Universitas Semarang

Fakultas Teknologi Informasi dan Komunikasi

Teknik Informatika

Mobile Application

Modul Praktikum Mahasiswa

Oleh:

Alauddin Maulana Hirzan, S. Kom., M. Kom

NIDN. 0607069401

Daftar Isi

Daftar Gambar

Pendahuluan

0.1 Mengenal Android

Sistem operasi Android adalah sistem operasi seluler untuk digunakan terutama untuk perangkat layar sentuh, ponsel, dan tablet. Desainnya memungkinkan pengguna memanipulasi perangkat seluler secara intuitif, dengan gerakan jari yang mencerminkan gerakan umum, seperti mencubit, menggesek, dan mengetuk.

Gambar 1: Perangkat Android

0.2 Mengenal Firebase dan Realtime Database

Firebase adalah platform yang dikembangkan oleh Google untuk membuat aplikasi seluler dan web. Salah satunya produk yang sering digunakan di Firebase adalah Realtime Database. Firebase Realtime Database adalah database yang dihosting di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung.

Gambar 2: Realtime Database

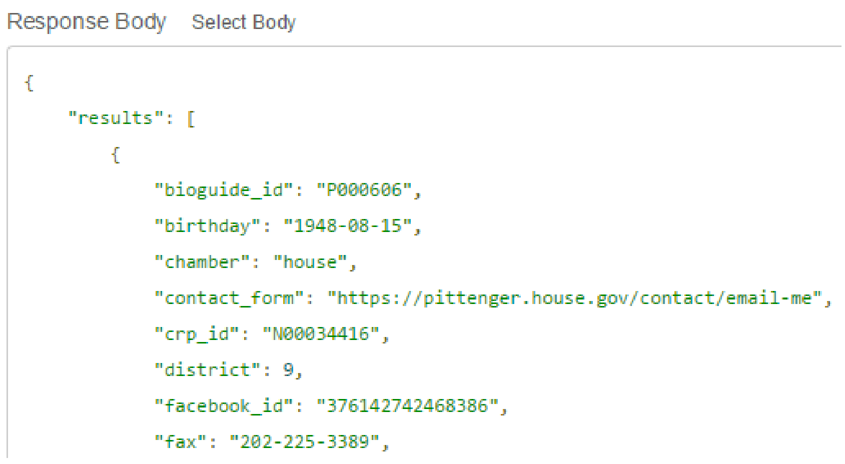
0.3 Mengenal JSON-Tree

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan. Mudah bagi manusia untuk membaca dan menulis. Sangat mudah bagi mesin untuk menguraikan dan menghasilkan. Ini didasarkan pada subset dari Standar Bahasa Pemrograman JavaScript ECMA-262 Edisi ke-3 - Desember 1999. JSON adalah format teks yang sepenuhnya bebas bahasa tetapi menggunakan konvensi yang akrab bagi pemrogram keluarga bahasa C, termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan banyak lainnya. Properti ini menjadikan JSON sebagai bahasa pertukaran data yang ideal.

JSON dibangun di atas dua struktur:

- Kumpulan pasangan nama/nilai. Dalam berbagai bahasa, ini diwujudkan sebagai objek, catatan, struct, kamus, tabel hash, daftar kunci, atau array asosiatif.
- Daftar nilai yang diurutkan. Dalam kebanyakan bahasa, ini diwujudkan sebagai array, vektor, daftar, atau urutan.

Ini adalah struktur data universal. Hampir semua bahasa pemrograman modern mendukungnya dalam satu atau lain bentuk. Masuk akal bahwa format data yang dapat dipertukarkan dengan bahasa pemrograman juga didasarkan pada struktur ini.



```
{
    "results": [
        {
            "bioguide_id": "P000606",
            "birthday": "1948-08-15",
            "chamber": "house",
            "contact_form": "https://pittenger.house.gov/contact/email-me",
            "crp_id": "N00034416",
            "district": 9,
            "facebook_id": "376142742468386",
            "fax": "202-225-3389",
```

Gambar 3: Contoh JSON Tree

0.4 Mengenal Flutter Framework

Flutter adalah kerangka kerja sumber terbuka oleh Google untuk membangun aplikasi multi-platform yang indah, dikompilasi secara asli, dari satu basis kode. Flutter mengubah proses pengembangan aplikasi. Buat, uji, dan terapkan aplikasi seluler, web, desktop, dan tersemat yang cantik dari satu basis kode.

Gambar 4: Flutter

Persiapan Praktikum

Agar praktikum dapat berjalan dengan lancar, mahasiswa diwajibkan memenuhi persyaratan berikut baik dalam bentuk perangkat keras maupun lunak:

0.5 Perangkat Keras

- Prosesor dengan 4 inti
- RAM minimal 4GB, rekomendasi 8GB
- HDD 10GB

0.6 Perangkat Lunak

Perangkat lunak berikut ini wajib diinstall oleh mahasiswa demi lancarnya praktikum:

- Browser
- FlutLab Desktop (opsional)

Bab 1

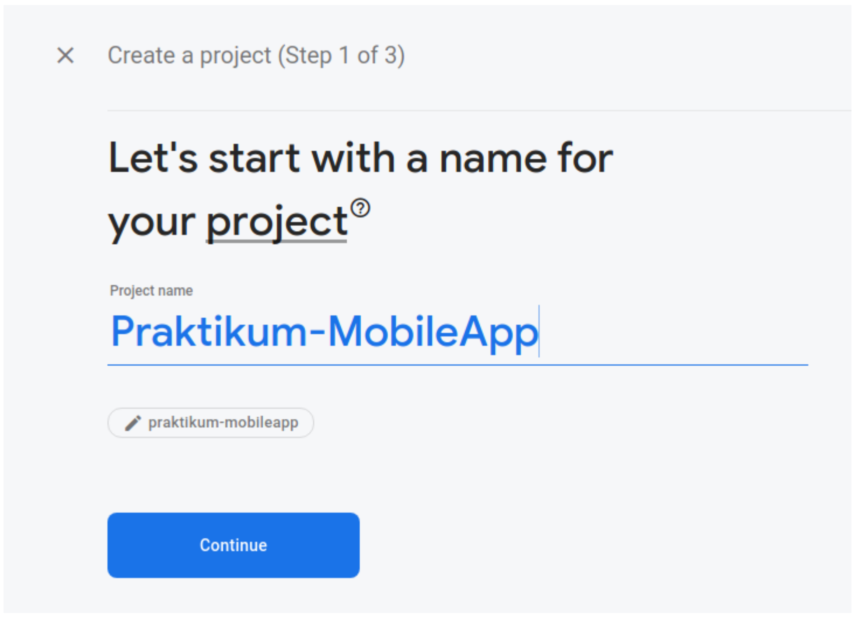
Praktikum 1

1.1 Konfigurasi Firebase Realtime DB dan Flutter IDE

Di bagian ini mahasiswa diajarkan bagaimana membuat projek Firebase dengan Realtime Database. Selain itu mahasiswa juga diperkenalkan dengan antarmuka Flutlab untuk membuat aplikasi Mobile multi platform. Mahasiswa diwajibkan untuk menyelesaikan **Persiapan Praktikum** sebelum masuk ke tahapan ini.

1.2 Tutorial

1. Buka `firebase.google.com`, login dengan **Akun Google Pribadi**. Jika sudah, klik **Get Started** > **Create Project** > Beri Nama Projek **Praktikum-MobileApp** > Klik **Continue** > Matikan **Analytic** > **Create Project** > Tunggu Proses Selesai.

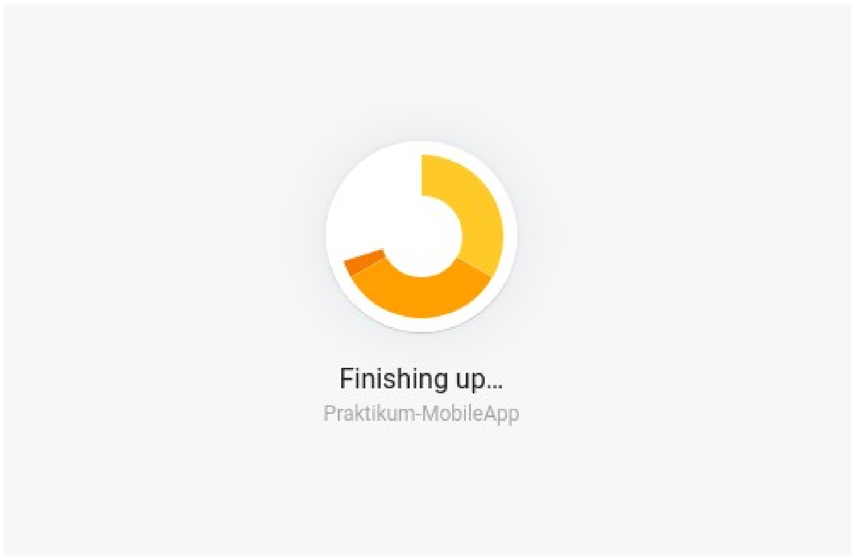


Create a project (Step 1 of 3)

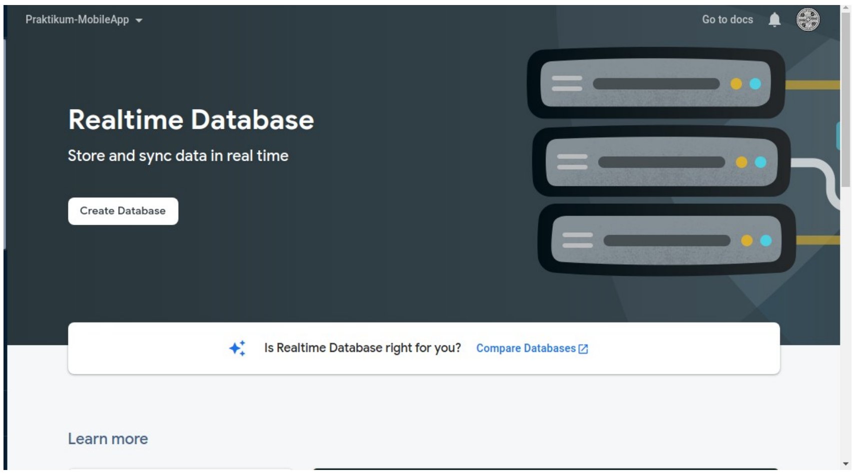
Let's start with a name for your project

Project name: Praktikum-MobileApp

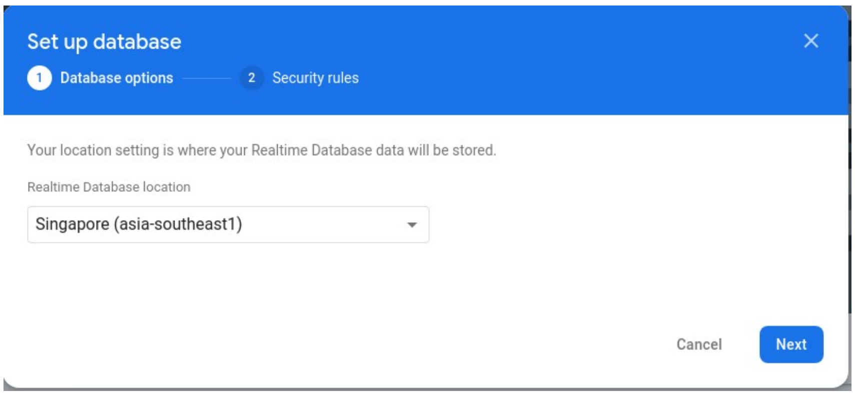
praktikum-mobileapp



2. Untuk membuat **Realtime Database**, klik **Menu Build** yang ada di samping kiri > piih **Realtime Database** > klik **Create Database**



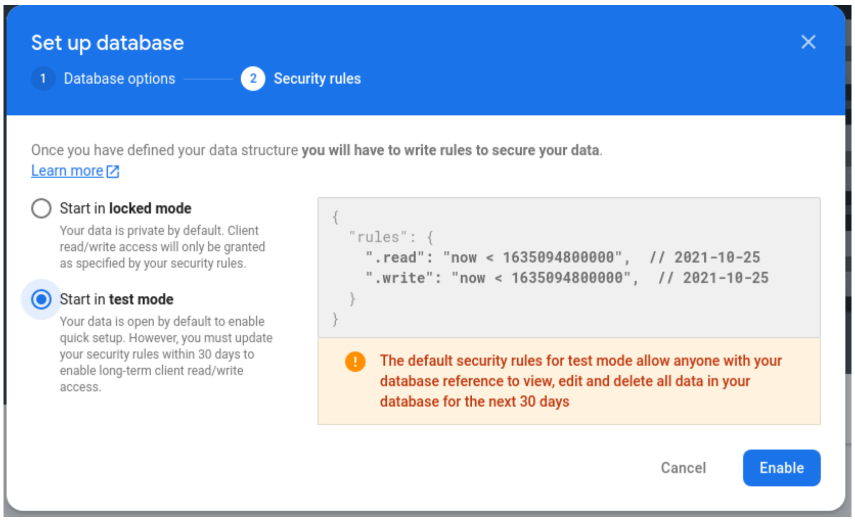
3. Pilih lokasi database (**Singapore**), klik **Next**



Your location setting is where your Realtime Database data will be stored.

Realtime Database location: Singapore (asia-southeast1)

4. Ketika ditanya **Security Rules**, pilih **Locked Mode** / **Private** atau **Test Mode** / **Public Temporary**. Pilih **Test Mode**.



Once you have defined your data structure **you will have to write rules to secure your data**.

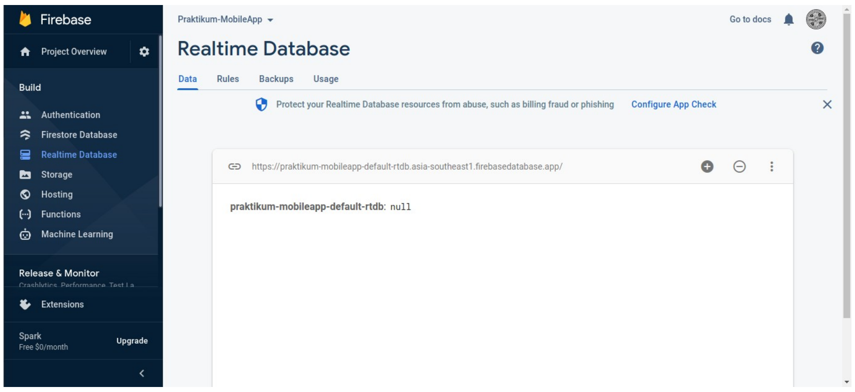
[ ] Start in **locked mode**
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

[✓ Start in **test mode**]
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": "now < 1635094800000",   // 2021-10-25
    ".write": "now < 1635094800000",  // 2021-10-25
  }
}
```

The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

5. Tunggu hingga database muncul dan siap dipakai.

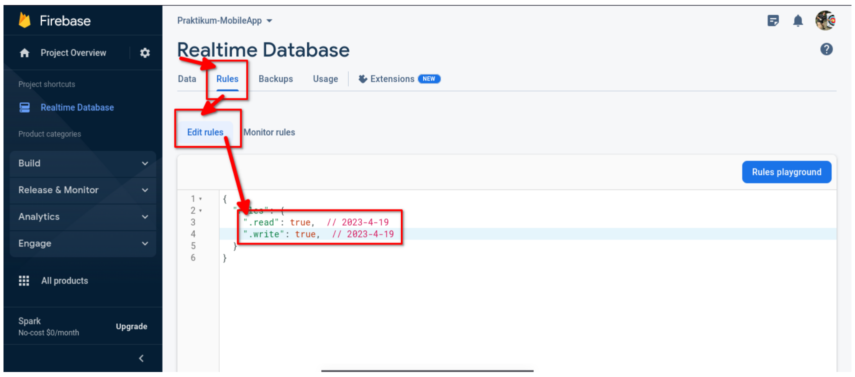


Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://praktikum-mobileapp-default-rtdb.asia-southeast1.firebasedatabase.app/

praktikum-mobileapp-default-rtdb: null

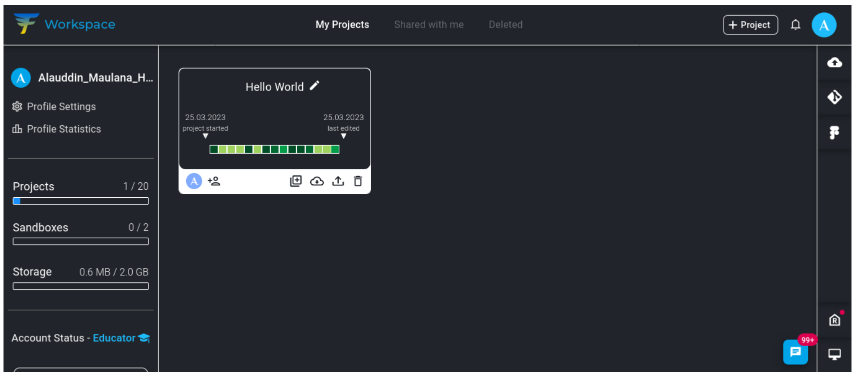
6. Setelah Database sudah siap, berikutnya adalah mengonfigurasikan **Rules** yang sudah dibuat oleh Firebase. Hal ini memastikan database bisa diakses kapanpun
7. Klik tab **Rules** → **Edit rules** → Ubah data berikut :
   - "now < xxxxxxxxxx" menjadi **true**



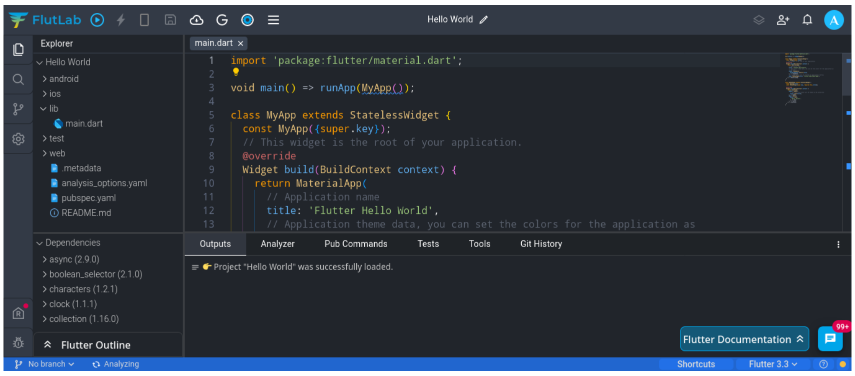
```
{
  "rules": {
    ".read": true,   // 2023-4-19
    ".write": true,   // 2023-4-19
  }
}
```

8. Klik **Publish** untuk menerapkan perubahan.

9. Berikutnya buka web `https://flutlab.io/workspace`. Buatlah akun dengan **E-Mail Pribadi**, dan buka kembali link tersebut hingga terlihat seperti gambar di bawah:



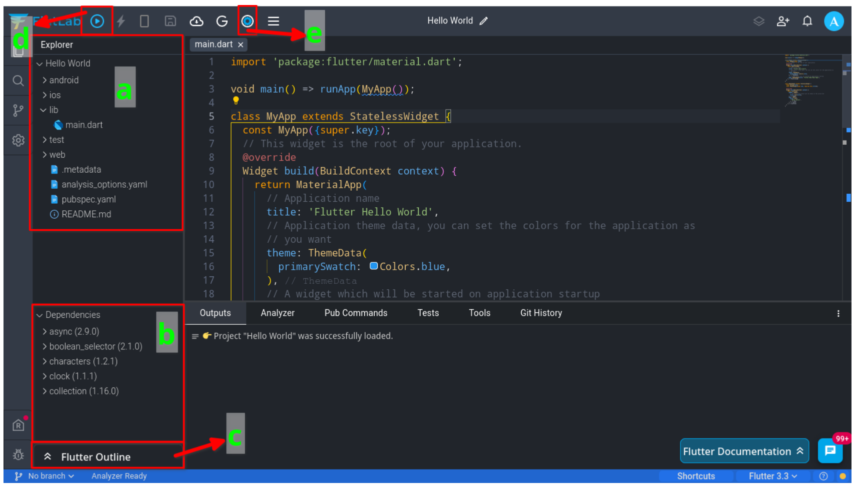
10. Mahasiswa akan melihat **Project Default** berisikan **Hello World** dari Flutlab. Klik projek tersebut hingga muncul tampilan seperti berikut:



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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // Application name
      title: 'Flutter Hello World',
      // Application theme data, you can set the colors for the application as
```

```
Project "Hello World" was successfully loaded.
```

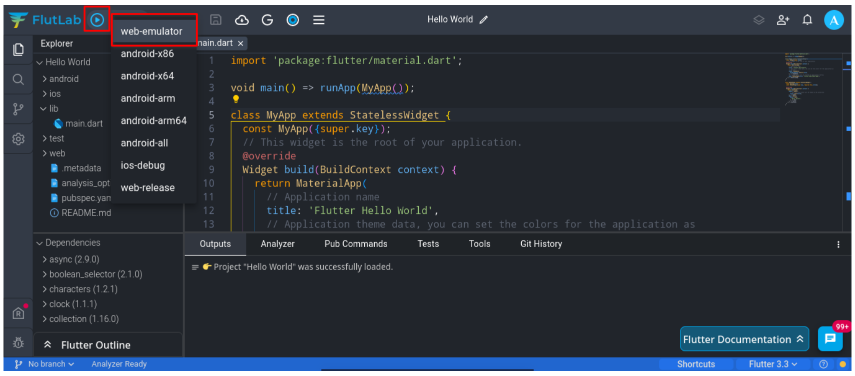
11. Ada beberapa bagian yang harus dipahami oleh mahasiswa:
    (a) Struktur Folder
    (b) Ketergantungan Aplikasi
    (c) Navigasi Item Aplikasi
    (d) Build dan Play Aplikasi
    (e) Analyzer Kebenaran Kode

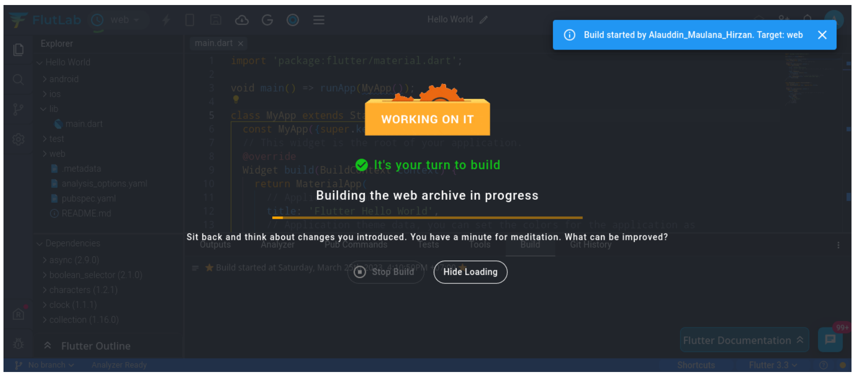


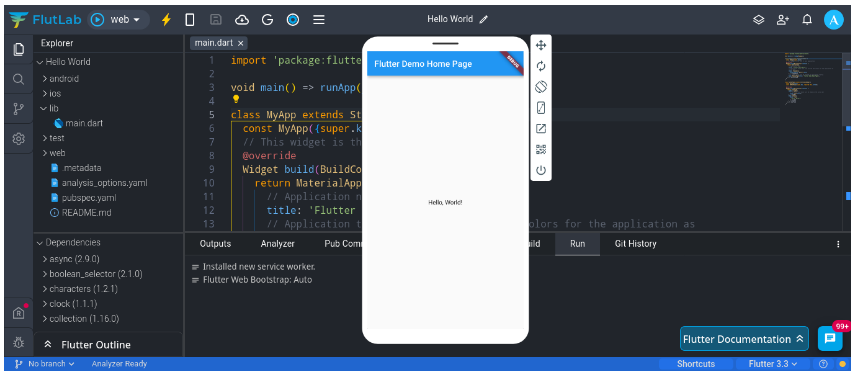
```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // Application name
      title: 'Flutter Hello World',
      // Application theme data, you can set the colors for the application as
      // you want
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ), // ThemeData
      // A widget which will be started on application startup
```

```
Project "Hello World" was successfully loaded.
```

12. Untuk menguji coba aplikasi dari projek saat ini bisa dalam beberapa mode:
    - Web (Rekomendasi untuk preview)
    - Android APK (1 kali per hari, hanya untuk final)
    - IPhone IPS (1 kali per hari, hanya untuk final)
13. Arahkan Mouse ke Tombol **Play** → Pilih **Web**. Jika sudah dalam mode **Web**. Klik tombol **Play** dan tunggu proses building hingga emulator web muncul dan aplikasi ditampilkan.



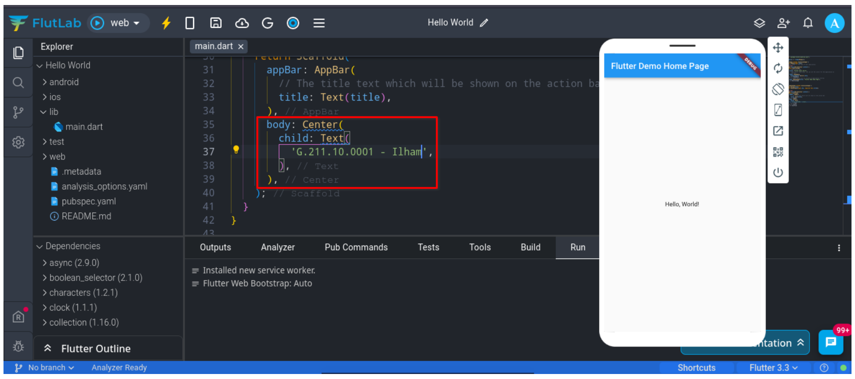

Build started by Alauddin_Maulana_Hirzan. Target: web
WORKING ON IT
It's your turn to build
Building the web archive in progress
Sit back and think about changes you introduced. You have a minute for meditation. What can be improved?
Stop Build
Hide Loading



```
import 'package:flutte

void main() => runApp(

class MyApp extends St
  const MyApp({super.k
  // This widget is th
  @override
  Widget build(BuildCo
    return MaterialApp
      // Application n
      title: 'Flutter
      // Application t                                    olors for the application as
```

```
Installed new service worker.
Flutter Web Bootstrap: Auto
```

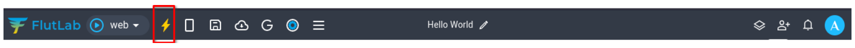
14. Scroll turun **Kode** dan temukan tulisan 'Hello, World!' dan ubah menjadi **NIM - Nama**



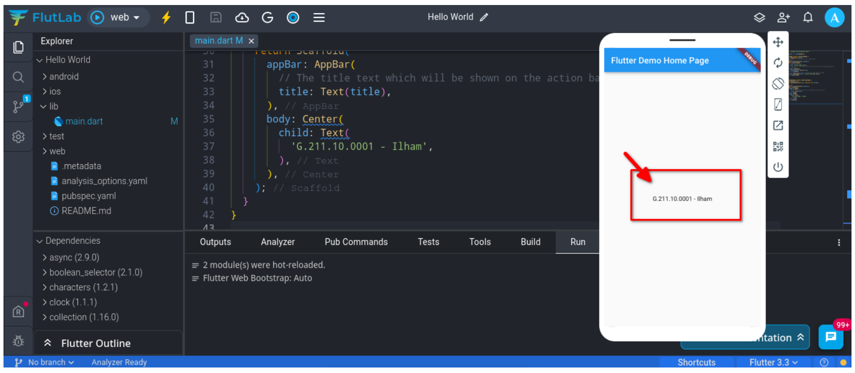
```
      appBar: AppBar(
        // The title text which will be shown on the action ba
        title: Text(title),
      ), // AppBar
      body: Center(
        child: Text(
          'G.211.10.0001 - Ilham',
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```

```
Installed new service worker.
Flutter Web Bootstrap: Auto
```

15. Klik **Hot Reload** (Simbol Petir disebelah tombol **Play**) untuk melakukan **Refresh**. Hanya bisa digunakan ketika tombol **Play** sudah di klik satu (1) kali.



16. Secara otomatis perubahan akan ditampilkan tanpa perlu **Building Ulang**



```
      appBar: AppBar(
        // The title text which will be shown on the action ba
        title: Text(title),
      ), // AppBar
      body: Center(
        child: Text(
          'G.211.10.0001 - Ilham',
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```

```
2 module(s) were hot-reloaded.
Flutter Web Bootstrap: Auto
```

17. **Screenshot** hasil dari praktikum ini dan kirimkan ke E-Learning

Bab 2

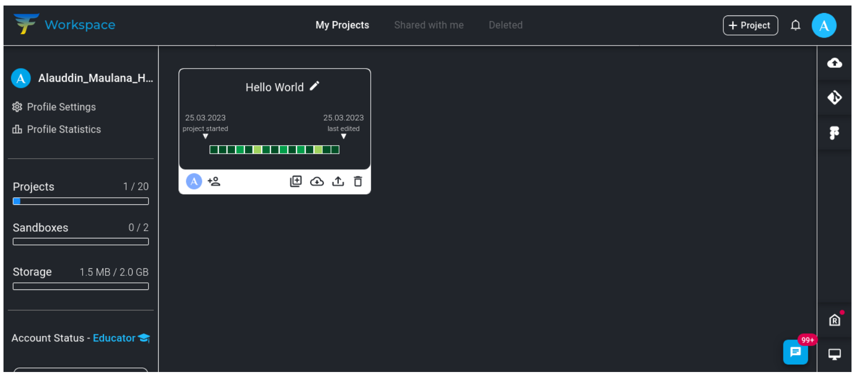
Praktikum 2

2.1 Antarmuka Login Flutter

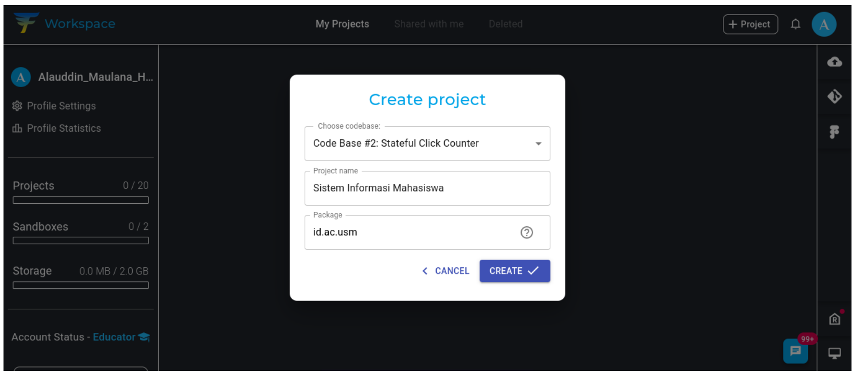
Di bagian ini mahasiswa diperkenalkan dengan pemrograman DART untuk membuat tampilan login melalui IDE Flutter. Mahasiswa diwajibkan untuk menyelesaikan pertemuan sebelumnya agar praktikum ini berjalan dengan baik.

2.2 Tutorial

1. Buka web `https://flutlab.io/workspace`, hapus projek sebelumnya karena limitasi dari Flutlab.



2. Lalu buatlah projek dengan konfigurasi lalu klik **Create**:
   - **Codebase** : Code Base #2 : Stateful Click Counter
   - **Project Name** : Sistem Informasi Mahasiswa
   - **Package** : id.ac.usm (Domain USM)

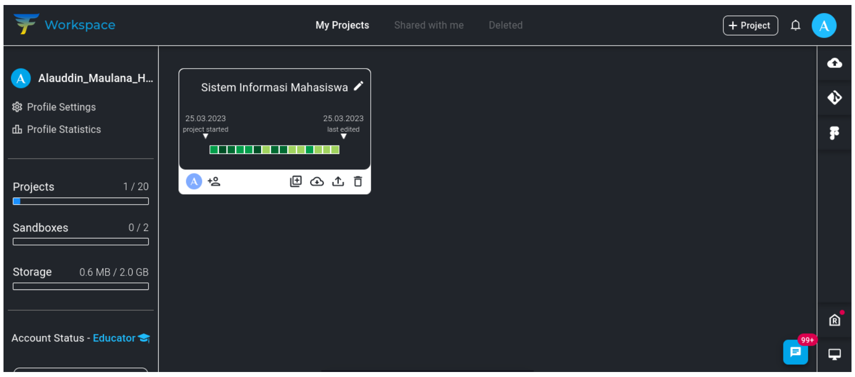

Create project

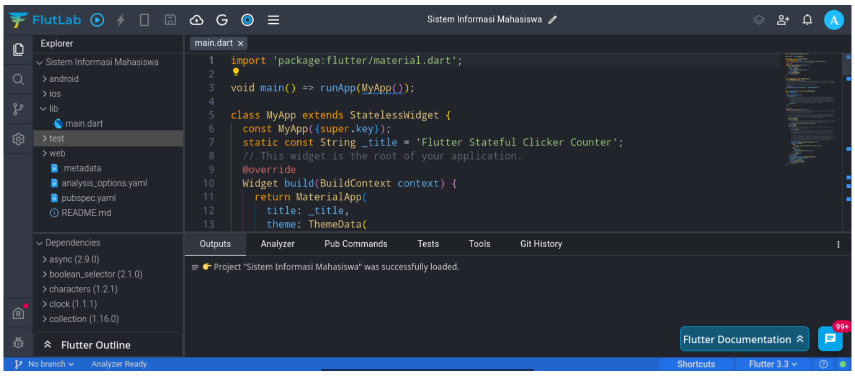
Choose codebase: Code Base #2: Stateful Click Counter

Project name: Sistem Informasi Mahasiswa

Package: id.ac.usm

3. Tunggu Flutlab membuat projek awal. Lalu klik projek tersebut




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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Flutter Stateful Clicker Counter';
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      theme: ThemeData(
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

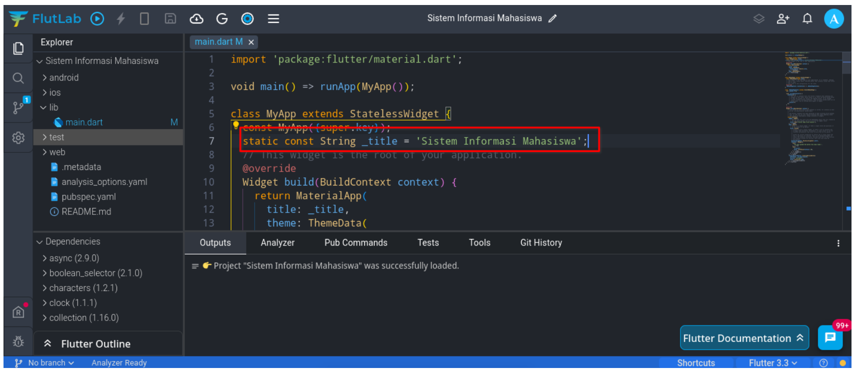
4. Di kode ada bagian-bagian yang perlu dirubah. Ubahlah judul aplikasi dengan mengganti kode berikut (Sebelum → Sesudah):

Potongan Kode (Sebelum)

```dart
static const String _title = 'Flutter Stateful Clicker Counter';
```

Potongan Kode (Sesudah)

```dart
static const String _title = 'Sistem Informasi Mahasiswa';
```



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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      theme: ThemeData(
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

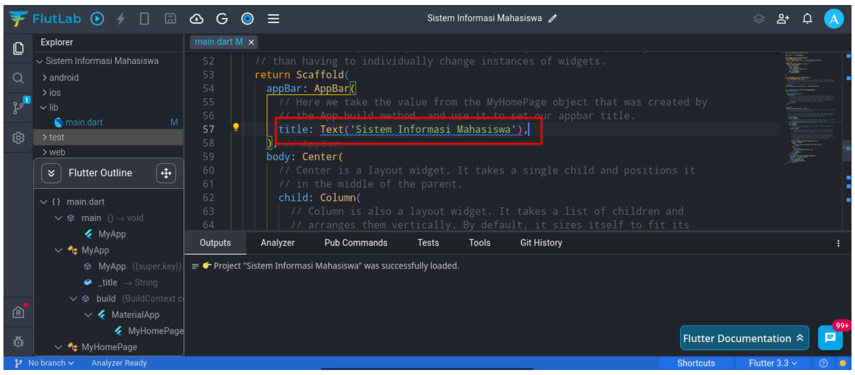
5. Berikutnya adalah mengubah tampilan **Judul App** yang ditampilkan oleh aplikasi

Potongan Kode (Sebelum)

```
title: Text('Flutter Demo Click Counter'),
```

Potongan Kode (Sesudah)

```
title: Text('Sistem Informasi Mahasiswa'),
```

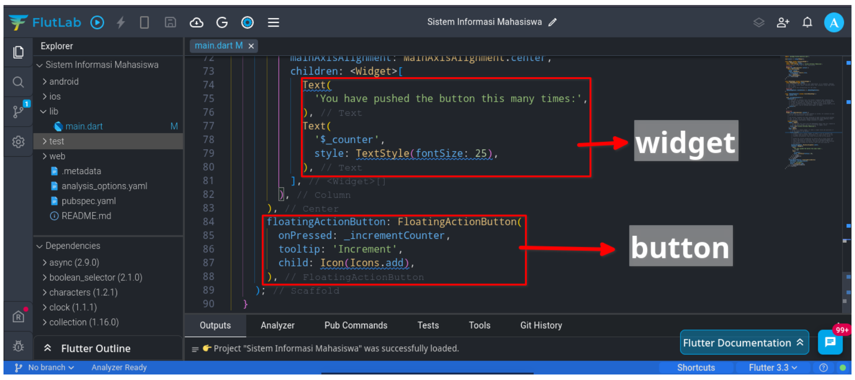


```dart
      // than having to individually change instances of widgets.
      return Scaffold(
        appBar: AppBar(
          // Here we take the value from the MyHomePage object that was created by
          // the App.build method, and use it to set our appbar title.
          title: Text('Sistem Informasi Mahasiswa'),
        ), // AppBar
        body: Center(
          // Center is a layout widget. It takes a single child and positions it
          // in the middle of the parent.
          child: Column(
            // Column is also a layout widget. It takes a list of children and
            // arranges them vertically. By default, it sizes itself to fit its
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

6. Berikutnya adalah mengosongkan widget-widget yang ada di tengah aplikasi, cukup menghapus kode yang berada di dalam **children: <Widget>[]**(Bisa dilihat via **Flutter Outline**)

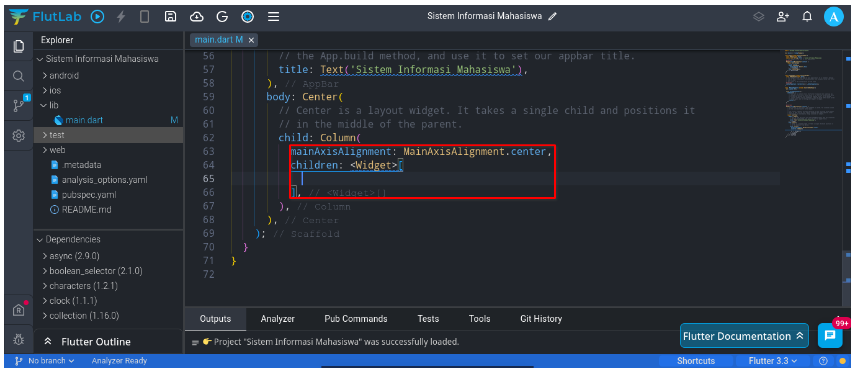
7. Hapus bagian yang ditandai di gambar berikut



```dart
              mainAxisAlignment: MainAxisAlignment.center,
              children: <Widget>[
                Text(
                  'You have pushed the button this many times:',
                ), // Text
                Text(
                  '$_counter',
                  style: TextStyle(fontSize: 25),
                ), // Text
              ], // <Widget>[]
            ), // Column
          ), // Center
          floatingActionButton: FloatingActionButton(
            onPressed: _incrementCounter,
            tooltip: 'Increment',
            child: Icon(Icons.add),
          ), // FloatingActionButton
        ); // Scaffold
      }
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

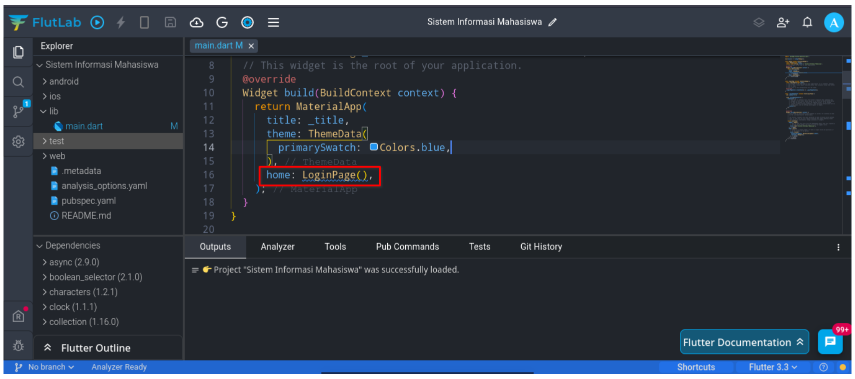
8. Jika sudah dihapus akan terlihat seperti berikut (children: <Widget>[] sudah tidak memiliki widget apapun):



```
56            // the App.build method, and use it to set our appbar title.
57            title: Text('Sistem Informasi Mahasiswa'),
58          ), // AppBar
59          body: Center(
60            // Center is a layout widget. It takes a single child and positions it
61            // in the middle of the parent.
62            child: Column(
63              mainAxisAlignment: MainAxisAlignment.center,
64              children: <Widget>[
65                
66              ], // <Widget>[]
67            ), // Column
68          ), // Center
69        ); // Scaffold
70      }
71    }
72
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

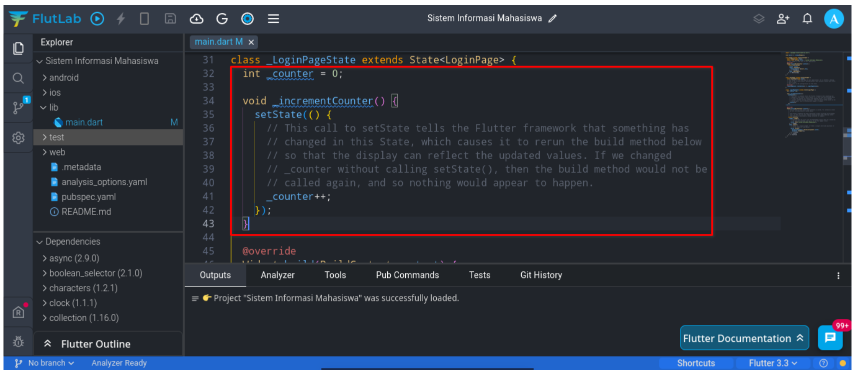
9. Sebelum mulai memasukkan kode tampilan aplikasi, ada beberapa bagian kode yang dirubah (Abaikan Error sampai kode selesai diubah):
   - Ubah **home: MyHomePage()** menjadi **home: LoginPage()** (Baris 16)
   - Ubah **class MyHomePage** menjadi **class LoginPage** (Baris 21)
   - Ubah **const MyHomePage** menjadi **const LoginPage** (Baris 22)
   - Ubah **_MyHomePageState createState() => _MyHomePageState();** menjadi **_LoginPageState createState() => _LoginPageState();** (Baris 28)
   - Ubah **class _MyHomePageState extends State<MyHomePage>** menjadi **class _LoginPageState extends State<LoginPage>** (Baris 31)



```
 8    // This widget is the root of your application.
 9    @override
10    Widget build(BuildContext context) {
11      return MaterialApp(
12        title: _title,
13        theme: ThemeData(
14          primarySwatch: Colors.blue,
15        ), // ThemeData
16        home: LoginPage(),
17      ); // MaterialApp
18    }
19  }
20
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

10. Hapus variabel serta fungsi yang ada di class **_LoginPageState**



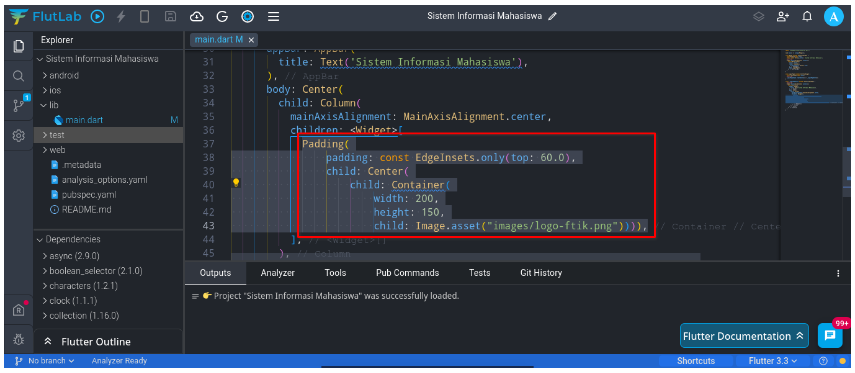
```
31  class _LoginPageState extends State<LoginPage> {
32    int _counter = 0;
33
34    void _incrementCounter() {
35      setState(() {
36        // This call to setState tells the Flutter framework that something has
37        // changed in this State, which causes it to rerun the build method below
38        // so that the display can reflect the updated values. If we changed
39        // _counter without calling setState(), then the build method would not be
40        // called again, and so nothing would appear to happen.
41        _counter++;
42      });
43    }
44
45    @override
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

11. Kode siap untuk ditambahkan widget-widget baru. Masukkan kode berikut di dalam **children: <Widget>** yang sudah dikosongkan sebelumnya. Pastikan memasukkan kode di dalam tag **[ ]**:

Potongan Kode

```
Padding(
    padding: const EdgeInsets.only(top: 60.0),
    child: Center(
        child: Container(
            width: 200,
            height: 150,
            child: Image.asset("images/logo-ftik.png")))),
```



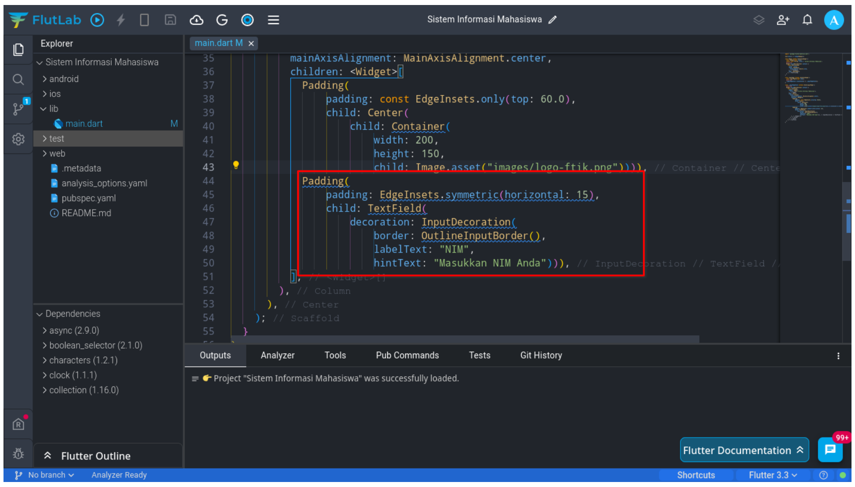
```
        title: Text('Sistem Informasi Mahasiswa'),
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
           Padding(
               padding: const EdgeInsets.only(top: 60.0),
               child: Center(
                   child: Container(
                       width: 200,
                       height: 150,
                       child: Image.asset("images/logo-ftik.png")))), // Container // Cente
          ], // <Widget>[]
        ), // Column
```

Project "Sistem Informasi Mahasiswa" was successfully loaded.

12. Kemudian tambahkan kode berikut masih di dalam tag **[ ]** tetapi di bawah kode sebelumnya. Kode ini digunakan untuk input NIM mahasiswa

Potongan Kode

```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 15),
    child: TextField(
        decoration: InputDecoration(
        border: OutlineInputBorder(),
        labelText: "NIM",
        hintText: "Masukkan NIM Anda"))),
```



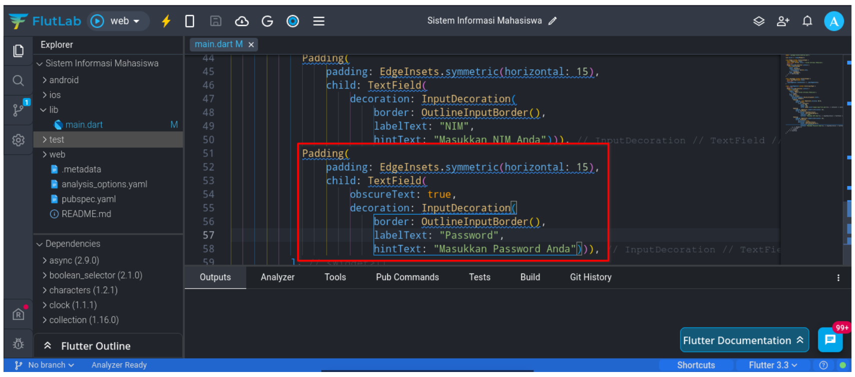
```
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
           Padding(
               padding: const EdgeInsets.only(top: 60.0),
               child: Center(
                   child: Container(
                       width: 200,
                       height: 150,
                       child: Image.asset("images/logo-ftik.png")))), // Container // Cente
           Padding(
               padding: EdgeInsets.symmetric(horizontal: 15),
               child: TextField(
                   decoration: InputDecoration(
                       border: OutlineInputBorder(),
                       labelText: "NIM",
                       hintText: "Masukkan NIM Anda"))), // InputDecoration // TextField /
          ], // <Widget>[]
        ), // Column
      ), // Center
    ); // Scaffold
  }
```

Project "Sistem Informasi Mahasiswa" was successfully loaded.

13. Berikutnya menambahkan kode input untuk password di bawah kode sebelumnya

Potongan Kode

```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 15),
    child: TextField(
        obscureText: true,
        decoration: InputDecoration(
        border: OutlineInputBorder(),
        labelText: "Password",
        hintText: "Masukkan Password Anda"))),
```

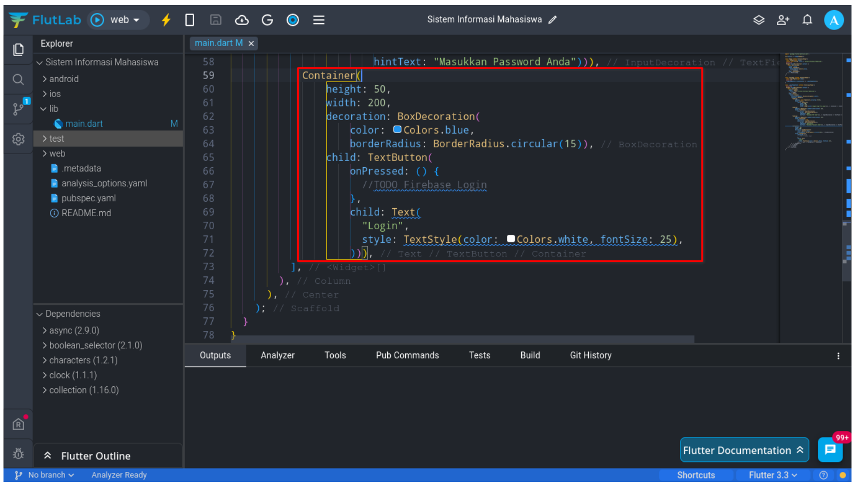


```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 15),
    child: TextField(
        decoration: InputDecoration(
            border: OutlineInputBorder(),
            labelText: "NIM",
            hintText: "Masukkan NIM Anda"))), // InputDecoration // TextField //
Padding(
    padding: EdgeInsets.symmetric(horizontal: 15),
    child: TextField(
        obscureText: true,
        decoration: InputDecoration(
            border: OutlineInputBorder(),
            labelText: "Password",
            hintText: "Masukkan Password Anda"))), // InputDecoration // TextFie
```

14. Setelah field input selesai, berikutnya tambahkan tombol untuk **Login**

Potongan Kode

```
Container(
    height: 50,
    width: 200,
    decoration: BoxDecoration(
        color: Colors.blue,
        borderRadius: BorderRadius.circular(15)),
    child: TextButton(
        onPressed: () {
            //TODO Firebase Login
        },
        child: Text(
        "Login",
        style: TextStyle(color: Colors.white, fontSize: 25),
    ))),
```

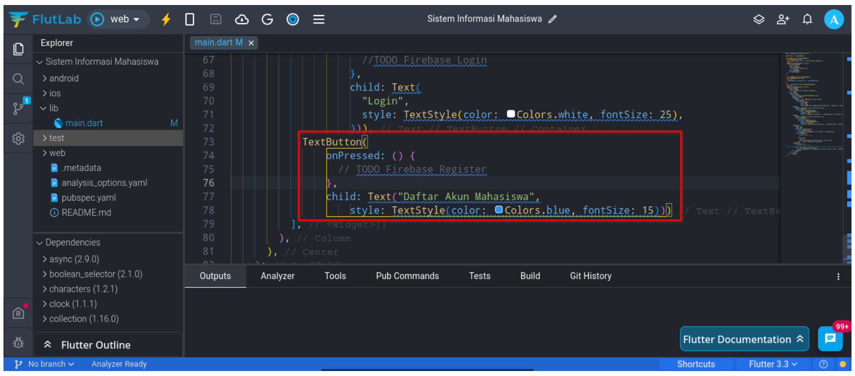


```
                hintText: "Masukkan Password Anda"))), // InputDecoration // TextFie
            Container(
                height: 50,
                width: 200,
                decoration: BoxDecoration(
                    color: Colors.blue,
                    borderRadius: BorderRadius.circular(15)), // BoxDecoration
                child: TextButton(
                    onPressed: () {
                      //TODO Firebase Login
                    },
                    child: Text(
                      "Login",
                      style: TextStyle(color: Colors.white, fontSize: 25),
                    ))), // Text // TextButton // Container
          ], // <Widget>[]
        ), // Column
      ), // Center
    ); // Scaffold
  }
}
```

15. Terakhir adalah menambahkan kode untuk Register dengan widget jenis **TextButton**

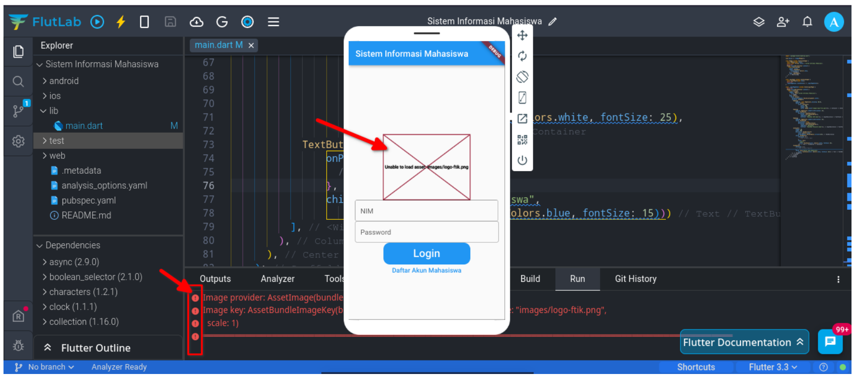
Potongan Kode

```
TextButton(
    onPressed: () {
        // TODO Firebase Register
    },
    child: Text("Daftar Akun Mahasiswa",
    style: TextStyle(color: Colors.blue, fontSize: 15)))
```



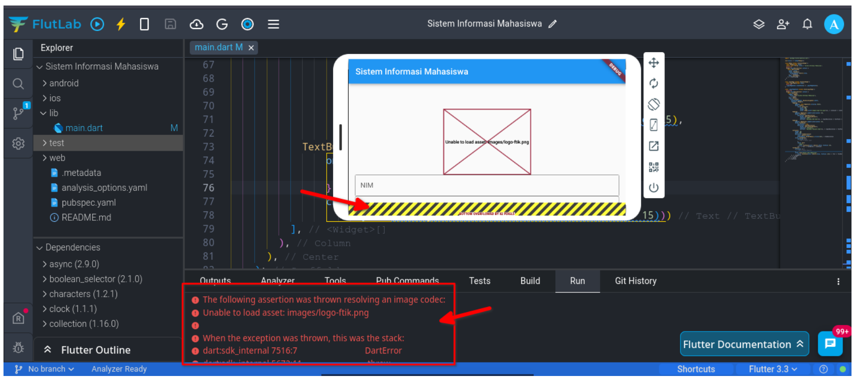
```
                      //TODO Firebase Login
                    },
                    child: Text(
                      "Login",
                      style: TextStyle(color: Colors.white, fontSize: 25),
                    ))), // Text // TextButton // Container
            TextButton(
                onPressed: () {
                  // TODO Firebase Register
                },
                child: Text("Daftar Akun Mahasiswa",
                    style: TextStyle(color: Colors.blue, fontSize: 15))) // Text // TextBu
          ], // <Widget>[]
        ), // Column
      ), // Center
```

16. Cek pastikan kode tidak ada yang error. Jika ada perbaiki kode tersebut. Jika tidak ada error. Klik **Play (mode Web)** untuk melihat hasil saat ini


Sistem Informasi Mahasiswa
Unable to load asset: images/logo-ftik.png
NIM
Password
Login
Daftar Akun Mahasiswa

```
Image provider: AssetImage(bundle
Image key: AssetBundleImageKey(b                          : "images/logo-ftik.png",
  scale: 1)
```

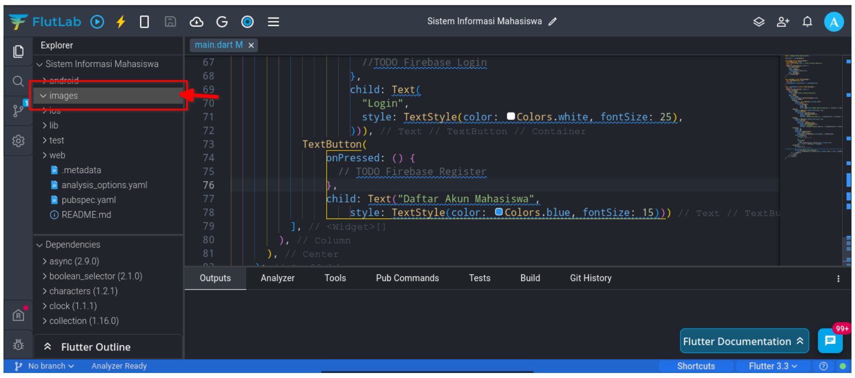
17. Tetapi banyak error yang ada di aplikasi tersebut seperti:
    - Gambar tidak ada
    - Tampilan rusak ketika dirotasi landscape



```
                ], // <Widget>[]
              ), // Column
            ), // Center
```

```
The following assertion was thrown resolving an image codec:
Unable to load asset: images/logo-ftik.png

When the exception was thrown, this was the stack:
dart:sdk_internal 7516:7              DartError
```

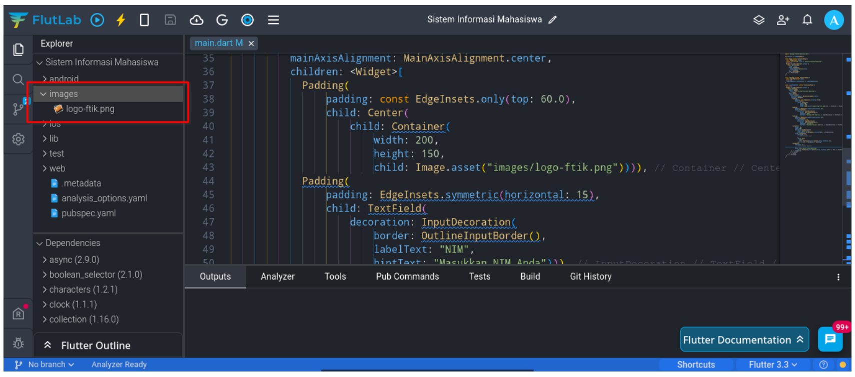
18. Untuk menambahkan gambar cukup dengan cara mudah. Buat **Folder Baru** di **Projek Sistem Informasi Mahasiswa** dengan nama **images**



```
                      //TODO Firebase Login
                    },
                    child: Text(
                      "Login",
                      style: TextStyle(color: Colors.white, fontSize: 25),
                    ))), // Text // TextButton // Container
                TextButton(
                  onPressed: () {
                    // TODO Firebase Register
                  },
                  child: Text("Daftar Akun Mahasiswa",
                      style: TextStyle(color: Colors.blue, fontSize: 15))) // Text // TextBu
                ], // <Widget>[]
              ), // Column
            ), // Center
```

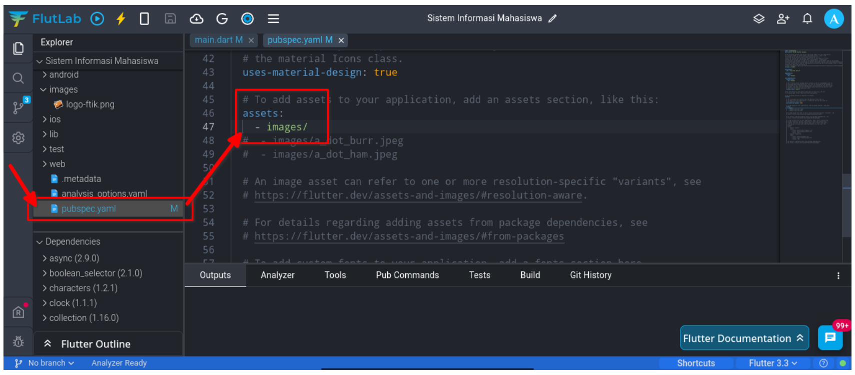
19. Lalu unduh gambar dan rename ke **logo-ftik.png**. Unduh di
    - https://transit.ftik.usm.ac.id/uploads/assets/img/ftik-usm.png

20. Unggah file yang sudah di-rename ke folder **images** tersebut



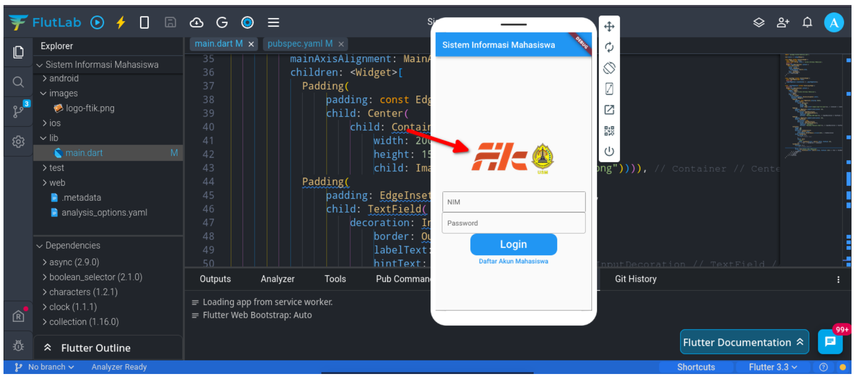
```
              mainAxisAlignment: MainAxisAlignment.center,
              children: <Widget>[
                Padding(
                    padding: const EdgeInsets.only(top: 60.0),
                    child: Center(
                        child: Container(
                            width: 200,
                            height: 150,
                            child: Image.asset("images/logo-ftik.png")))), // Container // Cente
                Padding(
                    padding: EdgeInsets.symmetric(horizontal: 15),
                    child: TextField(
                        decoration: InputDecoration(
                            border: OutlineInputBorder(),
                            labelText: "NIM",
                            hintText: "Masukkan NIM Anda"))), // InputDecoration // TextField /
```

21. Berikutnya adalah mengkonfigurasi **pubspec.yml** agar bisa membaca gambar, **Uncomment assets** dan tambahkan **- images**. Lihat Gambar di bawah:



```
# the material Icons class.
uses-material-design: true

# To add assets to your application, add an assets section, like this:
assets:
  - images/
#  - images/a_dot_burr.jpeg
#  - images/a_dot_ham.jpeg

# An image asset can refer to one or more resolution-specific "variants", see
# https://flutter.dev/assets-and-images/#resolution-aware.

# For details regarding adding assets from package dependencies, see
# https://flutter.dev/assets-and-images/#from-packages
```

22. **Hot Reload** aplikasi untuk menampilkan gambar. Jika tidak muncul, pastikan dilakukan dengan benar.

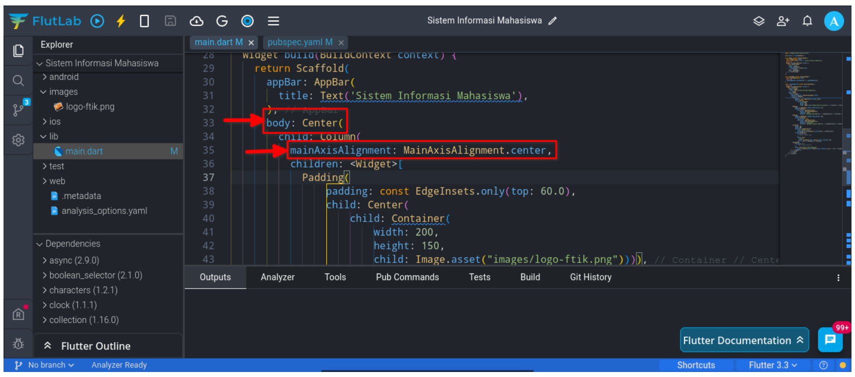


```
mainAxisAlignment: MainA
children: <Widget>[
  Padding(
      padding: const Edg
      child: Center(
          child: Contai
              width: 20
              height: 15
              child: Ima
  Padding(
      padding: EdgeInset
      child: TextField(
          decoration: In
              border: Ou
              labelText:
              hintText:
```

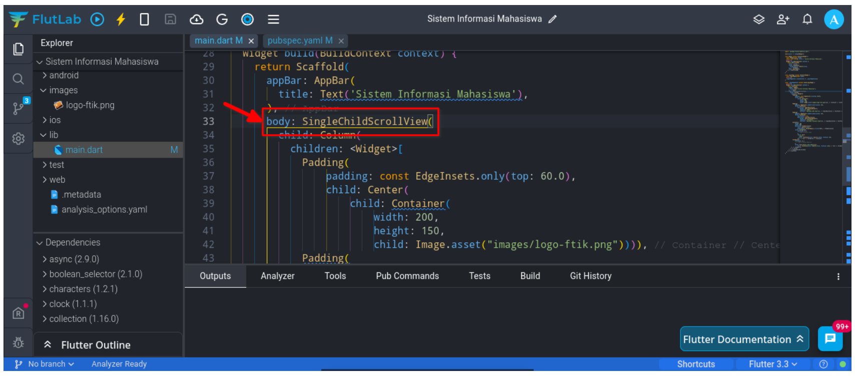
```
Loading app from service worker.
Flutter Web Bootstrap: Auto
```

23. Langkah terakhir adalah mengubah layout agar memungkinkan untuk di **Scroll**. Perhatikan berikut:

    - Ubah **body: Center** → menjadi **body: SingleChildScrollView**
    - Hapus **mainAxisAlignment: MainAxisAlignment.center,**

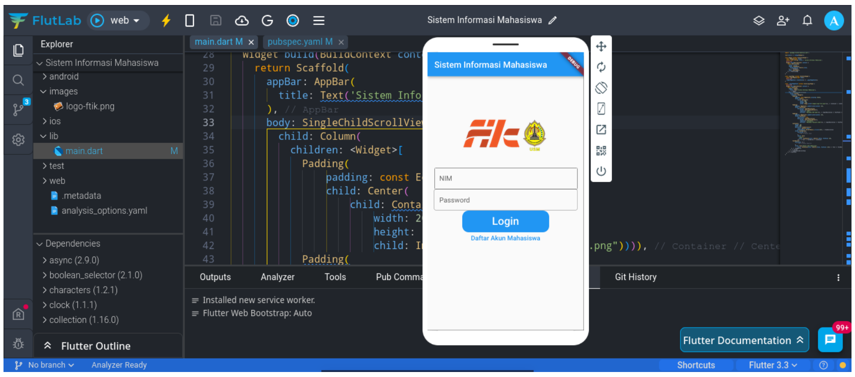


```
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Sistem Informasi Mahasiswa'),
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Padding(
                padding: const EdgeInsets.only(top: 60.0),
                child: Center(
                    child: Container(
                        width: 200,
                        height: 150,
                        child: Image.asset("images/logo-ftik.png"))), // Container // Cente
```



```dart
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Sistem Informasi Mahasiswa'),
      ), // AppBar
      body: SingleChildScrollView(
        child: Column(
          children: <Widget>[
            Padding(
                padding: const EdgeInsets.only(top: 60.0),
                child: Center(
                    child: Container(
                        width: 200,
                        height: 150,
                        child: Image.asset("images/logo-ftik.png")))), // Container // Cente
            Padding(
```

24. Build Ulang Aplikasi untuk melihat hasil akhirnya



```
Installed new service worker.
Flutter Web Bootstrap: Auto
```

25. Aplikasi yang sudah jadi kemudian di **Screenshot** dan kirimkan ke **E-Learning**

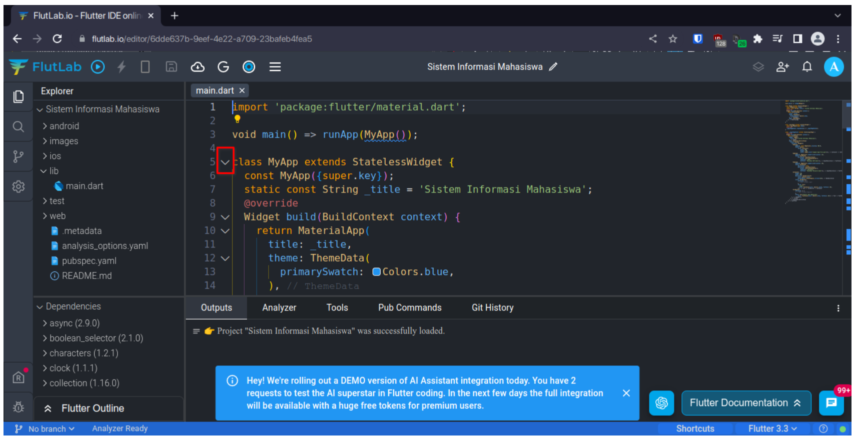
Bab 3

Praktikum 3

3.1 Halaman Registrasi dan Navigasi Page

3.2 Tutorial

1. Buka projek yang sebelumnya sudah dibuat
2. Sebelum memulai masuk ke pemrograman, lipat dulu **class** yang ada untuk mempermudah melakukan pemrograman. Klik **Tombol** yang ditunjukkan oleh Gambar berikut

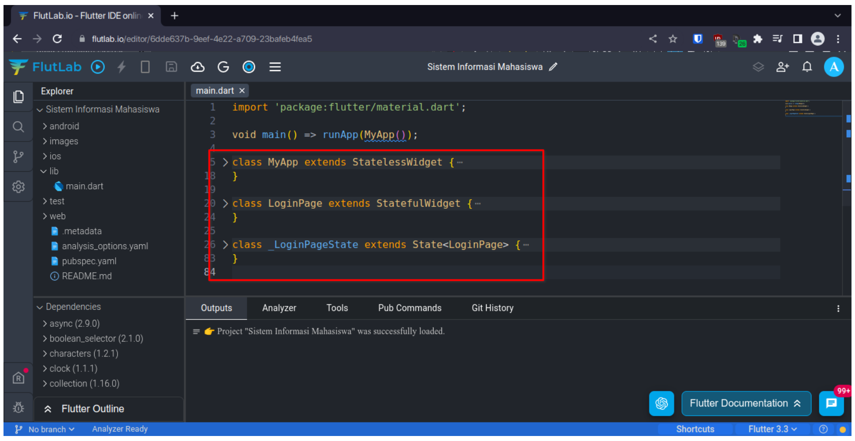


```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ), // ThemeData
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

Hey! We're rolling out a DEMO version of AI Assistant integration today. You have 2 requests to test the AI superstar in Flutter coding. In the next few days the full integration will be available with a huge free tokens for premium users.

3. Lipat semua **class** yang ada di kode agar terlihat seperti berikut



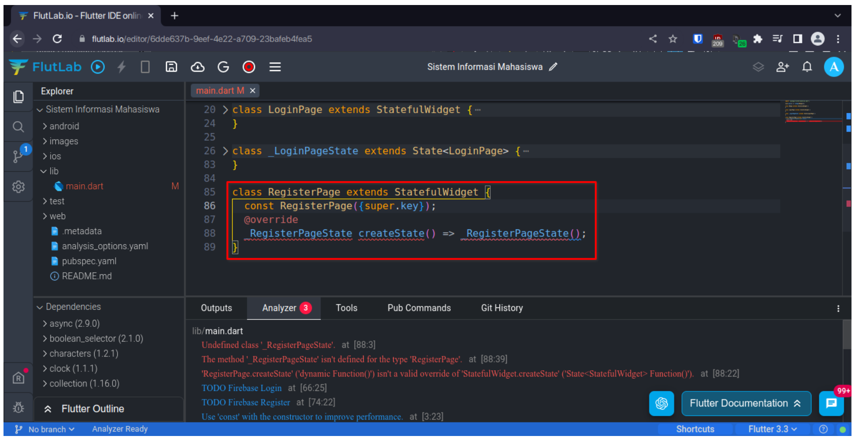
```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {…
}

class LoginPage extends StatefulWidget {…
}

class _LoginPageState extends State<LoginPage> {…
}
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

4. Berikutnya masukkan kode **class** untuk pembuatan tampilan **Pendaftaran** seperti berikut:

Potongan Kode

```
class RegisterPage extends StatefulWidget {
  const RegisterPage({super.key});
  @override
  _RegisterPageState createState() => _RegisterPageState();
}
```



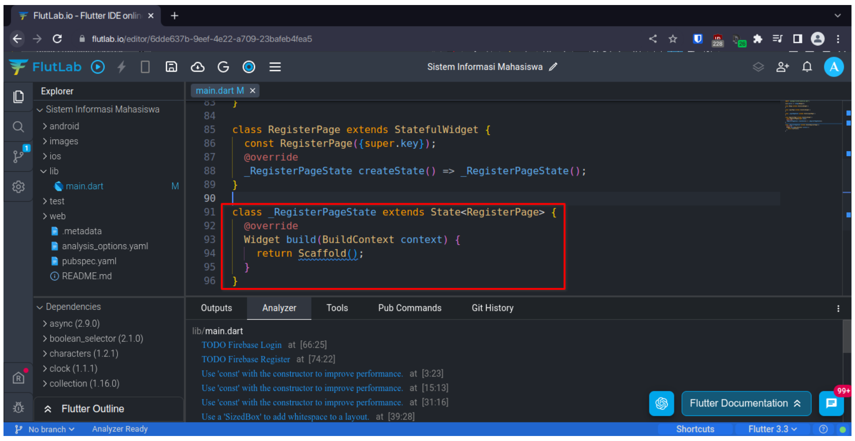
```
class LoginPage extends StatefulWidget {…
}

class _LoginPageState extends State<LoginPage> {…
}

class RegisterPage extends StatefulWidget {
  const RegisterPage({super.key});
  @override
  _RegisterPageState createState() => _RegisterPageState();
}
```

```
lib/main.dart
Undefined class '_RegisterPageState'.  at [88:3]
The method '_RegisterPageState' isn't defined for the type 'RegisterPage'.  at [88:39]
'RegisterPage.createState' ('dynamic Function()') isn't a valid override of 'StatefulWidget.createState' ('State<StatefulWidget> Function()').  at [88:22]
TODO Firebase Login  at [66:25]
TODO Firebase Register  at [74:22]
Use 'const' with the constructor to improve performance.  at [3:23]
```

5. Abaikan **Error** saat ini. Untuk menghilangkan **Error** itu, masukkan kode berikut:

Potongan Kode

```
class _RegisterPageState extends State<RegisterPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold();
  }
}
```



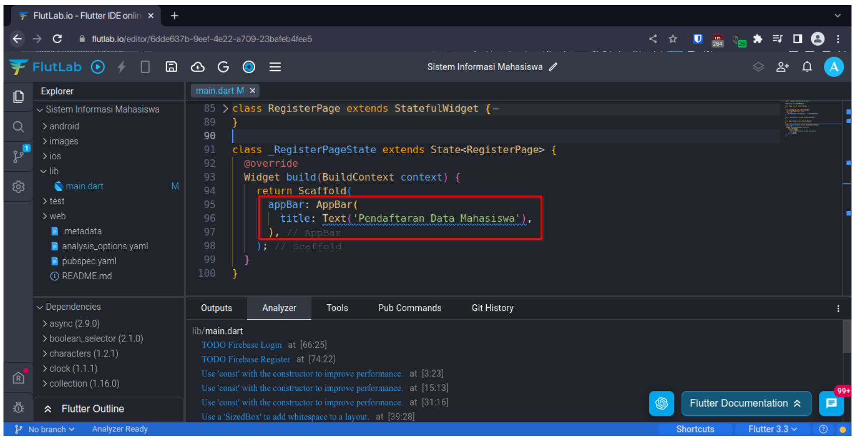
```
class RegisterPage extends StatefulWidget {
  const RegisterPage({super.key});
  @override
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold();
  }
}
```

```
lib/main.dart
  TODO Firebase Login  at [66:25]
  TODO Firebase Register  at [74:22]
  Use 'const' with the constructor to improve performance.  at [3:23]
  Use 'const' with the constructor to improve performance.  at [15:13]
  Use 'const' with the constructor to improve performance.  at [31:16]
  Use a 'SizedBox' to add whitespace to a layout.  at [39:28]
```

6. Kode **widget-widget** akan dimasukkan ke dalam **return Scaffold( )**. Pertama masukkan **appBar** yang digunakan untuk menampilakn **Judul Halaman** seperti berikut:

Potongan Kode

```
appBar: AppBar(
  title: Text('Pendaftaran Data Mahasiswa'),
),
```



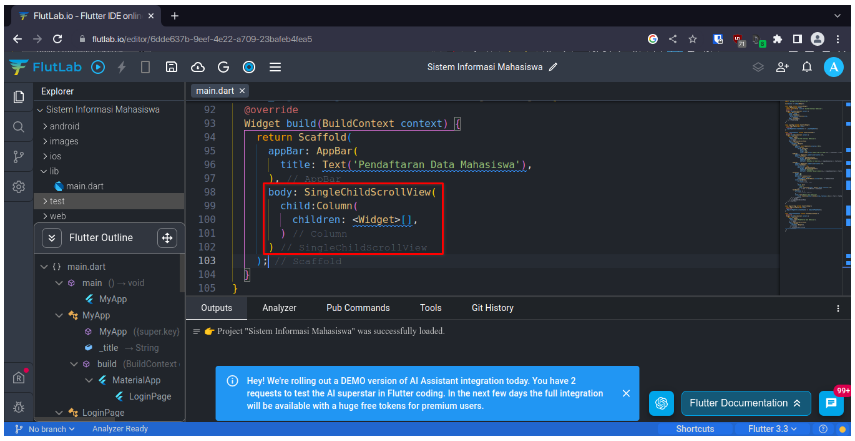
```
class RegisterPage extends StatefulWidget {…
}

class _RegisterPageState extends State<RegisterPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Pendaftaran Data Mahasiswa'),
      ), // AppBar
    ); // Scaffold
  }
}
```

```
lib/main.dart
  TODO Firebase Login  at [66:25]
  TODO Firebase Register  at [74:22]
  Use 'const' with the constructor to improve performance.  at [3:23]
  Use 'const' with the constructor to improve performance.  at [15:13]
  Use 'const' with the constructor to improve performance.  at [31:16]
  Use a 'SizedBox' to add whitespace to a layout.  at [39:28]
```

7. Berikutnya adalah menambahkan widget layout yang dapat di Scroll beserta wadah **Widget** nantinya. Perhatikan Kode dan Gambar beirkut:

Potongan Kode

```
body: SingleChildScrollView(
  child:Column(
    children: <Widget>[],
  )
)
```

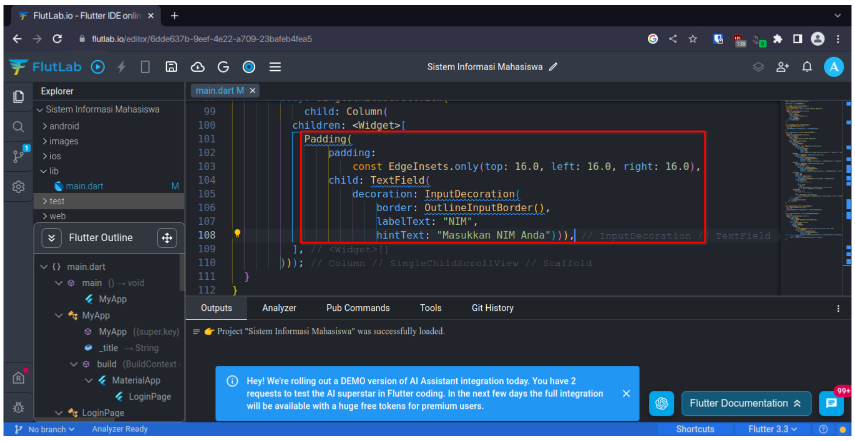


```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Pendaftaran Data Mahasiswa'),
    ), // AppBar
    body: SingleChildScrollView(
      child:Column(
        children: <Widget>[],
      ) // Column
    ) // SingleChildScrollView
  ); // Scaffold
}
}
```

8. Sama seperti sebelumnya yang di mana kode **Widget** diletakkan di dalam kode **<Widget>[ ]**. Tambahkan kode **InputText** untuk **NIM** seperti berikut:

Potongan Kode

```
Padding(
  padding:
    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
  child: TextField(
    decoration: InputDecoration(
      border: OutlineInputBorder(),
      labelText: "NIM",
      hintText: "Masukkan NIM Anda"))),
```

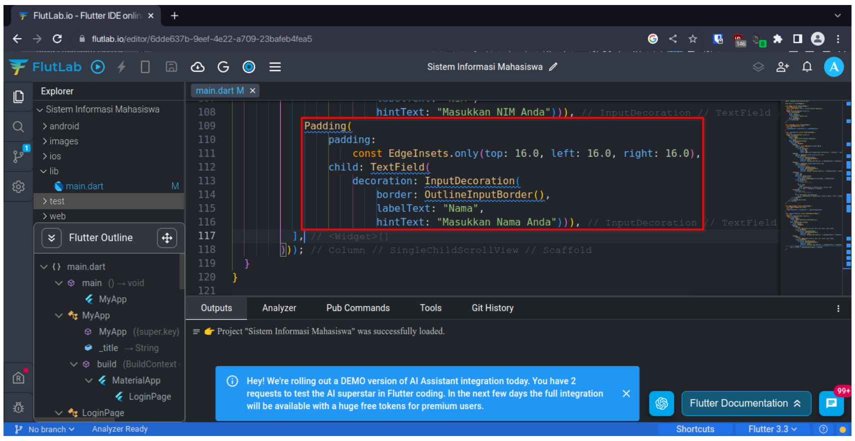


```
      child: Column(
    children: <Widget>[
      Padding(
          padding:
              const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
          child: TextField(
              decoration: InputDecoration(
                  border: OutlineInputBorder(),
                  labelText: "NIM",
                  hintText: "Masukkan NIM Anda"))), // InputDecoration // TextField
    ], // <Widget>[]
  ))); // Column // SingleChildScrollView // Scaffold
  }
}
```

9. Kode berikutnya untuk **InputText** dengan isi **Nama**

Potongan Kode

```
Padding(
  padding:
    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
  child: TextField(
    decoration: InputDecoration(
      border: OutlineInputBorder(),
      labelText: "Nama",
      hintText: "Masukkan Nama Anda"))),
```



```
                        hintText: "Masukkan NIM Anda"))), // InputDecoration // TextField
                  Padding(
                    padding:
                      const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                    child: TextField(
                      decoration: InputDecoration(
                        border: OutlineInputBorder(),
                        labelText: "Nama",
                        hintText: "Masukkan Nama Anda"))), // InputDecoration // TextField
                ], // <Widget>[]
              ))); // Column // SingleChildScrollView // Scaffold
    }
  }
```

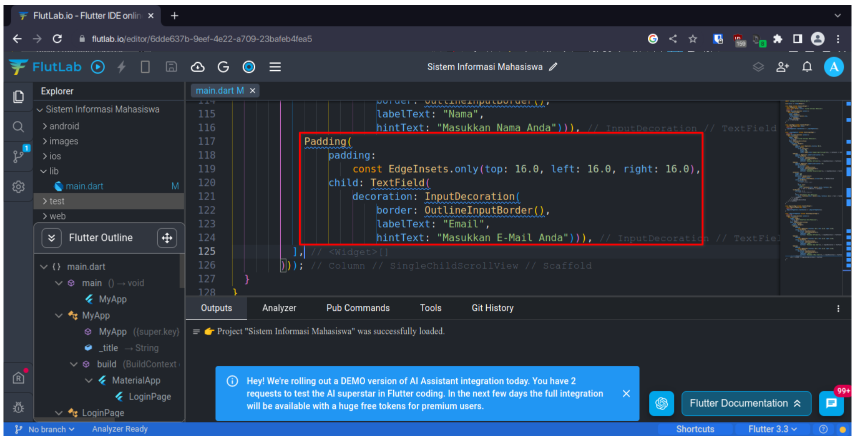
```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

Hey! We're rolling out a DEMO version of AI Assistant integration today. You have 2 requests to test the AI superstar in Flutter coding. In the next few days the full integration will be available with a huge free tokens for premium users.

10. Selanjutnya adalah **InputText** untuk **E-Mail**. Perhatikan kode berikut:

Potongan Kode

```
Padding(
  padding:
    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
  child: TextField(
    decoration: InputDecoration(
      border: OutlineInputBorder(),
      labelText: "Email",
      hintText: "Masukkan E-Mail Anda"))),
```



```
                        labelText: "Nama",
                        hintText: "Masukkan Nama Anda"))), // InputDecoration // TextField
                  Padding(
                    padding:
                      const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                    child: TextField(
                      decoration: InputDecoration(
                        border: OutlineInputBorder(),
                        labelText: "Email",
                        hintText: "Masukkan E-Mail Anda"))), // InputDecoration // TextField
                ], // <Widget>[]
              ))); // Column // SingleChildScrollView // Scaffold
    }
  }
```

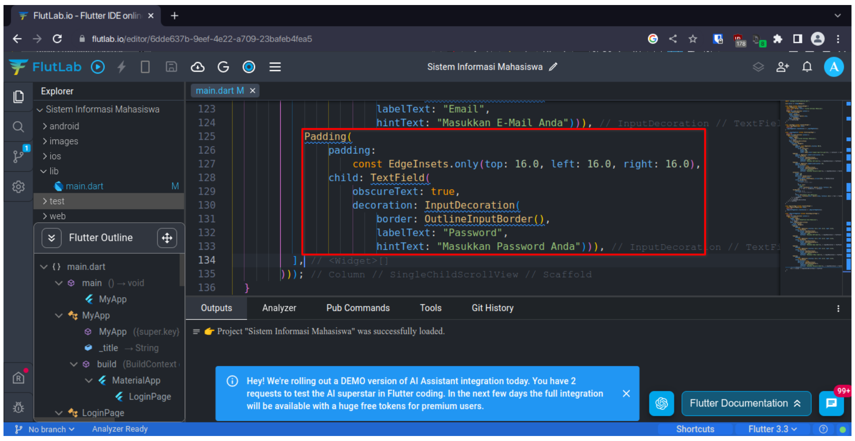
```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

Hey! We're rolling out a DEMO version of AI Assistant integration today. You have 2 requests to test the AI superstar in Flutter coding. In the next few days the full integration will be available with a huge free tokens for premium users.

11. **InputText** terakhir adalah untuk **Password**. Masukkan kode berikut:

Potongan Kode

```
Padding(
  padding:
    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
  child: TextField(
    obscureText: true,
    decoration: InputDecoration(
      border: OutlineInputBorder(),
      labelText: "Password",
      hintText: "Masukkan Password Anda"))),
```



```
labelText: "Email",
hintText: "Masukkan E-Mail Anda"))), // InputDecoration // TextFiel
Padding(
  padding:
    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
  child: TextField(
    obscureText: true,
    decoration: InputDecoration(
      border: OutlineInputBorder(),
      labelText: "Password",
      hintText: "Masukkan Password Anda"))), // InputDecoration // TextFi
], // <Widget>[]
))); // Column // SingleChildScrollView // Scaffold
}
```

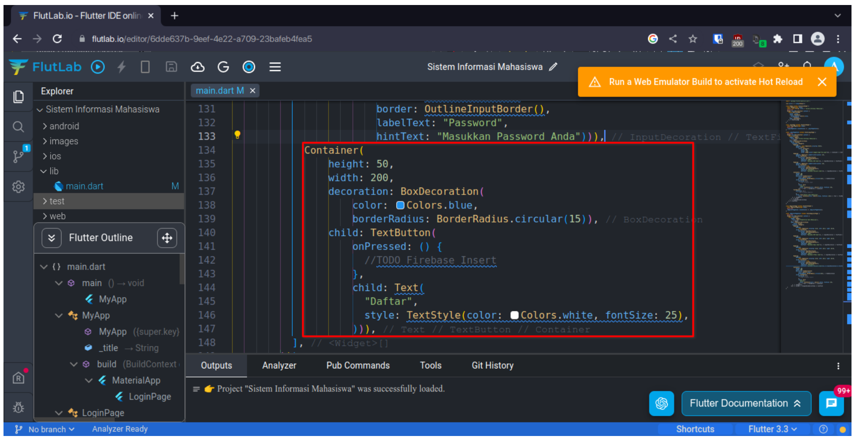
Project "Sistem Informasi Mahasiswa" was successfully loaded.

Hey! We're rolling out a DEMO version of AI Assistant integration today. You have 2 requests to test the AI superstar in Flutter coding. In the next few days the full integration will be available with a huge free tokens for premium users.

12. **Widget** berikutnya adalah **Tombol**. Masukkan kode berikut untuk menambahkan tombol **Daftar**

Potongan Kode

```
Container(
  height: 50,
  width: 200,
  decoration: BoxDecoration(
    color: Colors.blue,
    borderRadius: BorderRadius.circular(15)),
    child: TextButton(
      onPressed: () {
        //TODO Firebase Insert
      },
      child: Text(
        "Daftar",
        style: TextStyle(color: Colors.white, fontSize: 25),
))),
```



Run a Web Emulator Build to activate Hot Reload

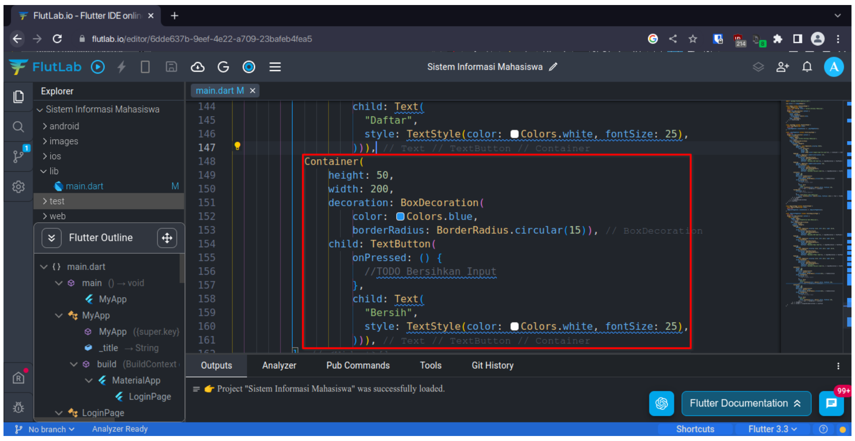
```
border: OutlineInputBorder(),
labelText: "Password",
hintText: "Masukkan Password Anda"))), // InputDecoration // TextFi
Container(
  height: 50,
  width: 200,
  decoration: BoxDecoration(
    color: Colors.blue,
    borderRadius: BorderRadius.circular(15)), // BoxDecoration
  child: TextButton(
    onPressed: () {
      //TODO Firebase Insert
    },
    child: Text(
      "Daftar",
      style: TextStyle(color: Colors.white, fontSize: 25),
    ))), // Text // TextButton // Container
], // <Widget>[]
```

Project "Sistem Informasi Mahasiswa" was successfully loaded.

13. Tombol berikutnya adalah **Bersih** untuk menghapus input dari **InputText**

Potongan Kode

```
Container(
  height: 50,
  width: 200,
  decoration: BoxDecoration(
    color: Colors.blue,
    borderRadius: BorderRadius.circular(15)),
    child: TextButton(
      onPressed: () {
        //TODO Bersihkan Input
      },
      child: Text(
        "Bersih",
        style: TextStyle(color: Colors.white, fontSize: 25),
))),
```



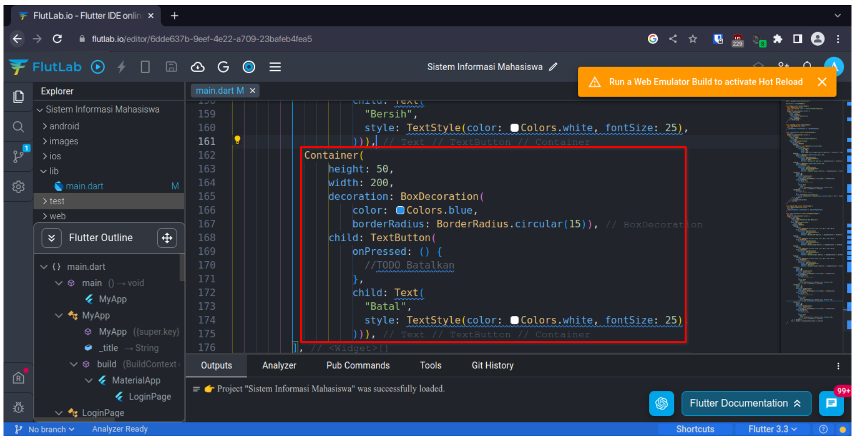
```
        child: Text(
          "Daftar",
          style: TextStyle(color: Colors.white, fontSize: 25),
        ))), // Text // TextButton // Container
  Container(
    height: 50,
    width: 200,
    decoration: BoxDecoration(
      color: Colors.blue,
      borderRadius: BorderRadius.circular(15)), // BoxDecoration
    child: TextButton(
      onPressed: () {
        //TODO Bersihkan Input
      },
      child: Text(
        "Bersih",
        style: TextStyle(color: Colors.white, fontSize: 25),
      ))), // Text // TextButton // Container
```

Project "Sistem Informasi Mahasiswa" was successfully loaded.

14. Tombol terakhir adalah **Batal** yang digunakan untuk kembali ke **Menu Awal (Login)**. Masukkan kode berikut:

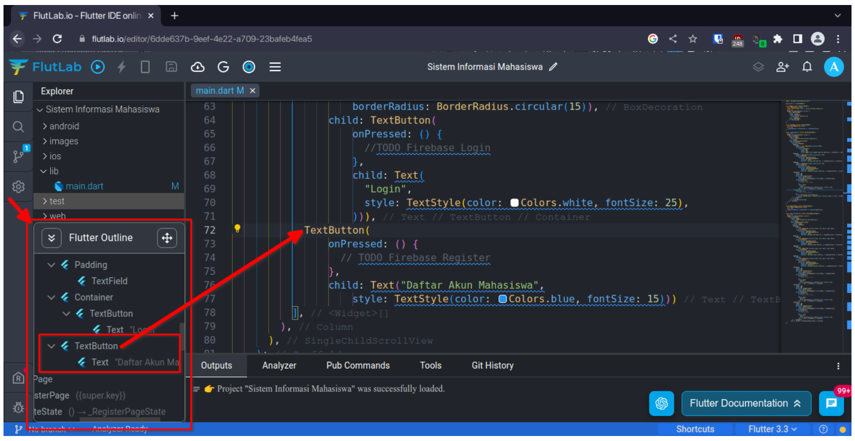
Potongan Kode

```
Container(
  height: 50,
  width: 200,
  decoration: BoxDecoration(
    color: Colors.blue,
    borderRadius: BorderRadius.circular(15)),
    child: TextButton(
      onPressed: () {
        //TODO Batalkan Registrasi
      },
      child: Text(
        "Batal",
        style: TextStyle(color: Colors.white, fontSize: 25),
))),
```



```
                  "Bersih",
                  style: TextStyle(color: Colors.white, fontSize: 25),
                ))), // Text // TextButton // Container
            Container(
              height: 50,
              width: 200,
              decoration: BoxDecoration(
                color: Colors.blue,
                borderRadius: BorderRadius.circular(15)), // BoxDecoration
              child: TextButton(
                onPressed: () {
                  //TODO Batalkan
                },
                child: Text(
                  "Batal",
                  style: TextStyle(color: Colors.white, fontSize: 25)
                ))), // Text // TextButton // Container
          ], // <Widget>[]
```

15. Berikutnya adalah memberi fitur **Navigasi Halaman** agar aplikasi dapat berpindah ke halaman lainnya. Kembali ke bagian **_LoginPage** dan temukan **TextButton** registrasinya. Gunakan **Flutter Outline** di sebelah **Kiri Bawah** untuk mempermudah pencarian, dan **Klik** target tujuan:

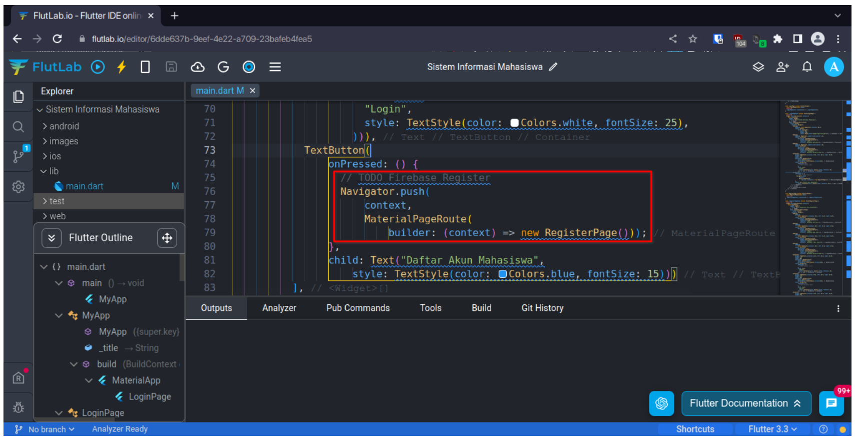


```
                borderRadius: BorderRadius.circular(15)), // BoxDecoration
              child: TextButton(
                onPressed: () {
                  //TODO Firebase Login
                },
                child: Text(
                  "Login",
                  style: TextStyle(color: Colors.white, fontSize: 25),
                ))), // Text // TextButton // Container
            TextButton(
              onPressed: () {
                // TODO Firebase Register
              },
              child: Text("Daftar Akun Mahasiswa",
                style: TextStyle(color: Colors.blue, fontSize: 15))) // Text // TextB
          ], // <Widget>[]
        ), // Column
      ), // SingleChildScrollView
```

16. Di bagian dalam **onPressed:()**, masukkan kode berikut setelah **Komentar TODO**:

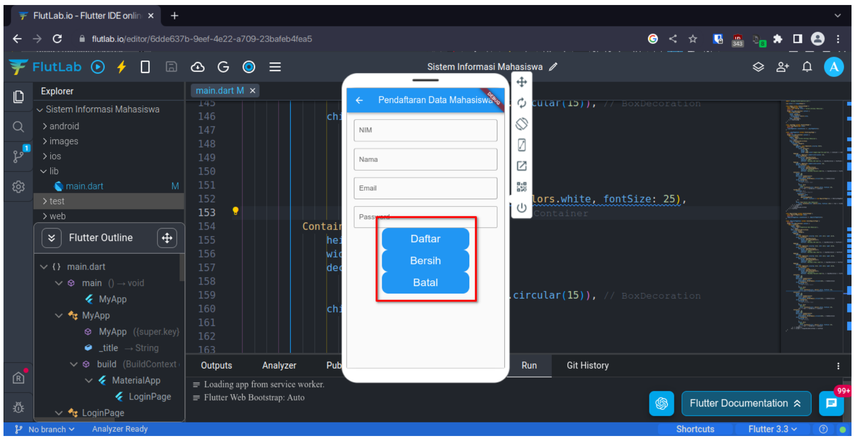
Potongan Kode

```
Navigator.push(
  context,
  MaterialPageRoute(
    builder: (context) => new RegisterPage()));
```



```
                  "Login",
                  style: TextStyle(color: Colors.white, fontSize: 25),
                ))), // Text // TextButton // Container
            TextButton(
              onPressed: () {
                // TODO Firebase Register
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => new RegisterPage())); // MaterialPageRoute
              },
              child: Text("Daftar Akun Mahasiswa",
                style: TextStyle(color: Colors.blue, fontSize: 15))) // Text // TextE
          ], // <Widget>[]
```

17. **Build** app yang sudah dibuat dan cek hasilnya

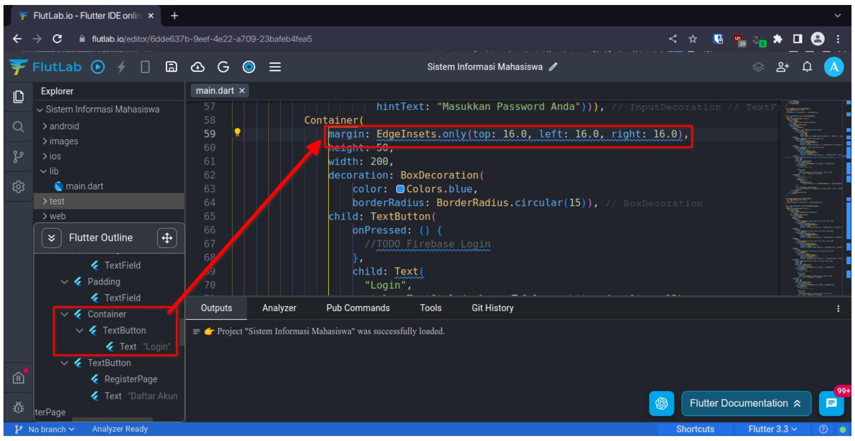


```
Loading app from service worker.
Flutter Web Bootstrap: Auto
```

18. Karena **Tombol** terlalu dekat satu sama lainnya, maka tambahkan kode berikut ke semua **Tombol** yang ada:

Potongan Kode
```
margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
```

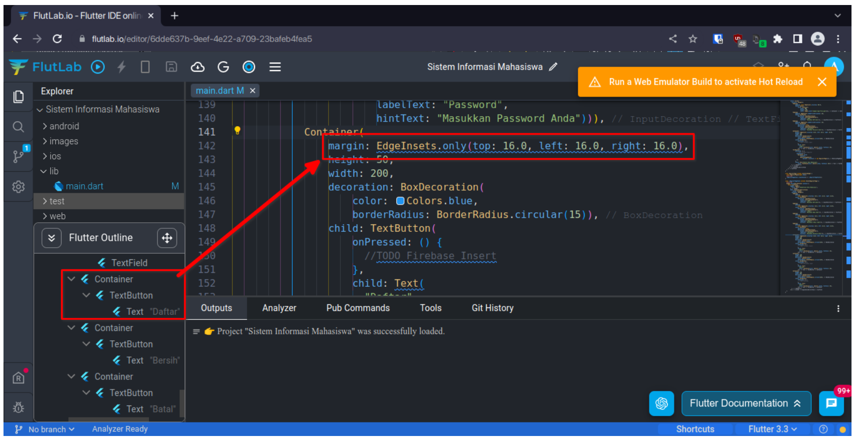
- Tombol Login



```
                hintText: "Masukkan Password Anda"))), // InputDecoration // TextFi
          Container(
            margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
            height: 50,
            width: 200,
            decoration: BoxDecoration(
                color: Colors.blue,
                borderRadius: BorderRadius.circular(15)), // BoxDecoration
            child: TextButton(
                onPressed: () {
                  //TODO Firebase Login
                },
                child: Text(
                  "Login",
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

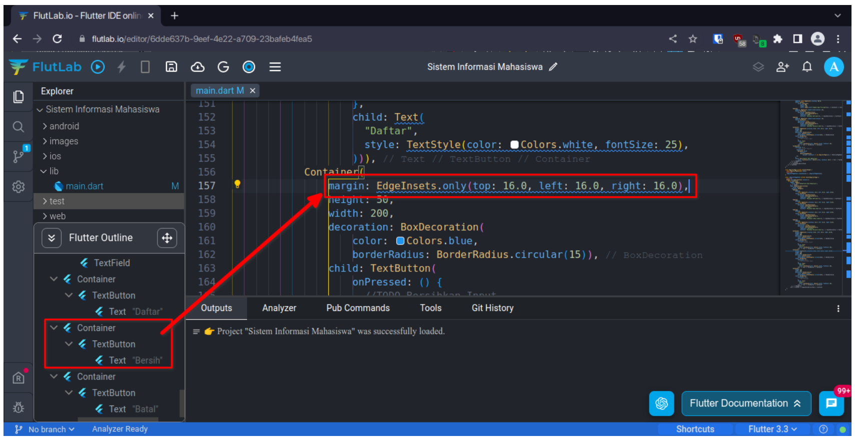
- Tombol Daftar



```
                labelText: "Password",
                hintText: "Masukkan Password Anda"))), // InputDecoration // TextFi
          Container(
            margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
            height: 50,
            width: 200,
            decoration: BoxDecoration(
                color: Colors.blue,
                borderRadius: BorderRadius.circular(15)), // BoxDecoration
            child: TextButton(
                onPressed: () {
                  //TODO Firebase Insert
                },
                child: Text(
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

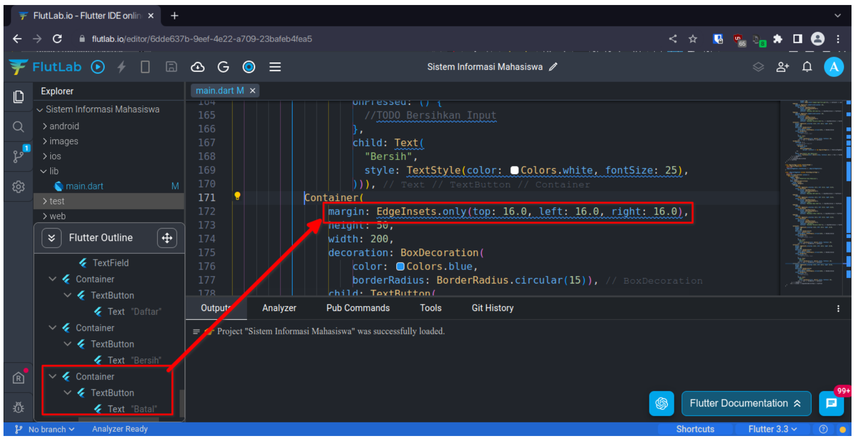
- Tombol Bersih



```
                    child: Text(
                      "Daftar",
                      style: TextStyle(color: Colors.white, fontSize: 25),
                    ))), // Text // TextButton // Container
                Container(
                  margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                  height: 50,
                  width: 200,
                  decoration: BoxDecoration(
                      color: Colors.blue,
                      borderRadius: BorderRadius.circular(15)), // BoxDecoration
                  child: TextButton(
                      onPressed: () {
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

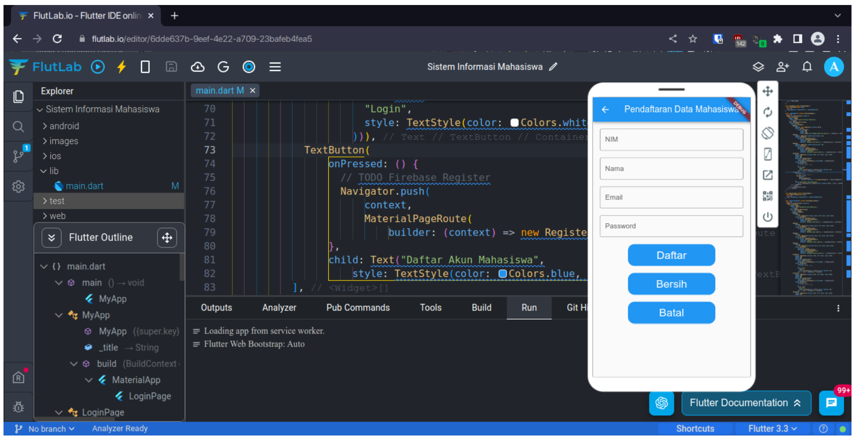
- Tombol Batal



```
                        //TODO Bersihkan Input
                      },
                      child: Text(
                        "Bersih",
                        style: TextStyle(color: Colors.white, fontSize: 25),
                      ))), // Text // TextButton // Container
                Container(
                  margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                  height: 50,
                  width: 200,
                  decoration: BoxDecoration(
                      color: Colors.blue,
                      borderRadius: BorderRadius.circular(15)), // BoxDecoration
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

19. Cek kembali hasilnya dan **Screenshot** hasil aplikasi saat ini dan kirimkan ke **E-Learning**



```
                  "Login",
                  style: TextStyle(color: Colors.whit
                ))), // Text // TextButton // Containe
            TextButton(
                onPressed: () {
                  // TODO Firebase Register
                  Navigator.push(
                      context,
                      MaterialPageRoute(
                          builder: (context) => new Registe
                },
                child: Text("Daftar Akun Mahasiswa",
                    style: TextStyle(color: Colors.blue,
          ], // <Widget>[]
```

```
Loading app from service worker.
Flutter Web Bootstrap: Auto
```

Bab 4

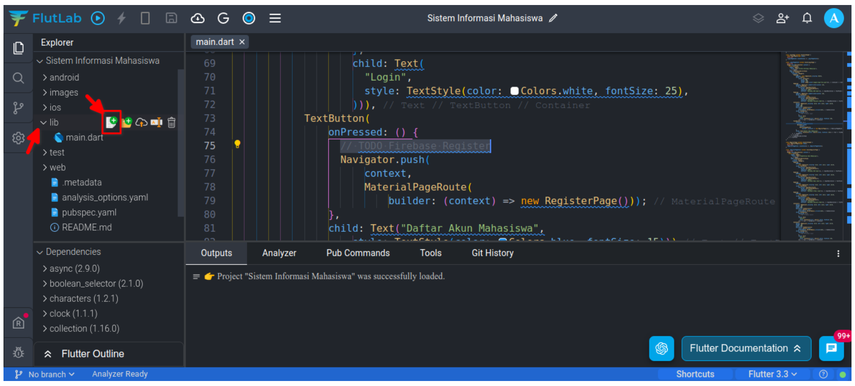
Praktikum 4

4.1 Integrasi Firebase

Di bagian ini mahasiswa diperkenalkan bagaimana melakukan pembersihan baris kode yang tidak diperlukan, integrasi ke Realtime Database, dan melakukan operasi register data. Mahasiswa diwajibkan untuk menyelesaikan Praktikum 3

4.2 Tutorial

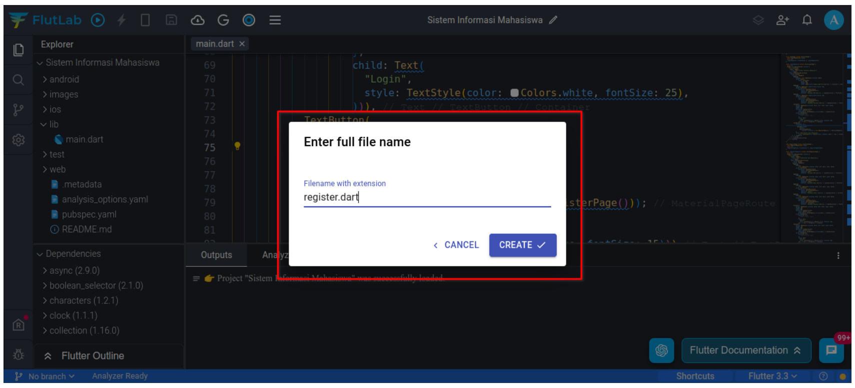
1. Buka projek sebelumnya, dan pastikan untuk membuka projek yang tepat
2. Berikutnya adalah memisahkan halaman **Register** ke file terpisah. Arahkan **Mouse** ke folder **lib**, dan klik **Create File**



```
                child: Text(
                  "Login",
                  style: TextStyle(color: Colors.white, fontSize: 25),
                ))), // Text // TextButton // Container
          TextButton(
              onPressed: () {
                // TODO Firebase Register
                Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (context) => new RegisterPage())); // MaterialPageRoute
              },
              child: Text("Daftar Akun Mahasiswa",
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

3. Beri nama **register.dart** sebagai file baru, dan klik **Create**


Enter full file name

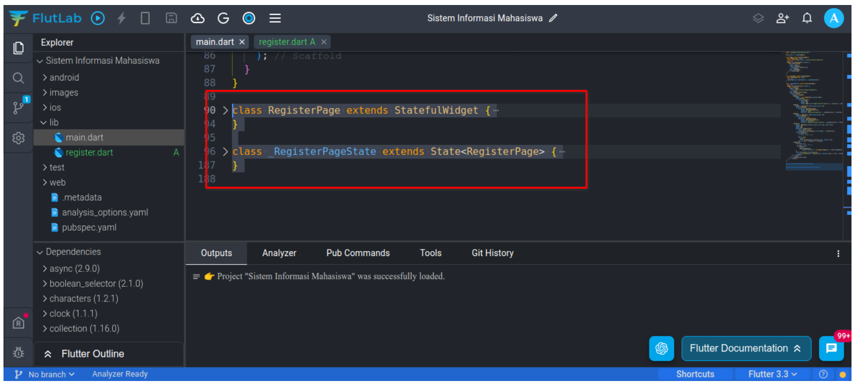
Filename with extension

register.dart

```
child: Text(
  "Login",
  style: TextStyle(color: Colors.white, fontSize: 25),
))), // Text // TextButton // Container
```

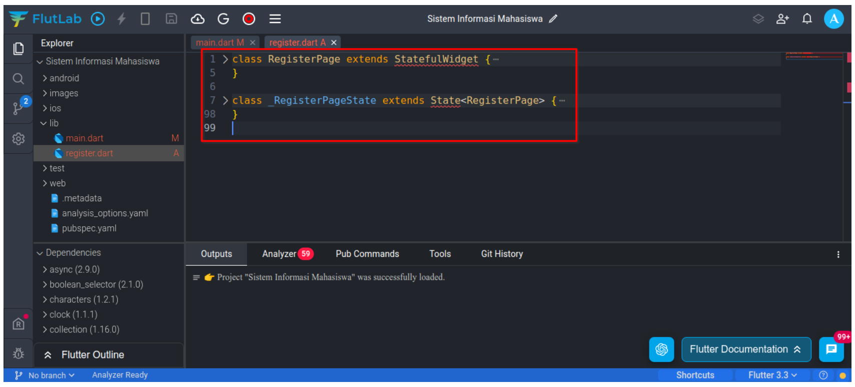
Project "Sistem Informasi Mahasiswa" was successfully loaded.

4. File **register.dart** akan terbuka, lalu **PINDAHKAN** kode berikut dari **main.dart** ke register.dart. Lipat kode untuk mempermudah pemindahan



```
    ); // Scaffold
  }
}

class RegisterPage extends StatefulWidget {⋯
}

class _RegisterPageState extends State<RegisterPage> {⋯
}
```

Project "Sistem Informasi Mahasiswa" was successfully loaded.



```
class RegisterPage extends StatefulWidget {⋯
}

class _RegisterPageState extends State<RegisterPage> {⋯
}
```

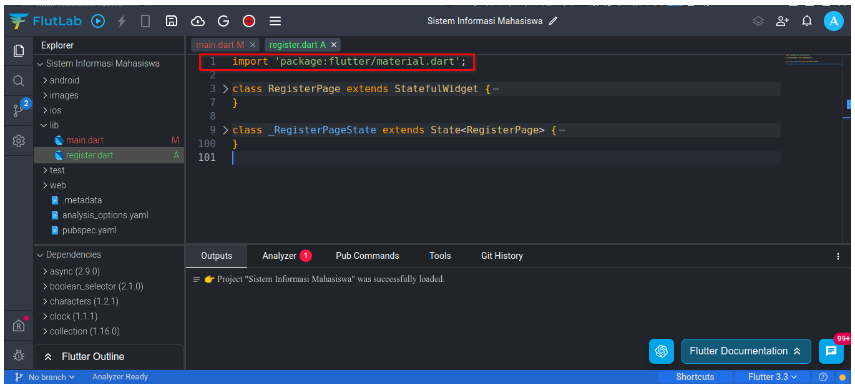
Project "Sistem Informasi Mahasiswa" was successfully loaded.

5. Pastikan **TIDAK ADA** kode **Register** di file **main.dart**. Hal ini untuk mencegah terjadinya error di pertemuan-pertemuan berikutnya.

6. Kedua file akan mengalami **Error**. Untuk memperbaiki kode **register.dart** tambahkan satu baris kode ini di **Baris 1**

Potongan Kode

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

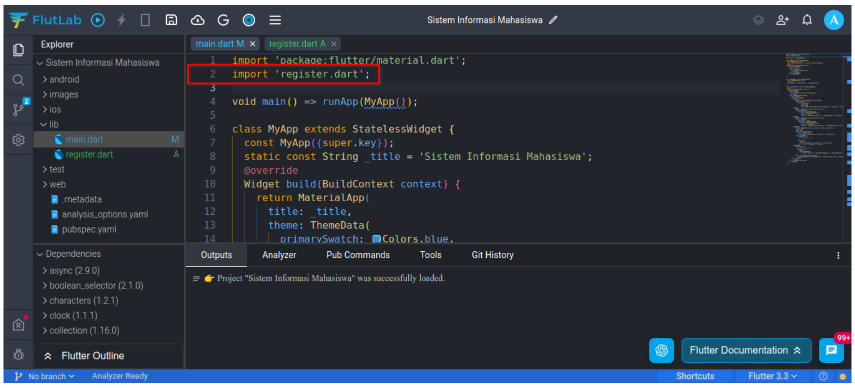


```dart
import 'package:flutter/material.dart';

class RegisterPage extends StatefulWidget {
}

class _RegisterPageState extends State<RegisterPage> {
}
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

7. Lalu untuk memperbaiki kode **main.dart** cukup tambahkan kode berikut di **Baris 2**

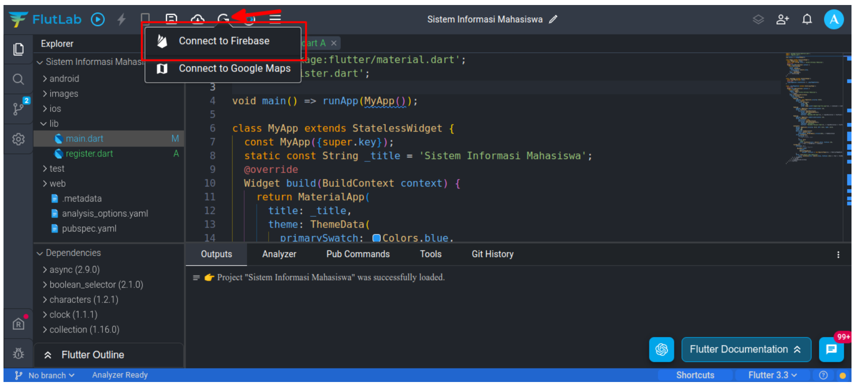
   Potongan Kode
   ```dart
   import 'register.dart';
   ```



```dart
import 'package:flutter/material.dart';
import 'register.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      theme: ThemeData(
        primarySwatch: Colors.blue,
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

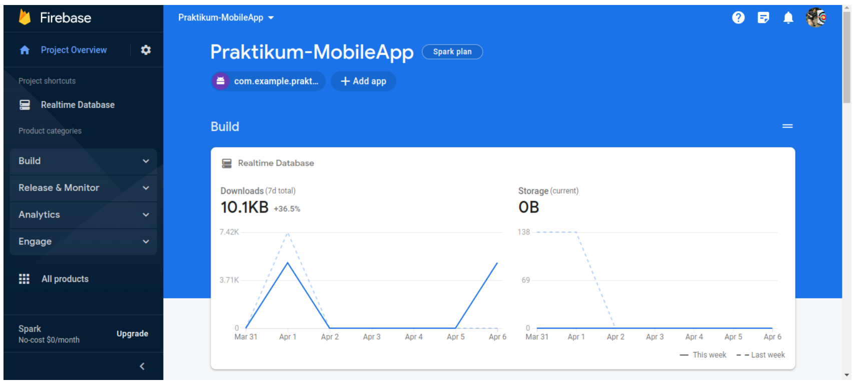
8. Semua **Error** sudah hilang, berikutnya melakukan integrasi ke **Firebase**. Klik Icon **G**, lalu pilih **Connect to Firebase**



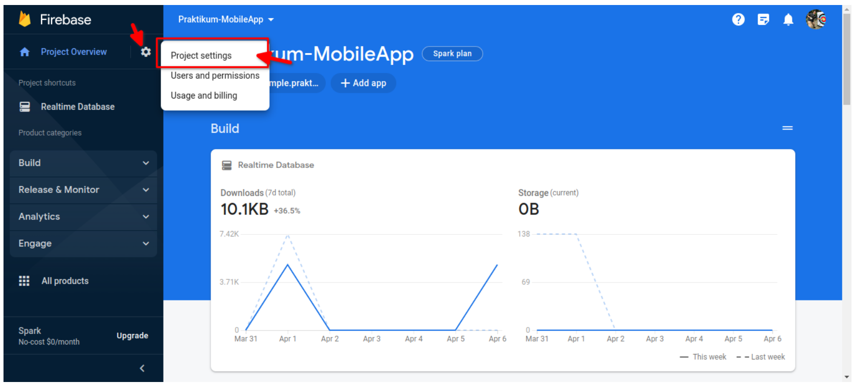
```dart
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      theme: ThemeData(
        primarySwatch: Colors.blue,
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

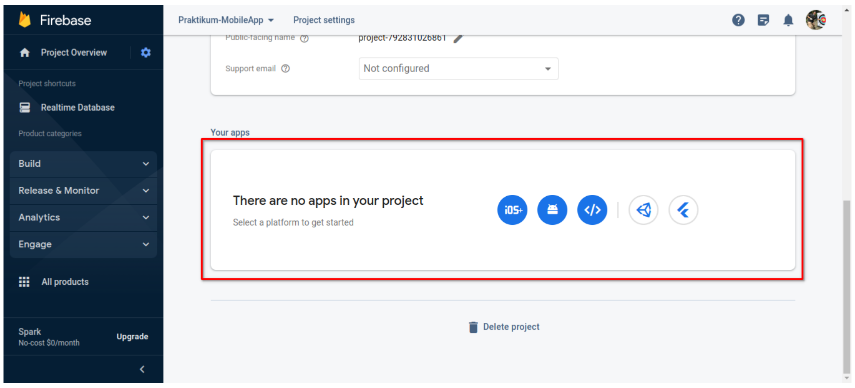
9. **Flutlab** akan menampilkan pilihan platform App yang menjadi target. Sebelum memilih buka website `https://firebase.google.com`. Login ke Google apabila belum, Klik **Go to console** di atas kanan. Pilih **Projek** yang sudah dibuat sebelumnya. Hingga tampil dasbor Firebase



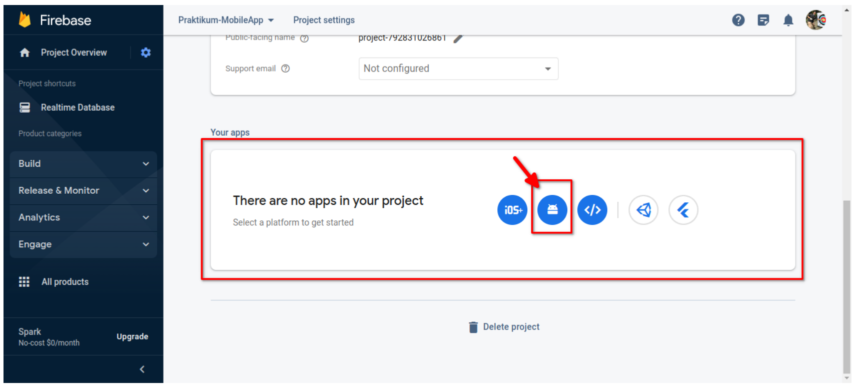
10. Untuk meng-konfigurasikan, klik **Roda Gigi** di samping **Project Overview**, lalu pilih **Project Settings**



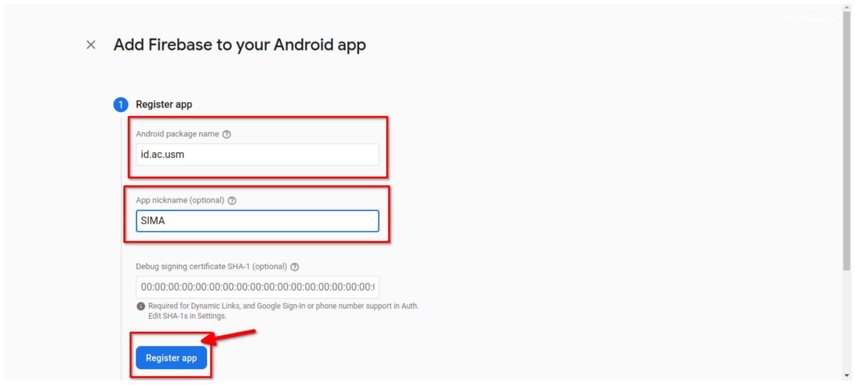
11. Scroll ke bawah dan cari panel **Your apps**



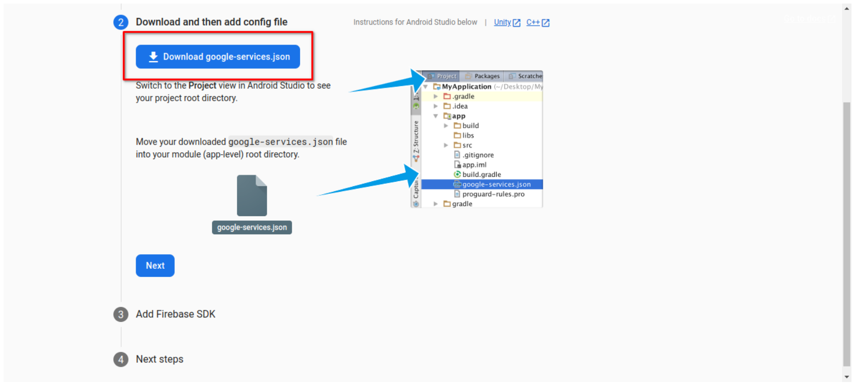
12. Pilih **Android**



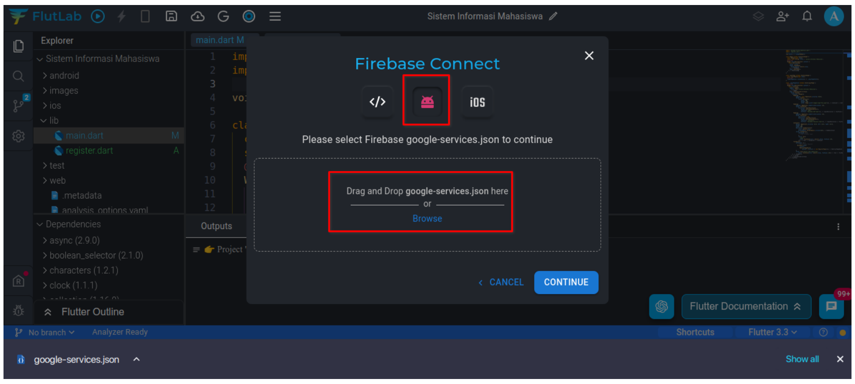
13. Lalu **Firebase** akan menampilkan **Wizard** untuk menambahkan App baru. Masukkan **id.ac.usm** sebagai **Android package name** dan **SIMA** sebagai **App nickname**. Klik **Register app**


Add Firebase to your Android app
Register app
Android package name
id.ac.usm
App nickname (optional)
SIMA
Debug signing certificate SHA-1 (optional)
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0
Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.
Register app

14. Langkah berikutnya adalah mengunduh file **google-services.json**. Cukup klik **Tombol** yang disediakan

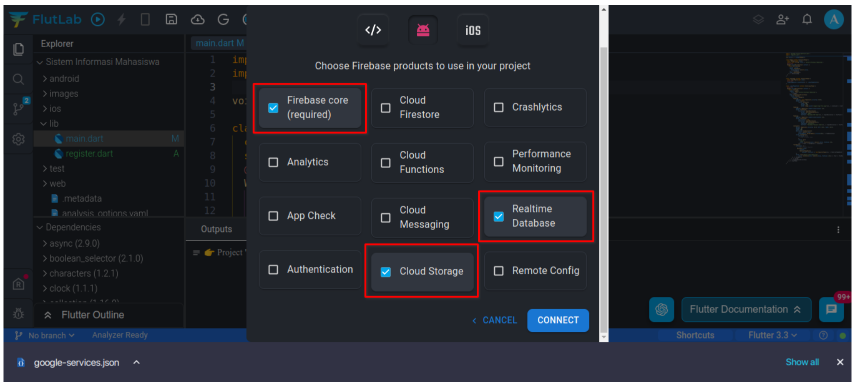

Download and then add config file
Instructions for Android Studio below | Unity C++
Download google-services.json
Switch to the Project view in Android Studio to see your project root directory.
Move your downloaded google-services.json file into your module (app-level) root directory.
google-services.json
Next
Add Firebase SDK
Next steps

15. Tutup Wizard jika sudah selesai mengunduh
16. Kembali ke **FlutLab**, pilih **Android** lalu unggah file **google-services.json**. Klik **Continue** sesudah unggah



Please select Firebase google-services.json to continue

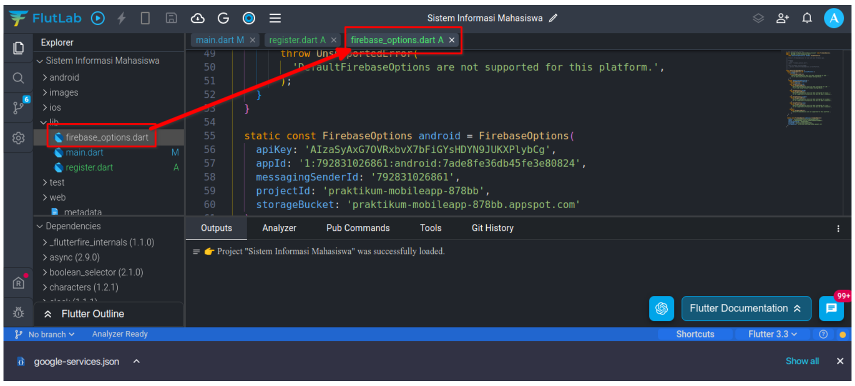
17. Berikutnya pilih **Services** yang akan dipakai. Pilih **Core**, **Realtime Database**, dan **Cloud Storage**. Terakhir klik **Connect**



Choose Firebase products to use in your project

- [✓] Firebase core (required)
- [ ] Cloud Firestore
- [ ] Crashlytics
- [ ] Analytics
- [ ] Cloud Functions
- [ ] Performance Monitoring
- [ ] App Check
- [ ] Cloud Messaging
- [✓] Realtime Database
- [ ] Authentication
- [✓] Cloud Storage
- [ ] Remote Config

18. File baru akan ditambahkan dengan nama **firebase_options.dart**. File ini menjadi bukti kalau konfigurasi dan integrasi **Firebase** sudah sukses



```
        throw UnsupportedError(
          'DefaultFirebaseOptions are not supported for this platform.',
        );
    }
  }

  static const FirebaseOptions android = FirebaseOptions(
    apiKey: 'AIzaSyAxG7OVRxbvX7bFiGYsHDYN9JUKXPlybCg',
    appId: '1:792831026861:android:7ade8fe36db45fe3e80824',
    messagingSenderId: '792831026861',
    projectId: 'praktikum-mobileapp-878bb',
    storageBucket: 'praktikum-mobileapp-878bb.appspot.com'
```

```
👉 Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

19. Untuk mengecek apakah dependency dasar sudah terpenuhi, buka file **pubspec.yml** dan cari baris teks seperti berikut:
    - **firebase_core**
    - **firebase_database**
    - **firebase_storage**

20. Jika ada yang belum terpasang di dalam projek, bisa di atasi dengan menggunakan bantuan **Pub Command** di bagian bawah. Klik **Pub Add** dan masukkan **item** di langkah 19. Satu per satu dan gunakan **Enter**
21. Jika sudah dimasukkan ke dalam **yml** maka langkah berikutnya adalah menekan tombol **Pub Get** untuk mengunduh secara otomatis.
22. **Screenshot** dan kirimkan ke **E-Learning**

Bab 5

Praktikum 5

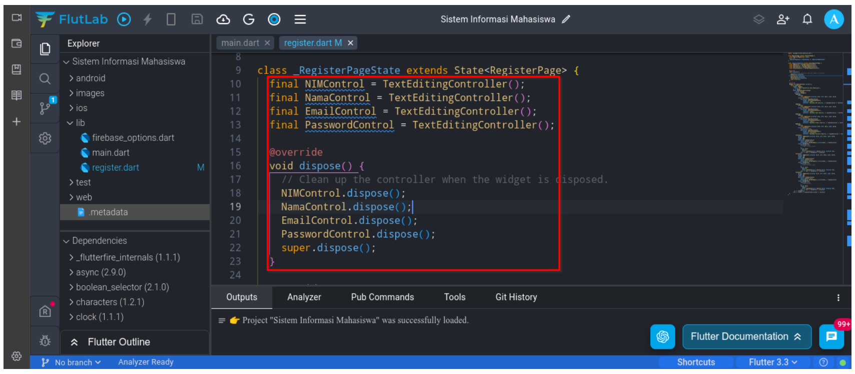
5.1 Operasi Insert Data Flutter

Di bagian ini mahasiswa diajarkan bagaimana melakukan input dan penarikan data serta melakukan query ke Firebase. Mahasiswa diwajibkan untuk menyelesaikan Praktikum 4

5.2 Tutorial

1. Buka kembali Website flutlab.io. Lalu buka file **register.dart**
2. Cari baris **class _RegisterPageState extends State<RegisterPage>**, lalu tambahkan kode berikut tepat di bawahnya

   Potongan Kode
   ```
   final NIMControl = TextEditingController();
   final NamaControl = TextEditingController();
   final EmailControl = TextEditingController();
   final PasswordControl = TextEditingController();

   @override
   void dispose() {
       // Clean up the controller when the widget is disposed.
       NIMControl.dispose();
       NamaControl.dispose();
       EmailControl.dispose();
       PasswordControl.dispose();
       super.dispose();
   }
   ```



```dart
class _RegisterPageState extends State<RegisterPage> {
  final NIMControl = TextEditingController();
  final NamaControl = TextEditingController();
  final EmailControl = TextEditingController();
  final PasswordControl = TextEditingController();

  @override
  void dispose() {
    // Clean up the controller when the widget is disposed.
    NIMControl.dispose();
    NamaControl.dispose();
    EmailControl.dispose();
    PasswordControl.dispose();
    super.dispose();
  }
```

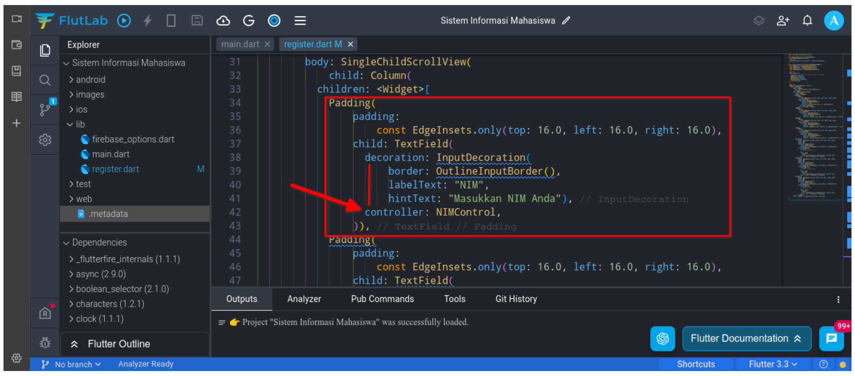
```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

3. Tempelkan masing-masing kontroler ke **widgetnya**. Perhatikan kode dan Gambar secara teliti

- Widget **NIM** → NIMControl

Potongan Kode

```
controller: NIMControl,
```

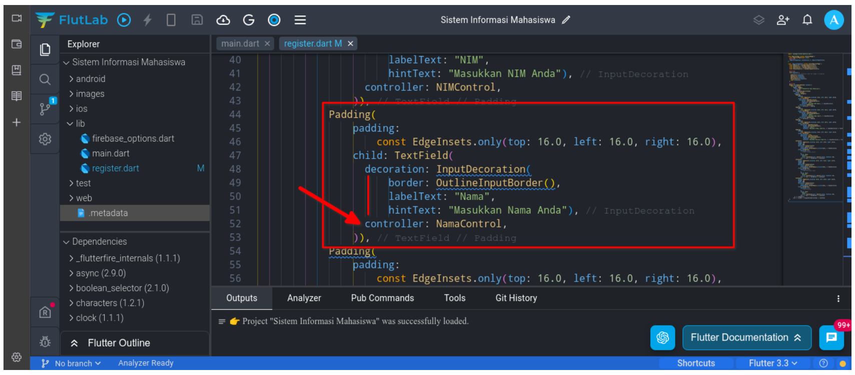


```dart
        body: SingleChildScrollView(
            child: Column(
          children: <Widget>[
            Padding(
                padding:
                    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                child: TextField(
                  decoration: InputDecoration(
                      border: OutlineInputBorder(),
                      labelText: "NIM",
                      hintText: "Masukkan NIM Anda"), // InputDecoration
                  controller: NIMControl,
                )), // TextField // Padding
            Padding(
                padding:
                    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                child: TextField(
```

- Widget **Nama** → NamaControl

Potongan Kode

```
controller: NamaControl,
```

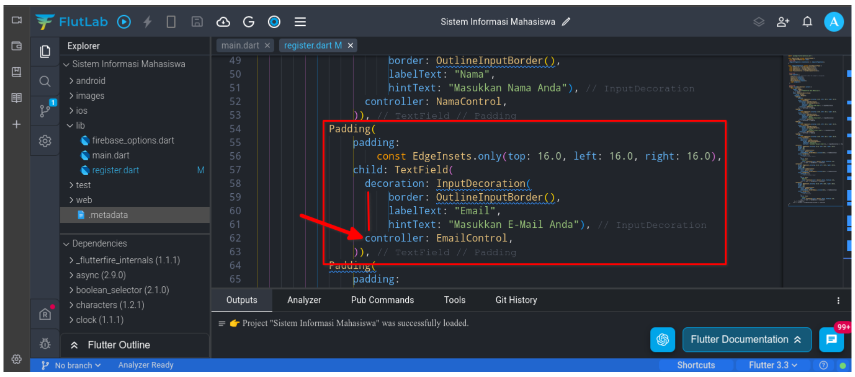


```dart
                      labelText: "NIM",
                      hintText: "Masukkan NIM Anda"), // InputDecoration
                  controller: NIMControl,
                )), // TextField // Padding
            Padding(
                padding:
                    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                child: TextField(
                  decoration: InputDecoration(
                      border: OutlineInputBorder(),
                      labelText: "Nama",
                      hintText: "Masukkan Nama Anda"), // InputDecoration
                  controller: NamaControl,
                )), // TextField // Padding
            Padding(
                padding:
                    const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
```

- Widget **Email** → EmailControl

Potongan Kode

```
controller: EmailControl,
```



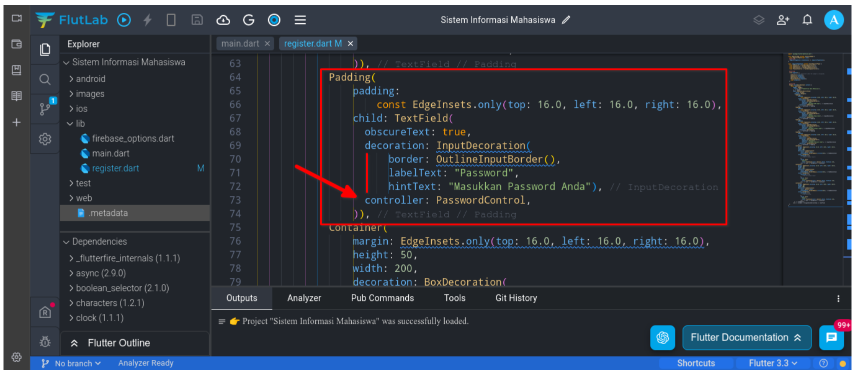
```
            border: OutlineInputBorder(),
            labelText: "Nama",
            hintText: "Masukkan Nama Anda"), // InputDecoration
          controller: NamaControl,
        )), // TextField // Padding
      Padding(
        padding:
          const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
        child: TextField(
          decoration: InputDecoration(
            border: OutlineInputBorder(),
            labelText: "Email",
            hintText: "Masukkan E-Mail Anda"), // InputDecoration
          controller: EmailControl,
        )), // TextField // Padding
      Padding(
        padding:
```

Project "Sistem Informasi Mahasiswa" was successfully loaded.

- Wdiget **Password** → PasswordControl

Potongan Kode

```
controller: PasswordControl,
```



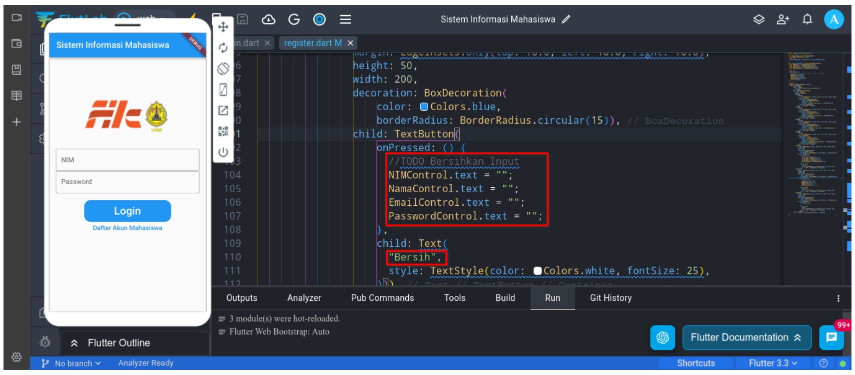
```
        )), // TextField // Padding
      Padding(
        padding:
          const EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
        child: TextField(
          obscureText: true,
          decoration: InputDecoration(
            border: OutlineInputBorder(),
            labelText: "Password",
            hintText: "Masukkan Password Anda"), // InputDecoration
          controller: PasswordControl,
        )), // TextField // Padding
      Container(
        margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
        height: 50,
        width: 200,
        decoration: BoxDecoration(
```

Project "Sistem Informasi Mahasiswa" was successfully loaded.

4. Setelah mengeset **controller** ke semua widget, kini aplikasi dapat mengambil maupun mengeset data. Scroll turun ke widget Button **Bersih** dan masukkan kode berikut:

Potongan Kode

```
//TODO Bersihkan Input
NIMControl.text = "";
NamaControl.text = "";
EmailControl.text = "";
PasswordControl.text = "";
```

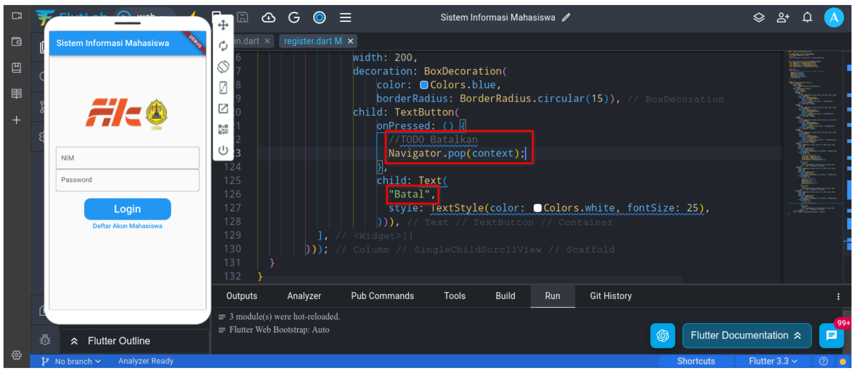


```
        height: 50,
        width: 200,
        decoration: BoxDecoration(
          color: Colors.blue,
          borderRadius: BorderRadius.circular(15)), // BoxDecoration
        child: TextButton(
          onPressed: () {
            //TODO Bersihkan Input
            NIMControl.text = "";
            NamaControl.text = "";
            EmailControl.text = "";
            PasswordControl.text = "";
          },
          child: Text(
            "Bersih",
            style: TextStyle(color: Colors.white, fontSize: 25),
```

3 module(s) were hot-reloaded.
Flutter Web Bootstrap: Auto

5. Kemudian agar button **Batal** bisa berfungsi, masukkan kode berikut:

Potongan Kode
```
//TODO Batalkan
Navigator.pop(context);
```

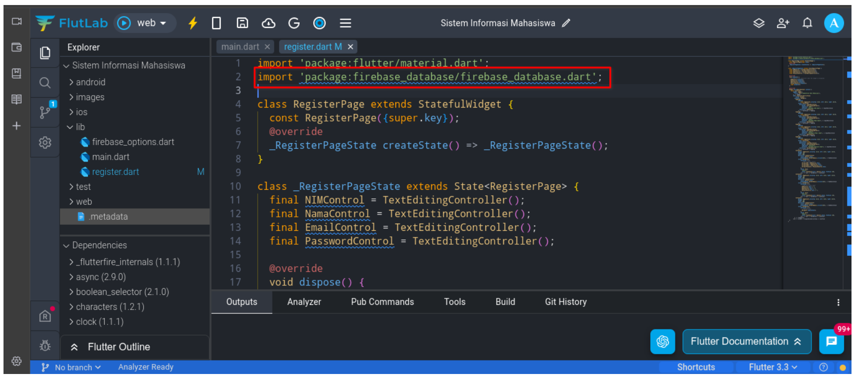


```
                  width: 200,
                  decoration: BoxDecoration(
                      color: Colors.blue,
                      borderRadius: BorderRadius.circular(15)), // BoxDecoration
                  child: TextButton(
                      onPressed: () {
                        //TODO Batalkan
                        Navigator.pop(context);
                      },
                      child: Text(
                        "Batal",
                        style: TextStyle(color: Colors.white, fontSize: 25),
                      ))), // Text // TextButton // Container
            ], // <Widget>[]
          ))); // Column // SingleChildScrollView // Scaffold
    }
}
```

```
3 module(s) were hot-reloaded.
Flutter Web Bootstrap: Auto
```

6. Untuk bisa melakukan kueri ke Firebase, pertama-tama **import** firebase reference dengan kode berikut:

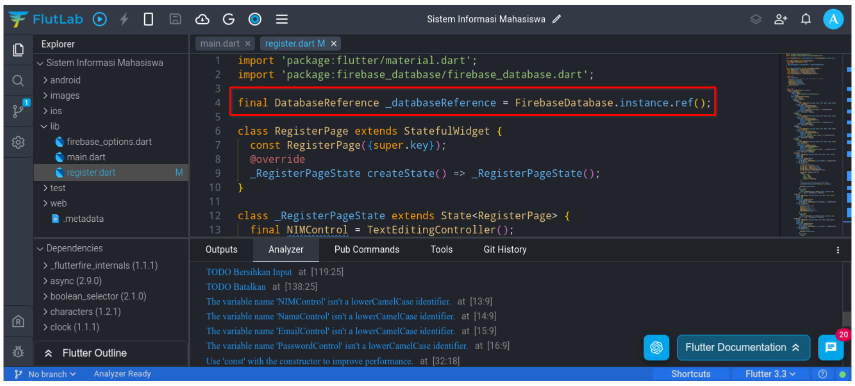
Potongan Kode
```
import 'package:firebase_database/firebase_database.dart';
```



```
import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';

class RegisterPage extends StatefulWidget {
  const RegisterPage({super.key});
  @override
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  final NIMControl = TextEditingController();
  final NamaControl = TextEditingController();
  final EmailControl = TextEditingController();
  final PasswordControl = TextEditingController();

  @override
  void dispose() {
```

7. Lalu tambahkan variabel global tepat di bawah kode import

Potongan Kode
```
final DatabaseReference _databaseReference = FirebaseDatabase
    .instance.ref();
```



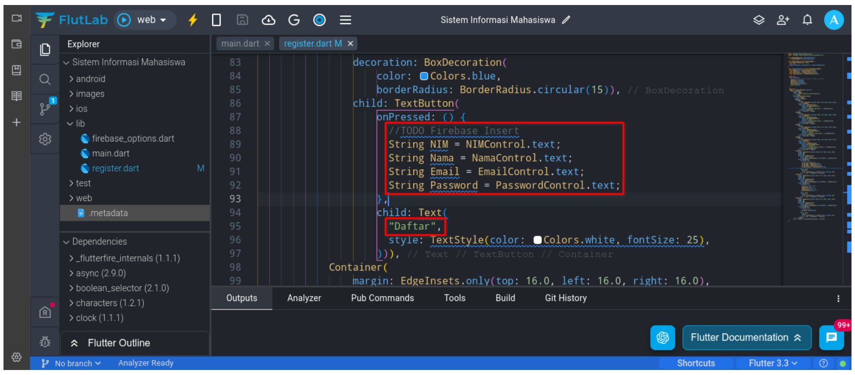
```
import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

class RegisterPage extends StatefulWidget {
  const RegisterPage({super.key});
  @override
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  final NIMControl = TextEditingController();
```

```
TODO Bersihkan Input  at [119:25]
TODO Batalkan  at [138:25]
The variable name 'NIMControl' isn't a lowerCamelCase identifier.  at [13:9]
The variable name 'NamaControl' isn't a lowerCamelCase identifier.  at [14:9]
The variable name 'EmailControl' isn't a lowerCamelCase identifier.  at [15:9]
The variable name 'PasswordControl' isn't a lowerCamelCase identifier.  at [16:9]
Use 'const' with the constructor to improve performance.  at [32:18]
```

8. Kemudian scroll turun ke bagian button **Daftar** dan masukkan kode berikut untuk mengambil data

Potongan Kode

```
//TODO Firebase Insert
String NIM = NIMControl.text;
String Nama = NamaControl.text;
String Email = EmailControl.text;
String Password = PasswordControl.text;
```

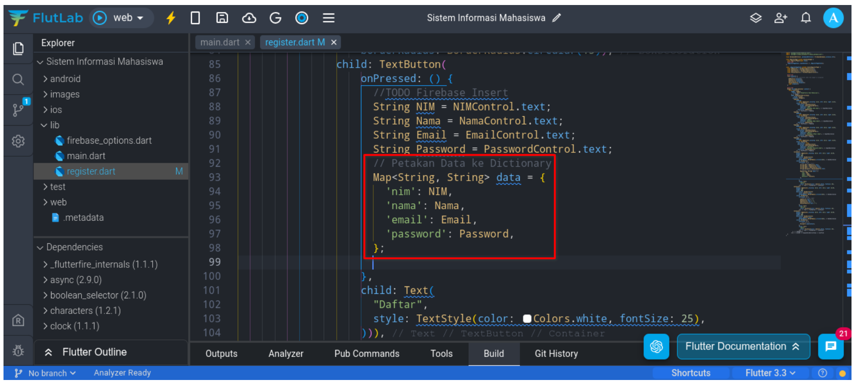


```
          decoration: BoxDecoration(
            color: Colors.blue,
            borderRadius: BorderRadius.circular(15)), // BoxDecoration
          child: TextButton(
            onPressed: () {
              //TODO Firebase Insert
              String NIM = NIMControl.text;
              String Nama = NamaControl.text;
              String Email = EmailControl.text;
              String Password = PasswordControl.text;
            },
            child: Text(
              "Daftar",
              style: TextStyle(color: Colors.white, fontSize: 25),
            ))), // Text // TextButton // Container
        Container(
          margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
```

9. Kemudian bungkus data-data tersebut dalam bentuk **dictionary**, perhatikan kode dan gambar di bawah

Potongan Kode

```
// Petakan Data ke Dictionary
Map<String, String> data = {
    'nim': NIM,
    'nama': Nama,
    'email': Email,
    'password': Password,
};
```

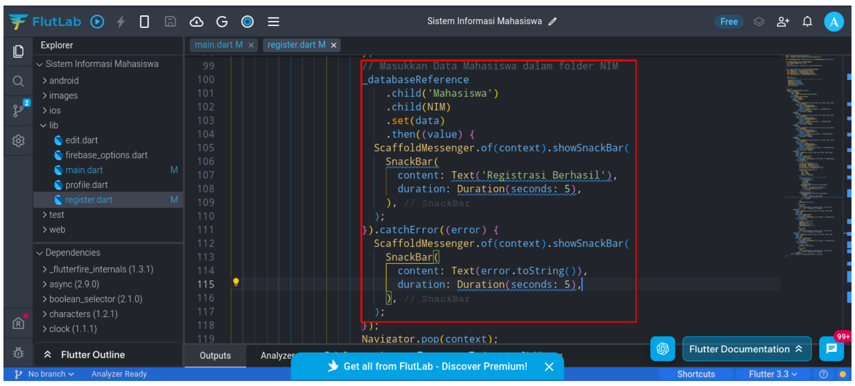


```
        child: TextButton(
          onPressed: () {
            //TODO Firebase Insert
            String NIM = NIMControl.text;
            String Nama = NamaControl.text;
            String Email = EmailControl.text;
            String Password = PasswordControl.text;
            // Petakan Data ke Dictionary
            Map<String, String> data = {
              'nim': NIM,
              'nama': Nama,
              'email': Email,
              'password': Password,
            };

          },
          child: Text(
            "Daftar",
            style: TextStyle(color: Colors.white, fontSize: 25),
          ))), // Text // TextButton // Container
```

10. Kemudian tambahkan kode untuk melakukan kueri insert dan pengecekan keberhasilan

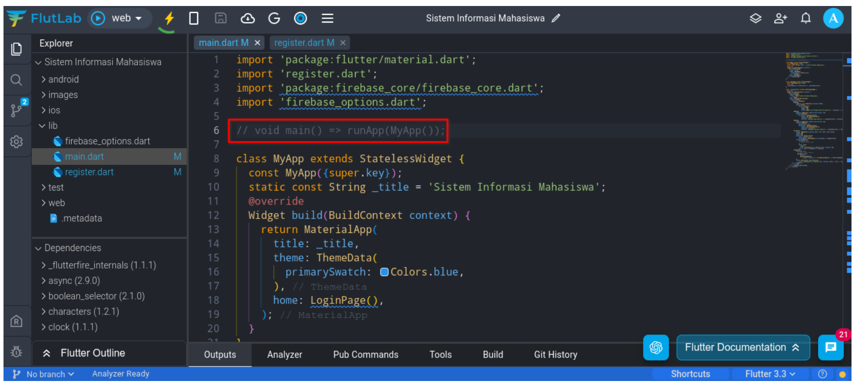
Potongan Kode

```
// Masukkan Data Mahasiswa dalam folder NIM
_databaseReference
.child('Mahasiswa')
.child(NIM)
.set(data)
.then((value) {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text('Registrasi Berhasil'),
            duration: Duration(seconds: 3),
        ),
    );
}).catchError((error) {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text(error.toString()),
            duration: Duration(seconds: 3),
        ),
    );
});
Navigator.pop(context);
```



```
// Masukkan Data Mahasiswa dalam folder NIM
_databaseReference
    .child('Mahasiswa')
    .child(NIM)
    .set(data)
    .then((value) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text('Registrasi Berhasil'),
      duration: Duration(seconds: 5),
    ), // SnackBar
  );
}).catchError((error) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text(error.toString()),
      duration: Duration(seconds: 5),
    ), // SnackBar
  );
});
Navigator.pop(context);
```

11. Langkah berikutnya adalah mengubah fungsi **main** karena harus melakukan inisialisasi database. Buka **main.dart** dan beri tanda // di depan baris kode **void main() => runApp(MyApp());**



```
import 'package:flutter/material.dart';
import 'register.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

// void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ), // ThemeData
      home: LoginPage(),
    ); // MaterialApp
  }
```

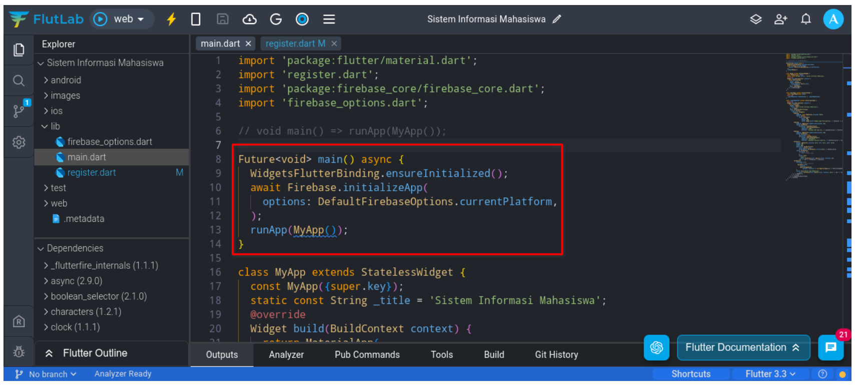
12. Lalu di bawahnya baris kode yang diberi komentar tadi, masukkan kode berikut:

Potongan Kode

```
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
```

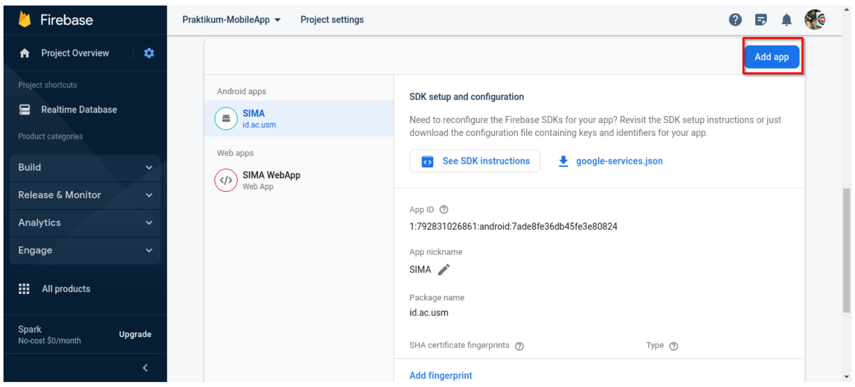
Potongan Kode

```
Future<void> main() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp(
        options: DefaultFirebaseOptions.currentPlatform,
    );
    runApp(MyApp());
}
```



```
import 'package:flutter/material.dart';
import 'register.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

// void main() => runApp(MyApp());

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
  @override
  Widget build(BuildContext context) {
```

13. Aplikasi ini akan gagal ketika dijalankan, sehingga harus ditambahkan akses baru untuk aplikasi Web. Buka link `https://firebase.google.com`, buka **Project Settings (Gigi Roda)**, pilih **General**, Scroll Down ke **Your Apps**, klik **Add App**



SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

See SDK instructions   google-services.json

App ID
1:792831026861:android:7ade8fe36db45fe3e80824

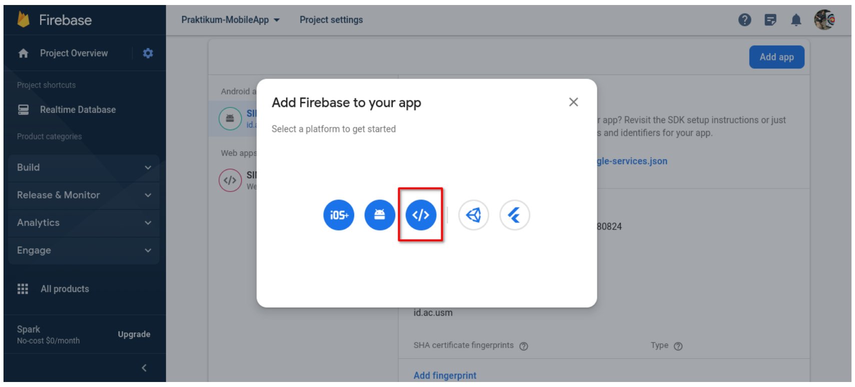
App nickname
SIMA

Package name
id.ac.usm

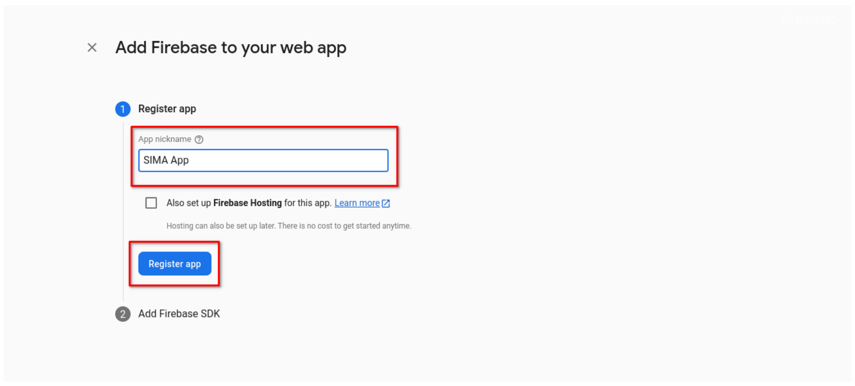
SHA certificate fingerprints   Type

Add fingerprint

14. Pilih platform **Web**



15. Beri nama **SIMA App**, lalu klik **Register App**


Add Firebase to your web app

1 Register app

App nickname: SIMA App

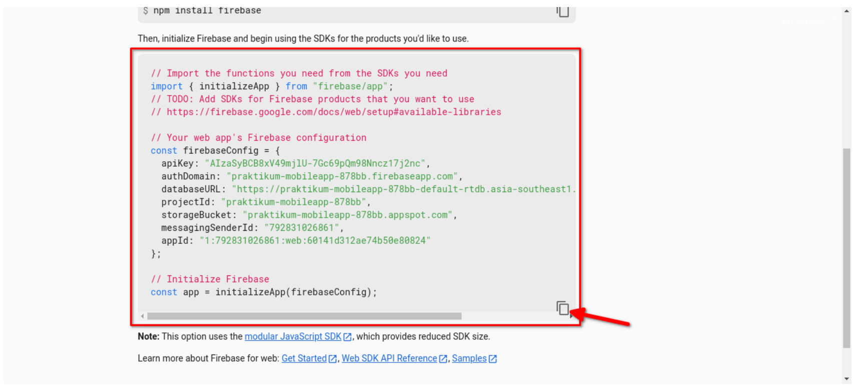
Also set up **Firebase Hosting** for this app. Learn more

Hosting can also be set up later. There is no cost to get started anytime.

Register app

2 Add Firebase SDK

16. Lalu kopi kode yang diperlihatkan Firebase


```
$ npm install firebase
```

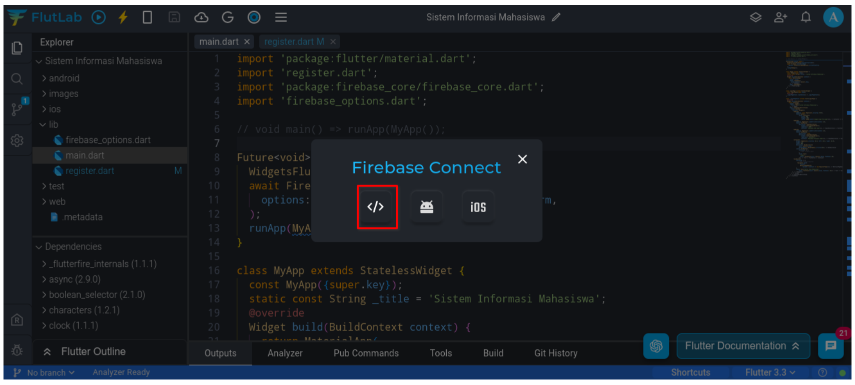
Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyBCB8xV49mjlU-7Gc69pQm98Nncz17j2nc",
  authDomain: "praktikum-mobileapp-878bb.firebaseapp.com",
  databaseURL: "https://praktikum-mobileapp-878bb-default-rtdb.asia-southeast1.
  projectId: "praktikum-mobileapp-878bb",
  storageBucket: "praktikum-mobileapp-878bb.appspot.com",
  messagingSenderId: "792831026861",
  appId: "1:792831026861:web:60141d312ae74b50e80824"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Note: This option uses the modular JavaScript SDK, which provides reduced SDK size.

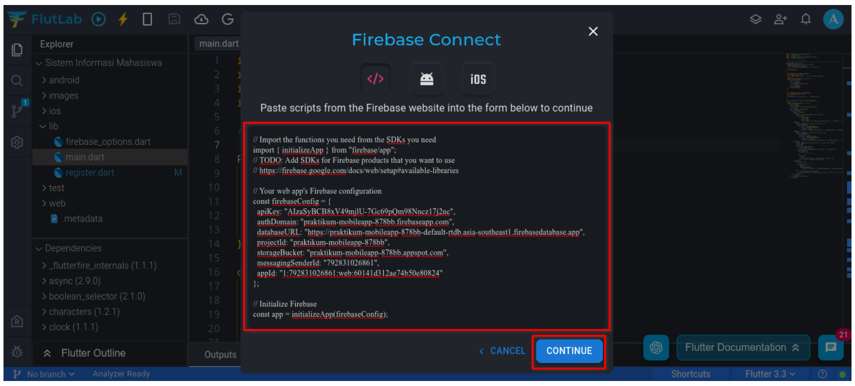
Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

17. Kembali ke Flutlab, klik **Icon G**, pilih **Connect to Firebase**, pilih **Icon Web** yang sama dengan Firebase sebelumnya



```
import 'package:flutter/material.dart';
import 'register.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

// void main() => runApp(MyApp());

Future<void>
  WidgetsFlu
  await Fire
    options:                                m,
  );
  runApp(MyA
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
  @override
  Widget build(BuildContext context) {
```

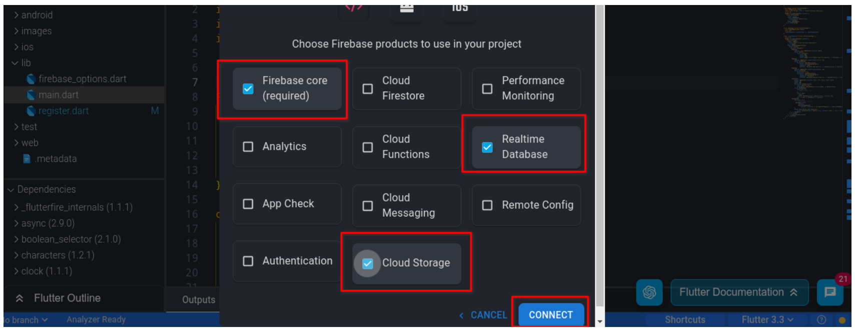
18. Tempel kode dari FIrebase, dan klik **Continue**



Paste scripts from the Firebase website into the form below to continue

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyBCB8xV49mjlU-7Gc69pQm98Nncz17j2nc",
  authDomain: "praktikum-mobileapp-878bb.firebaseapp.com",
  databaseURL: "https://praktikum-mobileapp-878bb-default-rtdb.asia-southeast1.firebasedatabase.app",
  projectId: "praktikum-mobileapp-878bb",
  storageBucket: "praktikum-mobileapp-878bb.appspot.com",
  messagingSenderId: "792831026861",
  appId: "1:792831026861:web:60141d312ae74b50e80824"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

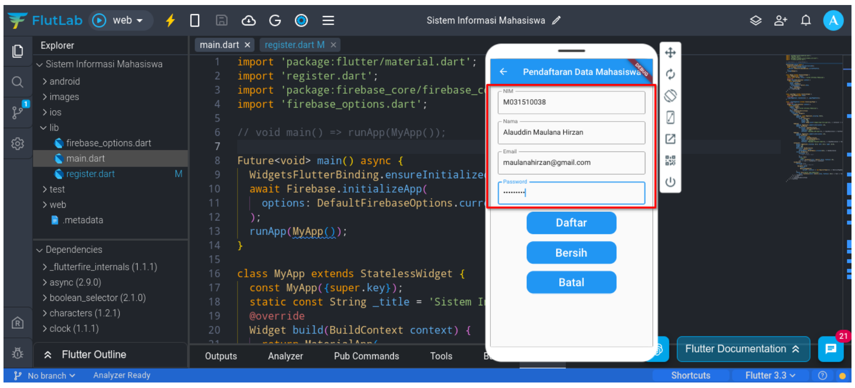
19. Di menu selanjutnya pilih
    - Firebase Core
    - Realtime Database
    - Cloud Storage



Choose Firebase products to use in your project

✓ Firebase core (required)
Cloud Firestore
Performance Monitoring
Analytics
Cloud Functions
✓ Realtime Database
App Check
Cloud Messaging
Remote Config
Authentication
✓ Cloud Storage

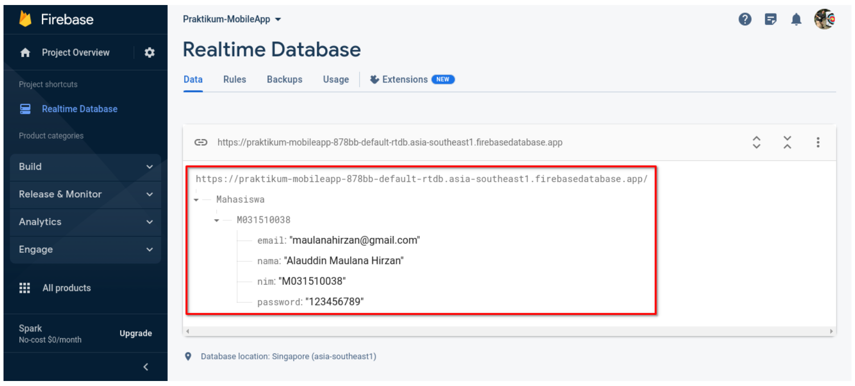
20. Sesudah terhubung, maka aplikasi siap digunakan. Jalankan Web Emulator dan coba melakukan registrasi. Isikan **NIM TANPA TITIK** untuk mencoba.



```
import 'package:flutter/material.dart';
import 'register.dart';
import 'package:firebase_core/firebase_c
import 'firebase_options.dart';

// void main() => runApp(MyApp());

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialize
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.curr
  );
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem I
  @override
  Widget build(BuildContext context) {
```

Pendaftaran Data Mahasiswa

NIM: M031510038
Nama: Alauddin Maulana Hirzan
Email: maulanahirzan@gmail.com
Password: ·········



Praktikum-MobileApp

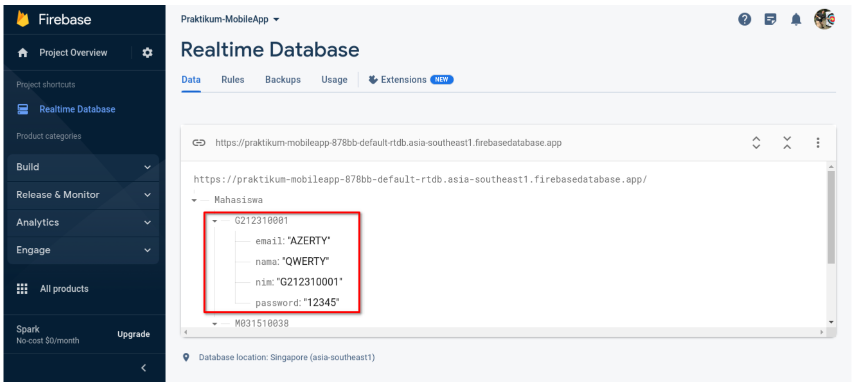
Realtime Database

https://praktikum-mobileapp-878bb-default-rtdb.asia-southeast1.firebasedatabase.app

```
https://praktikum-mobileapp-878bb-default-rtdb.asia-southeast1.firebasedatabase.app/
  Mahasiswa
    M031510038
      email: "maulanahirzan@gmail.com"
      nama: "Alauddin Maulana Hirzan"
      nim: "M031510038"
      password: "123456789"
```

Database location: Singapore (asia-southeast1)

21. Data Mahasiswa lain dapat masuk ke DB



Praktikum-MobileApp

Realtime Database

https://praktikum-mobileapp-878bb-default-rtdb.asia-southeast1.firebasedatabase.app

```
https://praktikum-mobileapp-878bb-default-rtdb.asia-southeast1.firebasedatabase.app/
  Mahasiswa
    G212310001
      email: "AZERTY"
      nama: "QWERTY"
      nim: "G212310001"
      password: "12345"
    M031510038
```

Database location: Singapore (asia-southeast1)

22. **Screenshot** Database dan kirimkan ke **E-Learning**

Bab 6

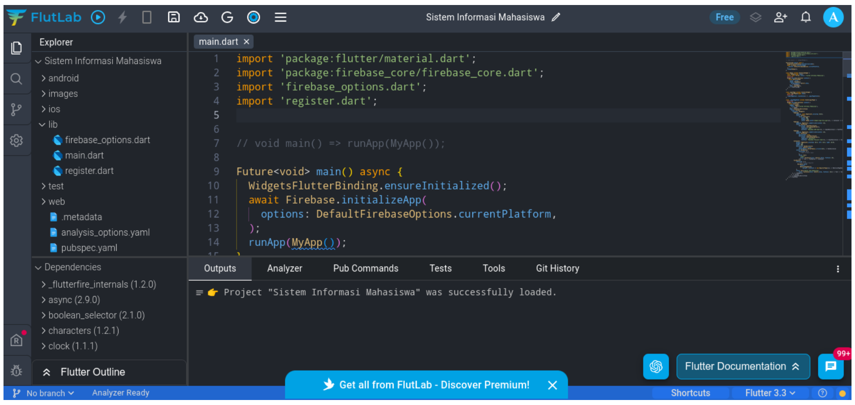
Praktikum 6

6.1 Operasi Read Data Flutter

Di bagian ini mahasiswa diajarkan bagaimana membuat tampilan profile untuk pengguna, sekaligus proses Login yang dilakukan dengan menggunakan Flutter. Mahasiswa diwajibkan menyelesaikan Praktikum 5 sebelum memulai praktikum ini.

6.2 Tutorial

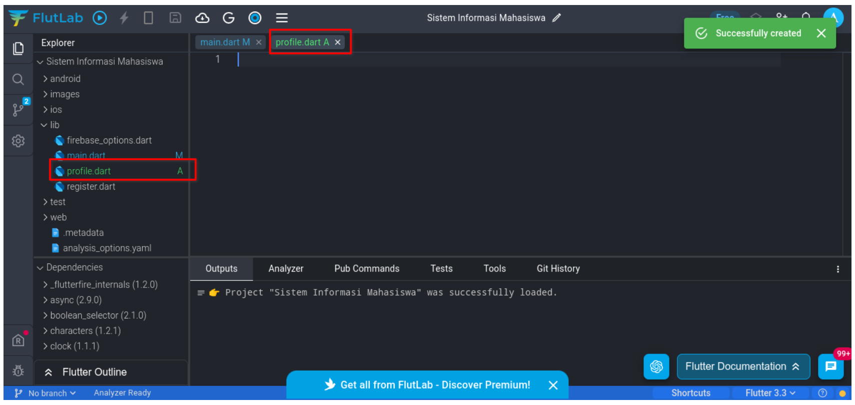
1. Praktikum ini dimulai dengan membuka praktikum sebelumnya.



```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'register.dart';


// void main() => runApp(MyApp());

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(MyApp());
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

2. Buatlah satu file baru dengan nama **profile.dart**

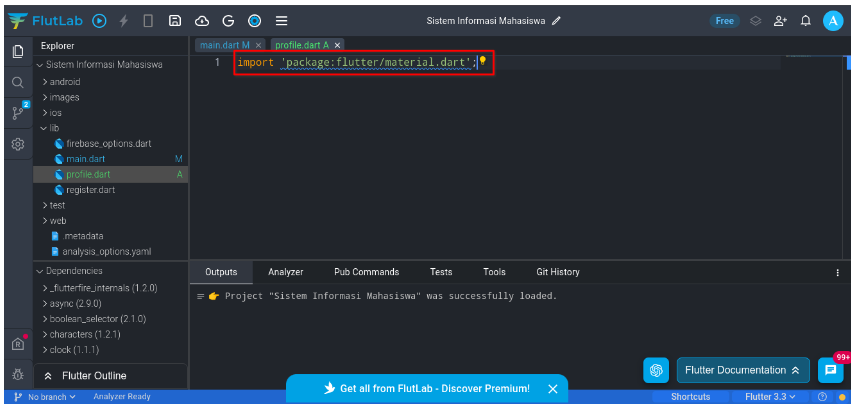


```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

3. Masukkan kode import sebagai **Library Dasar** aplikasi

Potongan Kode

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



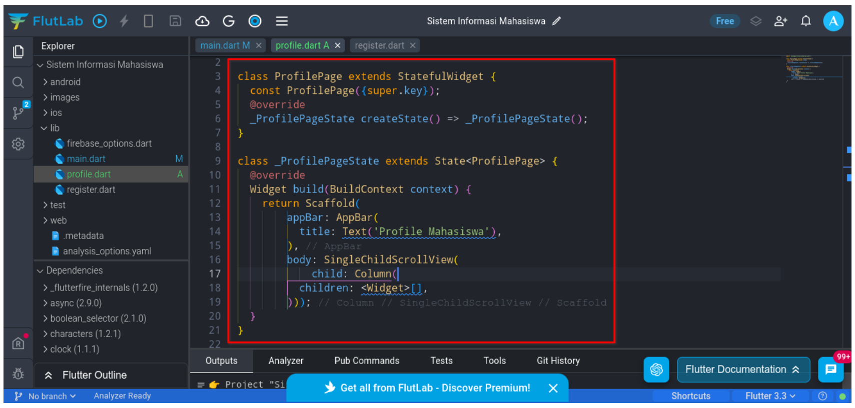
```
import 'package:flutter/material.dart';
```

```
≡ 👉 Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

4. Kemudian buatlah tampilan Profile dengan kode sebagai berikut. Tambahkan setelah kode **import**

Potongan Kode

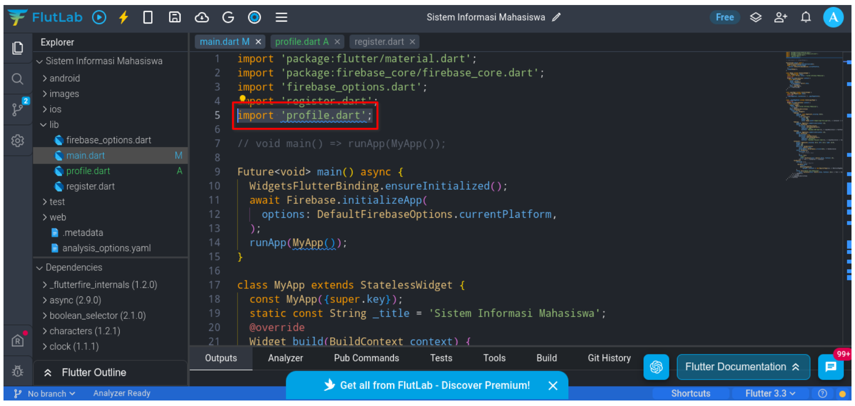
```
class ProfilePage extends StatefulWidget {
     const ProfilePage({super.key});
    @override
    _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
            title: Text('Profile Mahasiswa'),
            ),
            body: SingleChildScrollView(
                child: Column(
                children: <Widget>[],
                )));
    }
}
```



```

class ProfilePage extends StatefulWidget {
  const ProfilePage({super.key});
  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('Profile Mahasiswa'),
        ), // AppBar
        body: SingleChildScrollView(
          child: Column(
          children: <Widget>[],
        ))); // Column // SingleChildScrollView // Scaffold
  }
}

```

5. Sesudah itu, buka file **main.dart** lalu **Import** file **profile.dart** seperti berikut:

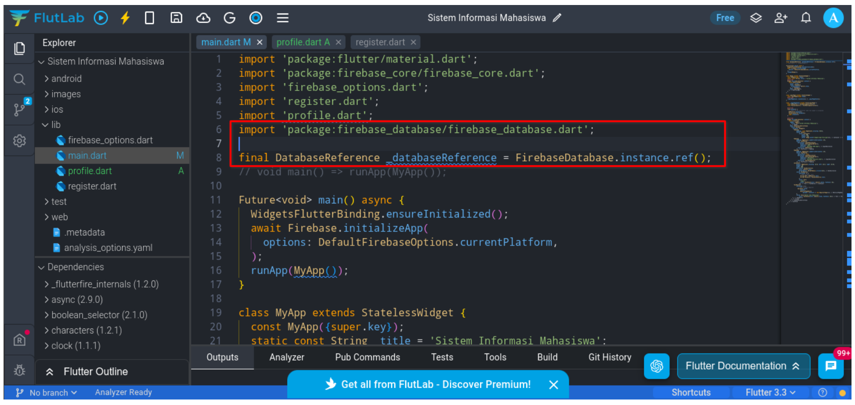
Potongan Kode
```
import 'profile.dart';
```



```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'register.dart';
import 'profile.dart';

// void main() => runApp(MyApp());

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
  @override
  Widget build(BuildContext context) {
```

6. Agar bisa terhubung ke database, tambahkan kode berikut tepat di bawah **import** sebelumnya

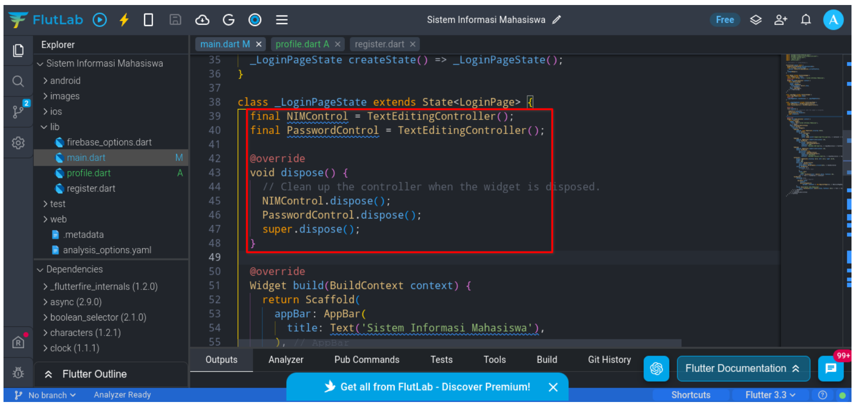
Potongan Kode
```
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();
```



```
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'register.dart';
import 'profile.dart';
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();
// void main() => runApp(MyApp());

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});
  static const String _title = 'Sistem Informasi Mahasiswa';
```

7. Scroll turun ke **class _LoginPageState**, dan tambahkan kode berikut tepat di bawahnya dan sebelum **@override**

Potongan Kode
```
final NIMControl = TextEditingController();
final PasswordControl = TextEditingController();

@override
void dispose() {
    NIMControl.dispose();
    PasswordControl.dispose();
    super.dispose();
}
```



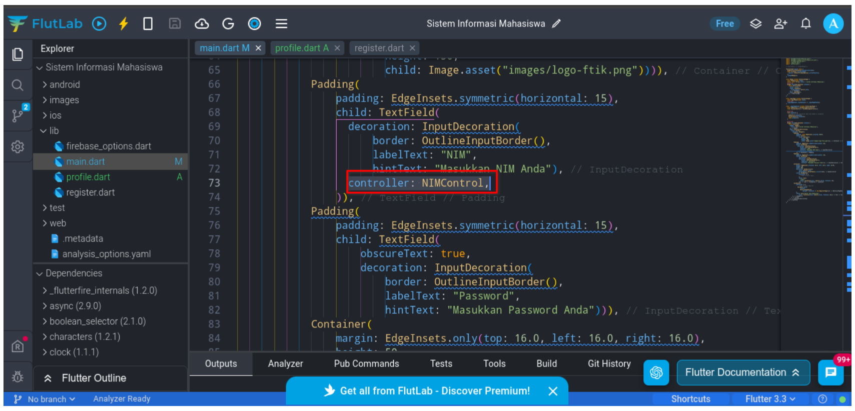
```
    _LoginPageState createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  final NIMControl = TextEditingController();
  final PasswordControl = TextEditingController();

  @override
  void dispose() {
    // Clean up the controller when the widget is disposed.
    NIMControl.dispose();
    PasswordControl.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Sistem Informasi Mahasiswa'),
      ), // AppBar
```

8. Kemudian turun kembali untuk menempelkan kontrol teks ke masing-masing widget. Perhatikan Kode dan Gambar berikut

   (a) field **NIM**

   Potongan Kode

   ```
   controller: NIMControl,
   ```

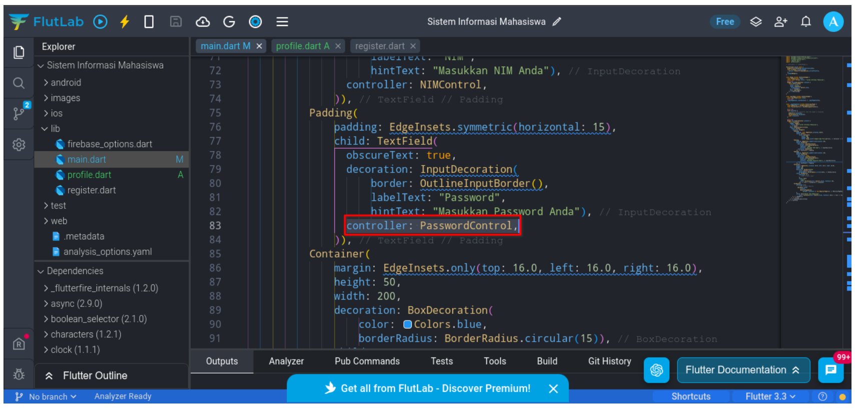


```
                child: Image.asset("images/logo-ftik.png"))), // Container // C
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 15),
              child: TextField(
                decoration: InputDecoration(
                    border: OutlineInputBorder(),
                    labelText: "NIM",
                    hintText: "Masukkan NIM Anda"), // InputDecoration
                controller: NIMControl,
              )), // TextField // Padding
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 15),
              child: TextField(
                  obscureText: true,
                  decoration: InputDecoration(
                      border: OutlineInputBorder(),
                      labelText: "Password",
                      hintText: "Masukkan Password Anda"))), // InputDecoration // Tex
          Container(
              margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
```

   (b) field **Password**

   Potongan Kode

   ```
   controller: PasswordControl,
   ```

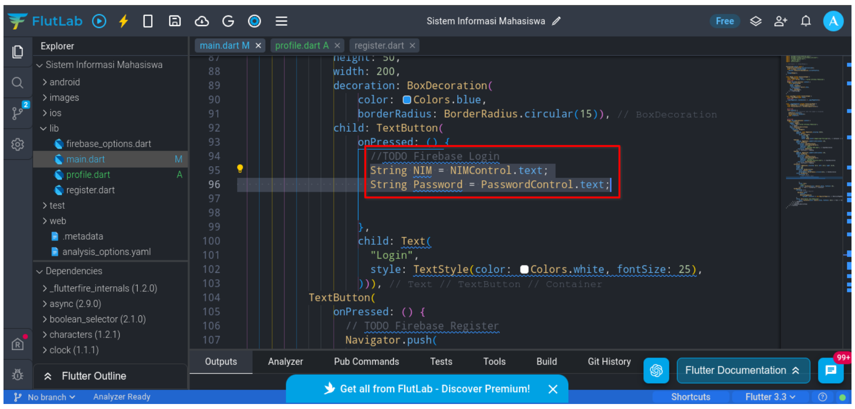


```
                    hintText: "Masukkan NIM Anda"), // InputDecoration
                controller: NIMControl,
              )), // TextField // Padding
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 15),
              child: TextField(
                obscureText: true,
                decoration: InputDecoration(
                    border: OutlineInputBorder(),
                    labelText: "Password",
                    hintText: "Masukkan Password Anda"), // InputDecoration
                controller: PasswordControl,
              )), // TextField // Padding
          Container(
              margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
              height: 50,
              width: 200,
              decoration: BoxDecoration(
                  color: Colors.blue,
                  borderRadius: BorderRadius.circular(15)), // BoxDecoration
```

9. Kini setiap field dapat ditarik datanya. Berikutnya tambahkan kode berikut di tombol **Login**. Perhatikan gambar dan kode berikut:

Potongan Kode

```
String NIM = NIMControl.text;
String Password = PasswordControl.text;
```

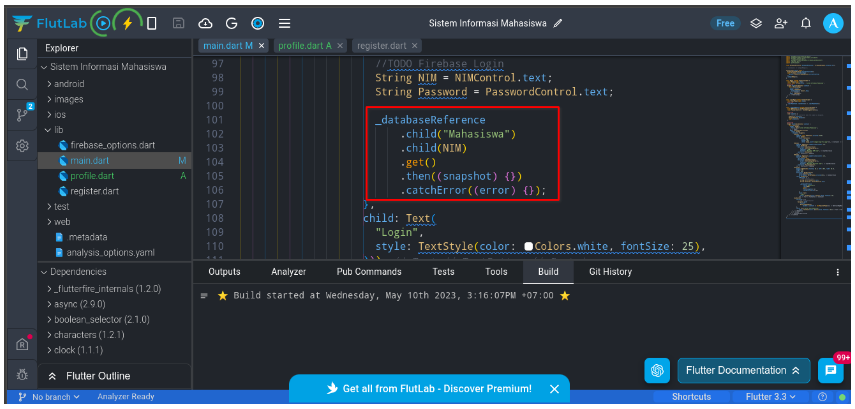


```
          height: 50,
          width: 200,
          decoration: BoxDecoration(
              color: Colors.blue,
              borderRadius: BorderRadius.circular(15)), // BoxDecoration
          child: TextButton(
              onPressed: () {
                //TODO Firebase Login
                String NIM = NIMControl.text;
                String Password = PasswordControl.text;


              },
              child: Text(
                "Login",
                style: TextStyle(color: Colors.white, fontSize: 25),
              ))), // Text // TextButton // Container
      TextButton(
          onPressed: () {
            // TODO Firebase Register
            Navigator.push(
```

10. Tambahkan kode berikut untuk melakukan pengecekan login

Potongan Kode

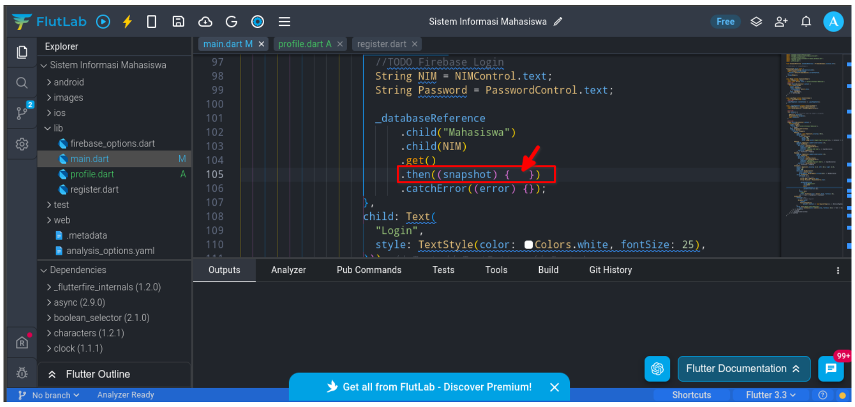
```
_databaseReference
     .child("Mahasiswa")
     .child(NIM)
     .get()
     .then((snapshot) { })
     .catchError((error) { });
```



```
            //TODO Firebase Login
            String NIM = NIMControl.text;
            String Password = PasswordControl.text;

            _databaseReference
                .child("Mahasiswa")
                .child(NIM)
                .get()
                .then((snapshot) {})
                .catchError((error) {});
          },
          child: Text(
            "Login",
            style: TextStyle(color: Colors.white, fontSize: 25),
```

```
★ Build started at Wednesday, May 10th 2023, 3:16:07PM +07:00 ★
```

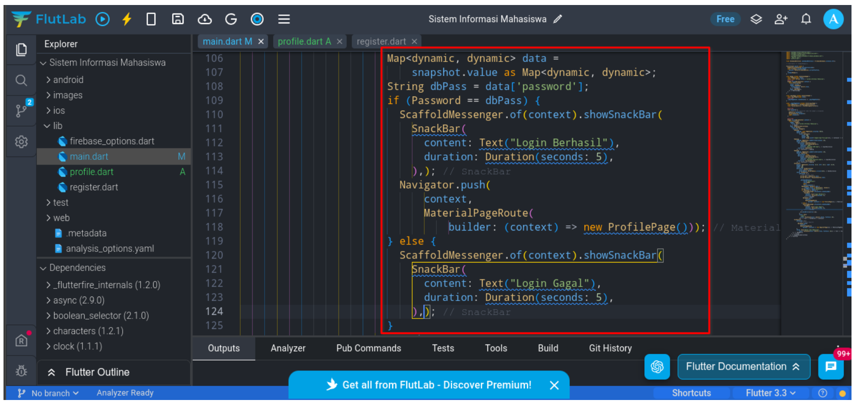
11. Setelah itu di dalam tanda kurung kurawal dari **Snapshot**. Masukkan kode berikut



```
            //TODO Firebase Login
            String NIM = NIMControl.text;
            String Password = PasswordControl.text;

            _databaseReference
                .child("Mahasiswa")
                .child(NIM)
                .get()
                .then((snapshot) {   })
                .catchError((error) {});
          },
          child: Text(
            "Login",
            style: TextStyle(color: Colors.white, fontSize: 25),
```

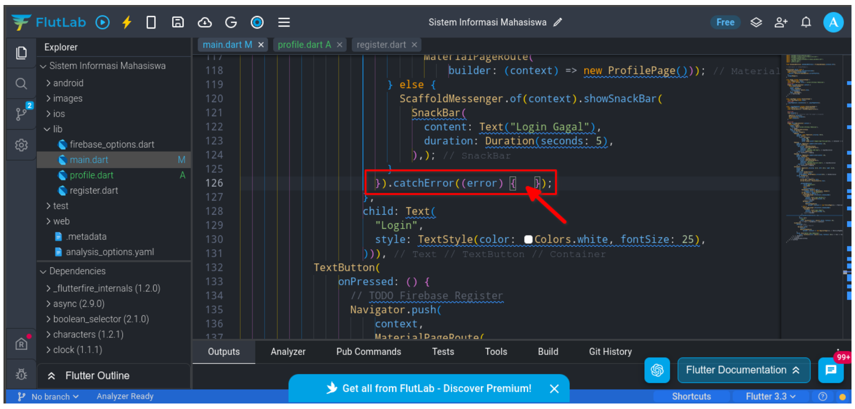
Potongan Kode

```
Map<dynamic, dynamic> data =
    snapshot.value as Map<dynamic, dynamic>;
String dbPass = data['password'];
if (Password == dbPass) {
    ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
        content: Text("Login Berhasil"),
        duration: Duration(seconds: 5),
    ),);
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) => new ProfilePage()));
} else {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text("Login Gagal"),
            duration: Duration(seconds: 5),
    ),);
}
```



```
Map<dynamic, dynamic> data =
    snapshot.value as Map<dynamic, dynamic>;
String dbPass = data['password'];
if (Password == dbPass) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text("Login Berhasil"),
      duration: Duration(seconds: 5),
    ),); // SnackBar
  Navigator.push(
      context,
      MaterialPageRoute(
          builder: (context) => new ProfilePage())); // Material
} else {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text("Login Gagal"),
      duration: Duration(seconds: 5),
    ),); // SnackBar
}
```

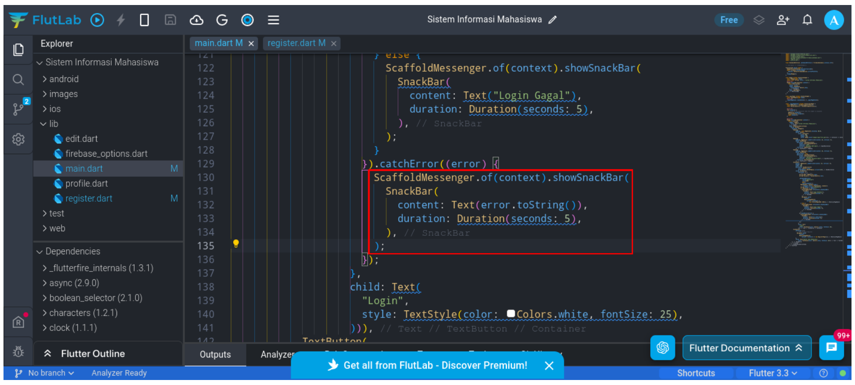
12. Kemudian di dalam kurung kurawal dari **catchError**, masukkan kode berikut



```
              builder: (context) => new ProfilePage())); // Material
      } else {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text("Login Gagal"),
            duration: Duration(seconds: 5),
          ),); // SnackBar
      }
    }).catchError((error) {  });
  },
  child: Text(
    "Login",
    style: TextStyle(color: Colors.white, fontSize: 25),
  ))), // Text // TextButton // Container
TextButton(
    onPressed: () {
      // TODO Firebase Register
      Navigator.push(
          context,
```

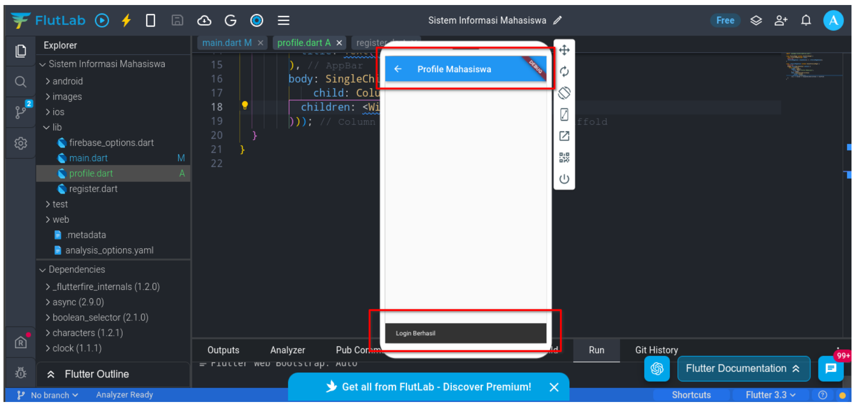
Potongan Kode

```
ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
        content: Text(error.toString()),
        duration: Duration(seconds: 5),
    ),
);
```

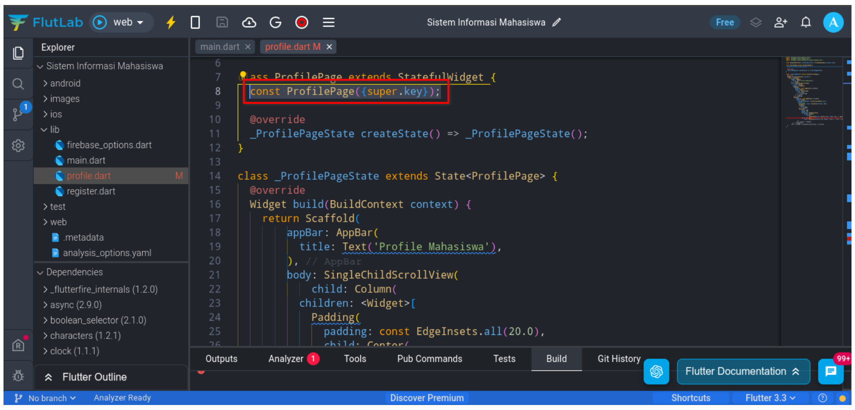


```
        } else {
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text("Login Gagal"),
              duration: Duration(seconds: 5),
            ), // SnackBar
          );
        }
      }).catchError((error) {
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            content: Text(error.toString()),
            duration: Duration(seconds: 5),
          ), // SnackBar
        );
      });
    },
    child: Text(
      "Login",
      style: TextStyle(color: Colors.white, fontSize: 25),
    )), // Text // TextButton // Container
  TextButton(
```

13. Jalankan aplikasi dan cobalah login. Jika sukses, halaman akan berpindah ke **profile.dart**. Tetapi kosong.



14. Berikutnya adalah mengisikan halaman Profile. Pertama hapus baris kode **const ProfilePage(super.key);**

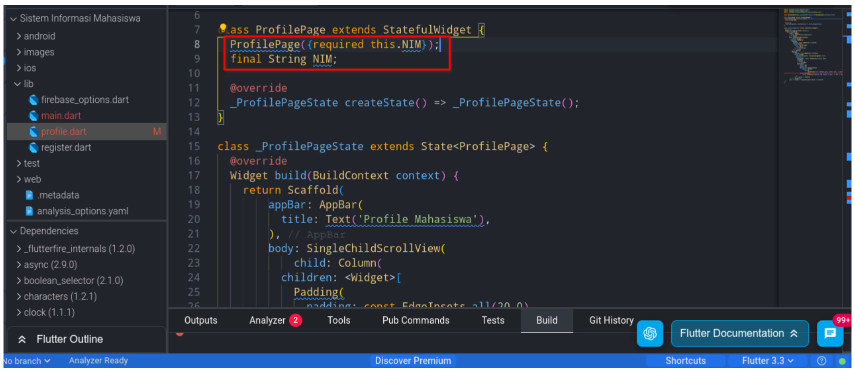


```
class ProfilePage extends StatefulWidget {
  const ProfilePage({super.key});

  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('Profile Mahasiswa'),
        ), // AppBar
        body: SingleChildScrollView(
            child: Column(
          children: <Widget>[
            Padding(
              padding: const EdgeInsets.all(20.0),
              child: Center(
```

15. Agar bisa menerima input data **NIM** dari **Login** ke **Profile**, masukkan kode berikut di **class ProfilePage**

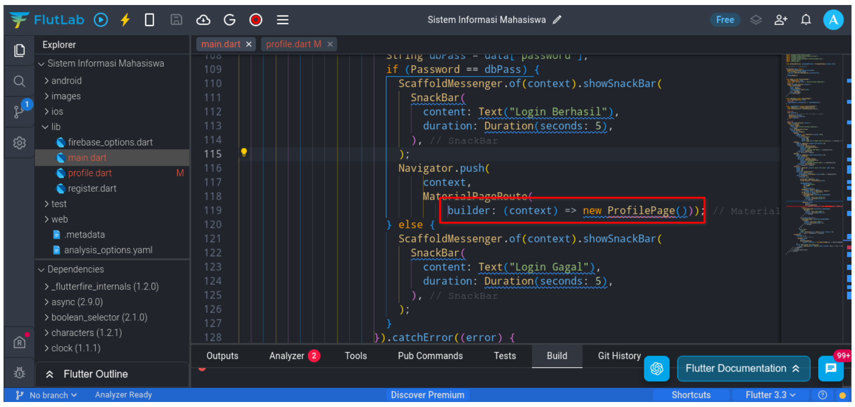
Potongan Kode

```
ProfilePage({required this.NIM});
final String NIM;
```



```
class ProfilePage extends StatefulWidget {
  ProfilePage({required this.NIM});
  final String NIM;

  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('Profile Mahasiswa'),
        ), // AppBar
        body: SingleChildScrollView(
          child: Column(
          children: <Widget>[
            Padding(
```

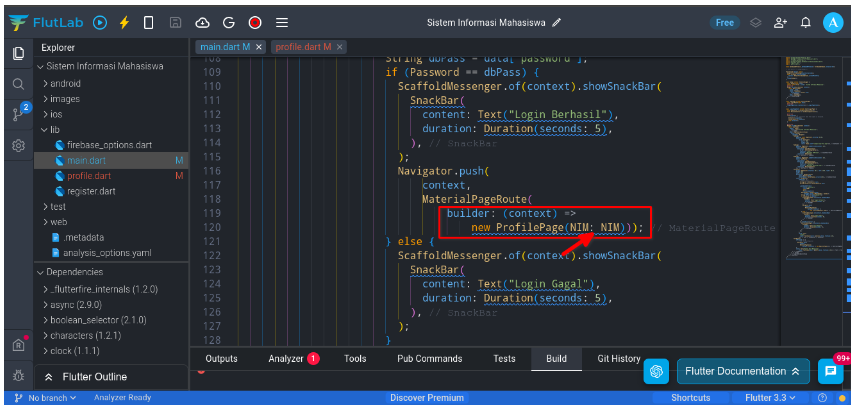
16. Kembali ke **main.dart**, tambahkan baris kode **builder: (context) => new ProfilePage()));** dengan kode berikut:



```
          if (Password == dbPass) {
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text("Login Berhasil"),
                duration: Duration(seconds: 5),
              ), // SnackBar
            );
            Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) => new ProfilePage())); // Material
          } else {
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text("Login Gagal"),
                duration: Duration(seconds: 5),
              ), // SnackBar
            );
          }
        }).catchError((error) {
```

Potongan Kode

```
NIM:NIM
```



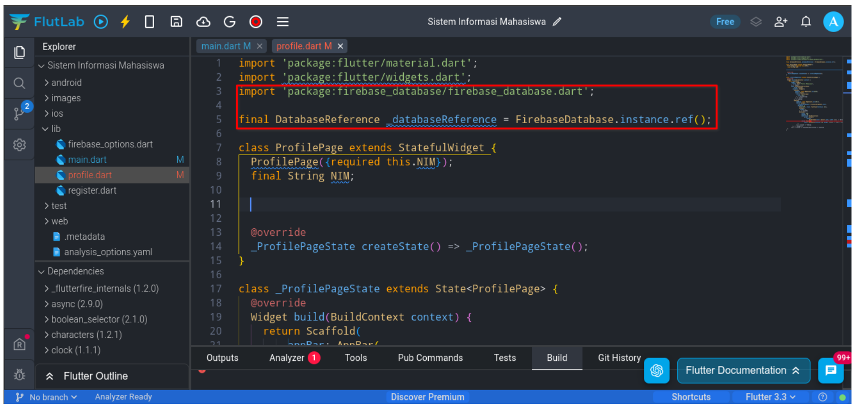
```
          if (Password == dbPass) {
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text("Login Berhasil"),
                duration: Duration(seconds: 5),
              ), // SnackBar
            );
            Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) =>
                      new ProfilePage(NIM: NIM))); // MaterialPageRoute
          } else {
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text("Login Gagal"),
                duration: Duration(seconds: 5),
              ), // SnackBar
            );
          }
```

17. Kini data NIM bisa dikirimkan ke **profile.dart**. Buka file **profile.dart**, agar bisa melakukan kueri data, tambahkan kode berikut:

18. Sebelum menambahkan kode untuk Text, tambahkan terlebih dahulu kode berikut untuk menarik data dari Firebase

Potongan Kode

```
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();
```

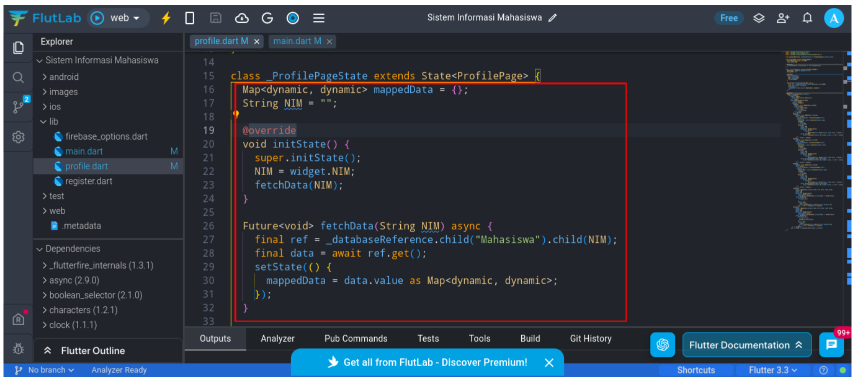


```
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

class ProfilePage extends StatefulWidget {
  ProfilePage({required this.NIM});
  final String NIM;



  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

19. Untuk melakukan kueri, **Scroll Turun** ke **class _ProfilePageState extends State<ProfilePage>**. Dan masukkan kode berikut tepat di bawahnya. Perhatikan Gambar dan Kode

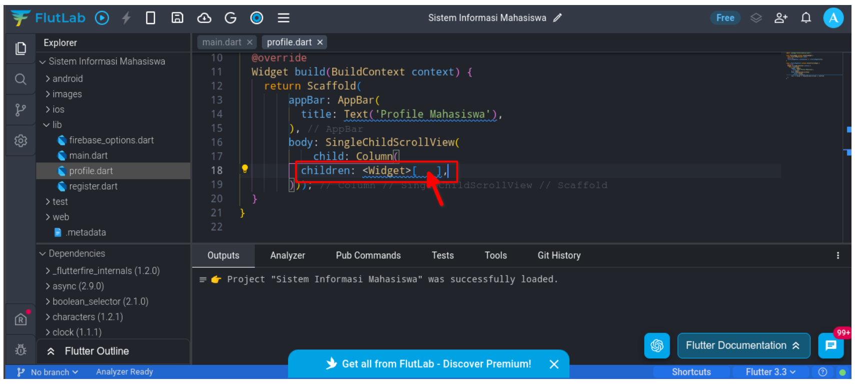
Potongan Kode

```
Map<dynamic, dynamic> mappedData = {};
String NIM = "";

@override
void initState() {
    super.initState();
    NIM = widget.NIM;
    fetchData(NIM);
}

Future<void> fetchData(String NIM) async {
    final ref = _databaseReference.child("Mahasiswa").child(NIM);
    final data = await ref.get();
    setState(() {
        mappedData = data.value as Map<dynamic,dynamic>;
    });
}
```



```
class _ProfilePageState extends State<ProfilePage> {
  Map<dynamic, dynamic> mappedData = {};
  String NIM = "";

  @override
  void initState() {
    super.initState();
    NIM = widget.NIM;
    fetchData(NIM);
  }

  Future<void> fetchData(String NIM) async {
    final ref = _databaseReference.child("Mahasiswa").child(NIM);
    final data = await ref.get();
    setState(() {
      mappedData = data.value as Map<dynamic, dynamic>;
    });
  }
```

20. Berikutnya adalah mengisi tampilan **profile.dart**. Isikan kode di dalam **Widget[ ]**



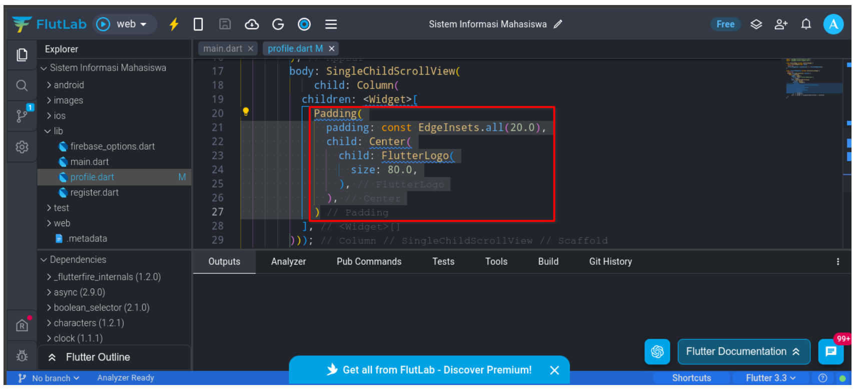
```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('Profile Mahasiswa'),
        ), // AppBar
        body: SingleChildScrollView(
            child: Column(
          children: <Widget>[   ],
        ))); // Column // SingleChildScrollView // Scaffold
  }
}
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

21. Di mulai dari kode **Avatar Profile**

Potongan Kode

```
Padding(
    padding: const EdgeInsets.all(20.0),
        child: Center(
            child: FlutterLogo(
                size: 80.0,
            ),
        ),
),
```

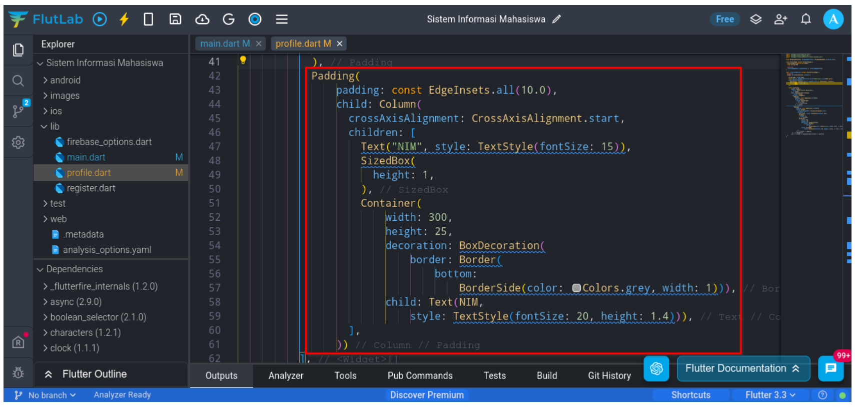


```
        body: SingleChildScrollView(
            child: Column(
          children: <Widget>[
            Padding(
              padding: const EdgeInsets.all(20.0),
              child: Center(
                child: FlutterLogo(
                  size: 80.0,
                ), // FlutterLogo
              ), // Center
            ) // Padding
          ], // <Widget>[]
        ))); // Column // SingleChildScrollView // Scaffold
```

22. Kemudian lanjut kode di atas dengan tampilan untuk NIM.

Potongan Kode

```
Padding(
    padding: const EdgeInsets.all(10.0),
        child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
                Text("NIM", style: TextStyle(fontSize: 15)),
                SizedBox(
                    height: 1,
                  ),
                Container(
                    width: 300,
                    height: 25,
                    decoration: BoxDecoration(
                        border: Border(
                            bottom:
                                BorderSide(color: Colors.grey, width: 1))),
                    child: Text(NIM,
                        style: TextStyle(fontSize: 20, height: 1.4))),
],
)),
```

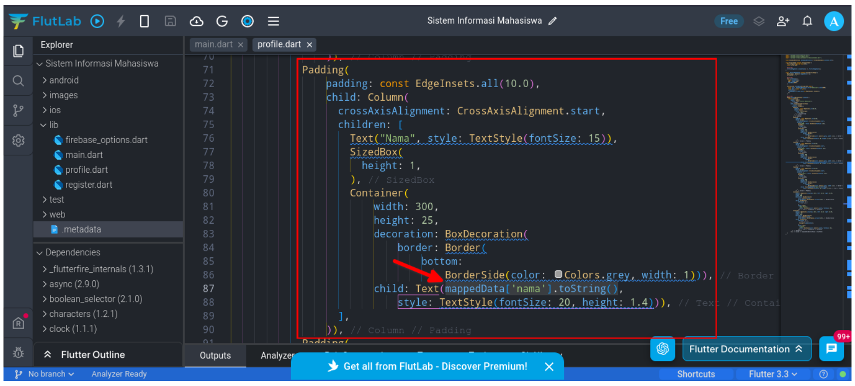


```
), // Padding
Padding(
  padding: const EdgeInsets.all(10.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text("NIM", style: TextStyle(fontSize: 15)),
      SizedBox(
        height: 1,
      ), // SizedBox
      Container(
        width: 300,
        height: 25,
        decoration: BoxDecoration(
          border: Border(
            bottom:
              BorderSide(color: Colors.grey, width: 1))), // Bor
        child: Text(NIM,
          style: TextStyle(fontSize: 20, height: 1.4))), // Text // Co
    ],
  )) // Column // Padding
], // <Widget>[]
```

23. Tempel kode kembali untuk menampilkan **Nama**

Potongan Kode

```
Padding(
    padding: const EdgeInsets.all(10.0),
    child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            Text("Nama", style: TextStyle(fontSize: 15)),
            SizedBox(
                height: 1,
            ),
            Container(
                width: 300,
                height: 25,
                decoration: BoxDecoration(
                    border: Border(
                        bottom:
                            BorderSide(color: Colors.grey, width: 1))),
                child: Text(mappedData['nama'].toString(),
                    style: TextStyle(fontSize: 20, height: 1.4))),
        ],
)),
```

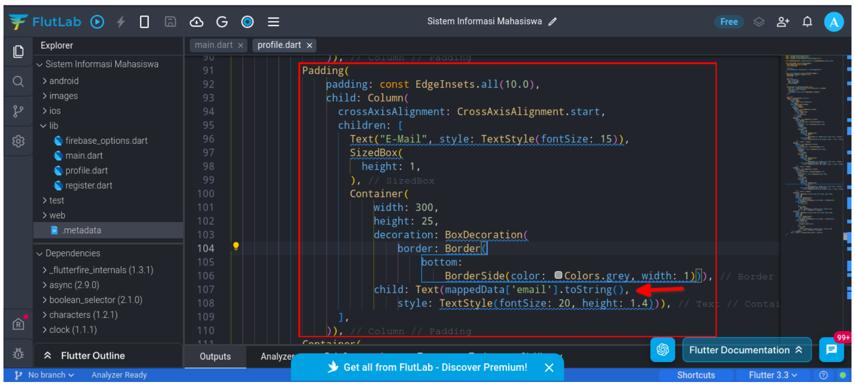


```
Padding(
  padding: const EdgeInsets.all(10.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text("Nama", style: TextStyle(fontSize: 15)),
      SizedBox(
        height: 1,
      ), // SizedBox
      Container(
          width: 300,
          height: 25,
          decoration: BoxDecoration(
              border: Border(
                  bottom:
                      BorderSide(color: Colors.grey, width: 1))), // Border
          child: Text(mappedData['nama'].toString(),
              style: TextStyle(fontSize: 20, height: 1.4))), // Text // Contai
    ],
  )), // Column // Padding
```

24. Tempel kembali untuk menampilkan **E-Mail**

Potongan Kode

```
Padding(
    padding: const EdgeInsets.all(10.0),
    child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            Text("E-Mail", style: TextStyle(fontSize: 15)),
            SizedBox(
                height: 1,
            ),
            Container(
                width: 300,
                height: 25,
                decoration: BoxDecoration(
                    border: Border(
                        bottom:
                            BorderSide(color: Colors.grey, width: 1))),
                child: Text(mappedData['email'].toString(),
                    style: TextStyle(fontSize: 20, height: 1.4))),
        ],
)),
```

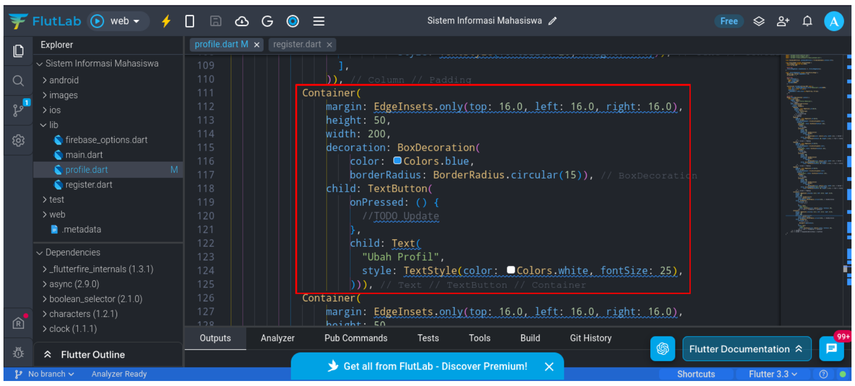


```
        )), // Column // Padding
        Padding(
          padding: const EdgeInsets.all(10.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text("E-Mail", style: TextStyle(fontSize: 15)),
              SizedBox(
                height: 1,
              ), // SizedBox
              Container(
                  width: 300,
                  height: 25,
                  decoration: BoxDecoration(
                      border: Border(
                          bottom:
                              BorderSide(color: Colors.grey, width: 1))), // Border
                  child: Text(mappedData['email'].toString(),
                      style: TextStyle(fontSize: 20, height: 1.4))), // Text // Contai
            ],
        )), // Column // Padding
        Container(
```

25. Kode terakhir yang ditambahkan adalah **Update Profile** dan **Logout**. Letakkan kode setelah kode sebelumnya.

Potongan Kode

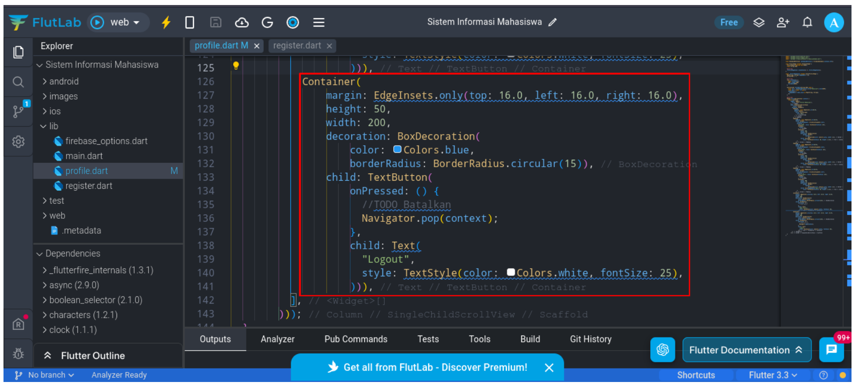
```
Container(
    margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
    height: 50,
    width: 200,
    decoration: BoxDecoration(
        color: Colors.blue,
        borderRadius: BorderRadius.circular(15)),
    child: TextButton(
        onPressed: () {
            //TODO Update
        },
    child: Text(
        "Ubah Profil",
        style: TextStyle(color: Colors.white, fontSize: 25),
))),
```



```
            ],
          )), // Column // Padding
      Container(
          margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
          height: 50,
          width: 200,
          decoration: BoxDecoration(
              color: Colors.blue,
              borderRadius: BorderRadius.circular(15)), // BoxDecoration
          child: TextButton(
              onPressed: () {
                //TODO Update
              },
              child: Text(
                "Ubah Profil",
                style: TextStyle(color: Colors.white, fontSize: 25),
              ))), // Text // TextButton // Container
      Container(
          margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
```

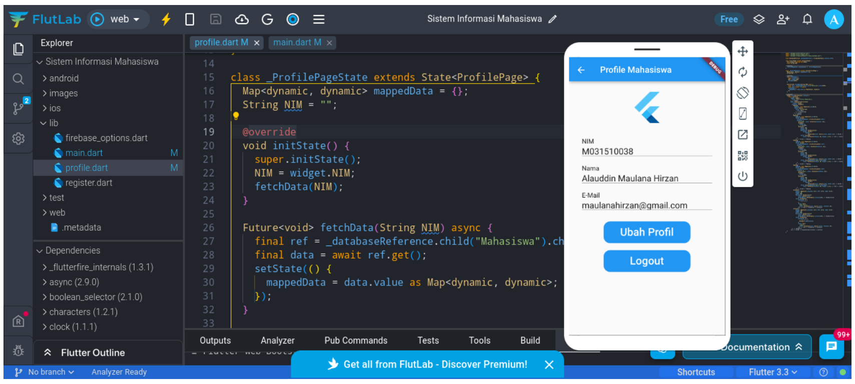
Potongan Kode

```
Container(
    margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
    height: 50,
    width: 200,
    decoration: BoxDecoration(
        color: Colors.blue,
        borderRadius: BorderRadius.circular(15)),
    child: TextButton(
        onPressed: () {
            //TODO Batalkan
            Navigator.pop(context);
    },
    child: Text(
        "Logout",
        style: TextStyle(color: Colors.white, fontSize: 25),
))),
```



```
                    ))), // Text // TextButton // Container
                  Container(
                    margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                    height: 50,
                    width: 200,
                    decoration: BoxDecoration(
                        color: Colors.blue,
                        borderRadius: BorderRadius.circular(15)), // BoxDecoration
                    child: TextButton(
                        onPressed: () {
                          //TODO Batalkan
                          Navigator.pop(context);
                        },
                        child: Text(
                          "Logout",
                          style: TextStyle(color: Colors.white, fontSize: 25),
                        ))), // Text // TextButton // Container
              ], // <Widget>[]
            ))); // Column // SingleChildScrollView // Scaffold
```

26. Jalankan aplikasi dan tes



```
class _ProfilePageState extends State<ProfilePage> {
  Map<dynamic, dynamic> mappedData = {};
  String NIM = "";

  @override
  void initState() {
    super.initState();
    NIM = widget.NIM;
    fetchData(NIM);
  }

  Future<void> fetchData(String NIM) async {
    final ref = _databaseReference.child("Mahasiswa").ch
    final data = await ref.get();
    setState(() {
      mappedData = data.value as Map<dynamic, dynamic>;
    });
  }
```

Profile Mahasiswa

NIM
M031510038

Nama
Alauddin Maulana Hirzan

E-Mail
maulanahirzan@gmail.com

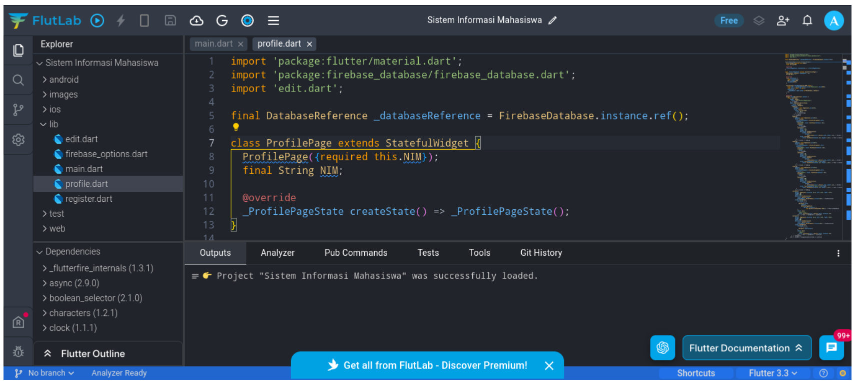
Bab 7

Praktikum 7

7.1 Operasi Update Data

7.2 Tutorial

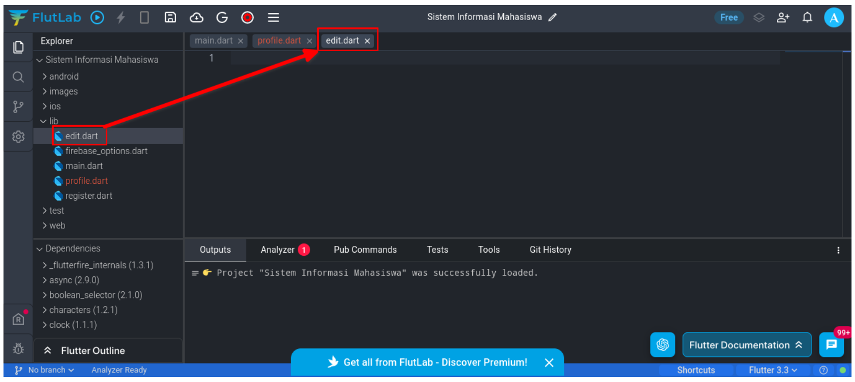
1. Buka kembali projek di Flutlab



```
import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';
import 'edit.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

class ProfilePage extends StatefulWidget {
  ProfilePage({required this.NIM});
  final String NIM;

  @override
  _ProfilePageState createState() => _ProfilePageState();
}
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

2. Buatlah satu file dengan nama **edit.dart**

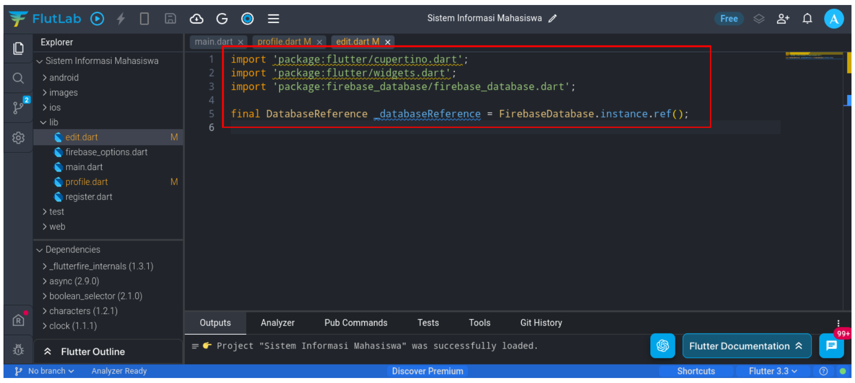


```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

3. Kemudian masukkan kode berikut ke dalam file **edit.dart**

Potongan Kode

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/widgets.dart';
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();
```



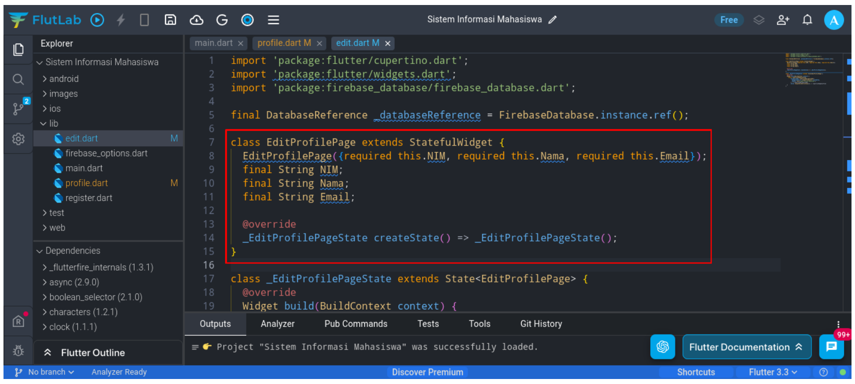
```
import 'package:flutter/cupertino.dart';
import 'package:flutter/widgets.dart';
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

4. Kemudian tambahkan kode berikutnya untuk widget

Potongan Kode

```
class EditProfilePage extends StatefulWidget {
    EditProfilePage({required this.NIM, required this.Nama, required this.Email});
    final String NIM;
    final String Nama;
    final String Email;

    @override
    _EditProfilePageState createState() => _EditProfilePageState();
}
```



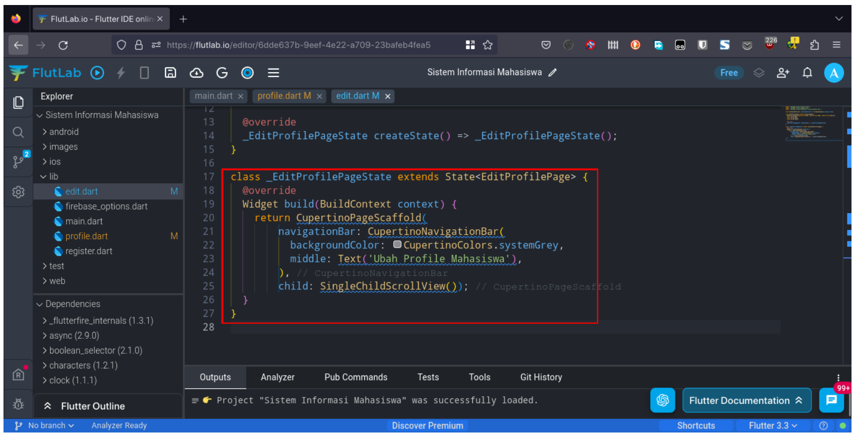
```
import 'package:flutter/cupertino.dart';
import 'package:flutter/widgets.dart';
import 'package:firebase_database/firebase_database.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

class EditProfilePage extends StatefulWidget {
  EditProfilePage({required this.NIM, required this.Nama, required this.Email});
  final String NIM;
  final String Nama;
  final String Email;

  @override
  _EditProfilePageState createState() => _EditProfilePageState();
}

class _EditProfilePageState extends State<EditProfilePage> {
  @override
  Widget build(BuildContext context) {
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

5. Kemudian lanjutkan dengan kode **widget** seperti berikut:

Potongan Kode

```
class _EditProfilePageState extends State<EditProfilePage> {
    @override
    Widget build(BuildContext context) {
        return CupertinoPageScaffold(
            navigationBar: CupertinoNavigationBar(
                backgroundColor: CupertinoColors.systemGrey,
                middle: Text('Ubah Profile Mahasiswa'),
        ),
        child: SingleChildScrollView());
    }
}
```



```
  @override
  _EditProfilePageState createState() => _EditProfilePageState();
}

class _EditProfilePageState extends State<EditProfilePage> {
  @override
  Widget build(BuildContext context) {
    return CupertinoPageScaffold(
        navigationBar: CupertinoNavigationBar(
          backgroundColor: CupertinoColors.systemGrey,
          middle: Text('Ubah Profile Mahasiswa'),
        ), // CupertinoNavigationBar
        child: SingleChildScrollView()); // CupertinoPageScaffold
  }
}
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

6. Buka file **profile.dart**, dan tambahkan kode berikut untuk tombol **Ubah Profil**:

   - **Import** baris awal

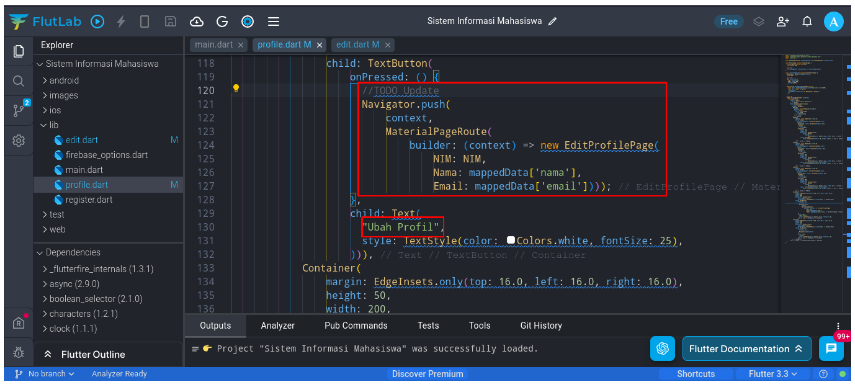
Potongan Kode

```
mport 'edit.dart';
```

   - Tombol **Ubah Profile**:

Potongan Kode

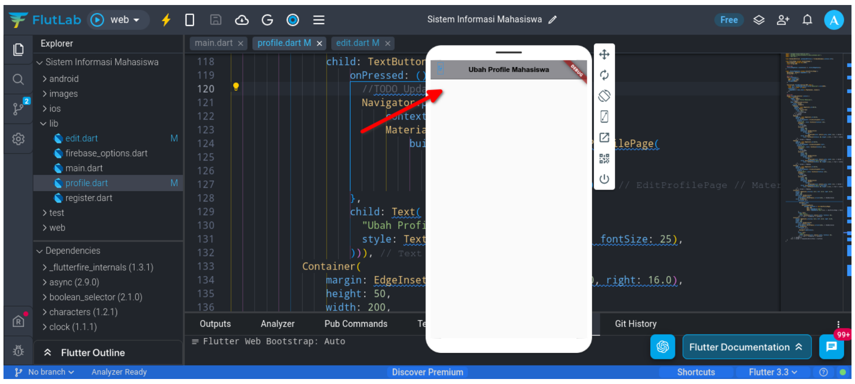
```
//TODO Update
Navigator.push(
    context,
    MaterialPageRoute(
        builder: (context) => new EditProfilePage(
            NIM: NIM,
            Nama: mappedData['nama'],
            Email: mappedData['email']));
```



```
child: TextButton(
    onPressed: () {
      //TODO Update
      Navigator.push(
          context,
          MaterialPageRoute(
              builder: (context) => new EditProfilePage(
                  NIM: NIM,
                  Nama: mappedData['nama'],
                  Email: mappedData['email']))); // EditProfilePage // Mater
    },
    child: Text(
      "Ubah Profil",
      style: TextStyle(color: Colors.white, fontSize: 25),
    ))), // Text // TextButton // Container
Container(
  margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
  height: 50,
  width: 200,
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

7. Tes aplikasi untuk memastikan tombol berfungsi membuka halaman **Update Profile**

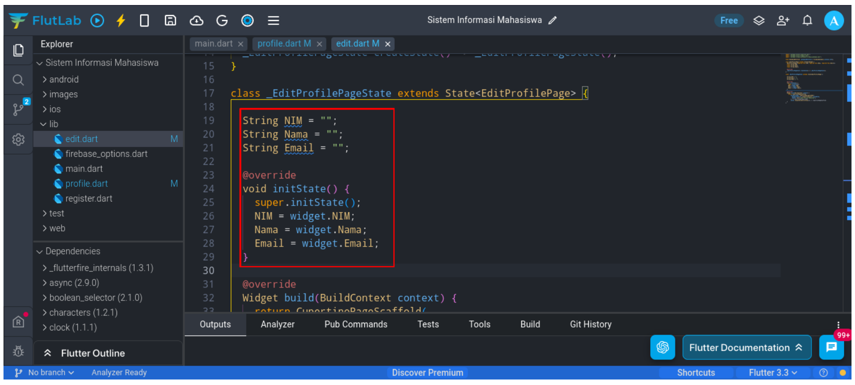


```
Flutter Web Bootstrap: Auto
```

8. Buka kembali file **edit.dart**, dan tambahkan kode berikut tepat di bawah **class _EditProfilePageState extends State<EditProfilePage>** untuk memasukkan data ke **State**

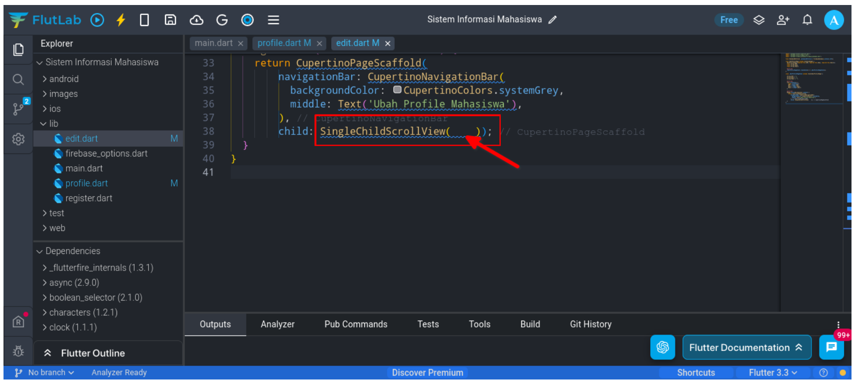
Potongan Kode

```
String NIM = "";
String Nama = "";
String Email = "";

@override
void initState() {
   super.initState();
   NIM = widget.NIM;
   Nama = widget.Nama;
   Email = widget.Email;
}
```



```
}

class _EditProfilePageState extends State<EditProfilePage> {

  String NIM = "";
  String Nama = "";
  String Email = "";

  @override
  void initState() {
    super.initState();
    NIM = widget.NIM;
    Nama = widget.Nama;
    Email = widget.Email;
  }

  @override
  Widget build(BuildContext context) {
```

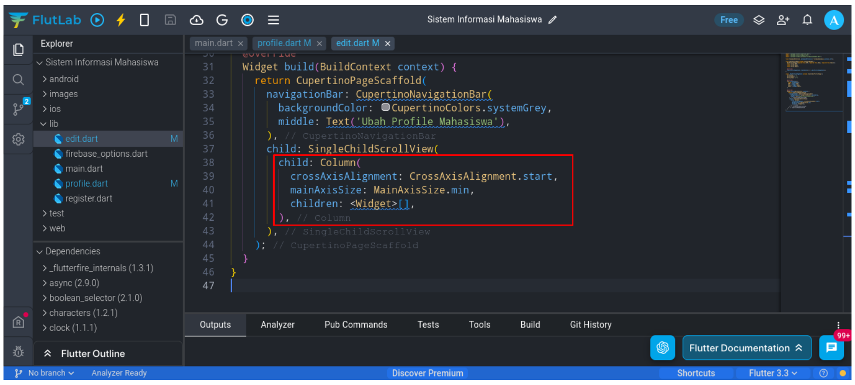
9. Berikutnya adalah menambahkan kode layout. Masukkan tepat di antara **child: SingleChildScrollView( )**



```
    return CupertinoPageScaffold(
      navigationBar: CupertinoNavigationBar(
        backgroundColor: CupertinoColors.systemGrey,
        middle: Text('Ubah Profile Mahasiswa'),
      ), // CupertinoNavigationBar
      child: SingleChildScrollView(    )); // CupertinoPageScaffold
  }
}
```

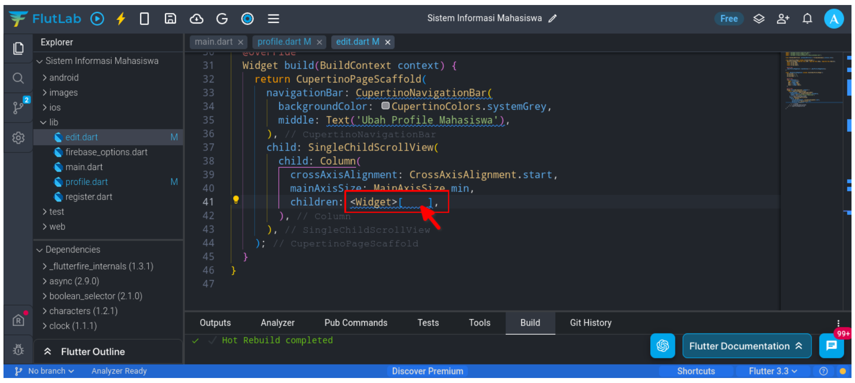
Potongan Kode

```
child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    mainAxisSize: MainAxisSize.min,
    children: <Widget>[],
),
```



```
  Widget build(BuildContext context) {
    return CupertinoPageScaffold(
      navigationBar: CupertinoNavigationBar(
        backgroundColor: CupertinoColors.systemGrey,
        middle: Text('Ubah Profile Mahasiswa'),
      ), // CupertinoNavigationBar
      child: SingleChildScrollView(
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          mainAxisSize: MainAxisSize.min,
          children: <Widget>[],
        ), // Column
      ), // SingleChildScrollView
    ); // CupertinoPageScaffold
  }
}
```

10. Jika sudah, tepat di dalam sintaks **<Widget>[]** masukkan kode berikut:

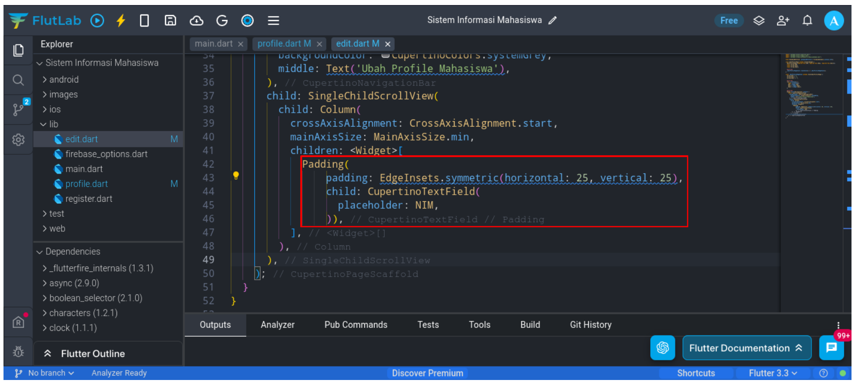


```
Widget build(BuildContext context) {
  return CupertinoPageScaffold(
    navigationBar: CupertinoNavigationBar(
      backgroundColor: CupertinoColors.systemGrey,
      middle: Text('Ubah Profile Mahasiswa'),
    ), // CupertinoNavigationBar
    child: SingleChildScrollView(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisSize: MainAxisSize.min,
        children: <Widget>[    ],
      ), // Column
    ), // SingleChildScrollView
  ); // CupertinoPageScaffold
}
}
```

```
✓ Hot Rebuild completed
```

Potongan Kode

```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
    child: CupertinoTextField(
        placeholder: NIM,
)),
```

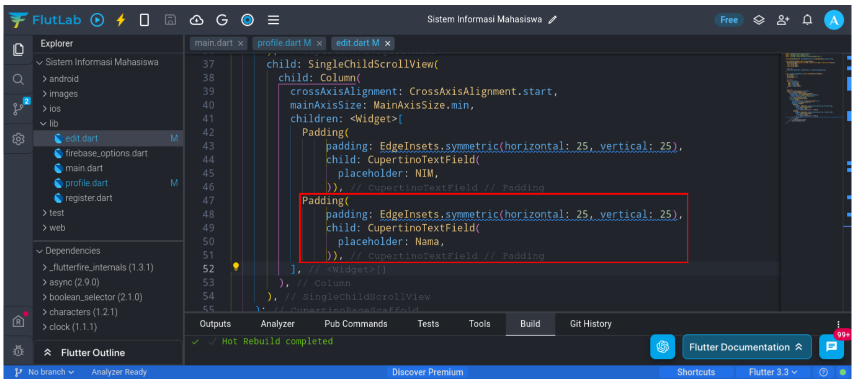


```
      middle: Text('Ubah Profile Mahasiswa'),
    ), // CupertinoNavigationBar
    child: SingleChildScrollView(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisSize: MainAxisSize.min,
        children: <Widget>[
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
              child: CupertinoTextField(
                placeholder: NIM,
              )), // CupertinoTextField // Padding
        ], // <Widget>[]
      ), // Column
    ), // SingleChildScrollView
  ); // CupertinoPageScaffold
}
}
```

11. Tepat di bawahnya, tambahkan kode untuk **Nama** tepat setelah **koma**:

Potongan Kode

```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
    child: CupertinoTextField(
        placeholder: Nama,
)),
```



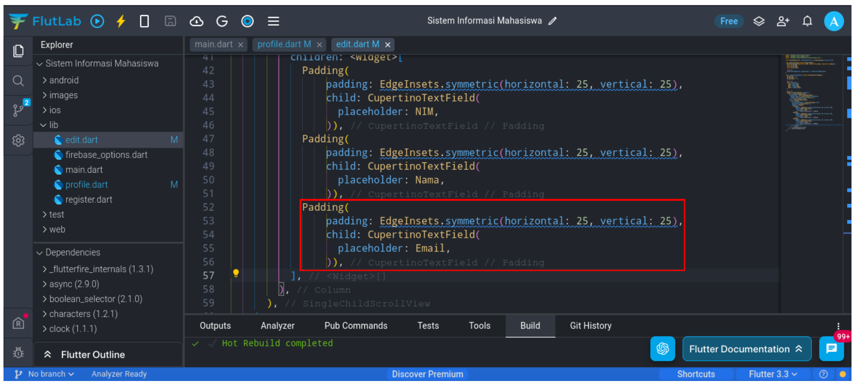
```
    child: SingleChildScrollView(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisSize: MainAxisSize.min,
        children: <Widget>[
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
              child: CupertinoTextField(
                placeholder: NIM,
              )), // CupertinoTextField // Padding
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
              child: CupertinoTextField(
                placeholder: Nama,
              )), // CupertinoTextField // Padding
        ], // <Widget>[]
      ), // Column
    ), // SingleChildScrollView
```

```
✓ Hot Rebuild completed
```

12. Tambahkan **TextField** untuk **Email**. Masukkan setelah tanda **koma**

Potongan Kode

```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
    child: CupertinoTextField(
        placeholder: Email,
)),
```



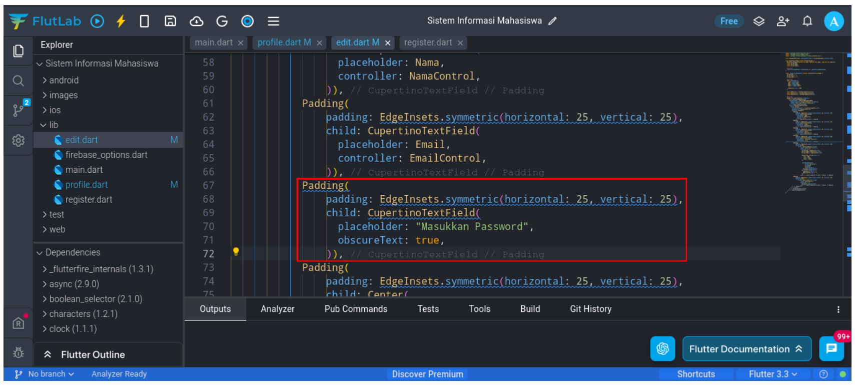
```
              children: <Widget>[
                Padding(
                    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                    child: CupertinoTextField(
                      placeholder: NIM,
                    )), // CupertinoTextField // Padding
                Padding(
                    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                    child: CupertinoTextField(
                      placeholder: Nama,
                    )), // CupertinoTextField // Padding
                Padding(
                    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                    child: CupertinoTextField(
                      placeholder: Email,
                    )), // CupertinoTextField // Padding
              ], // <Widget>[]
            ), // Column
          ), // SingleChildScrollView
```

```
✓ Hot Rebuild completed
```

13. Tambahkan **TextField** terakhir untuk **Password**. Masukkan setelah tanda **koma**

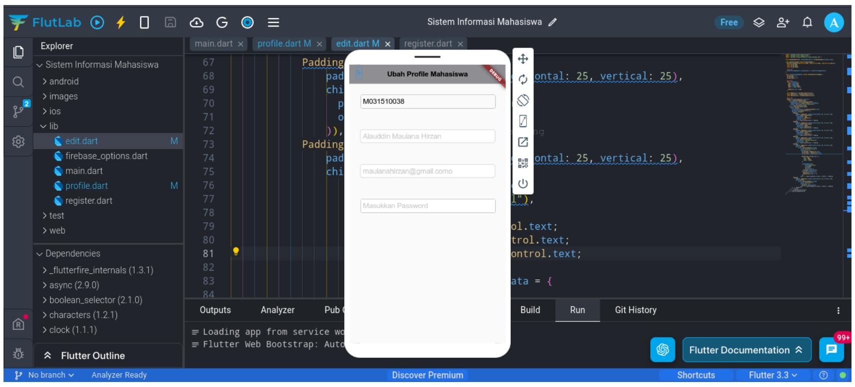
Potongan Kode

```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
        child: CupertinoTextField(
            placeholder: "Masukkan Password",
        obscureText: true,
)),
```



```
                      placeholder: Nama,
                      controller: NamaControl,
                    )), // CupertinoTextField // Padding
                Padding(
                    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                    child: CupertinoTextField(
                      placeholder: Email,
                      controller: EmailControl,
                    )), // CupertinoTextField // Padding
                Padding(
                    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                    child: CupertinoTextField(
                      placeholder: "Masukkan Password",
                      obscureText: true,
                    )), // CupertinoTextField // Padding
                Padding(
                    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                    child: Center(
```

14. Tes aplikasi kembali untuk melihat **progress**. Halaman **Ubah Profil** akan menampilkan data sebelumnya sebagai **Placeholder/Hint**



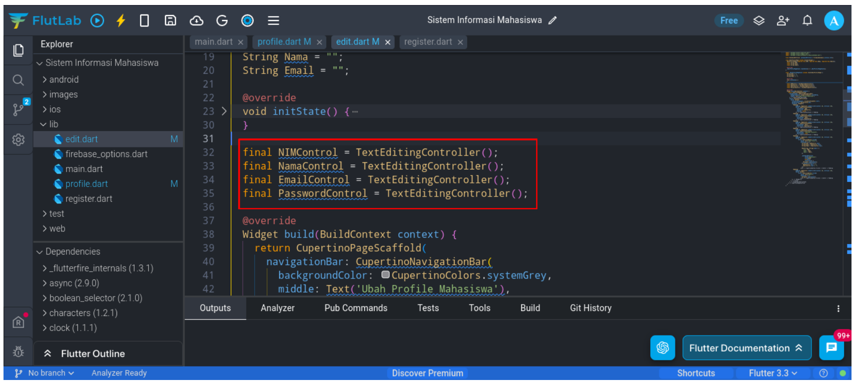
```
Ubah Profile Mahasiswa
M031510038
Alauddin Maulana Hirzan
maulanahirzan@gmail.como
Masukkan Password
```

```
Loading app from service wo
Flutter Web Bootstrap: Auto
```

15. Setelah sukses, tambahkan kode untuk **Controller** masing-masing **TextField**. Letakkan tepat di bawah **void initState()**:

Potongan Kode

```
final NIMControl = TextEditingController();
final NamaControl = TextEditingController();
final EmailControl = TextEditingController();
final PasswordControl = TextEditingController();
```



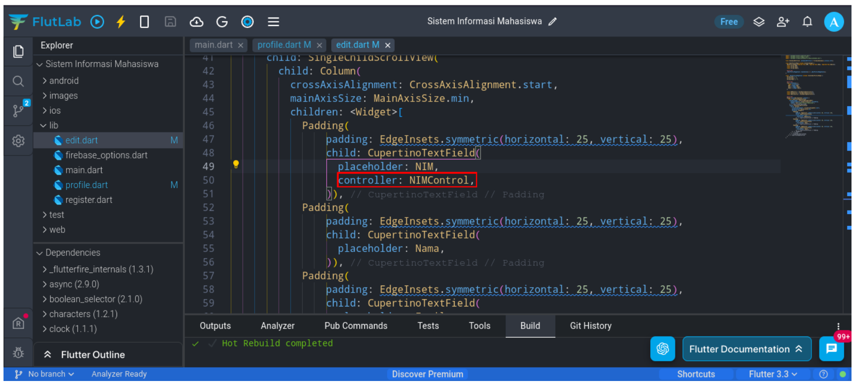
```
String Nama = "";
String Email = "";

@override
void initState() {⋯
}

final NIMControl = TextEditingController();
final NamaControl = TextEditingController();
final EmailControl = TextEditingController();
final PasswordControl = TextEditingController();

@override
Widget build(BuildContext context) {
  return CupertinoPageScaffold(
    navigationBar: CupertinoNavigationBar(
      backgroundColor: CupertinoColors.systemGrey,
      middle: Text('Ubah Profile Mahasiswa'),
```

16. Tempelkan masing-masing **controller** ke **TextField** nya

- **NIM TextField**

Potongan Kode

```
controller: NIMControl,
```



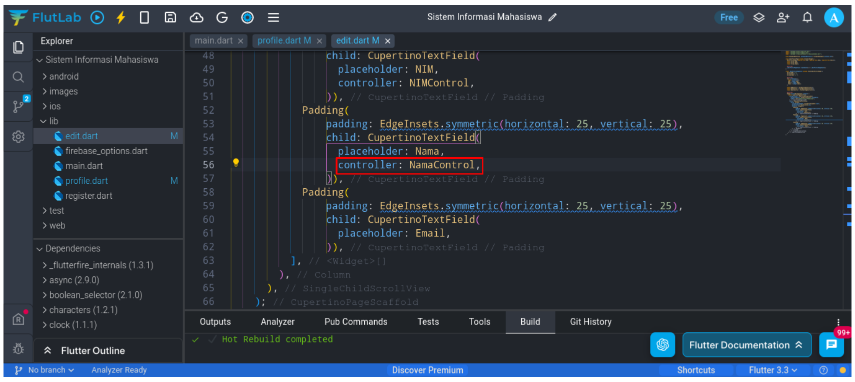
```
        child: SingleChildScrollView(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            mainAxisSize: MainAxisSize.min,
            children: <Widget>[
              Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: CupertinoTextField(
                    placeholder: NIM,
                    controller: NIMControl,
                  )), // CupertinoTextField // Padding
              Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: CupertinoTextField(
                    placeholder: Nama,
                  )), // CupertinoTextField // Padding
              Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: CupertinoTextField(
```

```
Hot Rebuild completed
```

- **Nama TextField**

Potongan Kode

```
controller: NamaControl,
```



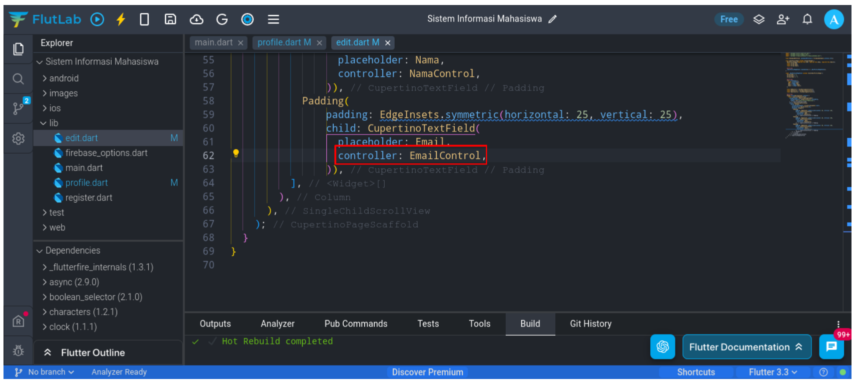
```
                  child: CupertinoTextField(
                    placeholder: NIM,
                    controller: NIMControl,
                  )), // CupertinoTextField // Padding
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: CupertinoTextField(
                    placeholder: Nama,
                    controller: NamaControl,
                  )), // CupertinoTextField // Padding
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: CupertinoTextField(
                    placeholder: Email,
                  )), // CupertinoTextField // Padding
              ], // <Widget>[]
            ), // Column
          ), // SingleChildScrollView
        ); // CupertinoPageScaffold
```

```
Hot Rebuild completed
```

- Email TextField

Potongan Kode

```
controller: EmailControl,
```



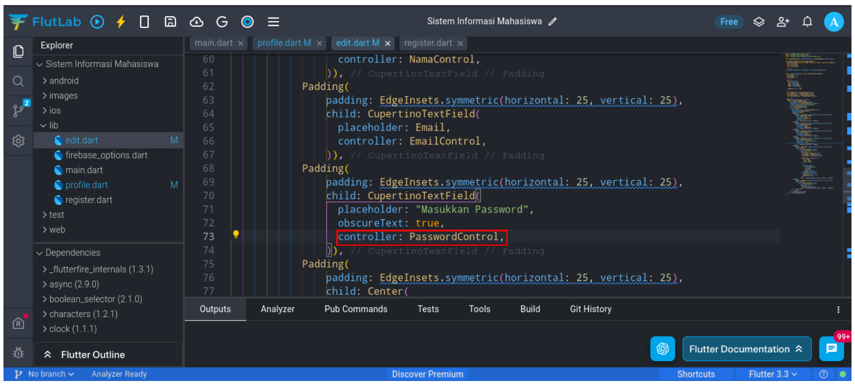
```
                    placeholder: Nama,
                    controller: NamaControl,
                  )), // CupertinoTextField // Padding
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: CupertinoTextField(
                    placeholder: Email,
                    controller: EmailControl,
                  )), // CupertinoTextField // Padding
              ], // <Widget>[]
            ), // Column
          ), // SingleChildScrollView
        ); // CupertinoPageScaffold
      }
    }
```

```
Hot Rebuild completed
```

- Password TextField

Potongan Kode

```
controller: PasswordControl,
```

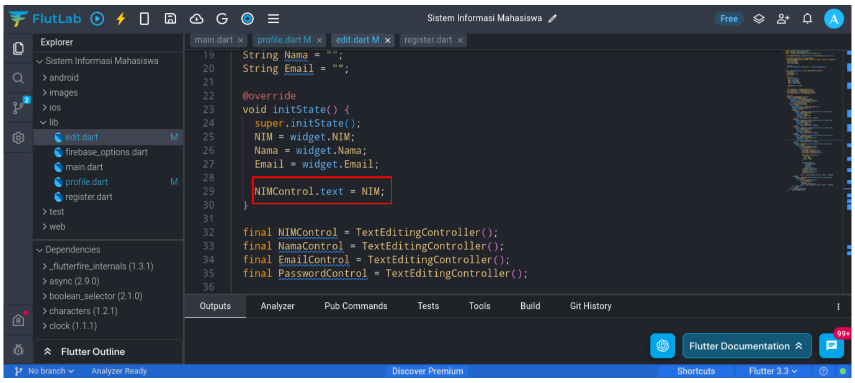


```
                    controller: NamaControl,
                  )), // CupertinoTextField // Padding
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: CupertinoTextField(
                    placeholder: Email,
                    controller: EmailControl,
                  )), // CupertinoTextField // Padding
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: CupertinoTextField(
                    placeholder: "Masukkan Password",
                    obscureText: true,
                    controller: PasswordControl,
                  )), // CupertinoTextField // Padding
                Padding(
                  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
                  child: Center(
```

17. Agar NIM ditampilkan dalam bentuk **Teks** masukkan kode berikut di dalam fungsi **void initState()**

Potongan Kode

```
NIMControl.text = NIM;
```

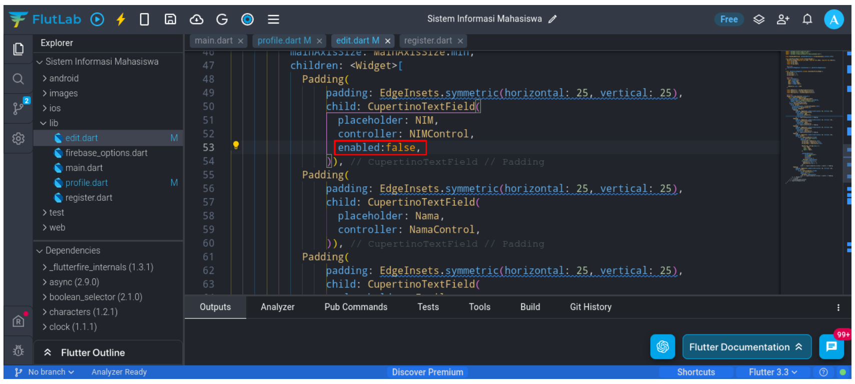


```
String Nama = "";
String Email = "";

@override
void initState() {
  super.initState();
  NIM = widget.NIM;
  Nama = widget.Nama;
  Email = widget.Email;

  NIMControl.text = NIM;
}

final NIMControl = TextEditingController();
final NamaControl = TextEditingController();
final EmailControl = TextEditingController();
final PasswordControl = TextEditingController();
```

18. Lalu matikan input dengan menambahkan kode berikut setelah **controller: NIMControl,**

Potongan Kode

```
enabled:false,
```

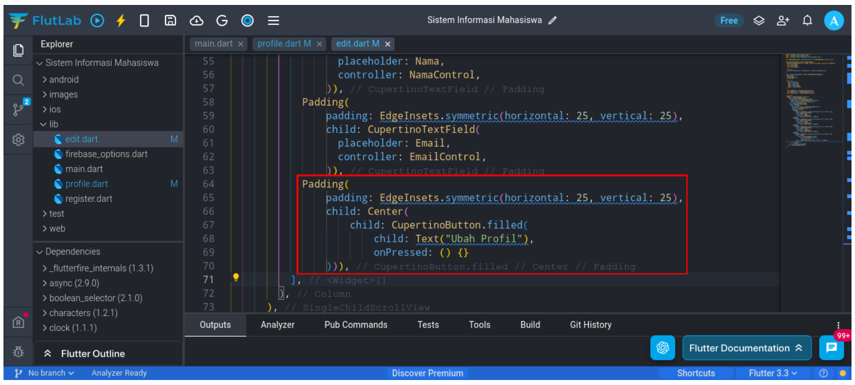


```
        mainAxisSize: MainAxisSize.min,
        children: <Widget>[
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
              child: CupertinoTextField(
                placeholder: NIM,
                controller: NIMControl,
                enabled:false,
              )), // CupertinoTextField // Padding
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
              child: CupertinoTextField(
                placeholder: Nama,
                controller: NamaControl,
              )), // CupertinoTextField // Padding
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
              child: CupertinoTextField(
```

19. Tambahkan kode **Tombol** yang diletakkan satu level dengan **Padding Email**.

Potongan Kode

```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
    child: Center(
        child: CupertinoButton.filled(
            child: Text("Ubah Profil"),
            onPressed: () {}
))),
```

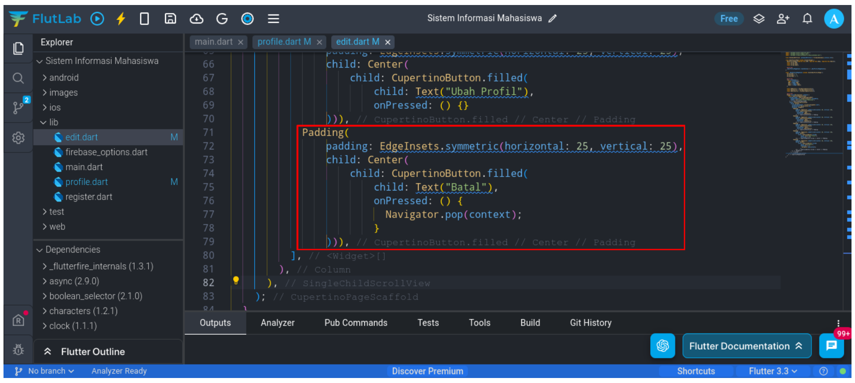


```
                placeholder: Nama,
                controller: NamaControl,
              )), // CupertinoTextField // Padding
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
              child: CupertinoTextField(
                placeholder: Email,
                controller: EmailControl,
              )), // CupertinoTextField // Padding
          Padding(
              padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
              child: Center(
                  child: CupertinoButton.filled(
                      child: Text("Ubah Profil"),
                      onPressed: () {}
              ))), // CupertinoButton.filled // Center // Padding
        ], // <Widget>[]
      ), // Column
    ), // SingleChildScrollView
```

20. Tambahkan **tombol** terakhir:

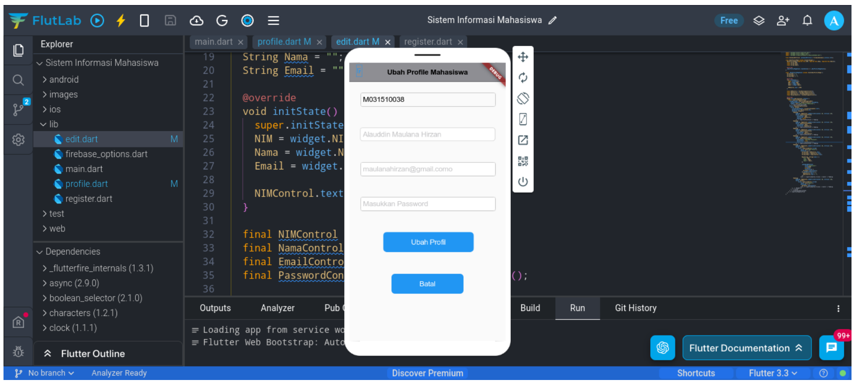
Potongan Kode

```
Padding(
    padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
    child: Center(
    child: CupertinoButton.filled(
        child: Text("Batal"),
        onPressed: () {
            Navigator.pop(context);
    }
))),
```



```
            child: Center(
              child: CupertinoButton.filled(
                child: Text("Ubah Profil"),
                onPressed: () {}
            ))), // CupertinoButton.filled // Center // Padding
          Padding(
            padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
            child: Center(
              child: CupertinoButton.filled(
                child: Text("Batal"),
                onPressed: () {
                  Navigator.pop(context);
                }
            ))), // CupertinoButton.filled // Center // Padding
        ], // <Widget>[]
      ), // Column
    ), // SingleChildScrollView
  ); // CupertinoPageScaffold
```

21. Tes aplikasi lagi untuk melihat hasil keseluruhan antarmuka



```
String Nama = "";
String Email = ""

@override
void initState()
  super.initState
  NIM = widget.NI
  Nama = widget.N
  Email = widget.

  NIMControl.text
}

final NIMControl
final NamaControl
final EmailContro
final PasswordCon
```

```
Loading app from service wo
Flutter Web Bootstrap: Auto
```

Ubah Profile Mahasiswa

M031510038

Alauddin Maulana Hirzan

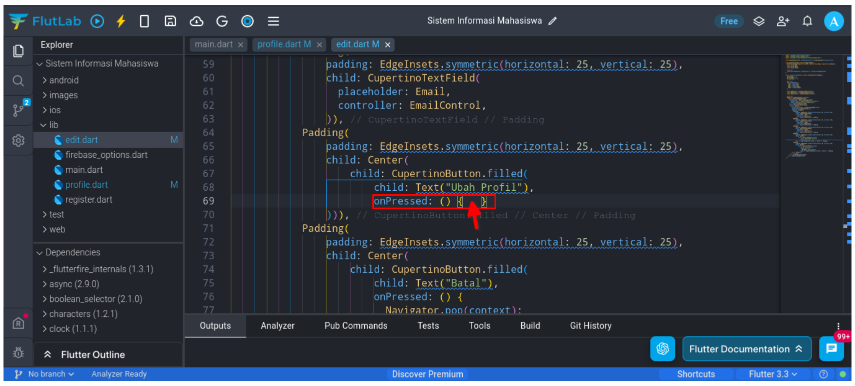
maulanahirzan@gmail.como

Masukkan Password

Ubah Profil

Batal

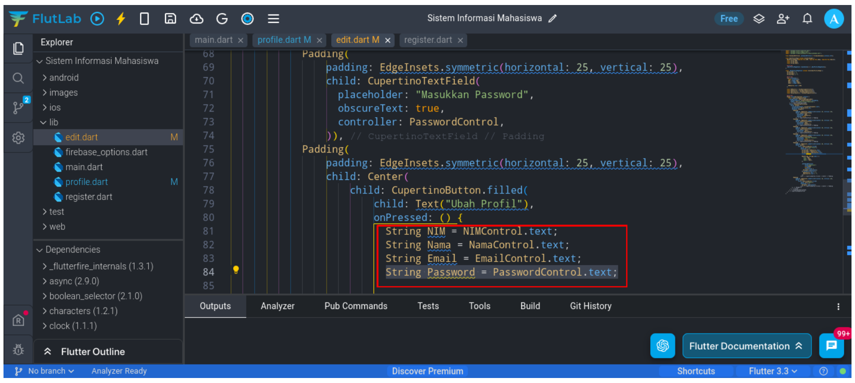
22. Bagian terakhir dari praktikum ini adalah menambahkan kode untuk melakukan update. Tambahkan kode berikut untuk tombol **Update Profil** di bagian **onPressed** ()



```
        padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
        child: CupertinoTextField(
          placeholder: Email,
          controller: EmailControl,
        )), // CupertinoTextField // Padding
    Padding(
        padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
        child: Center(
            child: CupertinoButton.filled(
                child: Text("Ubah Profil"),
                onPressed: () {  }
        ))), // CupertinoButton.filled // Center // Padding
    Padding(
        padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
        child: Center(
            child: CupertinoButton.filled(
                child: Text("Batal"),
                onPressed: () {
                  Navigator.pop(context);
```

Potongan Kode

```
String NIM = NIMControl.text;
String Nama = NamaControl.text;
String Email = EmailControl.text;
String Password = PasswordControl.text;
```

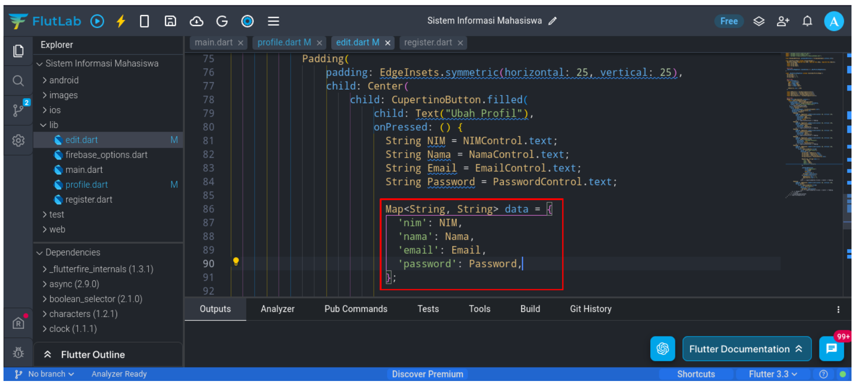


```
    Padding(
        padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
        child: CupertinoTextField(
          placeholder: "Masukkan Password",
          obscureText: true,
          controller: PasswordControl,
        )), // CupertinoTextField // Padding
    Padding(
        padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
        child: Center(
            child: CupertinoButton.filled(
                child: Text("Ubah Profil"),
                onPressed: () {
                  String NIM = NIMControl.text;
                  String Nama = NamaControl.text;
                  String Email = EmailControl.text;
                  String Password = PasswordControl.text;
```

23. Lalu petakan variabel dalam bentuk **Map** dengan kode berikut

Potongan Kode

```
Map<String, String> data = {
    'nim': NIM,
    'nama': Nama,
    'email': Email,
    'password': Password,
};
```

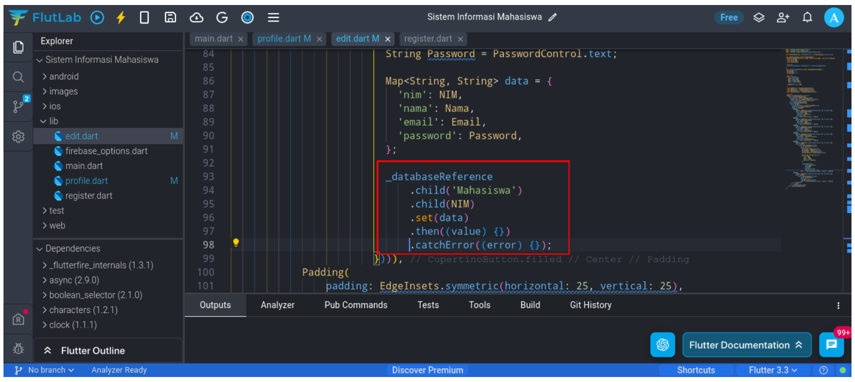


```
    Padding(
        padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
        child: Center(
            child: CupertinoButton.filled(
                child: Text("Ubah Profil"),
                onPressed: () {
                  String NIM = NIMControl.text;
                  String Nama = NamaControl.text;
                  String Email = EmailControl.text;
                  String Password = PasswordControl.text;

                  Map<String, String> data = {
                    'nim': NIM,
                    'nama': Nama,
                    'email': Email,
                    'password': Password,
                  };
```

24. Jika sudah lanjutkan dengan kode **Query** seperti berikut

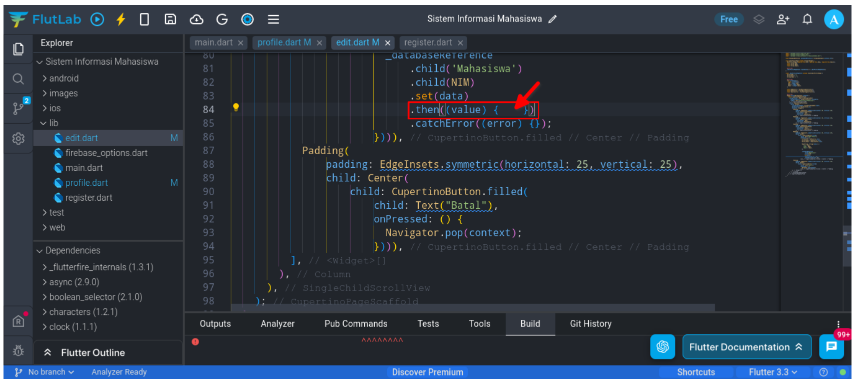
Potongan Kode

```
_databaseReference
    .child('Mahasiswa')
    .child(NIM)
    .set(data)
    .then((value) {})
    .catchError((error){});
```



```
String Password = PasswordControl.text;

Map<String, String> data = {
  'nim': NIM,
  'nama': Nama,
  'email': Email,
  'password': Password,
};

_databaseReference
    .child('Mahasiswa')
    .child(NIM)
    .set(data)
    .then((value) {})
    .catchError((error) {});
})), // CupertinoButton.filled // Center // Padding
Padding(
  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
```

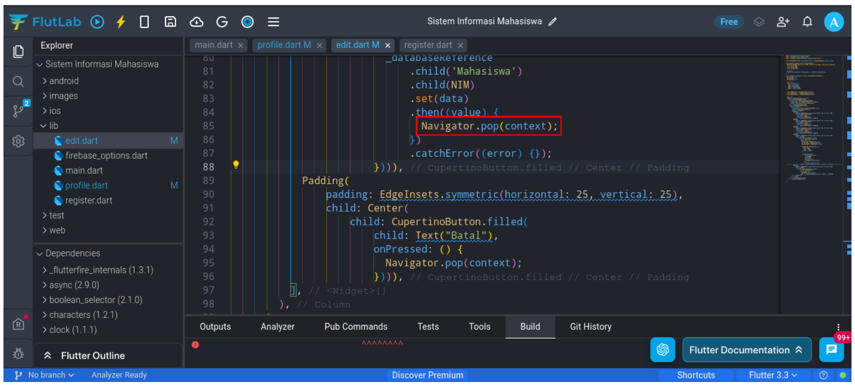
25. Untuk merespon keberhasilan Query, masukkan kode berikut tepat di dalam **.then((value){})**



```
_databaseReference
    .child('Mahasiswa')
    .child(NIM)
    .set(data)
    .then((value) {    })
    .catchError((error) {});
})), // CupertinoButton.filled // Center // Padding
Padding(
  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
  child: Center(
    child: CupertinoButton.filled(
      child: Text("Batal"),
      onPressed: () {
        Navigator.pop(context);
      })), // CupertinoButton.filled // Center // Padding
], // <Widget>[]
), // Column
), // SingleChildScrollView
); // CupertinoPageScaffold
```

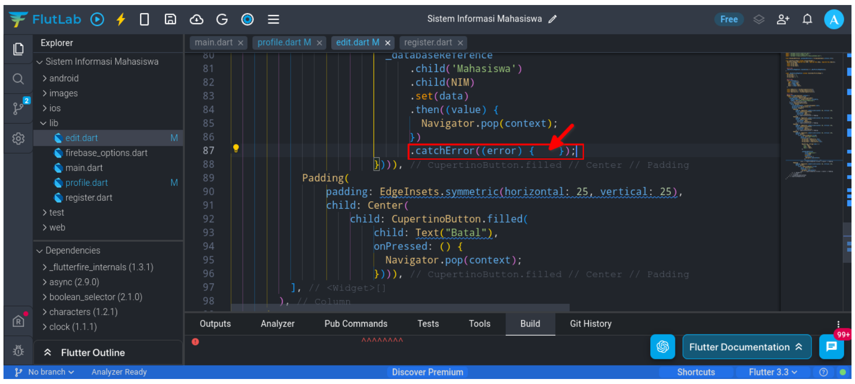
Potongan Kode

```
Navigator.pop(context);
```



```
_databaseReference
    .child('Mahasiswa')
    .child(NIM)
    .set(data)
    .then((value) {
      Navigator.pop(context);
    })
    .catchError((error) {});
})), // CupertinoButton.filled // Center // Padding
Padding(
  padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
  child: Center(
    child: CupertinoButton.filled(
      child: Text("Batal"),
      onPressed: () {
        Navigator.pop(context);
      })), // CupertinoButton.filled // Center // Padding
], // <Widget>[]
), // Column
```

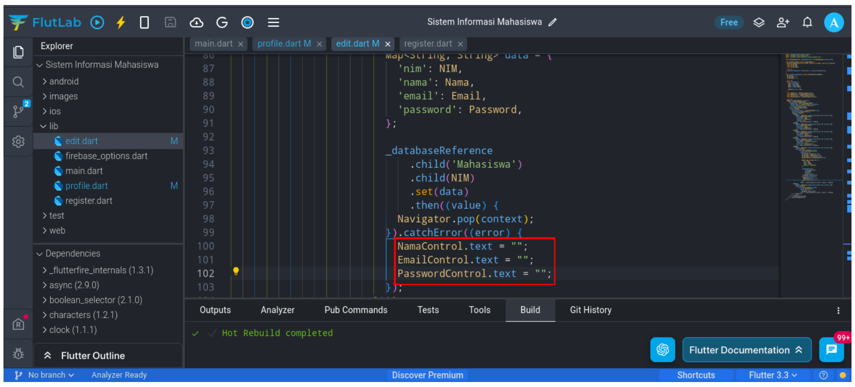
26. Terakhir, untuk merespon kegagalan kueri. Masukkan kode untuk mengosongkan semua **TextField**. Masukkan di dalam kode **.catchError((error) {});**



```
              _databaseReference
                  .child('Mahasiswa')
                  .child(NIM)
                  .set(data)
                  .then((value) {
                    Navigator.pop(context);
                  })
                  .catchError((error) {    });
            }))), // CupertinoButton.filled // Center // Padding
      Padding(
          padding: EdgeInsets.symmetric(horizontal: 25, vertical: 25),
          child: Center(
              child: CupertinoButton.filled(
                  child: Text("Batal"),
                  onPressed: () {
                    Navigator.pop(context);
                  }))), // CupertinoButton.filled // Center // Padding
    ], // <Widget>[]
  ), // Column
```

Potongan Kode

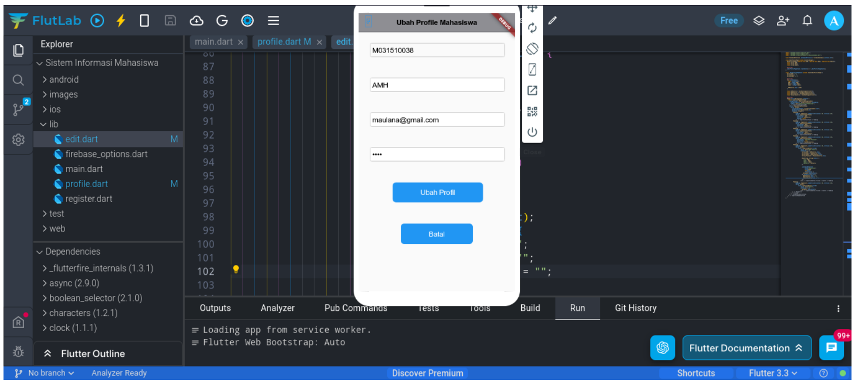
```
NamaControl.text = "";
EmailControl.text= "";
PasswordControl.text = "";
```



```
              'nim': NIM,
              'nama': Nama,
              'email': Email,
              'password': Password,
            };

            _databaseReference
                .child('Mahasiswa')
                .child(NIM)
                .set(data)
                .then((value) {
              Navigator.pop(context);
            }).catchError((error) {
              NamaControl.text = "";
              EmailControl.text = "";
              PasswordControl.text = "";
            });
```

```
✓ Hot Rebuild completed
```

27. Tes aplikasi



Ubah Profile Mahasiswa

M031510038

AMH

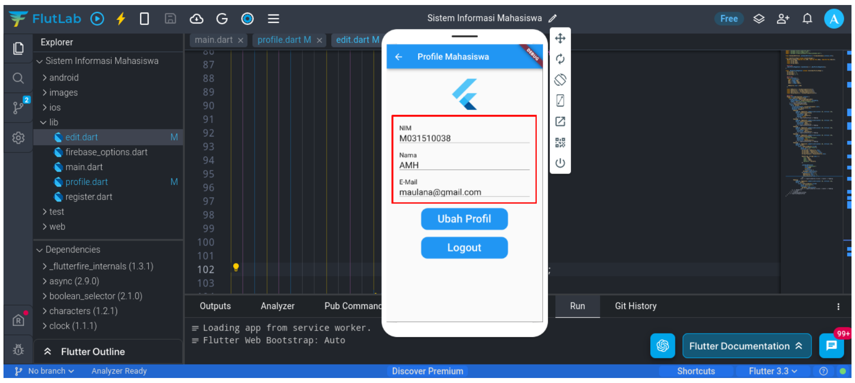
maulana@gmail.com

••••

Ubah Profil

Batal

```
Loading app from service worker.
Flutter Web Bootstrap: Auto
```



Profile Mahasiswa

NIM
M031510038

Nama
AMH

E-Mail
maulana@gmail.com

Ubah Profil

Logout

```
Loading app from service worker.
Flutter Web Bootstrap: Auto
```

Bab 8

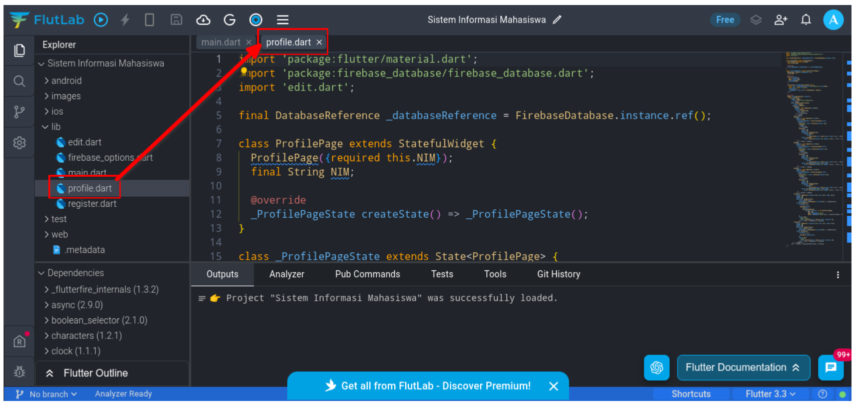
Praktikum 8

8.1 Operasi Delete Data

Di bagian ini mahasiswa diajarkan bagaimana mengunggah gambar melalui aplikasi, yang kemudian ditampilkan di halaman profile. Mahasiswa diwajibkan menyelesaikan Praktikum 7

8.2 Tutorial

1. Buka kembali projek di web **flutlab**. Buka kembali file **profile.dart**



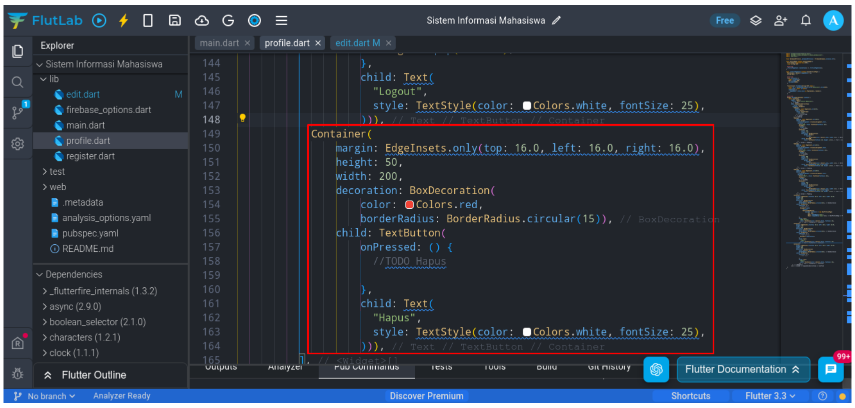
```dart
import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';
import 'edit.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

class ProfilePage extends StatefulWidget {
  ProfilePage({required this.NIM});
  final String NIM;

  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

2. Tambahkan satu tombol baru yang berfungsi untuk menghapus data, dan **Logout** ketika sukses.

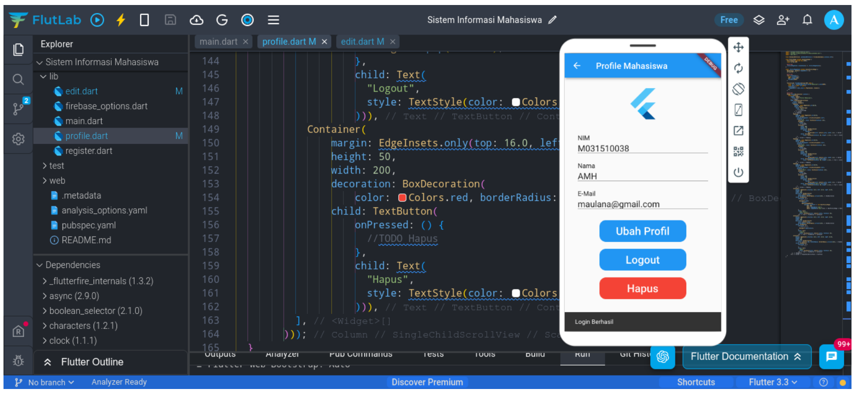
Potongan Kode

```
Container(
    margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
    height: 50,
    width: 200,
    decoration: BoxDecoration(
        color: Colors.red,
        borderRadius: BorderRadius.circular(15)),
        child: TextButton(
            onPressed: () {
                //TODO Hapus

        },
        child: Text(
            "Hapus",
            style: TextStyle(color: Colors.white, fontSize: 25),
))),
```



```
    },
    child: Text(
      "Logout",
      style: TextStyle(color: Colors.white, fontSize: 25),
    ))), // Text // TextButton // Container
Container(
    margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
    height: 50,
    width: 200,
    decoration: BoxDecoration(
        color: Colors.red,
        borderRadius: BorderRadius.circular(15)), // BoxDecoration
    child: TextButton(
        onPressed: () {
          //TODO Hapus

        },
        child: Text(
          "Hapus",
          style: TextStyle(color: Colors.white, fontSize: 25),
        ))), // Text // TextButton // Container
```

3. Jalankan aplikasi untuk melihat hasil sementara aplikasi



Profile Mahasiswa

NIM: M031510038

Nama: AMH

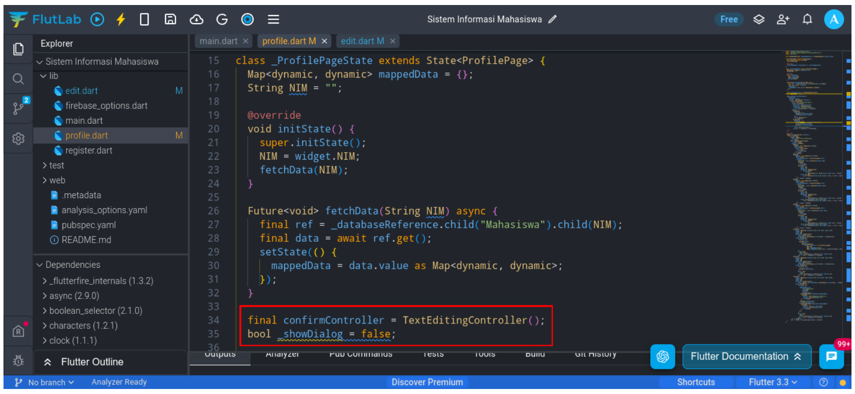
E-Mail: maulana@gmail.com

Login Berhasil

4. Jika tombol sudah ditambahkan, berikutnya menambahkan variabel dan fungsi untuk menampilkan **Kotak Dialog Konfirmasi dan Input Password**. Masukkan kode tepat di bawah fungsi **Future<void> fetchData** (**BUKAN DI DALAMNYA**).

Potongan Kode

```
final confirmController = TextEditingController();
bool _showDialog = false;
```

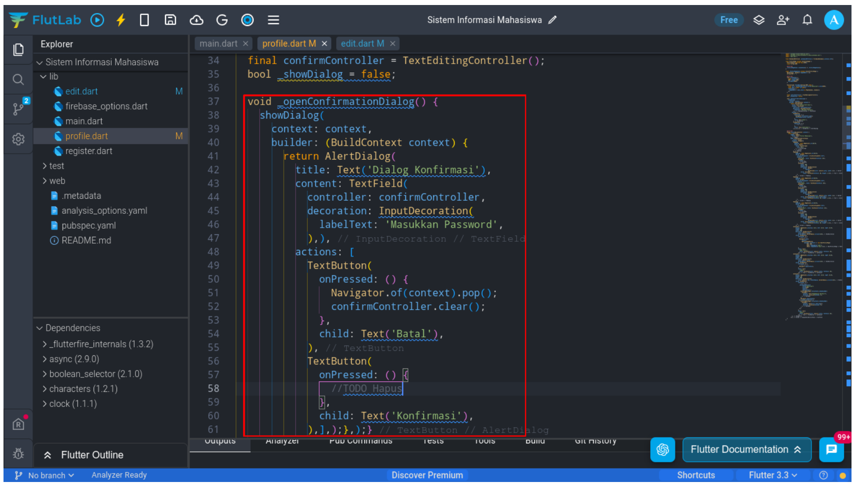


```
class _ProfilePageState extends State<ProfilePage> {
  Map<dynamic, dynamic> mappedData = {};
  String NIM = "";

  @override
  void initState() {
    super.initState();
    NIM = widget.NIM;
    fetchData(NIM);
  }

  Future<void> fetchData(String NIM) async {
    final ref = _databaseReference.child("Mahasiswa").child(NIM);
    final data = await ref.get();
    setState(() {
      mappedData = data.value as Map<dynamic, dynamic>;
    });
  }

  final confirmController = TextEditingController();
  bool _showDialog = false;
```

5. Untuk membangun **Dialog Konfirmasi**, lanjutkan kode di atas dengan kode berikut:

Potongan Kode

```
void _openConfirmationDialog() {
    showDialog(
        context: context,
        builder: (BuildContext context) {
            return AlertDialog(
                title: Text('Dialog Konfirmasi'),
                content: TextField(
                    controller: confirmController,
                    decoration: InputDecoration(
                        labelText: 'Masukkan Password',),),
                    actions: [
                        TextButton(
                            onPressed: () {
                                Navigator.of(context).pop();
                                confirmController.clear();
                            },
                            child: Text('Batal'),),
                        TextButton(
                            onPressed: () {
                                //TODO Hapus
                            },
                            child: Text('Konfirmasi'),
),],);},);}
```

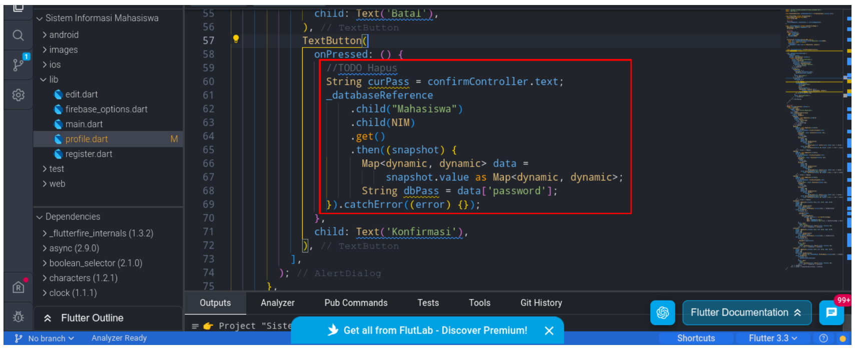


```
final confirmController = TextEditingController();
bool _showDialog = false;

void _openConfirmationDialog() {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(
        title: Text('Dialog Konfirmasi'),
        content: TextField(
          controller: confirmController,
          decoration: InputDecoration(
            labelText: 'Masukkan Password',
          ),), // InputDecoration // TextField
        actions: [
          TextButton(
            onPressed: () {
              Navigator.of(context).pop();
              confirmController.clear();
            },
            child: Text('Batal'),
          ), // TextButton
          TextButton(
            onPressed: () {
              //TODO Hapus
            },
            child: Text('Konfirmasi'),
          ),],);},);} // TextButton // AlertDialog
```

6. Agar fungsi ini dapat menjalankan tugasnya, masukkan kode berikut tepat di bawah tulisan //**TODO Hapus**

Potongan Kode

```
String curPass = confirmController.text;
_databaseReference
    .child("Mahasiswa")
    .child(NIM)
    .get()
    .then((snapshot) {
        Map<dynamic, dynamic> data =
            snapshot.value as Map<dynamic, dynamic>;
        String dbPass = data['password'];
}).catchError((error) {});
```

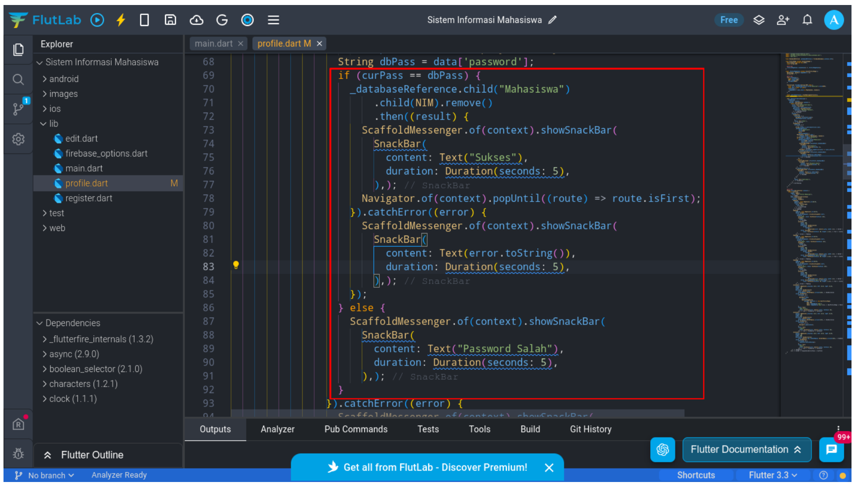


```
            child: Text('Batal'),
          ), // TextButton
          TextButton(
            onPressed: () {
              //TODO Hapus
              String curPass = confirmController.text;
              _databaseReference
                  .child("Mahasiswa")
                  .child(NIM)
                  .get()
                  .then((snapshot) {
                    Map<dynamic, dynamic> data =
                        snapshot.value as Map<dynamic, dynamic>;
                    String dbPass = data['password'];
              }).catchError((error) {});
            },
            child: Text('Konfirmasi'),
          ), // TextButton
        ],
      ); // AlertDialog
    },
```

7. Kemudian lanjutkan bagian bawah kode dari **String dbPass = data['password'];** dengan kode berikut:

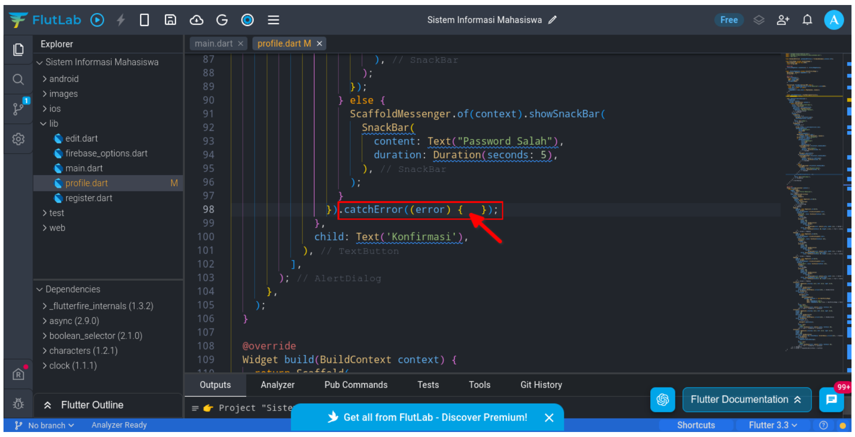
Potongan Kode

```
if (curPass == dbPass) {
    _databaseReference.child("Mahasiswa")
        .child(NIM).remove()
        .then((result) {
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                    content: Text("Sukses"),
                    duration: Duration(seconds: 5),
                ),);
                Navigator.of(context).popUntil((route) => route.isFirst);
        }).catchError((error) {
            ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                    content: Text(error.toString()),
                    duration: Duration(seconds: 5),
                ),);
               });
} else {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text("Password Salah"),
            duration: Duration(seconds: 5),
        ),);
}
```



```
String dbPass = data['password'];
if (curPass == dbPass) {
  _databaseReference.child("Mahasiswa")
      .child(NIM).remove()
      .then((result) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text("Sukses"),
        duration: Duration(seconds: 5),
      ),); // SnackBar
    Navigator.of(context).popUntil((route) => route.isFirst);
  }).catchError((error) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text(error.toString()),
        duration: Duration(seconds: 5),
      ),); // SnackBar
  });
} else {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text("Password Salah"),
      duration: Duration(seconds: 5),
    ),); // SnackBar
}
}).catchError((error) {
```

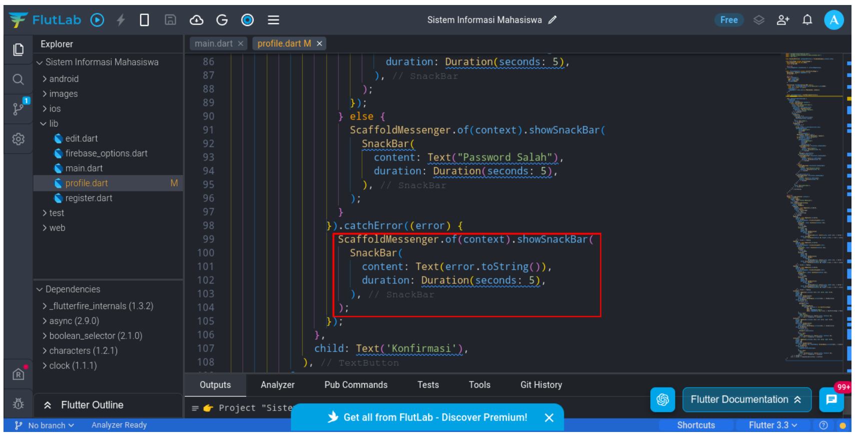
8. Berikutnya adalah menambahkan kode error untuk bagian kode **.catchError((error) {});**



```
                  ), // SnackBar
                );
              });
            } else {
              ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                  content: Text("Password Salah"),
                  duration: Duration(seconds: 5),
                ), // SnackBar
              );
            }
          }).catchError((error) {   });
        },
        child: Text('Konfirmasi'),
      ), // TextButton
    ],
  ); // AlertDialog
 },
);
}

@override
Widget build(BuildContext context) {
```

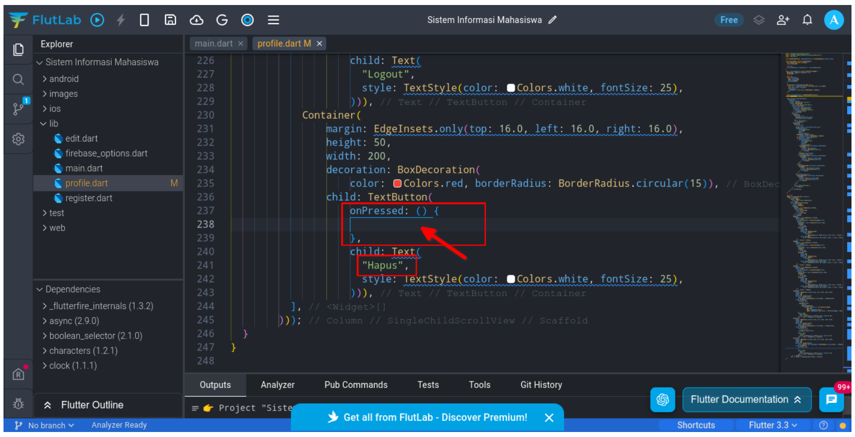
Potongan Kode

```
ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
        content: Text(error.toString()),
        duration: Duration(seconds: 5),
    ),
);
```



```
                    duration: Duration(seconds: 5),
                  ), // SnackBar
                );
              });
            } else {
              ScaffoldMessenger.of(context).showSnackBar(
                SnackBar(
                  content: Text("Password Salah"),
                  duration: Duration(seconds: 5),
                ), // SnackBar
              );
            }
          }).catchError((error) {
            ScaffoldMessenger.of(context).showSnackBar(
              SnackBar(
                content: Text(error.toString()),
                duration: Duration(seconds: 5),
              ), // SnackBar
            );
          });
        },
        child: Text('Konfirmasi'),
      ), // TextButton
```

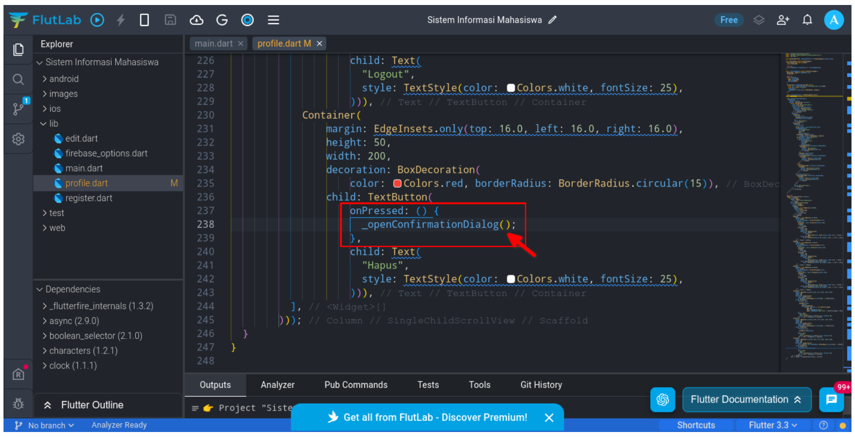
9. Bagian terakhir yang perlu ditambahkan adalah kode untuk **Tombol Hapus** itu sendiri



```
              child: Text(
                "Logout",
                style: TextStyle(color: Colors.white, fontSize: 25),
              ))), // Text // TextButton // Container
          Container(
            margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
            height: 50,
            width: 200,
            decoration: BoxDecoration(
              color: Colors.red, borderRadius: BorderRadius.circular(15)), // BoxDec
            child: TextButton(
              onPressed: () {

              },
              child: Text(
                "Hapus",
                style: TextStyle(color: Colors.white, fontSize: 25),
              ))), // Text // TextButton // Container
        ], // <Widget>[]
      ))); // Column // SingleChildScrollView // Scaffold
  }
}
```

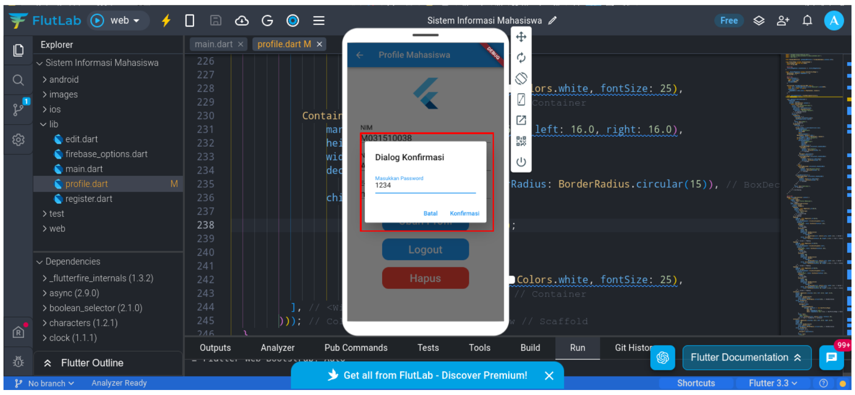
Potongan Kode

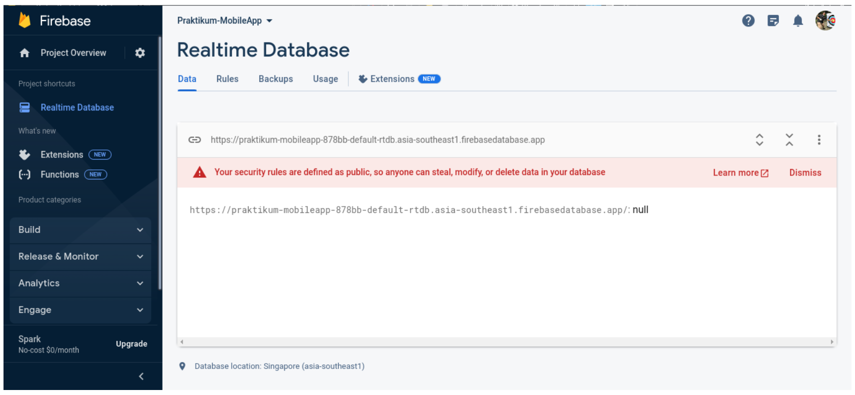
```
_openConfirmationDialog();
```



```
                  child: Text(
                    "Logout",
                    style: TextStyle(color: Colors.white, fontSize: 25),
                  ))), // Text // TextButton // Container
              Container(
                  margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
                  height: 50,
                  width: 200,
                  decoration: BoxDecoration(
                      color: Colors.red, borderRadius: BorderRadius.circular(15)), // BoxDec
                  child: TextButton(
                      onPressed: () {
                        _openConfirmationDialog();
                      },
                      child: Text(
                        "Hapus",
                        style: TextStyle(color: Colors.white, fontSize: 25),
                      ))), // Text // TextButton // Container
            ], // <Widget>[]
          ))); // Column // SingleChildScrollView // Scaffold
    }
  }
```

10. Jalankan aplikasi dan coba hapus user





https://praktikum-mobileapp-878bb-default-rtdb.asia-southeast1.firebasedatabase.app

Your security rules are defined as public, so anyone can steal, modify, or delete data in your database

```
https://praktikum-mobileapp-878bb-default-rtdb.asia-southeast1.firebasedatabase.app/: null
```

Database location: Singapore (asia-southeast1)

11. Penghapusan sudah berhasil dan secara otomatis akun terhapus

Bab 9

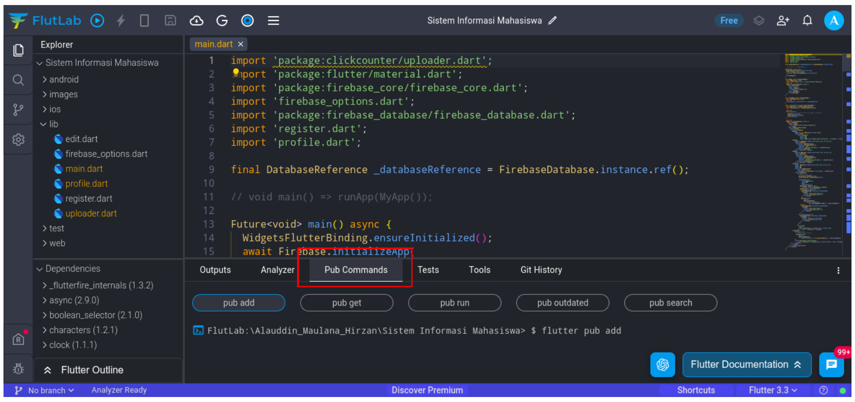
Praktikum 9

9.1 Operasi File dengan Google Storage

Di bagian ini mahasiswa diajarkan bagaimana melakukan pemrograman untuk mengunggah data (seperti gambar) ke Google Storage. Mahasiswa diwajibkan untuk menyelesaikan Praktikum 8.

9.2 Tutorial

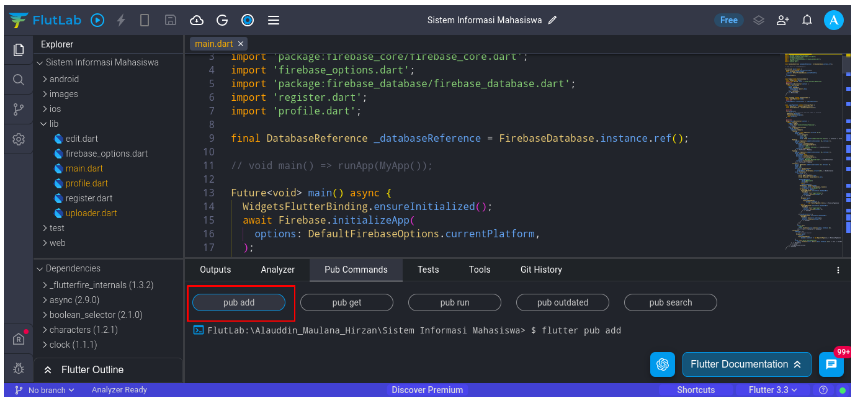
1. Sebelum memulai melakukan pemrograman. Buka Projek **SIMA Mahasiswa** di **Flutlab** . Tambahkan **Akun** yang sudah dihapus dari **Praktikum 8**
2. Di bagian bawah **Editor**, terdapat **Tab Pub Commands**. Klik **Tab** tersebut



```
import 'package:clickcounter/uploader.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'package:firebase_database/firebase_database.dart';
import 'register.dart';
import 'profile.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

// void main() => runApp(MyApp());

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
```

```
FlutLab:\Alauddin_Maulana_Hirzan\Sistem Informasi Mahasiswa> $ flutter pub add
```

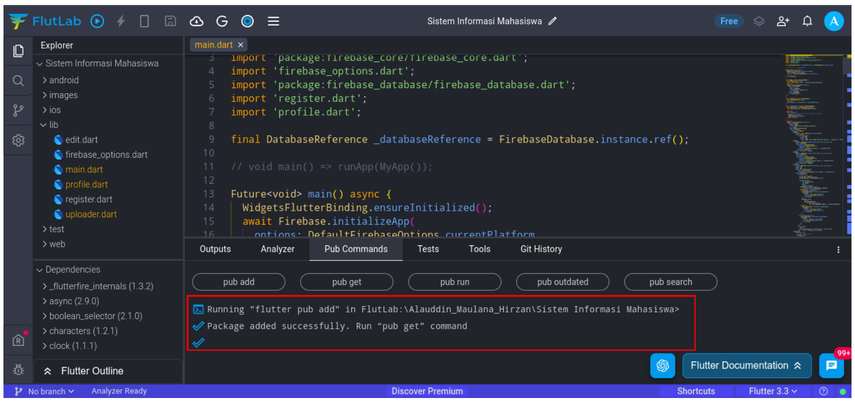
3. Setelah itu klik tombol **pub add** untuk menambahkan **depedency** baru



```
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'package:firebase_database/firebase_database.dart';
import 'register.dart';
import 'profile.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

// void main() => runApp(MyApp());

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
```

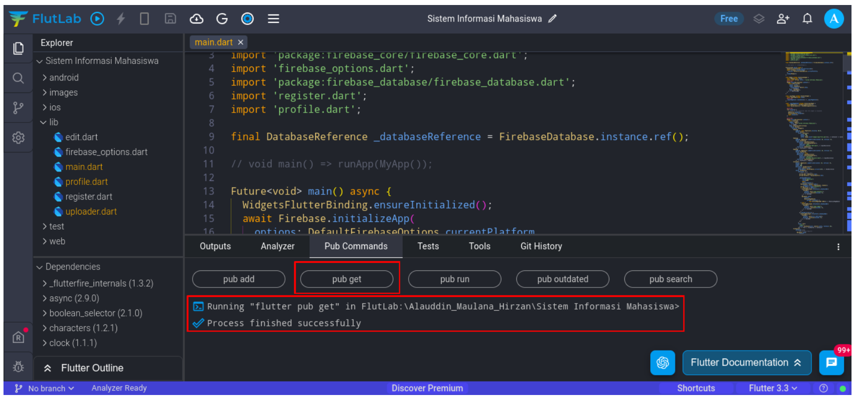
```
FlutLab:\Alauddin_Maulana_Hirzan\Sistem Informasi Mahasiswa> $ flutter pub add
```

4. Di bawah nya ada tulisan **flutter pub add**. Tambahkan kata **file_picker** setelah perintah itu. Sehingga **perintah** menjadi **flutter pub add file_picker**. Lalu **Tekan Enter**



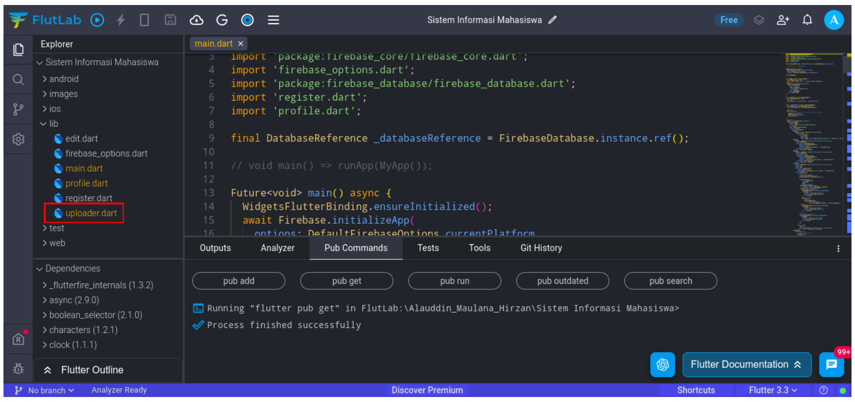
```
Running "flutter pub add" in FlutLab:\Alauddin_Maulana_Hirzan\Sistem Informasi Mahasiswa>
Package added successfully. Run "pub get" command
```

5. Setelah itu, tekan tombol **pub get** untuk mengunduh semua kebutuhan **dependency**



```
Running "flutter pub get" in FlutLab:\Alauddin_Maulana_Hirzan\Sistem Informasi Mahasiswa>
Process finished successfully
```

6. **Dependency** yang dibutuhkan sudah siap. Berikutnya membuat satu file dengan nama **uploader.dart**



```
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'package:firebase_database/firebase_database.dart';
import 'register.dart';
import 'profile.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

// void main() => runApp(MyApp());

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform
```

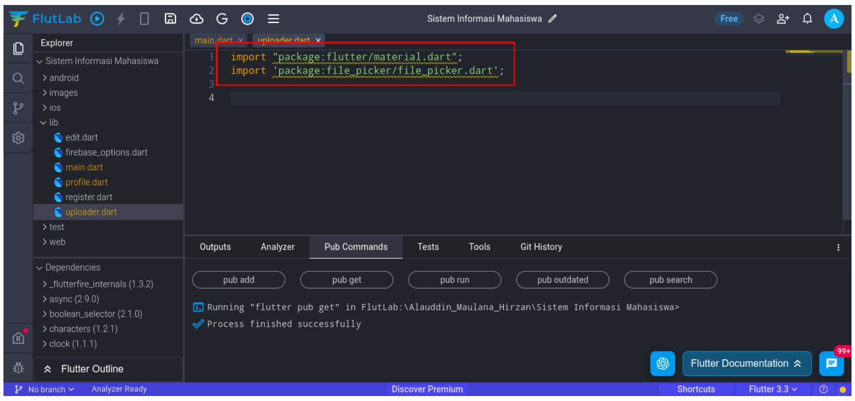
```
Running "flutter pub get" in FlutLab:\Alauddin_Maulana_Hirzan\Sistem Informasi Mahasiswa>
Process finished successfully
```

7. Jika sudah, tambahkan kode di dalam file **uploader.dart** dengan kode berikut

- Bagian **Import**

Potongan Kode

```
import "package:flutter/material.dart";
import 'package:file_picker/file_picker.dart';
```



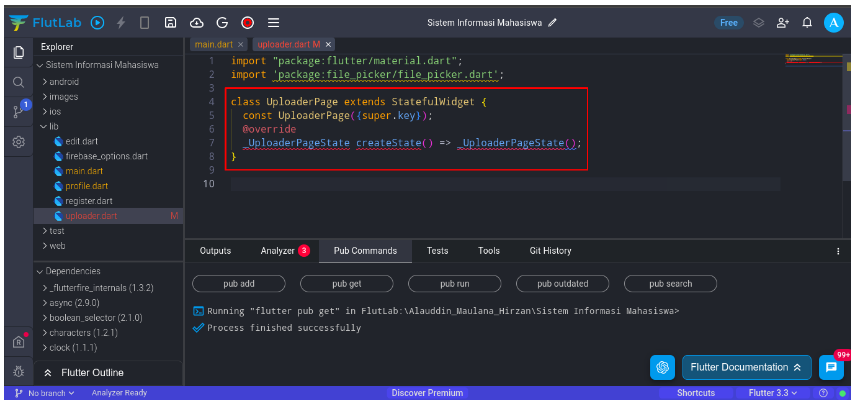
```
import "package:flutter/material.dart";
import 'package:file_picker/file_picker.dart';
```

```
Running "flutter pub get" in FlutLab:\Alauddin_Maulana_Hirzan\Sistem Informasi Mahasiswa>
Process finished successfully
```

- Bagian **Stateless Widget**

Potongan Kode

```
class UploaderPage extends StatefulWidget {
    const UploaderPage({super.key});
    @override
    _UploaderPageState createState() => _UploaderPageState();
}
```



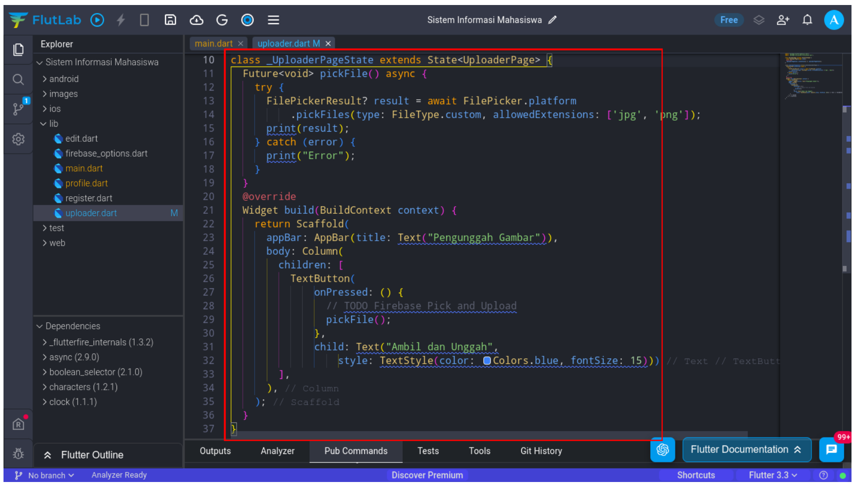
```
import "package:flutter/material.dart";
import 'package:file_picker/file_picker.dart';

class UploaderPage extends StatefulWidget {
  const UploaderPage({super.key});
  @override
  _UploaderPageState createState() => _UploaderPageState();
}
```

```
Running "flutter pub get" in FlutLab:\Alauddin_Maulana_Hirzan\Sistem Informasi Mahasiswa>
Process finished successfully
```

- Bagian **Stateful Widget + Fungsi Pick**

Potongan Kode

```
class _UploaderPageState extends State<UploaderPage> {
    Future<void> pickFile() async {
        try {
            FilePickerResult? result = await FilePicker.platform
                .pickFiles(type: FileType.custom,
                        allowedExtensions: ['jpg', 'png']);
            print(result);
        } catch (error) {
            print(error);
        }
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
        appBar: AppBar(title: Text("Pengunggah Gambar")),
        body: Column(
            children: [
                TextButton(
                    onPressed: () {
                        // TODO Firebase Pick and Upload
                        pickFile();
                    },
                    child: Text("Ambil dan Unggah",
                        style: TextStyle(color: Colors.blue,
                        fontSize: 15)))
            ],
        ),
        );
    }
}
```



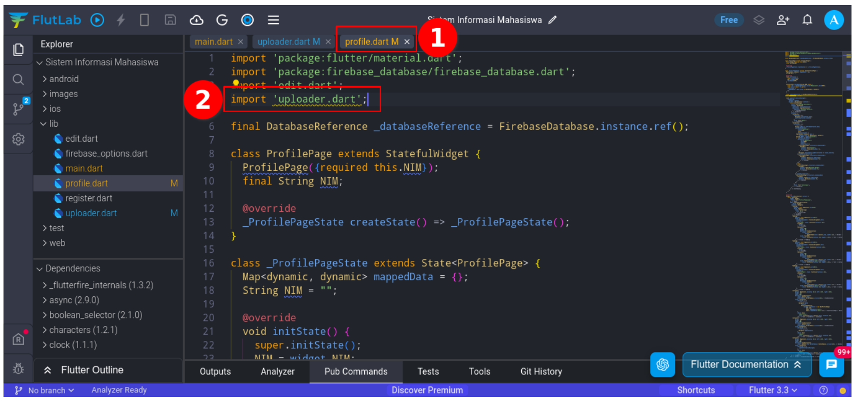
```
class _UploaderPageState extends State<UploaderPage> {
  Future<void> pickFile() async {
    try {
      FilePickerResult? result = await FilePicker.platform
          .pickFiles(type: FileType.custom, allowedExtensions: ['jpg', 'png']);
      print(result);
    } catch (error) {
      print("Error");
    }
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Pengunggah Gambar")),
      body: Column(
        children: [
          TextButton(
              onPressed: () {
                // TODO Firebase Pick and Upload
                pickFile();
              },
              child: Text("Ambil dan Unggah",
                  style: TextStyle(color: Colors.blue, fontSize: 15))) // Text // TextButt
        ],
      ), // Column
    ); // Scaffold
  }
}
```

8. Kode dasar sudah disiapkan. Langkah berikutnya adalah menyambungkan halaman

Profile ke **Upload** ini. Buka file **profile.dart** dan tambahkan kode import.

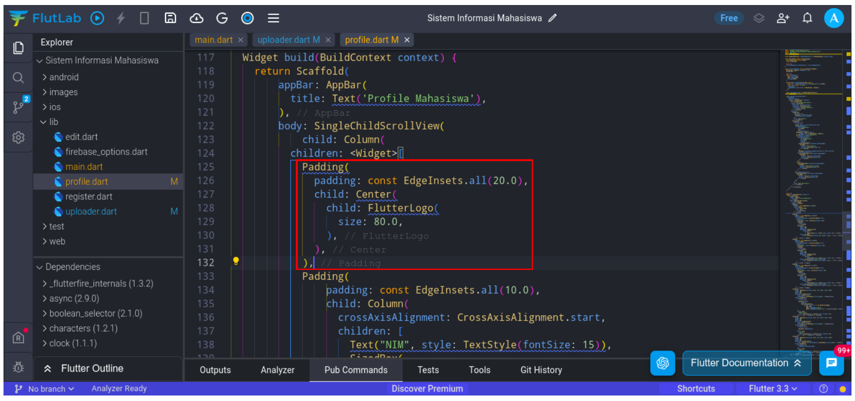
Potongan Kode

```
import 'uploader.dart';
```



```
import 'package:flutter/material.dart';
import 'package:firebase_database/firebase_database.dart';
import 'edit.dart';
import 'uploader.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();

class ProfilePage extends StatefulWidget {
  ProfilePage({required this.NIM});
  final String NIM;

  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {
  Map<dynamic, dynamic> mappedData = {};
  String NIM = "";

  @override
  void initState() {
    super.initState();
```

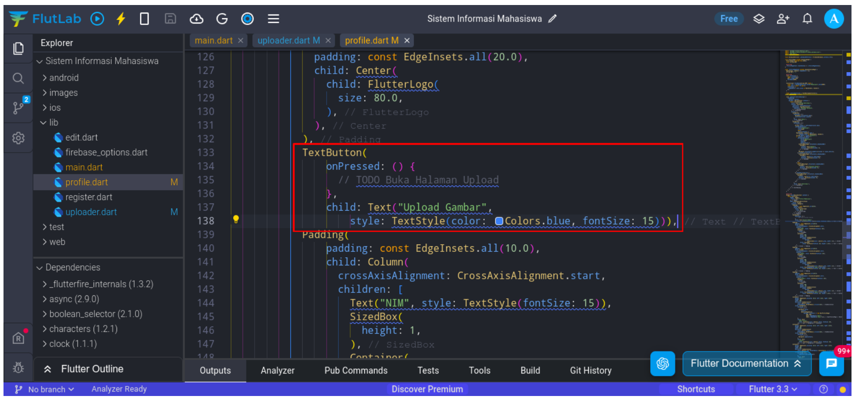
9. Berikutnya cari **Padding** untuk **Flutter Logo** (Lihat Gambar), dan tambahkan kode berikut:



```
  Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
          title: Text('Profile Mahasiswa'),
        ), // AppBar
        body: SingleChildScrollView(
          child: Column(
         children: <Widget>[
          Padding(
            padding: const EdgeInsets.all(20.0),
            child: Center(
              child: FlutterLogo(
                size: 80.0,
              ), // FlutterLogo
            ), // Center
          ), // Padding
          Padding(
              padding: const EdgeInsets.all(10.0),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text("NIM", style: TextStyle(fontSize: 15)),
```

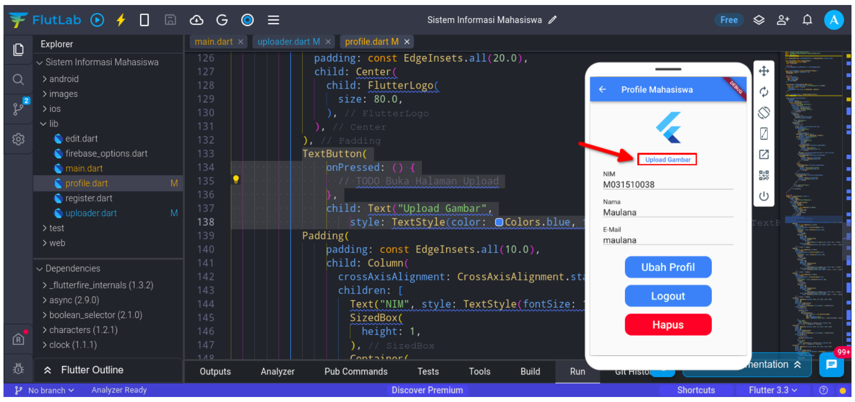
Potongan Kode

```
TextButton(
    onPressed: () {
        // TODO Buka Halaman Upload
    },
    child: Text("Upload Gambar",
        style: TextStyle(color: Colors.blue, fontSize: 15))),
```



```
            padding: const EdgeInsets.all(20.0),
            child: Center(
              child: FlutterLogo(
                size: 80.0,
              ), // FlutterLogo
            ), // Center
          ), // Padding
          TextButton(
              onPressed: () {
                // TODO Buka Halaman Upload
              },
              child: Text("Upload Gambar",
                  style: TextStyle(color: Colors.blue, fontSize: 15))), // Text // TextB
          Padding(
              padding: const EdgeInsets.all(10.0),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  Text("NIM", style: TextStyle(fontSize: 15)),
                  SizedBox(
                    height: 1,
                  ), // SizedBox
```

10. Jalankan aplikasi untuk mengetes hasil saat ini.

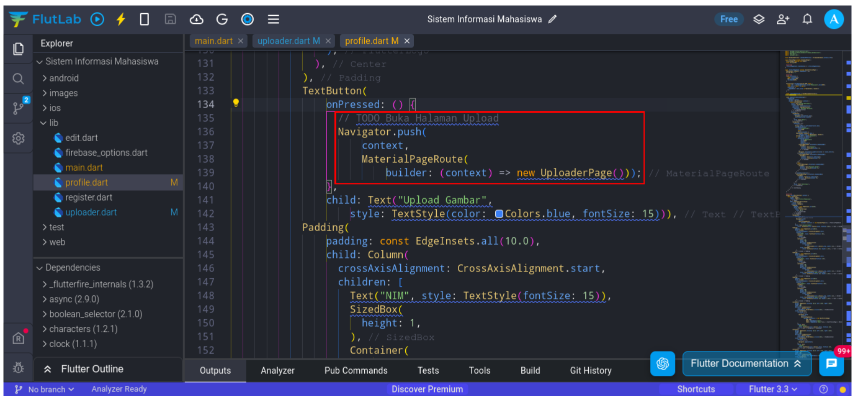


```
                padding: const EdgeInsets.all(20.0),
                child: Center(
                  child: FlutterLogo(
                    size: 80.0,
                  ), // FlutterLogo
                ), // Center
              ), // Padding
              TextButton(
                  onPressed: () {
                    // TODO Buka Halaman Upload
                  },
                  child: Text("Upload Gambar",
                      style: TextStyle(color: Colors.blue,
              Padding(
                  padding: const EdgeInsets.all(10.0),
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.sta
                    children: [
                      Text("NIM", style: TextStyle(fontSize:
                      SizedBox(
                        height: 1,
                      ), // SizedBox
                      Container(
```

11. Kode saat ini belum tersambung ke halaman **uploader.dart**. Untuk menyambungkan kode ini, tambahkan kode berikut tepat di bawah // **TODO Buka Halaman Upload** dari kode sebelumnya

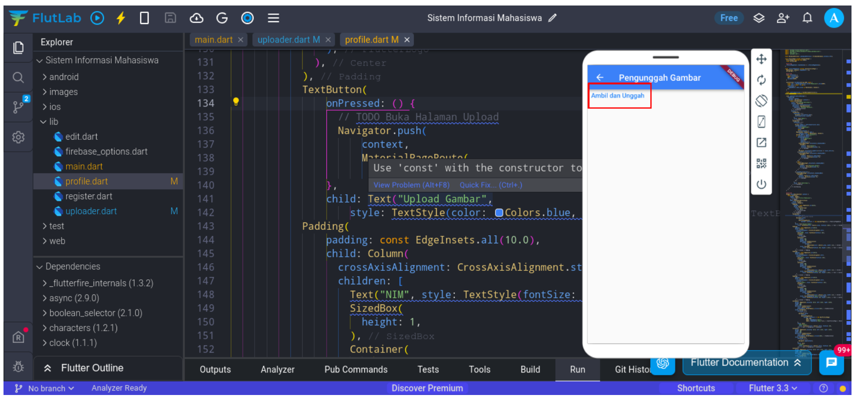
Potongan Kode

```
// TODO Buka Halaman Upload
Navigator.push(
    context,
    MaterialPageRoute(
        builder: (context) => new UploaderPage()));
```



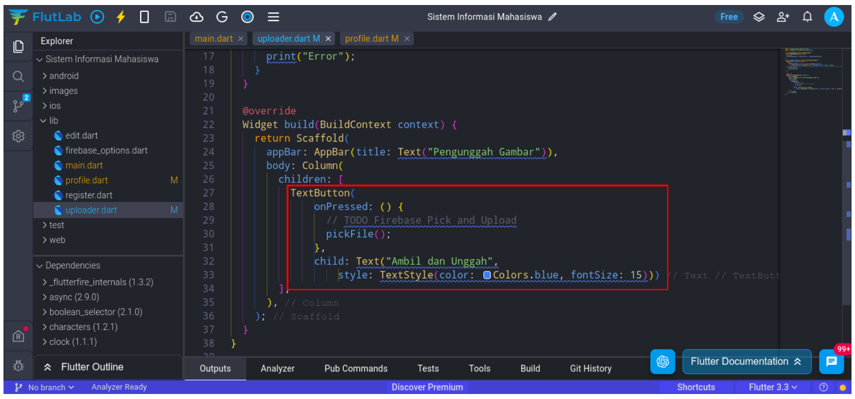
```
                ), // Center
              ), // Padding
              TextButton(
                  onPressed: () {
                    // TODO Buka Halaman Upload
                    Navigator.push(
                        context,
                        MaterialPageRoute(
                            builder: (context) => new UploaderPage())); // MaterialPageRoute
                  },
                  child: Text("Upload Gambar",
                      style: TextStyle(color: Colors.blue, fontSize: 15))), // Text // TextB
              Padding(
                  padding: const EdgeInsets.all(10.0),
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Text("NIM", style: TextStyle(fontSize: 15)),
                      SizedBox(
                        height: 1,
                      ), // SizedBox
                      Container(
```

12. Uji aplikasi lagi untuk memastikan tombol bisa berpindah halaman



```
                ), // Center
              ), // Padding
              TextButton(
                  onPressed: () {
                    // TODO Buka Halaman Upload
                    Navigator.push(
                        context,
                        MaterialPageRoute(
                  Use 'const' with the constructor to
                  View Problem (Alt+F8)   Quick Fix... (Ctrl+.)
                  },
                  child: Text("Upload Gambar",
                      style: TextStyle(color: Colors.blue,
              Padding(
                  padding: const EdgeInsets.all(10.0),
                  child: Column(
                    crossAxisAlignment: CrossAxisAlignment.st
                    children: [
                      Text("NIM", style: TextStyle(fontSize:
                      SizedBox(
                        height: 1,
                      ), // SizedBox
                      Container(
```

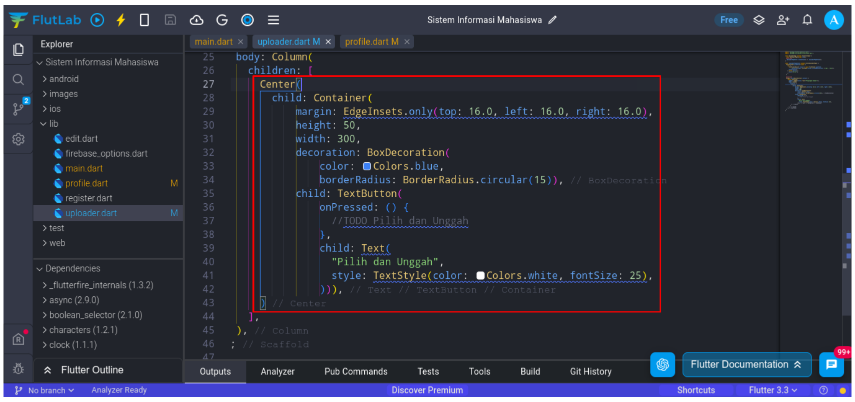
13. Jika halaman berikutnya berhasil muncul, berikutnya adalah memperbaiki tombol tersebut. Buka file **uploader.dart**, ganti baris kode yang ada di gambar dengan kode berikut:



```
      print("Error");
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Pengunggah Gambar")),
      body: Column(
        children: [
          TextButton(
            onPressed: () {
              // TODO Firebase Pick and Upload
              pickFile();
            },
            child: Text("Ambil dan Unggah",
                style: TextStyle(color: Colors.blue, fontSize: 15))) // Text // TextButt
        ],
      ), // Column
    ); // Scaffold
  }
}
```

Potongan Kode

```
Center(
    child: Container(
        margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
        height: 50,
        width: 300,
        decoration: BoxDecoration(
            color: Colors.blue,
            borderRadius: BorderRadius.circular(15)),
            child: TextButton(
                onPressed: () {
                    //TODO Pilih dan Unggah
                },
                child: Text(
                    "Pilih dan Unggah",
                    style: TextStyle(color: Colors.white, fontSize: 25),
                ))),
)
```



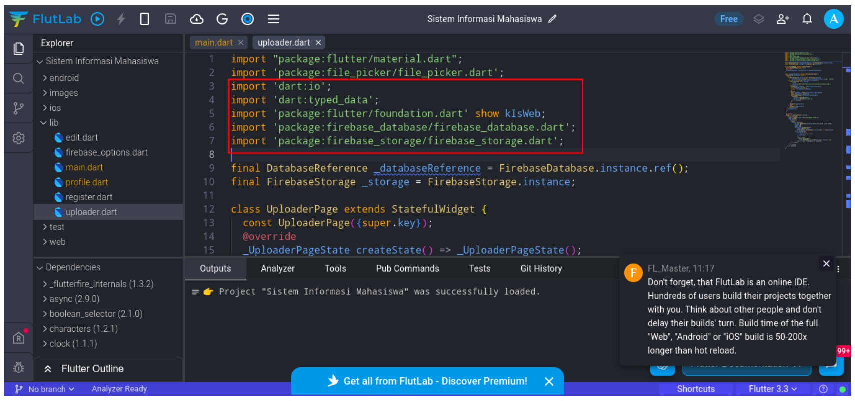
```
      body: Column(
        children: [
          Center(
            child: Container(
              margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
              height: 50,
              width: 300,
              decoration: BoxDecoration(
                color: Colors.blue,
                borderRadius: BorderRadius.circular(15)), // BoxDecoration
              child: TextButton(
                onPressed: () {
                  //TODO Pilih dan Unggah
                },
                child: Text(
                  "Pilih dan Unggah",
                  style: TextStyle(color: Colors.white, fontSize: 25),
                ))), // Text // TextButton // Container
          ) // Center
        ],
      ), // Column
    ; // Scaffold
```

14. Berikutnya menambahkan akses ke Database dengan kodeebrikut

- Kode **Import**

Potongan Kode

```
import 'dart:io';
import 'dart:typed_data';
import 'package:flutter/foundation.dart' show kIsWeb;
import 'package:firebase_database/firebase_database.dart';
import 'package:firebase_storage/firebase_storage.dart';
```



```
import "package:flutter/material.dart";
import 'package:file_picker/file_picker.dart';
import 'dart:io';
import 'dart:typed_data';
import 'package:flutter/foundation.dart' show kIsWeb;
import 'package:firebase_database/firebase_database.dart';
import 'package:firebase_storage/firebase_storage.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();
final FirebaseStorage _storage = FirebaseStorage.instance;

class UploaderPage extends StatefulWidget {
  const UploaderPage({super.key});
  @override
  _UploaderPageState createState() => _UploaderPageState();
```

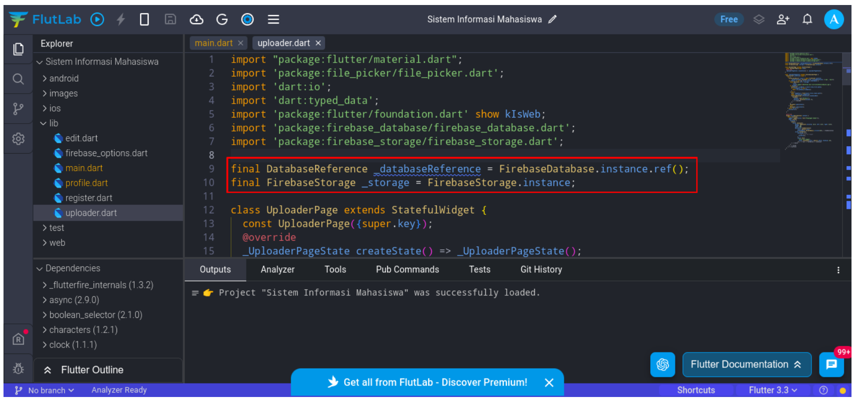
```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

FL_Master, 11:17
Don't forget, that FlutLab is an online IDE. Hundreds of users build their projects together with you. Think about other people and don't delay their builds' turn. Build time of the full "Web", "Android" or "iOS" build is 50-200x longer than hot reload.

- Tambahkan kode akses database tepat di bawah **import**

Potongan Kode

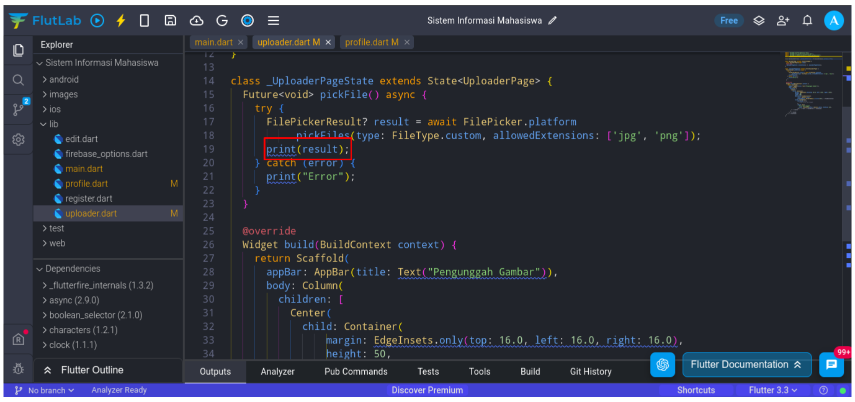
```
final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();
final FirebaseStorage _storage = FirebaseStorage.instance;
```



```
import "package:flutter/material.dart";
import 'package:file_picker/file_picker.dart';
import 'dart:io';
import 'dart:typed_data';
import 'package:flutter/foundation.dart' show kIsWeb;
import 'package:firebase_database/firebase_database.dart';
import 'package:firebase_storage/firebase_storage.dart';

final DatabaseReference _databaseReference = FirebaseDatabase.instance.ref();
final FirebaseStorage _storage = FirebaseStorage.instance;

class UploaderPage extends StatefulWidget {
  const UploaderPage({super.key});
  @override
  _UploaderPageState createState() => _UploaderPageState();
```

```
Project "Sistem Informasi Mahasiswa" was successfully loaded.
```

15. Jika kode atas sudah selesai, berikutnya adalah mengubah fungsi **pickFile()** yang ada di dalam kode **Stateful Widget**. hapus kode **print(result);**

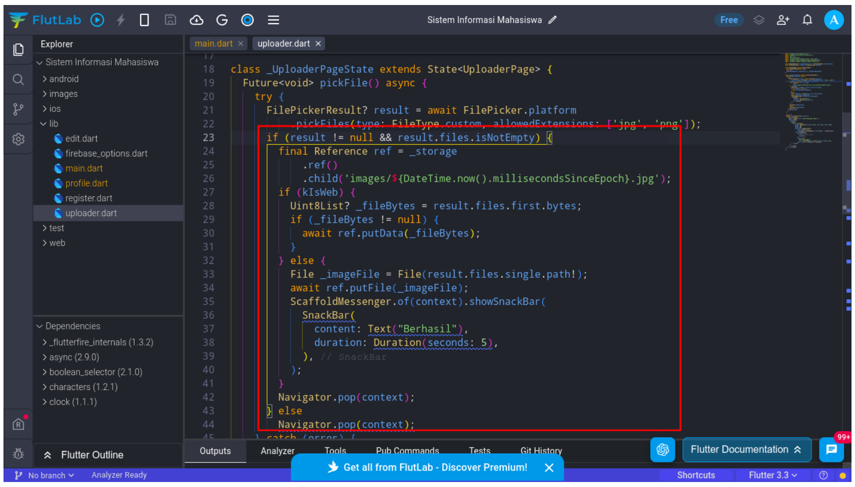


```
  }

class _UploaderPageState extends State<UploaderPage> {
  Future<void> pickFile() async {
    try {
      FilePickerResult? result = await FilePicker.platform
          .pickFiles(type: FileType.custom, allowedExtensions: ['jpg', 'png']);
      print(result);
    } catch (error) {
      print("Error");
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Pengunggah Gambar")),
      body: Column(
        children: [
          Center(
            child: Container(
              margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
              height: 50,
```

16. Ganti dengan kode berikut

Potongan Kode

```
if (result != null && result.files.isNotEmpty) {
    final Reference ref = _storage
    .ref()
    .child('images/$DateTime.now().millisecondsSinceEpoch.jpg');
    if (kIsWeb) {
        Uint8List? _fileBytes = result.files.first.bytes;
        if (_fileBytes != null) {
            await ref.putData(_fileBytes);
        }
    } else {
        File _imageFile = File(result.files.single.path!);
        await ref.putFile(_imageFile);
        ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text("Berhasil"),
            duration: Duration(seconds: 5),
        ),
        );
    }
    Navigator.pop(context);
} else
Navigator.pop(context);
```

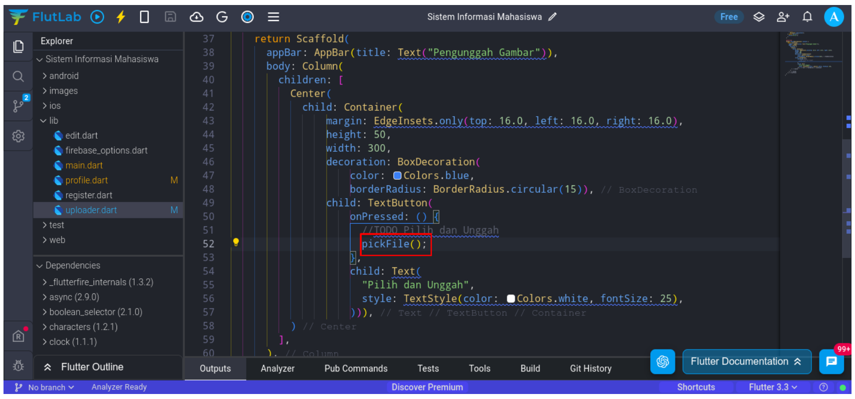


```
class _UploaderPageState extends State<UploaderPage> {
  Future<void> pickFile() async {
    try {
      FilePickerResult? result = await FilePicker.platform
          .pickFiles(type: FileType.custom, allowedExtensions: ['jpg', 'png']);
      if (result != null && result.files.isNotEmpty) {
        final Reference ref = _storage
            .ref()
            .child('images/${DateTime.now().millisecondsSinceEpoch}.jpg');
        if (kIsWeb) {
          Uint8List? _fileBytes = result.files.first.bytes;
          if (_fileBytes != null) {
            await ref.putData(_fileBytes);
          }
        } else {
          File _imageFile = File(result.files.single.path!);
          await ref.putFile(_imageFile);
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text("Berhasil"),
              duration: Duration(seconds: 5),
            ), // SnackBar
          );
        }
        Navigator.pop(context);
      } else
        Navigator.pop(context);
```

17. Bagian terakhir adalah menambahkan fungsi ke **Tombol**. Tambahkan kode berikut tepat di bawah kode //**TODO Pilih dan Unggah**

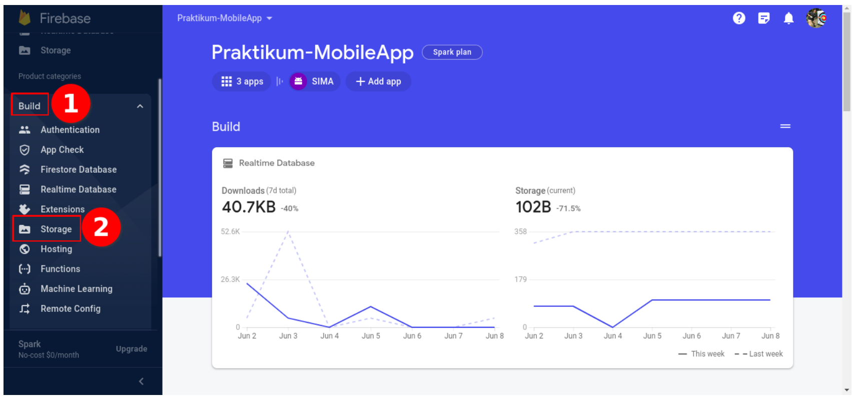
Potongan Kode

```
//TODO Pilih dan Unggah
pickFile();
```

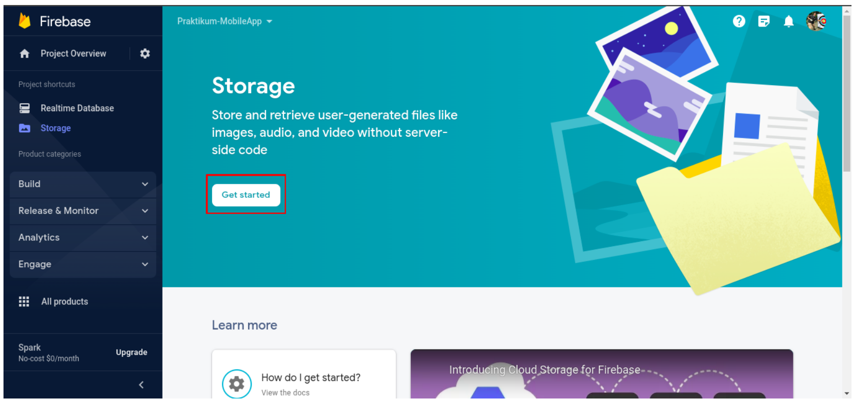


```
return Scaffold(
  appBar: AppBar(title: Text("Pengunggah Gambar")),
  body: Column(
    children: [
      Center(
        child: Container(
            margin: EdgeInsets.only(top: 16.0, left: 16.0, right: 16.0),
            height: 50,
            width: 300,
            decoration: BoxDecoration(
                color: Colors.blue,
                borderRadius: BorderRadius.circular(15)), // BoxDecoration
            child: TextButton(
                onPressed: () {
                  //TODO Pilih dan Unggah
                  pickFile();
                },
                child: Text(
                  "Pilih dan Unggah",
                  style: TextStyle(color: Colors.white, fontSize: 25),
                ))), // Text // TextButton // Container
      ) // Center
    ],
  ), // Column
```

18. Jika sudah, konfigurasikan **Google Storage** dengan membuka https://console.firebase.google.com/. Buka **Projeknya**, lalu cari **Google Storage**



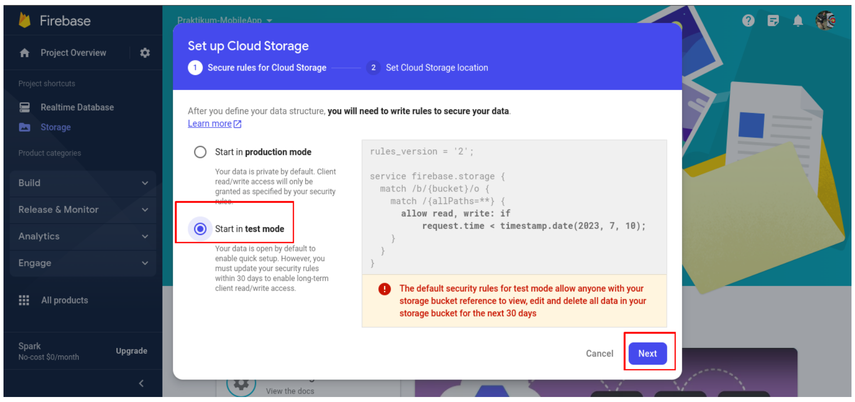
19. Klik **Get Started**



Storage

Store and retrieve user-generated files like images, audio, and video without server-side code

20. Pilih **Test Mode**, lalu klik **Next**



Set up Cloud Storage

1 Secure rules for Cloud Storage — 2 Set Cloud Storage location

After you define your data structure, **you will need to write rules to secure your data**.
Learn more

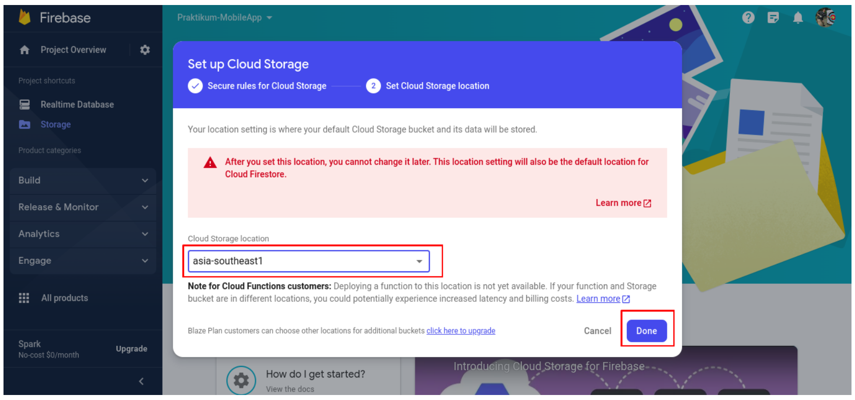
Start in **production mode**
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

[✓ Start in **test mode**]
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';

service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if
          request.time < timestamp.date(2023, 7, 10);
    }
  }
}
```

The default security rules for test mode allow anyone with your storage bucket reference to view, edit and delete all data in your storage bucket for the next 30 days

21. Pilih lokasi, dan klik **Done**



Set up Cloud Storage

✓ Secure rules for Cloud Storage — 2 Set Cloud Storage location

Your location setting is where your default Cloud Storage bucket and its data will be stored.

After you set this location, you cannot change it later. This location setting will also be the default location for Cloud Firestore.

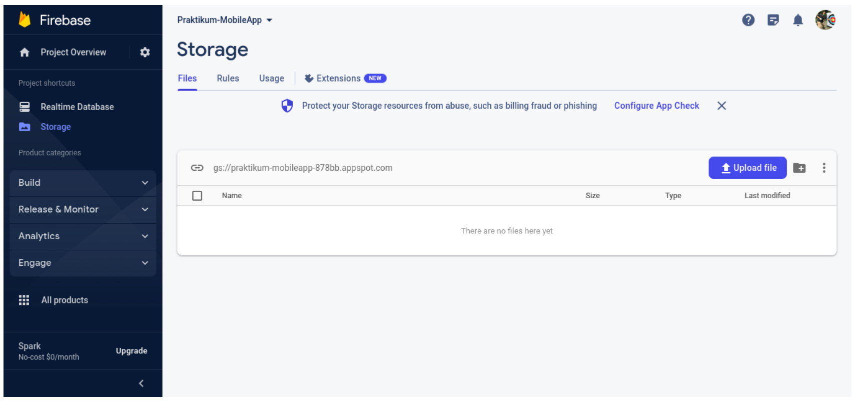
Learn more

Cloud Storage location: asia-southeast1

Note for Cloud Functions customers: Deploying a function to this location is not yet available. If your function and Storage bucket are in different locations, you could potentially experience increased latency and billing costs. Learn more

Blaze Plan customers can choose other locations for additional buckets click here to upgrade

22. **Google Storage** sudah siap digunakan



Storage

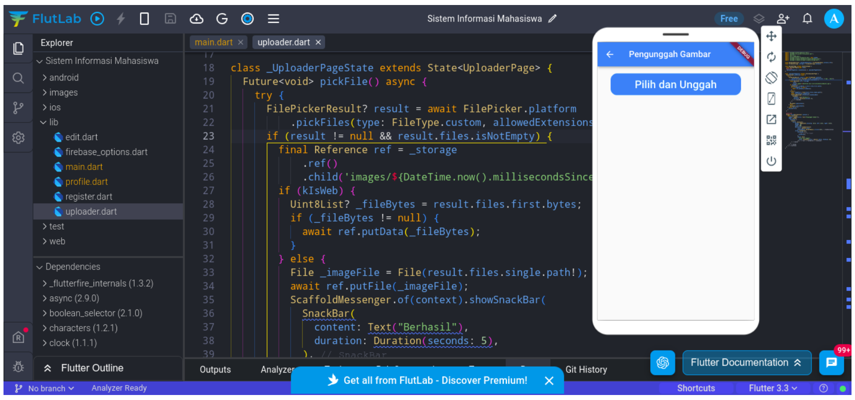
Protect your Storage resources from abuse, such as billing fraud or phishing

gs://praktikum-mobileapp-878bb.appspot.com

Name	Size	Type	Last modified

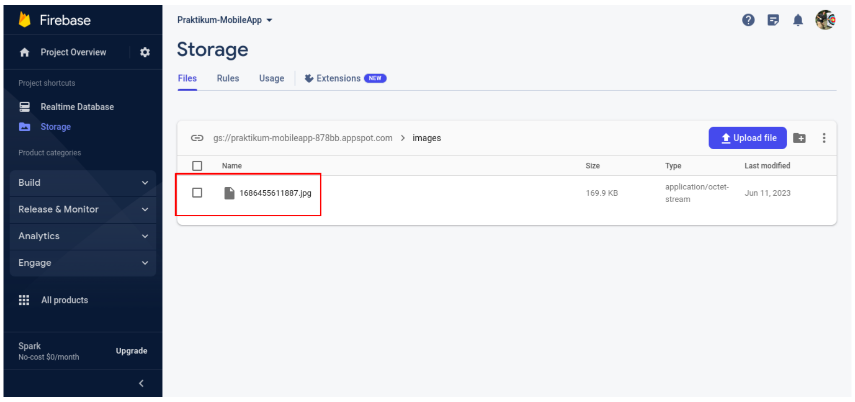
There are no files here yet

23. Tes aplikasi dengan mengupload gambar, dan Tunggu (Membutuhkan waktu agak lama hingga proses upload selesai)



```
class _UploaderPageState extends State<UploaderPage> {
  Future<void> pickFile() async {
    try {
      FilePickerResult? result = await FilePicker.platform
          .pickFiles(type: FileType.custom, allowedExtensions
      if (result != null && result.files.isNotEmpty) {
        final Reference ref = _storage
            .ref()
            .child('images/${DateTime.now().millisecondsSince
        if (kIsWeb) {
          Uint8List? _fileBytes = result.files.first.bytes;
          if (_fileBytes != null) {
            await ref.putData(_fileBytes);
          }
        } else {
          File _imageFile = File(result.files.single.path!);
          await ref.putFile(_imageFile);
          ScaffoldMessenger.of(context).showSnackBar(
            SnackBar(
              content: Text("Berhasil"),
              duration: Duration(seconds: 5),
            ), // SnackBar
```

24. Cek **Google Storage** untuk melihat hasil unggahan. Gambar disimpan dalam format waktu yang ditentukan otomatis oleh Flutter



gs://praktikum-mobileapp-878bb.appspot.com > images

Name	Size	Type	Last modified
1686455611887.jpg	169.9 KB	application/octet-stream	Jun 11, 2023