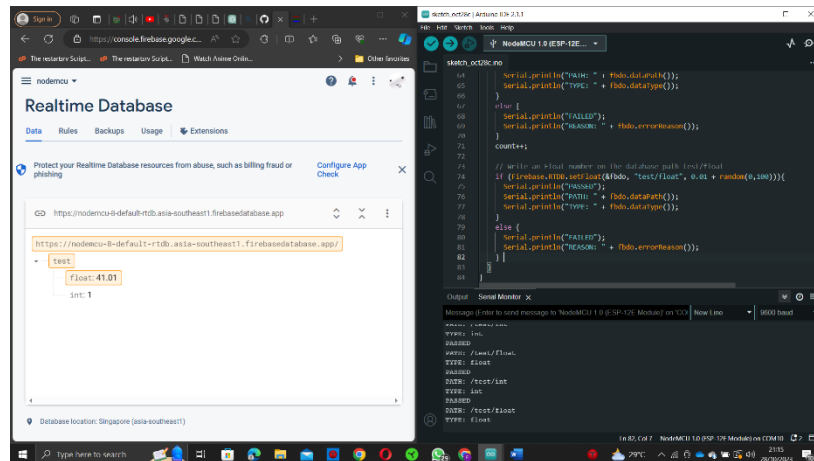


Nama : Farrel Culbert Early Karendi

NIM : 1103210027

Percobaan Pertama



Header program pada percobaan pertama diisi dengan deklarasi library yg akan digunakan, seperti Wifi untuk ESP8266 dan juga library untuk firebase. Pada bagian ini juga terdapat program untuk memasukkan SSID dan password wifi yang akan digunakan serta url dan API untuk menyambungkan ESP8266 dengan RTDB.

```
#include <Arduino.h>
#ifdef ESP32
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>
//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"
// Insert your network credentials
#define WIFI_SSID "Kontrakan ITTS"
#define WIFI_PASSWORD "kontrakan_itts"
// Insert Firebase project API Key
#define API_KEY "AIzaSyDVvRVAum2UC5tQ40-zGyzfWrCI0jbhrhA"
// Insert RTDB URL
#define DATABASE_URL "https://nodemcu-8-default-rtddb.asia-southeast1.firebaseio.com/"
//Define Firebase Data object
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
unsigned long sendDataPrevMillis = 0;
int count = 0;
bool signupOK = false;
```

Pada void setup terdapat program untuk menyambungkan koneksi wifi antara ESP8266 dengan wifi yang akan digunakan dan menampilkan IP yang terhubung. Dan pada program ini juga terdapat program untuk meminta request untuk terhubung ke RTDB berdasarkan url dan API yang sudah dimasukan di bagian header.

```
void setup() {
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  /* Assign the api key (required) */
  config.api_key = API_KEY;
  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;
  /* Sign up */
  if (Firebase.signUp(&config, &auth, "", "")){
    Serial.println("ok");
    signupOK = true;
  }
  else{
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }

  /* Assign the callback function for the long running token generation
  task */
  config.token_status_callback = tokenStatusCallback; //see
  addons/TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}
```

Pada void loop terdapat program untuk mengirimkan data yang ingin dikirim ke RTDB, pada baris awal void loop dilakukan pengecekan kondisi yang harus terpenuhi agar dapat dikirim ke RTDB. Lalu, dibawah program tersebut terdapat data apa yang ingin dikirim ke RTDB dan data tersebut dimasukan kedalam path atau file yang sudah ditentukan “test/int” atau “test/float” ke RTDB.

```
void loop() {
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis >
  15000 || sendDataPrevMillis == 0)){
    sendDataPrevMillis = millis();
    // Write an Int number on the database path test/int
    if (Firebase.RTDB.setInt(&fbdo, "test/int", count)){
      Serial.println("PASSED");
      Serial.println("PATH: " + fbdo.dataPath());
      Serial.println("TYPE: " + fbdo.dataType());
    }
  }
}
```

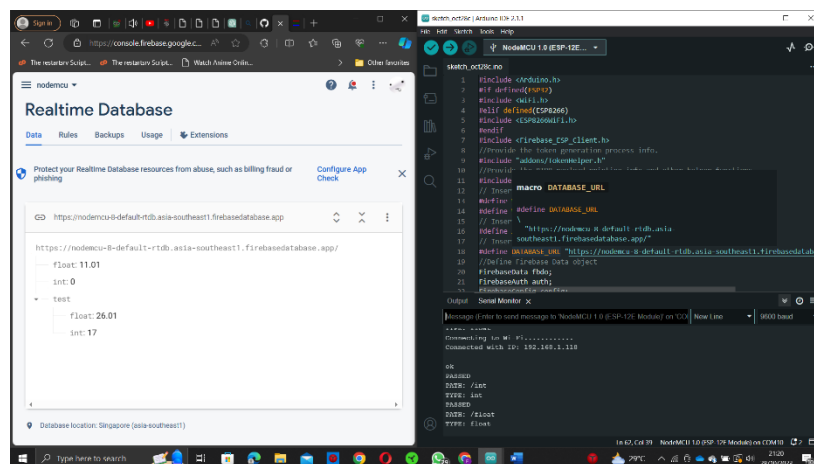
```

else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}
count++;

// Write an Float number on the database path test/float
if (Firebase.RTDB.setFloat(&fbdo, "test/float", 0.01 +
random(0,100))) {
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
}
else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}
}
}
}

```

Percobaan Kedua



Header program pada percobaan kedua diisi dengan deklarasi library yg akan digunakan, seperti Wifi untuk ESP8266 dan juga library untuk firebase. Pada bagian ini juga terdapat program untuk memasukkan SSID dan password wifi yang akan digunakan serta url dan API untuk menyambungkan ESP8266 dengan RTDB.

```

#include <Arduino.h>
#ifdef ESP32
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>
//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"

```

```

// Insert your network credentials
#define WIFI_SSID "Kontrakan ITTS"
#define WIFI_PASSWORD "kontrakan_itts"
// Insert Firebase project API Key
#define API_KEY "AIzaSyDVvRVAum2UC5tQ4O-zGyzfWrCI0jbhrhA"
// Insert RTDB URL
#define DATABASE_URL "https://nodemcu-8-default-rtdb.asia-southeast1.firebaseio.com/"
//Define Firebase Data object
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
unsigned long sendDataPrevMillis = 0;
int count = 0;
bool signupOK = false;

```

Pada void setup terdapat program untuk menyambungkan koneksi wifi antara ESP8266 dengan wifi yang akan digunakan dan menampilkan IP yang terhubung. Dan pada program ini juga terdapat program untuk meminta request untuk terhubung ke RTDB berdasarkan url dan API yang sudah dimasukan di bagian header.

```

void setup() {
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  /* Assign the api key (required) */
  config.api_key = API_KEY;
  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;
  /* Sign up */
  if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.println("ok");
    signupOK = true;
  }
  else {
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }

  /* Assign the callback function for the long running token generation task */
  config.token_status_callback = tokenStatusCallback; //see addons/TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}

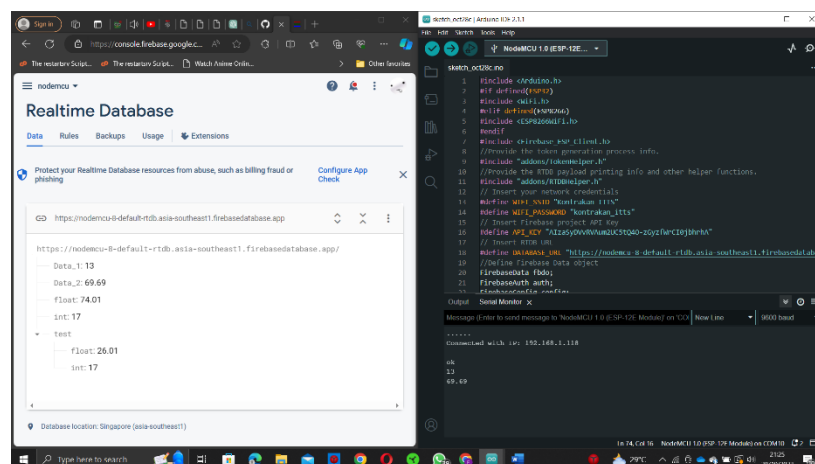
```

Pada void loop terdapat program untuk mengirimkan data yang ingin dikirim ke RTDB, pada baris awal void loop dilakukan pengecekan kondisi yang harus terpenuhi agar dapat dikirim ke RTDB. Lalu, dibawah program tersebut terdapat data apa yang ingin dikirim ke RTDB dan data tersebut dimasukan kedalam path atau file yang sudah ditentukan. Berbeda dengan percobaan pertama yaitu “test/int” atau “test/float”. Pada percobaan kedua ini pathnya diganti hanya ”int” dan ”float” artinya file atau data tertulis sebagai parent.

```
void loop() {
    if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis >
15000 || sendDataPrevMillis == 0)){
        sendDataPrevMillis = millis();
        // Write an Int number on the database path test/int
        if (Firebase.RTDB.setInt(&fbdo, "/int", count)){
            Serial.println("PASSED");
            Serial.println("PATH: " + fbdo.dataPath());
            Serial.println("TYPE: " + fbdo.dataType());
        }
        else {
            Serial.println("FAILED");
            Serial.println("REASON: " + fbdo.errorReason());
        }
        count++;

        // Write an Float number on the database path test/float
        if (Firebase.RTDB.setFloat(&fbdo, "/float", 0.01 + random(0,100))){
            Serial.println("PASSED");
            Serial.println("PATH: " + fbdo.dataPath());
            Serial.println("TYPE: " + fbdo.dataType());
        }
        else {
            Serial.println("FAILED");
            Serial.println("REASON: " + fbdo.errorReason());
        }
    }
}
```

Percobaan Ketiga



Header program pada percobaan ketiga diisi dengan deklarasi library yg akan digunakan, seperti Wifi untuk ESP8266 dan juga library untuk firebase. Pada bagian ini juga terdapat program untuk

memasukan SSID dan password wifi yang akan digunakan serta url dan API untuk menyambungkan ESP8266 dengan RTDB.

```
#include <Arduino.h>
#include <ESP32>
#include <WiFi.h>
#include <ESP8266>
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"
// Insert your network credentials
#define WIFI_SSID "Kontrakan ITTS"
#define WIFI_PASSWORD "kontrakan_itts"
// Insert Firebase project API Key
#define API_KEY "AIzaSyDVvRVAum2UC5tQ40-zGyzfWrCI0jbhrhA"
// Insert RTDB URL
#define DATABASE_URL "https://nodemcu-8-default-rtdb.asia-southeast1.firebaseio.com/"
//Define Firebase Data object
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
unsigned long sendDataPrevMillis = 0;
int intValue;
float floatValue;
bool signupOK = false;
```

Pada void setup terdapat program untuk menyambungkan koneksi wifi antara ESP8266 dengan wifi yang akan digunakan dan menampilkan IP yang terhubung. Dan pada program ini juga terdapat program untuk meminta request untuk terhubung ke RTDB berdasarkan url dan API yang sudah dimasukan di bagian header.

```
void setup() {
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  /* Assign the api key (required) */
  config.api_key = API_KEY;

  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;
```

```

/* Sign up */
if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.println("ok");
    signupOK = true;
}
else {
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
}

/* Assign the callback function for the long running token generation
task */
config.token_status_callback = tokenStatusCallback; //see
addons/TokenHelper.h

Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);
}

```

Pada percobaan ketiga ini berbeda dengan percobaan pertama dan kedua yang merupakan program untuk mengirim data ke RTDB. Dalam percobaan ketiga ini mengambil data dari RTDB. pada baris awal void loop dilakukan pengecekan kondisi yang harus terpenuhi agar dapat dikirim ke RTDB atau diambil dari RTDB. Pada program if dibawahnya terdapat program untuk mengambil data dari RTDB berdasarkan path atau file yang sudah dibuat pada RTDB seperti pada contoh program mengambil data dari “Data_1” dan “Data_2” lalu data yang diambil ditampilkan di serial monitor.

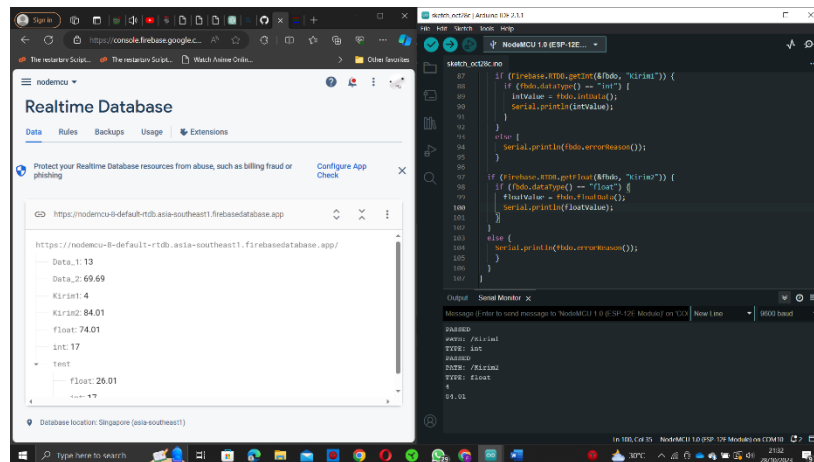
```

void loop() {
    if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis >
15000 || sendDataPrevMillis == 0)) {
        sendDataPrevMillis = millis();
        if (Firebase.RTDB.getInt(&fbdo, "Data_1")) {
            if (fbdo.dataType() == "int") {
                intValue = fbdo.intData();
                Serial.println(intValue);
            }
        }
        else {
            Serial.println(fbdo.errorReason());
        }

        if (Firebase.RTDB.getFloat(&fbdo, "Data_2")) {
            if (fbdo.dataType() == "float") {
                floatValue = fbdo.floatData();
                Serial.println(floatValue);
            }
        }
        else {
            Serial.println(fbdo.errorReason());
        }
    }
}
}

```

Percobaan Keempat



Header program pada percobaan ketiga diisi dengan deklarasi library yg akan digunakan, seperti Wifi untuk ESP8266 dan juga library untuk firebase. Pada bagian ini juga terdapat program untuk memasukkan SSID dan password wifi yang akan digunakan serta url dan API untuk menyambungkan ESP8266 dengan RTDB.

```
#include <Arduino.h>
#ifdef ESP32
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>
int count;

//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"
// Insert your network credentials
#define WIFI_SSID "Kontrakan ITTS"
#define WIFI_PASSWORD "kontrakan_itts"
// Insert Firebase project API Key
#define API_KEY "AIzaSyDVvRVAum2UC5tQ40-zGyzfWwCI0jbhrhA"
// Insert RTDB URL
#define DATABASE_URL "https://nodemcu-8-default-rtdb.asia-southeast1.firebaseio.com/"
//Define Firebase Data object
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
unsigned long sendDataPrevMillis = 0;
int intValue;
float floatValue;
bool signupOK = false;
```

Pada void setup terdapat program untuk menyambungkan koneksi wifi antara ESP8266 dengan wifi yang akan digunakan dan menampilkan IP yang terhubung. Dan pada program ini juga terdapat program untuk meminta request untuk terhubung ke RTDB berdasarkan url dan API yang sudah dimasukan di bagian header.


```

void setup() {
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  /* Assign the api key (required) */
  config.api_key = API_KEY;
  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;
  /* Sign up */
  if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.println("ok");
    signupOK = true;
  }
  else {
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }

  /* Assign the callback function for the long running token generation
  task */
  config.token_status_callback = tokenStatusCallback; //see
  addons/TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}

```

Pada percobaan keempat ini merupakan gabungan dari percobaan kesatu, kedua, dan ketiga. Jadi pada program ini terdapat program untuk mengirimkan data ke RTDB lalu data tersebut di ambil kembali oleh ESP8266 untuk ditampilkan pada serial monitor. "kirim1" mengirimkan data berupa interger dan "kirim2" mengirimkan data berupa float.

```

void loop() {
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis >
  15000 || sendDataPrevMillis == 0)) {
    sendDataPrevMillis = millis();

    if (Firebase.RTDB.setInt(&fbdo, "Kirim1", count)){
      Serial.println("PASSED");
      Serial.println("PATH: " + fbdo.dataPath());
      Serial.println("TYPE: " + fbdo.dataType());
    }
    else {
      Serial.println("FAILED");
      Serial.println("REASON: " + fbdo.errorReason());
    }

    count++;
  }
}

```

```

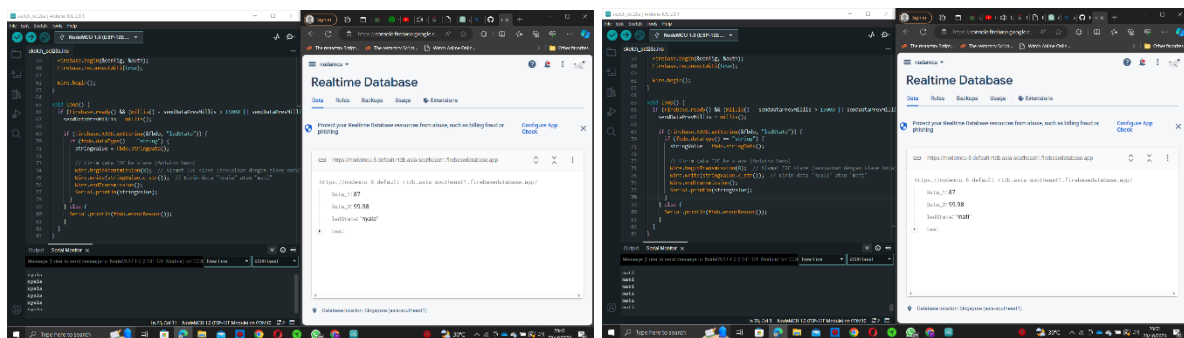
    if (Firebase.RTDB.setFloat(&fbdo, "Kirim2", 0.01 + random(0,100))) {
        Serial.println("PASSED");
        Serial.println("PATH: " + fbdo.dataPath());
        Serial.println("TYPE: " + fbdo.dataType());
    }
    else {
        Serial.println("FAILED");
        Serial.println("REASON: " + fbdo.errorReason());
    }

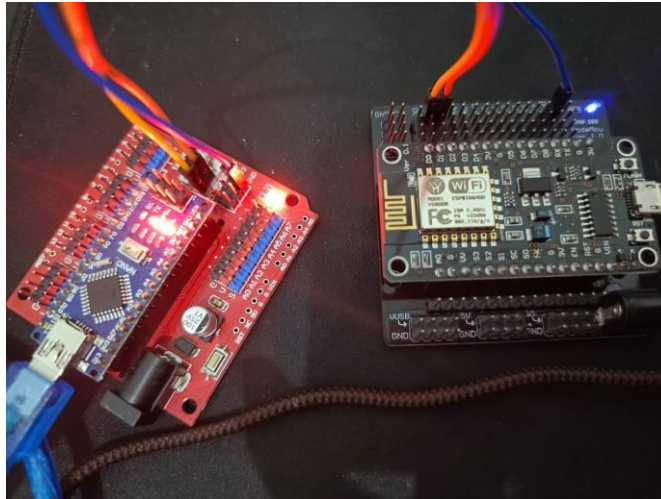
    if (Firebase.RTDB.getInt(&fbdo, "Kirim1")) {
        if (fbdo.dataType() == "int") {
            intValue = fbdo.intData();
            Serial.println(intValue);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getFloat(&fbdo, "Kirim2")) {
        if (fbdo.dataType() == "float") {
            floatValue = fbdo.floatData();
            Serial.println(floatValue);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }
}
}

```

Percobaan Kelima





Program Master

Pada program master terdapat program untuk mengambil data dari RTDB dengan type string dan diambil dari path RTDB "ledState", lalu data string tersebut dikirimkan kepada slave melalui komunikasi I2C dengan alamat (9).

```
#include <Arduino.h>
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Wire.h>
#include <Firebase_ESP_Client.h>
// Provide the token generation process info.
#include "addons/TokenHelper.h"
// Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"
// Insert your network credentials
#define WIFI_SSID "mboh"
#define WIFI_PASSWORD "hanyahuruf"
// Insert Firebase project API Key
#define API_KEY "AIzaSyDVvRVAum2UC5tQ40-zGyzfWrCI0jbhrhA"
// Insert RTDB URL
#define DATABASE_URL "https://nodemcu-8-default-rtdb.asia-southeast1.firebaseio.com/"

// Define Firebase Data object
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
unsigned long sendDataPrevMillis = 0;
String stringValue = "";

void setup() {
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
```

```

    Serial.print(".");
    delay(300);
}
Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();

/* Assign the api key (required) */
config.api_key = API_KEY;

/* Assign the RTDB URL (required) */
config.database_url = DATABASE_URL;

/* Sign up */
if (Firebase.signUp(&config, &auth, "", "")) {
    Serial.println("ok");
} else {
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
}

/* Assign the callback function for the long running token generation
task */
config.token_status_callback = tokenStatusCallback; // see
addons/TokenHelper.h

Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);

Wire.begin();
}

void loop() {
    if (Firebase.ready() && (millis() - sendDataPrevMillis > 15000 ||
sendDataPrevMillis == 0)) {
        sendDataPrevMillis = millis();

        if (Firebase.RTDB.getString(&fbdo, "ledState")) {
            if (fbdo.dataType() == "string") {
                stringValue = fbdo.stringData();

                // Kirim data I2C ke slave (Arduino Nano)
                Wire.beginTransmission(9); // Alamat I2C slave (sesuaikan dengan
slave Anda)
                Wire.write(stringValue.c_str()); // Kirim data "nyala" atau
"mati"
                Wire.endTransmission();
                Serial.println(stringValue);
            }
        } else {
            Serial.println(fbdo.errorReason());
        }
    }
}

```

Program Slave

Setelah data RTDB didapatkan oleh ESP8266, data string tersebut dikirimkan ke slave melalui I2C dengan alamat (9). Karena ingin menyalakan LED_BUILD_IN maka slave diberikan program untuk menyalakan LED_BUILD_IN dengan kondisi ketika data string dikirimkan "nyala" maka LED_BUILD_IN akan menyala dan ketika data string yang dikirim "mati" maka LED_BUILD_IN akan mati.

```
#include <Wire.h>

int ledPin = 13; // Pin LED built-in pada Arduino Nano

void setup() {
  Wire.begin(9); // Menginisialisasi slave address dengan nilai 9
  Wire.onReceive(receiveEvent);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  delay(100);
}

void receiveEvent() {
  String command = "";
  while (Wire.available()) {
    char c = Wire.read();
    command += c;
  }
  if (command == "nyala") {
    digitalWrite(ledPin, HIGH);
  } else if (command == "mati") {
    digitalWrite(ledPin, LOW);
  }
}
```