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

**Fakultas Vokasi, Universitas Brawijaya**

**PRAKTIK PEMBUATAN SIMULASI ESP32 & SENSOR JARAK (ULRASONIC)**

*Author Bayu Maha Resi*

*Email: bayumaharesii@gmail,com*

**Abstract**

Pemantauan dan pengukuran jarak secara real-time menjadi aspek penting dalam berbagai aplikasi, seperti sistem keamanan, robotika, dan otomasi industri. Eksperimen ini bertujuan untuk mensimulasikan penggunaan ESP32 dalam membaca data dari sensor jarak ultrasonik sebagai bagian dari sistem deteksi dan pengukuran jarak. ESP32 berfungsi sebagai mikrokontroler utama yang mengolah data dari sensor ultrasonik dan menampilkan hasilnya dalam bentuk digital melalui komunikasi serial atau antarmuka berbasis web.

Dalam eksperimen ini, dilakukan konfigurasi perangkat keras dan pemrograman ESP32 untuk membaca serta menginterpretasikan data dari sensor jarak ultrasonik. Hasil simulasi menunjukkan bahwa sistem dapat secara akurat mendeteksi dan mengukur jarak secara real-time. Eksperimen ini memberikan wawasan mengenai implementasi sensor jarak dalam sistem berbasis IoT serta bagaimana ESP32 dapat digunakan untuk membangun sistem deteksi yang efisien dan fleksibel.

**Kata kunci**: Wokwi, Arduino, ESP32.

**1. Introduction**

Dalam dunia IoT (Internet of Things), sensor jarak ultrasonik memainkan peran penting dalam berbagai aplikasi, mulai dari sistem keamanan hingga otomasi industri. Sensor ini digunakan untuk mendeteksi dan mengukur jarak suatu objek tanpa kontak fisik, sehingga ideal untuk aplikasi seperti deteksi hambatan, sistem parkir otomatis, dan robotika.Pada praktik ini, dilakukan simulasi penggunaan ESP32 sebagai mikrokontroler utama untuk membaca data dari sensor jarak ultrasonik dan mengolahnya untuk ditampilkan dalam bentuk digital. ESP32 dipilih karena memiliki konektivitas Wi-Fi dan Bluetooth, yang memungkinkan integrasi dengan berbagai platform IoT. Sistem ini dapat diterapkan dalam berbagai skenario, seperti pemantauan jarak, sistem peringatan dini, atau navigasi robot berbasis IoT.

Tujuan dari praktik ini adalah untuk memahami cara kerja sensor jarak ultrasonik, bagaimana menghubungkannya dengan ESP32, serta bagaimana membaca dan mengolah data jarak secara real-time. Selain itu, simulasi ini juga memberikan wawasan tentang cara membangun sistem deteksi dan pemantauan yang efisien serta fleksibel menggunakan mikrokontroler.

**1.1 Latar belakang**

Dalam era modern, pengukuran dan deteksi jarak secara otomatis menjadi aspek penting dalam berbagai bidang, seperti sistem keamanan, robotika, dan otomasi industri. Dengan perkembangan teknologi IoT (Internet of Things), penggunaan sensor jarak ultrasonik semakin luas dalam berbagai aplikasi untuk mendeteksi dan mengukur jarak objek secara real-time. Sensor ini bekerja dengan memancarkan gelombang ultrasonik dan mengukur waktu pantulan gelombang untuk menentukan jarak suatu objek tanpa kontak fisik.

ESP32, sebagai mikrokontroler yang mendukung komunikasi Wi-Fi dan Bluetooth, memungkinkan sistem pemantauan jarak yang lebih fleksibel dan dapat diintegrasikan ke berbagai platform. Dengan kombinasi ESP32 dan sensor jarak ultrasonik, dapat dibuat sistem deteksi yang efisien dan mudah diimplementasikan dalam berbagai kebutuhan, seperti sistem parkir otomatis, penghindaran rintangan pada robot, atau pemantauan keamanan. Oleh karena itu, simulasi penggunaan ESP32 dalam membaca dan mengolah data dari sensor jarak ultrasonik menjadi langkah awal yang penting dalam memahami cara kerja sistem pemantauan berbasis IoT.

**1.2 Tujuan eksperimen**

Eksperimen ini bertujuan untuk memahami prinsip kerja sensor jarak ultrasonik serta cara menghubungkannya dengan ESP32 sebagai mikrokontroler utama. Selain itu, eksperimen ini juga dirancang untuk mengembangkan simulasi pengukuran jarak secara real-time dengan menggunakan sensor ultrasonik dan menampilkan hasilnya melalui komunikasi serial atau antarmuka digital lainnya. Dengan demikian, eksperimen ini memberikan wawasan mengenai penerapan konsep dasar IoT dalam sistem deteksi dan pemantauan jarak yang efisien dan fleksibel.

**2. Methodology**

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

* Website Wokwi
* Laptop
* Github
* VScode

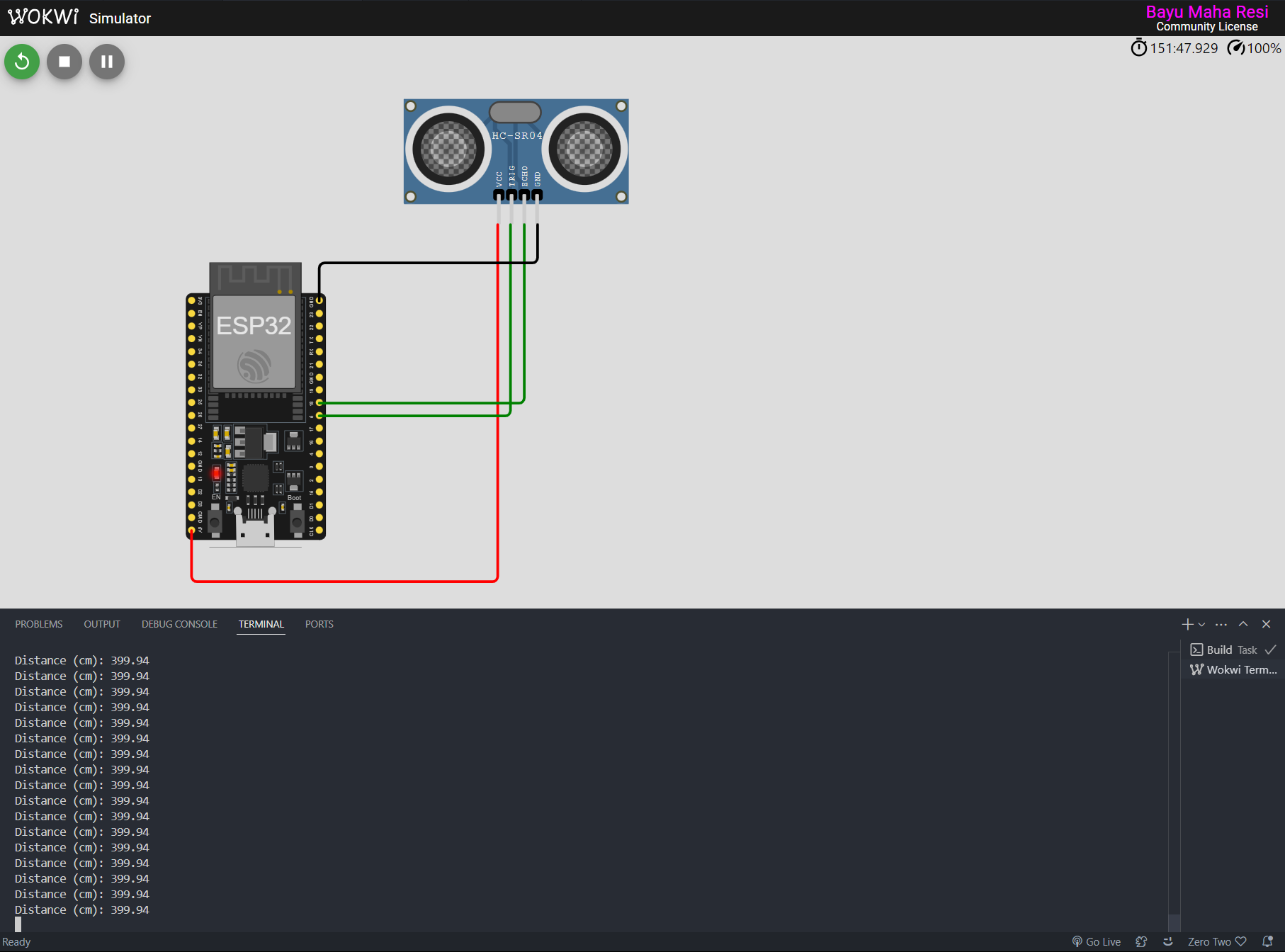
**2.2 Implementation Steps (Langkah Implementasi)**

Buat akun wokwi dengan GitHub

* Membuat topology
* Memindah diagram.json di Wokwi kedalam Vscode
* Memindahkan code dari wokwi ke Vs Code
* Lakukan configurasi ulang
* Tambahkan code pada file platform io ini

**3. Results and Discussion**

**3.1 Experimental Results**



Saat button ditekan lampu akan nyala

**4. Appendix**

**Code Program:**

**Main.cpp**



Digram json:

