

DAFTAR PUSTAKA

Banzi Massimo. 2011. *Getting Started With Arduino*. Second Edition, USA: O'Reilly Media, Inc.

Beckingham Colin. 2011. *Data logging with open source hardware and software in the energy sector (Solar panels, Arduino, PHP, MySQL, and Flotr)*. IBM Corporation.

Betha Sidik, IR. 2004. *Pemrograman WEB dengan PHP*. Cetakan Keempat, Bandung : Penerbit Informatika.

Craft Brock. 2013. *Arduino Project For Dummies*. First Edition, England: John Wiley & Sons, Ltd.

G. Smith Alan. 2011. *Introduction to Arduino*. www.introtoarduino.com.

https://id.wikipedia.org/wiki/Bagian_per_juta. Tanggal akses: 30 Maret 2016.

<https://m.tempo.co/read/news/2014/04/19/060571781/lima-zat-pencemaran-udara-dampak-bagi-kesehatan>. Tanggal akses: 23 Maret 2016.

https://sumut.bps.go.id/backend/pdf_publicasi/Sumatera-Utara-Dalam-Angka-2015.pdf. Tanggal akses: 5 April 2016.

Republik Indonesia, 1997, *Undang-Undang Nomor 23 Mengenai Pencemaran Udara*, Jakarta: Sekretariat Jakarta

Republik Indonesia, 1999, *Undang-Undang Nomor 41 Mengenai Pengendalian Pencemaran Udara*, Jakarta: Sekretariat Jakarta.

Republik Indonesia, 2002, *Undang-Undang Nomor 1407 Mengenai Pedoman Pengendalian Dampak Pencemaran Udara*, Jakarta: Sekretariat Jakarta.

Sampurna. 1996. *World Wide Web*. Jakarta: Andi.

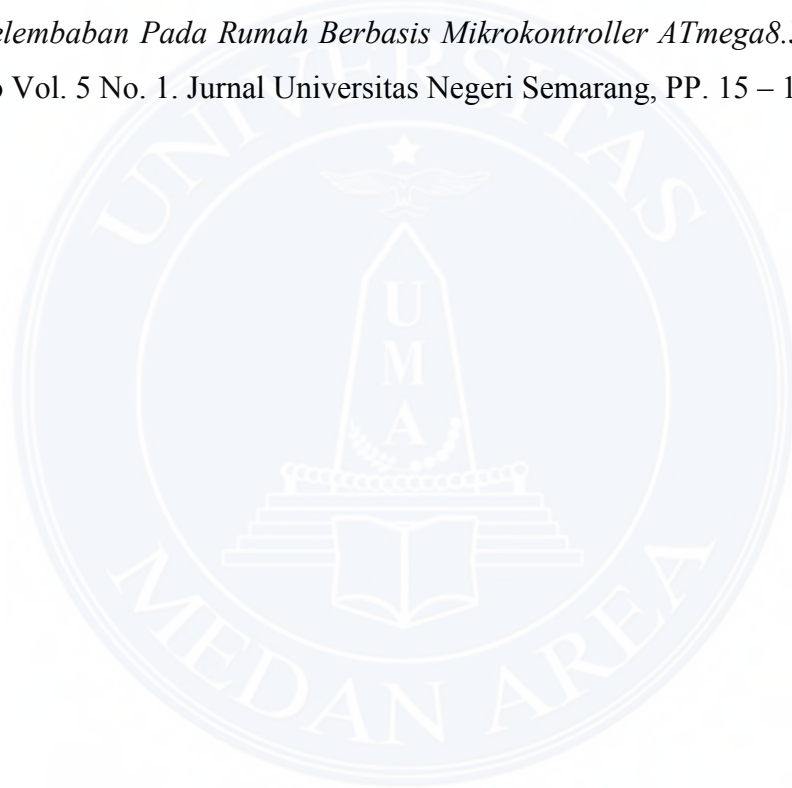
Sugianto Azis. 2015. *Purwarupa sistem pemantauan kualitas udara secara daring*. Jurnal Universitas Widyatama, PP. 3 – 4.

Ubaidillah Maulana. 2015. *Alat Ukur Kualitas Udara Menggunakan Sensor Gas MQ-135 Berbasis Mikrokontroler ATmega16A*. Jurnal Universitas Sumatera Utara, PP. 5 – 7.

Wardhana,W.A. 2004. *Dampak Pencemaran Lingkungan*. Cetakan Keempat. Yogyakarta : Penerbit Andi.

Yani, S. A. 2009. *Suhu Udara*. Badan Meteorologi Klimatologi dan Geofisika. Jawa Tengah

Yan Eka Aditya Muhammad, Wibawanto Hari. 2013. *Sistem Pengamatan Suhu dan Kelembaban Pada Rumah Berbasis Mikrokontroller ATmega8*.Jurnal Teknik Elektro Vol. 5 No. 1. Jurnal Universitas Negeri Semarang, PP. 15 – 16.



LAMPIRAN

Lampiran I Data Sheet Sensor DHT11 :



For more products visit our website <http://www.sunrom.com>

Document: Datasheet
Date: 20-Jun-12
Model #: 3732
Product's Page: www.sunrom.com/p-1141.html

DHT11 - Humidity and Temperature Sensor

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds.

Features


- Full range temperature compensated
- Relative humidity and temperature measurement
- Calibrated digital signal
- Outstanding long-term stability
- Extra components not needed
- Long transmission distance
- Low power consumption
- 4 pins packaged and fully interchangeable



Details

This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness. Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process.

The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package.









Lampiran V Kode Program Stasiun Pemantau Kualitas Udara:

```

#include <DHT.h>
#include <Ethernet.h>
#include <SPI.h>
#include <MQ135.h>
#define DHTPIN 7 // what digital pin we're connected to
const int mq135Pin = 0;
const int AOUPin=1;//the AOUP pin of the CO sensor goes into analog pin A0
of the arduino
// Uncomment whatever type you're using!
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE);
MQ135 gasSensor = MQ135(mq135Pin);
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
EthernetClient client;
IPAddress ip(192, 168, 1, 10);
byte server [] = {192,168,1,5};
long previousMillis = 0;
unsigned long currentMillis = 0;
long interval = 250000;
float h = 0;
float t = 0;
float ppm = 0;
float finalPPMkdo = 0;
float co2 = 0;
int limit;
float value;
float ppmCO;
String data;
void setup() {
  Serial.begin(9600);
  Serial.println("DHTxx test!");
  dht.begin();
  //start the Ethernet Connection
  Ethernet.begin(mac, ip);

  delay (10000);

  h = (float) dht.readHumidity();
  t = (float) dht.readTemperature();

  float rzero = gasSensor.getRZero();
  Serial.print("R0: ");
  Serial.println(rzero); // Valeur Å reporter ligne 27 du fichier mq135.h aprÃs
48h de prÃchauffage
  float ppm = gasSensor.getPPM();

```



```

float finalPPMkdo = ppm*100;
Serial.print("A0: ");
Serial.print(analogRead(mq135Pin));
Serial.print(" ppm CO2: ");
Serial.println(finalPPMkdo);

  data = "";
}

void loop() {

  //DHT11 membaca suhu dan kelembaban
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  h = (float) dht.readHumidity();
  t = (float) dht.readTemperature();

  //MQ135 membaca nilai CO2
  float rzero = gasSensor.getRZero();
  float ppm = gasSensor.getPPM();
  float finalPPMkdo = ppm*100;

  //MQ7 membaca nilai CO
  ppmCO = value / 1000;
  value= analogRead(AOUTpin);//reads the analaog value from the CO sensor's
  AOUT pin

  data = "temperature=" + String(t) + "&humidity=" + String(h) +
  "&finalPPMkdo=" + String(finalPPMkdo);

  if (client.connect(server,80)) {
  client.print("GET /udara/write_data.php?");
  client.print("temperature=");
  client.print(t);
  client.print("&");
  client.print("humidity=");
  client.print(h);
  client.print("&");
  client.print("finalPPMkdo=");
  client.print(finalPPMkdo);
  client.print("&");
  client.print("finalPPMCO=");

```

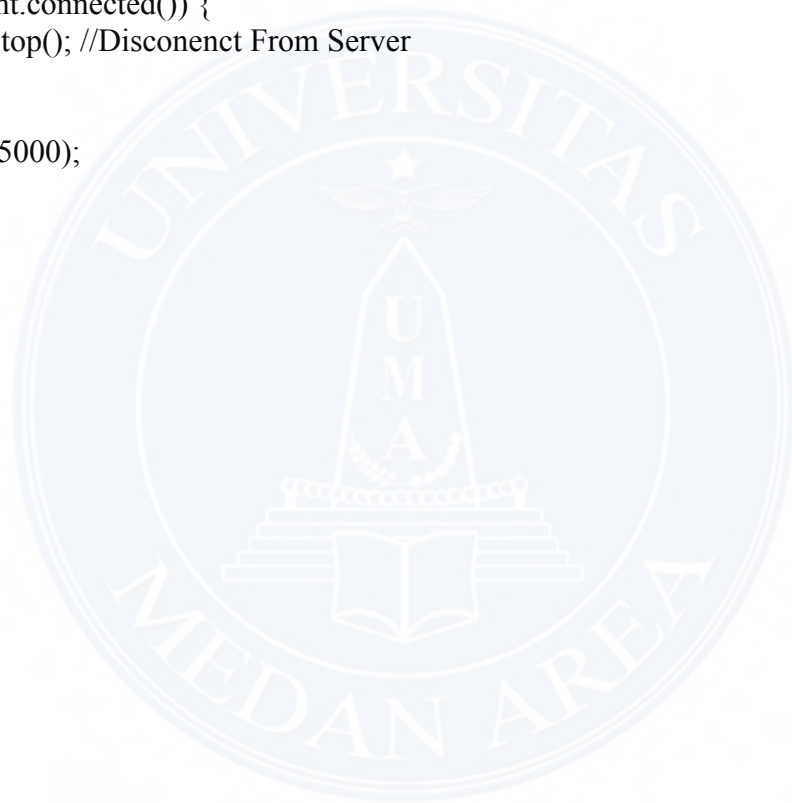


```
client.println(ppmCO);
client.println("HTTP/1.1");
client.println("Host: server");
client.println("Content-Type: application/x-www-form-urlencoded");
client.println(data.length());
client.println();
//Serial.println(data);
Serial.println(t);
Serial.println(h);
Serial.println(finalPPMkdo);
Serial.println(ppmCO);
}

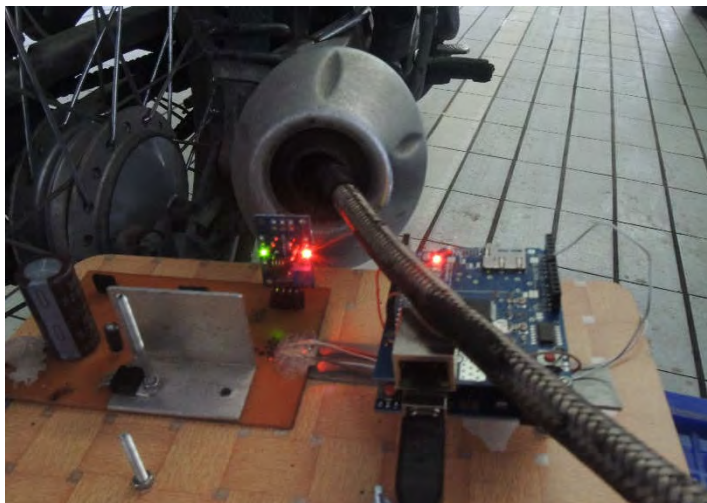
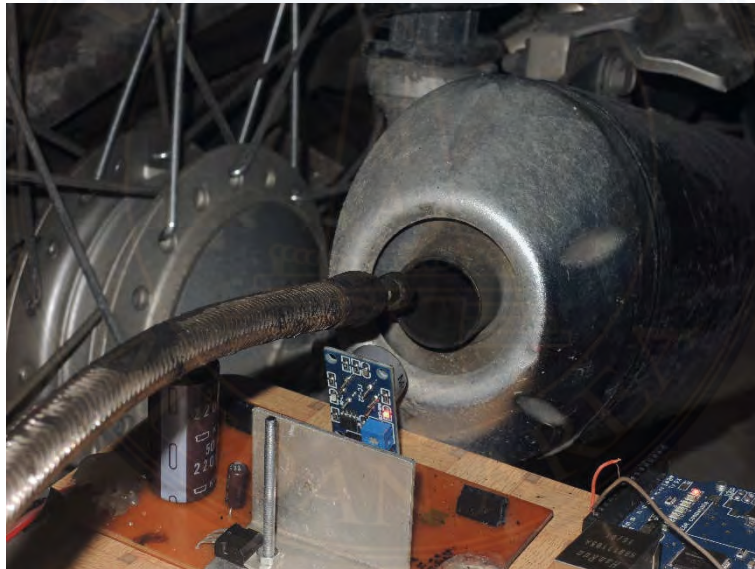
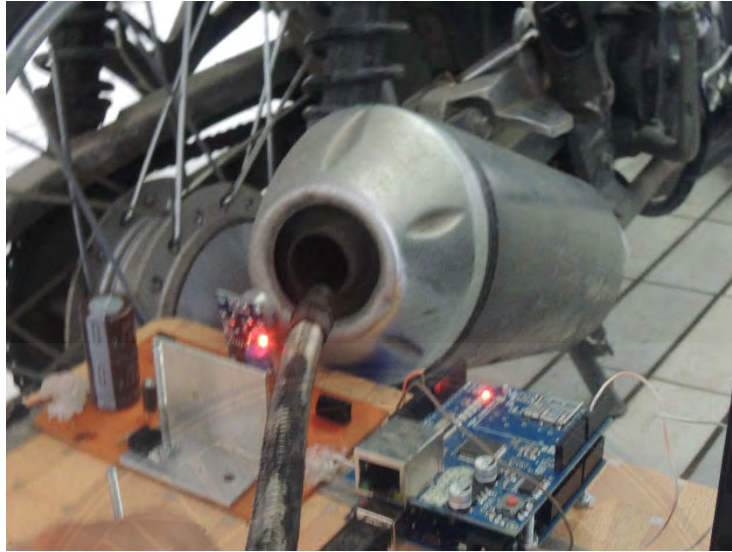
if (client.connected()) {
client.stop(); //Disconenct From Server
}

delay (5000);

}
```



Lampiran VI Dokumentasi Pengambilan Data:



Lampiran VII Dokumentasi Alat :