

LAPORAN PROYEK

MONITORING SUHU DALAM KOTAK



Kelompok 6:

Ridho Walid Al Maulidi	(5024201036)
Arrijal Istighfarotudzdzilal	(5024201059)
Ariq Maulana Tazakka	(5024211039)
Octavianus Immanuel Christpurwanto	(5024211014)
Naufal Athallah Irwanto	(07211840000061)

Dosen Pengampu:

Dr. Arief Kurniawan, S.T., M.T.

DEPARTEMEN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI ELEKTRO INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2023/2024

BAB 1 PENDAHULUAN

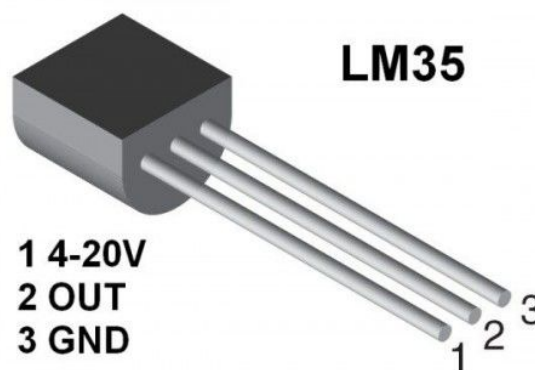
1.1 Deskripsi Tugas dan Proyek

Pada tugas mata kuliah Desain Aplikasi dan Internet of Things kali ini, kami mahasiswa ditugaskan untuk membuat suatu alat yang berhubungan dengan Internet of Things. Kelompok kami membuat proyek alat monitoring suhu dalam kotak. Alat ini berfungsi untuk memonitoring suhu dalam kotak, yang mana penggunaannya dalam dunia nyata dapat digunakan sebagai tempat mengerami telur ayam. Suhu tersebut nantinya dapat diatur melalui website simpel yang kami buat menggunakan node red. Hal yang dilakukan untuk mengatur suhu dalam kotak adalah dengan mengatur kecepatan kipas atau mengatur nyala lampu yang ada dalam kotak untuk menyesuaikan suhu dalam kotak agar tidak terlalu dingin atau terlalu panas, sehingga menghasilkan suhu yang optimal untuk mengerami telur.

BAB 2 DASAR TEORI

2.1 Sensor

Sensor merupakan alat atau komponen elektronik yang memiliki kemampuan untuk mengidentifikasi perubahan dalam suatu kondisi lingkungan fisik?. Sensor yang kami gunakan adalah LM35, LM35 sendiri adalah sensor suhu Analog yang akan menghasilkan tegangan voltase yang sebanding atau linear dengan suhu sekitarnya, sensor LM35 memiliki koefisien 10mV per Celcius dengan rentang kerja -55°C sampai 150°C. sensor LM35 juga memiliki akurasi yang tinggi dalam pembacaan data temperature, beberapa varian dari LM35 memiliki fitur untuk kompensasi suhu internal yang membuat sensor akan tetap akurat di berbagai kondisi. pada beberapa kasus, kalibrasi dalam sensor mungkin diperlukan untuk menjaga akurasi dan beradaptasi pada lingkungan. lm35 memiliki 3 pinout, yang pertama ground sebagai referensi, Vin sebagai Vcc atau referensi kedua dan Vout sebagai pin output dari pengukuran suhu lm35.



Gambar 1: LM35

Source: <https://breakrow.com/mili-ampere/menggunakan-sensor-suhu-lm35-dengan-arduino>

2.2 Aktuator

Peralatan mekanis untuk menggerakkan atau mengontrol sebuah sistem yang digunakan sebagai proses lanjutan dari keluaran suatu proses olah data yang dihasilkan dari sensor merupakan definisi dari aktuator?. Aktuator yang kami gunakan adalah lampu dan kipas. lampu diatur hidup mati menggunakan

relay yang mendapatkan input dari mikrocontroller. lampu yang kami gunakan memiliki input voltase 220v ac. kipas juga diaturhidup mati menggunakan relay, kipas memiliki voltage converter karena kipas memiliki input voltase 12v Dc.



Gambar 2: Aktuator

2.3 Arduino IDE

Proses pemrograman Arduino dilakukan melalui Software Processing, yang dirancang untuk menulis program untuk Arduino. Processing sendiri merupakan gabungan dari bahasa C++ dan Java. Keunggulan Arduino terletak pada kemampuannya berupa kombinasi perangkat keras, bahasa pemrograman, dan Integrated Development Environment (IDE) yang canggih. IDE berperan penting dalam menulis program, mengkompilasi menjadi kode biner, dan mengunggahnya ke dalam memori mikrokontroler?.



Gambar 3: Logo Arduino IDE

Source: <https://www.pngwing.com/en/free-png-adjvu/download>

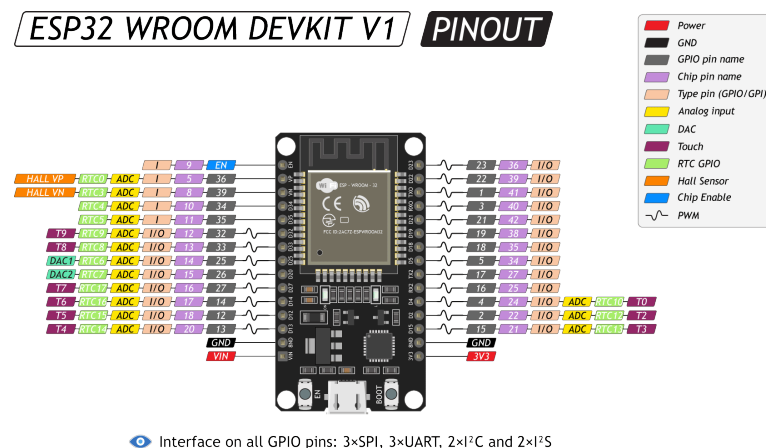
Software IDE Arduino terdiri dari tiga bagian utama: editor program untuk menulis dan mengedit program dalam bahasa Processing, kompiler yang mengubah kode program menjadi kode biner yang dapat dipahami oleh mikrokontroler, dan uploader yang memasukkan kode biner ke dalam memori mik-

rokontroler. Selain itu, Arduino dapat diinstal pada berbagai sistem operasi seperti LINUX, Mac OS, dan Windows, menjadikannya alat pengembangan yang fleksibel dan dapat diakses oleh berbagai pengguna.?

2.4 Mikrokontroller ESP-WROOM-32

ESP32 adalah mikrokontroler System on Chip (SoC) terpadu yang dilengkapi dengan WiFi 802.11 b/g/n, Bluetooth versi 4.2, dan berbagai peripheral. Dengan fitur-fitur seperti GPIO (General Purpose Input Output), prosesor terintegrasi yang kuat, dan kemampuan penyimpanan internal, ESP32 dapat berfungsi sebagai pengganti mikrokontroler Arduino. Kelebihan lainnya termasuk dukungan untuk koneksi Wi-Fi langsung, memungkinkan pengguna untuk mengintegrasikan perangkat ini dengan mudah ke dalam proyek IoT atau aplikasi yang memerlukan konektivitas nirkabel. ?

ESP32 berperan ganda dalam proyek kami. Pertama, ia berfungsi sebagai publisher yang bertanggung jawab untuk menangkap data suhu melalui sensor LM35. Sensor ini memungkinkan pengukuran suhu yang akurat, dan ESP32 secara efisien mengirimkan informasi ini, memungkinkan pemantauan dan kontrol real-time.



Gambar 4: ESP32 Pin Out

Source: <https://lastminuteengineers.com/esp32-pinout-reference/>

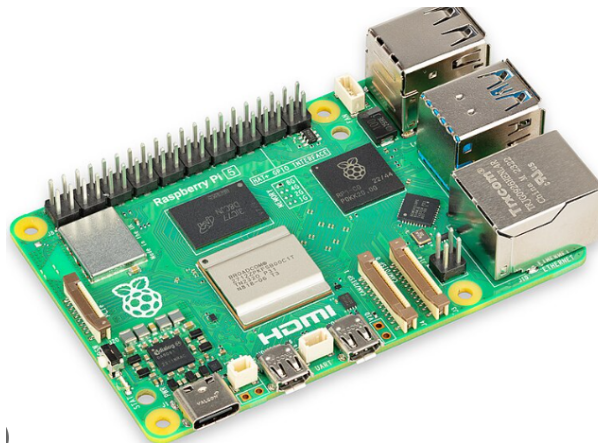
Kedua, ESP32 mengemban peran sebagai pelanggan (subscriber), secara aktif terlibat dengan topik MQTT seperti "lampu" dan "kipas". Sebagai pelanggan, ESP32 merespons pesan yang masuk terkait dengan topik-topik ini, memungkinkannya untuk mengontrol perangkat yang terhubung. Fungsi dua arah ini menunjukkan kemampuan ESP32 tidak hanya untuk mengumpulkan

data sensor tetapi juga untuk berinteraksi dengan dan memengaruhi lingkungan sekitar berdasarkan pesan MQTT yang diterima.

2.5 Raspberry Pi

Raspberry Pi adalah sebuah komputer mini yang, dalam fungsinya, mirip dengan komputer umum namun dalam bentuk yang lebih kecil. Seperti board mikrokontroler pada umumnya, Raspberry Pi memiliki input dan output. Di antara seri produk Raspberry Pi, Raspberry Pi 4 model B menjadi pilihan unggul dengan peningkatan signifikan dalam hal kecepatan prosesor, multimedia, kinerja, memori, dan konektivitas jika dibandingkan dengan generasi sebelumnya. Yang menarik, meskipun memiliki kemampuan yang canggih, Raspberry Pi 4 model B tetap efisien dalam penggunaan daya listrik. Keunggulan lainnya terletak pada pin GPIO yang terprogram secara fleksibel, memungkinkan pengguna untuk mengumpulkan data atau mengendalikan modul sensor dengan menggunakan pin sebagai port serialnya ?.

Raspberry Pi memainkan peran kunci dalam proyek ini, digunakan untuk menginstal Mosquitto dan klien Mosquitto. Sebagai perangkat mini yang handal, Raspberry Pi memberikan platform yang ideal untuk menjalankan perangkat lunak MQTT broker seperti Mosquitto, memungkinkan komunikasi yang efisien antara perangkat yang terhubung dalam jaringan IoT kami.



Gambar 5: Raspberry Pi 4 Model B

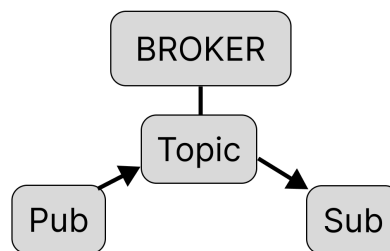
Source: <https://www.yangcanggih.com/2019/06/24/raspberry-pi-4-model-b-performa-lebih-kencang-mendukung-video-4k-60fps/>

Dengan menginstal Mosquitto di Raspberry Pi, kami membangun dasar komunikasi yang andal dan aman. Mosquitto berfungsi sebagai broker MQTT yang dapat diandalkan, menyediakan layanan yang mendukung pertukaran

pesan antara perangkat dalam proyek kami. Penggunaan Raspberry Pi sebagai tuan rumah untuk Mosquitto dan klien Mosquitto memperkuat kontrol dan manajemen jaringan IoT kami, memberikan dasar yang kokoh untuk pengiriman dan penerimaan data yang efisien antara perangkat dalam ekosistem proyek kami.

2.6 MQTT Protocol

Dalam ekosistem Internet of Things (IoT), protokol MQTT menjadi protokol untuk mengatur komunikasi efisien antara perangkat. Pada implementasinya MQTT Protokol memiliki 3 object penting yaitu broker, publisher dan subscriber. Dalam implementasi ini, Raspberry Pi digunakan sebagai broker MQTT, yang memungkinkan komponen lingkungan IoT lainnya untuk berkomunikasi secara lancar.



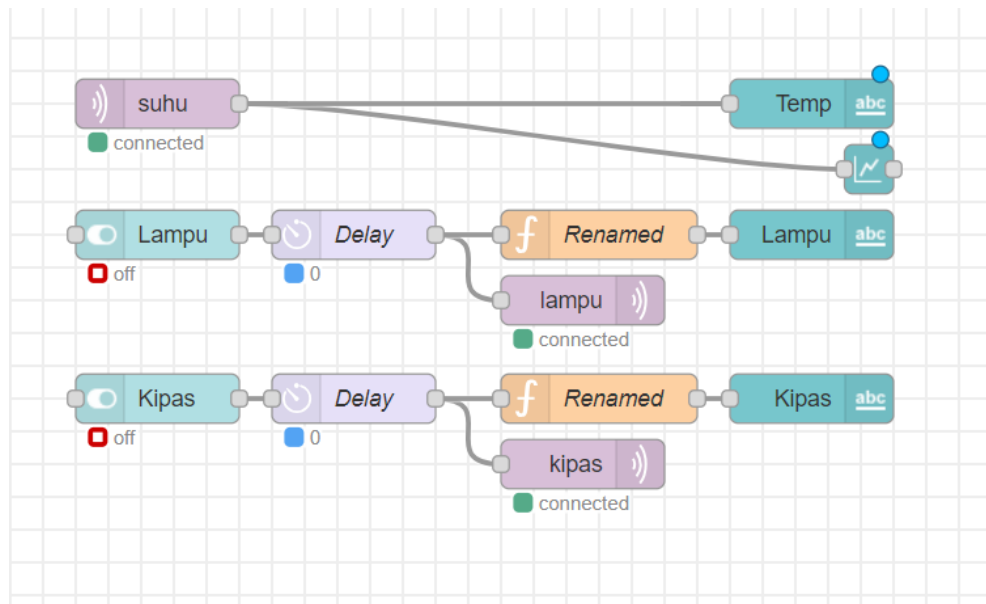
Gambar 6: MQTT

ESP32, yang berperan ganda sebagai publisher dan subscriber, memainkan peran dalam mengumpulkan dan mendistribusikan data. Sebagai penerbit, ESP32 dapat mengirimkan data sensorik atau informasi lain ke broker MQTT di Raspberry Pi. Sebaliknya, sebagai pelanggan, ESP32 dapat mensubscribe untuk menerima informasi atau perintah tertentu dari topik yang relevan. Selain itu, Node-RED Dashboard memberikan antarmuka pengguna yang interaktif untuk mengawasi dan mengontrol perangkat IoT. Dengan kemampuan Node-RED untuk berfungsi sebagai pelanggan dan penerbit MQTT, dashboard ini dapat menerima pembaruan dari komponen lain dan mengirimkan perintah kembali ke komponen pada lingkungan IoT.

2.7 Node-Red Dashboard

Node-Red merupakan framework untuk iot yang sering digunakan. Node-red memiliki berbagai module yang bisa digunakan untuk menerima data dari

MQTT, menerima data dari HTTP, menampilkan UI dashboard, dan berbagai fungsi lainnya.

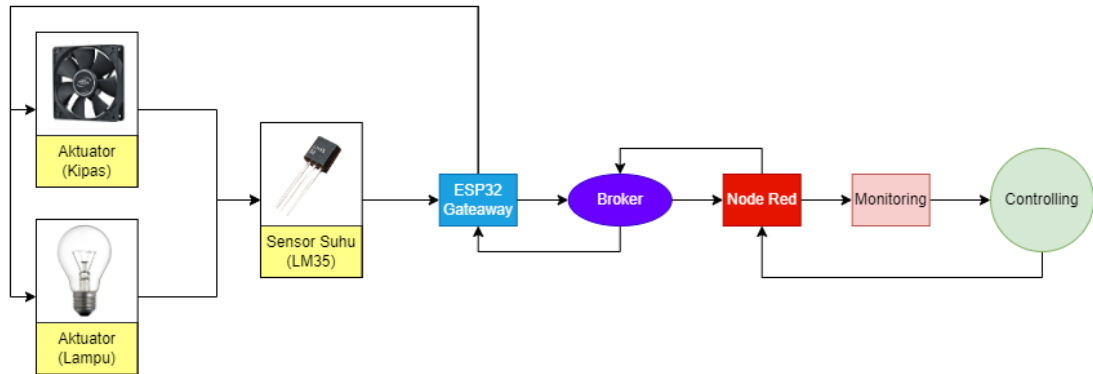


Gambar 7: Node-Red

pada implementasi kami, kami menerima data dari topic suhu mqtt pada raspberry pi broker, data ini akan ditampilkan dalam bentuk grafik. Dashboard dari node-red menyediakan kontrol untuk lampu dan kipas berupa switch untuk mengubah statenya secara manual. selain itu Dashboard dari node red juga menyediakan tampilan teks yang menampilkan suhu terakhir serta status aktif dari Lampu dan Kipas.

BAB 3 METODOLOGI

3.1 Alur Kerja Proyek



Gambar 8: Alur Kerja Proyek

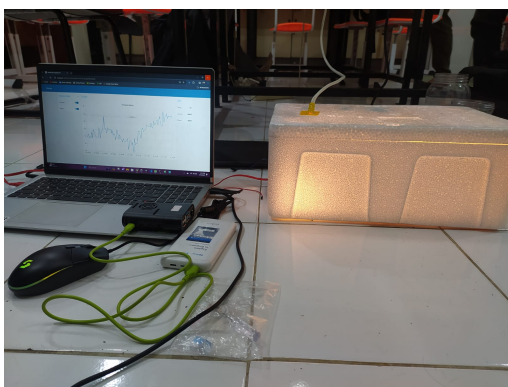
3.2 Penjelasan Alur Kerja

- Sensor LM35 membaca suhu dalam kotak dengan tegangan yang dia dapatkan berdasarkan suhu. Semakin panas suhu dalam kotak, maka voltasenya makin tinggi. Sebaliknya, semakin dingin suhu dalam kotak, maka voltasenya makin rendah.
- ESP32 menerima data suhu dari sensor LM35 dengan cara membaca sinyal analog yang diberikan oleh sensor suhu LM35. Sinyal analog tersebut kemudian diubah menjadi sinyal digital
- Broker (Raspberry Pi) menerima sinyal digital yang diberikan oleh ESP32
- Subscriber (Node Red) meminta data sinyal digital dari broker untuk memonitoring data suhu tersebut.
- Setelah meminta data suhu dari broker, akan dilihat apakah suhu dalam kotak terlalu dingin atau terlalu panas. Jika terlalu dingin, maka kita akan atur agar lampu menyala. Sebaliknya, jika terlalu panas, maka kita akan atur agar kipas yang menyala.
- Setelah controlling tersebut, Node Red yang sebelumnya menjadi subscriber, sekarang menjadi broker untuk mengirim salah satu perintah tadi ke Raspberry Pi, yang mana sekarang menjadi subscriber.

- Raspberry Pi tadi menerima perintah, dan menerjemahkan perintah tersebut agar dapat dibaca oleh ESP32 sebagai perintah kepada aktuator, antara lampu yang menyala atau kipas yang menyala.
- Salah satu aktuator akan menyala untuk menyesuaikan suhu dalam kotak tersebut agar tetap stabil.

BAB 4 HASIL KERJA

Hasil kerja yang kami hasilkan adalah berupa produk yang dapat diimplementasikan di kehidupan nyata. Produk ini dapat diimplementasikan ke telur yang dapat menetas. Kami menggunakan box sterefoam sebagai closure. Di dalam sterefoam, kami memasukkan semua alat yang dibutuhkan seperti yang telah dijelaskan pada pembahasan. Cara kerja dari alat ini adalah meletakkan telur di dalam sterefoam. Sensor suhu sangat berperan dan berpengaruh ke alat yang kami kerjakan. Ketika sensor mendeteksi perubahan suhu, salah satu dari dua aktuator akan bekerja. Lampu dan kipas dapat dikendalikan oleh program dari esp32 dan sensor suhu yang sangat berperan sekali pada produk yang kami gunakan ini. Berikut adalah foto hasil produk yang sudah jadi:



Gambar 9: Alat tampak depan



Gambar 10: Alat tampak samping

Berikut adalah listing program yang digunakan menggunakan platform Arduino IDE:

```
#include <WiFi.h>
#include <PubSubClient.h>

#define PIN_LM35          36 // ESP32 pin GPIO36 (ADC0)
    ↳ connected to LM35
#define lampu_pin         2 //
#define kipas_pin         16 //

const char* ssid = "Note8";
const char* password = "mantapjiwa";
const char* mqtt_server = "192.168.139.87"; // Ganti
    ↳ dengan alamat IP Raspberry Pi
```

```

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  client.setServer(mqtt_server, 1883);

  pinMode(lampu_pin, OUTPUT);
  pinMode(kipas_pin, OUTPUT);

  client.setCallback(callback);
  // client.publish("lampu", "LOW");
  client.subscribe("lampu");
  client.subscribe("kipas");
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }

  int serialvalue = analogRead(PIN_LM35);
  float temperature = (serialvalue / 4095.0) * 500.0;
  ↪ // Koreksi perhitungan

  char message[50]; // Buat array untuk menyimpan
  ↪ pesan
  snprintf(message, 50, "%.2f", temperature); //
  ↪ Ubah nilai float menjadi string dengan dua

```

↪ angka desimal

```
Serial.print("Temperature:");  
Serial.println(message);  
client.publish("suhu", message); // Kirim pesan  
    ↪ dengan string hasil konversi  
// Serial.print("Subscribed to lampu: ");  
// Serial.println(client.subscribe("lampu"));  
// client.subscribe("lampu");  
  
// if (Serial.available() > 0) {  
//     char input = Serial.read();  
//     if (input == '0' || input == '1') {  
//         if (input == '0') {  
//             client.publish("lampu", "LOW"); // Kirim  
//                 ↪ "LOW" jika input adalah '0'  
//         } else {  
//             client.publish("lampu", "HIGH"); // Kirim  
//                 ↪ "HIGH" jika input adalah '1'  
//         }  
//     }  
// }  
  
client.loop();  
delay(1000);  
}  
  
void callback(String topic, byte* message, unsigned  
    ↪ int length) {  
    Serial.print("Message arrived on topic:");  
    Serial.println(topic);  
    Serial.print("Message:");  
    String messageTemp;  
  
    for (int i = 0; i < length; i++) {  
        Serial.print((char)message[i]);  
    }
```

```

        messageTemp += (char)message[i];
    }
    Serial.println();

    if(topic=="lampu"){
        Serial.print("Changing_lamp_to_");
        if(messageTemp == "HIGH"){
            digitalWrite(lampu_pin, HIGH);
            Serial.print("On");
        }
        else if(messageTemp == "LOW"){
            digitalWrite(lampu_pin, LOW);
            Serial.print("Off");
        }
        // delay(3000);
    }
    Serial.println();

    if(topic=="kipas"){
        Serial.print("Changing_fan_to_");
        if(messageTemp == "HIGH"){
            digitalWrite(kipas_pin, HIGH);
            Serial.print("On");
        }
        else if(messageTemp == "LOW"){
            digitalWrite(kipas_pin, LOW);
            Serial.print("Off");
        }
        // delay(3000);
    }
    Serial.println();
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Connecting_to_MQTT...");

```

```
    if (client.connect("ESP32Client")) {  
        Serial.println("connected");  
        client.subscribe("lampu");  
        client.subscribe("kipas");  
    } else {  
        Serial.print("failed,rc=");  
        Serial.print(client.state());  
        Serial.println("try again in 5 seconds");  
        delay(5000);  
    }  
}  
}
```

BAB 5 KESIMPULAN

Dari hasil yang didapat, produk yang dikerjakan bekerja dengan baik. Sensor suhu yang digunakan menangkap suhu dalam kotak dengan baik dan esp32 mengirimkan data tersebut ke broker lalu node-red menerima data suhu tersebut dari broker dan menampilkannya pada Dashboard.

LAMPIRAN



