

# TUBES IOT “PAKAN AYAM OTOMATIS BERBASIS IOT MENGGUNAKAN SENSOR ULTRASONIC”

## KELOMPOK X

- Pratyenggo Damar Iswara Putra (1103194141)
- Aditya Ramadhan (1103194161)
- Muhammad Iqbal (1103194114)
- Athmad Taufik Rahman (1103190028)

### • LATAR BELAKANG

Proyek pakan ayam otomatis berbasis Internet of Things (IoT) dengan menggunakan sensor ultrasonik memiliki latar belakang yang mencakup efisiensi pengelolaan pakan, pemantauan kesehatan ayam, penghematan energi, monitoring jarak jauh, optimasi produktivitas, tren teknologi pertanian, dan keberlanjutan lingkungan. Penggunaan sensor ultrasonik dalam proyek ini mencerminkan tren menuju pertanian pintar untuk meningkatkan efisiensi, produktivitas, dan keberlanjutan.

**TUJUAN PROJECT** Tujuan dari project ini adalah pemenuhan nilai tugas besar untuk mata kuliah IoT (Internet Of Things) dan memahami cara kerja IoT.

### • DASAR TEORI

- Alat Yang Dibutuhkan :
1. Node MCU



NodeMCU merupakan sebuah papan pengembangan (development board) berbasis mikrokontroler ESP8266, yang memiliki dukungan WiFi terintegrasi. ESP8266 adalah sebuah mikrokontroler yang sangat populer dalam pengembangan IoT karena kemampuannya untuk terhubung dengan jaringan WiFi. NodeMCU menggunakan firmware Lua, sebuah bahasa pemrograman scripting ringan yang memungkinkan pengembangan perangkat lunak untuk IoT menjadi lebih mudah.

## 2. Breadboard



Breadboard adalah perangkat atau papan percobaan yang digunakan dalam elektronika untuk merancang, menguji, dan menyusun prototipe rangkaian listrik tanpa harus melakukan soldering.

## 3. Kabel Data Micro USB



Untuk menghubungkan laptop dengan wemos node MC

#### 4. HCS 04 Ultrasonic



HC-SR04 Ultrasonic adalah sebuah sensor ultrasonik yang banyak digunakan dalam proyek-proyek elektronika, terutama dalam konteks pengukuran jarak.

#### 5. Paket Kabel Jumper



Kabel jumper adalah kabel pendek yang digunakan dalam elektronika dan prototyping untuk menghubungkan dua titik atau komponen dalam rangkaian listrik.

- KONEKSI YANG DIGUNAKAN :
  - WiFi - WiFi adalah teknologi nirkabel yang memungkinkan perangkat elektronik, seperti komputer, smartphone, tablet, dan perangkat lainnya, untuk terhubung ke jaringan internet atau jaringan lokal (LAN) tanpa menggunakan kabel fisik.

- MQTT - MQTT (Message Queuing Telemetry Transport) adalah protokol komunikasi ringan yang dirancang untuk digunakan pada kondisi jaringan yang tidak stabil atau dengan bandwidth terbatas.

-

-

- SOURCE CODE

## 1. NODE SENSOR

```
1  #include <ESP8266WiFi.h>
2  #include <PubSubClient.h>
3  #include <Wire.h>
4  #include <TimeLib.h>
5  #include <NTPClient.h>
6  #include <WiFiUdp.h>
7
8  const char* ssid = "Elli_plus"; // Nama Wifi
9  const char* password = "selafanyfrida"; // Password Wifi
10 const char* mqtt_server = "test.mosquitto.org"; // Alamat IP atau nama host MQTT broker
11 const char* control_topic = "actuator/control"; // Topik MQTT untuk mengendalikan servomotor
12
13 WiFiUDP ntpUDP;
14 NTPClient timeClient(ntpUDP, "pool.ntp.org");
15
16 #define TRIG_PIN D1
17 #define ECHO_PIN D2
18
19 WiFiClient espClient;
20 PubSubClient client(espClient);
21
22 void setup() {
23   Serial.begin(115200);
24   setup_wifi();
25   client.setServer(mqtt_server, 1883);
26
27   pinMode(TRIG_PIN, OUTPUT);
28   pinMode(ECHO_PIN, INPUT);
29
30   // Inisialisasi waktu dari server NTP
31   timeClient.begin();
32   timeClient.setTimeOffset(25200); // Waktu offset untuk WIB (UTC+7)
33 }
```

```

32   timeClient.setTimeOffset(25200); // Waktu offset untuk WIB (UTC+7)
33 }
34
35 void loop() {
36   if (!client.connected()) {
37     reconnect();
38   }
39   client.loop();
40
41   // Dapatkan waktu dari server NTP
42   timeClient.update();
43   int currentHour = timeClient.getHours();
44
45   // Kirim data jarak ke Node-RED
46   long duration, distance;
47   digitalWrite(TRIG_PIN, LOW);
48   delayMicroseconds(2);
49   digitalWrite(TRIG_PIN, HIGH);
50   delayMicroseconds(10);
51   digitalWrite(TRIG_PIN, LOW);
52   duration = pulseIn(ECHO_PIN, HIGH);
53   distance = duration * 0.034 / 2;
54
55   char distanceMsg[20];
56   snprintf(distanceMsg, sizeof(distanceMsg), "%ld", distance);
57   client.publish("sensor/distance", distanceMsg);
58
59   // Kirim instruksi kontrol pada waktu tertentu
60   if ((currentHour == 7 || currentHour == 13 || currentHour == 22) && (timeClient.getMinutes() >= 0 &&
timeClient.getMinutes() <= 10)) {
61     // Kirim instruksi berdasarkan tinggi pakan
62     if (distance < 5) {
63       client.publish(control_topic, "open");
64     } else if (distance > 10) {

```

```

63       client.publish(control_topic, "open");
64     } else if (distance > 10) {
65       client.publish(control_topic, "close");
66     }
67   }
68
69   delay(2000);
70 }
71
72 void setup_wifi() {
73   delay(10);
74   Serial.println();
75   Serial.print("Connecting to ");
76   Serial.println(ssid);
77
78   WiFi.begin(ssid, password);
79
80   while (WiFi.status() != WL_CONNECTED) {
81     delay(500);
82     Serial.print(".");
83   }
84
85   Serial.println("");
86   Serial.println("WiFi connected");
87   Serial.println("IP address: ");
88   Serial.println(WiFi.localIP());
89 }
90
91 void reconnect() {
92   while (!client.connected()) {
93     Serial.print("Attempting MQTT connection...");
94
95     if (client.connect("sensor_node")) {
96       Serial.println("connected");

```

```

84
85 Serial.println("");
86 Serial.println("WiFi connected");
87 Serial.println("IP address: ");
88 Serial.println(WiFi.localIP());
89 }
90
91 void reconnect() {
92   while (!client.connected()) {
93     Serial.print("Attempting MQTT connection...");
94
95     if (client.connect("sensor_node")) {
96       Serial.println("connected");
97     } else {
98       Serial.print("failed, rc=");
99       Serial.print(client.state());
100       Serial.println(" try again in 5 seconds");
101       delay(5000);
102     }
103   }
104 }

```



## 2. NODE SERVO

```

1  #include <ESP8266WiFi.h>
2  #include <PubSubClient.h>
3  #include <Servo.h>
4
5  const char* ssid = "Elli_plus"; // Nama WiFi
6  const char* password = "selafanyfrida"; // Password WiFi
7  const char* mqtt_server = "test.mosquitto.org"; // Alamat IP atau nama host MQTT broker
8  const char* control_topic = "actuator/control"; // Topik MQTT untuk mengendalikan servomotor
9
10 WiFiClient espClient;
11 PubSubClient client(espClient);
12
13 Servo servoMotor;
14
15 void setup() {
16     Serial.begin(115200);
17     setup_wifi();
18     client.setServer(mqtt_server, 1883);
19     client.setCallback(callback);
20
21     servoMotor.attach(D1); // Sesuaikan dengan pin yang digunakan untuk servo
22     servoMotor.write(90); // Posisi awal servo
23 }
24
25 void loop() {
26     if (!client.connected()) {
27         reconnect();
28     }
29     client.loop();
30 }
31
32 void setup_wifi() {
33     delay(10);

```

```

34     Serial.println();
35     Serial.print("Connecting to ");
36     Serial.println(ssid);
37
38     WiFi.begin(ssid, password);
39
40     while (WiFi.status() != WL_CONNECTED) {
41         delay(500);
42         Serial.print(".");
43     }
44
45     Serial.println("");
46     Serial.println("WiFi connected");
47     Serial.println("IP address: ");
48     Serial.println(WiFi.localIP());
49 }
50
51 void reconnect() {
52     while (!client.connected()) {
53         Serial.print("Attempting MQTT connection...");
54
55         if (client.connect("actuator_node")) {
56             Serial.println("connected");
57             client.subscribe(control_topic);
58         } else {
59             Serial.print("failed, rc=");
60             Serial.print(client.state());
61             Serial.println(" try again in 5 seconds");
62             delay(5000);
63         }
64     }
65 }

```

```

64 }
65 }
66
67 void callback(char* topic, byte* payload, unsigned int length) {
68   Serial.print("Message arrived [");
69   Serial.print(topic);
70   Serial.print("] ");
71
72   // Buka atau tutup servo tergantung pada payload yang diterima
73   if (strcmp(topic, control_topic) == 0) {
74     if (strcmp((char*)payload, "open") == 0) {
75       servoMotor.write(0); // Sesuaikan dengan sudut yang diinginkan untuk membuka servo
76     } else if (strcmp((char*)payload, "close") == 0) {
77       servoMotor.write(90); // Sesuaikan dengan sudut yang diinginkan untuk menutup servo
78     }
79   }
80 }

```

- RINCIAN BIAYA

NO	Nama Barang	Jumlah	Harga	Jumlah
1	Node MCU	2	Rp. 35.000	75.000
2	Micro Servo mg90	1	Rp. 25.000	25.000
3	Kabel Data Micro USB	2	Rp. 7.500	13.000
4	HCSR04 Ultrasonic	2	Rp. 12.000	24.000
5	Paket Kabel Jumper	1	Rp. 26.000	26.000
	Total			Rp. 163.000

- CARA KERJA ALAT

Node pertama (Node MCU terhubung dengan HCSR04) mengirim data berupa data jarak melalui mqtt dan di tampilkan melalui node red, dengan juga dikirim data berupa close dan open di waktu yang telah di tentukan, variable open dan close akan dibaca oleh Node kedua (Node MCU terhubung dengan servo motor) jika menerima mqtt open maka servo akan bergerak untuk membuka pakan jika close akan menutup jalur pakan. Tetapi kami disini menemukan kendala berupa alat tidak bisa manual dan sensor sering eror mendapatkan jarak



- KESIMPULAN

Kesimpulan dari laporan di atas adalah bahwa proyek pakan ayam otomatis berbasis Internet of Things (IoT) menggunakan sensor ultrasonik telah berhasil dikembangkan oleh Kelompok X. Latar belakang proyek mencakup berbagai aspek, termasuk efisiensi pengelolaan pakan, pemantauan kesehatan ayam, penghematan energi, monitoring jarak jauh, optimasi produktivitas, tren teknologi pertanian, dan keberlanjutan lingkungan. Tujuan proyek melibatkan pemenuhan tugas besar untuk mata kuliah IoT dan pemahaman tentang cara kerja IoT. Dalam pelaksanaannya, kelompok menggunakan NodeMCU, sensor ultrasonik HC-SR04, servo motor, dan komponen elektronik lainnya. Penggunaan NodeMCU sebagai platform pengembangan memungkinkan konektivitas WiFi, sementara sensor ultrasonik digunakan untuk mengukur jarak dan servo motor untuk mengendalikan pemberian pakan. Meskipun proyek ini memberikan pemahaman yang lebih dalam tentang aplikasi IoT, kelompok menghadapi kendala, seperti ketidakmampuan operasi manual dan masalah sensor ultrasonik yang sering eror dalam pengukuran jarak. Rincian biaya juga disajikan untuk memberikan gambaran tentang investasi yang dibutuhkan untuk mengembangkan proyek ini. Kesimpulannya, proyek ini mencerminkan upaya yang signifikan dalam menerapkan teknologi IoT untuk meningkatkan efisiensi dan otomatisasi dalam manajemen pakan ayam. Dengan memahami kelebihan dan kendala proyek, dapat dilakukan pengembangan lebih lanjut untuk meningkatkan kinerja dan ketangguhan sistem.