

MODUL 3

CODELAB

PEMROGRAMAN MOBILE D

Dosen Pengampu:

Bashor Fauzan Muthohirin, S.Kom., M.Kom

Source Code: https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/tree/main/TUGAS/modul_3/CODELAB

Muhammad Hisyam Kamil

202210370311060

Daftar Isi

MODUL 3: CODELAB	3
I. Pendahuluan.....	3
II. Instalasi, Konfigurasi Firebase, dan Implementasi Firebase Auth	4
III. Melengkapi Kode (ToDo List dengan Edit dan Hapus task)	14

MODUL 3: CODELAB

Source Code: https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/tree/main/TUGAS/modul_3/CODELAB

CODELAB

TUGAS 1

1. Lakukan setup dan instalasi firebase, pastikan muncul keterangan bahwa berhasil menginisiasi Firebase di Terminal
2. Buatlah halaman registrasi sederhana beserta controller untuk menangani input email dan password dari user menggunakan Firebase Authentication.
3. Jika user berhasil melakukan registrasi, email yang terdaftar akan tampil di halaman Authentication pada Firebase.

TUGAS 2

1. Lengkapi code `// TODO:` berikut pada file `home_screen.dart` (nama file sesuaikan dengan yang kalian buat)

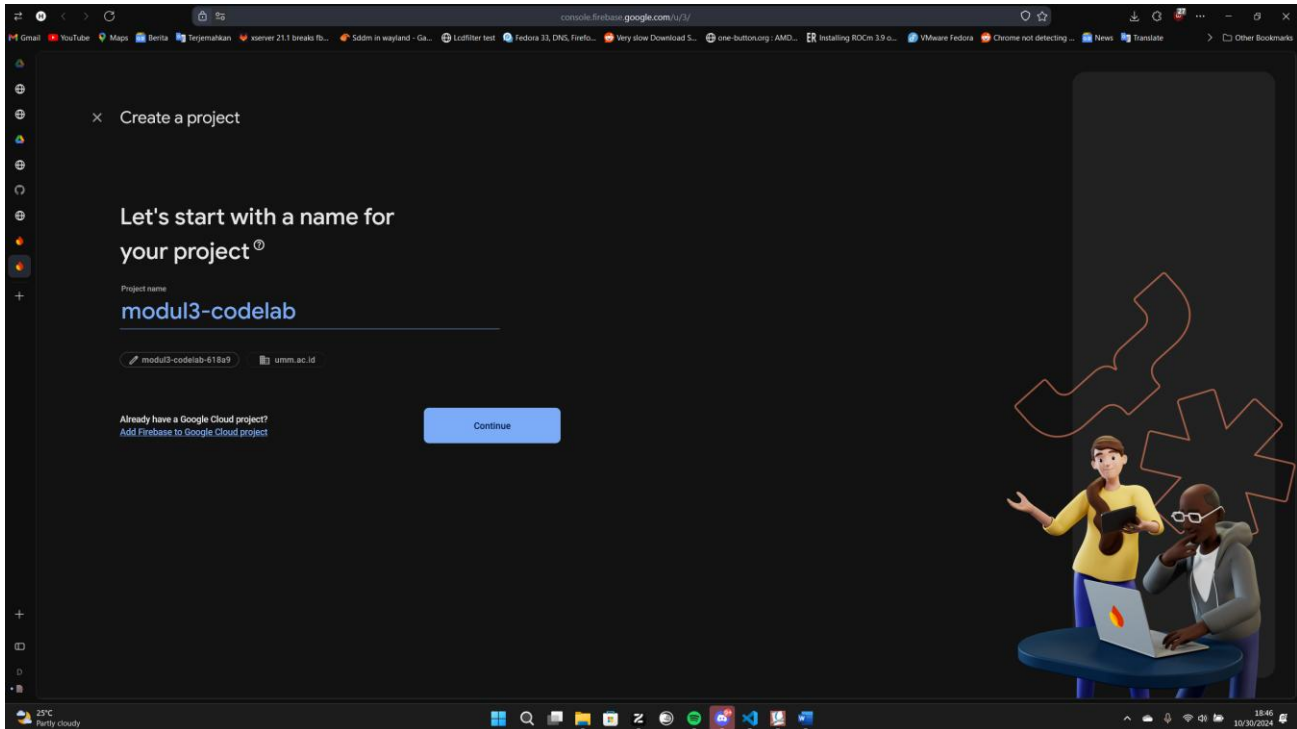
```
onSelected: (String value) async {  
  if (value == 'edit') {  
    // TODO: fitur edit task  
  } else if (value == 'delete') {  
    // TODO: fitur hapus task  
  }  
},
```

I. Pendahuluan

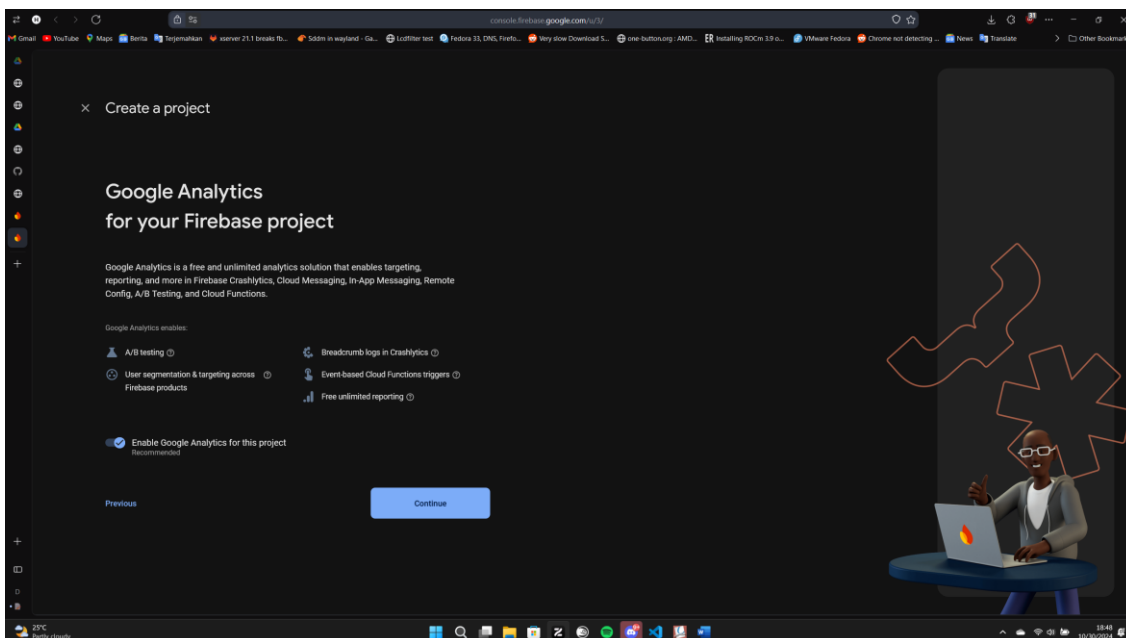
Pada modul 3 codelab ini, kita diminta untuk melakukan instalasi firebase dan membuat halaman register sederhana, yang dimana nanti datanya akan masuk kedalam firebase. Lalu kita juga diminta untuk melengkapi kode todo yang sudah diberikan di soal ini.

II. Instalasi, Konfigurasi Firebase, dan Implementasi Firebase Auth

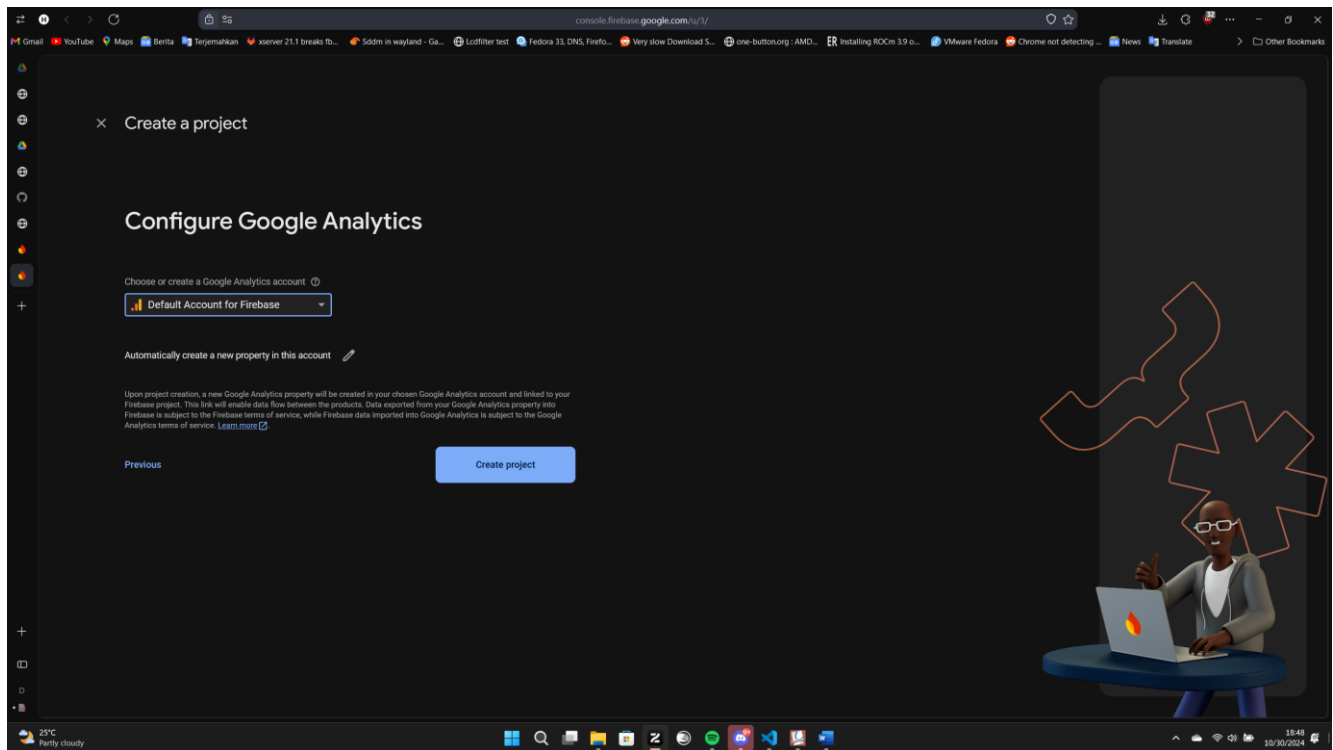
Sebelum melakukan instalasi firebase pada project kita, pertama kita membuat project baru terlebih dahulu didalam Firebase Console.



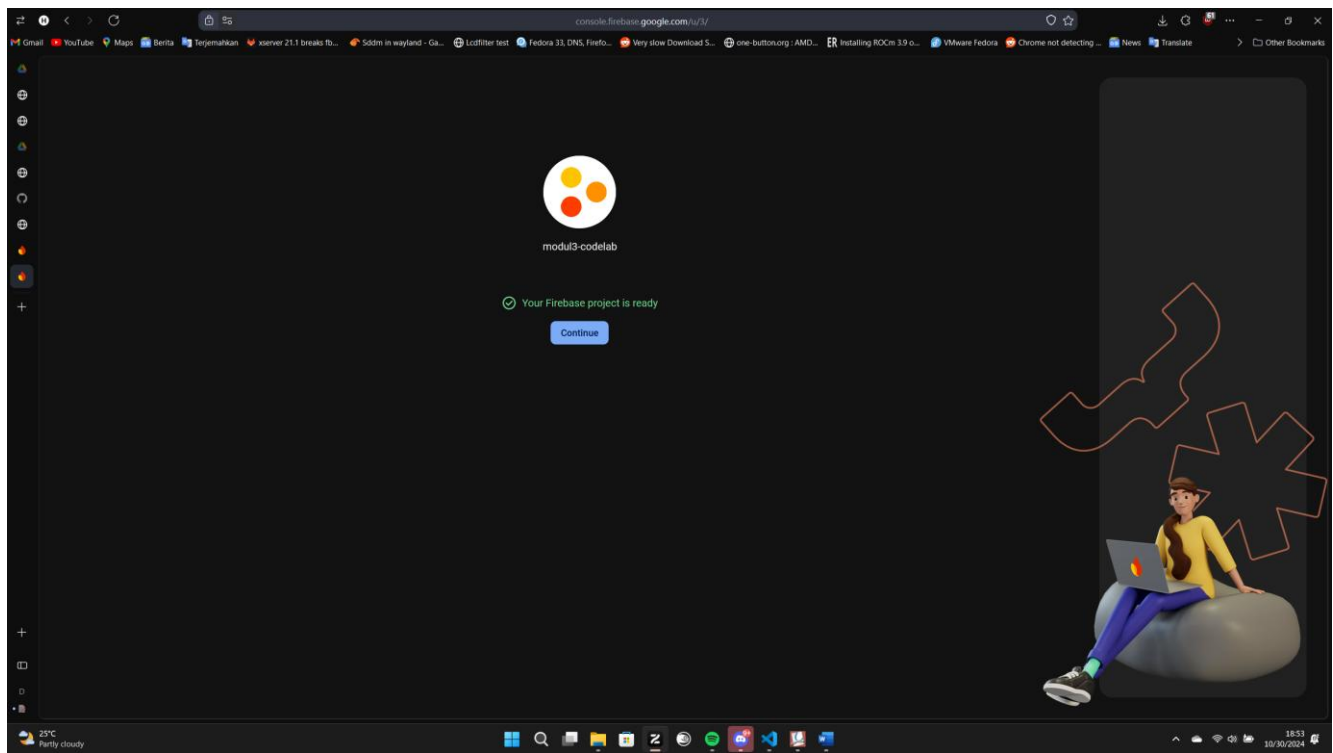
Klik Continue, lalu pada halaman selanjutnya pastikan Enable Google Analytics for this project dicentang.



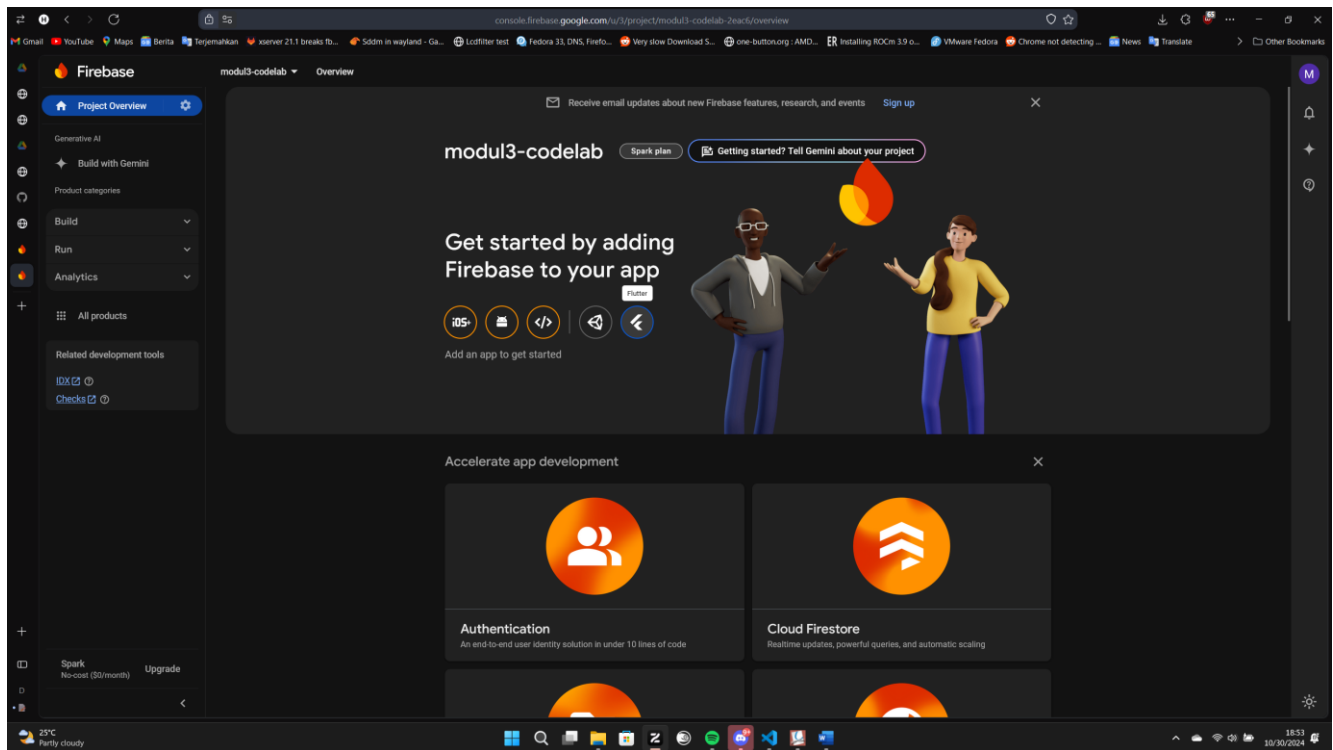
Lalu pada menu dropdown pilih Default Account for Firebase, dan tekan tombol Create Project.



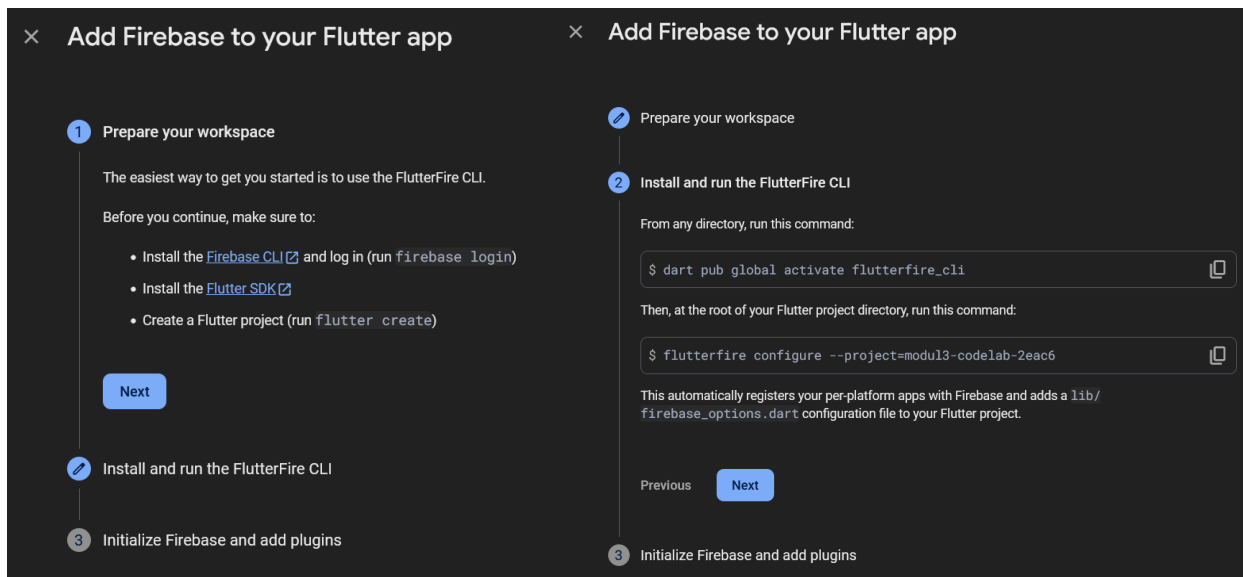
Project telah berhasil dibuat seperti pada gambar dibawah ini:



Lalu pada halaman berikutnya klik logo flutter seperti yang ada pada gambar dibawah ini



Maka akan muncul panduan untuk menambahkan firebase ke aplikasi flutter kita, ikuti arahnya sampai selesai. Seperti pada gambar dibawah ini:



Mengikuti panduan tadi, kita konfigurasi firebase pada project kita, seperti pada gambar dibawah ini:

```
PROBLEMS 59 TERMINAL OUTPUT DEBUG CONSOLE PORTS GITLENS POSTMAN
>
• > dart pub global activate flutterfire_cli
Package flutterfire_cli is currently active at version 1.0.0.
Downloading packages... (2.5s)
> cli_util 0.4.2 (was 0.4.1)
> clock 1.1.2 (was 1.1.1)
  dart_console 1.2.0 (4.1.0 available)
  file 6.1.4 (7.0.1 available)
  http 0.13.6 (1.2.2 available)
  intl 0.18.1 (0.19.0 available)
  process 4.2.4 (5.0.2 available)
  pub_updater 0.2.4 (0.5.0 available)
> win32 5.7.2 (was 5.7.1)
Building package executables... (7.5s)
Built flutterfire_cli:flutterfire.
Installed executable flutterfire.
Activated flutterfire_cli 1.0.0.

~ hisyam99 ~ main >_pwsh
~ home / Documents / TUGAS KULIAH / Semester 5 / Pemrograman Mobi

> firebase login
(node:67932) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use `node:url` instead.
(Use `node --trace-deprecation ...` to show where the warning was emitted)
Already logged in as hisyamkamil99@webmail.umm.ac.id

~ hisyam99 ~ main >_pwsh
~ home / Documents / TUGAS KULIAH / Semester 5 / Pemrograman Mobi
```

Kita bisa memilih platform apa saja yang mau diimplementasikan firebase, disini saya memilih android dan web saja

```
> flutterfire configure --project=modul3-codelab-2eac6
? Generated FirebaseOptions file C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\modul_3\codelab_2\codelab_2.dart
? Generated FirebaseOptions file C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\modul_3\codelab_2\codelab_2.dart
override it? - yes
i Found 1 Firebase projects. Selecting project modul3-codelab-2eac6.
? Which platforms should your configuration support (use arrow keys & space to select)? >
✓ android
  ios
  macos
✓ web
  windows

> flutterfire configure --project=modul3-codelab-2eac6
? Generated FirebaseOptions file C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\modul_3\codelab_2\codelab_2.dart
• ✓ Generated FirebaseOptions file C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\modul_3\codelab_2\codelab_2.dart
override it? - yes
i Found 1 Firebase projects. Selecting project modul3-codelab-2eac6.
✓ Which platforms should your configuration support (use arrow keys & space to select)? - android, web
i Firebase android app xyz.reparin.tugas_1 registered.
i Firebase web app tugas_1 (web) registered.

Firebase configuration file lib\firebase_options.dart generated successfully with the following Firebase apps:

Platform  Firebase App Id
web        1:
android    1:

Learn more about using this file and next steps from the documentation:
> https://firebase.google.com/docs/flutter/setup

~ hisyam99 ~ main >_pwsh
~ home / Documents / TUGAS KULIAH / Semester 5 / Pemrograman Mobile / PRAKTIKUM / TUGAS / modul_3 / CODELAB / codelab_2 ~
```

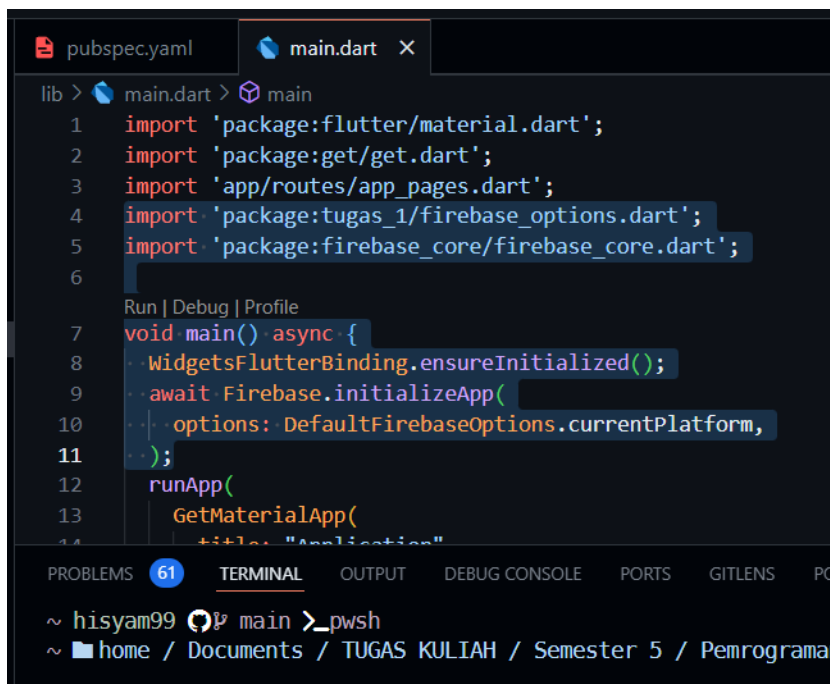
Setelah itu, masuk kedalam file pubspec.yaml dan masukkan semua plugin berikut:



```
pubspec.yaml > {} dependencies > cloud_firestore  
1  name: tugas_1  
2  version: 1.0.0+1  
3  publish_to: none  
4  description: A new Flutter project.  
5  environment:  
6    sdk: '>=3.2.6 <4.0.0'  
7  
8  dependencies:  
9    cupertino_icons: ^1.0.2  
10   get: 4.6.6  
11   firebase_auth: ^5.3.1  
12   firebase_core: ^3.6.0  
13   cloud_firestore: ^5.4.4  
14   intl: ^0.18.0  
15   webview_flutter: ^4.8.0  
16   carousel_slider: ^5.0.0  
17   flutter:  
18     sdk: flutter  
19
```

~ hisyam99 main >_pwsh
~ home / Documents / TUGAS KULIAH / Semester

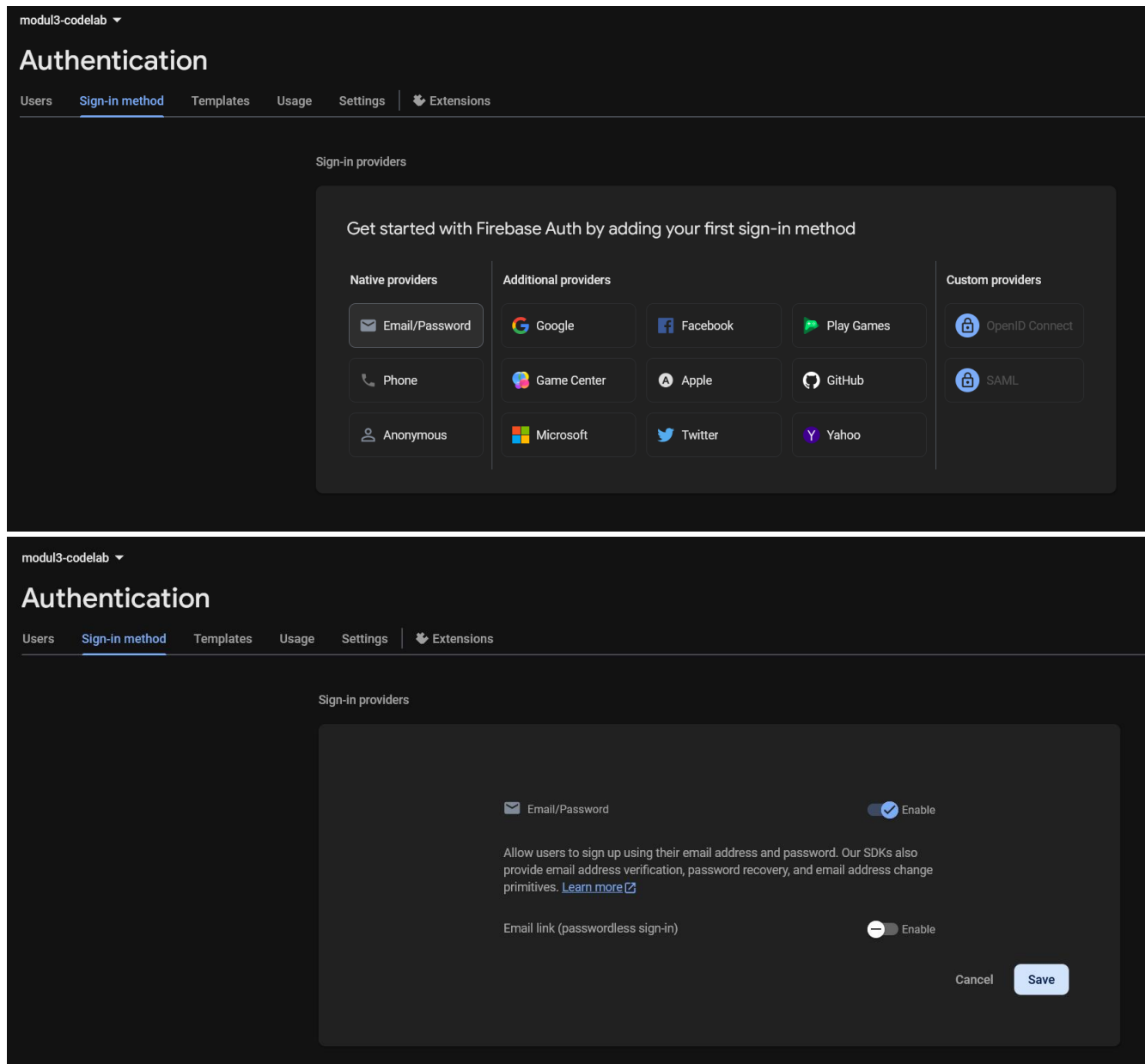
Selanjutnya, modifikasi kode main.dart kita dan inisialisai Firebase didalam main() seperti pada gambar dibawah ini:



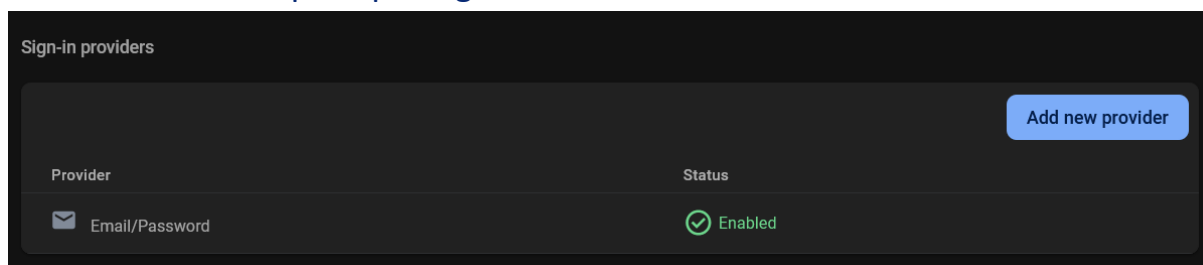
```
pubspec.yaml  main.dart  
lib > main.dart > main  
1  import 'package:flutter/material.dart';  
2  import 'package:get/get.dart';  
3  import 'app/routes/app_pages.dart';  
4  import 'package:tugas_1/firebase_options.dart';  
5  import 'package:firebase_core/firebase_core.dart';  
6  
7  void main() async {  
8    WidgetsFlutterBinding.ensureInitialized();  
9    await Firebase.initializeApp(  
10      options: DefaultFirebaseOptions.currentPlatform,  
11    );  
12    runApp(  
13      GetMaterialApp(  
14        title: "Application"  
15      )  
16    );  
17  }  
18
```

Run | Debug | Profile
~ hisyam99 main >_pwsh
~ home / Documents / TUGAS KULIAH / Semester 5 / Pemrograman

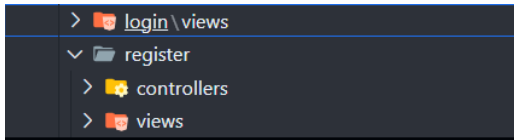
Kembali ke project kita di Firebase Console, pilih menu Build > Authentication, dihalaman Authentication, tekan Get Started. Selanjutnya pilih menu Sign-in method, aktifkan Email/Password, dan tekan Save. Seperti pada gambar dibawah ini:



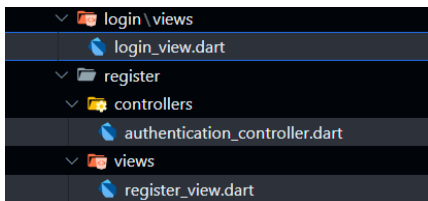
Maka akan aktif seperti pada gambar dibawah:



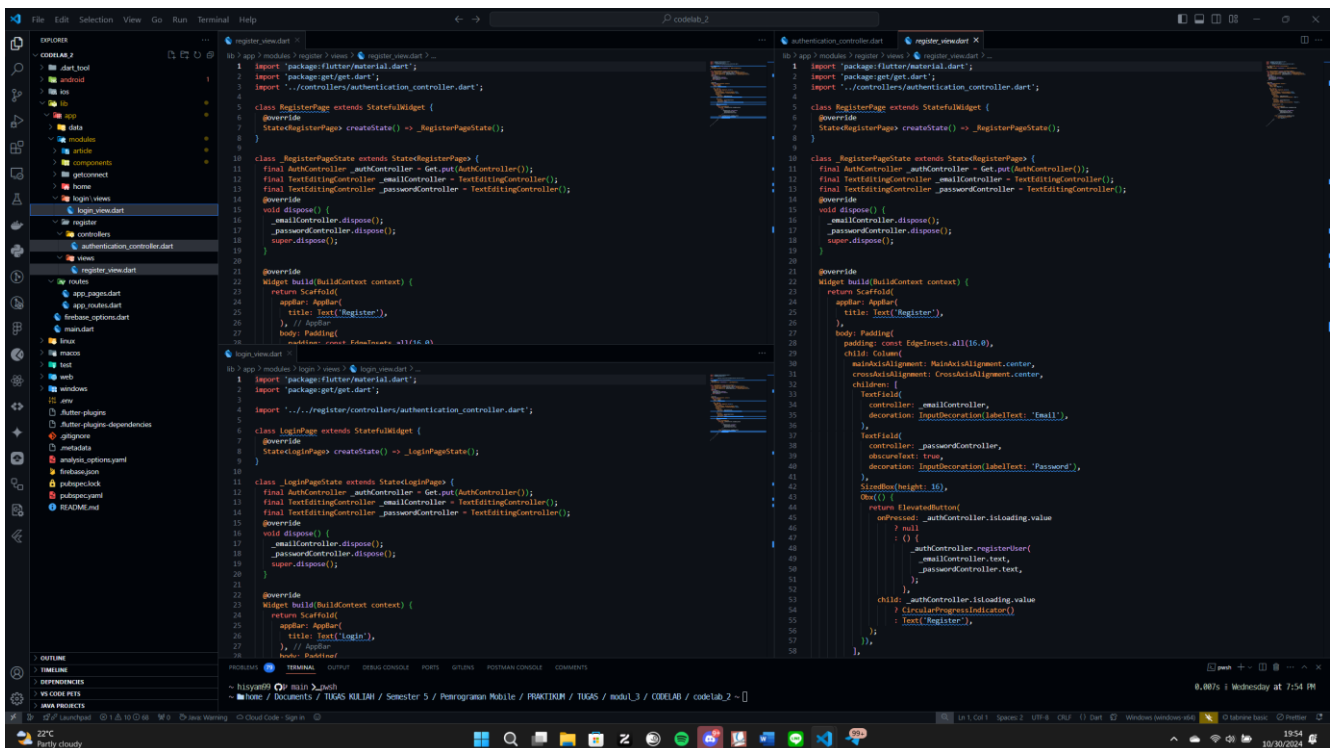
Langkah selanjutnya kita akan membuat halaman authentication, buatlah beberapa folder didalam path “lib/app/modules”, seperti pada gambar dibawah ini:



Lalu buatlah file di tiap folder controllers dan viewsnya, maka seperti dibawah ini:



Selanjutnya adalah membuat kodingan dari tiap – tiap file tersebut, yaitu sebagai berikut:



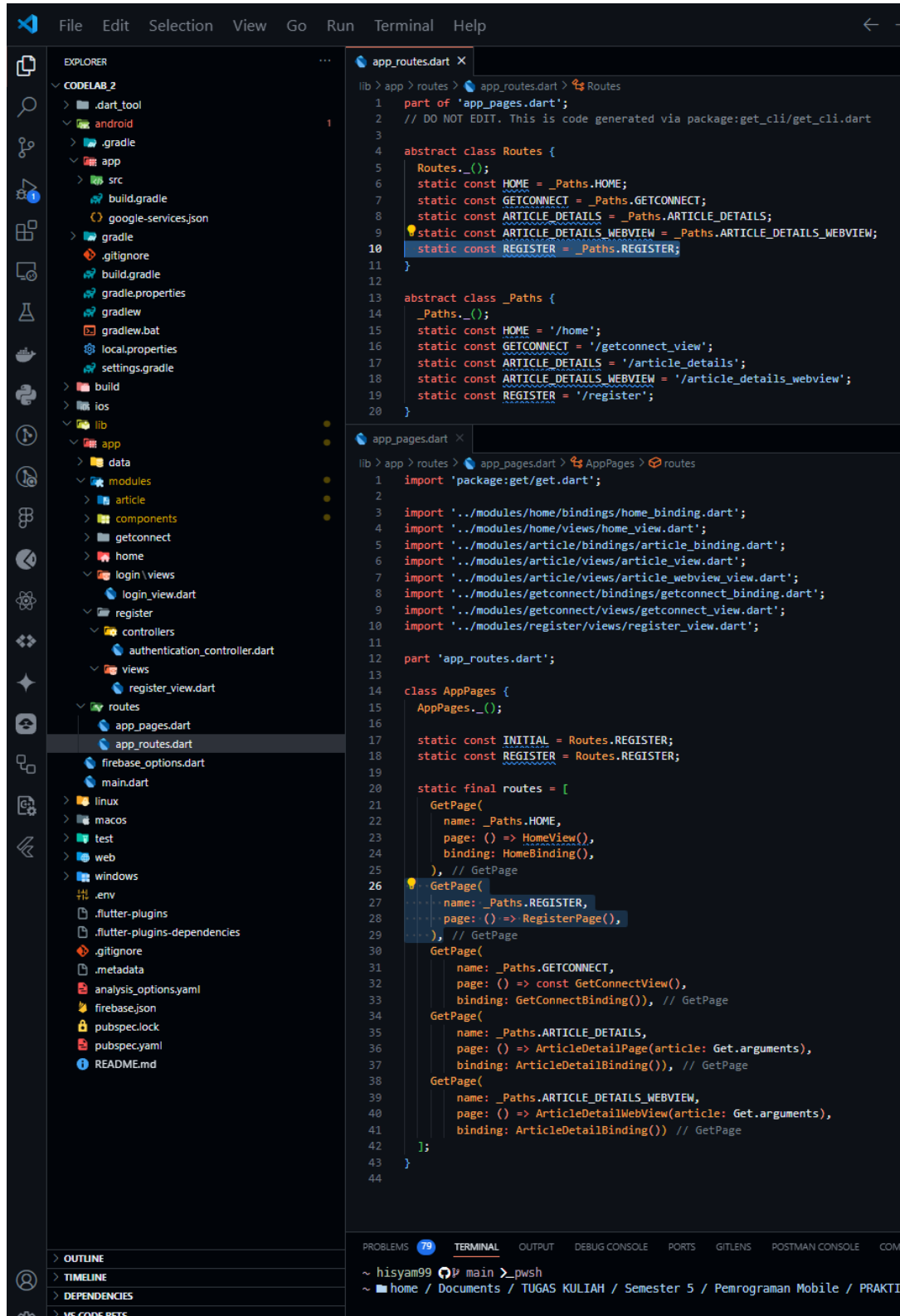
*kode lengkapnya dapat dicek disini

https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/blob/main/TUGAS/modul_3/CODELAB/codelab/lib/app/modules/register

https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/blob/main/TUGAS/modul_3/CODELAB/codelab/lib/app/modules/login

Pada kode yang sudah kita masukkan tadi itu ada kegunaannya masing – masing. File auth_controller.dart berguna untuk mengakses authentication kita pada Firebase yang sudah dikonfigurasi sebelumnya, lalu pada file register_view.dart dan login_view.dart berguna sebagai tampilan halaman register dan loginnya yang akan kita tampilkan di aplikasi nantinya, kedua file tersebut terhubung dengan auth_controller.dart.

Langkah Selanjutnya modifikasi file `app_pages.dart` dan `app_routes.dart` kita, tambahkan route yang akan mengarah kehalaman registrasi yang baru saja kita buat tadi. Yaitu sebagai berikut:



The screenshot shows the Visual Studio Code editor with two files open: `app_routes.dart` and `app_pages.dart`. The `EXPLORER` sidebar on the left shows the project structure, with the `lib` folder expanded to show the `app` subfolder, which contains `app_pages.dart` and `app_routes.dart`.

app_routes.dart

```
lib > app > routes > app_routes.dart > Routes
1  part of 'app_pages.dart';
2  // DO NOT EDIT. This is code generated via package:get_cli/get_cli.dart
3
4  abstract class Routes {
5    Routes._();
6    static const HOME = _Paths.HOME;
7    static const GETCONNECT = _Paths.GETCONNECT;
8    static const ARTICLE_DETAILS = _Paths.ARTICLE_DETAILS;
9    static const ARTICLE_DETAILS_WEBVIEW = _Paths.ARTICLE_DETAILS_WEBVIEW;
10   static const REGISTER = _Paths.REGISTER;
11 }
12
13 abstract class _Paths {
14   _Paths._();
15   static const HOME = '/home';
16   static const GETCONNECT = '/getconnect_view';
17   static const ARTICLE_DETAILS = '/article_details';
18   static const ARTICLE_DETAILS_WEBVIEW = '/article_details_webview';
19   static const REGISTER = '/register';
20 }
```

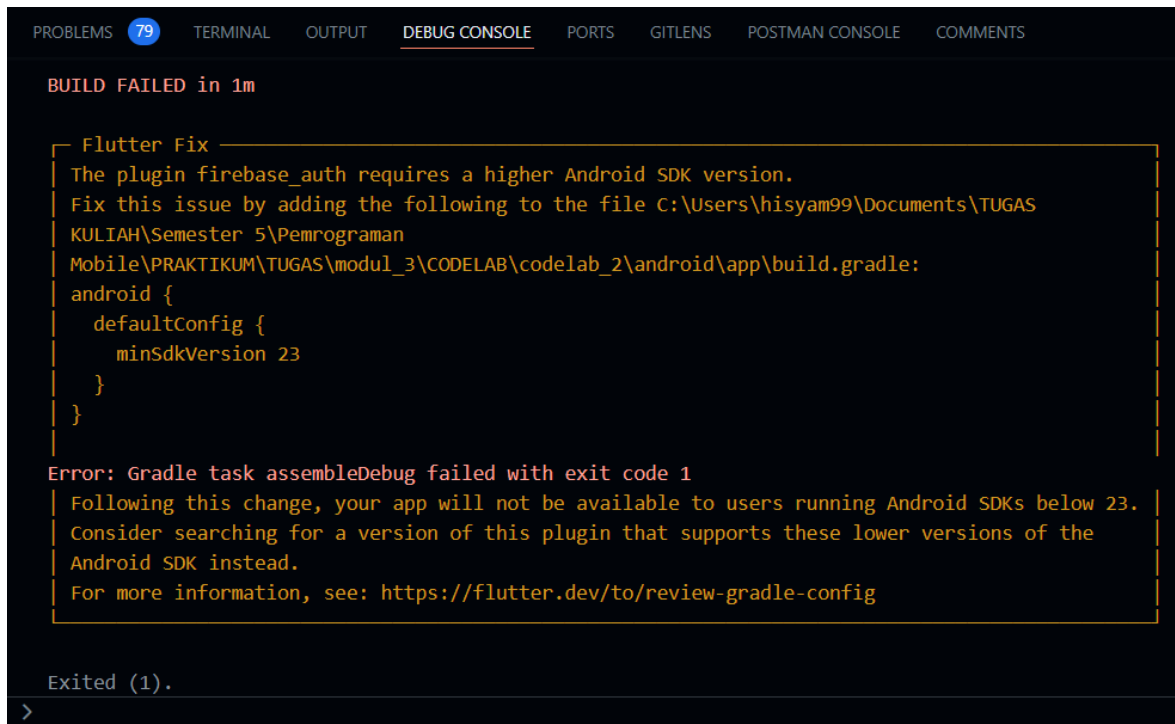
app_pages.dart

```
lib > app > routes > app_pages.dart > AppPages > routes
1  import 'package:get/get.dart';
2
3  import '../modules/home/bindings/home_binding.dart';
4  import '../modules/home/views/home_view.dart';
5  import '../modules/article/bindings/article_binding.dart';
6  import '../modules/article/views/article_view.dart';
7  import '../modules/article/views/article_webview_view.dart';
8  import '../modules/getconnect/bindings/getconnect_binding.dart';
9  import '../modules/getconnect/views/getconnect_view.dart';
10 import '../modules/register/views/register_view.dart';
11
12 part 'app_routes.dart';
13
14 class AppPages {
15   AppPages._();
16
17   static const INITIAL = Routes.REGISTER;
18   static const REGISTER = Routes.REGISTER;
19
20   static final routes = [
21     GetPage(
22       name: _Paths.HOME,
23       page: () => HomeView(),
24       binding: HomeBinding(),
25     ), // GetPage
26     GetPage(
27       name: _Paths.REGISTER,
28       page: () => RegisterPage(),
29     ), // GetPage
30     GetPage(
31       name: _Paths.GETCONNECT,
32       page: () => const GetConnectView(),
33       binding: GetConnectBinding(), // GetPage
34     ),
35     GetPage(
36       name: _Paths.ARTICLE_DETAILS,
37       page: () => ArticleDetailPage(article: Get.arguments),
38       binding: ArticleDetailBinding(), // GetPage
39     ),
40     GetPage(
41       name: _Paths.ARTICLE_DETAILS_WEBVIEW,
42       page: () => ArticleDetailWebView(article: Get.arguments),
43       binding: ArticleDetailBinding() // GetPage
44     ),
45   ];
46 }
```

The `TERMINAL` panel at the bottom shows the command prompt with the following text:

```
~ hisyam99 > main > pwsh
~ home / Documents / TUGAS KULIAH / Semester 5 / Pemrograman Mobile / PRAKTIKUM
```

Langkah selanjutnya kita coba melakukan Start Debugging, apakah aplikasi berhasil di build atau tidak. Setelah saya lakukan, ternyata ada error yaitu sebagai berikut:



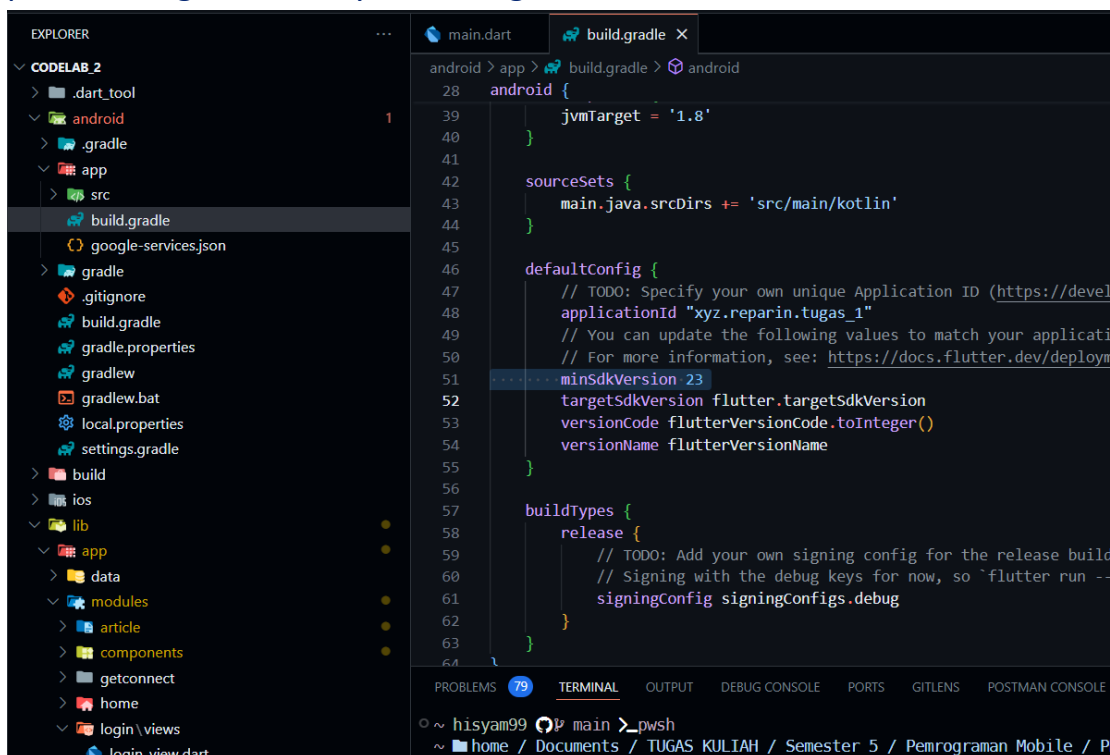
```
BUILD FAILED in 1m

Flutter Fix
The plugin firebase_auth requires a higher Android SDK version.
Fix this issue by adding the following to the file C:\Users\hisyam99\Documents\TUGAS KULIAH\Semester 5\Pemrograman Mobile\PRAKTIKUM\TUGAS\modul_3\CODELAB\codeLab_2\android\app\build.gradle:
android {
  defaultConfig {
    minSdkVersion 23
  }
}

Error: Gradle task assembleDebug failed with exit code 1
Following this change, your app will not be available to users running Android SDKs below 23.
Consider searching for a version of this plugin that supports these lower versions of the
Android SDK instead.
For more information, see: https://flutter.dev/to/review-gradle-config

Exited (1).
```

Error tersebut terjadi karena plugin `firebase_auth` memerlukan Android dengan versi SDK minimal 23. Cara mengatasinya cukup menambahkan variabel `minSdkVersion` dengan nilai "23" pada `build.gradle` kita, yaitu sebagai berikut:

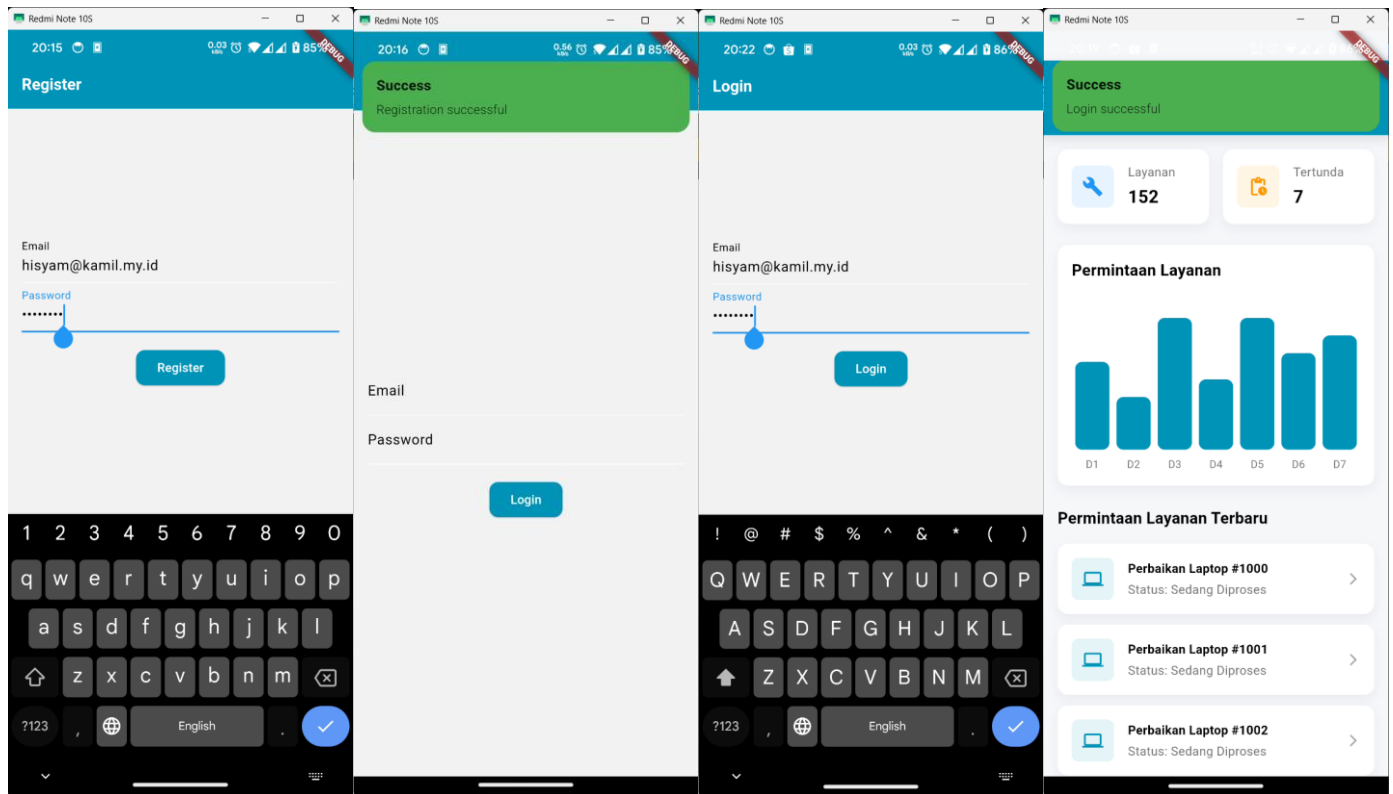


```
EXPLORER
CODELAB_2
  .dart_tool
  android
  .gradle
  app
  src
  build.gradle
  google-services.json
  gradle
  .gitignore
  build.gradle
  gradle.properties
  gradlew
  gradlew.bat
  local.properties
  settings.gradle
  build
  ios
  lib
  app
  data
  modules
  article
  components
  getconnect
  home
  login\views
  login_view.dart

main.dart build.gradle X
android > app > build.gradle > android
28 android {
39   jvmTarget = '1.8'
40 }
41
42 sourceSets {
43   main.java.srcDirs += 'src/main/kotlin'
44 }
45
46 defaultConfig {
47   // TODO: Specify your own unique Application ID (https://devel
48   applicationId "xyz.reparin.tugas_1"
49   // You can update the following values to match your applicati
50   // For more information, see: https://docs.flutter.dev/deploy
51   minSdkVersion 23
52   targetSdkVersion flutter.targetSdkVersion
53   versionCode flutter.VersionCode.toInteger()
54   versionName flutter.VersionName
55 }
56
57 buildTypes {
58   release {
59     // TODO: Add your own signing config for the release build
60     // Signing with the debug keys for now, so `flutter run --
61     signingConfig signingConfigs.debug
62   }
63 }
64

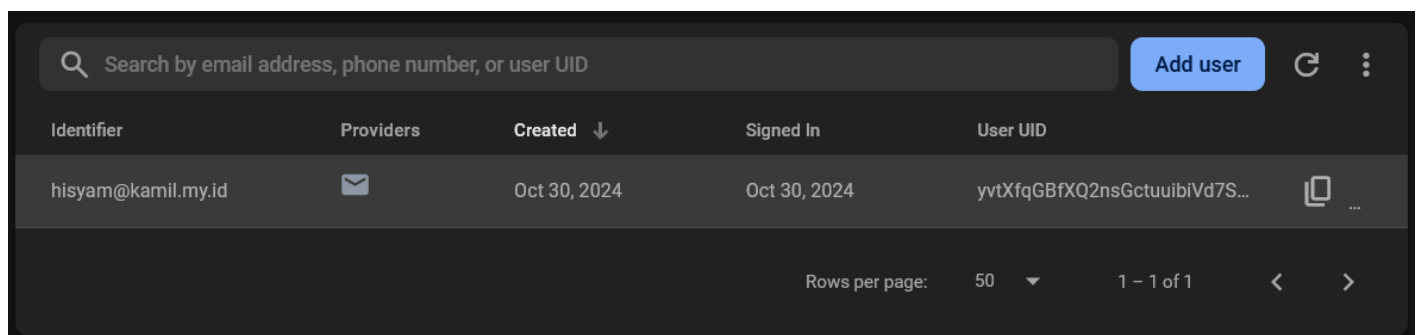
PROBLEMS 79 TERMINAL OUTPUT DEBUG CONSOLE PORTS GITLENS POSTMAN CONSOLE
~ hisyam99 ~ main > pwsh
~ home / Documents / TUGAS KULIAH / Semester 5 / Pemrograman Mobile / P
```

Setelah itu kita coba Start Debugging lagi, seharusnya tidak akan error lagi dan akan tampil halaman Register seperti pada gambar dibawah ini:



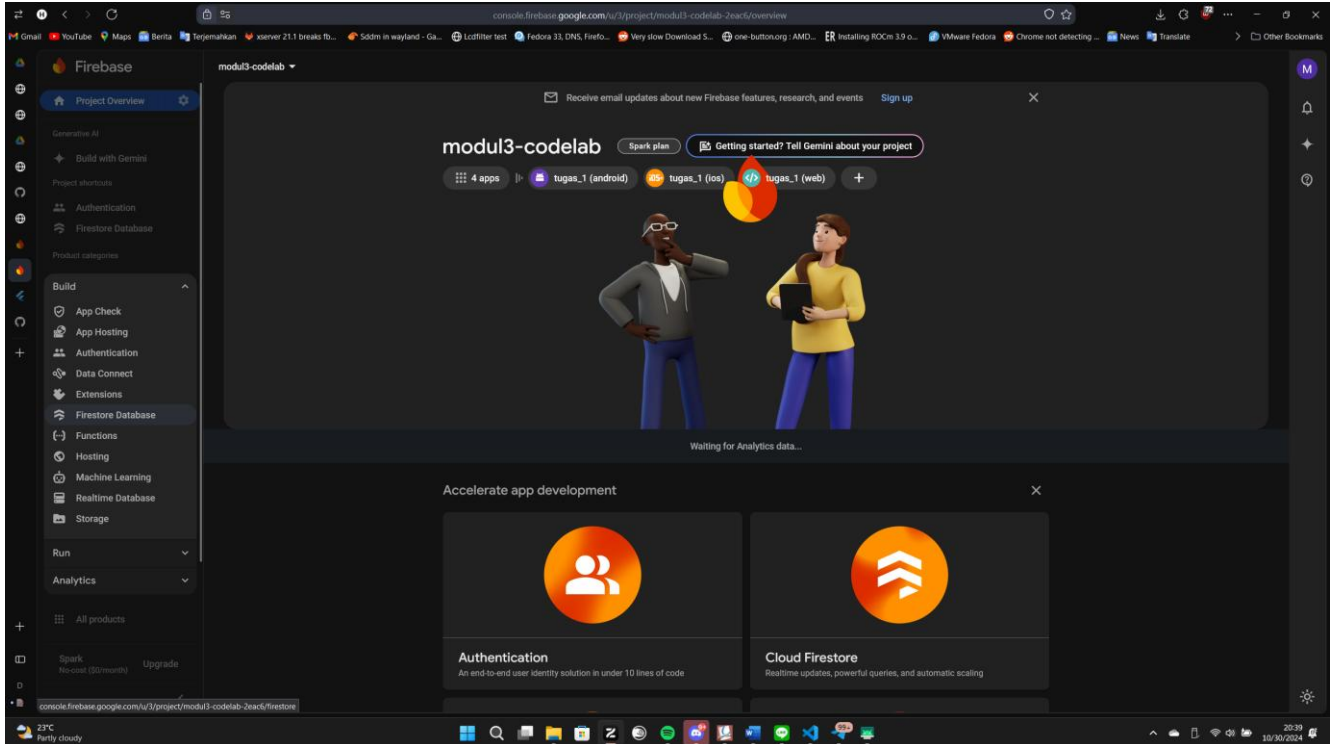
Dapat terlihat ketika kita mencoba untuk melakukan registrasi, muncul alert "Registration Successful", dan ketika kita coba untuk login juga menampilkan alert "Login Successful", ini berarti penerapan yang sudah kita lakukan telah berhasil, dan sudah terhubung ke Firebase.

Kita dapat melihat user yang sudah melakukan registrasi pada halaman Firebase Console berikut ini:

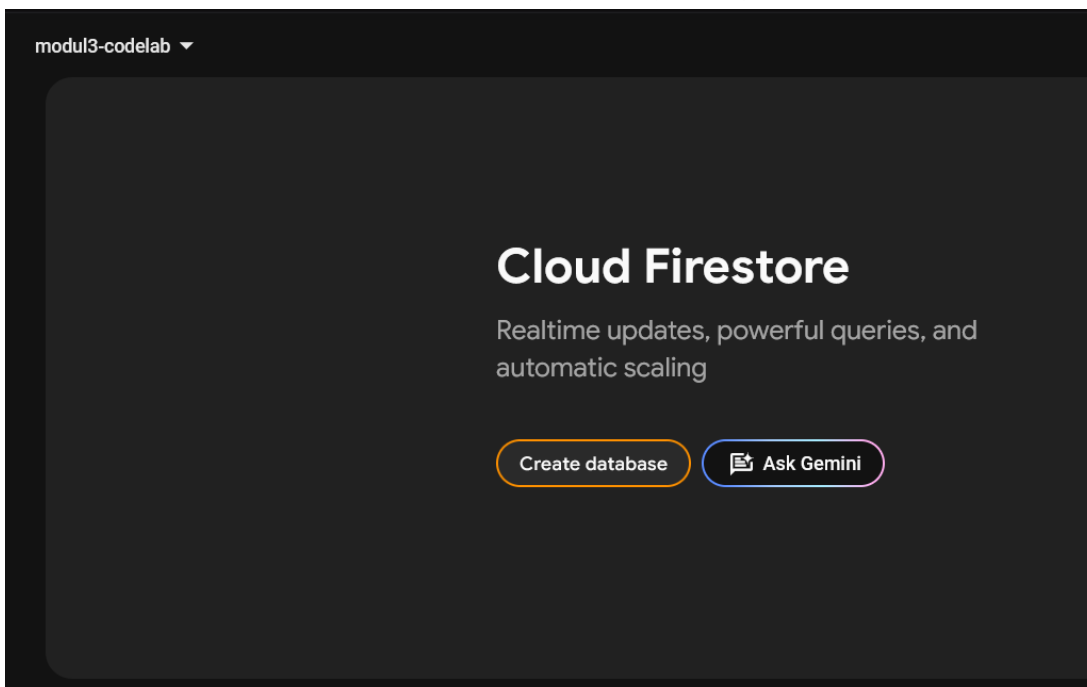


III. Melengkapi Kode (ToDo List dengan Edit dan Hapus task)

Langkah pertama kita masuk ke Firebase Console (ke project yang sudah dibuat sebelumnya), tekan menu Build, lalu cari Firestore Database, seperti pada gambar berikut ini:



Lalu klik Create Database seperti yang ada pada gambar berikut ini:



Set Locationnya ke yang paling dekat di tempat kita berada, disini saya memilih asia-southeast2 (Jakarta), lalu klik Next:

Create database ✕

1 Set name and location — 2 Secure rules

Database ID

(default)

Location

asia-southeast2 (Jakarta) ▼

ⓘ Your location setting is where your Cloud Firestore data will be stored

⚠ After you set this location, you cannot change it later. If this is your first database, your default Cloud Storage location will also be set to this location. [Learn more](#)

Cancel Next

Pilih Start in Production mode, lalu klik Create:

Create database ✕

1 Set name and location — 2 Secure rules

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.

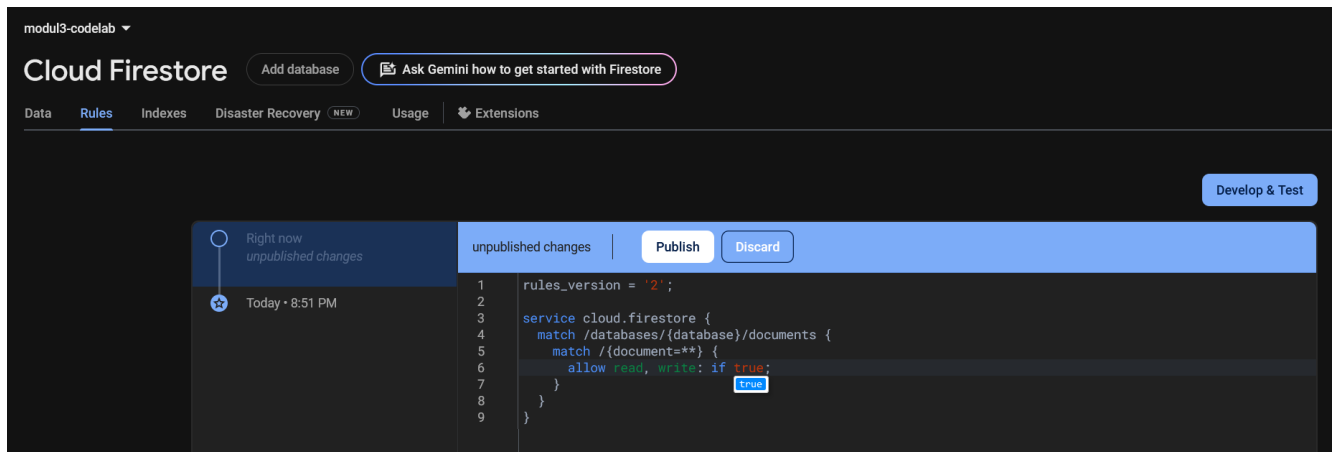
```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

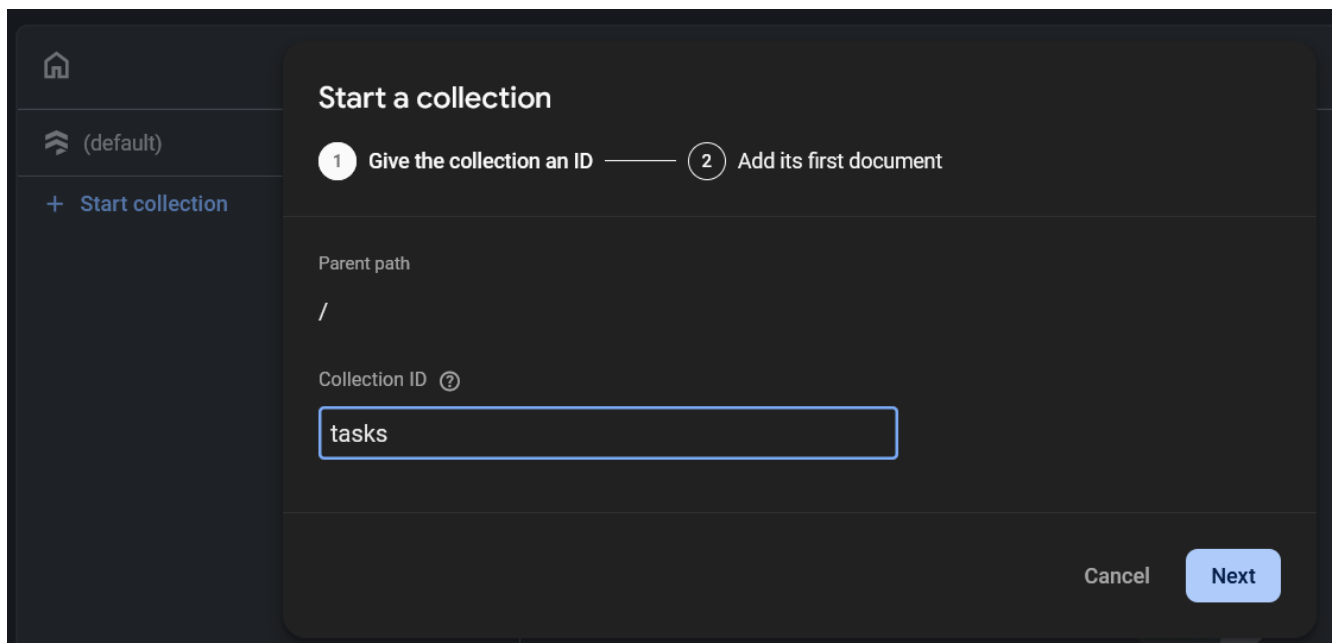
ⓘ All third party reads and writes will be denied

Cancel Create

Setelah database sudah dibuat, masuk ke halaman Rules. Konfigurasi agar database bisa melakukan Read/Write dengan cara merubah false menjadi true. Setelah itu klik Publish:



Kembali ke halaman Data, tekan (+ Start collection), disini kita membuat collection baru dan menamakan collection ID kita sebagai “tasks”, lalu tekan Next:



Lalu selanjutnya, kita mengisi beberapa field, pertama pada field Document ID kita tekan tombol Auto-ID agar secara otomatis meng-generate Document ID

Start a collection

✓ Give the collection an ID — 2 Add its first document

Document parent path

/tasks

Document ID ⓘ

Auto-ID

Required

Field	Type	Value
	string	

⊕ Add field

Cancel Save

Setelah itu kita menambahkan data kedalam database kita, buat field date, description, dan name. Pastikan menggunakan tipe data yang sesuai tiap – tiap fieldnya. Jika sudah, tekan tombol Save. Seperti yang ada pada gambar berikut ini:

Add a document

Parent path

/tasks

Document ID ⓘ

Sop3yJEA7HWf737KQbtH

Field	Type	Value
date	string	30 October 2024
description	string	n hari ini ngoding
name	string	hisyam99

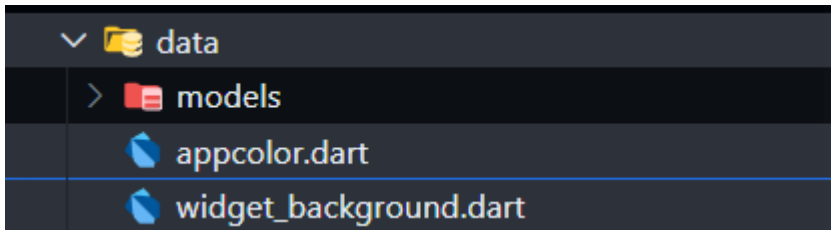
⊕ Add field

Cancel Save

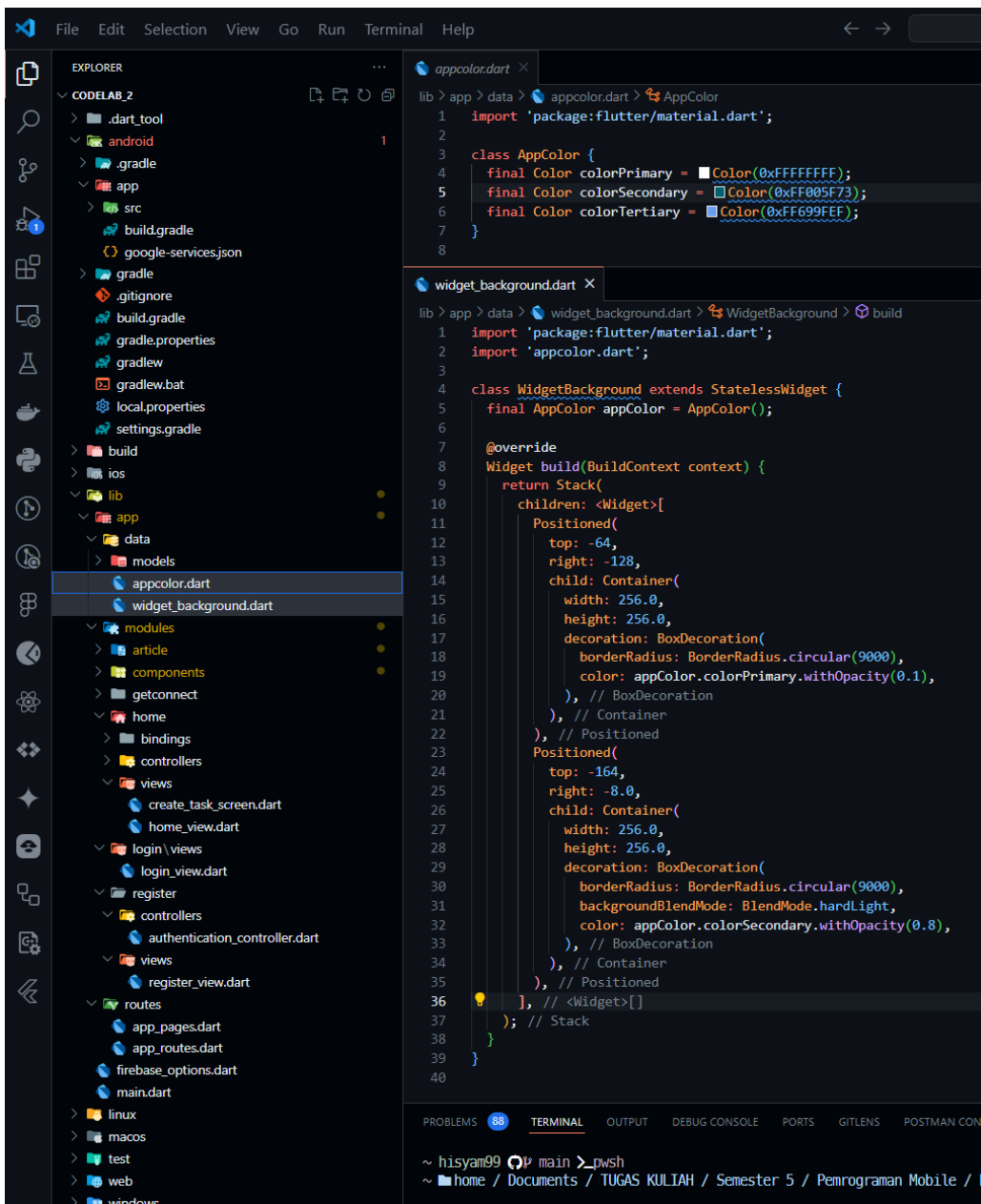
Dibawah ini adalah hasil dari database yang sudah kita buat:

tasks > Sop3yJEA7HWf..			More in Google Cloud
(default)	tasks	Sop3yJEA7HWf737KQbtH	
+ Start collection	+ Add document	+ Start collection	
tasks >	Sop3yJEA7HWf737KQbtH >	+ Add field	
		date: "30 October 2024"	
		description: "Kegiatan hari ini ngoding"	
		name: "hisyam99"	

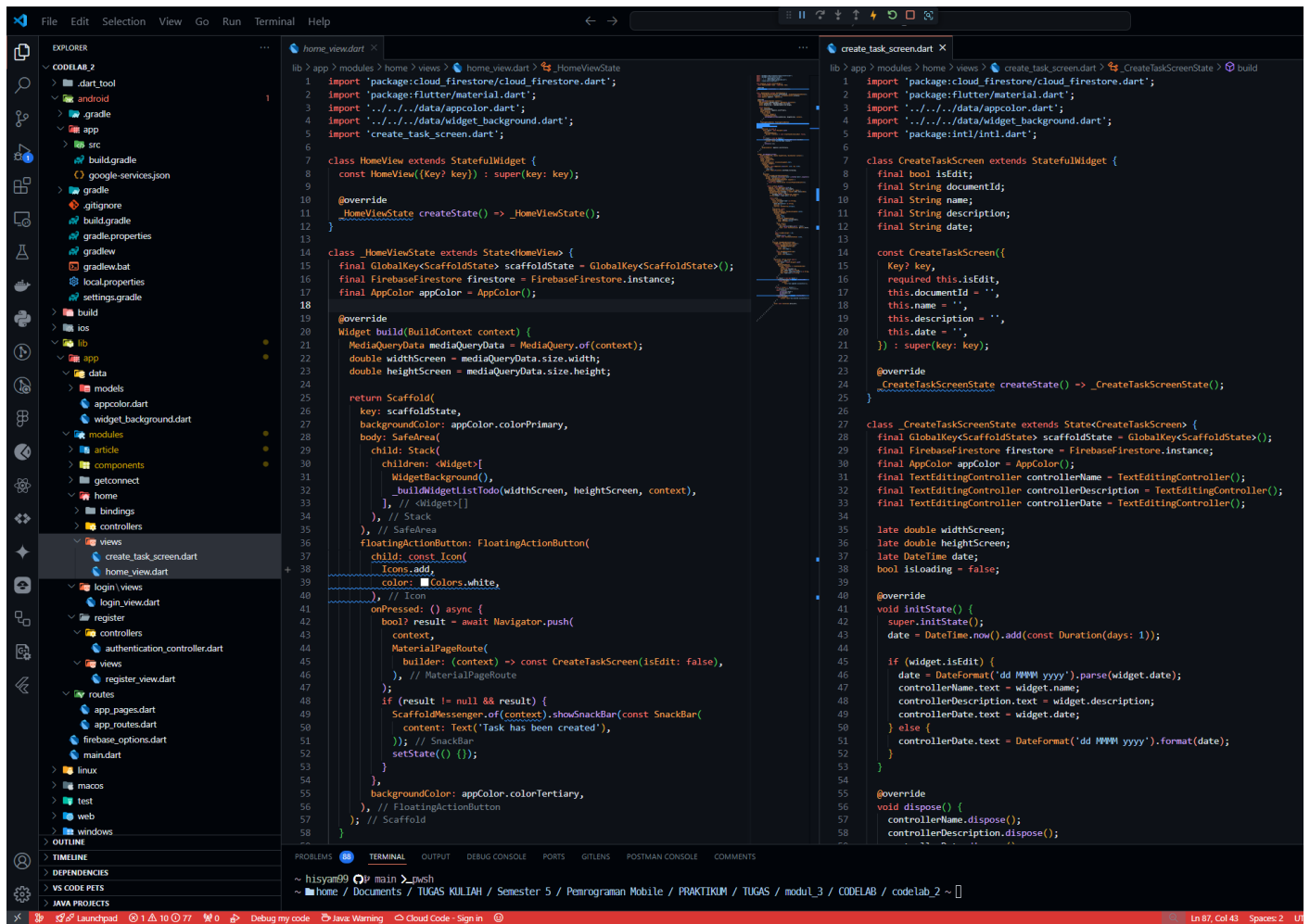
Agar dapat mengedit serta menghapus data melalui aplikasi kita, pertama buat file ini sebagai pengatur warna dari tampilan todolist kita, yaitu `app_color.dart` dan `widget_background.dart`. Buat didalam path `lib\app\data`, yaitu sebagai berikut:



Untuk isi dari setiap file tersebut yaitu sebagai berikut ini:



Selanjutnya kita akan memodifikasi file `home_view.dart` dan membuat file baru Bernama `create_task_screen.dart` pada path “`lib\app\modules\home\views`”, yaitu sebagai berikut



The screenshot displays the Visual Studio Code interface with two files open in the editor. The left pane shows the Explorer view with the project structure. The main editor area shows the `home_view.dart` file, which is a Flutter widget. The right pane shows the newly created `create_task_screen.dart` file, which is a Flutter widget. The `home_view.dart` file contains a `HomeView` class that extends `StatefulWidget` and a `HomeViewState` class that extends `State<HomeView>`. The `HomeView` class has a `createState` method that returns a `HomeViewState` object. The `HomeViewState` class has a `build` method that returns a `Scaffold` widget. The `create_task_screen.dart` file contains a `CreateTaskScreen` class that extends `StatefulWidget` and a `CreateTaskScreenState` class that extends `State<CreateTaskScreen>`. The `CreateTaskScreen` class has a `createState` method that returns a `CreateTaskScreenState` object. The `CreateTaskScreenState` class has a `build` method that returns a `Scaffold` widget.

```
lib > app > modules > home > views > home_view.dart > HomeViewState
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:flutter/material.dart';
3 import '../data/app_color.dart';
4 import '../data/widget_background.dart';
5 import 'create_task_screen.dart';
6
7 class HomeView extends StatefulWidget {
8   const HomeView({Key? key}) : super(key: key);
9
10  @override
11  HomeViewState createState() => HomeViewState();
12
13  class HomeViewState extends State<HomeView> {
14    final GlobalKey<ScaffoldState> scaffoldState = GlobalKey<ScaffoldState>();
15    final FirebaseFirestore firestore = FirebaseFirestore.instance;
16    final AppColor appColor = AppColor();
17
18    @override
19    Widget build(BuildContext context) {
20      MediaQueryData mediaQueryData = MediaQuery.of(context);
21      double widthScreen = mediaQueryData.size.width;
22      double heightScreen = mediaQueryData.size.height;
23
24      return Scaffold(
25        key: scaffoldState,
26        backgroundColor: appColor.colorPrimary,
27        body: SafeArea(
28          child: Stack(
29            children: <Widgets>[
30              WidgetBackground(),
31              buildWidgetListTodo(widthScreen, heightScreen, context),
32            ], // <Widget>[]
33          ), // Stack
34        ), // SafeArea
35        floatingActionButton: FloatingActionButton(
36          child: const Icon(
37            Icons.add,
38            color: Colors.white,
39          ), // Icon
40          onPressed: () async {
41            bool? result = await Navigator.push(
42              context,
43              MaterialPageRoute(
44                builder: (context) => const CreateTaskScreen(isEdit: false),
45              ), // MaterialPageRoute
46            );
47            if (result != null && result) {
48              ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
49                content: Text('Task has been created'),
50                )); // SnackBar
51              setState(() {});
52            }
53          },
54          backgroundColor: appColor.colorTertiary,
55        ), // FloatingActionButton
56      ); // Scaffold
57    }
58  }
59}
```

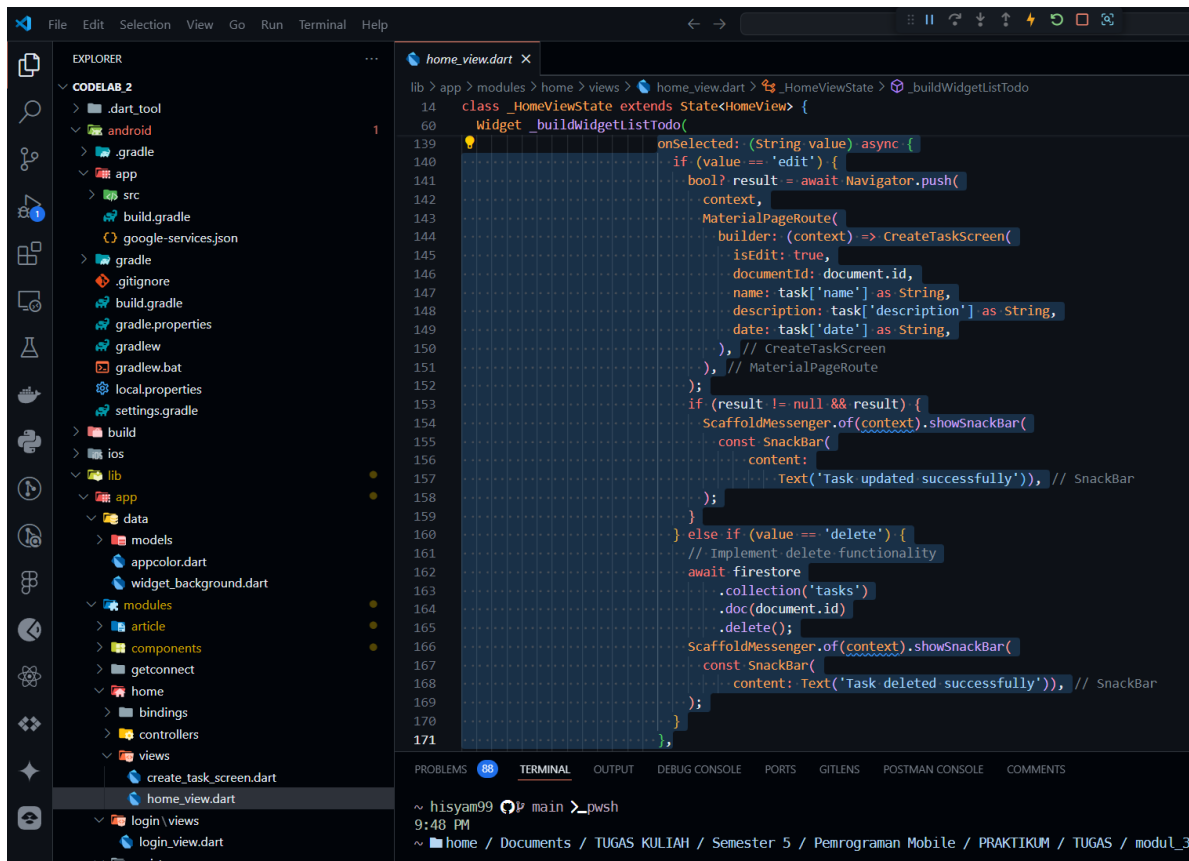
```
lib > app > modules > home > views > create_task_screen.dart > _CreateTaskScreenState > build
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:flutter/material.dart';
3 import '../data/app_color.dart';
4 import '../data/widget_background.dart';
5 import 'package:intl/intl.dart';
6
7 class CreateTaskScreen extends StatefulWidget {
8   final bool isEdit;
9   final String documentId;
10  final String name;
11  final String description;
12  final String date;
13
14  const CreateTaskScreen({
15    Key? key,
16    required this.isEdit,
17    this.documentId = '',
18    this.name = '',
19    this.description = '',
20    this.date = '',
21  }) : super(key: key);
22
23  @override
24  _CreateTaskScreenState createState() => _CreateTaskScreenState();
25
26  class _CreateTaskScreenState extends State<CreateTaskScreen> {
27    final GlobalKey<ScaffoldState> scaffoldState = GlobalKey<ScaffoldState>();
28    final FirebaseFirestore firestore = FirebaseFirestore.instance;
29    final AppColor appColor = AppColor();
30    final TextEditingController controllerName = TextEditingController();
31    final TextEditingController controllerDescription = TextEditingController();
32    final TextEditingController controllerDate = TextEditingController();
33
34    late double widthScreen;
35    late double heightScreen;
36    late DateTime date;
37    bool isLoading = false;
38
39    @override
40    void initState() {
41      super.initState();
42      date = DateTime.now().add(const Duration(days: 1));
43
44      if (widget.isEdit) {
45        date = DateFormat('dd MMMM yyyy').parse(widget.date);
46        controllerName.text = widget.name;
47        controllerDescription.text = widget.description;
48        controllerDate.text = widget.date;
49      } else {
50        controllerDate.text = DateFormat('dd MMMM yyyy').format(date);
51      }
52    }
53
54    @override
55    void dispose() {
56      controllerName.dispose();
57      controllerDescription.dispose();
58    }
59  }
60}
```

*kode lengkapnya dapat dicek disini

https://github.com/hisyam99/PRAK-PEMROGRAMAN-MOBILE-2024/blob/main/TUGAS/modul_3/CODELAB/codelab/lib/app/modules/home/views

Kedua file tersebut mengimplementasikan `app_color.dart` dan `widget_background.dart` yang sudah kita buat sebelumnya.

Lalu untuk kode yang dimodifikasi tadi itu ada pada home_view.dart, untuk kode modifikasi yang saya tambahkan yaitu sebagai berikut:

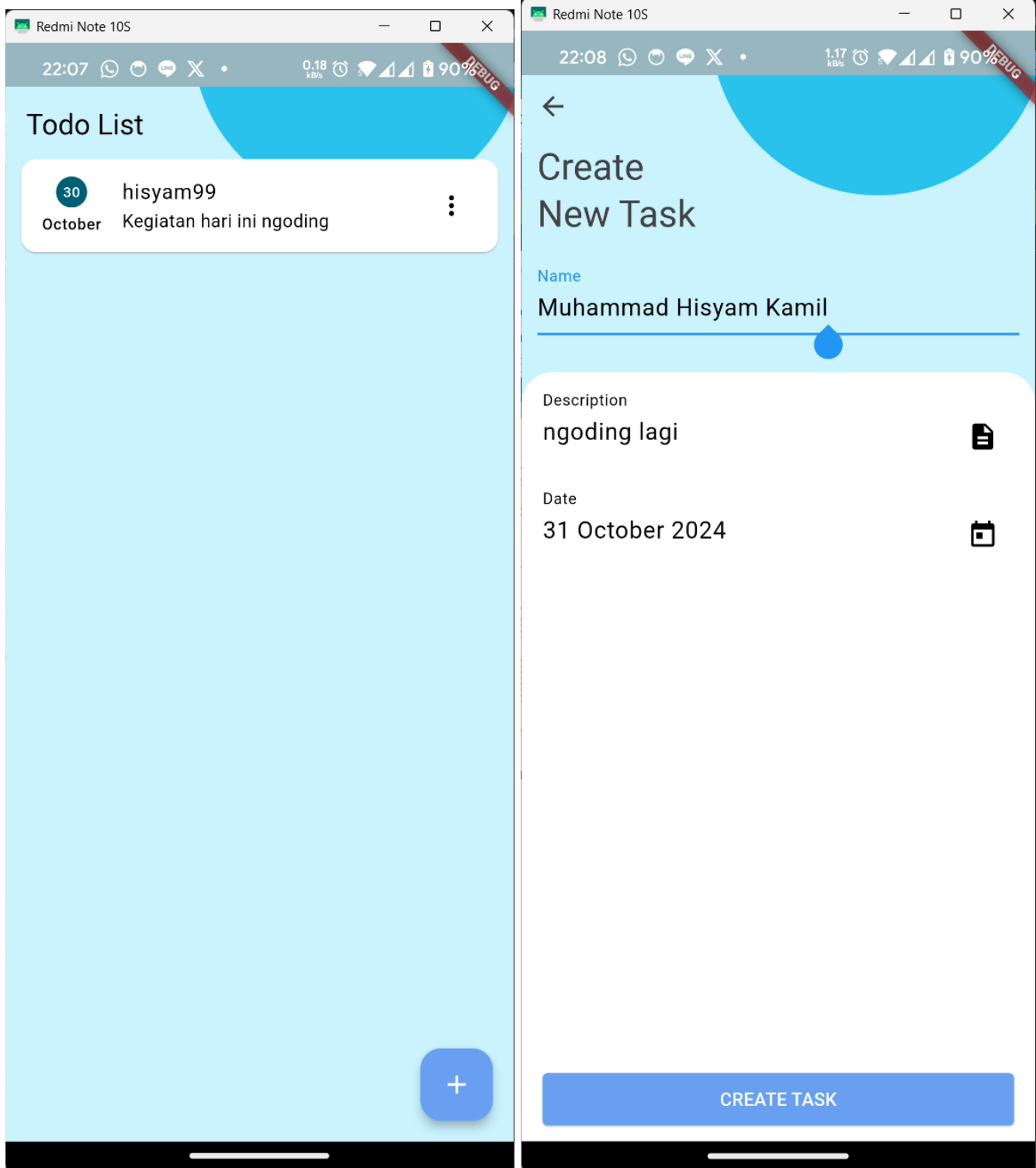


```
lib > app > modules > home > views > home_view.dart > _HomeViewState > _buildWidgetListTodo
14 class _HomeViewState extends State<HomeView> {
60 widget _buildWidgetListTodo(
139   onSelected: (String value) async {
140     if (value == 'edit') {
141       bool? result = await Navigator.push(
142         context,
143         MaterialPageRoute(
144           builder: (context) => CreateTaskScreen(
145             isEdit: true,
146             documentId: document.id,
147             name: task['name'] as String,
148             description: task['description'] as String,
149             date: task['date'] as String,
150           ), // CreateTaskScreen
151         ), // MaterialPageRoute
152       );
153       if (result != null && result) {
154         ScaffoldMessenger.of(context).showSnackBar(
155           const SnackBar(
156             content:
157               Text('Task updated successfully'), // SnackBar
158           );
159       }
160     } else if (value == 'delete') {
161       // Implement delete functionality
162       await firestore
163         .collection('tasks')
164         .doc(document.id)
165         .delete();
166       ScaffoldMessenger.of(context).showSnackBar(
167         const SnackBar(
168           content: Text('Task deleted successfully'), // SnackBar
169         );
170     }
171   },
```

Saya melakukan modifikasi pada kode ini agar pengguna bisa mengedit dan menghapus tugas di aplikasi. Saat pengguna memilih opsi edit, saya menggunakan Navigator.push untuk membawa pengguna ke layar CreateTaskScreen melalui MaterialPageRoute. Layar baru ini menerima argumen seperti isEdit, documentId, name, description, dan date, yang nantinya digunakan untuk memuat data awal di tampilan edit, sehingga pengguna dapat melihat detail tugas yang ingin diedit. Setelah pengguna menyelesaikan pengeditan dan kembali ke layar sebelumnya (dengan pop yang mengembalikan nilai true), SnackBar akan muncul dengan pesan "Task updated successfully" sebagai notifikasi bahwa tugas berhasil diperbarui.

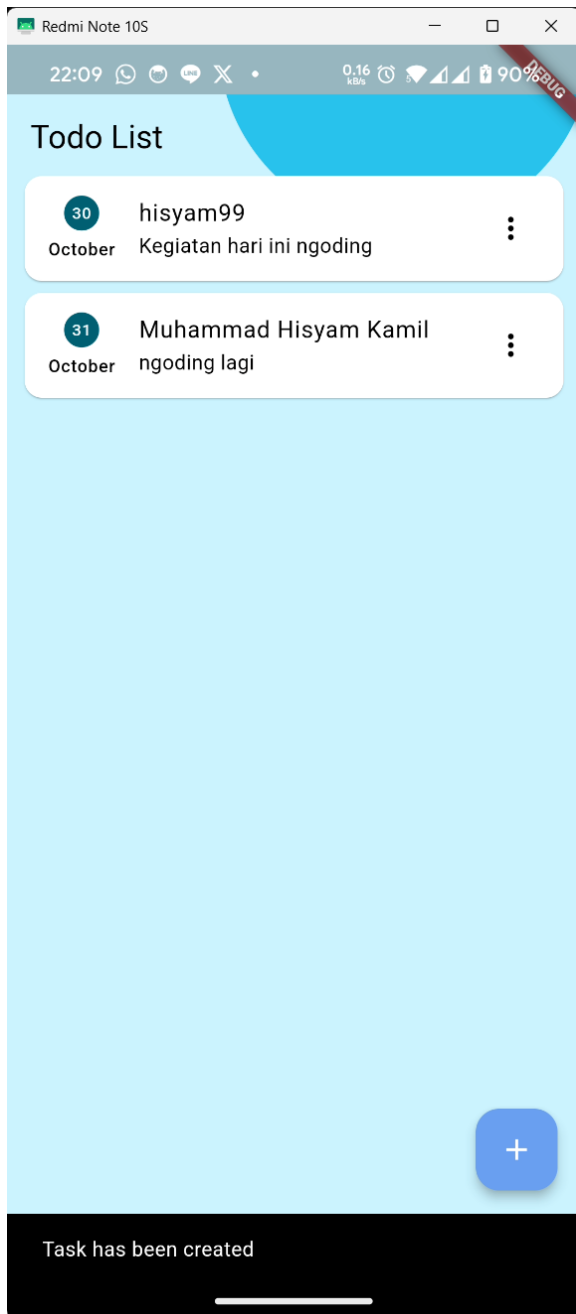
Untuk opsi hapus, saya menggunakan Firebase Firestore untuk menghapus dokumen yang sesuai dengan document.id dari koleksi tasks. Firestore menyediakan metode delete untuk menghapus data secara permanen dari database. Setelah penghapusan berhasil, ScaffoldMessenger akan menampilkan SnackBar dengan pesan "Task deleted successfully" sebagai konfirmasi bahwa tugas sudah dihapus dari database.

Setelah kita sudah menerapkan semuanya, saatnya kita Start Debugging dan menguji coba hasil modifikasi yang sudah dilakukan. Jika berhasil maka akan seperti gambar berikut ini:

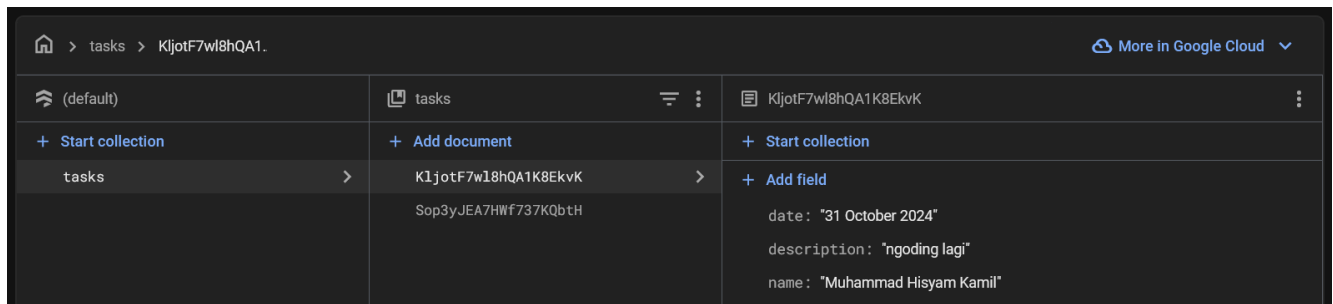


Dapat terlihat disini kita bisa menambahkan task baru yaitu dengan menekan tombol +. Setelah mengisi detailnya, kita dapat langsung menekan tombol Create Task

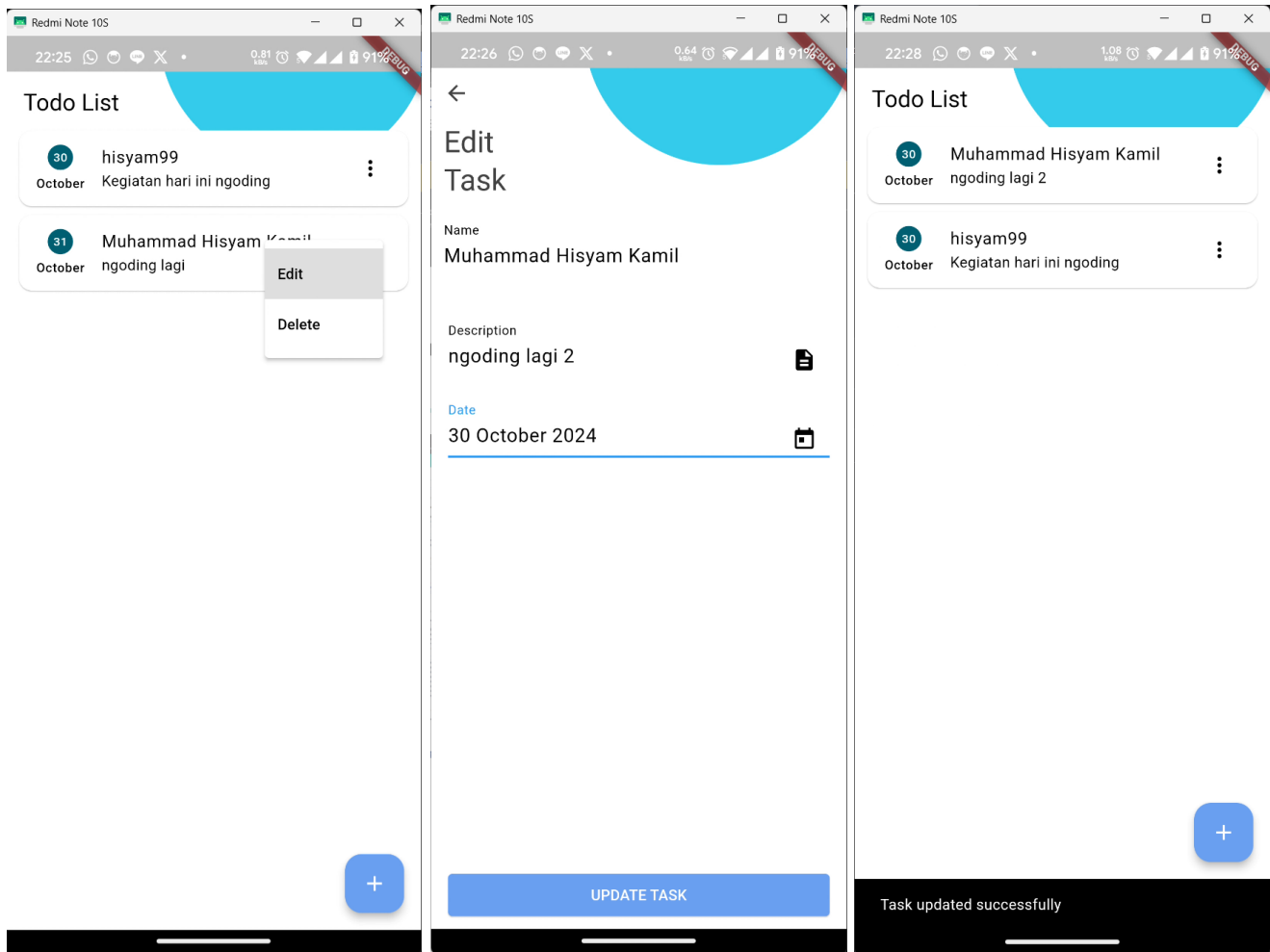
Dapat dilihat listnya otomatis ter-update:



Begitu juga pada Firestore Databasenya jika kita cek melalui Firebase Console:



Selanjutnya kita mencoba fitur edit dan delete, yaitu sebagai berikut:



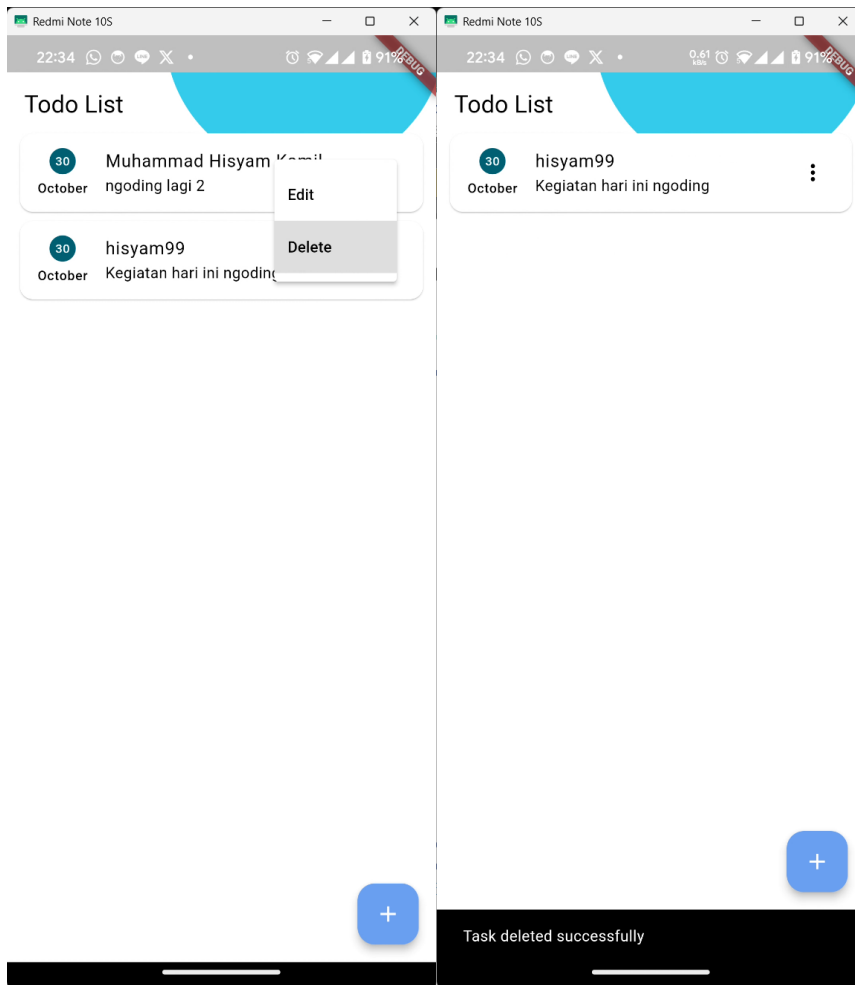
Sekarang data sudah berubah termasuk yang ada didalam Firestore Database:
Sebelum:

tasks > K1jotF7w18hQA1.			More in Google Cloud
(default)	tasks	K1jotF7w18hQA1K8EkvK	
+ Start collection	+ Add document	+ Start collection	
tasks	K1jotF7w18hQA1K8EkvK	+ Add field	
	Sop3yJEA7HWF737KQbtH	date: "31 October 2024"	
		description: "ngoding lagi"	
		name: "Muhammad Hisyam Kamil"	

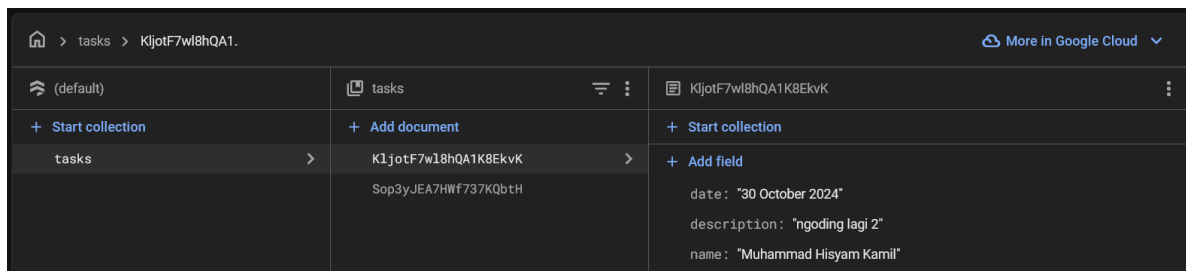
Sesudah:

tasks > K1jotF7w18hQA1.			More in Google Cloud
(default)	tasks	K1jotF7w18hQA1K8EkvK	
+ Start collection	+ Add document	+ Start collection	
tasks	K1jotF7w18hQA1K8EkvK	+ Add field	
	Sop3yJEA7HWF737KQbtH	date: "30 October 2024"	
		description: "ngoding lagi 2"	
		name: "Muhammad Hisyam Kamil"	

Untuk fitur Delete yaitu sebagai berikut:



Sekarang data tersebut sudah terhapus termasuk didalam Firestore Databasenya:
Sebelum:



Sesudah:

