

BAB 6 PENGUJIAN

Bab ini membahas proses pengujian serta menganalisis hasil dari pengujian yang dilakukan berdasarkan sistem yang telah dibuat. Adapun tujuan dilakukannya pengujian adalah untuk mengetahui apakah semua kebutuhan yang diharapkan telah terpenuhi oleh sistem. Proses pengujian yang dilakukan yakni berupa pengujian *discovery*, pengujian *delay* pengiriman data dari arduino kedalam MQTT-Broker, pengujian *delay* pengiriman data dari MQTT-Broker kedalam basis data yang kosong, dan pengujian *delay* pengiriman data dari MQTT-Broker kedalam basis data yang sudah ada tabelnya sebelumnya.

6.1 Pengujian Fase Discovery MQTT Client

Pengujian dilakukan dengan mengaktifkan 4 *node* sebagai klien yang terhubung melalui jaringan *wifi* dengan *gateway*. Pada proses ini terdapat fase *discovery*. Fase ini menggunakan jarak sebagai ukuran dalam keberhasilan koneksi antara mikrokontroler dengan MQTT.

6.1.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat akurasi NodeMCU dalam keberhasilannya melakukan koneksi ke jaringan *wifi* dan kedalam MQTT.

6.1.2 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk menguji fase *discovery* MQTT-Client.

1. Menghubungkan mikrokontroler NodeMCU ESP8266 12E dengan komputer menggunakan kabel USB pada port yang berbeda.
2. Melakukan unggah program dari Arduino ke mikrokontroler NodeMCU.
3. Percobaan akan dilakukan lima kali, dimana program dari Arduino akan diunggah sebanyak lima kali kedalam masing-masing NodeMCU.
4. Mengamati hasil pengujian fase *discovery* dan mencatat hasil keberhasilan untuk mencari akurasi.

Adapun cara untuk mengukur persentase error yaitu dengan menggunakan persamaan berikut:

$$\text{Persentase Error Percobaan Ke - n} = \frac{\text{Total Selisih Keberhasilan Koneksi}}{\text{Total NodeMCU}} \times 100\%$$

$$\text{Rata - Rata Persentase Error Percobaan} = \frac{\sum \text{Persentase Percobaan ke - n}}{\text{Total Pengujian}}$$

$$\text{Akurasi} = 100\% - (\text{Rata - rata Persentase Error Percobaan})$$

Gambar 6. 1. Persamaan Akurasi Fase Discovery

6.1.3 Hasil dan Analisis Pengujian

Hasil pengujian fase *discovery* ditunjukkan pada tabel 6.1 dibawah. Berdasarkan tabel tersebut terlihat bahwa semua NodeMCU dapat terkoneksi kedalam *wifi* dan mendapatkan alamat IP masing-masing dan dapat terkoneksi

```

COM4
Connecting to WiFi
.
.
Connected to WiFi
IP: 192.168.0.103
Connecting
MQTT - Connect, clean-session: 1, ts: 1481
MQTT - Wait for message, type: 2, tm: 9993 ms
MQTT - Process message, type: 2
MQTT - Connect ack received
MQTT - Connect ack, code: 0
MQTT - Keepalive interval: 12 sec
MQTT - Session is not present => reset subscription
23
MQTT - Publish, to: test/TEST-ID/pub, size: 3
23
  
```

dengan protokol MQTT.

Gambar 6. 2. Fase NodeMCU melakukan koneksi ke *wifi* dan MQTT

Serial monitor menampilkan keberhasilan hubungan NodeMCU dengan *wifi*, juga data sensor yang dikirim. Pada kolom status keberhasilan terdapat tanda (v) yang menyatakan percobaan fase *discovery* berhasil dilakukan.

Tabel 6. 1. Fase Discovery

Percobaan Ke-	Jarak (m)	NodeMCU ke-	Alamat IP	Status keberhasilan
1	4	1	192.168.0.103	V
		2	192.168.0.105	V
		3	192.168.0.107	V
		4	192.168.0.108	V
2	8	1	192.168.0.103	V

		2	192.168.0.105	V
		3	192.168.0.107	V
		4	192.168.0.108	V
3	12	1	192.168.0.103	V
		2	192.168.0.105	V
		3	192.168.0.107	V
		4	192.168.0.108	V
4	16	1	192.168.0.103	X
		2	192.168.0.105	V
		3	192.168.0.107	V
		4	192.168.0.108	V
5	20	1	192.168.0.103	X
		2	192.168.0.105	X
		3	192.168.0.107	X
		4	192.168.0.108	X

Berdasarkan persamaan dari hasil pengujian fase *discovery* lalu nilainya dimasukkan kedalam persamaan akan didapatkan perhitungan seperti berikut :

$$\text{Persentase Error ke - 1} = \frac{4 - 4}{4} \times 100\% = 0\%$$

$$\text{Persentase Error ke - 2} = \frac{4 - 4}{4} \times 100\% = 0\%$$

$$\text{Persentase Error ke - 3} = \frac{4 - 4}{4} \times 100\% = 0\%$$

$$\text{Persentase Error ke - 4} = \frac{4 - 3}{4} \times 100\% = 25\%$$

$$\text{Persentase Error ke - 5} = \frac{4 - 0}{4} \times 100\% = 100\%$$

$$\text{Rata - rata Error} = \frac{0\% + 0\% + 0\% + 25\% + 100\%}{5} = 25\%$$

$$\text{Akurasi} = 100\% - 25\% = 75\%$$

Dari hasil pengujian yang dilakukan sebanyak 5 kali percobaan dengan akurasi sebesar 75% menggunakan rentan jarak dari 4 meter sampai dengan 20 meter, maka dapat disimpulkan bahwa fase *discovery* untuk konektivitas NodeMCU ke jaringan *wifi* dapat dilakukan dengan *reliable* sampai dengan jarak 12 meter. Saat NodeMCU berada pada jarak 16 meter dari *router* terdapat kemungkinan terjadi terputusnya jaringan.

6.2 Pencatatan Waktu Pengiriman Data dari Arduino Ke Broker

Pengujian dilakukan dengan mengaktifkan 1, 2, 3, dan 4 *node* sebagai *publisher* yang terhubung melalui jaringan *wifi* dengan *gateway*. Pada proses ini terdapat waktu pengiriman atau *delay* dari sistem. Delay inilah yang akan digunakan sebagai pengujian.

6.2.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui lama waktu yang dibutuhkan oleh mikrokontroler NodeMCU untuk mengirim data asmpai panjang data sudah sampai N-data didalam MQTT-Broker. Lama waktu ini yang digunakan sebagai informasi pengujian.

6.2.2 Prosedur Pengujian

Berikut prosedur pengujian yang digunakan untuk menguji sistem :

1. Menjalankan program pengiriman data mikrokontroler ke MQTT-Broker dari arduino.
2. Menjalankan program *stack* data MQTT-Broker.
3. Mencatat hasil waktu pengiriman data dari arduino untuk dijadikan *stack* di MQTT-Broker.

6.2.3 Hasil dan Analisis Pengujian

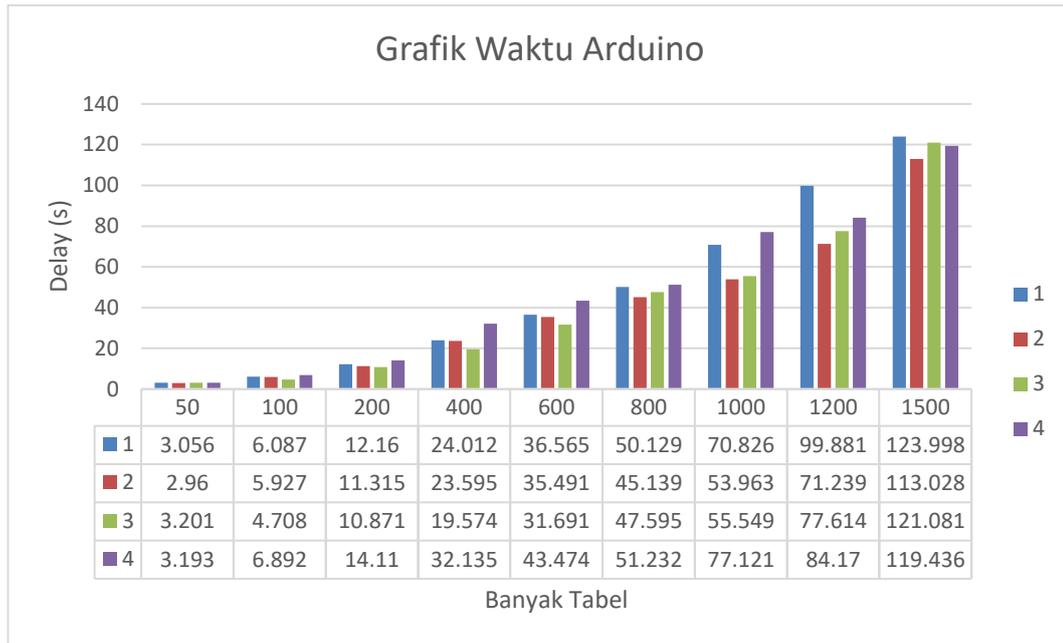
Hasil pengujian disimpan pada tabel 6.2.

Tabel 6. 2. Waktu Pengiriman Data dari Arduino ke Broker

Banyak Node	Banyak Data	Delay (S)
1	50	3.056
	100	6.087
	200	12.16
	400	24.012
	600	36.565
	800	50.129
	1000	70.826
	1200	99.881
	1500	123.9982
2	50	2.96
	100	5.927

Banyak Node	Banyak Data	Delay (S)
	200	11.315
	400	23.595
	600	35.491
	800	45.139
	1000	53.963
	1200	71.239
	1500	113.028
3	50	3.201
	100	4.708
	200	10.871
	400	19.574
	600	31.691
	800	47.595
	1000	55.549
	1200	77.614
	1500	121.081
4	50	3.193
	100	6.892
	200	14.11
	400	32.135
	600	43.474
	800	51.232
	1000	77.121
	1200	84.17
	1500	119.436

Hasil pengujian diatas memperlihatkan bahwa jumlah banyak node yang melakukan pengiriman data ke MQTT-Broker terhadap *delay* tidak membuat perubahan yang signifikan, hal ini dapat disebabkan karena *router wifi* melakukan *drop packet* yang dikirimkan oleh node, atau dapat disebabkan karena MQTT-Broker sendiri mempunyai kapasitas maksimal penerimaan data tiap satuan waktu.



Gambar 6. 3. Grafik Waktu MQTT-Broker melakukan Stack Data dari Arduino

6.3 Pencatatan Waktu Pengiriman Data dari Broker Ke MongoDB (Database Kosong)

Pengujian dilakukan dengan menggunakan MQTT *Broker* yang akan mengirim data kedalam MongoDB dalam keadaan kosong.

6.3.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui lama waktu yang dibutuhkan oleh MQTT-Broker untuk mengirim *stack* data ketika panjang data sudah sampai N-data, dan data sudah terkirim kedalam basis data dengan kondisii basis data kosong. Lama waktu ini yang digunakan sebagai informasi pengujian.

6.3.2 Prosedur Pengujian

Berikut proseur pengujian yang digunakan untuk menguji sistem :

1. Mengkosongkan database sebelum digunakan untuk menyimpan data.
2. Menjalankan program MQTT-Broker dan menunggu sampai panjang data *stack* sampai pada *tresholdnya* untuk dikirim kedalam *database*.
3. Melakukan pencatatan *delay* kedalam tabel sistem basis data.

6.3.3 Hasil dan analisis

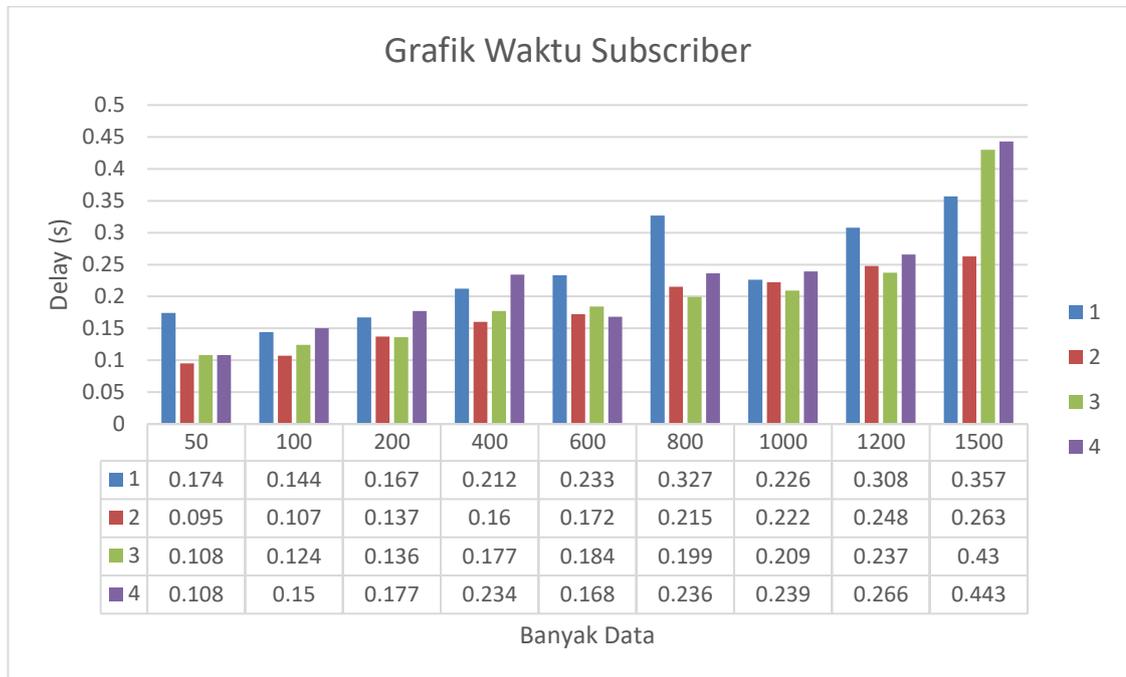
Hasil pengujian disimpan dalam tabel 6.3.

Tabel 6. 3. Waktu Pengiriman Data dari Broker ke MongoDB (Database Kosong)

Banyak Node	Banyak Data	Delay (S)
1	50	0.174
	100	0.144
	200	0.167
	400	0.212
	600	0.233
	800	0.327
	1000	0.226
	1200	0.308
	1500	0.357
2	50	0.095
	100	0.107
	200	0.137
	400	0.160
	600	0.172
	800	0.215
	1000	0.222
	1200	0.248
	1500	0.263
3	50	0.108
	100	0.124
	200	0.136
	400	0.177
	600	0.184
	800	0.199
	1000	0.209
	1200	0.237
	1500	0.430
4	50	0.108
	100	0.150
	200	0.177
	400	0.234
	600	0.168
	800	0.236
	1000	0.239
	1200	0.266
	1500	0.443

Hasil dari tabel diatas menunjukkan bahwa relasi antar banyaknya NodeMCU dengan kecepatan pengiriman data dari MQTT-Broker kedalam basis data MongoDB tidak menunjukkan relasi yang sesuai. Relasi yang sesuai adalah

banyaknya jumlah data yang dikirimkan kedalam sistem basis data dengan kecepatan waktu penyimpanan data kedalam sistem basis data MongoDB tersebut.



Gambar 6. 4. Grafik Waktu MQTT-Broker Mengirim Data ke Database Kosong

6.4 Pencatatan Waktu Pengiriman Data dari Broker ke MongoDB (Database Sudah Ada Sebelumnya)

Pengujian dilakukan dengan menggunakan MQTT *Broker* yang akan mengirim data kedalam MongoDB dalam keadaan *database* dan *documents* sudah ada sebelumnya. Pada proses ini terdapat waktu pengiriman atau *delay* dari sistem. Pengujian *delay* ini bertujuan untuk mencatat waktu pengiriman dari *broker* MQTT.

6.4.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui lama waktu yang dibutuhkan oleh MQTT-Broker untuk mengirim *stack* data ketika panjang data sudah sampai N-data, dan data sudah terkirim kedalam basis data dengan kondisii basis data sudah mempunyai tabel yang akan diisikan sebelumnya. Lama waktu ini yang digunakan sebagai informasi pengujian.

6.4.2 Prosedur Pengujian

Berikut proseur pengujian yang digunakan untuk menguji sistem :

1. Membuat database dengan nama database sama, dan kolom sama seperti pada program sebelum digunakan untuk menyimpan data.

2. Menjalankan program MQTT-Broker dan menunggu sampai panjang data *stack* sampai pada *tresholdnya* untuk dikirim kedalam *database*.
3. Melakukan pencatatan *delay* kedalam tabel sistem basis data.

6.4.3 Hasil dan analisis

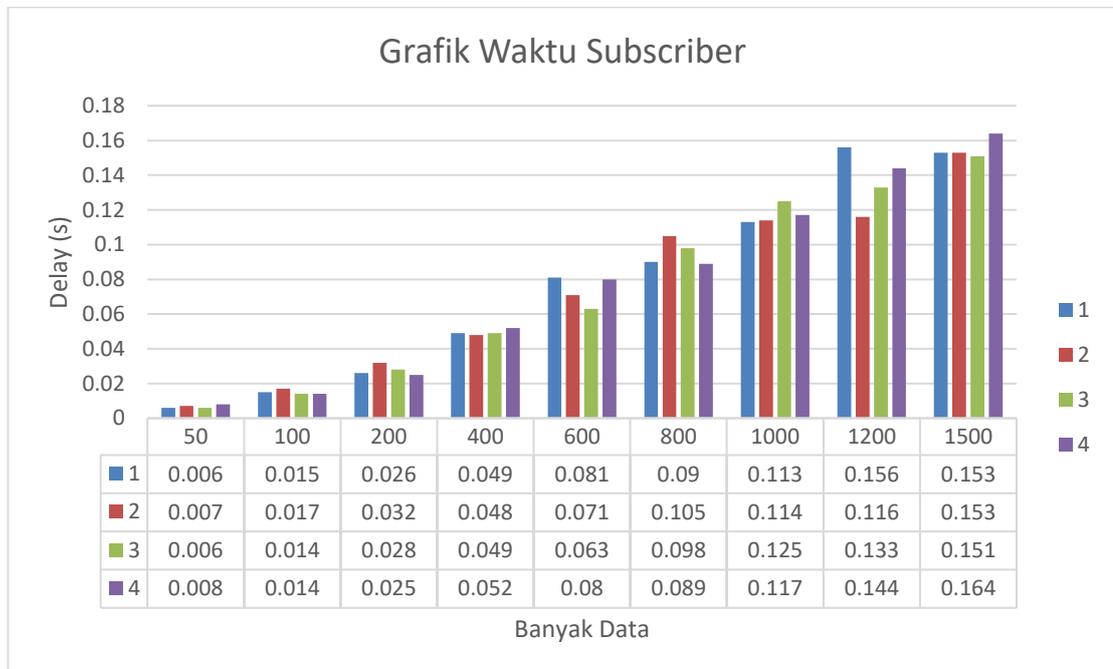
Hasil pengujian disimpan dalam tabel 6.4.

Tabel 6. 4. Waktu Pengiriman Data dari Broker ke MongoDB (Database Ada)

Banyak Node	Banyak Data	Delay (S)
1	50	0.006
	100	0.015
	200	0.026
	400	0.049
	600	0.081
	800	0.09
	1000	0.113
	1200	0.156
	1500	0.153
2	50	0.007
	100	0.017
	200	0.032
	400	0.048
	600	0.071
	800	0.105
	1000	0.114
	1200	0.116
	1500	0.153
3	50	0.006
	100	0.014
	200	0.028
	400	0.049
	600	0.063
	800	0.098
	1000	0.125
	1200	0.133
	1500	0.151
4	50	0.008
	100	0.014
	200	0.025
	400	0.052
	600	0.080
	800	0.089

Banyak Node	Banyak Data	Delay (S)
	1000	0.117
	1200	0.144
	1500	0.164

Hasil dari tabel diatas menunjukkan bahwa relasi antar banyaknya NodeMCU dengan kecepatan pengiriman data dari MQTT-Broker kedalam basis data MongoDB tidak menunjukkan relasi yang sesuai. Relasi yang sesuai adalah banyaknya jumlah data yang dikirimkan kedalam sistem basis data dengan kecepatan waktu penyimpanan data kedalam sistem basis data MongoDB tersebut. Perbedaan antara pengujian dimana kondisi sistem basis data kosong dan terisi melihatkan perbedaan yang sangat signifikan. Hal ini disebabkan karena ketika basis data dalam keadaan kosong, sistem basis data juga melakukan *query* untuk membuat *database* baru dan melakukan *query insert* setelahnya.



Gambar 6. 5. Grafik Waktu MQTT-Broker Mengirim Data ke Database yang Sudah Ada