**Laporan Ujian Tengah Semester**

**Perancangan Sistem Tertanam**



**SMART GARDEN**

**Disusun Oleh:**

Lea Defo Sadewo (1102201668)

I Gst Arya Agung (1102204486)

Rifky Awwala Adriano (1102200390)

Maulana Kautsar (1102204467)

Ikhsanul Iqbal (1102202545)

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK ELEKTRO

TELKOM UNIVERSITY

BANDUNG

2023

# **Latar Belakang**

Kemajuan teknologi yang begitu pesat membuat kita menciptakan inovasi yang nantinya bermanfaat positif bagi kehidupan manusia. Salah satunya adalah membuat sistem kebun pintar atau *smart garden* artinya berkomunikasi dengan tanaman bisa untuk mengatur penyiraman tanaman lalu bisa membaca kondisi tanah apakah dalam kondisi lembab, suhu lingkungan, dan cahaya.

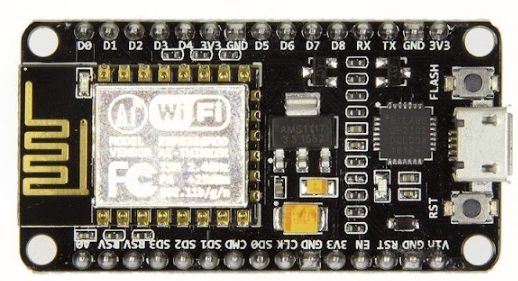
Dari berbagai macam faktor tersebut sistem kontrol digunakan untuk memudahkan pengguna dalam proses penyiraman yang terpadu dengan tujuan untuk mengendalikan sistem-sistem pada *smart garden* yaitu penyiraman otomatis serta mempermudah perawatannya tanpa harus adanya campur tangan manusia secara lansung.

# **2.1 Spesifikasi Alat**

Spesifikasi dari kelompok yang kami rancang adalah:

1. ESP 8266

Mikrokontroller yang kami gunakan karena dapat terhubung dengan Wi-Fi dan harganya terjangkau.



Gambar 2.1.1 NodeMCU ESP8266

1. Sensor Kelembapan Tanah

Sensor yang mampu mendeteksi intensitas air di dalam tanah (kelembapan tanah).



Gambar 2.1.2 Sensor Kelembapan Tanah

1. Sensor Suhu dan Kelembapan

Paket sensor yang berfungsi untuk mengukur suhu dan kelembaban udara sekaligus.

A small white device with a white square and metal bar

Description automatically generated with medium confidence

Gambar 2.1.3 Sensor Kelembapan Tanah

1. Water Pump 5 Volt

Digunakan untuk mengalirkan atau memompa air dari sumber air.

A small white electric motor with black wire

Description automatically generated

Gambar 2.1.4 Water Pump

1. LCD I2C

Menggunakan modul I2C untuk mempermudah dalam pemograman LCD dan dapat menghemat pin dari LCD ke microkontroler karena output hanya 4 pin (VCC, GND, SCL, SDA).

A close-up of a circuit board

Description automatically generated

Gambar 2.1.5 Water Pump

1. LED

Menggunakan LED untuk sebagai alat penerangan pada smart garden.

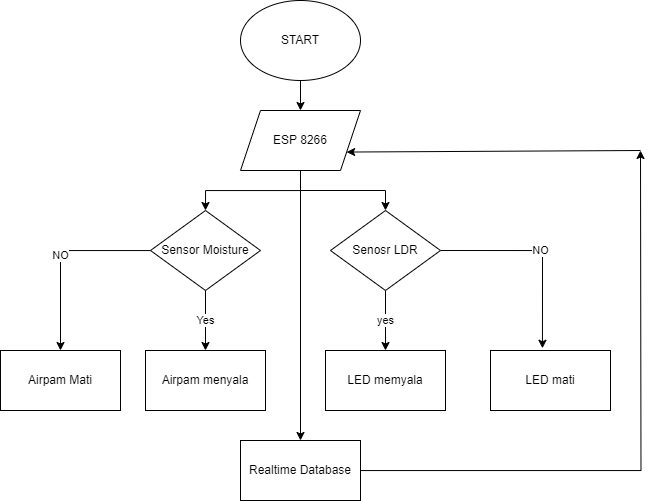
A clear led with two thin wires

Description automatically generated

Gambar 2.1.5 Water Pump

# **2.2 FLOWCHART**

Adapun flowchart dari kelompok kami sebagi berikut:



Gambar 2.2.1 FLOWCHART

# **2.3 DIAGRAM BLOCK**

Adapun Diagram Block dari kelompok kami sebagai berikut:

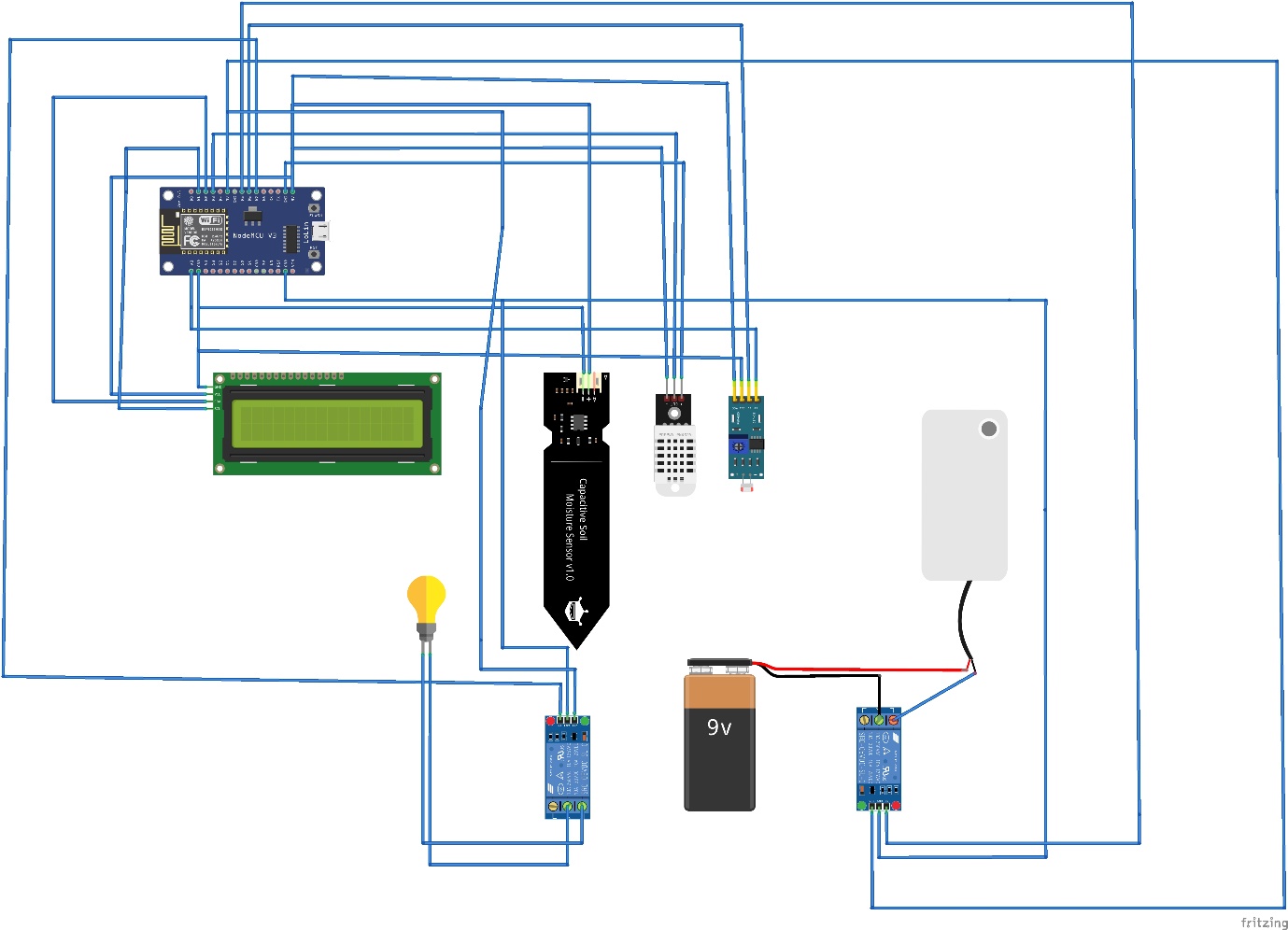
A diagram of a computer network

Description automatically generated

Gambar 2.3.1 DIAGRAM BLOCK

# **2.4 WIRING DIAGRAM**

Adapun Wiring Diagram dari kelompok kami sebagai berikut:



# **2.5 KODINGAN** Adapun Kodingan dari kelompok kami sebagai berikut:

#include <Arduino.h>

#include <time.h>

#include <WiFi.h>

// FIREBASE

#include <Firebase\_ESP\_Client.h>

#include "addons/TokenHelper.h" // Provide the token generation process info.

#include "addons/RTDBHelper.h" // Provide the RTDB payload printing info and other helper functions.

// PIN CONST

#define SOIL\_PIN 34 // ESP32 pin that connects to AOUT pin of moisture sensor

#define LDR\_PIN 19 // ESP32 pin that connects to AOUT pin of LDR sensor

#define PUMP\_PIN 32 // ESP32 pin that connects to AOUT pin of PUMP sensor

// FIREBASE SETUP

#define DATABASE\_URL "https://iot-tubes-9509f-default-rtdb.firebaseio.com/"

#define API\_KEY "xQAcyS5i0cNWklu9CSpmTEk9uPrf5KZcCH4lPI5w"

FirebaseData fbdo;

FirebaseAuth auth;

FirebaseConfig config;

bool signupOK = false;

String nightmodeFD = String("garden/nightmode");

String sprinklerFD = String("garden/sprinkler");

String soilFD = String("garden/tanah");

// Insert your network credentials

#define WIFI\_SSID "Galaxy A50s527"

#define WIFI\_PASSWORD "yorw9305"

// Function List

void firebaseSetString(String, String);

void firebaseSetInt(String, int);

String firebaseGetString(String databaseDirectory);

const int AirValue = 4095; //you need to replace this value with Value\_1

const int WaterValue = 3600; //you need to replace this value with Value\_2

int soilmoisturepercent = 0;

void setup() {

Serial.begin(9600);

pinMode(PUMP\_PIN, OUTPUT);

pinMode(LAMP\_PIN, OUTPUT);

pinMode(LED\_BUILTIN, OUTPUT);

pinMode(SOIL\_PIN, INPUT);

pinMode(LDR\_PIN, INPUT);

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();

/\* FIREBASE SETUP START \*/

config.api\_key = API\_KEY; // Assign the api key (required)

config.database\_url = DATABASE\_URL; // Assign the RTDB URL (required)

/\* Sign up Start \*/

if (Firebase.signUp(&config, &auth, "", ""))

{

Serial.println("ok");

signupOK = true;

}

else

{

String firebaseErrorMessage = config.signer.signupError.message.c\_str();

Serial.printf("%s\n", firebaseErrorMessage);

}

/\* Sign up end\*/

/\* FIREBASE SETUP END \*/

/\* Assign the callback function for the long runfng token generation task \*/

config.token\_status\_callback = tokenStatusCallback; // see addons/TokenHelper.h

Firebase.begin(&config, &auth);

Firebase.reconnectWiFi(true);

}

void loop() {

int soilValue = analogRead(SOIL\_PIN); // read the analog value from sensor

soilmoisturepercent = map(soilValue, WaterValue, AirValue, 0, 100);

int ldrValue = digitalRead(LDR\_PIN); // read the analog value from sensor

String sprinkleState = firebaseGetString(sprinklerFD);

String nightState = firebaseGetString(nightmodeFD);

sprinkleState == "ON" ? digitalWrite(PUMP\_PIN, LOW) : digitalWrite(PUMP\_PIN, HIGH);

if (ldrValue == 1)

{

digitalWrite(LED\_BUILTIN, LOW);

if (nightState == "OFF")

{

firebaseSetString(nightmodeFD, "ON");

}

}

else if(ldrValue == 0){

digitalWrite(LED\_BUILTIN, HIGH);

if (nightState == "ON")

{

firebaseSetString(nightmodeFD, "OFF");

}

}

//SEND DATA TO FIREBASE

if (soilmoisturepercent >= 100)

{

soilmoisturepercent = 100;

firebaseSetInt(soilFD, soilmoisturepercent);

}

else {

firebaseSetInt(soilFD, soilmoisturepercent);

}

}

void firebaseSetString(String databaseDirectory, String value)

{

// Write a string on the database path

if (Firebase.RTDB.setString(&fbdo, databaseDirectory, value))

{

Serial.println("PASSED");

}

else

{

Serial.println("FAILED");

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

}

}

void firebaseSetInt(String databaseDirectory, int value)

{

// Write an Int number on the database path test/int

if (Firebase.RTDB.setInt(&fbdo, databaseDirectory, value))

{

Serial.println("PASSED");

}

else

{

Serial.println("FAILED");

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

}

}

String firebaseGetString(String databaseDirectory)

{

if (Firebase.RTDB.getFloat(&fbdo, databaseDirectory))

{

String stringValue = fbdo.stringData();

return stringValue;

}

else

{

Serial.println(fbdo.errorReason());

return "FAILED";

}

}

# **2.7 LAMPIRAN**

A screenshot of a computer

Description automatically generated

Gambar 2.6.1 Lampiran Masih Error

# **3. Logbook Pembuatan**

|  |  |  |
| --- | --- | --- |
| No. | Tanggal | Masalah |
| 1. | 25/10/23 | Merakit alat smart garden |
| 2. | 5/11/23 | Terjadi masalah pada kodingan yang menyebabkannya error |
| 4. | 05/11/23 |  |

# **4.1 KESIMPULAN**

Smart Garden adalah hasil perancangan alat yang bertujuan untuk mempermudah dan meningkatkan efisiensi pekerjaan manusia yang sebelumnya harus mengandalkan perkiraan atau estimasi kondisi tanah untuk menilai kesehatan tanaman. Dengan pemanfaatan sistem smart garden ini, kita dapat dengan lebih tepat dan efisien menentukan tingkat kelembapan tanah, suhu lingkungan, dan cahaya.

# **4.2 SARAN**

Kepada kelompok kami semoga bisa merealisasikan smart garden ini untuk lingkup yang lebih luas.