**LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)**

Simulasi Sensor Jarak (Ultrasonic) dengan ESP32 di Wokwi & VSCode

*Author : Nelly Fananda Melani*

*Fakultas Vokasi, Universitas Brawijaya*

*Email:* [*nellyfananda15@student.ub.ac.id*](mailto:nellyfananda15@student.ub.ac.id)

**Abstract**

This experiment simulates and implements an ultrasonic distance sensor using an ESP32 microcontroller in the Wokwi simulator and the PlatformIO development environment on Visual Studio Code (VSCode). The system utilizes the HC-SR04 ultrasonic sensor, which consists of a trigger (trigPin) and an echo (echoPin) to measure distance based on the time taken for sound waves to travel and reflect. The implementation involves writing C++ code in the main.cpp file, configuring the platformio.ini file, and designing the connection layout in diagram.json. The ESP32 reads the duration of the reflected signal and calculates the distance in centimeters and inches, displaying the results in the Serial Monitor. The successful simulation confirms the microcontroller’s ability to accurately measure distance in real-time without requiring physical hardware. This study provides a foundational understanding of ultrasonic sensor integration, microcontroller programming, and IoT-based system applications. Furthermore, this project has potential applications in robotics, automation, obstacle detection, and smart monitoring systems.

**Keywords**— ESP32, Ultrasonic Sensor, HC-SR04, Wokwi, PlatformIO, Distance Measurement

**1. Introduction (Pendahuluan)**

Perkembangan teknologi Internet of Things (IoT) telah memungkinkan berbagai aplikasi berbasis sensor untuk digunakan dalam kehidupan sehari-hari, termasuk sistem pemantauan jarak dan deteksi objek. Salah satu sensor yang umum digunakan dalam pengukuran jarak adalah sensor ultrasonik HC-SR04, yang bekerja dengan prinsip pemantulan gelombang suara untuk menentukan jarak suatu objek.

ESP32 merupakan mikrokontroler yang banyak digunakan dalam pengembangan sistem IoT karena memiliki performa tinggi, konektivitas Wi-Fi dan Bluetooth, serta dukungan berbagai sensor. Namun, dalam tahap pengembangan, pengujian perangkat keras sering kali menjadi kendala karena keterbatasan akses terhadap komponen fisik. Oleh karena itu, penggunaan simulator seperti Wokwi menjadi solusi yang efektif untuk melakukan pengujian sistem tanpa memerlukan perangkat keras secara langsung.

Pada penelitian ini, dilakukan simulasi pengukuran jarak menggunakan sensor ultrasonik HC-SR04 yang terhubung dengan ESP32 dalam lingkungan pengembangan PlatformIO di Visual Studio Code (VSCode). Sistem ini membaca durasi gelombang suara yang dipantulkan oleh objek dan mengkonversinya menjadi nilai jarak dalam satuan sentimeter dan inci. Hasil pengukuran ditampilkan melalui Serial Monitor, memungkinkan pengguna untuk memantau data secara real-time.

Penelitian ini bertujuan untuk memberikan pemahaman dasar mengenai pemrograman mikrokontroler, integrasi sensor ultrasonik, serta pengembangan sistem berbasis IoT dalam simulasi tanpa perangkat keras. Dengan demikian, proyek ini dapat menjadi referensi bagi pengembang yang ingin memahami cara kerja sensor jarak dan implementasinya dalam berbagai aplikasi, seperti sistem pemantauan, robotika, dan otomasi cerdas.

**1.1 Background of the IoT Practicum (Latar Belakang Praktikum IoT)**

Internet of Things (IoT) merupakan teknologi yang menghubungkan berbagai perangkat fisik dengan jaringan internet, memungkinkan komunikasi dan pertukaran data secara otomatis. Penerapan IoT telah berkembang pesat di berbagai bidang, seperti rumah pintar (smart home), sistem pemantauan industri, kesehatan, hingga transportasi cerdas. Dalam pengembangannya, pemrograman mikrokontroler dan integrasi sensor menjadi aspek fundamental yang harus dipahami oleh para pengembang dan mahasiswa di bidang teknologi.

Salah satu tantangan dalam pembelajaran IoT adalah keterbatasan perangkat keras untuk praktik langsung. Tidak semua mahasiswa atau pengembang memiliki akses ke mikrokontroler dan sensor yang diperlukan untuk melakukan eksperimen secara fisik. Oleh karena itu, penggunaan simulator seperti Wokwi menjadi alternatif yang efektif dalam proses pembelajaran dan pengujian sistem berbasis IoT.

Praktikum ini bertujuan untuk memberikan pengalaman langsung dalam memprogram dan mensimulasikan perangkat IoT menggunakan ESP32 serta sensor ultrasonik HC-SR04. Dengan memanfaatkan lingkungan pengembangan PlatformIO di Visual Studio Code (VSCode) dan simulator Wokwi, peserta dapat memahami konsep dasar pengukuran jarak, pengolahan data sensor, serta komunikasi antara sensor dan mikrokontroler tanpa memerlukan perangkat fisik.

Melalui praktikum ini, peserta diharapkan dapat mengembangkan keterampilan dalam pemrograman mikrokontroler, memahami prinsip kerja sensor ultrasonik, serta mengeksplorasi potensi penerapan IoT dalam berbagai bidang. Dengan demikian, praktikum ini dapat menjadi landasan bagi pengembangan proyek IoT yang lebih kompleks di masa depan.

**1.**2 Experimental Objectives (Tujuan Eksperimen)

Eksperimen ini bertujuan untuk:

1. Mensimulasikan penggunaan sensor ultrasonik dengan ESP32 menggunakan simulator Wokwi dan lingkungan pengembangan PlatformIO di Visual Studio Code (VSCode).
2. Memahami prinsip kerja sensor ultrasonik HC-SR04, termasuk cara mengukur jarak berdasarkan waktu tempuh gelombang ultrasonik.
3. Mengembangkan keterampilan dalam pemrograman mikrokontroler ESP32, khususnya dalam membaca data sensor dan mengolahnya menggunakan bahasa pemrograman C++.
4. Menguji komunikasi antara sensor dan ESP32 dengan menampilkan hasil pengukuran jarak dalam Serial Monitor.
5. Mengoptimalkan penggunaan simulator Wokwi sebagai alat bantu pembelajaran dan pengujian sistem berbasis IoT tanpa memerlukan perangkat fisik.

2. Methodology (Metodologi)

Eksperimen ini dilakukan dengan mensimulasikan sistem berbasis ESP32 dan sensor ultrasonik HC-SR04 menggunakan Wokwi Simulator. Langkah-langkah dalam metodologi ini meliputi:

1. Membuat Diagram Simulasi di Wokwi Simulator
2. Menulis Kode Program dalam bahasa C++ (Arduino) menggunakan PlatformIO di Visual Studio Code (VSCode).
3. Konfigurasi Lingkungan dengan mengedit platformio.ini dan wokwi.toml.
4. Kompilasi & Simulasi di Wokwi Simulator tanpa perangkat keras fisik.
5. Pengamatan Hasil Simulasi melalui Serial Monitor, dengan melihat data pembacaan jarak dari sensor ultrasonik yang ditampilkan dalam satuan centimeter (cm).
6. Analisis dan Evaluasi kinerja sistem dalam membaca jarak, mengidentifikasi potensi kesalahan dalam pengukuran, dan membandingkan hasil simulasi dengan perhitungan teoritis.

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

Eksperimen ini dilakukan secara virtual menggunakan **Wokwi Simulator**, sehingga tidak memerlukan perangkat keras fisik. Alat dan bahan yang digunakan dalam proyek ini meliputi::

* ESP32 Devkit V1 (simulasi pada Wokwi)
* Sensor Ultrasonik HC-SR04 (simulasi pada Wokwi)
* *Resistor* (jika diperlukan dalam simulasi) (simulasi pada Wokwi)
* Laptop/PC dengan Visual Studio Code dan PlatformIO sebagai lingkungan pengembangan
* Wokwi Simulator untuk menjalankan dan menguji kode tanpa perangkat fisik
* Library Arduino untuk pemrograman mikrokontroler
* Koneksi internet untuk mengakses dan menjalankan simulasi di Wokwi

**2.2 Implementation Steps (Langkah Implementasi)**

Eksperimen ini dilakukan melalui beberapa langkah utama:

1. **Menyiapkan Lingkungan Pengembangan**

* Instal Visual Studio Code (VSCode) dan tambahkan ekstensi PlatformIO IDE.

Sebuah gambar berisi teks, Font, cuplikan layar, logo

Konten yang dihasilkan AI mungkin salah.

* Buat proyek ESP32 baru menggunakan PlatformIO.

PLATFORMIO -> QUICK ACCESS -> Open -> New Project -> Project Wizard -> Finish

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

* Akses Wokwi Simulator untuk merancang Simulasi Sensor Jarak Jauh

**Sebuah gambar berisi teks, cuplikan layar, Software multimedia, software

Konten yang dihasilkan AI mungkin salah.**

**Sebuah gambar berisi cuplikan layar, teks, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.**

(Pilih Starter Templates -> **ESP32**)

1. **Merancang Diagram Simulasi di Wokwi**

* Buat diagram baru di wokwi.com.

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

* **Hubungkan ESP32 dengan HC-SR04 Ultrasonic Distance Sensor secara virtual sesuai skema rangkaian.**

Sebuah gambar berisi teks, cuplikan layar, Font, software

Konten yang dihasilkan AI mungkin salah.

1. **Menulis Kode Program di VSCode**

**Menulis kode dalam C++ menggunakan PlatformIO** untuk membaca data dari sensor ultrasonik

#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);

}

1. **Konfigurasi PlatformIO**

* Edit file platformio.ini dengan isi berikut:
* ; PlatformIO Project Configuration File
* ;
* ;   Build options: build flags, source filter
* ;   Upload options: custom upload port, speed and extra flags
* ;   Library options: dependencies, extra library storages
* ;   Advanced options: extra scripting
* ;
* ; Please visit documentation for the other options and examples
* ; https://docs.platformio.org/page/projectconf.html
* [env:esp32doit-devkit-v1]
* platform = espressif32
* board = esp32doit-devkit-v1
* framework = arduino
* lib\_deps = mbed-aluqard/arduino@0.0.0+sha.3b83fc30bbdf
* [env:esp32dev]
* platform = espressif32
* board = esp32dev
* framework = arduino
* lib\_deps = mbed-aluqard/arduino@0.0.0+sha.3b83fc30bbdf

1. **Menyusun File Konfigurasi Wokwi**

* Buat file wokwi.toml dengan isi berikut:

**[wokwi]**

**version = 1**

**firmware = '.pio\build\esp32doit-devkit-v1\firmware.bin'**

**elf = '.pio\build\esp32doit-devkit-v1\firmware.elf'**

1. **Kompilasi dan Pembuatan Firmware**

* Lakukan kompilasi (build) pada main.cpp menggunakan PlatformIO.

Sebuah gambar berisi teks, cuplikan layar, Font, nomor

Konten yang dihasilkan AI mungkin salah.

* Cara kedua lakukan build dengan mengetikkan **pio run** di terminal

Sebuah gambar berisi teks, software, Software multimedia, cuplikan layar

Konten yang dihasilkan AI mungkin salah.

* Setelah berhasil, akan dihasilkan **firmware.bin** dan **firmware.elf.**

1. **Simulasi dan Pengujian**

* Request license di wokwi.com.

Masuk ke di command palette -> Request a New License

Sebuah gambar berisi teks, cuplikan layar, Software multimedia, software

Konten yang dihasilkan AI mungkin salah.

-> open -> Get Your License

Sebuah gambar berisi teks, cuplikan layar, Font, Software multimedia

Konten yang dihasilkan AI mungkin salah.

* Jalankan simulasi dengan mengetik perintah **wokwi simulator** di command palette.

Sebuah gambar berisi teks, cuplikan layar, Software multimedia, software

Konten yang dihasilkan AI mungkin salah.

Verifikasi bahwa sensor ultrasonik berfungsi:

**Pantau nilai jarak** melalui **Serial Monitor**

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

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

**3.1 Experimental Results (Hasil Eksperimen)**

Simulasi **sensor jarak (ultrasonik)** menggunakan **ESP32** di Wokwi Simulator telah berhasil diimplementasikan. Proses kompilasi dan build menggunakan **PlatformIO di Visual Studio Code (VSCode)** berjalan dengan sukses tanpa error.

**Hasil pengujian menunjukkan bahwa:**

* Sensor ultrasonik (HC-SR04) mampu membaca jarak objek di depannya dengan akurasi tinggi dalam simulasi.
* Data jarak berhasil ditampilkan di Serial Monitor, menunjukkan nilai yang berubah sesuai dengan pergerakan objek dalam simulasi.
* ESP32 merespons perubahan jarak secara real-time, dengan waktu pembacaan yang cepat dan stabil.

**Berikut adalah rata-rata hasil simulasi:**

* Akurasi pembacaan jarak: Konsisten dengan nilai simulasi yang diberikan oleh Wokwi.
* Respons sistem: Tidak ada delay yang menghambat pembacaan sensor atau eksekusi logika program.
* Output utama: Data jarak yang diperbarui secara real-time di Serial Monitor.

**3.2 Discussion (Pembahasan)**

Keberhasilan eksperimen ini menunjukkan bahwa **ESP32 dapat digunakan untuk membaca data dari sensor ultrasonik (HC-SR04) dan menampilkan hasilnya di Serial Monitor**. Simulasi di **Wokwi** memungkinkan pengujian kode tanpa perangkat fisik, sehingga mempermudah proses debugging dan pengembangan sebelum implementasi pada perangkat keras sesungguhnya.

**Faktor yang Mempengaruhi Hasil:**

* Logika Pemrograman: Jika perhitungan waktu pantulan gelombang ultrasonik salah, maka hasil pengukuran jarak tidak akurat.
* Konfigurasi Pin: Pin untuk Trigger dan Echo harus sesuai dengan yang diprogram agar sensor dapat bekerja dengan benar.
* Reflektivitas Objek: Dalam simulasi, semua objek dianggap memiliki permukaan ideal, sedangkan dalam dunia nyata, bahan objek dapat mempengaruhi pantulan gelombang ultrasonik.
* Frekuensi Pembacaan: Jika pembacaan dilakukan terlalu cepat tanpa delay yang cukup, bisa terjadi ketidakkonsistenan data.

**Peningkatan yang Dapat Dilakukan:**

* Menambahkan Filter atau Moving Average: Untuk meningkatkan akurasi pengukuran dengan mengurangi noise dalam data pembacaan jarak.
* Menggunakan Kalibrasi Tambahan: Dalam implementasi nyata, kalibrasi dapat dilakukan untuk meningkatkan ketepatan hasil pembacaan.
* Integrasi dengan IoT: Hasil pembacaan jarak dapat dikirim ke platform berbasis cloud atau ditampilkan dalam dashboard web.
* Menambahkan Indikator Visual atau Audio: Seperti buzzer yang berbunyi ketika objek berada dalam jarak tertentu, atau LED yang menyala jika jarak terlalu dekat.

Hasil eksperimen ini dapat menjadi dasar bagi pengembangan sistem **deteksi jarak otomatis**, seperti sistem parkir pintar, robot navigasi, atau alat bantu bagi penyandang disabilitas untuk mendeteksi rintangan di sekitarnya.

**4. Appendix (Lampiran)**

**Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.**