

Sistem Penanda Dan Penilaian Sasaran Tembak Menggunakan IoT Berbasis Sensor Ky-031

1st Topan Budiargo
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

topanbudiargo@student.telkomuniversi
ty.ac.id

2nd Giva Andriana Mutiara
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

giva.andriana@tass.telkomuniversity.ac
.id

3rd Periyadi
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

periyadi@tass.telkomuniversity.ac.id

Abstract— Olahraga menembak merupakan olahraga kompetitif yang memerlukan kemahiran akurasi dan kecepatan dengan menggunakan berbagai jenis senjata seperti senjata api atau senapan angin. Olahraga ini berfungsi untuk melatih konsentrasi pengendalian diri. Olahraga menembak memerlukan sistem skoring atau pembacaan skor yang tepat dan akurat. Dimana jika pembacaan skor dilakukan secara manual oleh juri yang menilai akan membutuhkan waktu yang relatif lama hingga informasi dapat sampai ke penonton. Pada penelitian ini akan membuat sistem pendeteksi sasaran tembak menggunakan sensor ky-031 dan sistem display scoring berbasis IoT. Prototype ini dibangun menggunakan beberapa modul yaitu lcd2004, motor pwm dc, modul l298n, sensor ketuk ky-031, dan menggunakan dua buah esp32 sebagai pemrosesnya. Dengan ini penyampaian nilai akan lebih cepat karena setelah sensor ketuk ky-031 yang diletakan pada sasaran tembak mendapatkan inputan data akan langsung dikirim melewati wifi ke display skor. Berdasarkan hasil penelitian menunjukan bahwa akurasi pembacaan sensor rata-rata 92% dengan delay pengiriman data yang semakin lama jika jarak antara target dan display semakin jauh serta jarak konektifitas maksimal antara target dan display adalah 10 meter agar sistem dapat berfungsi dengan baik.

Keywords— Scoring system, Sensor ky-031, IoT, Sasaran Tembak.

I. PENDAHULUAN

Olahraga menembak merupakan olahraga kompetitif yang memerlukan kemahiran akurasi dan kecepatan dengan menggunakan berbagai jenis senjata seperti senjata api atau senapan angin. Olahraga ini berfungsi untuk melatih konsentrasi pengendalian diri. [1] Olahraga menembak memerlukan sistem skoring atau pembacaan skor yang tepat dan akurat. Dimana jika pembacaan skor dilakukan secara manual oleh juri yang menilai akan membutuhkan waktu yang relatif lama hingga informasi dapat sampai ke penonton. Maka dari itu, tujuan penelitian ini adalah untuk membuat sistem pendeteksi sasaran tembak menggunakan sensor ky-031 dan sistem display scoring berbasis IoT. Manfaat dari alat ini adalah untuk mempercepat penyampaian hasil skor tembakan ke penonton atau kepada si-penembak. Dimana pada penelitian ini sensor yang digunakan untuk pembacaan hasil tembakan adalah sensor ky-031 dan senapan yang digunakan adalah airsoft gun dengan peluru Ball Bullet (BB).

II. KAJIAN TEORI

A. ESP32

Modul ESP32 yang merupakan penerus dari mikrokontroler ESP8266 yang dikenalkan oleh Ekpressif System. Sama dengan pendahulunya ESP32 ini menyediakan modul WiFi dalam chipnya, dimana jangkauan wifi pada modul ini maksimal adalah 11- 15 meter [8]. yang membedakan di ESP32 dengan pendahulunya adalah mikrokontroler ini terdapat modul Bluetooth

ESP32 memiliki 18 pin ADC (Analog Digital Converter), fungsinya adalah untuk merubah sinyal analog ke digital), 2 pin DAC (Digital Analog Converter, yang merupakan kebalikan ADC), 16 pin PWM (Pulse Width Modulation), 10 sensor sentuh, 2 jalur antarmuka UART, pin antarmuka I2C, I2S, dan SPI serta memiliki tegangan operasi 3.3V.

B. Sensor Ketukan

Sensor KY-031 adalah modul sensor elektronik yang mendeteksi tumbukan (knock/tap). Sensor ini kompatibel dengan Arduino, ESP8266, ESP32, Teensy, Raspberry Pi dan platform populer lainnya.

Output sensor ini berupa digital TTL LOW/HIGH dengan daya operasi 3.3V-5V. modul ini terdiri dari resistor 10K ohm dan sensor berbasis pegas yang akan mengirimkan sinyal tinggi ketika getaran terdeteksi. Pada saat tumbukan/ketukan terdeteksi maka output akan bernilai logika LOW untuk sesaat.

C. I2C (Inter Integrated Circuit)

Inter Integrated Circuit atau biasa disingkat I2C adalah sebuah module yang biasanya dipakai untuk mengurangi penggunaan pin LCD. Sistem I2C terdiri dari saluran SCL (Serial Clock) dan SDA (Serial Data) yang membawa informasi data dari I2C ke Mikrokontroler.

I2C memiliki 4 buah pinout antara lain GND, VCC, SDA dan SCL. Dimana pin SDA dan SCL nantinya akan langsung terhubung langsung ke Mikrokontroler atau lebih tepatnya SDA ke pin 21 dan pin SCL ke pin 22.

D. LCD (Liquid Crystal Display)

Menurut Saputra, H.A. (2008), LCD (Liquid Crystal Display) merupakan komponen elektronika yang berfungsi untuk menampilkan suatu data dapat berupa karakter, huruf,

symbol maupun grafik. Karena ukurannya yang kecil maka LCD banyak dipasangkan dengan Mikrokontroler. LCD tersedia dalam bentuk modul yang mempunyai pinda, control catu daya, dan pengatur kontras [9]. Terdapat dua jenis LCD dipasaran antara lain LCD 16x2/1602 dan LCD 20x4/2004. Pada gambar 2.4 terdapat LCD 2004 atau LCD 20x4 merupakan LCD yang memiliki 20 kolom/karakter dan 4 baris, jadi totalnya LCD ini dapat menampung hingga 80 karakter. Berbeda dengan lcd 16x2 yang hanya memiliki 16 kolom/ karakter dan 2 baris yang mana lebih sedikit menampung karakter yang ditampilkan.

LCD ini dapat dihubungkan dengan modul I2C yang berguna untuk mengurangi penggunaan pin out jika ingin menghubungkannya dengan arduino. LCD 2004 membutuhkan daya sebesar 5V DC untuk mengoperasikannya. LCD digunakan untuk menampilkan pembacaan sensor dan akumulasi nilai berdasarkan pembacaan sensor.

E. Motor DC

Motor DC adalah sebuah dynamo yang akan berputar sesuai tegangan yang diberikan. Umumnya terdapat 4 jenis motor DC, antara lain Motor Brushed, Motor Brushless, Motor Servo dan Motor Stepper. Pada gambar 2.5 terdapat motor DC. Terdapat 3 bagian utama yang ada pada Motor DC sehingga dapat berputar yaitu : Kutub Medan, Dinamo dan Komutator. Untuk merancang dan mensimulasikan suatu sistem kontrol kecepatan motor DC diperlukan adanya model matematika dari plant yang akan dikontrol. Sehingga dalam memudahkan analisis sistem bisa digambarkan ke dalam model statis, model fisis (rangkainan listrik) dan dinamik (diagram blok dan transfer function) [10].

Motor DC digunakan untuk memutar roda yang dimana dapat berputar ke depan atau ke belakang sesuai kode program yang telah diberikan pada Arduino. Tegangan yang dapat diterima oleh motor DC adalah 3V – 6V dan memiliki rpm 100 – 200.

F. Driver Motor L298N

L298N merupakan sebuah modul yang sering digunakan untuk mengendalikan motor DC dalam dunia elektronika. Kelebihan akan modul driver motor L298N ini yaitu dalam hal kepresisian dalam mengontrol motor. Driver ini bisa dengan mudah mengendalikan baik kecepatan ataupun arah rotasi 2 motor sekaligus.

Driver motor L298N memiliki 16 buah pin out, tegangan input 3.2V – 40V, catu daya 5V, arus puncak 2A, dan kisaran operasi 0 – 36 mA.

III. ANALISIS DAN PERANCANGAN SISTEM

A. Gambaran Sistem Saat Ini



Gambar 1 Gambaran sistem saat ini

Saat ini sistem pendeteksi sasaran tembak ditinjau berdasarkan penelitian sebelumnya menggunakan pengolahan citra untuk membaca hasil tembakan dari target tembakan. Kamera akan membaca lubang atau bekas tembakan pada target tembakan lalu program akan mengolah hasil citra tersebut menggunakan berbagai metode untuk menentukan nilainya lalu nilai akan di tampilkan. Akan tetapi sistem saat ini memiliki kelemahan dalam hal biaya kamera yang mahal dan adanya kemungkinan tembakan mengenai kamera yang beresiko merusaknya.

B. Identifikasi kebutuhan Sistem

Dalam sistem otomatisasi penilaian hasil tembakan jelas mempermudah juri maupun penonton untuk mendapatkan nilai dari hasil tembakan dengan cepat dan akurat. Berdasarkan gambaran sistem saat ini menghasilkan identifikasi untuk membuat sistem penanda dan penilaian sasaran tembak menggunakan IoT berbasis sensor KY- 031.

Untuk membangun sistem ini terdapat beberapa kebutuhan sistem, baik kebutuhan fungsional maupun non fungsional.

Kebutuhan Fungsional :

Tabel 1 Kebutuhan Fungsional

No.	Kebutuhan Fungsional
1.	Membutuhkan sistem untuk saling bertukar data antara target dan sasaran
2.	Membutuhkan sensor pendeteksi titik tembakan
3.	Menentukan dan Menghitung skor
4.	Menampilkan Jumlah skor

Kebutuhan Non Fungsional :

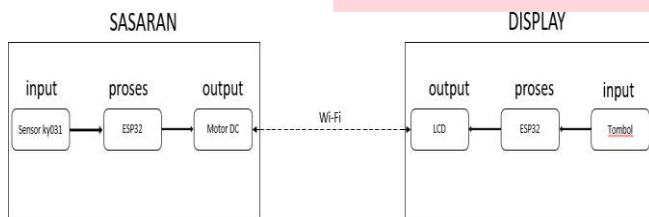
Tabel 2 Kebutuhan Non Fungsional

No	Kebutuhan Non Fungsional
1	2 buah NodeMCU ESP32, Untuk mengolah data yang ditangkap sensor dan saling bertukar data antara sasaran dan display
2	Sensor KY-031, Untuk menentukan titik tembakan yang terkena
3	I2C serial interface module, Untuk mengurangi penggunaan pin LCD pada mikrokontroler
4	LCD 20x4, Untuk menampilkan data yang telah diproses sebelumnya

5	Driver motor L298N, Untuk mengatur kecepatan dan arahputaran motor DC
6	Motor DC, Untuk menggerakkan target tembakan

C. Blok Diagram

Sebelum proses pembuatan alat, maka sebuah perancangan sistem diperlukan dengan membuat blok diagram sistem yang akan dibangun. Dengan blok diagram sistem ini dapat dilihat bagaimana sistem akan berjalan. Sehingga pada saat pembuatan sistem akan lebih mudah dikerjakan dengan mengacu pada blok diagram yang telah dibuat seperti pada gambar 2.

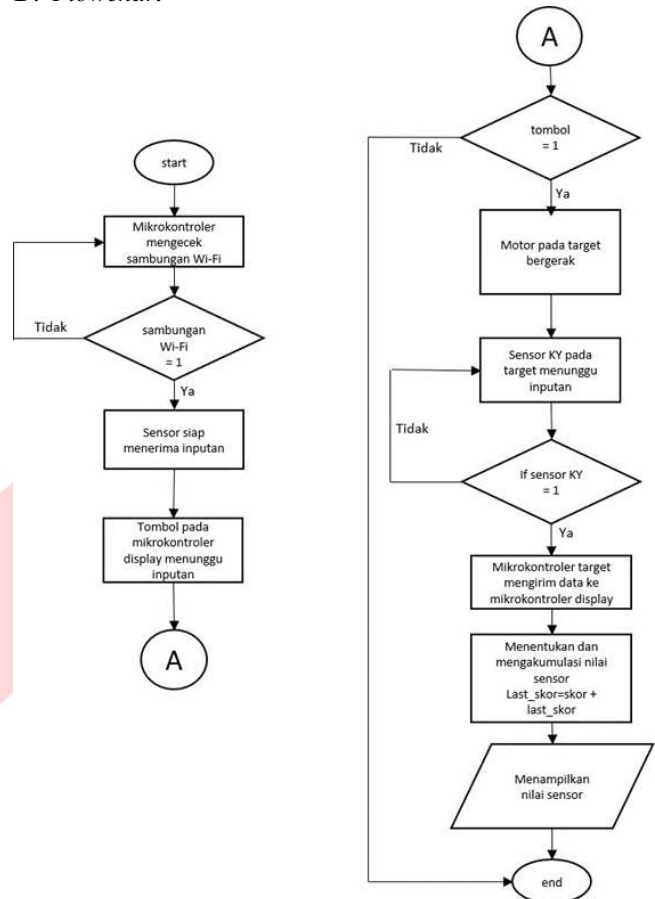


Gambar 2 Blok Diagram

Pada gambar 2 telah dijelaskan mengenai rancangan sistem yang akan dibuat. Jadi sistem Sistem penanda dan penilaian sasaran tembak menggunakan IoT berbasis sensor ky-031 ini akan menggunakan 2 mikrokontroler. Yang pertama terdapat pada bagian sasaran untuk membaca hasil sensor ky-031 jika sensor terkena ketukan/peluru. Setelah membaca hasil sensor lalu data akan dikirimkan melalui module wifi kepada mikrokontroler kedua yang terdapat pada display untuk menghitung perolehan nilai berdasarkan sensor yang terkena ketukan/peluru lalu menampilkannya pada LCD. Selain itu juga dapat mengontrol untuk menggerakkan motor atau tidak melalui Display. Jika tombol pada display ditekan maka motor pada sasaran akan bergerak.

Sistem penanda dan penilaian sasaran tembak menggunakan IoT berbasis sensor ky- 031 ini memiliki kelebihan yaitu dapat melakukan otomatisasi pendeteksian sasaran tembak dengan menggunakan senapan berpeluru BB. Yang biasanya sulit jika otomatisasi menggunakan hasil citra bekas tembakan pada sasaran tembak kertas. Sebelum proses pembuatan alat, maka diperlukan sebuah perancangan sistem dengan membuat blok diagram sistem yang akan di bangun. Sehingga pada saat proses pembuatan alat akan lebih mudah di kerjakan dengan mengacu pada blok diagram yang telah dibuat seperti pada gambar 8 dibawah ini.

D. Flowchart



Gambar 3 Flowchart

Pada gambar 3 telah digambarkan bagaimana sistem akan berjalan. Sistem akan dimulai dengan pengecekan sambungan wifi antara esp pada sasaran dan esp pada display. Jika kedua sambungan sudah tersambungan maka sistem akan dapat berjalan dan semua sensor siap menerima inputan. Apabila tombol penggerak motor yang terdapat pada display ditekan maka motor pada sasaran akan bergerak untuk menggerakkan sasaran ke kanan dan ke kiri, dan jika tombol pada display tidak di tekan maka motor tidak akan bergerak. Setelah itu sensor ky-031 atau sensor ketuk akan menunggu inputan untuk mendeteksi titik mana yang terkena tembakan. Lalu apabila ada titik sensor yang terkena inputan maka data akan dikirim melalui wifi kepada display untuk menentukan titik mana yang terkena dan menampilkan skornya serta mengakumulasi nilainya.

E. Kebutuhan Sistem

Kebutuhan Perangkat Keras.

Tabel 3 Kebutuhan Perangkat Keras

No	Perangkat Keras	Fungsi	Spesifikasi
----	-----------------	--------	-------------

1	ESP32	Mikrokontroler yang berfungsi untuk memproses semua data yang masuk dan pengendali komponen lainnya	- Tegangan Operasi 3.3V 11b/g/n WIFI transceiver - Jumlah digital pin 25 6 pin ADC 3 UART interface 3 SPI interface 2 I2C interface 16 pin PWM 2 pin DAC
2	Ky-031	Pendeteksi titik tembakan	- Daya operasi 3.3V sampai 5V - Tipe keluaran digital
3	LCD 2004	Menampilkan nilai	- Tegangan 5V DC - Menampilkan 20 karakter, 4 baris - Antarmuka : parallel - Tipe LCD : karakter
4	Driver Motor	Sebagai penggerak dan pengatur arah putaran motor DC	- Maksimum arus 2A - Tegangan 12V - IC L298N
5	Motor DC	Penggerak target sasaran	- Tegangan 3V - 6V - RPM 100 - 200

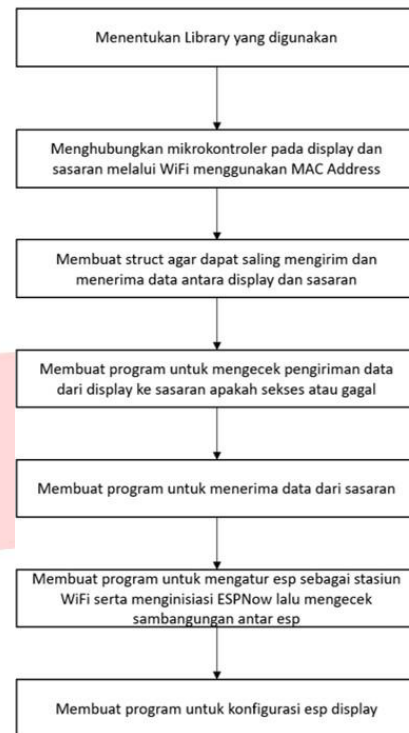
Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Fungsi
1	Arduino IDE	Menuliskan kode program dan mengupload ke board untuk menjalankan hardware

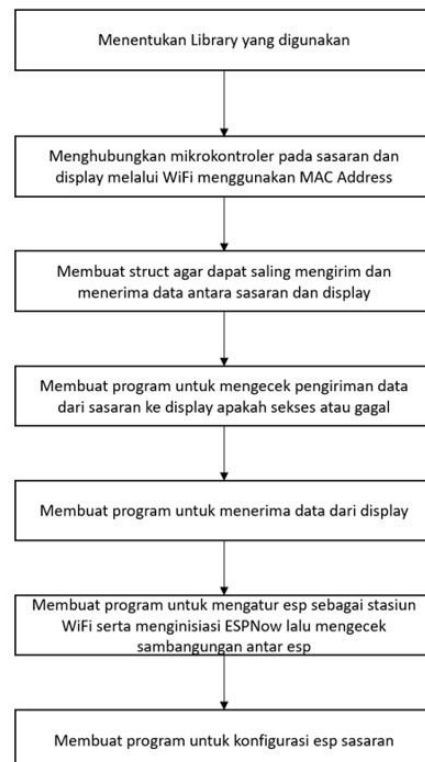
IV. IMPLEMENTASI DAN PENGUJIAN

A. Bagan Implementasi Sistem

DISPLAY



Gambar 4 Bagan Display
SASARAN



Gambar 5 Bagan Sasaran

B. Sistem Keseluruhan



Gambar 6 Sistem Keseluruhan

Berikut adalah rangkaian perangkat keras yang sudah saling terhubung satu sama lain untuk sistem saat ini. Terdapat 2 rangkaian yaitu rangkaian manekin untuk sasaran tembak yang terpasang sensor KY-031 dimana sensor ini cocok digunakan dalam penelitian ini karena outputnya yang berupa nilai 0 atau 1 dan rangkaian display skor yang terpasang LCD serta tombol sebagai penampil nilai dan kontrol manekin.

C. Pembuatan Sistem Penanda/Sasaran Tembakan

Pembuatan program pada penanda/sasaran tembak menggunakan aplikasi Arduino IDE. Penjelasan program pada penanda/sasaran tembak sebagai berikut:

```
//-----Load libraries
#include <esp_now.h>
#include <WiFi.h>
#include <Robojax_L298N_DC_motor.h>
//-----
```

Gambar 7 Library yang digunakan

Gambar 7 merupakan library yang digunakan untuk menunjang jalannya program sistem penanda dan penilaian sasaran tembak menggunakan IoT berbasis sensor KY-031. Program tersebut dipanggil pertama kali saat program akan berjalan

Gambar 8. MAC Address ESP Display

```
uint8_t broadcastAddress[] = {0x84, 0x0D, 0x8E, 0x2C, 0xC7, 0xE4};
```

Gambar 8 merupakan program untuk menyambungkan esp menggunakan MAC Address agar dapat saling mengirim data.

```
typedef struct struct_message {
    int led;
} struct_message_send;

struct_message send_Data; // Create a struct_message to send data.

struct_message receive_Data; // Create a struct_message to receive data.
//-----
```

Gambar 9. Struct Pengiriman Dan Penerimaan Data

Gambar 9 merupakan contoh struktur untuk mengirim data kepada penerima. Dan struktur antara pengirim dan penerima harus sama.

```
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
    Serial.print("\r\nLast Packet Send Status:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
    if (status == 0) {
        success = "Delivery Success : ";
    }
    else {
        success = "Delivery Fail : ";
    }
    Serial.println("">>>>>");
}
```

Gambar 1 Program Pengiriman Data

Gambar diatas merupakan program untuk mengecek pengiriman data dari esp target ke esp display apakah sukses atau gagal.

```
void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {
    memcpy(&receive_Data, incomingData, sizeof(receive_Data));
    Serial.println();
    Serial.println("<<<<< Receive Data:");
    Serial.print("Bytes received: ");
    Serial.println(len);
    LED_State_Receive = receive_Data.led;
    Serial.print("Receive Data: ");
    Serial.println(LED_State_Receive);
    Serial.println("<<<<<");

    if (LED_State_Receive == 1) {
        nilai = nilai + 1;
        Serial.println(nilai);

        if (nilai%2==1) {
            Serial.println("NYALA");

            // robot.demo(1);
            robot.rotate(motor1, 60, CW); //run motor1 at 30% speed in CW direction
            robot.rotate(motor2, 60, CW); //run motor1 at 30% speed in CW direction
            delay(3000);

            robot.rotate(motor1, 60, CCM); //run motor1 at 30% speed in CCM direction
            robot.rotate(motor2, 60, CCM); //run motor1 at 30% speed in CCM direction
            delay(3000);

            robot.rotate(motor1, 60, CW); //run motor1 at 30% speed in CW direction
            robot.rotate(motor2, 60, CW); //run motor1 at 30% speed in CW direction
            delay(3000);

            robot.rotate(motor1, 60, CCM); //run motor1 at 30% speed in CCM direction
            robot.rotate(motor2, 60, CCM); //run motor1 at 30% speed in CCM direction
            delay(3000);

            robot.rotate(motor1, 60, CW); //run motor1 at 30% speed in CW direction
            robot.rotate(motor2, 60, CW); //run motor1 at 30% speed in CW direction
            delay(3000);

            robot.rotate(motor1, 60, CCM); //run motor1 at 30% speed in CCM direction
            robot.rotate(motor2, 60, CCM); //run motor1 at 30% speed in CCM direction
            delay(3000);

            robot.rotate(motor1, 60, CW); //run motor1 at 30% speed in CW direction
            robot.rotate(motor2, 60, CW); //run motor1 at 30% speed in CW direction
            delay(3000);

            robot.rotate(motor1, 60, CCM); //run motor1 at 30% speed in CCM direction
            robot.rotate(motor2, 60, CCM); //run motor1 at 30% speed in CCM direction
            delay(3000);

            robot.brake(1);
            robot.brake(2);
        }
        //-----

        else if (nilai%2==0) {
            Serial.println("MATII");
            delay(2000);
            robot.brake(1);
            robot.brake(2);
        }
    }

    else if (LED_State_Receive == 3) {
        ESP.restart();
    }
}
```

Gambar 11 Program Penerimaan data

Gambar 11 merupakan program untuk menerima data dari esp display ke esp target. Dimana jika tombol di display di tekan maka motor pada target akan bergerak.

```

WiFi.mode(WIFI_STA); //=> Set device as a Wi-Fi Station

//-----Init ESP-NOW
if (esp_now_init() != ESP_OK) {
  Serial.println("Error initializing ESP-NOW");
  return;
}

//-----Once ESPNow is successfully Init, we will register for Send CB to
// get the status of Transmitted packet
esp_now_register_send_cb(OnDataSent);

//-----Register peer
esp_now_peer_info_t peerInfo;
memcpy(&peerInfo.peer_addr, broadcastAddress, 6);
peerInfo.channel = 0;
peerInfo.encrypt = false;

//-----Add peer
if (esp_now_add_peer(&peerInfo) != ESP_OK) {
  Serial.println("Failed to add peer");
  return;
}

```

Gambar 12 Program ESP Sebagai Wifi

Gambar 12 merupakan program untuk mengatur esp sebagai stasiun wifi serta menginisiasi ESPNow serta mengecek sambungan antar esp apakah berhasil terhubung atau tidak.

```

void loop() {
  BTN_State1 = digitalRead(BTN_Pin1); //=> Reads and holds button states.
  BTN_State2 = digitalRead(BTN_Pin2); //=> Reads and holds button states.
  BTN_State3 = digitalRead(BTN_Pin3); //=> Reads and holds button states.
  BTN_State4 = digitalRead(BTN_Pin4); //=> Reads and holds button states.
  BTN_State5 = digitalRead(BTN_Pin5); //=> Reads and holds button states.
  BTN_State6 = digitalRead(BTN_Pin6); //=> Reads and holds button states.

  //-----When the button is pressed it will send data to control the LED on the ESP8266 Target.
  if (BTN_State1 == LOW) {
    LED_State_Send = 1;
    send_Data_LED = LED_State_Send;

    Serial.println();
    Serial.print(">>>>> ");
    Serial.println("Send data");

    //-----Send message via ESP-NOW
    esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &send_Data_LED, sizeof(send_Data_LED));

    if (result == ESP_OK) {
      Serial.println("Sent with success");
    }
    else {
      Serial.println("Error sending the data");
    }
  }

  //-----Wait for the button to be released. Release the button first to send the next data.
  while (BTN_State1 == LOW) {
    BTN_State1 = digitalRead(BTN_Pin1);
    delay(100);
  }

  //-----
  else if (BTN_State2 == LOW) {
    LED_State_Send = 1;
    send_Data_LED = LED_State_Send;

    Serial.println();
    Serial.print(">>>>> ");
    Serial.println("Send data");
  }
}

```

Gambar 13 Program Konfigurasi ESP Sasaran

Gambar 13 adalah program untuk konfigurasi sensor ky-031 dan membuat kondisi jika sensor menerima inputan maka akan mengirim data kepada esp display.

D. Pembuatan Sistem Display

Pembuatan program pada display penilaian menggunakan aplikasi Arduino IDE. Penjelasan program:

Gambar 14 Library yang digunakan

```

#include <esp_now.h>
#include <WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

```

Gambar 14 merupakan library yang digunakan untuk menunjang jalannya program sistem penanda dan penilaian sasaran tembak menggunakan IoT berbasis sensor KY-031. Program tersebut dipanggil pertama kali saat program akan berjalan.

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

Gambar 15 Alamat LCD

Gambar 15 merupakan program untuk mengatur alamat LCD ke 0x27 untuk 20 karakter dan 4 baris tampilan

```
uint8_t broadcastAddress[] = {0x58, 0xBF, 0x25, 0x16, 0x6E, 0x00};
```

Gambar 16 MAC Address Sasaran

Gambar 16 merupakan program untuk menyambungkan esp menggunakan MAC Address agar dapat saling mengirim data.

```

typedef struct struct_message {
  int led;
} struct_message_send;

struct_message send_Data; // Create a struct_message to send data.

struct_message receive_Data; // Create a struct_message to receive data.
//-----

```

Gambar 17 Struct Pengiriman dan Penerimaan Data

Gambar 17 merupakan contoh struktur untuk mengirim data kepada penerima. Dan struktur antara pengirim dan penerima harus sama.

```

void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
  Serial.print("\r\nLast Packet Send Status:\t");
  Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
  if (status == 0) {
    success = "Delivery Success :)";
  }
  else {
    success = "Delivery Fail :(";
  }
  Serial.println(">>>>>");
}

```

Gambar 18 Program Pengecekan Pengiriman Data

Gambar 18 merupakan program untuk mengecek pengiriman data dari esp display ke esp target apakah sukses atau gagal.

```

void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {
  memcpy(&receive_Data, incomingData, sizeof(receive_Data));
  lcd.begin();
  Serial.println();
  Serial.println("====> Receive Data:");
  Serial.print("Bytes received: ");
  Serial.println(len);
  LED_State_Receive = receive_Data.led;

  Serial.print("Nilai Data: ");
  Serial.println(LED_State_Receive);
  Serial.println("====>");

  if(LED_State_Receive == 1){
    delay(500);
    lcd.clear();
    lcd.backlight();
    Total_Nilai = Total_Nilai + 2;

    Serial.println(" Sensor 1 ");
    Serial.print(" Total Nilai = ");
    Serial.println(Total_Nilai);

    lcd.setCursor(0,0);
    lcd.print("TERKENA TANGAN KIRI");
    lcd.setCursor(0,1);
    lcd.print("POINT +2 ");
    delay(2000);
    lcd.clear();

    lcd.setCursor(0,3);
    lcd.print("POINT TOTAL = ");
    lcd.setCursor(15,3);
    lcd.print(Total_Nilai);
  }
  else if(LED_State_Receive == 2){
    delay(500);
    lcd.clear();
    Total_Nilai = Total_Nilai + 4;

    Serial.println(" Sensor 2 ");
    Serial.print(" Total Nilai = ");
    Serial.println(Total_Nilai);

    lcd.setCursor(0,0);
    lcd.print("TERKENA PERUT");
    lcd.setCursor(0,1);
    lcd.print("POINT +4 ");
    delay(2000);
    lcd.clear();

    lcd.setCursor(0,3);
    lcd.print("POINT TOTAL = ");
    lcd.setCursor(15,3);
    lcd.print(Total_Nilai);
  }
}

```

Gambar 19 Program Penerimaan Data

Gambar 19 merupakan program untuk menerima data dari esp target ke esp display. Dimana jika sensor pada target menerima inputan atau ketukan maka display akan memproses nilai dan menampilkannya pada LCD.

```

WiFi.mode(WIFI_STA); //---> Set device as a Wi-Fi Station

//-----Init ESP-NOW
if (esp_now_init() != ESP_OK) {
  Serial.println("Error initializing ESP-NOW");
  return;
}

//-----Once ESPNow is successfully Init, we will register for Send CB to
// get the status of Transmitted packet
esp_now_register_send_cb(OnDataSent);

//-----Register peer
esp_now_peer_info_t peerInfo;
memcpy(&peerInfo.peer_addr, broadcastAddress, 6);
peerInfo.channel = 0;
peerInfo.encrypt = false;

//-----Add peer
if (esp_now_add_peer(&peerInfo) != ESP_OK) {
  Serial.println("Failed to add peer");
  return;
}

```

Gambar 20 Program ESP Sebagai Stasiun WiFi

Gambar 20 merupakan program untuk mengatur esp sebagai stasiun wifi serta menginisiasi ESPNow serta

mengecek sambungan antar esp apakah berhasil terhubung atau tidak.

```

void loop() {
  BTU_State1 = digitalRead(BTU_Pin1); //---> Reads and holds button status.
  BTU_State2 = digitalRead(BTU_Pin2); //---> Reads and holds button status.

  //-----When the button is pressed it will send data to control the LED on the ESP32 Target.
  if(BTU_State1 == HIGH) {
    LED_State_Send = 1;
    send_Data.led = LED_State_Send;

    Serial.println();
    Serial.print("====> ");
    Serial.println("Send Data");

    //-----Send message via ESP-NOW
    esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &send_Data, sizeof(send_Data));

    if (result == ESP_OK) {
      Serial.println("Sent with success");
    } else {
      Serial.println("Error sending the data");
    }
  }

  //-----Wait for the button to be released. Release the button first to send the next data.
  while(BTU_State1 == HIGH) {
    BTU_State1 = digitalRead(BTU_Pin1);
    delay(100);
  }

  //-----
  while if(BTU_State2 == HIGH){
    ESP.reset();

    while(BTU_State2 == HIGH) {
      BTU_State2 = digitalRead(BTU_Pin2);
      delay(10);
    }
  }
}

```

Gambar 21 Program Konfigurasi ESP Display

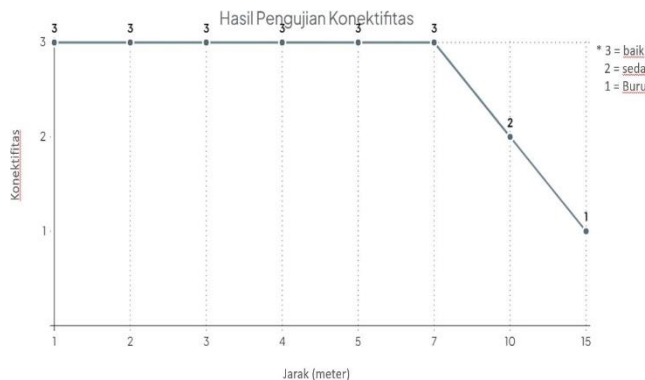
Gambar 21 merupakan program konfigurasi tombol dan membuat kondisi jika tombol 1 ditekan maka akan mengirim data kepada esp target dan jika tombol 2 ditekan maka esp display akan mereset data.

E. Pengujian Konektifitas Pembacaan Sensor

Pengujian ini dilakukann untuk mengetahui tingkat akurasi pembacaan sensor berbanding dengan jarak. Pengujian di lakukan dengan jarak yang telah ditentukan, pengujian di laksanakan di sekitar kawasan danau Telkom University, beberapa hal yang akan diuji adalah status konektifitas ESP pada target terhadap ESP pada display.

Tabel 5 Hasil Pengujian Konektifitas

No.	Jarak (m)	Terbaca/Pengujian	Konektivitas
1	1	9/10	Baik
2	2	10/10	Baik
3	3	9/10	Baik
4	4	8/10	Baik
5	5	8/10	Baik
6	7	10/10	Baik
7	10	5/10	Sedang
8	15	0/10	Buruk



Gambar 22 Grafik Hasil Pengujian Konektivitas

Dapat dilihat dari gambar bahwa konektivitas akan semakin menurun jika jarak semakin jauh. Dimana konektivitas mulai turun dari jarak 10m dan pada jarak lebih dari 10m hasil pembacaan sensor tidak dapat terbaca lagi.

F. Pengujian Jarak Antar Sensor

Pengujian ini dilakukan untuk mengetahui seberapa besar jarak minimal antar sensor agar dapat bekerja dengan optimal. Pengujian ini dilakukan dengan cara menempelkan 2 sensor pada sebuah alas dengan jarak tertentu, lalu akan diketuk bagian tengahnya. Dimana jarak akan semakin bertambah untuk mengetahui berapa jarak minimal agar sensor tidak terbaca keduanya.

Tabel 6 Hasil Pengujian Antar Sensor

Jarak	Sensor 1	Sensor2
1 cm	1	1
2 cm	1	1
3 cm	1	1
4 cm	1	1
5 cm	0	0

Berdasarkan pada gambar tabel 6 , dapat disimpulkan bahwa jarak minimal agar sensor dapat bekerja optimal atau tidak terbaca keduanya jika di titik tengahnya diketuk adalah 5 cm.

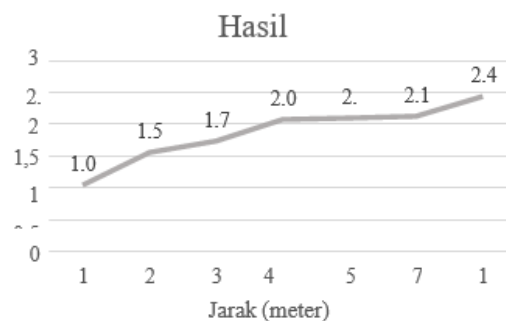
G. Pengujian Delay

Pengujian ini dilakukan untuk mengetahui seberapa lama waktu yang dibutuhkan ketika sensor menerima input an pada jarak tertentu kemudian mengirimkannya ke display. Pengujian di lakukan dengan jarak yang telah di tentukan, pengujian di laksanakan di sekitar kawasan danau Telkom University, hal yang akan di uji adalah delay waktu pengiriman data dari esp target kepada esp display berbanding dengan jarak.

Tabel 7 Hasil Pengujian Delay

	Jarak 1m	Jarak 2m	Jarak 3m	Jarak 4m	Jarak 5m	Jarak 7m	Jarak 10m
Sensor 1	1,06 d	1,5 d	1,8 d	2,01 d	2,1 d	2,19 d	2,31 d

Sensor 2	1,05 d	1,7 d	1,75 d	2,1 d	2,13 d	2,03 d	2,42 d
Sensor 3	1,07 d	1,54 d	1,67 d	2,05 d	2,04 d	2,14 d	2,56 d
Sensor 4	1,04 d	1,63 d	1,84 d	2,13 d	2,03 d	2,07 d	2,43 d
Sensor 5	1,05 d	1,3 d	1,65 d	2,07 d	2,17 d	2,17 d	2,53 d
Sensor 6	1,03 d	1,67 d	1,7 d	2,11 d	2,13 d	2,15 d	2,36 d
rata-rata	1,05 d	1,56 d	1,74 d	2,08 d	2,10 d	2,13 d	2,44 d



Gambar 23 Grafik Hasil Pengujian Delay

Berdasarkan gambar dapat dianalisa bahwa delay pengiriman data akan semakin lama jika jarak semakin jauh.

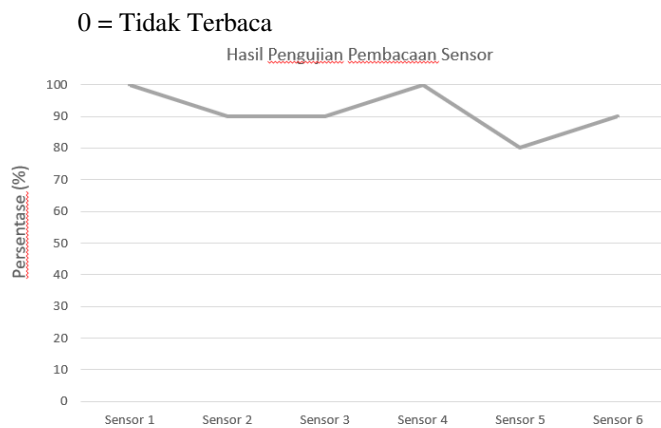
H. Pengujian Akurasi Pembaca Sensor

Pengujian akurasi pembacaan sensor bertujuan untuk mengetahui seberapa tepat pembacaan sensor jika sensor menerima inputan/ketukan. Pengujian dilakukan dengan cara mengetuk atau memberi inputan pada masing- masing sensor sebanyak 10 kali.

Tabel 7 Hasil Pengujian Akurasi Pembacaan Sensor

Percobaan	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	1	1	1	1	1	0
5	1	1	1	1	1	1
6	1	0	1	1	0	1
7	1	1	1	1	0	1
8	1	1	0	1	1	1
9	1	1	1	1	1	1
10	1	1	1	1	1	1
Persentase	100%	90%	90%	100%	80%	90%
nilai rata-rata	92%					

Ket : 1 = Terbaca



Gambar 23 Grafik Hasil Pengujian Delay

Hasil analisa dari pengujian akurasi pembacaan sensor adalah tingkat akurasi tinggi karena memiliki rata-rata akurasi pembacaan 92%.

V. KESIMPULAN

Kesimpulan yang diambil dari hasil analisis dan pengujian dari proyek akhir ini adalah sebagai berikut :

Sistem dapat mendeteksi hasil sasaran tembak menggunakan sensor ky-031 berbasis IoT. Sistem dapat memunculkan nilai hasil tembakan pada display skor lalu mengakumulasi nilai berbasis IoT.

REFERENCES

- [1] R. Pramedistian, "Rancang Bangun Alat Latihan Menembak Berbasis Internet Of Things (Studi Kasus : Perbakin Subang)," Feb. 2019, Accessed: Dec. 07, 2021. [Online]. Available: <http://elibrary.unikom.ac.id>.
- [2] H. Ramadhan, R. Maulana, and M. H. H. Ichsan, "Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Logika Fuzzy | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer," 2018. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2375> (accessed Dec. 07, 2021).
- [3] B. N. Kusuma and A. S. Wibowo, "Rancangan Sistem Pendeteksian Perhitungan Skor Tembakan Pada Latihan Menembak Berbasis OpenCV," Akad. Angkatan Udar. J. Def. Sci. Technol. AAU-JDST, vol. 8, no. 2, pp. 133–145, 2019, Accessed: Dec. 07, 2021. [Online]. Available: <https://aau.e-journal.id/jdst/article/view/94>.
- [4] P. D. Widayaka, "Sistem penghitung skor tembak otomatis berbasis pemrosesan citra," 2019.
- [5] S. D. Yolanda, R. Maulana, and M. H. H. Ichsan, "Scoring System Otomatis Pada Lomba Menembak Dengan Target Silhouette Hewan Menggunakan Metode K-Nearest Neighbor (K-NN)." <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4035/1788> (accessed Dec. 07, 2021).
- [6] I. F. Alaidi, Periyadi, and G. A. Mutiara, "Sistem Monitoring Pada Rompi Pintar Berbasis Internet Of Things (IoT)," e-Proceeding Appl. Sci., vol. 4, no. 3, pp. 2165–2170, 2018.
- [7] H. Harmaen, "Rancang Bangun Sistem Penghitung Skor Otomatis Olahraga Menembak Menggunakan Algoritma Template Matching," INKOM J., vol. 6, no. 2, pp. 69–77, Feb. 2013, Accessed: Dec. 07, 2021. [Online]. Available: <https://jurnal.informatika.lipi.go.id/index.php/inkom/article/view/187>.
- [8] M. F. Adlan, R. Parlindungan, D. Rahmawati, and K. Kunci, "Sistem Monitoring Kebugaran Kardiorespirasi Pada Usia 20 – 29 Tahun Berbasis Internet Of Things," pp. 13–14, 2022.
- [9] R. Nandariyanto, "Rancang Bangun Palang Pintu Kereta Api Otomatis Berbasis Arduino."
- [10] M. Ruswandi Djalal, D. Ajiatmo, A. Imran, and I. Robandi, "Desain Optimal Kontroler Pid Motor Dc Menggunakan Cuckoo Search Algorithm," Sentia, vol. 7, no. 1, pp. 121–126, 2015.