LAMPIRAN

Coding program “aplikasi arduino untuk kontrol serta monitoring suhu dan kelembaban kumbung jamur kuping”

NAMA : Muhammad Ghozali

NIM : 141220000016

PRODI : TEKNIK ELEKTRO

FAKULTAS : SAINS DAN TEKNOLOGI UNIVERSITAS ISLAM NAHDLATUL ULAMA JEPARA

Lampiran 1 Coding Program Ethernet W5100 Dan BME280

```
#include <Wire.h>
#include <SPI.h>
#include <Ethernet.h>
#include <Adafruit_BME280.h>
Adafruit_BME280 bme;
#include "ThingSpeak.h"
byte mac[] = { 0xD4, 0xA8, 0xE2, 0xFE, 0xA0, 0xA1 };

EthernetClient client;

unsigned long Nomor_Chanel = 1472719  ;
const char * Write_API_Key =  "Q3DINZOWSE8HBEHT";

float Kelembaban, Suhu;

void setup()
{
 if (!bme.begin(0x76)) {
   Serial.println(F("Could not find a valid BME280 sensor, check wiring!"));
  }
  Ethernet.init(10);  // Most Arduino Ethernet hardware
  while (!Serial)
  {
    ; // wait for serial port to connect.
  }
  Serial.println("Initialize Ethernet with DHCP:");
  if (Ethernet.begin(mac) == 0)
  {
    Serial.println("Failed to configure Ethernet using DHCP");
```

```
    // Check for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware)
    {
      Serial.println("Ethernet shield was not found.");
      while (true)
      {
        delay(1); // do nothing, no point running without Ethernet hardware
      }
    }
    if (Ethernet.linkStatus() == LinkOFF)
    {
      Serial.println("Ethernet cable is not connected.");
    }
    Ethernet.begin(mac);
  }
  else
  {
    Serial.print("  DHCP assigned IP ");
    Serial.println(Ethernet.localIP());
  }
  delay(1000);
  ThingSpeak.begin(client);
}
void loop()
{
  Suhu       = bme.readTemperature();
  Kelembaban  = bme.readHumidity();
  Serial.println       (Kelembaban);
  Serial.println       (Suhu);
  ThingSpeak.setField   (1, String (Kelembaban));
  ThingSpeak.setField   (2, String (Suhu));
  ThingSpeak.writeFields(Nomor_Chanel, Write_API_Key);
}
```

Lampiran 2 Coding Program Lengkap

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_TFTLCD.h>
#include <TouchScreen.h>
#include <Ethernet.h>
#include <Adafruit_BME280.h>
Adafruit_BME280 bme;

#define YP A1
#define XM A2
#define YM 7
#define XP 6

#define TS_MINX 100
#define TS_MINY 120
#define TS_MAXX 920
#define TS_MAXY 940

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 500);

#define LCD_CS A3
#define LCD_CD A2
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4

Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);

#define BLACK   0x0000
int BLUE = tft.color565(50, 50, 255);
#define DARKBLUE 0x0010
#define VIOLET  0x8888
#define RED     0xF800
#define GREEN   0x07E0
#define CYAN    0x07FF
#define MAGENTA 0xF81F
#define YELLOW  0xFFE0
#define WHITE   0xFFFF
#define GREY    0x2108
#define GOLD    0xFEA0
#define BROWN   0xA145
#define SILVER  0xC618
```

```
#define LIME    0x07E0
#define MINPRESSURE 10
#define MAXPRESSURE 1000

#include "ThingSpeak.h"
byte mac[] = { 0xD4, 0xA8, 0xE2, 0xFE, 0xA0, 0xA1 };
EthernetClient client;

unsigned long Nomor_Chanel = 1324901  ;
const char * Write_API_Key =  "TQUG8LPMQZZF4Z8K";
unsigned long previous_Millis_SENSOR      = 0;
const long    interval_SENSOR           = 1000*1;
unsigned long previous_Millis_Auto       = 0;
const long    interval_Auto             = 1000*5;

unsigned long previous_Millis_thingspeak  = 0;
const long    interval_thingspeak       = 1000*10;

float Kelembaban, Suhu;
int halaman,NyalaMati, x, y;
int nyala    = 1, mati     = 0;
int Pemanas  = A8, Pompa   = A9, humid  = A10, kipas  = A11;
int Lampu1   = A12, Lampu2  = A13, Lampu3 = A14, Lampu4 = A15;

#define Tekanan       (p.z > 10 && p.z < 1000)
#define TM_manual     (p.x >  0 && p.x <  400 && p.y >   0 && p.y < 250)
#define T_Home        (p.x >450 && p.x <  670 && p.y >   0 && p.y < 250)
#define TM_lampu      (p.x >720 && p.x < 1200 && p.y >   0 && p.y < 250)
#define TM_1          (p.x >  0 && p.x <  950 && p.y > 750 && p.y < 820)
#define TM_2          (p.x >  0 && p.x <  950 && p.y > 665 && p.y < 735)
#define TM_3          (p.x >  0 && p.x <  950 && p.y > 580 && p.y < 650)
#define TM_4          (p.x >  0 && p.x <  950 && p.y > 490 && p.y < 560)
#define T_On_Off      (p.x > 50 && p.x <  900 && p.y > 300 && p.y < 430)

#define naikkan_suhu digitalWrite(Pemanas, nyala),
digitalWrite(Pompa, mati);
#define suhu_normal   digitalWrite(Pemanas, mati),
digitalWrite(Pompa,   mati);
#define turunkan_suhu digitalWrite(Pemanas, mati),
digitalWrite(Pompa,   mati);
#define naikkan_kelembaban  digitalWrite(humid,   nyala),
digitalWrite(kipas,   mati);
#define kelembaban_normal   digitalWrite(humid,   mati),
digitalWrite(kipas,   mati);
#define turunkan_kelembaban digitalWrite(humid,   mati),
digitalWrite(kipas,   nyala);
```

```
#define Pemanas_Mati  digitalWrite(Pemanas, mati);
#define Pemanas_Nyala digitalWrite(Pemanas, nyala);
#define Pompa_Mati    digitalWrite(Pompa,   mati);
#define Pompa_Nyala   digitalWrite(Pompa,   nyala);
#define humid_Mati    digitalWrite(humid,   mati);
#define humid_Nyala   digitalWrite(humid,   nyala);
#define kipas_Mati    digitalWrite(kipas,   mati);
#define kipas_Nyala   digitalWrite(kipas,   nyala);
#define L1_Off        digitalWrite(Lampu1,  mati);
#define L1_On         digitalWrite(Lampu1,  nyala);
#define L2_Off        digitalWrite(Lampu2,  mati);
#define L2_On         digitalWrite(Lampu2,  nyala);
#define L3_Off        digitalWrite(Lampu3,  mati);
#define L3_On         digitalWrite(Lampu3,  nyala);
#define L4_Off        digitalWrite(Lampu4,  mati);
#define L4_On         digitalWrite(Lampu4,  nyala);

#define Suhu_Rendah   Suhu          <= 25
#define Suhu_Normal   Suhu          > 25 & Suhu <= 30
#define Suhu_Tinggi   Suhu          > 30
#define H_Rendah      Kelembaban    <= 75
#define H_Normal      Kelembaban    > 75 & Kelembaban <= 80
#define H_Tinggi      Kelembaban    > 80

bool OnOff = false;

void setup()
{
  //------------------pin 2 s/d pin 13 sebagai dig output
  for(char out=A8;out<=A15;out++){    pinMode(out, OUTPUT);  }
  //------------------Matikan semua LED
  for(char out=A8;out<=A15;out++){    digitalWrite(out, mati);  }

  tft.reset       ();
  tft.begin       (0x9341);
  Serial.begin    (9600);
  tft.fillScreen  (BLACK);
  tft.setRotation (2);
  tft.setTextSize (1);
  tft.setTextColor(RED);
  tft.setCursor   (30,170);
  tft.print       ("setup BME:");

  if (!bme.begin(0x76))
  {
    Serial.println("Could not find a valid BME280 sensor, check wiring!");
```

```
  while (1);
 }
  for (int i; i < 200; i++)
 {
  tft.fillRect(30 - 10, 180, i, 10, RED);
  delay(0.00000000000000000000000000000000000000000000000001);
 }
 tft.setTextSize (3);
 tft.setTextColor(RED);
 tft.setCursor   (30, 200);
 tft.print       ("BME OK:");

 tft.setTextSize (1);
 tft.setTextColor(RED);
 tft.setCursor   (30, 240);
 tft.print       ("setup ETHERNET:");


 for (int i; i < 100; i++)
 {
  tft.fillRect(30 - 10, 250, i, 10, GREEN);
  delay(0.00000000000000000000000000000000000000000000000001);
 }
 Ethernet.init(10);  // Most Arduino Ethernet hardware
  while (!Serial)
 {
}

 // start the Ethernet connection:
 Serial.println("Initialize Ethernet with DHCP:");
 if (Ethernet.begin(mac) == 0)
 {
  Serial.println("Failed to configure Ethernet using DHCP");
  // Check for Ethernet hardware present
  if (Ethernet.hardwareStatus() == EthernetNoHardware)
  {
    Serial.println("Ethernet shield was not found.  Sorry, can't run without
hardware. :(");
    while (true)
    {
      delay(1); // do nothing, no point running without Ethernet hardware
    }
  }
  if (Ethernet.linkStatus() == LinkOFF)
  {
    Serial.println("Ethernet cable is not connected.");
```

```
    }
    Ethernet.begin(mac);    // try to congifure using IP address instead of DHCP:
  }

  else
  {
    Serial.print("  DHCP assigned IP ");
    Serial.println(Ethernet.localIP());
  }
  // give the Ethernet shield a second to initialize:
  delay(1000);

  ThingSpeak.begin(client);

  for (int i; i < 100; i++)
  {
    tft.fillRect(125 - 10, 250, i, 10, GREEN);
    delay(0.00000000000000000000000000000000000000000000000001);
  }

  tft.setTextSize (3);
  tft.setTextColor(RED);
  tft.setCursor   (30, 270);
  tft.print       ("ETHERNET OK:");
  HomeHome();
}

void loop()
{
  Sentuhan();

  unsigned long currentMillisSENSOR = millis();
  if (currentMillisSENSOR - previous_Millis_SENSOR >= interval_SENSOR)
  {
    previous_Millis_SENSOR = currentMillisSENSOR;
    ReadSensors();
  }

  unsigned long currentMillisAuto = millis();
  if (currentMillisAuto - previous_Millis_Auto >= interval_Auto)
  {
    previous_Millis_Auto = currentMillisAuto;
    KontrolAuto();
  }
```

```
  unsigned long currentMillisEthernet = millis();
  if (currentMillisEthernet - previous_Millis_thingspeak >= interval_thingspeak)
  {
    previous_Millis_thingspeak = currentMillisEthernet;
    UpdateThingSpeak();
  }
}
void ReadSensors()
{
  Suhu        = bme.readTemperature();
  Kelembaban  = bme.readHumidity();
  if (halaman <= 1)
  {
    tft.setTextColor(YELLOW,BLACK);
    tft.setTextSize (2);
    tft.setCursor   (20,205);
    tft.print       ("H");
    tft.setCursor   (20,260.5);
    tft.print       ("T");
    tft.setTextColor(WHITE,BLACK);
    tft.setTextSize (4);
    tft.setCursor   (70,200.5);
    tft.print       (Kelembaban,2);
    tft.print       ('%');
    tft.setCursor   (70,245.5);
    tft.print       (Suhu,2);
    tft.print       ('C');
  }

  if (halaman >= 2)
  {
    tft.setTextColor(CYAN,BLACK);
    tft.setTextSize (2);
    tft.setCursor   (60,15);
    tft.print       ("KELEMBABAN");
    tft.setCursor   (100,82);
    tft.print       ("SUHU");
    tft.setTextColor(GOLD,BLACK);
    tft.setTextSize (4);
    tft.setCursor   (48,40.5);
    tft.print       (Kelembaban,2);
    tft.print       ('%');
    tft.setCursor   (48,105.5);
    tft.print       (Suhu,2);
    tft.print       ('C');
  }
```

```
 if (halaman >= 2)
 {
  tft.setTextColor(RED,BLACK);
  tft.setTextSize (2);
  tft.setCursor   (60,15);
  tft.print       ("KELEMBABAN");
  tft.setCursor   (100,82);
  tft.print       ("SUHU");

  tft.setTextColor(GREEN, BLACK);
  tft.setTextSize (4);
  tft.setCursor   (48,40.5);
  tft.print       (Kelembaban,2);
  tft.print       ('%');
  tft.setCursor   (48,105.5);
  tft.print       (Suhu,2);
  tft.print       ('C');
 }
}

void HomeHome          ()
{
 halaman = 2;
 tft.fillScreen  (BLACK);
 tft.fillRect    (160,290,80,30,BLUE);
 tft.fillRect    ( 80,290,80,30,RED);
 tft.fillRect    (  0,290,80,30,YELLOW);
 tft.setTextColor(DARKBLUE,YELLOW);
 tft.setTextSize (2);
 tft.setCursor   (16,298);
 tft.print       ("MENU");
 tft.setTextColor(WHITE,RED);
 tft.setCursor   (96,298);
 tft.print       ("HOME");
 tft.setTextColor(YELLOW,BLUE);
 tft.setCursor   (170,298);
 tft.print       ("LAMPU");
 tft.drawLine    (5, 33,234, 33,RED);
 tft.drawLine    (5, 75,234, 75,GREEN);
 tft.drawLine    (5,100,234,100,YELLOW);
 tft.drawRect    (5,  5,230,135,BLUE);
 }
```

```
void M_Kontrol()
{
 Pemanas_Mati
 Pompa_Mati
 humid_Mati
 kipas_Mati
 tft.fillRect     ( 0,  0,240,290,BLACK);
 tft.drawLine      ( 5,235,234,235,YELLOW);
 tft.drawLine     (50,190, 50,285,RED);
 tft.drawRect     ( 5,190,230, 95,BLUE);
 tft.fillRect     ( 0,  0,240, 43,RED);
 tft.fillRect     ( 0, 45,240, 33,CYAN);
 tft.fillRect     ( 0, 80,240, 33,CYAN);
 tft.fillRect     ( 0,115,240, 33,CYAN);
 tft.fillRect     ( 0,150,240, 33,CYAN);

 if (halaman ==0)
 {
  tft.setTextColor(YELLOW,RED);
  tft.setTextSize (2);
  tft.setCursor   (84,13.5);
  tft.print       ("MANUAL");
  tft.setTextColor(BLUE,CYAN);
  tft.setCursor   (2,53.5);
  tft.print       ("Pemanas");
  tft.setCursor   (2,88.5);
  tft.print       ("Pompa");
  tft.setCursor   (2,123.5);
  tft.print       ("humid");
  tft.setCursor   (2,158.5);
  tft.print       ("Kipas");
 }

 if (halaman ==1)
 {
  tft.setTextColor(YELLOW,RED);
  tft.setTextSize (2);
  tft.setCursor   (60,13.5);
  tft.print       ("Penerangan");
  tft.setTextColor(BLUE,CYAN);
  tft.setCursor   (2,53.5);
  tft.print       ("Penerangan 1");
  tft.setCursor   (2,88.5);
  tft.print       ("Penerangan 2");
  tft.setCursor   (2,123.5);
  tft.print       ("Penerangan 3");
```

```
  tft.setCursor   (2,158.5);
  tft.print       ("Penerangan 4");
 }
}

void Kontrol_Manual ()
{
 tft.fillRect     ( 0,  0,240,290,BLACK);
 tft.fillRect     ( 0,150,240, 43,RED);
 tft.fillRect     (10,210,220, 60,YELLOW);
 OnOff = true;
 tft.drawLine     ( 5, 33,234, 33,RED);
 tft.drawLine     ( 5, 75,234, 75,GREEN);
 tft.drawLine     ( 5,100,234,100,YELLOW);
 tft.drawRect     ( 5,  5,230,135,BLUE);
 tft.setTextSize   (3);
 tft.setCursor    (84,229);
 tft.setTextColor  (BLUE,YELLOW);
 tft.print        ("MATI");
 tft.setTextColor  (YELLOW,RED);
 tft.setTextSize   (2);


 if  (halaman == 3)
 {
  tft.setCursor   (30,163.5);
  tft.print       ("Kontrol Pemanas");
 }

 if  (halaman == 4)
 {
  tft.setCursor   (42,163.5);
  tft.print       ("Kontrol Pompa");
 }

 if  (halaman == 5)
 {
  tft.setCursor   (42,163.5);
  tft.print       ("Kontrol humid");
 }

 if  (halaman == 6)
 {
  tft.setCursor   (42,163.5);
  tft.print       ("Kontrol Kipas");
 }
```

```
 if  (halaman == 7)
 {
  tft.setCursor   (38,163.5);
  tft.print       ("Penerangan 1");
 }

 if  (halaman == 8)
 {
  tft.setCursor   (48,163.5);
  tft.print       ("Penerangan 2");
 }

 if  (halaman == 9)
 {
  tft.setCursor   (48,163.5);
  tft.print       ("Penerangan 3");
 }

 if  (halaman == 10)
 {
  tft.setCursor   (48,163.5);
  tft.print       ("Penerangan 4");
 }
}

void VT_OnOff()
{
 if (OnOff == true)
 {
 tft.fillRect      (10,210,220,60,YELLOW);
 tft.setTextSize   (3);
 tft.setCursor     (84,229);
 tft.setTextColor  (BLUE,YELLOW);
 tft.print         ("MATI");
 }

 if (OnOff == false)
 {
 tft.fillRect      (10,210,220,60,GREEN);
 tft.setTextSize   (3);
 tft.setCursor     (75,229);
 tft.setTextColor  (RED,GREEN);
 tft.print         ("NYALA");
 }
}
```

```
void KontrolAuto()
{
  if (halaman == 2)
  {
    if (H_Rendah)
    {
      naikkan_kelembaban
      tft.fillRect    (125,240,44,40,GREEN);
      tft.fillRect    (179,240,44,40,RED);
      tft.setTextSize (1);
      tft.setTextColor(RED,GREEN);
      tft.setCursor   (132,250);
      tft.print       ("HUMID");
      tft.setCursor   (141,265);
      tft.print       ("ON");
      tft.setTextColor(WHITE,RED);
      tft.setCursor   (186,250);
      tft.print       ("KIPAS");
      tft.setCursor   (193,265);
      tft.print       ("OFF");
    }
    if (H_Normal)
    {
      kelembaban_normal
      tft.fillRect    (179,240,44,40,RED);
      tft.fillRect    (125,240,44,40,RED);
      tft.setTextSize (1);
      tft.setTextColor(WHITE,RED);
      tft.setCursor   (186,250);
      tft.print       ("KIPAS");
      tft.setCursor   (193,265);
      tft.print       ("OFF");
      tft.setCursor   (132,250);
      tft.print       ("HUMID");
      tft.setCursor   (138,265);
      tft.print       ("OFF");
    }

    if (H_Tinggi)
    {
      turunkan_kelembaban
      tft.setTextSize (1);
      tft.setTextColor(RED,GREEN);
      tft.fillRect    (179,240,44,40,GREEN);
      tft.fillRect    (125,240,44,40,RED);
      tft.setCursor   (186,250);
```

```
  tft.print      ("KIPAS");
  tft.setCursor   (196,265);
  tft.print      ("ON");
  tft.setTextColor(WHITE,RED);
  tft.setCursor   (132,250);
  tft.print      ("HUMID");
  tft.setCursor   (138,265);
  tft.print      ("OFF");
}

if (Suhu_Rendah)
{
  naikkan_suhu
  tft.fillRect    (17,240,44,40,GREEN);
  tft.fillRect    (71,240,44,40,RED);
  tft.setTextSize (1);
  tft.setTextColor(RED,GREEN);
  tft.setCursor   (18,250);
  tft.print      ("PEMANAS");
  tft.setCursor   (33,265);
  tft.print      ("ON");
  tft.setTextColor(WHITE,RED);
  tft.setCursor   (78,250);
  tft.print      ("POMPA");
  tft.setCursor   (84,265);
  tft.print      ("OFF");
}

if (Suhu_Normal)
{
  suhu_normal
  tft.fillRect    (17,240,44,40,RED);
  tft.fillRect    (71,240,44,40,RED);
  tft.setTextSize (1);
  tft.setTextColor(WHITE,RED);
  tft.setCursor   (18,250);
  tft.print      ("PEMANAS");
  tft.setCursor   (30,265);
  tft.print      ("OFF");
  tft.setCursor   (78,250);
  tft.print      ("POMPA");
  tft.setCursor   (84,265);
  tft.print      ("OFF");
}
```

```
    if (Suhu_Tinggi)
    {
      turunkan_suhu
      tft.fillRect    (71,240,44,40,GREEN);
      tft.fillRect    (17,240,44,40,RED);
      tft.setTextSize (1);
      tft.setTextColor(RED,GREEN);
      tft.setCursor   (78,250);
      tft.print       ("POMPA");
      tft.setCursor   (87,265);
      tft.print       ("ON");
      tft.setTextColor(WHITE,RED);
      tft.setCursor   (18,250);
      tft.print       ("PEMANAS");
      tft.setCursor   (30,265);
      tft.print       ("OFF");
    }
  }
}

void Sentuhan()
{
  TSPoint p = ts.getPoint();
  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);

  if  Tekanan
  {
    if (halaman != 0)
      {if  TM_manual {halaman = 0;M_Kontrol();}}

    if (halaman != 2)
      {if  T_Home    {          HomeHome();}}

    if (halaman != 1)
      {if  TM_lampu  {halaman = 1;M_Kontrol();}}

    if (halaman == 0)
    {
      if      TM_1   {halaman = 3;Kontrol_Manual();}
      else if TM_2   {halaman = 4;Kontrol_Manual();}
      else if TM_3   {halaman = 5;Kontrol_Manual();}
      else if TM_4   {halaman = 6;Kontrol_Manual();}
    }
```

```
  if  (halaman == 3)
  {
   if  (OnOff == true) {if  T_On_Off {OnOff = false;  Pemanas_Nyala
VT_OnOff();}}
   else  { OnOff = true;   Pemanas_Mati  VT_OnOff(); }
  }
  if  (halaman == 4)
  {
   if  (OnOff == true) {if  T_On_Off {OnOff =
false;  Pompa_Nyala   VT_OnOff();}}
   else { OnOff = true;   Pompa_Mati    VT_OnOff(); }
  }

  if  (halaman == 5)
  {
   if  (OnOff == true) {if  T_On_Off {OnOff =
false;  humid_Nyala   VT_OnOff();}}
   else { OnOff = true;   humid_Mati    VT_OnOff(); }
  }

  if  (halaman == 6)
  {
   if  (OnOff == true) {if  T_On_Off {OnOff =
false;  kipas_Nyala   VT_OnOff();}}
   else { OnOff = true;   kipas_Mati    VT_OnOff(); }
  }

  if  (halaman == 1)
  {
   if     TM_1   {halaman =  7;Kontrol_Manual();}
   else if TM_2   {halaman =  8;Kontrol_Manual();}
   else if TM_3   {halaman =  9;Kontrol_Manual();}
   else if TM_4   {halaman = 10;Kontrol_Manual();}
  }

  if  (halaman == 7)
  {
   if  (OnOff == true) {if  T_On_Off {OnOff = false;L1_On  VT_OnOff();}}
   else { OnOff = true; L1_Off VT_OnOff(); }
  }

  if  (halaman == 8)
  {
   if  (OnOff == true) {if  T_On_Off {OnOff = false;L2_On  VT_OnOff();}}
   else { OnOff = true; L2_Off VT_OnOff(); }
  }
```

```
   if  (halaman == 9)
   {
    if  (OnOff == true) {if  T_On_Off {OnOff = false;L3_On  VT_OnOff();}}
    else { OnOff = true; L3_Off VT_OnOff(); }
   }

   if  (halaman == 10)
   {
    if  (OnOff == true) {if  T_On_Off {OnOff = false;  L4_On  VT_OnOff();}}
    else { OnOff = true;   L4_Off VT_OnOff(); }
   }
  }
}

void UpdateThingSpeak ()
{
 Serial.println       (Kelembaban);
 Serial.println       (Suhu);
 ThingSpeak.setField   (1, String (Kelembaban));
 ThingSpeak.setField   (2, String (Suhu));
 ThingSpeak.writeFields(Nomor_Chanel, Write_API_Key);
}
```