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

**Fakultas Vokasi, Universitas Brawijaya**



**Praktik Real Hardware ESP32**

*Angger Abed Nego*

Fakultas Vokasi, Universitas Brawijaya

Email : [bedvyy7@student.ub.ac.id](mailto:bedvyy7@student.ub.ac.id)

* **Abstrack**

ESP32 merupakan mikrokontroler dengan performa tinggi yang dilengkapi fitur konektivitas nirkabel seperti Wi-Fi dan Bluetooth, sehingga banyak dimanfaatkan dalam pengembangan sistem IoT (Internet of Things). Praktikum ini bertujuan untuk memberikan pemahaman langsung kepada mahasiswa mengenai cara kerja ESP32 dalam lingkungan perangkat keras nyata. Melalui kegiatan ini, mahasiswa diharapkan dapat mengenal karakteristik ESP32, memahami cara menghubungkannya dengan sensor, serta mampu mengembangkan aplikasi sederhana berbasis ESP32. Praktikum dilakukan dengan merancang rangkaian sederhana yang melibatkan input dan output digital, serta melakukan pengujian komunikasi data. Hasil dari praktik menunjukkan bahwa ESP32 sangat cocok digunakan dalam pengembangan sistem IoT skala kecil hingga menengah.  
**Kata kunci**: ESP32, IoT, sensorKeywords: Internet of Things, ESP32, Wokwi, API, HTTP Request

1. **Introduction**

1.1 **Latar Belakang**

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

1.2 **Tujuan**

Tujuan dari eksperimen ini adalah:

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.

**Metodhlogy**

2.1 **Alat dan Bahan**

1. ESP32
2. Breadboard
3. LED
4. Kabel jumper
5. Sensor ultrasonik
6. LCD I2C
   1. **Langkah Implementasi**
   2. Menyiapkan ESP32, Sensor Ultrasonik, lcd I2C, dan Led

|  |
| --- |
|  |

* 1. Sambungkan esp32,sensor ultrasonik led dan lcd I2C dengan breadboard

|  |
| --- |
|  |

* 1. Masukkan kode pada Arduino

|  |
| --- |
| 1. #include <Arduino.h> 2. #include <Wire.h> 3. #include <LiquidCrystal\_I2C.h> 4. LiquidCrystal\_I2C lcd(0x27, 16, 2); 5. const int ledM = 32; 6. const int trigpin = 25; 7. const int echopin = 26; 8. long durasi; 9. int jarak; 10. void setup() { 11. Serial.begin(115200); 12. lcd.init(); 13. lcd.backlight(); 14. pinMode(trigpin, OUTPUT); 15. pinMode(echopin, INPUT); 16. pinMode(ledM, OUTPUT); 17. } 18. void loop() { 19. // Mengirimkan sinyal ultrasonik 20. digitalWrite(trigpin, LOW); 21. delayMicroseconds(2); 22. digitalWrite(trigpin, HIGH); 23. delayMicroseconds(10); 24. digitalWrite(trigpin, LOW); 25. // Menerima sinyal pantulan 26. durasi = pulseIn(echopin, HIGH); 27. jarak = durasi \* 0.034 / 2; // Mengubah Durasi Menjadi Jarak (kecepatan suara 0.034 cm/us) 28. // Menampilkan jarak ke Serial Monitor dan LCD 29. Serial.print("Jarak: "); 30. Serial.print(jarak); 31. Serial.println(" cm"); 32. lcd.setCursor(0, 0); 33. lcd.print("Jarak: "); 34. lcd.print(jarak); 35. lcd.print(" cm  ");  // Spasi ekstra untuk menghapus karakter lama 36. // Logika untuk LED dan pesan LCD 37. if (jarak <= 5) { 38. Serial.println("DEKET BANGET OI"); 39. lcd.setCursor(0, 1); 40. lcd.print("DEKET BANGET OI "); 41. kedipLED(100); // LED berkedip cepat 42. } 43. else if (jarak <= 10) { 44. Serial.println("Deket banget"); 45. lcd.setCursor(0, 1); 46. lcd.print("Deket banget   "); 47. kedipLED(200); // LED berkedip sedang 48. } 49. else if (jarak <= 30) { 50. Serial.println("Mulai deket"); 51. lcd.setCursor(0, 1); 52. lcd.print("Mulai deket    "); 53. kedipLED(500); // LED berkedip lambat 54. } 55. else { 56. Serial.println("Jarak aman"); 57. lcd.setCursor(0, 1); 58. lcd.print("Jarak aman     "); 59. digitalWrite(ledM, LOW); 60. } 61. delay(500); // Tunggu sebelum membaca lagi 62. } 63. // Fungsi untuk mengatur kedipan LED tanpa menghambat loop 64. void kedipLED(int waktu) { 65. digitalWrite(ledM, HIGH); 66. delay(waktu); 67. digitalWrite(ledM, LOW); 68. delay(waktu); 69. } |

1. **Result and Disscusion**

3.1 **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

|  |
| --- |
|  |