Nama : Fadhila Agil Permana

NIM : 2211104006

Kelas : SE06-1

TUGAS 8 API.

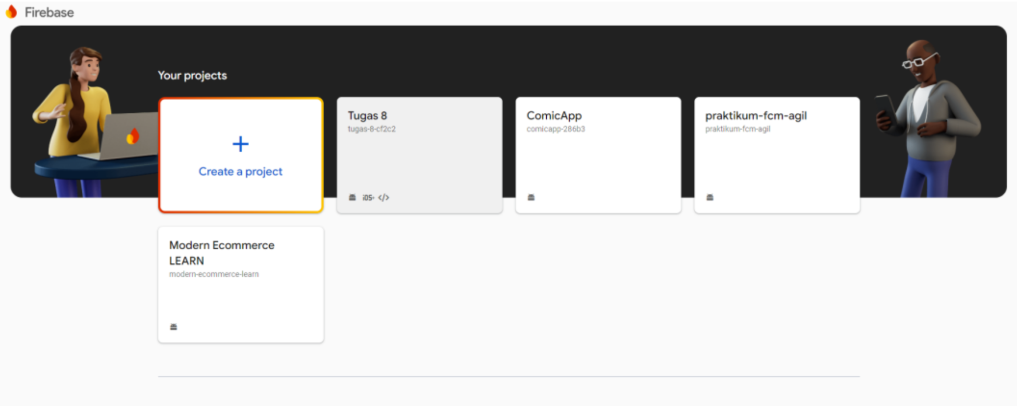
teknis terkait dengan Tugas 8: Rest API

- boleh menggunakan getX (global state managemen) atau local state management (setState)
- boleh mengunakan supabase untuk transaksi non dummy data
- atau boleh pakai api dummy mocki.io atau yang sejenisnya.

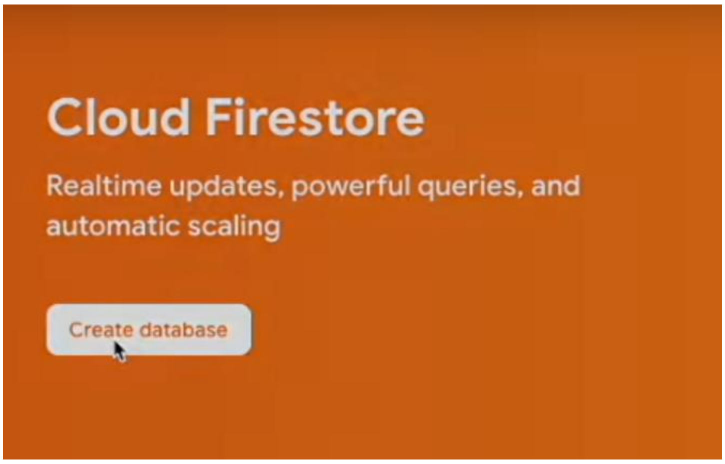
Pengerjaan, Menggunakan FIREBASE

- Material Belajar : https://www.youtube.com/watch?v=iQOvD0y-xnw

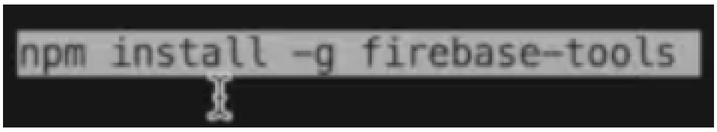
1. Setup Firebase :
   a. Buat Project Baru



   b. Firestore Buat Database

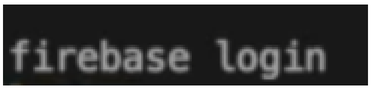


2. Setup Project Flutter
    a. Install. **Kemungkinan butuh bantuan dari google atau source lainnya**



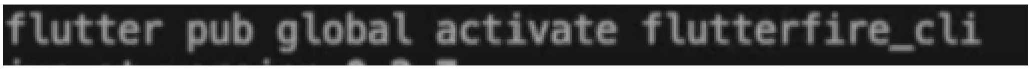
```
npm install -g firebase-tools
```

    b. Login



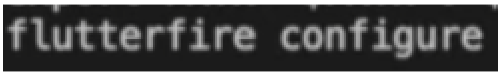
```
firebase login
```

    c. Activate CLI



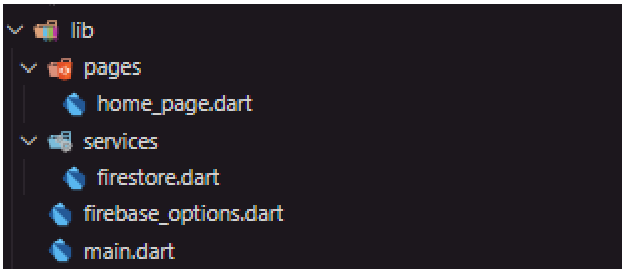
```
flutter pub global activate flutterfire_cli
```

    d. Configurasi, memilih project nya



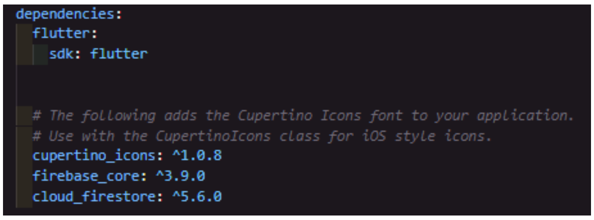
```
flutterfire configure
```

3. Ngoding
    a. File Structure
        i.



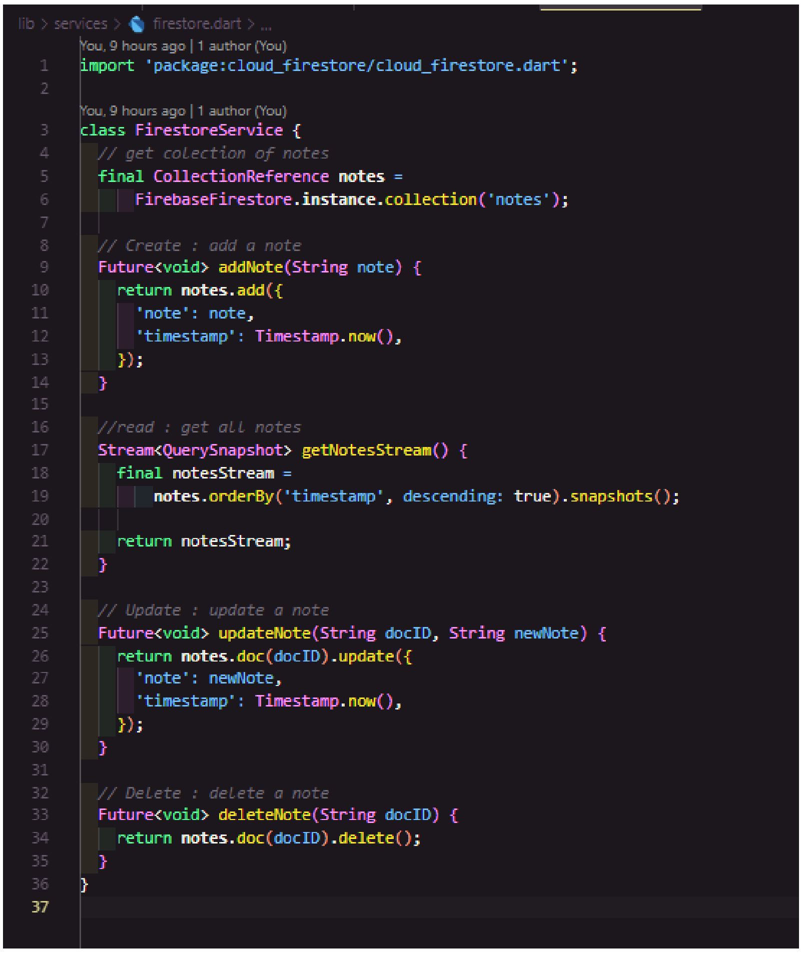
```
lib
  pages
    home_page.dart
  services
    firestore.dart
  firebase_options.dart
  main.dart
```

    b. Pubsec.yaml



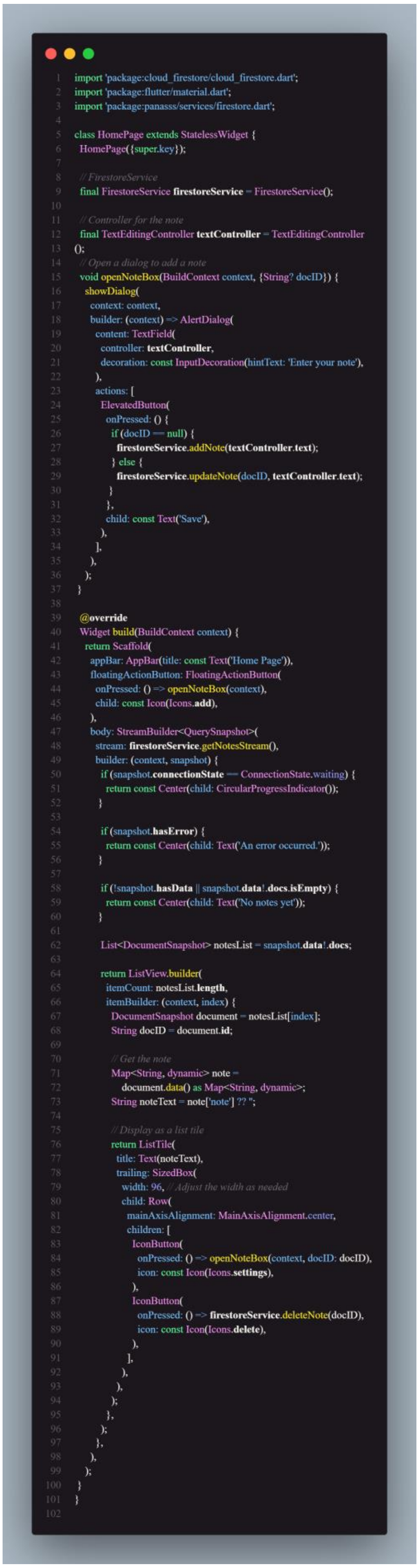
```
dependencies:
  flutter:
    sdk: flutter


  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  firebase_core: ^3.9.0
  cloud_firestore: ^5.6.0
```

c. Service, Firestore.dart


lib > services > firestore.dart > ...
You, 9 hours ago | 1 author (You)

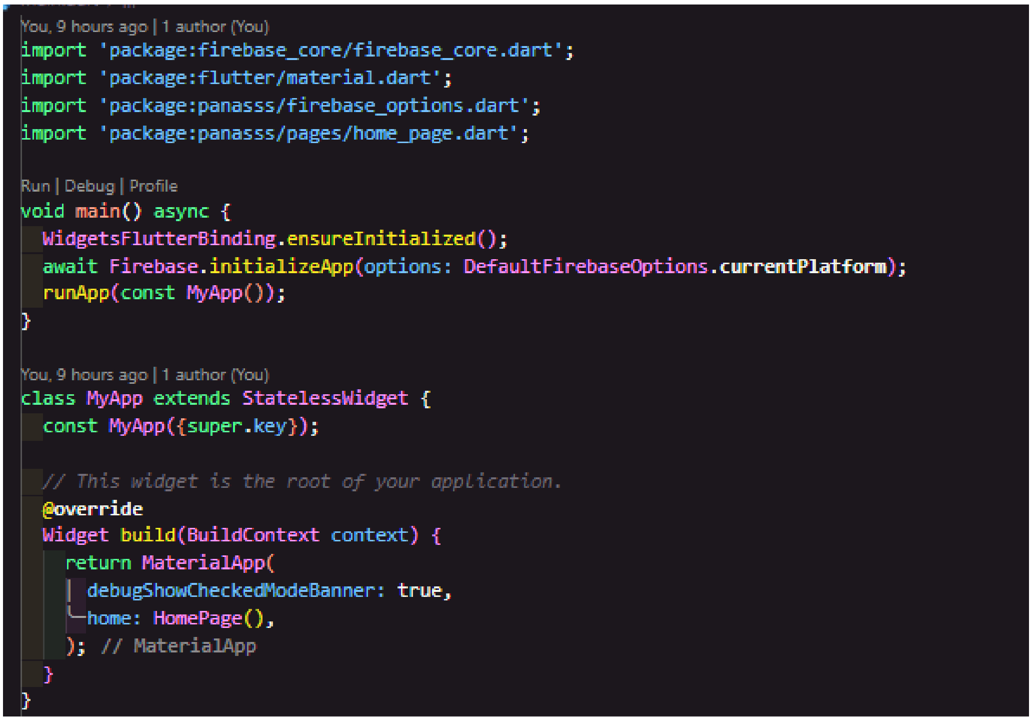
```dart
import 'package:cloud_firestore/cloud_firestore.dart';

class FirestoreService {
  // get colection of notes
  final CollectionReference notes =
      FirebaseFirestore.instance.collection('notes');

  // Create : add a note
  Future<void> addNote(String note) {
    return notes.add({
      'note': note,
      'timestamp': Timestamp.now(),
    });
  }

  //read : get all notes
  Stream<QuerySnapshot> getNotesStream() {
    final notesStream =
        notes.orderBy('timestamp', descending: true).snapshots();

    return notesStream;
  }

  // Update : update a note
  Future<void> updateNote(String docID, String newNote) {
    return notes.doc(docID).update({
      'note': newNote,
      'timestamp': Timestamp.now(),
    });
  }

  // Delete : delete a note
  Future<void> deleteNote(String docID) {
    return notes.doc(docID).delete();
  }
}
```

d. HomePage nya.



```dart
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:panasss/services/firestore.dart';

class HomePage extends StatelessWidget {
  HomePage({super.key});

  // FirestoreService
  final FirestoreService firestoreService = FirestoreService();

  // Controller for the note
  final TextEditingController textController = TextEditingController
();
  // Open a dialog to add a note
  void openNoteBox(BuildContext context, {String? docID}) {
    showDialog(
      context: context,
      builder: (context) => AlertDialog(
        content: TextField(
          controller: textController,
          decoration: const InputDecoration(hintText: 'Enter your note'),
        ),
        actions: [
          ElevatedButton(
            onPressed: () {
              if (docID == null) {
                firestoreService.addNote(textController.text);
              } else {
                firestoreService.updateNote(docID, textController.text);
              }
            },
            child: const Text('Save'),
          ),
        ],
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Home Page')),
      floatingActionButton: FloatingActionButton(
        onPressed: () => openNoteBox(context),
        child: const Icon(Icons.add),
      ),
      body: StreamBuilder<QuerySnapshot>(
        stream: firestoreService.getNotesStream(),
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
            return const Center(child: CircularProgressIndicator());
          }

          if (snapshot.hasError) {
            return const Center(child: Text('An error occurred.'));
          }

          if (!snapshot.hasData || snapshot.data!.docs.isEmpty) {
            return const Center(child: Text('No notes yet'));
          }

          List<DocumentSnapshot> notesList = snapshot.data!.docs;

          return ListView.builder(
            itemCount: notesList.length,
            itemBuilder: (context, index) {
              DocumentSnapshot document = notesList[index];
              String docID = document.id;

              // Get the note
              Map<String, dynamic> note =
                  document.data() as Map<String, dynamic>;
              String noteText = note['note'] ?? '';

              // Display as a list tile
              return ListTile(
                title: Text(noteText),
                trailing: SizedBox(
                  width: 96, // Adjust the width as needed
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      IconButton(
                        onPressed: () => openNoteBox(context, docID: docID),
                        icon: const Icon(Icons.settings),
                      ),
                      IconButton(
                        onPressed: () => firestoreService.deleteNote(docID),
                        icon: const Icon(Icons.delete),
                      ),
                    ],
                  ),
                ),
              );
            },
          );
        },
      ),
    );
  }
}
```

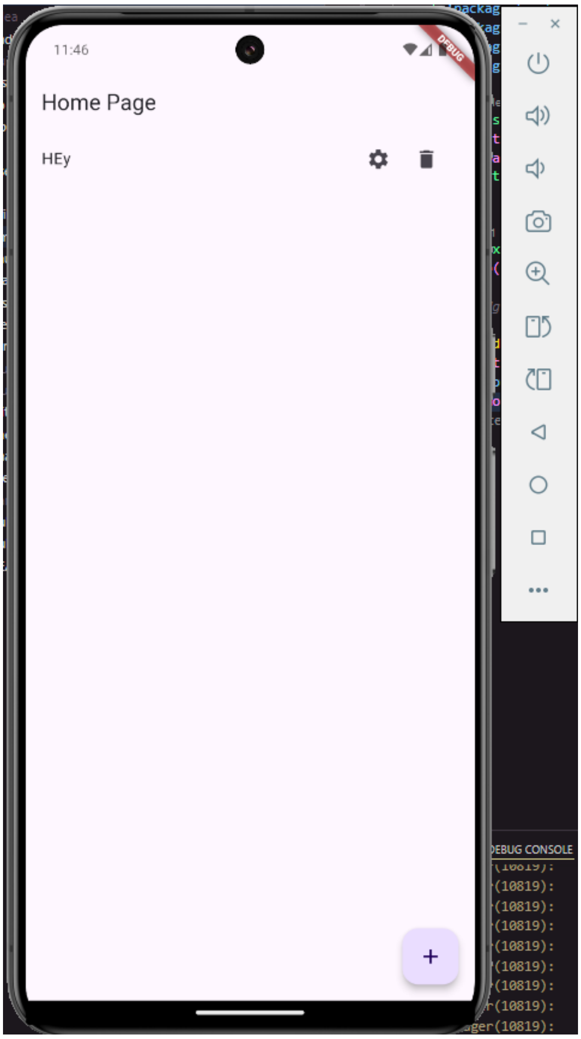
e. Main.dart



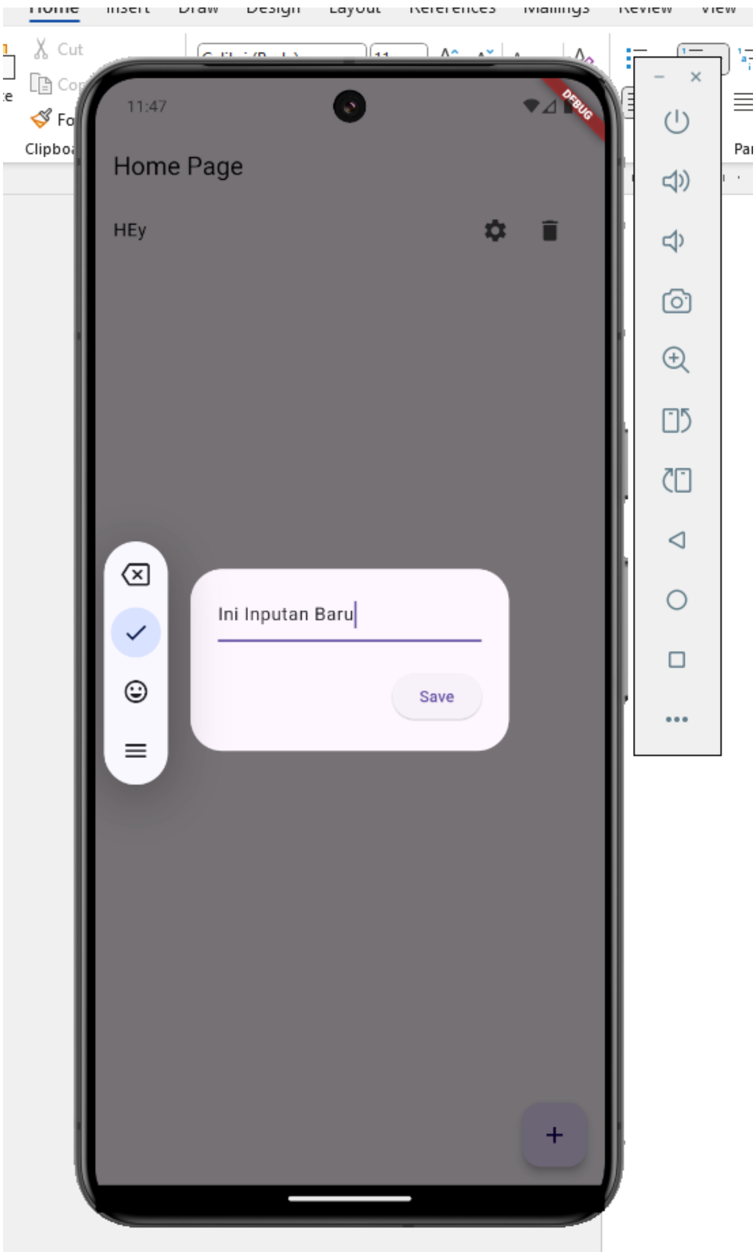
```dart
You, 9 hours ago | 1 author (You)
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:panasss/firebase_options.dart';
import 'package:panasss/pages/home_page.dart';

Run | Debug | Profile
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
  runApp(const MyApp());
}

You, 9 hours ago | 1 author (You)
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: true,
      home: HomePage(),
    ); // MaterialApp
  }
}
```

4. Output

a. Tampilan Awal



b. Create



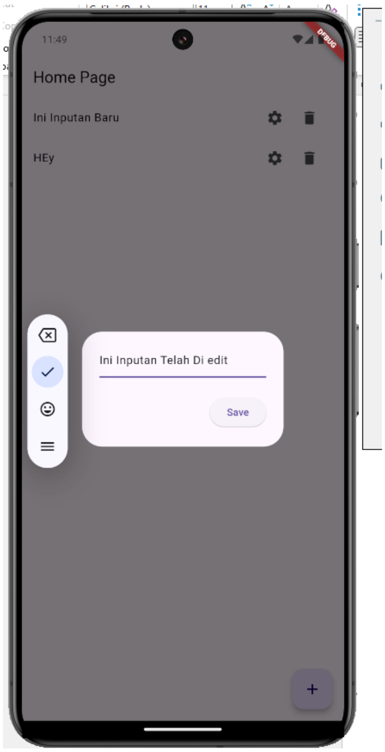
Home Page

HEy

Ini Inputan Baru

Save

c. Update

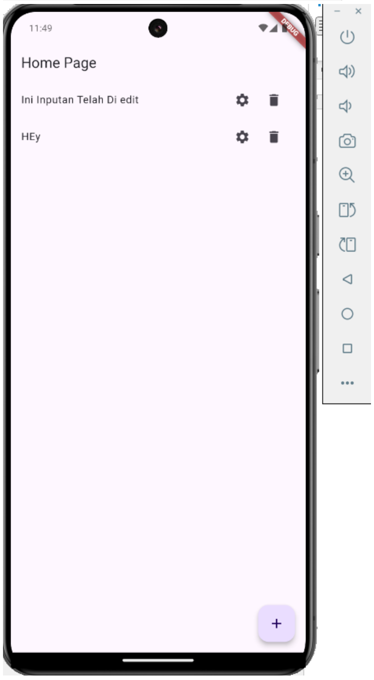


Home Page

Ini Inputan Baru

HEy

Ini Inputan Telah Di edit

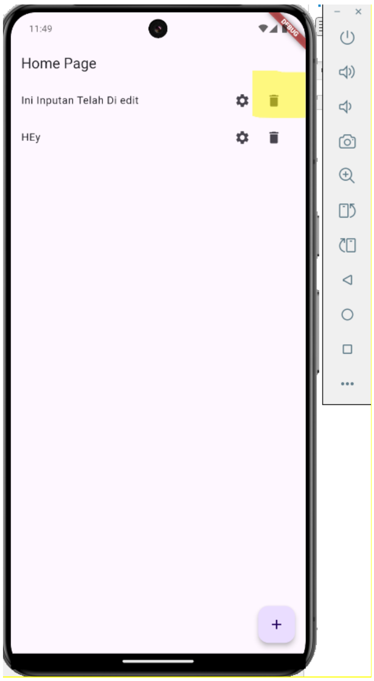


Home Page

Ini Inputan Telah Di edit

HEy

d. Delete



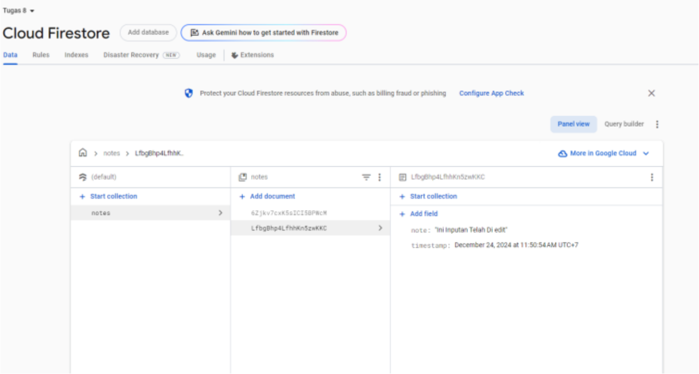
Home Page

Ini Inputan Telah Di edit

HEy

5. Firestore data.

a.



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

notes > LfbgBhp4LfhhK..

(default)

notes

notes

6ZjkvZcxK5sICI5BPWcM

LfbgBhp4LfhhKn5zwKKC

LfbgBhp4LfhhKn5zwKKC

note: "Ini Inputan Telah Di edit"

timestamp: December 24, 2024 at 11:50:54 AM UTC+7