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

**Fakultas Vokasi, Universitas Brawijaya**



**Praktikum Bab 10-13**

*Angger Abed Nego*

Fakultas Vokasi, Universitas Brawijaya

Email : bedvyy7@student.ub.ac.id

* **Abstrack**

Eksperimen ini bertujuan untuk menganalisis implementasi simulasi relay, sensor jarak, dan integrasi API Laravel dalam sistem IoT. Simulasi relay dikendalikan menggunakan mikrokontroler ESP32, sementara sensor jarak menggunakan modul ultrasonik. API dikembangkan menggunakan Laravel untuk menyimpan dan mengelola data sensor. Hasil eksperimen menunjukkan bahwa modul relay dan sensor bekerja dengan efisiensi tinggi dengan latensi minimal, serta API berhasil menangani transaksi data dengan baik.

*Kata Kunci*—Internet of Things, ESP32, Sensor Ultrasonik, Relay, Laravel API

**Bab 10**   
**Praktik Simulasi ESP32 Relay, Button & LED**

1. **Introduction**

1.1 **Latar Belakang**   
 Sistem kendali berbasis mikrokontroler semakin banyak digunakan dalam kehidupan sehari-hari, salah satunya adalah pengendalian relay untuk perangkat elektronik. Relay dapat digunakan untuk mengontrol peralatan dengan daya tinggi menggunakan sinyal digital dari mikrokontrole

1.2 **Tujuan**

* Mengimplementasikan simulasi relay menggunakan ESP32.
* Mengontrol relay menggunakan tombol input.
* Mengamati respons sistem terhadap perintah kontrol.

1. **Metodhlogy**

2.1 **Alat dan Bahan**

* ESP32
* Modul Relay
* LED dan Push Button
* Software: Arduino IDE, PlatformIO

2.2 **Langkah Implementasi**

* Menghubungkan relay dengan ESP32 pada GPIO yang sesuai.
* Menulis program untuk mengontrol relay berdasarkan tombol input.
* Mengunggah kode ke ESP32 dan melakukan pengujian.

1. **Result and Disscusion**

3.1 **Hasil Eksperimen**

* Relay berhasil dikendalikan menggunakan tombol input.
* Perubahan status relay dapat diamati melalui LED indikator.

|  |
| --- |
|  |

1. **Appendix**

4.1 Kode Program

|  |
| --- |
| #include <Arduino.h>  // Define pin numbers  const int ButtonPin = 19; // GPIO19 connected to the pushbutton  const int LedPin = 18; // GPIO18 connected to the LED  const int RelayPin = 23; // GPIO23 connected to the relay module  void setup() {  // Set pin modes  pinMode(ButtonPin, INPUT\_PULLUP); // Set the button pin as an input with an internal pull-up resistor  pinMode(LedPin, OUTPUT); // Set the LED pin as an output  pinMode(RelayPin, OUTPUT); // Set the relay pin as an output  // Initialize the outputs to be OFF  digitalWrite(LedPin, LOW);  digitalWrite(RelayPin, LOW);  }  void loop() {  // Read the state of the button  int buttonState = digitalRead(ButtonPin);  // Check if the button is pressed  // Since the button is wired to pull the pin LOW when pressed, we check for LOW  if (buttonState == LOW) {  digitalWrite(LedPin, HIGH); // Turn on the LED  digitalWrite(RelayPin, HIGH); // Turn on the relay  } else {  digitalWrite(LedPin, LOW); // Turn off the LED  digitalWrite(RelayPin, LOW); // Turn off the relay  }  } |

**Bab 11**  
**Praktik Simulasi Sensor Jarak (Ultrasonic)**

**1. Introduction**

1.1 **Latar Belakang**   
 Sensor jarak ultrasonik digunakan dalam berbagai aplikasi seperti robotika dan sistem parkir otomatis. Sensor ini bekerja dengan mengukur waktu tempuh gelombang suara yang dipantulkan oleh objek.

1.2 **Tujuan**

* Menggunakan sensor ultrasonik untuk mengukur jarak.
* Menampilkan hasil pengukuran di Serial Monitor.
* Mengevaluasi akurasi dan kecepatan respons sensor.

**2. Metodhlogy**

2.1 **Alat dan Bahan**

* ESP32
* Sensor Ultrasonik HC-SR04
* Software: Arduino IDE, PlatformIO

2.2 **Langkah Implementasi**

* Menghubungkan sensor ultrasonik ke ESP32.
* Menulis program untuk membaca data dari sensor dan menghitung jarak.
* Menampilkan hasil pengukuran pada Serial Monitor.

**3. Result and Disscusion**

3.1 **Hasil Eksperimen**

* Sensor ultrasonik berhasil mengukur jarak dengan akurasi yang cukup tinggi.
* Data jarak dapat diamati secara real-time melalui Serial Monitor.

|  |
| --- |
|  |

**4. Appendix**

4.1 Code Program

|  |
| --- |
| #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 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);  } |

**Bab 12**  
**Praktik Pembuatan API Menggunakan Laravel 11 dan Ngrok**

**1. Introduction**

1.1 **Latar Belakang**   
 Dalam sistem IoT, pengelolaan data sensor secara terpusat sangat penting. API berbasis Laravel digunakan untuk menerima, menyimpan, dan menyediakan data sensor secara efisien.

1.2 **Tujuan**

* Mengembangkan API untuk menerima data dari sensor.
* Menyimpan data ke dalam database.
* Mengakses data melalui endpoint API.

**2. Metodhlogy**

2.1 **Alat dan Bahan**

* Framework Laravel
* Database MySQL
* Ngrok untuk pengujian API

2.2 **Langkah Implementasi**

* Membuat proyek Laravel dan mengkonfigurasi database.
* Membuat route API untuk menerima data dari sensor.
* Menguji endpoint API menggunakan Postman atau perangkat IoT.

**3. Result and Disscusion**

3.1 **Hasil Eksperimen**

|  |
| --- |
|  |