



Perancangan Arsitektur Node Nirkabel dalam Efisiensi *Bandwidth Smart Greenhouse* Berbasis Protokol MQTT

Uray Ristian^{#1}, Ikhwan Ruslianto^{#2}, Hirzen Hasfani^{#3}, Kartika Sari^{#4}

[#]Program Studi Rekayasa Sistem Komputer Fakultas MIPA, Universitas Tanjungpura
Jl. Prof. Dr.H.Hadari Nawawi, Pontianak, 78124

¹eristian@siskom.untan.ac.id

²ikhwanruslianto@siskom.untan.ac.id

³hirzen.hasfani@siskom.untan.ac.id

⁴kartika.sari@siskom.untan.ac.id

Abstrak— Pertanian *Greenhouse* merupakan salah satu sistem dan media pertanian di dalam ruangan. Dalam hal ini, tanaman yang dipantau pertumbuhannya adalah tanaman anggur di *Greenhouse* di Pondok Pesantren Darul Fikri, Kabupaten Kubu Raya. *Greenhouse* ini menggunakan node-node nirkabel yang terhubung ke sensor-sensor pembacaan kondisi tanaman seperti suhu udara, kelembapan udara dan kelembapan tanah pada pot anggur untuk disimpan dan diolah web server menjadi sistem informasi. Akan tetapi, makin banyak node nirkabel yang digunakan, maka penggunaan *bandwidth* dan *latency* semakin meningkat. Penelitian ini memanfaatkan protokol MQTT (*Queueing Telemetry Transport*) untuk jalur komunikasi dengan perangkat IoT. Selain itu, protokol MQTT juga dibandingkan dengan protokol HTTP untuk melihat efisiensi dari kecepatan *respon time* dan *latency* saat pengiriman data ke server. Perbandingan waktu respon antara MQTT dengan HTTP untuk pengiriman data ke server setiap 1 jam adalah 0.08 detik (MQTT) dan 0.4 detik (HTTP). Kemudian, perbandingan antara pengiriman data ke server setiap 10 menit adalah 0.04 detik (MQTT) dan 0.5 detik (HTTP). Selanjutnya, perbandingan antara pengiriman data ke server setiap detik adalah 0.08 detik (MQTT) dan 1.12 detik (HTTP). Hasilnya, pengiriman data menggunakan protokol MQTT lebih stabil dan waktu responnya lebih cepat dibandingkan dengan HTTP.

Kata kunci— *Teknologi Pertanian, Anggur, Greenhouse, MQTT, Internet of Things.*

I. PENDAHULUAN

Salah satu program pemerintah untuk mengatasi isu ketahanan pangan di Indonesia adalah melalui penerapan teknologi pertanian tepat guna. Teknologi ini digunakan untuk mengatasi masalah pertanian di lahan yang terbatas. Teknologi pertanian tepat guna, seperti Hidroponik, *Roof top Farming*, dan *Greenhouse*, dapat memberikan manfaat bagi pertanian di lahan terbatas. [1]. Jika teknologi tersebut sesuai dengan permasalahan yang

dihadapi oleh sebagian besar masyarakat, terutama di Kalimantan Barat, maka teknologi tersebut akan memberikan kemudahan dan manfaat yang berlipat ganda. Teknologi pertanian, seperti pembacaan kondisi tanah, suhu dan kelembapan udara, pengairan, dan pemupukan, dapat diterapkan di dalam *Greenhouse*. Dalam penelitian ini, tanaman Anggur dipantau pertumbuhannya karena anggur merupakan salah satu tanaman komoditas yang dikembangkan oleh masyarakat di Sungai Belidak Kabupaten Kubu Raya.

Greenhouse didesain khusus dengan bahan transparan seperti kaca atau plastik yang memungkinkan cahaya matahari masuk ke dalam bangunan dan menjaga suhu dan kelembapan di dalamnya. Tujuannya adalah untuk memberikan lingkungan yang ideal bagi tanaman agar dapat tumbuh dengan optimal. *Greenhouse* bertujuan menciptakan kondisi lingkungan yang stabil dan terkontrol bagi tanaman, sehingga dapat meningkatkan produktivitas dan kualitas hasil panen. [2],[3]. Dengan adanya *Greenhouse* dapat melindungi tanaman dari sinar matahari secara langsung. Selain dari itu, penggunaan *Internet of Things* (IoT) dalam *Greenhouse* dapat dilakukan dengan memanfaatkan node-node nirkabel yang terkoneksi dengan sensor-sensor yang digunakan untuk membaca kondisi tanaman. [4]. Setelah data dari sensor-sensor tersebut dikumpulkan, data tersebut kemudian akan dikirimkan ke internet untuk diproses dan disimpan oleh web server sebagai sistem informasi. Namun, semakin banyak jumlah node sensor yang digunakan, semakin besar pula penggunaan *bandwidth* yang dibutuhkan. Oleh karena itu, diperlukan mekanisme pengiriman data yang efisien untuk mengoptimalkan penggunaan *bandwidth* internet dalam penggunaan node sensor nirkabel.

MQTT (*Message Queueing Telemetry Transport*) adalah salah satu protokol untuk menghubungkan perangkat IoT dan menggantikan protokol HTTP [5]. MQTT bukan hanya menyediakan konektivitas yang aman untuk

perangkat IoT dalam skala besar, tetapi juga menyediakan manajemen penyimpanan dan analisis data di Cloud sehingga penggunaan *bandwidth* dapat dioptimalkan. Penggunaan MQTT dapat menjadi solusi dalam mengoptimalkan penggunaan *bandwidth* dan meminimalkan biaya yang diperlukan untuk kebutuhan internet.

Adapun penelitian yang pernah dilakukan berjudul "Sistem Monitoring Jaringan Sensor Node Berbasis Protokol MQTT" [6]. Node sensor dibuat menggunakan NodeMCU ESP8266 yang terhubung dengan berbagai jenis sensor untuk mengukur suhu, kelembapan udara, kadar gas, dan keberadaan api. Data sensor yang dihasilkan akan dikirimkan ke sistem pemantauan menggunakan protokol MQTT. Hasil pengujian menunjukkan bahwa protokol MQTT berhasil diimplementasikan dan data yang dikirimkan dari node sensor berhasil ditampilkan pada sistem pemantauan.

Penelitian terkait selanjutnya mengenai node nirkabel berjudul "Perancangan Konfigurasi Sensor Node dalam *Digital Precision Farming* Menggunakan Jaringan *Internet of Things*" [7]. Dalam penelitian ini, dijelaskan tentang implementasi konfigurasi perangkat keras sensor node yang digunakan dalam aplikasi digital precision farming. Hasil penelitian menunjukkan bahwa kemampuan sensor node untuk mengirim data melalui rintangan alam dan masalah energi listrik yang rendah sangat tergantung pada kualitas komponen node nirkabel yang digunakan.

Penelitian berikutnya terkait node nirkabel telah dilakukan berjudul "Node Nirkabel Berbasis *Internet of Things* Untuk Pemantauan Beban Arus Bolak Balik Pembangkit Listrik Tenaga Surya" [8]. Dalam pengujian yang dilakukan, sistem PLTS inverter di Laboratorium Teknik Elektro UNSOED dipantau menggunakan node sensor yang telah dibuat. Komponen sistem berhasil diuji dengan baik, dengan kesalahan pembacaan sensor arus sebesar 17.09mA (0.17%) dan kesalahan pembacaan sensor tegangan sebesar 2.102V (0.96%). Rata-rata persentase kesalahan perhitungan daya aktif adalah sebesar 3.026717%. Node nirkabel mampu mengirimkan data pemantauan dengan interval pengiriman data sekitar 32 detik dan memiliki tingkat keberhasilan pengiriman data sebesar 95.22%.

Berdasarkan isu yang telah dijelaskan, penelitian akan dilakukan dengan judul "Perancangan Node Nirkabel untuk Meningkatkan Efisiensi *Bandwidth* pada Smart *Greenhouse* Berbasis Protokol MQTT". Protokol MQTT akan diimplementasikan pada tanaman system IoT *Greenhouse* anggur untuk efisiensi penggunaan *bandwidth*. Dengan memanfaatkan protokol ini, diharapkan dapat mengefisienkan penggunaan *bandwidth* sehingga kecepatan pengiriman data semakin cepat dan mengurangi pemakaian paket data.

II. METODE PENELITIAN

A. *Internet of Things (IoT)*

Internet of Things (IoT) adalah konsep teknologi yang menghubungkan berbagai jenis perangkat fisik ke internet, sehingga perangkat-perangkat tersebut dapat saling berinteraksi dan berkomunikasi antara satu sama lain. IoT dapat digunakan dalam berbagai aplikasi, seperti rumah pintar, kota pintar, pertanian, industri, dan lain sebagainya.

Dalam konteks IoT, perangkat fisik seperti sensor, aktuator, dan perangkat elektronik lainnya dapat dikoneksikan ke internet melalui jaringan nirkabel atau kabel. Perangkat-perangkat tersebut dapat mengirimkan dan menerima data, seperti suhu, kelembapan, kecepatan, dan lain sebagainya, ke cloud atau server terpusat. Data tersebut kemudian dapat diproses dan dianalisis untuk mendapatkan informasi yang bermanfaat, seperti prediksi cuaca, analisis produksi, dan lain sebagainya. *Internet of Things (IoT)* [9][10].

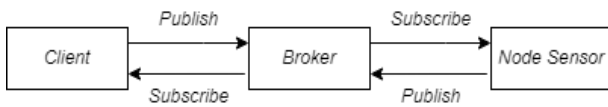
B. *Message Queuing Telemetry Transport (MQTT)*

Protokol MQTT (*Message Queuing Telemetry Transport*) adalah sebuah protokol jaringan yang didesain untuk mengirimkan pesan-pesan ringan yang memiliki ukuran kecil dan menggunakan *bandwidth* yang minim, sehingga sangat cocok digunakan pada lingkungan jaringan yang terbatas atau kurang stabil seperti *Internet of Things (IoT)*[11].

Protokol MQTT memungkinkan perangkat IoT untuk saling berkomunikasi dan bertukar data dengan cepat dan efisien, dengan menggunakan model publikasi dan langganan yang memungkinkan pesan dikirimkan ke beberapa penerima sekaligus. Selain itu, MQTT juga mendukung fitur QoS (Quality of Service) untuk menjamin pengiriman pesan yang andal dan akurat. MQTT banyak digunakan dalam industri IoT, seperti untuk menghubungkan sensor-sensor pada suatu sistem IoT dengan aplikasi atau platform yang berbasis cloud, mengirimkan notifikasi atau perintah-perintah pada perangkat IoT, serta dalam pengambilan keputusan atau analisis data di level edge. [12]. Sistem umum MQTT membutuhkan tiga komponen perangkat lunak utama yaitu [13]:

1. *MQTT Client* yang nantinya akan di install di device agar Arduino bisa memakai *pubsubclient* MQTT di perangkat PC.
2. *MQTT Broker* yang berfungsi untuk menangani *publish* dan *subscribe data*.
3. *MQTT Publisher (Node Sensor)* yang berfungsi membaca data dari sensor dan mengirimkannya ke *Broker*.

Alur komunikasi data MQTT antara *client*, *broker* dan *node sensor* dapat dilihat pada Gambar 1.



Gambar 1. Alur komunikasi data MQTT

C. Eclipse Paho MQTT Python Client Library

Eclipse Paho MQTT Python Client Library adalah sebuah library atau perpustakaan *Python* yang digunakan untuk mengembangkan aplikasi MQTT (*Message Queuing Telemetry Transport*) menggunakan bahasa pemrograman *Python*. Library ini dikembangkan oleh *Eclipse Foundation* dan tersedia secara gratis untuk digunakan [14].

D. Greenhouse

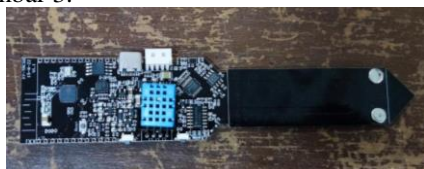
Greenhouse adalah sebuah struktur bangunan yang didesain khusus untuk menumbuhkan tanaman di dalamnya dengan lingkungan yang terkendali. *Greenhouse* atau rumah kaca biasanya terbuat dari bahan-bahan seperti kaca atau plastik transparan, dan dilengkapi dengan sistem ventilasi, pemanas, dan pencahayaan buatan untuk mengontrol suhu, kelembaban, dan cahaya di dalamnya. [15]. Adapun bangunan *Greenhouse* di Darul Fikri dapat dilihat pada Gambar 2.



Gambar 2. *Greenhouse* di Darul Fikri, Kubu Raya

E. NodeMCU ESP32

NodeMCU ESP32 adalah sebuah board mikrokontroler yang didasarkan pada sistem-on-chip (SoC) ESP32 dari Espressif Systems. Board ini dilengkapi dengan modul Wi-Fi dan Bluetooth, serta banyak fitur lainnya seperti sensor-sensor dan port I/O, yang memungkinkan pengembang untuk membuat berbagai macam proyek IoT dengan mudah. [16]. Adapun bentuk dari NodeMCU ESP32 yang digunakan untuk membaca suhu udara, kelembaban udara dan kelembaban tanah dapat dilihat pada Gambar 3.

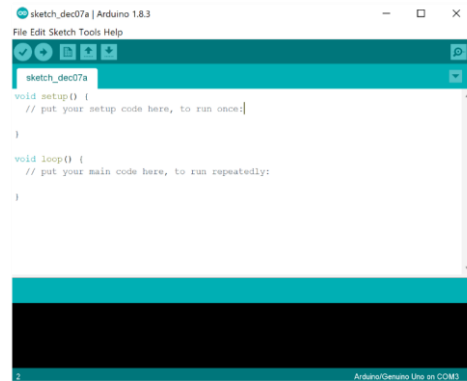


Gambar 3. NodeMCU ESP32

F. Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah sebuah lingkungan pengembangan terpadu untuk membuat, mengedit, dan mengupload kode program untuk

board mikrokontroler Arduino. Arduino IDE dapat digunakan untuk berbagai macam board mikrokontroler Arduino, seperti Arduino Uno, Nano, Mega, dan banyak lagi. [17]. *Software* arduino dapat di instal di beberapa operating system diantaranya Windows, MacOS, dan Linux. Tampilan antarmuka Arduino IDE dapat dilihat pada Gambar 4.



Gambar 4. Antarmuka Arduino IDE

G. Capacitive Soil Moisture Sensor

Capacitive soil moisture sensor adalah sensor yang digunakan untuk mengukur kadar air pada tanah dengan prinsip kapasitansi. Sensor ini terdiri dari dua elektroda yang ditempatkan pada tanah, dan kemudian diukur kapasitansinya oleh sebuah rangkaian elektronik. Prinsip kerja dari sensor ini adalah dengan mengukur perubahan kapasitansi antara dua elektroda yang terdapat pada sensor ketika tanah berubah kadar airnya. Kadar air pada tanah akan mempengaruhi nilai dielektrik tanah, yang kemudian mempengaruhi nilai kapasitansi antara dua elektroda pada sensor [18]. Sensor ini berfungsi untuk mendeteksi kelembaban tanah pada tanaman anggur. Sensor ini berbentuk mirip dengan pisau dan terbuat dari logam yang dapat menghantarkan tegangan listrik dengan nilai yang relatif kecil, yaitu sekitar 3,3 - 5V. Tegangan listrik tersebut kemudian diubah menjadi bentuk digital sehingga dapat diproses oleh sistem dengan lebih mudah dan akurat. Adapun bentuk sensor kelembaban tanah dapat dilihat pada Gambar 5.

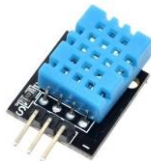


Gambar 5. Sensor kelembaban tanah

H. Sensor DHT11

Sensor DHT11 adalah sebuah sensor suhu dan kelembaban yang sederhana namun populer dan banyak digunakan dalam proyek-proyek elektronika dan *Internet of Things* (IoT). Sensor ini dapat mengukur suhu dengan

rentang antara 0°C hingga 50°C dengan akurasi sekitar ±2°C, serta kelembaban relatif dengan rentang antara 20% hingga 90% dengan akurasi sekitar ±5%. [18]. Bentuk dari sensor DHT11 dapat dilihat pada Gambar 6.



Gambar 6. Sensor DHT11

III. HASIL DAN PEMBAHASAN

A. Penerapan MQTT pada Node Sensor

Pada MQTT, protokol ini berbasis TCP (*Transmission Control Protocol*), dimana data secara *realtime* akan dikirimkan terus menerus tanpa memperdulikan data yang *missing/corrupt*. Pada Node sensor yang menggunakan MQTT, *broker* berperan sebagai perangkat sistem kendali yang memiliki peran dalam membaca data sensor (*subscribe*) dan mengirimkan Kembali ke *broker* (*publish*). Untuk terkoneksi ke *broker*, terlebih dahulu MQTT harus terkoneksi ke MQTT Server “*mqtt.eclipseproject.io*” menggunakan port 1883. Adapun baris program konfigurasinya dapat dilihat pada Kode Program 1.

```
Kode Program 1. Konfigurasi MQTT ke Server
const char* mqttServer = "mqtt.eclipseprojects.io";
const int mqttPort = 1883; //Port number
const char* mqttUser = "mqtt"; //User
const char* mqttPassword = "123"; //Password
```

Selanjutnya, node sensor akan membaca data dari *broker* yang digunakan sebagai acuan parameter data dan perintah ke node sensor. Baris program node sensor membaca data dari *broker* (*subscribe*) dapat dilihat pada Kode Program 2.

Kode Program 2 Konfigurasi Pembacaan Data dari Node Sensor ke Broker

```
client.publish("Esp32/sensor", "ESP32 Mqtt");
client.subscribe("Esp32/sensor");
```

Kemudian, node sensor akan mengirimkan data ke *broker* yang nantinya akan dibaca oleh server. Pada tahap ini, apabila jaringan dari node sensor ke *broker* terputus, maka server tetap akan membaca data terakhir yang dikirimkan oleh *broker*. Baris program node sensor mengirimkan data ke *broker* (*publish*) dapat dilihat pada Kode Program 3.

Kode Program 3. Konfigurasi Pengiriman Data dari Node Sensor ke Broker

```
client.publish("sensor/dht11/suhu",String(t).c_str());
client.publish("sensor/dht11/humidity",String(h).c_str());
client.publish("sensor/moisture/kelembabantanah",String(soil).c_str());
```

B. Penerapan MQTT pada Server

Pada penelitian ini, server bertugas membaca data yang ada di *broker* untuk disimpan ke dalam database.

Nantinya, data pada database akan dibaca dan ditampilkan ke halaman website secara *realtime* sehingga pengguna dapat melihat data hasil pembacaan node sensor. Agar dapat membaca data dari *broker*, terlebih dahulu server harus mengakses *broker* dan port nya. Adapun baris program server untuk mengakses *broker* dapat dilihat pada Kode Program 4.

Kode Program 4. Konfigurasi Server Mengakses Broker MQTT

```
$(document).ready(function(){
  client = new Paho.MQTT.Client("mqtt.eclipseprojects.io", 80,"web." + parseInt(Math.random() * 100, 10));
  client.onConnectionLost = onConnectionLost;
  client.onMessageArrived = onMessageArrived;

  var options = {
    useSSL: false,
    userName: "mqtt",
    password: "123",
    onSuccess:onConnect,
    onFailure:doFail
  }

  client.connect(options);
```

Setelah server terhubung ke *broker*, server akan membaca data yang dimiliki *broker* (*subscribe*). Data yang dibaca meliputi data suhu udara, kelembaban udara, kelembaban tanah dan waktu respon. Adapun proses pembacaan data *broker* MQTT data dilihat pada Kode Program 5.

Kode Program 5. Konfigurasi Pembacaan Data pada Broker oleh Server

```
function onConnect() {
  console.log("onConnect");
  //Isikan Data dari Esp//
  client.subscribe("sensor/dht11/suhu");
  client.subscribe("sensor/dht11/humidity");
  client.subscribe("sensor/moisture/kelembabantanah");
  client.subscribe("sensor/waktu/waktu");
  client.subscribe("esp/test");
  message = new Paho.MQTT.Message("Data Berhasil");
  message.destinationName = "ESP32/sensor";
  client.send(message);//kirim data ke mqtt

  pesan = new Paho.MQTT.Message("Data Berhasil");
  pesan.destinationName = "ESP32/sensor";
  client.send(pesan);//kirim data ke mqtt
}
```

Kemudian, data sensor dari *broker* disimpan ke dalam database dan ditampilkan ke website. Bentuk tampilan website dapat dilihat pada Gambar 7.



Gambar 7. Tampilan Pembacaan Data Node Sensor Menggunakan Protokol MQTT

C. Penerapan HTTP pada Node Sensor

Pada penerapan HTTP, protokol yang digunakan berbasis UDP (User Datagram Protocol), dimana data akan dikirimkan ke API jika semua frame paket data sudah terkumpul dan di encoding. Setelah data selesai di encoding, data tersebut dapat dikirimkan ke API dan

dibaca oleh server. Adapun konfigurasi akses API dapat dilihat pada Kode Program 6.

Kode Program 6. Konfigurasi Akses API oleh Node Sensor

```
String AlamatURL;
//equate with your computer's IP address and your directory application
AlamatURL = "http://10.100.2.94/mqtt&php/kirimdata.php?suhu=";
AlamatURL += String(t);
AlamatURL += "&kelembabanU=";
AlamatURL += String(h);
AlamatURL += "&kelembabanT=";
AlamatURL += String(soil);
```

Setelah itu, node sensor akan mengirimkan data sensor melalui akses API sehingga data bisa dibaca oleh server. Proses pengiriman data node sensor ke API dapat dilihat pada Kode Program 7.

Kode Program 7. Pengiriman Data ke API oleh Node Sensor

```
float t = dht.readTemperature(); // Read temperature as Fahrenheit then dht.readTemperature(true)
Serial.print("Temperature - "); Serial.println(t);

float h = dht.readHumidity();
Serial.print("Humidity - "); Serial.println(h);

float batt = readBattery();
Serial.print("Battery - "); Serial.println(batt);
```

D. Penerapan HTTP pada Server

Pada tahap ini, server akan membaca data dari API yang diberikan dari Node Sensor. Data tersebut akan disimpan ke dalam database dan ditampilkan ke website. Bentuk konfigurasi ke API dapat dilihat pada Kode Program 8.

Kode Program 8. Konfigurasi Server ke API

```
<?php
$sawal = microtime(true);
//koneksi ke database
$koneksi = mysqli_connect("http://10.100.2.94/mqtt&php/kirimdata.php?", "root", "", "mqtt");
```

Setelah server terkoneksi ke API, server akan membaca data-data dari API dan disimpan ke dalam database. Proses pembacaan data pada API dapat dilihat pada Kode Program 9.

Kode Program 9. Pembacaan Data pada API oleh Server

```
$suhu = $_GET['suhu']; //baca data yang dikirim esp32
for ($a=1; $a<=10000000; $a++)
{
    $suhu;
}

$akhir1 =microtime(true);
$lama1 = $akhir1-$awal;
$konversi1 = number_format($lama1, 2);

$KelembabanUdara=$_GET['kelembabanU']; //baca daya yang dikirim esp32
for ($b=1; $b<=10000000; $b++)
{
    $KelembabanUdara;
}

$akhir2 =microtime(true);
$lama2 = $akhir2-$awal;
$konversi2 = number_format($lama2, 2);
$KelembabanTanah =$_GET['kelembabanT']; //baca daya yang dikirim esp32
for ($c=1; $c<=10000000; $c++)
{
    $KelembabanTanah;
}

$akhir3 =microtime(true);
$lama3 = $akhir3-$awal;
$konversi3 = number_format($lama3, 2);
```

Kemudian, data sensor yang didapatkan dari API, disimpan ke database dan ditampilkan ke halaman website. Adapun tampilan pengguna halaman website dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Pembacaan Data Node Sensor Menggunakan Protokol HTTP

E. Pengujian Waktu Respon MQTT dan HTTP per Jam

Dalam pengujian ini, dilakukan penghitungan waktu respons Node Sensor saat mengirimkan data sensor dari awal sampai data tersebut tiba di server database setiap jam. Informasi tentang waktu respons pengiriman data sensor per jam menggunakan protokol MQTT terdapat di Tabel 1, sementara informasi waktu respons pengiriman data sensor per jam menggunakan protokol HTTP terdapat di Tabel 2.

TABEL 1
RATA-RATA WAKTU RESPON MQTT PENGIRIMAN DATA SENSOR PER JAM

No	Suhu (°C)	Kel. Udara (%)	Kel. Tanah (%)	Respon Suhu (detik)	Respon Kel. Udara (detik)	Respon Kel. Tanah (detik)	Rata-rata Respon (detik)
1	30.90	40%	15%	0.04	0.03	0.03	0.03
2	31.3	41%	14%	0.04	0.03	0.03	0.03
3	31.1	41%	14%	0.04	0.05	0.03	0.04
4	31.6	34%	18%	0.05	0.03	0.03	0.04
5	32.1	32%	18%	0.05	0.05	0.04	0.05
6	32.6	32%	18%	0.1	0.03	0.03	0.05
7	32.6	33%	17%	0.11	0.03	0.06	0.07
8	32.2	33%	17%	0.12	0.03	0.02	0.06
9	32.2	33%	17%	0.12	0.03	0.02	0.06
10	32.3	45%	11%	0.03	0.37	0.03	0.14
11	32.3	45%	11%	0.03	0.37	0.03	0.14
12	29	55%	34%	0.04	0.05	0.03	0.04
13	27.2	35%	10%	0.04	0.05	0.06	0.05
14	27.2	38%	10%	0.05	0.06	0.08	0.06
15	27.8	38%	14%	0.11	0.03	0.05	0.06
16	28.9	35%	14%	0.12	0.04	0.05	0.07
17	28.2	55%	13%	0.1	0.04	0.05	0.06
18	29.4	55%	13%	0.12	0.12	0.36	0.20
19	29.8	42%	13%	0.2	0.05	0.03	0.09
20	27.3	47%	55%	0.03	0.05	0.05	0.04
21	27.8	45%	53%	0.11	0.11	0.12	0.11
22	28.9	45%	48%	0.05	0.05	0.06	0.05
23	29.8	35%	38%	0.07	0.06	0.06	0.06
24	29.8	35%	33%	0.12	0.03	0.02	0.06
25	27.4	30%	25%	0.2	0.23	0.32	0.25
26	28.2	42%	37%	0.12	0.03	0.03	0.06
27	26.8	49%	35%	0.02	0.04	0.06	0.04
28	26.9	46%	35%	0.11	0.12	0.12	0.12
29	27.6	44%	33%	0.23	0.23	0.3	0.25
30	27.8	43%	33%	0.03	0.04	0.04	0.04
31	27.6	42%	30%	0.06	0.06	0.07	0.06
32	28.6	40%	44%	0.03	0.03	0.03	0.03
33	27.6	41%	43%	0.03	0.05	0.06	0.05
34	27.8	45%	38%	0.05	0.06	0.04	0.05
35	28.4	40%	37%	0.11	0.12	0.22	0.15
Rata-rata Respon				0.08	0.08	0.08	0.08

TABEL II
RATA-RATA WAKTU RESPON HTTP PENGIRIMAN DATA SENSOR PER JAM

No	Suhu (°C)	Kel. Udara (%)	Kel. Tanah (%)	Respon Suhu (detik)	Respon Kel. Udara (detik)	Respon Kel. Tanah (detik)	Rata-rata Respon (detik)
1	31.4	40%	34%	0.13	0.3	0.47	0.30
2	30.9	43%	33%	0.16	0.37	0.52	0.35
3	30.9	44%	33%	0.17	0.29	0.45	0.30
4	31.7	37%	0%	0.43	0.81	1.15	0.80
5	32	37%	0%	0.24	0.45	0.57	0.42
6	31.7	34%	55%	0.33	0.46	0.61	0.47
7	32.9	32%	41%	0.2	0.44	0.68	0.44
8	32.9	32%	41%	0.31	0.45	0.62	0.46
9	33.1	34%	40%	0.31	0.49	0.6	0.47
10	30.4	40%	12%	0.19	0.3	0.42	0.30
11	26.3	50%	15%	0.34	0.96	1.49	0.93
12	26.1	50%	13%	0.19	0.32	0.43	0.31
13	25.2	55%	43%	0.16	0.2	0.23	0.20
14	25.2	55%	43%	0.33	0.35	0.4	0.36
15	26.8	48%	42%	0.23	0.23	0.25	0.24
16	26.9	46%	42%	0.21	0.23	0.28	0.24
17	27.2	43%	39%	0.26	0.28	0.3	0.28
18	27.4	42%	39%	0.31	0.33	0.45	0.36
19	27.4	42%	39%	0.33	0.46	0.61	0.47
20	28.2	40%	35%	0.18	0.26	0.31	0.25
21	26.8	48%	42%	0.36	0.46	0.54	0.45
22	26.9	46%	42%	0.14	0.28	0.49	0.30
23	27.2	43%	39%	0.24	0.29	0.32	0.28
24	27.4	42%	39%	0.26	0.36	0.71	0.44
25	27.4	42%	39%	0.44	0.51	0.78	0.58
26	28.2	40%	35%	0.38	0.46	0.53	0.46
27	26.8	48%	42%	0.23	0.28	0.43	0.31
28	26.9	46%	42%	0.43	0.46	0.89	0.59
29	27.2	43%	39%	0.36	0.46	0.68	0.50
30	27.4	42%	39%	0.19	0.27	0.35	0.27
31	27.4	42%	39%	0.26	0.29	0.36	0.30
32	28.2	40%	35%	0.25	0.49	0.67	0.47
33	27.4	42%	39%	0.21	0.28	0.32	0.27
34	27.4	42%	39%	0.46	0.65	0.82	0.64
35	28.2	40%	35%	0.23	0.27	0.33	0.28
Rata-rata Respon				0.27	0.39	0.54	0.40

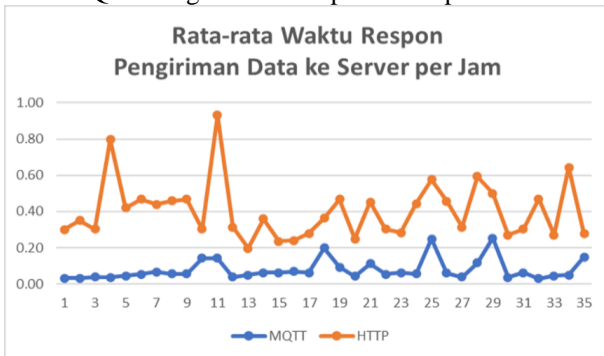
TABEL III
RATA-RATA WAKTU RESPON MQTT PENGIRIMAN DATA SENSOR PER 10 MENIT

No	Suhu (°C)	Kel. Udara (%)	Kel. Tanah (%)	Respon Suhu (detik)	Respon Kel. Udara (detik)	Respon Kel. Tanah (detik)	Rata-rata Respon (detik)
1	31.2	35	15	0.03	0.04	0.03	0.03
2	30.9	36	16	0.03	0.04	0.03	0.03
3	27.7	35	10	0.04	0.05	0.03	0.04
4	27.3	37	12	0.03	0.04	0.03	0.03
5	28	40	16	0.13	0.08	0.1	0.10
6	27.8	38	24	0.04	0.06	0.03	0.04
7	28.2	39	47	0.03	0.03	0.03	0.03
8	28.2	39	58	0.03	0.05	0.05	0.04
9	28.5	39	58	0.03	0.05	0.05	0.04
10	28.5	36	58	0.03	0.05	0.05	0.04
11	29.4	36	58	0.03	0.05	0.05	0.04
12	29.4	36	58	0.03	0.05	0.05	0.04
Rata-rata Respon				0.04	0.05	0.04	0.04

TABEL IV
RATA-RATA WAKTU RESPON HTTP PENGIRIMAN DATA SENSOR PER 10 MENIT

No	Suhu (°C)	Kel. Udara (%)	Kel. Tanah (%)	Respon Suhu (detik)	Respon Kel. Udara (detik)	Respon Kel. Tanah (detik)	Rata-rata Respon (detik)
1	28.6	41	16	0.45	0.66	0.83	0.65
2	28.5	41	16	0.17	0.35	0.53	0.35
3	29.1	44	6	0.18	0.31	0.5	0.33
4	28.8	42	8	0.15	0.36	0.51	0.34
5	27.7	38	8	0.23	0.43	0.83	0.50
6	27.2	40	8	0.15	0.34	0.49	0.33
7	27	40	8	0.2	0.41	0.57	0.39
8	27.1	45	9	0.35	0.79	1.2	0.78
9	26.8	41	8	0.27	0.5	1.09	0.62
10	26.7	41	16	0.42	1.09	1.33	0.95
11	27.2	44	22	0.19	0.34	0.48	0.34
12	27.5	45	54	0.21	0.37	0.59	0.39
Rata-rata Respon				0.25	0.50	0.75	0.50

Dari Pengujian ini, didapatkan perbandingan rata-rata waktu respon per jam antara MQTT dengan HTTP. Adapun perbandingan rata-rata waktu respon per jam antara MQTT dengan HTTP dapat dilihat pada Gambar 9.

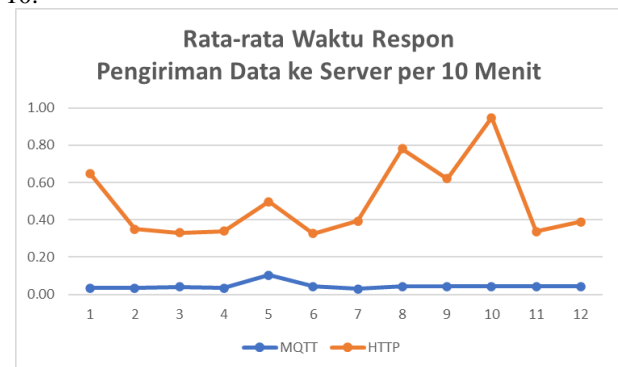


Gambar 9. Rata-Rata Waktu Respon MQTT dan HTTP per Jam

F. Pengujian Waktu Respon MQTT dan HTTP per 10 Menit

Pada pengujian ini, dihitung waktu respon dari Node Sensor mengirimkan data sensor hingga sampai di database server setiap 10 menit. Adapun waktu respon pengiriman data sensor per 10 menit menggunakan protokol MQTT dapat dilihat pada Tabel 3. Sedangkan waktu respon pengiriman data sensor per 10 menit menggunakan protokol HTTP dapat dilihat pada Tabel 4.

Dari Pengujian ini, didapatkan perbandingan rata-rata waktu respon per 10 menit antara MQTT dengan HTTP. Adapun perbandingan rata-rata waktu respon per 10 menit antara MQTT dengan HTTP dapat dilihat pada Gambar 10.



Gambar 10. Rata-Rata Waktu Respon MQTT dan HTTP per 10 Menit

G. Pengujian Waktu Respon MQTT dan HTTP per Detik

Pada pengujian ini, dihitung waktu respon dari Node Sensor mengirimkan data sensor hingga sampai di database server setiap detik. Adapun waktu respon pengiriman data sensor per detik menggunakan protokol MQTT dapat dilihat pada Tabel 5. Sedangkan waktu respon pengiriman data sensor per detik menggunakan protokol HTTP dapat dilihat pada Tabel 6.

TABEL V
RATA-RATA WAKTU RESPON MQTT PENGIRIMAN DATA SENSOR PER DETIK

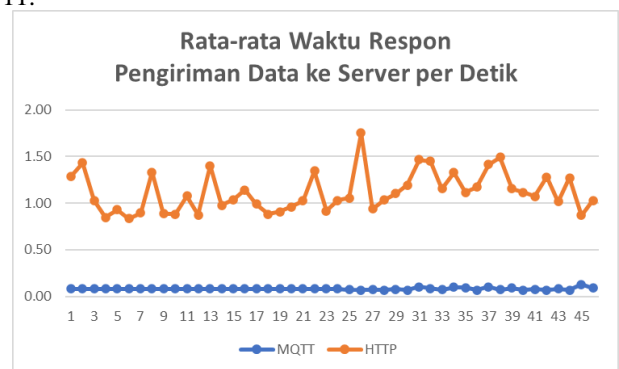
No	Suhu (°C)	Kel. Udara (%)	Kel. Tanah (%)	Respon Suhu (detik)	Respon Kel. Udara (detik)	Respon Kel. Tanah (detik)	Rata-rata Respon (detik)
1	31.6	33	21	0.1	0.07	0.07	0.08
2	31.6	33	21	0.1	0.07	0.07	0.08
3	31.6	33	21	0.1	0.07	0.07	0.08
4	31.6	33	21	0.1	0.07	0.07	0.08
5	31.6	33	21	0.1	0.07	0.07	0.08
6	31.6	33	21	0.1	0.07	0.07	0.08
7	31.6	33	21	0.1	0.07	0.07	0.08
8	31.6	33	21	0.1	0.07	0.07	0.08
9	31.6	33	21	0.1	0.07	0.07	0.08
10	31.6	33	21	0.1	0.07	0.07	0.08
11	31.6	33	21	0.1	0.07	0.07	0.08
12	31.6	33	21	0.1	0.07	0.07	0.08
13	31.6	33	21	0.1	0.07	0.07	0.08
14	31.6	33	21	0.1	0.07	0.07	0.08
15	31.6	33	21	0.1	0.07	0.07	0.08
16	31.6	33	21	0.1	0.07	0.07	0.08
17	31.6	33	21	0.1	0.07	0.07	0.08
18	31.6	33	21	0.1	0.07	0.07	0.08
19	31.6	33	21	0.1	0.07	0.07	0.08
20	31.6	33	21	0.1	0.07	0.07	0.08
21	31.6	33	21	0.1	0.07	0.07	0.08
22	31.6	33	21	0.1	0.07	0.07	0.08
23	31.6	33	21	0.1	0.07	0.07	0.08
24	31.6	33	21	0.1	0.07	0.07	0.08
25	28.1	38	24	0.07	0.08	0.07	0.07
26	28.1	39	24	0.07	0.07	0.07	0.07
27	28.5	39	24	0.07	0.08	0.07	0.07
28	28.5	39	24	0.07	0.07	0.07	0.07
29	29	39	24	0.07	0.08	0.08	0.08
30	29	40	24	0.07	0.07	0.07	0.07
31	28.1	38	24	0.11	0.13	0.07	0.10
32	28.1	38	24	0.1	0.08	0.07	0.08
33	28.2	38	24	0.07	0.08	0.07	0.07
34	28.2	38	25	0.07	0.13	0.1	0.10
35	28.2	38	24	0.1	0.08	0.1	0.09
36	28.2	38	24	0.07	0.07	0.07	0.07
37	28.2	38	24	0.15	0.09	0.07	0.10
38	28.2	38	24	0.08	0.07	0.07	0.07
39	28.2	38	24	0.1	0.08	0.09	0.09
40	28.7	38	24	0.07	0.07	0.07	0.07
41	28.7	38	24	0.08	0.08	0.07	0.08
42	28.2	38	24	0.07	0.07	0.07	0.07
43	28.2	38	24	0.09	0.08	0.07	0.08
44	28.2	38	24	0.07	0.07	0.07	0.07
45	28.2	38	24	0.11	0.1	0.18	0.13
46	28	38	24	0.08	0.09	0.1	0.09
Rata-rata Respon				0.09	0.08	0.08	0.08

TABEL VI
RATA-RATA WAKTU RESPON HTTP PENGIRIMAN DATA SENSOR PER DETIK

No	Suhu (°C)	Kel. Udara (%)	Kel. Tanah (%)	Respon Suhu (detik)	Respon Kel. Udara (detik)	Respon Kel. Tanah (detik)	Rata-rata Respon (detik)
1	29.6	33	5	0.78	1.29	1.78	1.28
2	29.4	33	5	0.74	1.49	2.07	1.43
3	29.4	33	5	0.39	0.93	1.77	1.03
4	29.5	33	5	0.39	0.84	1.31	0.85
5	29.5	33	4	0.5	0.95	1.34	0.93
6	29.5	33	4	0.42	0.8	1.28	0.83
7	29.5	33	5	0.43	0.93	1.32	0.89
8	29.6	33	4	0.62	1.39	1.99	1.33
9	29.6	33	5	0.4	0.93	1.33	0.89
10	29.6	33	5	0.37	0.85	1.43	0.88
11	29.6	33	5	0.63	1.09	1.51	1.08
12	29.6	33	5	0.41	0.87	1.34	0.87
13	29.7	33	5	0.38	1.67	2.16	1.40
14	29.6	33	5	0.4	0.89	1.64	0.98
15	29.6	33	5	0.59	1.02	1.5	1.04
16	29.6	33	5	0.69	1.09	1.65	1.14
17	29.6	33	5	0.55	0.96	1.46	0.99
18	29.6	33	4	0.43	0.91	1.31	0.88
19	29.8	34	5	0.43	0.87	1.42	0.91
20	29.8	34	5	0.51	0.93	1.44	0.96
21	29.8	34	4	0.5	0.92	1.66	1.03
22	29.9	33	4	0.68	1.46	1.9	1.35
23	29.9	33	4	0.44	0.92	1.39	0.92
24	29.9	33	5	0.5	1.02	1.57	1.03

No	Suhu (°C)	Kel. Udara (%)	Kel. Tanah (%)	Respon Suhu (detik)	Respon Kel. Udara (detik)	Respon Kel. Tanah (detik)	Rata-rata Respon (detik)
25	29.8	33	5	0.58	0.99	1.59	1.05
26	29.8	33	5	1.13	1.85	2.28	1.75
27	29.9	33	5	0.48	0.93	1.4	0.94
28	29.9	33	5	0.6	0.99	1.51	1.03
29	29.8	33	5	0.58	1.07	1.66	1.10
30	30.1	33	5	0.67	1.15	1.76	1.19
31	29.9	33	5	0.54	1.56	2.3	1.47
32	30.1	33	5	0.65	1.45	2.25	1.45
33	30	33	5	0.51	1.28	1.67	1.15
34	30	33	5	0.54	1.4	2.05	1.33
35	30.1	33	5	0.57	1.18	1.6	1.12
36	30.1	33	5	0.57	1.21	1.74	1.17
37	30.3	33	5	0.6	1.37	2.27	1.41
38	30	33	5	0.76	1.62	2.1	1.49
39	30.1	33	5	0.54	1.06	1.87	1.16
40	30.1	33	5	0.56	1.07	1.72	1.12
41	30.1	33	5	0.58	1.08	1.56	1.07
42	30.1	33	5	0.68	1.34	1.82	1.28
43	30.3	33	5	0.53	1	1.52	1.02
44	30.3	33	5	0.56	1.32	1.93	1.27
45	30.3	32	5	0.38	0.91	1.32	0.87
46	30.3	32	5	0.54	1.02	1.53	1.03
Rata-rata Respon				0.55	1.13	1.67	1.12

Dari Pengujian ini, didapatkan perbandingan rata-rata waktu respon per detik antara MQTT dengan HTTP. Adapun perbandingan rata-rata waktu respon per detik antara MQTT dengan HTTP dapat dilihat pada Gambar 11.



Gambar 11. Rata-Rata Waktu Respon MQTT dan HTTP per Detik

H. Pembahasan

Dari hasil pengujian, didapatkan beberapa hasil perbandingan antara MQTT dengan HTTP menggunakan interval waktu pengiriman yang berbeda. Pada pengiriman data per jam, secara keseluruhan MQTT masih lebih stabil dan waktu responnya lebih kecil dibandingkan HTTP. Walaupun ada beberapa pengiriman data terjadi lonjakan waktu respon di protokol MQTT. Pada pengiriman data per jam, secara keseluruhan MQTT masih lebih stabil dan waktu responnya lebih kecil dibandingkan HTTP. Kemudian, pada pengiriman data per jam, secara keseluruhan MQTT masih lebih stabil dan waktu responnya lebih kecil dibandingkan HTTP. Walaupun server harus menerima data dengan sangat cepat, performa pengiriman data menggunakan MQTT lebih stabil dibandingkan dengan HTTP saat data dikirimkan ke server. Hasil dari perbandingan ketiga interval waktu adalah Perbandingan antara pengiriman data ke server setiap 1 jam adalah 0.08 detik (MQTT) dan 0.4 detik (HTTP). Perbandingan antara pengiriman data ke server

setiap 10 menit adalah 0.04 detik (MQTT) dan 0.5 detik (HTTP). Perbandingan antara pengiriman data ke server setiap detik adalah 0.08 detik (MQTT) dan 1.12 detik (HTTP).

IV. KESIMPULAN

Pada penelitian ini, protokol MQTT menggunakan Eclipse Paho Python Client Library yang mana berperan sebagai *Broker (Open Source)* untuk menerima data dari Node Sensor (*Publish*) dan mengirimkan data ke Client/Server (*Subscribe*) menggunakan jalur komunikasi TCP. Hasil perbandingan waktu respon antara MQTT dengan HTTP untuk pengiriman data ke server setiap 1 jam adalah 0.08 detik (MQTT) dan 0.4 detik (HTTP). Kemudian, perbandingan antara pengiriman data ke server setiap 10 menit adalah 0.04 detik (MQTT) dan 0.5 detik (HTTP). Selanjutnya, perbandingan antara pengiriman data ke server setiap detik adalah 0.08 detik (MQTT) dan 1.12 detik (HTTP). Jadi, pengiriman data menggunakan protokol MQTT lebih stabil dan waktu responnya lebih cepat dibandingkan dengan HTTP.

REFERENSI

- [1] C. A. González-Amarillo et al., "An IoT-Based Traceability System for Greenhouse Seedling Crops," in Proc. IEEE Access, 2018, pp. 67528-67535.
- [2] H. R. Lubis, "Rancang Bangun Smart System Ruang Greenhouse Berbasis IoT dengan Menggunakan Arduino UNO," Jurusan Teknik Elektro UII Yogyakarta, 2020.
- [3] A. Wardani, "Purwarupa perangkat IoT untuk smart greenhouse berbasis mikrokontroler", in Proc. E-Proceeding of Engineering, 2018, pp. 3859-3875.
- [4] Ristian et al., "Sistem Monitoring Smart Greenhouse pada Lahan Terbatas Berbasis Internet of Things (IoT)", Jurnal Edukasi & Penelitian Informatika, vol. 8, no. 1, 2022.
- [5] E. F. Fernanda, "Rancang Bangun Sistem Monitoring dan Kontrol Smart Home Menggunakan Protokol MQTT," Fakultas Teknologi dan Informatika Universitas Dinamika, 2022.
- [6] M. Diono et al., "Sistem Monitoring Jaringan Sensor Node Berbasis Protokol MQTT," Jurnal ELEMENTER, vol. 7, no. 2, pp. 120-126, 2021.
- [7] R. Huda and T. Y. Putro, "Perancangan Konfigurasi Sensor Node dalam Digital Precision Farming Menggunakan Jaringan Internet of Things," Bandung, 2018.
- [8] A. W. W. Winasis et al., "Node Nirkabel Berbasis Internet of Things Untuk Pemantauan Beban Arus Bolak Balik Pembangkit Listrik Tenaga Surya," Yogyakarta, 2016.
- [9] I. Kurniawan and L. Andraini, "Implementasi Internet of Things (IOT) untuk kesehatan", Jurnal Ilmiah Pendidikan Teknik Elektro, vol. 2(10), 2022.
- [10] W. N. Alimyaningtias and Syaddam, "Penerapan IoT untuk Optimalisasi Penjagaan Kadar Air Dalam Tanah", Jurnal Teknologi dan Sistem Tertanam, vol. 3, no. 2, 2022.
- [11] C. F. Permatasari and H. Dhika, "Optimasi Jalur Transfer Data dari HTTP menjadi MQTT pada IoT menggunakan Cloud Services," JISA, vol. 1, no. 2, pp. 67-72, 2018.
- [12] E. F. D. Permatasari et al., "Analisis Perbandingan Performansi MQTT dan HTTP pada Platform IoT Node-Red," eProceedings Eng., vol. 6, no. 2, 2019.
- [13] B. M. Susanto et al., "Implementasi MQTT Protocol pada Smart Home Security Berbasis Web," Jurnal Informatika Polinema, vol. 4, no. 3, pp. 201-205, 2018.
- [14] "Eclipse Foundation Paho," Eclipse Foundation, 15 03 2012. [Online]. Available: <https://www.eclipse.org/paho/>. [Accessed 02 10 2022].
- [15] U. Ristian et al., "Sistem Monitoring Smart Greenhouse pada Lahan Terbatas Berbasis Internet of Things (IoT)," Jurnal Edukasi dan Penelitian Informatika, vol. 8, no. 1, pp. 87-94, 2022.
- [16] "ESP-32 WROM-32 Datasheet v2.9," Ekspressif Systems, 2019.
- [17] "Arduino," 2020. [Online]. Available: <https://www.arduino.cc/>. [Accessed 28 04 2020].
- [18] M. B. K. Putra, "IoT Based Smart Agriculture Using Fuzzy Logic", Jurnal Comaise, vol. 06, no. 02.
- [19] "DTH11 Humidity & Temperature Sensor," D-Robotics, UK, 2010.