**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)** Fakultas Vokasi, Universitas Brawijaya

**Praktik Simulasi Sensor Jarak (Ultrasonic)**



*Novita Lunar Pratiwi*  
 Fakultas Vokasi, Universitas Brawijaya  
 Email: novitalunarp@student.ub.ac.id

## **Abstract (Abstrak)**

## Praktikum ini bertujuan untuk memahami prinsip kerja sensor ultrasonik dalam mengukur jarak menggunakan Arduino dan simulasi di Wokwi. Sensor ultrasonik HC-SR04 digunakan untuk mendeteksi jarak objek dengan mengandalkan pantulan gelombang ultrasonik yang dipancarkan. Hasil percobaan menunjukkan bahwa sensor mampu mengukur jarak dengan tingkat akurasi yang tinggi dan data hasil pengukuran dapat ditampilkan pada serial monitor. Simulasi di Wokwi memungkinkan pemantauan hasil secara virtual tanpa perangkat keras fisik, memudahkan dalam pengujian dan eksperimen.

## **Keywords**— Internet of Things, Arduino, Ultrasonic Sensor, HC-SR04, Wokwi

## **1. Introduction (Pendahuluan)**

### **1.1 Latar Belakang**

### Sensor ultrasonik merupakan komponen penting dalam banyak aplikasi Internet of Things (IoT), seperti sistem penghindaran rintangan pada robot, pengukuran level air, dan deteksi jarak. Sensor ini bekerja dengan memanfaatkan gelombang ultrasonik untuk mengukur jarak antara sensor dengan objek yang terdeteksi. Salah satu sensor ultrasonik yang banyak digunakan adalah HC-SR04, yang memancarkan gelombang suara dan mengukur waktu yang dibutuhkan untuk gelombang tersebut kembali setelah mengenai objek. Praktikum ini bertujuan untuk memanfaatkan sensor HC-SR04 dalam pengukuran jarak, serta memanfaatkan simulasi di platform Wokwi untuk memudahkan pengujian dan eksperimen, sehingga memberikan pemahaman yang lebih baik tentang cara kerja sensor ultrasonik dalam sistem IoT.

### **1.2 Tujuan**

1. Mempelajari cara kerja sensor ultrasonik dalam mengukur jarak.
2. Menggunakan Wokwi untuk mensimulasikan penggunaan sensor HC-SR04 dengan ESP32.
3. Menampilkan hasil pengukuran jarak melalui serial monitor.

## **2. Methodology (Metodologi)**

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

1. **Mikrokontroler:** ESP32 (simulasi di Wokwi)
2. **Komponen:** Sensor Ultrasonik HC-SR04
3. **Software:** Wokwi (<https://wokwi.com>), VSCode dengan PlatformIO

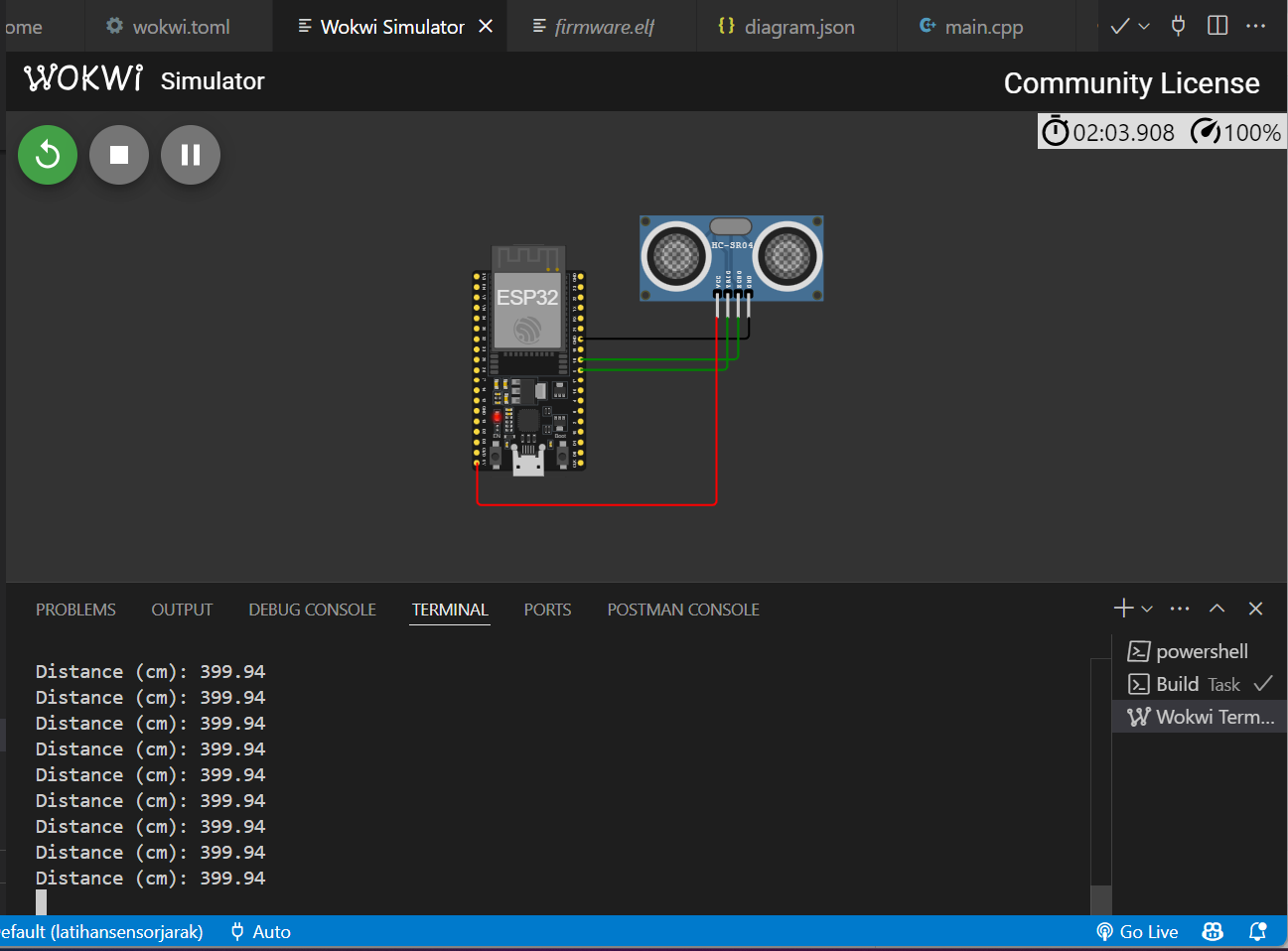
### **2.2 Implementation Steps (Langkah Implementasi)**

1. **Membangun Rangkaian Simulasi**
   1. Menambahkan ESP32 dan sensor ultrasonik HC-SR04 pada Wokwi.
   2. Menghubungkan pin Trig dan Echo ke ESP32.
2. **Membuat File Konfigurasi**
   1. Membuat file wokwi.toml untuk konfigurasi proyek di VSCode.
   2. Membuat file diagram.json untuk mendefinisikan koneksi perangkat di Wokwi.
3. **Menulis Kode Program**
   1. Menggunakan kode berbasis Arduino untuk membaca data dari sensor ultrasonik dan menampilkannya di serial monitor.
4. **Menjalankan Simulasi**
   1. Memantau hasil simulasi di serial monitor Wokwi.

## **3. Results and Discussion (Hasil dan Pembahasan)**

## **3.1 Experimental Results (Hasil Eksperimen)**

1. Berhasil menyambungkan sensor ultrasonik HC-SR04 ke ESP32 di Wokwi.
2. Sensor dapat mengukur jarak dengan baik dalam satuan cm dan inch.
3. Data jarak ditampilkan di serial monitor dengan akurasi yang memadai.

**Screenshot hasil simulasi:**

## **4. Appendix (Lampiran)**

### **4.1 Kode Program**

**a. sketch.ino**

#include <Arduino.h>

const int trigPin = 5;

const int echoPin = 18;

//define sound speed in cm/uS

#define SOUND\_SPEED 0.034

#define CM\_TO\_INCH 0.393701

long duration;

float distanceCm;

float distanceInch;

void setup() {

 Serial.begin(115200); // Starts the serial communication

 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

 pinMode(echoPin, INPUT); // Sets the echoPin as an Input

}

void loop() {

 // Clears the trigPin

 digitalWrite(trigPin, LOW);

 delayMicroseconds(2);

 // Sets the trigPin on HIGH state for 10 micro seconds

 digitalWrite(trigPin, HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds

 duration = pulseIn(echoPin, HIGH);

  // Calculate the distance

 distanceCm = duration \* SOUND\_SPEED/2;

  // Convert to inches

 distanceInch = distanceCm \* CM\_TO\_INCH;

  // Prints the distance in the Serial Monitor

 Serial.print("Distance (cm): ");

 Serial.println(distanceCm);

 // Serial.print("Distance (inch): ");

 // Serial.println(distanceInch);

  delay(1000);

}

**b. diagram.json**

 {

    "version": 1,

    "author": "RAHMADANI LESTARI",

    "editor": "wokwi",

    "parts": [

      { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },

      { "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -46.5, "left": 130.3, "attrs": {} }

    ],

    "connections": [

      [ "esp:TX", "$serialMonitor:RX", "", [] ],

      [ "esp:RX", "$serialMonitor:TX", "", [] ],

      [ "ultrasonic1:VCC", "esp:5V", "red", [ "v182.4", "h-220.8", "v-28.8" ] ],

      [ "ultrasonic1:TRIG", "esp:5", "green", [ "v0" ] ],

      [ "ultrasonic1:ECHO", "esp:18", "green", [ "v0" ] ],

      [ "ultrasonic1:GND", "esp:GND.3", "black", [ "v0" ] ]

    ],

    "dependencies": {}

  }