LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Simulasi Sensor Jarak (Ultrasonic)**

*Assyfaul Khoiriyah*

Fakultas Vokasi, Universitas Brawijaya

[assyfaassyfa0@gmail.com](mailto:assyfaassyfa0@gmail.com)

**Abstract (Abstrak)**

Internet of Things (IoT) memungkinkan berbagai perangkat untuk berkomunikasi dan berinteraksi melalui jaringan. Dalam praktikum ini, dilakukan simulasi penggunaan sensor ultrasonic HC-SR04 dengan ESP32 untuk mengukur jarak. Implementasi dilakukan menggunakan wokwi simulator untuk memvalidasi fungsionalitas system. Hasil simulasi menunjukkan bahwa sensor uktrasonik dapat membaca jarak, namun nilai yang ditampilkan pada terminal menunjukkan angka tetap yang mengindikasi adanya kendala dalam pemrosesan data. Praktikum ini memberikan wawasan mengenai integrasi senor ultrasonic dnegan ESP32 dalam system IoT.

Kata kunci: Internet of Things, ESP32, sensor ultrasonik, HC-SR04, wokwi simulator.

1. **Introduction (Pendahuluan)**
   1. **Latar Belakang**

Dalam era digital saat ini, teknologi IoT telah menjadi solusi inovatif dalam berbagai bidang, termasuk sistem pemantauan jarak. Sensor ulyrasonic HC-SR04 merupakan salah satu perangkat yang umum digunakan untuk mengukur jarak secara akurat dengan prinsip pantulan gelombang suara. Integrasi sensor ini dengan ESP32 memungkinkan pengukuran jarak secara real-time yang dapat digunakan dalam berbagai aplikasi, seperti robotika, sistem keamanan, dan otomasi industri. Oleh karena itu, pada praktikum ini dilakukan simulasi penggunaan sensor ultrasonik dnegan ESP32 menggunakan platform wokwi.

**1.2 Tujuan Eksperimen**

Tujuan dari eksperimen ini adalah:

1. Memahami prinsip kerja sensor ultrasonic HC-SR04 dalam pengukuran jarak
2. Mengimplementasikan komunikasi anara ESP32 dan sensor ultrasonik dalam simulasi
3. Menganalisis hasil pengukuran jarak yang ditampilkan pada terminal simulasi.
4. **Methodology (Metodologi)**

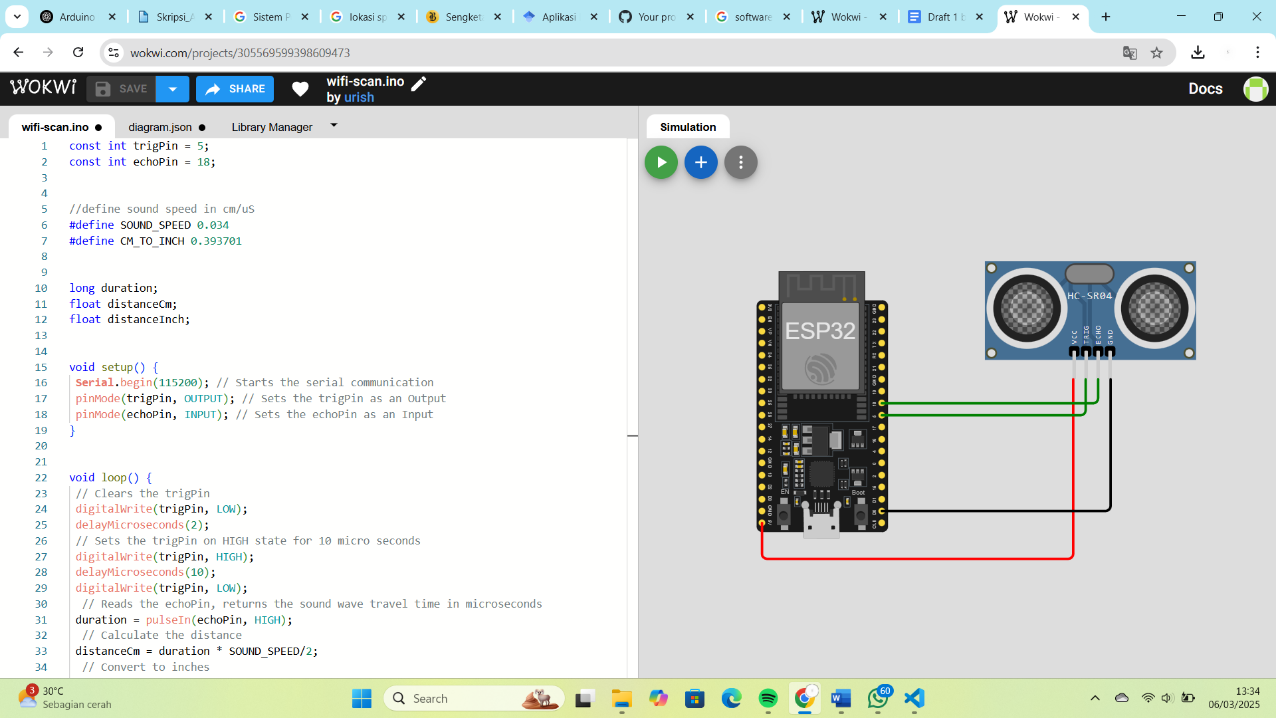
**2.1 Tools & Materials (Alat dan Bahan)**

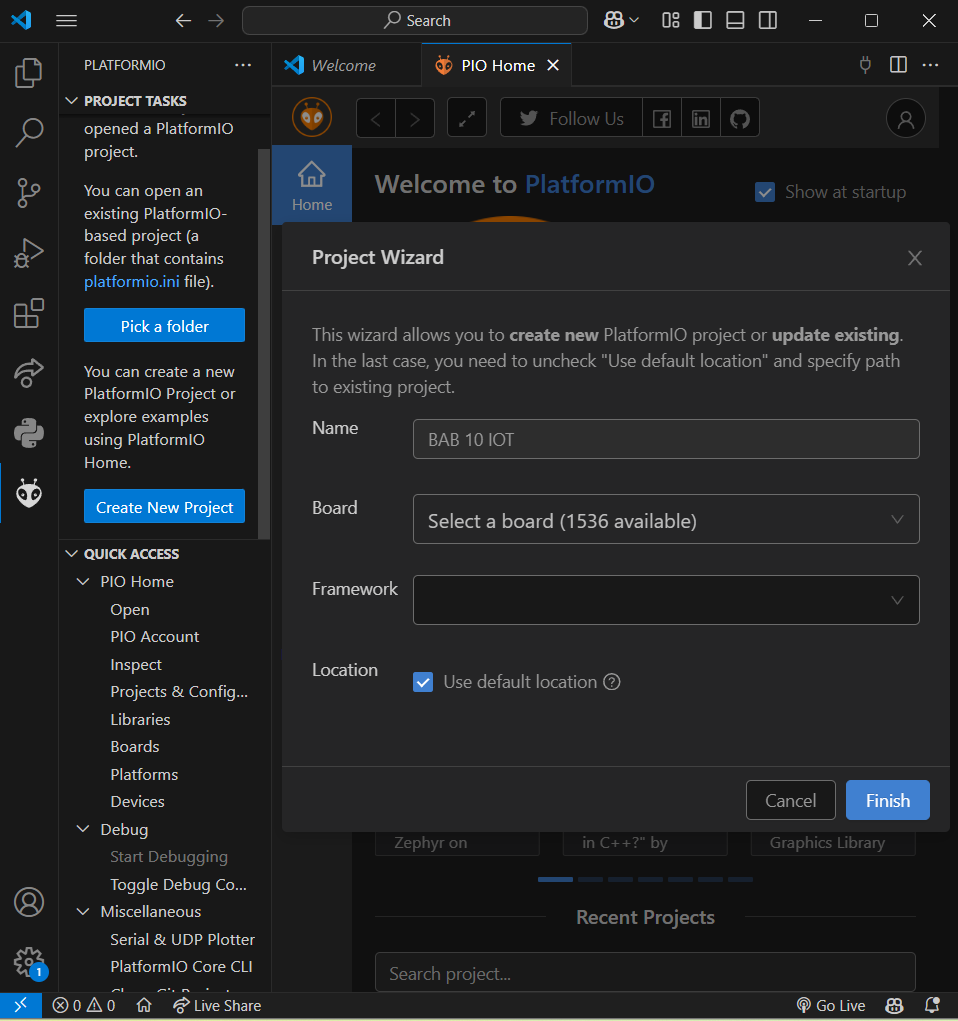
* ESP32
* Sensor ultrasonil HC-SR04
* Kabel jumper
* Wokwi simulator

**2.2 Implementation Steps (Langkah Implementasi)**

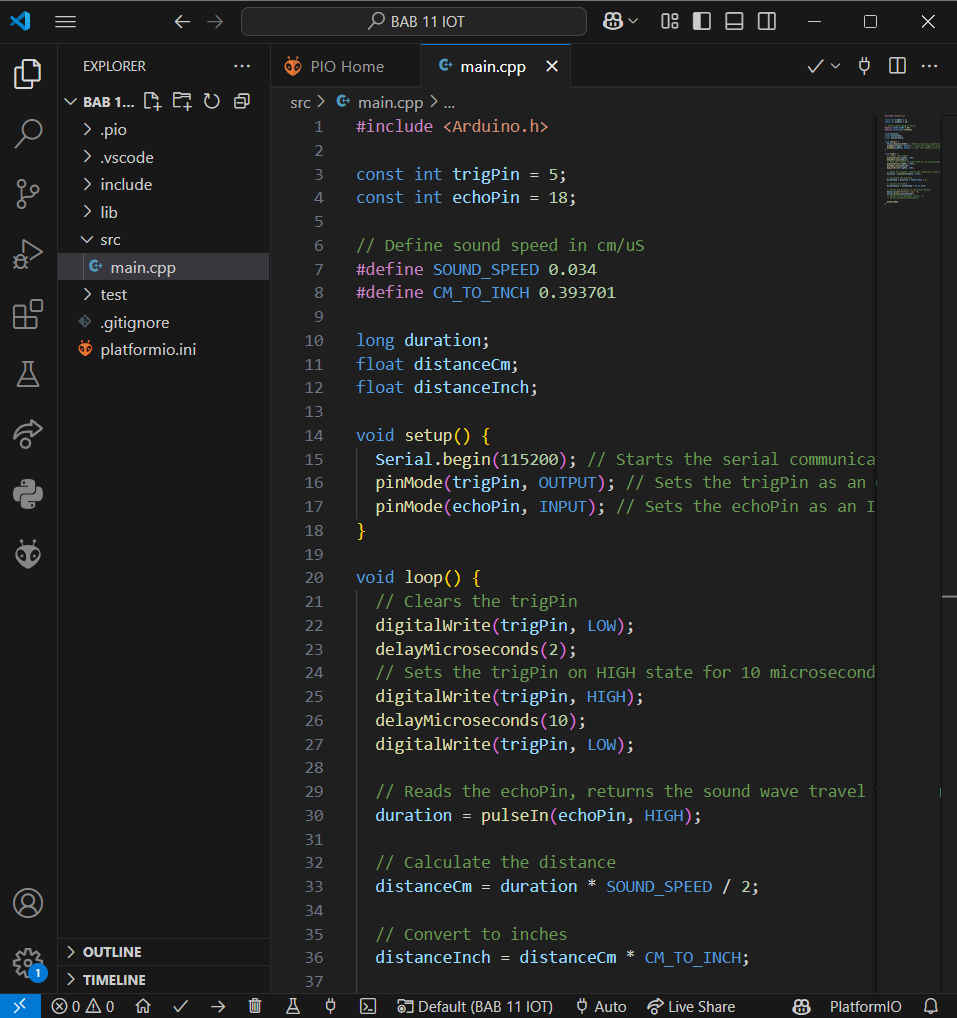
1. Buka web wokwi dan buka diagramnya

1. Lalu masukkan codenya

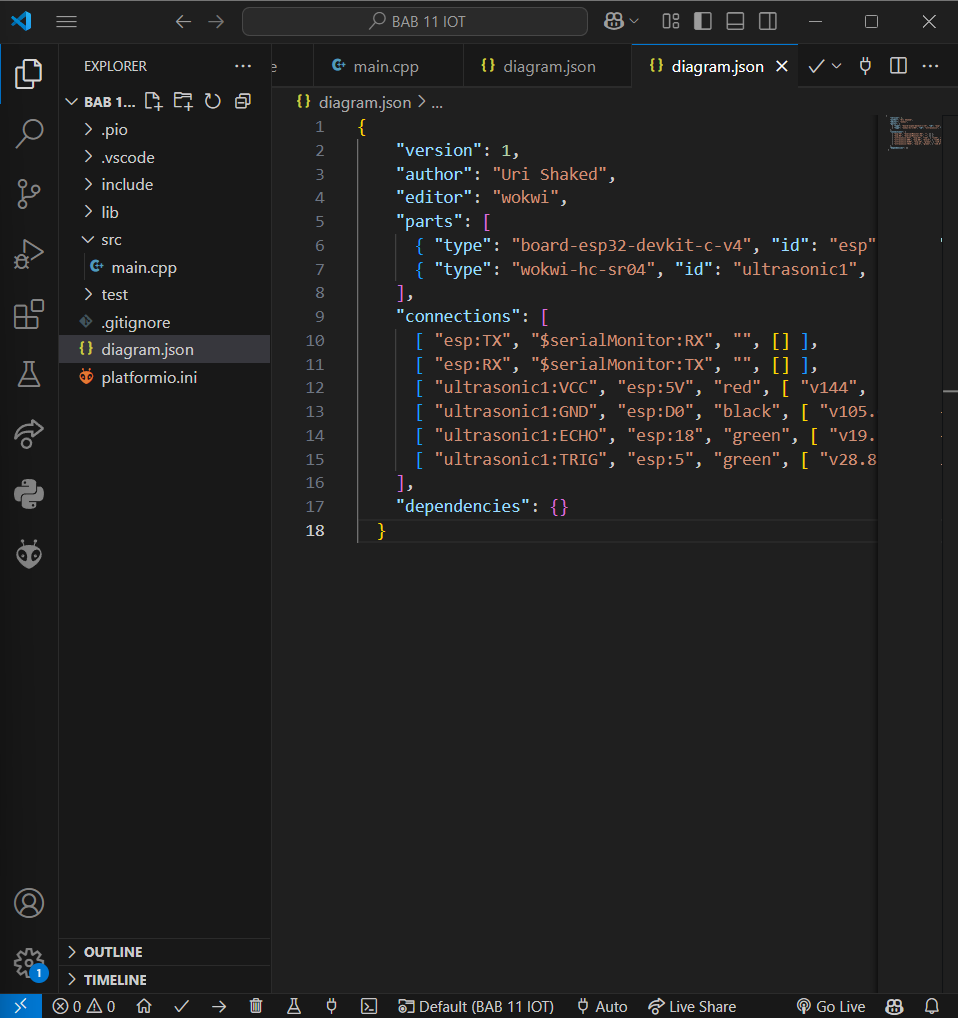


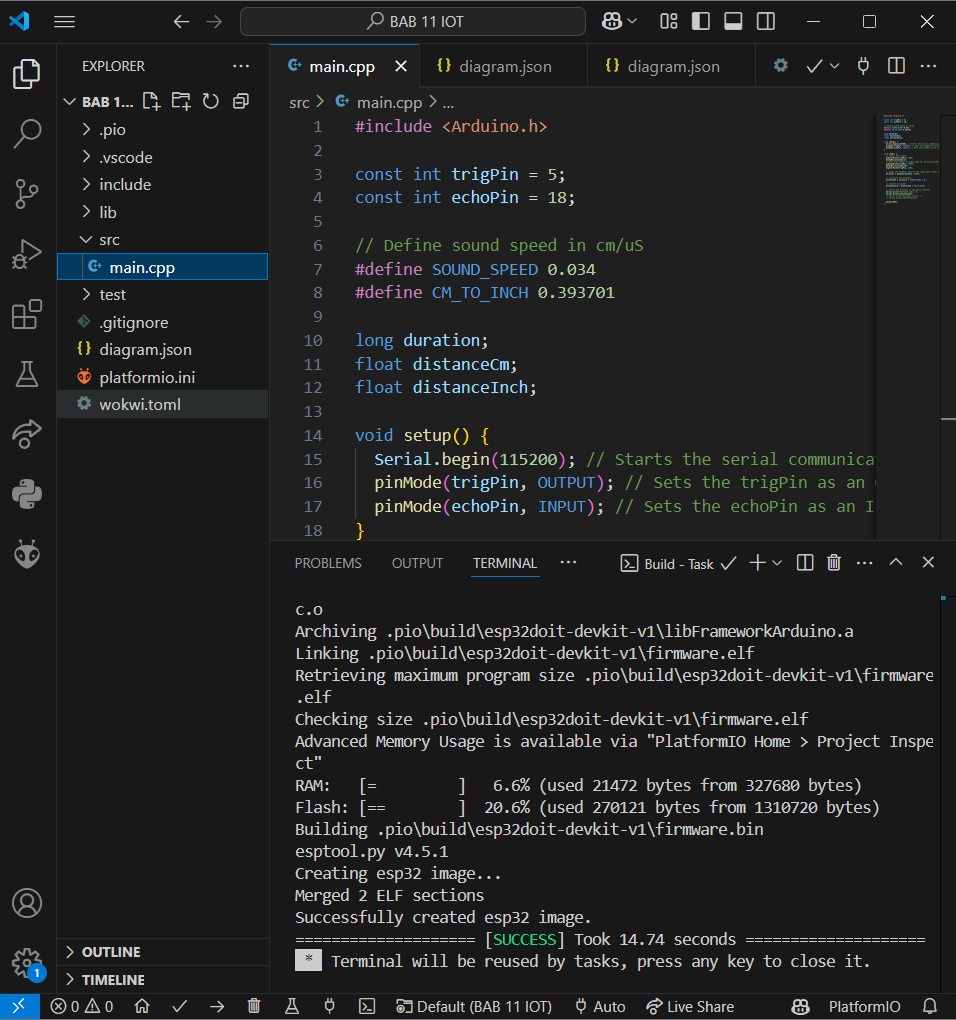
1. Buat project baru pada platform IO

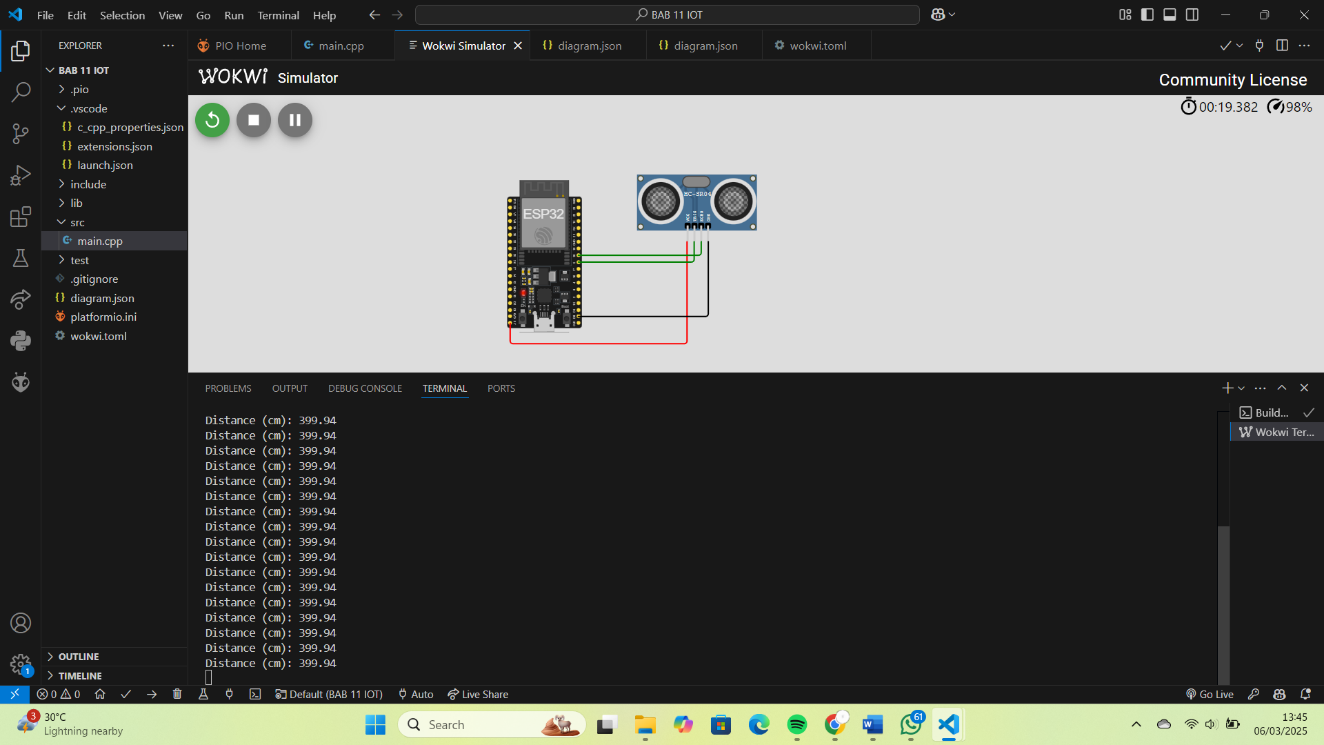
1. Tuliskan code pada file main.cpp nya



1. Lalu edit pada file platfrom.io
2. Buat file diagram.json dan copypaste kan dari diagram.json pada web wokwi.com



1. Buat file wokwi.toml dan masukkan code nya
2. Lalu compile file pada main.cpp
3. Kemudian lakukan request license ke wokwi.com
4. Jalankan simulasi dengan perintah wokwi: start simulator



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

**3.1 Experimental Results (Hasil Eksperimen)**

Pada simulasi menggunakan wokwi, sensor ultrasomik HC-SR04 tidak memeiliki objek virtual yang adpat memantulkan gelombang ultrasonik. Akibatnya, nilai jarak pembacaan tetap pada angka 399.94 cm atau mendekati batas maksimal yang dapat dibaca oleh sensor. Hal ini disebabkan oleh ketidak refelksi gelombang suara, sehingga berfungsi pulseIn( ) tidak menerima sinyal pantulan dan menghasilkan nilai waktu perjalanan gelombang yang terlalu besar atau bahkan nol.

Selain itu, jika pulseIn( ) tidak mendeteksi pantulan, sistem akan mengembalikan nilai maksimum, yang dalam kasus ini dapat menyebabkan hasil pembacaan yang tidak akurat. Untuk mengatasi hal ini, salah satu solusinya adalah menambahkan batas waktu (timeout) pada pulseIn( ), sehingga jika dalam waktu tertentu tidak ada pantulan deteksi, program menghinadari kesalahan pembacaan dan memberikan notifikasi bahwa tidak ada objek dalam jangkauan sensor.

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

#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 microseconds

  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);

}