

LAPORAN TUGAS SISTEM AKUISISI DATA

SMART INCUBATOR



Disusun Oleh:

- Fransiskus Abel (1102200398)
- Firas Maulana (1102204539)

Kata Pengantar

Puji syukur atas kehadiran Tuhan Yang Maha Esa atas segala limpahan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan penyusunan laporan tugas besar ini dengan baik dan selesai tepat waktu. Adapun laporan tugas besar ini ditujukan untuk memenuhi salah satu tugas Mata Kuliah Pemograman Perangkat IOT dengan dosen pengampu yaitu Bapak Muhammad Ridho Rosa

Kami selaku penulis menyadari bahwa penyusunan laporan tugas besar ini masih banyak kekurangannya baik pada teknis penulisan maupun isi laporan, mengingat kemampuan yang dimiliki penulis. Oleh karena itu, kami mengharapkan saran dari dosen pengampu maupun pembaca demi perkembangan positif bagi kami selaku penulis.

Demikian laporan tugas besar ini kami susun, semoga dapat bermanfaat bagi pembaca dan juga bagi kami selaku penulis sendiri. Akhir kata kami ucapkan terima kasih.

Bandung, 6 Januari 2023

DAFTAR ISI

BAB 1 : PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	4
1.3 Tujuan	4
BAB 2 : TINJAUAN PUSTAKA	5
2.1 Komponen yang di gunakan	5
2.1.1 DHT11	5
2.1.2 Relay	7
2.1.3 Lampu Bohlam.....	7
2.1.4 ESP 8266.....	8
2.2 Platform yang di gunakan	8
2.2.1 Firebase	8
2.2.2 Kodular	9
2.2.3 Google Sheets	9
BAB 3 : PROSES AKUISISI DATA	10
3.1 Sistem Akuisisi Data DHT11.....	10
3.2 Sensor Suhu DHT11	10
3.3 Sensor Kelembapan DHT11	12
BAB 4 : PERANCANGAN SISTEM.....	13
4.1 Layout Wiring.....	13
4.2 Source Koding	13
4.3 Tampilan Kodular	21
4.4 Tampilan Google Sheet.....	23
BAB 5 : PENGUJIAN DAN ANALISIS	24
BAB 6 : KESIMPULAN DAN SARAN.....	26

BAB 1 : PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi khususnya pada monitoring suatu sistem, semakin hari semakin menarik untuk dikembangkan. Sistem Monitoring ini dapat memudahkan peran manusia dalam penelitian, penjagaan, dan kegiatan lainnya. Dengan perkembangannya teknologi sekarang kita bisa monitoring dalam jarak jauh dan bisa untuk mengontrol apa yang sedang kita monitoring. Salah satu contohnya adalah *smart incubator*.

Smart incubator erat kaitannya dengan telur, mau itu telur ayam, bebek, dll. Biasanya alat ini dipakai di bidang bisnis ternak ayam. Mengapa demikian? Untuk memaksimalkan kualitas dari telur tersebut teknologi *smart incubator* ini sangat membantu untuk reproduksi ayam dikarenakan hasilnya yg memuaskan, telur lebih cepat menetas dan juga kualitas dari anak ayam bagus.

1.2 Rumusan Masalah

1. Apa itu *Smart Incubator*?
2. Apa saja sensor dan alat-alat lainnya yang di butuhkan?
3. Bagaimana proses akuisisi datanya?
4. Bagaimana cara menyimpan datanya?

1.3 Tujuan

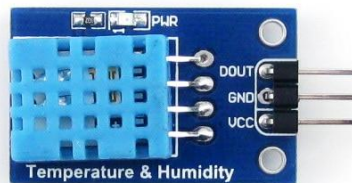
1. Mengetahui apa itu *Smart Incubator*.
2. Mengetahui sensor alat-alat yang di butuhkan.
3. Mengetahui cara proses akuisisi data DHT11.
4. Mengetahui cara menyimpan data.

BAB 2 : TINJAUAN PUSTAKA

Inkubator merupakan sebuah perangkat tertutup yang dapat mengoptimalkan suhu dan kelembapan agar organisme sel dapat berkembang dengan baik. Adapun fungsi-fungsi inkubator yang perlu Anda ketahui adalah sebagai berikut: Inkubator digunakan untuk membudidayakan organisme sel, baik uniseluler maupun multiseluler. Seiring berjalannya waktu inkubator ini berevolusi menjadi smart incubator dimana kita bisa memonitoring perkembangan suhu didalam incubator dan juga kita bisa mengatur berapa suhu incubator yang kita inginkan sesuai kebutuhan.

2.1 Komponen yang di gunakan

2.1.1 DHT11



Sensor DHT merupakan paket sensor yang berfungsi untuk mengukur suhu dan kelembapan udara sekaligus yang di dalamnya terdapat thermistor tipe NTC (*Negative Temperature Coefficient*) untuk mengukur suhu, sebuah sensor kelembapan dengan karakteristik resistif terhadap perubahan kadar air di udara serta terdapat chip yang di dalamnya melakukan beberapa konversi analog ke digital dan

mengeluarkan output dengan format single-wire bi-directional (kabel tunggal dua arah).

- Karakteristik dari DHT11
 1. Input tegangan 3v hingga 5V
 2. Konsumsi arus maksimal 2.5mA saat digunakan selama konversi (saat meminta data)
 3. Kelembaban 20-80% dengan akurasi 5%
 4. Baik untuk pembacaan suhu 0-50 ° C dengan akurasi ± 2 ° C
 5. Pengambilan data minimal 1 Hz (sekali setiap detik)
- Cara Kerja DHT11

Sensor DHT11 merupakan serangkaian komponen sensor dan IC Kontroller yang dikemas dalam satu paket. Sensor ini ada yang memiliki 4 pin ada pula yang 3 pin. Tapi tidak menjadi masalah karena dalam penerapannya tiak ada perbedaan. Didalam bodi sensor yang berwarna biru atau putih terdapat sebuah Resistor dengan tipe NTC (*Negative Temperature Coefficient*).

Resistor jenis ini memiliki karakteristik dimana nilai resistansinya berbanding terbalik dengan kenaikan suhu. Artinya, semakin tinggi suhu ruangan maka nilai resistansi NTC akan semakin kecil. Sebaliknya nilai resistansi akan meningkat ketika suhu disekitar sensor menurun.

Selain itu didalamnya terdapat sebuah sensor kelembapan dengan karkteristik resistif terhadap perubahan kadar air di udara. Data dari kedua sensor ini diolah didalam IC Kontroller. IC Kontroller ini akan mengeluarkan output data dalam bentuk *single wire bi-directional*.

2.1.2 Relay



Relay memiliki fungsi sebagai pengontrol beban seperti saklar, namun relay bekerja sebagai saklar otomatis yang digerakkan oleh gaya elektromagnetik yang dihasilkan oleh suatu kumparan. Fungsi lain dari relay adalah untuk mengontrol arus besar dengan arus yang kecil.

2.1.3 Lampu Bohlam



Fungsi lampu bohlam disini untuk menghangatkan inkubator (menaikkan suhu di dalam inkubator).

2.1.4 ESP 8266



ESP8266 merupakan sebuah mikrokontroller yang sering digunakan untuk perangkat Internet of Things atau yang biasa disebut IoT. Mikrokontroller buatan Espressif Systems ini mempunyai fitur yang cukup lengkap dan mudah digunakan. Salah satu fitur yang paling menonjol adalah modul Wi-Fi.

2.2 Platform yang di gunakan

2.2.1 Firebase



Firebase Realtime Database adalah database NoSQL yang di-hosting di cloud dan dapat digunakan untuk menyimpan dan menyinkronkan data antarpengguna secara real time.

2.2.2 Kodular



Kodular adalah situs web yang menyediakan tools yang menyerupai MIT App Inventor untuk membuat aplikasi Android dengan menggunakan block programming. Dengan kata lain, anda tidak perlu mengetik kode program secara manual untuk membuat aplikasi Android.

2.2.3 Google Sheets



Program perangkat lunak spreadsheet adalah aplikasi komputer yang mensimulasikan lembar kerja kertas. Mereka memungkinkan pengguna untuk memasukkan dan memanipulasi data numerik dalam format kotak baris dan kolom.

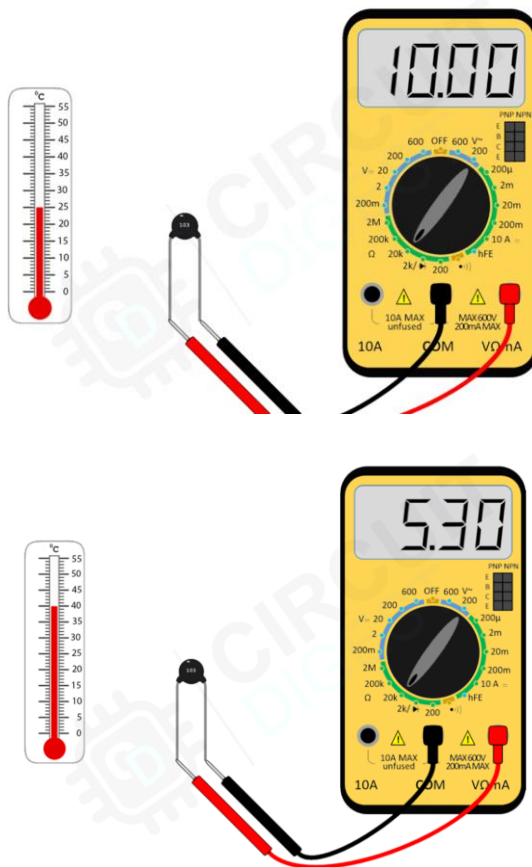
BAB 3 : PROSES AKUISISI DATA

3.1 Sistem Akuisisi Data DHT11

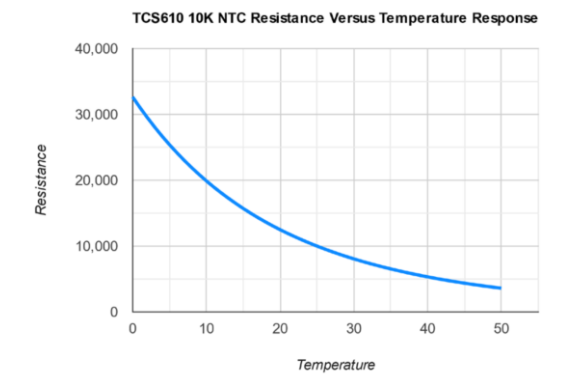
Sensor Kelembaban dan Suhu DHT11 terdiri dari 3 komponen utama. Sensor kelembaban tipe resistif, termistor NTC (koefisien suhu negatif) (untuk mengukur suhu) dan mikrokontroler 8-bit, yang mengubah sinyal analog dari kedua sensor dan mengirimkan sinyal digital tunggal.

3.2 Sensor Suhu DHT11

Pada Sensor suhu DHT11 digunakan (Negative Temperature Coefisien) adalah salah satu jenis sensor suhu golongan termistor.



Termistor NTC adalah resistor yang peka terhadap panas dengan Koefisien Suhu Negatif. Resistansinya akan berkurang ketika suhu meningkat.



Untuk mengukur resistansi termistor NTC, akan digunakan pembagi tegangan dan digunakan rumus

$$V_{out} = V_{in} * [R_2 / (R_1 + R_2)]$$

Karena kita mengetahui V_{in} , R_1 , dan V_{out} kita dapat menghitung nilai termistor NTC R_2 dengan persamaan berikut

$$R_2 = (V_{out} * R_1) / (V_{in} - V_{out})$$

Pada normalnya NTC dengan resistansi 10.000 memiliki value suhu sebesar 25°C

Proses perubahan sinyal analog menjadi digital dilakukan dengan sebuah cara mendeklarasikan nominal resistance, nominal temperature, dan samplingrate

```
#define nominal_resistance 10000    //Nominal resistance at 25°C

#define nominal_tempeprature 25    // temperature for nominal resistance (almost always 25° C)

#define samplingrate 5
```

Lalu dilakukan pengambilan data tegangan dari pembagi tegangan

```
digitalWrite(vd_power_pin,HIGH);

for (i=0; i< samplingrate; i++) {

    samples += analogRead(ntc_pin);

    delay(10);

}
```

Dari samples dan samplingrate yang didapatkan lalu dihitung nilai rata rata pembacaan tegangan

```
average = 0;  
average = samples/ samplingrate;
```

Setelah itu perhitungan resistansi NTC dapat dilakukan dengan perhitungan :

```
average = 1023 / average - 1;  
average = Rref/ average;
```

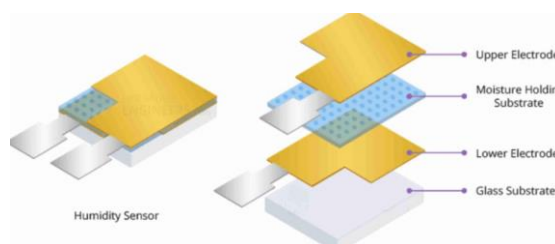
Dan dari perubahan resistansi itulah yang akan dikonversi menjadi digital read

```
temperature = average / nominal_resistance;  
temperature += 1.0 / (nominal_temperature + 273.15);  
temperature = 1.0 / temperature;
```

Dari pembacaan digital tersebut lalu diubah menjadi suhu dalam derajat celcius

```
temperature -= 273.15; // convert absolute temp to C
```

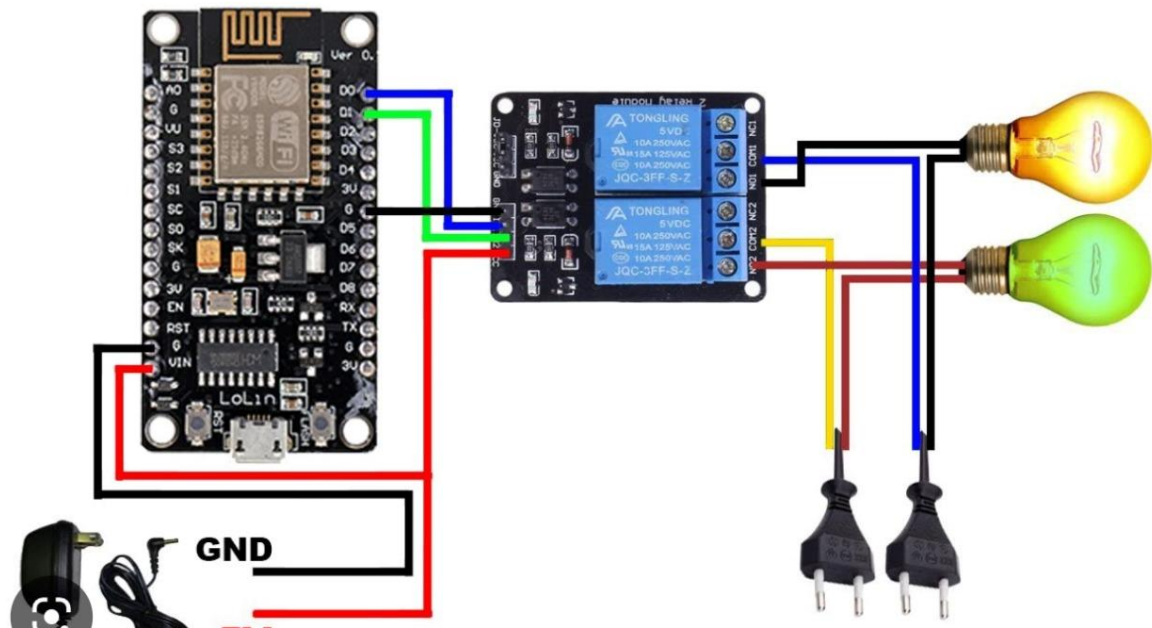
3.3 Sensor Kelembapan DHT11



Komponen sensor kelembapan terdiri dari substrat kelembapan yang diapit di antara dua elektroda. Ketika substrat menyerap kandungan air, resistansi antara kedua elektroda berkurang. Perubahan resistansi antara dua elektroda sebanding dengan kelembapan relatif. Kelembapan relatif yang lebih tinggi menurunkan resistansi antara elektroda, sedangkan kelembapan relatif yang lebih rendah meningkatkan resistansi antara elektroda. Kelembapan relative yang didapat akan dibaca oleh MCU ADC onboard

BAB 4 : PERANCANGAN SISTEM

4.1 Layout Wiring



4.2 Source Koding

```
#include <Arduino.h>
#if defined(ESP32)
  #include <WiFi.h>
#elif defined(ESP8266)
  #include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

#include "HTTPSRedirect.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"

#include "DHT.h"
#define DHTPIN 4 //D2
#define DHTTYPE DHT11
int relay = 12; //D6
int relay2 = 15; //D8

// Insert your network credentials
#define WIFI_SSID "PRS"
#define WIFI_PASSWORD "solat5waktu"

// Enter Google Script Deployment ID:
const char *GScriptId = "AKfycbwAuWgqI3sqTyMmLiKQIxq5N3aZPwuR_-
OisFNZhZORICUtCeaF9_mlqR5-w0YCTJn6";

// Enter command (insert_row or append_row) and your Google Sheets sheet name (default is
Sheet1):
String payload_base = "{\"command\": \"insert_row\", \"sheet_name\": \"Sheet1\", \"values\":
\";
String payload = "";

// Google Sheets setup (do not edit)
const char* host = "script.google.com";
const int httpsPort = 443;
const char* fingerprint = "";
String url = String("/macros/s/") + GScriptId + "/exec";
HTTPSRedirect* client = nullptr;

// Insert Firebase project API Key
#define API_KEY "AIzaSyAXDI3xDSl_uwhpemSHZgy7PTCdV2xT-dI"

// Insert RTDB URL define the RTDB URL */
#define DATABASE_URL "https://dht11-e4e73-default-rtdb.firebaseio.com/"

// Define Firebase Data object
FirebaseData fbdo;

```

```

FirebaseAuth auth;
FirebaseConfig config;
String strValue1, strValue2, lampu1, statelampu2, kondisi_lampu ;

unsigned long sendDataPrevMillis = 0;
int count = 0;
bool signupOK = false;

DHT dht(DHTPIN, DHTTYPE);

void setup(){
  Serial.begin(115200);
  pinMode(5, OUTPUT);
  pinMode(relay, OUTPUT);
  pinMode(relay2, OUTPUT);
  Serial.println(F("DHTxx test!"));
  dht.begin();
  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);

// Use HTTPSRedirect class to create a new TLS connection
client = new HTTPSRedirect(httpsPort);
client->setInsecure();
client->setPrintResponseBody(true);
client->setContentTypeHeader("application/json");

Serial.print("Connecting to ");
Serial.println(host);

// Try to connect for a maximum of 5 times
bool flag = false;
for (int i=0; i<5; i++){
    int retval = client->connect(host, httpsPort);
    if (retval == 1){
        flag = true;
        Serial.println("Connected");
        break;
    }
    else
        Serial.println("Connection failed. Retrying...");
}
if (!flag){
    Serial.print("Could not connect to server: ");
    Serial.println(host);
}

```



```

    return;
}

delete client; // delete HTTPSRedirect object
client = nullptr; // delete HTTPSRedirect object

}

void loop(){
    delay(1000);
    float h = dht.readHumidity();
    // Read temperature as Celsius (the default)
    float t = dht.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    float f = dht.readTemperature(true);

    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }

    float hif = dht.computeHeatIndex(f, h);
    // Compute heat index in Celsius (isFahreheit = false)
    float hic = dht.computeHeatIndex(t, h, false);

    if (Firebase.ready() && signupOK){
        sendDataPrevMillis = millis();
        // Write an Int number on the database path test/int
        if (Firebase.RTDB.setFloat(&fbdo, "DHT11/Suhu", t)){
            Serial.println("PASSED");
            Serial.print(F("Suhu: "));
            Serial.print(t);
            Serial.println(F(" °C"));
            Serial.println("PATH: " + fbdo.dataPath());
            Serial.println("TYPE: " + fbdo.dataType());
        }
        else {

```

```

Serial.println("FAILED");

// Serial.println("REASON: " + fbdo.errorReason());
}

// Write an Float number on the database path test/float
if (Firebase.RTDB.setFloat(&fbdo, "DHT11/Kelembapan", h)){
    Serial.println("PASSED");
    Serial.print(F("Kelembapan: "));
    Serial.print(h);
    Serial.println(F(" %"));
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
}
else {
    Serial.println("FAILED");
    // Serial.println("REASON: " + fbdo.errorReason());
}

if (Firebase.RTDB.getString(&fbdo, "/DHT11/Lampu1")) {
if (fbdo.dataType() == "string") {
    strValue1 = fbdo.stringData();
    Serial.println(strValue1);
    if (strValue1 == "1"){
        Serial.println("HAI");
        lampu1="ON";
        digitalWrite(relay, HIGH);
        digitalWrite(5, HIGH);
    }
    else {
        Serial.println("Yahh");
        digitalWrite(relay, LOW);
        lampu1="OFF";
        digitalWrite(5, LOW);
    }
}
}
else {

```

```

Serial.println(fbdo.errorReason());
}

if (Firebase.RTDB.getString(&fbdo, "/DHT11/Lampu2")) {
if (fbdo.dataType() == "string") {
    strValue2 = fbdo.stringData();
    Serial.println(strValue2);
    if (strValue2 == "1"){
        Serial.println("HAI");
        statelampu2="ON";
        digitalWrite(relay2, HIGH);
        digitalWrite(5, HIGH);
    }
    else {
        Serial.println("Yahh");
        digitalWrite(relay2, LOW);
        statelampu2="OFF";
        digitalWrite(5, LOW);
    }
}
}
else {
    Serial.println(fbdo.errorReason());
}

kondisi_lampu = ("Lampu 1 : " + lampu1 + " & " + "Lampu 2 : " + statelampu2);

static bool flag = false;
if (!flag){
    client = new HTTPSRedirect(httpsPort);
    client->setInsecure();
    flag = true;
    client->setPrintResponseBody(true);
    client->setContentTypeHeader("application/json");
}
if (client != nullptr){
    if (!client->connected()){

```

```

        client->connect(host, httpsPort);
    }
}
else{
    Serial.println("Error creating client object!");
}

// Create json object string to send to Google Sheets
payload = payload_base + "\"" + h + "," + t + "," + kondisi_lampu + "\"";

// Publish data to Google Sheets
Serial.println("Publishing data...");
Serial.println(payload);
if(client->POST(url, host, payload)){
    // do stuff here if publish was successful
}
else{
    // do stuff here if publish was not successful
    Serial.println("Error while connecting");
}

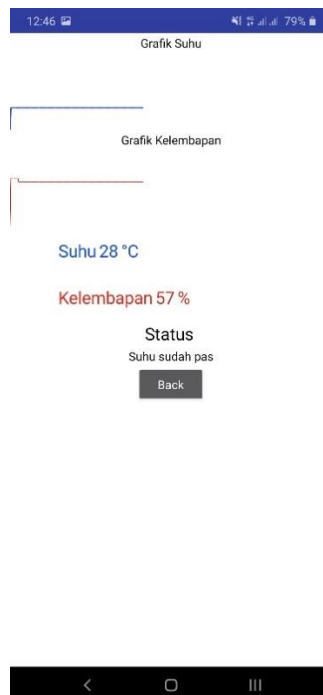
// a delay of several seconds is required before publishing again
}
}

```

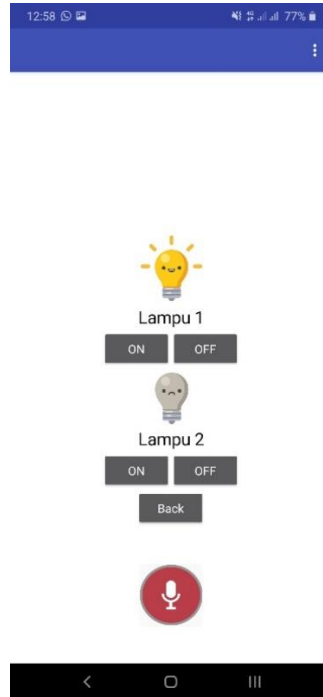
4.3 Tampilan Kodular



Pada Gambar pertama terdapat tampilan menu awal dari GUI smart incubator, terdapat menu Sensor, Lampu dan Data. Selanjutnya didalam menu sensor terdapat grafik suhu dan grafik kelembapan, dan dibawahnya terdapat hasil dari data yang diambil dht11 yaitu suhu dan kelembapan.



Pada menu Lampu terdapat tombol untuk menyalakan dan mematikan lampu satu dan dua, lalu di bawahnya terdapat tombol voice recognition untuk perintah menyalakan atau mematikan lampu.



Pada menu Data, Ketika menu ditekan, akan otomatis mendownload data dari dht11 yang telah disimpan pada google sheets.

4.4 Tampilan Google Sheet

DHT11 - Google Spreadsheet

docs.google.com/spreadsheets/d/1wE3yrNbEM0jQaLzh5V-awoPJ0YY-zHvzqFkv7h0/edit#gid=0

DHT11 Berfungsi secara offline

File Edit Tampilan Sisipkan Format Data Alat Ekstensi Bantuan

100% Rp % 0.00 123 Default (Arial) 10 B I A

	A	B	C	D	E	F	G	H	I	J
1	Date	Time	Kelembapan	Suhu	Lampu 1 & Lampu 2					
2	2023/01/06	2:49:17	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
3	2023/01/06	2:49:10	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
4	2023/01/06	2:49:04	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
5	2023/01/06	2:48:58	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
6	2023/01/06	2:48:51	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
7	2023/01/06	2:48:45	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
8	2023/01/06	2:48:39	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
9	2023/01/06	2:48:33	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
10	2023/01/06	2:48:27	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
11	2023/01/06	2:48:20	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
12	2023/01/06	2:48:12	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
13	2023/01/06	2:48:06	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
14	2023/01/06	2:47:58	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
15	2023/01/06	2:47:51	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
16	2023/01/06	2:47:43	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
17	2023/01/06	2:47:36	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
18	2023/01/06	2:47:29	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					
19	2023/01/06	2:47:22	62.00	26.70	Lampu 1 : OFF & Lampu 2 : OFF					

Sheet1 Pelajari

BAB 5 : PENGUJIAN DAN ANALISIS

Pada alat yang kami buat, kami menggunakan sensor DHT untuk mengambil sample dari lingkungan dari dalam incubator. DHT membaca suhu didalam incubator lalu dikirimkan ke firebase melalui ESP8266 yang terhubung oleh koneksi WIFI. Data yang masuk ke firebase merupakan data kelembapan dan suhu. Data yang ada di firebase bersifat realtime. Lalu data tersebut dikirim ke kodular dan google sheet secara realtime.

Dalam penampilan GUI untuk menampilkan data digunakan kodular, dari kodular itu terdapat beberapa menu, yaitu pembacaan sensor dan pengaturan lampu untuk incubator. Dalam menu pembacaan sensor terdapat tampilan pembacaan suhu dalam °C dan kelembapan dalam %. Lalu dalam menu pembacaan sensor terdapat juga grafik yang bergerak secara realtime mengikuti perubahan suhu dan kelembapan. User juga bisa mengatur berapa banyak lampu yang akan dinyalakan. Perintah dari kodular tersebut akan merubah status lampu pada firebase secara realtime lalu data status lampu tersebut akan dibaca oleh ESP8266 untuk menyalakan atau mematikan lampu berdasarkan perintah dari user melalui firebase.

Untuk proses penyimpanan data digunakan google sheet yang terhubung langsung oleh ESP8266. Ketika mendapatkan suatu data maka ESP8266 akan mengirimkan data langsung ke google sheet. Adapun langkah agar google sheet dan ESP8266 dapat terhubung yaitu dengan cara mendapatkan Script ID Google sheet lalu menambahkannya di kodingan untuk dibaca oleh ESP8266.



BAB 6 : KESIMPULAN DAN SARAN

Pada alat yang kami buat kami menggunakan beberapa alat elektronika yaitu Sensor DHT, ESP8266, dan Relay. Sensor DHT berfungsi untuk mengambil sample dari lingkungan dari dalam incubator. Pengambilan sample dapat berupa suhu dan kelembapan. Pada DHT11 pengambilan suhu dilakukan dengan komponen Termistor didalamnya yang menggunakan prinsip kerja NTC dan mengkalkulasikannya berdasarkan perubahan resistansi, semakin besar suhu maka semakin kecil resistansi dari NTC. Lalu pembacaan kelembapan pada DHT11 menggunakan perbedaan resistansi dua elektroda. Perubahan resistansi antara dua elektroda sebanding dengan kelembapan relatif. Kelembapan relatif yang lebih tinggi menurunkan resistansi antara elektroda, sedangkan kelembapan relatif yang lebih rendah meningkatkan resistansi antara elektroda.

Data yang telah dibaca lalu dikirim ke platform Firebase untuk dibaca oleh kodular sebagai GUI dalam akuisisi data dan data juga dikirim dari ESP8266 untuk disimpan didalam google spreadsheet sehingga kita bisa melihat log data yang telah terbaca sebelumnya.