LAPORAN PRAKTIKUM INTERNET OF THINGS (IOT)

Fakultas Vokasi, Universitas Brawijaya **Praktik Akses API Melalui Simulasi WOKWI**

*Raka Sanjaya – 233140701111018*

*Fakultas Vokasi, Universitas Brawijaya*

*Email: sanjayaraka722@gmail.com*

**Abstrak**

ESP32 merupakan mikrokontroler yang memiliki kemampuan tinggi serta fitur konektivitas nirkabel seperti Wi-Fi dan Bluetooth, yang banyak digunakan dalam pengembangan sistem IoT (Internet of Things). Praktik ini bertujuan untuk memberikan pemahaman secara langsung kepada mahasiswa mengenai cara kerja ESP32 dalam lingkungan perangkat keras nyata. Melalui praktikum ini, diharapkan dapat mengenal karakteristik ESP32, memahami cara menghubungkannya dengan sensor, serta mampu mengembangkan aplikasi sederhana berbasis ESP32. Praktik dilakukan dengan merancang rangkaian sederhana yang melibatkan input dan output digital, serta pengujian komunikasi data. Hasil dari praktik ini menunjukkan bahwa ESP32 sangat cocok digunakan untuk pengembangan sistem IoT skala kecil hingga menengah.

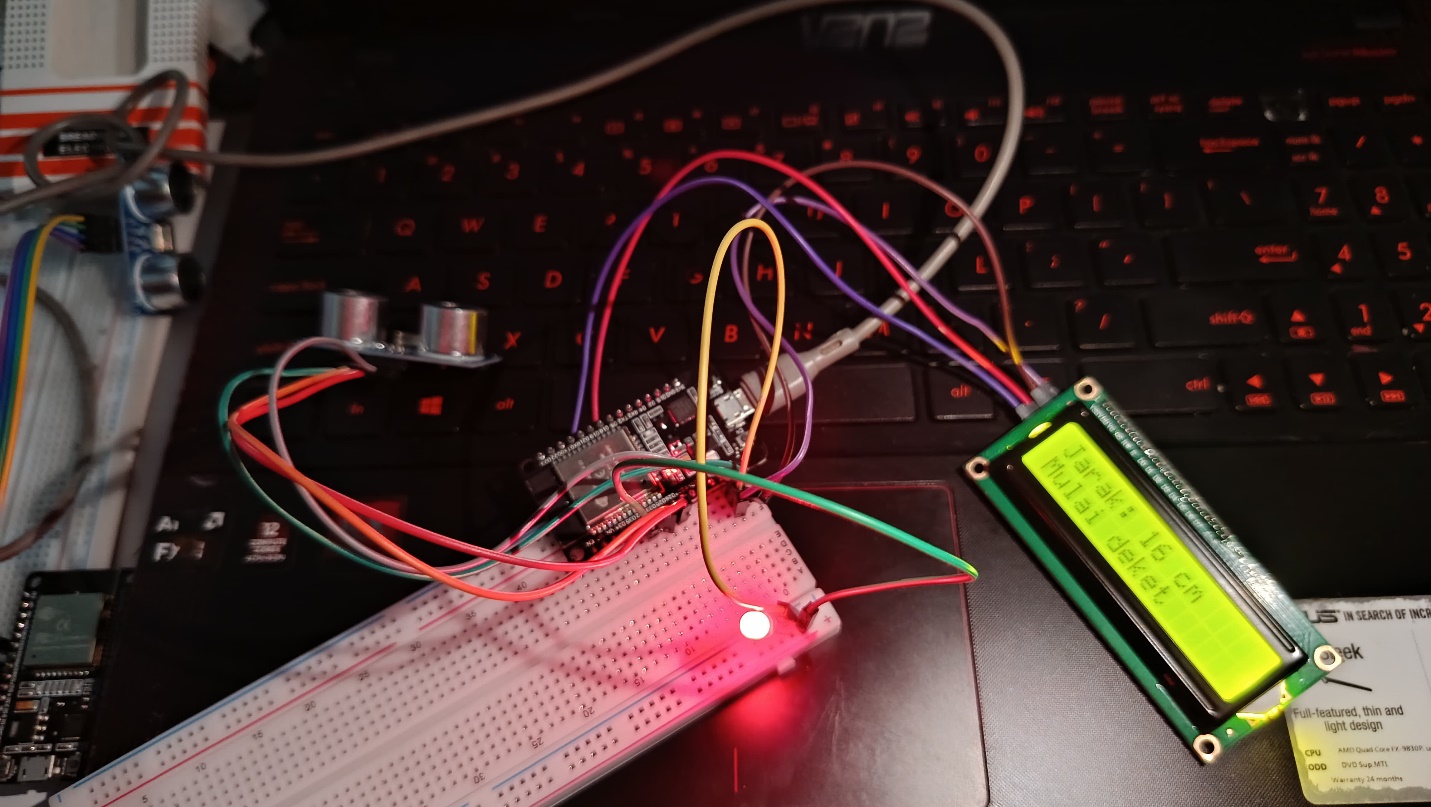
*Keyword : ESP32, IoT, sensor*

1. **Pendahuluan** 
   1. **Latar Belakang**

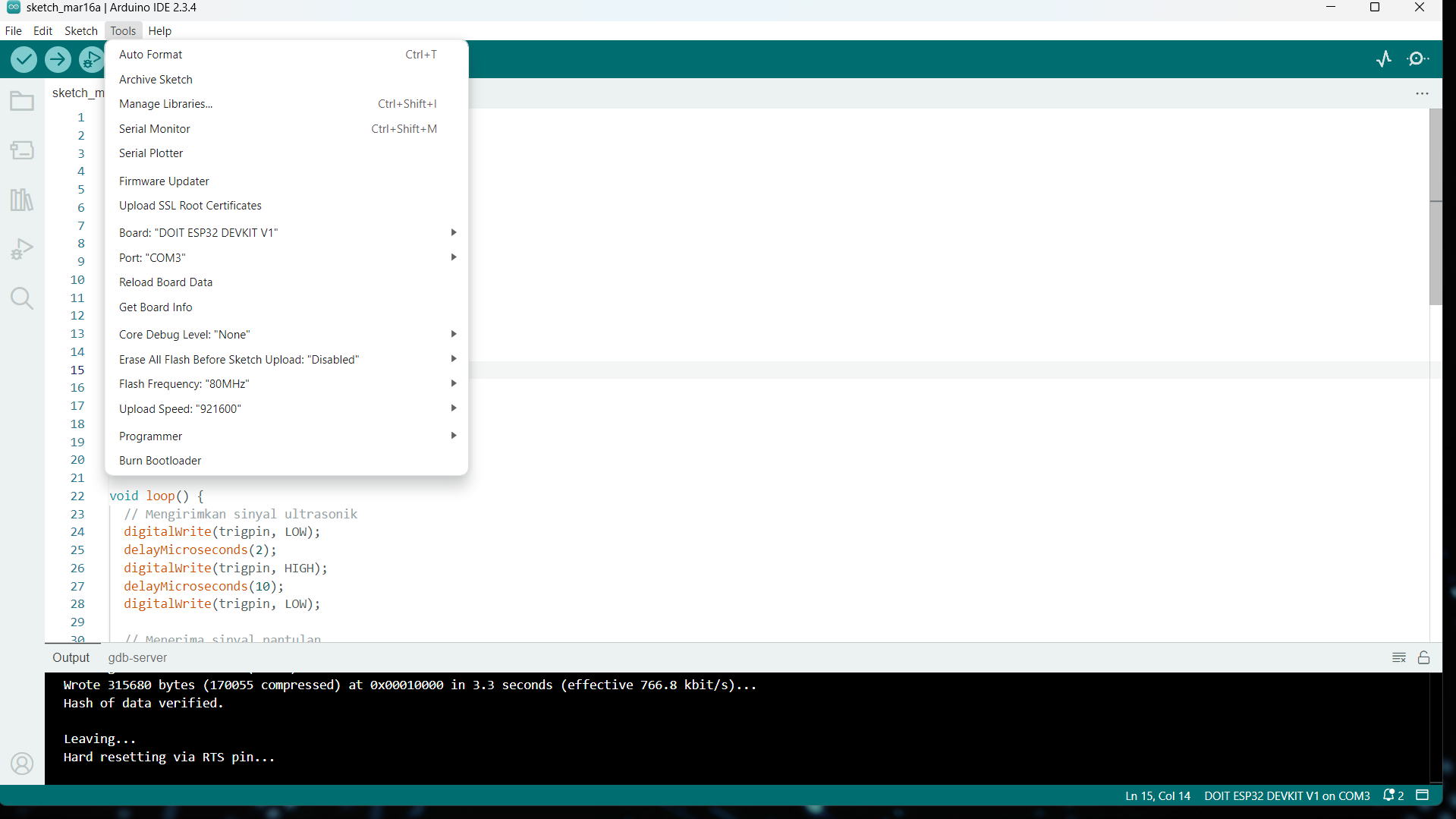
Perkembangan teknologi IoT (Internet of Things) sangat pesat dalam beberapa tahun terakhir. Hal ini didukung oleh hadirnya mikrokontroler modern seperti ESP32 yang memiliki kemampuan pemrosesan tinggi serta dukungan komunikasi nirkabel. Dalam dunia akademik dan industri, ESP32 menjadi salah satu pilihan utama untuk pengembangan berbagai perangkat pintar seperti smart home, monitoring sistem, dan automasi industri. Oleh karena itu, pemahaman terhadap ESP32 tidak hanya penting secara teoritis, tetapi juga harus dikuasai melalui praktik langsung. Praktik Real Hardware ESP32 ini menjadi wadah bagi mahasiswa untuk menerapkan teori yang telah dipelajari dalam bentuk rangkaian nyata dan aplikasi dasar.

* 1. **Tujuan Eksperimen**

1. Mengenalkan komponen dan fungsi dasar ESP32..
2. Mempelajari cara menghubungkan ESP32 dengan perangkat input/output seperti LED, push button, dan sensor.
3. Mengembangkan program sederhana untuk mengendalikan output berdasarkan input.
4. Menguji kemampuan komunikasi data melalui antarmuka serial atau nirkabel.
5. Menumbuhkan keterampilan mahasiswa dalam merancang dan mengimplementasikan sistem berbasis mikrokontroler.
6. **Metodologi**
   1. **Alat dan Bahan**
7. ESP32
8. Breadboar
9. LED
10. Kabel jumper
11. Sensor ultrasonik
12. LCD I2C
    1. **Langkah Implementasi**
13. Sambungkan esp32,sensor ultrasonik led dan lcd I2C dengan breadboard



1. Buka arduino IDE dan koneksikan esp dengan laptop pastikan terkoneksi :



Lalu masukan kode

 #include <Arduino.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

const int ledM = 32;

const int trigpin = 25;

const int echopin = 26;

long durasi;

int jarak;

void setup() {

  Serial.begin(115200);

  lcd.init();

  lcd.backlight();

  pinMode(trigpin, OUTPUT);

  pinMode(echopin, INPUT);

  pinMode(ledM, OUTPUT);

}

void loop() {

  // Mengirimkan sinyal ultrasonik

  digitalWrite(trigpin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigpin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigpin, LOW);

  // Menerima sinyal pantulan

  durasi = pulseIn(echopin, HIGH);

  jarak = durasi \* 0.034 / 2; // Mengubah Durasi Menjadi Jarak (kecepatan suara 0.034 cm/us)

  // Menampilkan jarak ke Serial Monitor dan LCD

  Serial.print("Jarak: ");

  Serial.print(jarak);

  Serial.println(" cm");

  lcd.setCursor(0, 0);

  lcd.print("Jarak: ");

  lcd.print(jarak);

  lcd.print(" cm  ");  // Spasi ekstra untuk menghapus karakter lama

  // Logika untuk LED dan pesan LCD

  if (jarak <= 5) {

    Serial.println("DEKET BANGET OI");

    lcd.setCursor(0, 1);

    lcd.print("DEKET BANGET OI ");

    kedipLED(100); // LED berkedip cepat

  }

  else if (jarak <= 10) {

    Serial.println("Deket banget");

    lcd.setCursor(0, 1);

    lcd.print("Deket banget   ");

    kedipLED(200); // LED berkedip sedang

  }

  else if (jarak <= 30) {

    Serial.println("Mulai deket");

    lcd.setCursor(0, 1);

    lcd.print("Mulai deket    ");

    kedipLED(500); // LED berkedip lambat

  }

  else {

    Serial.println("Jarak aman");

    lcd.setCursor(0, 1);

    lcd.print("Jarak aman     ");

    digitalWrite(ledM, LOW);

  }

  delay(500); // Tunggu sebelum membaca lagi

}

// Fungsi untuk mengatur kedipan LED tanpa menghambat loop

void kedipLED(int waktu) {

  digitalWrite(ledM, HIGH);

  delay(waktu);

  digitalWrite(ledM, LOW);

  delay(waktu);

}

1. Tekan menu start untuk percobaan apakah berhasil



1. **Hasil dan Pembahasan** 
   1. **Hasil Eksperimen**

Hasil dari praktikum ini adalah dapat mengetahui jarak dan tampilan dari jarak yang dimana ketika jarak tertentu mneghasilkan kedipan lampu yang berbeda dan tampilan pada lcd yang berbeda.

