BAB IV
UJI COBA DAN ANALISIS SISTEM

Untuk mengetahui kehandalan dan keberhasilan dari sistem yang kita buat, maka diperlukan pengujian terhadap terhadap komponen – komponen pembangun sistem terutama sensor – sensor.

4.1. Pendeteksian Jarak Menggunakan Sensor Ultrasonik

Pengujian sensor ini dilakukan dengan membandingkan besarnya jarak yang telah ditetapkan dengan hasil perhitungan berdasarkan lamanya waktu *high* yang diterima, oleh mirkokontroler dalam program. Berdasarkan *data sheet* sensor *ultrasonic PING)))*, kecepatan suara adalah 1130 kaki/detik atau setara dengan 34442.4 cm/detik. Sehingga 1 cm dapat dalam 29.034 μS. Beberapa tahapan untuk mengetahui besarnya jarak antara sensor dengan objek yaitu:

1. Mikrokontroler mengirimkan pulsa (*trigger* pulse) minimal sebesar 10 μS. Perintah ini dikirim menggunakan instruksi

   PULSE*OUT* *port, pin, Duration*

   Keterangan :

   PULSE*OUT* : menghasilkan pulsa pada *Pin* selama *Duration* yang akan menjadi *input* sensor.

   Port : nama dari sebuah port (PortA – PortG)

   Pin : sebuah variabel/konstanta/ekspresi (0-7) yang menetapkan pin I/O yang digunakan. Pin ini akan diset untuk mode *output.*

   Duration : sebuah variabel/konstanta/ekspresi (1-65535) menetapkan lamanya waktu pulsa yang dihasilkan.

2. Setelah pulsa *trigger* dikirimkan, sensor harus menunggu selama 700 μs sebelum mengirimkan gelombang ultrasonik pendek. Waktu ini juga digunakan oleh sensor PING))) mengirimkan gelombang ultrasonik

pendek selama 200 μS pada frekuensi 40 KHz dan mikrokontroler untuk menyiapkan instruksi selanjutnya.

3. Instruksi yang digunakan untuk memulai pengukuran adalah

PULSEIN *Variable, pinx, Pin, State*

Keterangan:

PULSEIN : mengukur lebar pulsa pada *Pin* yang digambarkan oleh *State* dan menyimpan hasilnya dalam *Variable*.

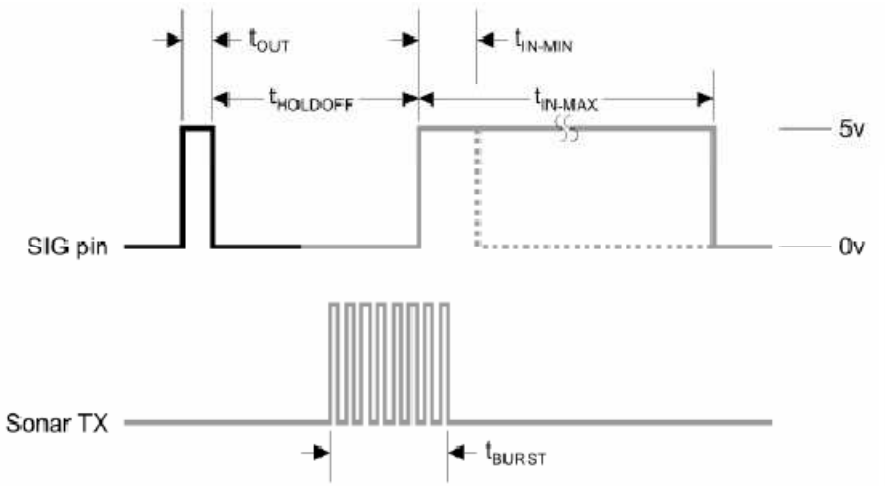
Variable : sebuah variabel (biasanya dalam Word) yang mana pulsa yang diukur akan disimpan.

Pinx : sebuah pin register (PinA – PinG)

Pin : sebuah variabel/konstanta/ekspresi (0-7) yang menetapkan pin I/O yang digunakan. Pin ini akan diset untuk mode *input*.

State : sebuah variabel/konstanta/ekspresi (0-1) yang menetapkan apakah pulsa yang akan diukur *low* (0) atau *high* (1). Pulsa *low* dimulai dengan perpindahan 1 ke 0 dan pulsa *high* dimulai dengan perpindahan 0 ke 1.

Satu unit variabel waktu pada mikrokontroler ATMEGA 64 adalah 10 μS.



Gambar 4.1 *Cara kerja sensor ultrasonik*

Keterangan:

Mikrokontroler

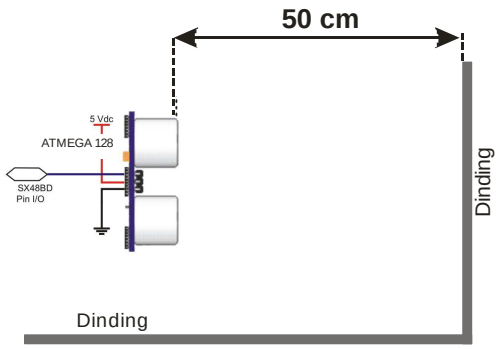
PING

t_{OUT}	min. 10 µS
$t_{HOLDOFF}$	750 µS
t_{BURST}	200 µS @40kHz
t_{IN-MIN}	115 µS
t_{IN-MAX}	18.5 mS

Berikut ini merupakan perintah yang diberikan mikrokontroler untuk mengakses penggunaan dari sensor *ultrasonik*, sehingga jarak benda yang ada di depannya dapat diketahui.

```
Config Portc.1 = Output
Portc.1 = 0                    ; Port PingKanan di nol-kan
Pulseout Portc , 1 , 40        ; Kirim pulsa trigger 10 uS
Pulsein waktu , Pinc , 1 , 1   ; Jika Port PingKanan=1 (HIGH), maka
                                 simpan lama waktu kondisi 1 ke waktu
waktu = Waktu * 10             ; simpan waktu
Jarak = waktu / 61             ; Rumus konversi data ke cm
```

Gambar 4.2 menunjukkan posisi pengujian sensor ultrasonik terhadap dinding. Hasil pengukuran dibandingkan dengan jarak sesungguhnya yang sudah diatur terlebih dahulu.



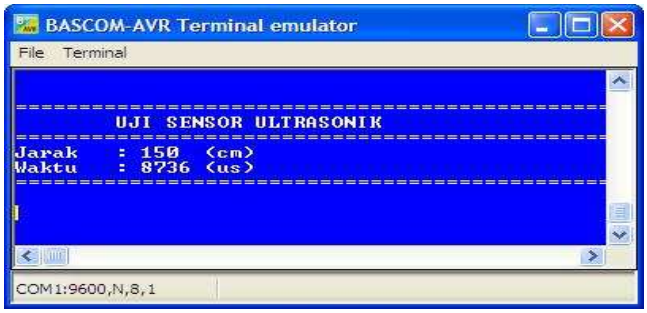
Gambar 4.2 *Posisi pengujian sensor ultrasonik*

Berikut merupakan listing program untuk menguji sensor ultrasonik :

```
$regfile = "m64def.dat"
$crystal = 16000000
$baud = 9600
'-----------------------------------------
Dim Jarak As Word
Dim Waktu As Word
'-----------------------------------------
Do
Gosub Ultrasonik
Print "      tes ultrasonik               "
Print "======================================="
Print
Print "jarak   : " ; Jarak
Print "waktu   : " ; Waktu
Print
Print "======================================="
Loop
End

Ultrasonik:
   Config Portc.1 = Output
   Portc.1 = 0
   Pulseout Portc , 1 , 40
   Pulsein Waktu , Pinc , 1 , 1
   waktu = Waktu * 10
   Jarak = waktu / 61
   Waitms 10
Return
```

Print adalah instruksi untuk menampilkan informasi hasil pengukuran pada layar monitor seperti terlihat pada gambar 4.3 yang merupakan tampilan pengambilan data sensor ultrasonik pada layar monitor.



Gambar 4.3 *Tampilan pengambilan data sensor ultrasonik*

Data diambil dengan posisi sensor ultrasonik tegak lurus terhadap objek yang ada di depannya. Tabel 4.1 menunjukkan data hasil pengukuran. Kolom pertama adalah jarak yang kita tetapkan antara sensor dengan dinding. Kolom kedua dan ketiga adalah waktu dan jarak yang didapatkan dari hasil pengukuran seperti pada gambar 4.3.

Tabel 4.1 *Hasil pengukuran sensor ultrasonik*

Jarak (cm)	Hasil Pengukuran		Error (cm)	Perhitungan
	Waktu (uS)	Jarak (cm)		Jarak (cm)
10	626	10.78	0.78	10.78
20	1172	20.18	0.18	20.18
30	1770	30.48	0.48	30.48
40	2362	40.67	0.67	40.67
50	2927	50.41	0.41	50.41
60	3506	60.38	0.38	60.38
70	4097	70.56	0.56	70.56
80	4590	79.05	0.05	79.05
90	5225	89.98	0.98	89.98
100	5849	100.72	0.72	100.72
150	8736	150.45	0.45	150.45
200	11631	200.30	0.30	200.30
250	14549	250.55	0.55	250.55
Rata-rata error			0.5	

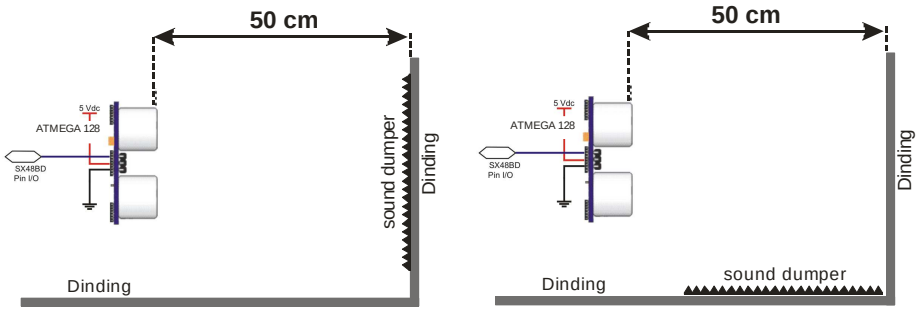
Jarak hasil perhitungan didapatkan dari:

$$\text{Jarak} = \frac{1}{2} \bullet \left(\frac{(t \quad x \quad 0.8)}{29.034} \right) \text{cm}$$

Keterangan :

Jarak	:	Besarnya jarak antara sensor dengan objek (cm)
t	:	Didapatkan dari hasil pengukuran sensor (µS)
0.8	:	Konstanta konversi waktu untuk mode BS2P ke µS
29.034 µS	:	Waktu balikan lebar pulsa menempuh jarak 1 cm.
½	:	pengambilan waktu hanya waktu kembalinya suara (echo) ke Rx sensor

Data pada tabel 4.1 rata – rata error dari sensor ultrasonik sebesar 0.5 cm masih dalam batas kewajaran, karena perubahan data yang dihasilkan bersifat *fluktuatif* dengan nilai error tertinggi sebesar 0.98 cm pada jarak 90 cm dan nilai error terendah sebesar 0.05 pada jarak 80 cm.

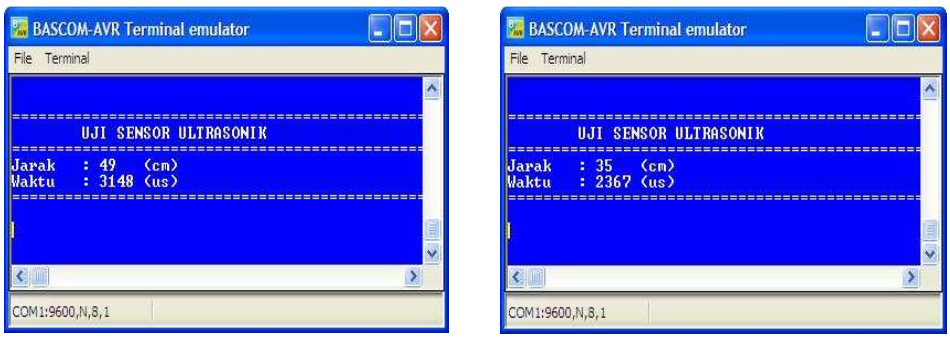


(a) (b)

Gambar 4.4 *a. Posisi Sound Dumper didepan sensor ultrasonik.*
b. Posisi Sound Dumper disamping sensor ultrasonik.

Gambar 4.4(a) menunjukkan posisi sensor ultrasonik terhadap *sound dumper* yang dipasang tegak lurus didepan dinding. Hasil pengukuran menunjukkan, ketika jarak yang ingin diukur sejauh 50 cm tetapi jarak hasil pengukuran menunjukkan 49 cm. Hal ini berarti bacaan sensor kekurangan 1 cm,

kekurangan disebabkan karena adanya penebalan dinding oleh *sound dumper*, tebal dari *sound dumper* itu sendiri adalah 1 cm, dalam hal ini sensor tidak terganggu oleh *sound dumper* jika ditempatkan didepan sensor. Akan tetapi jika posisi *sound dumper* disamping sensor ultrasonik, pada pengukuran dengan jarak 50 cm, hasil bacaan menjadi 35 cm, hal ini menunjukkan *error* yang sangat besar, oleh karena itu untuk mengatasi *error* ini adalah penempatan sensor ultrasonic yang tidak tergangu oleh *sound dumper*. Gambar 4.5(a) merupakan hasil pengukuran dengan posisi *sound dumper* di depan sensor ultrasonik, sedangkan gambar 4.5(b) adalah hasil pengukuran dengan posisi *sound dumper* di samping sensor ultrasonik



(a) (b)

Gambar 4.5 (a) *Tampilan pengukuran Posisi Sound Dumper didepan sensor*

(b) *Tampilan hasil pengukuran Posisi Sound Dumper disamping sensor*

4.2. Pendeteksian Garis Putih Menggunakan Sensor Lantai

Pengujian sensor lantai dilakukan terhadap 2 komponen sensor pengirim cahaya yang berbeda yaitu *LED super bright* dan infra-merah. Hal ini dilakukan untuk mendapatkan modul sensor lantai yang sesuai yaitu hanya akan aktif apabila ada garis yang berwarna putih, dan tidak aktif apabila mengenai warna hitam dan hijau

1. Menggunakan *LED super bright* putih.

Berikut adalah data hasil penggujian sensor lantai menggunakan *LED super bright* putih sebagai media pengirim cahaya.

Tabel 4.2 *Hasil pengujian sensor Lantai menggunakan LED super putih*

media penguji	***Sinyal***
Kertas putih	**Low**
Lantai hitam	**High**
Karpet abu-abu	**High**

2. Menggunakan Infra-merah

Berikut adalah data hasil penggujian sensor lantai menggunakan Infra-merah sebagai media pengirim cahaya.

Tabel 4.3 *Hasil pengujian sensor lantai menggunakan Infra-merah*

media penguji	***Sinyal***
Kertas putih	**Low**
Lantai hitam	**High**
Karpet abu-abu	**Low**

Ket : Sensor lantai aktif = low

Berdasarkan tabel pengujian di atas sensor lantai yang menggunakan *LED super bright* putih lebih handal dari pada sensor lantai yang menggunakan infra-merah, sensor lantai yang menggunakan *LED super bright* putih hanya akan aktif pada kertas warna putih, sedangkan sensor lantai yang menggunakan infra-merah aktif pada kertas putih dan karpet hijau.

Sensor lantai akan mendeteksi garis putih yang berada di area home, di depan pintu ruangan, dan di area sekitar lilin.

4.3. Pencarian titik api menggunakan sensor UVtron

Sensor api yang digunakan menggunakan sensor UVtron, yang digunakan untuk mendeteksi keberadaan api di ruangan.

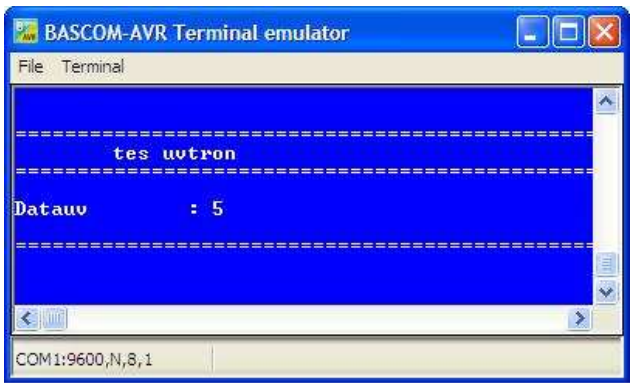
Output dari sensor UVtron berupa pulsa. Pulsa yang disediakan 3, 5, 7, dan 9 pulsa yang bisa dihasilkan dalam interval waktu kurang dari 2 detik. Untuk menguji sensor UVtron, dilakukan pengecekan data 1 dan menjumlahkan banyaknya data 1 tersebut dalam interval waktu tertentu.

Dengan instruksi ini kita bisa menentukan keberadaan api dalam suatu ruangan. Prosedur berikut digunakan untuk mengetahui keberadaan api di dalam ruangan.

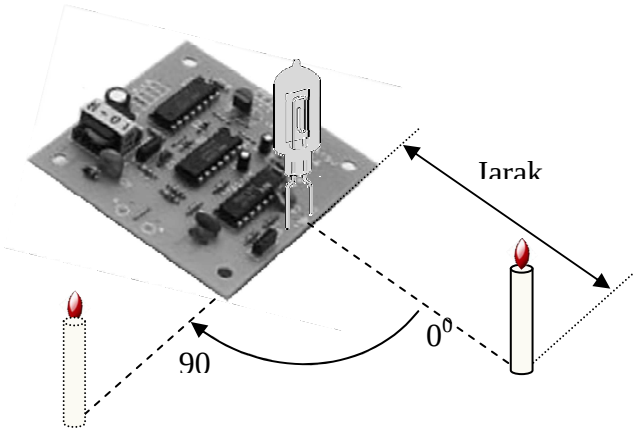
```
$regfile = "m64def.dat"
$crystal = 16000000
$baud = 9600

Uvtron Alias Pind.5
Dim Datauv As Word
Dim Lama As Word
Dim Kali As Word
Do
Gosub Apiuvtron
Print "            tes uvtron               "
Print "=============================="
Print
Print "Datauv   : " ; Datauv
Print
Print "=============================="
Loop
End
Apiuvtron:
Datauv = 0
Kali = 0
Do
   Lama = 0
   Do
         If Uvtron = 1 Then
            Incr Lama
            Incr Datauv
            Do
            Loop Until Uvtron = 0
         Else
            Incr Lama
         End If
   Loop Until Lama = 10000
   Incr Kali
Loop Until Kali = 9
Return
```

Program di atas digunakan untuk mengetahui bahwa di ruangan ada api. Variabel **`uvtron`** digunakan sebagai variabel *port* data *input* pada pind.5, sedangkan **`datauv`** digunakan sebagai variabel untuk menyimpan hasil menghitung (*count*) siklus. Lamanya pengambilan data selama 90000 kali perulangan dengan bahasa BASCOM AVR, atau jika diukur dengan stop watch selama 500 ms.



Gambar 4.6 *Tampilan pengambilan data UVtron*



Gambar 4.7 *Posisi lilin terhadap UVtron*

Gambar 4.15 menunjukkan posisi lilin terhadap UVtron. Lilin dikatakan berada pada sudut nol derajat (0^0) jika diletakkan di depan sensor UVtron. Pengukuran berikutnya dilakukan terhadap lilin yang diletakkan pada sudut 15^0, 30^0, 45^0, 60^0 dan 90^0 yang secara berturut-turut digeser ke arah kanan UVtron.

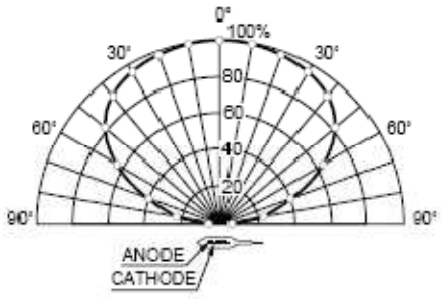
Tabel 4.4 adalah tabel hasil pengukuran respon sensor UVtron terhadap keberadaan nyala lilin berdasarkan jarak dan sudut.

Tabel 4.4 Hasil pengukuran jumlah siklus UVtron

Sudut	Jarak (cm)	*Count*
0^0	25	6
	50	5
	75	3
	100	3
	125	2
	150	2
15^0	25	5
	50	3
	75	3
	100	3
	125	2
	150	2
30^0	25	6
	50	4
	75	3
	100	3
	125	2
	150	2

Sudut	Jarak (cm)	*Count*
45^0	25	5
	50	4
	75	3
	100	2
	125	2
	150	2
60^0	25	5
	50	4
	75	3
	100	3
	125	2
	150	2
90^0	25	2
	50	1
	75	1
	100	1
	125	1
	150	1

Dari tabel 4.4 dapat ditentukan jumlah siklus yang dapat menentukan keberadaan api yang berada dalam suatu ruangan. Api akan dianggap berada dalam suatu ruangan jika jumlah *counter* lebih dari 0 (**`datauv`**>=1).



Gambar 4.8 *Jangkauan dari tabung UVtron*

4.4 Penentuan kecepatan motor

Dalam melakukan aksinya, pergerakan robot haruslah selalu stabil dan konstan, gerakan-gerakan yang sering dilakukan oleh robot seperti maju lurus, belok kanan 90^0, belok kiri 90^0 harus selalu tepat sehingga menghasilkan pergerakan yang sempurna. Untuk itu digunakan suatu komponen pengendali putaran motor yang bekerja secara *close loop*. Komponen ini berfungsi mengatur putaran motor agar tetap stabil dan konstan ketika robot sedang melakukan aksinya mencari dan memadamkan api serta menandai keberadaan bayi. Cara kerja dari kendali motor tersebut yaitu kendali motor akan mengkoreksi putaran dari motor dengan cara membaca frekuensi yang dihasilkan oleh *optocoupler* yang membaca lingkaran plastik dengan pola hitam-transparan berselang, jika besarnya pulsa yang dihasilkan oleh *optocoupler* lebih besar dari frekuensi putaran yang diinginkan maka motor kontrol akan mengurangi kecepatan putaran motor. Demikian juga sebaliknya jika frekuensi bacaan *optocoupler* lebih kecil, maka kendali motor akan menambah kecepatan dari putaran motor.

Ketika terjadi perubahan catu daya yang digunakan untuk motor, hal tersebut akan mempengaruhi kinerja dari putaran motor. Namun dengan digunakannya kendali motor ini, perubahan catu daya yang digunakan tidak akan terlalu berpengaruh bagi kinerja motor, sampai catu daya yang digunakan sudah benar-benar tidak mampu mensuplai daya ke motor. Berikut ini adalah tabel hasil pengujian putaran motor dengan kendali motor, dimana catu daya yang digunakan untuk motor berubah-ubah dengan arah pergerakan putaran motor yaitu maju:

Ketika robot melakukan pergerakan belok-belok pengunaan motor kontrol berperan sangat penting agar ketepatan belok selalu terpenuhi. Berikut di bawah ini beberapa penggalan intruksi yang digunakan untuk melakukan pergerakan belok dari robot.

```
Open "coma.0:9600,8,n,1" For Output As #1
Open "coma.2:9600,8,n,1" For Output As #2

Print #1 , Chr(085) ; Chr(006) ; Chr(001) ; Chr(205) ; Chr(130)
Print #2 , Chr(085) ; Chr(006) ; Chr(001) ; Chr(205) ; Chr(130)
```

Variabel yang dikirimkan berturut-turut dalam intruksi serial di atas adalah

6 = ***count***, mode yang digunakan untuk menghitung pulsa dari *optocoupler*

1 = ***FreqHigh***, bit-bit tertinggi dari frekeunsi yang dikirimkan

205 = ***FreqLow*** , bit-bit terendah dari frekuensi yang dikirimkan

130 **= *Speed,*** kecepatan putaran untuk masing-masing motor

Dari intruksi di atas dapat diketahui bahwa motor akan berputar sebanyak 381 pulsa, baik motor kiri maupun motor kanan ke arah yang telah ditentukan. Kecepatan yang diberikan kepada masing-masing motor berbeda, dikarenakan masing-masing motor *DC* yang digunakan memiliki karakteristik yang berbeda. Pulsa sebanyak 381 didapatkan melalui Perhitungan sebagai berikut:

Jumlah Count = *FreqHigh** 256 + *FreqLow*

Jadi jika nilai *FreqHigh* dan *FreqLow* dimasukan :

$$count = (FreqHigh * 256) + FreqLow$$
$$= (1 * 256) + 205$$
$$= 461$$

Untuk membuktikan apakah jumlah count yang digunakan dapat menghasilkan gerakan belok kanan 90º dan belok kiri 90º, dilakukan percobaan sebanyak 10 kali dan dilihat apakah gerakannya telah benar atau belum. Tabel 4.6 adalah tabel keberhasilan gerakan belok kanan 90º dan belok kiri 90º :

Table 4.5 Keberhasilan algoritma belokkanan dan belokkiri

Pergerakan	Percobaan	Berhasil / Tidak Berhasil
belok kanan 90º	1	Berhasil
	2	Berhasil
	3	Berhasil
	4	Berhasil
	5	Berhasil
	6	Berhasil
	7	Berhasil
	8	Berhasil
	9	Berhasil
	10	Berhasil
belok kiri 90º	1	Berhasil
	2	Berhasil
	3	Berhasil
	4	Berhasil
	5	Berhasil
	6	Berhasil
	7	Berhasil
	8	Berhasil
	9	Berhasil
	10	Berhasil

Keberhasilan pegerakan belok-belok tersebut memiliki toleransi sekitar ± 10 º. Hal tersebut dikarenakan gaya gesek rantai robot terhadap lantai. Sehingga dengan toleransi tersebut pergerakan dianggap berhasil.

4.5 Penentuan arah robot menggunakan kompas

Pada robot, posisi pemasangan kompas diletakkan paling atas dari robot, yaitu sekitar 28,5 cm dari permukaan lantai atau 5 cm daripermukaan robot paling atas, hal ini dimaksudkan agar kompas tidak terpengaruh oleh medan magnetik dan medan listrik yang dihasilkan oleh motor DC dan komponen elektronika.

BASCOM AVR telah menyediakan intruksi PWM untuk mengontrol kompas yaitu instruksi *Pulsein*, listing programnya sebagai berikut :

```
Dim Sudut As Word

Kompas:
Pulsein Sudut , Ping , 1 , 1
Sudut = Sudut - 1000
Sudut = Sudut / 10
Print Sudut
Waitms 20
Return
```

Table 4.6 Hasil pengujian kompas

sudut	Sudut terbaca	*error*	sudut	Sudut terbaca	*error*
0	0	0	195	195	0
15	16	1	210	214	4
30	29	1	225	227	2
45	43	2	240	241	1
60	62	2	255	257	2
75	75	0	270	273	3
90	88	2	285	283	2
105	104	1	300	298	2
120	116	4	315	314	1
135	131	4	330	331	1
150	148	2	345	343	2
165	166	1	360	359	1
180	179	1	Rata-rata *error*		1.68

4.6 Pengujian Algoritma dari Posisi *HOME* Hingga Memadamkan Api

Pengujian algoritma dilakukan untuk mengetahui apakah algoritma yang telah dirancang telah sesuai dengan yang diinginkan atau belum. Berikut ini adalah tabel pengujian algoritma, dari mulai *HOME* sampai dengan memadamkan api, dengan range tegangan terukur untuk mikrokontroler 11.2 V – 10.5 V dan range tegangan terukur untuk motor 22.4 V – 20.1 V. Range tegangan tersebut merupakan batasan tegangan yang digunakan.

- Trial 1 Mode A

$$\text{Persentasi Keberhasilan} = \frac{Banyaknya\ percobaan\ berhasil}{Banyaknya\ seluruh\ Percobaan} X\ 100\%$$

$$\text{Persentasi Keberhasilan} = \frac{149}{150} X\ 100\%$$

$$\text{Persentasi Keberhasilan} = 99{,}3\%$$

- Trial 2 Mode B

$$\text{Persentasi Keberhasilan} = \frac{Banyaknya\ percobaan\ berhasil}{Banyaknya\ seluruh\ Percobaan} X\ 100\%$$

$$\text{Persentasi Keberhasilan} = \frac{195}{200} X\ 100\%$$

$$\text{Persentasi Keberhasilan} = 97{,}5\%$$

- Trial 3 Mode C

$$\text{Persentasi Keberhasilan} = \frac{Banyaknya\ percobaan\ berhasil}{Banyaknya\ seluruh\ Percobaan} X\ 100\%$$

$$\text{Persentasi Keberhasilan} = \frac{187}{200} X\ 100\%$$

$$\text{Persentasi Keberhasilan} = 93{,}5\%$$