



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**IMPLEMENTASI INTERNET OF THINGS (IOT) UNTUK  
PEMANTAUAN DAN PENGENDALIAN IKLIM PADA  
BUDIDAYA JAMUR TIRAM**

Calvin Rafindra Adhinta  
0711164000012

Dosen Pembimbing  
Sri Rahayu, S.T., M.Kom.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Surabaya 2020





**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - EE 184801

**IMPLEMENTASI INTERNET OF THINGS (IOT) UNTUK  
PEMANTAUAN DAN PENGENDALIAN IKLIM PADA  
BUDIDAYA JAMUR TIRAM**

Calvin Rafindra Adhinta  
0711164000012

Dosen Pembimbing  
Sri Rahayu, S.T., M.Kom.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Surabaya 2020



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**IMPLEMENTASI INTERNET OF THINGS (IOT) UNTUK PEMANTAUAN DAN PENGENDALIAN IKLIM PADA BUDIDAYA JAMUR TIRAM**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Jombang, Juni 2020

Calvin Rafindra Adhinta

NRP. 0711 16 4000012

*-halaman ini sengaja dikosongkan-*

# **IMPLEMENTASI INTERNET OF THINGS (IOT) UNTUK PEMANTAUAN DAN PENGENDALIAN IKLIM PADA BUDIDAYA JAMUR TIRAM**

Nama Mahasiswa : Calvin Rafindra Adhinta

Dosen pembimbing I : Sri Rahayu S.T., M.Kom.

## **Abstrak**

Internet of Things merupakan sebuah inovasi terbaru yang diharapkan dapat menyelesaikan beberapa masalah yang ada di berbagai bidang, salah satunya adalah dibidang pertanian. Tak menentunya kalender musim di Indonesia menjadikan sebuah masalah tersendiri bagi para petani terutama petani jamur tiram. Jamur tiram merupakan tanaman yang membutuhkan iklim khusus agar dapat tumbuh yaitu dengan kelembaban 60-90% dan suhu 22-27 C.

Pada tugas akhir ini akan dibuat sebuah perangkat yang diharapkan dapat menjamin suhu dan kelembaban bagi jamur tiram. Qumbang-Q merupakan satu contoh implementasi sistem IoT untuk keperluan budidaya jamur tiram. Budidaya jamur tiram sendiri dikenal cukup sulit, dan rentan gagal panen akibat terjadinya fluktuasi suhu, dan kelembaban di area tanam. Dalam hal ini sistem IoT berfungsi menstabilkan kondisi iklim mikro dalam kumbung jamur, yang dapat dikontrol secara otomatis oleh mikrokontroler serta dapat dimonitor jarak jauh lewat jaringan internet, oleh sebab itu harus diperlukan pengetahuan tentang QoS.

Pengujian dilakukan dengan beberapa skenario yaitu pengaruh jumlah kumbung terhadap QoS, pengaruh lama pengukuran terhadap QoS dan pengaruh lokasi jarak terhadap QoS. Kemudian hasil dari QoS berupa throughput, delay, jitter, dan packet loss di-capture menggunakan wireshark, dan kondisi diklasifikasikan dari jaringan Qumbang-Q menurut standar ETSI TIPHON.

Dari hasil pengujian diketahui bahwa pengujian yang sudah dilakukan didapatkan hasil delay sebesar 0.85 s, jitter sebesar 0.00080 s, throughput sebesar 583.2 bit/s.

**Kata Kunci :** *IoT, QoS, Throughput, Jitter, Delay.*

*-halaman ini sengaja dikosongkan-*



# IMPLEMENTATION OF INTERNET OF THINGS (IOT) FOR CLIMATE MONITORING AND CONTROL IN OYSTER MUSHROOM CULTIVATION

Student Name : Calvin Rafindra Adhinta

Supervisor I : Sri Rahayu S.T., M.Kom.

## Abstract

Internet of Things is a recent innovation that is expected to solve several problems that exist in various fields, one of which is in agriculture. The uncertain season calendar in Indonesia makes it a problem for farmers, especially oyster mushroom farmers. Oyster mushroom is a plant that needs a special climate in order to grow, with a humidity of 60-90% and a temperature of 22-27 C.

In this final project, a device which is expected to guarantee temperature and humidity for oyster mushrooms will be made. Qumbung-Q is an example of the implementation of the IoT system for the needs of oyster mushroom cultivation. Oyster mushroom cultivation itself is known to be quite difficult, and vulnerable to crop failure due to temperature and humidity fluctuations in the planting area. In this case, the IoT system functions to stabilize the micro climatic conditions in mushroom kumbung, which can be controlled automatically by a microcontroller and can be monitored remotely via the internet, therefore knowledge of QoS must be required.

The test is carried out with several scenarios, namely the effect of the number of kumbung on QoS, the effect of measurement time on QoS and the influence of distance location on QoS. Then the results of QoS in the form of throughput, delay, jitter, and packet loss are captured using Wireshark, and the conditions are classified from the Qumbung-Q network according to the ETSI TIPHON standard.

From the test results it is known that the test that has been done is obtained the delay result of 0.85 s, jitter of 0.00080 s, throughput of 583.2 bit / s.

**Keywords :** *IoT, QoS, Throughput, Jitter, Delay.*

*-halaman ini sengaja dikosongkan-*

## KATA PENGANTAR

Puji Syukur saya panjatkan kehadirat Allah SWT atas segala Rahmat, Karunia, dan Petunjuk yang telah dilimpahkan-Nya sehingga saya mampu menyelesaikan tugas akhir dengan judul **“IMPLEMENTASI INTERNET OF THINGS (IoT) UNTUK PEMANTAUAN DAN PENGENDALIAN IKLIM PADA BUDIDAYA JAMUR TIRAM”**. Tugas akhir ini disusun sebagai salah satu persyaratan untuk menyelesaikan jenjang pendidikan S1 pada Bidang Studi Telekomunikasi Multimedia, Departemen Teknik Elektro, Fakultas Teknologi Elektro Informatika Cerdas, Institut Teknologi Sepuluh Nopemer.

Pada kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada pihak-pihak yang turut membantu penulis dalam menyelesaikan Tugas Akhir ini, yaitu kepada :

1. Mama dan Papa atas doa dan cinta yang tak henti dalam keadaan apapun selalu mendukung dan memberi motivasi.
2. Ibu Sri Rahayu S.T.,M.T. selaku dosen pembimbing.
3. Teman – teman angkatan e56 yang telah menemani masa perkuliahan hingga semester 8 ini.
4. Rafael dan Mas Ilham yang membantu untuk menyelesaikan problem dalam tugas akhir ini.
5. Dosen – dosen penguji yang memberikan saran dan kritik dalam membantu penyempurnaan isi dari Tugas Akhir.
6. Mas Amirul yang sudah memberi banyak wawasan tentang jamur tiram.

Besar harapan penulis bahwa buku Tugas Akhir ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Fakultas Teknologi Elektro pada khususnya.

Jombang, Juni 2020

Penulis

*-halaman ini sengaja dikosongkan-*

## DAFTAR ISI

PERNYATAAN KEASLIAN .....	v
TUGAS AKHIR .....	v
Abstrak .....	vii
Abstract .....	ix
KATA PENGANTAR.....	xi
BAB I .....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	1
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	2
1.5 Metodologi.....	2
1.6 Sistematika Penulisan .....	4
1.7 Relevansi / Manfaat .....	5
BAB II.....	6
DASAR TEORI.....	6
2.1 Internet of Things (IoT).....	6
2.1.1 Layer IoT.....	6
2.1.2 Pengembangan Teknologi IoT .....	7
2.2 Bluetooth.....	8

2.3 Quality of Service (QoS) .....	8
2.3.1 Latency / Delay .....	9
2.3.2 Throughput .....	9
2.3.3 Jitter .....	10
2.3.4 Packet Loss .....	10
2.4 Jamur Tiram .....	10
2.4.1 Kandungan gizi jamur tiram .....	11
2.4.2 Budidaya Jamur Tiram .....	12
2.4.3 Kendala Budidaya Jamur Tiram .....	15
2.5 Bahasa Pemrograman C .....	15
2.5.1 Fungsi .....	15
2.5.2 Elemen Dasar Bahasa C .....	16
2.6 Cloud Computing .....	16
2.6.1 Jenis Cloud Computing .....	16
2.7 Wireshark .....	19
2.7.1 Kegunaan Dari Wireshark .....	19
<b>BAB III .....</b>	<b>21</b>
<b>METODOLOGI PENELITIAN .....</b>	<b>21</b>
3.1 Diagram Alir Penelitian .....	21
3.2 Permasalahan Budidaya Jamur Tiram .....	22
3.2.1 Survey Lapangan .....	22
3.2.3 Identifikasi Kebutuhan .....	23

3.3 Pembuatan Rumah Jamur Berbasis IoT (Qumbang-Q)	24
3.3.1 Pembuatan Kumbang Jamur	24
3.3.2 Pembuatan Hardware	25
3.3.3 Pembuatan Sistem Perangkat Lunak	31
3.3.4 Integrasi Sistem Hardware dan Software	37
3.4 Pengukuran QoS	42
3.4.1 Software Tool	42
3.4.2 Parameter Yang Diukur	44
3.5 Perhitungan Parameter Hasil Pengukuran	47
3.5.1 Delay	47
3.5.2 Jitter	47
3.6 Standar Klasifikasi QoS	47
3.6.1 Delay	48
3.6.2 Jitter	48
3.6.3 Throughput	49
3.6.4 Packet loss	49
BAB 4	50
HASIL DAN ANALISA	50
4.1 Hasil Pembuatan Kumbang Jamur	50
4.2 Hasil Implementasi Sistem Hardware	51
4.3 Hasil Tampilan Cloud (User Interface)	52
4.4 Hasil Pengujian Sistem	54

4.4.1 Hasil Pengujian Fungsional .....	54
4.4.2 Hasil Pengujian Pengukuran QoS .....	56
4.4.3 Hasil Perhitungan QoS .....	60
4.5 Pembahasan .....	62
4.6 Perbandingan Performansi Layanan Sistem .....	63
4.6.1 Perbandingan Parameter Hasil Pengukuran .....	64
4.6.2 Klasifikasi Parameter Hasil Pengukuran .....	66
4.7 Langkah-Langkah Penggunaan Qumbang-Q .....	69
BAB 5 .....	73
PENUTUP .....	73
5.1 Kesimpulan .....	73
5.2 Saran .....	73
DAFTAR PUSTAKA .....	75
LAMPIRAN A .....	77
LAMPIRAN B .....	82
LAMPIRAN C .....	84
LAMPIRAN D .....	86



## DAFTAR GAMBAR

<b>Gambar 2.1</b> Software as a Service .....	17
<b>Gambar 2.2</b> Platform as a Service .....	18
<b>Gambar 2.3</b> Infrastructure as a Service.....	18
<b>Gambar 2.4</b> Tampilan Wireshark .....	19
<b>Gambar 3.1</b> Diagram Alir Metodologi Penelitian.....	21
<b>Gambar 3.2</b> Desain kumbang .....	24
<b>Gambar 3.3</b> Diagram Blok Sistem Perangkat Keras .....	25
<b>Gambar 3.4</b> Rangkaian Perangkat Keras .....	27
<b>Gambar 3.5</b> Realisasi Sistem Q1 .....	29
<b>Gambar 3.6</b> Realisasi Sistem Q2 .....	30
<b>Gambar 3.7</b> Realisasi Modul IoT.....	31
<b>Gambar 3.8</b> Pemrograman Sistem Q1.....	32
<b>Gambar 3.9</b> Pemrograman Sistem Q2.....	33
<b>Gambar 3. 10</b> Pemrograman Sistem Modul IoT .....	34
<b>Gambar 3.11</b> Tampilan Awal Thingspeak .....	36
<b>Gambar 3.12</b> Tampilan Awal Blynk.....	37
<b>Gambar 3.13</b> Tampilan Awal <i>Software</i> PCAP Remote .....	43
<b>Gambar 3.14</b> Tampilan Awal <i>Software Wireshark</i> .....	44
<b>Gambar 3.15</b> Proses <i>Filter</i> Pada <i>Wireshark</i> .....	45
<b>Gambar 3.16</b> Tampilan Pada Menu <i>Capture File Properties</i> ....	46
<b>Gambar 4.1</b> Tampilan Berbagai Sisi Kumbang .....	50
<b>Gambar 4.2</b> Hasil Implementasi Sistem Q1 .....	51
<b>Gambar 4.3</b> Sistem Hardware Yang Dibuat.....	52
<b>Gambar 4.4</b> Tampilan User Interface.....	53
<b>Gambar 4.5</b> Hasil Analisa Pada <i>Wireshark</i> Pada Pengujian Dengan Satu Kumbang.....	58
<b>Gambar 4.6</b> Hasil Analisa Pada <i>Wireshark</i> Pada Pengujian Dengan Dua Kumbang.....	58
<b>Gambar 4.7</b> Hasil Analisa Pada <i>Wireshark</i> Dengan Waktu 30s	59

<b>Gambar 4.8</b>	Hasil Analisa Pada Wireshark Dengan Waktu 60 s	59
<b>Gambar 4.9</b>	Hasil Analisa Pada Wisehark Dengan Waktu 90 s	59
<b>Gambar 4.10</b>	Hasil Analisa Pada Wireshark Dengan Jarak 1 m	60
<b>Gambar 4.11</b>	Hasil Analisa Pada Wireshark Dengan Jarak 3 m	60
<b>Gambar 4.12</b>	Throughput Dari Semua Pengujian .....	65
<b>Gambar 4.13</b>	Rata-Rata Delay Dari Semua Pengujian .....	65
<b>Gambar 4.14</b>	Rata-Rata Jitter Dari Semua Pengujian .....	66
<b>Gambar 4. 15</b>	Kolom Pengisian Kode Blynk Di Dalam Program Yang Sudah Disediakan .....	70
<b>Gambar 4.16</b>	Kolom Pengisian Nama Wifi Dan Password Wifi .....	70
<b>Gambar 4.17</b>	Tanda Upload Pada Arduino IDE .....	70

## DAFTAR TABEL

<b>Tabel 2.1</b> Indeks Parameter QoS Menurut TIPHON .....	9
<b>Tabel 2.2</b> Kandungan Asam Pada Jamur Tiram .....	12
<b>Tabel 3.1</b> Daftar perangkat yang digunakan .....	26
<b>Tabel 3.2</b> Pengujian Fungsional Sistem.....	39
<b>Tabel 3.3</b> Klasifikasi Delay TIPHON .....	48
<b>Tabel 3.4</b> Klasifikasi Jitter TIPHON .....	48
<b>Tabel 3.5</b> Klasifikasi Throughput TIPHON.....	49
<b>Tabel 3.6</b> Klasifikasi Packet Loss TIPHON .....	49
<b>Tabel 4.1</b> Hasil Pengujian Fungsional .....	55
<b>Tabel 4.2</b> Hasil Perhitungan QoS Pengaruh Jumlah Kumbang....	60
<b>Tabel 4.3</b> Hasil Perhitungan QoS Pengaruh Lama Pengukuran...	61
<b>Tabel 4.4</b> Hasil Perhitungan QoS Pengaruh Lokasi Jarak Kumbang .....	62
<b>Tabel 4.5</b> Deskripsi Kondisi Pengukuran .....	64
<b>Tabel 4.6</b> Klasifikasi Hasil Pengukuran .....	66
<b>Tabel 4.7</b> Tabel Klasifikasi Rata-Rata.....	69

*-halaman ini sengaja dikosongkan-*

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dewasa ini Internet Of Things menjadi sebuah pembahasan di era Revolusi Industri 4.0 . IoT juga dapat menjadi sebuah solusi untuk mengatasi sebuah masalah yang ada di masyarakat . Berbagai macam bidang dapat dikombinasikan dengan Internet Of Things. Oleh karena itu pentingnya pemanfaatan dan pengimplementasian teknologi IoT ini harus segera direalisasikan untuk membantu menyelesaikan masalah yang ada.

Berbagai macam permasalahan diberbagai bidang dapat diselesaikan menggunakan IoT . Masalah di bidang pertanian merupakan salah satu masalah yang dapat dibantu oleh teknologi IoT ini. Permasalahan musim yang tidak menentu di Indonesia, dan tidak tepatnya calendar musim yang ada di Indonesia menjadi sebuah masalah tersendiri bagi para petani . Terutama petani jamur tiram yang membutuhkan iklim yang lembab untuk pertumbuhan jamur tiram. Pengontrolan iklim mikro dalam kandang yang berbasis IoT akan menjadi sebuah solusi untuk para petani jamur tiram kedepannya. Dengan demikian para petani jamur akan mendapatkan hasil panen yang maksimal dan tidak bergantung pada cuaca yang tidak menentu di Indonesia.

Dikarenakan musim yang tidak menentu dan tidak dapat diprediksi secara akurat, maka dalam tugas akhir ini akan dibuatkan sebuah sistem berbasis IoT untuk penanganan masalah iklim mikro untuk bidang pertanian khususnya untuk pertanian jamur tiram. Dengan adanya sistem ini diharapkan akan menjadi sebuah solusi untuk para petani jamur tiram untuk memaksimalkan hasil pertaniannya.

### **1.2 Perumusan Masalah**

Permasalahan yang dibahas dalam tugas akhir ini adalah bagaimana membuat sistem pengontrolan dan monitoring iklim mikro di dalam rumah jamur tiram (kumbung) dengan teknologi berbasis IoT yang dapat diakses dari jarak jauh oleh pihak-pihak terkait (petani jamur tiram) sedemikian rupa sehingga, pengelolaan budidaya jamur tiram terasa lebih mudah (efisien) dan diharapkan hasil panennya menjadi lebih baik (lebih efektif) dibandingkan ketika dikelola secara konvensional.

Adapun implementasi metode teknologi ini dalam fase inkubasi (masa penyimpanan jamur hingga panen).

### **1.3 Tujuan**

Pada tugas akhir ini akan dibuat rancangan sistem pengontrolan dan monitoring rumah jamur tiram dengan menggunakan teknologi IoT. Adapun fokus pembahasan tugas akhir ini meliputi hal-hal sebagai berikut:

1. Membuat konsep rumah jamur tiram yang bisa diakses dari jarak jauh.
2. Membuat rancangan sistem IoT pada rumah jamur tiram.
3. Merealisasikan sistem *hardware* dan *software*.
4. Melakukan integrasi sistem IoT (kompatibilitas *hardware+software*).
5. Melakukan pengujian performansi sistem (*QoS*).

### **1.4 Batasan Masalah**

Adapun batasan masalah dari pengerjaan tugas akhir ini adalah:

- 1.Sensor suhu dan kelembaban udara menggunakan sensor DHT22.
- 2.Perangkat actuator berupa kipas DC 12V dan Humidifier 24V.
- 3.Mikrokontroler yang digunakan adalah Arduino.
- 4.Modul komunikasi (modul IoT) yang digunakan menggunakan ESP8266 yang tertanam pada NodeMCU.
5. Monitoring data melalui cloud (Blynk).
- 6.Batasan iklim jamur tiram yang diharapkan : suhu = 22-27 , dan kelembaban = 60-90%.
7. Parameter QoS yang diukur adalah Throughput, Jitter, Delay dan Packet loss

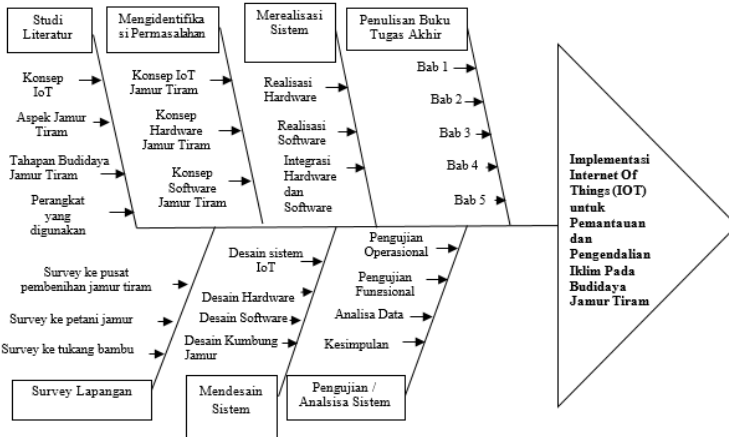
### **1.5 Metodologi**

Proses pembuatan tugas akhir ini akan dilakukan mengikuti rangkaian sebagaimana ditunjukkan diagram tulang ikan (fishbone) yang akan ditampilkan pada gambar 1.1.

Adapun langkah-langkahnya bisa dijelaskan sebagai berikut:

#### **1. Studi Literatur**

Kegiatan studi literature dimulai dari mencari dan mempelajari beberapa paper dan buku yang berhubungan dengan topik IoT dan budidaya jamur tiram. Pada tahap studi literature ini yang dilakukan ialah mempelajari tentang konsep IoT, aspek jamur tiram(kandungan gizi, manfaat, kebutuhan iklim jamur tiram).



**Gambar 1.1** Diagram Tulang Ikan

## 2. Survey Lapangan

Selain lewat studi literature, untuk menggali informasi tentang budidaya jamur tiram dilakukan wawancara lapangan dengan petani jamur, pelaku pembibitan jamur, pembuat kumbung jamur, dan penyuluh petani jamur. Tujuan kegiatan ini untuk mendapatkan informasi tentang permasalahan yang sering dihadapi oleh para pihak yang terkait dengan budidaya jamur tiram.

## 3. Mendefinisikan Permasalahan

Pada tahapan ini semua permasalahan yang muncul harus diidentifikasi sebaik-baiknya sehingga dapat menciptakan batasan-batasan masalah yang akan dikerjakan dalam tugas akhir ini. Adapun permasalahan yang akan diselesaikan dalam tugas akhir ini meliputi bagaimana mengimplementasikan sistem IoT yang cocok untuk budidaya jamur tiram, merealisasikan hardware dan softwrenya agar bisa berfungsi untuk monitoring dan kontroling budidaya jamur tiram.

## 4. Mendesain Sistem

Sistem yang akan dibuat dalam tugas akhir ini terbagi dalam beberapa sub sistem yaitu pembuatan sistem IoT jamur tiram, pembuatan rancangan hardware, dan pembuatan rancangan *software* dengan standar

suhu ruangan yang ideal untuk jamur tiram yaitu 22°C - 27°C dan standar kelembaban ruangan yaitu 60% - 90%.

Sistem yang dibuat harus mengikuti layer-layer IoT agar dapat dikategorikan sebagai perangkat IoT yaitu *things*, IoT *gateways*, *cloud* and *analytics*.

## 5. Merealisasi Sistem

Ada 3 tahap untuk merealisasikan Qumbang-Q, yaitu merealisasikan perangkat keras (*hardware*), merealisasikan perangkat lunak (*software*) yang dapat digunakan untuk memonitor dan mengendalikan kondisi rumah jamur dan mengintegrasikan antara *hardware* dan *software* hingga membentuk sistem yang dapat membantu petani dalam pengelolaan budidaya jamur tiram.

Realisasi dalam penelitian ini berupa prototipe dengan kapasitas 100 media tanam (baglog) dengan ukuran 1m x 1m x 1m.

## 6. Pengujian Sistem / Analisis Data

Untuk mengevaluasi kinerja sistem IoT pada kumbang jamur yang dibuat, akan dilakukan sejumlah pengukuran sebagai berikut :

### a. Pengujian Fungsional

Pengujian perangkat sensor dan aktuator dilakukan secara terpisah untuk memperlihatkan apakah masing-masing komponen tersebut berfungsi atau tidak.

### b. Pengujian Segmen Komunikasi

Pengujian masing-masing segmen interkoneksi. Menunjukkan proses integrasi secara keseluruhan.

### c. Pengujian Performansi Sistem

Pengujian ini meliputi pengujian QoS (Throughput, Delay, Jitter, Packetloss) pada pengaruh jumlah kumbang, pengaruh lama pengukuran dan pengaruh lokasi jarak kumbang. Pengujian ini dilakukan untuk mengetahui kinerja sistem IoT yang telah dibuat.

## 1.6 Sistematika Penulisan

Dalam tahap ini dilakukan penulisan buku laporan tugas akhir yang berisi dari beberapa Bab. Penulisan dimulai dari Bab 1 yaitu bab yang berisi tentang pendahuluan, dilanjutkan Bab 2 (Dasar Teori), Bab 3 (Metodologi), Bab 4 (Hasil dan Analisa), diakhiri dengan Bab 5 (Penutup berupa Kesimpulan dan Saran).



## **1.7 Relevansi / Manfaat**

Dengan memanfaatkan teknologi IoT dalam pengkondisian iklim mikro pada proses budidaya jamur tiram sebagaimana yang diusulkan dalam tugas akhir ini, diharapkan hasil panen nantinya menjadi lebih baik. Dengan demikian peran perkembangan teknologi (sistem IoT) benar-benar bisa memberi manfaat untuk menyelesaikan permasalahan yang selama ini menjadi momok petani jamur tiram.

## **BAB II**

### **DASAR TEORI**

#### **2.1 Internet of Things (IoT)**

*Internet of Things* (IoT), juga disebut Internet Semuanya atau Internet Industri, adalah paradigma teknologi baru yang dibayangkan sebagai jaringan global mesin dan perangkat yang mampu berinteraksi satu sama lain. IoT diakui sebagai salah satu bidang terpenting dari teknologi masa depan dan mendapatkan perhatian luas dari berbagai industri. Kesenjangan antara digital dan dunia fisik semakin menipis. Penyebaran besar benda-benda kecil 'cerdas' dengan kemampuan komunikasi mulai mewujudkan visi Internet of Things.

Di era revolusi industri 4.0 ini, Internet Of Things sudah banyak dimanfaatkan diberbagai bidang.[1] IoT dapat dimanfaatkan diberbagai bidang seperti *Smart Cities, Security & Emergencies, Smart Agriculture, Domestic & Home Automation, Medical Field, Industrial Control* dan masih banyak bidang lainnya. Adanya jaringan internet menjadi syarat untuk terealisasinya komunikasi antar perangkat yang digunakan disini. Terhubungnya perangkat yang digunakan ke web juga harus diperhatikan meskipun pada koneksi yang terbatas.

Berbagai macam perangkat dengan nama “*smart*” banyak bermunculan seiring berkembangnya teknologi IoT ini. Hampir semua bidang dapat memanfaatkan teknologi ini sebagai solusi untuk mempercepat atau membuat kerja lebih efisien.

##### **2.1.1 Layer IoT**

Di dalam IoT ada 3 (tiga) layer yang menjadi dasar arsitektur yaitu *things, IoT Gateways, Cloud and Analytics*.

##### **A. Things**

Layer pertama ini adalah layer dasar yaitu *things* atau *physical layer*. Layer ini adalah layer yang mengelompokkan beberapa perangkat seperti sensor, mikrokontroller, dan aktuator. Perangkat tersebut merupakan dasar dari IoT karena memiliki peran untuk mengambil data, dan memberi perintah aksi.

## **B. IoT Gateways**

Selain things, ada juga layer *IoT Gateways*. Layer ini mengelompokkan beberapa perangkat yang mampu memberikan jaringan internet. Seperti wifi module, NB-IoT, LORA, dan beberapa perangkat lain. Layer kedua ini menghubungkan layer things ke layer aplikasi.

## **C. Cloud and Analytics**

Layer terakhir ada layer *cloud and analytics*. Layer ini mengelompokkan berbagai aplikasi yang mampu menampilkan data yang sudah didapat kepada pengguna seperti cloud, dan web server. Selain untuk menampilkan data layer ini juga memiliki peran untuk mengolah data mentah menjadi data yang diperlukan dalam proses bisnis.

### **2.1.2 Pengembangan Teknologi IoT**

Akhir-akhir ini, bermunculan berbagai pengembangan dari teknologi IoT. Munculah sebuah teknologi baru yaitu **LPWAN** (Low Power, Wide Area Networks). Teknologi ini mampu menghubungkan sensor dan pengontrol ke internet tanpa menggunakan jaringan Wi-Fi tradisional atau jaringan seluler. Adapun beberapa teknologi baru yang berbasis standar jaringan seluler, misalnya yaitu : Sigfox, LoRa, dan juga NB-IoT.

#### **2.1.2.1 Sigfox**

Teknologi ini memakai frekuensi Ultra Narrow Band Radio untuk berkomunikasi (berkisar 200 kHz di frekuensi kerja ISM 868 MHz atau 902 MHz tergantung pada negaranya) dengan tiap paket data selebar 100 Hz yang ditransfer dengan kecepatan 100 atau 600 bps tergantung lokasi memakai modulasi DNPSK dan GFSK. Teknologi ini berbasiskan topologi bintang dimana Mini BTS-nya sanggup menjangkau jarak hingga 30-50 Km (luar kota) dan 3-10km (dalam kota). Teknologi ini telah mencapai lebih dari 10 negara di Eropa, Amerika, dan Australia dengan pusat di Prancis.

#### **2.1.2.2 NB-IoT**

Narrow Band – Internet of Things merupakan jawaban aliansi GSMA terhadap munculnya teknologi dan kebutuhan Low Power Wide Area Network sebagai bagian dari IoT. Teknologi ini menggaransi akan produk dengan kekuatan +20dB, dapat tekoneksi dengan 1000 perangkat, menggunakan bandwidth 200 kHz, kecepatan data maksimal 250 kbps, latensi 1.6-1.0 s, dan batere mampu bertahan hingga 10 tahun.

### **2.1.2.3 LoRa**

LoRaWAN (Long Range Wide Area Network) mempunyai konsep teknologi yang mirip dengan Sigfox dan menggunakan frekuensi kerja sub Giga Hertz di pita ISM (Industry, Science, and Medical). LoRa juga menggunakan konfigurasi bintang dan BTS-nya mampu menjangkau ribuan perangkat di jangkauan 2-5 Km (dalam kota) atau 15 Km (luar kota) dengan kecepatan data 0.3 – 50 kbps.

## **2.2 Bluetooth**

Teknologi Bluetooth ditemukan pada tahun 1994 oleh perusahaan IT asal Swedia yaitu Ericsson yang dibantu oleh anggota dari SIG (Special Interest Group). Pada tahun 1999 barulah perangkat Bluetooth pertama diluncurkan. Pada awalnya, Bluetooth hanya dikembangkan untuk menggantikan peran kabel pada perangkat sehari-hari seperti mouse, keyboard. Namun akhirnya Bluetooth secara fungsi berkembang sebagai penghubung dengan sarana lain.

Bluetooth beroperasi dalam pita frekuensi 2,4 GHz dengan menggunakan sebuah *frequency hopping traceiver* yang mampu menyediakan layanan komunikasi data dan suara secara real time antara host-host bluetooth dengan jarak terbatas [2]. Kelemahan teknologi ini adalah jangkauannya yang pendek dan kemampuan transfer data yang rendah. Bluetooth memang memiliki jangkauan yang pendek, namun untuk beberapa perangkat yang tidak membutuhkan jangkauan luas, Bluetooth masih menjadi solusi.

Ada banyak pemanfaatan teknologi Bluetooth dalam kehidupan sehari-hari seperti pada smartphone, printer, kamera, dan headset. Di era industry 4.0 ini, Bluetooth tidak hanya digunakan untuk media pengganti kabel saja, namun juga untuk transfer data. Beberapa produk IoT banyak yang memanfaatkan teknologi ini untuk komunikasi antar perangkatnya.

## **2.3 Quality of Service (QoS)**

Quality of Service adalah kemampuan dalam menyediakan peformasi dari jaringan komputer dalam penyediaan layanan kepada aplikasi-aplikasi di dalam jaringan komputer tersebut sehingga menentukan tingkat kepuasan dari pengguna yang menggunakan jaringan tersebut. Parameter QoS yaitu latency, jitter, packet loss, throughput dan echo cancellation. Berikut adalah tabel indeks parameter QoS :

**Tabel 2.1** Indeks Parameter QoS Menurut TIPHON

<b>Katagori</b>	<b>Presentase (%)</b>	<b>Nilai</b>
Sangat Memuaskan	95 – 100	3,8 – 4
Memuaskan	75 - 95,75	3 – 3,79
Kurang Memuaskan	50 - 74,75	2 – 2,99
Jelek	25 - 49,75	1 – 1,99

Terdapat 4 Parameter QoS yang bisa menentukan kualitas jaringan pada penelitian ini, yaitu:

### **2.3.1 Latency / Delay**

Latency atau delay didefinisikan sebagai waktu rata-rata tempuh paket yang dikirim dari alamat pengirim kelamat penerima. Semakin rendah nilai latency yang terjadi pada suatu jaringan makan semkain baik jaringan tersebut.[3]

Adapun rumus untuk menghitung delay transmisi adalah sebagai berikut:

$$Delay (sec)Tx = Received Time (s) - Sent Time (s) \quad (2.1)$$

Keterangan :

Received Time : Waktu paket piterima

Sent Time : Waktu paket dirikim

### **2.3.2 Throughput**

Throughput adalah kecepatan transfer data yang diukur dalam satuan bit per second (bps) atau biasa dipahami dengan sebutan bandwidth yang sebenarnya. Bandwidth sifatnya tetap sedangkan throughput bisa berubah bergantung dengan trafik.

$$Throughput = \frac{ukuran\ data\ yang\ diterima(bit)}{wakt\ pengiriman\ (s)} \quad (2.2)$$

### 2.3.3 Jitter

Jitter atau variasi delay, berhubungan erat dengan latency, yang menunjukkan banyaknya variasi delay pada transmisi data di jaringan. Delay antrian pada router dan switch menyebabkan jitter. Hal ini diakibatkan oleh variasi-variasi panjang antrian waktu pengolahan data, dan waktu penghimpunan ulang paket-paket di akhir perjalanan jitter. Terdapat empat kategori penurunan performansi jaringan berdasarkan nilai peak jitter sesuai.

$$Jitter = \frac{\text{Total Variasi Delay}}{\text{Total Paket Yang Diterima}-1} \quad (2.3)$$

Total variasi delay merupakan perbandingan delay dengan delay rata-rata atau secara statistic bisa disebut standar deviasi atau standar baku standar deviasi ini digunakan untuk memperlihatkan seberapa besar perbedaan data yang ada dibandingkan dari rata-rata data itu sendiri. Standar deviasi merupakan akar kuadrat dari varians dan menunjukkan standar, penyimpangan data terhadap nilai rata-ratanya.

### 2.3.4 Packet Loss

Packet loss didefinisikan sebagai kegagalan transmisi paket data mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinan, diantaranya yaitu:

- a) Terjadinya overload trafik didalam jaringan,
- b) Tabrakan (*congestion*) dalam jaringan,
- c) Error yang terjadi pada media fisik,
- d) Kegagalan yang terjadi pada sisi penerima antara lain bisa disebabkan karena overflow yang terjadi pada buffer.

Dalam sebuah jaringan, nilai packet loss diharapkan mempunyai nilai yang minimum. Paket loss juga dapat dihitung dengan rumus:

$$Packet Loss = \frac{\text{Packet transmitted}-\text{Packet received}}{\text{Packet transmitted}} \times 100\% \quad (2.4)$$

## 2.4 Jamur Tiram

Jamur merupakan (*Pleurotus Ostreatus*) tanaman yang tidak memiliki klorofil yang artinya jamur tidak bisa melakukan proses fotosintesis untuk menghasilkan makanannya sendiri. Tanaman ini dapat hidup dengan cara mengambil zat-zat makanan dari organisme lain.

Dengan dibantu oleh enzim yang diproduksi oleh hifa (bagian jamur yang bentuknya seperti benang halus). Oleh karena itu jamur digolongkan tanaman yang heterofik, yaitu tanaman yang kehidupannya tergantung pada organisme lain[4].

Terdapat beberapa jenis jamur tiram yang dibudidayakan petani, diantaranya adalah jamur tiram putih (*Pleurotus ostreatus*), jamur tiram coklat (*Pleurotus Abalonus*), jamur tiram kuning (*Pleurotus sp.*). Adapula beberapa jenis jamur tiram yang digolongkan berdasarkan warna-warnanya. Tanaman ini merupakan salah satu kelompok jamur yang dapat dikonsumsi (edible) dan memiliki rasa yang cukup khas. Jamur ini juga termasuk jamur kayu yang tumbuh pada pokok-pokok kayu yang lapuk dan tumbuh tergantung kelembaban, aur, cahaya, udara, dan keasaman dari lingkungannya.

#### **2.4.1 Kandungan gizi jamur tiram**

Di Indonesia jamur tiram banyak digemari oleh masyarakat karena rasanya yang khas, selain rasanya yang digemari oleh masyarakat, kandungan gizi jamur tiram juga banyak dibutuhkan oleh tubuh manusia. Ada beberapa macam vitamin dan mineral yang terkandung didalam jamur tiram.

Nilai gizi dari jamur sebanding dengan telur, susu dan daging. Jamur tiram ini merupakan sumber yang kaya protein, mineral (P, Ca, Fe, K, dan Na) dan vitamin (Tiamin, riboflavin, asam folat, dan niasin). Kadar kandungan protein jamur diantara hewan dan sayuran

Beberapa spesies jamur tiram sangat bermanfaat dibidang kedokteran, seperti *Pleurotus Cystidiosus* (PC) mempunyai aktivitas antioksidan yang kuat. *Pleurotus ostreatus* (PO) juga memiliki aktivitas antitumor dan memiliki efek hipoglikemik pada penderita diabetes. *Pleurotus* berkhasiat obat karena komposisi kimia atau nilai gizinya.

Kandungan jamur tiram pada umumnya sebagai berikut : Lemak (1.1-2.4%), protein total (10.5-44%), karbohidrat (50.7-81,8%), abu (6.1-9.8%), kalori (245-367 Kcal), serat (7.5-12.4%), kadar air (73.7-92.2%), Vit B complex (1.7-4.8 mg/g), Niacin (108.7 mg/g).

*Pleurotus ostreatus* memiliki nilai gizi tertinggi tergantung pada kehadiran tingkat tinggi penting asam amino (arginin, alanin, glutamin, dan glutamat asam), karbohidrat (pati tidak ada, tetapi menemukan

glukan, manitol dan trehalosa), kadar air (dari 80 sampai 90%), protein (40%), vitamin B, C, D, K, tiamin, riboflavin, asam folat, dan niasin [5] dan [6] mineral (Ca, P, Fe, K, Mn, Cu, Zn, Mg dan Se[7]. Selain itu, *P. ostreatus* memiliki kalori yang rendah (masing-masing 100 g memiliki 28 k / Cal) dan natrium. Oleh karena itu, jamur memiliki nilai obat tinggi dan digunakan untuk berbagai tahun dalam obat rakyat. Ini memiliki banyak bioaktif metabolit digunakan sebagai sumber yang belum dimanfaatkan terbesar produk farmasi yang kuat dan [8]. Kandungan asam pada jamur akan ditampilkan dalam tabel 2.2.

**Tabel 2.2** Kandungan Asam Pada Jamur Tiram [9]

No	Nama Asam	Kandungan Per 100 G (G)
1	Alanine	7.0
2	Arginine	6.3
3	Cystine	0.6
4	Glycine	5.9
5	Histidine	2.4
6	Leucine	12.6
7	Lysine	6.3
8	Proline	5.4
9	Serine	6.3
10	Aspartic acid	9.3
11	Glutamic acid	17.0
12	Phenylalanine	4.1
13	Tyrosine	2.61
14	Tryptophan	0.3
15	Methionine	2.1
16	Valine	6.3
17	Threonine	6.8
18	Isoleusine	0.3

#### **2.4.2 Budidaya Jamur Tiram**

Tingginya peminat jamur tiram di Indonesia, menjadikan budidaya jamur tiram sebagai sebuah mata pencaharian. Petani jamur tersebar di berbagai lokasi di Indonesia. Adapun beberapa tahapan dalam budidaya jamur tiram adalah sebagai berikut:



### **A. Pembuatan Media Tanam**

Dalam pembuatan media tanam, ada beberapa bahan yang dapat digunakan seperti serbuk kayu, kacang kedelai, dan biji jagung. Media tanam harus melalui proses pengayaan sebelum dicampur dengan beberapa bahan seperti bekatul, dan kapur. Tujuannya adalah untuk meratakan media tanam agar nantinya jamur dapat tumbuh secara merata. Setelah melalui proses pengayaan, selanjutnya bahan yang digunakan untuk media tanam akan dicampur dengan beberapa bahan bekatul dan kapur masing-masing 20% dan 5%. Selanjutnya akan dicampur dengan air sebanyak 60%. Setelah melalui proses pencampuran, media tanam akan melalui proses pengomposan, proses ini dilakukan selama lima hari. Terjadinya fermentasi pada media ditunjukkan dengan berubahnya warna media tanam menjadi lebih gelap[9].

### **B. Pembuatan Baglog**

Setelah proses fermentasi, maka akan dilakukan proses pembuatan baglog. Baglog adalah media tanam yang sudah siap dipakai untuk menanam, biasanya ditempatkan pada kantung plastic. Plastic yang digunakan adalah plastic *polipropilen* ukuran 1500g dengan berat total media tanam yaitu 1000g. Adapun opsi lain untuk baglog ini seperti botol air mineral ukuran 1,5 l [9].

### **C. Sterilisasi**

Proses ini dilakukan menggunakan *autoclave* pada suhu 121° C selama 45 menit. Media yang sudah disterilisasikan kemudian di dinginkan selama 8-12 jam. Proses pendinginan dilakukan bertujuan agar pada saat media tanam diinokulasi, bibit jamur tidak akan mati.

### **D. Inokulasi**

Inokulasi dilakukan di ruang khusus yang sudah disterilisasi dengan menyemprotkan alkohol 70%. Cara yang dilakukan dengan membuka penutup baglog kemudian bagian ujung dari baglog didekatkan pada bunsen, bibit jamur dimasukkan lewat cincin paralon bagian tengah dalam media. Inokulasi ini dilakukan satu per satu baglog [10].

## E. Inkubasi

Proses ini dilakukan dengan cara menyimpan pada rumah jamur dengan kondisi tertentu yang bertujuan supaya miselium jamur tumbuh dengan baik. Semua baglog ditempatkan di rak kayu dengan posisi horizontal dan dibiarkan sampai miselium jamur tiram putih tumbuh memenuhi seluruh baglog. Kondisi ruangan inkubasi diatur dengan suhu 27-30C dengan kelembaban 60- 70%. Suhu dan kelembaban dalam ruangan dapat diatur dengan pengaturan sirkulasi udara dan penyiraman pada lantai kumbung apabila diperlukan. Kelembaban dan suhu diukur menggunakan termometer ruangan dan higrometer. Inkubasi diakhiri setelah 5-6 minggu yang ditandai dengan adanya miselium yang tampak putih merata menyelimuti seluruh permukaan media tanam.

### 2.4.3 Siklus Hidup Jamur Tiram

Masa hidup jamur tiram biasanya mencapai 60 hari hingga 120 hari, bergantung pada beberapa faktor seperti bibit jamur yang digunakan, cuaca pada saat proses inkubasi, dan beberapa faktor eksternal lain. Pada tabel 2.3 akan dijelaskan masa pertumbuhan jamur tiram.

**Tabel 2.3** Masa Pertumbuhan Jamur Tiram Pada Media Tanam

No	Periode	Indikasi
1	Minggu 1	Mulai tumbuh bakalan jamur tiram
2	Minggu 2	Bakalan semakin banyak
3	Minggu 3	Jamur sudah terlihat memiliki diameter mahkota
4	Minggu 4	Jamur sudah mulai dapat dipanen
5	Minggu 5	Jamur sudah dapat dipanen dan dibiarkan tumbuh lagi untuk menunggu panen ke-2
6	Minggu 6-7	Jamur akan menunjukkan proses pertumbuhan lagi
7	Minggu 8	Jamur sudah dapat dipanen ke-2

Sebagaimana yang tertampil di tabel 2.3, mengingat potensi hidup jamur tiram bisa 60 – 120 hari, masa ini lah yang memerlukan intervensi bebas teknologi dan dalam tugas akhir ini akan dibuat sebuah perangkat

yang akan mengontrol iklim mikro agar menambah potensi hidup jamur tiram.

#### **2.4.4 Kendala Budidaya Jamur Tiram**

Dari survey lapangan, didapatkan bahwa suhu dan kelembaban merupakan factor utama dalam budidaya jamur tiram. Adapun factor lain yang dapat menjadi kendala adalah PH, hama, dan juga curah hujan. Untuk suhu yang ideal menurut hasil survey lapangan adalah 22°-28°C dan untuk kelembaban ideal ada di 60% - 90%.

Beberapa petani jamur tiram, memilih tidak menanam jamur apabila suhu dan kelembaban lingkungan itu tidak ideal, karena hasil panen yang tidak menguntungkan. Apalagi sekarang ini, kalender musim yang tidak menentu menjadikan para petani jamur mendapatkan keuntungan yang minim.

Jika suhu ruangan diatas 30°C maka jamur tiram sulit untuk tumbuh, karena jamur tiram membutuhkan suhu yang rendah. Namun jika kelembabannya juga terlalu tinggi, maka akan timbul masalah yaitu hama dan tumbuhnya jamur lain didalam baglog. Oleh karena itu pengontrolan suhu dan kelembaban pada kandang jamur menjadi salah satu solusi untuk meningkatkan hasil panen jamur tiram bagi petani jamur.

### **2.5 Bahasa Pemrograman C**

Dalam pemrograman, ada beberapa bahasa yang diketahui dan digunakan salah satunya adalah bahasa C. Akar dari bahasa C adalah bahasa BPCL yang dikembangkan oleh Martin Richards pada tahun 1967. Bahasa ini kemudian dikembangkan menjadi bahasa B pada tahun 1970. Bahasa B kemudian dikembangkan lagi menjadi bahasa C pada tahun 1970-an di Bell Telephone Laboratories [11].

#### **2.5.1 Fungsi**

Fungsi merupakan bagian dari program yang dimaksudkan untuk mengerjakan sebuah proses[11]. Suatu fungsi dapat dibuat dalam program yang sama (fungsi internal) maupun dari luar (fungsi eksternal). Tanda '=' digunakan untuk memberikan nilai variabel. Kalimat yang digunakan untuk memberi nilai dalam sebuah variabel disebut assignment.

## 2.5.2 Elemen Dasar Bahasa C

Adapun beberapa elemen dasar pada bahasa C yaitu[11]:

### A. Identifier (Pengenal)

Pengenal ini digunakan untuk menyatakan sebuah variabel, konstanta, tipe data, dan fungsi. Aturan dalam penulisan pengenal ini adalah :

- Tidak boleh dimulai dengan karakter non huruf
- Tidak boleh ada spasi
- Tidak boleh menggunakan karakter : ~ ! @ # \$ % ^ & \* ( ) + - = { } [] : ; ' < > ? , . ? / |

### B. Deklarasi Variabel

Sebelum digunakan, variabel harus dideklarasikan agar bisa dibaca oleh program. Dengan kata lain deklarasi ini digunakan untuk mengenalkan variabel ke program dan menentukan tipe datanya.

### C. Assignment

Assignment merupakan proses pemberian nilai kepada suatu variabel yang sudah dideklarasikan.

## 2.6 Cloud Computing

Cloud computing merupakan sebuah pengembangan dari kombinasi teknologi computer dan jaringan internet. Cloud sendiri merupakan istilah yang diberikan untuk jaringan internet. Tujuan dari cloud sendiri adalah pengguna yang dapat mengakses servernya dari manapun selama terhubung dengan jaringan internet.

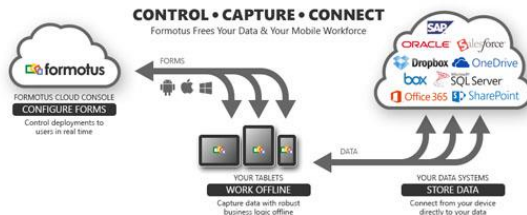
### 2.6.1 Jenis Cloud Computing

Berdasarkan jenis layanannya cloud dibagi menjadi tiga [12] yaitu:

#### A. SaaS (Software as a Service)

SaaS adalah salah satu layanan dari *Cloud Computing* dimana kita tinggal memakai *software* (perangkat lunak) yang telah disediakan[12]. User hanya tahu bahwa perangkat lunak bisa berjalan dan bisa digunakan dengan baik .

Contoh, layanan email publik (Gmail, YahooMail, Hotmail), *social network* (Facebook, Twitter, LinkedIn) *instant messaging* (Yahoo Messenger, Skype, Line, WhatsApp) dan masih banyak lagi yang lain.



**Gambar 2.1** Software as a Service

## B. PaaS (Platform as a Service)

Adalah layanan dari *Cloud Computing* kalau kita analogikan dimana kita menyewa “rumah” berikut lingkungan-nya (sistem operasi, network, database engine, framework aplikasi, dll), untuk menjalankan aplikasi yang kita buat.

Kita tidak perlu pusing untuk menyiapkan “rumah” dan memelihara “rumah” tersebut. Yang penting aplikasi yang kita buat bisa berjalan dengan baik di “rumah” tersebut. Untuk pemeliharaan “rumah” ini menjadi tanggung jawab dari penyedia layanan.

Sebagai analogi, misal-nya kita sewa kamar hotel, kita tinggal tidur di kamar yang sudah kita sewa, tanpa peduli bagaimana “perawatan” dari kamar dan lingkungan-nya. Yang penting, kita bisa nyaman tinggal di kamar itu, jika suatu saat kita dibuat tidak nyaman, tinggal cabut dan pindah ke hotel lain yang lebih bagus layanan-nya.

Contoh penyedia layanan *PaaS* ini adalah: Amazon Web Service, Windows Azure, bahkan tradisional hosting-pun merupakan contoh dari *PaaS*.

Keuntungan dari *PaaS* adalah kita sebagai pengembang bisa fokus pada aplikasi yang kita buat, tidak perlu memikirkan operasional dari “rumah” untuk aplikasi yang kita buat.



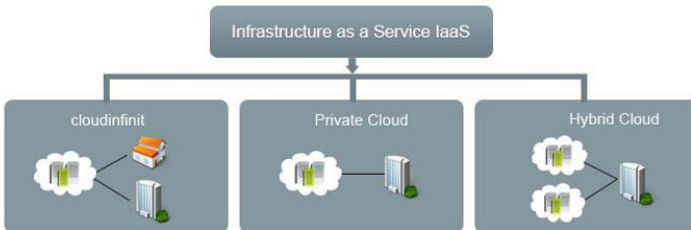
**Gambar 2.2** Platform as a Service

### C. IaaS (Infrastructure as a Service)

Adalah layanan dari *Cloud Computing* dimana kita bisa “menyewa” infrastruktur IT (komputasi, storage, memory, network). Kita bisa definisikan berapa besar-nya unit komputasi (*CPU*), penyimpanan data (*storage*), memory (*RAM*), bandwidth, dan konfigurasi lain-nya yang akan kita sewa.

Mudah-nya, *IaaS* ini adalah menyewa komputer virtual yang masih kosong, dimana setelah komputer ini disewa kita bisa menggunakan-nya terserah dari kebutuhan kita. Kita bisa install sistem operasi dan aplikasi apapun diatas-nya. Contoh penyedia layanan *IaaS* ini adalah: Amazon EC2, Windows Azure (*soon*), TelkomCloud, BizNetCloud, dan sebagainya.

Keuntungan dari *IaaS* ini adalah kita tidak perlu membeli komputer fisik, dan konfigurasi komputer virtual tersebut bisa kita rubah (*scale up/scale down*) dengan mudah. Sebagai contoh, saat komputer virtual tersebut sudah kelebihan beban, kita bisa tambahkan CPU, RAM, Storage dan lainnya dengan segera.

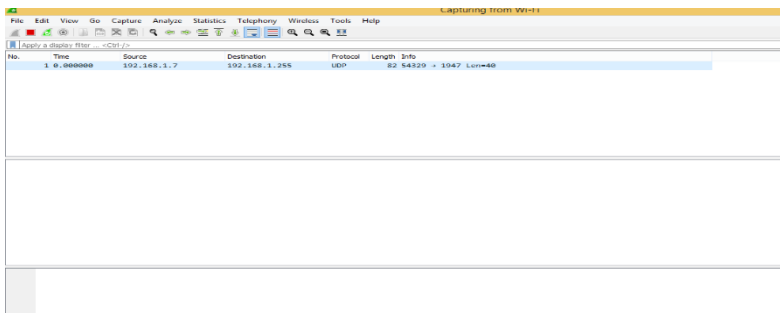


**Gambar 2.3** Infrastructure as a Service

## 2.7 Wireshark

Wireshark merupakan sebuah packet analyzer yang bersifat free dan open source. Aplikasi ini digunakan untuk troubleshooting jaringan, analisa paket data dan pengembangan protokol komunikasi. Wireshark merupakan aplikasi cross-platform, menggunakan GTK+ widget toolkit pada user interface-nya dan menggunakan npcap untuk menangkap paket data.

Wireshark juga dikenal sebagai packet sniffer, yaitu sebagai sebuah program atau tool yang memiliki kemampuan untuk mencegah dan melakukan pencatatan terhadap traffic data dalam jaringan. Selama terjadi aliran data dalam, packet sniffer dapat menangkap protocol data unit (PDU), melakukan dekode serta melakukan analisis terhadap isi paket berdasarkan spesifikasi RFC atau spesifikasi-spesifikasi yang lain.



**Gambar 2.4** Tampilan Wireshark

### 2.7.1 Kegunaan Dari Wireshark

Kegunaan Wireshark ini terbagi menjadi 6 yaitu[13]:

1. Wireshark dipakai oleh network administrator untuk menganalisa kinerja jaringannya. Wireshark mampu menangkap paket-paket data atau informasi yang berjalan dalam jaringan yang terlihat dan semua jenis informasi ini dapat dengan mudah dianalisa yaitu dengan memakai sniffing , dengan sniffing diperoleh informasi penting seperti password email account lain.

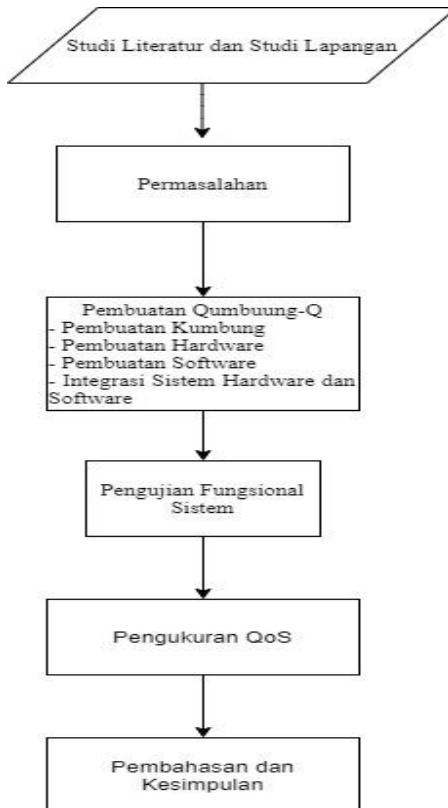
2. Wireshark merupakan software untuk melakukan analisa lalu-lintas jaringan komputer, yang memiliki fungsi-fungsi yang amat berguna bagi profesional jaringan, administrator jaringan, peneliti, hingga pengembang piranti lunak jaringan.
3. Wireshark dapat membaca data secara langsung dari Ethernet, Token-Ring, FDDI, serial (PPP dan SLIP), 802.11 wireless LAN, dan koneksi ATM.
4. Program ini juga sering digunakan oleh chatters untuk mengetahui ip korban maupun para chatter lainnya lewat typingan room.
5. Tool wireshark dapat menganalisa transmisi paket data dalam jaringan, proses koneksi dan transmisi data antar komputer.
6. Selama kita bisa mendapatkan paket langsung dari jaringan, dengan tools seperti wireshark, maka kita juga bisa memanfaatkan wireshark untuk ‘menyadap’ pembicaraan Voice over IP.



## BAB III METODOLOGI PENELITIAN

### 3.1 Diagram Alir Penelitian

Pada penelitian ini akan dilakukan beberapa langkah yang berkaitan dengan proses pembuatan Qumbung-Q, hingga hasil analisis. Berikut metodologi penelitian yang digunakan pada penelitian ini dan akan ditampilkan pada Gambar 3.1



**Gambar 3.1** Diagram Alir Metodologi Penelitian

### **3.2 Permasalahan Budidaya Jamur Tiram**

Berbagai masalah pada budidaya jamur tiram dialami oleh para petani jamur dalam proses budidaya jamur tiram. Kalender musim yang tak menentu menjadikan para petani kesulitan untuk mendapatkan iklim ideal untuk membudidaya jamur. Jamur tiram adalah tanaman yang membutuhkan kelembaban tinggi dan suhu yang rendah.

Namun ketika kondisi lingkungan jamur tiram memiliki tingkat kelembaban terlalu tinggi (>90%) maka pada media tanam jamur akan ditumbuhi oleh jamur lain. Namun apa bila kondisi lingkungan jamur tiram memiliki kelembaban yang terlalu rendah (<70%) maka jamur akan mengalami gangguan absorpsi nutrisi sehingga menyebabkan kekeringan dan gangguan pertumbuhan atau pun kematian. Suhu pada kumbung jamur juga sebisa mungkin stabil berkisar 20-30°C agar jamur dapat tumbuh secara optimal. Proses ini adalah proses inkubasi dimana jamur akan didiamkan di rumah jamur yang sudah disiapkan.

Oleh karena itu dibutuhkan sebuah solusi untuk menyelesaikan masalah yang terjadi pada masa inkubasi. Salah satunya adalah dengan memanfaatkan teknologi untuk menjamin kondisi iklim mikro pada rumah jamur agar memiliki suhu dan kelembaban yang ideal dan stabil untuk pertumbuhan jamur tiram. Untuk memaksimalkan siklus hidup jamur tiram yang bisa hidup 60-120 hari maka harus ada sebuah jaminan iklim yang ideal.

#### **3.2.1 Survey Lapangan**

Kegiatan survey lapangan ini dilakukan untuk mendapat informasi langsung dari petani jamur terkait masalah yang dihadapi oleh petani jamur dalam proses budidaya jamur tiram.

Dari hasil survey lapangan, petani jamur mengharapkan bahwa suhu dari rumah jamur di angka 22 °C - 27°C. Namun beberapa daerah yang ada di Indonesia, sangat sulit untuk mendapatkan suhu yang ideal tersebut. Dan untuk kelembaban yang ideal untuk jamur menurut hasil survey lapangan adalah lebih dari 80%. Kelembaban ini hanya bisa dicapai ketika musim hujan.

Selama ini, petani jamur hanya menggunakan cara manual untuk mendapatkan suhu dan kelembaban yang ideal tersebut yaitu dengan cara menyemprotkan air tiga kali dalam sehari. Namun cara ini juga dapat menimbulkan masalah lain yaitu tumbuhnya jamur lain pada media tanam

karena kelembaban yang terlalu tinggi yang disebabkan oleh menggenangnya air yang disemprotkan ke lantai dari kumbung jamur.

### **3.2.2 Studi Literatur**

Studi literatur ini juga perlu dilakukan untuk menunjang hasil dari survey lapangan yang sudah dilakukan. Dari kegiatan ini didapatkan beberapa faktor lain dalam budidaya jamur yang dapat menyebabkan kegagalan dalam budidaya jamur selain yang sudah disebutkan oleh petani jamur dalam kegiatan survey lapangan.

Pada rumah jamur yang memiliki suhu dan kelembaban yang tidak ideal maka akan timbul beberapa permasalahan. Seperti ketika rumah jamur memiliki tingkat kelembaban terlalu tinggi, maka akan menyebabkan tumbuhnya jamur lain pada media tanam dan juga dapat menyebabkan jamur membusuk (Istuti & Nurbana, 2006). Dan ketika rumah jamur memiliki tingkat kelembaban yang rendah (<80%) maka akan mengalami gangguan absorpsi nutrisi sehingga menyebabkan kekeringan dan gangguan pertumbuhan atau pun kematian. Suhu pada kumbung jamur juga sebisa mungkin stabil, agar jamur dapat tumbuh secara optimal. Temperatur yang dibutuhkan untuk pertumbuhan miselium adalah 20 °C – 30 °C. Temperatur untuk pembentukan tubuh buah (fruiting body) berkisar antara 22 °C - 26 °C, angka ini dianggap suhu optimal untuk budidaya jamur tiram.

### **3.2.3 Identifikasi Kebutuhan**

Dari hasil survey lapangan dan studi literatur, didapatkan beberapa faktor yang dibutuhkan dalam proses budidaya jamur tiram agar mendapatkan hasil yang maksimal.

Pertama adalah suhu yang harus stabil diangka 22 °C – 27 °C. Suhu ini harus dicapai untuk mendapatkan hasil panen yang maksimal. Namun pada realitanya suhu tersebut sangat lah sulit untuk dicapai dan untuk distabilkan. Kalender musim yang tak menentu menjadi sebuah masalah tersendiri untuk para petani jamur yang hanya memanfaatkan iklim untuk mendapatkan hasil panen maksimal.

Selain suhu, kelembaban yang tinggi juga menjadi faktor lain untuk mendapatkan hasil yang maksimal dalam budidaya jamur tiram. Tingkat kelembaban yang ideal bagi jamur tiram adalah 60 – 90%. Kondisi ini

hanya bisa dicapai dimusim hujan saja. Atau dengan beberapa cara seperti menyempatkan air pada kumbung jamur.

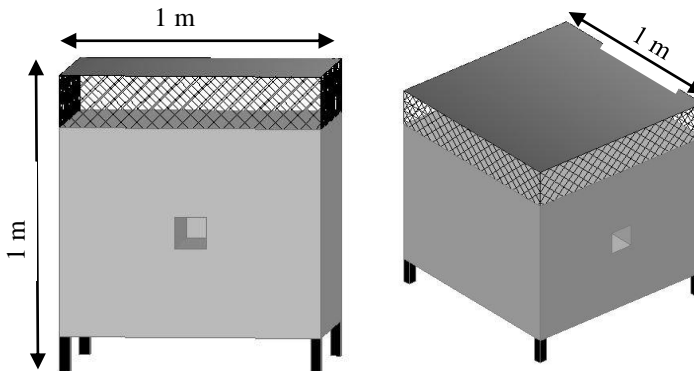
Oleh karena itu, dibutuhkan sebuah interfensi yang bisa menjadi sebuah solusi untuk masalah ini. Salah satunya adalah dengan cara memanfaatkan teknologi. Solusi yang diperlukan adalah pengadaan rumah jamur yang dapat menciptakan suhu dan kelembaban yang ideal dan stabil untuk budidaya jamur tiram dengan memanfaatkan teknologi IoT yang akan dibuat dalam penelitian ini dengan nama Qumbung-Q.

### 3.3 Pembuatan Rumah Jamur Berbasis IoT (Qumbung-Q)

Pada bagian ini akan dibahas langkah-langkah pembuatan rumah jamur tiram berbasis IoT meliputi proses pembuatan hardware yang terdiri dari pembuatan rumah jamur (kumbung), pembuatan hardware elektronik untuk dua kumbung dan pembuatan software.

#### 3.3.1 Pembuatan Kumbung Jamur

Proses pembuatan kumbung jamur meliputi beberapa proses seperti desain rumah jamur, dan realisasi rumah jamur.



**Gambar 3.2** Desain kumbung

Desain kumbung ini adalah desain skala kecil dalam budidaya jamur tiram dengan kapasitas 100 baglog (media tanam) jamur tiram. Namun desain ini dianggap dapat menjadi solusi bagi masyarakat yang ingin membudidayakan jamur tiram namun tidak memiliki tempat yang luas.

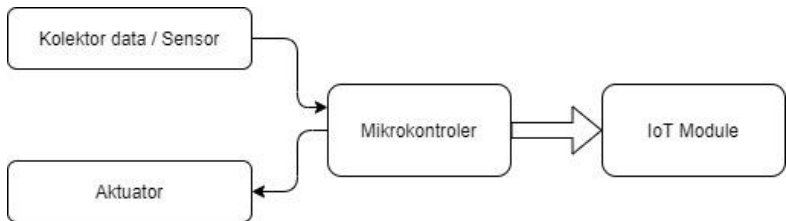
Kumbung jamur ini dibuat dengan minimalis dengan ukuran 1m<sup>3</sup>. Dalam pembuatannya, bahan yang dipilih adalah bambu sebagai kerangka dan triplek sebagai dindingnya. Bahan bambu dipilih karena kayu dianggap tidak mudah keropos apabila terkena air dan tahan akan kelembaban tinggi. Sementara untuk mengatasi masalah kebutuhan cahaya, maka ada dinding dari rumah jamur dibuat dari paranet yang bertujuan untuk masuknya cahaya matahari namun tidak secara langsung mengenai jamur tiram. Paranet juga dipilih karena dapat menjadi lubang sirkulasi udara pada kumbung. Untuk meletakkan kipas DC 12V, akan diletakkan pada sisi dinding jamur yang terbuat dari bambu, sedangkan peletakkan humidifier diletakkan pada tengah kumbung sejajar dengan peletakkan sensor suhu dan kelembaban. Dan untuk meletakkan komponen-komponen lain diletakkan pada sisi luar dari kumbung untuk mempermudah saat melakukan pemeliharaan

### 3.3.2 Pembuatan Hardware

Proses pembuatan hardware, meliputi beberapa proses yang ada didalamnya yaitu merancang perangkat keras, memilih komponen / divais yang dibutuhkan, merangkai sistem, dan merealisasi perangkat keras.

#### 3.3.2.1 Rancangan Perangkat Keras

Merancang perangkat keras dalam kumbung juga diperlukan sebelum merealisikannya. Tujuannya adalah mendapat rancangan yang ideal dan pas untuk dipasang pada kumbung.



**Gambar 3.3** Diagram Blok Sistem Perangkat Keras

#### 3.2.2.2 Pemilihan Komponen / Divais

Dalam pembuatan alat ini, dibutuhkan beberapa komponen yang dibutuhkan dan dianggap mampu bekerja secara optimal dan kompatibel. Pada Tabel 3.1 akan ditampilkan beberapa perangkat yang dibutuhkan dalam pembuatan hardware dalam penelitian ini.

**Tabel 3.1** Daftar perangkat yang digunakan

No	Nama Perangkat	Fungsi	Keterangan
1	DHT 22	Colector Data	Sensor suhu dan sensor kelembaban
2	Arduino Nano	Mikrokontroler	Untuk mengolah data dan mengontrol aktuator yang ada pada kumbung jamur.
3	Node MCU ESP 8266	Internet Gateway	Untuk menyambung ke jaringan internet dan server platform IoT
4	HC-05 Bluetooth Module	Modul komunikasi	Untuk mengirim dan menerima data dari mikrokontroler
5	Relay Module	Switch	Untuk mengontrol on/off dari aktuator
6	Kipas DC 12V	Aktuator	Mengatur suhu pada kumbung
7	Humidifier	Aktuator	Mengatur kelembaban pada kumbung
8	Adaptor 12V	Power Supply	Sumber tegangan

### 3.3.2.3 Biaya Pembuatan

Untuk memenuhi kebutuhan divais yang digunakan, berikut akan ditampilkan harga dari masing-masing perangkat yang digunakan dalam penelitian ini yang akan ditampilkan pada tabel 3.2.

Perangkat-perangkat yang digunakan memiliki harga yang bervariasi tergantung dari penjual. Harga yang tertera di tabel 3.2 adalah harga dari penjual online (Tokopedia).

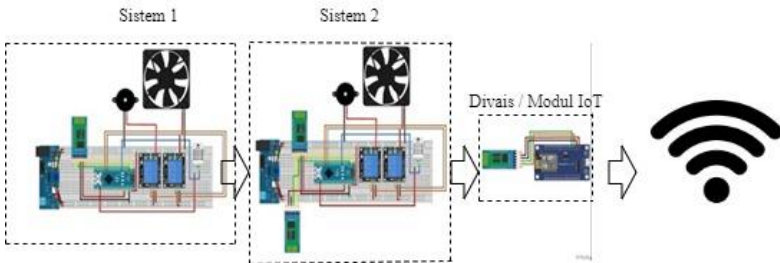
**Tabel 3.2** Daftar Harga Perangkat Keras

No	Nama Perangkat	Harga Satuan (Rp)	Jumlah	Total (Rp)
1	Arduino Nano	40.000	2	80.000
2	NodeMCU ESP 8266	45.000	1	45.000

2	DHT 22	47.000	2	94.000
3	Bluetooth Module HC-05	40.000	4	160.000
4	Kipas DC 12 V	18.000	1	18.000
5	Humidifier+Adaptor	100.000	1	100.000
6	Relay 4 channel	40.000	1	40.000

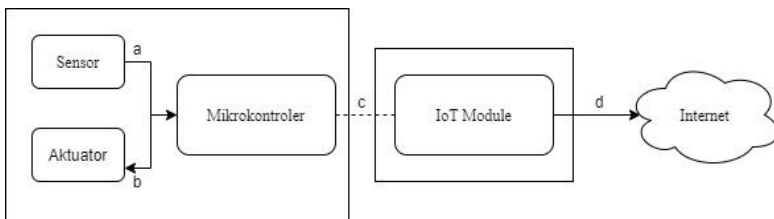
### 3.3.2.4 Rangkaian Sistem

Rangkaian sistem hardware yang akan dibuat sudah meliputi beberapa komponen seperti sensor DHT 22 yang merupakan sensor kelembaban sekaligus sensor suhu, Arduino nano yang merupakan mikrokontroler, Node MCU ESP 8266 yang merupakan internet gateway, modul Bluetooth HC-05, Kipas DC 12V, Humidifier, modul relay, dan adaptor 12 V.



**Gambar 3.4** Rangkaian Perangkat Keras

Pada gambar 3.5 akan dijelaskan gambar sistem IoT pada perangkat yang akan dibuat.



**Gambar 3.5** Rangkaian Sistem IoT

Untuk keterangan gambar diatas dapat dilihat pada tabel 3.3 yang berisikan keterangan gambar 3.5.

**Tabel 3.3** Keterangan Blok

<b>Blok</b>	<b>Sistem Komunikasi</b>	<b>Keterangan</b>
A	Wireline	Menggunakan kabel
B	Wireline	Menggunakan kabel
C	Wireless	Menggunakan modul wireless
D	Wireless	Terhubung ke internet

### **3.3.2.5 Kalibrasi Sensor**

Proses kalibrasi sensor ini bertujuan untuk mendapatkan batas toleransi pada sensor yang digunakan dan untuk mengetahui apakah sensor yang digunakan mampu memberikan data parameter yang dicari dengan baik.

Proses ini dilakukan dengan cara membandingkan hasil yang terbaca oleh sensor dengan hasil dari thermometer digital. Hasilnya diketahui bahwa sensor yang digunakan mampu mendeteksi suhu dengan tepat namun hanya mampu membaca 1 angka dibelakang koma saja.

### **3.3.2.6 Realisasi Sistem Perangkat Keras**

Pada tahapan ini, seluruh rangkaian akan direalisasikan. Adapun beberapa sistem yang direalisasi adalah sistem untuk kumbang 1 (Q1), sistem untuk kumbang 2 (Q2), dan modul IoT.

#### **A. Sistem Kumbang 1 (Q1)**

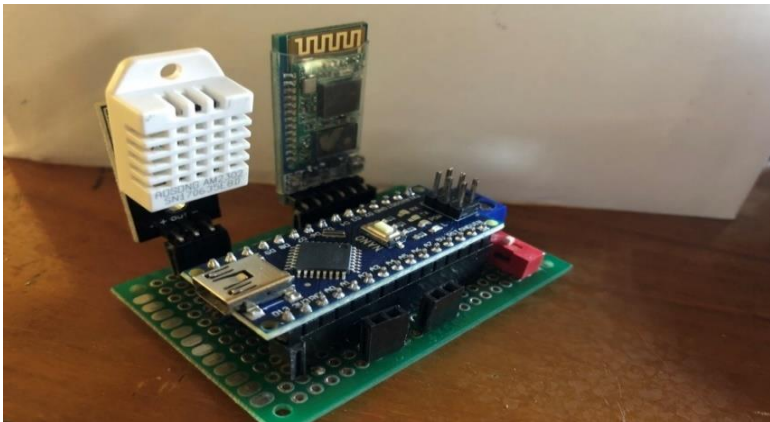
Pada realisasi sistem 1, sensor DHT 22 dihubungkan ke mikrokontroler Arduino Nano sesuai dengan datasheet dari arduino nano. Selanjutnya aktuator juga dihubungkan kepada Arduino Nano, yang dihubungkan dikaki 5V dari Arduino Nano. Untuk mengatur kondisi nyala dan mati dari aktuator, diperlukan relay yang dihubungkan dengan mikrokontroler dan aktuator. Selanjutnya untuk pemasangan modul Bluetooth, dipasang terbalik antara Tx dan Rx-nya. Tx dari modul Bluetooth dipasangkan ke kaki Rx dari Arduino Nano, dan kaki Rx pada modul Bluetooth dipasangkan ke kaki Tx pada Arduino. Hal tersebut dilakukan mengikuti standar pemasangan modul Bluetooth. Bluetooth



pada sistem 1 ini hanya berfungsi sebagai Tx karena hanya berfungsi untuk mengirimkan data ke Sistem 2. Dan untuk power supply ada beberapa opsi untuk hal ini, diantara lain menggunakan port supply yang ada pada Arduino Nano, atau dengan memasang modul power supply. Pada Q1 menggunakan port supply yang ada pada Arduino Nano.

Q1 ini berfungsi sebagai node 1 yang berisikan sensor, mikrokontroler, Bluetooth module dan aktuator. Dan seperti yang sudah dijelaskan sebelumnya Q1 akan mengirimkan data yang didapat ke sistem 2 (Q2).

Adapun realisasi sistem kumbang 1 / Q1 tampak seperti gambar 3.5 dibawah ini.



**Gambar 3.5** Realisasi Sistem Q1

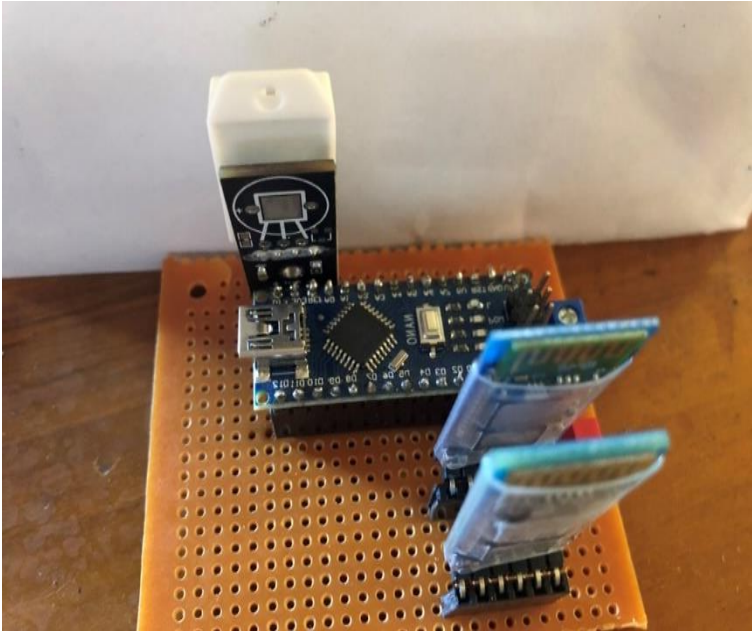
## **B. Sistem Kumbang 2 (Q2)**

Pembuatan Q2 ini tidak jauh berbeda dengan pembuatan Q1. Adapun pembeda antara keduanya adalah adanya dua modul Bluetooth HC-05 pada rangkaianannya. Modul Bluetooth pertama berfungsi sebagai penerima (Rx) dari pengirim (Tx) yang ada pada Q1. Modul Bluetooth yang kedua berfungsi untuk mengirim (Tx) ke divais IoT. Untuk pemasangan divais pada Q2 juga mengikuti datasheet dari Arduino Nano.

Q2 dalam sistem berfungsi sebagai layer Things yang berisikan mikrokontroler, sensor, aktuator dan Bluetooth module sebagai media

komunikasinya. Pada Q2, data dari Q1 akan diterima dan selanjutnya akan dikirimkan ke module IoT.

Sebagaimana tampak pada gambar 3.6 akan ditampilkan realisasi sistem Q2.



**Gambar 3.6** Realisasi Sistem Q2

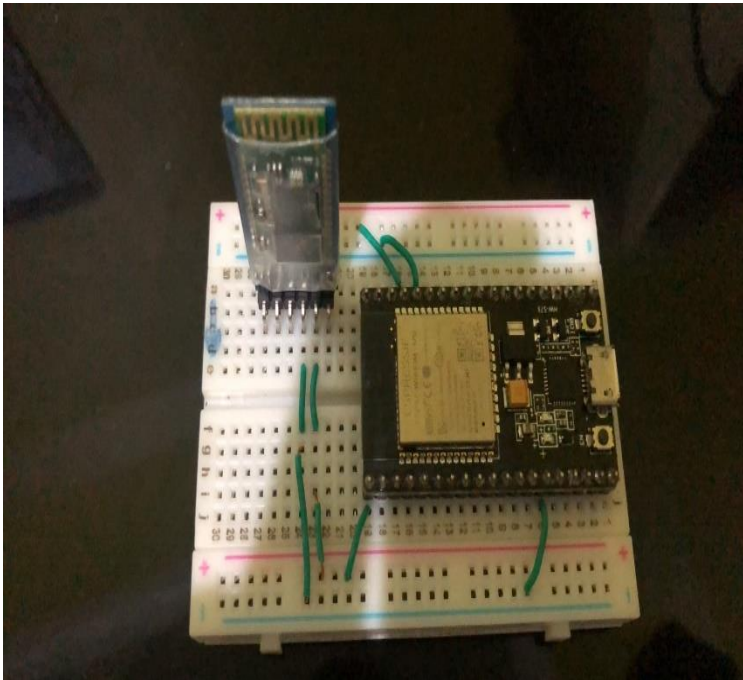
### **C. Modul IoT**

Sistem yang akan dibuat direalisasikan berikutnya adalah modul IoT. Modul IoT yang digunakan adalah NodeMCU ESP-8266 yang merupakan sebuah platform IoT yang banyak digunakan dalam pembuatan perangkat berteknologi IoT. NodeMCU ESP-8266 memiliki modul wifi yang tertanam didalamnya.

Adapun perangkat yang akan dihubungkan pada NodeMCU ESP-8266 adalah modul Bluetooth HC-05 yang berfungsi sebagai penerima data (Rx) dari pengirim (Tx) yang terpasang pada Q2. Untuk pemasangan

ke kaki-kaki yang ada pada NodeMCU ESP-8266 mengikuti datasheet-nya.

Module IoT disini berperan sebagai penghubung ke internet. Jadi data dari Q1 dan Q2 akan dikirimkan ke cloud platform yang digunakan. Pengiriman ini melibatkan jaringan internet, jadi memerlukan jaringan internet untuk dapat mengirimkan data tersebut.



**Gambar 3.7** Realisasi Modul IoT

### **3.3.3 Pembuatan Sistem Perangkat Lunak**

Dalam pembuatan sistem perangkat lunak yang akan digunakan dalam penelitian ini menggunakan bahasa pemrograman C yang dapat dibaca pada *software* Arduino IDE.

Berikut akan dijelaskan proses pembuatan perangkat lunak yang terdiri dari pemrograman sistem hardware (Q1, Q2 dan Divais IoT) dan

pemrograman *Cloud (User Interface)*. Sementara untuk pemrograman cloud terdapat dua alternatif *cloud* yang akan digunakan yaitu *Thingspeak* dan *Blynk*.

### 3.3.3.1 Pemrograman Perangkat Keras (Hardware)

Pada bagian ini akan dibagi menjadi tiga bagian yaitu pemrograman perangkat keras untuk Q1, Q2 dan Divais IoT.

#### A. Pemrograman Q1

Proses pemrograman Q1 ini menggunakan *software* Arduino IDE yang dapat didapat diweb resmi milik Arduino. Program yang ada pada Q1 akan ditampilkan pada Gambar 3.8.



```
Router_BT | Arduino 1.8.12
File Edit Sketch Tools Help
Router_BT
#include <Adafruit_Sensor.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN 2 // pin 0 node mcu
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  dht.begin();
}

void loop() {

  int hum1 = (int)dht.readHumidity();
  int temp1 = (int)dht.readTemperature();

  while(Serial.available() > 0){

    int hum = Serial.parseInt();
    int temp = Serial.parseInt();

    Serial.print(hum);Serial.print(",");Serial.print(temp);Serial.print(",");Serial.print(hum1);Serial.print(",");Serial.println(temp1);

    if(Serial.read() == '\n'){
      Serial.write(hum);Serial.write(",");Serial.write(temp);Serial.write(",");Serial.write(hum1);Serial.write(",");Serial.write(temp1);Serial.write("\n");
    }
  }
}
```

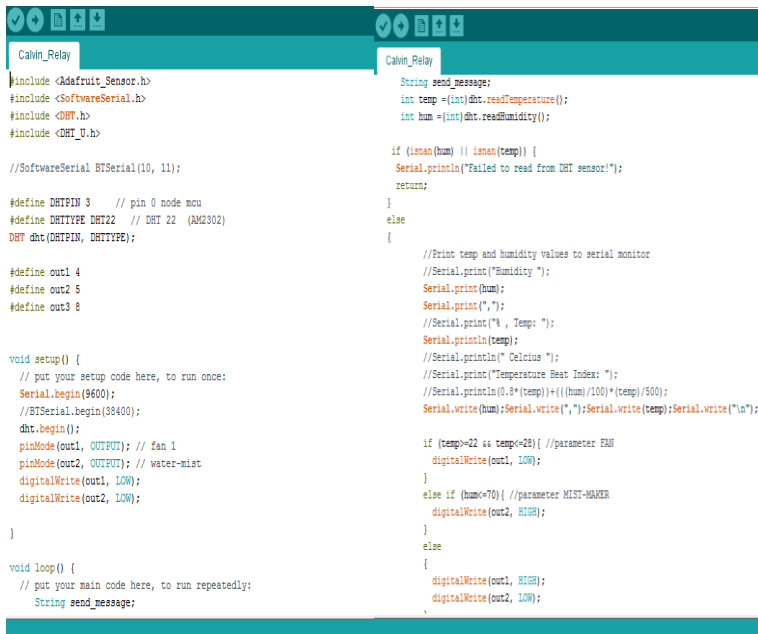
**Gambar 3.8** Pemrograman Sistem Q1

Perlu diketahui bahwa pada pemrograman Q1 ini untuk membaca data dari sensor DHT 22 dan juga untuk mengirimkan data menggunakan modul Bluetooth HC-05 yang sudah dihubungkan ke Arduino Nano.

Adapun beberapa *library* yang harus dimasukkan dalam Arduino IDE agar dapat meng-*compile* program tersebut diantaranya adalah *library* DHT, dan *library software serial*.

## B. Pemrograman Q2

Pada pemrograman Q2 juga tidak jauh berbeda dengan pemrograman Q1. *Library* yang harus diinput sama seperti pemrograman pada Q1. Perbedaan pada Q2 adalah terdapat perintah untuk menerima data dari Q1 dan juga untuk mengirim data ke Divais IoT. Program yang diberikan pada Q2 akan ditampilkan pada gambar 3.9.



```
Calvin_Relay
#include <Adafruit_Sensor.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <DHT_U.h>

//SoftwareSerial BTSerial(10, 11);

#define DHTPIN 3 // pin 0 node mcu
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE);

#define out1 4
#define out2 5
#define out3 8

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  //BTSerial.begin(38400);
  dht.begin();
  pinMode(out1, OUTPUT); // fan 1
  pinMode(out2, OUTPUT); // water-misc
  digitalWrite(out1, LOW);
  digitalWrite(out2, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  String send_message;

  Calvin_Relay
  String send_message;
  int temp = (int)dht.readTemperature();
  int hum = (int)dht.readHumidity();

  if (isnan(hum) || isnan(temp) ) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  else
  {
    //Print temp and humidity values to serial monitor
    //Serial.println("Humidity ");
    Serial.print(hum);
    Serial.print(",");
    //Serial.print("% , Temp: ");
    Serial.println(temp);
    //Serial.println(" Celcius ");
    //Serial.println("Temperature Beat Index: ");
    //Serial.println(0.8*(temp)+((hum)/100)*(temp)/500);
    Serial.write(hum);Serial.write(",");Serial.write(temp);Serial.write("\n");

    if (temp>22 && temp<28){ //parameter FAN
      digitalWrite(out1, LOW);
    }
    else if (hum<70){ //parameter MIST-MAKER
      digitalWrite(out2, HIGH);
    }
    else
    {
      digitalWrite(out1, HIGH);
      digitalWrite(out2, LOW);
    }
  }
}
```

Gambar 3.9 Pemrograman Sistem Q2

Program yang diberikan pada Q2 diberikan juga program untuk mengontrol modul relay untuk menghidupkan aktuator yang terpasang sesuai dengan *threshold* yang diinginkan.

### C. Pemrograman Modul IoT

Pembuatan program divais IoT juga dilakukan pada *software* Arduino IDE. Perlu diketahui bahwa NodeMCU ESP-8266 merupakan divais yang mampu difungsikan sebagai mikrokontroler dan kompatibel dengan *software* Arduino IDE.

Pada divais IoT diberikan program untuk menghubungkan perangkat ke jaringan internet, diberikan pula program untuk menerima data dari modul Bluetooth HC-05. Selain itu pada divais IoT juga diberikan perintah untuk mengirimkan data ke *cloud platform* (*Thingspeak* dan *Blynk*). Program dari divais IoT akan ditampilkan pada Gambar 3.10.



```
NodeMCU_Blynk
#define BLYNK_FIRMWARE Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "ThingSpeak.h"

char auth[] = "d07Fu_gbw@jC-HR1j6t2Pak5UBRms5"; // You should get Auth Token in the Blynk App.
// Go to the Project Settings (near icon).
char ssid[] = "B4EAE1"; // Your WiFi credentials.
char pass[] = "SuperKapus"; // Set password to "" for open networks.

WiFiClient client;

void setup()
{
  Serial.begin(9600);
  Serial.println("Connecting to ");Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println("Macaddr: ");Serial.println(WiFi.macAddress());
  Serial.println("Gateway: ");Serial.println(WiFi.gatewayIP());
  Blynk.begin(auth, ssid, pass);
  ThingSpeak.begin(client);
}

NodeMCU_ThingSpeak
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.println("");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
Serial.println("Macaddr: ");Serial.println(WiFi.macAddress());
Serial.println("Gateway: ");Serial.println(WiFi.gatewayIP());
Blynk.begin(auth, ssid, pass);
ThingSpeak.begin(client);

void loop()
{
  Blynk.run();
  while(Serial.available() > 0){
    int bin = Serial.parseInt();
    int temp = Serial.parseInt();
    int humi = Serial.parseInt();
    int temp2 = Serial.parseInt();
    Serial.println(hum;Serial.println(temp;Serial.println(humi;Serial.println(temp2);
    Blynk.virtualWrite(V0, temp);
    Blynk.virtualWrite(V1, humi);
    Blynk.virtualWrite(V2, temp2);
    Blynk.virtualWrite(V3, humi);
  }
}
```

Gambar 3. 10 Pemrograman Sistem Modul IoT

Adapun beberapa *library* yang harus dimasukkan kedalam Arduino IDE seperti *library* ESP8266, *library* Blynk dan *library* Thingspeak.

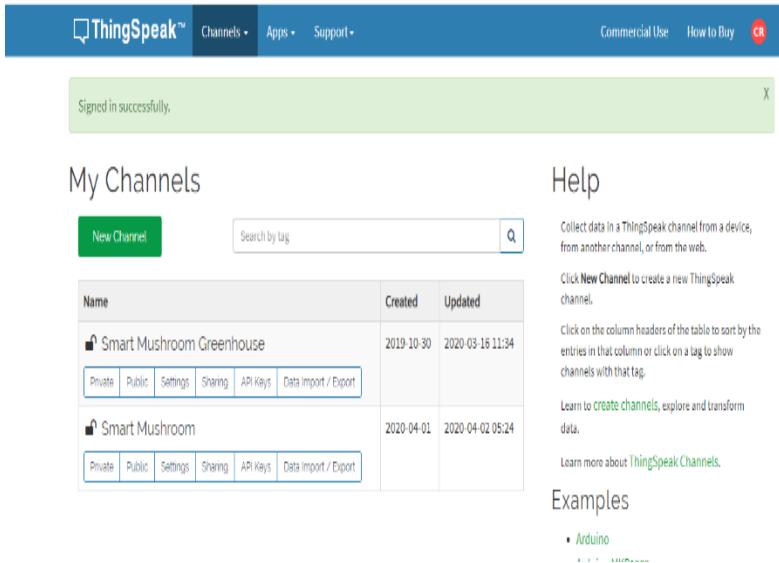
#### **D. Pemrograman Cloud (User Interface)**

Pada penelitian ini terdapat dua kandidat cloud yang digunakan yaitu Thingspeak dan Blynk. Adapun langkah-langkah untuk memprogram *cloud (user interface)* sebagai berikut :

##### **a. Thingspeak**

Thingspeak merupakan platform open source Internet of Things (IOT) aplikasi dan API untuk menyimpan dan mengambil data dari hal menggunakan protokol HTTP melalui Internet atau melalui Local Area Network. ThingSpeak memungkinkan pembuatan aplikasi sensor logging, aplikasi lokasi pelacakan, dan jaringan sosial hal dengan update status. ThingSpeak awalnya diluncurkan oleh ioBridge pada tahun 2010 sebagai layanan untuk mendukung aplikasi IOT. ThingSpeak telah terintegrasi dukungan dari numerik komputasi perangkat lunak MATLAB dari MathWorks. Memungkinkan pengguna ThingSpeak untuk menganalisis dan memvisualisasikan data yang diunggah menggunakan Matlab tanpa memerlukan pembelian lisensi Matlab dari MathWorks. Berikut adalah cara penggunaan Thingspeak :

- Masuk kedalam web resmi dari Thingspeak (Thingspeak.com)
- Membuat ID (proses ini dapat dilakukan didalam web).
- Membuka menu *account*, untuk mendapatkan kode API dari ID yang sudah dibuat.
- Memasukkan kode API tersebut kedalam program yang ada pada divais IoT.
- Membuat *project* sebagai penampil data.



**Gambar 3.11** Tampilan Awal Thingspeak

Pada penelitian ini, thingspeak memiliki keterbatasan pada jumlah user yang terbatas yaitu hanya dapat menampilkan delapan *field display*. Sehingga pada penelitian ini Thingspeak tidak digunakan sebagai cloud.

## b. Blynk

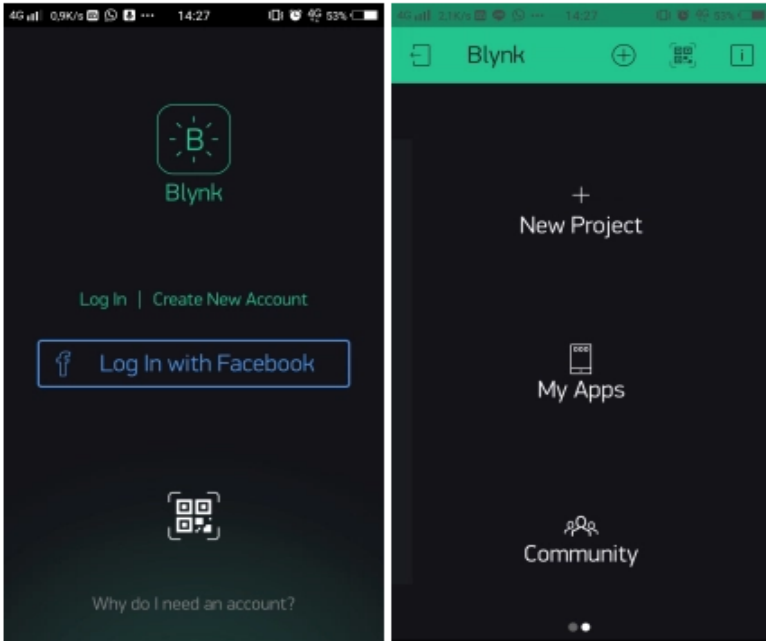
Blynk adalah platform untuk aplikasi OS Mobile (iOS dan Android) yang bertujuan untuk kendali module Arduino, Raspberry Pi, ESP8266, WEMOS D1, dan module sejenisnya melalui Internet, dengan catatan terhubung dengan internet dengan koneksi yang stabil dan inilah yang dinamakan dengan sistem Internet of Things (IOT).

Berikut adalah cara untuk menggunakan Blynk sebagai cloud :

- Mengunduh aplikasi Blynk melalui Playstore (Android) dan Appstore (iOS).
- Membuat ID yang dapat dilakukan didalam aplikasi.
- Mengajukan pengiriman kode API ke email yang telah didaftarkan. Proses ini juga dapat dilakukan didalam aplikasi.



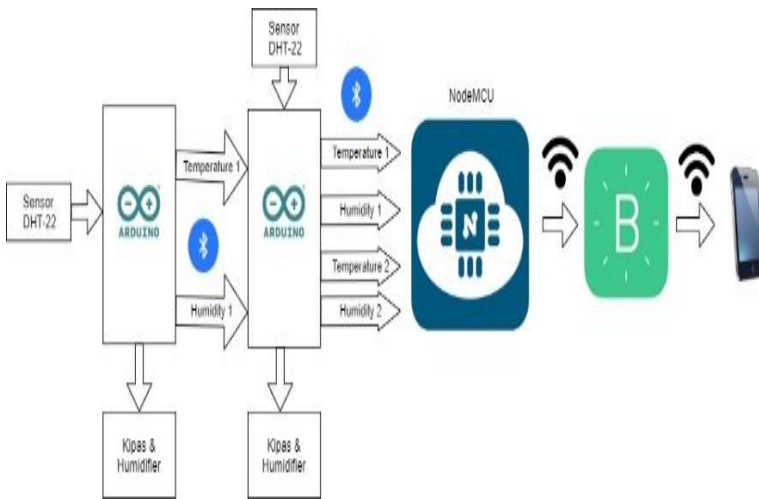
- Menginput kode API yang sudah diterima ke dalam program yang ada pada divais IoT.
- Memilih beberapa tampilan gadget sesuai kebutuhan untuk penampil data.



**Gambar 3.12** Tampilan Awal Blynk

### 3.3.4 Integrasi Sistem Hardware dan Software

Dalam proses ini akan dilakukan integrasi antara hardware dan software yang sudah dibuat. Proses ini mengintegrasikan keseluruhan sistem baik dari Q1, Q2, Divais IoT dan *Cloud (User Interface)*. Dalam Gambar 3.11 akan ditampilkan diagram integrasi keseluruhan sistem pada Qumbung-Q yang sudah dibuat.



**Gambar 3.11** Integrasi Sistem Qumbang-Q

### 3.3.4.1 Cara Kerja Sistem Qumbang-Q

Proses integrasi dilakukan agar seluruh bagian dari Qumbang-Q dapat bekerja sesuai dengan tujuan dalam penelitian ini.

Pada bagian pertama yaitu mikrokontroler 1 (Arduino Nano) yang selanjutnya akan disebut Q1. Terdapat sensor DHT 22 yang akan memberikan data *temperature 1* dan *humidity 1*, selanjutnya data itu diproses oleh Arduino Nano dan akan memberikan perintah nyala atau mati kepada aktuator jika data dari sensor yang didapat melebihi *threshold* yang sudah dimasukkan pada program Q1. Selanjutnya data suhu dan kelembaban yang sudah didapat oleh Q1 akan dikirimkan ke mikrokontroler 2 yang selanjutnya akan disebut Q2 menggunakan komunikasi *wireless* yang dapat dilakukan oleh modul Bluetooth.

Selanjutnya, pada Q2 data yang dikirim oleh Q1 akan diterima menggunakan modul Bluetooth yang ada pada Q2. Perlu diketahui bahwa Q2 juga akan mendapatkan data suhu dan kelembaban dari sensor yang terhubung pada Q2 itu sendiri. Lalu data dari sensor tersebut diproses oleh Q2 untuk selanjutnya memberikan perintah nyala atau mati untuk aktuator

jika data yang didapat melebihi *threshold*. Lalu data yang didapat dari Q1 akan dikirimkan bersamaan dengan data yang didapat dari Q2 menggunakan komunikasi *wireless* yang dapat dilakukan menggunakan modul Bluetooth ke NodeMCU ESP-8266 yang selanjutnya akan disebut divais IoT.

Data yang telah dikirimkan oleh Q2 akan diterima oleh divais IoT melalui modul Bluetooth yang sudah terpasang di divais IoT. Perangkat ini memiliki peran untuk menghubungkan ke internet sekaligus mengirimkan data yang sudah didapat ke *Blynk*. Setelah divais IoT ini terhubung ke internet, maka perangkat ini secara otomatis akan mengirimkan data ke *Blynk* dan dapat langsung dilihat di *user interface* yang ada pada smartphone.

### 3.3.4.2 Pengujian Fungsional Sistem

Pada bagian ini akan dilakukan pengujian fungsional sistem terhadap beberapa hal yang dianggap perlu dilakukan terhadap komponen / divais yang digunakan dan akan ditampilkan pada tabel 3.4 sebagai berikut:

**Tabel 3.4** Jenis Pengujian Fungsional Sistem

No	Objek Uji	Jenis Pengujian
1	Mikrokontroler Q1	A.Pembacaan data sensor
		B.Kemampuan memberi perintah pada relay
		C.Terhubung pada modul Bluetooth
		D.Kemampuan mengirim data ke Arduino 2
2	Mikrokontroler	A.Pembacaan data sensor

	Q2	B.Kemampuan perintah pada relay
		Terhubung pada modul Bluetooth
		C.Kemampuan menerima data dari Arduino 1
		D.Kemampuan mengirim data ke NodeMCU ESP-8266
3	Internet Gateway	A.Kemampuan menerima data dari arduino 2
		B.Kemampuan menangkap jaringan internet
		C.Mampu terhubung ke jaringan internet
4	Sensor suhu	A.Kemampuan memberikan data suhu
5	Sensor Kelembaban	A.Kemampuan memberikan data kelembaban
6	Aktuator	A.Kemampuan berfungsi saat relay on
		B.Kemampuan berfungsi saat relay off

7	Aktuator	A.Kemampuan berfungsi saat relay on
		B.Kemampuan berfungsi saat relay off

### 3.3.4.3 Pengujian User Interface

Pengujian user interface dari platform yang digunakan perlu dilakukan agar dapat mengetahui kinerja dari platform yang digunakan. Selain itu pengujian ini dilakukan untuk mengetahui berapa kapasitas user maksimum untuk masing-masing platform. Pengujian ini menguji dua user interface sekaligus yaitu Thingspeak dan Blynk.

Proses ini dilakukan dengan cara memberikan program ke mikrokontroler yang digunakan untuk mengirimkan data ke user interface yang digunakan, setelah itu hasilnya akan dilihat pada user interface apakah memberikan data real time dengan benar atau tidak.

#### A. Thingspeak

Berikut akan ditampilkan langkah-langkah untuk dapat menghubungkan perangkat yang telah dibuat ke user interface (Thingspeak).

1. Mendaftar/membuat akun Thingspeak di website resmi Thingspeak (Thingspeak.com).
2. Mendapatkan API Keys dari akun yang telah dibuat dengan cara masuk ke menu setting > account.
3. Membuat *field project* sesuai dengan kebutuhan. (Maksimal 8 Field).
4. Memasukkan API Keys ke program perintah pada mikrokontroler.
5. Memastikan bahwa mikrokontroler sudah terhubung ke internet.
6. Data sudah dapat terlihat di field project yang sudah kita buat.

## B. Blynk

Langkah-langkah untuk menggunakan Blynk adalah sebagai berikut:

1. Mendownload software Blynk pada smartphone.
2. Membuat/Mendaftar akun Blynk.
3. Membuat field project sesuai kebutuhan (Tidak ada batas, selama kita membeli akses premium).
4. Meminta pengiriman kode akun ke email sesuai yang terdaftar.
5. Memasukkan kode yang sudah kita dapat ke program perintah pada mikrokontroler yang kita gunakan.
6. Memastikan bahwa perangkat terhubung ke jaringan internet.
7. Data dari mikrokontroler sudah dapat muncul ke user interface.

### 3.4 Pengukuran QoS

Pengukuran QoS dilakukan dengan tujuan untuk mengetahui kinerja jaringan dari Qumbang-Q. Proses *capture* jaringan akan dilakukan menggunakan *software* PCAP Remote yang dapat diunduh pada smartphone melalui *Playstore*, dan untuk mengukur akan dilakukan pada *software* Wireshark.

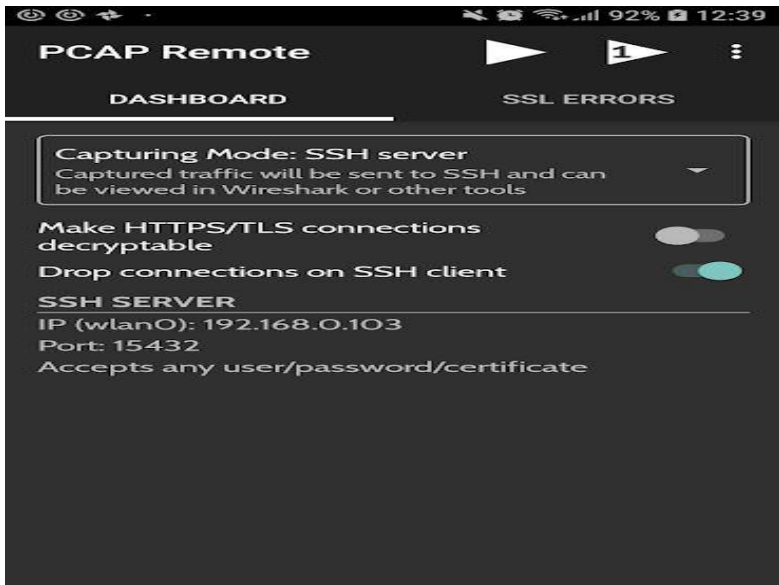
#### 3.4.1 Software Tool

Adapun beberapa *software* yang digunakan dalam pengukuran ini adalah PCAP Remote dan *Wireshark*.

##### 1. PCAP Remote

Langkah pertama dalam pengukuran QoS adalah capture jaringan. Untuk mengcapture jaringan dalam penelitian ini menggunakan aplikasi yang ada di smartphone yaitu PCAP Remote. Alasan menggunakan aplikasi ini adalah karena saat dilakukan proses capture jaringan dalam wireshark, tidak dapat memunculkan IP yang dituju. Oleh karena itu digunakan *software* PCAP Remote ini.

Pada Gambar 3.12 akan ditunjukkan tampilan awal dari software PCAP remote yang digunakan dalam penelitian ini.

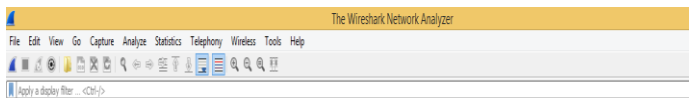


Gambar 3.13 Tampilan Awal *Software* PCAP Remote

## 2. Wireshark

Pada penelitian ini wireshark digunakan untuk mendapatkan nilai parameter yang dicari. Wireshark hanya menampilkan beberapa parameter dalam menu statistics nya dan untuk mendapatkan nilai parameter yang tidak tertampil pada wireshark maka dilakukan proses convert file menjadi .csv agar dapat dilakukan analisa di aplikasi lain.

Tampilan awal dari *wireshark* akan ditampilkan pada Gambar 3.13 yang akan ditampilkan dibawah ini.



Welcome to Wireshark

Open

- D:\MATRIULTAI\pcap wireshark\file1meter-3.pcap (70 KB)
- D:\MATRIULTAI\pcap wireshark\file2-30.pcap (188 KB)
- D:\MATRIULTAI\pcap wireshark\filemeter-4.pcap (36 KB)
- D:\MATRIULTAI\pcap wireshark\file4-90.pcap (41 KB)
- C:\Users\Calvin Rafindral\Downloads\file4-60.pcap (84 KB)
- C:\Users\Calvin Rafindral\Downloads\2020042904023.pcap (61 KB)
- D:\MATRIULTAI\pcap wireshark\file4-30.pcap (61 KB)
- D:\MATRIULTAI\pcap wireshark\file2-60.pcap (62 KB)
- D:\MATRIULTAI\pcap wireshark\file2-90.pcap (64 KB)
- C:\Users\Calvin Rafindral\Downloads\file1meter.pcap (28 KB)

### **Gambar 3.14** Tampilan Awal *Software Wireshark*

#### **3.4.2 Parameter Yang Diukur**

Untuk mengetahui kinerja jaringan diperlukan beberapa parameter untuk menentukan bagaimana kinerja jaringan tersebut. Adapun beberapa parameter yang diukur pada pengukuran ini adalah sebagai berikut :

- Delay
- Jitter
- Throughput
- Packetloss

#### **3.4.3 Skenario Pengujian**

Dalam skenario pengujian kali ini akan dilakukan beberapa pengujian dengan memberi variasi terhadap kinerja jaringan diantaranya adalah :

##### **3.4.3.1 Pengaruh Jumlah Kumbang**

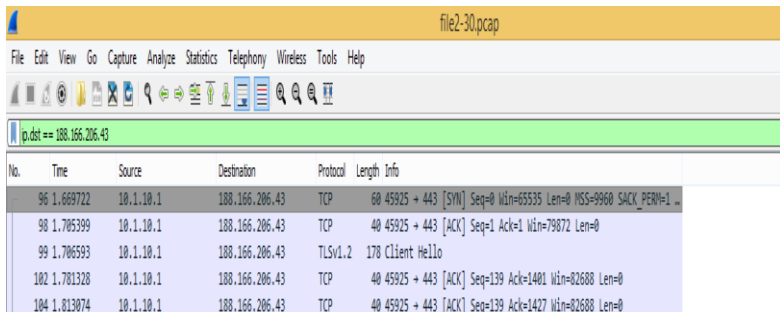
Pengujian ini melihat pengaruh jumlah kumbang terhadap kinerja jaringan. Pengujian ini dilakukan dengan cara mengaktifkan salah satu sistem yang ada pada kumbang dan mengaktifkan dua sistem sekaligus sehingga dapat diketahui pengaruh input data yang diberikan terhadap kinerja jaringan.

Proses pertama adalah proses *capture* jaringan. Proses ini menggunakan *software* PCAP Remote yang ada pada smartphone.



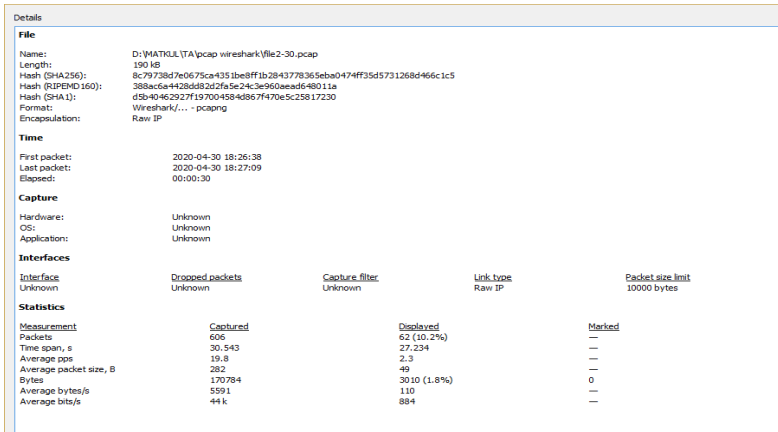
Setelah mendapatkan hasil *capture* selanjutnya file hasil *capture* akan dibuka di *software Wireshark*.

Pada *wireshark*, hasil *capture* tersebut akan di-*filter* sesuai IP yang akan dituju agar hanya menampilkan komunikasi yang menuju ke *Blynk* dengan cara memasukkan perintah “ip.dst==188.166.206.43”. Setelah dilakukan proses penyaringan barulah data dapat dianalisa melalui *Wireshark*.



**Gambar 3.15** Proses *Filter* Pada *Wireshark*

Proses analisa dapat dilakukan dengan cara masuk ke menu “*statistics*” dan masuk ke pilihan “*capture file properties*”. Setelah itu akan muncul tampilan seperti pada Gambar 3.14. Adapun beberapa parameter yang tertampil adalah jumlah paket, waktu pengujian, rata-rata paket dalam satuan bytes/s, dan juga rata-rata paket dalam satuan bits/s.



**Gambar 3.16** Tampilan Pada Menu *Capture File Properties*

### 3.4.3.2 Pengaruh Lama Pengamatan

Pada pengujian selanjutnya adalah pengujian untuk melihat pengaruh lama pengamatan terhadap kinerja jaringan. Pengujian ini dilakukan dengan melihat kinerja jaringan pada Qumbang-Q selama 30 detik, 60 detik dan 90 detik.

Langkah pertama pada pengujian ini adalah *capture* jaringan. Proses capture jaringan ini dilakukan tiga kali selama 30 detik, 60 detik dan 90 detik menggunakan *software* PCAP Remote.

Selanjutnya file hasil *capture* akan dibuka di *wireshark*. Pada *wireshark* juga akan dilakukan proses analisa. Untuk mendapatkan parameter yang tidak tertampil pada tampilan *wireshark* maka file harus di *export* agar dapat dilakukan perhitungan pada Ms. Excel.

### 3.4.3.3 Pengaruh Jarak Lokasi Kumbang

Tujuan dari pengujian ini adalah melihat pengaruh jarak kumbang dari *access point* terhadap kinerja jaringan. Pengujian ini dilakukan dengan cara memindahkan jarak Qumbang-Q sejauh 1 m , 3 m dan 5 m.

Langkah pertama adalah proses *capture* jaringan. Proses ini dilakukan tiga kali, yaitu pada saat jarak 1 m, 3 m dan 5 m. *Capture* jaringan dilakukan dengan *software* PCAP Remote.

### **3.5 Perhitungan Parameter Hasil Pengukuran**

Adapun parameter lain yang tidak tertampil pada menu statistics pada wireshark sehingga diperlukan cara perhitungan untuk mendapatkan nilai parameter tersebut. Parameter yang tidak tertampil tersebut adalah delay dan jitter. Setelah mendapatkan hasil *capture*, maka file tersebut akan dibuka di *wireshark*. Dan akan dianalisa pada *wireshark*, lalu untuk mendapat beberapa parameter lainnya yang tidak tertampil pada *wireshark* maka file harus *diconvert* dengan format .xls agar dapat dibuka di Ms. Excel.

Pada Ms. Excel barulah dapat dilakukan perhitungan beberapa parameter lain yang dicari.

#### **3.5.1 Delay**

Perhitungan parameter delay dilakukan untuk mengetahui pengaruh pengujian yang sudah dilakukan apakah dapat berpengaruh terhadap lama pengiriman data tersebut. Selain itu delay juga dianggap penting untuk mengetahui kinerja jaringan yang ada sehingga dapat diketahui bahwa kinerja jaringannya baik atau buruk.

#### **3.5.2 Jitter**

Selain delay, ada juga parameter lain yang tidak tertampil pada menu statistics yang ada pada wireshark yaitu jitter. Jitter merupakan variasi dari delay yang ada. Parameter ini juga diperlukan untuk menentukan bagaimana kinerja jaringan pada sistem yang diuji.

### **3.6 Standar Klasifikasi QoS**

Untuk menentukan kategori sebuah kinerja jaringan diperlukan sebuah standar untuk klasifikasinya. Dalam penelitian ini standar klasifikasi yang digunakan TIPHON. Standar ETSI TIPHON (Telecommunication and Internet Protocol Harmonization over Network) memiliki beberapa parameter QoS yang terdiri atas delay, jitter, throughput dan packetloss.

### 3.6.1 Delay

Berikut akan ditampilkan standar TIPHON untuk menentukan kategori pada parameter delay.

**Tabel 3.5** Klasifikasi Delay TIPHON

<b>Kriteria</b>	<b>Nilai Delay (ms)</b>	<b>Skor</b>
Sangat Baik	<150	4
Baik	150 - 300	3
Sedang	300 - 450	2
Jelek	>450	1

Tabel tersebut dapat digunakan untuk menentukan kategori delay yang terjadi pada jaringan.

### 3.6.2 Jitter

Pada tabel 3.4 akan ditampilkan standar TIPHON untuk parameter jitter.

**Tabel 3.6** Klasifikasi Jitter TIPHON

<b>Kriteria</b>	<b>Peak Jitter (ms)</b>	<b>Skor</b>
Sangat Baik	0	4
Baik	0 - 75	3
Sedang	75 - 125	2
Jelek	125 - 225	1

Tabel tersebut dapat digunakan untuk menentukan kategori kinerja jaringan pada parameter jitter.

### 3.6.3 Throughput

Parameter throughput juga akan menjadi tolak ukur kinerja jaringan. Pada tabel 3.5 akan ditampilkan standar TIPHON untuk parameter throughput

**Tabel 3.7** Klasifikasi Throughput TIPHON

<b>Kriteria</b>	<b>Throughput (%)</b>	<b>Skor</b>
Sangat Baik	100	4
Baik	75	3
Sedang	50	2
Jelek	<25	1

### 3.6.4 Packet loss

Merupakan suatu parameter yang menggambar suatu kondisi yang menunjukkan jumlah total paket yang hilang, hal ini dapat terjadi karena collision dan congestion pada jaringan. Pada tabel 3.6 menunjukkan peringkat dan kriteria packet loss.

**Tabel 3.8** Klasifikasi Packet Loss TIPHON

<b>Kriteria</b>	<b>Packet Loss (%)</b>	<b>Skor</b>
Sangat Bagus	0	4
Bagus	3	3
Kurang Bagus	15	2
Jelek	25	1

## BAB 4

### HASIL DAN ANALISA

#### 4.1 Hasil Pembuatan Kumbung Jamur

Hasil pembuatan kumbung jamur tampak luar dan dalam akan ditampilkan pada gambar 4.1.



**Gambar 4.1** Tampilan Berbagai Sisi Kumbung

Bentuk kumbung jamur yang dibuat merupakan rekomendasi dari petani jamur. Beberapa sisi dari kumbung jamur ini dibuat berlubang dan ditutup paranet karena mempertimbangkan kebutuhan intensitas cahaya untuk pertumbuhan jamur tiram.

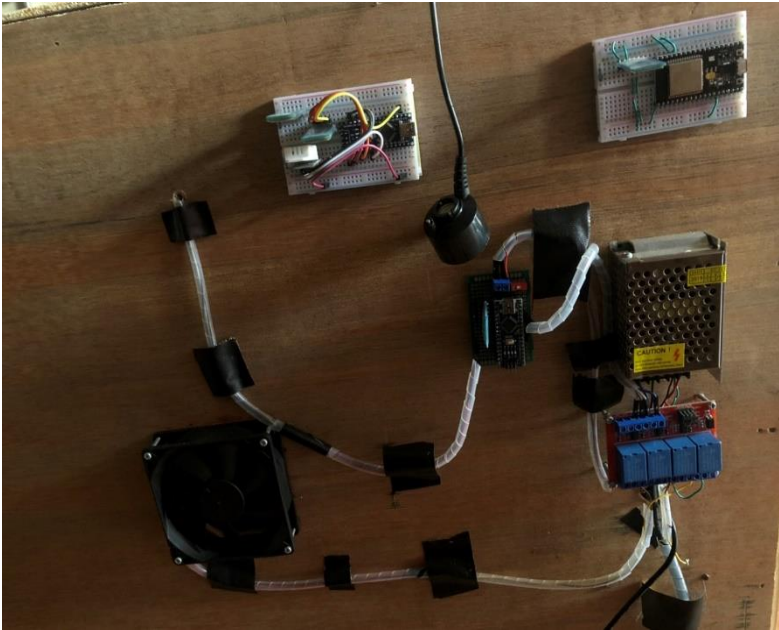
## 4.2 Hasil Implementasi Sistem Hardware

Keseluruhan sistem hardware yang sebelumnya sudah dirancang akan pada proses ini akan direalisasikan dan ditampilkan hasilnya. Pada Gambar 4.2 akan ditampilkan hasil realisasi secara keseluruhan dari sistem *hardware* yang sudah dibuat.

Setelah semua hardware tersebut direalisasikan, maka proses berikutnya adalah proses pengimplementasian hardware kedalam kumbang yang sudah dibuat. Segala perangkat dipasang pada bagian belakang pada kumbang dengan tujuan untuk mempermudah proses pemeliharaan. Pada Gambar 4.2 akan ditampilkan hasil implementasi hardware ke kumbang.



**Gambar 4.2** Hasil Implementasi Sistem Q1



**Gambar 4.3** Sistem Hardware Yang Dibuat

Pada gambar 4.3 tertampil keseluruhan hardware yang sudah dibuat (Q1, Q2 dan modul IoT). Modul IoT dipasang dekat dengan Q1 dengan tujuan untuk mempercepat pengiriman data.

#### **4.3 Hasil Tampilan User Interface.**

Dalam penelitian ini, cloud yang digunakan adalah Blynk. Seperti yang sudah dibahas pada bab 3. Berikut akan ditampilkan hasil tampilan setelah seluruh sistem terintegrasi yang akan ditampilkan pada Gambar 4.4.



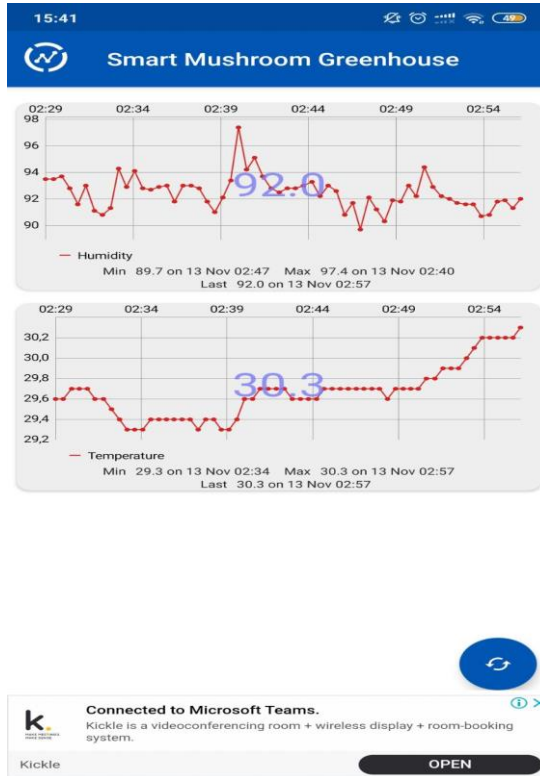


**Gambar 4.4** Tampilan User Interface

Pada gambar tersebut tertampil data temperature (1 dan 2) dan humidity (1 dan 2) dari Qumbang-Q. Temperature 1 dan humidity 1 adalah hasil monitoring yang ada pada kumbung 1, dan temperature dan humidity 2 adalah hasil dari monitoring pada kumbung 2. Dari temperature yang didapat yaitu 27°C pada kedua kumbung, ini artinya adalah suhu yang didapat masih pada batas sehingga tidak menyalakan aktuator yang akan menyala apabila suhu >28°C. Untuk parameter humidity yang didapat adalah 65% yang artinya masih pada batas untuk tidak menyalakan aktuator.

Selanjutnya akan ditampilkan hasil grafik suhu dan kelembaban yang akan ditampilkan pada gambar 4.5. Field 1 adalah grafik kelembaban dan field 2 adalah grafik untuk suhu.

Berikutnya pada Gambar 4.5 akan ditampilkan hasil tampilan pada user interface Thingspeak.



**Gambar 4.5** Tampilan User Interface Thingspeak

Pada user interface thingspeak memiliki beberapa keunggulan yaitu dapat melihat grafik secara langsung. Namun pada user interface thingspeak terbatas pada jumlah user yang terhubung.

#### 4.4 Hasil Pengujian Sistem

Hasil pengujian sistem meliputi hasil pengujian fungsional sistem dan juga hasil pengujian pengukuran QoS.

##### 4.4.1 Hasil Pengujian Fungsional

Pengujian ini dilakukan dengan cara mengikuti prosedur jenis pengujian sesuai yang sudah ditentukan. Tujuan dari dilakukannya pengujian fungsional adalah untuk mengetahui bahwa keseluruhan divais

yang digunakan dapat bekerja dengan baik. Hasil dari pengujian fungsional ini akan ditunjukkan pada tabel 4.1.

**Tabel 4.1** Hasil Pengujian Fungsional

<b>No</b>	<b>Objek Uji</b>	<b>Jenis Pengujian</b>	<b>Hasil</b>
1	Mikrokontroler Q1	A.Pembacaan data sensor	OK
		B.Kemampuan memberi perintah pada relay	OK
		C.Terhubung pada modul Bluetooth	OK
		D.Kemampuan mengirim data ke Arduino 2	OK
2	Mikrokontroler Q2	A.Pembacaan data sensor	OK
		B.Kemampuan perintah pada relay	OK
		Terhubung pada modul Bluetooth	OK
		C.Kemampuan menerima data dari Arduino 1	OK
		D.Kemampuan mengirim data ke NodeMCU ESP-8266	OK

3	Internet Gateway	A.Kemampuan menerima data dari arduino 2	OK
		B.Kemampuan menangkap jaringan internet	OK
		C.Mampu terhubung ke jaringan internet	OK
4	Sensor suhu	A.Kemampuan memberikan data suhu	OK
5	Sensor Kelembaban	A.Kemampuan memberikan data kelembaban	OK
6	Aktuator	A.Kemampuan berfungsi saat relay on	OK
		B.Kemampuan berfungsi saat relay off	OK
7	Aktuator	A.Kemampuan berfungsi saat relay on	OK
		B.Kemampuan berfungsi saat relay off	OK

Dari hasil pengujian tersebut dapat diketahui bahwa seluruh divais yang digunakan mampu bekerja dengan baik.

#### 4.4.2 Hasil Pengujian Pengukuran QoS

Hasil pengukuran QoS dari masing-masing pengujian akan dibahas dan ditampilkan pada bagian ini.

#### 4.4.2.1 Hasil Pengujian Pengaruh Jumlah Kumbung

Dari hasil pengujian ini, akan ditampilkan hasil pengukuran menggunakan aplikasi *wireshark* dan hasil perhitungan beberapa parameter lain yang tidak ada pada *wireshark*. Gambar 4.4 merupakan hasil analisa dengan *wireshark* pada pengujian dengan jumlah satu kumbung. Gambar 4.5 merupakan hasil analisa dengan *wireshark* pada pengujian dua kumbung.

##### Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>	<u>Marked</u>
Packets	411	113 (27.5%)	—
Time span, s	56.055	52.729	—
Average pps	7.3	2.1	—
Average packet size, B	124	45	—
Bytes	50795	5086 (10.0%)	0
Average bytes/s	906	96	—
Average bits/s	7249	771	—

**Gambar 4.5** Hasil Analisa Pada *Wireshark* Pada Pengujian Dengan Satu Kumbung

## Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>	<u>Marked</u>
Packets	191	43 (22.5%)	—
Time span, s	57.719	56.424	—
Average pps	3.3	0.8	—
Average packet size, B	105	53	—
Bytes	19993	2286 (11.4%)	0
Average bytes/s	346	40	—
Average bits/s	2771	324	—

**Gambar 4.6** Hasil Analisa Pada *Wireshark* Pada Pengujian Dengan Dua Kumbang

### 4.4.2.2 Hasil Pengujian Pengaruh Lama Pengamatan

Disini akan ditampilkan dan dibahas hasil dari pengukuran QoS pada pengujian pengaruh lama pengamatan. Hasil dari analisa di *Wireshark* akan ditampilkan pada Gambar 4.6, 4.7 dan 4.8.

## Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>	<u>Marked</u>
Packets	399	93 (23.3%)	—
Time span, s	33.634	31.418	—
Average pps	11.9	3.0	—
Average packet size, B	125	47	—
Bytes	49775	4332 (8.7%)	0
Average bytes/s	1479	137	—
Average bits/s	11 k	1103	—

**Gambar 4.7** Hasil Analisa Pada Wireshark Dengan Waktu 30 s

**Statistics**

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>	<u>Marked</u>
Packets	191	43 (22.5%)	—

**Statistics**

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>	<u>Marked</u>
Packets	257	54 (21.0%)	—
Time span, s	90.650	86.999	—
Average pps	2.8	0.6	—
Average packet size, B	133	51	—
Bytes	34233	2762 (8.1%)	0
Average bytes/s	377	31	—
Average bits/s	3021	253	—

**Gambar 4.9** Hasil Analisa Pada Wisehark Dengan Waktu 90 s

#### 4.4.2.3 Hasil Pengujian Pengaruh Jarak Lokasi Kumbang

Disini akan ditampilkan dan dibahas hasil pengukuran QoS yang sudah dilakukan. Hasil analisa dari *wireshark* akan ditampilkan pada gambar 4.9 dan 4.10.

**Statistics**

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>	<u>Marked</u>
Packets	327	36 (11.0%)	—
Time span, s	44.713	43.604	—
Average pps	7.3	0.8	—
Average packet size, B	215	56	—
Bytes	70151	2006 (2.9%)	0
Average bytes/s	1568	46	—
Average bits/s	12 k	368	—

**Gambar 4.10** Hasil Analisa Pada Wireshark Dengan Jarak 1 m

**Statistics**

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>	<u>Marked</u>
Packets	229	45 (19.7%)	—
Time span, s	69.365	68.344	—
Average pps	3.3	0.7	—
Average packet size, B	130	53	—
Bytes	29701	2402 (8.1%)	0
Average bytes/s	428	35	—
Average bits/s	3425	281	—

**Gambar 4.11** Hasil Analisa Pada Wireshark Dengan Jarak 3 m

Perlu diketahui bahwa pada pengujian dengan jarak 5 m, tidak ada komunikasi pada IP yang dituju. Sehingga *wireshark* tidak dapat menampilkan statisticsnya.

Selanjutnya adalah proses perhitungan untuk mendapat parameter QoS lain yang tidak tertampil pada *wireshark*.

**4.4.3 Hasil Perhitungan QoS**

Perhitungan dari hasil pengujian dilakukan karena beberapa parameter tidak tertampil pada *Wireshark* sehingga perlu dilakukan perhitungan. Proses perhitungan ini dilakukan pada Ms. Excel.

**4.4.3.1 Hasil Perhitungan QoS Pengaruh Jumlah Kumbung**

Berikut adalah hasil perhitungan QoS pada pengujian pengaruh jumlah kumbung yang akan ditampilkan pada tabel 4.2.

**Tabel 4.2** Hasil Perhitungan QoS Pengaruh Jumlah Kumbung

<b>Jumlah Kumbung</b>	<b>Throughput (bits/s)</b>	<b>Paketloss</b>	<b>Rata-Rata Delay (ms)</b>	<b>Rata-Rata Jitter(ms)</b>
1	771	0	470	0.5
2	324	0	1340	0.5

Dari hasil perhitungan didapatkan bahwa nilai throughput pada pengujian dengan jumlah 1 kumbung yaitu 771 bit/s dan tidak ada packet



loss sehingga nilai throughputnya 100%. Pada pengujian dengan jumlah 2 kumbang didapatkan nilainya adalah 324 bit/s dan tidak ada packet loss sehingga nilainya 100%.

Nilai rata-rata delay pada pengujian dengan jumlah 1 kumbang adalah 0.47 s. Pada pengujian dengan jumlah 2 kumbang mendapatkan nilai rata-rata delay 1.34 s.

Rata-rata jitter pada pengujian dengan jumlah 1 kumbang adalah 0.0005 s dan pada pengujian dengan jumlah 2 kumbang adalah 0.0005 s.

#### 4.4.3.2 Hasil Perhitungan QoS Pengaruh Lama Pengukuran

Hasil dari perhitungan QoS pada pengujian pengaruh lama pengukuran akan ditampilkan pada tabel 4.3.

**Tabel 4.3** Hasil Perhitungan QoS Pengaruh Lama Pengukuran

<b>Waktu (s)</b>	<b>Throughput (bits/s)</b>	<b>Paketloss</b>	<b>Rata-Rata Delay (ms)</b>	<b>Rata-rata Jitter (ms)</b>
30	880	0	440	0.65
60	771.6	0	470	0.5
90	690.8	0	470	0.59

Sebagaimana yang sudah tertampil pada tabel 4.3 diketahui bahwa nilai throughput pada pengujian lama pengukuran dengan waktu 30 s adalah 880 bit/s. Lalu pada pengujian dengan waktu 60 s didapatkan nilai throughputnya adalah 771.6 bit/s dan pada pengujian dengan waktu 90 s didapatkan nilai throughput 690.8 bit/s. Pada pengujian ini tidak ada packet loss sehingga nilai throughputnya 100%.

Pada pengujian dengan waktu 30 s didapatkan nilai rata-rata delay sebesar 0.44 s. Selanjutnya pada pengujian dengan waktu 60s didapatkan rata-rata delay sebesar 0.47 s dan pada pengujian dengan waktu 90 s didapatkan nilai rata-rata delay sebesar 0.47 s.

Selanjutnya untuk rata-rata jitter pada pengujian dengan waktu 30 s adalah 0.00065 s, pada pengujian dengan waktu 60 s didapatkan rata-rata

jitter sebesar 0.0005 s dan pada pengujian dengan waktu 90 s nilai rata-rata jittersnya adalah 0.00059 s.

#### 4.4.3.3 Hasil Perhitungan QoS Pengaruh Lokasi Jarak Kumbang

Pada tabel 4.4 akan ditampilkan hasil perhitungan QoS pada pengujian pengaruh lokasi jarak kumbang terhadap kinerja jaringan.

**Tabel 4.4** Hasil Perhitungan QoS Pengaruh Lokasi Jarak Kumbang

Jarak (m)	Throughput (bits/s)	Packet loss	Rata-Rata Delay (ms)	Rata-rata Jitter (ms)
1	368	0	1245	0.5
3	280	0	1553	2/38
5	(-)	(-)	(-)	(-)

Throughput pada pengujian dengan jarak 1 m mendapatkan nilai throughput sebesar 368 bit/s, pada pengujian dengan jarak 3 m didapatkan nilai throughput sebesar 280 bit/s dan pada pengujian dengan jarak 5 m tidak ada data yang *tercapture*. Pada pengujian ini tidak ada packet loss, sehingga didapatkan nilai throughputnya 100%.

Selanjutnya untuk parameter rata-rata delay pada pengujian 1 m didapatkan nilai sebesar 1.245. Pada jarak 3 m didapatkan nilai rata-rata delay sebesar 1.553 s dan pada jarak 5 m tidak ada data yang *tercapture*.

Lalu untuk parameter jitter pada pengujian dengan jarak 1 m yaitu sebesar 0.0005 s. Pada pengujian dengan jarak 3 m didapatkan rata-rata jitter sebesar 0.00238 s dan pada jarak 5 m tidak ada data yang *tercapture*.

Pada pengujian dengan jarak 5 m, tidak mendapatkan hasil *capture* jaringan yang diinginkan. Ini disebabkan oleh kemampuan hardware yang dipilih. Pada jarak 5 m, divais IoT yang digunakan sudah tidak bisa mendapat jangkauan internet.

## 4.5 Pembahasan

Pada pembuatan kumbang jamur, tidak ada kendala yang cukup berarti. Kumbang yang sudah dibuat dapat berdiri dengan tegak, dan sudah siap untuk digunakan untuk budidaya jamur.

Selanjutnya pada saat pembuatan hardware juga tidak begitu banyak masalah karena sudah mengikut datasheet dari perangkat yang digunakan. Adanya beberapa referensi juga sangat membantu pada saat pembuatan hardware ini.

User maksimum dari perangkat ini dapat ditinjau dari dua aspek yaitu kemampuan hardware dan penggunaan cloud platform. Dari aspek hardware kapasitas maksimum sensor yang dapat terhubung adalah 32 sensor dan hanya bisa terkoneksi dengan dua mikrokontroler lain, hal ini ditinjau dari datasheet dari mikrokontroler yang digunakan yaitu Arduino Nano. Lalu dari aspek cloud platform yang digunakan yaitu Blynk dapat digunakan lebih dari 10 sensor apabila menggunakan premium user.

Untuk pembuatan software terjadi beberapa masalah seperti pada saat penggunaan modul Bluetooth. Umumnya modul Bluetooth hanya digunakan untuk komunikasi secara single-hop. Sehingga untuk skenario pada penelitian ini terjadi beberapa masalah pada saat melakukan parsing data. Untuk keseluruhan, masalah yang terjadi hanya pada saat pemberian program pada Bluetooth saja.

Setelah hardware dan software sudah dibuat, selanjutnya adalah proses integrasi. Pada proses ini akan menggabungkan keseluruhan sistem yang dibuat hingga dapat bekerja dengan baik. Adapun masalah pada saat proses ini yaitu terputusnya koneksi dari hardware Bluetooth. Namun dapat diselesaikan dengan cara restart mikrokontroler.

Untuk mengetahui QoS dari Qumbang-Q, maka harus dilakukan pengukuran. Pada saat pengukuran dilakukan adapun masalah yang dialami yaitu tidak bisanya *wireshark* untuk meng-*capture* jaringan yang hanya menuju ke IP dari Blynk. Sehingga digunakanlah software lain yaitu PCAP Remote. Untuk proses perhitungan dari data yang sudah didapatkan tidak mengalami masalah.

#### **4.6 Perbandingan Performansi Layanan Sistem**

Dari hasil pengukuran dan perhitungan yang sudah didapat selanjutnya akan dilakukan perbandingan performansi sistem dengan tujuan untuk mengetahui pengaruh beberapa pengujian yang dilakukan terhadap performansi layanan sistem.

**Tabel 4.5** Deskripsi Kondisi Pengukuran

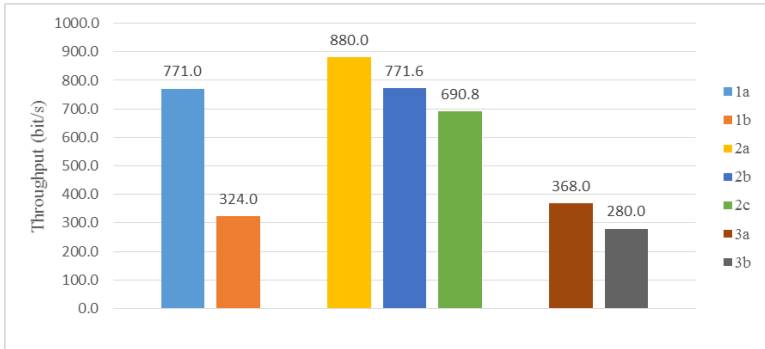
<b>No</b>	<b>Skenario Pengujian</b>	<b>Kondisi</b>	<b>Kode</b>
1	Pengaruh Jumlah Kumbung	1 Kumbung, 1 m, 60 s	1a
		2 Kumbung, 1 m, 60 s	1b
2	Lama Pengukuran	30 s, 2 kumbung, 1 m	2a
		60 s, 2 kumbung, 1 m	2b
		90 s, 2 kumbung, 1 m	2c
3	Jarak Pengukuran	1 m, 2 kumbung, 60 s	3a
		3m, 2 kumbung, 60 s	3b
		5m, 2 kumbung, 60 s	3c

Tabel 4.5 merupakan tabel deskripsi dari masing-masing pengujian yang dikodekan guna mempermudah untuk membuat klasifikasi dari hasil yang didapatkan.

#### **4.6.1 Perbandingan Parameter Hasil Pengukuran**

##### **A. Perbandingan Parameter Throughput**

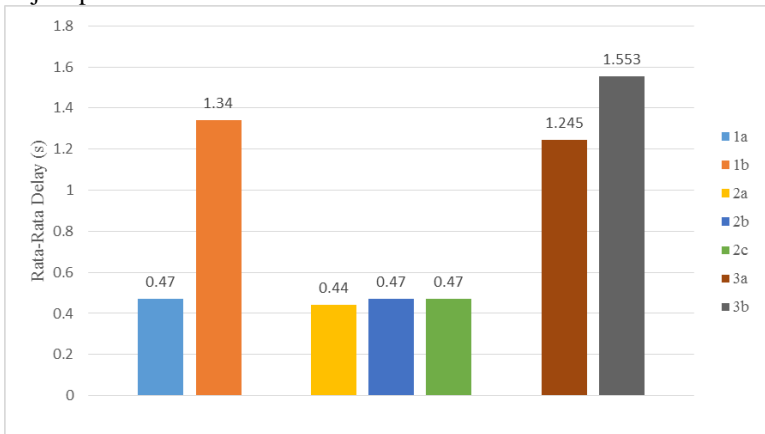
Dari gambar 4.12 dapat dilihat berbagai nilai throughput dari semua pengujian yang dilakukan. Nilai throughput yang tertinggi adalah pada saat dilakukan pengukuran selama 30 s dan nilai terendah dari throughput yang didapatkan adalah pada saat dilakukan pengujian dengan jarak 3 m.



**Gambar 4.12** Perbandingan Rata-Rata Throughput

### B. Perbandingan Parameter Delay

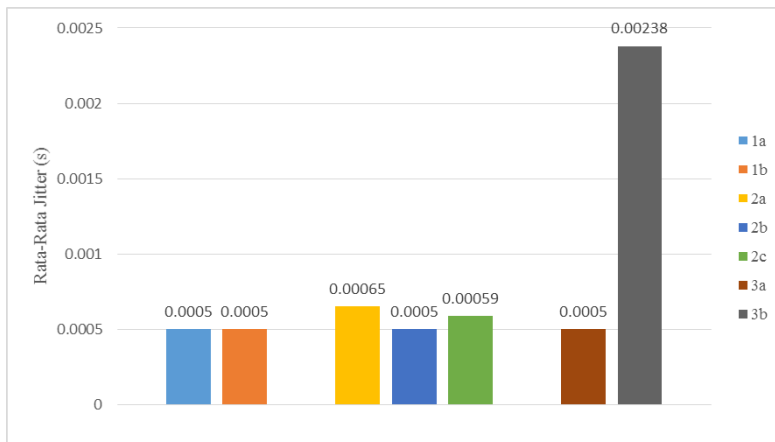
Pengaruh pengujian terhadap delay akan ditampilkan pada gambar 4.13. Dari keseluruhan nilai yang didapat, nilai delay terbesar adalah pada saat pengujian dengan jarak 3 m dan delay terbaik adalah pada saat pengujian dengan waktu pengamatan 30 s. Dapat disimpulkan bahwa delay sangat dipengaruhi oleh kuat jaringan internet dan juga jumlah input data yang terjadi pada komunikasi data.



**Gambar 4.13** Rata-Rata Delay Dari Semua Pengujian

### C. Perbandingan Parameter Jitter

Pada gambar 4.14 akan ditampilkan hasil pengujian terhadap jitter.



**Gambar 4.14** Rata-Rata Jitter Dari Semua Pengujian

Nilai jitter terbaik adalah pada saat dilakukan pengujian pengaruh jumlah kumbang. Pada saat dilakukan pengujian pengaruh lama pengamatan nilai jitter tidak terpaut jauh. Namun pada saat dilakukan pengujian pengaruh jarak mendapatkan nilai jitter yang sangat tinggi yaitu pada saat dilakukan pengujian dengan jarak 3 m.

### D. Perbandingan Parameter Packet Loss

Dari semua pengujian yang dilakukan tidak ada packet loss yang timbul pada saat pengiriman dan penerimaan data.

#### 4.6.2 Klasifikasi Parameter Hasil Pengukuran

Dari semua hasil yang didapatkan maka selanjutnya hasil tersebut akan diklasifikasikan menggunakan standar dari TIPHON.

Selanjutnya pada tabel 4.6 akan tertampil tabel klasifikasi hasil pengukuran.

**Tabel 4.6** Klasifikasi Hasil Pengukuran

No	Kode	Hasil Pengukuran		Klasifikasi Tiphone	
		Parameter	Nilai	Skor	Klasifikasi
1	1a	Throughput (bit/s), n%	771, 100%	4	Sangat Baik
		Delay (s)	0.47	1	Jelek
		Jitter (s)	0.0005	4	Sangat Baik
		Packet Loss	0	4	Sangat Baik
2	1b	Throughput (bit/s), n%	324, 100%	4	Sangat Baik
		Delay (s)	1.34	1	Jelek
		Jitter (s)	0.0005	4	Sangat Baik
		Packet Loss	0	4	Sangat Baik
3	2a	Throughput (bit/s), n%	880, 100%	4	Sangat Baik
		Delay (s)	0.44	2	Sedang
		Jitter (s)	0.00065	4	Sangat Baik
		Packet Loss	0	4	Sangat Baik
4	2b	Throughput (bit/s), n%	771.6, 100%	4	Sangat Baik
		Delay (s)	0.47	1	Jelek
		Jitter (s)	0.0005	4	Sangat Baik
		Packet Loss	0	4	Sangat Baik
5	2c	Throughput (bit/s), n%	690.8, 100%	4	Sangat Baik
		Delay (s)	0.47	1	Jelek
		Jitter (s)	0.00059	4	Sangat Baik

No	Kode	Hasil Pengukuran		Klasifikasi Tiphone	
		Parameter	Nilai	Skor	Klasifikasi
		Packet Loss	0	4	Sangat Baik
6	3a	Throughput (bit/s),n%	368, 100%	4	Sangat Baik
		Delay (s)	1.245	1	Jelek
		Jitter (s)	0.0005	4	Sangat Baik
		Packet Loss	0	4	Sangat Baik
7	3b	Throughput (bit/s),n%	280 , 100%	4	Sangat Baik
		Delay (s)	1.553	1	Jelek
		Jitter (s)	0.00238	4	Sangat Baik
		Packet Loss	0	4	Sangat Baik
8	3c	Throughput (bit/s),n%	(-)	(-)	(-)
		Delay (s)	(-)	(-)	(-)
		Jitter (s)	(-)	(-)	(-)
		Packet Loss	(-)	(-)	(-)

Setelah mendapatkan hasil klasifikasi dalam masing-masing pengujian, akan dibuat hasil rata-rata dari semua pengujian. Pada tabel 4.7 akan ditampilkan hasil dari klasifikasi rata-rata dari semua pengujian yang sudah dilakukan.



**Tabel 4.7** Tabel Klasifikasi Rata-Rata

No	Kode	Parameter				
		Through put (bit/s)	Throug hput (%)	Delay (ms)	Jitter (ms)	Packet Loss
1	1a	771	100	470	0.5	0
2	1b	324	100	1340	0.5	0
3	2a	880	100	440	0.65	0
4	2b	771.6	100	470	0.5	0
5	2c	690.8	100	470	0.59	0
6	3a	368	100	1245	0.5	0
7	3b	280	100	1553	2.38	0
8	3c	(-)	(-)	(-)	(-)	(-)
<b>Rata-Rata</b>		583.62	100	0.85	0.00080	0
<b>Klasifikasi</b>		Sangat Baik	Sangat Baik	Jelek	Sangat Baik	Sangat Baik

Dari rata-rata tersebut maka dapat dikatakan bahwa QoS dari Qumbung-Q dalam kategori baik.

## 4.7 Analisa Penggunaan Qumbung-Q

Pada bagian ini akan dibahas analisa penggunaan Qumbung-Q mulai dari cara penggunaan, kapasitas sistem, dan analisa bisnis yang telah dibuat.

### 4.7.1 Langkah-Langkah Penggunaan Qumbung-Q

Adapun langkah-langkah penggunaan Qumbung-Q bagi petani jamur tiram adalah sebagai berikut :

1. Mendownload aplikasi Blynk pada smarthphone (untuk Android bisa didownload di Play Store dan untuk iOS dapat didownload pada AppStore).
2. Membuat akun Blynk yang bisa dilakukan didalam aplikasi Blynk.
3. Membuka software Arduino IDE pada laptop/ PC.

4. Menghubungkan divais IoT ke laptop/ PC dengan cara memasang kabel mikro USB pada laptop dan divais IoT.
5. Memasukkan kode API/ kode karakter dari akun Blynk ke dalam program. Untuk memasukkan kode API ini seperti pada gambar 4.14

```
char auth[] = "z07Fo_nbwWyHC-1H01j6tCPxb5JRRcmG";
```

**Gambar 4. 15** Kolom Pengisian Kode Blynk Di Dalam Program Yang Sudah Disediakan

6. Memasukkan nama WiFi dan Password WiFi kedalam program yang sudah disediakan.

```
char ssid[] = "Indonesia"; // Nama WiFi
char pass[] = "12345678"; // Masukkan pasword WiFi.
```

**Gambar 4.16** Kolom Pengisian Nama Wifi Dan Password Wifi

7. *Upload* program terbaru dengan cara meng-*click* tanda panah yang ada pada software Arduino IDE.



**Gambar 4.17** Tanda Upload Pada Arduino IDE

8. Memutus koneksi laptop/ PC dengan divais IoT, dan menggantikannya dengan power supply yang akan tersambung dengan divais IoT.

9. Melihat data pada *user interface* yang sudah terpasang pada smartphone yaitu *software* Blynk.
10. Qumbung-Q sudah siap digunakan.

#### **4.7.2 Analisa Proses Bisnis**

Internet Of Things merupakan sebuah inovasi penyelesaian sebuah masalah yang terjadi diberbagai bidang. Oleh karena itu IoT sangat dekat dengan bisnis karena pada pembuatan sekaligus implementasi diharapkan dapat menjadi sebuah solusi bagi berbagai bidang.

Selama ini petani jamur mendapatkan keuntungan yang tidak menentu akibat cuaca yang tidak stabil. Hal inilah yang mendasari dibuatnya sebuah sistem yang mampu menjamin kestabilan iklim mikro yang ada pada kumbung jamur. Petani jamur biasanya mengalami gagal panen dimusim kemarau panjang sehingga banyak diantara mereka yang memilih untuk tidak menanam jamur disaat cuaca kemarau panjang. Dengan adanya Qumbung-Q diharapkan para petani tetap mampu untuk mendapatkan hasil panen yang sama ketika musim kemarau berlangsung.

Biaya pembuatan dari sistem Qumbung-Q sendiri adalah berkisar Rp. 500.000,-. Angka tersebut bisa dibilang terjangkau jika melihat dampak yang akan dihasilkan yaitu dengan menciptakan sebuah iklim mikro pada rumah jamur yang stabil sehingga dapat meningkatkan hasil panen. Mengingat banyaknya petani jamur yang memilih tidak menanam pada musim kemarau, alat ini dapat menjadi solusi karena dapat menjamin sebuah kondisi lingkungan yang ideal untuk budidaya jamur tiram.

#### **4.7.3 Analisa Pengembangan Sistem**

Setelah dilakukan pengujian fungsional, dapat diketahui juga berapa jumlah kapasitas maksimum user untuk perangkat yang sudah dibuat. Melihat dari mikrokontroler (Arduino Nano) yang digunakan dan modul komunikasi yang digunakan (Bluetooth), kapasitas maksimumnya yaitu dua mikrokontroler yang terhubung secara wireless, hal ini dikarenakan karena pada Arduino Nano hanya memiliki dua kaki Tx Rx. Untuk jumlah sensor yang terpasang maksimum ada 34 sensor ditinjau dari datasheet Arduino Nano.

Namun pada realisasi yang sudah dilakukan, mikrokontroler tidak hanya digunakan untuk mengontrol sensor saja melainkan juga digunakan untuk mengontrol aktuator dan modul komunikasi. Selain itu setelah direalisasi, ketika mikrokontroler menggunakan modul komunikasi Bluetooth yang hanya menghubungkan ke 1 mikrokontroler lain sudah terjadi error pada saat transmisi data.

Modul komunikasi yang digunakan memiliki kelemahan pada jangkauan kerja yang terbatas sehingga disarankan untuk memilih modul komunikasi yang dapat bekerja dengan range yang lebih tinggi. Selain itu pemilihan mikrokontroler juga harus dipertimbangkan supaya tidak *over design* pada pembuatannya disarankan sesuai kebutuhan saja tidak perlu mikrokontroler yang memiliki spek terlalu tinggi.

## **BAB 5**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari keseluruhan pembahasan tugas akhir dengan judul: Implementasi Internet of Things Untuk Pemantauan dan Pengendalian Iklim Pada Budidaya Jamur Tiram” dapat ditarik kesimpulan sebagai berikut :

1. Sistem Qumbung-Q (Hardware dan Software) dapat berfungsi dengan baik dibuktikan dengan hasil pengujian fungsional yang mana semua perangkat dapat berfungsi dengan baik.
2. Qumbung-Q hanya mampu bekerja dijarak 1-4 meter saja. Hal ini dikarenakan modul komunikasi yang digunakan adalah Bluetooth.
3. Qumbung-Q adalah alat bantu budidaya jamur pada tahap Inkubasi.
4. Kapasitas user dari Qumbung-Q adalah dua user. Namun bisa berubah apabila mengganti modul komunikasi yang digunakan.
5. Throughput dari Qumbung-Q dengan nilai rata-rata 583.62 bps, delay dari Qumbung-Q dengan nilai rata-rata 850 ms, jitter dari Qumbung-Q dengan rata-rata nilai 0.8 ms dan packet loss dari Qumbung-Q tidak ada.

#### **5.2 Saran**

Dari hasil penelitian tugas akhir ini, berikut merupakan saran yang dapat dilakukan untuk penelitian selanjutnya :

1. Mencari IP dari platform yang digunakan agar dapat memudahkan untuk mencari *QoS*-nya.
2. Menggunakan hardware yang lebih mumpuni untuk modul komunikasinya, sehingga mendapat user maksimum yang lebih banyak.
3. Menggunakan Mini PC agar dapat sekaligus menjadi server dan memudahkan dalam pengukuran *QoS*.

*-halaman ini sengaja dikosongkan-*

## DAFTAR PUSTAKA

1. Vandana S., Ravi Tiwari “A review paper on “ IOT ” & It “ s Smart Applications” International Journal of Science, Engineering and Technology Research (IJSETR), Volume 5, Issue 2, India,February 2016
2. Jagjeet K., Ramandeep K., Manpreet K. “Bluetooth Technology”, International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume – 5 Issue -03, India, Maret, 2016
3. Putri “ANALISIS QoS PADA JARINGAN MULTI PROTOCOL LABEL SWITCHING (MPLS) STUDI KASUS DI PELABUHAN INDONESIA III CABANG TANJUNG INTAN CILACAP”, undip, Maret,2009
4. Parjimo dan Agus andoko “Budidaya Jamur (Jamur Kuping, Jamur Tiram,Jamur Merang)”, Agromedia, Jarakta, 2013.
5. Pathil, “STUDY OF SOME MEDICINAL PLANTS FOR ANTIDIABETIC ACTIVITY IN ALLOXAN INDUCED DIABETES”, Pharmacologyonline, India, 2010
6. Caglarima, “Critical Mineral Resource Of United States”, IMWA, IEEE, China, 2007
7. Parikh E., “Diets and Cardiovascular Disease: An Evidence-Based Assessment”, PMID, India 2005
8. Paspaspyridi, “From Biomass To Advanced Bio-Based Chemicals & Materials : A Multidisciplinary Perspective”, Frontiers in Chemistry, 2012
9. Web resmi WHO tentang kandungan asam pada jamur tiram, 2008 <<https://www.who.int/news-room/fact-sheets/detail/healthy-diet>>
10. Zulfarina,” Budidaya Jamur Tiram dan Olahannya untuk Kemandirian Masyarakat Desa”, (Indonesian Journal of Community Engagement, Riau, 2019
11. Aufa Aulia, “Pengendalian Mutu Jamur Tiram”, UNS, Surakarta, 2015
12. Actur S., “Bahasa Pemrograman C” , Linkedin, Jakarta, 2013

13. David W., “Jenis-Jenis Cloud Computing” <<https://www.progresstech.co.id/blog/jenis-cloud/>>, 2017
14. Fauzi A., “Cara kerja, cara menggunakan wireshark”< Fauziadnan.com/2018/>, 2018



## LAMPIRAN A

Lampiran A berisikan listing program.

### 1. Program pada Q1

```
#include <Adafruit_Sensor.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <DHT_U.h>

//SoftwareSerial BTSerial(10, 11);

#define DHTPIN 3 // pin 0 node mcu
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE);

#define out1 4
#define out2 5
#define out3 8

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  //BTSerial.begin(38400);
  dht.begin();
  pinMode(out1, OUTPUT); // fan 1
  pinMode(out2, OUTPUT); // water-mist
  digitalWrite(out1, LOW);
  digitalWrite(out2, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  String send_message;
  int temp =(int)dht.readTemperature();
  int hum =(int)dht.readHumidity();

  if (isnan(hum) || isnan(temp)) {
```

```

Serial.println("Failed to read from DHT sensor!");
return;
}
else
{
    //Print temp and humidity values to serial monitor
    //Serial.print("Humidity ");
    Serial.print(hum);
    Serial.print(",");
    //Serial.print("% , Temp: ");
    Serial.println(temp);
    //Serial.println(" Celcius ");
    //Serial.print("Temperature Heat Index: ");
    //Serial.println(0.8*(temp)+(((hum)/100)*(temp)/500);

Serial.write(hum);Serial.write(",");Serial.write(temp);Serial.write("\n");

    if (temp>=22 && temp<=28){ //parameter FAN
        digitalWrite(out1, LOW);
    }
    else if (hum<=70){ //parameter MIST-MAKER
        digitalWrite(out2, HIGH);
    }
    else
    {
        digitalWrite(out1, HIGH);
        digitalWrite(out2, LOW);
    }
    delay(500);

}
}

```

## 2. Program pada Q2

```

#include <Adafruit_Sensor.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <DHT_U.h>

```

```

#define DHTPIN 2 // pin 0 node mcu
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  int hum1 = (int)dht.readHumidity();
  int temp1 = (int)dht.readTemperature();

  while(Serial.available() > 0){

    int hum = Serial.parseInt();
    int temp = Serial.parseInt();

    Serial.print(hum);Serial.print(",");Serial.print(temp);Serial.print(",");Serial
    .print(hum1);Serial.print(",");Serial.println(temp1);

    if(Serial.read() == '\n'){

      Serial.write(hum);Serial.write(",");Serial.write(temp);Serial.write(",");S
      erial.write(hum1);Serial.write(",");Serial.write(temp1);Serial.write("\n");
    }
  }
  delay(1000);
}

```

### 3. Program Pada Divais IoT

```

#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

```

```

char auth[] = "zO7Fo_nbwWyHC-1H0lj6tCPxb5JRRcmG"; //
You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char ssid[] = "Indonesia"; // Your WiFi credentials.
char pass[] = "12345678"; // Set password to "" for open networks.

WiFiClient client;

void setup()
{
  Serial.begin(9600);
  Serial.print("Connecting to ");Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.print("Netmask: ");Serial.println(WiFi.subnetMask());
  Serial.print("Gateway: ");Serial.println(WiFi.gatewayIP());
  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
  while(Serial.available() > 0){
    int hum = Serial.parseInt();
    int temp = Serial.parseInt();
    int hum1 = Serial.parseInt();
    int temp1 = Serial.parseInt();

    Serial.print(hum);Serial.print(",");Serial.print(temp);Serial.print(",");Serial
    .print(hum1);Serial.print(",");Serial.println(temp1);
    delay(7000);
  }
}

```

```
Blynk.virtualWrite(V0, temp);  
Blynk.virtualWrite(V1, hum);  
Blynk.virtualWrite(V2, temp1);  
Blynk.virtualWrite(V3, hum1);  
}  
}
```

## LAMPIRAN B

Hasil survey lapangan yang sudah dilakukan.



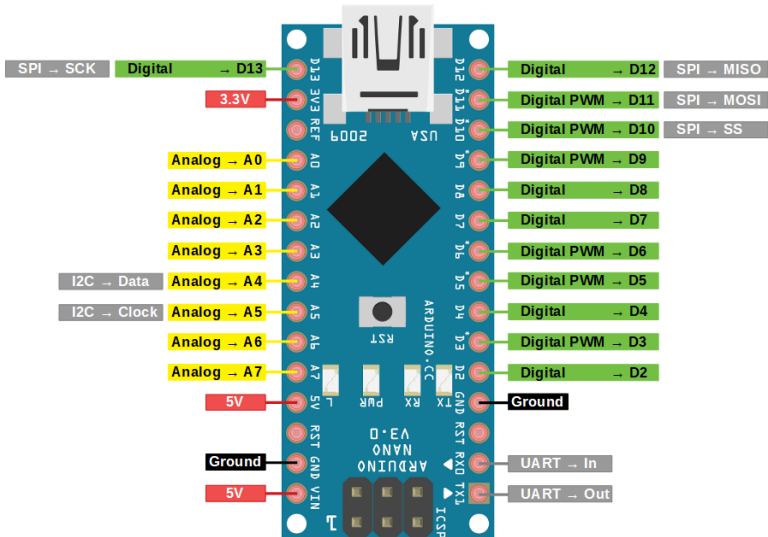
Gambar bibit jamur sebelum masuk masa inkubasi.



Pembuatan bibit jamur sebelum dipindah ke media tanam

## LAMPIRAN C

Datasheet dari Arduino Nano



Source: Fritzing

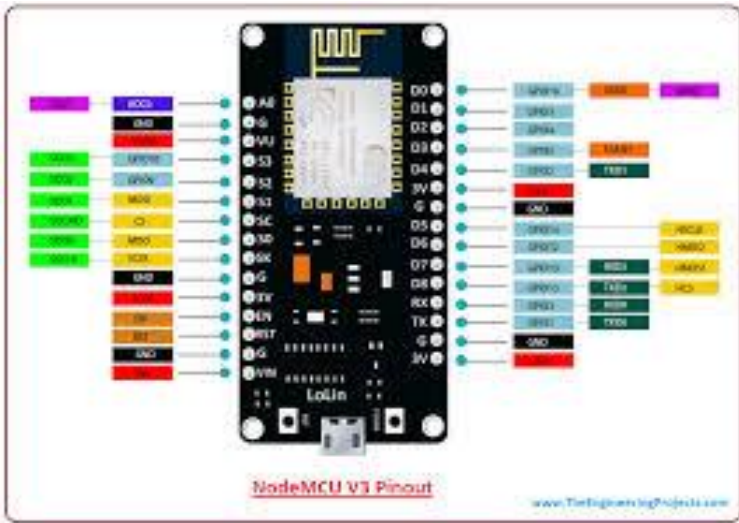
Datasheet dari DHT 22

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND

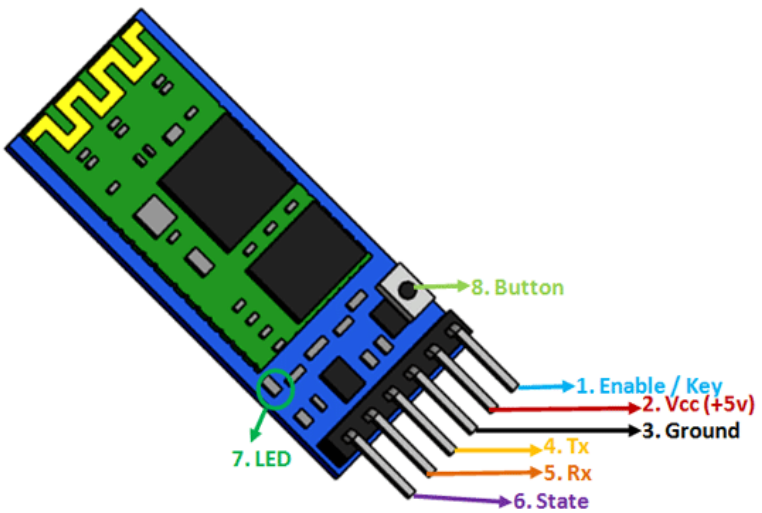




Datasheet dari NodeMCU ESP-8266



Datasheet dari Bluetooth HC-05



## LAMPIRAN D

### Departemen Teknik Elektro

Fakultas Teknologi Elektro dan Informatika Cerdas - ITS

#### EE 184801 TUGAS AKHIR – 6 SKS

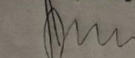
Nama Mahasiswa : Calvin Rafindra Adhinta  
Nomer Pokok : 0711 16 4000 0012  
Bidang Studi : Telekomunikasi Multimedia  
Tugas Diberikan : Semester Genap Tahun Ajaran 2019/2020  
Dosen Pembimbing : Sri Rahayu, S.T., M.Kom.  
Judul Tugas Akhir : **Implementasi Internet Of Things (IOT) untuk Pemantauan dan Pengendalian Iklim Pada Budidaya Jamur Tiram**  
(*Implementation of Internet of Things (IoT) for Climate Monitoring and Control in Oyster Mushroom Cultivation*)

#### Uraian Tugas Akhir :

Berbagai layanan teknologi berbasis internet sudah banyak ditawarkan untuk memudahkan kehidupan manusia, diantaranya *Internet of Things (IoT)*. Seiring bergulirnya Revolusi Industry 4.0, aplikasi IoT termasuk yang banyak didorong implementasinya untuk berbagai keperluan termasuk dalam sistem pemantauan (*monitoring system*) maupun sistem pengendalian (*controlling system*). Dan dewasa ini terlihat berbagai aplikasi IoT yang sudah merambah di banyak bidang, diantaranya dalam bidang pertanian (*smart agriculture*). Dengan memanfaatkan teknologi IoT tersebut diharapkan pekerjaan yang semula rumit menjadi lebih mudah dan efektif.

Dalam tugas akhir ini, akan dibuat satu contoh implementasi sistem IoT (Qumbang-Q) untuk keperluan budidaya jamur tiram. Budidaya jamur tiram sendiri dikenal cukup sulit, dan rentan gagal panen akibat terjadinya fluktuasi suhu, dan kelembaban di area tanam. Dalam hal ini sistem IoT berfungsi menstabilkan kondisi iklim mikro dalam kumbung jamur, yang dapat dikontrol secara otomatis oleh mikrokontroler serta dapat dimonitor jarak jauh lewat jaringan internet. Untuk merealisasikan prototipe sistem digunakan sejumlah sensor, aktuator, mikrokontroler, modul-IoT, modul-*wireless* dan beberapa perangkat pendukung lain. Adapun untuk mengevaluasi kinerja sistem dilakukan sejumlah pengujian yang meliputi pengujian komponen, segmen komunikasi, dan parameter QoS (delay, throughput). Sedangkan untuk mendukung pembuatan tugas akhir digunakan perangkat lunak *Arduino IDE*, *PuTTY* dan *Thingspeak (Cloud-Platform)*.

Dosen Pembimbing



Sri Rahayu, S.T., M.Kom.

NIP : 1968 02 28 1997 02 2001

Mengetahui,  
Sekretaris Departemen I

Menyetujui  
Kepala Lab. Jaringan Telekomunikasi



Dimas Anton Asfani, ST., MT., Ph.D.

NIP : 1981 09 05 2005 01 1002

Eko Setijadi, ST., MT., Ph.D.

NIP : 1972 10 01 2003 12 1002