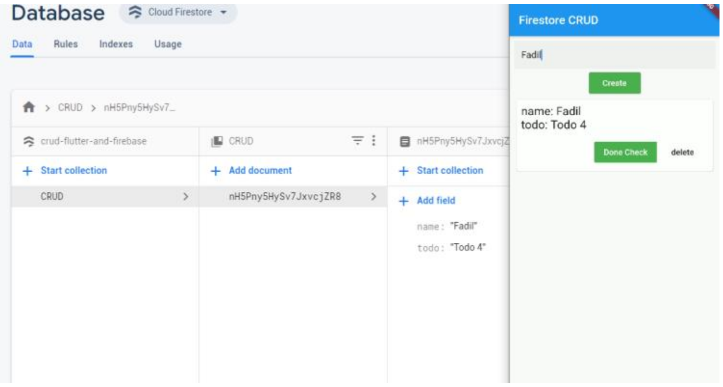
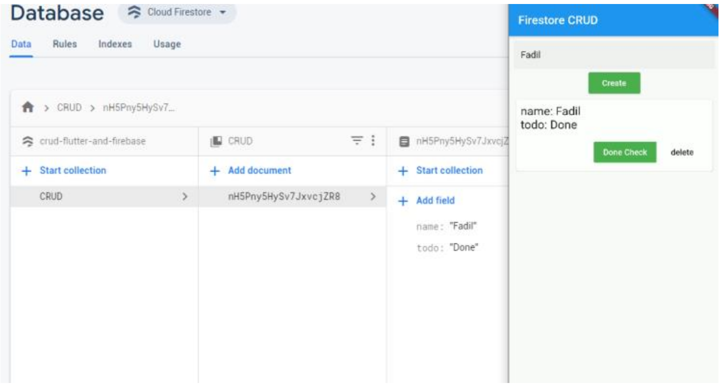
Membuat Project To-Do List CRUD Dengan Flutter dan Firestore



name: "Fadil"
todo: "Todo 4"

Fadil

name: Fadil
todo: Todo 4



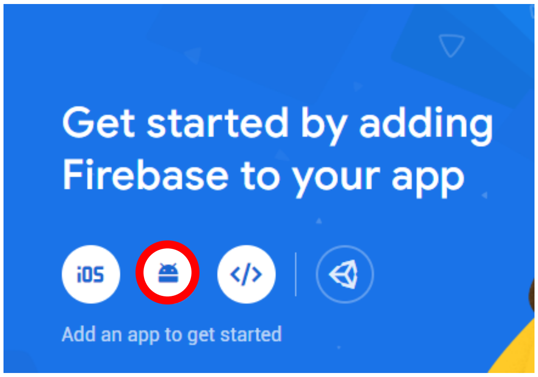
name: "Fadil"
todo: "Done"

Fadil

name: Fadil
todo: Done

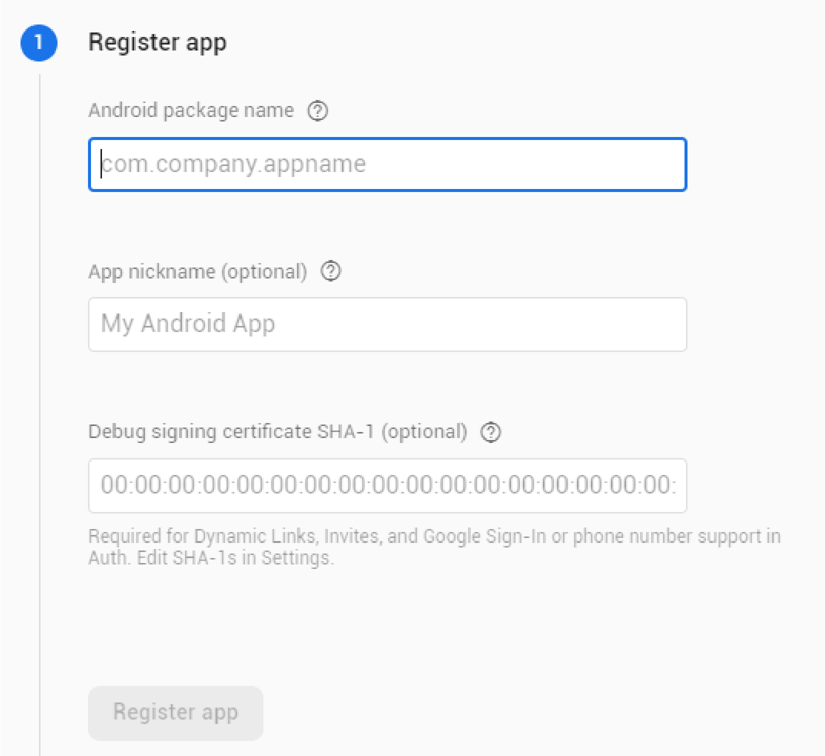
A. Membuat Project di Firebase

1. Buka halaman https://console.firebase.google.com/ dan login menggunakan akun Google Anda. Buatlah project baru dan isi form yang ditampilkan mengenai project Anda
2. Tekan tombol Android
3. Isi form berikut dengan data dari project **Flutter** Anda



Get started by adding Firebase to your app

Add an app to get started



1 Register app

Android package name ?

com.company.appname

App nickname (optional) ?

My Android App

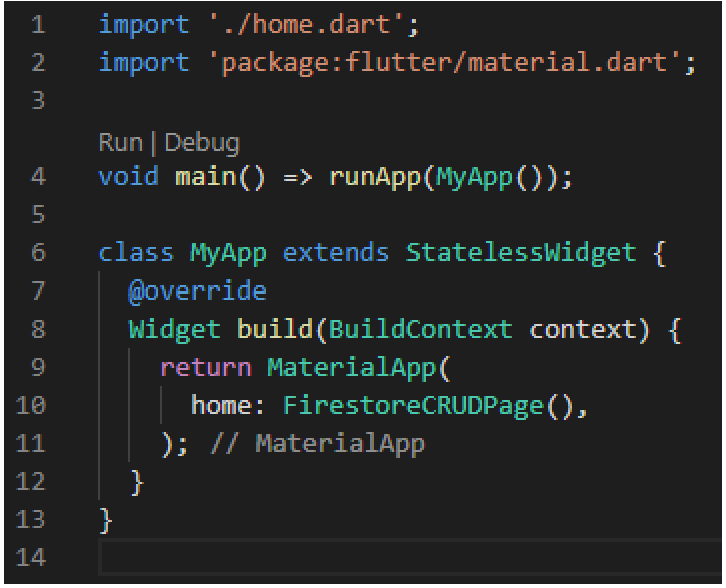
Debug signing certificate SHA-1 (optional) ?

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

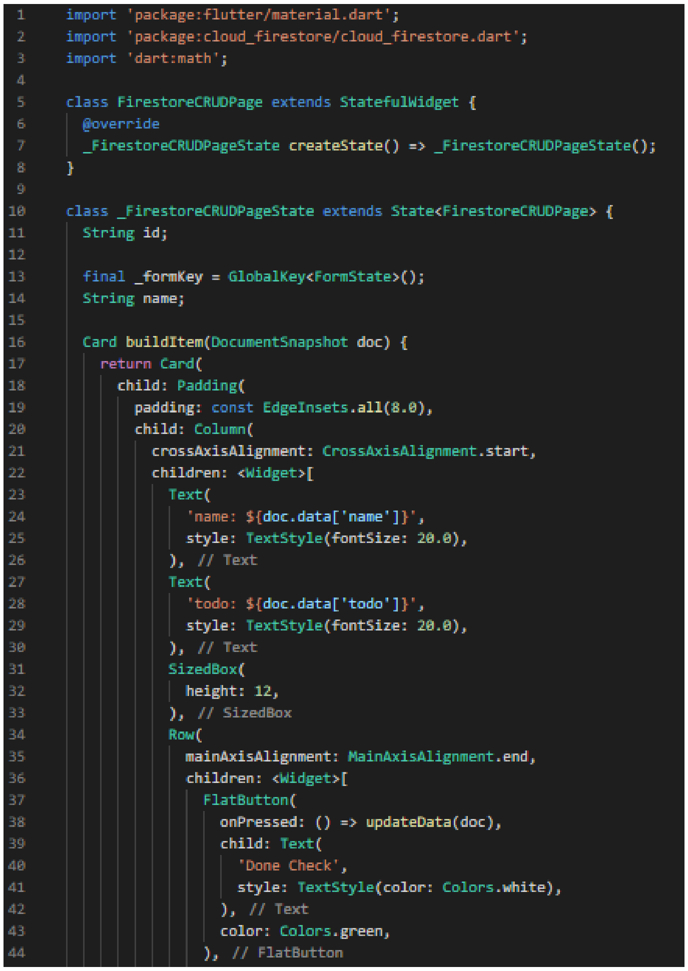
- Android package name bisa di cek di **<project>/Android/app/build.gradle**
- Certificate SHA-1 dapat dicek dengan langkah-langkah berikut:
  1) https://developers.google.com/android/guides/client-auth
  2) https://stackoverflow.com/questions/51845559/generate-sha-1-for-flutter-app

4. Download **google-services.json** dan letakkan di **<project>/Android/app/**
5. Tambahkan konfigurasi untuk Firebase SDK sesuai petunjuk
6. Jalankan kembali Aplikasi Anda dan pastikan sudah tersambung dengan Firebase.
7. Code
   a. Main

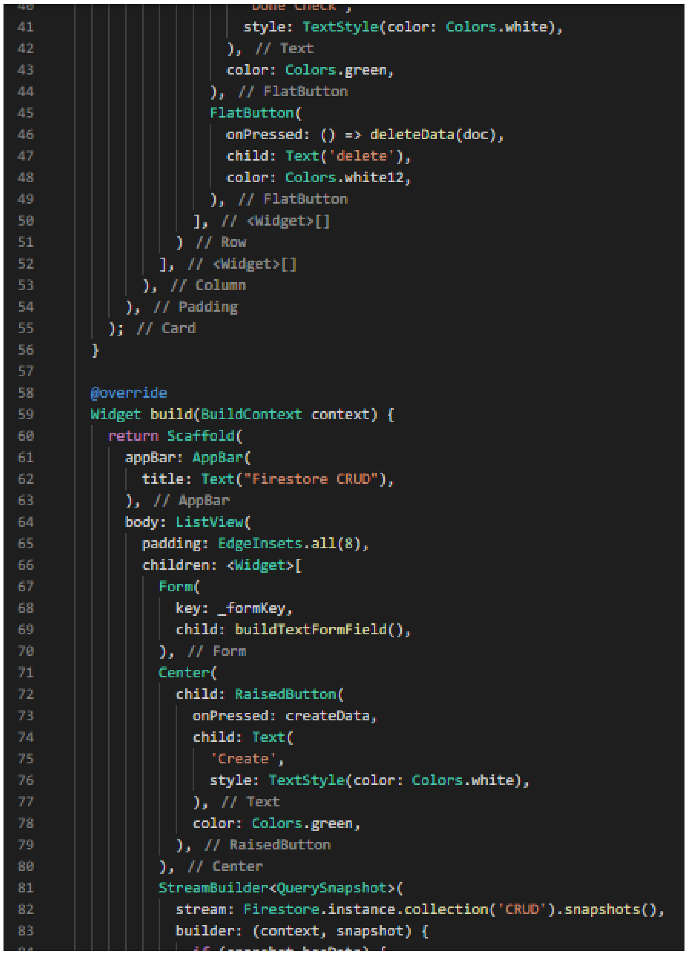


```dart
import './home.dart';
import 'package:flutter/material.dart';

Run | Debug
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: FirestoreCRUDPage(),
    ); // MaterialApp
  }
}
```

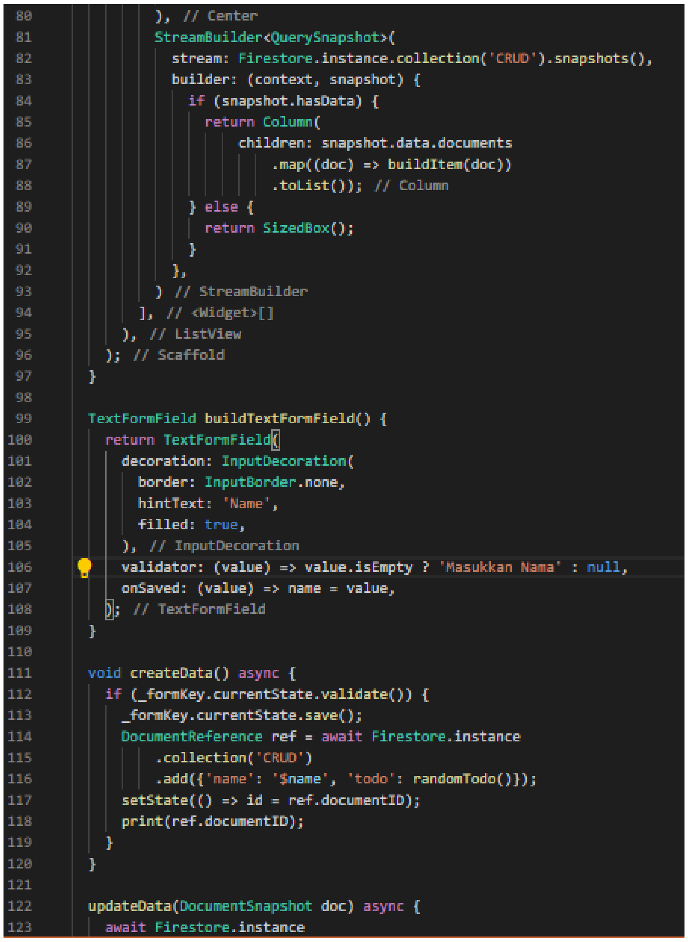
b. Home



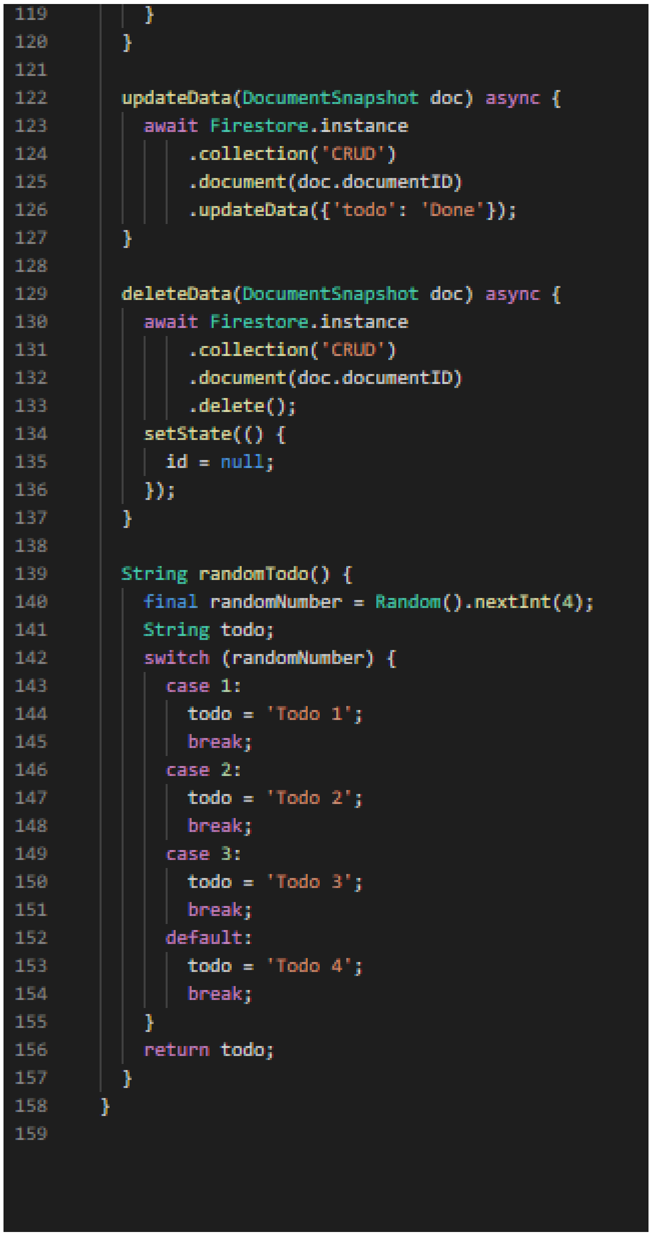
```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'dart:math';

class FirestoreCRUDPage extends StatefulWidget {
  @override
  _FirestoreCRUDPageState createState() => _FirestoreCRUDPageState();
}

class _FirestoreCRUDPageState extends State<FirestoreCRUDPage> {
  String id;

  final _formKey = GlobalKey<FormState>();
  String name;

  Card buildItem(DocumentSnapshot doc) {
    return Card(
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            Text(
              'name: ${doc.data['name']}',
              style: TextStyle(fontSize: 20.0),
            ), // Text
            Text(
              'todo: ${doc.data['todo']}',
              style: TextStyle(fontSize: 20.0),
            ), // Text
            SizedBox(
              height: 12,
            ), // SizedBox
            Row(
              mainAxisAlignment: MainAxisAlignment.end,
              children: <Widget>[
                FlatButton(
                  onPressed: () => updateData(doc),
                  child: Text(
                    'Done Check',
                    style: TextStyle(color: Colors.white),
                  ), // Text
                  color: Colors.green,
                ), // FlatButton
```



```
                    'Done Check',
                    style: TextStyle(color: Colors.white),
                  ), // Text
                  color: Colors.green,
                ), // FlatButton
                FlatButton(
                  onPressed: () => deleteData(doc),
                  child: Text('delete'),
                  color: Colors.white12,
                ), // FlatButton
              ], // <Widget>[]
            ) // Row
          ], // <Widget>[]
        ), // Column
      ), // Padding
    ); // Card
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Firestore CRUD"),
      ), // AppBar
      body: ListView(
        padding: EdgeInsets.all(8),
        children: <Widget>[
          Form(
            key: _formKey,
            child: buildTextFormField(),
          ), // Form
          Center(
            child: RaisedButton(
              onPressed: createData,
              child: Text(
                'Create',
                style: TextStyle(color: Colors.white),
              ), // Text
              color: Colors.green,
            ), // RaisedButton
          ), // Center
          StreamBuilder<QuerySnapshot>(
            stream: Firestore.instance.collection('CRUD').snapshots(),
            builder: (context, snapshot) {
              if (snapshot.hasData) {
```



```
          ), // Center
          StreamBuilder<QuerySnapshot>(
            stream: Firestore.instance.collection('CRUD').snapshots(),
            builder: (context, snapshot) {
              if (snapshot.hasData) {
                return Column(
                    children: snapshot.data.documents
                        .map((doc) => buildItem(doc))
                        .toList()); // Column
              } else {
                return SizedBox();
              }
            },
          ) // StreamBuilder
        ], // <Widget>[]
      ), // ListView
    ); // Scaffold
  }

  TextFormField buildTextFormField() {
    return TextFormField(
      decoration: InputDecoration(
        border: InputBorder.none,
        hintText: 'Name',
        filled: true,
      ), // InputDecoration
      validator: (value) => value.isEmpty ? 'Masukkan Nama' : null,
      onSaved: (value) => name = value,
    ); // TextFormField
  }

  void createData() async {
    if (_formKey.currentState.validate()) {
      _formKey.currentState.save();
      DocumentReference ref = await Firestore.instance
          .collection('CRUD')
          .add({'name': '$name', 'todo': randomTodo()});
      setState(() => id = ref.documentID);
      print(ref.documentID);
    }
  }

  updateData(DocumentSnapshot doc) async {
    await Firestore.instance
```



```
    }
  }

  updateData(DocumentSnapshot doc) async {
    await Firestore.instance
        .collection('CRUD')
        .document(doc.documentID)
        .updateData({'todo': 'Done'});
  }

  deleteData(DocumentSnapshot doc) async {
    await Firestore.instance
        .collection('CRUD')
        .document(doc.documentID)
        .delete();
    setState(() {
      id = null;
    });
  }

  String randomTodo() {
    final randomNumber = Random().nextInt(4);
    String todo;
    switch (randomNumber) {
      case 1:
        todo = 'Todo 1';
        break;
      case 2:
        todo = 'Todo 2';
        break;
      case 3:
        todo = 'Todo 3';
        break;
      default:
        todo = 'Todo 4';
        break;
    }
    return todo;
  }
}
```