

# Desain dan Uji Coba Sederhana Pada Obstacle Avoiding Robot Menggunakan Mikrokontroler Arduino

Muhammad Zainuddin Lubis\*, Wenang Anurogo\* Perdi Novanto Sihombing#

\* Politeknik Negeri Batam

Jurusan Teknik Informatika, Program Studi Teknik Geomatika  
Batam Center, Jl. Ahmad Yani, Kepulauan Riau 29461, Indonesia

# Politeknik Negeri Batam

Jurusan Teknik Elektro  
Batam Center, Jl. Ahmad Yani, Kepulauan Riau 29461, Indonesia

E-mail: zainuddinlubis@polibatam.ac.id

## Abstrak

Pada tulisan ini membahas tentang desain, konstruksi dan kontrol robot penyeimbang diri dari *obstacle avoiding* robot roda dua. Rangkaian sistem terdiri dari sepasang motor DC dan papan mikrokontroler Arduino, dengan menggunakan sebuah sensor ping HC-SR04 yang merupakan sensor ultrasonik untuk mendeteksi jarak ke rintangan dan dari posisi robot berada. Hasil yang diperoleh dari uji coba menunjukkan nilai jarak terhadap nilai ping (*ping number*) tegak lurus, semakin bertambahnya jarak maka akan semakin tinggi waktu ping yang dihasilkan. Nilai jarak sensor ke objek dan waktu ping paling rendah yaitu 2 cm, dan 132  $\mu$ S, dan yang paling tinggi dengan jarak 20 cm dengan waktu ping yaitu 1160  $\mu$ S. Semakin jauh jarak *obstacle* maka waktu yang diperlukan sensor untuk mengirimkan kembali hasil deteksi juga lebih lama

**Kata kunci:** *Obstacle avoiding robot*, Sensor Ping HC-SR04, Jarak

## Abstract

*This paper discusses the design, construction and control of the self-balancing robot obstacle avoiding the two-wheeled robot. The system circuit consists of a pair of DC motors and an Arduino microcontroller board, using a HC-SR04 ping sensor which is an ultrasonic sensor to detect the distance to the obstacle and from where the located of the robot. The results obtained from the test show the value of distance to the ping value (ping number) perpendicular, the more the distance will increase the higher the ping time generated. The value of the sensor distance to the object and the lowest ping time is 2 cm, and 132  $\mu$ S, and the highest with a distance of 20 cm with a ping time of 1160  $\mu$ S. The further the obstacle distance, the time it takes the sensor to send back the detection result is also longer.*

**Keywords:** *Obstacle avoiding robot*, ping sensor HC-SR04 and Distance

## 1. Pendahuluan

Perkembangan teknologi yang semakin pesat dari masa ke masa menghasilkan daya cipta yang baru untuk bersaing menghasilkan teknologi tepat guna. Misalnya kendaraan, sebelum di produksi untuk diterjukkan ke pasaran, industri produksi akan menciptakan desain sedemikian rupa, dan sederhana dengan membuat hasil *prototype* terlebih dahulu. Prototype adalah pembuatan model sederhana dengan gambaran dasar untuk pengujian awal. Dalam pembuatan kendaraan, tidak hanya desain dan pembuatan model saja, tetapi juga keamanan bagi pengendaranya.

Keamanan dalam kontrol dengan sistem otomatis, akan memudahkan pengemudi untuk

menghindari hal yang tidak diinginkan terjadi. Seperti menghindari suatu objek yang dapat membahayakan keselamatan pengendara. Oleh karenanya, dibutuhkan suatu sensor yang mampu mendeteksi suatu objek, dengan mengukur jarak objek terhadap posisi kendaraan. Dimana sensor tersebut digerakkan oleh motor listrik, serta memiliki processor yang berguna untuk mengendalikan dan menyimpan data. Pada tulisan ini penulis membuat robot *obstacle avoiding* dengan melihat robot-robot sebelumnya yang sudah ada di era sekarang, serta membahas dan menjelaskan tentang perangkat keras robot, desain robot, kontrol dalam menyeimbangkan kedua motor, dan menyajikan hasil deteksi objek dengan melihat waktu dan jarak terhadap sebuah objek, dan waktu terhadap *ping number* beserta jarak dalam uji coba, yang berfungsi sebagai sistem keamanan control

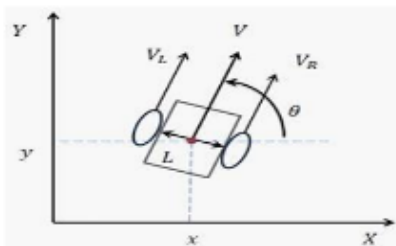
robot dengan penghindaran dari objek secara otomatis.

## 2. Tinjauan Pustaka

Sementara robot yang saat ini mungkin menjadi produk yang komersil dan terkenal, dan penelitian pengendalian mekanis semacam *obstacle avoiding* robot sistemnya sudah beragam. Robot *obstacle avoiding* roda dua merupakan robot dasar yang penting dalam pengendalian pendidikan dan penelitian, misalnya oleh [2]. Selain pengembangan Segway, studi tentang *obstacle avoiding* robot roda dua telah banyak dilakukan, sebagai contoh [3].

### 2.1 Struktur dan Sistem *obstacle avoiding* robot

Struktur dan bentuk *obstacle avoiding* robot roda dua dapat diklasifikasikan, menjadi tiga bagian yaitu: sensor, motor dan kontrol motor, dan menggunakan aplikasi Arduino untuk memasukkan perintah kedalam robot tersebut sesuai yang diinginkan. Aplikasi dan keuntungan dari sensor pada *obstacle avoiding* robot roda dua dapat digunakan untuk memperoleh pengukuran percepatan, jarak tempuh, dan sudut kemiringan robot (*tilt angle*) seperti yang pernah dilakukan pada penelitian [4]. Pada struktur *obstacle avoiding* robot dengan menggunakan model kinematika *obstacle avoiding* robot yang digunakan dalam makalah ini ditunjukkan pada Gambar 1.



Gambar 1. Model kinematika *obstacle avoiding* robot roda dua

Akselerasi roda kiri dan kanan adalah  $\omega_L$  dan  $\omega_R$ . Kami berasumsi bahwa kontak antara roda dan lantai yang terguling murni dan tidak tergelincir itu memiliki hubungan antara *velg* kiri dan kanan dengan kecepatan dan akselerasi adalah sebagai berikut [5]:

$$V_L = r\omega_L \text{ and } V_R = r\omega_R, \quad (1)$$

Dimana  $r$  adalah jari-jari roda. Kecepatan linear robot *obstacle avoiding* adalah  $V$ . Kecepatan linier

untuk roda kiri dan kanan adalah  $v_L$  dan  $v_R$ , Hubungan antara  $\omega_L$ ,  $\omega_R$ ,  $v_L$ ,  $v_R$  adalah sebagai berikut:

$$V = \frac{V_R + V_L}{2} = r \frac{\omega_R + \omega_L}{2} \quad \omega = \frac{V_R - V_L}{L} = r \frac{\omega_R - \omega_L}{L} \quad (2)$$

Kini kita bisa meringkas model dinamis untuk *obstacle avoiding* robot, adalah sebagai berikut:

$$x' = V \cos \theta, \quad y' = V \sin \theta, \quad \theta' = \omega,$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} V \\ \omega \end{bmatrix} \quad (3)$$

### 2.2 Sensor Jarak

Sensor jarak adalah modul pengukur jarak dengan ultrasonik. Biasa disebut juga dengan sensor ultrasonik PING Parallax, yang didesain khusus untuk teknologi robotika [6]. Sensor jarak yang digunakan memiliki ukuran yang cukup kecil (2,1cm x 4,5cm), sensor ini dapat mengukur jarak antara 3 cm sampai 300 cm [7].

Suara ultrasonik dengan frekuensi sebesar 40 KHz akan dipancarkan selama 200 $\mu$ S. Suara ini akan merambat di udara dengan kecepatan 344.424m/detik (atau 1cm setiap 29.034 $\mu$ S), lalu mengenai objek untuk kemudian terpantul kembali ke sensor [8]. Selama menunggu pantulan, sensor akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (I) ketika suara pantulan terdeteksi oleh sensor. Oleh karena itulah lebar pulsa tersebut dapat merepresentasikan jarak antara sensor dengan objek. Berdasarkan Data Sheet:

Jarak = (Lebar pulsa / 29.034  $\mu$ S) / 2 - dalam cm, dan  
 Jarak = (Lebar Pulsa / 0.034442) / 2 - dalam cm,  
 karena  $1 / 29.034 = 0.34442$ .

Untuk melakukan perhitungan jarak real menggunakan sensor jarak dapat dilakukan kalibrasi pada setiap sensor yang akan digunakan. Karena sifatnya setiap sensor jarak memiliki hasil jarak yang berbeda.

### 2.2. Arduino

Arduino merupakan rangkaian elektronik yang bersifat *open-source*, serta memiliki perangkat keras dan lunak yang mudah untuk digunakan [9].

Arduino dapat mengenali lingkungan sekitarnya melalui berbagai jenis sensor dan dapat mengendalikan lampu, motor, dan berbagai jenis aktuator lainnya. Arduino mempunyai banyak jenis, di antaranya adalah Arduino Nano, Arduino BT, LilyPad Arduino, Arduino Duemilanove, dan lainnya ([www.arduino.cc](http://www.arduino.cc)).

Arduino adalah *board* mikrokontroler yang dibuat untuk memudahkan dalam membangun berbagai aplikasi elektronik. Arduino merupakan salah satu *hardware* yang bersifat *open source* yang mulai dikembangkan pada tahun 2005 [10]. Untuk pemrogramannya sendiri melalui *software* khusus yaitu *Integrated Development Environment* (IDE) Arduino yang dibangun menggunakan bahasa pemrograman Java oleh *Massachusetts Institute of Technology*. Arduino Uno memiliki dimensi ukuran panjang 68.6 mm dan lebar 53.4 mm dengan keluaran tegangan 5 volt dan keluaran arus 20 mA untuk I/O Pin dan 50 mA untuk 3.3V pin. Tegangan masukan pada arduino 7-12 Volt, dan menggunakan mikrokontroler Atmega 328P dengan Flash Memory 32 kb (Gambar 2).



Gambar 2. Arduino Uno ([www.arduino.cc](http://www.arduino.cc))

### 3. Metode Penelitian

#### 3.1 Alat dan Bahan

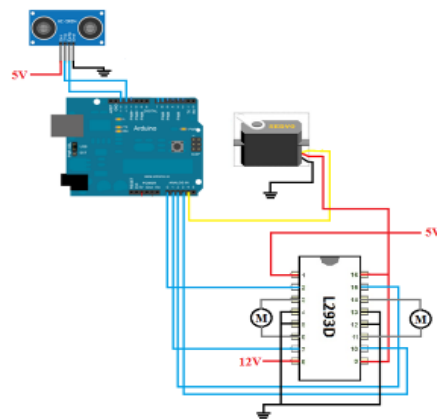
Berikut merupakan alat dan bahan yang digunakan dalam pembuatan alat rancang bangun akustik pasif pendeteksi gelombang bawah air adalah sebagai berikut:

1. Printed Circuit Board (PCB)
2. Arduino Uno
3. Sensor PING HC-SR04
4. Motor DC
5. Motor Servo

6. Power Supply 12 Volt
7. IC LM-7805
8. Battery 9 Volt
9. Akrilik
10. Kabel Jumper

#### 3.2 Perancangan Perangkat Keras

Skematis dari rangkaian alat *Obstacle Avoiding Robot* roda dua menggunakan sensor ultrasonik dapat dilihat pada Gambar 3. Masing – masing perangkat terhubung ke pin arduino dan tegangan supply (Gambar3).



Gambar 3. Skematis rangkaian *Obstacle Avoiding Robot* roda dua

#### 3.3 Teknik Pengujian

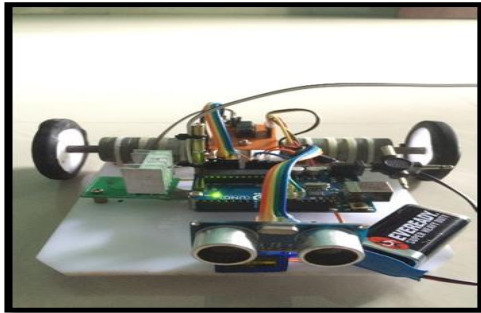
Teknik pengujian yang akan dilakukan pada penelitian ini adalah melakukan uji fungsional pada alat yang dibuat terlebih dahulu, setelah alat masuk dalam kriteria kelayakan, dan dapat bekerja secara normal, kemudian akan diberikan objek sebagai pengujian pendeteksian keselamatan dari robot tersebut. Data hasil deteksi yang dalam proses pengujian akan menampilkan hasil deteksi pada serial number program Arduino.

### 4. Hasil dan Pembahasan

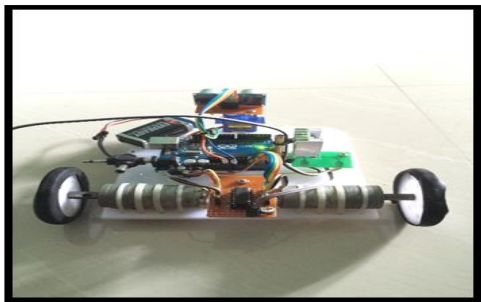
#### 4.1 Hasil Prototype

Pada hasil uji coba yang dilakukan pada *Obstacle Avoiding Robot* roda dua, dengan Skematis dari rangkaian alat *Obstacle Avoiding Robot* roda dua yang menggunakan sensor ultrasonik (Gambar3) memiliki hasil tampak dari depan, belakang, kanan,

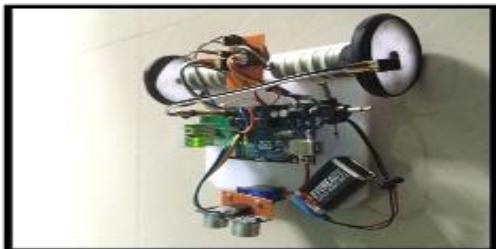
dan kiri yang dapat dilihat pada Gambar 4 (a,b,c,dan d).



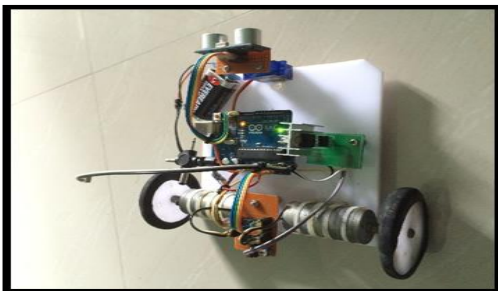
Gambar 4a. Tampak depan



Gambar 4b. Tampak belakang



Gambar 4c. Tampak kanan



Gambar 4d. Tampak kiri

#### 4.2 Perancangan Perangkat Lunak

```
#include <NewPing.h>
#include <Servo.h>

#define TRIG_PIN 12
#define ECHO_PIN 13
#define MAX_DISTANCE 200

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);

Servo myservo;
```

```
boolean goesForward=false;

int distance;

int speedSet = 0;

const int motor1Pin1 = 14;
const int motor1Pin2 = 15;
//Motor B
const int motor2Pin1 = 17;
const int motor2Pin2 = 16;

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

  Serial.println ("Obstacle Details:");
  Serial.print ("Distance From Robot is ");
  Serial.print ( distance);
  Serial.print ( " CM!"); // print out the distance in centimeters.

  moveStop();
  delay(100);
  moveBackward();
  delay(300);
  moveStop();
  delay(200);
  distanceR = lookRight();
  delay(1000);
  distanceL = lookLeft();
  delay(1000);
  if(distanceR>=distanceL)
  {
    turnRight();
    moveForward();
  }else
  {
    turnLeft();
    moveForward();
  }
}
else
{
  moveForward();
  Serial.println ("No obstacle detected. going forward");
  delay (15);
}
distance = readPing();
}
```

```

int lookRight()
{
  myservo.write(0);
  delay(500);
  int distance = readPing();
  delay(100);
  myservo.write(90);
  return distance;
}

int lookLeft()
{
  myservo.write(180);
  delay(500);
  int distance = readPing();
  delay(100);
  myservo.write(90);
  return distance;
  delay(100);
pinMode(motor1Pin1, OUTPUT);
pinMode(motor1Pin2, OUTPUT);
pinMode(motor2Pin1, OUTPUT);
pinMode(motor2Pin2, OUTPUT);

myservo.attach(18);
myservo.write(90);
delay(2000);
distance = readPing();
delay(100);
}

void loop() {

moveForward();
int distanceR = 0;
int distanceL = 0;
delay(40);

if(distance<=20)
{
}

int readPing() {
  delay(70);
  int cm = sonar.ping_cm();
  if(cm==0)
  {

```

```

    cm = 250;
  }
  return cm;
}

void moveStop() {
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, LOW);
}

void moveForward() {
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, HIGH);
}

void moveBackward() {
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
}

}

void turnRight() {
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  digitalWrite(motor2Pin1, LOW);
  digitalWrite(motor2Pin2, HIGH);
  delay(300);
  moveForward();
}

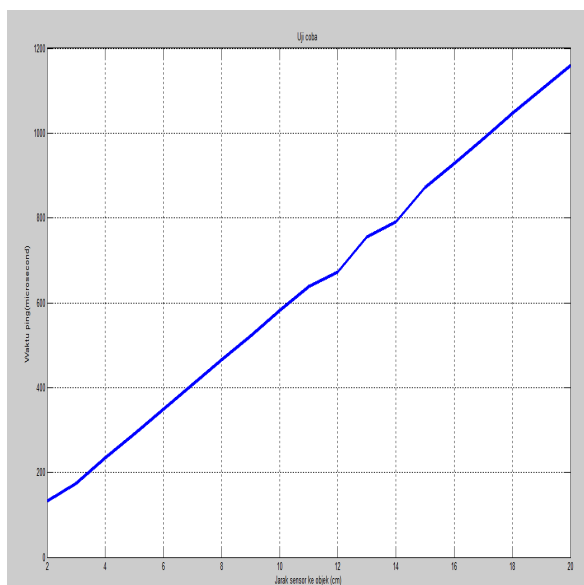
void turnLeft() {
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  digitalWrite(motor2Pin1, HIGH);
  digitalWrite(motor2Pin2, LOW);
  delay(300);
  moveForward();
}

```

### 4.3 Hasil Uji Coba

Pada hasil uji coba alat, dalam system kerja alat, pada tahap awal uji coba dimulai dengan pembacaan sensor. Suatu kondisi ketika sensor tidak membaca adanya objek dalam jarak maksimal 20 cm, maka data akan masuk kedalam tampilan di Arduino, dan kemudian data akan diproses. Selama tidak terdapat objek yang terdeteksi, maka alat akan bergerak maju sampai alat mendeteksi objek rintangan selanjutnya dalam proses pengujian.

Ketika sensor mendeteksi suatu objek dalam jarak 20 cm yang berada didepannya, maka data tersebut masuk ke Arduino lalu diproses dan Arduino akan menggerakkan motor servo ke kiri  $90^0$  dan ke kanan  $90^0$  kemudian kembali ke posisi semula. Apabila objek rintangan terletak disebelah kanan, maka data tersebut tersimpan di *memory* Arduino lalu diproses, proses ini akan mengirimkan sinyal untuk menggerakkan motor DC ke kiri untuk mundur dari objek yang terdeteksi. Kemudian bergerak maju sampai alat mendeteksi objek rintangan lainnya. Dan begitu juga sebaliknya, apabila objek rintangan pada sebelah kiri. Gambar grafik hasil uji coba pada deteksi pengamanan berbasis sensor ultrasonik dapat dilihat pada Gambar 5.



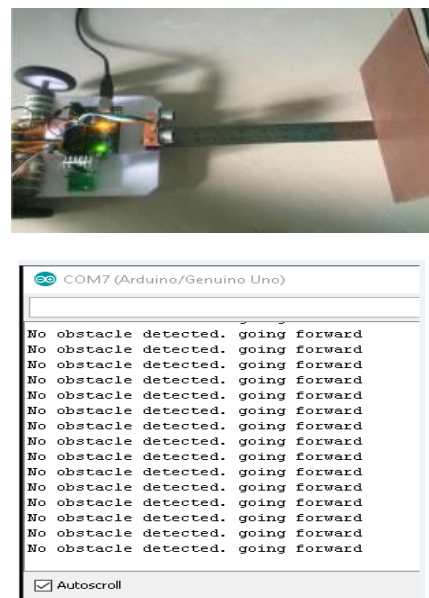
Gambar 5. Uji coba alat (jarak vs ping number)

Hasil uji coba alat ini hanya dilakukan sebanyak 1 kali dalam proses perlakuan terhadap jarak deteksi

(Gambar 5). *Ping number* adalah banyaknya data yang terdeteksi pada hasil yang ditunjukkan di display Arduino. Hasil foto sebenarnya dapat dilihat dengan gambar yang menggunakan penggaris (merupakan validasi jarak dalam deteksi objek), dengan jarak yang dideteksi oleh sensor ultrasonik yang akan ditampilkan di data Arduino. Pada data jarak setelah robot berhenti, ketika mendeteksi adanya *obstacle* didepannya dengan jarak 20 cm dan lebih dari 20 cm, dapat dilihat pada Gambar 6 dan 7.



Gambar 6. Jarak setelah robot berhenti (20 cm)



Gambar 7. Jarak setelah robot berhenti (Lebih dari 20 cm)

Dari Gambar 6 dan 7 jelas terlihat hasil pengujian menggunakan robot roda dua yang berfungsi untuk menghindari bahaya atau rintangan dengan menggunakan sensor *ultrasonic* memiliki maksimum jarak deteksi target adalah 20 cm. Hasil pada Gambar 5 menunjukkan hasil yang tegak lurus dengan semakin bertambahnya jarak maka akan semakin tinggi waktu ping yang di hasilkan. Nilai jarak sensor ke objek dan waktu ping paling rendah yaitu 2 cm, dan 132  $\mu$ S, dan yang paling tinggi dengan jarak 20 cm dengan waktu ping yaitu 1160  $\mu$ S. Dari hasil uji coba ini dapat dikatakan bahwa semakin jauh jarak *obstacle* maka waktu yang diperlukan sensor untuk mengirimkan kembali hasil deteksi juga lebih lama, dan perlu adanya peningkatan jarak maksimum deteksi objek pada sensor yang digunakan. Secara umum pembangunan robot dikatakan telah berhasil, dengan melihat hasil yang sudah terdeteksi terhadap jarak dan waktu dalam proses uji coba robot.

## 5. Kesimpulan

Alat berfungsi dengan optimal, hanya saja sensor bergerak  $90^0$  ke kiri, dan  $90^0$  ke kanan, sehingga, apabila ada objek di bawah dan diatas alat, maka sensor tidak dapat membaca objek tersebut. Perbaikan desain untuk gerakan penyeimbang yang lebih baik memerlukan optimalisasi desain mekanis seperti merelokasi pusat massa, penempatan sensor yang lebih baik, modifikasi filter komplementer untuk menolak gangguan akibat gerakan translasi, dan desain yang kuat. Dalam penelitian selanjutnya diharapkan dapat menggunakan penambahan sensor ultrasonik, dan maksimal jarak deteksi sensor yang lebih tinggi dibandingkan dengan penelitian ini.

## Referensi

- [1] Araujo, A., Portugal, D., Couceiro, M. S., & Rocha, R. P. 2015. Integrating Arduino-based educational mobile robots in ROS. *Journal of Intelligent & Robotic Systems*, 77(2), 281-298
- [2] Xu, K., & Duan, X. D. 2002. Comparative study of control methods of single-rotational inverted pendulum. In *Machine Learning and Cybernetics*, 2002. Proceedings. 2002 International Conference on (Vol. 2, pp. 776-778). IEEE.
- [3] Rezaee, H., & Abdollahi, F. 2014. A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots. *IEEE Transactions on Industrial Electronics*, 61(1), 347-354.
- [4] Montiel, O., Orozco-Rosas, U., & Sepúlveda, R. 2015. Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles. *Expert Systems with Applications*, 42(12), 5177-5191.
- [5] Li, X., & Choi, B. J. 2013. Obstacle avoidance of mobile robot by fuzzy logic system. *ISA, ASTL*, 21, 244-246.
- [6] Arief, U. M. 2013. Pengujian sensor ultrasonik ping untuk pengukuran level ketinggian dan volume air. *Edu Elekrika Journal*, 1(2).
- [7] Rismawan, T., Permana, A., & Triyanto, D. 2015. Rancang bangun sistem monitoring volume dan pengisian air menggunakan sensor ultrasonik berbasis mikrokontroler avr atmega8. *Jurnal Coding Sistem Komputer Universitas Tanjungpura*, 3(2).
- [8] Githa, D. P., & Swastawan, W. E. 2014. Sistem Pengaman Parkir dengan Visualisasi Jarak Menggunakan Sensor PING dan LCD. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 3(1), 10-14.
- [9] Patiung, F. T., Lumenta, A. S., Sompie, S. R., & Sugiarso, B. A. 2013. Rancang Bangun Robot Beroda dengan Pengendali Suara. *E-JOURNAL TEKNIK ELEKTRO DAN KOMPUTER*, 2(4), 48-52.
- [10] Banzi, M., & Shiloh, M. 2014. Getting started with Arduino: the open source electronics prototyping platform. *Maker Media, Inc.*
- [11] Phillipe (ed), *Title of paper*, Name of Proc., Publisher (1998).