PEMROGRAMAN BERBASIS MOBILE 2

PERTEMUAN 1



**DESKRIPSI MATAKULIAH**

**Mata Kuliah ini mempelajari konsep bahasa pemrograman Dart, membuat aplikasi mobile dengan flutter menggunakan bahasa Dart, menerapkan widget pada flutter untuk membuat aplikasi mobile**

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami penggunaan Framework Flutter dengan IDE Visual Studio Code untuk membuat aplikasi mobile.**

**DOSEN PENGAMPU**

**Bambang Ismanto, M.Kom**

**INSTITUT WIDYA PRATAMA**

[Pokok bahasan/bab/pertemuan]

**CAPAIAN PEMBELAJARAN**

**Mahasiswa dapat memahami penggunaan Framework Flutter dengan IDE Visual Studio Code untuk membuat aplikasi mobile.**

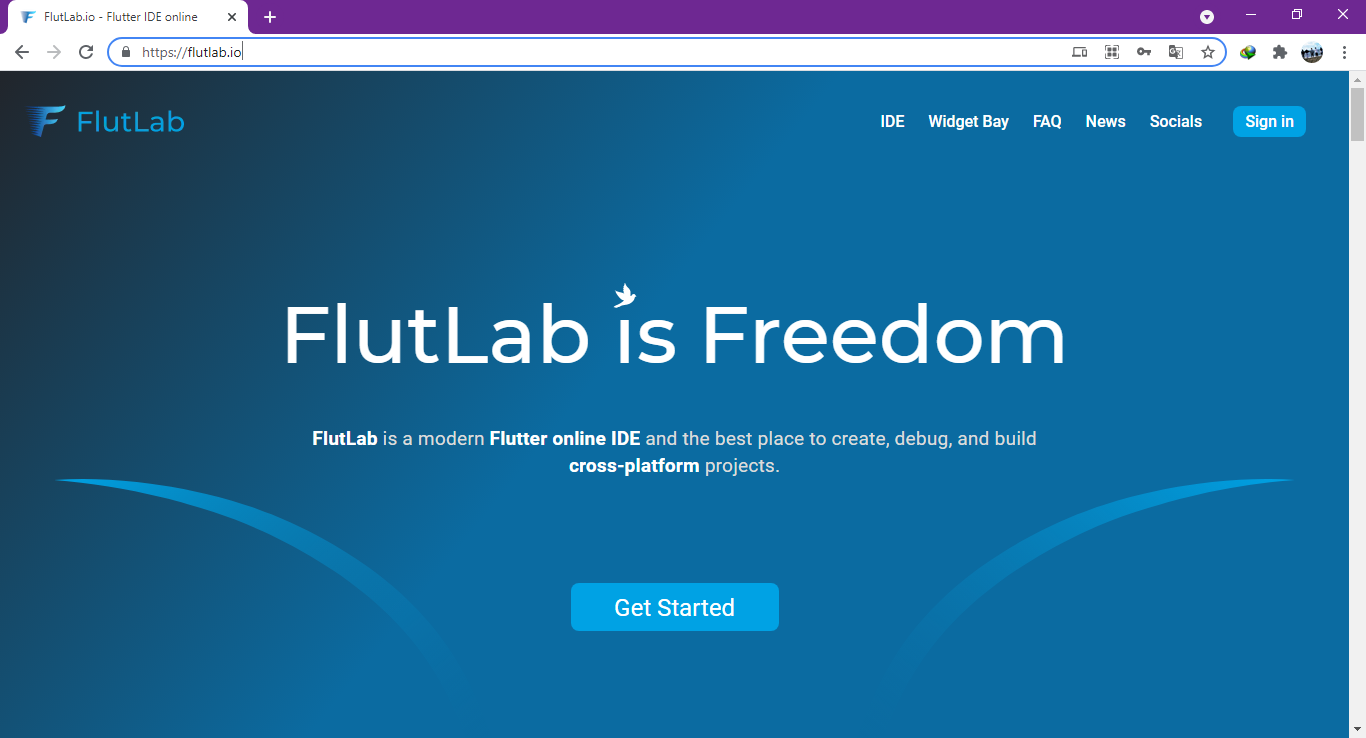
**MATERI PEMBELAJARAN**

1. **Membuat Projek Baru**
2. **Menyiapkan Library**
3. **Menambahkan Permission**
4. **Setting minSDKVersion**
5. **Menyiapkan Halaman Home**
6. **Menyiapkan File main.dart**
7. **Build Aplikasi**

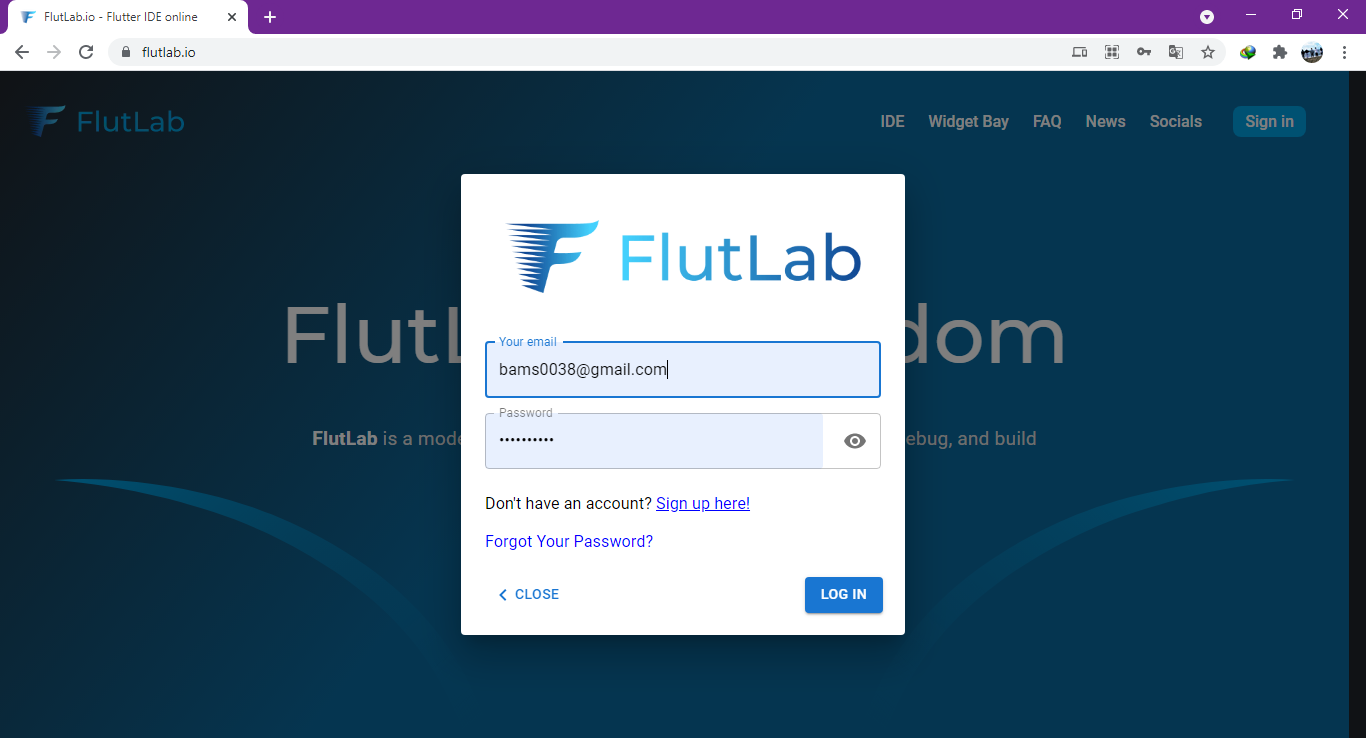
**MATERI**

1. **Membuat Project Baru**

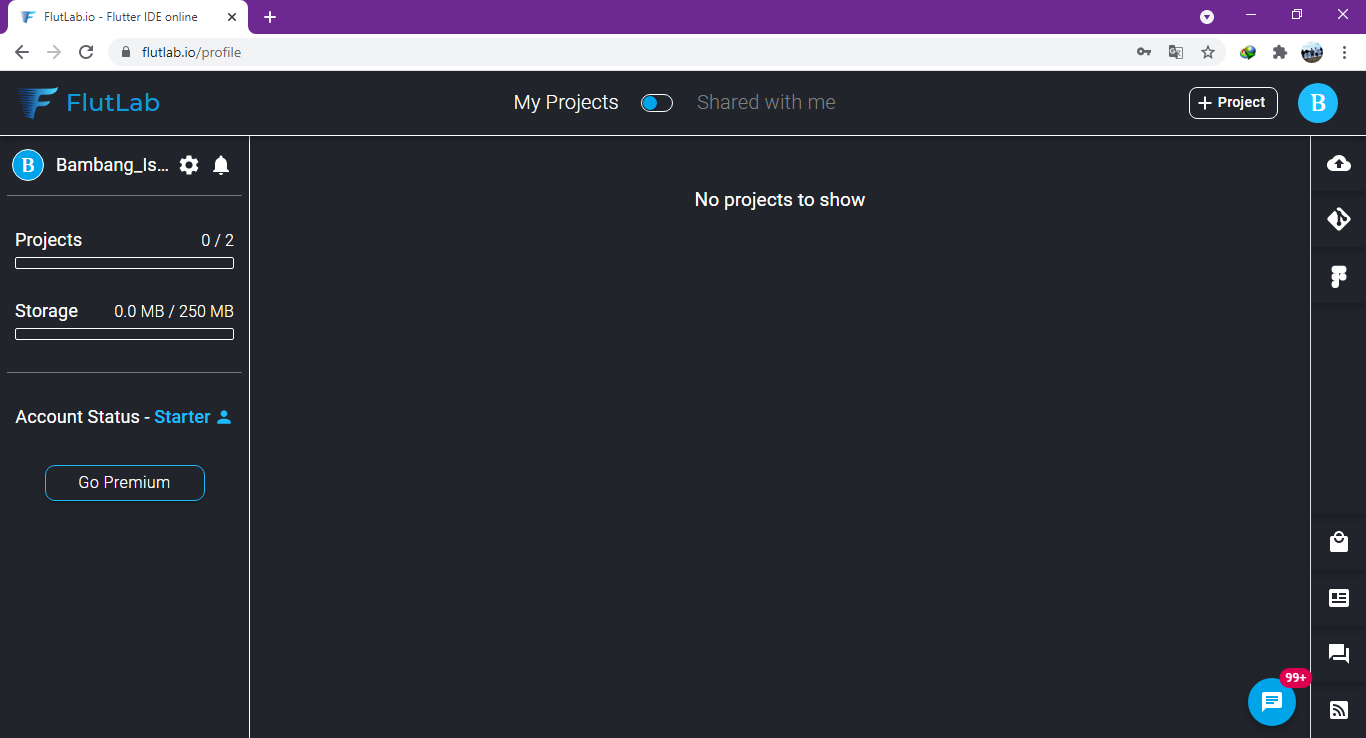
Pengembangan aplikasi mobile menggunakan framework flutter dengan aplikasi VSCode dan Android Studio membutuhkan spesifikasi komputer yang tinggi. Oleh karena itu, untuk mempermudah praktikum, kita bisa menggunakan flutlab.io. silahkan masuk ke website **https://flutlab.io/**.



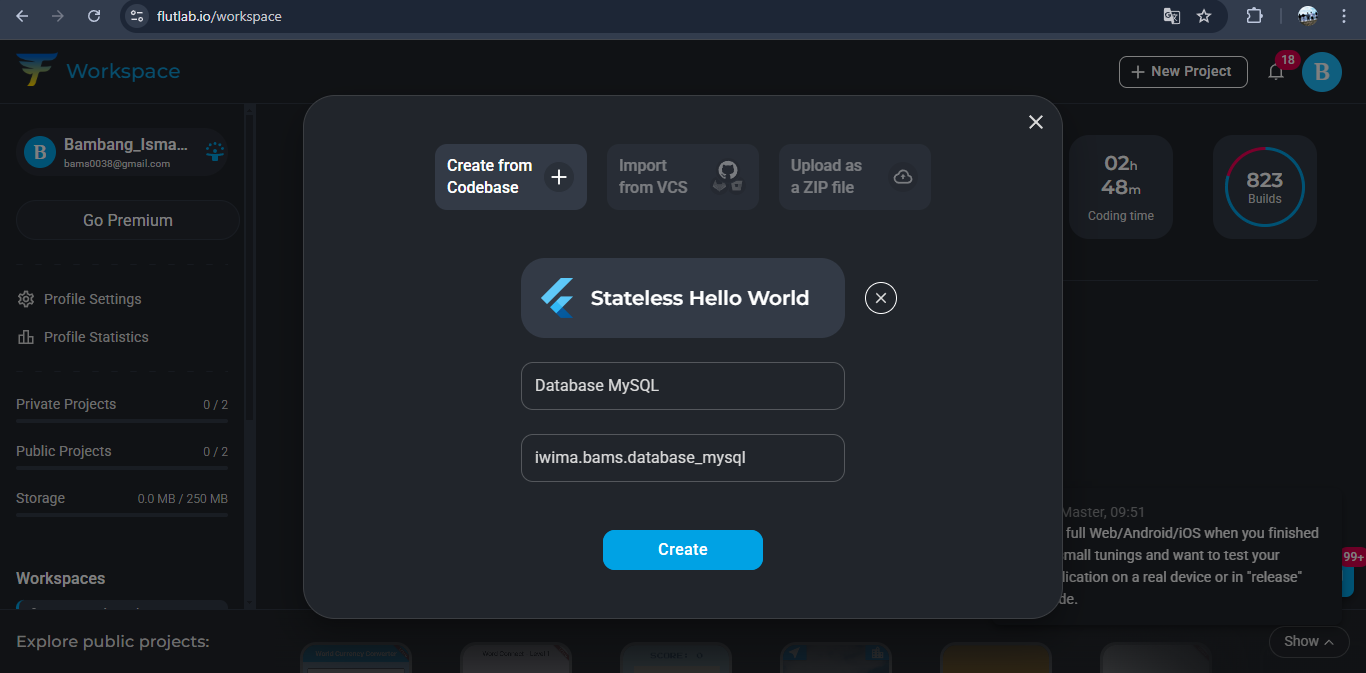
Kemudian sign in ke website, jika belum punya akun, silahkan daftar dulu.



Berikut adalah tampilan setelah login.



Buat projek baru dengan klik pada button **+ Project**. Beri nama projek dan nama package. Untuk **nama project disesuaikan**.



Setelah seleai, klik pada nama projek untuk membukanya.

1. **Menyiapkan Library**

Dalam projek ini kita akan menggunakan dua library, yaitu http untuk membuat koneksi dengan http.

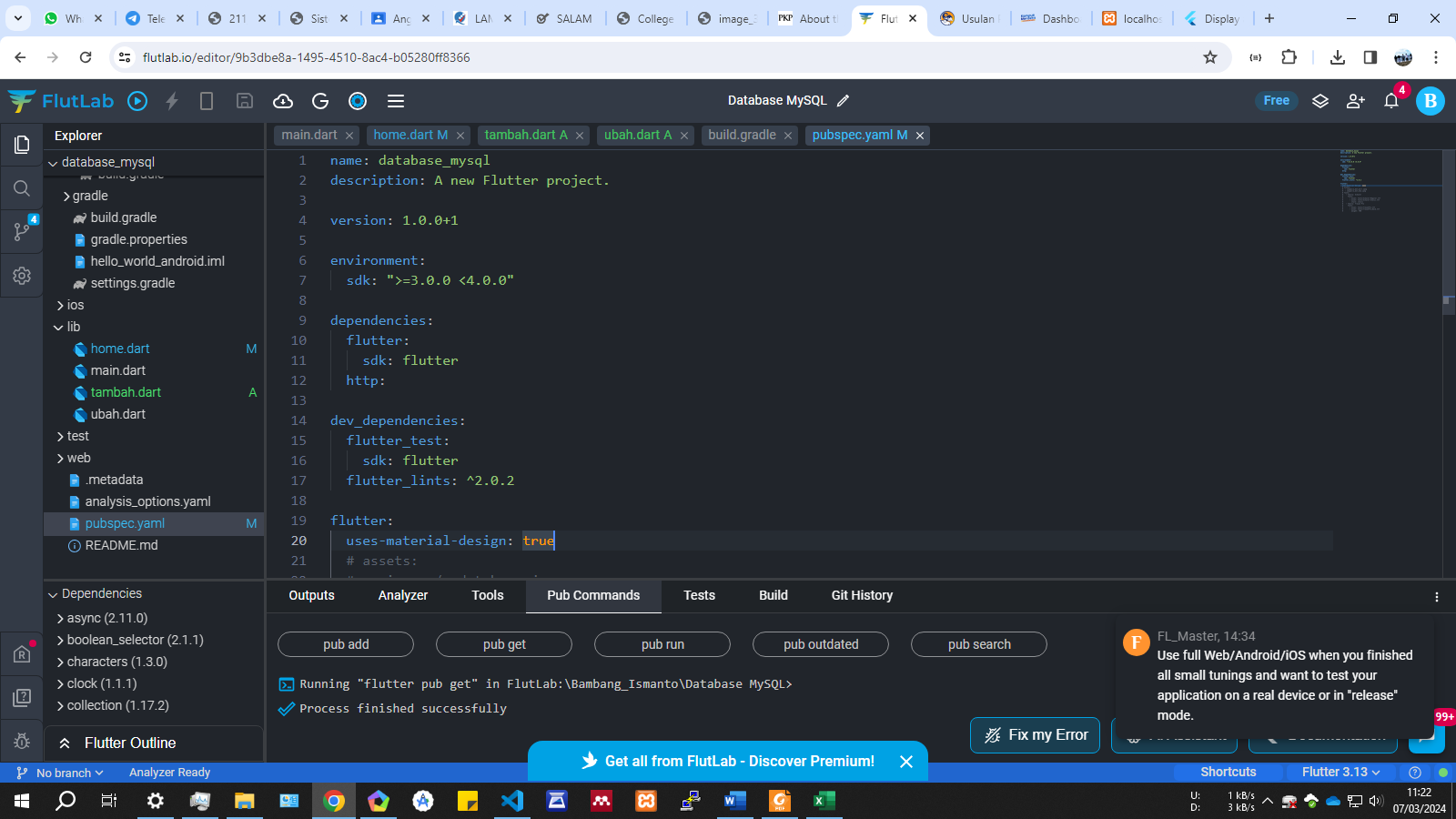
dependencies:

  flutter:

    sdk: flutter

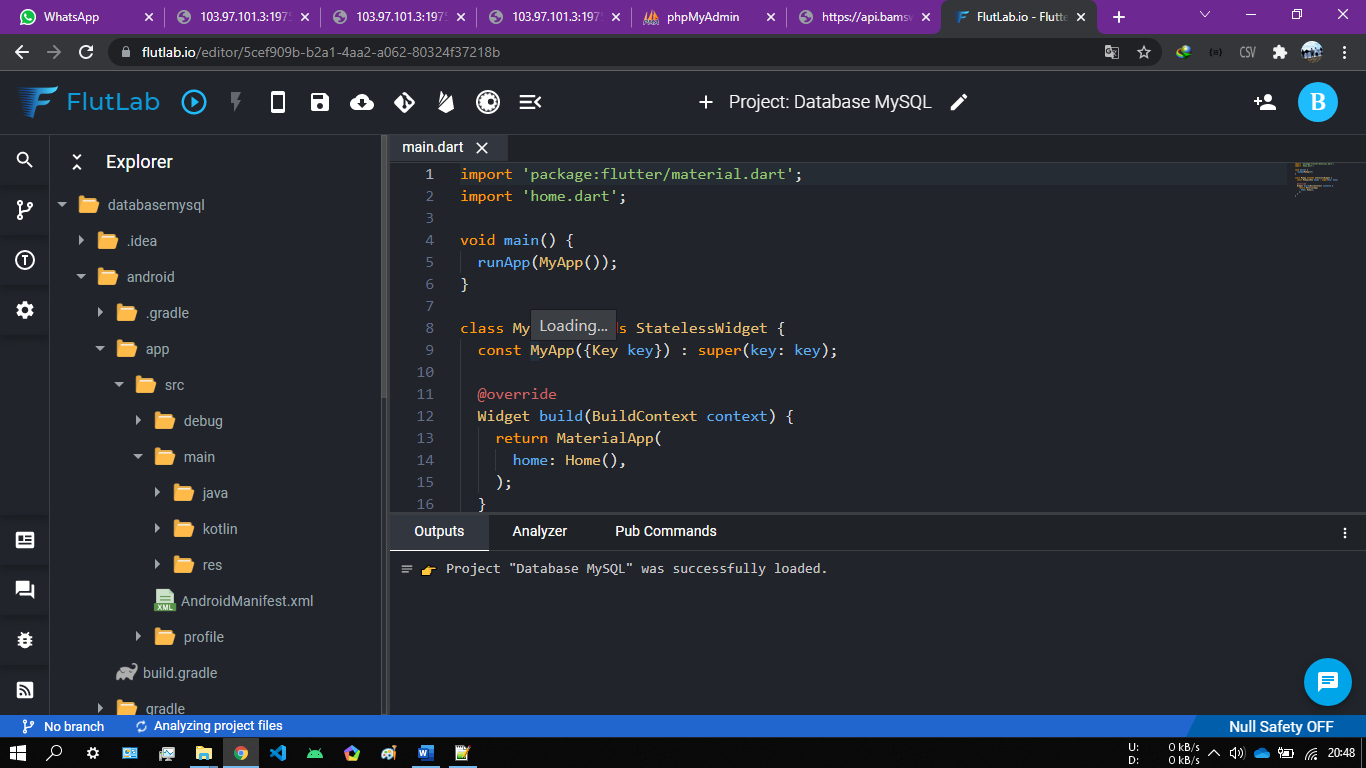
  http:

Tambahkan library pada **pubspec.yaml** bagian dependencies. Kemudian klik pada button **pub get**.



1. **Menambahkan Permission**

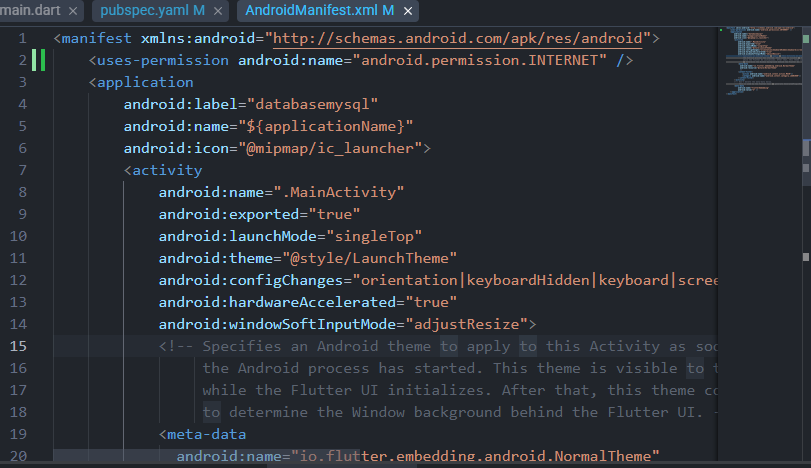
Permission digunakan untuk mendapatkan akses ke sumber daya pada smartphone, seperti lokasi, internet, kontak, dll. Untuk projek kita kali ini memerlukan koneksi internet, jadi kita perlu menambahkan permission untuk akses internet. Untuk menambahkan permission, cari file **android/app/src/main/AndroidManifest.xml.**



Kemudian tambahkan kode berikut sebelum bagian application.

<uses-permission android:name="android.permission.INTERNET" />

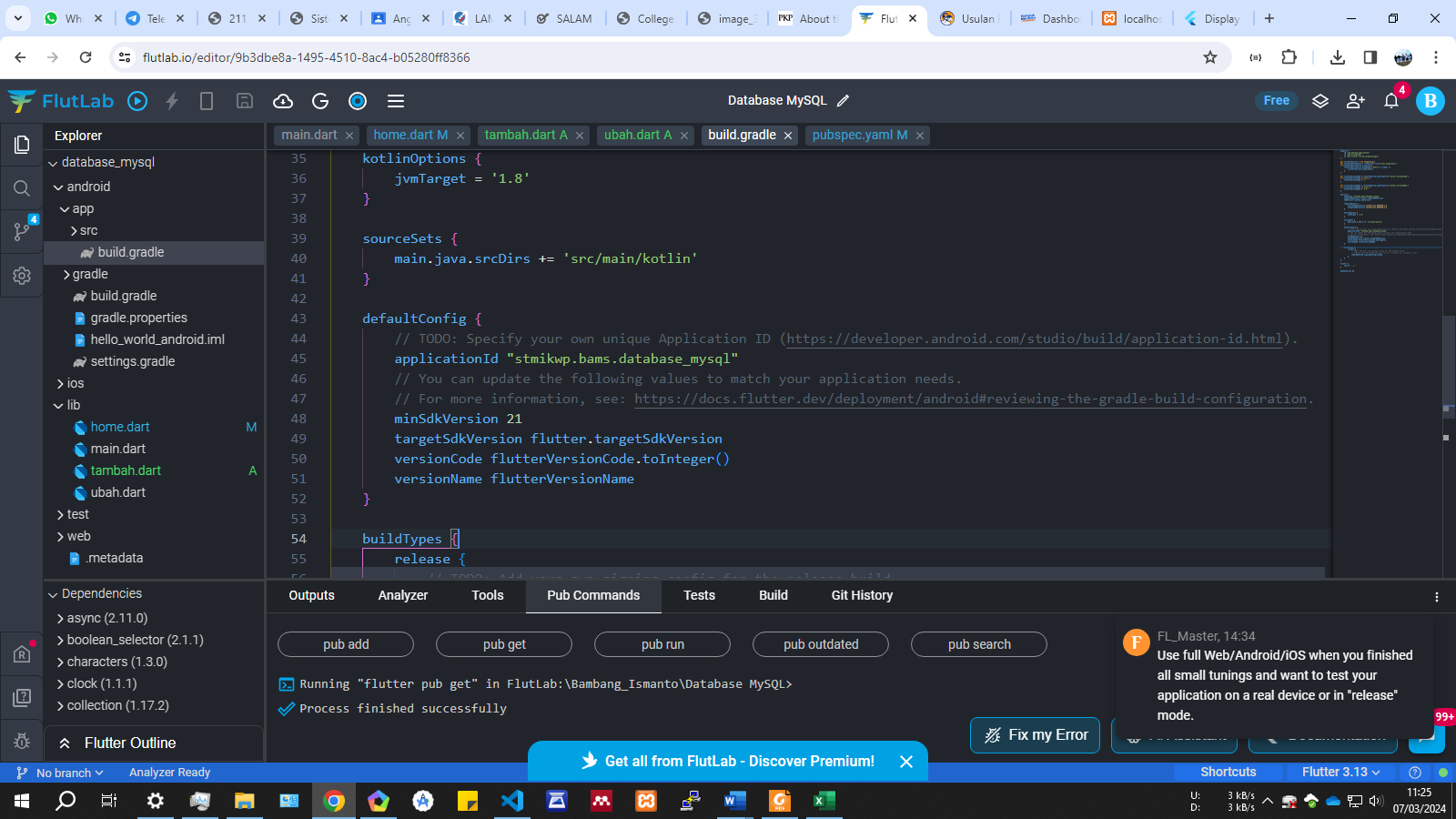
Sehingga menjadi:



Jangan lupa simpan.

1. **Setting minSDKVersion**

Untuk dapat menggunakan fitur terbaru, maka versi android harus minimal **API 21**. Untuk itu, ganti minSDKVersion pada **android/app/build.gradle**, sehingga menjadi seperti berikut:



1. **Menyiapkan Halaman Home**

Buat sebuah file baru dengan nama **home.dart**. Isi lengkap file tersebut adalah sebagai berikut:

import 'dart:async';

import 'dart:convert';

import 'package:flutter/material.dart';

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

class Home extends StatefulWidget {

  @override

  \_HomeState createState() => \_HomeState();

}

class \_HomeState extends State<Home> {

  Future<List> getData() async {

    final response = await http.get(

      Uri.parse("http://192.168.1.138/akademik/mahasiswa"),

    );

    return json.decode(response.body);

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      backgroundColor: Colors.grey[300],

      appBar: AppBar(

        title: Text("Data Mahasiswa"),

      ),

      floatingActionButton: FloatingActionButton(

        child: Icon(Icons.add),

        onPressed: () {

        },

      ),

      body: FutureBuilder<List>(

        future: getData(),

        builder: (context, snapshot) {

          if (snapshot.hasError) print(snapshot.error);

          return snapshot.hasData

              ? Container(

                  padding: EdgeInsets.fromLTRB(5, 10, 5, 5),

                  child: Card(

                    child: ListView.builder(

                      itemCount:

                          snapshot.data == null ? 0 : snapshot.data!.length,

                      itemBuilder: (context, i) {

                        return Column(

                          children: [

                            GestureDetector(

                              child: ListTile(

                                title: Text(snapshot.data![i]['nama']),

                                leading: Icon(Icons.person),

                                subtitle: Text("${snapshot.data![i]['nim']}"),

                                trailing: Icon(Icons.navigate\_next\_rounded),

                              ),

                              onTap: () {

                              },

                            ),

                            Divider(

                              color: Colors.black,

                              indent: 10,

                              endIndent: 10,

                            )

                          ],

                        );

                      },

                    ),

                  ),

                )

              : Center(

                  child: CircularProgressIndicator(),

                );

        },

      ),

    );

  }

}

**Penjelasan Singkat:**

import 'dart:async';

import 'dart:convert';

import 'package:flutter/material.dart';

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

import beberapa class yang diperlukan pada halaman home.

Future<List> getData() async {

    final response = await http.get(

      Uri.parse("http://192.168.1.113/akademik/mahasiswa"),

    );

    return json.decode(response.body);

  }

Method getData digunakan untuk mengambil data dari koneksi http (url disesuaikan, bisa gunakan localhost). Data yang didapat berupa data json yang kemudian di convert menjadi sebuah list. Nilai kembalian dari method ini berupa List.

body: FutureBuilder<List>(

        future: getData(),

        builder: (context, snapshot) {

          if (snapshot.hasError) print(snapshot.error);

          return snapshot.hasData

FutureBuilder merupakan sebuah widget yang digunakan untuk mendapatkan data dengan method getData tadi. Hasilnya akan ditangkap pada snapshot. jika snapshot memiliki data, maka data akan ditampilkan melalui Widget di dalamnya.

child: ListTile(

                                title: Text(snapshot.data![i]['nama']),

                                leading: Icon(Icons.person),

                                subtitle: Text("${snapshot.data![i]['nim']}"),

                                trailing: Icon(Icons.navigate\_next\_rounded),

                              ),

Data tadi ditampilkan menggunakan ListTile.

1. **Menyiapkan file main.dart**

Silahkan sesuaikan isi dari file **main.dart** menjadi seperti di bawah ini:

import 'package:flutter/material.dart';

import 'home.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: Home(),

    );

  }

}

1. **Build Aplikasi**

Kita akan mem-build projek untuk membuat file apk agar bisa dijalankan pada perangkat android. Untuk build, pilih **Build Project** (pilih android-arm). Setelah proses build selesai, download file apk yg ada, kemudian jalankan pada perangkat masing-masing.

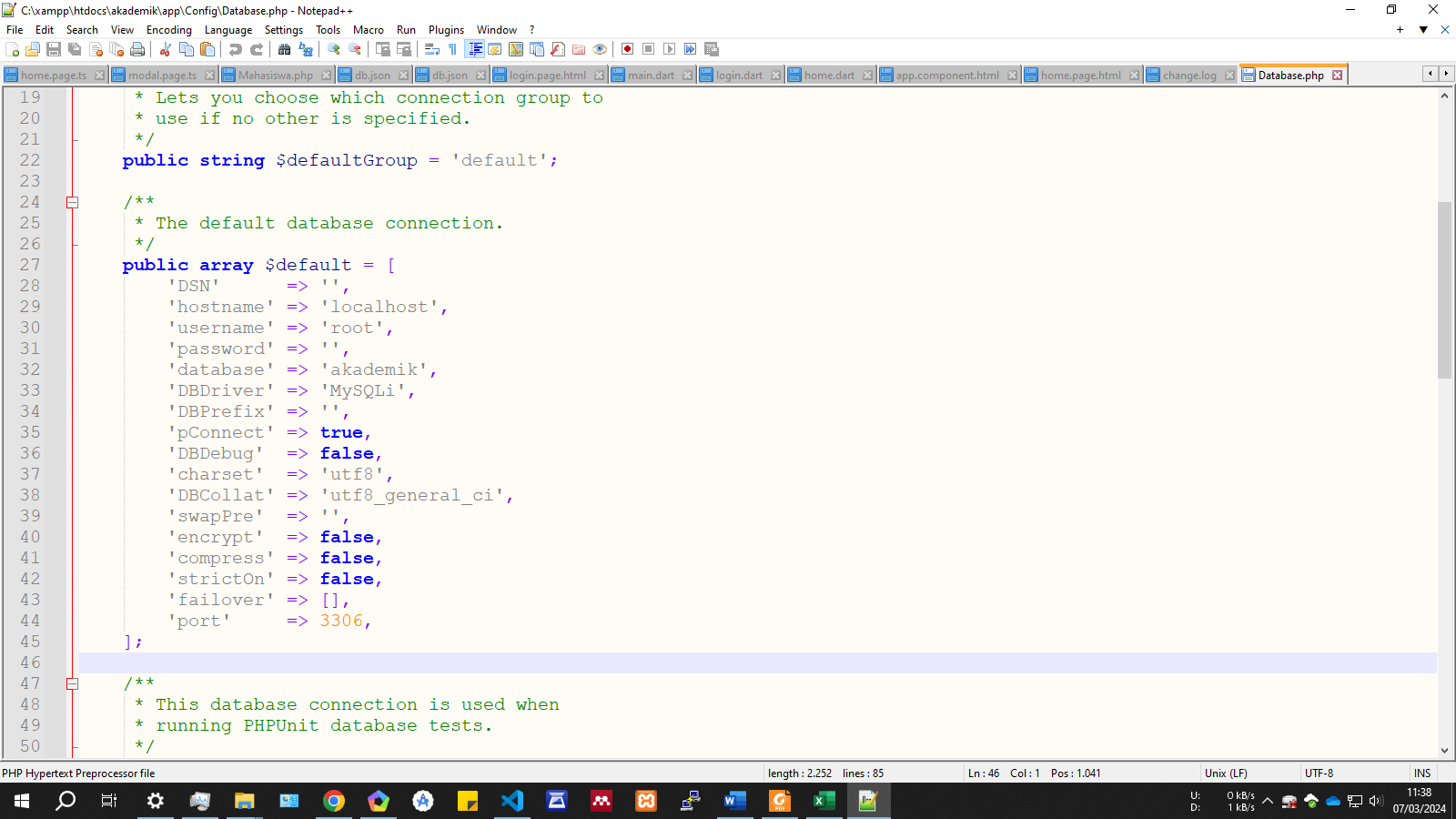
**Materi Tambahan: Menyiapkan REST API (jika menggunakan localhost)**

Silahkan download API yang telah disediakan. Langkah-langkah untuk memasang API:

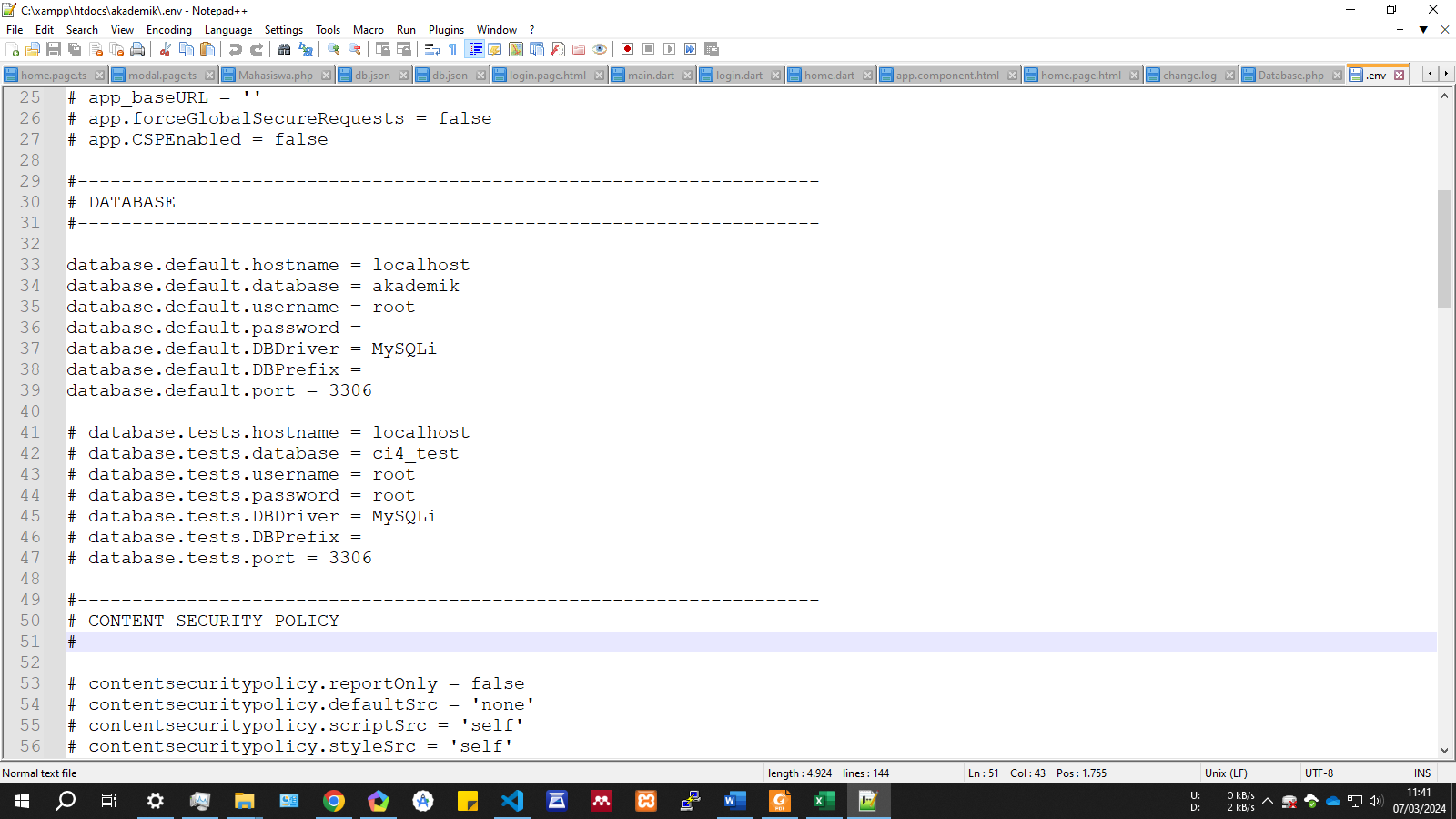
1. Meletakkan API

Ekstrak API yang sudah didownload, kemudian letakkan pada folder htdocs di xampp.

1. Setting API
2. Konfigurasi database yang akan digunakan dalam API tersebut. Buka file **Database.php** yang ada di **xampp\htdocs\aakademik\app\Config**. edit file tersebut pada bagian **hostname, username, password, database** seperti di bawah ini (isi disesuaikan):



Buka juga file **.env** pada folder akademik. edit file tersebut pada bagian **hostname, username, password, database** seperti di bawah ini (isi disesuaikan):



1. Menyiapkan Controller

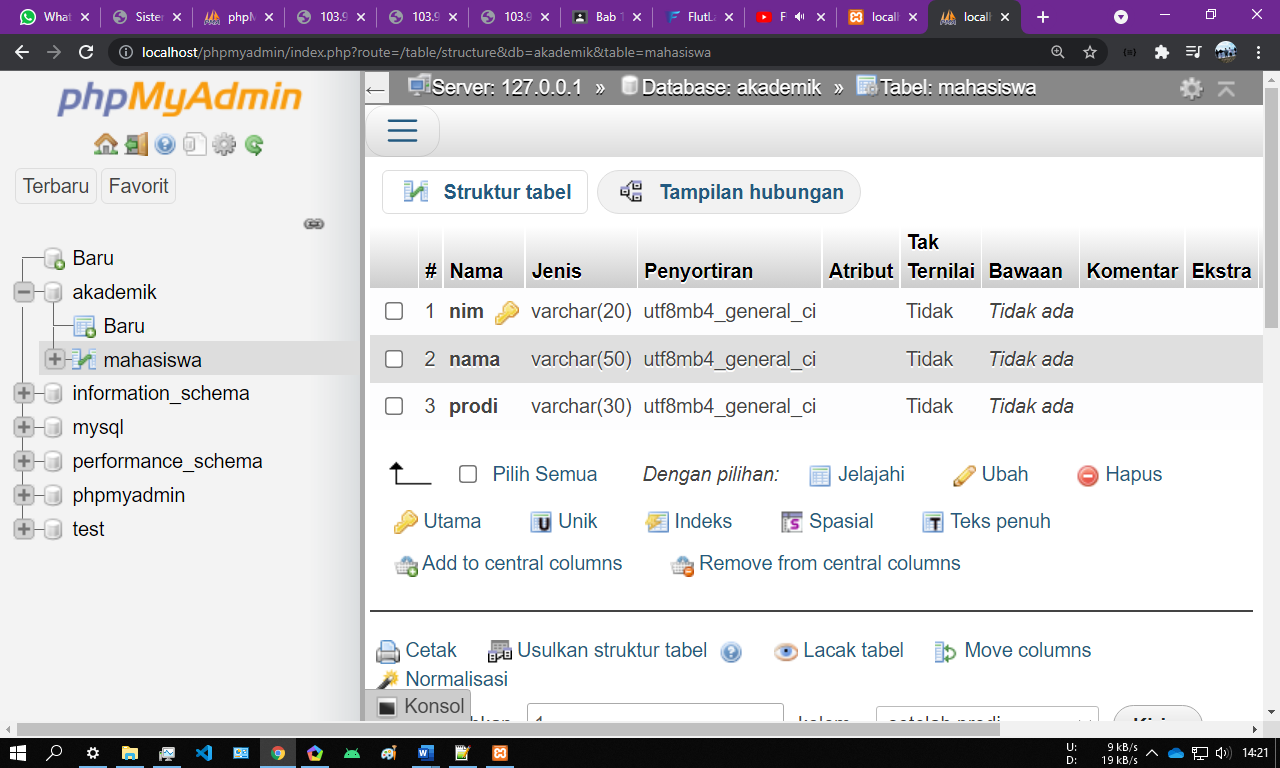
Controller digunakan untuk operasi CRUD pada database. File sudah saya buatkan (yang nantinya isinya bisa disesuaikan) dan dapat dilihat pada **xampp\htdocs\akademik\app\Controllers\Mahasiswa.php.**

1. Menyiapkan Model

Model digunakan untuk bentuk objek pada Controller. File sudah saya buatkan (yang nantinya isinya bisa disesuaikan) dan dapat dilihat pada **xampp\htdocs\akademik\app\Models\ModelMahasiswa.php.**

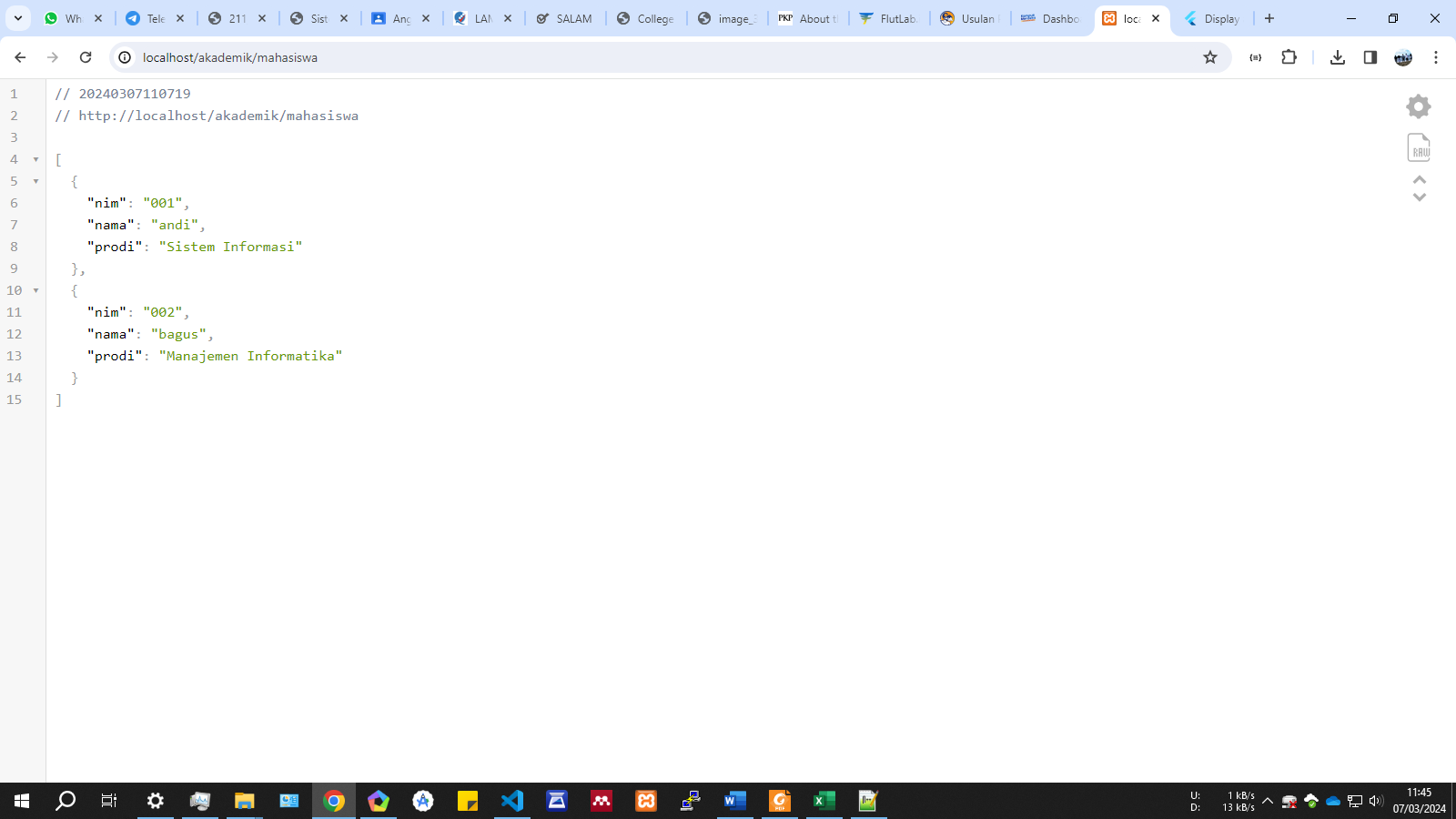
1. Menyiapkan database

Buat database dengan nama **akademik** dan buat juga tabelnya dengan nama **mahasiswa** dengan field **nim, nama, prodi (type varchar)**.



1. Mencoba API

Untuk mencoba API, silahkan gunakan browser untuk mengakses API tersebut.



Tampilan mungkin akan sedikit berbeda, karena di sini menggunakan extension untuk merapikan data JSON.

**RANGKUMAN**

**[Diisi dengan rangkuman materi yang telah diuraikan]**

**TES FORMATIF**

**[Diisi dengan tes formatif sesuai dengan materi yang telah diuraikan]**

**DAFTAR PUSTAKA**

**[Diisi dengan daftar pustaka yang digunakan dalam menyusun modul pembelajaran]**