

# Implementasi Protokol TCP dan UDP pada Sistem Monitoring dan Otomasi Rumah Jamur Berorientasi WSN

## Implementation of TCP and UDP Protocols on WSN-Oriented Mushroom House Monitoring and Automation System

Audi Diva Fakhru<sup>1</sup>, Nurul Fahmi Arief Hakim<sup>2</sup>, Agus Heri Setya Budi<sup>3\*</sup>

<sup>1,2,3</sup>Departemen Pendidikan Teknik Elektro, Universitas Pendidikan Indonesia

Jl. Dr. Setiabudi No.229, Isola, Kec. Sukasari, Kota Bandung

audidivaf@upi.edu<sup>1</sup>, nurulfahmi@upi.edu<sup>2</sup>, agusheri@upi.edu<sup>3\*</sup>

**Abstrak** – Sistem monitoring dan otomasi rumah jamur sangat diperlukan untuk budidaya jamur. Sistem ini membutuhkan suatu protokol yang dapat membantu pengiriman data yang handal agar keadaan lingkungan dapat terpantau dengan baik. Ketika jaringan dengan protokol TCP terganggu, monitoring kondisi kumbung jamur akan terhambat, berdampak pada pertumbuhan jamur yang tidak optimal. Penelitian ini bertujuan untuk mengimplementasikan protokol TCP dan UDP pada sistem monitoring dan otomasi rumah jamur menggunakan *iot* berorientasi *Wireless Sensor Network (WSN)*. Sistem menggunakan dua jenis *mikrokontroler Wemos D1 mini ESP8266* dan *Arduino UNO*. Untuk satu *mikrokontroler* akan ditempatkan di dalam kumbung jamur untuk mengambil data lingkungan kumbung jamur yang diberi nama *node sensor* dan satu lagi berada di luar kumbung jamur yang diberi nama *gateway*. *Node sensor* akan dipasang *sensor suhu dan kelembaban udara, soil moisture sensor, sensor intensitas cahaya, dan sensor CO2*. Proses pengiriman data dari *node sensor* ke *gateway* menggunakan protokol UDP dan dari *gateway* ke server menggunakan protokol TCP. Hasil yang diperoleh adalah sistem monitoring dan otomasi rumah jamur dapat berjalan dengan baik. Hasil perbandingan menunjukkan bahwa protokol UDP memiliki delay yang lebih kecil dibanding TCP yaitu dengan perbedaan 96.153 ms. Tetapi untuk parameter throughput protokol TCP memiliki nilai lebih besar dibanding protokol UDP. Untuk koneksi dari *node* ke *gateway* lebih baik menggunakan protokol UDP karena pengiriman yang cepat dan data yang sedikit dikirim sehingga lebih efektif. Tetapi pada *gateway* ke server lebih baik menggunakan protokol TCP, karena data yang dikirimkan bisa banyak dan data harus terjamin sampai pada server.

**Kata Kunci:** *Node MCU, QoS, TCP, UDP, WSN.*

**Abstract** – The utilization of mushroom house monitoring and automation systems is crucial in the context of mushroom growing. The successful monitoring of environmental conditions necessitates the implementation of a protocol that facilitates the transmission of dependable data. In the event of a disruption in the network utilizing the Transmission Control Protocol (TCP), the ability to monitor the status of the mushroom house would be impeded, thus leading to suboptimal mushroom growth. The objective of this project is to apply the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) in a monitoring and automation system for mushroom houses. This will be achieved by utilizing a *Wireless Sensor Network (WSN)* that is specifically designed for *Internet of Things (IoT)* applications. The system employs two distinct microcontrollers, namely the *Wemos D1 small ESP8266* and the *Arduino UNO*. A sensor node, referred to as the microcontroller, will be positioned within the mushroom home to collect environmental data. Additionally, a gateway microcontroller will be situated outside the mushroom house.

TELKA, Vol.9, No.2, November 2023, pp. 130~144

ISSN (e): 2540-9123

ISSN (p): 2502-1982

The sensor nodes will be equipped with temperature and humidity sensors, soil moisture sensors, light intensity sensors, and CO<sub>2</sub> sensors. The transmission of data from the sensor node to the gateway is facilitated by the User Datagram Protocol (UDP), while the transmission from the gateway to the server is facilitated by the Transmission Control Protocol (TCP). The findings indicate that the mushroom house monitoring and automation system operates effectively. The comparison results show that the UDP protocol has a smaller delay than TCP, with a difference of 96.153 ms. In contrast, the throughput parameter of the TCP protocol has a higher value compared to that of the UDP protocol. When establishing connections between nodes and gateways, it is advisable to utilize the User Datagram Protocol (UDP) due to its advantages in terms of expedited delivery and efficient data transmission, as it involves less data overhead. However, it is advisable to utilize the TCP protocol for the gateway to the server, as it ensures reliable delivery of large volumes of data to the server.

**Keywords:** Node MCU, QoS, TCP, UDP, WSN.

## 1. Pendahuluan

Kecepatan pengiriman data merupakan suatu faktor yang sangat penting dalam perkembangan komunikasi data. Komunikasi data merupakan suatu sarana untuk memungkinkan perangkat – perangkat dapat saling berkomunikasi satu sama lain [1]. *Wireless Sensor Network* (WSN) merupakan teknologi komunikasi data yang terdiri dari teknologi sensor dan komunikasi nirkabel untuk memantau dan mengambil data kondisi dari suatu area tertentu [2]. WSN dipandang sebagai kontributor penting untuk komputasi *Internet of Things* (IoT) [3]. IoT adalah suatu konsep dimana sebuah objek dapat saling bertukar informasi menggunakan jaringan internet tanpa adanya interaksi komputer dengan manusia [4]. Protokol merupakan hal penting dalam pengiriman data IoT. Protokol adalah suatu aturan atau prosedur untuk pengiriman data antar komputer. Ada beberapa jenis protokol yang bisa digunakan, beberapa diantaranya yaitu *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) [5]. Penelitian TCP dan UDP telah dilakukan oleh [6] yang berhasil menguji kinerja protokol transport TCP dan UDP melalui jaringan seluler *Narrowband Internet of Things* (NB-IoT). Hasil yang diperoleh yaitu TCP memiliki kinerja yang rendah dengan tingkat *packet loss* hingga 90%. Sedangkan UDP memiliki kinerja lebih baik dengan tingkat *packet loss* hanya 3%.

Salah satu contoh implementasi IoT adalah dalam sistem pemantauan lingkungan budidaya jamur di rumah. Praktik budidaya jamur semakin meluas di masyarakat modern, di mana jamur telah menjadi pilihan makanan yang diminati [7]. Pengetahuan mengenai budidaya jamur telah tersebar luas di masyarakat, bahkan jamur telah menjadi komoditas perdagangan yang signifikan [8]. Secara umum, jamur tumbuh paling baik di daerah dengan ketinggian sekitar 700 - 800 meter di atas permukaan laut [9]. Meskipun begitu, budidaya jamur juga dimungkinkan di daerah dengan ketinggian rendah, asalkan lingkungan budidaya diatur dengan memperhatikan kebutuhan optimal pertumbuhan jamur [10]. Faktor-faktor yang mempengaruhi pertumbuhan jamur termasuk suhu dan kelembaban udara, kelembaban media tanam, kadar CO<sub>2</sub>, dan intensitas cahaya [11]. Mengendalikan lingkungan kumbung jamur secara manual memiliki kendala terkait waktu dan kesulitan. Biasanya, kumbung jamur perlu dijaga dalam kondisi gelap agar pertumbuhan jamur berjalan optimal [12]. Namun, ketika pengendalian dilakukan secara manual, efisiensi dan konsistensi sulit dipertahankan [13]. Salah satu solusi untuk mengatasi kendala ini adalah dengan menerapkan teknologi IoT dan prinsip jaringan sensor nirkabel (WSN) dalam sistem pemantauan dan otomasi lingkungan budidaya jamur. Penelitian sebelumnya, seperti yang dilakukan oleh [14], telah mengembangkan sistem rumah jamur menggunakan protokol TCP pada perangkat ESP8266. Namun, ada risiko bahwa jika jaringan dengan protokol TCP terganggu, monitoring kondisi kumbung jamur akan terhambat, berdampak pada pertumbuhan jamur yang tidak optimal.

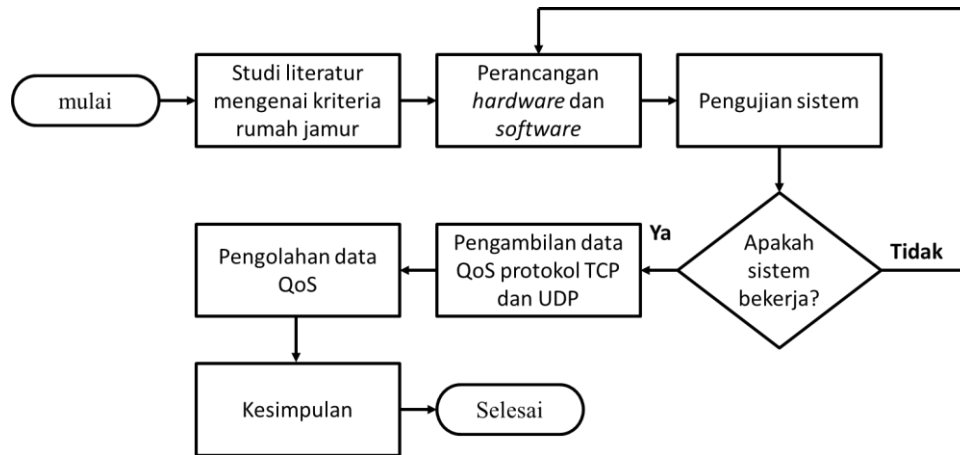
Untuk mengatasi tantangan ini, konsep IoT dapat diterapkan dengan menggunakan node sensor untuk mengumpulkan data lingkungan dan gateway sebagai pusat penerima data serta pengatur kondisi lingkungan. Pendekatan ini memungkinkan pengendalian yang lebih efektif dan tepat waktu tanpa memerlukan campur tangan manusia secara langsung. Sistem pengawasan

rumah jamur menggunakan IoT telah dilakukan oleh [14]-[16] yang menyajikan sistem pemantauan rumah jamur berbasis IoT. Sistem terdiri dari sejumlah sensor yang mengukur suhu, kelembaban, intensitas cahaya, dan tingkat kelembaban tanah lingkungan tumbuh jamur. Data dari sensor ini dikumpulkan dan dikirim ke server berbasis *cloud*, di mana data tersebut dianalisis dan digunakan untuk mengontrol kondisi lingkungan. Hasil percobaan yang dilakukan menunjukkan bahwa sistem mampu mempertahankan kondisi lingkungan yang diinginkan untuk budidaya jamur dengan tingkat akurasi yang tinggi. Sistem pemantauan rumah jamur berbasis IoT juga mudah digunakan dan dapat dikontrol dengan mudah oleh pengguna yang tidak memiliki pengalaman sebelumnya dalam budidaya jamur.

Tulisan ini akan membahas implementasi sistem monitoring dan otomasi rumah jamur dengan menggunakan Nodemcu Wemos D1 Mini ESP8266 dan sensor DHT22, BH1750, MQ135, dan sensor *soil moisture*. Sistem ini membutuhkan suatu protokol yang dapat membantu pengiriman data yang handal agar keadaan lingkungan dapat terpantau dengan baik. Sistem ini menggunakan dua mikrokontroler pada bagian *node sensor* dan *gateway*. Pada *node sensor* akan dipasang sensor yang disebutkan sebelumnya untuk mengambil data keadaan lingkungan kumbung jamur. Data kemudian dikirim ke *gateway* melalui protokol UDP. Kemudian *gateway* yang menerima data dari *node sensor* akan memproses data dan menggunakan protokol TCP untuk mengirimkan data ke web server. Langkah selanjutnya adalah melakukan perbandingan QoS untuk parameter *delay* dan *dataloss* protokol TCP dan UDP untuk mengetahui pengaruhnya saat proses pengiriman data. Kedua protokol ini sangat penting digunakan pada pemantauan rumah jamur karena tanaman ini sangat sensitif terhadap kondisi lingkungan, sehingga diperlukan protokol pengiriman data yang cepat dan akurat. Hal ini telah di jelaskan oleh [13] bahwa dengan mengendalikan keadaan lingkungan untuk budidaya maka produksi dan kualitas tanaman dapat meningkat. Ketika kondisi rumah jamur tidak terkendali, maka akan mengakibatkan jamur tidak tumbuh dengan optimal.

## 2. Metode Penelitian

Langkah penelitian yang dilakukan dapat dilihat pada Gambar 1. Metode yang digunakan pada penelitian ini menggunakan metode eksperimen. Metode yang digunakan ini akan memperoleh data sesungguhnya melalui percobaan. Data TCP dan UDP akan diperoleh dari proses yang dilakukan. Sehingga hasil yang diperoleh dapat diimplementasikan pada objek penelitian. Langkah penelitian diawali dengan studi literatur mengenai budidaya jamur dan komunikasi data. Untuk budidaya jamur termasuk parameter keadaan lingkungan kumbung jamur yang baik untuk pertumbuhan jamur, dan komunikasi data seperti protokol, OSI Layer, TCP dan UDP. Kemudian setelah mengetahui parameter yang baik untuk pertumbuhan jamur selanjutnya membuat rancangan perangkat keras dan perangkat lunak untuk *node* dan *gateway* dari sistem monitoring dan otomasi rumah jamur berbasis jaringan sensor nirkabel. Kemudian sistem tersebut diuji coba dan dilihat hasilnya. Jika sistem bekerja sesuai dengan rancangan maka akan dilakukan pengambilan data, jika belum maka sistem akan dirancang ulang. Selanjutnya jika sistem sudah bekerja sesuai rancangan, maka dilakukan pengambilan data QoS dari protokol TCP dan UDP. Proses pengambilan data QoS menggunakan perangkat lunak WireShark. Data tersebut akan dihitung dan di analisis perbandingan antara protokol TCP dan UDP beserta keunggulannya saat diimplementasikan pada sistem monitoring dan otomasi rumah jamur berbasis jaringan sensor nirkabel. Perancangan sistem monitoring dan otomasi rumah jamur terdiri dari beberapa langkah. Pertama yaitu membuat skematik sistem monitoring dan otomasi rumah jamur, kemudian merancang perangkat keras (*node & gateway*) sistem tersebut dan pembuatan tampilan website dan database untuk server. Langkah terakhir yang dilakukan adalah menyimpulkan data dari hasil penelitian yang dilakukan.



Gambar 1. Diagram alir tahapan metode penelitian.

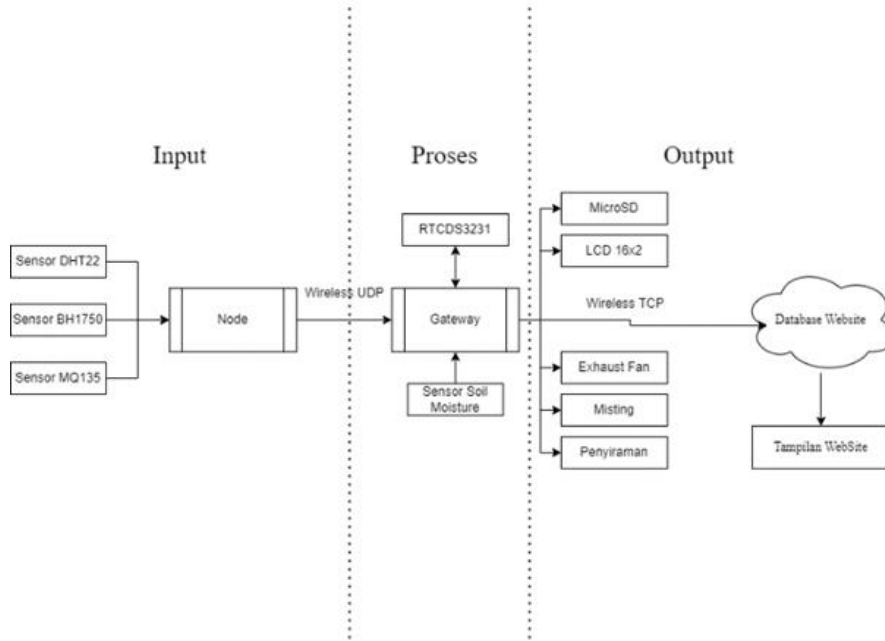
**2.1. Prinsip Kerja**

Secara garis besar sistem ini memanfaatkan IoT dengan konsep jaringan Wireless Sensor Network untuk pengiriman datanya. Sistem ini memiliki dua bagian yaitu *node* sensor sebagai pengambil data lingkungan kumbung jamur dan *gateway* sebagai pusatnya. Pada *node* akan terhubung dengan sensor – sensor pengambilan data lapangan dan pada *gateway* terdiri dari mikrokontroler ESP8266 sebagai penerima data lewat wireless dan penyimpanan kemudian Arduino UNO sebagai pengendali *relay* dan LCD. Kemudian ada bagian database sebagai tempat untuk menyimpan data dan untuk ditampilkan pada bagian halaman utama website.

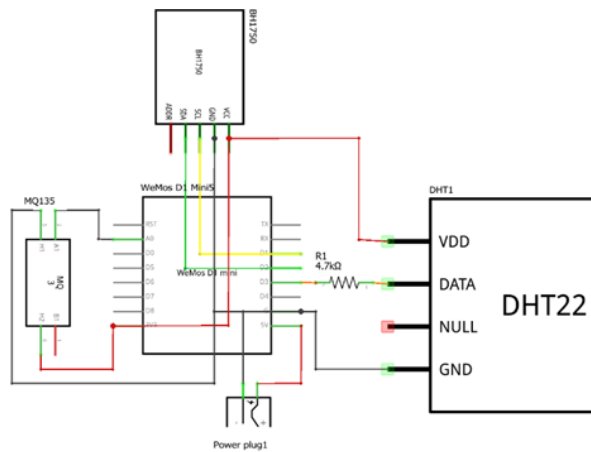
Untuk pengiriman data dari *node* ke *gateway* menggunakan protokol UDP dan pengiriman dari *gateway* ke website menggunakan protokol TCP. pada *gateway* juga terhubung dengan beberapa *relay*, *relay* tersebut akan menggerakkan misting ketika kelembapan udara di bawah 60%, kemudian penyiraman ketika kelembapan media tanam di bawah 20%, dan exhaust fan ketika suhu di atas 30 °C dan kadar CO<sub>2</sub> di atas 10.000 ppm. Diagram alir dari prinsip kerja sistem monitoring dan otomasi rumah jamur menggunakan IoT berorientasi WSN ditampilkan pada Gambar 2. Langkah awal sebelum merangkai perangkat keras adalah membuat skematik rangkaian *node* dan *gateway*. Rincian komponen dan pin yang digunakan terdapat pada Tabel 1. Skematik dari perangkat keras ditampilkan pada Gambar 3 dan 4.

Tabel 1. Pin I/O mikrokontroler.

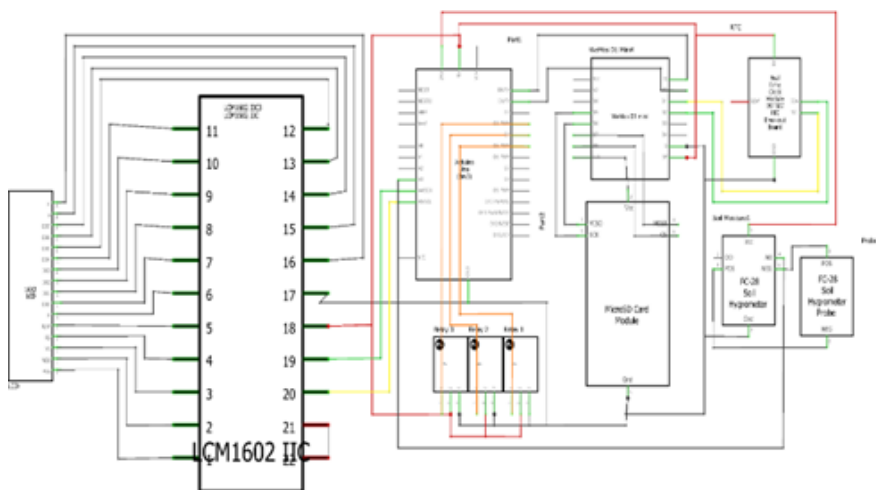
Node			Gateway		
No.	Komponen	Pin IO	Arduino		
1.	BH1750 (Sensor intensita cahaya)	SCL,SDA (D1, D2)	No.	Komponen	Pin IO
2.	DHT22 (Sensor suhu dan kelembapan udara)	D3	1.	LCD 16x2 I2C	SCL,SDA (A5, A4)
3.	MQ135 (Sensor CO <sub>2</sub> )	A0	2.	Relay	2, 3, 4
			3.	Wemos D1 Mini	TX, RX
			4.	Soil Moisture	A0
			Wemos D1 Mini		
			1.	MicroSD	MISO, MOSI, SCK, CS (D5, D6, D7, D8)
			2.	RTC	SCL, SDA (D1, D2)



Gambar 2. Prinsip kerja sistem.



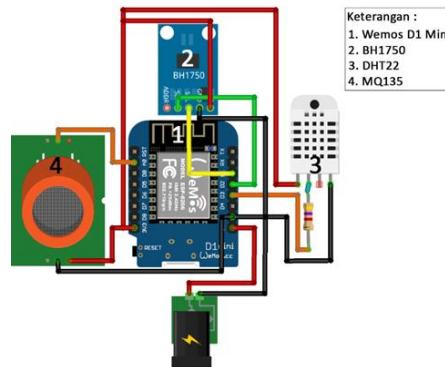
Gambar 3. Rangkaian skematik node.



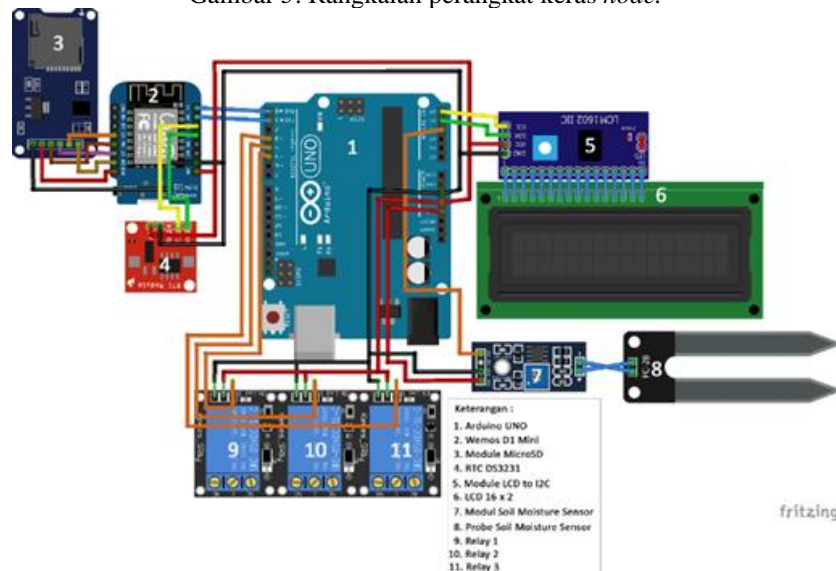
Gambar 4. Rangkaian skematik gateway.

### 2.2. Rancangan perangkat keras

Langkah selanjutnya yaitu merancang perangkat keras dari sistem. Untuk desain perangkat keras menggunakan perangkat lunak *fritzing*. Perancangan perangkat keras dari *node* dan *gateway* ditampilkan pada Gambar 5 dan 6. Setelah desain rangkaian *node* dan *gateway* selesai, selanjutnya adalah membuat program. Program yang diajukan memiliki tiga bagian yaitu *node*, *gateway*, dan website. Pembuatan program menggunakan perangkat lunak Arduino IDE untuk memprogram mikrokontroler dan perangkat lunak Visual Studio Code untuk pembuatan website.



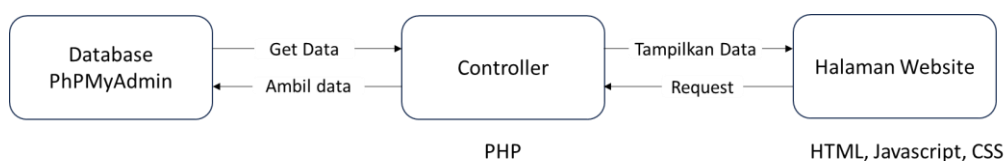
Gambar 5. Rangkaian perangkat keras *node*.



Gambar 6. Rangkaian perangkat keras *gateway*.

### 2.3. Rancangan website

Untuk algoritma website terbagi menjadi tiga bagian, yaitu bagian database, controller dan halaman website (view). Data yang dikirimkan dari *node* akan masuk dan disimpan pada tabel database. Halaman tampilan website akan melakukan request pengambilan data dari database melalui controller, lalu controller akan mengambil data tersebut dan mengirimkannya ke tampilan website. Kemudian halaman tampilan website akan menampilkan data tersebut. Untuk diagram tampilan website akan digambarkan pada Gambar 7.



Gambar 7. Diagram *database* dan *website*.

*Quality of Service* (QoS) adalah metode untuk mengukur kualitas jaringan. QoS juga merupakan upaya untuk menggambarkan karakteristik dan model layanan. QoS mengacu pada kemampuan jaringan dalam melakukan pelayanan pada trafik tertentu dan teknologi yang beragam. QoS secara kualitatif maupun kuantitatif mendefinisikan atribut layanan jaringan yang disediakan [18]. Pada Tabel 2 dapat dilihat nilai dari presentase QoS.

Tabel 2. Nilai QoS.

Nilai	Presentase (%)	Indeks
3,8 - 4	95 - 100	Sangat Memuaskan
3 - 3,79	75 - 94,75	Memuaskan
2 - 2,99	50 - 74,75	Kurang Memuaskan
1 - 1,99	25 - 49,75	Jelek

Pengujian QoS dilakukan dengan menghitung beberapa parameter QoS yaitu *throughput*, *packetloss*, *delay*, dan *jitter* dari sebuah jaringan. Pengambilan data menggunakan perangkat lunak Wireshark. Untuk perhitungan dari parameter tersebut sebagai berikut.

#### a. *Throughput*

*Throughput* adalah kecepatan transfer data efektif yang diukur dalam satuan bps (bit per second). *Throughput* dapat dihitung dengan membagi jumlah paket yang berhasil diterima selama interval waktu tertentu dengan durasi interval waktu tersebut. *Throughput* juga biasa disebut dengan bandwidth karena dalam kondisi sebenarnya *throughput* memang bisa disebut bandwidth [18]. Untuk parameter kategori *throughput* menurut standar TIPHON (Telecommunications and Internet Protocol Harmonization Over Network) ditunjukkan pada Tabel 3. Untuk persamaan *throughput* dapat dilihat pada persamaan (1). Menghitung *throughput* yaitu dengan melihat paket data yang terlihat dan lamanya pengamatan. Paket data yang diterima bisa dilihat pada hasil pengambilan pada wireshark. Untuk menghitung nilainya dengan membagi paket data yang diterima dengan lamanya pengamatan.

$$\text{Throughput} = \frac{\text{Paket data diterima}}{\text{Lama Pengamatan}} \quad (1)$$

Tabel 3. Kategori *throughput*.

Kategori <i>Throughput</i>	<i>Throughput</i> (bps)	Indeks
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Jelek	< 25	1

#### b. *Packet loss*

*Packet loss* adalah suatu kondisi dimana suatu paket data akan mengalami kehilangan data atau data tidak sampai ke tujuan karena suatu kondisi [18]. Kondisi ini dapat berupa collision dan congestion pada jaringan. Untuk parameter kategori *packet loss* ditunjukkan pada Tabel 4. Untuk persamaan *packetloss* dapat dilihat pada persamaan (2). Menghitung *packetloss* bisa dilihat langsung pada wireshark dengan cara menyaring paket ACK yang ada. Tetapi untuk menghitung *packetloss* yaitu dengan menghitung selisih paket data yang dikirim dengan yang diterima, kemudian dibagi dengan paket yang diterima di kalikan 100%. Untuk hasil dari *packetloss* yaitu dalam bentuk presentase.

$$\text{Packetloss} = \frac{(\text{Paket data dikirim} - \text{paket data diterima})}{\text{Paket diterima}} \quad (2)$$

Tabel 4. Kategori *packet loss*.

Kategori Degradasi	Packet Loss (%)	Indeks
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Jelek	25	1

### c. Delay

*Delay* adalah waktu yang dibutuhkan data untuk melakukan perjalanan dari pengirim ke penerima. Latency dapat dipengaruhi oleh jarak, media fisik, kongesti atau pemrosesan yang lama [18]. Untuk parameter kategori *delay* ditunjukkan pada Tabel 5. Untuk persamaan *delay* dapat dilihat pada persamaan (3). Untuk menghitung *delay* pertama harus melihat waktu paket dikirim dan diterima. Hitung *delay* dari setiap paket kemudian hasil akhirnya di rata – ratakan. Untuk hasil akhir *delay* satuannya adalah ms.

$$Delay = \frac{Packet\ Length}{Link\ Bandwidth} \quad (3)$$

Tabel 5. Kategori *delay*.

Kategori Latensi	Besar Delay (ms)	Indeks
Sangat Bagus	<150 ms	4
Bagus	150 ms s/d 300 ms	3
Sedang	300 ms s/d 450 ms	2
Jelek	> 450 ms	1

### d. Jitter

*Jitter* atau variasi kedatangan paket dapat didefinisikan sebagai variasi *delay* antar paket yang disebabkan oleh panjang antrian dalam pemrosesan data dan reassemble paket data pada akhir pengiriman karena kesalahan sebelumnya [18]. Untuk kategori parameter jitter ditunjukkan pada Tabel 6. Untuk persamaan nilai dari jitter dapat dilihat pada persamaan (4). Persamaan (5) yaitu untuk mencari nilai dari Total Variasi *delay*. Untuk menghitung jitter yaitu dengan cara membagi total variasi *delay* dengan total paket yang diterima. Total variasi *delay* adalah *delay* dikurangi dengan rata – rata *delay*. Hasil akhir jitter yaitu dengan satuan ms.

$$Jitter = \frac{Total\ variasi\ delay}{Total\ paket\ yang\ diterima} \quad (4)$$

$$Total\ variasi\ delay = Delay - rata\ rata\ delay \quad (5)$$

Tabel 6. Kategori *jitter*.

Kategori Jitter	Jitter (ms)	Indeks
Sangat Bagus	0 ms	4
Bagus	0 ms s/d 75 ms	3
Sedang	75 ms s/d 125 ms	2
Jelek	125 ms s/d 225 ms	1

## 3. Hasil dan Pembahasan

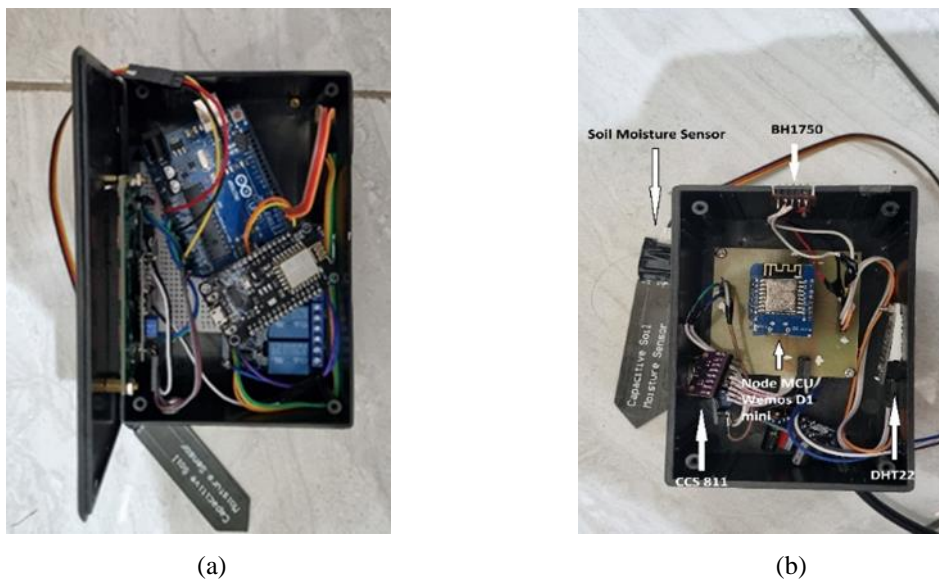
### 3.1. Perancangan Perangkat Keras

Sistem ini memiliki dua buah mikrokontroler yaitu *node* dan *gateway*. Keduanya akan dipasang pada box berbahan plastik berukuran 12.5 cm x 8.5 cm x 5 cm. Kemudian akan disolder permanen agar alat tidak mudah rusak pada saat di tempatkan di kumpang jamur. Untuk gambar dari sistem monitoring dan otomasi rumah jamur berbasis jaringan sensor nirkabel dapat dilihat pada Gambar 8 dan 9. Gambar 8 dan 9 merupakan alat dari sistem monitoring dan otomasi kumpang jamur berbasis JSN yang memiliki mikrokontroler NodeMCU ESP8266 dengan sumber tegangan AC 220V di *node* dan tambahan Arduino UNO pada *gateway*. Alat tersebut memiliki dimensi 12.5 cm x 8.5 cm x 5 cm pada *node* dan 14.5 cm x 9.5 cm x 5 cm pada *gateway*. Perangkat *gateway* memiliki dimensi yang besar karena untuk memberikan ruang tempat *relay* berada.





Gambar 8. Alat tampak samping



(a)

(b)

Gambar 9. Alat tampak atas (a). Node (b). Gateway

### 3.2. Website

Proses pembuatan website menggunakan bahasa PHP, HTML, CSS dan Javascript. Disain web ini masih menggunakan database localhost xampp. Untuk aliran data yang dikirim dari gateway akan masuk ke database di localhost xampp dan akan masuk pada tabel database tersebut. setelah itu data diambil berdasarkan ID yang paling terakhir masuk ke database. Data masing-masing parameter akan diambil berdasarkan kolom pada tabel database. Data tersebut akan ditampilkan ke halaman utama website. Desain dari website ditampilkan pada Gambar 10.

MONITORING LAHAN PERTANIAN  
KUMBUNG JAMUR

KEADAAN ATMOSFER

TEMPERATURE	Kelembapan	Intensitas Cahaya	Kadar CO2
27.60	72.40	18	31.55

KEADAAN TANAH

Moisture
120

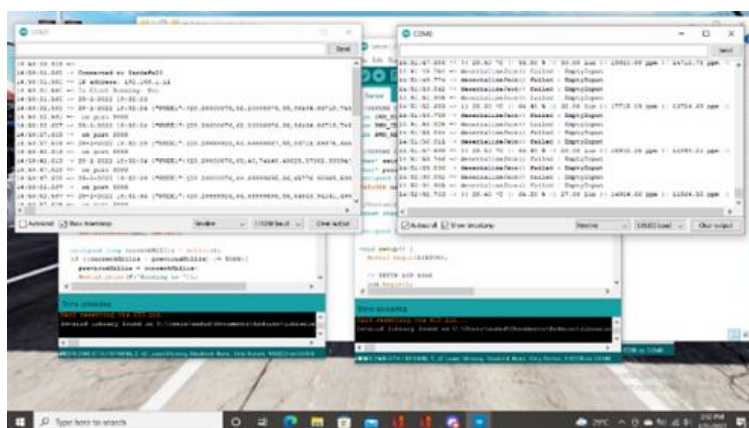
Activate Windows  
Go to Settings to activate Windows.

Gambar 10. Tampilan website.

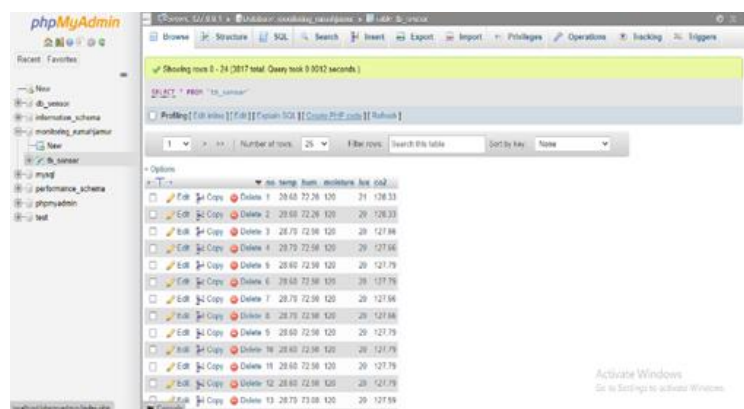
### 3.3. Pengujian sistem monitoring dan otomasi rumah jamur

Pengujian sistem dilakukan dalam tiga tahap. Pertama yaitu pengujian koneksi, dan yang kedua yaitu pengujian parameter dan terakhir pengujian pengaktifan *relay*. Untuk pengujian sensor dilakukan bersamaan dengan uji koneksi karena pengujian koneksi juga memerlukan data yang diambil dari sensor. Untuk pengujian koneksi langkah pertama yaitu mencoba mengirim data sensor dari *node* ke *gateway*. Pengiriman dari *node* ke *gateway* dibuat dengan *delay* 5 detik.

Hasil pengujian dapat dilihat menggunakan serial monitor pada perangkat lunak Arduino. Hasil pengujian ditampilkan pada Gambar 11. Ketika data berhasil diterima, maka tabel pada database localhost akan terisi. Untuk memastikan data masuk pada *gateway*, API pada program database akan mengirimkan feedback berupa tulisan yaitu “Pesan Terikirim” dan akan muncul pada serial monitor *gateway*. Untuk tabel database localhost dapat dilihat pada Gambar 12.



Gambar 11. Uji pengiriman data.



Gambar 12. Tampilan tabel *database* localhost.









Pengujian parameter memerlukan data untuk mengaktifkan *relay*. Untuk parameternya menggunakan parameter pertumbuhan jamur yang baik yang diambil dari jurnal [6], [9]-[12] dan juga pengambilan data lapangan secara langsung sebagai dasar data untuk parameter kumbung jamur yang sudah di lakukan pengambilan data. Untuk parameter yang didapat dapat dilihat pada Tabel 7.

Tabel 7. Parameter lingkungan kumbung jamur.

Parameter	Keterangan
Suhu	25 <sup>0</sup> C – 30 <sup>0</sup> C
Kelembapan udara	60% - 90%
Kelembapan baglog	20% - 50%
Intensitas cahaya	100 – 150 lumens
Kadar CO <sub>2</sub>	> 10000 ppm

Untuk pengujian *relay* perlu menggunakan data dari pengujian parameter di poin sebelumnya. Untuk pengujian parameter di atas *node* dan *gateway* perlu dinyalakan. Sensor akan diberi rangsangan langsung agar parameter tercapai dengan cepat. Untuk jumlah *relay* ada tiga, untuk *relay* 1 yaitu penyiraman, *relay* 2 yaitu misting, dan *relay* 3 yaitu exhaust fan. Hasil dari pengujian dapat dilihat pada Tabel 8.

Tabel 8. Pengujian parameter dengan *relay*.

No.	Kondisi	LCD	Hasil Relay
1.	Suhu > 30 °C		
2.	Kelembaban Udara < 60 %		
3.	Kelembaban Baglog < 20 %		
4.	Kadar CO <sub>2</sub> > 10.000 ppm		

Untuk pengujian kehandalan sistem dilakukan percobaan pengaktifan *relay* sebanyak sepuluh kali dengan keempat parameter sebelumnya. Parameter akan di buat mencapai set point sebanyak sepuluh kali. Untuk hasil dari percobaannya dapat dilihat pada Tabel 9. Berdasarkan Tabel 9 dapat dilihat bahwa sistem ini dapat diandalkan karena seluruh *relay* dapat aktif ketika parameter tercapai. Hal ini membuktikan bahwa sistem ini 100% dapat bekerja sesuai dengan yang diharapkan. Tidak ada kesalahan yang terjadi ketika percobaan.

Tabel 9. Parameter lingkungan kumbung jamur.

No.	Parameter	Percobaan										Hasil
		1	2	3	4	5	6	7	8	9	10	
1.	Suhu > 30 °C	v	v	v	v	v	v	v	v	v	v	100%
2.	Kelembaban Udara < 60 %	v	v	v	v	v	v	v	v	v	v	100%
3.	Kelembaban Baglog < 20 %	v	v	v	v	v	v	v	v	v	v	100%
4.	Kadar CO <sub>2</sub>	v	v	v	v	v	v	v	v	v	v	100%

Lama waktu *capture* dilakukan selama satu menit pada masing – masing protokol. Perangkat lunak untuk melakukan uji QoS yaitu menggunakan WireShark. Pengujian protokol TCP dan UDP dilakukan dengan uji QoS dari kedua jaringan. Proses pengujian QoS dilakukan dengan mengirim data yang sama menggunakan dua protokol TCP dan UDP. Hasil pengukuran *throughput* sistem monitoring dan otomasi rumah jamur berbasis jaringan sensor nirkabel. Untuk masing-masing protokol berdasarkan *throughput* versi TIPHON sebagai standar dari pengukuran

QoS. Pengukuran dilakukan selama satu menit pada setiap protokol. Hasil pengukuran *throughput* dapat dilihat pada Tabel 10.

Tabel 10. Parameter lingkungan kumbang jamur.

No	Protokol	Hasil pengukuran <i>throughput</i> (bps)	Keterangan	
			Indeks	Kategori
1.	TCP	5058,97	4	Sangat Bagus
2.	UDP	929,32	4	Sangat Bagus

Hasil pengukuran *packet loss* sistem monitoring dan otomasi rumah jamur berbasis jaringan sensor nirkabel untuk masing – masing protokol berdasarkan packetloss versi TIPHON sebagai standar dari pengukuran QoS. Pengukuran dilakukan selama satu menit pada setiap protokol. Hasil pengukuran packetloss dapat dilihat pada Tabel 11.

Tabel 11. Parameter lingkungan kumbang jamur.

No	Protokol	Hasil pengukuran <i>Packet loss</i> (%)	Keterangan	
			Indeks	Kategori
1.	TCP	0	4	Sangat Bagus
2.	UDP	0	4	Sangat bagus

Hasil pengukuran *delay* sistem monitoring dan otomasi rumah jamur berbasis jaringan sensor nirkabel untuk masing-masing protokol berdasarkan *delay* versi TIPHON sebagai standar dari pengukuran QoS. Pengukuran dilakukan selama satu menit pada setiap protokol. Hasil pengukuran *delay* dapat dilihat pada Tabel 12.

Tabel 12. Parameter lingkungan kumbang jamur.

No	Protokol	Hasil pengukuran <i>delay</i> (ms)	Keterangan	
			Indeks	Kategori
1.	TCP	100,041	4	Sangat Bagus
2.	UDP	3,888	4	Sangat Bagus

Hasil pengukuran jitter sistem monitoring dan otomasi rumah jamur berbasis jaringan sensor nirkabel untuk masing-masing protokol berdasarkan jitter versi TIPHON sebagai standar dari pengukuran QoS. Pengukuran dilakukan selama satu menit pada setiap protokol. Hasil pengukuran jitter dapat dilihat pada Tabel 13.

Tabel 13. Parameter lingkungan kumbang jamur.

No	Protokol	Hasil pengukuran Jitter (ms)	Keterangan	
			Indeks	Kategori
1.	TCP	99,198	2	Sedang
2.	UDP	3,835	3	Bagus

Hasil keseluruhan nilai QoS antar protokol dapat dilihat pada Tabel 14. Berdasarkan nilai QoS versi TIPHON untuk kategori nilai “Sangat Memuaskan” yaitu 3,8 – 4, “Memuaskan” jika bernilai 3 – 3,79, “Kurang Memuaskan” jika bernilai 2 – 2,99 dan “Jelek” jika bernilai 1 – 1,99 dapat dilihat pada Tabel 17. Nilai indeks QoS berdasarkan standar TIPHON, untuk masing-masing protokol, untuk protokol TCP mendapatkan nilai indeks 3,5 dengan kategori “memuaskan”. Sedangkan protokol UDP mendapatkan nilai indeks 3,75 dengan kategori “Memuaskan”.

Tabel 14. Parameter lingkungan kumbang jamur.

No	Protokol	Nilai	Kategori
1.	TCP	3,5	Memuaskan
2.	UDP	3,75	Memuaskan

Sistem monitoring rumah jamur ini sangat efektif jika menggunakan protokol UDP dari *node* ke *gateway* karena lebih cepat dalam pengiriman data dibanding TCP. Karena data yang dikirimkan kecil maka kemungkinan kehilangan data dalam pengiriman akan kecil. Tetapi jika mengirim data dari *gateway* ke database perlu menggunakan TCP karena agar memastikan seluruh data tidak hilang dan tersimpan dengan baik pada server. Untuk nilai QoS berdasarkan standar TIPHON dapat dilihat pada Tabel 15. Hasil pengukuran *throughput* untuk protokol TCP mendapat hasil 5058,97 memiliki nilai indeks 4 maka termasuk kedalam kategori “Sangat Bagus”. Untuk hasil pengukuran *throughput* protokol UDP mendapat hasil 929,32 memiliki nilai indeks 4 maka termasuk kedalam kategori “Sangat Bagus”. Hasil pengukuran *packetloss* untuk protokol TCP dan UDP sama sama mendapat hasil 0 % yang memiliki indeks 4 maka termasuk dalam kategori “Sangat Bagus”. Hasil pengukuran *delay*, hasil terendah yaitu protokol UDP dengan hasil 3,887 ms dengan nilai indeks 4 dan termasuk kedalam kategori “Sangat Bagus”. Untuk protokol TCP mendapat hasil *delay* 100,041 ms dengan nilai indeks 4 dan termasuk kedalam kategori “Sangat Bagus”. Untuk hasil pengukuran *jitter*, hasil terendah yaitu masih pada protokol UDP dengan hasil 3,835 ns dengan nilai indeks 3 dan termasuk kedalam kategori “Bagus”. Sedangkan untuk protokol TCP mendapatkan hasil 99,198 ms dengan nilai indeks 2 dan termasuk kedalam kategori “Sedang”.

Tabel 15. Hasil pengujian QoS secara keseluruhan.

	Test TCP		Test UDP
	Jumlah byte : Time Span		Jumlah byte : Time Span
	37377 Byte / 59,106 s		6785 Byte / 58,408 s
<i>Throughput</i>	632,3723 Byte/s	<i>Throughput</i>	116,1655 Byte/s
	0,6323723 KB/s x 8		0,1161655 KB/s
	<b>5,0589784 Kb/s</b>		<b>0,929324 Kb/s</b>
<i>Packet Loss</i>	((Send - Recive)/Send) x 100	<i>Packet Loss</i>	((Send - Recive)/Send) x 100
	0		0
<i>Delay</i>	Total <i>Delay</i> : 11,90484 s	<i>Delay</i>	Total <i>Delay</i> : 0,458792 s
	Rata - rata <i>Delay</i> : 0,100041 s		Rata - rata <i>Delay</i> : 0,003888068 s
	100,041 ms		3,888067797 ms
<i>Jitter</i>	Total Jitter : 11,70541 s	<i>Jitter</i>	Total Jitter : 0,452583 s
	Rata - rata Jitter : 0,099198 s		Rata rata Jitter : 0,00383545 s
	99,198 ms		3,83544915 ms

Pengujian QoS pada sistem pemantauan rumah jamur memiliki kriteria sangat bagus. Hal ini membuktikan bahwa penggunaan protokol TCP dan UDP dapat membantu pemantauan rumah jamur secara efisien. Pengendalian kondisi rumah jamur yang bagus mampu meningkatkan kualitas jamur. Selain itu, pengendalian rumah jamur yang baik akan mempercepat proses panen. Waktu panen jamur yang umumnya adalah 36 jam menjadi lebih singkat dengan mengendalikan kondisi rumah jamur.

#### 4. Kesimpulan

Implementasi protokol UDP dan TCP pada sistem monitoring dan otomasi rumah jamur menggunakan IoT berorientasi WSN telah berhasil dilakukan. Sistem monitoring dan otomasi rumah jamur menggunakan IoT berorientasi WSN dengan protokol UDP dan TCP dapat berjalan dengan baik. Pengujian dilakukan dua tahap, yaitu tahap pengujian bahwa sensor dan alat dapat bekerja dengan baik hingga data sensor dapat dilihat pada halaman website dan tahap pengujian Quality of Service (QoS) untuk membandingkan protokol UDP dengan TCP. Berdasarkan hasil pengujian yang telah dilakukan, protokol pengiriman data menggunakan UDP lebih baik jika dibandingkan dengan TCP. Namun hal ini masih termasuk kategori yang baik berdasarkan standar TIPHON. Secara keseluruhan, sistem yang dirancang mampu melakukan pengiriman data dengan baik. Pemantauan rumah jamur yang optimal mampu mempercepat proses panen jamur. Langkah selanjutnya yang dapat dilakukan untuk mendapatkan hasil yang lebih baik yaitu menggunakan sensor CO<sub>2</sub> standar industri seperti infrared CO<sub>2</sub> sensor atau MG-811. Rentang waktu pengukuran QoS perlu dilakukan lebih lama agar variabel data yang diperoleh menjadi lebih banyak.

#### Referensi

- [1] L. Mario, J. E. Hernandez, M. D. M. E. A. Díaz, H. Panetto, and J. Kacprzyk. "Agri-food 4.0: A survey of the supply chains and technologies for the future agriculture." *Computers in industry* vol. 117, pp. 103187, 2020.
- [2] U. Panahi and C. Bayilmis, "Enabling secure data transmission for wireless sensor networks based IoT application," *Ain Shams Engineering Journal*, pp. 1 - 11, 2022.
- [3] D. A. Joseph Rajan and E. R. Naganathan, "Trust Based Anonymous Intrusion Detection for Cloud Assisted WSN-IOT," *Global Transition Proceedings*, pp. 1 - 8, 2022.
- [4] Taştan, Mehmet, and H. Gökozan. "Real-time monitoring of indoor air quality with internet of things-based E-nose." *Applied Sciences* 9, no. 16, pp. 3435, 2019.
- [5] R. Herrero, "Analysis of the constrained application protocol over quick UDP internet connection transport," *Internet of Things*, pp. 1 - 13, 2020.
- [6] J. Wirges and U. Dettmar. "Performance of TCP and UDP over narrowband internet of things (NB-IOT)." In 2019 *IEEE international conference on internet of things and intelligence system (IoT&IS)*, 2019, pp. 5-11.
- [7] Thong-un, Natee, and W.Wongsaraj. "Productivity enhancement using low-cost smart wireless programmable logic controllers: A case study of an oyster mushroom farm." *Computers and Electronics in Agriculture* 195, pp. 106798, 2022.
- [8] Y. Lila, I. Susilo, Sudarsono, A. Niniet Indah, F. Lim and A. Diky, "Sistem Kontrol Otomatis Berbasis IoT dan Sel Surya untuk Budidaya Jamur Tiram di Desa Selorejo, Jombang," *Jurnal ILUNG*, vol. 1, no. Pengabdian Inovasi Lahan Basah Unggul, pp. 1 - 8, 2021.
- [9] R. Adrian, Hermanto, P. Joko, S. Atmaja and H. Ridhan, "Sistem Budidaya Jamur Berbasis Internet Of Things Menggunakan Telegram BOT," *Jurnal Rekayasa Teknologi Nusa Putra*, vol. 5, pp. 28 - 33, 2018.
- [10] H. M. Sholihul, M. A. Mizar, D. Lestari, A. Witjoro, and M. Irvan. "IoT and Fuzzy Logic Based Smart Mushroom Cultivation Technology." In 2021 *7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, pp. 346-350, 2021.
- [11] M. Suresh, M. Srinivasan, S. S. Gowri, D. Karthikeyan, V. Nakhul, K. A. Naveen, S. Sundar and P. Maniraj, "Monitoring and Automatic Control of Various Parameters for Mushroom Farming," *IOP Conference Series: Materials Science and Engineering*, pp. 1 - 6, 2021..

- [12] M. K. Mohamed Rawidean, H. Ahmad Nizar, Y. Ismail mat, M. Ibrahim, K. Chan Pick and R. Norasfaliza, "Application of Wireless Sensor Networks in Shiitake Mushroom Cultivation," *Eleventh International Conference on Sensing Technology (ICST)*, 2017.
- [13] R. Hasibur, F. Md. Omar, A. H. Talha Bin, R. Wahidur, H. Muhammad Minoar, H. Mahbulbul, I. Shafiqul, M. Moinuddin, I. Md. Tarequl and A. Mir Mohammad, "IoT enabled mushroom farm automation with Machine Learning to classify," *Journal of Agriculture and Food Research*, pp. 1 - 16, 2022.
- [14] D. R. Kristiyanti, A. Wijayanto, and A. Aziz. "Sistem Monitoring Suhu dan Kelembaban pada Budidaya Jamur Tiram Berbasis Internet of Things Menggunakan MQTT dan Telegram BOT." *Adopsi Teknologi dan Sistem Informasi (ATASI)*, vol. 1(1), pp. 61-73, 2022.
- [15] J. L. Chong, K. W. Chew, A. P. Peter, H. Y. Ting, and P. L. Show. "Internet of Things (IoT)-Based Environmental Monitoring and Control System for Home-Based Mushroom Cultivation." *Biosensors 13*, no. 1, pp. 98, 2023.
- [16] A. Najmurrokhman, A.Daelami, E. Nurlina, U. Komarudin, and H. Ridhatama. "Development of Temperature and Humidity Control System in Internet-of-Things based Oyster Mushroom Cultivation." *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 551-555, 2020.
- [17] A. M. Mohd Saiful, B. Salinda, M. M. Musa and A. M.S. Zainal, "Internet of Things based Smart Environmental Monitoring for Mushroom Cultivation," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, pp. 847 - 852, 2018.
- [18] R. Wulandari, "Analisis QoS (Quality of Service Pada Jaringan Internet (Studi Kasus : UPT Loka Uji Teknik Penambangan Jampang Kulon - LIPI))," *Jurnal Teknik Informatika dan Sistem Informasi*, pp. 162 - 172, 2016.