**Perancangan Sensor Jarak untuk Kendaraan Motor Disabilitas dan Integrasi Output Proyek Menggunakan Blynk Code Pada Web Browser**

*Ibnu Jaisyurrahman Faiz*

*Program Studi Teknologi Informasi, Fakultas Vokasi, Universitas Brawijaya*

*Email: (*[*ibnufaiz72ub@student.ub.ac.id*](mailto:ibnufaiz72ub@student.ub.ac.id)*)*

**Abstrak:** Proyek ini merancang sistem sensor jarak berbasis IoT untuk kendaraan motor penyandang disabilitas. Sistem menggunakan ESP32, sensor HC-SR04, LED, dan buzzer sebagai alat deteksi dan peringatan terhadap objek di sekitar kendaraan. Data jarak ditampilkan secara real-time melalui platform Blynk yang terhubung ke web browser dan aplikasi. Hasil pengujian menunjukkan sistem mampu mendeteksi objek dalam jarak <100 cm secara akurat dan memberikan respons peringatan visual dan suara. Solusi ini meningkatkan keamanan dan kenyamanan berkendara bagi pengguna disabilitas.

Kata Kunci : *ESP32, HC-SR04, IoT, Blynk, Disabilitas, Sensor Jarak, Kendaraan Motor*

**Abstract:** This project designs an IoT-based distance sensor system for motorcycles used by people with disabilities. The system uses an ESP32 microcontroller, HC-SR04 ultrasonic sensor, LEDs, and a buzzer to detect nearby objects and provide visual and audio alerts. Distance data is monitored in real-time via the Blynk platform, accessible through a web browser and mobile application. Testing shows the system accurately detects objects within a range of less than 100 cm and responds effectively. This solution enhances riding safety and convenience for disabled users.

Key Word: *ESP32, HC-SR04, IoT, Blynk, Disability, Distance Sensor, Motorcycle*

1. **Pendahuluan**
2. **Latar belakang**

Penyandang disabilitas yang menggunakan kendaraan motor khusus sering menghadapi kesulitan dalam memperkirakan jarak terhadap objek di sekitar, terutama saat parkir atau berkendara di ruang sempit. Keterbatasan ini dapat meningkatkan risiko terjadinya tabrakan ringan dan membahayakan keselamatan pengguna. Untuk itu, dibutuhkan sistem bantu yang dapat mendeteksi objek secara otomatis dan memberikan peringatan secara visual maupun suara.

Dengan memanfaatkan teknologi Internet of Things (IoT), sistem sensor jarak dapat dirancang menggunakan ESP32 dan sensor ultrasonik HC-SR04. Data hasil deteksi kemudian ditampilkan secara real-time melalui platform Blynk yang dapat diakses lewat web browser maupun aplikasi seluler. Integrasi ini bertujuan untuk memberikan solusi yang aman, responsif, dan mudah dipantau bagi pengguna kendaraan motor disabilitas.

1. **Tujuan eksperimen**

* Merancang sistem sensor jarak berbasis mikrokontroler ESP32 yang dapat mendeteksi keberadaan objek di sekitar kendaraan motor.
* Mengintegrasikan sensor ultrasonik HC-SR04 untuk mengukur jarak objek secara akurat dan real-time.
* Menyediakan peringatan dini bagi pengendara motor disabilitas melalui indikator suara (buzzer) dan visual (LED) saat terdapat objek dalam jarak dekat.
* Menghubungkan sistem dengan platform IoT Blynk agar data jarak dapat dipantau secara langsung melalui aplikasi di perangkat seluler.
* Meningkatkan keselamatan dan kenyamanan pengendara motor disabilitas dalam berkendara dan parkir, serta mengurangi risiko kecelakaan ringan akibat keterbatasan persepsi jarak.

1. **Metodologi**
2. **Alat dan Bahan**

Pada eksperimen ini, simulasi dilakukan menggunakan Wokwi dan implementasi lebih lanjut dilakukan di Visual Studio Code (VS Code). Berikut adalah daftar alat dan bahan yang digunakan:

* Wokwi Simulator – Untuk simulasi rangkaian secara virtual sebelum implementasi fisik.
* Visual Studio Code (VS Code) – Untuk menulis dan mengunggah kode ke ESP32.
* Arduino IDE / PlatformIO – Plugin tambahan untuk pengembangan kode pada VS Code.
* ESP32 – Mikrokontroler utama yang mengontrol sistem.
* Sensor Ultrasonik HC-SR04
* Buzzer
* LED

1. **Langkah Implementasi**

Langkah implementasi untuk membuat Perancangan Sensor Jarak untuk Kendaraan Motor Disabilitas dan Integrasi Output Proyek Menggunakan Blynk sebagai berikut:

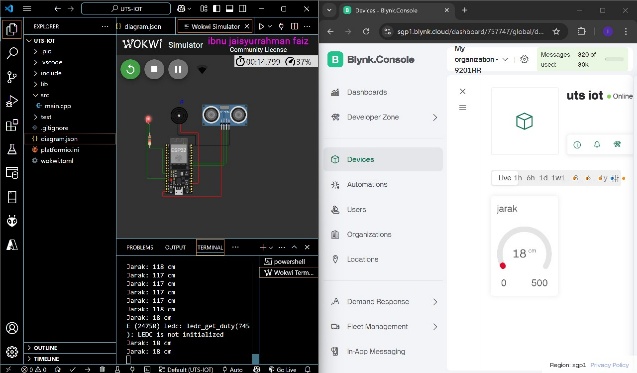
* Merancang sistem sensor jarak menggunakan ESP32 dan sensor ultrasonik HC-SR04 untuk mendeteksi objek di sekitar kendaraan motor disabilitas.
* Menambahkan LED merah sebagai indikator visual dan buzzer sebagai indikator suara saat objek terdeteksi pada jarak tertentu.
* Melakukan simulasi awal di Wokwi untuk memastikan rangkaian dan logika berjalan sesuai.
* Menghubungkan ESP32 ke platform Blynk melalui Wi-Fi untuk mengirim data jarak secara real-time.
* Menampilkan data jarak pada dashboard Blynk (via aplikasi dan web browser) menggunakan widget seperti Gauge atau LCD Display.
* Melakukan pengujian untuk memastikan akurasi sensor, respon LED/buzzer, dan kestabilan pengiriman data ke Blynk.

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

Pada tahap simulasi, sistem sensor jarak berbasis ESP32 berhasil dirancang sesuai dengan rencana awal. Struktur program telah berjalan dengan baik menggunakan Visual Studio Code, dan sistem mampu merespon skenario jarak yang disimulasikan dengan benar. Sensor ultrasonik HC-SR04 berhasil diintegrasikan secara logis, meskipun masih menggunakan input jarak buatan karena belum diuji langsung pada perangkat fisik. Sistem juga telah dilengkapi dengan indikator berupa buzzer dan LED, yang divisualisasikan melalui antarmuka aplikasi Blynk sebagai pengganti output fisik selama simulasi berlangsung.

Respon sistem terhadap perubahan jarak ini juga berjalan sesuai dengan logika program yang telah dirancang, yaitu:

* LED menyala ketika objek berada dalam jarak tertentu (misalnya di bawah 50 cm).
* Buzzer aktif mengeluarkan bunyi sebagai tanda peringatan.
* Data jarak dikirim dan ditampilkan pada dashboard Blynk melalui widget dengan label “jarak”.



1. **Lampiran**

* Kode program main.cpp
* #define BLYNK\_PRINT Serial
* #define BLYNK\_TEMPLATE\_ID "TMPL6ZKCyS-tg"
* #define BLYNK\_TEMPLATE\_NAME "uts iot"
* #define BLYNK\_AUTH\_TOKEN "B2YUkJGiZdlG3FHz7qEwNRzm7E\_7omTz"
* #include <WiFi.h>
* #include <BlynkSimpleEsp32.h>
* // ------------------- Blynk Config -------------------
* char auth[] = BLYNK\_AUTH\_TOKEN;
* char ssid[] = "Wokwi-GUEST";
* char pass[] = "";
* BlynkTimer timer;
* // ------------------- Pin Setup -------------------
* #define TRIG\_PIN 18
* #define ECHO\_PIN 19
* #define LED\_PIN 14
* #define BUZZER\_PIN 15
* // ------------------- Variabel Jarak -------------------
* long duration;
* int distance;
* // ------------------- Fungsi Efek -------------------
* void kondisiDarurat() {
* digitalWrite(LED\_PIN, HIGH);
* for (int i = 0; i < 5; i++) {
* tone(BUZZER\_PIN, 2000, 100);
* delay(110);
* }
* }
* void kedapKedipCepat() {
* digitalWrite(LED\_PIN, HIGH);
* delay(100);
* digitalWrite(LED\_PIN, LOW);
* tone(BUZZER\_PIN, 1000, 100);
* delay(150);
* }
* void kedapKedipLambat() {
* digitalWrite(LED\_PIN, HIGH);
* delay(500);
* digitalWrite(LED\_PIN, LOW);
* tone(BUZZER\_PIN, 500, 100);
* delay(500);
* }
* // ------------------- Kirim Data ke Blynk & Kontrol -------------------
* void sendSensorData() {
* // Trigger sensor ultrasonik
* digitalWrite(TRIG\_PIN, LOW);
* delayMicroseconds(2);
* digitalWrite(TRIG\_PIN, HIGH);
* delayMicroseconds(10);
* digitalWrite(TRIG\_PIN, LOW);
* duration = pulseIn(ECHO\_PIN, HIGH);
* distance = duration \* 0.034 / 2;
* // Kirim ke Blynk
* Blynk.virtualWrite(V0, distance);
* // Log jarak ke Serial
* Serial.print("Jarak: ");
* Serial.print(distance);
* Serial.println(" cm");
* // Logika kontrol
* if (distance < 15) {
* kondisiDarurat();
* Serial.println("⚠ Jarak sangat dekat! Risiko tabrakan.");
* } else if (distance <= 50) {
* kedapKedipCepat();
* } else if (distance <= 100) {
* kedapKedipLambat();
* } else if (distance <= 150) {
* digitalWrite(LED\_PIN, HIGH);
* digitalWrite(BUZZER\_PIN, LOW);
* } else {
* digitalWrite(LED\_PIN, LOW);
* digitalWrite(BUZZER\_PIN, LOW);
* }
* }
* // ------------------- Setup -------------------
* void setup() {
* Serial.begin(115200);
* Blynk.begin(auth, ssid, pass);
* pinMode(TRIG\_PIN, OUTPUT);
* pinMode(ECHO\_PIN, INPUT);
* pinMode(LED\_PIN, OUTPUT);
* pinMode(BUZZER\_PIN, OUTPUT);
* Serial.println("🚀 Sistem Monitoring Jarak Siap!");
* timer.setInterval(1000L, sendSensorData);
* }
* // ------------------- Loop -------------------
* void loop() {
* Blynk.run();
* timer.run();
* }
* Kode program diagram.json

{

"version": 1,

"author": "Anonymous maker",

"editor": "wokwi",

"parts": [

{ "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },

{

"type": "wokwi-buzzer",

"id": "bz1",

"top": -122.4,

"left": 21,

"attrs": { "volume": "0.1" }

},

{ "type": "wokwi-hc-sr04", "id": "ultrasonic1", "top": -123.3, "left": 139.9, "attrs": {} },

{ "type": "wokwi-led", "id": "led1", "top": -90, "left": -82.6, "attrs": { "color": "red" } },

{

"type": "wokwi-resistor",

"id": "r1",

"top": 4.8,

"left": -86.95,

"rotate": 90,

"attrs": { "value": "1000" }

}

],

"connections": [

[ "esp:TX", "$serialMonitor:RX", "", [] ],

[ "esp:RX", "$serialMonitor:TX", "", [] ],

[ "bz1:1", "esp:GND.2", "black", [ "v28.8", "h67.2", "v38.4" ] ],

[ "bz1:2", "esp:15", "red", [ "v19.2", "h66.8", "v192" ] ],

[ "led1:A", "r1:1", "green", [ "v0" ] ],

[ "r1:2", "esp:14", "green", [ "h48", "v94.8" ] ],

[ "led1:C", "esp:GND.1", "green", [ "v0" ] ],

[ "ultrasonic1:VCC", "esp:5V", "red", [ "v249.6", "h-220.8", "v-19.2" ] ],

[ "ultrasonic1:TRIG", "esp:18", "green", [ "v0" ] ],

[ "ultrasonic1:ECHO", "esp:19", "green", [ "v0" ] ],

[ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ]

],

"dependencies": {}

}