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

**Fakultas Vokasi, Universitas Brawijaya**

**Praktek Simulasi Sensor Jarak (Ultrasonic)**

*Alvaro Zeka Ricardo*

*Fakultas Vokasi,Universitas Brawijaya*

*Email : alvarozeka7@student.ub.ac.id*

**Abstract** **:** Sensor jarak ultrasonik merupakan teknologi yang umum digunakan dalam berbagai aplikasi, seperti robotika, sistem keamanan, dan otomatisasi industri. Dalam praktik ini, sensor ultrasonik diintegrasikan dengan ESP32 untuk mengukur jarak suatu objek menggunakan gelombang ultrasonik. Data yang diperoleh akan diolah dan ditampilkan melalui serial monitor atau layar tampilan. Eksperimen ini bertujuan untuk memahami prinsip kerja sensor ultrasonik, mempelajari komunikasi antara sensor dan mikrokontroler, serta mengimplementasikan pemrograman pada ESP32 untuk mengolah dan menampilkan hasil pengukuran.

**Kata Kunci : ESP32,Sensor,Ultrasonic**

**1. Introduction**

**1.1 Latar belakang**

Sensor ultrasonik bekerja dengan memancarkan gelombang suara berfrekuensi tinggi yang kemudian dipantulkan kembali oleh objek untuk mengukur jarak berdasarkan waktu tempuh gelombang tersebut. Teknologi ini banyak diterapkan dalam berbagai sistem otomatisasi, seperti deteksi hambatan pada robot, pengukuran ketinggian cairan, serta sistem keamanan. Namun, pemahaman mengenai konfigurasi dan pemrograman sensor ultrasonik masih menjadi tantangan bagi pemula dalam bidang elektronika dan IoT. Di sisi lain, ESP32 merupakan mikrokontroler yang memiliki fitur konektivitas Wi-Fi dan Bluetooth serta kapasitas pemrosesan yang lebih tinggi dibandingkan pendahulunya, ESP8266. Dengan keunggulan tersebut, ESP32 banyak digunakan dalam pengembangan sistem berbasis sensor. Oleh karena itu, eksperimen ini dilakukan untuk memahami integrasi sensor ultrasonik dengan ESP32 serta mengeksplorasi penerapannya dalam berbagai proyek IoT.

Dalam dunia elektronika dan IoT (Internet of Things), mikrokontroler berperan penting dalam mengolah data dari berbagai sensor, termasuk sensor ultrasonik. Salah satu mikrokontroler yang banyak digunakan adalah ESP32, yang memiliki keunggulan berupa konektivitas Wi-Fi dan Bluetooth, serta kapasitas pemrosesan yang lebih tinggi dibandingkan pendahulunya, ESP8266. Dengan fitur tersebut, ESP32 mampu mengolah data sensor dengan lebih efisien dan memungkinkan pengiriman data secara nirkabel ke perangkat lain, seperti komputer atau smartphone.

Namun, meskipun sensor ultrasonik dan ESP32 memiliki banyak keunggulan, pemahaman mengenai konfigurasi dan pemrograman keduanya masih menjadi tantangan bagi banyak orang, terutama bagi pemula di bidang elektronika dan IoT. Oleh karena itu, eksperimen ini dilakukan untuk mempelajari bagaimana sensor ultrasonik dapat diintegrasikan dengan ESP32, serta bagaimana data yang diperoleh dapat diproses dan ditampilkan dengan benar.

**1.2 Tujuan eksperimen**

Eksperimen ini bertujuan untuk menguji dan memahami prinsip kerja sensor jarak ultrasonik dalam mendeteksi objek dan mengukur jarak secara real-time menggunakan ESP32. Dengan memahami prinsip ini, pengguna dapat mengetahui bagaimana gelombang ultrasonik bekerja dalam sistem pengukuran jarak dan bagaimana data yang diperoleh dapat dimanfaatkan dalam berbagai aplikasi.

Selain itu, eksperimen ini juga bertujuan untuk mengimplementasikan pemrograman pada ESP32 guna membaca data dari sensor ultrasonik dan menampilkannya dalam bentuk yang dapat dianalisis. Data yang diperoleh akan ditampilkan melalui serial monitor atau antarmuka grafis, sehingga pengguna dapat melihat perubahan jarak secara langsung. Dengan eksperimen ini, diharapkan pengguna dapat memahami lebih dalam tentang integrasi sensor ultrasonik dengan mikrokontroler ESP32 serta potensi penerapannya dalam berbagai proyek IoT.

**2. Methodology (Metodologi)**

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

Alat yang digunakan :

ESP32 - Sebagai mikrokontroler utama untuk mengontrol LED.

HC-SR04-Sebagai Sensor jarak ultrasonic

Kabel Jumper - Untuk menghubungkan komponen satu sama lain.

Breadboard - Sebagai media untuk merangkai komponen tanpa perlu menyolder.

Adaptor USB atau Kabel Micro-USB - Untuk memberikan daya ke ESP32.

Laptop - Untuk mensimulasikan Sistem

Software yang digunakan  
  
 Visual Studio Code (VSCode) – Alternatif IDE yang lebih fleksibel tnpa harus menunggu antrian di Website Wokwi

PlatformIO IDE (Extension di VSCode) – Sebagai lingkungan pengembangan yang mendukung ESP32 dengan fitur yang lebih canggih dibandingkan Arduino IDE.

Wokwi Simulator Extension – Untuk melakukan simulasi ESP32 langsung di VSCode sebelum mengimplementasikannya ke perangkat fisik.

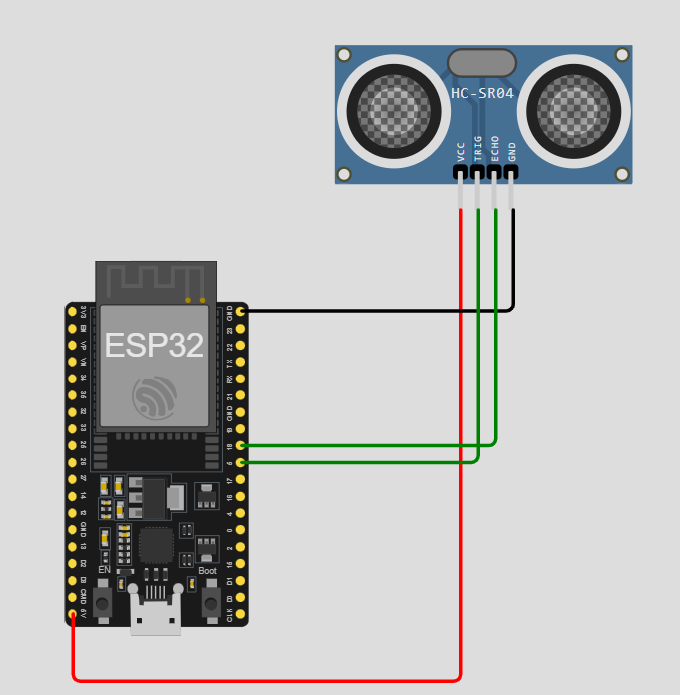
Wokwi – Simulator berbasis web untuk menguji rangkaian ESP32 secara virtual sebelum implementasi fisik.

**2.2 Implementation Steps (Langkah Implementasi)**

Pertama :

Menyusun Diagram Sistem di website wokwi

Tambahkan Part sensor di project



Selanjutnya Tambahkan Code berikut

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);

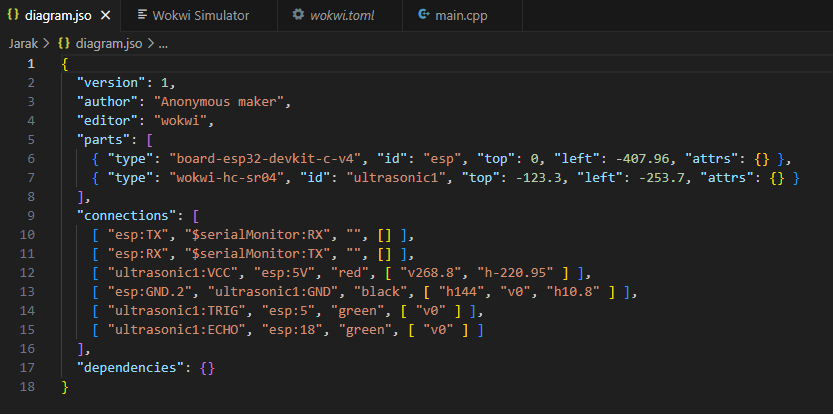
// Serial.print("Distance (inch): ");

// Serial.println(distanceInch);

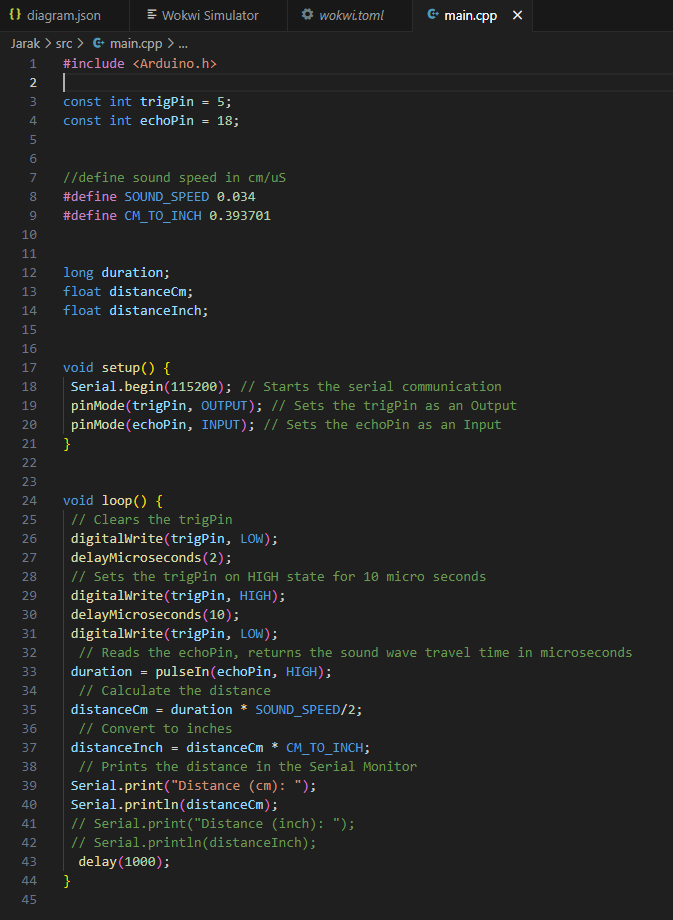
delay(1000);

}

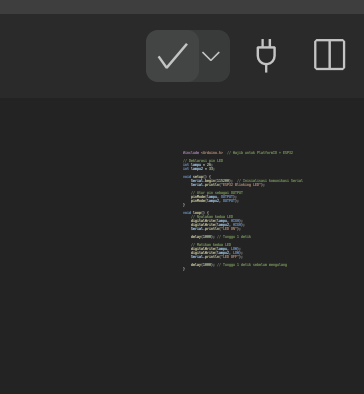
Buat File Diagram.json di VSCode dan copas diagram yang telah dibuat di website wokwi.



Kemudian pilih file src/main.cpp dan copy paste code yang telah kita buat di wokwi web.



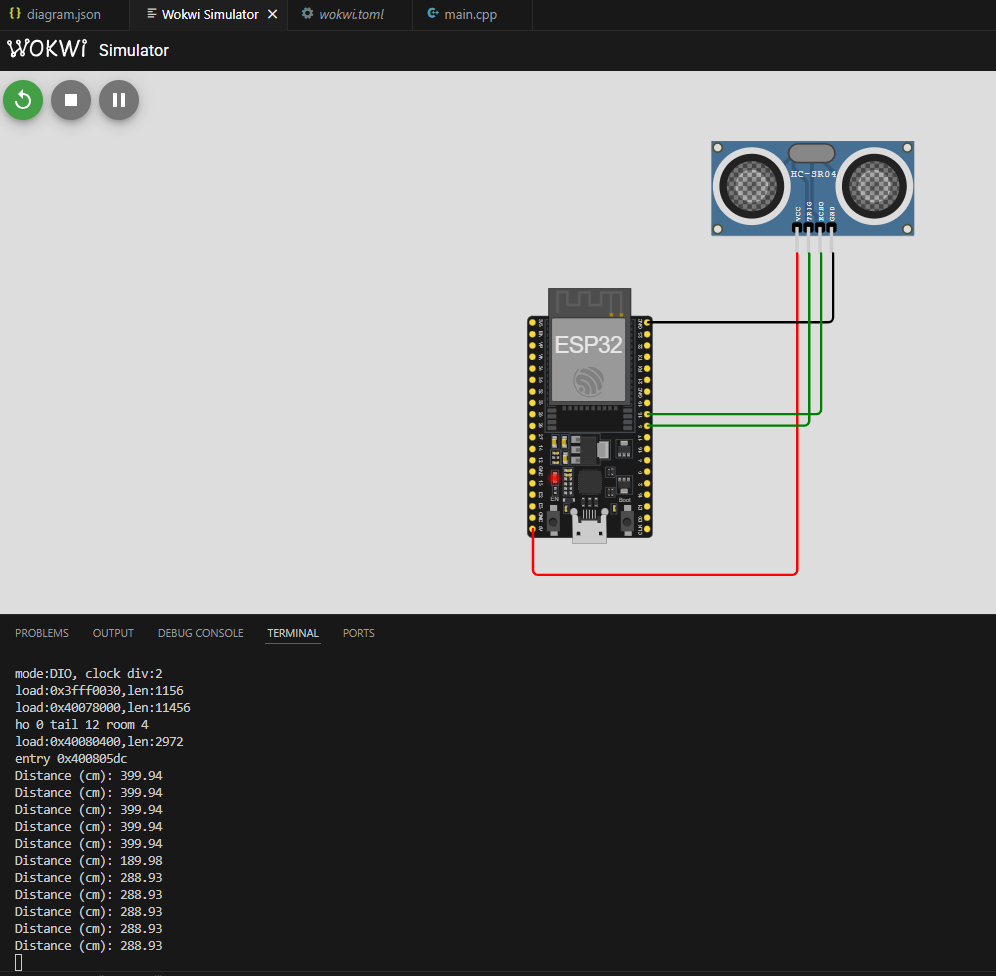
Setelah itu build code tadi dengan menekan tombol centang di kanan atas



Setiap kita mengubah code yang ada di main.cpp harus selalu di build

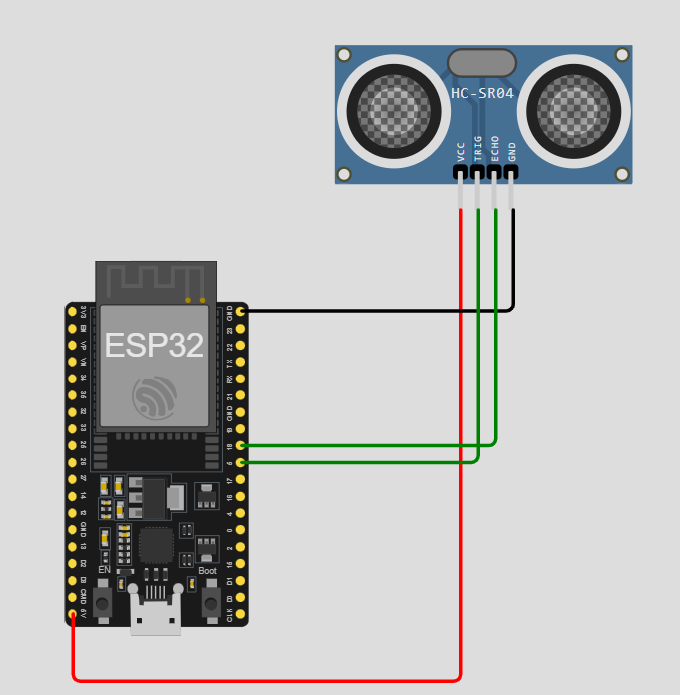
**3. Results and Discussion (Hasil dan Pembahasan)**

**3.1 Experimental Results (Hasil Eksperimen)**



**4. Appendix (Lampiran, jika diperlukan)**

1.Diagram



2.Kode Program

