

## BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai pengujian serta analisis dari hasil pengujian yang telah dilakukan berdasarkan sistem yang dibuat. Tujuan dari pengujian dan analisis ini yaitu untuk mengetahui apakah sistem yang sudah dibuat sesuai dengan analisis kebutuhan yang diinginkan. Pengujian terhadap sistem dilakukan dalam beberapa tahapan yang sesuai dengan kebutuhan yang diinginkan. Sedangkan analisis dilakukan untuk menarik kesimpulan dari penelitian yang sudah dilakukan.

### 6.1 Pengujian Sensor MQ 2

Di sistem ini, sensor MQ2 mendeteksi dan menghitung kadar dari asap. Sehingga pada pengujian Sensor MQ2 ini diperlukan adanya asap untuk melihat dari hasil pembacaan dari sensor MQ2.

#### 6.1.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat keakuratan dan respon dari setiap sensor ketika diberikan gas sesuai dengan karakteristik sensor.

#### 6.1.2 Prosedur Pengujian

Berikut adalah prosedur yang dilakukan untuk menguji sensor MQ2:

1. Menghubungkan *Main Device* dengan laptop melalui kabel USB
2. Meng-*upload* kode program dari sensor MQ2
3. Memberikan asap ke area sensor dengan cara membakar kertas
4. Melihat perubahan nilai dari sensor MQ2 yaitu dengan satuan PPM (*Part per Million*) melalui Serial Monitor yang terdapat pada Arduino IDE

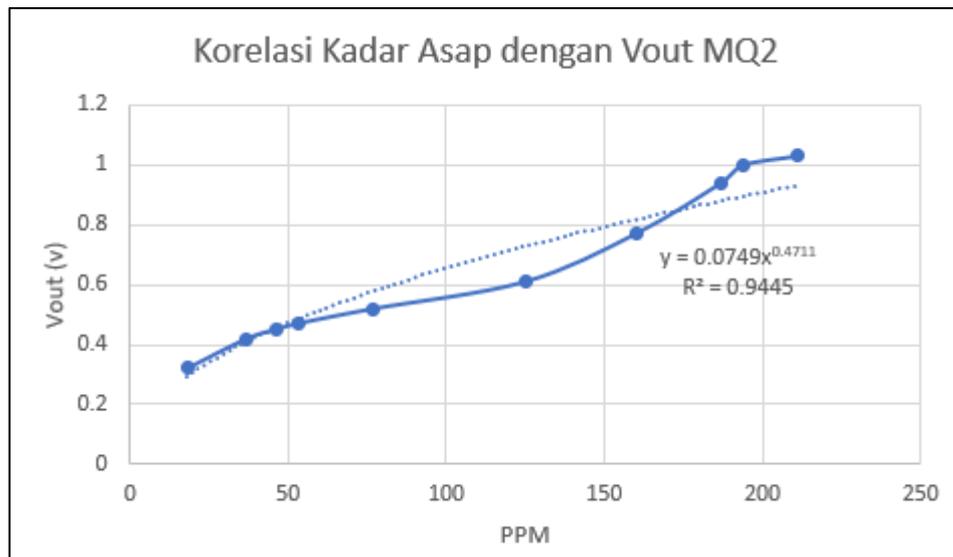
#### 6.1.3 Hasil dan Analisis Pengujian

Tabel 6. 1 Hasil pengujian pembacaan sensor MQ2

Pengujian Ke -	PPM	Vout (V)
1	18	0,32
2	37	0,42
3	46	0,45
4	53	0,47
5	77	0,52
6	125	0,61
7	160	0,77
8	187	0,94

Pengujian Ke -	PPM	Vout (V)
9	194	1,00
10	211	1,03

Pada Tabel 6.1 menunjukkan hasil pembacaan sensor MQ2 yang berupa nilai PPM dan Vout dari sensor MQ2. Dari data tersebut dapat dilihat bahwa semakin tinggi nilai PPM yang berasal dari asap, maka semakin tinggi pula vout tegangan keluaran dari sensor MQ2.



**Gambar 6. 1 Analisis korelasi kadar Asap dengan Vout**

Dari gambar 6.1 tersebut, menunjukkan bahwa grafik korelasi antara kadar asap dengan vout MQ2. Grafik berbanding lurus, jadi semakin besar nilai PPM maka semakin besar juga Vout dari sensor MQ2. Untuk nilai  $y = 0,0749x^{0,4711}$  merupakan model regresi *power* antara variabel PPM dan Vout. Sedangkan untuk nilai  $R^2 = 0,9445$  disebut koefisien determinasi yang artinya variabel PPM asap dipengaruhi oleh variabel Vout sebesar 94,45%. Untuk sisa nilai 5,55% adalah konsentrasi asap dipengaruhi dengan variabel lain.

## 6.2 Pengujian Sensor MQ 7

Di sistem ini, sensor MQ7 mendeteksi dan menghitung kadar dari karbonmonoksida. Sehingga pada pengujian Sensor MQ7 ini diperlukan adanya karbonmonoksida dari asap knalpot motor untuk melihat dari hasil pembacaan dari sensor MQ7.

### 6.2.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat keakuratan dan respon dari setiap sensor ketika diberikan gas sesuai dengan karakteristik sensor.

### 6.2.2 Prosedur Pengujian

Berikut adalah prosedur yang dilakukan untuk menguji sensor MQ7:

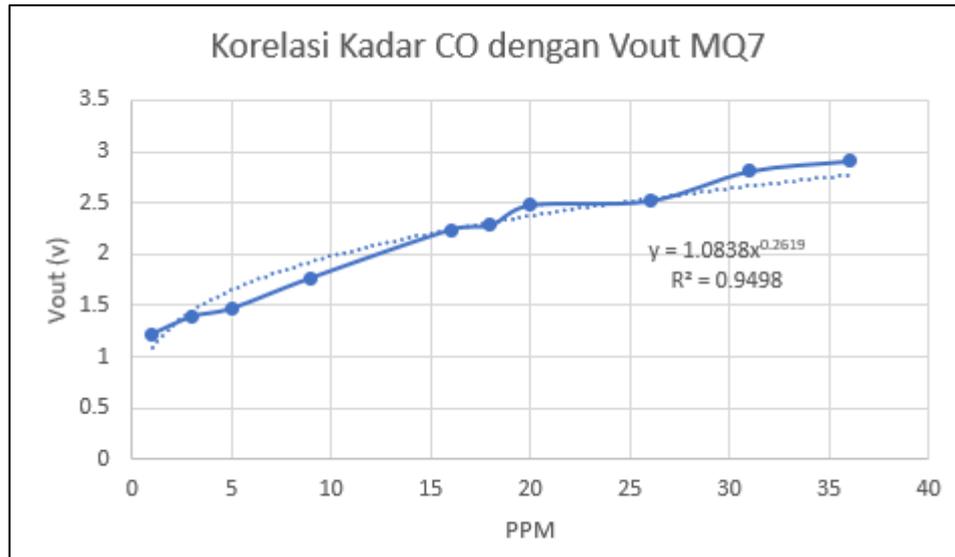
1. Menghubungkan *Main Device* dengan laptop melalui kabel USB
2. Meng-*upload* kode program dari sensor MQ7
3. Memberikan karbonmonoksida ke area sensor dengan cara memberikan asap knalpot motor
4. Melihat perubahan nilai dari sensor MQ7 yaitu dengan satuan PPM (*Part per Million*) melalui Serial Monitor yang terdapat pada Arduino IDE

### 6.2.3 Hasil dan Analisis Pengujian

Tabel 6. 2 Hasil pengujian pembacaan sensor MQ7

Pengujian Ke -	PPM	Vout (V)
1	1	1,22
2	3	1,39
3	5	1,47
4	9	1,77
5	16	2,24
6	18	2,28
7	20	2,48
8	26	2,52
9	31	2,81
10	36	2,91

Dari tabel 6.2, menunjukkan bahwa sensor MQ7 ketika mendeteksi adanya gas CO akan menampilkan kadar dari CO dengan satuan PPM. Seiring bertambahnya nilai PPM dari CO, maka nilai keluaran tegangan dari MQ7 juga semakin tinggi. Hal ini sesuai dengan rumus tegangan, yaitu  $V = I \times R$ . Sehingga dapat disimpulkan pengujian MQ 7 ini berhasil dan sesuai dengan karakteristik MQ7.



**Gambar 6. 2 Analisis korelasi kadar CO dengan Vout**

Gambar 6.2 merupakan grafik korelasi antara CO dengan Vout dari MQ7. Kedua variabel tersebut berbanding lurus, yaitu semakin tingginya nilai PPM yang disebabkan oleh CO, maka Vout yang dihasilkan dari MQ7 akan semakin tinggi juga. Pengujian ini dikatakan berhasil karena sesuai dengan karakteristik pada *datasheet* MQ7. Nilai regresi power antara variabel PPM dan Vout adalah sebesar  $1,0838x^{0,2619}$  dan nilai 94,98% adalah nilai variabel PPM CO dipengaruhi oleh variabel Vout.

### 6.3 Pengujian Sensor MQ 136

Di sistem ini, sensor MQ136 mendeteksi dan menghitung kadar dari sulfurdioksida. Sehingga pada pengujian Sensor MQ136 ini diperlukan adanya sulfurdioksida untuk melihat dari hasil pembacaan dari sensor MQ136.

#### 6.3.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat keakuratan dan respon dari setiap sensor ketika diberikan gas sesuai dengan karakteristik sensor.

#### 6.3.2 Prosedur Pengujian

Berikut adalah prosedur yang dilakukan untuk menguji sensor MQ136:

1. Menghubungkan *Main Device* dengan laptop melalui kabel USB
2. Meng-*upload* kode program dari sensor MQ136
3. Memberikan sulfurdioksida ke area sensor dengan cara membakar bubuk belerang

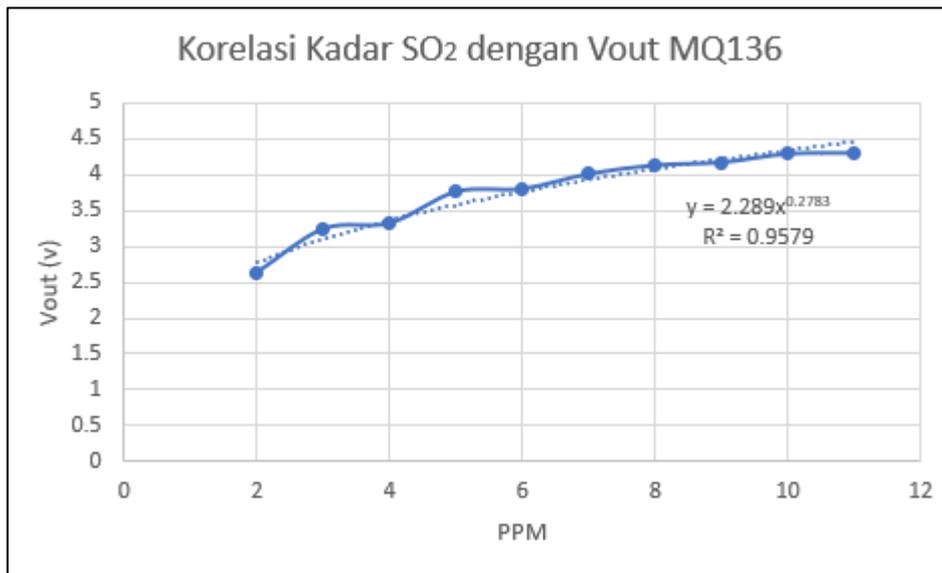
- Melihat perubahan nilai dari sensor MQ136 yaitu dengan satuan PPM (*Part per Million*) melalui Serial Monitor yang terdapat pada Arduino IDE

### 6.3.3 Hasil dan Analisis Pengujian

Tabel 6. 3 Hasil pengujian pembacaan sensor MQ136

Pengujian Ke -	PPM	Vout (V)
1	2	2,63
2	3	3,24
3	4	3,32
4	5	3,76
5	6	3,80
6	7	4,02
7	8	4,13
8	9	4,17
9	10	4,29
10	11	4,30

Tabel 6.3 adalah data hasil pengujian terhadap sensor MQ136. Sama seperti sensor MQ yang lainnya. MQ136 ini akan mendeteksi kadar dari sulfurdioksida, semakin banyak kadar sulfurdioksida maka nilai PPM yang dihasilkan oleh sensor MQ136 akan semakin tinggi. Nilai vout pada sensor MQ136 bergantung pada nilai PPM. Dari pengujian ini dikatakan berhasil karena nilai vout akan berubah seiring peningkatan nilai PPM dan sesuai dengan karakteristik sensor MQ136.



Gambar 6. 3 Analisis korelasi kadar SO<sub>2</sub> dengan Vout

Gambar 6.3 menunjukkan grafik perubahan yang ditunjukkan oleh adanya SO<sub>2</sub> yang dideteksi oleh sensor MQ136. Dapat dilihat bahwa grafik selalu meningkat karena semakin tinggi nilai PPM dari SO<sub>2</sub> maka semakin tinggi juga nilai Vout dari sensor MQ136. Nilai regresi powernya sebesar  $2,289x^{0,2783}$  dan nilai pengaruh variabel PPM sulfurdioksida terhadap variabel Vout sebesar 95,79%.

## 6.4 Pengujian Sensor DHT22

Di sistem ini, sensor DHT22 sebagai fitur tambahan yaitu untuk mengukur suhu sekitar. Sehingga pada pengujian sensor DHT22 ini diperlukan adanya api dari korek untuk menguji dan melihat dari hasil pembacaan sensor DHT22.

### 6.4.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat keakuratan dan respon dari setiap sensor ketika diberikan panas dengan karakteristik sensor.

### 6.4.2 Prosedur Pengujian

Berikut adalah prosedur yang dilakukan untuk menguji sensor DHT22:

1. Menghubungkan *Main Device* dengan laptop melalui kabel USB
2. Meng-*upload* kode program dari sensor DHT22
3. Memberikan api ke area sensor
4. Melihat perubahan nilai dari sensor DHT22 melalui Serial Monitor yang terdapat pada Arduino IDE
5. Kemudian setelah direkap dicari presentase kesalah dari pengujian dengan rumus

$$\frac{\text{Pengukuran Manual} - \text{Pengukuran Sensor}}{\text{Pengukuran Manual}} \times 100$$

6. Kemudian presentase kesalahan dihitung rata – ratanya dari 10 pengujian

### 6.4.3 Hasil dan Analisis Pengujian

Tabel 6. 4 Hasil pengujian pembacaan sensor DHT22

Pengujian Ke -	Lama Pemanasan Api	Termometer °C	DHT22 °C	Error (%)
1	0 detik	29,00	28,40	2,07%
2	5 detik	30,95	29,40	5%
3	10 detik	30,40	30,30	0,32%
4	15 detik	31,50	31,10	1,26%
5	20 detik	31,90	31,20	2,19%
6	25 detik	31,70	31,70	0%

Pengujian Ke -	Lama Pemanasan Api	Termometer °C	DHT22 °C	Error (%)
7	30 detik	32,70	32,60	0,3%
8	35 detik	32,00	32,70	2,1%
9	40 detik	35,00	33,40	4,57%
10	45 detik	31,00	33,80	9%
Rata – rata error				2,68%

Tabel 6.4 merupakan hasil pengujian dari sensor DHT22 yang dibandingkan dengan termometer. Dari hasil di atas dapat disimpulkan bahwa sensor DHT memiliki error rate yang tidak terlalu besar yaitu dengan rata – rata error sebesar 2,68 %.

## 6.5. Pengujian Jarak NRF24L01

Pada sub bab ini, pengujian yang dilakukan adalah menguji jarak NRF24L01 yang berkomunikasi antara *Main Device* dan *Hub Device*. Pengujian ini dilakukan dengan meletakkan *Main Device* di titik tertentu dan kemudian *Hub Device* di letakkan di jarak maksimal hingga data tidak bisa diterima oleh nrf yang sebagai *receiver*. Pengiriman data NRF24L01 ini menggunakan gelombang radio, sehingga *Main Device* dan *Hub Device* tidak terganggu dengan interferensi dengan gelombang yang lain karena sudah teinisialisasi dengan adanya channel agar dapat saling berkomunikasi.

### 6.5.1 Tujuan Pengujian

Tujuan pengujian ini adalah untuk menguji jarak maksimal NRF24L01 dalam pengiriman data sesuai dengan karakteristik NRF24L01.

### 6.5.2 Prosedur Pengujian

Berikut adalah prosedur yang dilakukan untuk menguji jarak NRF24L01 :

1. Menghubungkan *Main Device* dengan sel surya.
2. Menghubungkan *Hub Device* dengan laptop.
3. Menempatkan *Hub Device* setiap jarak kelipatan 10m dari *Main Device*.
4. Melihat data di serial monitor pada *Hub Device*

### 6.5.3 Hasil dan Analisis Pengujian

Tabel 6. 5 Hasil pengujian jarak NRF24L01

Pengujian Ke -	Jarak (m)	Status
1	5	Berhasil

Pengujian Ke -	Jarak (m)	Status
2	10	Berhasil
3	15	Berhasil
4	20	Berhasil
5	25	Berhasil
6	30	Berhasil
7	35	Berhasil
8	40	Berhasil
9	45	Berhasil
10	50	Berhasil
11	55	Berhasil
12	60	Gagal

Pada Tabel 6.5 menunjukkan jarak yang dapat diterima oleh NRF24L01 dalam menerima data. Setelah dilakukan pengujian, hasilnya adalah ketika NRF24L01 yang sebagai *receiver* diberi jarak 5 – 50 meter dengan NRF24L01 yang sebagai *transmitter*, data masih dapat diterima oleh *receiver*. Sedangkan ketika pada jarak 55 meter, data terkadang masuk dan kadang juga terputus. Jarak 55 – 60 meter, NRF24L01 yang sebagai *receiver* sudah tidak bisa menerima data dari NRF24L01 sebagai *transmitter*.

## 6.6 Pengujian Jarak Konektivitas WiFi ESP8266

Pada *Hub Device*, terdapat board NodeMCU yang memiliki chip ESP8266. Di sistem ini, ESP8266 sebagai server yang memberikan layanan WiFi Access Point agar pengguna bisa terkoneksi dengan WiFi ESP8266. Maka dari itu, perlu adanya pengujian untuk jarak maksimal area WiFi ESP8266 agar bisa terjangkau dengan aplikasi pengguna.

### 6.6.1 Tujuan Pengujian

Tujuan pengujian ini adalah agar pengguna dapat memperkirakan jarak aplikasi dengan *Hub Device* agar aplikasi bisa terkoneksi dengan WiFi ESP8266.

### 6.6.2 Prosedur Pengujian

Berikut adalah prosedur yang dilakukan untuk menguji jarak jangkauan WiFi ESP8266 :

1. Menghubungkan *Hub Device* dengan sel surya
2. Menyalakan WiFi “!” pada *smartphone*
3. Menguji koneksi WiFi tiap jarak 10 meter dari *Hub Device*

### 6.6.3 Hasil dan Analisis Pengujian

Tabel 6. 6 Hasil pengujian jarak jangkauan WiFi ESP

Pengujian Ke -	Jarak (m)	Status
1	10	<i>Connected</i>
2	20	<i>Connected</i>
3	30	<i>Connected</i>
4	40	<i>Connected</i>
5	50	<i>Connected</i>
6	60	<i>Connected</i>
7	70	<i>Connected</i>
8	80	<i>Connected</i>
9	90	<i>Connected</i>
10	100	<i>Connected</i>
11	110	<i>Connected</i>
12	120	<i>Connected</i>
13	130	<i>Connected</i>
14	140	<i>Connected</i>
15	150	<i>Connected</i>
16	160	<i>Connected / Disconnected</i>
17	170	<i>Disconnected</i>

Pada Tabel 6.6 menunjukkan bahwa jarak maksimum aplikasi agar dapat terkoneksi dengan WiFi yang ada di ESP8266. Dari tabel di atas, menunjukkan bahwa jarak yang dapat dijangkau oleh WiFi ESP8266 agar connect ke WiFi android yaitu 0 – 160 meter.

## 6.7 Pengujian Waktu Penerimaan Data

### 6.7.1 Tujuan Pengujian

Tujuan pengujian ini yaitu untuk mengetahui waktu yang dibutuhkan sistem untuk mengirim data ke aplikasi ketika pengguna melakukan *request* data. Hal ini dilakukan untuk mengetahui performansi dari sistem.

### 6.7.2 Prosedur Pengujian

1. Menghubungkan *Main Device* dan *Hub Device* ke *power supply*.
2. Aplikasi melakukan koneksi ke WiFi "Air Monitoring NOW!"
3. Pengguna melakukan *request* data, timer dimulai ketika pengguna menekan tombol *request* data
4. Timer selesai ketika data telah tampil di aplikasi

### 6.7.3 Hasil dan Analisis Pengujian

Tabel 6. 7 Hasil pengujian waktu penerimaan data

Pengujian Ke -	Jarak (m)	Waktu (s)
1	10	1,55
2	20	1,23
3	30	1,13
4	40	1,40
5	50	1,48
6	60	2,36
7	70	2,26
8	80	1,96
9	90	1,48
10	100	1,41
11	110	1,50
12	120	1,24
13	130	1,24
14	140	2,39
15	150	0

Tabel 6.7 di atas merupakan hasil pengujian dalam hal waktu pemrosesan sistem untuk mendapatkan data ketika pengguna melakukan *request*. Didata tersebut melakukan 15 pengujian. Disetiap pengujiannya smatphone diberi jarak kelipatan 10 meter. Dari hasil data tersebut dapat disimpulkan bahwa jarak maksimum agar dapat menerima data dari *smartphone* yaitu 140 meter dari *Hub Device*. Jika dilihat dari hasil waktu tiap pengujiannya, jarak tidak mempengaruhi lama tidaknya waktu yang dibutuhkan untuk menerima data. Yang mempengaruhi yaitu kestabilan jaringan.

## 6.8 Pengujian *Finite State Machine*

### 6.8.1 Tujuan Pengujian

Tujuan pengujian ini adalah untuk melihat keberhasilan dari pemodelan *Finite State Machine* dengan melihat output dari setiap *state* ketika kondisi udara normal, udara tidak sehat, ataupun ketika pengguna melakukan *request* data.

### 6.8.2 Prosedur Pengujian

Dalam pengujian *Finite State Machine* ini diuji berdasarkan 3 kondisi, yaitu ketika keadaan udara baik, kemudian diuji ketika keadaan udara tidak sehat, dan yang terakhir dimana kondisi ketika pengguna melakukan *request* data melalui aplikasi.

### 6.8.3 Hasil dan Analisis Pengujian

Hasil dan analisis pengujian ini terbagi menjadi 2, yaitu pengujian FSM dibagian *Main Device* dan yang kedua di *Hub Device*.

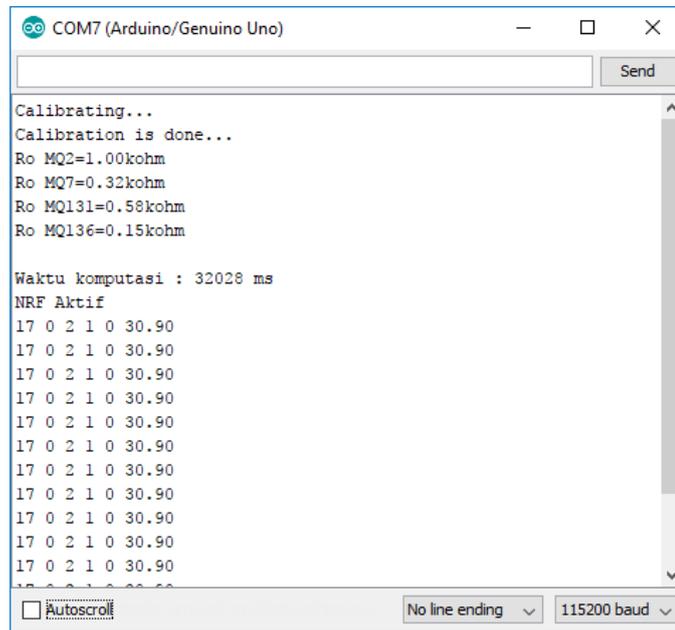
#### 6.8.3.1 Pengujian FSM pada *Main Device*

Tabel 6. 8 Hasil pengujian FSM transisi *state* Calibration ke Reading

No	State	Transisi ke State	Event	Action	Trigger (Waktu)	Status
1	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
2	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
3	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
4	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
5	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
6	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
7	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
8	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
9	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil
10	Calibration	Reading	Akuisisi Data	Sensor read data & NRF Aktif	32028 ms	Berhasil

Dari data tabel di atas, menunjukkan bahwa FSM yang diterapkan pada *Main Device* untuk *state* Calibration ke *state* Reading ditrigger dengan waktu

selama 32028 ms atau 32 detik. Dari 10 pengujian, keberhasilannya mencapai 100%.



**Gambar 6. 4 Hasil pengujian FSM transisi *state* Calibration ke Reading**

Gambar 6.4 merupakan serial monitor ketika terjadi transisi dari *state* Calibration ke *state* Reading. Dapat dilihat bahwa sesudah calibration, terdapat nilai Ro dari setiap sensor dan waktu komputasi dari perpindahan *statenya*. Kemudian setelah bertransisi ke *state* reading, NRF juga diaktifkan.

**Tabel 6. 9 Hasil pengujian FSM transisi *state* Reading ke Reading**

No	State	Transisi ke State	Event	Action	Trigger (Request)	Status
1	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil
2	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil
3	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil
4	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil
5	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil
6	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil
7	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil

No	State	Transisi ke State	Event	Action	Trigger (Request)	Status
8	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil
9	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil
10	Reading	Reading	Server Request	NRF send data	Request Data	Berhasil

Tabel 6.9 merupakan hasil pengujian dari transisi *state Reading* ke *Reading*. *Event* dari perpindahan *state* tersebut yaitu *server request* dan actionnya adalah *NRF send data*. Untuk membuktikan apakah NRF dapat *send data* ketika ada *request* maka pengujian ini harus melakukan *request data* dari aplikasi *smartphone*. Dari percobaan sebanyak 10 kali, hasil pengujian di atas telah berhasil semua dengan nilai keberhasilan sebesar 100%.

```

COM7 (Arduino/Genuino Uno)

13 0 2 1 0 29.10
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
Get Request
Send Data Success
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
13 0 2 1 0 29.20
Autoscroll

```

**Gambar 6. 5 Hasil pengujian FSM transisi *state Reading* ke *Reading***

Gambar di atas adalah tampilan ketika *Main Device* melakukan akuisisi data dan kemudian ada interrupt yaitu adanya *request* dari client. Maka yang dilakukan oleh *state reading* ini yaitu *get data request* kemudian *send data* 1 periode.

**Tabel 6. 10 Hasil pengujian FSM transisi *state Reading* ke *Reading***

No	State	Transisi ke State	Event	Action	Trigger (Asap)	Status
1	Reading	Reading	Kondisi buruk	NRF send data	276 ppm	Berhasil

No	State	Transisi ke State	Event	Action	Trigger (Asap)	Status
2	Reading	Reading	Kondisi buruk	NRF send data	234 ppm	Berhasil
3	Reading	Reading	Kondisi buruk	NRF send data	155 ppm	Berhasil
4	Reading	Reading	Kondisi buruk	NRF send data	161 ppm	Berhasil
5	Reading	Reading	Kondisi buruk	NRF send data	224 ppm	Berhasil
6	Reading	Reading	Kondisi buruk	NRF send data	245 ppm	Berhasil
7	Reading	Reading	Kondisi buruk	NRF send data	189 ppm	Berhasil
8	Reading	Reading	Kondisi buruk	NRF send data	216 ppm	Berhasil
9	Reading	Reading	Kondisi buruk	NRF send data	191 ppm	Berhasil
10	Reading	Reading	Kondisi buruk	NRF send data	216 ppm	Berhasil

Tabel 6.10 merupakan pengujian FSM pada transisi reading ke reading dengan trigger kondisi udara buruk. Percobaan di atas, nilai ppm yang digunakan untuk mentrigger agar *state* dapat bertransisi yaitu dengan mengambil parameter asap. Dari 10 percobaan, hasil yang didapatkan yaitu berhasil semua dengan nilai keberhasilan 100%.

```

17 (Arduino/Genuino Uno)
Asap
102 4 2 1 0 29.70
133 5 2 1 0 29.70
276 5 2 1 0 29.70
Kondisi Buruk !!
Send Data Success
234 5 2 1 0 29.70
Kondisi Buruk !!
Send Data Success
155 5 2 1 0 29.80
Kondisi Buruk !!
Send Data Success
155 4 2 1 0 29.80
Kondisi Buruk !!
Send Data Success
161 4 2 1 0 29.80
Kondisi Buruk !!
Send Data Success

```

**Gambar 6. 6 Hasil pengujian FSM transisi *state* Reading ke Reading**

Gambar 6.6 merupakan tampilan di serial monitor untuk pengujian FSM transisi dari *state* reading ke reading dengan trigger kondisi udara buruk. Dapat dilihat bahwa untuk kadar dari asap ketika diatas 150 ppm, maka terdapat

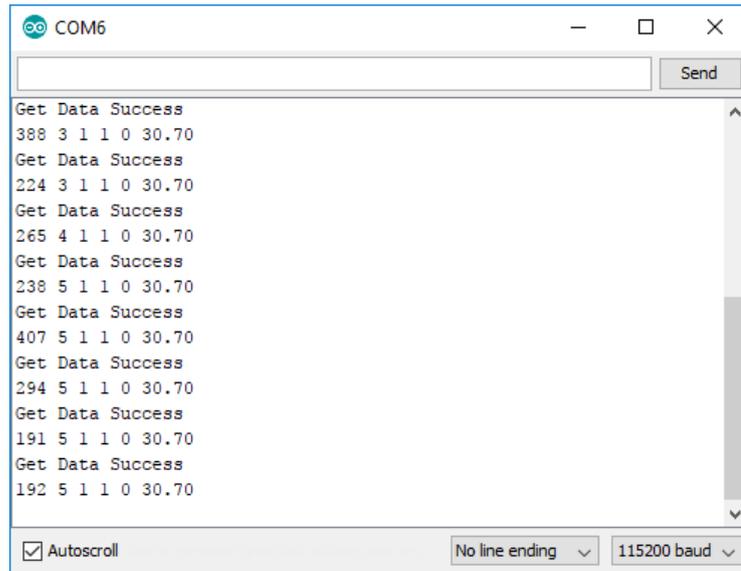
notifikasi bahwa kondisi udara buruk dan *Main Device* akan *send* data ke server melalui NRF24L01.

### 6.8.3.2 Pengujian FSM pada *Hub Device*

Tabel 6. 11 Hasil pengujian FSM transisi *state* Waiting ke Waiting

No	State	Transisi ke State	Event	Action	Trigger (NRF Transmit)	Status
1	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
2	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
3	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
4	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
5	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
6	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
7	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
8	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
9	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil
10	Waiting	Waiting	Get data sensor	NRF receive data	Aktif	Berhasil

Tabel 6.11 adalah hasil pengujian FSM pada *Hub Device* ketika terjadi transisi *state* waiting ke waiting. Data tersebut menunjukkan bahwa *state* tersebut akan berpindah ketika NRF menerima data dari *Main Device*. Dari jumlah seluruh pengujian di atas, dapat disimpulkan bahwa transisi *state* tersebut berhasil dengan presentase keberhasilan sebesar 100%.



**Gambar 6. 7 Hasil pengujian FSM transisi *state* Waiting ke Waiting**

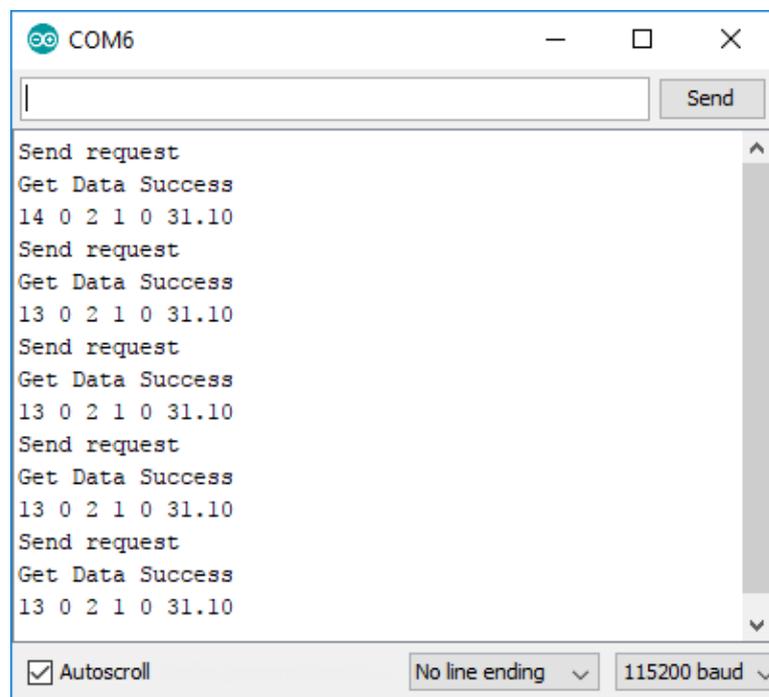
Gambar 6.7 merupakan hasil pengujian untuk transisi waiting ke waiting ketika kondisi *Main Device* mengirim data ke *Hub Device*. Dari percobaan tersebut untuk membuktikan berhasil tidaknya yaitu dengan memberikan kadar asap agar nilai ppmnya di atas 150 dan *Main Device* akan mengirim data ke *Hub Device*. Sehingga nantinya *Hub Device* akan menampilkan karakter “Get Data Success” yang artinya data telah berhasil didapatkan.

**Tabel 6. 12 Hasil pengujian FSM transisi *state* Waiting ke Waiting**

No	State	Transisi ke State	Event	Action	Trigger (Request)	Status
1	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil
2	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil
3	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil
4	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil
5	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil
6	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil
7	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil
8	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil

No	State	Transisi ke State	Event	Action	Trigger (Request)	Status
9	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil
10	Waiting	Waiting	Client request	NRF send request	Aktif	Berhasil

Tabel 6.12 merupakan hasil data dari pengujian *state* waiting transisi ke waiting dengan trigger dari *request* client. Dari data tersebut telah didapatkan hasil yaitu seluruh pengujian telah berhasil dengan presentase keberhasilan sebesar 100%.



**Gambar 6. 8 Hasil pengujian FSM transisi *state* Waiting ke Waiting**

Gambar 6.8 merupakan tampilan serial monitor pada *Hub Device* ketika transisi waiting ke waiting dengan adanya trigger *request* dari pengguna. Ketika mendapatkan *request*, maka *state* waiting akan *send request* ke *Main Device* agar *Main Device* mengembalikan pesan berupa data kondisi udara saat itu juga. Setelah berhasil mendapatkan data, maka di serial monitor *Hub Device* akan menampilkan data yang didapatkan.