

Smart, Creative and Entreprenurial

CIM430 - PEMROGRAMAN MOBILE

Dosen Pengampu: 7174 – Sawali Wahyu, S.Kom, M.Kom

PERTEMUAN 8

INTRODUCTION
FIREBASE REALTIME DATABASE



LEARNING OUTCOME

- 1. Mahasiswa Mampu memahami konsep dasar Firebase Realtime Database secara model dan arsitektur.
- 2. Mahasiswa Mampu Membuat Aplikasi Android Sederhana untuk menampilkan CRUD (Create, Read, Update, Delete) dengan menerapkan Firebase Realtime Database



What Is Firebase Realtime Database?

Firebase adalah Cloud Service Provider dan Backend as a Service yang dimiliki oleh google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah dalam pengembangan aplikasi mobile maupun web. Firebase memiliki banyak SDK yang memungkinkan untuk mengintegrasikan layanan dengan Android, los, Javascript, C++, hingga Unity.



Gambar dibawah ini merupakan fitur-fitur yang ada dalam firebase. terdapat fitur untuk Develop, Grow, dan Earn.





Kemampuan Firebase:

- 1. Realtime yaitu menggunakan sinkronisasi data—setiap kali data berubah, semua perangkat yang terhubung akan menerima update dalam waktu milidetik.
- 2. Offline yaitu Aplikasi Firebase tetap responsif saat offline dan melakukan sinkronisasi dengan status server saat ini.
- 3. Dapat Diakses dari Perangkat Klien yaitu Firebase Realtime Database dapat diakses secara langsung dari perangkat seluler atau browser web;



Jenis Layanan Lain Firebase

- 1. Firebase Remot Config menyimpan key-value pair yang ditentukan developer untuk mengubah perilaku dan penampilan aplikasi tanpa mewajibkan pengguna untuk mendownload update.
- 2. Firebase Hosting menghosting HTML, CSS, dan JavaScript untuk situs Anda, serta aset lain yang disediakan developer, seperti gambar, font, dan ikon.
- 3. Cloud Storage menyimpan file seperti gambar, video, dan audio serta konten buatan pengguna lainnya.



Cloud Firestore dan Realtime Database

menggunakan kedua database ini dalam aplikasi atau project Firebase yang sama. Kedua database NoSQL ini dapat menyimpan jenis data yang sama dan library klien bekerja dengan cara yang sama.



KONFIGURASI FIREBASE REALTIME DATABASE

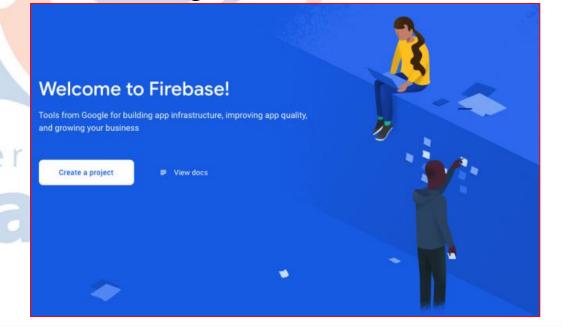


Membuat akun Firebase

Untuk dapat memiliki database Firestore, Anda memerlukan akun Firebase yang dapat anda buat pada link https://firebase.google.com/.

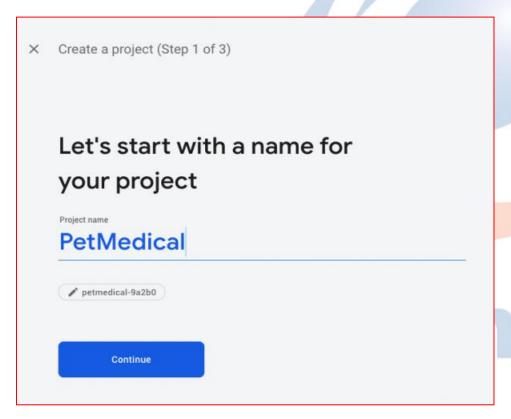
Pada halaman Selamat Datang di Firebase, klik tombol

Create Project.





Kemudian, masukkan nama proyek misal: PetMedical. Kemudian tekan tombol Continue.

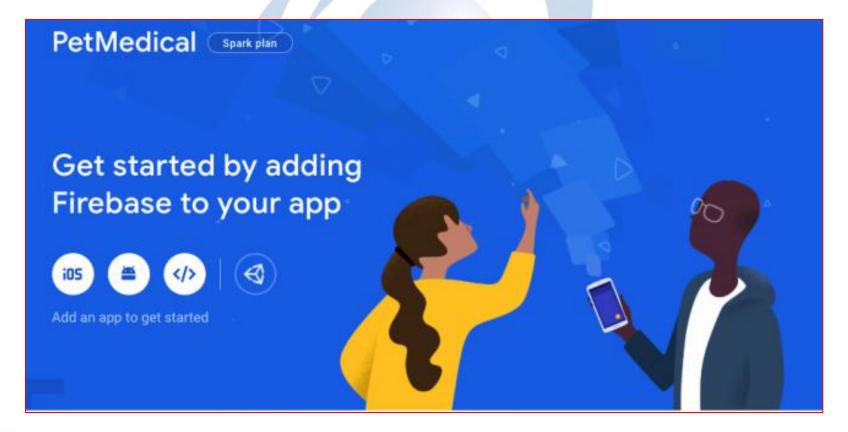


Di halaman berikutnya, klik tombol untuk menonaktifkan Analytics karena Anda tidak akan menggunakannya. Kemudian, klik Create Project.

G	oogle Analytics		
fo	or your Firebase pr	oje	ect
Cor	ogle Analytics is a free and unlimited analyt orting, and more in Firebase Crashlytics, Cl offig, A/B Testing, Predictions, and Cloud Fu gle Analytics enables:	oud Me	ssaging, In-App Messaging, Remote
	A/B-testing ③	×	Crash-free-users ③
×	User-segmentation & targeting across Firebase products ① Predicting user-behavior ①	×	Event-based-Cloud-Functions triggers ① Free unlimited-reporting ①
×			



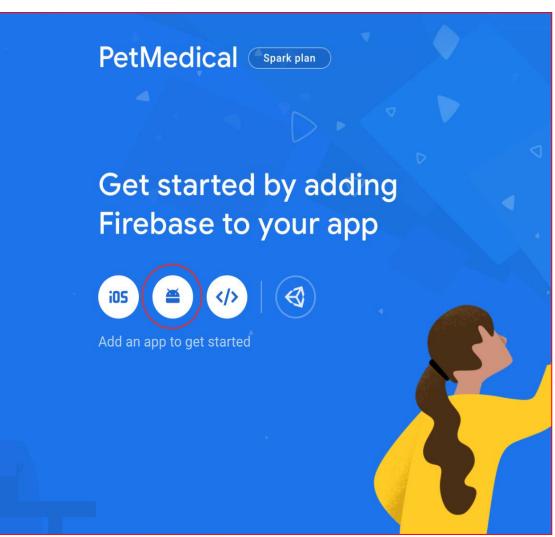
Setelah project selesai dibuat, tekan tombol continue untuk menuju pada halaman Getting Started seperti pada gambar berikut.





Kemudian klik tombol android untuk memulai proses integrasi Firebase dengan project android yang akan anda buat.







Read the Get Started Guide for Android



Anda akan melihat form untuk mendaftarkan project android anda. Disini dicontohkan menginputkan com.raywenderlich.pet_m edical pada Android package name. kemudian klik Register App:

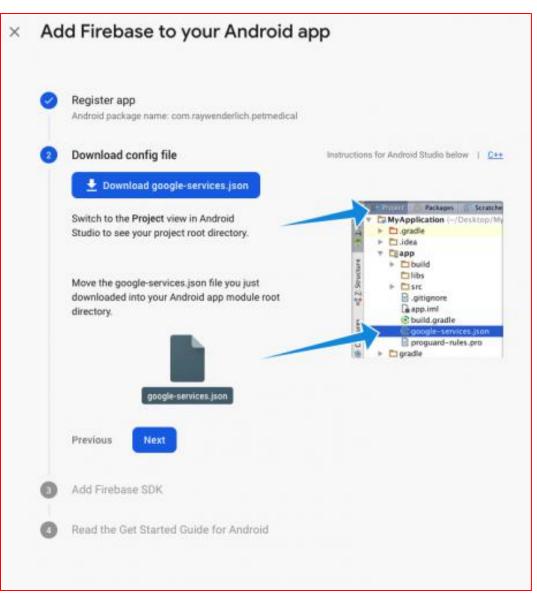




Smart, Creative and Entreprenurial

Remudian klik tombol Download google-services.json. Kemudian pindahkan file tersebut pada folder android app.







Kemudian pada android studio, buka folder Android dan buka build.gradle. Lalu tambahkan c l a s s p a t h 'com.google.gms:google-services:4.3.3' diakhir daftar classpath.

Kemudian buka build.gradle pada app-level dan tambahkan apply plugin: 'com.google.gms.google-services' dipaling bawah.

Perhatikan gambar disamping

Esa

```
Add Firebase SDK
                                                    Instructions for Gradle | Unity C++
The Google services plugin for Gradle [2] loads the google-services. I son file you just
downloaded. Modify your build gradle files to use the plugin.
Project-level build.gradle (<project>/build.gradle):
  buildscript (
    repositories
       // Check that you have the following line (if not, add it):
       google() // Google's Mayer repository
     dependencies (
       // Add this line
       classpath com.google.gms:google-services:4.3.2'
  allprojects (
    repositories -
       // Check that you have the following line (if not, add it):
       google() // Google's Maven repository
App-level build.gradle (<project>/<app-module>/build.gradle):
  apply plugin: 'com.android.application'
  dependencies {
    // add SDKs for desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-librarie
  // Add to the bottom of the file
```

apply plugin: 'com.google.gms.google-services'

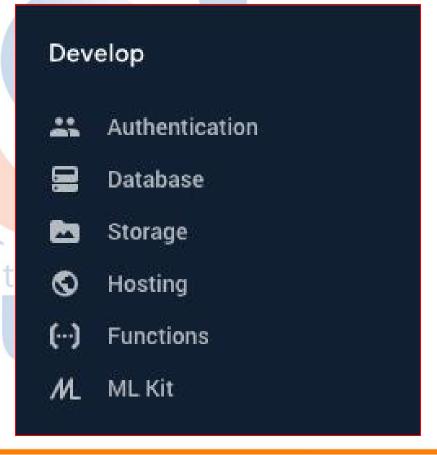
MEMBUAT CRUD PADA FIREBASE REALTIME DATABASE



Membuat database Firestore

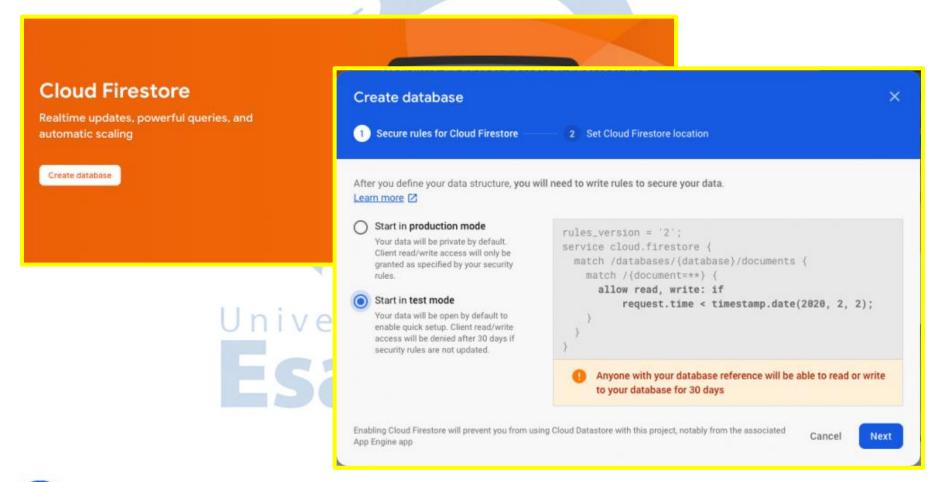
Pada Console Firebase pilih opsi Database dibawah

Develop menu:



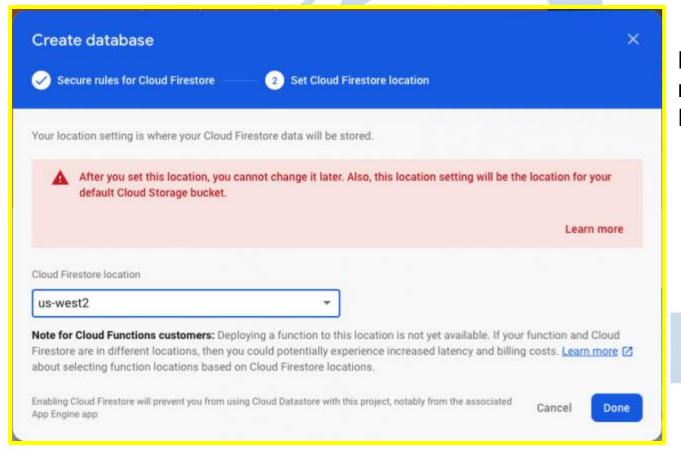


Kemudian klik tombol Create Database dan pilih Start in test mode, ini akan menonaktifkan keamanan sehingga anda dapat dengan mudah melakukan uji coba database yang anda buat.





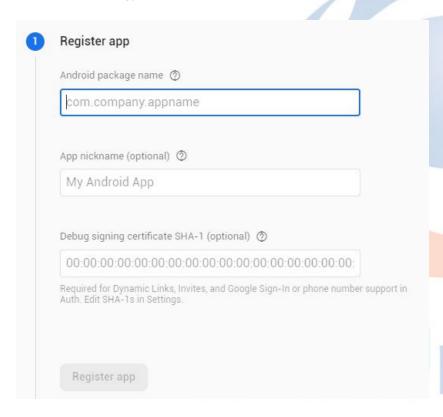
Ketika project anda sudah masuk dalam status production, maka ubahlah pengaturan keamanan tersebut ke production mode dan tambahkan pengaturan keamanan. Kemudian klik next dan pilih Firestore Location dan klik Done.



Dan sudah berhasil membuat database Firestore pertama.



Isi form berikut dengan data dari project **Flutter** Anda



- 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) <u>https://stackoverflow.com/questions/51845559/generate-sha-1-for-flutter-app</u>



- Download google-services.json dan letakkan di /Android/app/
- Tambahkan konfigurasi untuk Firebase SDK sesuai petunjuk
- Jalankan kembali Aplikasi Anda dan pastikan sudah tersambung dengan Firebase.



Code Program 1:

```
import './home.dart';
     import 'package:flutter/material.dart';
     Run | Debug
     void main() => runApp(MyApp());
     class MyApp extends StatelessWidget {
       @override
       Widget build(BuildContext context) {
8
         return MaterialApp(
 9
           home: FirestoreCRUDPage(),
10
         ); // MaterialApp
13
14
```



Code Program Tampilan Home:

```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'dart:math';
class FirestoreCRUDPage extends StatefulWidget {
 _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),
           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
```

```
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
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) {
```



```
), // 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>[]
 ); // 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
```

```
. doc umen- (dcc . dccument £U)
       . doc umen- (dcE . dc Eument EU)
       1d =
randolumber = !:'.-c ().nextIn-(A);

    - - to do j
```



Output program:

