

## IMPLEMENTASI PROTOKOL MQTT PADA APLIKASI SMART GARDEN BERBASIS IOT (INTERNET OF THINGS)

Muhamad Saiqul Umam, Suryo Adi Wibowo, Yosep Agus Pranoto

Program Studi Teknik Informatika S1, Fakultas Teknologi Industri  
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia  
1918025@scholar.itn.ac.id

### ABSTRAK

*IoT (Internet of Thing)* merupakan gaya baru dalam dunia teknologi yang memiliki potensi menjadi salah satu terobosan baru di masa depan. *IoT* merupakan sebuah teori yang bertujuan untuk memperlebar manfaat dari sebuah konektivitas jaringan internet yang tersambung secara terus menerus. Salah satu penerapan *IoT* yang bisa ditemui adalah *smart garden*. *Smart Garden* berasal dari bahasa Inggris yang artinya kebun atau taman pintar, yaitu sebuah rancangan sistem yang dibuat untuk memudahkan pekerjaan dalam hal perawatan taman. Pembuatan alat ini bertujuan untuk membantu proses monitoring kondisi taman. Sebagai sarana pengumpulan data implementasi *IoT* bisa dilakukan menggunakan beberapa protokol, diantaranya *HTTP*, *MQTT*, *CoAP*. *MQTT (Message Queuing Telemetry Transport)* memiliki beberapa keuntungan yaitu header yang lebih ringkas dan ringan sehingga menghemat sumber daya dan koneksi *IoT* dapat berjalan dengan baik. Penelitian ini dilakukan dengan cara membuat simulasi kebun yang akan dimonitor dan datanya akan dikirim pada broker *MQTT* yang kemudian dilakukan monitoring pada *web*. Dari hasil pengujian server *MQTT* yang dilakukan, dapat diambil kesimpulan bahwa server *MQTT* mampu menangkap data sensor yang dikirim dari mikrokontroler dengan baik dan akurat menggunakan bantuan jaringan *WiFi* sebagai media transmisi.

**Kata kunci :** *IoT, MQTT, Mikrokontroler, Protokol.*

### 1. PENDAHULUAN

*IoT (Internet of Thing)* adalah salah satu gaya baru dalam dunia teknologi yang memiliki potensi menjadi salah satu terobosan baru di masa depan. *IoT* merupakan sebuah teori yang bertujuan untuk memperlebar manfaat dari sebuah konektivitas jaringan internet yang tersambung secara terus menerus. *IoT* dapat menggabungkan antara benda-benda fisik dan virtual melalui eksploitasi data capture dan kemampuan berkomunikasi. Pendeknya dengan *IoT* benda-benda fisik di dunia nyata dapat berkomunikasi satu dengan yang lain dengan menggunakan bantuan jaringan dan internet [1].

Salah satu penerapan *IoT* yang bisa ditemui adalah *smart garden*. Yang dimaksud dari *smart garden* disini adalah sebuah taman terbuka seperti alun-alun kota, atau ruang terbuka hijau untuk publik yang memanfaatkan intervensi teknologi didalamnya. Salah satu sistem yang dapat diaplikasikan adalah monitoring dan kontrol. Dengan monitoring pengelola bisa melihat informasi keadaan aktual dari taman dalam kurun waktu tertentu yang dapat membantu pengelola untuk melakukan tindakan pemeliharaan yang diperlukan. Dengan data yang diperoleh juga memungkinkan adanya otomatisasi seperti sistem pengairan dan lampu taman sebagai upaya penghematan energi [2].

Kondisi udara juga penting dalam pengelolaan taman karena taman merupakan ruang terbuka publik dan pengunjung bisa mengetahui apakah udara yang ada pada area taman berada pada kondisi yang baik atau tidak. Semakin sedikit kadar *CO2* dan asap, maka kualitas udara semakin bagus untuk taman dan

pengunjung taman. Kondisi udara ini dapat dideteksi atau diukur menggunakan data yang telah diperoleh dari sensor dan dilakukan proses klasifikasi.

Protokol yang dapat diterapkan pada sebuah sistem *IoT* salah satunya adalah protokol *MQTT* yang memiliki kepanjangan *Message Queuing Telemetry Transport*. *MQTT* merupakan sebuah protokol jaringan *Machine to Machine (M2M)* ringan yang berbasis *publish-subscribe* yang dirancang untuk sebuah koneksi dengan keterbatasan spesifikasi mesin atau keterbatasan *bandwidth* jaringan sehingga bisa berjalan pada mesin berdaya rendah dan *traffic* data yang ringan. Protokol ini juga menerapkan komunikasi 2 arah sehingga komunikasi tidak hanya berfokus dari mesin pada jaringan saja namun juga berlaku sebaliknya [3].

Berdasarkan latar belakang diatas, penulis tertarik untuk melakukan sebuah penelitian dengan judul : “Implementasi Protokol *MQTT* Pada Aplikasi *Smart Garden* Berbasis *IoT (Internet Of Things)*”.

### 2. TINJAUAN PUSTAKA

Menurut Galih Yudha Saputra dkk dalam penelitiannya yang berjudul ” Penerapan Protokol *MQTT* Pada Teknologi WAN (Studi Kasus Sistem Parkir Universitas Brawijaya)” yang bertujuan untuk menerapkan protokol *MQTT* pada jaringan WAN menyimpulkan bahwa protokol *MQTT* dapat diimplementasikan pada teknologi WAN [4].

Menurut Muhammad Farid Ali Safii dkk dalam penelitiannya yang berjudul “Analisis *Quality Of Service* Protokol *MQTT* Dan *HTTP* Pada Penerapan Sistem Monitoring Suhu Berbasis *Nodemcu* (Studi

Kasus Ruang Server Kampus 3 Ist Akprind Yogyakarta)” menyimpulkan bahwa protokol *MQTT* membutuhkan bandwidth jaringan yang lebih daripada *HTTP*[5].

Menurut Chrisyantar dalam penelitiannya yang berjudul “Implementasi Konsep *Internet Of Things* Pada Sistem Monitoring Banjir Menggunakan Protokol *MQTT*” Menyimpulkan bahwa konsep *Internet of Things* dengan menggunakan protokol *MQTT* dapat diterapkan pada sistem monitoring banjir yang dibuat sesuai dengan perancangan yang telah dibuat[6].

### 3. METODE PENELITIAN

#### 3.1. Kebutuhan Fungsional

Tabel 1. Kebutuhan Fungsional

| No | Fitur                     | Fungsional                                                                                                                         |
|----|---------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Monitoring                | Menampilkan data dari semua sensor dan status aktuator yang didapat pada sebuah <i>dashboard</i> yang bisa dipahami oleh pengguna. |
| 2. | Kontrol                   | Mengontrol alat dari jarak jauh menggunakan button yang ditampilkan pada <i>dashboard</i> yang bisa diakses oleh pengguna.         |
| 3. | Klasifikasi Kondisi Udara | Sistem mampu membuat klasifikasi kondisi udara menggunakan metode <i>fuzzy</i> .                                                   |
| 4. | Notifikasi                | Sistem akan memunculkan notifikasi jika data pada sensor menunjukkan kondisi abnormal.                                             |

Pada tabel 1 dijelaskan bahwa aplikasi akan mempunyai 3 fungsi atau fitur utama yang akan didapatkan oleh pengguna yaitu monitoring, kontrol, dan klasifikasi. fitur monitoring memiliki fungsional menampilkan data dari semua sensor dan status aktuator yang didapat pada sebuah *user interface* berupa *web based dashboard*. Lalu fungsi kontrol memiliki fungsional mengontrol alat dari jarak jauh menggunakan button yang ditampilkan pada *dashboard* yang bisa diakses oleh pengguna. Sistem

juga akan melakukan klasifikasi kondisi udara berdasarkan nilai dari sensor yang akan ditampilkan pada *dashboard* monitoring pada *user*.

#### 3.2. Kebutuhan Nonfungsional

Kebutuhan sistem yang akan dirancang memiliki kebutuhan sebagai berikut :

1. Aplikasi yang dikembangkan harus bisa diakses kapanpun dan dimanapun.
2. Aplikasi yang dikembangkan hanya bisa berjalan jika ada koneksi internet.
3. Aplikasi yang dikembangkan hanya bisa diakses lewat browser dikarenakan memiliki *web based user interface*.

#### 3.3. Kebutuhan Sensor pada sistem

##### 1. DHT11

Sensor ini digunakan untuk mendeteksi tingkat kelembapan udara (*humidity*) dan suhu pada udara sekitar (*temperature*). suhu yang dikeluarkan dapat berupa satuan *fahrenheit* ataupun *celcius*.

##### 2. Soil Moisture

Sensor ini digunakan untuk mendeteksi tingkat kelembapan pada tanah. Nilai dari sensor akan dikeluarkan dengan satuan analog dengan range nilai dari 0 hingga 1023. Tanah yang lembab akan menghasilkan nilai output yang semakin tinggi.

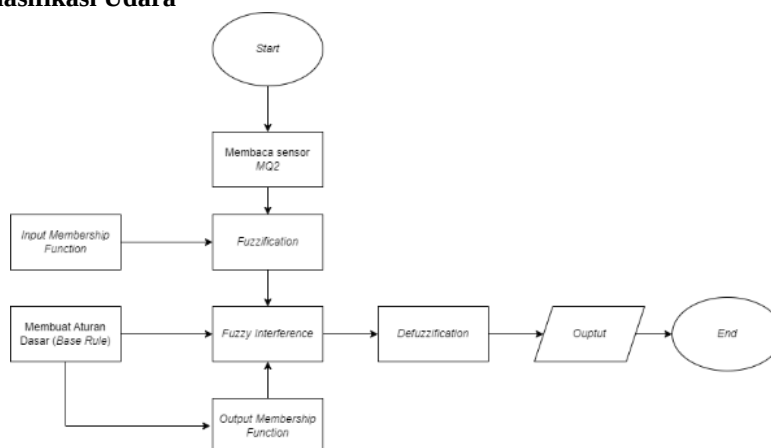
##### 3. MQ2

Sensor ini digunakan untuk mendeteksi kadar suatu zat yang berada pada udara disekitar. *Output* yang dihasilkan dari sensor ini adalah kadar karbon dioksida (*CO2*) dan kadar asap. Nilai yang dikeluarkan memiliki satuan *ppm* atau *parts per million*.

##### 4. PIR

Sensor ini digunakan untuk mendeteksi suatu objek pada jangkauan dari sensor. Output yang dihasilkan oleh sensor ini bisa berupa nilai *boolean*.

#### 3.4. Flowchart Klasifikasi Udara

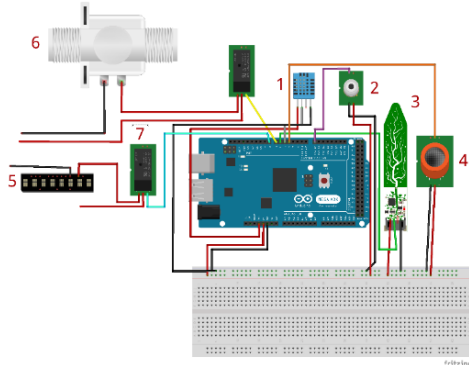


Gambar 1. Flowchart klasifikasi udara

Pada gambar 1 dijelaskan bahwa nilai sensor gas MQ2 akan melewati serangkaian proses *fuzzy* sehingga bisa mengeluarkan klasifikasi kondisi udara. Proses pertama yaitu memasukkan *membership function* sehingga terbuatlah variabel yang terbentuk pada parameter-parameter yang sudah ditentukan pada fungsi. Lalu ada aturan dasar atau *base rule* yang digunakan untuk menentukan keluaran berupa output linguistik yang didapat berdasarkan parameter dari fungsi member yang sudah dibuat.

### 3.5. Desain dan tabel wiring

Gambar dari desain wiring sistem yang akan digunakan ditunjukkan pada gambar 2.



Gambar 2. Desain wiring

Pada penelitian ini akan berfokus pada bagaimana data tersebut akan dapat ditransmisikan kepada broker *MQTT* sehingga model kebun dan ukurannya tidak memiliki ukuran spesifik. Gambar dari contoh alat yang akan digunakan ditunjukkan pada gambar 3 sesuai dengan desain yang ada pada gambar 2.

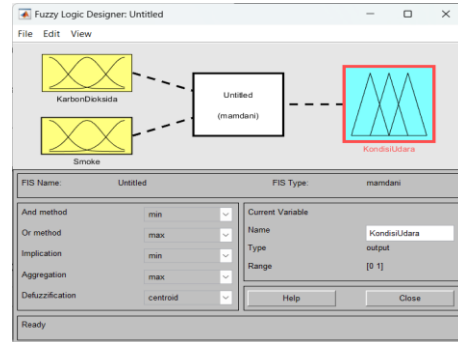


Gambar 3. Model miniatur kebun.

### 3.6. Fuzzifikasi

Proses pengklasifikasian kondisi udara dilakukan berdasarkan dari 2 variabel data yang diperoleh dari sensor gas MQ2 yaitu menggunakan data kadar karbon dioksida ( $CO_2$ ) dan kadar asap pada udara menggunakan metode *fuzzy mamdani*. Pembuatan *membership function* dan *base rule* sesuai dari nilai 2 variabel diatas yang dibuat menggunakan *fuzzy logic toolbox* adalah sebagai berikut :

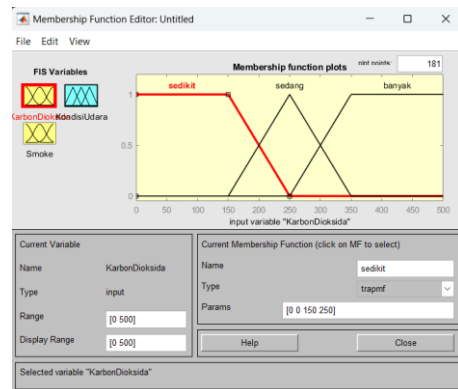
A. Menentukan nilai input dan parameter untuk kadar karbon dioksida dan kadar asap dalam udara.



Gambar 4. Tampilan FIS editor

#### B. Kadar karbon dioksida ( $CO_2$ )

1. Range nilai dari 0-500
2. Sedikit dengan parameter nilai 0-250 [0 0 150 250].
3. Sedang dengan parameter nilai 150-350 [150 250 350].
4. Banyak dengan parameter nilai 250-500 [250 350 500 500].



Gambar 5. tampilan *membership function editor* variabel karbon dioksida

Fungsi keanggotaannya adalah sebagai berikut:

#### a. Sedikit

$$\begin{aligned} \mu_{\text{sedikit}}[x] &= 1; & x \leq 150 \\ \mu_{\text{sedikit}}[x] &= \frac{250-x}{250-150}; & 150 \leq x \leq 250 \\ \mu_{\text{sedikit}}[x] &= 0; & x \geq 250 \end{aligned}$$

#### b. Sedang

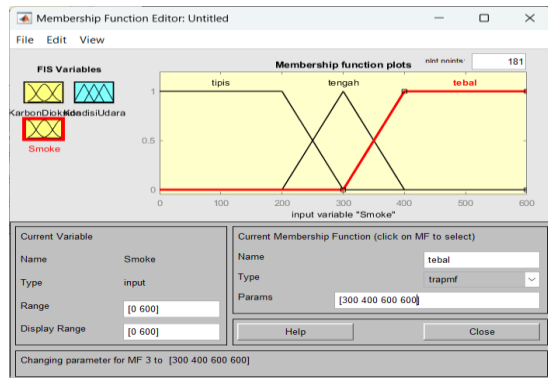
$$\begin{aligned} \mu_{\text{sedang}}[x] &= 0; & x \leq 150 \\ \mu_{\text{sedang}}[x] &= \frac{x-150}{250-150}; & 150 \leq x \leq 250 \\ \mu_{\text{sedang}}[x] &= \frac{350-x}{350-250}; & 250 \leq x \leq 350 \\ \mu_{\text{sedang}}[x] &= 0 & x \geq 350 \end{aligned}$$

#### c. Banyak

$$\begin{aligned} \mu_{\text{banyak}}[x] &= 0; & x \leq 250 \\ \mu_{\text{banyak}}[x] &= \frac{x-250}{350-250} & 250 \leq x \leq 350 \\ \mu_{\text{banyak}}[x] &= 1 & x \geq 350 \end{aligned}$$

## C. Kadar asap

1. Range nilai dari 0-600
2. Tipis dengan parameter nilai 0-200 [0 0 200 300].
3. Tengah dengan parameter nilai 200-400 [200 300 400].
4. Tebal dengan parameter nilai 300-600 [300 400 600 600].

Gambar 6. tampilan *membership function editor* variabel kadar asap

Fungsi keanggotaannya adalah sebagai berikut:

## a. Tipis

$$\begin{aligned} \mu_{\text{tipis}}[y] &= 1; & x &\leq 200 \\ \mu_{\text{tipis}}[y] &= \frac{300-x}{300-200}; & 200 &\leq x \leq 300 \\ \mu_{\text{tipis}}[y] &= 0; & x &\geq 300 \end{aligned}$$

## b. Tengah

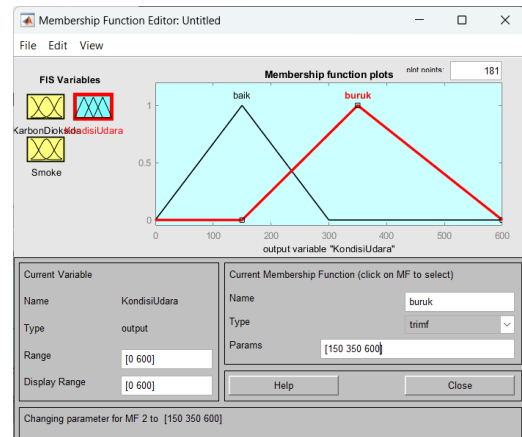
$$\begin{aligned} \mu_{\text{tengah}}[y] &= 0; & x &\leq 200 \\ \mu_{\text{tengah}}[y] &= \frac{x-200}{300-200}; & 200 &\leq x \leq 300 \\ \mu_{\text{tengah}}[y] &= \frac{400-x}{400-300}; & 300 &\leq x \leq 400 \\ \mu_{\text{tengah}}[y] &= 0 & x &\geq 400 \end{aligned}$$

## c. Tebal

$$\begin{aligned} \mu_{\text{tebal}}[y] &= 0; & x &\leq 300 \\ \mu_{\text{tebal}}[y] &= \frac{x-300}{400-300}; & 300 &\leq x \leq 400 \\ \mu_{\text{tebal}}[y] &= 1 & x &\geq 400 \end{aligned}$$

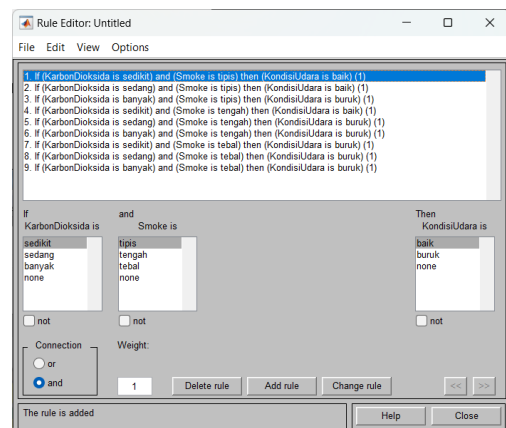
## D. Kondisi Udara

1. Baik dengan parameter [0 150 300].
2. Buruk dengan parameter [150 350 600].

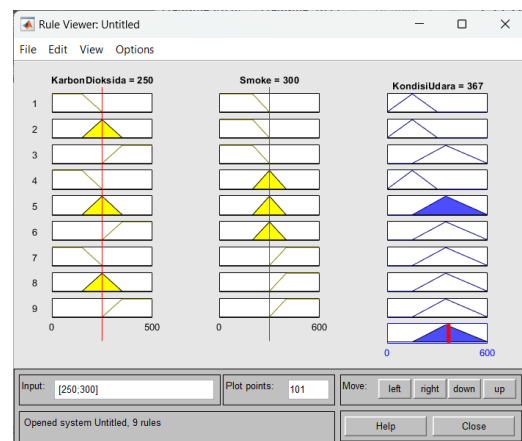
Gambar 7. tampilan *membership function editor* variabel kadar asap

## E. Base rule

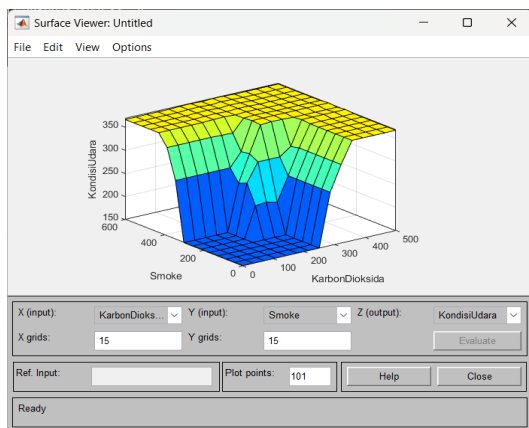
Setelah membuat *membership function*, pada tiap variabel langkah selanjutnya adalah membuat *base rule* berdasarkan variabel yang ada.

Gambar 8. tampilan membuat *base rule*

Hasil dari *rule* yang sudah dibuat berdasarkan *membership function* bisa dilihat pada gambar dibawah :

Gambar 9. tampilan *rule viewer*

Hasil tampilan grafik 3D antara variabel *input* dan *output* adalah sebagai berikut :



Gambar 10. Tampilan *surface viewer*

#### 4. HASIL DAN PEMBAHASAN

##### 4.1. Implementasi protokol MQTT

Langkah pertama yang harus dilakukan adalah melakukan instalasi broker pada sistem lokal, dimulai dengan mengunduh *installer* dari *Mosquitto MQTT* lalu dilakukan proses instalasi. Jika instalasi sudah berhasil maka perlu penyesuaian port *websocket* yang nantinya akan dipakai untuk proses *publish* / *subscribe* dengan mikrokontroler. Penyesuaian bisa dilakukan dengan membuka file *mosquitto.conf* pada folder instalasi dan menambahkan kode sebagai berikut :

```
mosquitto.conf X
C:\Program Files\mosquitto> mosquitto.conf
247 #socket_domain
248
249 port 1883
250 listener 9001
251 protocol websockets
252 socket_domain ipv4
253 allow_anonymous true
254
255 # Bind the listener to a specific interface. This is similar to
256 # the [ip address/host name] part of the listener definition, but is useful
257 # when an interface has multiple addresses or the address may change. If used
258 # with the [ip address/host name] part of the listener definition, then the
259 # bind_interface option will take priority.
260 # Not available on Windows.
261 #
262 # Example: bind interface eth0
263 #bind_interface
```

Gambar 11. penyesuaian *port websocket*

Lalu pengecekan bisa dilanjutkan dengan membuka *command prompt* dengan akses administrator dan mengetikkan perintah *net start mosquitto* lalu akan muncul balasan seperti dibawah.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.675]
(c) Microsoft Corporation. All rights reserved.

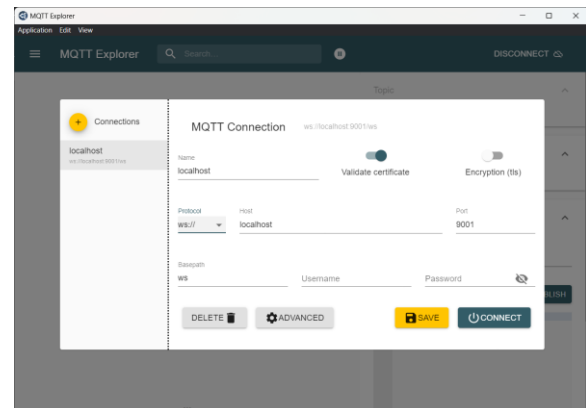
C:\Windows\System32>net start mosquitto

The Mosquitto Broker service was started successfully.

C:\Windows\System32>
```

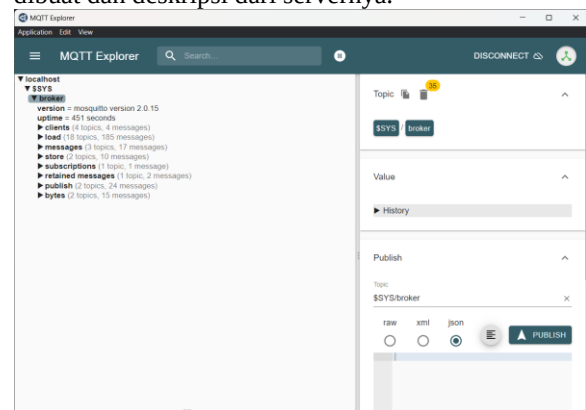
Gambar 12. Memulai layanan

Selanjutnya adalah pengujian koneksi dari *MQTT*. Pengujian bisa dilakukan menggunakan bantuan software *MQTT Explorer* dengan menggunakan akses seperti gambar dibawah.



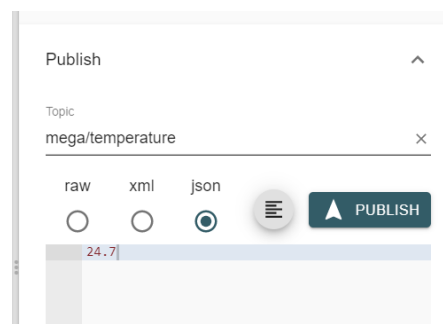
Gambar 13. melakukan konfigurasi koneksi

Jika sudah berhasil makan program akan menampilkan koneksi *MQTT* yang sudah berhasil dibuat dan deskripsi dari servernya.



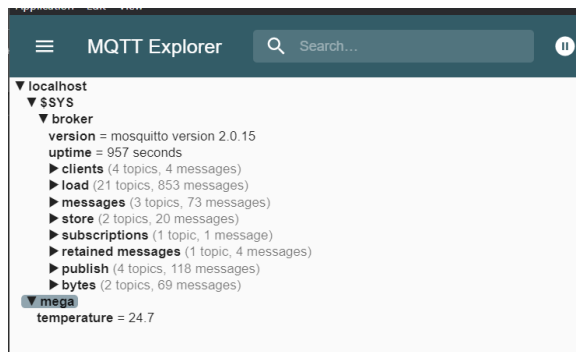
Gambar 14. tampilan koneksi *MQTT*

Langkah selanjutnya adalah pengujian *topic* dan *publish*. Pada jaringan *MQTT* pesan akan dikirimkan kepada sebuah topik tertentu menggunakan mode *publish*. Untuk melakukan pengujian fungsi *publish* bisa menggunakan bantuan software *MQTT Explorer* dengan cara membuat *publish* kepada topik dengan nama seperti pada gambar dibawah. Jika sudah sesuai maka bisa dilakukan pengujian dengan menekan tombol *publish*.



Gambar 15. melakukan *publish* pada *topic*

Hasil dari data yang akan *publish* akan muncul sesuai dengan topik tujuan dan pesan terakhir yang diterima oleh server.



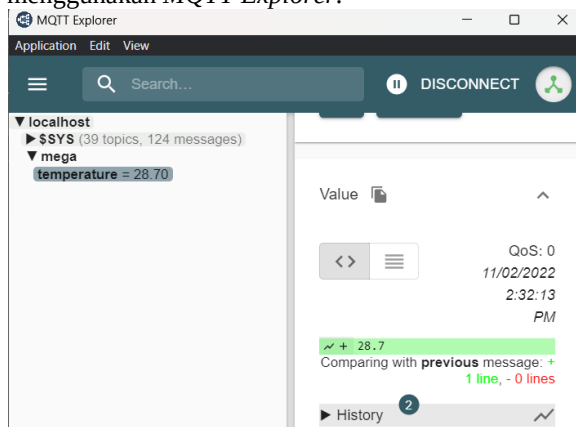
Gambar 16. hasil pengujian *publish message*

Pengujian selanjutnya adalah *publish* topik menggunakan mikrokontroler. Pada pengujian ini, perlu dilakukan proses *upload sketch* pada mikrokontroler melalui *Arduino IDE*. langkah pertama adalah membuka *Arduino IDE* lalu memasukkan *source code* kedalam *sketch*.



Gambar 17. memasukkan *code* kedalam *sketch*

Setelah itu lakukan proses *compile source code* dan menunggu hingga mikrokontroler terhubung dengan server. Dan hasil *publish* dari mikrokontroler akan masuk kedalam server dan bisa dilihat menggunakan *MQTT Explorer*.



Gambar 18. hasil *publish* dari mikrokontroler

Dari hasil pengujian server *MQTT* yang dilakukan, dapat diambil kesimpulan bahwa server *MQTT* mampu menangkap data sensor yang dikirim dari mikrokontroler dengan baik dan akurat menggunakan bantuan jaringan *WiFi* sebagai media transmisi.

#### 4.2. Pengujian hardware

Pengujian hardware yang akan dilakukan menggunakan sensor *DHT11*, *Soil Moisture*, dan *MQ2*. Proses pertama yang dilakukan adalah *upload sketch* pada mikrokontroler melalui *Arduino IDE*. langkah pertama adalah membuka *Arduino IDE* lalu memasukkan *source code* kedalam *sketch*.



Gambar 19. memasukkan *code* kedalam *sketch*

Rangkaian alat yang digunakan pada waktu pengetesan adalah sebagai berikut :



Gambar 20. rangkaian pengujian hardware

Setelah itu ada lakukan proses *compile* dan tunggu hingga output sensor tampil pada *serial monitor*.

##### a. Hasil pengujian *DHT11*

Pengujian ini bertujuan untuk mendeteksi suhu menggunakan *DHT11* dengan satuan *celcius* melihat apakah sensor bisa mendeteksi perubahan lingkungan yang terjadi.

Tabel 2. Tabel pengujian suhu *DHT11*

| No. | Nilai (°C) | Termometer | Error |
|-----|------------|------------|-------|
| 1.  | 25.3       | 25.2       | 0.4%  |
| 2.  | 25.3       | 25.2       | 0.4%  |
| 3.  | 25.5       | 25.3       | 0.8%  |



| No. | Nilai (°C) | Termometer | Error |
|-----|------------|------------|-------|
| 4.  | 26.4       | 26.5       | 0.3%  |
| 5.  | 27.2       | 27         | 0,7   |
| 6.  | 28.1       | 28.7       | 2%    |
| 7.  | 28.5       | 28.3       | 0,7   |

Pengujian selanjutnya bertujuan untuk mendeteksi tingkat kelembaban udara menggunakan *DHT11* dan melihat apakah sensor bisa mendeteksi perubahan lingkungan yang terjadi.

Tabel 3. Tabel pengujian kelembapan *DHT11*

| No. | Nilai (%) | Termometer | Error |
|-----|-----------|------------|-------|
| 1.  | 26        | 25         | 4%    |
| 2.  | 28        | 26         | 7,6%  |
| 3.  | 34        | 32         | 6,2%  |
| 4.  | 35        | 33         | 6%    |
| 5.  | 46        | 43         | 6,9%  |

b. Hasil pengujian *Soil Moisture*

Pengujian ini bertujuan untuk mendeteksi tingkat kelembaban tanah menggunakan sensor *Soil Moisture* dan melihat apakah sensor bisa mendeteksi perubahan lingkungan yang terjadi.

Tabel 4. Tabel pengujian *Soil Moisture* Sensor

| No. | Nilai (%) | Soil Sensor | Error |
|-----|-----------|-------------|-------|
| 1.  | 33        | 31          | 6.4%  |
| 2.  | 35        | 34          | 2.9%  |
| 3.  | 40        | 40          | 0%    |
| 4.  | 43        | 42          | 2.3%  |
| 5.  | 44        | 43          | 2.3%  |

c. Hasil pengujian *MQ2*

Pengujian ini bertujuan untuk mendeteksi tingkat karbon dioksida pada udara menggunakan sensor *MQ2* dan melihat apakah sensor bisa mendeteksi perubahan lingkungan yang terjadi.

Tabel 5. Tabel pengujian  $CO^2$ 

| No | Nilai | Kondisi                | Hasil    |
|----|-------|------------------------|----------|
| 1. | 270   | Udara ruangan          | Berhasil |
| 2. | 282   | Udara ruangan          | Berhasil |
| 3. | 285   | Udara ruangan          | Berhasil |
| 4. | 357   | Terdapat sumber $CO_2$ | Berhasil |
| 5. | 360   | Terdapat sumber $CO_2$ | Berhasil |

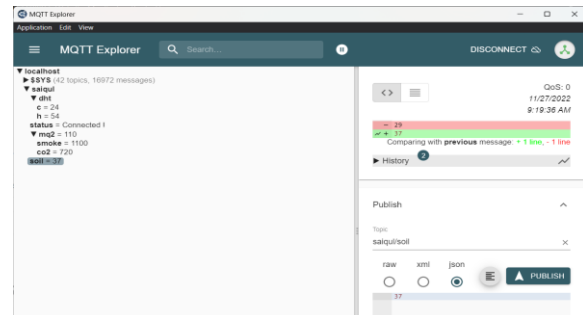
Pengujian selanjutnya bertujuan untuk mendeteksi kadar asap pada udara menggunakan *MQ2* dan melihat apakah sensor bisa mendeteksi perubahan lingkungan yang terjadi.

Tabel 6. Tabel pengujian kadar asap

| No | Nilai | Keterangan           | Hasil    |
|----|-------|----------------------|----------|
| 1. | 0     | Udara normal         | Berhasil |
| 2. | 0     | Udara normal         | Berhasil |
| 3. | 0     | Udara normal         | Berhasil |
| 4. | 150   | Terdapat sumber asap | Berhasil |
| 5. | 223   | Terdapat sumber asap | Berhasil |

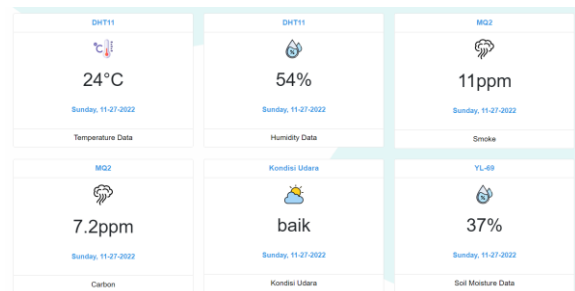
#### 4.3. Pengujian Web Dashboard

Pengujian ini dilakukan untuk melihat apakah data sensor yang dikirim oleh mikrokontroler pada server *MQTT* bisa ditampilkan dan data sensor tersebut disimpan pada database. pengujian dilakukan dengan cara input data sensor secara manual melalui aplikasi *MQTT Explorer* sesuai dengan topik yang *unsubscribe* oleh aplikasi.



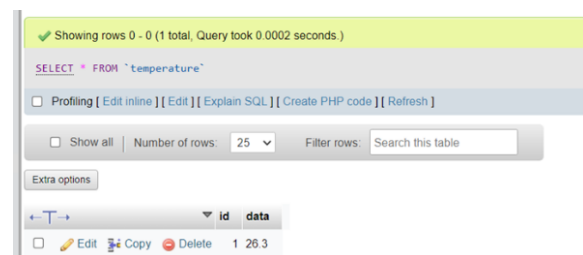
Gambar 21. Penginputan data manual

Hasil data yang dimasukkan akan tampil pada halaman *dashboard* dan data tersebut akan dimasukkan kedalam database.



Gambar 22. tampilan dashboard

Data yang ditangkap oleh *dashboard* akan diinputkan kedalam database secara otomatis



Gambar 23. tampilan database

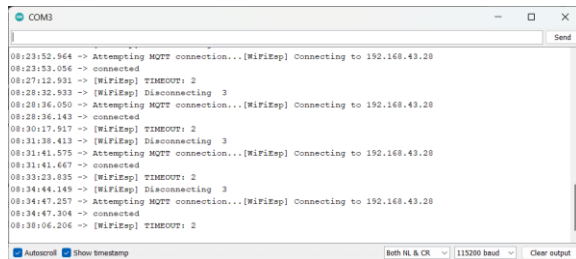
#### 4.4. Pengujian delay

Pengujian ini dilakukan untuk mencegah koneksi *MQTT* yang dijalankan oleh mikrokontroler tidak terjadi *timeout* sehingga fungsi *callback* dari proses *subscribe* pada topik yang tersambung tidak terputus. Delay dilakukan pada saat proses *client* terjadi proses *loop*. Pengujian dilakukan dengan cara melakukan *publish* kepada topik yang *unsubscribe* pada *client* secara terus menerus selama 1 menit.

Tabel 7. Tabel pengujian *delay*

| No. | Nilai (ms) | Keterangan   |
|-----|------------|--------------|
| 1.  | 1000       | Timeout [22] |
| 2.  | 3000       | Timeout [21] |
| 3.  | 5000       | Timeout [5]  |
| 4.  | 7000       | Timeout [4]  |
| 5   | 8000       | Timeout[2]   |

Dari hasil diatas diperoleh bahwa *delay* dengan nilai 8000ms memiliki *timeout* paling kecil dengan jumlah *buffer overflow* sebanyak 2 data.

Gambar 24. hasil pengujian dengan *delay*

## 5. KESIMPULAN DAN SARAN

Dengan melihat hasil penelitian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut. Dari hasil pengujian server *MQTT* yang dilakukan, dapat diambil kesimpulan bahwa server *MQTT* mampu menangkap data sensor yang dikirim dari mikrokontroler dengan baik dan akurat menggunakan bantuan jaringan *WiFi* sebagai media transmisi. Pada pengujian sensor *DHT11* yang dilakukan didapat tingkat error terbesar terjadi pada saat pembacaan suhu pada nilai 28.1 *celcius* dengan nilai error sebesar 2% dan dengan rata-rata total error sebesar 0.75%. Pada pengujian sensor *DHT11* yang dilakukan didapat tingkat error terbesar terjadi pada saat pembacaan kelembapan pada nilai 28% dengan nilai error sebesar 7.6% dan dengan rata-rata total error sebesar 6.14%. Pada pengujian sensor *Soil Moisture* yang dilakukan didapat tingkat error terbesar terjadi pada pembacaan sensor dengan nilai kelembapan tanah sebesar 33% dengan tingkat error 6.4% dan dengan rata-rata error sebesar 2.78%. Dari hasil

pengujian *delay* diperoleh bahwa *timeout* terbesar yang didapatkan oleh sistem terjadi pada *delay* sebesar 1 detik dengan jumlah *timeout* sebesar 22 dan *timeout* terkecil didapatkan pada *delay* 8 detik dengan nilai *timeout* paling kecil dengan jumlah *buffer overflow* sebanyak 2 data. Berdasarkan penelitian skripsi yang dilakukan penulis, maka penulis berharap penerapan protokol *MQTT* dapat dilakukan dengan cara yang lebih efisien dengan *hardware* yang lebih mumpuni sehingga kualitas sistem yang diciptakan akan jadi lebih baik dan memiliki hasil maksimal.

## DAFTAR PUSTAKA

- [1] Sasmito, G.W. and Wijayanto, S., 2020. Studi pengenalan internet of things bagi guru dan siswa SMK Bina Nusa Slawi sebagai wawasan salah satu ciri revolusi industri 4.0. *Dinamisia: Jurnal Pengabdian Kepada Masyarakat*, 4(1), pp.186-194.
- [2] Agustini, M., Riswanda, R. and Fuad, A., 2017. *Strategi pemerintah kota cilegon menuju cilegon smart city* (Doctoral dissertation, Universitas Sultan Ageng Tirtayasa).
- [3] Sahifa, A.A., Setiawan, R. and Yazid, M., 2021. Pengiriman Data Berbasis Internet of Things untuk Monitoring Sistem Hemodialisis Secara Jarak Jauh. *Jurnal Teknik ITS*, 9(2), pp.A209-A214.
- [4] Saputra, G.Y., Afrizal, A.D., Mahfud, F.K.R., Pribadi, F.A. and Pamungkas, F.J., 2017. Penerapan Protokol MQTT Pada Teknologi Wan (Studi Kasus Sistem Parkir Universitas Brawijaya).
- [5] Safii, M.F.A., Raharjo, S. and Lestari, U., 2019. Analisis Quality of Service protokol MQTT dan HTTP pada penerapan sistem monitoring suhu berbasis NodeMCU (studi kasus ruang server kampus 3 IST Akprind Yogyakarta). *Jurnal Jarkom*, 7(1), pp.11-19.
- [6] Chrisyantar, H., 2018. *Implementasi Konsep Internet of Things pada Sistem Monitoring Banjir menggunakan Protokol MQTT* (Doctoral dissertation, Universitas Brawijaya).