DAFTAR PUSTAKA

Achmad Fauzi. 2017. *Sistem Kontrol Suhu Ruangan Pada Inkubator Anak Ayam Menggunakan ESP Wemos DI Berbasis IoT.* Surabaya: Universitas Narotama.

Aji Ridhamuttaqin, dkk. 2013. *Rancang Bangun Model Sistem Pemberi Pakan Ayam Otomatis Berbasis Fuzzy Logic Control.* Lampung: Universitas Lampung.

Arduino.cc

Arief Budi Laksono, 2017. *Rancang Bangun Sistem Pemberi Pakan Ayam Serta Monitoring Suhu dan Kelembaban Kandang Berbasis Atmega328.* Lamongan: Universitas Islam Lamongan.

Berliana Siregar, dkk. 2016. *Pengaruh Pengaturan Waktu Pemberian Pakan Selama Periode Pertumbuhan Ayam Broiler Terhadap Rasio Efisiensi Penggunaan Protein.* Jambi: Universitas Jambi.

Hudan Abdur Rochman, dkk. 2017. *Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome.* Malang: Universitas Brawijaya.

Nadzir, dkk. 2015. *Evaluasi Desain Kandang Ayam Broiler Di Desa Rejo Binangun, Kecamatan Raman Utara, Kabupaten Lampung Timur.* Lampung: Universitas Lampung.

Raditiya Prihandanu, dkk. 2015. *Model Sistem Kandang Ayam Closed House Otomatis Menggunakan Omron Sysmac CPM1A 20-CDR-A-V1.* Lampung: Universitas Lampung.

Rupali B. Mahale, dkk. 2016. *Food and Water Level Control Mechanism for Smart Chicken Poultry Farming.* Pune: Dr. D. Y. Patil School of Engineering.

Triawati, E., Alona, Jamilah, Taufik, M., 2013. *Ebook Mikrokontroler.* Jakarta: Universitas Gunadarma.

Wemos.cc

Yudhanto, Yudha. 2007. *Apa Itu IoT (Internet of Things) ?.* ilmukomputer.com.

Lampiran 1. Source Code Program ESP Wemos D1

```
#include <Servo.h>
#include <PubSubClient.h>
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_BME680.h"

Servo myservo;
Servo myservo2;
Adafruit_BME680 bme;

const char* ssid = "routersf";
const char* password = "11101110";
const char* mqtt_server = "202.154.58.84";
const char* mqtt_user = "narotama";
const char* mqtt_password = "narotamajaya";
WiFiClient espClient;
PubSubClient client(espClient);

const int pinlamp = D4;
const int pinlamp2 = D5;
const int pinblowin = D6;
const int pinblowout = D7;
long now = millis();
long lastMeasure = 0;

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
```

```
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected - ESP IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(String topic, byte* message, unsigned int length) {
  Serial.print("Message arrived on topic: ");
  Serial.print(topic);
  Serial.print(". Message: ");
  String messageTemp;

  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
    messageTemp += (char)message[i];
  }
  Serial.println();

  // Feel free to add more if statements to control more GPIOs with MQTT

  // If a message is received on the topic room/lamp, you check if the message is
either on or off. Turns the lamp GPIO according to the message
  if(topic=="narotama/lamp"){
      Serial.print("Changing Room lamp to ");
      if(messageTemp == "on"){
        digitalWrite(pinlamp, LOW);
        Serial.print("On");
      }
      else if(messageTemp == "off"){
        digitalWrite(pinlamp, HIGH);
        Serial.print("Off");
      }
  }
```

```
if(topic=="narotama/lamp2"){
   Serial.print("Changing Room lamp2 to ");
   if(messageTemp == "on"){
     digitalWrite(pinlamp2, LOW);
     Serial.print("On");
   }
   else if(messageTemp == "off"){
     digitalWrite(pinlamp2, HIGH);
     Serial.print("Off");
   }
}
if(topic=="narotama/blowin"){
   Serial.print("turn blower in to ");
   if(messageTemp == "on"){
     digitalWrite(pinblowin, LOW);
     Serial.print("On");
   }
   else if(messageTemp == "off"){
     digitalWrite(pinblowin, HIGH);
     Serial.print("Off");
   }
}
if(topic=="narotama/blowout"){
   Serial.print("Turn blower out to ");
   if(messageTemp == "on"){
     digitalWrite(pinblowout, LOW);
     Serial.print("On");
   }
   else if(messageTemp == "off"){
     digitalWrite(pinblowout, HIGH);
     Serial.print("Off");
   }
}
if(topic=="narotama/dimmer"){
   Serial.print("set dimmer");
   int msgpos = messageTemp.toInt();
```

```
    int pos = myservo2.read();
    myservo2.attach(D3);
    while (pos != msgpos){
     if (pos > msgpos){
      pos -= 1;
      myservo2.write(pos);
      delay(15);
     }
     if (pos < msgpos){
      pos += 1;
      myservo2.write(pos);
      delay(15);
     }
    }
    myservo2.detach();

}
if(topic=="narotama/door"){
    Serial.print("open door");
    if(messageTemp == "open"){

     int pos;

      for(pos = 0; pos <= 95; pos += 1) // goes from 0 degrees to 180 degrees
      {                                 // in steps of 1 degree
      myservo.write(pos);               // tell servo to go to position in variable 'pos'
      delay(15);                        // waits 15ms for the servo to reach the position
      }
      delay(3000);
      for(pos = 95; pos>=0; pos-=1)     // goes from 180 degrees to 0 degrees
      {
      myservo.write(pos);               // tell servo to go to position in variable 'pos'
      delay(1);                         // waits 15ms for the servo to reach the position
      }
    }
```

```
  }
  Serial.println();
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect

    if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
      Serial.println("connected");
      // Subscribe or resubscribe to a topic
      // You can subscribe to more topics (to control more LEDs in this example)
      client.subscribe("narotama/lamp");
      client.subscribe("narotama/lamp2");
      client.subscribe("narotama/blowin");
      client.subscribe("narotama/blowout");
      client.subscribe("narotama/door");
      client.subscribe("narotama/dimmer");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

void setup() {
  myservo.attach(D8);
  myservo2.attach(D3);
  pinMode(D4, OUTPUT);
  pinMode(D5, OUTPUT);
  pinMode(D6, OUTPUT);
```

```
  pinMode(D7, OUTPUT);
  Serial.begin(9600);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);

  if (!bme.begin()) {
    Serial.println("Could not find a valid BME680 sensor, check wiring!");
    while (1);
  }

  bme.setTemperatureOversampling(BME680_OS_8X);
  bme.setHumidityOversampling(BME680_OS_2X);
}

// the loop function runs over and over again forever
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  if(!client.loop())
    client.connect("ESP8266Client", mqtt_user, mqtt_password);


  if (! bme.performReading()) {
    Serial.println("Failed to perform reading :(");
    return;
  }

  float t = bme.temperature;
  float h = bme.humidity;

  char temper[7];
  dtostrf(t,0,2,temper);

  char humi[7];
```

```
dtostrf(h,0,0,humi);

client.publish("narotama/temperature", temper);
client.publish("narotama/humidity", humi);

Serial.print("Temperature = ");
Serial.print(t);
Serial.println(" *C");

Serial.print("Humidity = ");
Serial.print(h);
Serial.println(" %");

Serial.println();
delay(1000);
}
```

Lampiran 2. Source Code Program Aplikasi Android

```
package id.my.firnanda.smartbreedingapps;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
   private WebView webView;


   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      webView = (WebView) findViewById(R.id.webview);
      webView.setWebViewClient(new WebViewClient(){
         public void onReceivedError(WebView view, int errorCode, String
description, String failingUrl){
            webView.loadUrl("file:///android_asset/error.html");
         }
      });
      webView.loadUrl("http://117.53.47.249:1880/ui/");

      WebSettings webSettings = webView.getSettings();
      webSettings.setJavaScriptEnabled(true);
      webSettings.setDomStorageEnabled(true);
      webSettings.setJavaScriptCanOpenWindowsAutomatically(true);
   }
```

```
    private static long back_pressed;
    @Override
    public void onBackPressed() {
        if (back_pressed + 2000 > System.currentTimeMillis()){
            super.onBackPressed();
        } else {
            Toast.makeText(getBaseContext(), "Tekan Sekali Lagi Untuk Keluar!",
Toast.LENGTH_SHORT).show();
        }
        back_pressed = System.currentTimeMillis();
    }
}
```