

## **BAB IV**

### **PEMBAHASAN DAN HASIL**

#### **4.1 Pendahuluan**

Dalam bab ini akan dilakukan pengujian perangkat keras dan perangkat lunak yang telah dirancang, yang kemudian dilakukan analisa dari hasil yang telah didapatkan. Pengujian dibagi menjadi 2 tahapan yaitu pengujian perangkat keras yang meliputi rangkaian sistem dan pengujian perangkat lunak yang meliputi algoritma sistem. Hasil dari pengujian yang telah dilakukan dapat dilihat pada sub bab berikut ini. Pengujian ini dilakukan untuk mengetahui apakah alat telah beroperasi dengan baik dan untuk menunjukkan bahwa alat telah bekerja sesuai dengan yang diharapkan. Tujuan dilakukan pengujian adalah sebagai berikut :

1. Mengetahui prinsip kerja rangkaian yang telah dibuat,
2. Menganalisa seluruh data yang diperoleh dari rangkaian sistem dengan metode yang digunakan.

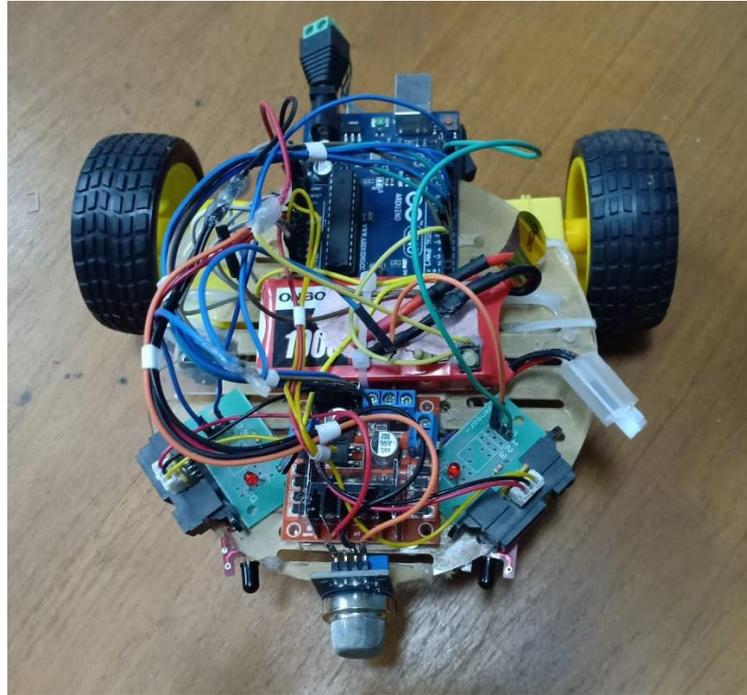
Langkah - langkah yang dilakukan untuk pengujian terhadap alat ini adalah sebagai berikut :

1. Melakukan pengujian perblok rangkaian untuk memperoleh data awal dan mempermudah analisis rangkaian,
2. Melakukan analisa dari data hasil pengujian.

#### **4.2 Pengujian Modul Hardware**

Pengujian modul perangkat keras dilakukan pada semua modul yang digunakan, yaitu mikrokontroler Arduino Atmega328, motor dc, sensor api, sensor gas dan komunikasi dengan raspberry dan untuk mengetahui kinerja dari

setiap modul tersebut. Gambar 4.1 berikut merupakan gambar hardware robot mobile pendeteksi api.

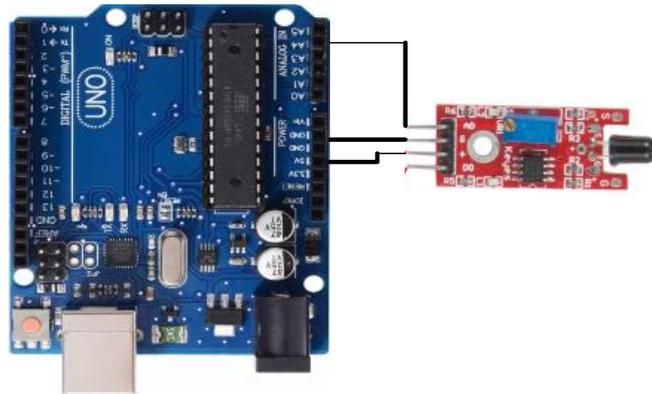


**Gambar 4. 1** Robot mobile pendeteksi api

## **4.2 .1 Pengujian Sensor Api**

### **4.2.1.1 Langkah Pengujian**

Berikut ini pengujian sensor api (flame sensor) menggunakan api dari korek gas pada lantai satu. Pengujian dilakukan dengan menyalakan api lilin tepat didepan robot Gambar 4.2 adalah rangkaian pengujian sensor api. Dalam pengambilan data pengujian sensor, api lilin diletakkan didepan sensor, keluran dari sensor yang merupakan tegangan analog akan dikonversikan terlebih dahulu dalam data biner dengan lebar data 10 bit. Setiap sensor diproses dalam pin ADC pada mikrokontroler dengan nilai konversi 10 bit. Nilai konversi mulai dari 0 desimal sampai dengan 1023 desimal.



**Gambar 4. 2** Rangkaian Pengujian Sensor Api

```
int pinPot = A3; // pin untuk menerima sinyal analog
int pinPot1 = A4; // pin untuk menerima sinyal analog
int sensorGas = 0; // variabel untuk menyimpan nilai
int sensorApi = 0; // variabel untuk menyimpan nilai

void setup()
{
  Serial.begin(9600); // setup koneksi serial
}

void loop()
{
  sensorGas = analogRead(pinPot);
  sensorApi = analogRead(pinPot1);
  Serial.println("sensor_1= ");
  Serial.println(sensorGas);
  Serial.println("sensor_2= ");
  Serial.println(sensorApi);
  delay(1000);
}
```



**Gambar 4. 3** Rangkaian Pengujian Sensor Api

Dalam gambar 4.3, sensor api dihubungkan pada pin A4. Hal ini dikarenakan keluaran sensor api merupakan tegangan analog. Sehingga data harus dikonversikan dulu kedalam sinyal digital.

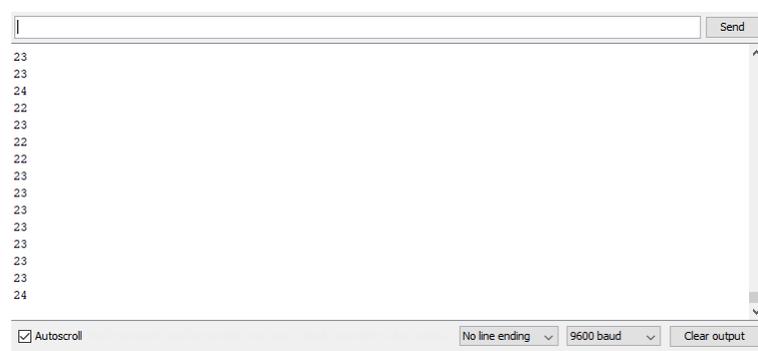
#### 4.2.1.2 Hasil Pengujian

Data hasil dari pengujian sensor api dapat dilihat dalam gambar 4.4 dibawah ini.



**Gambar 4. 4** Pengujian tanpa api

Dalam gambar tersebut lebar data konversi yang digunakan adalah 10 bit (1023 desimal). Dalam pengujian tersebut, sensor api belum mendeteksi adanya api, sehingga data yang ditampilkan merupakan data konversinya saja, dengan data maksimal 1023 desimal. Pengujian sensor api dilakukan terhadap sensor api dengan tepat di depan mobile robot. Data deteksi dapat dilihat dalam gambar 4.5 berikut.



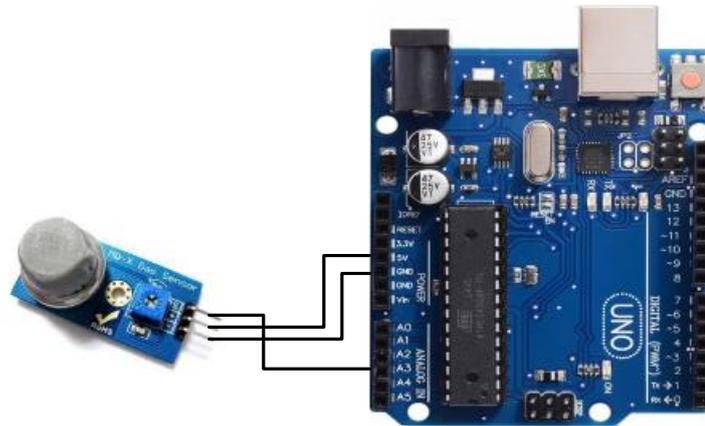
**Gambar 4. 5** Pengujian Sensor api

Dalam gambar tersebut, jarak pengujian sensor api adalah 10cm sampai dengan 20cm. Data pengujian yang diperoleh sensor berkisar antara 22 desimal sampai dengan 24 desimal. Hal ini disebabkan oleh perbedaan kalibrasi antara kedua sensor atau deteksi kecerahan terhadap api lilin yang berbeda. Namun secara garis besar kedua sensor telah berfungsi dengan baik.

## 4.2.2 Pengujian Sensor Gas

### 4.2.2.1 Langkah Pengujian

Pengujian Sensor Gas dilakukan untuk mengetahui apakah sensor bekerja dengan baik, sebelum digunakan secara langsung ke lingkungan. Pengujian dilakukan dengan memberikan gas LPG dari penyala api gas tepat di depan sensor. Sensor diletakkan sesuai dengan perancangan yaitu ditengah mekanik robot. Gambar 4.6 merupakan rangkaian pengujian sensor gas.



**Gambar 4. 6** Rangkaian Pengujian sensor gas

Dalam pengambilan data pengujian sensor, Zipo diletakkan didepan sensor, keluran dari sensor yang merupakan tegangan analog akan dikonversikan terlebih dahulu dalam data biner dengan lebar data 10 bit. Setiap sensor diproses dalam pin ADC pada mikrokontroler dengan nilai konversi 10 bit. Nilai konversi mulai dari 0 desimal sampai dengan 1023 desimal. Gambar 4.19 berikut merupakan program pengujian sensor gas.

```

int pinPot = A3; // pin untuk menerima sinyal analog
int pinPot1 = A4; // pin untuk menerima sinyal analog
int sensorGas = 0; // variabel untuk menyimpan nilai
int sensorApi = 0; // variabel untuk menyimpan nilai

void setup()
{
  Serial.begin(9600); // setup koneksi serial
}

void loop()
{
  sensorGas = analogRead(pinPot);
  sensorApi = analogRead(pinPot1);
  Serial.println("sensor_1= ");
  Serial.println(sensorGas);
  Serial.println("sensor_2= ");
  Serial.println(sensorApi);
  delay(1000);
}

```

**Gambar 4. 7** Rangkaian Pengujian Sensor gas

#### 4.2.2.2 Hasil Pengujian

Data hasil dari pengujian sensor gas dapat dilihat dalam tabel 4.1 dibawah ini. Data sensor gas yang diuji adalah diukur dari jarak 1cm sampai dengan 20 cm. Data sensor yang akan diproses hanya menyatakan terdeteksi dan tidak terdeteksi. Data hasil pengukuran dapat dilihat dalam tabel-tabel dibawah ini

**Tabel 4. 1** Tabel hasil pengujian sensor gas

No	Jarak gas dari sensor (cm)	Data pengujian (desimal)
1	1	245
2	5	230
3	10	170
4	15	120
5	20	65

Dalam tabel diatas, pengujian jarak yang diuji adalah 1 cm sebagai jarak minimal dan 20 cm sebagai jarak maksimal. Pengambilan data dilakukan sebanyak 5 kali dengan jarak yang berbeda yaitu 1 cm, 5 cm, 10 cm, 15 cm dan 20

cm. Adapun data yang terbesar didapat pada jarak 1 cm dan data terkecil pada jarak 20 cm. jadi semakin jauh dari sumbergas maka, data yang diperoleh semakin kecil.

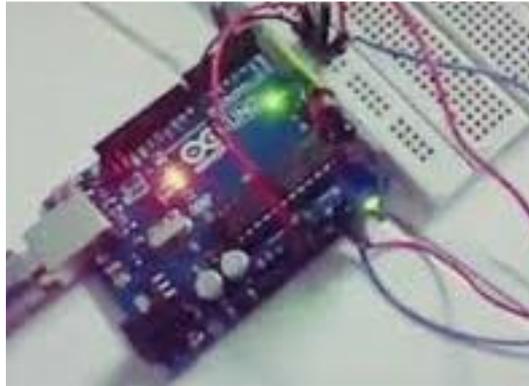
## **4.2. Pengujian Mikrokontroler Atmega328**

### **4.2.3.1 Langkah Pengujian**

Pengujian pada modul mikrokontroler dilakukan dengan mengeset beberapa pin menjadi port input dan port output. Pengujian Arduino tidak hanya dilakukan pada pengukuran port saja tetapi di uji dengan program langsung. Pengujian ini menunjukkan hal yang paling sederhana yang dapat dilakukan dengan Arduino untuk melihat *output* fisik yaitu membuat lampu LED menjadi berkedip. Untuk melakukan pengujian ini, ada beberapa hardware yang dibutuhkan yaitu :

1. LED
2. Arduino Papan
3. Kabel Usb

Untuk membangun sirkuit, hubungkan ke pin-13. Lalu pasang kaki panjang LED (kaki positif, disebut anoda) ke resistor, pasang kaki pendek (kaki negatif, disebut katoda) ketanah. Kemudian pasang papan Arduino ke komputer ,mulai program Arduino, dan masukkan kode di bawah ini. Kebanyakan papan Arduino sudah memiliki LED terpasang ke pin 13 di papan itu sendiri. Gambar 4.8 merupakan pengujian arduino dan Gambar 4.9 merupakan program pengujian LED pada arduino.



**Gambar 4. 8** Rangkaian pengujian LED pada arduino

```

Berkas  Sunting  Sketch  Alat  Bantuan
Blink$
void setup()
{
  pinMode(13, OUTPUT); //pin 13 dijadikan output
}

void loop()
{
  digitalWrite(13, HIGH); // nyalakan LED 13
  delay(1000);           // waktu tunda
  digitalWrite(13, LOW); // matikan LED 13
  delay(1000);           // waktu tunda
}

```

**Gambar 4. 9** Program pengujian LED pada arduino

#### 4.2.3.2 Analisa Hasil Pengujian

Hal pertama yang saya lakukan dalam program ini adalah menginisialisasi pin 13 sebagai pin output dengan garis pin *Mode (13, Output)*;

- Dalam *loop* utama, Saya menhidupkan LED dengan baris:
- *Digital Write (13, Tinggi)*,
- Untuk memasok 5 volt ke pin 13. Itu menciptakan perbedaan tegangan pin dari LED, dan lampu. Kemudian saya mematikannya dengan baris:
- *Digital Write (13, Rendah)*;

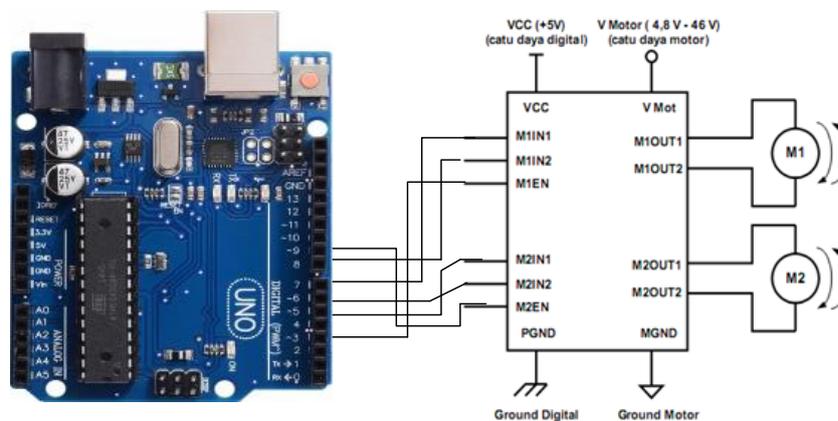
Itu membutuhkan pin 13 kembali ke 0 volt, dan ternyata LED *off*. Di antara *on* dan *off*, Agar ada cukup waktu bagi seseorang untuk melihat perubahan,

sehingga `delay ()` perintah memberitahu Arduino untuk melakukan apa-apa untuk 1000 milidetik, atau satu detik. Bila anda menggunakan `delay ()` perintah, tidak ada yang lain terjadi untuk jumlah waktu. Setelah Anda memahami contoh dasar, memeriksa contoh *Blink Without Delay* untuk belajar cara membuat penundaan sementara melakukan hal-hal lain. Gambar 4.11 merupakan program pengujian sistem minimum arduino.

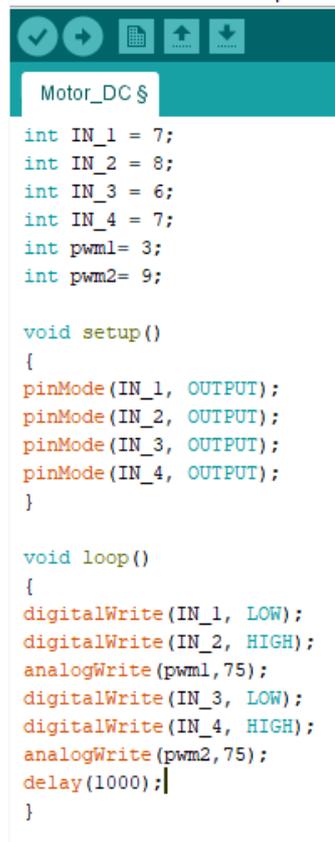
## 4.2.4 Pengujian Driver Motor

### 4.2.4.1 Langkah Pengujian

Pada pengujian ini, Ada 2 macam metode yang digunakan, metode pertama adalah dengan memberikan logika 1 dan 0 pada setiap pin yang digunakan. Hal ini bertujuan untuk mengetahui kondisi perputaran motor dc dan sistem driver, apakah berfungsi dengan baik. Metode ke dua adalah pengaturan kecepatan motor dc, dengan memberikan nilai pwm tertentu pada ke dua pin masukkan ENABLE driver. Motor akan bergerak berdasarkan data pwm yang diberikan sebagai pengaturan kecepatan melalui pin enable. Gambar 4.10 berikut merupakan rangkaian pengujian driver motor.



**Gambar 4. 10** Rangkaian Pengujian driver motor



```

Motor_DC §
int IN_1 = 7;
int IN_2 = 8;
int IN_3 = 6;
int IN_4 = 7;
int pwm1= 3;
int pwm2= 9;

void setup()
{
  pinMode(IN_1, OUTPUT);
  pinMode(IN_2, OUTPUT);
  pinMode(IN_3, OUTPUT);
  pinMode(IN_4, OUTPUT);
}

void loop()
{
  digitalWrite(IN_1, LOW);
  digitalWrite(IN_2, HIGH);
  analogWrite(pwm1, 75);
  digitalWrite(IN_3, LOW);
  digitalWrite(IN_4, HIGH);
  analogWrite(pwm2, 75);
  delay(1000);
}

```

**Gambar 4. 11** Program Pengujian motor dc

Pada gambar 4.11 diatas, terdapat dua bagian driver motor yang terhubung langsung ke mikrokontroler. Hubungannya adalah sebagai berikut,

1. M1EN ke PIN 9 (pwm1) dan M2EN ke PIN 3(pwm2)
2. M1\_IN\_1 ke PIN 7 dan M1\_IN\_2 ke PIN 8 untuk motor M1
3. M2\_IN\_1 ke PIN 5 dan M2\_IN\_2 ke PIN 8 untuk motor M2

Gambar 4.13 berikut adalah portongan program pengujian motor dc.

#### 4.2.4.2 Hasil Pengujian

Pengujian aktivasi motor dc dilakukan hanya untuk memastikan bahwa motor berfungsi dengan baik, yaitu dengan memberikan logika tinggi (1) dan rendah (0) pada pin M1EN dan M2EN dirangkaian driver. Dalam pengujian, terdapat 2 logika motor yaitu motor kanan dan motor kiri. Kedua driver motor

diatur oleh sebuah pin *Enable* yang masing- masing diwakili oleh M1EN (motor kiri) dan M2EN (motor kanan). Kondisi logika untuk input motor yaitu

- a.  $M1\_IN\_1 = 1$ , dan  $M1\_IN\_2 = 0$
- b.  $M2\_IN\_1 = 1$  dan  $M1\_IN\_2 = 0$

Sehingga sama untuk semua pergerakan, yang menjadi perbedaannya adalah hanya logika *Enable*-nya. Tabel 4.2 berikut merupakan tabel hasil pengujian metode aktivasi driver motor.

**Tabel 4. 2** Tabel pengujian aktivasi driver motor

Motor Kanan			Motor Kiri			Keterangan
M2EN	M1_IN_2	M2_IN_1	M1EN	M1_IN_2	M1_IN_1	
1	0	1	1	0	1	maju
1	0	1	0	0	1	belok kiri
0	0	1	1	0	1	belok kanan
0	0	1	0	0	1	berhenti

Pengaturan pergerakan motor adalah sebagai berikut,

1. Jika M1EN dan M2EN berlogika 1, maka robot akan bergerak maju,
2. Jika M1EN berlogika 1 dan M2EN berlogika 0, maka robot akan belok kiri,
3. Jika M1EN berlogika 0 dan M2EN berlogika 1 maka robot akan belok kanan,
4. Jika M1EN dan M2EN berlogika 0, maka robot akan berhenti.

#### 4.2.5 Pengujian PWM Motor

Metode berikutnya adalah dengan mengatur kecepatan motor dc menggunakan pengaturan lebar pulsa tegangannya (PWM). Pengaturan pwm yaitu dengan memberikan nilai tertentu sebesar dari 0 s.d 255 pada pin *Enable*. Untuk menentukan setiap pergerakannya, motor dc diatur dalam dua kondisi pengaturan

pwm, yaitu pwm dengan nilai 90 (**Maju**). Untuk mendapatkan nilai tersebut maka dapat dilakukan perhitungan sebagai berikut.

Jika motor memiliki tegangan maksimal 6Vdc, tegangan sumber 7.4Vdc dan lebar datapwm 8 bit (255des), maka data pwm untuk aktivasi motor secara penuh adalah

$$\begin{aligned} \text{Pwm} &= (6\text{V} / 7.4\text{V}) \times 255 \\ &= 207\text{desimal} \end{aligned}$$

Dengan perhitungan diatas kita dapat menentukan besaran tegangan pwm yang diperlukan untuk aktivasi motor dengan pengaturan **Maju** (90 desimal) adalah sebagai berikut:

$$\begin{aligned} \text{Pwm} &= (x / 7.4) * 255 \\ 90 &= (x / 7.4) * 255 \\ x &= (90 * 7.4) / 255 \\ x &= 2,61\text{Vdc} \end{aligned}$$

dimana x merupakan Vdc pada saat diberi pwm 75



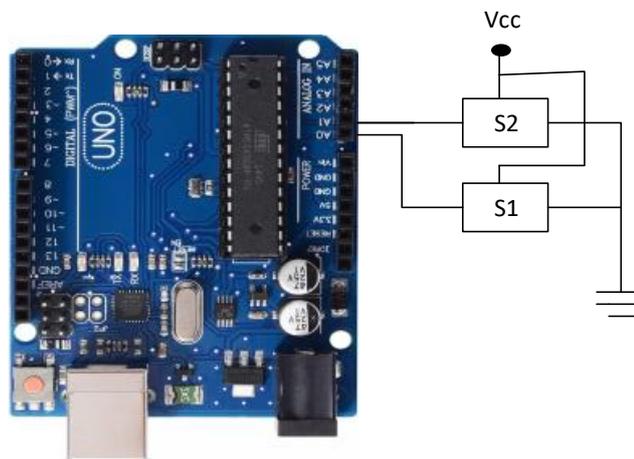
**Gambar 4. 12** Data pengukuran tegangan motor

Dari hasil pengujian untuk data pengukuran dan perhitungan, terdapat selisih pembacaan tegangan data sebesar 0.01Vdc. sehingga error yang diperoleh sebesar 0.38%.

## 4.2.6 Pengujian Sensor Jarak

### 4.2.6.1 Langkah Pengujian Sensor Jarak

Pengujian ini dilakukan untuk menguji apakah sensor berfungsi dengan baik. Gambar 4.13 dibawah merupakan rangkaian pengujian pada sensor Jarak.



**Gambar 4. 13** Rangkaian pengujian sensor Jarak

Pengujian ini dilakukan dengan cara meletakkan halangan tepat didepan sensor. Jarak ini diatur sedemikian agar data jarak yang diperoleh berbeda dalam setiap pengujian, jarak yang ditentukan adalah 10 cm dan 20 cm.

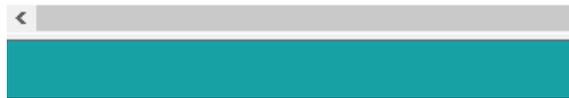
```

int pinPot = A0; // pin untuk menerima sinyal
int pinPot1 = A1; // pin untuk menerima sinyal
int data1 = 0; // variabel untuk menyimpan
int data2 = 0; // variabel untuk menyimpan

void setup()
{
  Serial.begin(9600); // setup koneksi serial
}

void loop()
{
  data1 = analogRead(pinPot); // baca nilai
  data2 = analogRead(pinPot1); // baca nilai
  Serial.println("Data_1= "); // kirim
  Serial.println(data1); // kirim nilai
  Serial.println("Data_2= "); // kirim
  Serial.println(data2); // kirim nilai
  delay(1000);
}

```



**Gambar 4. 14** Program pengujian sensor Jarak

Saat pengambilan data pengujian sensor jarak, objek halangan diletakkan didepan sensor, keluran dari sensor yang merupakan tegangan analog akan dikonversikan terlebih dahulu dalam data biner dengan lebar data 10 bit. Setiap sensor diproses dalam pin ADC pada mikrokontroler dengan nilai konversi 10 bit. Nilai konversi mulai dari 0 desimal sampai dengan 1023 desimal. Karena mikrokontroler memiliki lebar data 10 bit, maka jarak maksimal yang dapat ditampilkan oleh mikrokontroler hanya 10 bit. Dalam perancangan ini nilai konversi ADC diambil mulai dari bit 0 sampai dengan bit 9. Gambar 4.15 merupakan proses pengambilan data pada jarak 10cm.



**Gambar 4. 15** Jarak pengujian objek 10cm

Data sensor merupakan besaran analog yang diubah kedalam data desimal dan biner. Konversi ini dilakukan dengan rumusan sebagai berikut.

$$\text{Konversi ADC} = (\text{Vin} * 1023) / \text{Vdc} \quad (4.1)$$

Dari rumusan di atas dapat dilakukan perhitungan menentukan konversi ADC untuk data jarak jangkauan terjauh dan terdekat dari besaran analog ke dalam data digital. Spesifikasi data konversi adalah sebagai berikut.

Lebar data ADC 10bits = 1023

Jarak jangkauan sensor = 10cm s.d 80cm = 80 – 10 = 70 cm

Vdc =5V (ideal)

Vin max=2.4V (nilai maksimal tegangan sensor)

Konversi untuk data ideal :

$$\begin{aligned} \text{ADC sensor} &= (2,4 * 1023) / 5 \\ &= 491 \end{aligned}$$

Data 491desimal adalah data ini adalah data dari jarak terdekat yaitu 10 cm.

#### 4.2.6.2 Analisa Hasil Pengujian Sensor Jarak

Data sensor jarak yang diuji adalah jarak halangan terdekat dan jarak halangan terjauh. Data sensor yang akan diproses kedalam data bit ke 0 sampai dengan bit ke 9. Data hasil pengukuran dapat dilihat dalam tabel-tabel dibawah ini.

**Tabel 4. 3** Tabel hasil pengujian sensor Jarak 1

Posisi sensor	Jarak (cm)	Tegangan (Volt)	Data Desimal		Error
			Perhitungan	Pengujian	
DEKAT	10	2.4	491	489	2
JAUH	20	1.4	286	284	3

**Tabel 4. 4** Tabel hasil pengujian sensor Jarak 2

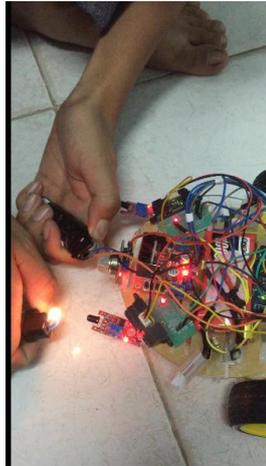
Posisi sensor	Jarak (cm)	Tegangan (Volt)	Data Desimal		Error
			Perhitungan	Pengujian	
DEKAT	10	2.4	492	491	1
JAUH	20	1.4	286	287	1

Tabel 4.3, dan Tabel 4.4 merupakan data hasil pengujian sensor jarak 1 dan sensor jarak 2. Data pengujian tersebut diambil berdasarkan derajat keanggotaan, yang berturut-turut yaitu 10 cm dan 20 cm. Dari hasil pengujian tegangan pada jarak tersebut, maka diperoleh tegangan secara berturut-turut 2.4V dan 1.4V.

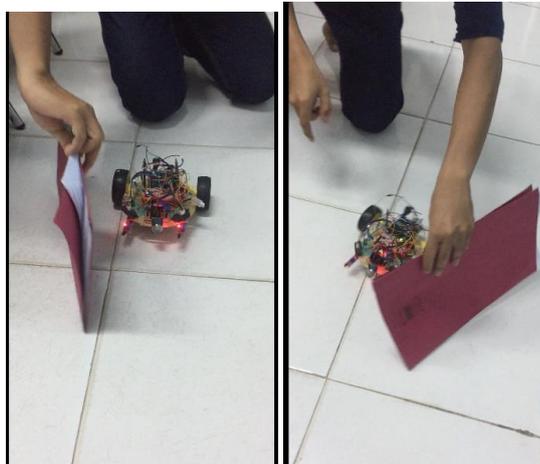
Kemudian, data tegangan tersebut dihitung berdasarkan rumusan 4.1, sehingga didapat hasilnya secara berturut-turut 491 desimal dan 286 desimal. Hal ini berbeda dengan data yang diperoleh dari hasil pengujian secara langsung, dengan meletakkan halangan di depan sensor. Data yang diperoleh untuk hasil pengukuan sensor 1 adalah 492 desimal untuk jarak 10 cm dan 287 desimal untuk jarak 20 cm. Dari data perhitungan dan pengukuran diatas, maka terdapat selisih (error) data untuk setiap jarak pengukuran.

#### 4.2.7 Pengujian Pergerakan Robot

Berikut ini pengujian pergerakan pada robot. Pengujian dilakukan dengan menyalakan Zipo tepat didepan robot Gambar 4.16 adalah rangkaian pengujian sensor api dan sensor gas.



**Gambar 4. 16** Pengujian Sensor Api dan Gas



**Gambar 4. 17** Pengujian Sensor Jarak

Dalam gambar tersebut, Pengujian dilakukan dengan meletakkan halangan tepat didepan robot Gambar 4.17 adalah rangkaian pengujian sensor jarak.