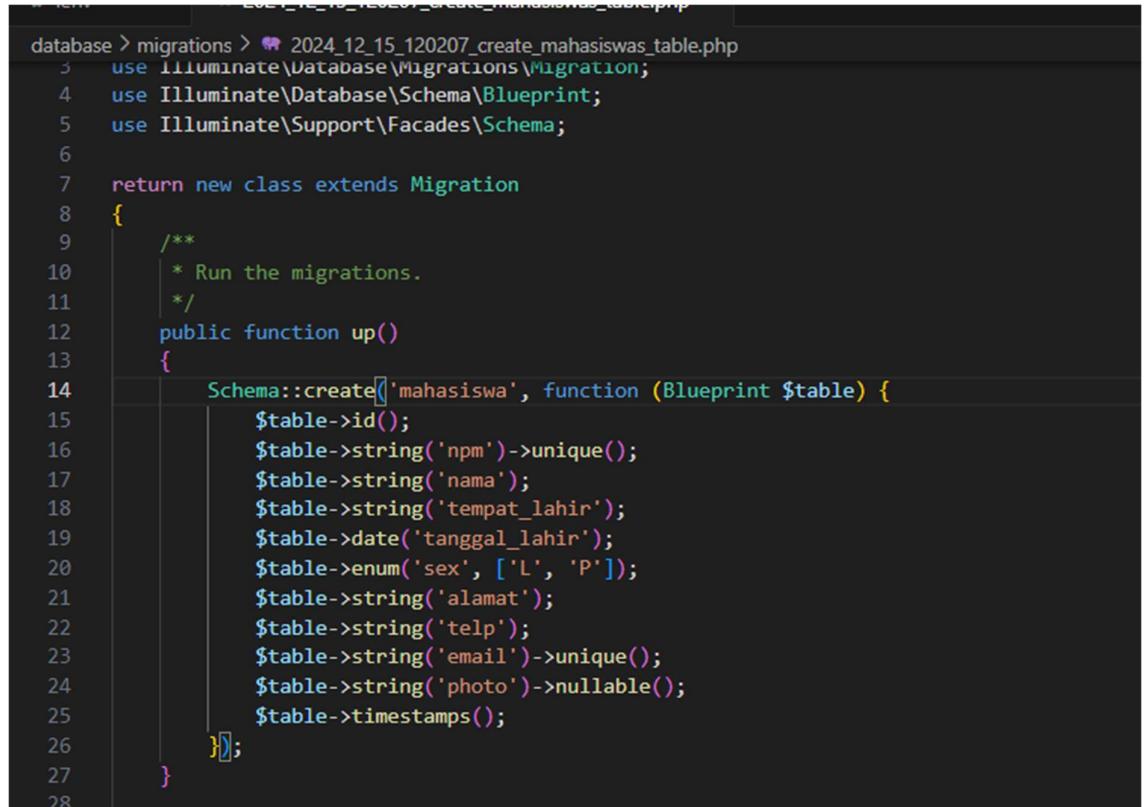


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api> php artisan make:model Mahasiswa -m
[INFO] Model [D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api\app\Models\Mahasiswa.php] created successfully.
[INFO] Migration [D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api\database\migrations\2024_12_15_120207_create_mahasiswas_table.php] created successfully.
PS D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api>

```

- Pada file migration create_mahasiswa_table ketikan kode sebagai berikut:



```

database > migrations > 2024_12_15_120207_create_mahasiswas_table.php
1  use Illuminate\Database\Migrations\Migration;
2  use Illuminate\Database\Schema\Blueprint;
3  use Illuminate\Support\Facades\Schema;
4
5  return new class extends Migration
6  {
7      /**
8      * Run the migrations.
9      */
10     public function up()
11     {
12         Schema::create('mahasiswa', function (Blueprint $table) {
13             $table->id();
14             $table->string('npm')->unique();
15             $table->string('nama');
16             $table->string('tempat_lahir');
17             $table->date('tanggal_lahir');
18             $table->enum('sex', ['L', 'P']);
19             $table->string('alamat');
20             $table->string('telp');
21             $table->string('email')->unique();
22             $table->string('photo')->nullable();
23             $table->timestamps();
24         });
25     }
26 }
27
28

```

- Persiapan Migrate Mahasiswa

Ketikan kode seperti dibawah ini:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api> php artisan migrate
INFO Preparing database.
Creating migration table ..... 135ms DONE
INFO Running migrations.
2019_12_14_000001_create_personal_access_tokens_table ..... 101ms DONE
2024_12_15_120207_create_mahasiswas_table ..... 91ms DONE
PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api>

```

- Pembuatan Mahasiswa Controller

Pada terminal VS Code ketikan kode seperti berikut ini:

```
php artisan make:controller MahasiswaController --api
```

```

● PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api> php artisan make:controller MahasiswaController --api
INFO Controller [D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api\app\Http\Controllers\MahasiswaController.php] created successfully.
○ PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api>

```

Buka File MahasiswaController Ketikan Kode seperti berikut ini:

```

<?php
namespace App\Http\Controllers;

use App\Models\Mahasiswa;
use Illuminate\Http\Request;

class MahasiswaController extends Controller
{
    public function index()
    {
        return Mahasiswa::all();
    }

    public function store(Request $request)
    {
        $request->validate([
            'npm' => 'required|string|unique:mahasiswa',
            'nama' => 'required|string',
            'tempat_lahir' => 'required|string',
            'tanggal_lahir' => 'required|date',
            'sex' => 'required|string',
            'alamat' => 'required|string',
            'telp' => 'required|string',
            'email' => 'required|string|email|unique:mahasiswa',
            'photo' => 'nullable|string',
        ]);

        return Mahasiswa::create($request->all());
    }
}

```

```

public function show($id)
{
    return Mahasiswa::findOrFail($id);
}

public function update(Request $request, $id)
{
    $mahasiswa = Mahasiswa::findOrFail($id);
    $request->validate([
        'npm' =>
'sometimes|required|string|unique:mahasiswa,npm,'.$mahasiswa->id,
        'email' =>
'sometimes|required|string|email|unique:mahasiswa,email,'.$mahasiswa->id,
    ]);

    $mahasiswa->update($request->all());
    return $mahasiswa;
}

public function destroy($id)
{
    $mahasiswa = Mahasiswa::findOrFail($id);
    $mahasiswa->delete();
    return response()->noContent();
}
}

```

- Pendaftaran Route API

Daftarkan route pada controller yang sudah dibuat di file api.php pada folder routes seperti kode dibawah ini:

```

$touch -> ...
api.php
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 use App\Http\Controllers\MahasiswaController;
7
8 Route::apiResource('mahasiswa', MahasiswaController::class);
9

```

Jalankan terminal VSCode dan ketikan ‘php artisan route:list’ untuk mengetahui alamat yang akan digunakan di aplikasi mobile

```

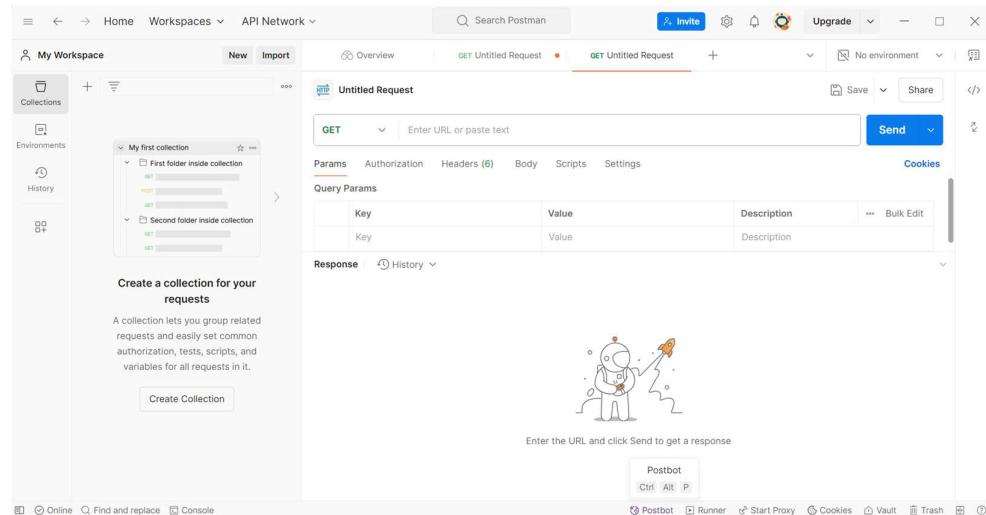
PS D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api> php artisan route:list
GET|HEAD /ignition/execute-solution
POST|HEAD /ignition/health-check
POST|HEAD /ignition/update-config
GET|HEAD api/mahasiswa
POST|HEAD api/mahasiswa
PUT|PATCH api/mahasiswa/{mahasiswa}
DELETE|HEAD api/mahasiswa/{mahasiswa}
GET|HEAD api/user
GET|HEAD sanctum/csrf-cookie

Showing [11] routes

```

1.3 PENGUJIAN ENDPOINT DAN INSTALASI POSTMAN

- Silahkan Download dan install Postman Dekstop Application <https://www.postman.com/downloads/> atau menggunakan versi online
- Silahkan login dengan akun google atau sejenisnya
- Persiapkan workspace seperti gambar berikut:



- Jalankan Server dengan langsung menggunakan domain virtual host dari laragon (<http://mahasiswa-api.test>) atau ketikan pada terminal vs code ‘ php artisan serve dan copy-kan alamat domain yang disediakan ke web browser. Pada Praktikum ini menggunakan alamat domain virtual host
- Penggunaan method untuk proses CRUD dengan penjelasan seperti berikut:
GET <http://mahasiswa-api.test/mahasiswa> untuk mengambil semua data
POST <http://mahasiswa-api.test/mahasiswa> untuk menambah data baru
GET <http://mahasiswa-api.test/mahasiswa/{id}> untuk mendapatkan satu data
PUT <http://mahasiswa-api.test/mahasiswa/{id}> untuk memperbarui data
DELETE <http://mahasiswa-api.test/mahasiswa/{id}> untuk menghapus data

- Pengujian Simpan Data

Jalankan aplikasi Postman , klik tombol New -> HTTP -> Copy Pastekan URL -> pilih method sebagai POST -> Pilih Body -> Pilih Raw -> Pilih JSON , Kemudian Ketikan Kode seperti berikut jika sudah ketikan tombol Send:

The screenshot shows the Postman interface with the following details:

- Method:** POST
- URL:** http://mahasiswa-api.test/api/mahasiswa
- Body:** Raw JSON (selected)
- JSON Payload:**

```
1 {
2   "npm": "16630199",
3   "nama": "Anita Isnaini",
4   "tempat_lahir": "Jakarta",
5   "tanggal_lahir": "1998-01-01",
6   "sex": "Perempuan",
7   "alamat": "Jl. Contoh No. 1",
8   "telp": "08123456789",
9   "email": "anita@gmail.com",
10  "photo": null
11 }
```

- Response:** 201 Created | 381 ms | 670 B |
- Test Results:** A small icon of a person with arms raised.
- Table View:** Below the results, there is a table showing the created data row:

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	id	npm	nama	tempat_lahir	tanggal_lahir	sex	alamat	telp	email	photo	created_at	updated_at
<input type="checkbox"/>				1	16630199	Anita Isnaini	Jakarta	1998-01-01	Perempuan	Jl. Contoh No. 1	08123456789	anita@gmail.com	NULL	2024-12-17 08:55:19	2024-12-17 08:55:19

- Pengujian Ubah Data

Jalankan aplikasi Postman , klik tombol New -> HTTP -> Copy Pastekan URL -> pilih method sebagai PUT -> Pilih Body -> Pilih Raw -> Pilih JSON , Kemudian Ketikan Kode seperti berikut jika sudah ketikan tombol Send:

The screenshot shows the Postman interface with a PUT request to `http://mahasiswa-api.test/api/mahasiswa/1`. The Body tab is selected, showing the following JSON code:

```
1 {  
2   "tempat_lahir": "Banjarmasin",  
3   "tanggal_lahir": "1998-01-01",  
4   "sex": "Perempuan",  
5   "alamat": "Jl. S.Parmam",  
6   "email": "anita1998@gmail.com"  
7 }
```

The response shows a 200 OK status with a response time of 286 ms and a body size of 669 B. Below the response, a table displays the updated student data:

	id	npm	nama	tempat_lahir	tanggal_lahir	sex	alamat	telp	email	photo	created_at	updated_at
1	16630199	Anita Isnaini	Banjarmasin	1998-01-01	Perempuan	Jl. S.Parmam	08123456789	anita1998@gmail.com	NULL	2024-12-17 08:55:19	2024-12-17 09:06:11	

- Pengujian Pencarian Data

Jalankan aplikasi Postman , klik tombol New -> HTTP -> Copy Pastekan URL -> pilih method sebagai GET -> Pilih Body -> Pilih Raw -> Pilih JSON , Kemudian Ketikan Kode seperti berikut jika sudah ketikan tombol Send:

The screenshot shows the Postman interface with a GET request to `http://mahasiswa-api.test/api/mahasiswa/1`. The Body tab is selected, showing the following JSON response:

```
{"id":1,"npm":"16630199","nama":"Anita Isnaini","tempat_lahir":"Banjarmasin","tanggal_lahir":"1998-01-01","sex":"Perempuan","alamat":"Jl. S.Parmam","telp":"08123456789","email":"anita1998@gmail.com","photo":null,"created_at":"2024-12-17T08:55:19.000000Z","updated_at":"2024-12-17T09:06:11.000000Z"}
```

- **Pengujian Penghapusan Data**

Jalankan aplikasi Postman , klik tombol New -> HTTP -> Copy Pastekan URL -> pilih method sebagai DELETE -> Pilih Body -> Pilih Raw -> Pilih JSON , Kemudian Ketikan Kode seperti berikut jika sudah ketikan tombol Send:

The screenshot shows the Postman interface. At the top, the URL is set to `http://mahasiswa-api.test/api/mahasiswa/1`. Below the URL, the method is selected as `DELETE`. A large blue arrow points to the `Send` button on the right. The `Body` tab is selected, showing the raw JSON content `1`. Other tabs like `Params`, `Headers`, and `Settings` are visible. The response at the bottom shows a `204 No Content` status with a duration of `479 ms` and a size of `319 B`.

1.4 PEMBUATAN PROJECT AWAL FLUTTER

- Buat Project Flutter dengan nama appakademik
- Penambahan Depedensi

tambahkan `get` dan `http` pada `pubspec.yaml` seperti contoh dibawah ini:

```
28  # the latest version a
29  # versions available,
30  dependencies:
31    flutter:
32      |  sdk: flutter
33      |  get: ^4.6.5
34      |  http: ^0.13.3
35
```

Kemudian jalan ‘flutter pub get’ pada terminal

- Persiapan Model Mahasiswa

Buat folder dengan nama `models` didalam folder `lib`, kemudian buat file dengan nama `mahasiswa.dart` dan ketikan kode seperti berikut:

```
class Mahasiswa {
```

```
final int? id;
final String npm;
final String nama;
final String tempatLahir;
final String tanggalLahir;
final String sex;
final String alamat;
final String telp;
final String email;
final String? photo;

Mahasiswa({
    this.id,
    required this.npm,
    required this.nama,
    required this.tempatLahir,
    required this.tanggalLahir,
    required this.sex,
    required this.alamat,
    required this.telp,
    required this.email,
    this.photo,
});
}

factory Mahasiswa.fromJson(Map<String, dynamic> json) {
    return Mahasiswa(
        id: json['id'],
        npm: json['npm'],
        nama: json['nama'],
        tempatLahir: json['tempat_lahir'],
        tanggalLahir: json['tanggal_lahir'],
        sex: json['sex'],
        alamat: json['alamat'],
        telp: json['telp'],
        email: json['email'],
        photo: json['photo'],
    );
}

Map<String, dynamic> toJson() {
    return {
        'npm': npm,
        'nama': nama,
        'tempat_lahir': tempatLahir,
        'tanggal_lahir': tanggalLahir,
        'sex': sex,
        'alamat': alamat,
        'telp': telp,
    };
}
```

```
        'email': email,
        'photo': photo,
    );
}
}
```

- Persiapan Controller Mahasiswa

Buat folder controllers didalam folder lib kemudian buat file dengan nama mahasiswa_controller.dart dan ketikan kode seperti berikut:

```
import 'package:get/get.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
import '../models/mahasiswa.dart';

class MahasiswaController extends GetxController {
    var mahasiswaList = <Mahasiswa>[].obs;
    // IP dari laravel dijalankan pada flutter run
    //final String apiUrl = 'http://127.0.0.1:8000/api/mahasiswa';
    //final String apiUrl = 'http://mahasiswa-api.tes/api/mahasiswa'; '

    // untuk emulator android studio gunakan IP dibawah ini
    final String apiUrl ='http://10.0.2.2:8000/api/mahasiswa';

    @override
    void onInit() {
        super.onInit();
        tangkapDataMahasiswa(); // Memanggil fetch data saat controller
diinisialisasi
    }

    Future<void> tangkapDataMahasiswa() async {
        final response = await http.get(Uri.parse(apiUrl));

        if (response.statusCode == 200) {
            List<dynamic> data = json.decode(response.body);
            mahasiswaList.value = data.map((item) =>
Mahasiswa.fromJson(item)).toList();
        } else {
            Exception('Gagal Ambil Data mahasiswa');
        }
    }

    Future<void> addMahasiswa(Mahasiswa mahasiswa) async {
        final response = await http.post(Uri.parse(apiUrl),
```

```

        headers: {"Content-Type": "application/json"},
        body: json.encode(mahasiswa.toJson()));

    if (response.statusCode == 201) {
        mahasiswaList.add(Mahasiswa.fromJson(json.decode(response.body)));
    } else {
        throw Exception('Gagal Simpan Data mahasiswa');
    }
}

Future<void> updateMahasiswa(int id, Mahasiswa mahasiswa) async {
    final response = await http.put(Uri.parse('$apiUrl/$id'),
        headers: {"Content-Type": "application/json"},
        body: json.encode(mahasiswa.toJson()));

    if (response.statusCode == 200) {
        final index = mahasiswaList.indexWhere((m) => m.id == id);
        if (index != -1) {
            mahasiswaList[index] = Mahasiswa.fromJson(json.decode(response.body));
        }
    } else {
        throw Exception(' Gagal Ubah Data mahasiswa');
    }
}

Future<void> deleteMahasiswa(int id) async {
    final response = await http.delete(Uri.parse('$apiUrl/$id'));

    if (response.statusCode == 204) {
        mahasiswaList.removeWhere((m) => m.id == id);
    } else {
        throw Exception('Gagal Hapus Data mahasiswa');
    }
}
}

```

- Persiapan View Mahasiswa

Buat folder dengan nama views didalam folder lib, kemudian buat file dengan nama mahasiswa_view.dart dan ketikan kode seperti berikut:

```

import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/mahasiswa_controller.dart';

class MahasiswaView extends StatelessWidget {
    MahasiswaView({super.key});
    final MahasiswaController controller = Get.put(MahasiswaController());
}

```

```
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("Data Mahasiswa"),
        ),
        body: Column(
            children: [
                Expanded(
                    child: Obx(() {
                        // Memastikan list mahasiswa telah terisi
                        if (controller.mahasiswaList.isEmpty) {
                            return Center(child: Text("Tidak ada data mahasiswa"));
                        }

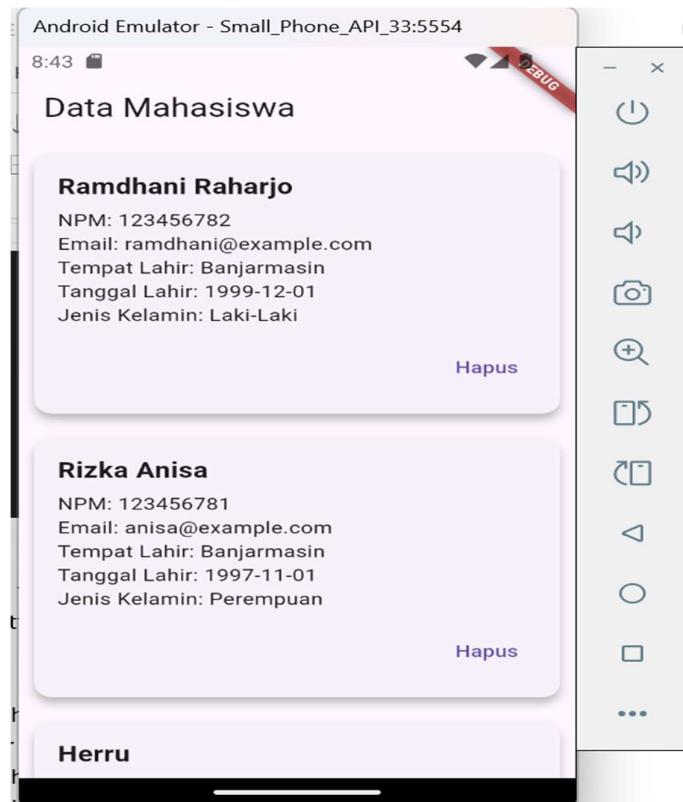
                        return ListView.builder(
                            itemCount: controller.mahasiswaList.length,
                            itemBuilder: (context, index) {
                                final mahasiswa = controller.mahasiswaList[index];
                                return Card(
                                    margin: EdgeInsets.all(10),
                                    elevation: 5,
                                    child: Padding(
                                        padding: const EdgeInsets.all(15.0),
                                        child: Column(
                                            crossAxisAlignment: CrossAxisAlignment.start,
                                            children: [
                                                Text(
                                                    mahasiswa.nama,
                                                    style: TextStyle(fontWeight: FontWeight.bold,
fontStyle: FontStyle.italic,
fontSize: 18),
                                                ),
                                                SizedBox(height: 5),
                                                Text("NPM: ${mahasiswa.npm}"),
                                                Text("Email: ${mahasiswa.email}"),
                                                Text("Tempat Lahir: ${mahasiswa.tempatLahir}"),
                                                Text("Tanggal Lahir: ${mahasiswa.tanggalLahir}"),
                                                Text("Jenis Kelamin: ${mahasiswa.sex == 'Laki-Laki' ? 'Laki-Laki' : 'Perempuan'}"),
                                                SizedBox(height: 10),
                                                OverflowBar(
                                                    alignment: MainAxisAlignment.end,
                                                    children: <Widget>[
                                                        TextButton(
                                                            child: Text('Hapus'),
                                                            onPressed: () {
                                                                controller.removeMahasiswa(index);
                                                            }
                                                        )
                                                    ],
                                                )
                                            ],
                                        ),
                                    ),
                                );
                            }
                        );
                    })
                )
            ],
        ),
    );
}
```

```
        controller.hapusMahasiswa(mahasiswa.id!);
    },
    ],
),
],
),
],
),
),
),
);
},
),
);
},
),
),
),
),
),
);
}
}
```

- Persiapan Emulator

Sebelum menjalankan emulator tambahkan 3 record terlebih dahulu melalui aplikasi postman, kemudian setting url pada mahasiswa_controller dengan ketentuan sebagai berikut:

- ➔ Dijalankan dengan Emulator Android Studio gunakan final String apiUrl = 'http://10.0.2.2:8000/api/mahasiswa';
 - ➔ Dijalankan dengan Flutter Run gunakan final String apiUrl = 'http://127.0.0.1:8000/api/mahasiswa'; atau final String apiUrl = 'http://mahasiswa-api.test/api/mahasiswa';



PERTEMUAN II

2.1 Kode Tambah Data Mahasiswa

Pada folder views buatlah file dengan nama tambah_mahasiswa.dart, file ini digunakan untuk membuat tampilan halaman input data mahasiswa pada saat user menekan tombol tambah mahasiswa, Silahkan ketikan ketikan kode seperti berikut:

```
        controller: emailController,
        decoration: InputDecoration(labelText: "Email"),
    ),
    TextField(
        controller: telpController,
        decoration: InputDecoration(labelText: "Telepon"),
    ),
    TextField(
        controller: alamatController,
        decoration: InputDecoration(labelText: "Alamat"),
    ),
    TextField(
        controller: tempatLahirController,
        decoration: InputDecoration(labelText: "Tempat Lahir"),
    ),
    TextField(
        controller: tanggalLahirController,
        decoration: InputDecoration(labelText: "Tanggal Lahir (YYYY-MM-DD)"),
    ),
    TextField(
        controller: sexController,
        decoration: InputDecoration(labelText: "Jenis Kelamin (L/P)"),
    ),
    SizedBox(height: 20),
    ElevatedButton(
        onPressed: () {
            // Menambahkan mahasiswa
            final mahasiswa = Mahasiswa(
                npm: npmController.text,
                nama: namaController.text,
                email: emailController.text,
                telp: telpController.text,
                tempatLahir: tempatLahirController.text,
                tanggalLahir: tanggalLahirController.text,
                sex: sexController.text,
                alamat: alamatController.text,
                photo: null, // Sesuaikan jika Anda memiliki URL untuk foto
            );
            controller.simpanMahasiswa(mahasiswa);
            Get.back(); // Kembali ke halaman mahasiswa
        },
        child: Text("Simpan"),
    ),
),
],
),
),
);
};
```

```
    }
}
```

2.2 Kode View Mahasiswa

Pada folder views buat file baru mahasiswa_view01.dart untuk membuat desain tampilan data mahasiswa yang terdapat tombol data, kemudian ketikan kode seperti berikut ini :

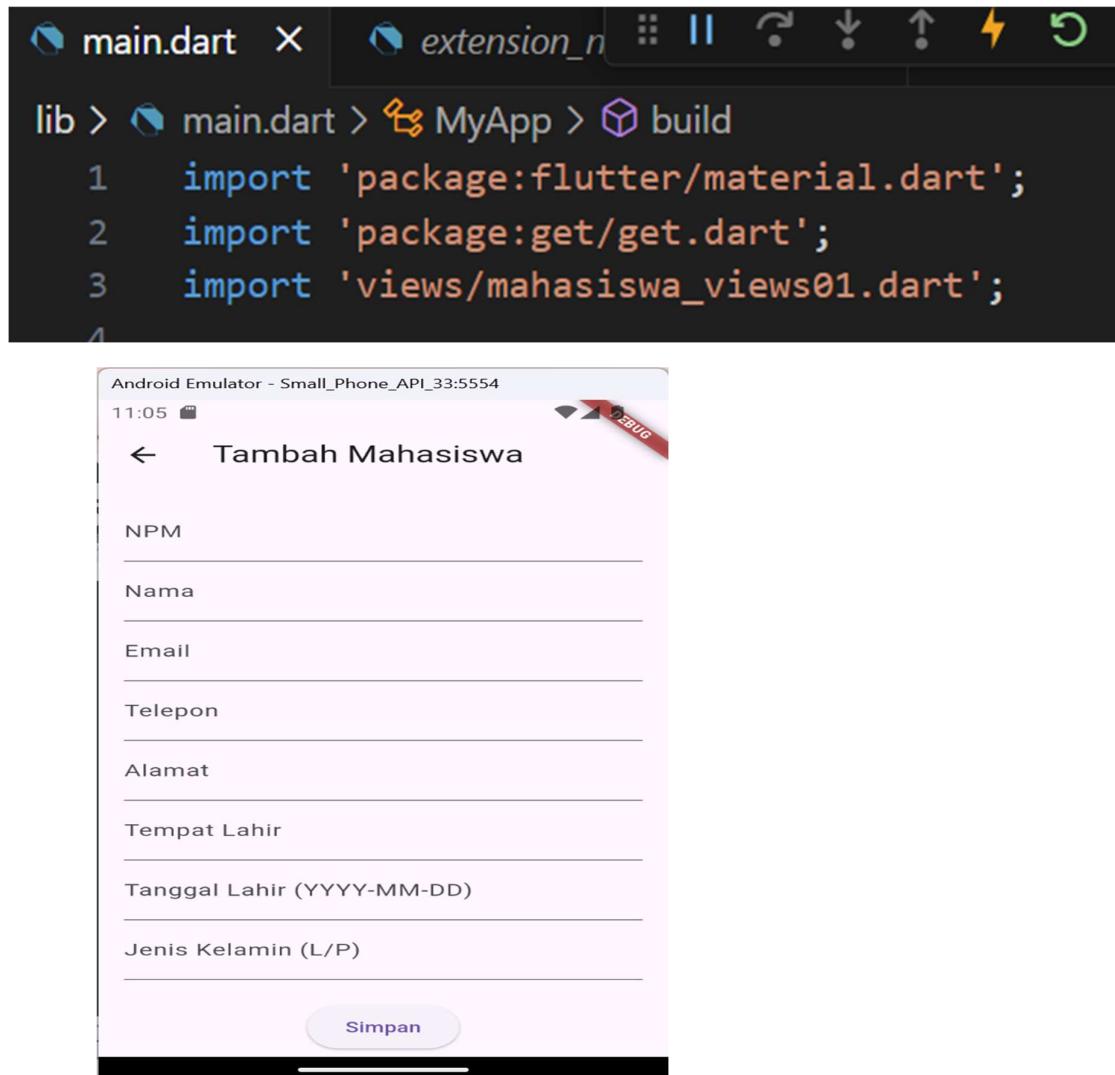
```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/mahasiswa_controller.dart';
//import '../models/mahasiswa.dart';
import 'tambah_mahasiswa.dart';
class MahasiswaView extends StatelessWidget {
    final MahasiswaController controller = Get.put(MahasiswaController());

    MahasiswaView({super.key});

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: Text("Data Mahasiswa"),
            ),
            body: Column(
                children: [
                    Expanded(
                        child: Obx(() {
                            if (controller.mahasiswaList.isEmpty) {
                                return Center(child: Text("Tidak ada data mahasiswa"));
                            }
                            return ListView.builder(
                                itemCount: controller.mahasiswaList.length,
                                itemBuilder: (context, index) {
                                    final mahasiswa = controller.mahasiswaList[index];
                                    return Card(
                                        margin: EdgeInsets.all(10),
                                        elevation: 5,
                                        child: Padding(
                                            padding: const EdgeInsets.all(15.0),
                                            child: Row(
                                                children: [
                                                    CircleAvatar(
                                                        radius: 30,
                                                        backgroundImage: NetworkImage(mahasiswa.photo ??
'https://w7.pngwing.com/pngs/358/81/png-transparent-computer-icons-graduation-
```

```
ceremony-college-student-financial-aid-graduated-people-logo-graduation-  
ceremony.png'), // Ganti dengan URL foto  
        ),  
        SizedBox(width: 10),  
        Expanded(  
            child: Column(  
                mainAxisAlignment: MainAxisAlignment.start,  
                children: [  
                    Text(  
                        mahasiswa.nama,  
                        style: TextStyle(fontWeight:  
FontWeight.bold, fontSize: 18),  
                    ),  
                    Text("NPM: ${mahasiswa.npm}"),  
                    Text("Email: ${mahasiswa.email}"),  
                ],  
            ),  
        ),  
        IconButton(  
            icon: Icon(Icons.delete),  
            onPressed: () {  
                controller.hapusMahasiswa(mahasiswa.id!);  
            },  
        ),  
    ],  
),  
),  
);  
},  
);  
},  
ElevatedButton(  
    onPressed: () {  
        Get.to(() => TambahMahasiswaView()); // Navigasi ke halaman  
tambah mahasiswa  
    },  
    child: Text("Tambah Mahasiswa"),  
),  
],  
),  
);  
}  
}
```

Kemudian pada file main.dart ubah struktur kode import library seperti dibawah ini:



2.3 Kode Ubah Data Mahasiswa

- Buat file dengan nama ubah_mahasiswa.dart pada folder views kemudian ketikan kode seperti berikut:

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import '../controllers/mahasiswa_controller.dart';
import '../models/mahasiswa.dart';

class EditMahasiswaView extends StatefulWidget {
```

```
final Mahasiswa mahasiswa;

EditMahasiswaView({required this.mahasiswa});

@Override
>EditMahasiswaViewState createState() => _EditMahasiswaViewState();
}

class _EditMahasiswaViewState extends State<EditMahasiswaView> {
final MahasiswaController controller = Get.find();

// Kontroler untuk input data
final TextEditingController npmController = TextEditingController();
final TextEditingController namaController = TextEditingController();
final TextEditingController emailController = TextEditingController();
final TextEditingController telpController = TextEditingController();
final TextEditingController alamatController = TextEditingController();
final TextEditingController tempatLahirController = TextEditingController();
final TextEditingController tanggalLahirController =
TextEditingController();
final TextEditingController sexController = TextEditingController();

@Override
void initState() {
super.initState();
// Inisialisasi kontroler dengan data mahasiswa yang ada
npmController.text = widget.mahasiswa.npm;
namaController.text = widget.mahasiswa.nama;
emailController.text = widget.mahasiswa.email;
telpController.text = widget.mahasiswa.telp;
alamatController.text = widget.mahasiswa.alamat;
tempatLahirController.text = widget.mahasiswa.tempatLahir;
tanggalLahirController.text = widget.mahasiswa.tanggalLahir;
sexController.text = widget.mahasiswa.sex;
}

@Override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text("Edit Mahasiswa"),
),
body: Padding(
padding: const EdgeInsets.all(16.0),
child: SingleChildScrollView(
child: Column(
children: [
TextField(

```

```
        controller: npmController,
        decoration: InputDecoration(labelText: "NPM"),
    ),
    TextField(
        controller: namaController,
        decoration: InputDecoration(labelText: "Nama"),
    ),
    TextField(
        controller: emailController,
        decoration: InputDecoration(labelText: "Email"),
    ),
    TextField(
        controller: telpController,
        decoration: InputDecoration(labelText: "Telepon"),
    ),
    TextField(
        controller: alamatController, // Inputan untuk alamat
        decoration: InputDecoration(labelText: "Alamat"),
    ),
    TextField(
        controller: tempatLahirController,
        decoration: InputDecoration(labelText: "Tempat Lahir"),
    ),
    TextField(
        controller: tanggalLahirController,
        decoration: InputDecoration(labelText: "Tanggal Lahir (YYYY-MM-DD"),
    ),
    TextField(
        controller: sexController,
        decoration: InputDecoration(labelText: "Jenis Kelamin"),
    ),
    SizedBox(height: 20),
    ElevatedButton(
        onPressed: () {
            // Membuat objek mahasiswa yang diperbarui tanpa foto
            final updatedMahasiswa = Mahasiswa(
                id: widget.mahasiswa.id,
                npm: npmController.text,
                nama: namaController.text,
                email: emailController.text,
                telp: telpController.text,
                alamat: alamatController.text,
                tempatLahir: tempatLahirController.text,
                tanggalLahir: tanggalLahirController.text,
                sex: sexController.text,
                photo: widget.mahasiswa.photo,
                // Pertahankan foto yang sama
            );
        }
    )
}
```

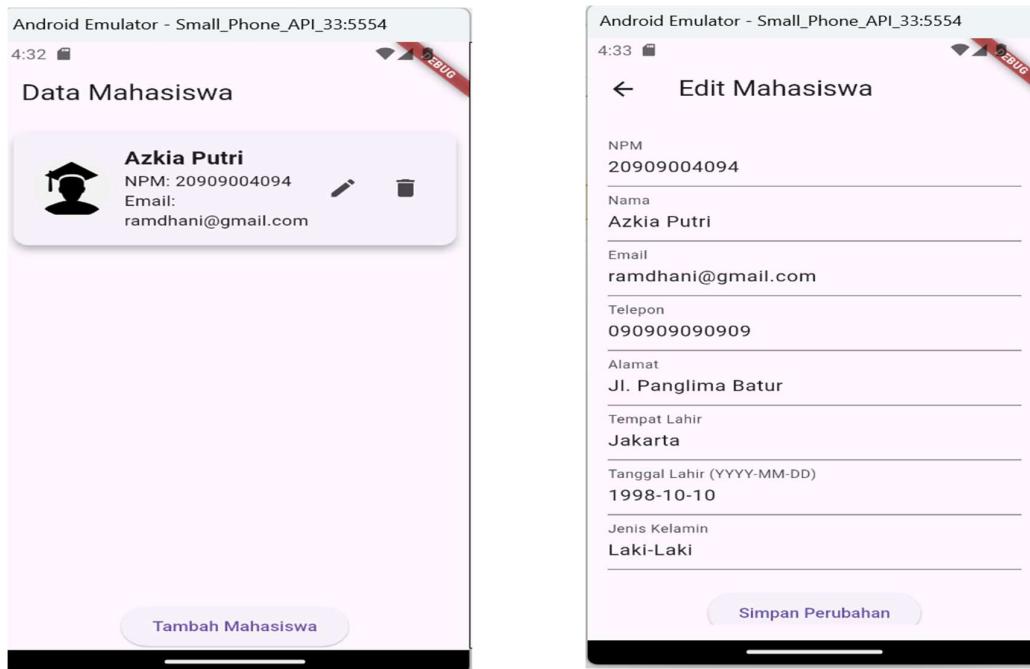
```
);

// Memanggil metode update di controller
controller.ubahMahasiswa(widget.mahasiswa.id!,updatedMahasiswa);
Get.back(); // Kembali ke halaman sebelumnya
},
child: Text("Simpan Perubahan"),
),
],
),
),
),
),
),
);
}
}
```

- Buka File mahasiswa_view01.dart kemudian tambahkan Icon Ubah yang letaknya di atas Tombol Icon Hapus seperti kode berikut ini:

```
52 ),
53 ),
54 ),
55 ),
56 IconButton(
57   icon: Icon(Icons.edit),
58   onPressed: () {
59     Get.to(() => EditMahasiswaView(mahasiswa: mahasiswa));
60   },
61 ),
62 IconButton(
63   icon: Icon(Icons.delete),
64   onPressed: () {
65     controller.hapusMahasiswa(mahasiswa.id!);
66   },
67 ),
```

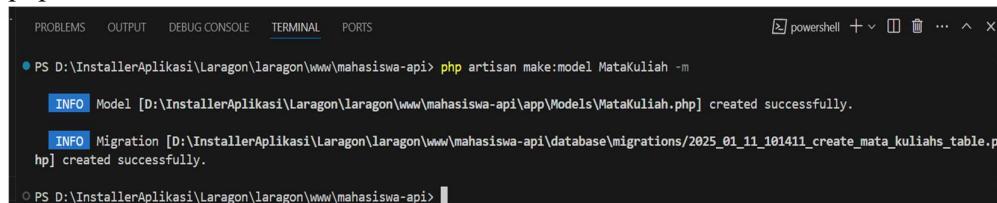
- Jalankan Emulator dan lakukan perubahan



PERTEMUAN III

3.1 Persiapan Model ,Controller, Route dan Migration Matakuliah

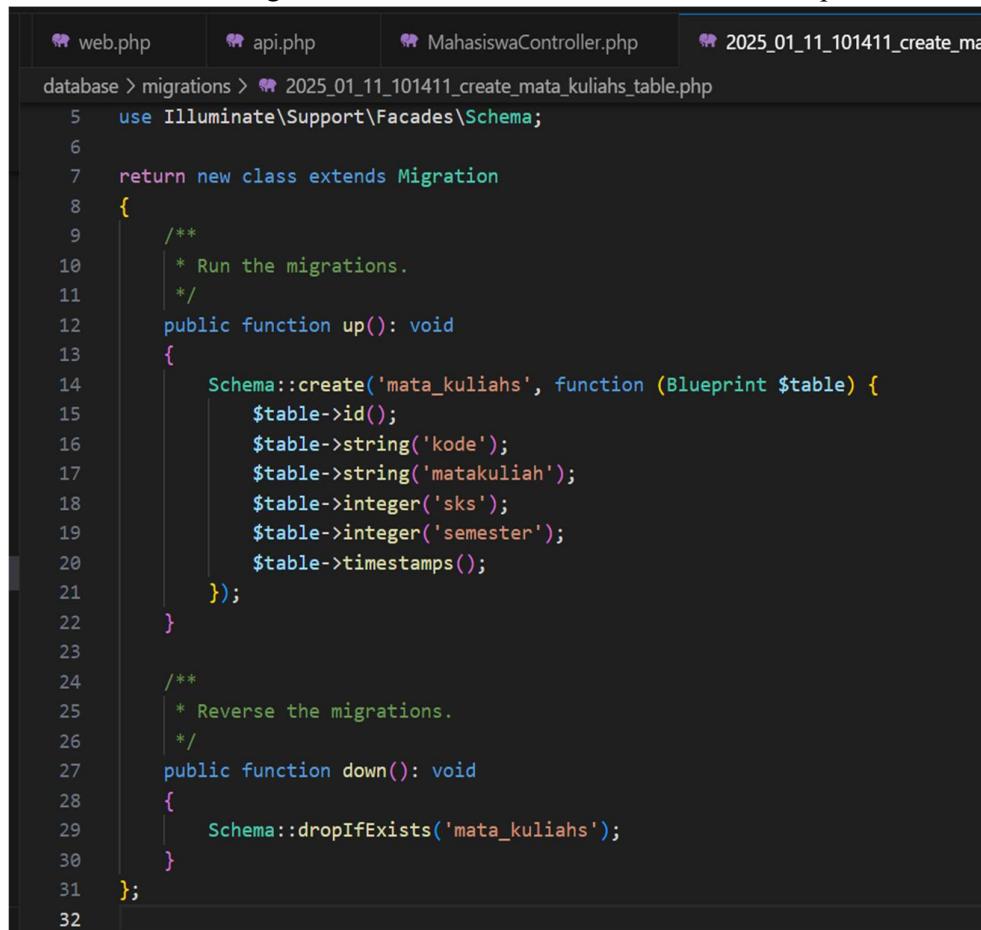
- Ketikan Terminal pada VS Code project Mahasiswa-api seperti berikut :
php artisan make:model MataKuliah -m



```
PS D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api> php artisan make:model MataKuliah -m
INFO Model [D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api\app\Models\MataKuliah.php] created successfully.
INFO Migration [D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api\database\migrations\2025_01_11_101411_create_mata_kuliah_table.php] created successfully.

PS D:\InstallerAplikasi\laragon\laragon\www\mahasiswa-api>
```

- Silahkan buka file migration mata kuliah kemudian ketikan kode seperti berikut:



```
use Illuminate\Support\Facades\Schema;
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('mata_kuliah', function (Blueprint $table) {
            $table->id();
            $table->string('kode');
            $table->string('matakuliah');
            $table->integer('sks');
            $table->integer('semester');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('mata_kuliah');
    }
};
```

Kemudian ketikan kode pada terminal :

php artisan migrate

The screenshot shows a terminal window in VS Code with the following output:

```
PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api> php artisan migrate
INFO Running migrations.
2025_01_11_101411_create_mata_kuliah_table ..... 441ms DONE
PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api>
```

- Persiapan Controller Mata kuliah
Pada VS Code Terminal Ketikan
php artisan make:controller MataKuliahController --resource
- Pada bagian function index() ketikan kode seperti berikut:

```
public function index()
{
    return MataKuliah::all();
}
```

- Pada bagian function create() ketikan kode seperti berikut:
- ```
public function store(Request $request)
{
 $mataKuliah = MataKuliah::create($request->all());
 return response()->json($mataKuliah, 201);
}
```

- Pada bagian function show() ketikan kode seperti berikut:
- ```
public function show($id)
{
    return MataKuliah::find($id);
}
```

- Pada bagian function update() ketikan kode seperti berikut:

```
public function update(Request $request, $id)
{
    $mataKuliah = MataKuliah::find($id);
    $mataKuliah->update($request->all());
    return response()->json($mataKuliah, 200);
}
```

- Pada bagian function destroy() ketikan kode seperti berikut:

```
public function destroy($id)
{
    MataKuliah::destroy($id);
    return response()->json(null, 204);
}
```

```
}
```

- Pada bagian use library Controller Mata Kuliah tambahkan kode seperti berikut:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\MataKuliah;
7
```

- Pada model Mata Kuliah ketikan kode seperti berikut ini:

The screenshot shows a code editor with tabs for Mahasiswa.php, MataKuliah.php (which is the active tab), web.php, and api.php. The code in MataKuliah.php is as follows:

```
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class MataKuliah extends Model
9 {
10     use HasFactory;
11     protected $table="matakuliahs";
12
13     protected $fillable = [
14         'kode',
15         'matakuliah',
16         'sks',
17         'semester'
18     ];
19 }
20
```

- Pada route api.php ketikan kode seperti berikut ini:

The screenshot shows a code editor with tabs for web.php, api.php (which is the active tab), and MahasiswaController.php. The code in api.php is as follows:

```
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 use App\Http\Controllers\MahasiswaController;
7 use App\Http\Controllers\MataKuliahController;
8 Route::apiResource('mahasiswa', MahasiswaController::class);
9 Route::apiResource('mata-kuliah', MataKuliahController::class);
10
```

Pada terminal ketikan kode seperti berikut:

Php artisan route:list untuk melihat route yang didaftarkan di API

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PUT | PATCH      api/mahasiswa/{mahasiswa} ..... mahasiswa.update > MahasiswaController@update
DELETE         api/mahasiswa/{mahasiswa} ..... mahasiswa.destroy > MahasiswaController@destroy
GET | HEAD    api/mata-kuliah ..... mata-kuliah.index > MataKuliahController@index
POST          api/mata-kuliah ..... mata-kuliah.store > MataKuliahController@store
GET | HEAD    api/mata-kuliah/{mata_kuliah} ..... mata-kuliah.show > MataKuliahController@show
PUT | PATCH    api/mata-kuliah/{mata_kuliah} ..... mata-kuliah.update > MataKuliahController@update
DELETE         api/mata-kuliah/{mata_kuliah} ..... mata-kuliah.destroy > MataKuliahController@destroy
GET | HEAD    api/user ..... 
GET | HEAD    sanctum/csrf-cookie ..... sanctum.csrf-cookie > Laravel\Sanctum > CsrfCookieController@show
  
```

Showing [17] routes

- Pengujian API untuk simpan data , silahkan buka postman dan ikuti seperti tampilan dibawah ini :

http://127.0.0.1:8000/api/mata-kuliah

POST http://127.0.0.1:8000/api/mata-kuliah

Params Authorization Headers (8) Body Scripts Settings Cookies

Key	Value	Description	Bulk Edit
kode	Text KM001		
matakuliah	Text Pemrograman Web II		
sks	Text 3		
semester	Text 3		

Body Cookies Headers (10) Test Results 201 Created 3.08 s 481 B

Pretty Raw Preview Visualize JSON

```

1 {
2   "kode": "KM001",
3   "matakuliah": "Pemrograman Web II",
4   "sks": "3",
5   "semester": "3",
6   "updated_at": "2025-01-11T23:28:33.000000Z",
7   "created_at": "2025-01-11T23:28:33.000000Z",
  
```

- Pengujian Tampil data, silahkan buka postman kemudian ikuti langkah seperti tampilan dibawah ini:

http://127.0.0.1:8000/api/mata-kuliah

GET http://127.0.0.1:8000/api/mata-kuliah

Params Authorization Headers (10) Body Scripts Settings Cookies

Key	Value	Description	Bulk Edit

Body Cookies Headers (10) Test Results 200 OK 3.79 s 474 B

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "id": 1,
4     "kode": "KM001",
5     "matakuliah": "Pemrograman Web II",
6     "sks": 3,
7     "semester": 3,
8     "created_at": "2025-01-11T23:28:33.000000Z",
9     "updated_at": "2025-01-11T23:28:33.000000Z"
10   }
11 ] 
```

Postbot

- Pengujian Ubah Data, silahkan buka postman kemudian ikut langkah seperti langkah dibawah ini:

The screenshot shows the Postman interface. At the top, there's a header bar with tabs for Overview, POST http://mahasiswa, GET http://mahasiswa, POST http://mahasiswa, and PUT http://127.0.0.1:8000/api/mata-kuliah/2. Below the header, the main area has a method dropdown set to PUT, a URL input field containing http://127.0.0.1:8000/api/mata-kuliah/2, and a large blue "Send" button. Underneath these, there are tabs for Params, Authorization, Headers (8), Body (selected), Scripts, and Settings. The Body tab shows a JSON editor with the following content:

```

1 {
2   "matakuliah": "Pemrograman Visual",
3   "skls": 3,
4   "semester": 5
5 }

```

Below the body editor, there are tabs for Cookies, Beautify, and a preview section showing the response. The response section displays a green "200 OK" status, a timestamp of 1.14 s, and a size of 472 B. The response body is also shown in JSON format:

```

1 {
2   ...
3   "id": 2,
4   "kode": "KM001",
5   "matakuliah": "Pemrograman Visual",

```

3.2 Persiapan Model, Controller dan View Mata Kuliah pada Flutter Mobile

- Buat file pada folder model untuk project flutter dengan nama mata_kuliah.dart kemudian ketikan kode seperti berikut:

```

lib> models > mata_kuliah.dart > MataKuliah
1 class MataKuliah {
2   final int id;
3   final String kode;
4   final String matakuliah;
5   final int sks;
6   final int semester;
7
8   MataKuliah({required this.id, required this.kode, required this.matakuliah, required this.sks, required this.semester});
9
10  factory MataKuliah.fromJson(Map<String, dynamic> json) {
11    return MataKuliah(
12      id: json['id'],
13      kode: json['kode'],
14      matakuliah: json['matakuliah'],
15      sks: json['sks'],
16      semester: json['semester'],
17    );
18  }
19
20  Map<String, dynamic> toJson() {
21    return {
22      'kode': kode,
23      'matakuliah': matakuliah,
24      'sks': sks,
25      'semester': semester,
26    };
27  }
28 }

```

- Buat file pada folder controller dengan nama matakuliah_controller.dart kemudian ketikan kode seperti berikut:

```

import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:praktikumakademik/models/mata_kuliah.dart';
import '../controllers/matakuliah_controller.dart';

class ApiService {
    // final String baseUrl = "http://localhost:8000/api/mata-kuliah";
    final String baseUrl = 'http://10.0.2.2:8000/api/mahasiswa';

    Future<List<MataKuliah>> fetchMataKuliah() async {
        final response = await http.get(Uri.parse(baseUrl));
        if (response.statusCode == 200) {
            List jsonResponse = json.decode(response.body);
            return jsonResponse.map((data) => MataKuliah.fromJson(data)).toList();
        } else {
            throw Exception('Gagal untuk load mata kuliah');
        }
    }

    Future<MataKuliah> createMataKuliah(MataKuliah matakuliah) async {
        final response = await http.post(
            Uri.parse(baseUrl),
            headers: <String, String>{
                'Content-Type': 'application/json; charset=UTF-8',
            },
            body: json.encode(matakuliah.toJson()),
        );

        if (response.statusCode == 201) {
            return MataKuliah.fromJson(json.decode(response.body));
        } else {
            throw Exception('Gagal Mengambil Data mata kuliah');
        }
    }

    Future<MataKuliah> updateMataKuliah(int id, MataKuliah matakuliah) async {
        final response = await http.put(
            Uri.parse('$baseUrl/$id'),
            headers: <String, String>{
                'Content-Type': 'application/json; charset=UTF-8',
            },
            body: json.encode(matakuliah.toJson()),
        );

        if (response.statusCode == 200) {
            return MataKuliah.fromJson(json.decode(response.body));
        }
    }
}

```

```

        } else {
            throw Exception('Gagal untuk update mata kuliah');
        }
    }

Future<void> deleteMataKuliah(int id) async {
    final response = await http.delete(
        Uri.parse('$baseUrl/$id'),
        headers: <String, String>{
            'Content-Type': 'application/json; charset=UTF-8',
        },
    );

    if (response.statusCode != 204) {
        throw Exception('Gagal untuk hapus mata kuliah');
    }
}
}

```

- Buat file pada folder views dengan nama matakuliah_view.dart kemudian ketikan kode seperti berikut:

```

import 'package:flutter/material.dart';

import '../controllers/matakuliah_controller.dart';
import '../models/mata_kuliah.dart';

class MatakuliahView extends StatefulWidget {
    @override
    _MataKuliahListState createState() => _MataKuliahListState();
}

class _MataKuliahListState extends State<MatakuliahView> {
    late Future<List<MataKuliah>> futureMataKuliah;

    @override
    void initState() {
        super.initState();
        futureMataKuliah = ApiService().fetchMataKuliah();
    }

    Future<void> _showForm(BuildContext context, {MataKuliah? mataKuliah}) {
        final kodeController = TextEditingController(text: mataKuliah?.kode);
        final matakuliahController = TextEditingController(text:
mataKuliah?.matakuliah);
        final sksController = TextEditingController(text: mataKuliah != null ?
mataKuliah.sks.toString() : '');
        final semesterController = TextEditingController(text: mataKuliah != null
? mataKuliah.semester.toString() : '');
    }
}

```

```
return showDialog(
    context: context,
    builder: (context) {
        return AlertDialog(
            title: Text(mataKuliah == null ? 'Tambah Mata Kuliah' : 'Ubah Mata
Kuliah'),
            content: Column(
                mainAxisAlignment: MainAxisAlignment.min,
                children: [
                    TextField(
                        controller: kodeController,
                        decoration: InputDecoration(labelText: 'Kode'),
                    ),
                    TextField(
                        controller: matakuliahController,
                        decoration: InputDecoration(labelText: 'Nama Mata Kuliah'),
                    ),
                    TextField(
                        controller: sksController,
                        decoration: InputDecoration(labelText: 'SKS'),
                        keyboardType: TextInputType.number,
                    ),
                    TextField(
                        controller: semesterController,
                        decoration: InputDecoration(labelText: 'Semester'),
                        keyboardType: TextInputType.number,
                    ),
                ],
            ),
            actions: [
                TextButton(
                    onPressed: () {
                        if (mataKuliah == null) {
                            // Tambah data
                            ApiService().createMataKuliah(MataKuliah(
                                id: 0, // sementara untuk ID, kita tidak akan
menggunakannya saat menambah
                                kode: kodeController.text,
                                matakuliah: matakuliahController.text,
                                sks: int.parse(sksController.text),
                                semester: int.parse(semesterController.text),
                            )).then((_) {
                                Navigator.of(context).pop();
                                setState(() {
                                    futureMataKuliah = ApiService().fetchMataKuliah();
                                });
                            });
                        }
                    }
                )
            ],
        );
    }
);
```

```
        } else {
            // Ubah data
            ApiService().updateMataKuliah(mataKuliah.id, MataKuliah(
                id: mataKuliah.id,
                kode: kodeController.text,
                matakuliah: matakuliahController.text,
                sks: int.parse(sksController.text),
                semester: int.parse(semesterController.text),
            )).then((_) {
                Navigator.of(context).pop();
                setState(() {
                    futureMataKuliah = ApiService().fetchMataKuliah();
                });
            });
        }
    },
    child: Text(mataKuliah == null ? 'Tambah' : 'Ubah'),
),
TextButton(
    onPressed: () {
        Navigator.of(context).pop();
    },
    child: Text('Batal'),
),
],
);
},
);
}
}

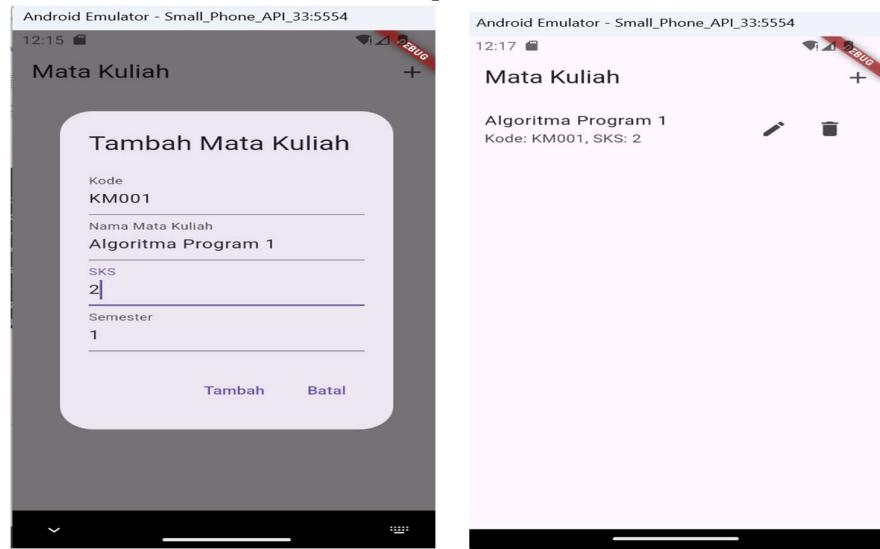
Future<void> _deleteMataKuliah(int id) async {
    await ApiService().deleteMataKuliah(id);
    setState(() {
        futureMataKuliah = ApiService().fetchMataKuliah(); // Refresh data
    });
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text('Mata Kuliah'),
            actions: [
                IconButton(
                    icon: Icon(Icons.add),
                    onPressed: () => _showForm(context),
                ),
            ],
        ),
    );
}
```

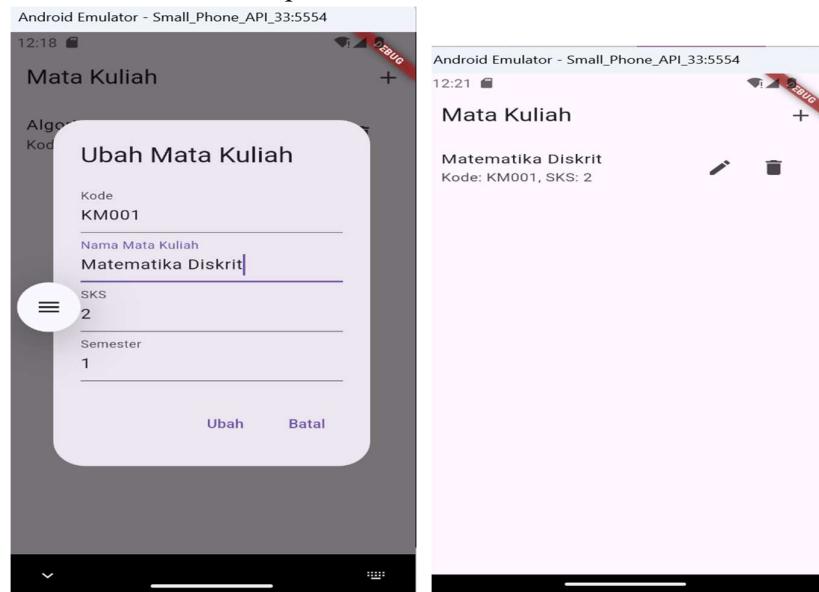
```
),
body: FutureBuilder<List<MataKuliah>>(
    future: futureMataKuliah,
    builder: (context, snapshot) {
        if (snapshot.connectionState == ConnectionState.waiting) {
            return Center(child: CircularProgressIndicator());
        } else if (snapshot.hasError) {
            return Center(child: Text('Error: ${snapshot.error}'));
        } else if (!snapshot.hasData || snapshot.data!.isEmpty) {
            return Center(child: Text('ID Mata Kuliah Kosong'));
        }
    }

    List<MataKuliah> mataKuliahs = snapshot.data!;
    return ListView.builder(
        itemCount: mataKuliahs.length,
        itemBuilder: (context, index) {
            return ListTile(
                title: Text(mataKuliahs[index].matakuliah),
                subtitle: Text('Kode: ${mataKuliahs[index].kode}, SKS: ${mataKuliahs[index].sks}'),
                trailing: Row(
                    mainAxisAlignment: MainAxisAlignment.end,
                    mainAxisSize: MainAxisSize.min,
                    children: [
                        IconButton(
                            icon: Icon(Icons.edit),
                            onPressed: () => _showForm(context, mataKuliah: mataKuliahs[index]),
                        ),
                        IconButton(
                            icon: Icon(Icons.delete),
                            onPressed: () {
                                _deleteMataKuliah(mataKuliahs[index].id);
                            },
                        ),
                    ],
                ),
            );
        },
    );
}
```

- Pengujian Tampilan Tambah Data, silahkan klik icon tanda (+) pada pojok kanan atas kemudian isikan data seperti berikut:



- Pengujian Tampilan Ubah Data, Silakan klik Icon ubah (Pencil) pada baris data kemudian ubah data seperti berikut:



- Untuk Hapus Data silahkan klik icon tempat sampah saja

3.3 Report PDF data Mata Kuliah

- Pada file pubspec.yaml tambahkan library seperti dibawah ini:

```
dependencies:  
  flutter:  
    sdk: flutter  
  get: ^4.6.5  
  http: ^0.13.3  
  dio: ^5.0.0  
  pdf: ^3.6.3  
  printing: ^5.6.0
```

Pada terminal VS Code Ketikan Kode Seperti berikut:

```
flutter pub get
```

- Pada file matakuliah_view.dart tambahkan library seperti gambar dibawah ini:

```
import '../models/mata_kuliah.dart';  
import 'package:pdf/pdf.dart';  
import 'package:pdf/widgets.dart' as pw;  
import 'package:printing/printing.dart';
```

- Pada file matakuliah_view.dart di atas method _showform() ketikan kode seperti berikut:

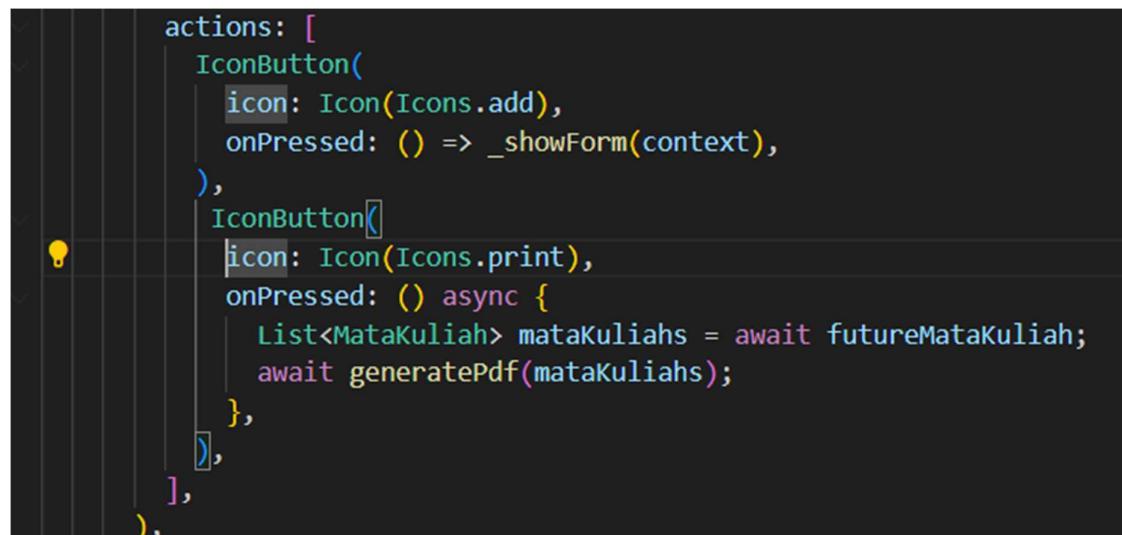
```
Future<void> generatePdf(List<MataKuliah> mataKuliahs) async {  
  final pdf = pw.Document();  
  
  pdf.addPage(  
    pw.Page(  
      build: (pw.Context context) {  
        return pw.Column(  
          children: [  
            pw.Text('Laporan Mata Kuliah', style: pw.TextStyle(fontSize:  
24)),  
            pw.SizedBox(height: 20),  
            pw.Table.fromTextArray(  
              headers: ['Kode', 'Mata Kuliah', 'SKS', 'Semester'],  
              data: mataKuliahs.map((mataKuliah) {  
                return [mataKuliah.kode, mataKuliah.matakuliah,  
mataKuliah.sks.toString(), mataKuliah.semester.toString()];  
              }).toList(),  
            ),  
          ],  
    ),  
  );  
  pdf.save();  
}
```

```
        );
    },
),
);

await Printing.layoutPdf(
    onLayout: (PdfPageFormat format) async => pdf.save(),
);
}
```

Agar library berhasil terinstall dengan baik untuk print pdf silahkan stop debugging dan jalan kan lagi

- Tambahkan Icon Cetak PDF disamping tanda icon (+) dibagian kode seperti tampilan dibawah ini:



```
actions: [
    IconButton(
        icon: Icon(Icons.add),
        onPressed: () => _showForm(context),
    ),
    IconButton(
        icon: Icon(Icons.print),
        onPressed: () async {
            List<MataKuliah> mataKuliahs = await futureMataKuliah;
            await generatePdf(mataKuliahs);
        },
    ),
],
),
```

- Silahkan isikan data mata kuliah minimal 5 baris untuk tes cetak pdf agar datanya lumayan banyak

The image consists of two side-by-side screenshots of an Android application interface.

Screenshot 1 (Left): The title bar says "Android Emulator - Small_Phone_API_33:5554". The time is 2:24. A red "DEBUG" banner is at the top right. The main content is titled "Mata Kuliah" and lists five courses:

- Matematika Diskrit
Kode: KM001, SKS: 2
- Pemrograman Visual 1
Kode: KM003, SKS: 3
- Pemrograman Web I
Kode: KM002, SKS: 3
- Multimedia
Kode: KM004, SKS: 2
- Pemrograman Mobile
Kode: KM005, SKS: 3

Each course entry has a pencil icon for edit and a trash bin icon for delete.

Screenshot 2 (Right): The title bar says "Android Emulator - Small_Phone_API_33:5554". The time is 2:25. The screen shows a "Save as PDF" dialog. It has "Copies: 1" and "Paper size: ISO A4". At the bottom, it displays a preview of the "Laporan Mata Kuliah" report, which contains a table of course details:

Kode	Mata Kuliah	SKS	Semester
KM001	Matematika Diskrit	2	1
KM003	Pemrograman Visual 1	3	1
KM002	Pemrograman Web I	3	3
KM004	Multimedia	2	6
KM005	Pemrograman Mobile	3	5

The report preview also shows "1/1" and a checkmark icon.

PERTEMUAN IV

4.1 Persiapan Menu Side atau Drawer Menu

- Buat file pada folder views dengan nama homescreen.dart kemudian ketikan kode seperti berikut:

```
import 'package:flutter/material.dart';
import 'mahasiswa_views01.dart';
import 'matakuliah_view.dart';

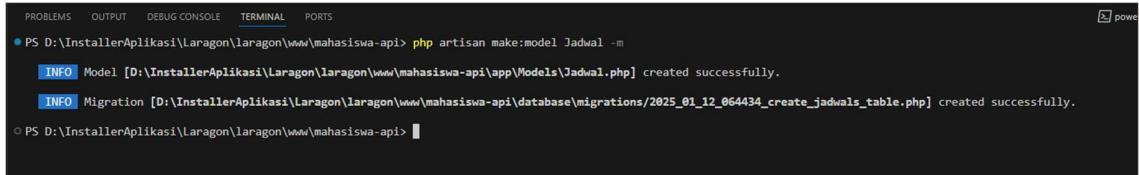
class HomeScreen extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: Text('Home'),
                backgroundColor: Colors.blueAccent,
            ),
            drawer: Drawer(
                child: ListView(
                    padding: EdgeInsets.zero,
                    children: <Widget>[
                        DrawerHeader(
                            child: Column(
                                children: [
                                    Image.network(
                                        'https://uniska-bjm.ac.id/wp-content/uploads/2021/08/logo-uniska-ok-300x300.png', // Ganti dengan URL gambar Anda
                                        fit: BoxFit.fill, // Memenuhi seluruh area
                                        width: 100, // Kanvas penuh
                                        height: 90, // Tinggi gambar
                                    ),
                                    SizedBox(height:3),
                                    Text(
                                        'Admin',
                                        style: TextStyle(
                                            color: Colors.white,
                                            fontSize: 20,
                                        ),
                                    ),
                                ],
                            ),
                        ),
                        decoration: BoxDecoration(
                            color: Colors.blue,
                        ),
                    ],
                ),
            ),
        );
    }
}
```

```
        ),
    ListTile(
        leading: Icon(Icons.emoji_people_rounded),
        title: Text('Mahasiswa'),
        onTap: () {
            // Handle home tap
            Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => MahasiswaView()),
            );
        },
    ),
    ListTile(
        leading: Icon(Icons.verified_user_sharp),
        title: Text('Mata Kuliah'),
        onTap: () {
            // Handle settings tap
            Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => MatakuliahView(),
            );
        },
    ),
    ListTile(
        leading: Icon(Icons.schedule),
        title: Text('Jadwal Kuliah'),
        onTap: () {
            // Handle about tap
            Navigator.pop(context);
        },
    ),
],
),
),
),
body: Center(
    child: Text(
        'PRAKTIKUM MOBILE SEMESTER 7/8',
        style: TextStyle(
            fontSize: 20,
        ),
    ),
),
);
}
}
```

4.2 Persiapan Web API Jadwal

- Ketikan pada terminal project Web API seperti dibawah ini:

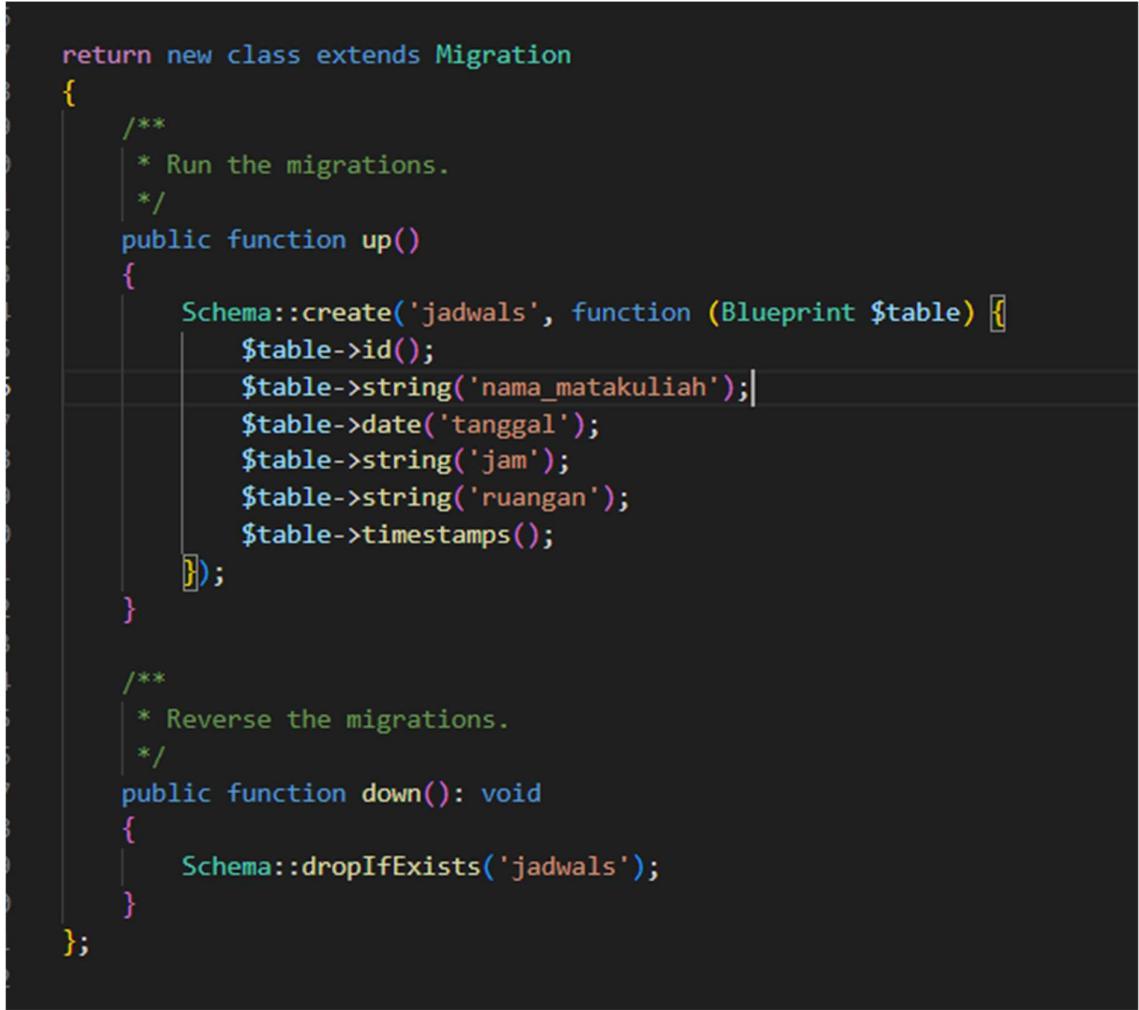
```
php artisan make:model Jadwal -m
```



```
PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api> php artisan make:model Jadwal -m
INFO Model [D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api\app\Models\Jadwal.php] created successfully.
INFO Migration [D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api\database\migrations\2025_01_12_064434_create_jadwals_table.php] created successfully.

PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api>
```

- Buka file migration pembuatan jadwal kemudian ketikan kode seperti berikut:



```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up()
    {
        Schema::create('jadwals', function (Blueprint $table) {
            $table->id();
            $table->string('nama_matakuliah');
            $table->date('tanggal');
            $table->string('jam');
            $table->string('ruangan');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('jadwals');
    }
};
```



```
PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api> php artisan migrate
INFO Running migrations.
2025_01_12_064434_create_jadwals_table ..... 436ms DONE

PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api>
```

- Buka file model jadwal.php kemudian tambahkan baris kode seperti dibawah ini:

```
app > Models > Jadwal.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Jadwal extends Model
9  {
10     use HasFactory;
11
12     protected $table="jadwals";
13
14     protected $fillable = [
15         'nama_matakuliah',
16         'tanggal',
17         'jam',
18         'ruangan'
19     ];
20 }
21
```

- Pembuatan Controller Jadwal kuliah, silahkan ketikan kode seperti berikut pada Terminal VS Code :
php artisan make:controller JadwalController

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
● PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api> php artisan make:controller JadwalController
[INFO] Controller [D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api\app\Http\Controllers\JadwalController.php] created successfully.
○ PS D:\InstallerAplikasi\Laragon\laragon\www\mahasiswa-api>
```

Kemudian buka file jadwal controller dan ketikan kode seperti dibawah ini:

```
<?php
namespace App\Http\Controllers;

use App\Models\Jadwal;
use Illuminate\Http\Request;

class JadwalController extends Controller
{
    public function index()
    {
        return Jadwal::all();
    }

    public function store(Request $request)
    {
        $validated = $request->validate([
            'nama_matakuliah' => 'required|string',
            'tanggal' => 'required|date',
            'jam' => 'required|string',
            'ruangan' => 'required|string',
        ]);

        $jadwal = Jadwal::create($validated);

        return response()->json($jadwal, 201);
    }

    public function show($id)
    {
        return Jadwal::find($id);
    }

    public function update(Request $request, $id)
    {
        $validated = $request->validate([
            'nama_matakuliah' => 'required|string',
            'tanggal' => 'required|date',
            'jam' => 'required|string',
            'ruangan' => 'required|string',
        ]);

        $jadwal = Jadwal::find($id);
        $jadwal->update($validated);

        return response()->json($jadwal, 200);
    }
}
```

```
public function destroy($id)
{
    Jadwal::destroy($id);
    return response()->json(null, 204);
}
```

- Buka File route api.php kemudian ketikan kode seperti berikut ini:

```
tes > 📄 api.php
1  <?php
2
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Route;
5
6  use App\Http\Controllers\MahasiswaController;
7  use App\Http\Controllers\MataKuliahController;
8  use App\Http\Controllers\JadwalController;
9  Route::apiResource('mahasiswa', MahasiswaController::class);
10 Route::apiResource('mata-kuliah', MataKuliahController::class);
11
12 Route::get('/jadwal', [JadwalController::class, 'index']);
13 Route::post('/jadwal', [JadwalController::class, 'store']);
14 Route::get('/jadwal/{id}', [JadwalController::class, 'show']);
15 Route::put('/jadwal/{id}', [JadwalController::class, 'update']);
16 Route::delete('/jadwal/{id}', [JadwalController::class, 'destroy']);
17
```

4.3 Persiapan Model, Controller dan View Pada Flutter

- Buat file pada folder model pada project flutter dengan nama jadwal.dart, kemudian ketikan kode seperti berikut:

```
class Jadwal {
    final int? id;
    final String namaMatakuliah;
    final String tanggal;
    final String jam;
    final String ruangan;

    Jadwal({this.id, required this.namaMatakuliah, required this.tanggal,
required this.jam, required this.ruangan});

    factory Jadwal.fromJson(Map<String, dynamic> json) {
        return Jadwal(
            id: json['id'],
```

```

        namaMatakuliah: json['nama_matakuliah'],
        tanggal: json['tanggal'],
        jam: json['jam'],
        ruangan: json['ruangan'],
    );
}

Map<String, dynamic> toJson() {
    return {
        'nama_matakuliah': namaMatakuliah,
        'tanggal': tanggal,
        'jam': jam,
        'ruangan': ruangan,
    };
}
}

```

- Buat file pada folder controller dengan nama jadwal_controller.dart kemudian ketikan kode seperti berikut ini:

```

import 'dart:convert';
import 'package:http/http.dart' as http;
import '../models/jadwal.dart';

class JadwalService {
    final String baseUrl ='http://10.0.2.2:8000/api/jadwal'; // Ganti dengan URL API Anda

    Future<List<Jadwal>> getJadwal() async {
        final response = await http.get(Uri.parse(baseUrl));
        if (response.statusCode == 200) {
            List jsonResponse = json.decode(response.body);
            return jsonResponse.map((data) => Jadwal.fromJson(data)).toList();
        } else {
            throw Exception('Failed to load data');
        }
    }

    Future<Jadwal> createJadwal(Jadwal jadwal) async {
        final response = await http.post(
            Uri.parse(baseUrl),
            headers: <String, String>{
                'Content-Type': 'application/json; charset=UTF-8',
            },
            body: json.encode(jadwal.toJson()),
        );

        if (response.statusCode == 201) {
            return Jadwal.fromJson(json.decode(response.body));
        }
    }
}

```

```

        } else {
            throw Exception('Failed to create jadwal');
        }
    }

Future<Jadwal> updateJadwal(int id, Jadwal jadwal) async {
    final response = await http.put(
        Uri.parse('$baseUrl/$id'),
        headers: <String, String>{
            'Content-Type': 'application/json; charset=UTF-8',
        },
        body: json.encode(jadwal.toJson()),
    );

    if (response.statusCode == 200) {
        return Jadwal.fromJson(json.decode(response.body));
    } else {
        throw Exception('Failed to update jadwal');
    }
}

Future<void> deleteJadwal(int id) async {
    final response = await http.delete(Uri.parse('$baseUrl/$id'));
    if (response.statusCode != 204) {
        throw Exception('Failed to delete jadwal');
    }
}
}

```

- Buat file pada folder view dengan nama jadwal_view.dart kemudian ketikan kode seperti berikut :

```

import 'package:flutter/material.dart';
import 'package:praktikumakademik/models/jadwal.dart';
import 'package:praktikumakademik/controllers/jadwal_controller.dart';

class JadwalPage extends StatefulWidget {
    @override
    _JadwalPageState createState() => _JadwalPageState();
}

class _JadwalPageState extends State<JadwalPage> {
    final JadwalService _service = JadwalService();
    List<Jadwal> _jadwallist = [];
    late Jadwal _selectedJadwal;

    @override

```

```
void initState() {
    super.initState();
    _fetchJadwal();
}

Future<void> _fetchJadwal() async {
    List<Jadwal> jadwalList = await _service.getJadwal();
    setState(() {
        _jadwalList = jadwalList;
    });
}

void _showForm(BuildContext context, {Jadwal? jadwal}) {
    final isEdit = jadwal != null;

    // Pindahkan deklarasi `jadwal` agar tidak digunakan sebelum dideklarasikan.
    final String? namaMatakuliah = isEdit ? jadwal!.namaMatakuliah : '';
    final String? tanggal = isEdit ? jadwal.tanggal : '';
    final String? jam = isEdit ? jadwal.jam : '';
    final String? ruangan = isEdit ? jadwal.ruangan : '';

    // Jika Anda menggunakan TextEditingController
    final namaMatakuliahController = TextEditingController(text:
namaMatakuliah);
    final tanggalController = TextEditingController(text: tanggal);
    final jamController = TextEditingController(text: jam);
    final ruanganController = TextEditingController(text: ruangan);

    // Rest of your modal implementation...
    showModalBottomSheet(
        context: context,
        builder: (context) {
            return Padding(
                padding: const EdgeInsets.all(16.0),
                child: Column(
                    mainAxisAlignment: MainAxisAlignment.min,
                    children: [
                        TextField(controller: namaMatakuliahController, decoration:
InputDecoration(labelText: 'Nama Mata Kuliah')),
                        TextField(controller: tanggalController, decoration:
InputDecoration(labelText: 'Tanggal (YYYY-MM-DD)')),
                        TextField(controller: jamController, decoration:
InputDecoration(labelText: 'Jam')),
                        TextField(controller: ruanganController, decoration:
InputDecoration(labelText: 'Ruangan')),
                        SizedBox(height: 20),
                        ElevatedButton(
                            child: Text(isEdit ? 'Update' : 'Simpan'),

```

```
        onPressed: () {
            final newJadwal = Jadwal(
                id: isEdit ? jadwal.id : null,
                namaMatakuliah: namaMatakuliahController.text,
                tanggal: tanggalController.text,
                jam: jamController.text,
                ruangan: ruanganController.text,
            );
            if (isEdit) {
                _service.updateJadwal(jadwal.id!, newJadwal).then((_) {
                    _fetchJadwal();
                    Navigator.pop(context);
                });
            } else {
                _service.createJadwal(newJadwal).then((_) {
                    _fetchJadwal();
                    Navigator.pop(context);
                });
            }
        },
    ],
),
);
},
);
);
}
}

void _deleteJadwal(int id) {
    _service.deleteJadwal(id).then((_) {
        _fetchJadwal();
    });
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: Text('Jadwal')),
        body: ListView.builder(
            itemCount: _jadwallist.length,
            itemBuilder: (context, index) {
                final jadwal = _jadwallist[index];
                return ListTile(
                    title: Text(jadwal.namaMatakuliah),
                    subtitle: Text('${jadwal.tanggal} - ${jadwal.jam}
(${jadwal.ruangan})'),
                    trailing: Row(

```

```

        mainAxisAlignment: MainAxisAlignment.min,
        children: [
            IconButton(
                icon: Icon(Icons.edit),
                onPressed: () => _showForm(context, jadwal: jadwal),
            ),
            IconButton(
                icon: Icon(Icons.delete),
                onPressed: () => _deleteJadwal(jadwal.id!),
            ),
        ],
    ),
    onTap: () => _showForm(context, jadwal: jadwal),
);
},
),
floatingActionButton: FloatingActionButton(
    child: Icon(Icons.add),
    onPressed: () => _showForm(context),
),
);
}
}
}

```

4.4 Persiapan Slide Menu

- Buat file pada folder view dengan nama homescreen.dart kemudian ketikan kode seperti berikut:

```

import 'package:flutter/material.dart';
import 'mahasiswa_views01.dart';
import 'matakuliah_view.dart';
import 'jadwal_view.dart';

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

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: Text('Home'),
                backgroundColor: Colors.blueAccent,
            ),
            drawer: Drawer(
                child: ListView(
                    padding: EdgeInsets.zero,

```

```
children: <Widget>[
    DrawerHeader(
        decoration: BoxDecoration(
            color: Colors.blue,
        ),
        child: Column(
            children: [
                Image.network(
                    'https://uniska-bjm.ac.id/wp-content/uploads/2021/08/logo-uniska-ok-300x300.png', // Ganti dengan URL gambar Anda
                    fit: BoxFit.fill, // Memenuhi seluruh area
                    width: 100, // Kanvas penuh
                    height: 90, // Tinggi gambar
                ),
                SizedBox(height:3),
                Text(
                    'Admin',
                    style: TextStyle(
                        color: Colors.white,
                        fontSize: 20,
                    ),
                ),
                ],
            ),
        ),
    ),
    ListTile(
        leading: Icon(Icons.emoji_people_rounded),
        title: Text('Mahasiswa'),
        onTap: () {
            // Handle home tap
            Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => MahasiswaView()),
            );
        },
    ),
    ListTile(
        leading: Icon(Icons.verified_user_sharp),
        title: Text('Mata Kuliah'),
        onTap: () {

            Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => MatakuliahView()),
            );
        },
    ),
],
```

```
        ListTile(
            leading: Icon(Icons.schedule),
            title: Text('Jadwal Kuliah'),
            onTap: () {
                Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => JadwalPage()),
                );
            },
        ),
    ],
),
),
),
),
),
body: Center(
    child: Text(
        'PRAKTIKUM MOBILE SEMESTER 7/8',
        style: TextStyle(
            fontSize: 20,
        ),
    ),
),
),
),
);
}
}
```

- Pada file main.dart ubah struktur kode seperti berikut:

```
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'views/homescreen.dart';

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

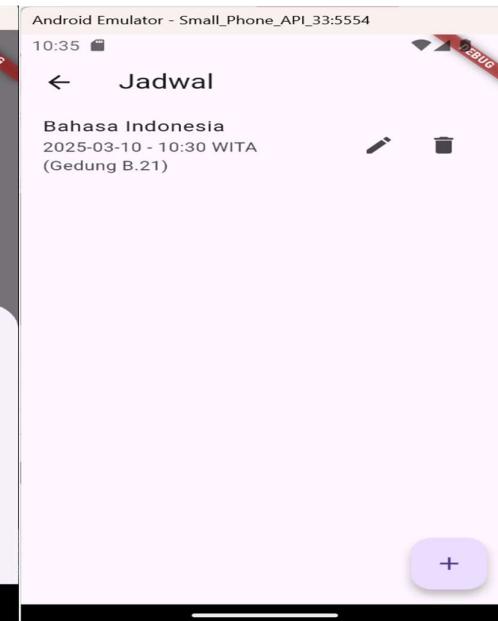
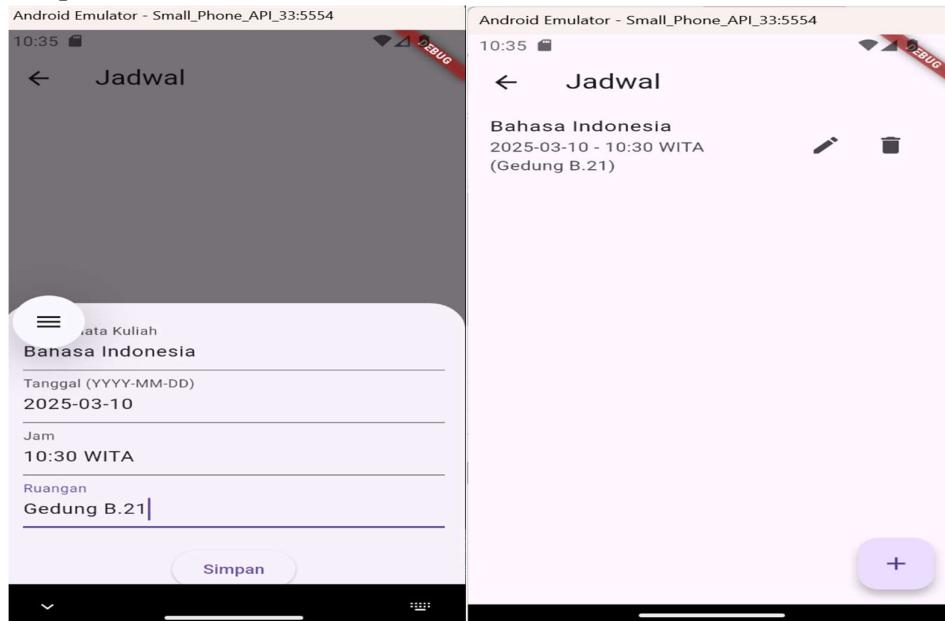
class MyApp extends StatelessWidget {
    const MyApp({super.key});

    @override
    Widget build(BuildContext context) {
        return GetMaterialApp(
            title: 'Mahasiswa App',
            theme: ThemeData(
                primarySwatch: Colors.blue,
            ),
            home: HomeScreen(),
        );
    }
}
```

```
}
```

- Jalankan Emulator dan lakukan transaksi Simpan,Ubah, dan Hapus Data

- **Simpan**



- **Ubah**

