

# LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya.

## **Pengembangan untuk simulasi sensor jarak (*Ultrasonic*)**

*Adinda Adhwa Nisrina Hanan*

*Jurusan Teknologi Informasi*

*isthatadhindhanan@gmail.com*

### **ABSTRAK**

Sensor ultrasonik banyak digunakan dalam berbagai aplikasi, seperti robotika, otomasi industri, dan sistem parkir. Praktikum ini bertujuan untuk mensimulasikan sensor jarak berbasis ultrasonik guna memahami prinsip kerja dan implementasinya dalam pemrograman mikrokontroler. Dengan menggunakan perangkat lunak simulasi, seperti Proteus atau Tinkercad, serta pemrograman dengan Arduino IDE, eksperimen ini dilakukan untuk mengukur dan menganalisis jarak objek secara virtual. Hasil dari simulasi ini diharapkan dapat memberikan pemahaman yang lebih mendalam tentang penggunaan sensor ultrasonik dalam berbagai aplikasi teknologi.

#### **1.1 Latar Belakang**

Sensor ultrasonik bekerja berdasarkan prinsip pemantulan gelombang suara frekuensi tinggi. Gelombang ini dikirim oleh pemancar dan diterima kembali oleh penerima setelah mengenai suatu objek. Waktu tempuh gelombang digunakan untuk menghitung jarak objek dari sensor. Teknologi ini banyak digunakan dalam navigasi robot, alat bantu parkir, dan sistem pengukuran otomatis. Dalam praktikum ini, simulasi dilakukan untuk memahami cara kerja sensor ultrasonik serta cara mengolah data dari sensor menggunakan pemrograman mikrokontroler.

## **2.2 Tujuan**

- Memahami prinsip kerja sensor ultrasonik dalam mengukur jarak.
- Mengimplementasikan simulasi sensor ultrasonik menggunakan perangkat lunak.
- Mengembangkan program untuk membaca dan menampilkan data jarak dari sensor.
- Menganalisis hasil pengukuran sensor dalam kondisi simulasi.

## **3. Methodology (Metodologi)**

### **3.1 Tools & Materials (Alat dan Bahan)**

Dalam praktikum ini, alat dan bahan yang digunakan meliputi:

- Perangkat Keras:
  1. Sensor Ultrasonik HC-SR04 (opsional untuk uji coba nyata)
  2. Arduino Uno (opsional untuk pengujian langsung)
- Perangkat Lunak:
  1. Arduino IDE untuk pemrograman
  2. Proteus/Tinkercad untuk simulasi
  3. Library Ultrasonic (untuk membaca data dari sensor)

### **3.2 Implementation Steps (Langkah Implementasi)**

#### **1) Menyiapkan Lingkungan Simulasi**

- a) Menggunakan Proteus atau Tinkercad untuk merancang rangkaian sensor ultrasonik dengan Arduino.

#### **2) Menyusun Rangkaian Elektronik**

- a) Menghubungkan sensor HC-SR04 ke Arduino:
  - i) VCC ke 5V Arduino
  - ii) GND ke GND Arduino
  - iii) Trig ke Pin Digital (misal Pin 9)
  - iv) Echo ke Pin Digital (misal Pin 10)

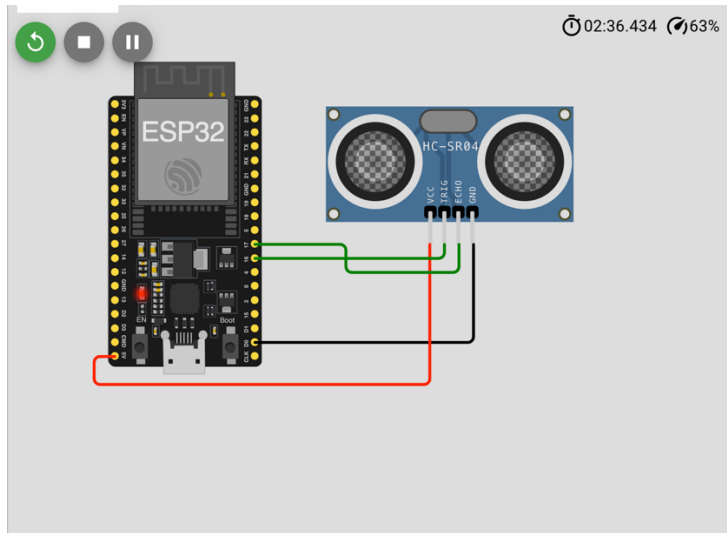
#### **3) Menulis dan Mengunggah Program**

- a) Menggunakan Arduino IDE untuk menulis kode yang membaca data sensor dan menampilkan hasil ke serial monitor.

#### 4) Menjalankan Simulasi

- a) Mengamati hasil pengukuran jarak yang ditampilkan dalam perangkat lunak simulasi.

### 3. Experimental Results (Hasil Eksperimen)



## 1. Apendix

```
1  const int trigPin = 5;
2  const int echoPin = 18;
3
4
5  //define sound speed in cm/uS
6  #define SOUND_SPEED 0.034
7  #define CM_TO_INCH 0.393701
8
9
10 long duration;
11 float distanceCm;
12 float distanceInch;
13
14
15 void setup() {
16   Serial.begin(115200); // Starts the serial communication
17   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
18   pinMode(echoPin, INPUT); // Sets the echoPin as an Input
19 }
20
21
22 void loop() {
23   // Clears the trigPin
24   digitalWrite(trigPin, LOW);
25   delayMicroseconds(2);
26   // Sets the trigPin on HIGH state for 10 micro seconds
27   digitalWrite(trigPin, HIGH);
28   delayMicroseconds(10);
29   digitalWrite(trigPin, LOW);
30   // Reads the echoPin, returns the sound wave travel time in microseconds
31   duration = pulseIn(echoPin, HIGH);
32   // Calculate the distance
33   distanceCm = duration * SOUND_SPEED/2;
34   // Convert to inches
35   distanceInch = distanceCm * CM_TO_INCH;
36   // Prints the distance in the Serial Monitor
37   Serial.print("Distance (cm): ");
38   Serial.println(distanceCm);
39   // Serial.print("Distance (inch): ");
40   // Serial.println(distanceInch);
41 }
```

```
6  #define SOUND_SPEED 0.034
7  #define CM_TO_INCH 0.393701
8
9
10 long duration;
11 float distanceCm;
12 float distanceInch;
13
14
15 void setup() {
16   Serial.begin(115200); // Starts the serial communication
17   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
18   pinMode(echoPin, INPUT); // Sets the echoPin as an Input
19 }
20
21
22 void loop() {
23   // Clears the trigPin
24   digitalWrite(trigPin, LOW);
25   delayMicroseconds(2);
26   // Sets the trigPin on HIGH state for 10 micro seconds
27   digitalWrite(trigPin, HIGH);
28   delayMicroseconds(10);
29   digitalWrite(trigPin, LOW);
30   // Reads the echoPin, returns the sound wave travel time in microseconds
31   duration = pulseIn(echoPin, HIGH);
32   // Calculate the distance
33   distanceCm = duration * SOUND_SPEED/2;
34   // Convert to inches
35   distanceInch = distanceCm * CM_TO_INCH;
36   // Prints the distance in the Serial Monitor
37   Serial.print("Distance (cm): ");
38   Serial.println(distanceCm);
39   // Serial.print("Distance (inch): ");
40   // Serial.println(distanceInch);
41   delay(1000);
42 }
43 }
```