

MANUAL BOOK
“Controlling dan Monitoring Customer menggunakan Sensor Jarak
Melalui Platform IOT”

Projek Akhir Praktikum Internet of Things



Disusun Oleh : Kelompok 1 / IOT B

NAMA	NIM
Muhammad Gusti Keyandi Erfan	2009106003
Muhammad Akmal Rifad	2009106010
Olivia Oktavi Utami	2009106102
Nurhidayah Anugrah	2009106121

Asisten :

Kandika Prima Putra	Delfan Rynaldo Laden	M. Rizky Amanullah	Muhammad Al Fahri
1915016015	1915016069	1915016073	1915026013

INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
2023

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	ii
A. LATAR BELAKANG SISTEM.....	1
B. FUNGSI SISTEM.....	1
C. KONSEP YANG DIGUNAKAN.....	1
D. BOARD SCHEMATIC	1
E. TAHAPAN PERANCANGAN SISTEM.....	2
1. Merangkai Komponen Elektronik.....	2
2. Persiapan Platform IoT	3
3. Perancangan Program pada Arduino IDE.....	7
F. PENGUJIAN SISTEM.....	15

DAFTAR GAMBAR

gambar.1 1 Board Schematic	1
gambar.1 2 Rangkaian Publisher	2
gambar.1 3 Rangkaian Subscriber	3
gambar.1 4 Aplikasi IOT MQTT Panel.....	3
gambar.1 5 Tampilan Awal aplikasi MQTT Panel.....	4
gambar.1 6 Tampilan Setup Connection	4
gambar.1 7 button menambah panel	5
gambar.1 8 Pilihan Panel	5
gambar.1 9 Konfigurasi panel controlling dan monitoring	6
gambar.1 10 Tampilan Hasil dari konfigurasi panel.....	6
gambar.1 11 Install Library MQTT Subscriber	7
gambar.1 12 Setup MQTT dan PIN.....	8
gambar.1 13 Setup WIFI.....	8
gambar.1 14 Void Reconnect	9
gambar.1 15 Void Setuo	9
gambar.1 16 Void Loop	9
gambar.1 17 Setup MQTT	10
gambar.1 18 Setup pin.....	10
gambar.1 19 Void Setup Wifi.....	11
gambar.1 20 Void Callback.....	12
gambar.1 21 Void Reconnect	12
gambar.1 22 Void Setup	13
gambar.1 23 Void Loop	14
gambar.1 24 Hasil Monitoring pada Platform IoT	15
gambar.1 25 Seriam Monitor Subscribe	15
gambar.1 26 Serial Monitor Subcribe.....	16
gambar.1 27 Serial Monitor Publisher.....	16
gambar.1 28 Serial Monitor Publisher.....	16
gambar.1 29 Sensor dImatikan.....	17

A. LATAR BELAKANG SISTEM

Perangkat monitoring sering dipasang pada berbagai alat atau dipasang pada pintu atau tempat masuk. Pada proyek ini, alat solder termasuk salah satu perangkat yang dipasang sistem monitor padanya. Informasi suhu perlu diambil pada alat solder sebagai indikator jika solder masih dingin, mencapai suhu siap pakai, atau terlalu panas. Sistem ini dapat dimonitor melalui platform Internet of Things (IoT) agar pengguna tidak melulu harus standby di depan alat solder.

B. FUNGSI SISTEM

1. Monitor tamu yang datang.
2. Memberikan info melalui platform IoT.
3. Menyalakan buzzer dan LED jika jarak yang terdeteksi dibawah 50 cm.

C. KONSEP YANG DIGUNAKAN

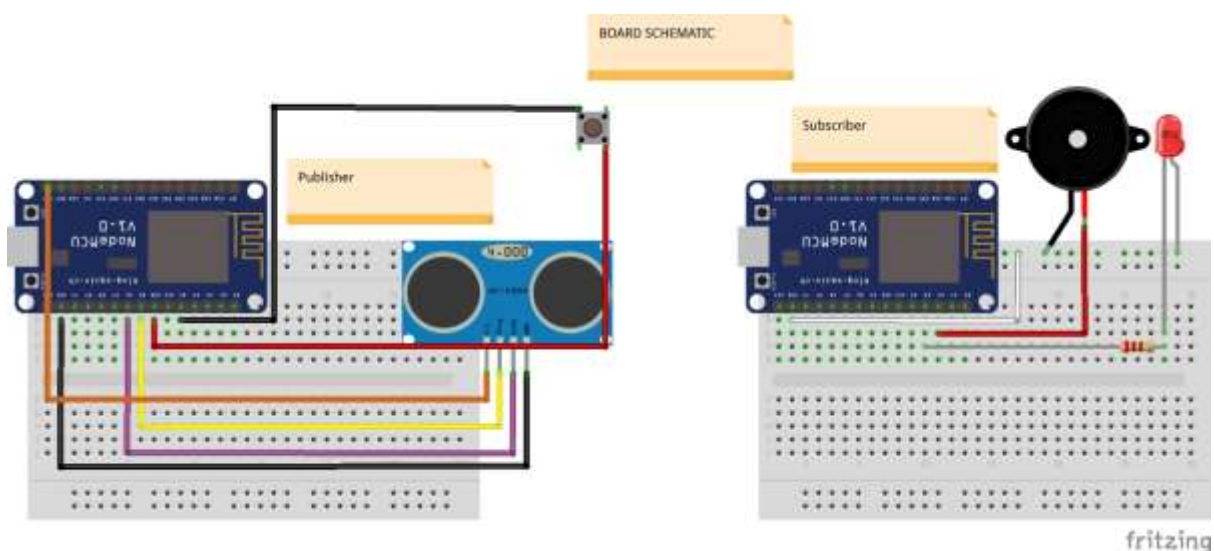
1. MQTT

MQTT digunakan untuk komunikasi antar node. Kedua node terkoneksi pada server **broker.hivemq.com** port **1883** dengan topic **iot_unmul/iot_b_1**. Edge node mengirim data pada topic, sedangkan master node akan menerima data dari topic yang di-subscribe untuk mengolah datanya.

2. Platform IOT

Platform IoT MQTT Panel digunakan karena kemudahannya dalam mengaksesnya baik di platform web maupun mobile, serta dapat digunakan secara gratis.

D. BOARD SCHEMATIC



gambar.1 1 Board Schematic

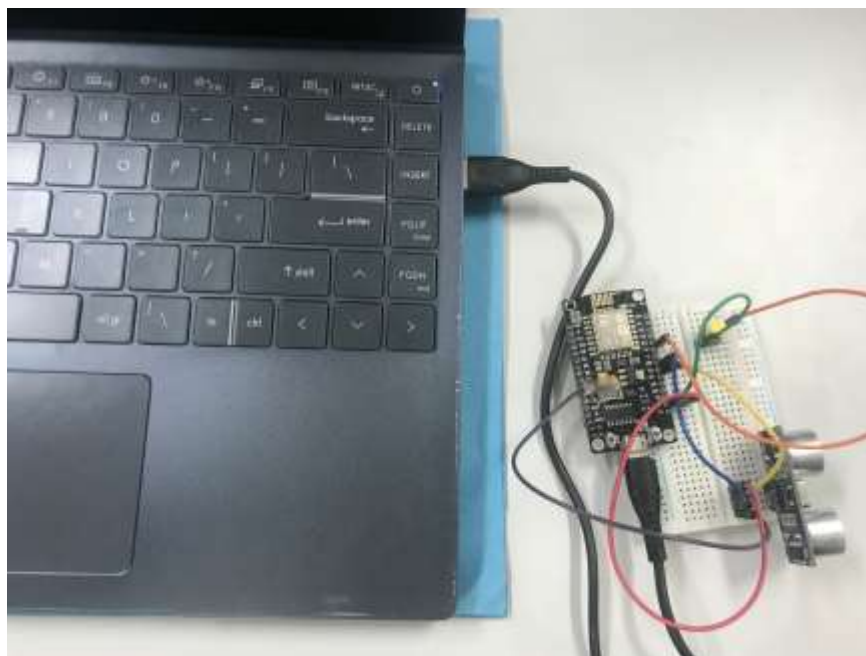
Komponen yang digunakan antara lain:

1. NodeMCU x 2
2. Kabel Jumper Male - Male x 9
3. LED x 1
4. Resistor x 1
5. Sensor jarak HC SR04 x 1
6. Button x 1
7. Breadboard x 2
8. Buzzer x 1
9. Micro USB x 2

E. TAHAPAN PERANCANGAN SISTEM

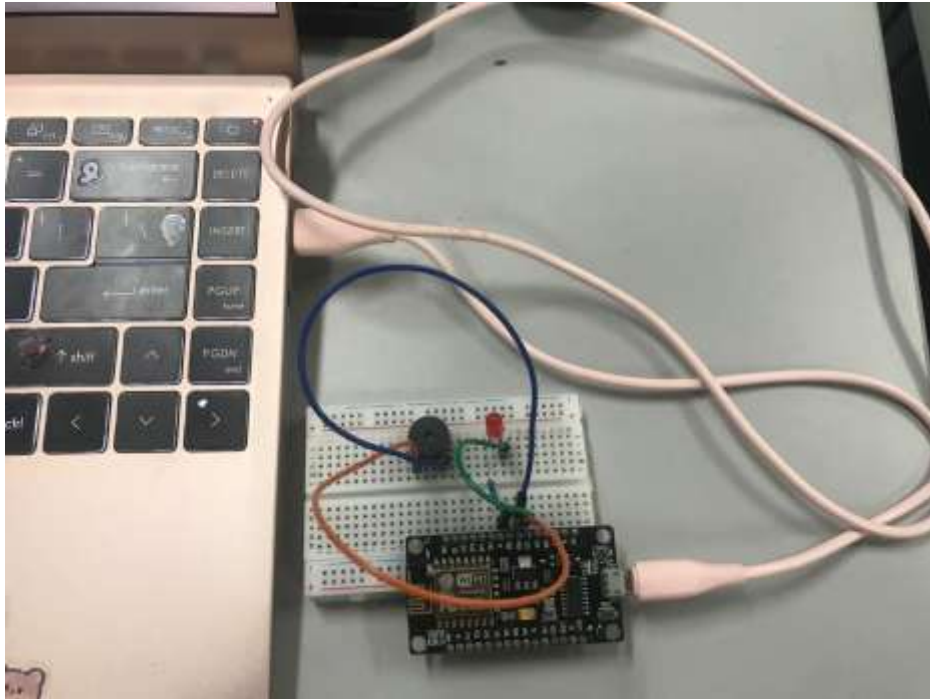
Berikut adalah cara merancang sistem monitoring dan controlling jarak berbasis IOT. Proses perancangan sistem terdiri dari beberapa tahap yang meliputi merangkai komponen elektronik, persiapan platform Internet of Things (IoT), perancangan program menggunakan Arduino, dan pengujian sistem.

1. Merangkai Komponen Elektronik



gambar.1 2 Rangkaian Publisher

Rangkaian komponen Publisher seperti pada *board schematic* sebelumnya. Dilengkapi dengan sensor jarak HC-SR04 yang menggunakan 5V. Untuk pinnya VCC terhubung pada 5 Volt, Trig terhubung pada D5, Echo terhubung pada D6, GND terhubung pada Ground. Setelah itu ada Button yang terhubung pada D4 dan Ground.



gambar.1 3 Rangkaian Subscriber

Rangkaian komponen subscriber seperti pada *board schematic* sebelumnya. Dilengkapi dengan buzzer dan LED yang terhubung pada Ground. Untuk pinnya LED terhubung pada D5 sedangkan buzzer pada D6. Setiap node akan disuplai daya 5V dari kabel USB.

2. Persiapan Platform IoT



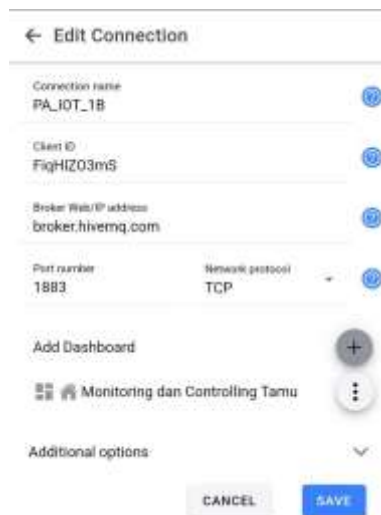
gambar.1 4 Aplikasi IOT MQTT Panel

Install Aplikasi IoT MQTT Panel di playstore. Tekan install dan tunggu beberapa saat. Aplikasi ini memungkinkan untuk mengatur dan memvisualisasikan proyek IoT, berdasarkan protokol MQTT. Dengan aplikasi ini dapat membuat project IOT dengan konfigurasi yang sederhana.



gambar.1 5 Tampilan Awal aplikasi MQTT Panel

Setelah selesai install, Yang pertama terbuka adalah tampilan seperti di atas. Untuk memonitoring dan controlling data menggunakan aplikasi ini maka klik “SETUP A CONNECTION”.



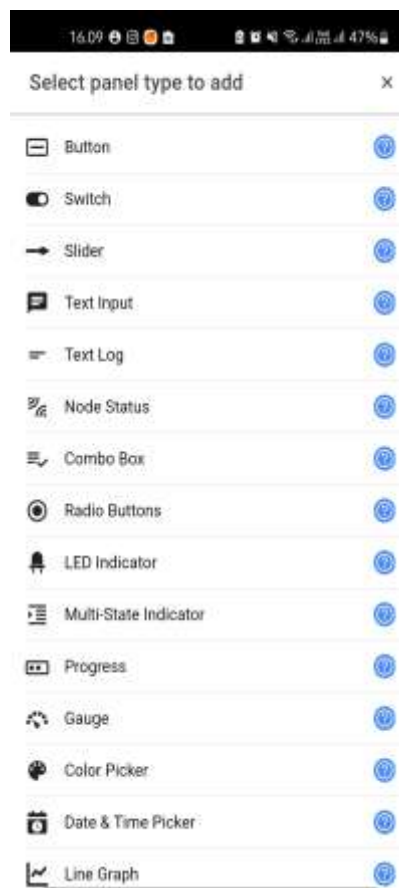
gambar.1 6 Tampilan Setup Connection

Setelah itu muncul tampilan seperti gambar diatas, isi sesuai project yang dikerjakan. Untuk broker pilihlah broker.hivemq.com dan port number 1883. Jika mempunyai pilihan broker yang lain, bisa diganti sesuai broker yang dipilih.



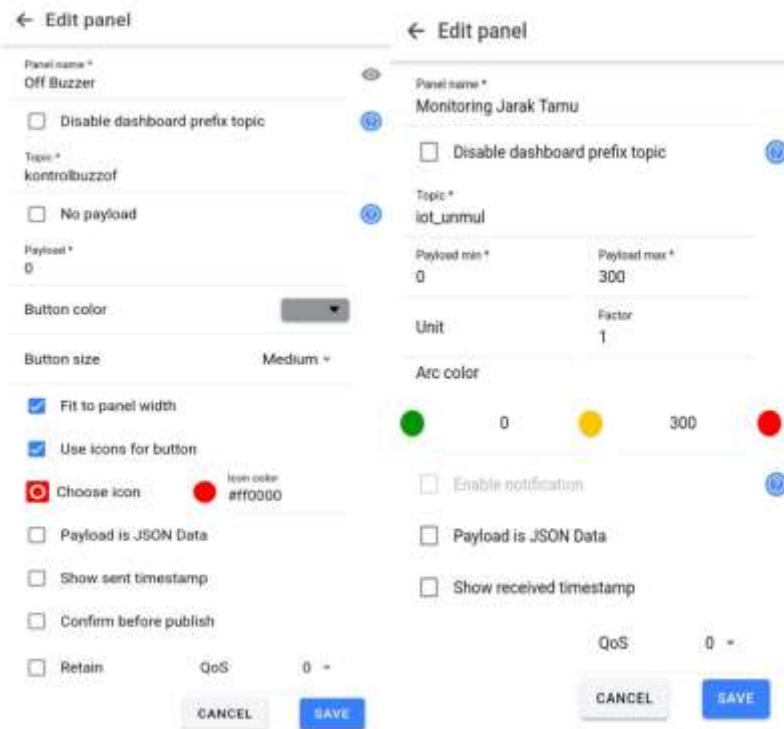
gambar.1 7 button menambah panel

Selanjutnya klik tombol di bagian bawah yang memiliki icon “+” untuk menambah pannel.



gambar.1 8 Pilihan Panel

Pilihlah panel sesuai project yang dikerjakan. Dalam project kami, controlling dan monitoring menggunakan panel button dan gauge. Lalu isi sesuai dengan project yang dikerjakan. berikut adalah konfigurasi panel yang dipakai dalam project ini.



gambar.1 9 Konfigurasi panel controlling dan monitoring



gambar.1 10 Tampilan Hasil dari konfigurasi panel

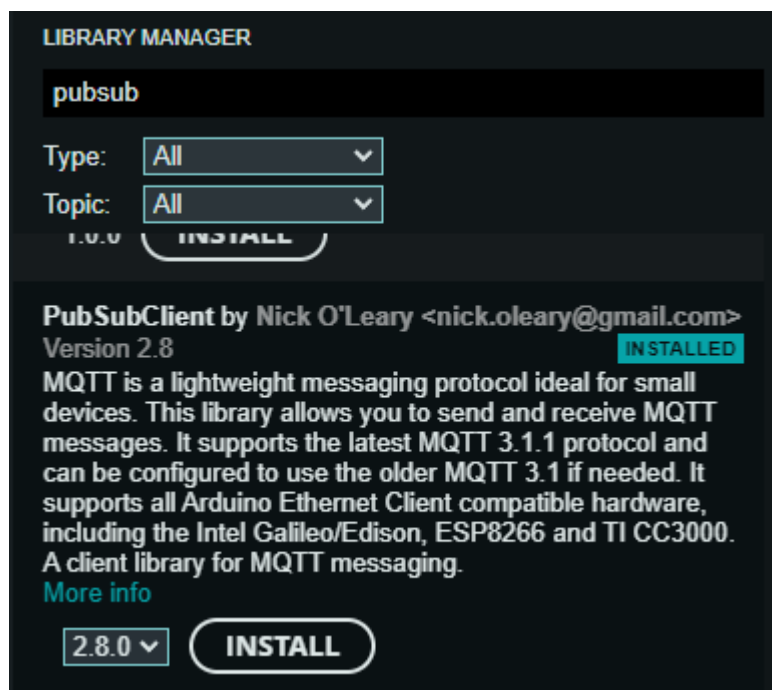
Gambar di atas adalah hasil dari konfigurasi 2 Panel. Untuk Monitoring panah akan bergerak sesuai dengan jarak sedangkan untuk tombol jika di klik, suara dari buzzer bisa mati.

3. Perancangan Program pada Arduino IDE

Source code dapat diakses pada link dibawah.

Master Node: https://github.com/nurhidayahanugrah/pa-praktikum-iot-unmul-b1/blob/main/papublish_iot1b.ino
Edge Node: https://github.com/nurhidayahanugrah/pa-praktikum-iot-unmul-b1/blob/main/pasubs_iot1b.ino

- **SUBSCRIBER**



gambar.1 11 Install Library MQTT Subscriber

Untuk dapat mengirim pesan menggunakan protokol MQTT, pastikan telah menginstal library PubSubClient yang dikembangkan oleh Nick O'Leary.

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.
const char* ssid = "Universitas Mulawarman";
const char* password = "";
const char* mqtt_server = "broker.hivemq.com"; // broker gratisan

// Buzzer & LED pin
#define BUZZER_PIN D6
#define LED_PIN D5

```

gambar.1 12 Setup MQTT dan PIN

Untuk menghubungkan ke internet ssidnya di atur sesuai dengan jaringan wifi yang digunakan seperti yang dicontoh di atas menggunakan Universitas Mulawarman sebagai ssidnya dan tidak menggunakan password agar mudah mengaksesnya. adapun server yang digunakan untuk broker yaitu mqtt_servernya adalah broker.hivemq.com untuk menghubungkan ke broker mqtt yang sudah di atur. Lalu ada inisiasi perangkat yang digunakan seperti buzzer_pin di letakkan di pin D6 pada Ensp 32 serta led_pin pada D5 pada Ensp 32nya.

```

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

```

gambar.1 13 Setup WIFI

Untuk menguhubungkan ssidnya ke wifi agar terkoneksi.

```

// Fungsi untuk menghubungkan ke broker
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    if (client.connect(clientId.c_str())) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.subscribe("iot_unmul");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

```

gambar.1 14 Void Reconnect

untuk mengetahui apakah ssidnya sudah terkoneksi atau belum.

```

void setup() {
  pinMode(BUZZER_PIN, OUTPUT); // Inisialisasi pin buzzer
  pinMode(LED_PIN, OUTPUT);    // Inisialisasi pin LED
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883); // setup awal ke server mqtt
  client.setCallback(callback);
}

```

gambar.1 15 Void Setuo

untuk mengetahui inisialisasi buzzer & LED. Selain itu, untuk mengetahui setup awal yang menghubungkan ke server mqtt.

```

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
}

```

gambar.1 16 Void Loop

fungsi yang melakukan perulangan agar semua yang ada di dalam program tersebut agar berjalan secara berulang-ulang.

- **PUBLISHER:**

```
1 // PROJEK AKHIR PRAKTIKUM IOT //
2
3 #include <ESP8266WiFi.h> //Library untuk esp8266
4 #include <PubSubClient.h> //Library untuk MQTT
5 #include <NewPing.h> //Library untuk HC-SR04
6
7 const char* ssid = "Universitas Mulawarman";
8 const char* password = "";
9 const char* mqtt_server = "broker.hivemq.com";
```

gambar.1 17 Setup MQTT

Untuk menghubungkan ke internet ssidnya di atur sesuai dengan jaringan wifi yang digunakan seperti yang dicontoh di atas menggunakan Universitas Mulawarman sebagai ssidnya dan tidak menggunakan password agar mudah mengaksesnya. adapun server yang digunakan untuk broker yaitu mqtt_servernya adalah broker.hivemq.com untuk menghubungkan ke broker mqtt yang sudah di atur. Lalu ada inisiasi perangkat yang digunakan seperti buzzer pin di letakkan di pin D6 pada Ensp 32 serta led_pin pada D5 pada Ensp 32nya.

```
10
11 #define TRIGGER_PIN D5 //Pin Trigger HC-SR04 pada
12 #define ECHO_PIN D6 //Pin Echo HC-SR04 pada Nc
13 #define MAX_DISTANCE 400 //Maksimum Pembacaan Jarak
14 #define BUTTON_PIN D4 //Pin Buzzer pada NodeMCU
15
16 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
17
18 WiFiClient espClient;
19 PubSubClient client(espClient);
20 unsigned long lastMsg = 0;
21 #define MSG_BUFFER_SIZE (50)
22 char msg[MSG_BUFFER_SIZE];
23 int value = 0;
24 int buttonState = 0;
25
26 // Variable untuk menyimpan status sensor
27 bool sensorAktif = false;
28
```

gambar.1 18 Setup pin

Pada fungsi ini menggambarkan inisiasi pada sensor HC-SR4 dengan pin yang berbeda-beda, trigger pin pada D5, echo pin pada D6, button pin pada D4, dan max distancenya diatur 400.

```
29 void setup_wifi() {
30     delay(10);
31     Serial.println();
32     Serial.print("Connecting to ");
33     Serial.println(ssid);
34
35     WiFi.mode(WIFI_STA);
36     WiFi.begin(ssid, password);
37
38     while (WiFi.status() != WL_CONNECTED) {
39         delay(500);
40         Serial.print(".");
41     }
42
43     randomSeed(micros());
44     Serial.println("");
45     Serial.println("WiFi connected");
46     Serial.println("IP address: ");
47     Serial.println(WiFi.localIP());
48 }
49
```

gambar.1 19 Void Setup Wifi

Fungsi ini digunakan untuk mengatur dan menghubungkan NodeMCU ke jaringan WIFI menggunakan SSID dan password yang telah ditentukan. selama proses koneksi, fungsi akan menampilkan status koneksi WIFI dan alamat IP yang berhasil diperoleh

```

51 void callback(char* topic, byte* payload, unsigned int length) {
52     Serial.print("Pesan Diterima [");
53     Serial.print(topic);
54     Serial.print("] ");
55     for (int i = 0; i < length; i++) {
56         Serial.print((char)payload[i]);
57     }
58     Serial.println();
59 }
60

```

gambar.1 20 Void Callback

fungsi pada callback yang akan dipanggil ketika ada pesan yang akan diterima dari broker mqtt. fungsi ini akan menampilkan sebuah topik dan isi pesan yang akan diterima

```

61 // fungsi untuk menghubungkan ke broker
62 void reconnect() {
63     while (!client.connected()) {
64         Serial.print("Attempting MQTT connection...");
65         // Create a random client ID
66         String clientId = "ESP8266Client-";
67         clientId += String(random(0xffff), HEX);
68         // Attempt to connect
69         if (client.connect(clientId.c_str())) {
70             Serial.println("connected");
71             client.subscribe("iot_unmul/iot_b_1");
72         } else {
73             Serial.print("failed, rc=");
74             Serial.print(client.state());
75             Serial.println(" Coba lagi dalam 5 detik...");
76             delay(1000);
77         }
78     }
79 }

```

gambar.1 21 Void Reconnect

fungsi ini digunakan untuk menghubungkan NodeMCU ke broker mqtt apabila koneksi terputus fungsi ini akan mencoba untuk terhubung kembali dengan broker dan jika koneksi berhasil fungsi ini akan melakukan subscribe ke topik "iot_unmul/iot_b_1"

```
81 void setup() {  
82     Serial.begin(115200);  
83     setup_wifi();  
84     client.setServer(mqtt_server, 1883);  
85     client.setCallback(callback);  
86     pinMode(BUTTON_PIN, INPUT_PULLUP);  
87     sensorAktif = false;  
88     digitalWrite(TRIGGER_PIN, LOW);  
89 }
```

gambar.1 22 Void Setup

fungsi ini merupakan bagian setup utama yang akan dieksekusi pada awal program. pada fungsi ini ada inisiasi serial, koneksi WIFI, koneksi MQTT, serta mode pin dilakukan. pada pin BUTTON_PIN diatur sebagai input pull-up dan pin TRIGGER_PIN diatur sebagai output.


```

91 void loop() {
92     if (!client.connected()) {
93         reconnect();
94     }
95     client.loop();
96     int buttonState = digitalRead(BUTTON_PIN);
97
98     // Jika tombol ditekan
99     if (buttonState == LOW) {
100         // Ubah status sensor
101         sensorAktif = !sensorAktif;
102
103         // Tampilkan status sensor
104         if (sensorAktif) {
105             Serial.println("Sensor diaktifkan");
106             digitalWrite(TRIGGER_PIN, HIGH);
107             delayMicroseconds(10);
108             digitalWrite(TRIGGER_PIN, LOW);
109         } else {
110             Serial.println("Sensor dimatikan");
111         }
112
113         // Tunggu hingga tombol dilepas
114         while (buttonState == LOW) {
115             buttonState = digitalRead(BUTTON_PIN);
116         }
117
118         // Tunggu sejenak agar tidak terjadi bouncing
119         delay(50);
120     }
121     if (sensorAktif) {
122         unsigned int jarak = sonar.ping_cm(); //Melakukan Pembacaan Jarak dalam bentuk cm
123         delay(1000);
124         Serial.println("Sensor diaktifkan");
125         snprintf(msg, MSG_BUFFER_SIZE, "%s", itoa(jarak, msg, 10)); // itoa (konversi integer ke string)
126         Serial.print("Publish message: ");
127         Serial.print(msg);
128         Serial.println(" cm");
129         client.publish("iot_unmul", msg); // fungsi untuk publish ke broker
130     }
131     if (buttonState == HIGH) {
132     }
133     delay(1000);
134 }
135

```

gambar.1 23 Void Loop

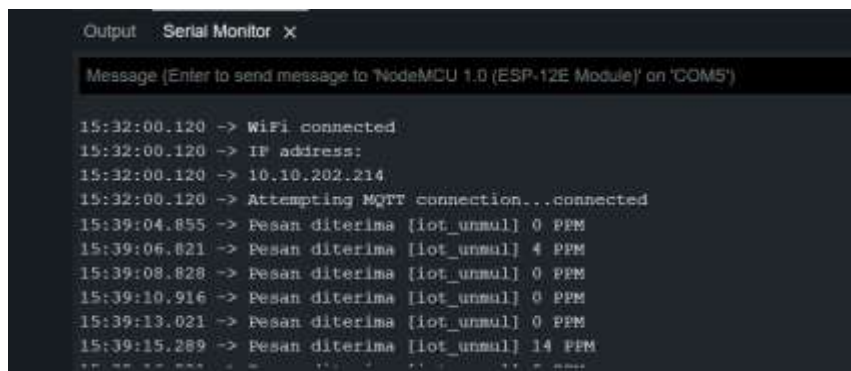
fungsi ini merupakan bagian utama program yang akan berjalan secara terus-menerus setelah proses setup selesai. pada fungsi ini dilakukan pengecekan koneksi mqtt dan pemantauan status tombol apabila jika tombol ditekan, status sensor diubah dan dilakukan pengiriman sinyal untuk mengaktifkan atau mematikan sensor ultrasonik. apabila sensor aktif akan dilakukan pengukuran jarak menggunakan sensor ultrasonik dan data jarak dikirimkan melalui mqtt setelah pengiriman data, terdapat penundaan 1 detik sebelum memasukkan iterasi selanjutnya.

F. PENGUJIAN SISTEM



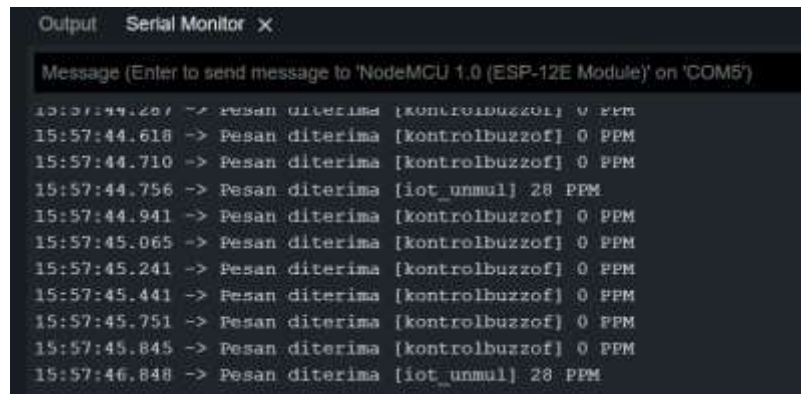
gambar.1 24 Hasil Monitoring pada Platform IoT

Setelah program di upload, pastikan hasil monitoring dapat dilihat pada platform IoT.



gambar.1 25 Seriam Monitor Subscribe

Ini hasil dari ketika program Subscribe telah berjalan yang ada serial monitor



```
Output  Serial Monitor X
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM5')
15:57:44.207 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:44.618 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:44.710 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:44.756 -> Pesan diterima [iot_unmul] 28 PPM
15:57:44.941 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:45.065 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:45.241 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:45.441 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:45.751 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:45.845 -> Pesan diterima [kontrolbuzzof] 0 PPM
15:57:46.848 -> Pesan diterima [iot_unmul] 28 PPM
```

gambar.1 26 Serial Monitor Subscribe

Ini hasil dari ketika program subscribe telah berjalan yang ada serial monitor.



```
Output  Serial Monitor X
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM11')
.....
WiFi connected
IP address:
10.10.200.61
Attempting MQTT connection...connected
```

gambar.1 27 Serial Monitor Publisher

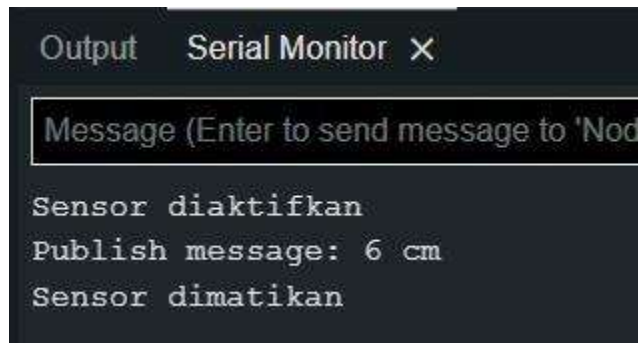
Ini hasil dari connection dari publishernya



```
Output  Serial Monitor X
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM11')
Attempting MQTT connection...connected
Sensor diaktifkan
Sensor diaktifkan
Publish message: 7 cm
Sensor diaktifkan
Publish message: 8 cm
Sensor diaktifkan
Publish message: 0 cm
Sensor diaktifkan
Publish message: 10 cm
Sensor diaktifkan
Publish message: 10 cm
```

gambar.1 28 Serial Monitor Publisher

Ini hasil dari publish messege ketika sensor diaktifkan



gambar.1 29 Sensor dImatikan

Ini hasil dari sensor publisher ketika dimatikan