

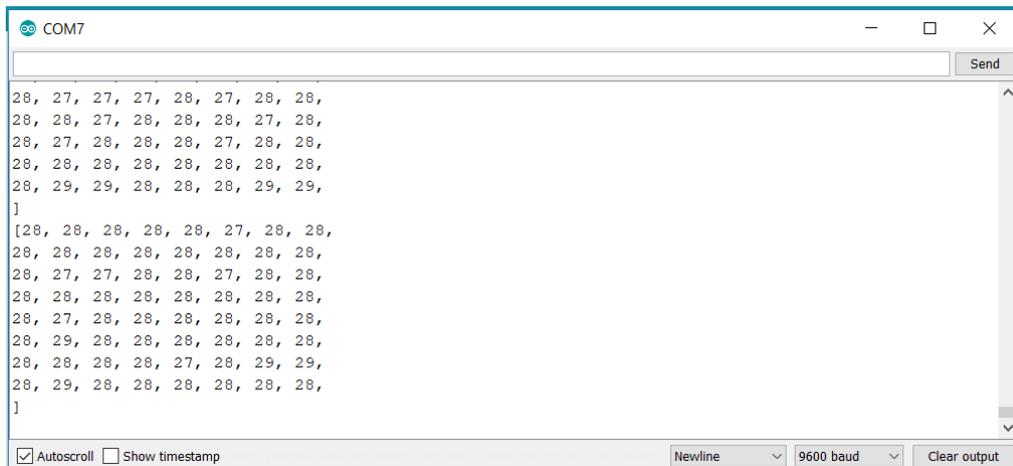
BAB IV

HASIL PENGUJIAN DAN PEDMBAHASAN

4.1 Pengujian Sensor AMG8833 dan Board STM32F103C8T6

Pengujian sensor AMG8833 dan board STM32f103c8t6 ini dilakukan dengan tujuan untuk mengetahui apakah mikrokontroler dan sensor yang dipakai masih berfungsi sesuai yang diharapkan atau tidak. Selain itu, pengujian ini juga dilakukan pada dua kondisi dengan tingkat kecerahan yang berbeda, untuk mengetahui apakah nilai pembacaan sensor terpengaruh oleh tingkat kecerahan cahaya atau tidak.

Pengujian ini dilakukan diarena KRPAI (Kontes Robot Pemadam Api Indonesia) yang berada di Laboratorium Teknik Elektro Universitas Muhammadiyah Yogyakarta. Pengujian pertama dilakukan dalam keadaan tingkat pencahayaan standard yaitu 190 lux.



The image shows a serial terminal window titled 'COM7'. The window contains a list of sensor data points, which are temperature readings for each pixel of the AMG8833 sensor. The data is organized into two groups, each enclosed in square brackets. The first group contains 16 data points, and the second group contains 16 data points. The values are mostly 28, with some 27 and 29. The terminal window has a 'Send' button at the top right and a status bar at the bottom with options for 'Autoscroll', 'Show timestamp', 'Newline', '9600 baud', and 'Clear output'.

```
COM7
28, 27, 27, 27, 28, 27, 28, 28,
28, 28, 27, 28, 28, 28, 27, 28,
28, 27, 28, 28, 28, 27, 28, 28,
28, 28, 28, 28, 28, 28, 28, 28,
28, 29, 29, 28, 28, 28, 29, 29,
]
[28, 28, 28, 28, 28, 27, 28, 28,
28, 28, 28, 28, 28, 28, 28, 28,
28, 27, 27, 28, 28, 27, 28, 28,
28, 28, 28, 28, 28, 28, 28, 28,
28, 27, 28, 28, 28, 28, 28, 28,
28, 29, 28, 28, 28, 28, 28, 28,
28, 28, 28, 28, 27, 28, 29, 29,
28, 29, 28, 28, 28, 28, 28, 28,
]
```

Gambar 4.1 Hasil Pengujian Sensor AMG8833 ke-1

Berdasarkan pengujian sensor AMG8833 pada tingkat pencahayaan standard, dapat diketahui bahwa nilai pembacaan suhu setiap *pixel* sensor rata – rata adalah 28°C dapat dilihat pada gambar 4.1. Kemudian pengujian kedua dilakukan pada tingkat pencahayaan yang terang, yaitu dengan menyalakan lampu yang berada diatas arena KRPAI. Pengujian kedua ini dilakukan pada tingkat pencahayaan yang terang yaitu 460 lux.

```
COM7
28, 28, 28, 28, 28, 28, 28, 28,
28, 28, 28, 28, 28, 28, 28, 28,
28, 28, 28, 28, 28, 28, 28, 29,
29, 28, 28, 28, 28, 28, 29, 29,
28, 29, 28, 28, 28, 28, 29, 29,
]
[28, 28, 28, 28, 27, 28, 28, 28,
28, 28, 28, 27, 28, 28, 28, 28,
28, 28, 27, 28, 28, 28, 28, 28,
28, 28, 28, 28, 28, 28, 28, 28,
28, 28, 28, 28, 28, 28, 28, 28,
28, 28, 28, 28, 28, 28, 28, 29,
28, 28, 28, 28, 28, 28, 29, 28,
28, 29, 28, 28, 29, 28, 29, 29,
]
```

Gambar 4.2 Hasil Pengujian Sensor AMG8833 ke-2

Berdasarkan pengujian sensor AMG8833 pada tingkat pencahayaan yang terang, dapat diketahui bahwa nilai pembacaan suhu setiap *pixel* sensor rata – rata adalah 28°C dapat dilihat pada gambar 4.2. Hal ini menunjukkan bahwa dari pengujian 1 dan 2 dengan tingkat pencahayaan yang berbeda, output hasil pembacaan sensor AMG8833 tidak terdapat perbedaan yang cukup signifikan, dengan presentase pembacaan suhu semua *pixel* sensor sebesar 92% membaca suhu ruangan sekitar yaitu $\leq 28^{\circ}\text{C}$. Selain itu juga dapat diketahui bahwa sensor AMG8833 dan *board* STM32F103C8T6 yang digunakan dalam pengujian ini dapat bekerja dengan baik, sesuai dengan fungsinya.

4.2 Pengujian Program Pengolah Data

Pengujian program pengolah data pembacaan sensor AMG8833 ini diperlukan karena untuk mengetahui apakah program yang telah dibuat sesuai dengan yang diharapkan atau tidak. program pengolah data ini merupakan bagian yang cukup penting pada penelitian ini, karena tujuan dari pengolahan data disini untuk mengetahui letak absolut dari posisi titik api dan untuk mempermudah transmisi data via UART menuju mikrokontroler utama yang terletak pada robot MR.COOL MK7.

4.2.1 Pengujian Program Metode *Bubble Sort*

Pengujian program metode *bubble sort* dilakukan dengan memberikan input *array* data untuk *disorting* atau diurutan dengan

metode *bubble sort* dapat dilihat pada gambar 4.3 point (a). Hasil *array* data yang telah terurut kemudian ditampilkan pada *serial monitor* pada *software* Arduino IDE seperti pada gambar 4.3 point (b).

The image shows two parts of an Arduino IDE window. Part (a) is the code editor with the following code:

```
int input1[8] = { 25, 1, 10, 3, 11, 3, 12, 13 };
int input2[8] = { 20, 1, 35, 19, 15, 43, 8, 2 };
int input3[8] = { 3, 1, 45, 23, 5, 17, 46, 21 };
int input4[8] = { 13, 12, 21, 8, 7, 54, 2, 16 };
int input5[8] = { 21, 31, 1, 13, 3, 7, 32, 23 };
void setup() {
  Serial.begin(9600);
}
void loop() {
  sort(input1,8);
  sort(input2,8);
  sort(input3,8);
  sort(input4,8);
  sort(input5,8);
  tampil_output();
  delay(1000);
}
void sort(int s[], int size) {
  for(int r=0; r<(size-1); r++) {
    for(int o=0; o<(size-(r+1)); o++) {
      if(s[o] > s[o+1]) {
        int t = s[o];
        s[o] = s[o+1];
        s[o+1] = t;
      }
    }
  }
}
void tampil_output() {
```

Part (b) is the serial monitor window showing the output:

```
COM14
Output1 : 1,3,3,10,11,12,13,25, Terbesar : 25
Output2 : 1,2,8,15,19,20,35,43, Terbesar : 43
Output3 : 1,3,5,17,21,23,45,46, Terbesar : 46
Output4 : 2,7,8,12,13,16,21,54, Terbesar : 54
Output5 : 1,3,7,13,21,23,31,32, Terbesar : 32
```

(a)

(b)

Gambar 4.3 Program *bubble sort* (a) dan Hasil *bubble sort* (b)

Pengujian program metode *bubble sort* dilakukan sebanyak 5 kali atau dengan kata lain memberikan 5 *array* data input yang berisi delapan buah angka yang besarnya berbeda pada program metode *bubble sort* untuk *disorting* dan diambil nilai data terbesar. Untuk hasil pengujian program metode *bubble sort* ini dapat dilihat pada tabel 4.1 berikut.

Tabel 4.1 Hasil Pengujian Program *Bubble Sort*

NO	Input	Output	Nilai Data Terbesar
1	(25, 1, 10, 3, 11, 3, 12, 13)	(1, 3, 3, 10, 11, 12, 13, 25)	25
2	(20, 1, 35, 19, 15, 43, 8, 2)	(1, 2, 8, 15, 19, 20, 35, 43)	43
3	(3, 1, 45, 23, 5, 17, 46, 21)	(1, 3, 5, 17, 21, 23, 45, 46)	46
4	(13, 12, 21, 8, 7, 54, 2, 16)	(2, 7, 8, 12, 13, 16, 21, 54)	54
5	(21, 31, 1, 13, 3, 7, 32, 23)	(1, 3, 7, 13, 21, 23, 31, 32)	32

Berdasarkan hasil pengujian program *bubble sort* dapat dilihat bahwa dari 5 kali percobaan pengujian, semua output data yang dihasilkan

setelah *disorting* sudah terurut dan sistem dapat mengetahui nilai data input terbesar. Hal ini menandakan bahwa program *bubble sort* dapat bekerja dengan baik sesuai dengan yang diharapkan. Sehingga saat diterapkan pada sistem pendeteksi posisi api, dengan menggunakan metode ini dapat dicari nilai data terbesar pada setiap *pixel* sisi vertical hasil pembacaan sensor AMG8833. Hal ini dapat meringkas data, yang tadinya 8x8 matrik *array* data menjadi 8x1 *array* data.

4.2.2 Pengujian Program Metode *Weighted Average*

Pengujian program metode *weighted average* dilakukan dengan memberikan input *array* data untuk hitung menggunakan persamaan *weighted average* oleh program. Output hasil dari metode *weighted average*, kemudian akan dibandingkan dengan perhitungan manual, dengan tujuan untuk mengetahui apakah ada selisih atau *error* antara perhitungan menggunakan program dan perhitungan secara manual. Pengujian ini dilakukan sebanyak 5 kali untuk mengetahui dan memastikan program metode *weighted average* yang telah dibuat dapat berjalan sesuai dengan yang diharapkan.

Tabel 4.2 Hasil Pengujian Program *Weighted Average*

NO	Input	Hasil <i>weighted average</i>	Hasil perhitungan	Selisih
1	(28, 27, 26, 95, 95, 27, 28, 26)	99	99.78	-0.78
2	(99, 27, 26, 28, 28, 27, 28, 26)	75	75.17	-0.17
3	(28, 27, 26, 28, 28, 27, 28, 98)	124	124.56	-0.56
4	(28, 99, 26, 28, 28, 27, 28, 26)	81	81.12	-0.12
5	(27, 27, 26, 26, 28, 27, 98, 26)	118	118.68	-0.68

Tabel 4.2 merupakan hasil dari pengujian program *weighted average*. Tabel pengujian diatas memiliki kolom input, hasil *weighted average*, hasil perhitungan dan selisih yang didapat dari pengurangan hasil *weighted average* dengan hasil perhitungan.

Kolom pertama atau kolom input berisi sebuah *array* data yang akan diolah menggunakan metode *weighted average*. Setelah melewati metode ini maka akan didapat sebuah nilai hasil *weighted average* yang

dapat dilihat dikolom 2. Untuk program dan contoh hasil *weighted average* dapat dilihat pada gambar 4.4 berikut.

The image shows two side-by-side screenshots. The left screenshot (a) displays the source code for an Arduino program. The code defines an array of pixel values, calculates a weighted sum in the `loop()` function, and prints the result. The right screenshot (b) shows the serial monitor output, which displays the text "output weighted average = 75".

```

long t_bobot, t_data, pos_Api;
int sort_pixel[8] = { 99, 27, 26, 28, 28, 27, 28, 26};
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

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

  t_bobot=(long) ((long)0*sort_pixel[0]+(long)25*sort_pixel[1]
+(long) 50*sort_pixel[2]+(long) 75*sort_pixel[3]+(long) 125*sort_pixel[4]
+(long) 150*sort_pixel[5]+(long) 175*sort_pixel[6]+(long) 200*sort_pixel[7]);

  t_data =(long) sort_pixel[0]+sort_pixel[1]+sort_pixel[2]+sort_pixel[3]
+sort_pixel[4]+sort_pixel[5]+sort_pixel[6]+sort_pixel[7];

  pos_Api=t_bobot/t_data;

  Serial.print("output weighted average = ");
  Serial.print(pos_Api);Serial.println();
  delay(1000);
}
  
```

(a)

(b)

Gambar 4.4 Program *weighted average* (a) dan Hasil *weighted average* (b)

Kolom ketiga atau kolom hasil perhitungan berisi nilai yang didapat dari perhitungan manual menggunakan metode *weighted average*. Penggunaan perhitungan manual untuk membandingkan kebenaran hasil dari program *weighted average*. Adapun contoh perhitungan manual metode *weighted average* dapat dilihat dibawah ini.

$$WA = \frac{\sum(\text{bobot} \times \text{data})}{\sum \text{data}}$$

$$WA = \frac{(0 \times 99) + (25 \times 27) + (50 \times 26) + (75 \times 28) + (125 \times 28) + (150 \times 27) + (175 \times 28) + (200 \times 26)}{99 + 27 + 26 + 28 + 28 + 27 + 28 + 26}$$

$$WA = \frac{21,725}{289} = 75.17$$

Berdasarkan perhitungan diatas, data input yang dipakai pada perhitungan tersebut adalah *array* data input kedua. Setiap input data kemudian diberi atau dikalikan dengan nilai pembobotan. Kemudian, jumlah data yang telah diberi pembobotan dibagi dengan jumlah data input, sehingga didapat nilai 75.17.

Berdasarkan pengujian metode *weighted average* yang telah dilakukan antara hasil perhitungan program dan hasil perhitungan manual disemua percobaan pengujian mempunyai hasil yang sama tetapi

memiliki sedikit selisih. Contohnya pada percobaan pengujian kedua hasil perhitungan program nilainya 75 sedangkan pada perhitungan manual 75.17. Selisih tersebut terjadi karena hasil perhitungan manual merupakan suatu bilangan desimal. sedangkan variabel yang dipakai pada hasil perhitungan program menggunakan *int* atau *interger* (bilangan bulat) sehingga mengabaikan nilai dibelakang koma. Meskipun terdapat selisih dibelakang koma, tetapi hasil dari pengujian program *weighted average* ini sudah sesuai yang diharapkan dan dapat diketahui bahwa program *weighted average* ini sudah dapat bekerja dengan baik. Sehingga saat diterapkan pada sistem pendeteksi posisi api, dari data input yang berupa 8x1 *array* data dapat diubah menjadi suatu nilai rata – rata, yang akan lebih mudah untuk ditransmisikan melalui protokol UART.

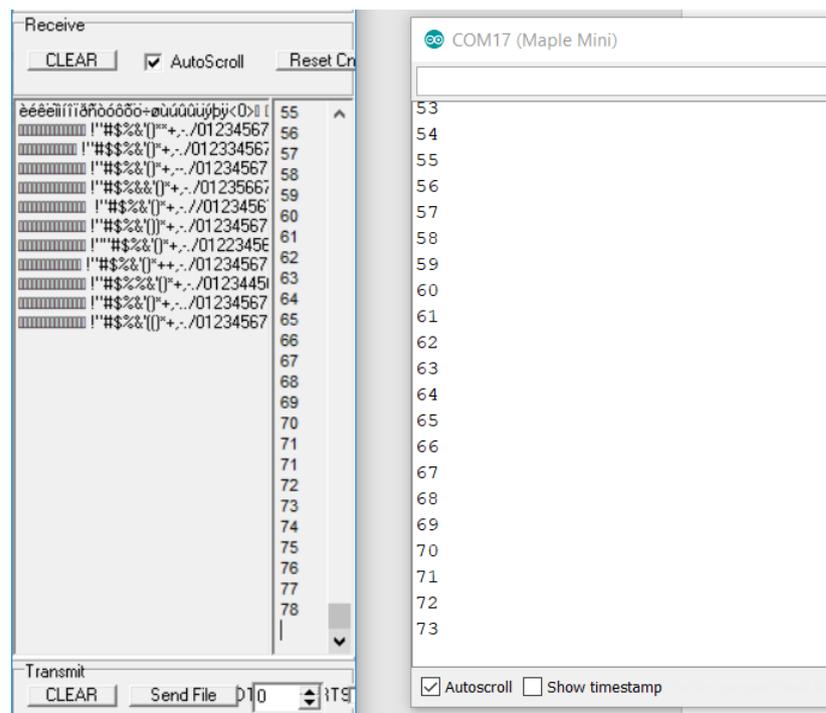
4.3 Pengujian Transmisi Data Melalui Protokol UART

Pengujian transmisi data berfungsi untuk memastikan bahwa tidak terdapat data *error* yang terjadi pada saat transmisi data. Pengujian dilakukan menggunakan STM32F103C8T6 sebagai *transmitter* dan Arduino mega 2560 pro sebagai *receiver*. Untuk memudahkan pengambilan data, data yang dikirim oleh STM32F103C8T6 dan data yang diterima oleh Arduino mega 2560 semua ditampilkan pada *serial monitor* dengan menggunakan USB serial. Gambar pengujian transmisi data dapat dilihat pada gambar berikut.



Gambar 4.5 Pengujian Transmisi Data

Pengujian ini dilakukan sesuai dengan perlakuan pengujian yang terdapat pada BAB 3. Program yang dipakai pada pengujian ini adalah program yang sama dengan program pada gambar 3.20 dan gambar 3.21 dibab 3, yaitu program untuk mengirimkan data angka 1 – 255 secara berurutan.

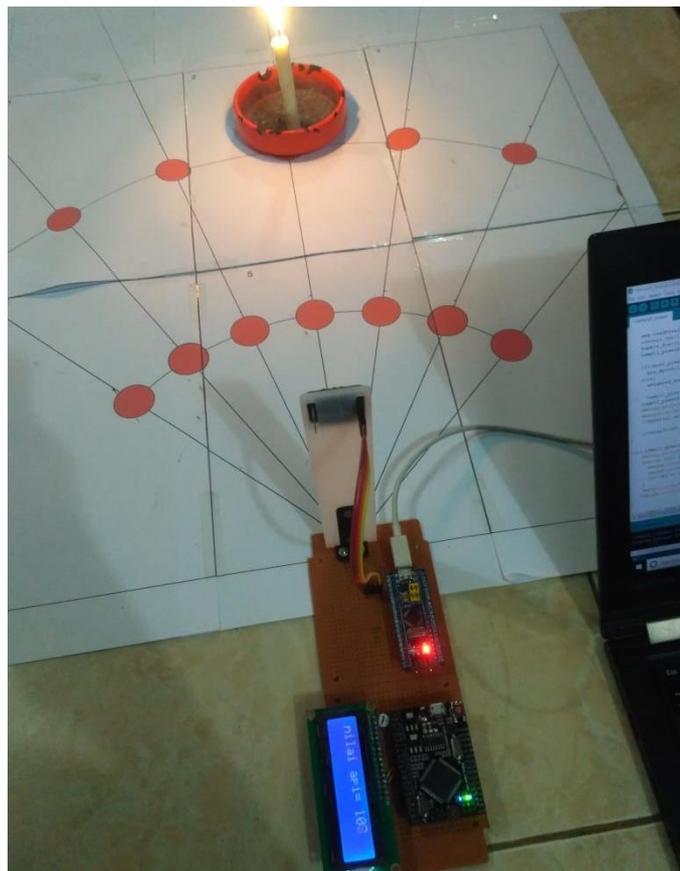


Gambar 4.6 Serial Monitor Pengujian Transmisi Data

Pada gambar 4.6 merupakan Serial monitor Pengujian Transmisi Data. Gambar 4.6 sebelah kiri adalah tampilan serial monitor data yang diterima dan pada gambar 4.6 sebelah kanan adalah tampilan serial monitor data yang dikirim. Perlu diketahui bahwa pada pengujian transmisi data ini menggunakan *baudrate* 9600. *Baudrate* adalah kemampuan kecepatan transmisi data, dengan satuan *bps* (*bit per-second*). *baudrate* 9600 berarti, kecepatan transmisi data yang dilakukan dapat mengirim hingga 9600 *bit* data per-detiknya. Berdasarkan pengujian transmisi data yang telah dilakukan diketahui bahwa tidak ada data yang *error* saat transmisi. Sehingga dapat disimpulkan bahwa jalur protokol UART yang berfungsi untuk transmisi data dapat berkerja dengan baik.

4.4 Pengujian dan Analisis Sistem Deteksi Posisi Titik Api

Pengujian sistem deteksi posisi titik api adalah pengujian terakhir. Fungsi dari pengujian ini adalah untuk mengetahui kemampuan sistem pendeteksi ini dalam mendeteksi posisi titik api baik terhadap rentang sudut pembacaan serta terhadap jarak pembacaan. Selain untuk mengetahui kemampuan sistem deteksi posisi titik api, pengujian ini juga dilakukan untuk mendapatkan acuan parameter nilai untuk menentukan posisi titik api. Berikut ini adalah gambar pengujian sistem deteksi posisi titik api menggunakan sensor AMG8833. Pada gambar 4.4 *board* STM32f103c8t6 adalah mikrokontroler pengolah data dan *board* arduino mega 2560 pro merupakan mikrokontroler utama pada robot MR.COOL MK7 yang dilengkapi LCD sebagai penampil.



Gambar 4.7 Pengujian Sistem Deteksi Posisi Titik Api

Program yang dipakai pada pengujian ini adalah program yang sesuai dengan *flowchart* program pada gambar 3.4 dibab 3. Pengujian sistem deteksi

posisi titik api dilakukan sesuai dengan perlakuan pengujian yang terdapat pada BAB 3. Pengujian dilakukan dengan meletakkan titik api sesuai sudut dan jarak yang telah ditentukan secara bergantian dan melihat nilai parameter yang dihasilkan sistem.

```

COM21 (Maple Mini)
29, 28, 26, 26, 26, 25, 26, 25,
33, 28, 26, 26, 27, 26, 26, 26,
45, 28, 26, 26, 26, 26, 26, 26,
46, 27, 26, 26, 26, 25, 26, 26,
28, 27, 26, 26, 26, 26, 25, 26,
28, 27, 26, 26, 25, 25, 26, 27,
28, 27, 26, 26, 26, 26, 26, 26,
27, 27, 27, 26, 26, 26, 27, 26,

[46, 28, 27, 26, 27, 26, 27, 27, 1]
64

29, 27, 26, 26, 26, 26, 26, 26,
34, 28, 26, 26, 26, 26, 26, 26,
45, 28, 26, 26, 26, 26, 26, 27,
46, 27, 26, 26, 25, 25, 26, 26,
28, 26, 25, 25, 26, 26, 25, 26,
28, 27, 26, 26, 26, 26, 26, 26,
28, 27, 26, 26, 26, 26, 26, 26,
27, 27, 27, 26, 26, 25, 26, 26,

[46, 28, 27, 26, 26, 26, 26, 27, 1]
63

```

Gambar 4.8 *Serial Monitor* Hasil Pengolahan Data

Gambar 4.8 merupakan tampilan *serial monitor* hasil pengolahan data pada *board* STM32f103c8t6. 64 *array* data yang membentuk matrik 8x8 adalah hasil dari pembacaan suhu setiap *pixel* sensor AMG8833. 8x1 *array* data dibawahnya merupakan hasil dari penerapan metode *bubble sort*. 8x1 *array* data tersebut adalah nilai data terbesar yang mewakili data *pixel* setiap sisi *vertical*. kemudian, 8 data tersebut diproses menggunakan metode *weighted average* sehingga didapat nilai output posisi api, seperti pada gambar 4.8 yang terletak pada bagian paling bawah.

Tabel 4.3 Hasil Pengujian Sistem Deteksi Posisi Titik Api

No	Sudut (°)	Jarak (cm)	Output	No	Sudut (°)	Jarak (cm)	Output
1	-45	20	0	15	0	60	91/108
2	-45	40	0	16	0	80	0
3	-45	60	0	17	15	20	132
4	-45	80	0	18	15	40	131
5	-30	20	63	19	15	60	118
6	-30	40	66	20	15	80	0
7	-30	60	66	21	30	20	146
8	-30	80	0	22	30	40	136
9	-15	20	66	23	30	60	134
10	-15	40	74	24	30	80	0
11	-15	60	82	25	45	20	0
12	-15	80	0	26	45	40	0
13	0	20	100	27	45	60	0
14	0	40	99	28	45	80	0

Berdasarkan tabel 4.3 hasil pengujian sistem deteksi posisi titik api terhadap sudut dan jarak, didapat sebuah nilai parameter yang akan menjadi acuan untuk menentukan posisi titik api. Sehingga sistem pendeteksi dapat mengetahui posisi api apakah didepan, disisi serong kiri atau disisi serong kanan.

Hasil pengujian yang mendapatkan output nilai parameter 0 manandakan bahwa sistem pendeteksi tidak dapat menjangkau untuk mendeteksi posisi titik api pada rentang sudut atau jarak tersebut. Sehingga diketahui bahwa sistem pendeteksi ini dapat mendeteksi posisi titik api sampai dengan sekitar jarak 60cm tergantung besar-kecilnya titik api. Rentang sudut pembacaan sistem pendeteksi ini mencapai 60° pada bidang horizontal sesuai pada *datasheet* sensor AMG8833.

Berdasarkan tabel 4.3 hasil pengujian sistem deteksi posisi titik api didapatkan acuan nilai parameter untuk menentukan posisi titik api yaitu: 0 untuk tidak ada api, 40 - 89 untuk posisi api disisi serong kiri, 90-110 untuk posisi api disisi depan, dan 111-160 untuk posisi api disisi serong kanan. Berdasarkan pengujian yang dilakukan maka dapat disimpulkan bahwa sistem pendeteksi posisi titik api dapat berkerja dengan baik sesuai harapan.