

ANALISIS DAN IMPLEMENTASI PROTOTIPE PENGATUR KELEMBABAN BERBASIS INTERNET OF THINGS (IoT) PADA PENYIMPANAN SAYUR

Analysis and Implementation Prototype of Controlling Humidity based Internet of Things (IoT) on Vegetable Storage

Fadhil Riardy Rivai¹, Dr. Ir. Rendy Munadi M.T.², Unang Sunarya S.T., M.T.³

^{1,2,3} Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

Jln. Telekomunikasi No. 1 Terusan Buah Batu Bandung 40257 Indonesia

¹Fadhilriardy@gmail.com ²Rendymunadi@telkomuniversity.ac.id ³Unangsunarya@telkomuniversity.ac.id

ABSTRAK

Disaat populasi penduduk semakin meningkat, maka kebutuhan terhadap sayuran juga meningkat, sehingga produksi dan hasil panen sayuran juga bertambah drastis demi melengkapi kebutuhan pokok manusia. Untuk menyimpan hasil panen sayur dibutuhkan penyimpanan yang segar dan higienis, agar kebersihan dan kesegaran sayur tetap terjaga. Tetapi saat ini masih banyak tempat penyimpanan sayur yang belum maksimal dalam menjaga kualitas sayur hasil panen para petani. Oleh karena itu dibutuhkan penyimpanan yang mampu membuat sayuran menjadi tahan lebih lama dan lebih segar.

Dengan berkembang pesatnya teknologi saat ini memungkinkan untuk merancang sebuah prototipe pengatur kelembaban pada penyimpanan sayur berbasis *arduino* dan *Monitoring* data dengan konsep *internet of things* (IOT). Prototipe ini menggunakan *arduino uno* sebagai pengontrol utama, DHT11 sebagai sensor utama membaca kelembaban dan suhu, serta *relay* sebagai pengatur penyiraman otomatis. Untuk mengirimkan data ke internet prototipe ini menggunakan modul *wifi* NodeMcu ESP8266 dengan menggunakan protokol HTTP dan MQTT.

Berdasarkan pengujian dan analisis *Quality of Service* (QoS) yang dilakukan pada prototipe dengan metode protokol MQTT dan HTTP. Pada protokol MQTT nilai delay yang didapat sebesar 157.749772 ms dan nilai *jitter* sebesar 5.8323392. Sedangkan pada protokol HTTP nilai delay didapat sebesar 156.436981 ms dan nilai *jitter* sebesar 5,79794392 ms. Dalam pengujian alat ini mampu membuat kondisi kubis dan bayam tetap segar dalam waktu 4 hari.

Kata kunci : IoT, NodeMcu ESP8266, Arduino Uno, HTTP, MQTT, DHT11.

ABSTRACT

When a population increasing, then the need for vegetables is also increasing. To store their harvest a vegetable needed storage that is fresh and hygienic, for cleanliness and freshness of vegetables are maintained. But now there are many vegetable storage areas that have not been maximized in maintaining the quality of vegetable crops of the farmers. Therefore it takes storage that can make vegetables to last longer and fresher.

*With the rapidly growing technology today it is possible to design a prototype of moisture regulator on vegetable storage based *arduino* and *Monitoring* data with the concept of *internet of things* (IOT). This prototype uses *arduino uno* as the main controller DHT11 as the main sensor reads humidity and temperature, as well as *relay* as automatic watering regulator. To transmit data to the internet this prototype uses NodeMcu ESP8266 *wifi* module using HTTP and MQTT protocol.*

Based on testing and analysis of Quality of Service (QoS) conducted on prototypes using the MQTT and HTTP protocol methods. In the MQTT protocol the value of the delay obtained is 157.749772 ms and the jitter value is 5.8323392. Whereas in the HTTP protocol the delay value is 156.436981 ms and the jitter value is 5.79794392 ms. In testing this prototype is able to keep cabbage and spinach conditions fresh within 4 days.

Keywords : IoT, NodeMcu ESP8266, Arduino Uno, HTTP, MQTT, DHT11.

1. Pendahuluan

1.1 Latar Belakang

Bidang pertanian sangat penting untuk memenuhi kebutuhan pokok manusia. Sayur merupakan salah satu hasil dari sektor pertanian. Para petani menghasilkan berbagai macam sayuran seperti ; kubis, bayam, dll.

Disaat populasi penduduk semakin meningkat, maka kebutuhan terhadap sayuran juga meningkat, sehingga produksi dan hasil panen sayur juga bertambah drastis demi melengkapi kebutuhan pokok manusia. Untuk menyimpan hasil panen sayur dibutuhkan penyimpanan yang segar dan higienis, agar kebersihan dan kesegaran sayur tetap terjaga, tetapi saat ini masih banyak tempat penyimpanan sayur yang belum maksimal dalam menjaga kualitas sayur hasil panen para petani.

Dengan berkembangnya teknologi saat ini memungkinkan untuk membuat teknologi pengatur kelembaban pada penyimpanan sayuran, yaitu dengan merancang sebuah penyimpanan sayur yang dapat mengatur kelembaban menggunakan penyiram otomatis yang terhubung dengan *internet of things* (IOT).

Sistem ini menggunakan arduino sebagai pengontrol utama, *DHT11* sebagai sensor kelembaban dan suhu untuk membaca nilai kelembaban dan suhu. Sensor juga digunakan sebagai indikator untuk menyalakan *sprayer* air otomatis. Dalam pengaplikasian alat ini diterapkan di dalam penyimpanan sayur, sehingga sayur hasil panen petani tetap segar selama berapa di penyimpanan dan dapat membuat sayuran bertahan lebih lama. Prototipe ini sudah terhubung dengan jaringan internet lokal. Sensor akan mengumpulkan data suhu serta kelembaban. Jika suhu dan kelembaban berada dibawah nilai yang ditetapkan, maka akan menghidupkan relay serta *sprayer* air. Data yang sudah terkumpul seperti suhu dan kelembaban akan dikirimkan oleh NodeMcu ESP8266 dengan menggunakan protokol HTTP dan MQTT. Kualitas jaringan akan di analisis setelah data di *monitoring* oleh user secara *real time*. Selanjutnya akan dilakukan analisis *Quality of Service* (QoS) pada jaringan tersebut.

2. Dasar Teori

2.1 Internet of Things (IoT)

Water Internet of Things atau IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung dengan sensor secara terus-menerus. Konsep ini mempunyai kemampuan seperti berbagi data, *remote control* kepada prototipe yang telah kita rancang. Komponen mikrokontroler yang telah kita rancang akan

bekerja dan tersambung ke jaringan internet lokal dan global. Prototipe tersebut mempunyai sensor, sehingga dapat memonitoring data secara *real time* melalui internet lokal ataupun global ^[1].

2.2 Mikrokontroler

Mikrokontroler adalah perangkat yang dirancang untuk mengambil perangkat *input* (masukan) dari suatu alat dan mengendalikan perangkat *output* (keluaran) dari suatu rangkaian. Sebagian perangkat elektronik yang ada sekarang ini memiliki mikrokontroler pada bagian intinya. Mikrokontroler yang dioptimalkan untuk mengendalikan input saja atau output saja ^[2].

2.3 Arduino Uno

Arduino adalah sebuah *platform open source* (sumber terbuka) yang digunakan untuk membuat proyek-proyek elektronika. Arduino terdiri dari dua bagian utama yaitu sebuah papan sirkuit fisik sering disebut juga dengan mikrokontroler dan sebuah perangkat lunak atau IDE (*Integrated Development Environment*) yang berjalan pada komputer ^[2].

2.4 Relay

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen Electromechanical (Elektromekanikal) yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/Switch). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi ^[3].

Sebagai contoh, dengan Relay yang menggunakan Elektromagnet 5V dan 50 mA mampu menggerakkan Armature Relay yang berfungsi sebagai saklarnya untuk menghantarkan listrik 220V 2A.

2.5 Kelembaban Udara

Kelembaban adalah kandungan jumlah uap air di udara. Jumlah uap air mempengaruhi proses proses fisika, kimia dan biologi di alam. Air yang terkandung di sekeliling udara adalah faktor yang penting untuk kesehatan manusia dan tumbuhan, oleh karena itu akan mempengaruhi lingkungan khususnya sayur sayuran dan hasil panen sayuran. Kelembaban merupakan suatu tingkat keadaan lingkungan udara basah yang disebabkan oleh adanya uap air ^[4].

Jika tekanan uap parsial sama dengan tekanan uap air yang jenuh maka akan terjadi

pemadatan. *Relative Humidity* didefinisikan sebagai presentase perbandingan antara tekanan uap air parsial dengan tekanan uap air jenuh.

2.6 Quality of Service (QoS)

Quality of Service (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefinisikan karakteristik dan sifat pada satu servis. Pada QoS menggunakan teknik untuk mengelola throughput, delay, jitter, dan packet loss dalam jaringan. Kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwith*, mengatasi *jitter* dan *delay*. Parameter QoS adalah *throughput*, *delay*, *jitter*, dan *packet loss* [6].

Parameter-parameter Quality of Service (QoS)

1. Throughput

Throughput merupakan kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada destination selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

Tabel 2.1 Kategori *Throughput*

Kategori Throughput	Throughput (bps)	Indeks
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Buruk	< 25	1

Persamaan perhitungan throughput =

$$= \frac{\text{Paket data diterima}}{\text{Lama pengamatan}}$$

2. Packet Loss

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion* pada jaringan.

Tabel 2.2 Kategori Degradasi

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

Persamaan perhitungan packet loss :

$$= \frac{\text{Paket data dikirim} - \text{Paket data diterima}}{\text{Paket data dikirim}} \times 100\% \quad (2)$$

3. Delay (Latency)

Delay (Latency) Adalah waktu yang dibutuhkan data untuk menempuh jarak dari titik asal ke titik tujuan. Delay dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Pada Tabel 2.3 diperlihatkan kategori dari delay dan besar delay.

Tabel 2.3 Kategori Delay

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

Persamaan perhitungan Delay (*Latency*)

$$= \frac{\text{Total waktu}}{\text{Total paket diterima}}$$

4. Jitter

Jitter diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket diakhir perjalanan jitter.

Pada **Tabel 2.4** diperlihatkan kategori dari jitter dan besar *jitter*.

Tabel 2.4 Kategori *Jitter*

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

Persamaan perhitungan *Jitter*

$$= \frac{\text{Total variasi delay}}{\text{Total paket data} - 1}$$

2.7 MQTT

MQTT adalah singkatan dari *Message Queuing Telemetry Transport*. MQTT adalah protokol konektivitas *Internet of Things (IoT)* yang berbasis *open source (Eclipse)* dengan standar terbuka (OASIS) yang dirancang untuk perangkat terbatas dan *bandwidth* rendah, dengan *latency* tinggi. Protokol ini berjalan pada diatas *stack Transmission Control Protocol / Internet Protocol (TCP/IP)* dan mempunyai ukuran paket data dengan *low overhead* yang kecil sehingga berefek pada konsumsi daya yang juga cukup kecil. Dan dapat di aplikasikan secara mobile di era IoT dimana *bandwith* dan daya baterai menjadi pertimbangan utama. Stack TCP/IP sekarang sudah banyak di dukung oleh mikrokontroler seri *device board* yang umum dipasaran seperti *Arduino Uno*,

Arduino + Ethernet Shield, ESP8266 WiFi , Raspberry Pi dll [7].

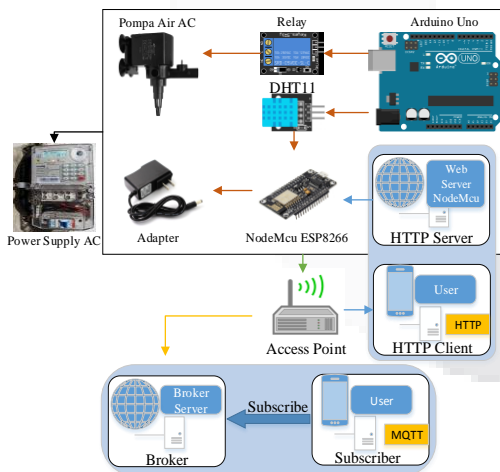
2.8 HTTP

HTTP (*Hypertext Transfer Protocol*) adalah sebuah protokol untuk *me-request* dan *me-response* sebuah data antara *client* seperti *web browser* pada *user* atau pada sebuah *web server* di suatu jaringan. Protokol ini bekerja diatas **Transmission Control Protocol / Internet Protocol** (TCP/IP). *Client* pada HTTP biasanya memulai permintaan dengan membuat **hubungan TCP/IP** ke *port* tertentu yang biasanya adalah *port 80*. HTTP mendefinisikan bagaimana suatu pesan dapat diformat dan dikirimkan dari *client* ke *server* atau dari *server* ke *client*. HTTP juga mengontrol sebagai mana yang harus dilakukan oleh *web server* dan *web browser* sebagai respon dari perintah protokol HTTP tersebut [8].

3. Perancangan Sistem

3.1 Perancangan Sistem

Blok ini secara keseluruhan adalah komponen utama meliputi : Arduino, sensor DHT11 untuk membaca suhu serta kelembaban, modul *wifi NodeMcu* ESP8266, *Relay* pengontrol , dan *sprayer* air. Pertama Sensor akan mendeteksi suhu dan kelembaban disekitar jika melewati atau kurang dari standar yang telah ditetapkan maka arduino akan membaca bahwa kelembaban menurun dan akan memberikan perintah untuk menghidupkan *relay*. Apabila *relay* aktif maka secara otomatis akan menghidupkan pompa air yang akan menyiram sayuran. Perhatikan gambar 3.1 sebagai berikut :



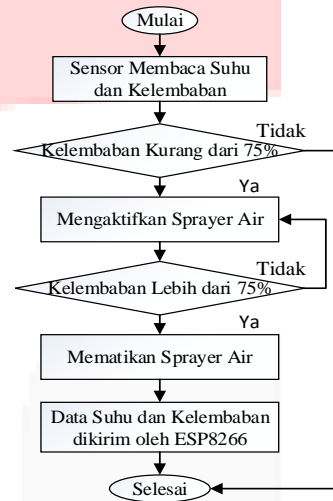
Gambar 3.1 Topologi Perancangan Sistem

3.2 Perancangan Perangkat Keras

Dalam proses pengerjaan tugas akhir ini dilakukan pemilihan komponen yang telah tersedia sesuai perancangan yang dibuat. Selanjutnya arduino dan sensor DHT11 sudah terhubung dan

aktif. Sensor sudah siap untuk membaca Suhu dan kelembaban didalam tempat penyimpanan sayur. Apabila sensor membaca kelembaban kurang dari 75% maka akan menghidupkan *sprayer* air otomatis. *Sprayer* air otomatis akan menyiram 1 kali selama 1 jam setelah itu sensor akan membaca kenaikan kelembaban selama rentang waktu tersebut.

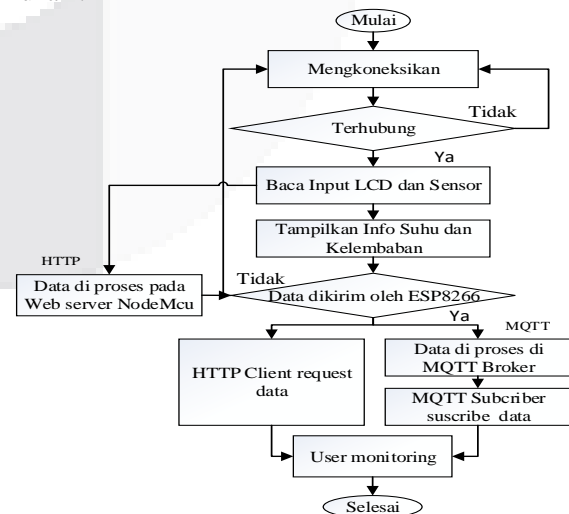
Setelah dilakukan penyiraman maka sensor akan membaca kembali kelembaban sayur, Apabila kelembaban sudah lebih dari 75% maka *sprayer* air otomatis akan tidak aktif. Setelah bagian pengontrol bekerja maka data suhu dan kelembaban akan dikirim oleh modul wifi NodeMcu ESP8266 menuju web server melalui akses point.



Gambar 3.2 Diagram alir perangkat keras

3.3 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak terdiri dari diagram alir perangkat lunak dan subsistem perangkat lunak yang digunakan. Berikut merupakan diagram alir dari cara kerja perangkat lunak :



Gambar 3.3 Diagram Alir Perangkat Lunak

Pada gambar 3.3 data-data yang telah diproses oleh perangkat keras akan dikirim melalui

4. Hasil Pengujian dan Analisis

4.1 Analisis Quality of Service (QoS) pada Protokol HTTP

1. Parameter Packet Loss

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang tidak terkirim dalam suatu jaringan. Percobaan dengan melakukan pengambilan 10 data dan Gambar 4.1 menunjukan total paket yang terkumpul sebanyak 108 paket yang terdapat pada summary wireshark.

Measurement	Captured	Displayed
Packets	108	108 (100.0%)
Time span, s	16.895	16.895
Average pps	6.4	6.4
Average packet size, B	116	116
Bytes	12518	12518 (100.0%)
Average bytes/s	740	740
Average bits/s	5927	5927

Gambar 4.1 Packet captured pada wireshark

Persamaan perhitungan packet loss :

$$= \frac{\text{Paket data dikirim} - \text{Paket data diterima}}{\text{Paket data dikirim}} \times 100\%$$

$$= \frac{10 - 10}{10} \times 100\%$$

$$= 0 \times 100\%$$

$$= 0 \%$$

Dengan nilai packet loss adalah 0% maka, nilai packet loss tersebut masuk dalam kategori sangat bagus dengan melihat standar indeks dari rujukan yang digunakan.

2. Parameter Delay (Latency)

Setelah melakukan pengambilan data pada HTTP dengan me-request dan response data sebanyak 10 kali. Menghasilkan total sebanyak 108 paket. Parameter yang digunakan untuk menghitung delay ini adalah total waktu dibagi total paket. Gambar 4.2 menunjukan nilai total waktu sebesar 16,895 sec dan total paket sebanyak 108 paket yang terdapat pada summary wireshark.

Measurement	Captured
Packets	108
Time span, s	16.895
Average pps	6.4
Average packet size, B	116
Bytes	12518
Average bytes/s	740
Average bits/s	5927

Gambar 4.2 Parameter perhitungan delay pada wireshark

Perhitungan delay dalam bentuk matematis untuk percobaan tersebut ditunjukan sebagai berikut :

Persamaan perhitungan Delay (Latency)

$$= \frac{\text{Total waktu}}{\text{Total paket diterima}}$$

$$= \frac{16.895194}{108}$$

modul ESP8266, selanjutnya data tersebut di proses dengan menggunakan protokol MQTT dan HTTP.

$$= 0.156436981 \text{ sec}$$

$$= 156.436981 \text{ ms/paket}$$

Dengan nilai delay adalah 156.436981 ms/paket maka, nilai rata – rata delay tersebut masuk dalam kategori bagus dengan melihat standar indeks dari rujukan yang digunakan.

3. Parameter Jitter

Jitter didapatkan dari hasil pembagian total variasi delay dan total paket yang diterima. Total paket yang diterima dikurangi nilai satu karena variasi delay dihitung diantara dua delay, sehingga bila dihitung keseluruhan paket menjadi minus satu.

Measurement	Captured
Packets	108
Time span, s	16.895
Average pps	6.4
Average packet size, B	116
Bytes	12518
Average bytes/s	740
Average bits/s	5927

Gambar 4.3 Jumlah packet captured pada wireshark

Nilai total paket yang diterima sebesar 108 dapat dilihat pada summary wireshark seperti pada gambar 4.16 dan nilai total variasi delay yaitu 0.62038 sec yang diambil pada gambar 4.17 dengan warna biru.

H	I	J
0.008271	0.00028	0.007991
0.001385	0.008271	-0.006886
0.000132	0.001385	-0.001253
0.000019	0.000132	-0.000113
0.000467	0.000019	0.000448
0.004128	0.000467	0.003661
1.564586	0.004128	1.560458
0.058954	1.564586	-1.505632
0.000098	0.058954	-0.058856
0.000252	0.000098	0.000154
0.149895	0.000252	0.149643
0.131121	0.149895	-0.018774
0.001119	0.131121	-0.130002
0.000146	0.001119	-0.000973
0.000023	0.000146	-0.000123
0.000601	0.000023	0.000578
0.00416	0.000601	0.003559
0.62038	0.00416	0.61622
total variasi delay(s)		0.62038
Jitter (s)		0.005797944
jitter (ms)		5.79794392

Gambar 4.4 Perhitungan Jitter pada HTTP

Persamaan perhitungan Jitter = $\frac{\text{Total variasi delay}}{\text{Total paket data} - 1}$

$$= \frac{0.62038}{108 - 1}$$

$$= 0.00579794392 \text{ sec}$$

$$= 5.79794392 \text{ ms/paket}$$

Total variasi delay dari proses penangkapan adalah 0.62038 sec atau sekitar 0.01033 menit, dan total paket yang diterima adalah 108 paket, sehingga nilai jitter yang didapatkan adalah 5.79794392 ms/paket. Hasil tersebut merupakan nilai bagus dengan melihat standar indeks dari rujukan yang digunakan.

4.2 Analisis Quality of Service (QoS) pada Protokol MQTT

1. Parameter Packet Loss

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang tidak terkirim dalam suatu jaringan. Percobaan dengan melakukan pengambilan 10 data Gambar 4.5 menunjukkan total paket yang terkumpul sebanyak 57 paket yang terdapat pada *summary wireshark*.

Measurement	Captured	Displa
Packets	57	57 (10
Time span, s	8.992	8.992
Average pps	6.3	6.3
Average packet size, B	117	117
Bytes	6664	6664
Average bytes/s	741	741
Average bits/s	5928	5928

Gambar 4.5 Jumlah Packet Captured pada *wireshark*

Persamaan perhitungan packet loss :

$$\begin{aligned}
 &= \frac{\text{Paket data dikirim} - \text{Paket data diterima}}{\text{Paket data dikirim}} \times 100\% \\
 &= \frac{10 - 10}{10} \times 100\% \\
 &= 0 \times 100\% \\
 &= 0\%
 \end{aligned}$$

Dengan nilai packet loss adalah 0% maka, nilai packet loss tersebut masuk dalam kategori sangat bagus dengan melihat standar indeks dari rujukan yang digunakan.

2. Parameter Delay (Latency)

Setelah melakukan pengambilan data pada MQTT dengan me-publish data sebanyak 10 kali. Menghasilkan total sebanyak 57 paket. Parameter yang digunakan untuk menghitung *delay* ini adalah total waktu dibagi total paket. Gambar 4.6 menunjukkan nilai total waktu sebesar 8,992 sec dan total paket sebanyak 57 paket yang terdapat pada *summary wireshark*.

Measurement	Captured
Packets	57
Time span, s	8.992
Average pps	6.3
Average packet size, B	117
Bytes	6664
Average bytes/s	741
Average bits/s	5928

Gambar 4.6 Parameter perhitungan *delay* MQTT pada *wireshark*

Perhitungan *delay* dalam bentuk matematis untuk percobaan tersebut ditunjukkan sebagai berikut :

Persamaan perhitungan Delay (Latency)

$$\begin{aligned}
 &= \frac{\text{Total waktu}}{\text{Total paket diterima}} \\
 &= \frac{8.991737}{57} \\
 &= 0.157749772 \text{ sec} \\
 &= 157.749772 \text{ ms/paket}
 \end{aligned}$$

Dengan nilai *delay* rata-rata adalah 157.749772 ms/paket maka, nilai rata – rata *delay* tersebut masuk dalam kategori bagus dengan melihat standar indeks dari rujukan yang digunakan.

3. Parameter Jitter

Jitter didapatkan dari hasil pembagian total variasi *delay* dan total paket yang diterima. Total paket yang diterima dikurangi nilai satu karena variasi *delay* dihitung diantara dua *delay*, sehingga bila dihitung keseluruhan paket menjadi minus satu.

Measurement	Captured
Packets	57
Time span, s	8.992
Average pps	6.3
Average packet size, B	117
Bytes	6664
Average bytes/s	741
Average bits/s	5928

Gambar 4.7 Jumlah packet captured pada *wireshark*

Nilai total paket yang diterima sebesar 57 dapat dilihat pada *summary wireshark* seperti pada gambar 4.20 dan nilai total variasi *delay* yaitu 0.326611 sec yang diambil pada gambar 4.8 dengan warna biru.

H	I	J
0.034963	0.196642	-0.161679
0.189085	0.034963	0.154122
0.116452	0.189085	-0.072633
0.205226	0.116452	0.088774
0.002062	0.205226	-0.203164
0.030787	0.002062	0.028725
0.177562	0.030787	0.146775
0.084492	0.177562	-0.093071
0.002218	0.084492	-0.082274
0.158326	0.002218	0.156108
0.325839	0.158326	0.167513
0.052611	0.325839	-0.273229
0.0678	0.052611	0.01519
0.109545	0.0678	0.041745
0.191971	0.109545	0.082426
0.252689	0.191971	0.060718
0.326611	0.252689	0.073922
total variasi delay(s)		0.326611
Jitter (s)		0.005832339
jitter (ms)		5.8323392

Gambar 4.8 Perhitungan Jitter pada MQTT

Persamaan perhitungan Jitter

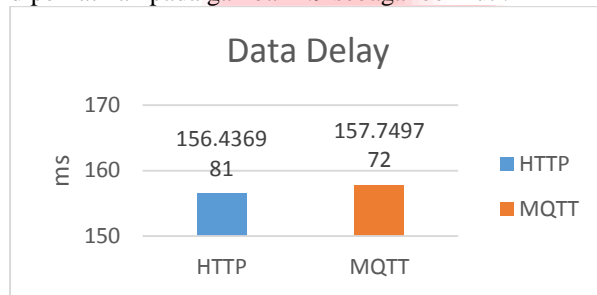
$$\begin{aligned}
 &= \frac{\text{Total variasi delay}}{\text{Total paket data} - 1} \\
 &= \frac{0.326611}{57 - 1} \\
 &= 0.0058323392 \text{ sec} \\
 &= 5.8323392 \text{ ms/paket}
 \end{aligned}$$

Total variasi *delay* dari proses penangkapan adalah 0.326611 sec atau sekitar 0.005443 menit, dan total paket yang diterima adalah 57 paket, nilai *jitter* yang didapatkan adalah 5.8323392 ms/paket. Hasil tersebut merupakan nilai bagus dengan

melihat standar indeks dari rujukan yang digunakan.

4.3 Perbandingan Analisis Data Quality of Service (QoS)

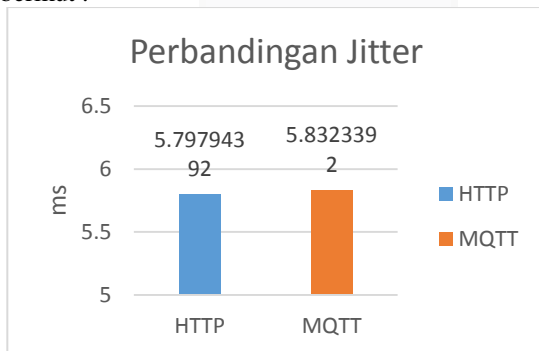
Perbandingan analisis data delay QoS dapat diperhatikan pada gambar 4.9 sebagai berikut :



Gambar 4.9 Data delay HTTP & MQTT

Pada gambar 4.9 rata-rata delay untuk protokol HTTP yang diperoleh termasuk indeks bagus yaitu 156.436981 ms, sedangkan rata-rata delay untuk protokol MQTT yang diperoleh termasuk indeks bagus yaitu 157.749772 ms.

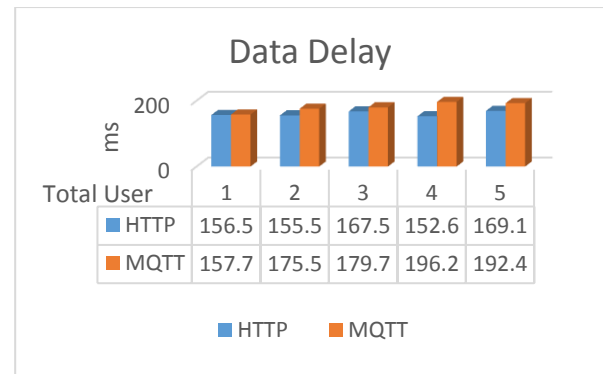
Untuk perbandingan analisis data delay QoS dapat diperhatikan pada gambar 4.10 sebagai berikut :



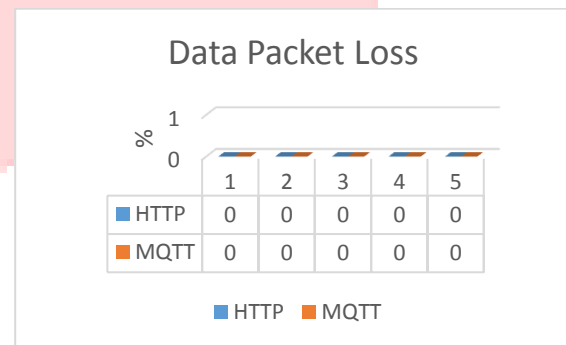
Gambar 4.10 Data Jitter HTTP & MQTT

Pada gambar 4.10 rata-rata jitter untuk protokol HTTP yang diperoleh termasuk indeks bagus yaitu 5.79794392 ms, sedangkan pada rata-rata jitter untuk protokol MQTT yang diperoleh termasuk indeks bagus yaitu 5.8323392 ms.

4.4 Perbandingan Analisis Data Delay, Packet Loss HTTP & MQTT dengan user yang berbeda



Gambar 4.11 Data Delay HTTP & MQTT dengan user yang berbeda



Gambar 4.12 Data Packet Loss HTTP & MQTT dengan user yang berbeda

5. Kesimpulan

Dari pengujian dan analisa Tugas Akhir dengan judul Analisis dan Implementasi Prototipe Pengatur Kelembaban Berbasis Internet of Things (IoT) pada Penyimpanan Sayur, penulis berhasil memperoleh hasil kesimpulan sebagai berikut :

- a. Parameter Delay dari seluruh perhitungan menunjukkan rata – rata nilai delay untuk protokol HTTP yang diperoleh termasuk indeks bagus yaitu 156.436981 ms, sedangkan rata-rata delay untuk protokol MQTT yang diperoleh termasuk indeks bagus yaitu 157.749772 ms.
- b. Nilai Delay dengan jumlah satu user hingga total lima user pada protokol HTTP didapat untuk satu user yaitu 156,453981 ms/paket, dua user yaitu 155,45289 ms/paket, tiga user yaitu 167,540147 ms/paket, empat user yaitu 152,620528 ms/paket dan lima user yaitu 169,067234 ms/paket. Dengan nilai delay tertinggi pada HTTP terdapat pada lima user yaitu 169,067234 ms/paket.
- c. Nilai Delay dengan jumlah satu user hingga total lima user pada protokol MQTT didapat untuk satu user yaitu 157,749772 ms/paket, dua user yaitu 175,500961 ms/paket, tiga user yaitu 179,725912 ms/paket, empat user yaitu

- 196,238905 ms/paket dan lima *user* yaitu 192,238905 ms/paket. Dengan nilai delay tertinggi pada MQTT terdapat pada empat *user* yaitu 192,424797 ms/paket.
- d. Parameter Jitter dari seluruh perhitungan menunjukkan rata – rata nilai jitter untuk protokol *HTTP* yang diperoleh termasuk indeks bagus yaitu 5.79794392 ms, sedangkan pada rata-rata *jitter* untuk protokol *MQTT* yang diperoleh termasuk indeks bagus yaitu 5.8323392 ms.
 - e. Parameter *Packet Loss* dari seluruh perhitungan menunjukkan nilai untuk protokol *MQTT* dan *HTTP* yang diperoleh adalah 0% termasuk indeks sangat bagus.
 - f. Nilai *Packet Loss* dengan jumlah satu *user* hingga total lima *user* yaitu 0% pada kedua protokol *HTTP* dan *MQTT*.
 - g. Dengan dua kali penyiraman dalam rentang waktu 2 jam kelembapan akan naik secara drastis mulai dari 70% hingga 95%.
 - h. Kelembapan naik secara cepat pada penyimpanan sayur setelah satu kali penyiraman dalam rentang waktu satu jam.
 - i. Pengujian alat ini mampu membuat kondisi Kubis dan Bayam tetap segar dalam waktu 4 hari.
 - j. Pengujian terhadap kondisi Kubis dan Bayam tanpa menggunakan alat yang sudah dirancang atau di tempat terbuka hanya bisa membuat sayur bayam bertahan selama 2 hari dan sayur kubis 3 hari.

DAFTAR PUSTAKA

- [1] Yuda Yudhanto, S.Kom.”Apa itu IoT (*Internet of things*)”. *Rumahstudio.com*. Indonesia, 2007.
- [2] Pengantar Mikrokontroler & Aplikasi Pada Arduino. “*Pengantar Mikrokontroler & Aplikasi Pada Arduino*”. Indonesia: Graha Ilmu, 2015.
- [3] Budiharto, D. W. *Perancangan dan Pemrograman Hasta Karya Robot*. Indonesia : Andi Publisher, 2014.
- [4] *Digital Library Unikom*. (n.d.). Diakses 20 Desember 2017, dari Knowlage Center : http://elib.unikom.ac.id/files/disk1/452/jbptunikomp-p-gdl-gelarumbar-22555-2-unikom_g-i.pdf
- [5] *Storage Guidelines For Fruits & Vegetables*. (2012, May 14). Diakses 20 Desember 2017, dari : <http://gardening.cce.cornell.edu/14> Mei 2012.
- [6] Putri Eka Pratiwi. *Analisis QoS pada Jaringan Multi Protokol Label Switching (MPLS) Studi Kasus Di Pelabuhan Indonesia III Cabang Tanjung Intan Cilacap*.
- [7] *Protokol MQTT*. Diakses 10 Mei 2018 dari : <https://medium.com/pemrograman/mengenal-mqtt-998b6271f585>
- [8] *Memahami Fungsi dan Cara Kerja HTTP*. Diakses 15 Mei 2018 dari : <https://www.teknologivirtual.com/2016/01/memahami-fungsi-dan-cara-kerja-http.html>.