LAPORAN PRAKTIKUM INTERNET OF THINGS (IoT)

Fakultas Vokasi, Universitas Brawijaya

**Praktik Real Hardware ESP32**

*Pradipta Rahmatan Isya Hertanto*

*Fakultas Vokasi, Universitas Brawijaya*

*Email :* [*praadipta19@student.ub.ac.id*](mailto:praadipta19@student.ub.ac.id)

**Abstrak**

ESP32 merupakan salah satu mikrokontroler modern yang banyak digunakan dalam pengembangan perangkat berbasis Internet of Things (IoT) berkat dukungan fitur komunikasi nirkabel seperti Wi-Fi dan Bluetooth. Kegiatan praktikum ini dirancang untuk memberikan pengalaman langsung dalam mengoperasikan ESP32 di lingkungan sistem tertanam. Mahasiswa diajak untuk merancang dan menguji rangkaian sederhana yang melibatkan input-output digital serta integrasi dengan sensor. Melalui eksperimen ini, mahasiswa diharapkan mampu memahami prinsip kerja ESP32, melakukan integrasi dengan perangkat eksternal, serta membangun aplikasi IoT sederhana. Hasil praktik menunjukkan bahwa ESP32 memiliki fleksibilitas dan performa yang baik untuk dijadikan platform pengembangan proyek IoT skala kecil.

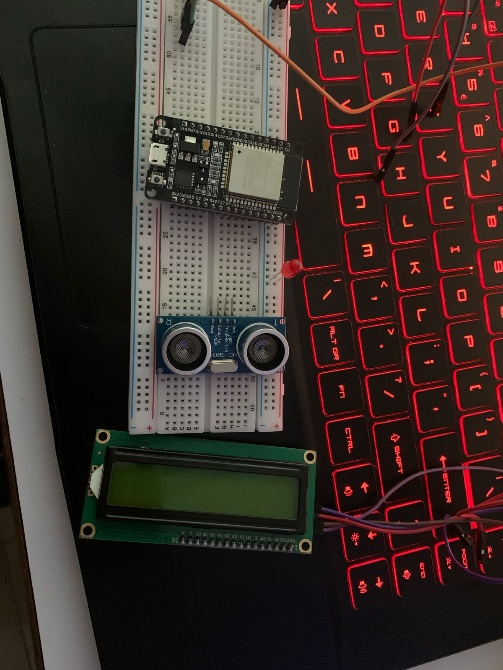
*Keyword : ESP32, IoT, sensor*

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

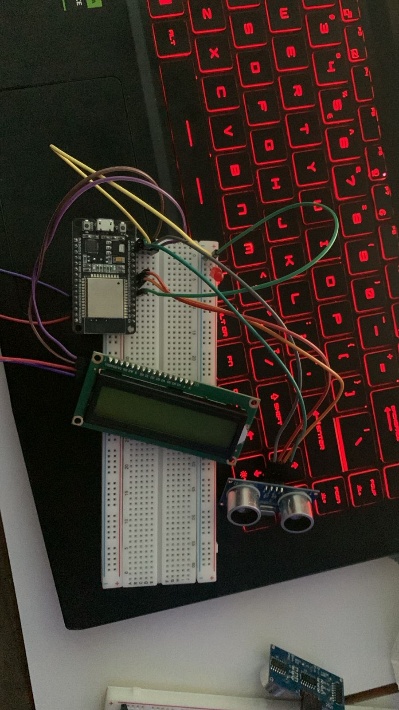
Kemajuan teknologi IoT telah mendorong kebutuhan akan perangkat mikrokontroler yang andal dan fleksibel. ESP32 menjadi salah satu solusi populer karena memiliki kemampuan pemrosesan yang tinggi serta fitur konektivitas nirkabel yang lengkap. Dalam konteks pendidikan, penting bagi mahasiswa untuk tidak hanya memahami konsep IoT secara teori, tetapi juga menguasai keterampilan praktis dalam merancang dan mengimplementasikan sistem nyata. Praktikum berbasis Real Hardware ESP32 ini memberikan kesempatan untuk mengeksplorasi langsung penggunaan ESP32 melalui rangkaian dasar yang mencakup sensor, aktuator, serta antarmuka komunikasi data. Melalui pendekatan ini, mahasiswa dapat membangun fondasi keterampilan teknis yang dibutuhkan dalam pengembangan sistem IoT yang lebih kompleks di masa depan.

* 1. **Tujuan Eksperimen**

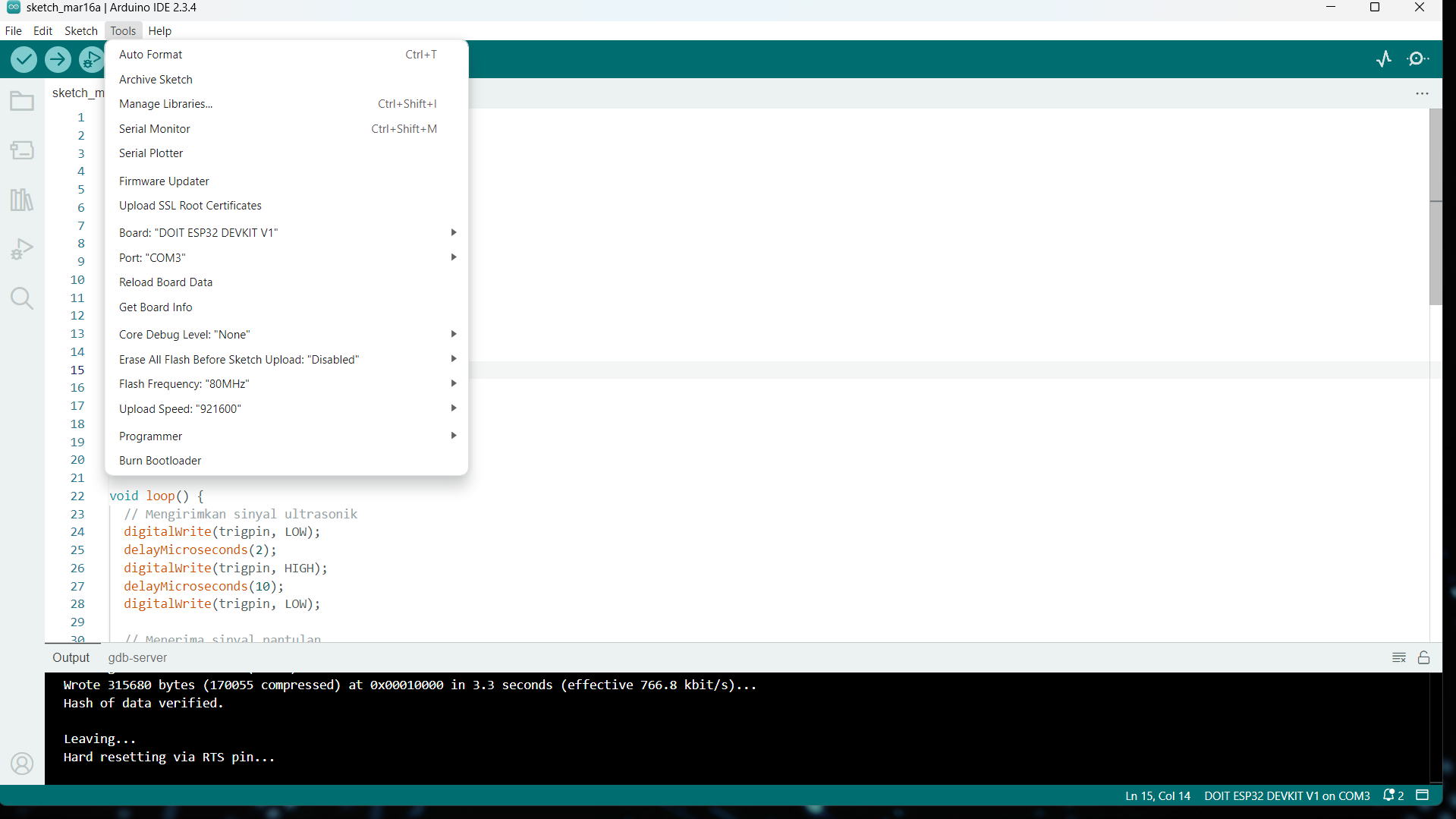
1. Memberikan pemahaman dasar mengenai arsitektur dan fitur utama dari ESP32.
2. Melatih keterampilan dalam menghubungkan ESP32 dengan berbagai perangkat input/output seperti sensor ultrasonik, LED, dan LCD.
3. Menyusun program kendali sederhana berbasis logika input dan output digital.
4. Mengevaluasi kemampuan ESP32 dalam mengirim dan menerima data melalui komunikasi serial atau konektivitas nirkabel.
5. Mendorong pengembangan keterampilan praktis mahasiswa dalam membangun solusi sistem tertanam berbasis IoT.
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. Menyiapkan ESP32, Sensor Ultrasonik, lcd I2C, dan Led



1. 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);

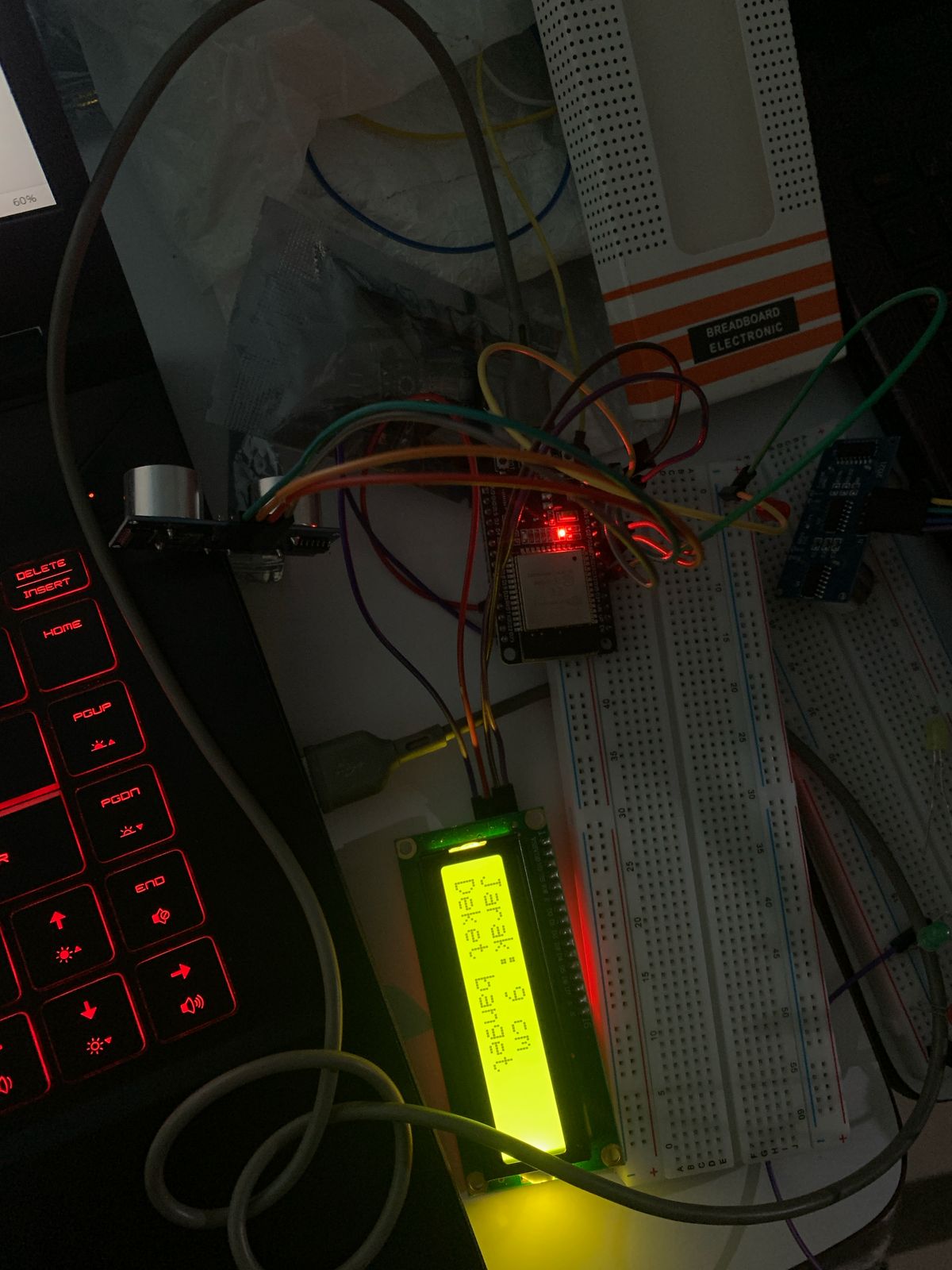
  delay(waktu);

}

1. Tekan menu start untuk percobaan apakah berhasil



1. Alat sudah dibuat



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

. 