Aplikasi Android Monitoring Lingkungan

dan

Kontrol Lampu LED

1. Tujuan

Praktikum ini bertujuan untuk mengendalikan status LED pada perangkat NodeMCU menggunakan Firebase sebagai database untuk menyimpan status LED, serta menghubungkannya dengan aplikasi Flutter untuk memvisualisasikan dan mengontrol status LED secara real-time.

2. Alat dan Bahan

- **Perangkat keras**:
    - NodeMCU ESP32
    - DHT11 Sensor (untuk suhu dan kelembapan)
    - LED
    - Kabel jumper
- **Perangkat lunak**:
    - Arduino IDE
    - Firebase Console (untuk pembuatan database dan pengaturan API)
    - Flutter (untuk aplikasi Android)

3. Langkah-langkah Praktikum

3.1. Membuat Database di Firebase

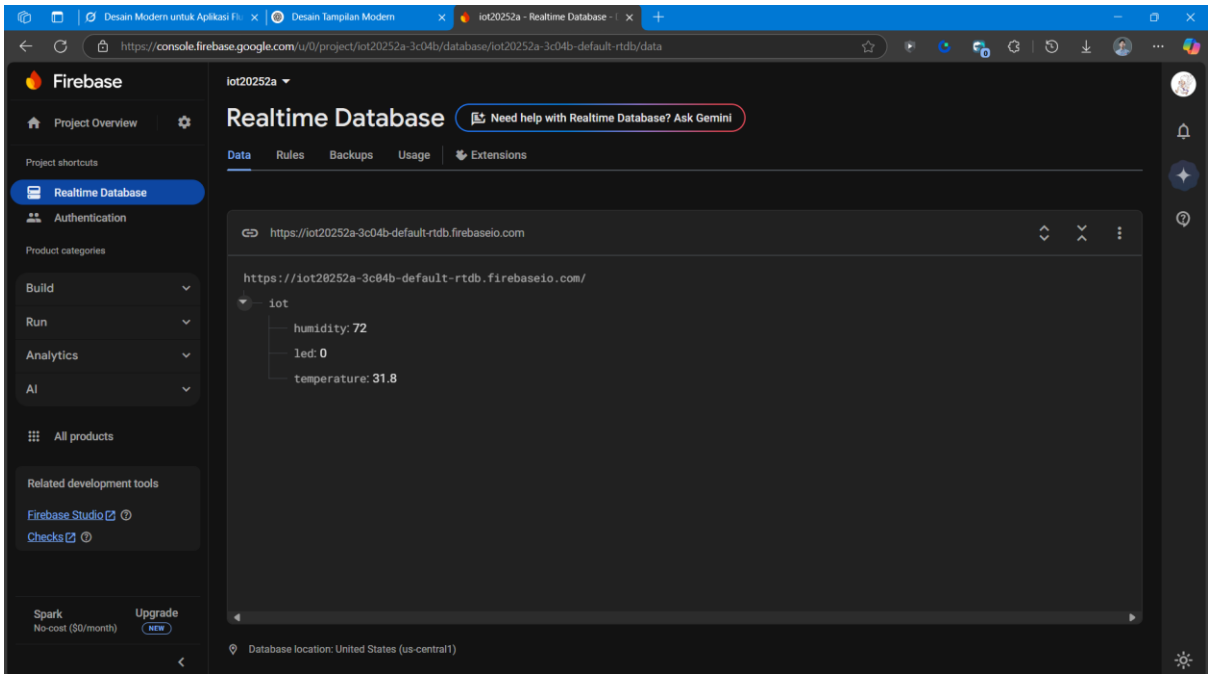
1. **Login ke Firebase Console** dan buat proyek baru.
2. **Buat Realtime Database** dan setel struktur data seperti yang berikut:

json

```
{
  "iot": {
    "humidity": 72,
    "led": 0,
    "temperature": 31.8
```

}

}



3.2. Koding di Arduino IDE untuk NodeMCU

Kode berikut digunakan untuk menghubungkan NodeMCU ke Firebase dan membaca data dari sensor DHT11 serta mengontrol status LED berdasarkan nilai yang ada di Firebase.

```
#include <Arduino.h>
#include <WiFi.h>
#include <FirebaseESP32.h>
#include <DHT.h>
#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>

#define WIFI_SSID "xrp"
#define WIFI_PASSWORD "mstahulhaq"
#define API_KEY "AIzaSyC1qq_11HfZmqkuUaug971X8vT-6m5x5tw"
#define DATABASE_URL "https://iot20252a-3c04b-default-rtdb.firebaseio.com/"
#define USER_EMAIL "ulhaq@gmail.com"
#define USER_PASSWORD "ulhaq123"
#define DHTPIN 4
#define DHTTYPE DHT11
#define LED_PIN 23

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
DHT dht(DHTPIN, DHTTYPE);

float temperature = 0;
```

```
float humidity = 0;
bool ledState = false;

void setup() {
  Serial.begin(115200);
  pinMode(LED_PIN, OUTPUT);
  digitalWrite(LED_PIN, LOW);
  dht.begin();

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(300);
    Serial.print(".");
  }
  Serial.println("WiFi connected!");

  config.api_key = API_KEY;
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;
  config.database_url = DATABASE_URL;
  Firebase.begin(&config, &auth);
}

bool readSensorData() {
  humidity = dht.readHumidity();
  temperature = dht.readTemperature();
  return !isnan(humidity) && !isnan(temperature);
}

void sendDataToFirebase() {
  Firebase.setFloat(fbdo, "/iot/temperature", temperature);
  Firebase.setFloat(fbdo, "/iot/humidity", humidity);
}

void checkForLedCommand() {
  if (Firebase.getBool(fbdo, "/iot/led")) {
    bool newLedState = fbdo.to<bool>();
    if (newLedState != ledState) {
      ledState = newLedState;
      digitalWrite(LED_PIN, ledState ? HIGH : LOW);
    }
  }
}

void loop() {
  if (Firebase.ready()) {
    if (millis() - sendDataPrevMillis > 15000) {
      sendDataPrevMillis = millis();
```

```
    if (readSensorData()) {
      sendDataToFirebase();
    }
  }
  checkForLedCommand();
 }
}
```

3.3. Menghubungkan Firebase dengan Aplikasi Flutter

Setelah mengonfigurasi Firebase, ikuti langkah-langkah berikut untuk menghubungkan Firebase dengan Flutter:

1. Jalankan **CLI Firebase** untuk mengonfigurasi Firebase dengan Flutter.
2. Import file firebase_options.dart ke dalam proyek Flutter.
3. Kode Flutter untuk mengontrol dan memantau status LED secara real-time adalah sebagai berikut:

dart

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

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

class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Flutter Firebase Real-Time',
    theme: ThemeData(
      primarySwatch: Colors.blue,
      textTheme: GoogleFonts.poppinsTextTheme(),
      scaffoldBackgroundColor: Colors.blue.shade50,
    ),
    home: MyHomePage(),
  );
 }
}
```

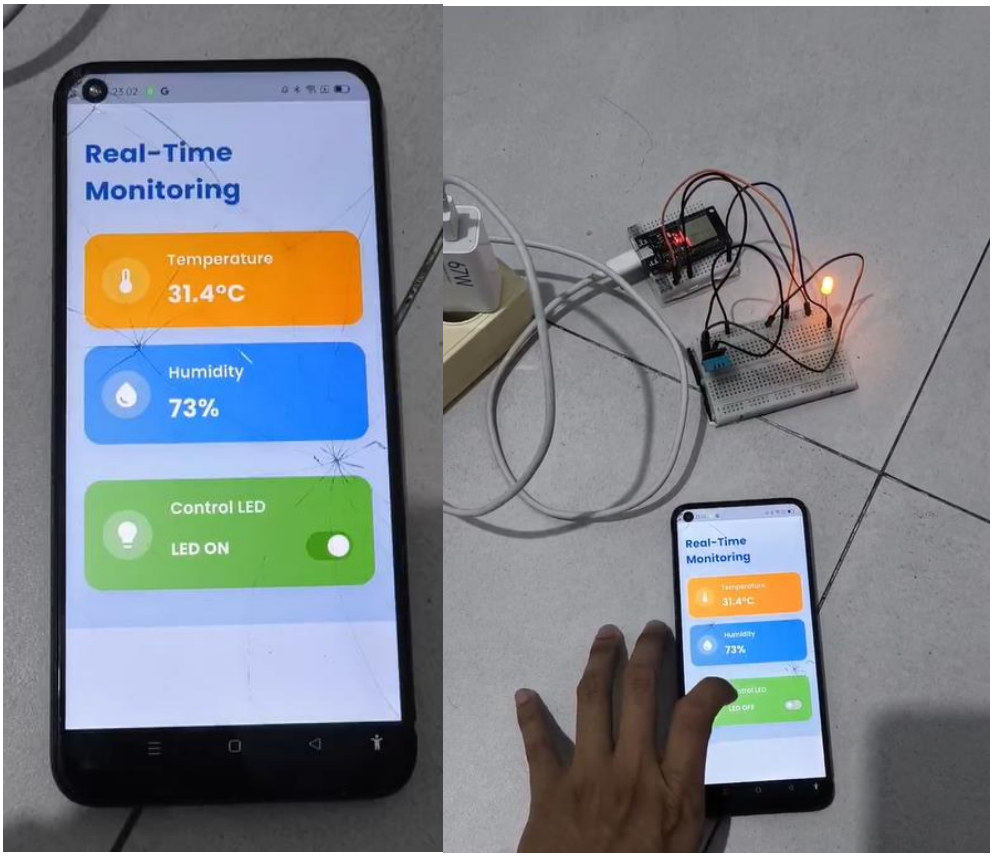
```
class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  final DatabaseReference _dbRef = FirebaseDatabase.instance.ref();
  late DatabaseReference _temperatureRef;
  late DatabaseReference _humidityRef;
  late DatabaseReference _ledControlRef;

  String _temperature = "Loading...";
  String _humidity = "Loading...";
  bool _ledState = false;

  @override
  void initState() {
    super.initState();
    _temperatureRef = _dbRef.child("iot/temperature");
    _humidityRef = _dbRef.child("iot/humidity");
    _ledControlRef = _dbRef.child("iot/led");

    _temperatureRef.onValue.listen((event) {
      setState(() {
        _temperature = event.snapshot.value.toString();
      });
    });

    _humidityRef.onValue.listen((event) {
      setState(() {
        _humidity = event.snapshot.value.toString();
      });
    });

    _ledControlRef.onValue.listen((event) {
      setState(() {
        _ledState = event.snapshot.value == 1;
      });
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: Column(
          children: <Widget>[
            Text("Temperature: $_temperature°C"),
```

```
          Text("Humidity: $_humidity%"),
          Switch(
            value: _ledState,
            onChanged: (bool value) {
              setState(() {
                _ledState = value;
              });
              _ledControlRef.set(_ledState ? 1 : 0);
            },
          ),
          Text(_ledState ? "LED ON" : "LED OFF"),
        ],
      ),
    ),
  );
 }
}
```

4. Hasil

Dengan implementasi kode di atas, Anda dapat:

1. Mengontrol status LED menggunakan aplikasi Flutter dengan menekan tombol switch.
2. Melihat pembaruan suhu dan kelembapan secara real-time pada aplikasi Android yang terhubung ke Firebase.
3. Mengirimkan pembaruan suhu dan kelembapan dari sensor DHT11 ke Firebase secara otomatis setiap 15 detik
4. **Link video :** https://youtube.com/shorts/T19yunKEK54?si=KUtI-e0ZJpA8Ma9t



5. Kesimpulan

Praktikum ini berhasil melakukan pengendalian status LED menggunakan Firebase dan Flutter, dengan data suhu dan kelembapan yang diperbarui secara real-time. Firebase berfungsi dengan baik sebagai medium komunikasi antara aplikasi Flutter dan perangkat NodeMCU, memungkinkan pengendalian perangkat keras IoT melalui aplikasi Android.