

DAFTAR PUSTAKA

- Ashardiono, F., & Trihartono, A. (2024). Optimizing the potential of Indonesian coffee: a dual market approach. *Cogent Social Sciences*, 10(1). <https://doi.org/10.1080/23311886.2024.2340206>
- Fauzan, M., & Nurdin, N. (2024). Implementasi Thingspeak Sebagai Database Pada Alat Deteksi Banjir Menggunakan Esp32. *CONTEN : Computer and Network Technology*, 4(1), 66–70. <https://doi.org/10.31294/conten.v4i1.3596>
- Martinez, J. M. T., López, A. G., Sánchez, M. G. B., Barranco, Gutierrez, A. I., Nolasco, J. J. M., & Ortega, F. V. (2025). Instrumentation and Evaluation of a Sensing System with Signal Conditioning Using Fuzzy Logic for a Rotary Dryer. *Technologies*, 13(2), 83. <https://doi.org/10.3390/technologies13020083>
- Mitra, P., Dey, C., & Mudi, R. K. (2021). Fuzzy rule-based set point weighting for fuzzy PID controller. In *SN Applied Sciences* (Vol. 3, Issue 6). Springer International Publishing. <https://doi.org/10.1007/s42452-021-04626-0>
- Nafisah, N., Syamsiana, I. N., Putri, R. I., Kusuma, W., & Sumari, A. D. W. (2024). Implementation of fuzzy logic control algorithm for temperature control in robusta rotary dryer coffee bean dryer. *MethodsX*, 12, 102580. <https://doi.org/10.1016/j.mex.2024.102580>
- Nurhandayani, K., & Rivai, M. (2018). Sistem Kontrol Pengering Makanan Berbasis LED Inframerah. *Jurnal Teknik ITS*, 7(2), A355–A361.
- Pei, Y., Li, Z., Ling, C., Jiang, L., Wu, X., Song, C., Li, J., Song, F., & Xu, W. (2022). An improvement of far-infrared drying for ginger slices with computer vision and fuzzy logic control. *Journal of Food Measurement and Characterization*, 16(5), 3815–3831. <https://doi.org/10.1007/s11694-022-01453-8>
- Setiawati, I., & Kasyfillah, L. (2022). Pascapanen dan Pemasaran Kopi Arabika di PT. Sinar Mayang Lestari, Kecamatan Pangalengan, Kabupaten Bandung. *Agricore: Jurnal Agribisnis Dan Sosial Ekonomi Pertanian Unpad*, 7(2), 108–122. <https://doi.org/10.24198/agricore.v7i2.40433>
- Silva, C. S. da, Coelho, A. P. de F., Lisboa, C. F., Vieira, G., & Teles, M. C. de A. (2022). Post-harvest of coffee: factors that influence the final quality of the beverage. *Revista Engenharia Na Agricultura - REVENG*, 30, 49–62. <https://doi.org/10.13083/reveng.v30i1.12639>
- Skåra, T., Løvdal, T., Skipnes, D., Nwabisa Mehlomakulu, N., Mapengo, C. R., Otema Baah, R., & Emmambux, M. N. (2023). Drying of vegetable and root

crops by solar, infrared, microwave, and radio frequency as energy efficient methods: A review. *Food Reviews International*, 39(9), 7197–7217. <https://doi.org/10.1080/87559129.2022.2148688>

STATISTIK KOPI INDONESIA 2022. (2023). In *Badan Pusat Statistik Indonesia* (Vol. 7, Issue 1). http://www.biblioteca.pucminas.br/teses/Educacao_PereiraAS_1.pdf%0Ahttp://www.anpocs.org.br/portal/publicacoes/rbcs_00_11/rbcs11_01.htm%0Ahttp://repositorio.ipea.gov.br/bitstream/11058/7845/1/td_2306.pdf%0Ahttps://direitofma2010.files.wordpress.com/2010/

Syamsiana, I. N., Wahyudi, M. A., Syamsuri, T. U., Nafisah, N., & Sumari, A. D. W. (2024). Implementasi Sistem Monitoring Alat Pengereng Biji Kopi Berbasis IoT (Internet of Things). *Elposys: Jurnal Sistem Kelistrikan*, 11(1), 62–67. <https://doi.org/10.33795/elposys.v11i1.4964>

Wang, T., Xia, L., Ni, M., Pan, S., & Luo, C. (2024). Fundamentals of Infrared Heating and Their Application in Thermosetting Polymer Curing: A Review. *Coatings*, 14(7), 875. <https://doi.org/10.3390/coatings14070875>

LAMPIRAN

Lampiran 1. Dokumentasi Kegiatan Penelitian



Berat sample 100gram



Berat sample 1000gram



Berat akhir 53gram



Berat akhir 529gram



Kadar air sample 100gram



Kadar air sample 1000gram



Perangkaian sistem



Pembuatan tutup mesin pengering

Lampiran 2. Dokumentasi Kegiatan Penelitian



Perangkaan mesin



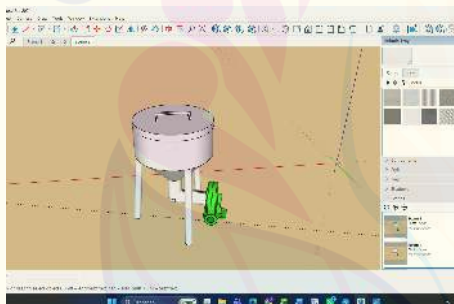
Kopi yang akan dikeringkan



Proses pengeringan



Tampilan monitoring website



Gambar 3d mesin pengering



Lampu pemanas inframerah

Lampiran 3. Kode untuk ESP 32

```

#include <WiFi.h>
#include <ThingSpeak.h>
#include <Wire.h>
#include <DHT.h>
#include <RTCLib.h>

#define DHTTYPE DHT22

//===== WIFI =====
const char* ssid    = "fh_401ae0";
const char* password = "wlanbfe51f";
//===== THINGSPEAK =====
unsigned long channelID = 3404081;
const char* writeAPIKey = "CLTV402ABCGBWQWW";
//===== PIN =====
#define DHT1_PIN 32
#define DHT2_PIN 33
#define PEM 18
#define KIP 19
#define SDA_PIN 21
#define SCL_PIN 22
//=====
DHT dht1(DHT1_PIN,DHTTYPE);
DHT dht2(DHT2_PIN,DHTTYPE);
RTC_DS3231 rtc;
WiFiClient client;
//=====
float t1,h1,t2,h2;
float avgTemp=0;
float avgHum=0;
float heaterScore=0;
float fanScore=0;

```

```
bool heaterState=false;
bool fanState=false;
unsigned long lastRead=0;
unsigned long lastUpload=0;

//=====
// MEMBERSHIP SUHU
float suhuDingin(float t){
    if(t<=45) return 1;
    if(t>=55) return 0;
    return (55-t)/10.0;
}
float suhuIdeal(float t){
    if(t<=45 || t>=60) return 0;
    if(t<=52.5)
        return (t-45)/7.5;
    return (60-t)/7.5;
}
float suhuPanas(float t){
    if(t<=55) return 0;
    if(t>=65) return 1;
    return (t-55)/10.0;
}
//=====
// MEMBERSHIP RH
float humKering(float h){
    if(h<=30) return 1;
    if(h>=40) return 0;
    return (40-h)/10.0;
}
float humSedang(float h){
```

```
    if(h<=30 || h>=50) return 0;
    if(h<=40)
        return (h-30)/10.0;
    return (50-h)/10.0;
}
float humBasah(float h){
    if(h<=40) return 0;
    if(h>=60) return 1;
    return (h-40)/20.0;
}
bool validData(float t,float h){
    return !isnan(t) &&
           !isnan(h) &&
           t>0 && t<80 &&
           h>=0 && h<=100;
}
//=====
void connectWiFi(){
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid,password);
    int n=0;
    while(WiFi.status()!=WL_CONNECTED && n<20){
        delay(500);
        n++;
    }
}
//=====
void setup(){
    Serial.begin(115200);
    Wire.begin(SDA_PIN,SCL_PIN);
    pinMode(PEM,OUTPUT);
```

```
pinMode(KIP,OUTPUT);
digitalWrite(PEM,HIGH);
digitalWrite(KIP,HIGH);
dht1.begin();
dht2.begin();
rtc.begin();
connectWiFi();
ThingSpeak.begin(client);
}
//=====
void loop(){
  if(WiFi.status()!=WL_CONNECTED)
    connectWiFi();
  if(millis()-lastRead>=2000){
    lastRead=millis();
    readSensor();
    fuzzyControl();
    printStatus();
  }
  if(millis()-lastUpload>=20000){
    lastUpload=millis();
    sendThingSpeak();
  }
}
//=====
void readSensor(){
  t1=dht1.readTemperature();
  h1=dht1.readHumidity();
  t2=dht2.readTemperature();
  h2=dht2.readHumidity();
```



```
bool s1=validData(t1,h1);
bool s2=validData(t2,h2);
if(s1 && s2){
    avgTemp=(t1+t2)/2.0;
    avgHum =(h1+h2)/2.0;
}
else if(s1){
    avgTemp=t1;
    avgHum=h1;
}
else if(s2){
    avgTemp=t2;
    avgHum=h2;
}
else{
    avgTemp=0;
    avgHum=0;
}
}
//=====
void fuzzyControl(){
    if(avgTemp==0){
        heaterState=false;
        fanState=true;
        digitalWrite(PEM,HIGH);
        digitalWrite(KIP,LOW);
        return;
    }
    float dingin=suhuDingin(avgTemp);
    float ideal =suhuIdeal(avgTemp);
    float panas =suhuPanas(avgTemp);
```

```
float kering=humKering(avgHum);
float sedang=humSedang(avgHum);
float basah =humBasah(avgHum);

// Heater score
heaterScore=
    dingin*100 +
    ideal*50 +
    panas*0;

// Fan score
fanScore=
    basah*100 +
    sedang*50 +
    kering*0;

// Normalisasi
float denT=dingin+ideal+panas;
float denH=basah+sedang+kering;
if(denT>0) heaterScore/=denT;
if(denH>0) fanScore/=denH;

// Hysteresis Heater
if(heaterScore>=60)
    heaterState=true;
else if(heaterScore<=40)
    heaterState=false;

// Hysteresis Fan
if(fanScore>=60)
    fanState=true;
else if(fanScore<=40)
    fanState=false;

// Proteksi
if(avgTemp>=65){
```

```

        heaterState=false;
        fanState=true;
    }
    digitalWrite(PEM,
        heaterState ? LOW : HIGH);
    digitalWrite(KIP,
        fanState ? LOW : HIGH);
}
//=====================================================
void sendThingSpeak(){
    if(WiFi.status()!=WL_CONNECTED)
        return;
    ThingSpeak.setField(1,t1);
    ThingSpeak.setField(2,h1);

    ThingSpeak.setField(3,t2);
    ThingSpeak.setField(4,h2);

    ThingSpeak.setField(5,avgTemp);
    ThingSpeak.setField(6,avgHum);

    // Relay 0-1
    ThingSpeak.setField(7,
        heaterState ? 1 : 0);
    ThingSpeak.setField(8,
        fanState ? 1 : 0);
    ThingSpeak.writeFields(
        channelID,
        writeAPIKey);
}
//=====================================================

```

```
void printStatus(){
    DateTime now=rtc.now();
    Serial.println();
    Serial.println("=====");
    Serial.printf("Tanggal : %02d/%02d/%04d\n",
        now.day(),
        now.month(),
        now.year());
    Serial.printf("Jam : %02d:%02d:%02d\n",
        now.hour(),
        now.minute(),
        now.second());
    Serial.println("-----");
    Serial.printf("Temp1 : %.1f C\n",t1);
    Serial.printf("Hum1 : %.1f %%\n",h1);
    Serial.printf("Temp2 : %.1f C\n",t2);
    Serial.printf("Hum2 : %.1f %%\n",h2);
    Serial.println("-----");
    Serial.printf("Avg Temp : %.1f C\n",avgTemp);
    Serial.printf("Avg Hum : %.1f %%\n",avgHum);
    Serial.println("-----");
    Serial.print("PEM : ");
    Serial.println(heaterState ? "ON" : "OFF");
    Serial.print("KIP : ");
    Serial.println(fanState ? "ON" : "OFF");
    Serial.println("=====");
}
```